



Especificación HTML 4.01 (HTML 4.01 Specification)

Recomendación del W3C 24 de diciembre de 1999

Importante:

Este documento es una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification", publicada el 24 de diciembre de 1999.

Esta traducción ha sido publicada el 1 de junio de 2001.

El URI de este documento es <http://html.conclase.net/w3c/html401-es/cover.html>

El URI del documento original es <http://www.w3.org/TR/1999/REC-html401-19991224>

El URI de la última versión de la Recomendación en inglés es <http://www.w3.org/TR/html401>

La **única** versión normativa de este documento es la [versión original en inglés](#) que se encuentra en el sitio web del W3C. Ninguna parte del presente documento en castellano es normativa aunque se especifique lo contrario.

Véase el [Aviso de copyright](#) al final de este documento.

Este documento puede contener errores de traducción.

La traducción ha sido realizada por [Juan R. Pozo](#), <jrpozo@conclase.net>

Gracias a todos aquellos que han colaborado en la tarea.

Esta versión (en castellano):

<http://html.conclase.net/w3c/html401-es/cover.html>

([texto sin formato \[880Kb\]](#), [archivo zip de ficheros HTML comprimidos \[428Kb\]](#), [fichero PDF comprimido con winzip \[2346Kb\]](#))

Esta versión (en inglés):

<http://www.w3.org/TR/1999/REC-html401-19991224>

([texto sin formato \[794Kb\]](#), [archivo tar comprimido con gzip de ficheros HTML \[371Kb\]](#), [archivo zip de ficheros HTML comprimidos \[405Kb\]](#), [fichero Postscript comprimido con gzip \[746Kb, 389 pages\]](#), [fichero PDF comprimido con gzip \[963Kb\]](#))

Última versión de HTML 4.01 (en inglés):

<http://www.w3.org/TR/html401>

Última versión de HTML 4 (en inglés):

<http://www.w3.org/TR/html4>

Última versión de HTML (en inglés):

<http://www.w3.org/TR/html>

Versión anterior de HTML 4.01 (en inglés):

<http://www.w3.org/TR/1999/PR-html40-19990824>

Recomendación previa HTML 4 (en inglés):

<http://www.w3.org/TR/1998/REC-html40-19980424>

Editores:

[Dave Raggett](mailto:dsr@w3.org) <dsr@w3.org>

Arnaud Le Hors, W3C

Ian Jacobs, W3C

Copyright ©1997-1999 **W3C**[®] ([MIT](#), [INRIA](#), [Keio](#)), todos los derechos reservados. Se aplican todas las condiciones de [responsabilidad](#), [marcas registradas](#), [uso de documentos](#) y [licencias de software](#) del W3C.

Resumen

Esta especificación define el Lenguaje de Formato de Documentos para Hipertexto (*HyperText Markup Language, HTML*), el lenguaje de publicación de la World Wide Web. Esta especificación define HTML 4.01, que es una versión de HTML 4. Además de las características relativas a texto, multimedia e hipervínculos de las versiones anteriores de HTML (HTML 3.2 [\[HTML32\]](#) y HTML 2.0 [\[RFC1866\]](#)), HTML 4 soporta más opciones de multimedia, lenguajes de scripts, hojas de estilo, mejores capacidades de impresión, y documentos más accesibles a usuarios con discapacidades. HTML 4 también da un gran paso adelante hacia la internacionalización de los documentos, con la intención de hacer la Web auténticamente universal.

HTML4 es una aplicación de SGML conforme al estándar internacional ISO 8879 -- Standard Generalized Markup Language [\[ISO8879\]](#).

Estado de este documento

Esta sección describe el estado de este documento en la fecha de su publicación. Otros documentos pueden reemplazar a este documento. En el W3C se conserva el estado más reciente de esta serie de documentos.

Este documento especifica HTML 4.01, que es parte de la línea de especificaciones de HTML 4. La primera versión de HTML 4 fue HTML 4.0 [\[HTML40\]](#), publicado el 18 de diciembre de 1997 y revisado el 24 de abril de 1998. Esta especificación es la primera Recomendación HTML 4.01. Incluye [cambios no editoriales desde la versión del 24 de abril de HTML 4.0](#). Por ejemplo, se han hecho algunos cambios en los DTDs. Con este documento quedan obsoletas las versiones previas de

HTML 4.0, aunque el W3C seguirá poniendo a disposición del público dichas especificaciones y sus DTDs en su sitio web.

Este documento ha sido revisado por Miembros del W3C y otras partes interesadas y ha sido aprobado por el Director como Recomendación del W3C. Es un documento estable y puede ser usado como material de referencia o citado como referencia normativa en otros documentos. La intención del W3C al hacer esta Recomendación es llamar la atención sobre la especificación y promover su difusión general.

El W3C recomienda que los agentes de usuario y los autores (y en particular las herramientas de creación) produzcan documentos HTML 4.01 en vez de documentos HTML 4.0. El W3C recomienda que los autores produzcan documentos HTML 4 en vez de documentos HTML 3.2. Por razones de compatibilidad con versiones anteriores, el W3C también recomienda que las herramientas que interpreten HTML 4 sigan dando soporte a HTML 3.2 y a HTML 2.0.

Para obtener información sobre la próxima generación de HTML, "El Lenguaje Extensible para el Formato de Documentos de Hipertexto" [XHTML], consulte las [Actividades del W3C en el campo del HTML](#) y la lista de [Informes Técnicos del W3C](#).

Este documento ha sido producido como parte de las [Actividades del W3C en el campo del HTML](#). Los objetivos del [Grupo de Trabajo HTML \(sólo Miembros\)](#) se plantean en la [carta constitucional del Grupo de Trabajo HTML \(sólo Miembros\)](#).

Puede encontrarse una lista de las Recomendaciones actuales del W3C y otros documentos técnicos en <http://www.w3.org/TR>.

En la lista de correo www-html@w3.org se discuten públicamente las características del HTML ([archivos de www-html@w3.org](#)).

Idiomas disponibles

La versión en inglés de esta especificación es la única versión normativa. Sin embargo, para traducciones de este documento, véase <http://www.w3.org/MarkUp/html4-updates/translations>.

Erratas

La lista de errores conocidos de la versión original en inglés de esta especificación está disponible en: <http://www.w3.org/MarkUp/html4-updates/errata>

Le rogamos informe de errores en la versión original en inglés de este documento a www-html-editor@w3.org.

Nota: La corrección de estas erratas ya ha sido tomada en cuenta en la traducción al castellano del documento. Para informar sobre errores presentes en la traducción española de la especificación, puede ponerse en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Tabla abreviada de contenidos

1. [Sobre la especificación HTML 4](#)
2. [Introducción a HTML4](#)
3. [Sobre SGML y HTML](#)
4. [Conformidad: requisitos y recomendaciones](#)
5. [Representación de documentos HTML](#) - *Conjuntos de caracteres, codificaciones de caracteres y entidades*
6. [Tipos de datos básicos de HTML](#) - *Datos de caracteres, colores, longitudes, URIs, tipos de contenido, etc.*
7. [La estructura global de un documento HTML](#) - *La cabecera (HEAD) y el cuerpo (BODY) de un documento*
8. [Información sobre el idioma y la dirección del texto](#) - *Consideraciones internacionales sobre el texto*
9. [Texto](#) - *Párrafos, líneas y frases*
10. [Listas](#) - *Listas ordenadas, no ordenadas y de definiciones*
11. [Tablas](#)
12. [Vínculos](#) - *Hipertexto y vínculos independientes del medio*
13. [Objetos, Imágenes y Aplicaciones](#)
14. [Hojas de Estilo](#) - *Añadir estilo a los documentos HTML*
15. [Alineación, estilos de fuente y separadores horizontales](#)
16. [Marcos](#) - *Presentación multivista de documentos*
17. [Formularios](#) - *Formularios para entrada de datos por el usuario: campos de texto, botones, menús, etc.*
18. [Scripts](#) - *Documentos animados y formularios inteligentes*
19. [Información de referencia sobre SGML para HTML](#) - *Definición formal y validación de HTML*
20. [Declaración SGML de HTML 4](#)
21. [Definición del Tipo de Documento \(Document Type Definition\)](#)
22. [Definición del Tipo de Documento Transicional \(Transitional Document Type Definition\)](#)
23. [Definición del Tipo de Documento con Marcos \(Frameset Document Type Definition\)](#)
24. [Referencias a entidades de caracteres en HTML 4](#)
 - A. [Cambios](#)
 - B. [Aplicación, Implementación y Notas de Diseño](#)
 - [Referencias](#)
 - [Índice de Elementos](#)
 - [Índice de Atributos](#)
 - [Índice](#)

Tabla completa de contenidos

1. [Sobre la especificación HTML 4](#)
 1. [Cómo está organizada la especificación](#)
 2. [Convenciones del documento](#)
 1. [Elementos y atributos](#)
 2. [Notas y ejemplos](#)
 3. [Agradecimientos](#)
 1. [Agradecimientos por la revisión actual](#)
 4. [Aviso de Copyright](#)
2. [Introducción a HTML 4](#)
 1. [¿Qué es la World Wide Web?](#)
 1. [Introducción a los URIs](#)
 2. [Identificadores de fragmento](#)
 3. [URIs relativos](#)
 2. [¿Qué es el HTML?](#)
 1. [Breve historia del HTML](#)
 3. [HTML 4](#)
 1. [Internacionalización](#)
 2. [Accesibilidad](#)
 3. [Tablas](#)
 4. [Documentos compuestos](#)
 5. [Hojas de estilo](#)
 6. [Scripts](#)
 7. [Impresión](#)
 4. [Creación de documentos con HTML 4](#)
 1. [Separar estructura y presentación](#)
 2. [Considerar la accesibilidad universal a la Web](#)
 3. [Ayudar a los agentes de usuario con la representación incremental](#)
3. [Sobre SGML y HTML](#)
 1. [Introducción al SGML](#)
 2. [Estructuras SGML usadas en HTML](#)
 1. [Elementos](#)
 2. [Atributos](#)
 3. [Referencias de caracteres](#)

4. [Comentarios](#)
3. [Cómo leer el DTD de HTML](#)
 1. [Comentarios DTD](#)
 2. [Definiciones de entidades paramétricas](#)
 3. [Declaraciones de elementos](#)
 - [Definiciones del modelo de contenido](#)
 4. [Declaraciones de atributos](#)
 - [Entidades DTD en definiciones de atributos](#)
 - [Atributos booleanos](#)
4. [Conformidad: requisitos y recomendaciones](#)
 1. [Definiciones](#)
 2. [SGML](#)
 3. [El tipo de contenido text/html](#)
5. [Representación de Documentos HTML](#) - *Conjuntos de caracteres, codificaciones de caracteres y entidades*
 1. [El Conjunto de Caracteres del Documento](#)
 2. [Codificaciones de caracteres](#)
 1. [Elección de una codificación](#)
 - [Notas sobre codificaciones específicas](#)
 2. [Especificación de la codificación de caracteres](#)
 3. [Referencias de caracteres](#)
 1. [Referencias numéricas de caracteres](#)
 2. [Referencias a entidades de caracteres](#)
 4. [Caracteres no representables](#)
6. [Tipos de datos básicos de HTML](#) - *Datos de caracteres, colores, longitudes, URIs, tipos de contenido, etc.*
 1. [Mayúsculas y minúsculas](#)
 2. [Tipos básicos de SGML](#)
 3. [Cadenas de texto](#)
 4. [URIs](#)
 5. [Colores](#)
 1. [Notas sobre el uso de colores](#)
 6. [Longitudes](#)
 7. [Tipos de contenido \(tipos MIME\)](#)
 8. [Códigos de idioma](#)
 9. [Codificaciones de caracteres](#)

10. [Caracteres individuales](#)
 11. [Fechas y horas](#)
 12. [Tipos de vínculos](#)
 13. [Descriptores de medios](#)
 14. [Datos de scripts](#)
 15. [Datos de hojas de estilo](#)
 16. [Nombres de marcos destino](#)
7. [La estructura global de un documento HTML](#) - *La cabecera (HEAD) y el cuerpo (BODY) de un documento*
1. [Introducción a la estructura de un documento HTML](#)
 2. [Información sobre la versión de HTML](#)
 3. [El elemento HTML](#)
 4. [La cabecera del documento](#)
 1. [El elemento HEAD](#)
 2. [El elemento TITLE](#)
 3. [El atributo title](#)
 4. [Metadatos](#)
 - [Especificación de metadatos](#)
 - [El elemento META](#)
 - [Perfiles de metadatos](#)
 5. [El cuerpo del documento](#)
 1. [El elemento BODY](#)
 2. [Identificadores de elementos: los atributos id y class](#)
 3. [Elementos en bloque y elementos en línea](#)
 4. [Agrupación de elementos: los elementos DIV y SPAN](#)
 5. [Encabezados: los elementos H1, H2, H3, H4, H5, H6](#)
 6. [El elemento ADDRESS](#)
8. [Información sobre el idioma y la dirección del texto](#) - *Consideraciones internacionales sobre el texto*
1. [Especificación del idioma del contenido: el atributo lang](#)
 1. [Códigos de idioma](#)
 2. [Herencia de los códigos de idioma](#)
 3. [Interpretación de los códigos de idioma](#)
 2. [Especificación de la dirección del texto y de las tablas: el atributo dir](#)
 1. [Introducción al algoritmo bidireccional](#)
 2. [Herencia de la información sobre la dirección del texto](#)

3. [Especificación de la dirección del texto incluido](#)
4. [Anulación del algoritmo bidireccional: el elemento BDO](#)
5. [Referencias de caracteres para el control de la direccionalidad y de la unión](#)
6. [Efecto de las hojas de estilo en la bidireccionalidad](#)

9. [Texto](#) - *Párrafos, líneas y frases*

1. [Espacio en blanco](#)
2. [Texto estructurado](#)
 1. [Elementos de frase: EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, ABBR y ACRONYM](#)
 2. [Citas: Los elementos BLOCKQUOTE y Q](#)
 - [Representación de citas](#)
 3. [Subíndices y superíndices: los elementos SUB y SUP](#)
3. [Líneas y párrafos](#)
 1. [Párrafos: el elemento P](#)
 2. [Control de los saltos de línea](#)
 - [Forzar un salto de línea: el elemento BR](#)
 - [Impedir un salto de línea](#)
 3. [División de palabras](#)
 4. [Texto preformateado: el elemento PRE](#)
 5. [Representación visual de los párrafos](#)
4. [Anotación de cambios en los documentos: los elementos INS y DEL](#)

10. [Listas](#) - *Listas ordenadas, no ordenadas y de definiciones*

1. [Introducción a las listas](#)
2. [Listas no ordenadas \(UL\), listas ordenadas \(OL\), y objetos de lista \(LI\)](#)
3. [Listas de definiciones: los elementos DL, DT y DD](#)
 1. [Representación visual de las listas](#)
4. [Los elementos DIR y MENU](#)

11. [Tablas](#)

1. [Introducción a las tablas](#)
2. [Elementos para construir tablas](#)
 1. [El elemento TABLE](#)
 - [Direccionalidad de tablas](#)
 2. [Títulos de tablas: el elemento CAPTION](#)
 3. [Grupos de filas: los elementos THEAD, TFOOT y TBODY](#)
 4. [Grupos de columnas: los elementos COLGROUP y COL](#)
 - [El elemento COLGROUP](#)

- [El elemento COL](#)
- [Cálculo del número de columnas de una tabla](#)
- [Cálculo de la anchura de las columnas](#)
- 5. [Filas de una tabla: el elemento TR](#)
- 6. [Celdas de una tabla: los elementos TH y TD](#)
 - [Celdas que abarcan varias filas o columnas](#)
- 3. [Formato de tablas por agentes de usuario visuales](#)
 - 1. [Bordes y líneas de división](#)
 - 2. [Alineación horizontal y vertical](#)
 - [Herencia de las especificaciones de alineación](#)
 - 3. [Márgenes de las celdas](#)
- 4. [Representación de tablas por agentes de usuario no visuales](#)
 - 1. [Asociación de información de encabezado con celdas de datos](#)
 - 2. [Categorización de celdas](#)
 - 3. [Algoritmo para encontrar información de encabezado](#)
- 5. [Tabla de muestra](#)
- 12. [Vínculos](#) - *Hipertexto y vínculos independientes del medio*
 - 1. [Introducción a los vínculos](#)
 - 1. [Visitar un recurso vinculado](#)
 - 2. [Otras relaciones de vínculo](#)
 - 3. [Especificación de vínculos](#)
 - 4. [Títulos de vínculos](#)
 - 5. [Internacionalización y vínculos](#)
 - 2. [El elemento A](#)
 - 1. [Sintaxis de nombres de vínculos](#)
 - 2. [Los vínculos anidados son ilegales](#)
 - 3. [Vínculos con el atributo id](#)
 - 4. [Recursos no disponibles o no identificables](#)
 - 3. [Relaciones entre documentos: el elemento LINK](#)
 - 1. [Vínculos directos e inversos](#)
 - 2. [Vínculos y hojas de estilo externas](#)
 - 3. [Vínculos y motores de búsqueda](#)
 - 4. [Información sobre la ruta de acceso: el elemento BASE](#)
 - 1. [Cómo completar los URIs relativos](#)
- 13. [Objetos, Imágenes y Aplicaciones](#)

1. [Introducción a los objetos, imágenes y aplicaciones](#)
2. [Incluir una imagen: el elemento IMG](#)
3. [Inclusión genérica: el elemento OBJECT](#)
 1. [Reglas para representar objetos](#)
 2. [Inicialización de objetos: el elemento PARAM](#)
 3. [Esquemas globales de nombres para objetos](#)
 4. [Declaración y creación de un objeto](#)
4. [Incluir un applet: el elemento APPLET](#)
5. [Notas sobre la inclusión de documentos](#)
6. [Mapas de imágenes](#)
 1. [Mapas de imágenes en el lado del cliente: los elementos MAP y AREA](#)
 - [Ejemplos de mapas de imágenes en el lado del cliente](#)
 2. [Mapas de imágenes en el lado del servidor](#)
7. [Presentación visual de imágenes, objetos y aplicaciones](#)
 1. [Anchura y altura](#)
 2. [Espacio en blanco alrededor de imágenes y objetos](#)
 3. [Bordes](#)
 4. [Alineación](#)
8. [Cómo especificar texto alternativo](#)
14. [Hojas de estilo](#) - *Añadir estilo a los documentos HTML*
 1. [Introducción a las hojas de estilo](#)
 2. [Añadir estilos al HTML](#)
 1. [Especificación del lenguaje de hojas de estilo por defecto](#)
 2. [Información de estilo en línea](#)
 3. [Información de estilo en cabecera: el elemento STYLE](#)
 4. [Tipos de medios](#)
 3. [Hojas de estilo externas](#)
 1. [Hojas de estilo preferentes y alternativas](#)
 2. [Especificación de hojas de estilo externas](#)
 4. [Hojas de Estilo en Cascada](#)
 1. [Cascadas dependientes del medio](#)
 2. [Herencia y cascada](#)
 5. [Ocultar datos de estilo a los agentes de usuario](#)
 6. [Vinculación de hojas de estilo mediante encabezados HTTP](#)
15. [Alineación, estilos de fuente y separadores horizontales](#)

1. [Formato](#)
 1. [Color de fondo](#)
 2. [Alineación](#)
 3. [Objetos flotantes](#)
 - [Hacer flotar a un objeto](#)
 - [Texto flotante alrededor de un objeto](#)
 2. [Fuentes](#)
 1. [Elementos de estilo de fuente: los elementos TT, I, B, BIG, SMALL, STRIKE, S y U](#)
 2. [Elementos de modificación de fuentes: FONT y BASEFONT](#)
 3. [Separadores: el elemento HR](#)
16. [Marcos](#) - *Presentación multivista de documentos*
1. [Introducción a los marcos](#)
 2. [Disposición de los marcos](#)
 1. [El elemento FRAMESET](#)
 - [Filas y Columnas](#)
 - [Anidamiento de grupos de marcos](#)
 - [Compartir datos entre marcos](#)
 2. [El elemento FRAME](#)
 - [Especificación de los contenidos iniciales de un marco](#)
 - [Representación visual de un marco](#)
 3. [Especificación de información sobre el marco destino](#)
 1. [Especificación del destino de los vínculos por defecto](#)
 2. [Semántica de marcos destino](#)
 4. [Contenido alternativo](#)
 1. [El elemento NOFRAMES](#)
 2. [Descripciones largas de marcos](#)
 5. [Marcos en línea: el elemento IFRAME](#)
17. [Formularios](#) - *Formularios para entrada de datos por el usuario: campos de texto, botones, menús, etc.*
1. [Introducción a los formularios](#)
 2. [Controles](#)
 1. [Tipos de controles](#)
 3. [El elemento FORM](#)
 4. [El elemento INPUT](#)
 1. [Tipos de controles creados con INPUT](#)

2. [Ejemplos de formularios con controles INPUT](#)
5. [El elemento BUTTON](#)
6. [Los elementos SELECT, OPTGROUP y OPTION](#)
 1. [Opciones preseleccionadas](#)
7. [El elemento TEXTAREA](#)
8. [El elemento ISINDEX](#)
9. [Rótulos](#)
 1. [El elemento LABEL](#)
10. [Añadir estructura a los formularios: los elementos FIELDSET y LEGEND](#)
11. [Dirigir el foco hacia un elemento](#)
 1. [Navegación con tabulador](#)
 2. [Teclas de acceso](#)
12. [Controles deshabilitados y de sólo lectura](#)
 1. [Controles deshabilitados](#)
 2. [Controles de sólo lectura](#)
13. [Envío de formularios](#)
 1. [Método de envío del formulario](#)
 2. [Controles con éxito](#)
 3. [Procesamiento de los datos del formulario](#)
 - [Paso uno: Identificación de los controles con éxito](#)
 - [Paso dos: Construcción del conjunto de datos del formulario](#)
 - [Paso tres: Codificación del conjunto de datos del formulario](#)
 - [Paso cuatro: Envío del conjunto de datos del formulario codificado](#)
 4. [Tipos de contenido de formularios](#)
 - [application/x-www-form-urlencoded](#)
 - [multipart/form-data](#)
18. [Scripts](#) - *Documentos animados y formularios inteligentes*
 1. [Introducción a los scripts](#)
 2. [Diseño de documentos para agentes de usuario que soporten scripts](#)
 1. [El elemento SCRIPT](#)
 2. [Especificación del lenguaje de scripts](#)
 - [El lenguaje de scripts por defecto](#)
 - [Declaración local del lenguaje de un script](#)
 - [Referencias a elementos HTML desde un script](#)
 3. [Eventos intrínsecos](#)

4. [Modificación dinámica de documentos](#)
3. [Diseño de documentos para agentes de usuario que no soporten scripts](#)
 1. [El elemento NOSCRIPT](#)
 2. [Ocultar datos de scripts a los agentes de usuario](#)
19. [Información de referencia de SGML para HTML - Definición formal y validación de HTML](#)
 1. [Validación de documentos](#)
 2. [Catálogo SGML de ejemplo](#)
20. [Declaración SGML de HTML 4](#)
 1. [Declaración SGML](#)
21. [Definición del Tipo de Documento](#)
22. [Definición del Tipo de Documento Transicional](#)
23. [Definición del Tipo de Documento con Marcos](#)
24. [Referencias a entidades de caracteres en HTML 4](#)
 1. [Introducción a las referencias a entidades de caracteres](#)
 2. [Referencias a entidades de caracteres para caracteres de la ISO 8859-1](#)
 1. [Lista de caracteres](#)
 3. [Referencias a entidades de caracteres para símbolos, símbolos matemáticos y letras griegas](#)
 1. [Lista de caracteres](#)
 4. [Referencias a entidades de caracteres para caracteres con significado en el código y caracteres de internacionalización](#)
 1. [Lista de caracteres](#)

A. [Cambios](#)

1. [Cambios entre la versión de HTML 4.0 del 24 de abril de 1998 y la versión de HTML 4.01 del 24 de diciembre de 1999](#)
 1. [Cambios en la especificación](#)
 - [Cambios generales](#)
 - [Sobre SGML y HTML](#)
 - [Representación de Documentos HTML](#)
 - [Tipos de datos básicos de HTML](#)
 - [La estructura global de un documento HTML](#)
 - [Información sobre el idioma y la dirección del texto](#)
 - [Tablas](#)
 - [Vínculos](#)
 - [Objetos, Imágenes y Aplicaciones](#)
 - [Hojas de estilo en documentos HTML](#)

- [Marcos](#)
 - [Formularios](#)
 - [Declaración SGML](#)
 - [DTD Estricto](#)
 - [Notas](#)
 - [Referencias](#)
2. [Erratas corregidas](#)
 3. [Pequeños errores tipográficos corregidos](#)
 4. [Clarificaciones](#)
 5. [Problemas conocidos de los navegadores](#)
2. [Cambios entre las versiones del 18 de diciembre de 1997 y del 24 de abril de 1998](#)
 1. [Erratas corregidas](#)
 2. [Pequeños errores tipográficos corregidos](#)
 3. [Cambios entre HTML 3.2 y HTML 4.0 \(18 de diciembre de 1997\)](#)
 1. [Cambios en los elementos](#)
 - [Elementos nuevos](#)
 - [Elementos desaprobados](#)
 - [Elementos obsoletos](#)
 2. [Cambios en los atributos](#)
 3. [Cambios relacionados con la accesibilidad](#)
 4. [Cambios relacionados con los metadatos](#)
 5. [Cambios relacionados con el texto](#)
 6. [Cambios relacionados con los vínculos](#)
 7. [Cambios relacionados con las tablas](#)
 8. [Cambios relacionados con las imágenes, objetos y mapas de imágenes](#)
 9. [Cambios relacionados con los formularios](#)
 10. [Cambios relacionados con las hojas de estilo](#)
 11. [Cambios relacionados con los marcos](#)
 12. [Cambios relacionados con los scripts](#)
 13. [Cambios relacionados con la internacionalización](#)
- B. [Notas sobre Aplicación, Implementación y Diseño](#)
1. [Notas sobre documentos no válidos](#)
 2. [Caracteres especiales en valores de atributos URI](#)
 1. [Caracteres no ASCII en valores de atributos URI](#)
 2. [El signo & en valores de atributos URI](#)

3. [Notas de implementación de SGML](#)
 1. [Saltos de línea](#)
 2. [Especificación de datos no HTML](#)
 - [Contenido de elementos](#)
 - [Valores de atributos](#)
 3. [Características de SGML con soporte limitado](#)
 4. [Atributos booleanos](#)
 5. [Secciones marcadas](#)
 6. [Instrucciones de procesamiento](#)
 7. [Código abreviado](#)
4. [Cómo ayudar a los motores de búsqueda a indexar su sitio web](#)
 1. [Motores de búsqueda](#)
 - [El fichero robots.txt](#)
 - [Los robots y el elemento META](#)
5. [Notas sobre tablas](#)
 1. [Criterios de diseño](#)
 - [Reformateo dinámico](#)
 - [Representación incremental](#)
 - [Estructura y presentación](#)
 - [Grupos de filas y de columnas](#)
 - [Accesibilidad](#)
 2. [Algoritmos recomendados de composición](#)
 - [Algoritmo de Composición Fija](#)
 - [Algoritmo de Autocomposición](#)
6. [Notas sobre formularios](#)
 1. [Representación incremental](#)
 2. [Proyectos futuros](#)
7. [Notas sobre scripts](#)
 1. [Sintaxis reservada para macros de scripts futuras](#)
 - [Práctica actual en macros de scripts](#)
8. [Notas sobre marcos](#)
9. [Notas sobre accesibilidad](#)
10. [Notas sobre seguridad](#)
 1. [Cuestiones de seguridad relacionadas con los formularios](#)

● [Referencias](#)

1. [Referencias normativas](#)
2. [Referencias informativas](#)

- [Índice de elementos](#)
- [Índice de atributos](#)
- [Índice](#)

AVISO DE COPYRIGHT

Copyright © 1994-2001 [World Wide Web Consortium](#), ([Massachusetts Institute of Technology](#), [Institut National de Recherche en Informatique et en Automatique](#), [Keio University](#)). Todos los derechos reservados. <http://www.w3.org/Consortium/Legal/>

Los titulares del copyright proporcionan los documentos públicos que se encuentran en el sitio web del W3C de acuerdo con la siguiente Licencia. El software y las Definiciones de Tipo de Documento (*Document Type Definitions, DTDs*) asociadas con las especificaciones del W3C están gobernados por el [aviso de copyright de Software](#).

Por el uso y/o la copia de este documento, usted (el beneficiario de la Licencia del copyright) está de acuerdo en que ha leído, entendido y que respetará los siguientes términos y condiciones:

Se concede permiso para usar, copiar y distribuir los contenidos de este documento en cualquier medio y para cualquier propósito sin necesidad del pago de fianzas ni royalties, siempre y cuando se incluya la siguiente información en TODAS las copias de este documento, o partes del mismo, que usted utilice:

1. Un vínculo o URL al documento original del W3C.
2. El aviso de copyright pre-existente del autor original; caso de no existir éste, debería utilizarse un aviso de la forma siguiente: "Copyright © [fecha-del-documento] [World Wide Web Consortium](#), ([Massachusetts Institute of Technology](#), [Institut National de Recherche en Informatique et en Automatique](#), [Keio University](#)). All Rights Reserved. <http://www.w3.org/Consortium/Legal/>" (Preferiblemente en hipertexto, pero se permite una representación en texto simple.)
3. *Si existe*, el STATUS o ESTADO del documento W3C.

Cuando el espacio lo permita, debería incluirse una copia del texto completo de esta **NOTICIA**. Le instamos a que mencione a los autores en cualquier programa, documento, u otro artículo o producto que usted cree como consecuencia de la implementación de los contenidos de este documento o de cualquier parte del mismo.

Esta licencia no concede permiso para crear modificaciones o documentos derivados de documentos del W3C. Sin embargo, caso de satisfacerse adicionales (documentados en el [Copyright FAQ](#)), el W3C puede conceder permiso para crear modificaciones o documentos derivados a los individuos que satisfagan dichos requisitos.

ESTE DOCUMENTO SE OFRECE "TAL COMO ESTÁ". LOS PROPIETARIOS DEL COPYRIGHT NO HACEN GARANTÍAS NI REPRESENTACIONES DE NINGÚN TIPO, EXPRESAS O IMPLÍCITAS, INCLUYENDO, AUNQUE SIN LIMITARSE A, LAS GARANTÍAS DE COMERCIALIZACIÓN, ADECUACIÓN A UN PROPÓSITO ESPECÍFICO, O TÍTULO; QUE LOS CONTENIDOS DEL DOCUMENTO SEAN APROPIADOS PARA CUALQUIER

PROPÓSITO; NI QUE LA IMPLEMENTACIÓN DE TALES CONTENIDOS NO INFRINGIRÁ LAS PATENTES, DERECHOS DE AUTOR, MARCAS REGISTRADAS O COMERCIALES U OTROS DERECHOS DE TERCERAS PERSONAS.

LOS PROPIETARIOS DEL COPYRIGHT NO SERÁN RESPONSABLES DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL O CONSECUENTE DEBIDO A CUALQUIER USO DEL DOCUMENTO O DEL FUNCIONAMIENTO O IMPLEMENTACIÓN DE LOS CONTENIDOS DEL MISMO.

El nombre y marcas registradas de los propietarios del copyright NO pueden utilizarse en anuncios o publicidad relativos a este documento o a sus contenidos sin un permiso previo por escrito. Los derechos de autor de este documento pertenecerán en todo momento a los propietarios del copyright.

[siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

1 Sobre la especificación HTML 4

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Cómo está organizada la especificación](#)
2. [Convenciones del documento](#)
 1. [Elementos y atributos](#)
 2. [Notas y ejemplos](#)
3. [Agradecimientos](#)
 1. [Agradecimientos por la revisión actual](#)
4. [Aviso de Copyright](#)

1.1 Cómo está organizada la especificación

Esta especificación se divide en las siguientes secciones:

Secciones 2 y 3: Introducción al HTML 4

La introducción describe el lugar del HTML en el marco de la World Wide Web, proporciona una breve historia del desarrollo del HTML, explica lo que puede hacerse con HTML 4 y da algunos consejos para la creación de documentos con HTML.

Un tutorial escueto de SGML da a los lectores algo de información sobre la relación entre HTML y SGML, así como información resumida sobre cómo leer las Definiciones del Tipo de Documento (DTD).

Secciones 4 - 24: Manual de referencia de HTML 4

El grueso del manual de referencia consiste en la referencia del lenguaje HTML, que define todos los elementos y atributos del lenguaje.

Se ha organizado este documento por temas, y no por la gramática del HTML. Los temas se agrupan en tres categorías: estructura, presentación e interactividad. Aunque no es fácil dividir las estructuras HTML perfectamente entre estas tres categorías, el modelo refleja la experiencia del Grupo de Trabajo HTML de que separar la estructura del documento de su presentación produce documentos más efectivos y de mantenimiento más sencillo.

La referencia del lenguaje consiste en la siguiente información:

- Qué [caracteres](#) pueden aparecer en un documento HTML.

- [Tipos de datos](#) básicos de un documento HTML.
- Elementos que gobiernan la estructura de un documento HTML, incluyendo [texto](#), [listas](#), [tablas](#), [vínculos](#) y [objetos, imágenes y aplicaciones incluidos](#).
- Elementos que gobiernan la presentación de un documento HTML, incluyendo [hojas de estilo](#), [fuentes, colores, separadores y otras presentaciones visuales](#) y [marcos para presentaciones multiventana](#).
- Elementos que gobiernan la interactividad dentro de un documento HTML, incluyendo [formularios para la entrada de datos por el usuario](#) y [scripts para documentos activos](#).
- La definición formal SGML de HTML:
 - La [declaración SGML de HTML](#).
 - Tres DTDs: [strict](#) (estricto), [transitional](#) (transicional), y [frameset](#) (de documentos con marcos).
 - La [lista de referencias de caracteres](#).

Apéndices

El primer apéndice contiene información sobre los [cambios desde la versión HTML 3.2](#) para ayudar a los autores e implementadores durante la transición a HTML 4, los [cambios desde la especificación del 18 de diciembre de 1997](#), y los [cambios entre las versiones HTML 4.0 y HTML 4.01](#). El segundo apéndice contiene [notas de ejecución e implementación](#), y su finalidad principal es ayudar a los implementadores a crear agentes de usuario para HTML 4.

Referencias

Una lista de referencias normativas e informativas.

Índices

Tres índices permiten a los lectores acceder rápidamente a las definiciones de los [conceptos clave](#), [elementos](#) y [atributos](#).

1.2 Convenciones del documento

Este documento ha sido escrito con dos tipos de lectores en mente: autores e implementadores. Esperamos que la especificación proporcione a los autores las herramientas que necesitan para escribir documentos eficientes, atractivos y accesibles sin sobreexponerles a detalles sobre la implementación del HTML. Los implementadores, sin embargo, deberían encontrar todo lo que necesitan para crear agentes de usuario conformes.

Hay varias maneras de abordar esta especificación:

- **Leer de principio a fin.** Esta especificación comienza con una presentación general de HTML y se va haciendo más técnica y específica a medida que avanza.
- **Acceso rápido a la información.** Para poder obtener información sobre sintaxis y semántica lo más rápidamente posible, la versión en línea de la especificación presenta las siguientes características:
 1. Toda referencia a un elemento o atributo está enlazada con su definición en la especificación. Cada elemento o atributo se define en un único lugar.

2. Cada página incluye vínculos a los índices, de modo que el lector nunca está a más de dos vínculos de distancia de la definición de un [elemento](#) o [atributo](#).
3. La portada de cada sección del manual de referencia expande la tabla inicial de contenidos con más detalle sobre esa sección.

1.2.1 Elementos y atributos

Los nombres de elementos se escriben en mayúscula (p.ej., BODY). Los nombres de atributos se escriben en minúscula (p.ej., lang, onsubmit). Recuerde que HTML no diferencia entre mayúsculas y minúsculas con respecto a los nombres de elementos y atributos; esta convención sólo se sigue para facilitar la lectura.

En este documento los nombres de elementos y atributos se han codificado de modo que puedan ser representados de manera especial por algunos agentes de usuario.

La definición de cada atributo especifica el tipo de su valor. Si el tipo permite un conjunto pequeño de valores posibles, la definición enumera el conjunto de valores, separados por una barra (|).

Después de la información sobre el tipo, la definición de cada atributo indica si se distingue entre mayúsculas y minúsculas para sus valores, con un código entre corchetes ("[]"). Véase la sección sobre [mayúsculas y minúsculas](#) para más detalles.

1.2.2 Notas y ejemplos

Las notas informativas se han enfatizado para diferenciarlas del texto que las rodea y para poder ser representadas de manera especial por algunos agentes de usuario.

Todos los ejemplos que ilustran usos [desaprobados](#) se han marcado como "EJEMPLO DESAPROBADO". Los ejemplos desaprobados también incluyen soluciones alternativas recomendadas. Todos los ejemplos que ilustran usos ilegales se han marcado claramente como "EJEMPLO ILEGAL".

Los ejemplos y las notas han sido codificadas de modo que puedan ser representados de manera especial por algunos agentes de usuario.

1.3 Agradecimientos

Gracias a todos los que han ayudado a crear los borradores de trabajo que han cristalizado en la especificación HTML 4, y a todos aquellos que han enviado sugerencias y correcciones.

Muchas gracias a la Iniciativa por la Accesibilidad en la Web (grupo WAI HC) por su trabajo para mejorar la accesibilidad en HTML y a T.V. Raman (Adobe) por su trabajo previo en el desarrollo de formularios accesibles.

Los autores de esta especificación, los miembros del Grupo de Trabajo HTML del W3C, merecen un aplauso por la diligente revisión de este documento, sus comentarios constructivos, y su gran trabajo: John D. Burger (MITRE), Steve Byrne (JavaSoft), Martin J. Dürst (Universidad de Zurich), Daniel Glazman (Electricité de France), Scott Isaacs (Microsoft), Murray Maloney (GRIF), Steven Pemberton (CWI), Robert Pernet (Lotus), Jared Sorensen (Novell), Powell Smith (IBM), Robert Stevahn (HP), Ed Tecot (Microsoft), Jeffrey Veen (HotWired), Mike Wexler (Adobe), Misha Wolf

(Reuters) y Lauren Wood (SoftQuad).

Gracias a Dan Connolly (W3C) por sus rigurosas y abundantes contribuciones como editor a tiempo parcial y por sus amables consejos como presidente del Grupo de Trabajo HTML. Gracias a Sally Khudairi (W3C) por su trabajo indispensable en los comunicados de prensa.

Gracias a David M. Abrahamson y a Roger Price por su cuidadosa lectura de la especificación y sus comentarios constructivos.

Gracias a Jan Kärman, autor de [html2ps](#) por su gran ayuda en la creación de la versión Postscript de la especificación.

De particular ayuda para el W3C en Sophia-Antipolis fueron Janet Bertot, Bert Bos, Stephane Boyera, Daniel Dardailler, Yves Lafon, Håkon Lie, Chris Lilley y Colas Nahaboo (Bull).

Por último, gracias a Tim Berners-Lee, sin el cual nada de esto habría sido posible.

1.3.1 Agradecimientos por la revisión actual

Muchas gracias a Shane McCarron por el control de erratas para esta revisión de la especificación.

1.4 Aviso de Copyright

Para información sobre derechos de autor, léase la [Noticia de Propiedad Intelectual del W3C](#), la [Noticia y Licencia de Documentos del W3C](#), y la [Noticia de Propiedad Intelectual de Software del W3C](#).

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

2 Introducción a HTML 4

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [¿Qué es la World Wide Web?](#)
 1. [Introducción a los URIs](#)
 2. [Identificadores de fragmento](#)
 3. [URIs relativos](#)
2. [¿Qué es el HTML?](#)
 1. [Breve historia del HTML](#)
3. [HTML 4](#)
 1. [Internacionalización](#)
 2. [Accesibilidad](#)
 3. [Tablas](#)
 4. [Documentos compuestos](#)
 5. [Hojas de estilo](#)
 6. [Scripts](#)
 7. [Impresión](#)
4. [Creación de documentos con HTML 4](#)
 1. [Separar estructura y presentación](#)
 2. [Considerar la accesibilidad universal a la Web](#)
 3. [Ayudar a los agentes de usuario con la representación incremental](#)

2.1 ¿Qué es la World Wide Web?

La World Wide Web (Web), en castellano "Tejido Mundial", es una red de recursos de información. La Web se basa en tres mecanismos para hacer que estos recursos estén listos y a disposición de la mayor audiencia posible:

1. Un esquema uniforme de nombres para localizar recursos en la Web (p.ej., URIs).
2. Protocolos, para acceder a recursos con nombre en la Web (p.ej., HTTP).
3. Hipertexto, para navegar fácilmente entre recursos (p.ej., HTML).

Las relaciones entre los tres mecanismos son evidentes a todo lo largo de esta especificación.

2.1.1 Introducción a los URIs

Todos los recursos disponibles en la Web -- documentos HTML, imágenes, videoclips, programas, etc. -- tienen una dirección que puede ser codificada mediante un Universal Resource Identifier, o "URI", es decir, un Identificador Universal de Recursos.

Los URIs se componen normalmente de tres partes:

1. El esquema de nombres del mecanismo usado para acceder al recurso.
2. El nombre de la máquina que aloja el recurso.
3. El nombre en sí del recurso, dado en forma de "path" o "ruta de acceso".

Consideremos el URI que designa la página de Informes Técnicos del W3C:

```
http://www.w3.org/TR
```

Este URI puede leerse de la siguiente manera: Hay un documento disponible a través del protocolo HTTP (ver [RFC2616](#)), que se encuentra en la máquina `www.w3.org`, accesible a través de la ruta `/TR`. Entre otros esquemas que pueden encontrarse en documentos HTML se incluyen "mailto" para correo electrónico y "ftp" para FTP.

Aquí tenemos otro ejemplo de URI. Éste se refiere al buzón de correo electrónico de un usuario:

```
...aquí va texto...
```

Para cualquier comentario, envíe un mensaje a

```
<A href="mailto:jose@algunsitio.com">Jose Chévere</A>.
```

Nota. La mayoría de los lectores estará familiarizado con el término "URL" y no con el término "URI". Los URLs forman un subconjunto del esquema de nombres URI, que es más general.

2.1.2 Identificadores de fragmento

Algunos URIs se refieren a una localización dentro de un recurso. Este tipo de URIs termina con un "#" seguido de un identificador de vínculo (llamado identificador de fragmento). Por ejemplo, aquí tenemos un URI que apunta a un vínculo llamando `seccion_2`:

```
http://algunsitio.com/html/superior.html#seccion_2
```

2.1.3 URIs relativos

Un URI relativo no contiene información sobre el esquema de nombres. Su ruta de acceso se refiere generalmente a un recurso que está en la misma máquina que el documento actual. Los URIs relativos pueden contener indicadores relativos de ruta (p.ej., ".." significa un nivel superior en la jerarquía definida por la ruta de acceso), y puede contener [identificadores de fragmento](#).

Los URIs relativos se [convierten en URIs completos](#) a partir de un URI base. Como ejemplo de conversión de un URI relativo, supongamos que tenemos el URI base "http://www.acme.com/soporte/intro.html". El URI relativo de la siguiente línea de código de un

vínculo hipertextual:

```
<A href="proveedores.html">Proveedores</A>
```

se expandiría al URI completo "http://www.acme.com/soporte/proveedores.html", mientras que el URI relativo de la siguiente línea de código de una imagen:

```
<IMG src="../iconos/logo.gif" alt="logo">
```

se expandiría al URI completo "http://www.acme.com/iconos/logo.gif".

En HTML, los URIs se usan para:

- Crear un vínculo a otro documento o recurso (ver elementos [A](#) y [LINK](#)).
- Crear un vínculo a una hoja de estilo o script externos (ver elementos [LINK](#) y [SCRIPT](#)).
- Incluir una imagen, objeto o aplicación en una página (ver elementos [IMG](#), [OBJECT](#), [APPLET](#) e [INPUT](#)).
- Crear un mapa de imágenes (ver elementos [MAP](#) y [AREA](#)).
- Enviar un formulario (ver elemento [FORM](#)).
- Crear un documento con marcos (ver elementos [FRAME](#) e [IFRAME](#)).
- Citar una referencia externa (ver elementos [Q](#), [BLOCKQUOTE](#), [INS](#) y [DEL](#)).
- Hacer referencia a convenciones de metadatos que describen un documento (ver elemento [HEAD](#)).
- Designar una imagen de fondo (ver elemento [BODY](#)). Este es un uso [desaprobado](#) y deberían usarse hojas de estilo en su lugar.

Consulte la sección sobre el tipo [URI](#) para más información sobre URIs.

2.2 ¿Qué es el HTML?

Para publicar información y distribuirla globalmente, se necesita un lenguaje entendido universalmente, una especie de lengua franca de publicación que todas las computadoras puedan comprender potencialmente. El lenguaje de publicación usado por la World Wide Web es el HTML (acrónimo de HyperText Markup Language, Lenguaje para el Formato de Documentos de Hipertexto).

El HTML da a los autores las herramientas para:

- Publicar documentos en línea con encabezados, textos, tablas, listas, fotos, etc.
- Obtener información en línea a través de vínculos de hipertexto, haciendo clic con el botón de un ratón.
- Diseñar formularios para realizar transacciones con servicios remotos, para buscar información, hacer reservas, pedir productos, etc.
- Incluir hojas de cálculo, videoclips, sonidos, y otras aplicaciones directamente en sus documentos.

2.2.1 Breve historia del HTML

El HTML fue desarrollado originalmente por Tim Berners-Lee mientras estaba en el CERN, y fue popularizado por el navegador Mosaic desarrollado en el NCSA. Durante los años 90 ha proliferado con el crecimiento explosivo de la Web. Durante este tiempo, el HTML se ha desarrollado de diferentes maneras. La Web depende de que los autores de páginas Web y las compañías compartan las mismas convenciones de HTML. Esto ha motivado el trabajo colectivo en las especificaciones del HTML.

El HTML 2.0 (noviembre de 1995, ver [\[RFC1866\]](#)) fue desarrollado bajo los auspicios de la Internet Engineering Task Force (IETF) para codificar lo que era la práctica común a finales de 1994. HTML+ (1993) y HTML 3.0 (1995, ver [\[HTML30\]](#)) propusieron versiones mucho más ricas de HTML. A pesar de no haber logrado nunca el consenso en las discusiones sobre estándares, estos borradores llevaron a la adopción de un número de nuevas características. Los esfuerzos del Grupo de Trabajo HTML del World Wide Web Consortium para codificar la práctica común en 1996 condujeron a HTML 3.2 (enero de 1997, ver [\[HTML32\]](#)). Los cambios desde HTML 3.2 se resumen en el [Apéndice A](#).

La mayoría de las personas están de acuerdo en que los documentos HTML deberían funcionar bien en diferentes navegadores y plataformas. Gracias a la interoperabilidad los proveedores de contenidos reducen gastos, ya que sólo deben desarrollar una versión de cada documento. Si este esfuerzo no se realiza, hay un riesgo mucho mayor de que la Web se convierta en un mundo propietario de formatos incompatibles, que al final acabaría por reducir el potencial comercial de la Web para todos los que forman parte de ella.

Cada versión de HTML ha intentado reflejar un consenso cada vez mayor entre los interlocutores de la industria, de modo que no se desperdicien las inversiones hechas por los proveedores de contenidos y que sus documentos no dejen de ser legibles a corto plazo.

El HTML ha sido desarrollado con la premisa de que cualquier tipo de dispositivo debería ser capaz de usar la información de la Web: PCs con pantallas gráficas con distintas resoluciones y colores, teléfonos móviles, dispositivos de mano, dispositivos de salida y entrada por voz, computadoras con anchos de banda grandes o pequeños, etc.

2.3 HTML 4

El HTML 4 desarrolla el lenguaje HTML con mecanismos para hojas de estilo, ejecución de scripts, marcos, objetos incluidos, soporte mejorado para texto de derecha a izquierda y direcciones mezcladas, tablas más ricas y mejoras en formularios, ofreciendo mejoras de accesibilidad para personas con discapacidades.

El HTML 4.01 es una revisión de HTML 4.0 que corrige errores e introduce algunos [cambios desde la revisión anterior](#).

2.3.1 Internacionalización

Esta versión de HTML ha sido diseñada con la ayuda de expertos en el campo de la internacionalización, para que los documentos puedan ser escritos en cualquier idioma y fácilmente transportados por todo el mundo. Esto se ha logrado incorporando la referencia [\[RFC2070\]](#), que trata

sobre la internacionalización del HTML.

Un paso importante ha sido la adopción del estándar ISO/IEC:10646 (ver [\[ISO10646\]](#)) como el conjunto de caracteres del documento para HTML. Este es el estándar más exhaustivo del mundo relacionado con la representación de caracteres internacionales, dirección del texto, puntuación, y otros aspectos de los idiomas del mundo.

HTML ofrece ahora un mayor soporte para diversos lenguajes humanos dentro de un documento. Esto permite un indexado más efectivo de los documentos por parte de los motores de búsqueda, tipografía de mayor calidad, mejor conversión de texto a voz, mejor separación de palabras, etc.

2.3.2 Accesibilidad

A medida que la comunidad de la Web crece y sus miembros diversifican sus habilidades, es crucial que las tecnologías subyacentes sean apropiadas para sus fines específicos. El HTML se ha diseñado para hacer las páginas web más accesibles a aquéllos con limitaciones físicas. Los desarrollos de HTML 4 derivados de cuestiones de accesibilidad incluyen:

- Mejor distinción entre la estructura y la presentación de un documento, aconsejándose la utilización de hojas de estilo en lugar de elementos y atributos de presentación de HTML.
- Mejores formularios, incluyendo la adición de teclas de acceso, la posibilidad de agrupar semánticamente los controles de un formulario, la posibilidad de agrupar las opciones [SELECT](#) semánticamente, y los rótulos activos.
- La posibilidad de codificar una descripción textual de un objeto incluido (con el elemento [OBJECT](#)).
- Un nuevo mecanismo de mapas de imágenes en el lado del cliente (el elemento [MAP](#)) que permite a los autores integrar vínculos de imagen y de texto.
- El requerimiento de que se incluya texto alternativo acompañando a las imágenes dentro del elemento [IMG](#) y a los mapas de imágenes dentro del elemento [AREA](#).
- Soporte de los atributos [title](#) y [lang](#) en todos los elementos.
- Soporte de los elementos [ABBR](#) y [ACRONYM](#).
- Un espectro más amplio de medios utilizables por hojas de estilo (tty, braille, etc.).
- Tablas mejores, incluyendo títulos, grupos de columnas y mecanismos para facilitar su representación no visual.
- Descripciones largas para tablas, imágenes, marcos, etc.

Los autores que diseñen páginas pensando en el aspecto de la accesibilidad, no sólo recibirán el agradecimiento de la comunidad preocupada por el tema de la accesibilidad, sino que también se beneficiarán en otros aspectos: los documentos HTML bien diseñados que distingan entre estructura y presentación se adaptarán más fácilmente a las nuevas tecnologías.

Nota. Para más información sobre el diseño de documentos HTML accesibles, consúltese en [\[WAI\]](#).

2.3.3 Tablas

El nuevo modelo de tablas de HTML se basa en [\[RFC1942\]](#). Ahora los autores tienen mayor control sobre la estructura y la presentación (p.ej., grupos de columnas). La posibilidad que tienen los autores de recomendar anchuras para las columnas permite a los agentes de usuario mostrar los datos de la

tabla incrementalmente (a medida que los reciben) en lugar de tener que cargar toda la tabla antes de empezar a representarla.

Nota. En el momento de escribir esta especificación, algunas herramientas de creación de HTML se basan exhaustivamente en el uso de tablas para dar formato a los documentos, lo cual puede causar fácilmente problemas de accesibilidad.

2.3.4 Documentos compuestos

HTML ofrece ahora un mecanismo estándar para incluir objetos genéricos y aplicaciones dentro de documentos HTML. El elemento [OBJECT](#) (junto a los antiguos elementos [IMG](#) y [APPLET](#), más específicos) proporciona un mecanismo para incluir imágenes, vídeo, sonido, fórmulas matemáticas, aplicaciones especializadas y otros objetos en un documento. También permite a los autores especificar una jerarquía de representaciones alternativas para los agentes de usuario que no soporten una representación específica.

2.3.5 Hojas de estilo

Las hojas de estilo simplifican el código HTML y liberan en gran medida al HTML de las responsabilidades de presentación. Esto da tanto a los autores como a los usuarios control sobre la presentación de los documentos: fuentes, alineación, colores, etc.

La información de estilo puede especificarse para elementos individuales o para grupos de elementos. La información de estilo puede especificarse en un documento HTML o en hojas de estilo externas.

El mecanismo para asociar una hoja de estilo con un documento es independiente del lenguaje de la hoja de estilo.

Antes de la llegada de las hojas de estilo, los autores tenían un control limitado sobre la representación. HTML 3.2 incluía un número de atributos y elementos que ofrecían control sobre la alineación, el tamaño de la fuente y el color del texto. Además los autores utilizaban las tablas y las imágenes como medio de organizar la presentación de sus páginas. El tiempo relativamente largo que necesitan los usuarios para actualizar sus navegadores hará que estas características sigan siendo usadas durante algún tiempo. Sin embargo, al ofrecer las hojas de estilo mecanismos de presentación más potentes, el World Wide Web Consortium declarará obsoletos en el futuro muchos de los elementos y atributos de presentación del HTML. A lo largo de esta especificación dichos elementos y atributos se marcan como "[desaprobados](#)". Se acompañan con ejemplos de cómo lograr los mismos efectos con otros elementos o con hojas de estilo.

2.3.6 Ejecución de scripts

Gracias a los scripts, los autores pueden crear páginas web dinámicas (p.ej., "formularios inteligentes", que reaccionan a medida que los usuarios los rellenan) y utilizar el HTML para crear aplicaciones en red.

Los mecanismos proporcionados para incluir scripts en un documento HTML son independientes del lenguaje de programación de los scripts.

2.3.7 Impresión

En ocasiones los autores querrán facilitar a los usuarios la impresión de su trabajo, sin limitarse al documento actual. Cuando un documento forme parte de un trabajo mayor, las relaciones entre los documentos pueden describirse mediante el elemento HTML [LINK](#) o usando el Resource Description Framework (RDF, Marco de Descripción de Recursos) del W3C (ver [\[RDF10\]](#)).

2.4 Creación de documentos con HTML 4

Recomendamos a los autores e implementadores que sigan los siguientes principios generales cuando trabajen con HTML 4.

2.4.1 Separar estructura y presentación

El HTML tiene sus raíces en SGML, que siempre ha sido un lenguaje para la especificación de código estructural. A medida que el HTML madura, un número cada vez mayor de sus elementos y atributos presentacionales ha sido reemplazado por otros mecanismos, en particular las hojas de estilo. La experiencia ha demostrado que separando la estructura de un documento de sus aspectos presentacionales se reduce el coste de servir a un amplio espectro de plataformas, medios, etc. y se facilitan las revisiones del documento.

2.4.2 Considerar la accesibilidad universal a la Web

Para hacer la Web más accesible a todos, en especial a aquéllos con discapacidades, los autores deberían considerar cómo pueden representarse sus documentos en diferentes plataformas: navegadores basados en voz, lectores braille, etc. No estamos recomendando a los autores que limiten su creatividad, sólo que consideren representaciones alternativas de sus diseños. El HTML ofrece un número de mecanismos con este fin (p.ej., el atributo [alt](#), el atributo [accesskey](#), etc.)

Además de esto, los autores deberían recordar que sus documentos pueden llegar a una audiencia muy lejana con diferentes computadoras y configuraciones. Para que los documentos sean correctamente interpretados, los autores deberían incluir en sus documentos información sobre el idioma natural y la dirección del texto, cómo está codificado el documento, y otras cuestiones relacionadas con la internacionalización.

2.4.3 Ayudar a los agentes de usuario con la representación incremental

Mediante un diseño cuidadoso de las tablas y haciendo uso de las nuevas características de las tablas de HTML 4, los autores pueden ayudar a los agentes de usuario a representar los documentos más rápidamente. Los autores pueden aprender a diseñar tablas para su representación incremental (ver elemento [TABLE](#)). Los implementadores deberían consultar las [notas sobre tablas](#) del apéndice para obtener información sobre algoritmos incrementales.

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

3 Sobre SGML y HTML

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Introducción al SGML](#)
2. [Estructuras SGML usadas en HTML](#)
 1. [Elementos](#)
 2. [Atributos](#)
 3. [Referencias de caracteres](#)
 4. [Comentarios](#)
3. [Cómo leer el DTD de HTML](#)
 1. [Comentarios DTD](#)
 2. [Definiciones de entidades paramétricas](#)
 3. [Declaraciones de elementos](#)
 - [Definiciones del modelo de contenido](#)
 4. [Declaraciones de atributos](#)
 - [Entidades DTD en definiciones de atributos](#)
 - [Atributos booleanos](#)

Esta sección del documento presenta el SGML y habla sobre su relación con HTML. En el estándar puede encontrarse información completa sobre el SGML (ver [ISO8879](#)).

3.1 Introducción al SGML

El SGML es un sistema para definir lenguajes para dar formato a documentos (*markup languages*). Los autores utilizan un código de formato (en inglés *markup*) en sus documentos para representar información estructural, presentacional y semántica junto con el contenido. El HTML es un ejemplo de lenguaje de formato de documentos. Aquí tenemos un ejemplo de un documento HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Mi primer documento HTML</TITLE>
  </HEAD>
  <BODY>
    <P>¡Hola mundo!
  </BODY>
</HTML>
```

Un documento HTML se divide en una sección de cabecera (aquí, entre `<HEAD>` y `</HEAD>`) y un cuerpo (aquí, entre `<BODY>` y `</BODY>`). El título del documento aparece en la cabecera (junto con otras informaciones sobre el documento), y el contenido del documento aparece en el cuerpo. El cuerpo de este ejemplo contiene únicamente un párrafo, codificado o marcado como `<P>`.

Cada lenguaje de formato de documentos definido con SGML se llama aplicación SGML. Una aplicación SGML se caracteriza generalmente por:

1. [Una declaración SGML](#). La declaración SGML especifica qué caracteres y delimitadores pueden aparecer en la aplicación.
2. [Una definición del tipo de documento \(*document type definition, DTD*\)](#). El DTD define la sintaxis de las estructuras de formato. El DTD puede incluir definiciones adicionales, tales como [referencias a entidades de caracteres](#).
3. Una especificación que describe la semántica que se debe conferir al código de formato. Esta especificación también impone restricciones de sintaxis que no pueden expresarse dentro del DTD.
4. Documentos que contienen datos (contenido) y código (*markup*). Cada documento contiene una referencia al DTD que debe usarse para interpretarlo.

Esta especificación incluye una [declaración SGML](#), tres definiciones del tipo de documento (ver la sección sobre [información sobre la versión de HTML](#) para una descripción de las tres), y una lista de [referencias de caracteres](#).

3.2 Estructuras SGML usadas en HTML

Las secciones siguientes presentan las estructuras de SGML que se usan en HTML.

El apéndice enumera algunas [características de SGML](#) que no son soportadas por la mayoría de las herramientas y agentes de usuario HTML y que deberían evitarse.

3.2.1 Elementos

Una [definición del tipo de documento](#) SGML declara tipos de elementos que representan estructuras o un comportamiento deseado. HTML incluye elementos que representan párrafos, vínculos de hipertexto, listas, tablas, imágenes, etc.

Cada declaración de tipo de elemento describe generalmente tres partes: una etiqueta inicial, un contenido y una etiqueta final.

El nombre del elemento aparece en la etiqueta inicial (escrita <nombre-del-elemento>) y en la etiqueta final (escrita </nombre-del-elemento>); obsérvese la barra que precede al nombre del elemento en la etiqueta final. Por ejemplo, las etiquetas inicial y final del tipo de elemento [UL](#) delimitan los objetos de una lista:

```
<UL>
<LI><P>...objeto de lista 1...
<LI><P>...objeto de lista 2...
</UL>
```

Algunos tipos de elemento de HTML permiten a los autores omitir las etiquetas finales (p.ej., los tipos de elemento [P](#) y [LI](#)). Algunos elementos también permiten omitir la etiqueta inicial; por ejemplo, [HEAD](#) y [BODY](#). El DTD de HTML indica para cada tipo de elemento si la etiqueta inicial y la etiqueta final son necesarias.

Algunos tipos de elemento de HTML no tienen contenido. Por ejemplo, el elemento de salto de línea [BR](#) no tiene contenido; su único papel es terminar una línea de texto. Estos elementos vacíos nunca tienen etiquetas finales. La [definición del tipo de documento](#) y el texto de la especificación indican si un tipo de elemento es vacío (es decir, si no tiene contenido) o, en caso de que pueda tener contenido, lo que se considera como contenido legal.

Para los nombres de los elementos no se distingue nunca entre mayúsculas y minúsculas.

Consulte el estándar SGML para más información sobre las reglas que gobiernan a los elementos (p.ej., deben anidarse correctamente, una etiqueta final cierra hasta su etiqueta inicial correspondiente todas las etiquetas iniciales intermedias abiertas (sección 7.5.1.), etc.).

Por ejemplo, el siguiente párrafo:

```
<P>Este es el primer párrafo.</P>
...un elemento en bloque...
```

puede escribirse sin su etiqueta final:

```
<P>Este es el primer párrafo.
...un elemento en bloque...
```

ya que la etiqueta inicial <P> es cerrada por el siguiente elemento en bloque. Análogamente, si un párrafo está contenido en un

elemento en bloque, como aquí:

```
<DIV>
<P>Este es el párrafo.
</DIV>
```

la etiqueta final del elemento en bloque que lo contiene (aquí, </DIV>) implica la etiqueta final de la etiqueta inicial <P> abierta.

Los elementos no son etiquetas. Algunas personas se refieren a los elementos como etiquetas (p.ej., "la etiqueta P"). Recuerde que el elemento es una cosa, y la etiqueta (sea la inicial o la final) es otra cosa distinta. Por ejemplo, el elemento HEAD está siempre presente, incluso si tanto la etiqueta inicial de HEAD como la final están ausentes del código.

Todos los tipos de elemento declarados en esta especificación se enumeran en el [índice de elementos](#).

3.2.2 Atributos

Los elementos pueden tener propiedades asociadas, llamadas atributos, que pueden tener valores (por defecto o asignados por el autor o por un script). Las parejas atributo/valor aparecen antes del ">" final de la etiqueta inicial de un elemento. En la etiqueta inicial de un elemento puede aparecer cualquier número de parejas (legales) atributo/valor, separadas por espacios. Pueden aparecer en cualquier orden.

En este ejemplo se establece el atributo [id](#) de un elemento [H1](#):

```
<H1 id="seccion1">
Este es un título identificado gracias al atributo id
</H1>
```

Por defecto, SGML requiere que todos los valores de atributo estén delimitados ya sea mediante comillas dobles (ASCII decimal 34) o mediante comillas simples (ASCII decimal 39). Un valor de atributo delimitado por comillas dobles puede contener signos de comillas simples, y viceversa. Los autores también pueden utilizar [referencias numéricas de caracteres](#) para representar comillas dobles (") y simples ('). Para las comillas dobles los autores también pueden utilizar la [referencia a entidad de caracteres](#) ".

En ciertos casos, los autores pueden especificar el valor de un atributo sin ponerlo entre comillas. El valor del atributo sólo puede contener letras (a-z y A-Z), dígitos (0-9), guiones (ASCII decimal 45), puntos (ASCII decimal 46), subguiones (ASCII decimal 95) y dos puntos (ASCII decimal 58). Recomendamos usar comillas incluso cuando sea posible eliminarlas.

Para los nombres de atributos no se distingue nunca entre mayúsculas y minúsculas.

Para los valores de los atributos en general no se distingue entre mayúsculas y minúsculas. En el manual de referencia se indica, en la definición de cada atributo, si se distingue para su valor entre mayúsculas y minúsculas.

Todos los atributos definidos por esta especificación se enumeran en el [índice de atributos](#).

3.2.3 Referencias de caracteres

Las referencias de caracteres son nombres simbólicos o numéricos de caracteres que pueden incluirse en un documento HTML. Son útiles para hacer referencia a caracteres usados esporádicamente, o a aquellos que son difíciles o imposibles de introducir con las herramientas de creación. A lo largo de este documento pueden verse referencias de caracteres; comienzan con un signo "&" y terminan con un punto y coma (;). A continuación figuran algunos ejemplos comunes:

- "<" representa el signo <.
- ">" representa el signo >.
- """ representa el signo ".
- "å" (en decimal) representa la letra "a" con un pequeño círculo encima.
- "И" (en decimal) representa la letra mayúscula cirílica "I".
- "水" (en hexadecimal) representa el carácter chino para el agua.

Hablaremos con más detalle acerca de las [referencias de caracteres en HTML](#) más adelante en la sección sobre [el conjunto de caracteres de los documentos HTML](#). La especificación contiene también una [lista de referencias de caracteres](#) que pueden aparecer en documentos HTML 4.

3.2.4 Comentarios

Los comentarios en HTML tienen la sintaxis siguiente:

```
<!-- esto es un comentario -->
<!-- y esto también es un comentario,
      que ocupa más de una línea -->
```

No se permite espacio en blanco entre el delimitador de apertura de declaración de etiqueta ("**<!**") y el delimitador de apertura de comentario ("**--**"), pero sí se permite entre el delimitador de cierre de comentario ("**--**" y el delimitador de cierre de declaración de etiqueta ("**>**"). Un error común es incluir una cadena de guiones ("**---**") dentro de un comentario. Los autores deberían evitar poner dos o más guiones adyacentes dentro de un comentario.

La información que aparece entre comentarios no tiene significado especial (p.ej., [las referencias de caracteres](#) no se interpretan como tales).

Obsérvese que los comentarios forman parte del código.

3.3 Cómo leer el DTD de HTML

En esta especificación, todas las declaraciones de elementos y atributos vienen acompañadas del fragmento correspondiente de la [definición del tipo de documento](#). Hemos optado por incluir en la especificación fragmentos del DTD en lugar de intentar describir las propiedades de un elemento de una forma que sea más abordable, pero más larga y menos precisa. El tutorial que sigue permitirá a los lectores que no estén familiarizados con el SGML leer el DTD y entender los detalles técnicos de la especificación HTML.

3.3.1 Comentarios DTD

En los DTDs, los comentarios pueden extenderse a lo largo de una o más líneas. En el DTD, los comentarios están delimitados por un par de marcas "**--**", p.ej.:

```
<!ELEMENT PARAM - O EMPTY          -- valor de propiedad con nombre -->
```

Aquí, el comentario "valor de propiedad con nombre" explica el uso del tipo de elemento [PARAM](#). Los comentarios de un DTD son únicamente informativos.

3.3.2 Definiciones de entidades paramétricas

El [DTD HTML](#) comienza con una serie de definiciones de entidades paramétricas. Una definición de entidad paramétrica define una clase de macro a la que se puede hacer referencia y ser expandida en cualquier otra parte del DTD. Estas macros no pueden aparecer en documentos HTML, sólo en el DTD. Otros tipos de macros, llamadas [referencias de caracteres](#), sí pueden utilizarse en el texto de un documento HTML o dentro de los valores de los atributos.

Cuando se hace referencia a una entidad paramétrica por su nombre en el DTD, ésta se expande, sustituyéndose por una cadena.

Una definición de entidad paramétrica comienza con la palabra clave **<!ENTITY %** seguida por el nombre de la entidad, la cadena entre comillas a la que se expande la entidad, y por último un **>** de cierre. Las apariciones de entidades paramétricas dentro de un DTD comienzan con un **"%**", seguido por el nombre de la entidad paramétrica, y finalmente por un **;"** opcional.

El siguiente ejemplo define la cadena a la que se expandirá la entidad **"%fontstyle;"**.

```
<!ENTITY % fontstyle "TT | I | B | BIG | SMALL">
```

La cadena a la que se expande la entidad paramétrica puede contener otros nombres de entidades paramétricas. Estos nombres se expanden recursivamente. En el siguiente ejemplo, la entidad paramétrica **"%inline;"** se define de modo que incluya las entidades paramétricas **"%fontstyle;"**, **"%phrase;"**, **"%special;"** y **"%formctrl;"**.

```
<!ENTITY % inline "#PCDATA | %fontstyle; | %phrase; | %special; | %formctrl;">
```

Hay dos entidades DTD que aparecen frecuentemente en el [DTD HTML](#): **"%block;"** e **"%inline;"**. Se utilizan cuando el

modelo de contenido incluye [elementos en bloque y elementos en línea](#), respectivamente (definidos en la sección sobre la [estructura global de un documento HTML](#)).

3.3.3 Declaraciones de elementos

El grueso del DTD HTML consiste en la declaración de tipos de elementos y sus atributos. La palabra clave `<!ELEMENT` comienza una declaración y el carácter `>` la termina. Entre ellos se especifica:

1. El nombre del elemento.
2. Si las etiquetas del elemento son opcionales. Si aparecen dos guiones después del nombre del elemento tanto la etiqueta inicial como la final son obligatorias. Un guión seguido por la letra "O" indica que puede omitirse la etiqueta final. Un par de letras "O" indican que tanto la etiqueta inicial como la final pueden omitirse.
3. Los contenidos del elemento, si los hay. El contenido admitido para un elemento se denomina su modelo de contenido. Los tipos de elemento diseñados de modo que no tengan contenido se llaman elementos vacíos. El modelo de contenido de tales tipos de elemento se declara con la palabra clave "EMPTY".

En este ejemplo:

```
<!ELEMENT UL - - (LI)+>
```

- El tipo de elemento declarado es **UL**.
- Los dos guiones indican que tanto la etiqueta inicial `` como la final `` son obligatorias para este tipo de elemento.
- El modelo de contenido de este tipo de elemento se ha declarado como "al menos un elemento LI". Más adelante se explica cómo especificar los modelos de contenido.

Este ejemplo ilustra la declaración de un tipo de elemento vacío:

```
<!ELEMENT IMG - O EMPTY>
```

- El tipo de elemento declarado es **IMG**.
- El guión y la "O" que le sigue indican que la etiqueta final puede omitirse, pero eso, junto al modelo de contenido "EMPTY", implica que la etiqueta final **debe** ser omitida.
- La palabra clave "EMPTY" significa que las apariciones de este tipo no deben tener contenido.

Definiciones del modelo de contenido

El modelo de contenido describe cuál puede ser el contenido de una aparición de un tipo de elemento. Las definiciones del modelo de contenido pueden incluir:

- Nombres de los tipos de elementos permitidos o prohibidos (p.ej., el elemento [UL](#) contiene apariciones del tipo de elemento [LI](#), y el tipo de elemento [P](#) no puede contener otros elementos [P](#)).
- Entidades DTD (p.ej., el elemento [LABEL](#) contiene apariciones de la entidad paramétrica "%inline;").
- Texto de documento (indicado mediante la estructura SGML "#PCDATA"). El texto puede contener [referencias de caracteres](#). Recuérdese que éstas comienzan con `&` y terminan con un punto y coma (p.ej., "Las aventuras de Astérix" contiene la referencia a la entidad de caracteres del carácter "e con acento agudo").

El modelo de contenido de un elemento se especifica de acuerdo a la sintaxis siguiente. Obsérvese que la lista que sigue es una simplificación de las reglas de sintaxis de SGML y no habla, por ejemplo, de precedencias.

(...)

Delimita un grupo.

A

A debe aparecer una sola vez.

A+

A debe aparecer una o más veces.

A?

A debe aparecer cero o una veces.

A*

A puede aparecer cero o más veces.

+(A)

A puede aparecer.

-(A)

A no debe aparecer.

A | B

O bien A o bien B deben aparecer, pero no las dos.

A , B

Tanto A como B deben aparecer, y en ese orden.

A & B

Tanto A como B deben aparecer, en cualquier orden.

Aquí presentamos algunos ejemplos sacados del DTD de HTML:

```
<!ELEMENT UL - - (LI)+>
```

El elemento [UL](#) debe contener uno o más elementos [LI](#).

```
<!ELEMENT DL - - (DT|DD)+>
```

El elemento [DL](#) debe contener uno o más elementos [DT](#) o [DD](#) en cualquier orden.

```
<!ELEMENT OPTION - O (#PCDATA)>
```

El elemento [OPTION](#) sólo puede contener texto y entidades, tales como `&`. Esto se indica con el tipo de datos de SGML `#PCDATA`.

Algunos tipos de elementos HTML usan una característica adicional de SGML para excluir elementos de su modelo de contenido. Los elementos excluidos van precedidos de un guión, y van al final de la descripción del modelo de contenido. Las exclusiones explícitas prevalecen sobre los elementos permitidos.

En este ejemplo, el `-(A)` significa que el elemento [A](#) no puede aparecer en otro elemento [A](#) (es decir, los vínculos no pueden estar anidados).

```
<!ELEMENT A - - (%inline;)* -(A)>
```

Obsérvese que el tipo de elemento [A](#) es parte de la entidad paramétrica del DTD `"%inline;"`, pero está excluido explícitamente debido al `-(A)`.

Análogamente, la siguiente declaración del tipo de elemento [FORM](#) prohíbe los formularios anidados:

```
<!ELEMENT FORM - - (%block;|SCRIPT)+ -(FORM)>
```

3.3.4 Declaraciones de atributos

La palabra clave `<!ATTLIST` comienza la declaración de los atributos que puede tomar un elemento. Va seguida por el nombre del elemento en cuestión, una lista de definiciones de atributos, y un `>` de cierre. Cada definición de atributo es una terna que define:

- El nombre del atributo.
- El tipo del valor del atributo o un conjunto explícito de valores posibles. Los valores definidos explícitamente por el DTD no distinguen entre mayúsculas y minúsculas. Consulte la sección sobre [tipos de datos básicos de HTML](#) para más información sobre tipos de valores de atributos.
- Si el valor por defecto del atributo es implícito (palabra clave `"#IMPLIED"`), en cuyo caso el valor por defecto debe ser proporcionado por el agente de usuario (en algunos casos heredándolo de elementos padre); siempre requerido (palabra clave `"#REQUIRED"`); o fijo e igual a un valor dado (palabra clave `"#FIXED"`). Algunas definiciones de atributos especifican explícitamente un valor por defecto para el atributo.

En este ejemplo, se define el atributo [name](#) para el elemento [MAP](#). El atributo es opcional para este elemento.

```
<!ATTLIST MAP
  name          CDATA          #IMPLIED
>
```

El tipo de los valores permitidos para el atributo está dado como CDATA, un tipo de datos de SGML. CDATA es texto que puede contener [referencias de caracteres](#).

Para más información sobre "CDATA", "NAME", "ID", y otros tipos de datos, consúltese la sección sobre [tipos de datos de HTML](#).

Los siguientes ejemplos ilustran varias definiciones de atributos:

```
rowspan        NUMBER        1          -- número de filas abarcado por la celda --
http-equiv     NAME          #IMPLIED -- nombre de encabezado de respuesta HTTP --
id             ID           #IMPLIED   -- identificador único a nivel de documento --
valign         (top|middle|bottom|baseline) #IMPLIED
```

El atributo `rowspan` requiere valores de tipo NUMBER. El valor por defecto está dado explícitamente y es "1". El atributo opcional `http-equiv` requiere valores de tipo NAME. El atributo opcional `id` requiere valores de tipo ID. El atributo opcional `valign` está restringido a los valores del conjunto {top, middle, bottom, baseline}.

Entidades DTD en definiciones de atributos

Las definiciones de atributos también pueden contener referencias a entidades.

En este ejemplo, vemos que la lista de definiciones de atributos para el elemento [LINK](#) comienza con la entidad paramétrica "%attrs;".

```
<!ELEMENT LINK - O EMPTY          -- un vínculo independiente del medio -->
<!ATTLIST LINK
  %attrs;          -- %coreattrs, %il8n, %events --
  charset          %Charset;      #IMPLIED   -- codificación de caracteres del
                                     recurso vinculado --
  href             %URI;          #IMPLIED   -- URI del recurso vinculado --
  hreflang        %LanguageCode; #IMPLIED   -- código de idioma --
  type            %ContentType;  #IMPLIED   -- tipo consultivo de contenido --
  rel             %LinkTypes;    #IMPLIED   -- tipos de vínculos directos --
  rev            %LinkTypes;    #IMPLIED   -- tipos de vínculos inversos --
  media          %MediaDesc;    #IMPLIED   -- para representar en estos medios --
>
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **prohibida***

La entidad paramétrica "%attrs;" se define como sigue:

```
<!ENTITY % attrs "%coreattrs; %il8n; %events;">
```

La entidad paramétrica "%coreattrs;" de la definición de "%attrs;" se expande como sigue:

```
<!ENTITY % coreattrs
  "id            ID            #IMPLIED   -- identificador único a nivel de documento --
  class         CDATA         #IMPLIED   -- lista de clases separadas por espacios --
  style        %StyleSheet;  #IMPLIED   -- información de estilo asociada --
  title        %Text;        #IMPLIED   -- título consultivo --"
>
```

La entidad paramétrica "%attrs;" se ha definido por conveniencia ya que estos atributos se definen para la mayoría de los tipos de elemento de HTML.

Análogamente, el DTD define la entidad paramétrica "%URI;" que se expande a la cadena "CDATA".

```
<!ENTITY % URI "CDATA"
  -- un Identificador Uniforme de Recursos,
  ver \[URI\]
-->
```

Como se muestra en este ejemplo, la entidad paramétrica "%URI;" proporciona a los lectores del DTD más información sobre el tipo de datos que se espera para un atributo. Se han definido entidades similares para "%Color;", "%Charset;", "%Length;", "%Pixels;", etc.

Atributos booleanos

Algunos atributos actúan como variables booleanas (p.ej., el atributo [selected](#) del elemento [OPTION](#)). Su aparición en la etiqueta inicial de un elemento implica que el valor del atributo es "verdadero" ("true"). Su ausencia implica un valor "falso" ("false").

Los atributos booleanos sólo pueden tomar un valor legal: el propio nombre del atributo (p.ej., `selected="selected"`).

Este ejemplo define el atributo [selected](#) como un atributo booleano.

```
selected      (selected)  #IMPLIED  -- opción preseleccionada --
```

El atributo se iguala a "verdadero" si aparece en la etiqueta inicial del elemento:

```
<OPTION selected="selected">
...contenidos...
</OPTION>
```

En HTML, los atributos booleanos pueden aparecer en forma minimizada: sólo aparece el **valor** del atributo en la etiqueta inicial del elemento. Así, puede establecerse [selected](#) escribiendo:

```
<OPTION selected>
```

en lugar de:

```
<OPTION selected="selected">
```

Los autores deberían tener en cuenta que muchos agentes de usuario **sólo** reconocen la forma minimizada de los atributos booleanos, y no la forma completa.

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

4 Conformidad: requisitos y recomendaciones

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Definiciones](#)
2. [SGML](#)
3. [El tipo de contenido text/html](#)

En esta sección comenzamos la especificación de HTML 4, empezando por las obligaciones de y entre autores, documentos, usuarios y agentes de usuario.

Las palabras clave "DEBE", "NO DEBE", "REQUERIDO", "DEBERÁ", "NO DEBERÁ", "DEBERÍA", "NO DEBERÍA", "RECOMENDADO", "PUEDE", y "OPCIONAL" de este documento deben interpretarse como se describe en [\[RFC2119\]](#). Sin embargo, para facilitar la lectura, estas palabras no aparecen en mayúsculas en esta especificación.

En ocasiones, los autores de esta especificación recomiendan buena práctica a autores y agentes de usuario. Estas recomendaciones no son normativas, y la conformidad con esta especificación no depende de su ejecución. Estas recomendaciones contienen las expresiones "Recomendamos que...", "Esta especificación recomienda...", o similares.

4.1 Definiciones

Documento HTML

Un documento HTML es un documento SGML que se atiene a las restricciones de esta especificación.

Autor

Un autor es una persona o programa que escribe o genera documentos HTML. Una herramienta de creación es un caso especial de autor, a saber, un programa que genera HTML.

Recomendamos a los autores que escriban documentos que sean conformes con el [DTD estricto](#) mejor que con otros DTDs definidos por esta especificación. Consulte la sección sobre [información sobre versiones](#) para más detalles sobre los DTDs definidos en HTML 4.

Usuario

Un usuario es una persona que interactúa con un agente de usuario para ver, oír, o usar de

cualquier otra manera un documento HTML representado.

Agente de usuario HTML

Un agente de usuario HTML es cualquier dispositivo que interprete documentos HTML. Los agentes de usuario incluyen navegadores visuales (de texto o gráficos), navegadores no visuales (audio, Braille), robots de búsqueda, proxies, etc.

Un **agente de usuario conforme** con HTML 4 es el que cumple las condiciones obligatorias ("debe") establecidas en esta especificación, incluyendo los puntos siguientes:

- Un agente de usuario debería evitar imponer límites arbitrarios a la longitud de los valores literales de los atributos (ver la sección sobre capacidades en la [Declaración de SGML](#)). Para información preliminar sobre atributos SGML, consúltese la sección sobre [definiciones de atributos](#).
- Un agente de usuario debe asegurarse de que la representación no cambia por la presencia o la ausencia de etiquetas iniciales o finales cuando el DTD HTML indique que éstas son opcionales. Véase la sección sobre [definiciones de elementos](#) para información preliminar sobre elementos SGML.
- Por razones de compatibilidad con versiones anteriores, recomendamos que las herramientas que interpreten HTML 4 sigan soportando HTML 3.2 (ver [\[HTML32\]](#)) y HTML 2.0 (ver [\[RFC1866\]](#)).

Condiciones de error

Esta especificación no define cómo deben tratar los agentes de usuario las condiciones de error generales, incluyendo cómo se comportan los agentes de usuario cuando encuentran elementos, atributos, valores de atributos o entidades no especificadas en este documento.

Sin embargo, para un comportamiento recomendado de manejo de errores, consúltense las [notas sobre documentos no válidos](#).

Desaprobado

Un elemento o atributo desaprobado es aquel que ha quedado anticuado por la presencia de estructuras nuevas. Los elementos desaprobados se definen en el manual de referencia en los lugares apropiados, pero claramente marcados como desaprobados. Los elementos desaprobados pueden declararse obsoletos en versiones futuras de HTML.

Los agentes de usuario deberían seguir dando soporte a los elementos desaprobados por razones de compatibilidad con versiones anteriores.

Las definiciones de elementos y atributos indican claramente cuáles son desaprobados.

Esta especificación incluye ejemplos que ilustran cómo evitar el uso de elementos desaprobados. En la mayoría de los casos estos ejemplos dependen de que el agente de usuario soporte hojas de estilo. En general, los autores deberían usar hojas de estilo para lograr efectos estilísticos y de formato de presentación en lugar de utilizar atributos presentacionales de HTML. Los atributos de presentación de HTML han sido desaprobados cuando existen alternativas con hojas de estilo (véase, por ejemplo, [\[CSS1\]](#)).

Obsoleto

Un elemento o atributo obsoleto es aquél para el cual no hay garantía de soporte por parte de un agente de usuario. Los elementos obsoletos han dejado de estar definidos en la especificación, pero se enumeran por motivos históricos en la [sección de cambios](#) del manual de referencia.

4.2 SGML

El HTML 4 es una aplicación SGML conforme al Estándar Internacional ISO 8879 -- Standard Generalized Markup Language SGML (definido en [\[ISO8879\]](#)).

Los ejemplos del texto son conformes a la [definición del tipo de documento estricta](#) a menos que el ejemplo en cuestión se refiera a elementos o atributos definidos sólo por la [definición del tipo de documento transicional](#) o por la [definición del tipo de documento con marcos](#). En pos de la brevedad, la mayoría de los ejemplos de esta especificación no comienzan con la [declaración del tipo de documento](#) que es obligatoria al principio de todo documento HTML.

Los fragmentos de DTD de las definiciones de los elementos están sacados de la [definición del tipo de documento estricta](#) excepto para los elementos relacionados con marcos.

Consulte la sección sobre [información sobre versiones de HTML](#) para detalles sobre cuándo usar el DTD estricto, el transicional o el de documentos con marcos.

Los comentarios que aparecen en el [DTD de HTML 4](#) no tienen valor normativo; sólo son informativos.

Los agentes de usuario no deben representar instrucciones de procesamiento SGML (p.ej., `<?full volume>`) ni comentarios. Para más información sobre éstas y otras características de SGML que pueden ser legales en HTML pero que no son generalmente soportadas por agentes de usuario, consulte la sección sobre [características de SGML con soporte limitado](#).

4.3 El tipo de contenido text/html

Los documentos HTML se envían por Internet como una secuencia de bytes acompañada de información de codificación (descrita en la sección sobre [codificaciones de caracteres](#)). La estructura de la transmisión, denominada entidad de mensaje, está definida por [\[RFC2045\]](#) y [\[RFC2616\]](#). Una entidad de mensaje con un [tipo de contenido](#) "text/html" representa un documento HTML.

El tipo de contenido para los documentos HTML se define como sigue:

Nombre del tipo de contenido (Content type name):

text

Nombre del subtipo de contenido (Content subtype name):

html

Parámetros obligatorios:

ninguno

Parámetros opcionales:

charset

Consideraciones de codificación:

se permite cualquier codificación

Consideraciones de seguridad:

Véanse las [notas sobre seguridad](#)

El parámetro opcional "charset" se refiere a la [codificación de caracteres](#) usada para representar el documento HTML como una secuencia de bytes. Los valores legales para este parámetro están definidos en la sección sobre [codificaciones de caracteres](#). Aunque este parámetro es opcional, recomendamos que esté siempre presente.

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

5 Representación de Documentos HTML

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [El Conjunto de Caracteres del Documento](#)
2. [Codificaciones de caracteres](#)
 1. [Elección de una codificación](#)
 - [Notas sobre codificaciones específicas](#)
 2. [Especificación de la codificación de caracteres](#)
3. [Referencias de caracteres](#)
 1. [Referencias numéricas de caracteres](#)
 2. [Referencias a entidades de caracteres](#)
4. [Caracteres no representables](#)

En este capítulo hablamos sobre cómo se representan los documentos HTML en una computadora y en Internet.

La sección sobre el [conjunto de caracteres del documento](#) trata la cuestión de qué *caracteres* abstractos pueden formar parte de un documento HTML. Estos caracteres incluyen la letra latina "A", la letra cirílica "I", el carácter chino que significa "agua", etc.

La sección sobre las [codificaciones de caracteres](#) trata la cuestión de cómo pueden *representarse* esos caracteres en un fichero o cuando se transmiten por Internet. Como algunas codificaciones de caracteres no pueden representar directamente todos los caracteres que un autor puede querer incluir en un documento, HTML ofrece otros mecanismos, llamados [referencias de caracteres](#), para hacer referencia a cualquier carácter.

Al haber un gran número de caracteres en todos los lenguajes humanos, y una gran variedad de formas de representar estos caracteres, debe tenerse un cuidado especial para que los documentos puedan ser entendidos por agentes de usuario de todo el mundo.

5.1 El Conjunto de Caracteres del Documento

Para promover la interoperabilidad, SGML requiere que cada aplicación (incluyendo al HTML) especifique su conjunto de caracteres del documento. Un conjunto de caracteres del documento consiste en:

- Un Repertorio : Un conjunto de caracteres abstractos, tales como la letra latina "A", la letra cirílica "I", el carácter chino que significa "agua", etc.
- Posiciones de códigos: Un conjunto de referencias enteras a los caracteres del repertorio.

Cada documento SGML (incluidos los documentos HTML) es una secuencia de caracteres del repertorio. Los sistemas informáticos identifican cada carácter según la posición de su código; por ejemplo, en el conjunto de caracteres ASCII, las posiciones de los códigos 65, 66 y 67 se refieren a los caracteres 'A', 'B' y 'C' respectivamente.

El conjunto de caracteres ASCII no es suficiente para un sistema de información global como la Web, de modo que HTML usa un conjunto de caracteres mucho más completo llamado *Conjunto Universal de Caracteres (Universal Character Set, UCS)*, definido en [\[ISO10646\]](#). Este estándar define un repertorio de miles de caracteres usados por comunidades de todo el mundo.

El conjunto de caracteres definido en [\[ISO10646\]](#) es equivalente carácter por carácter a Unicode ([\[UNICODE\]](#)). Ambos estándares se actualizan de vez en cuando con nuevos caracteres, y las correcciones deberían consultarse en los sitios web respectivos. En la especificación actual, se utiliza "[ISO10646]" para hacer referencia al conjunto de caracteres del documento, mientras que "[UNICODE]" se reserva para hacer referencia al [algoritmo de texto bidireccional](#) de Unicode.

El conjunto de caracteres del documento no es suficiente, sin embargo, para permitir a los agentes de usuario interpretar correctamente los documentos HTML del modo en que éstos se intercambian normalmente: codificados como una secuencia de bytes en un fichero o durante una transmisión en red. Los agentes de usuario deben conocer además la [codificación de caracteres](#) específica que fue usada para transformar el flujo de caracteres del documento en un flujo de bytes.

5.2 Codificaciones de caracteres

Lo que en esta especificación se denomina codificación de caracteres se conoce con nombres diferentes en otras especificaciones (lo cual puede causar cierta confusión). Sin embargo, el concepto es básicamente el mismo en todo Internet. Asimismo, las cabeceras, atributos y parámetros del protocolo que se refieren a la codificación de caracteres comparten todos el mismo nombre -- "charset" -- y usan los mismos valores del registro [\[IANA\]](#) (ver [\[CHARSETS\]](#) para una lista completa).

El parámetro "charset" identifica una codificación de caracteres, que es un método para convertir una secuencia de bytes en una secuencia de caracteres. Esta conversión se adapta naturalmente al modo de funcionamiento de la Web: los servidores envían los documentos HTML a los agentes de usuario como un flujo de bytes; los agentes de usuario los interpretan como un flujo de caracteres. El método de conversión puede ir desde una correspondencia simple uno a uno hasta complejos algoritmos o procesos de intercambio.

Una técnica simple de codificación uno a uno no es suficiente para cadenas de texto con un repertorio

tan amplio como el de [\[ISO10646\]](#). Hay varias codificaciones diferentes de partes de [\[ISO10646\]](#) además de codificaciones del conjunto completo de caracteres (como por ejemplo UCS-4).

5.2.1 Elección de una codificación

Las herramientas de creación (p.ej., editores de texto) pueden codificar documentos HTML con una codificación de caracteres de su elección, y dicha elección depende en gran medida de las convenciones usadas por el software del sistema. Estas herramientas pueden emplear cualquier codificación que cubra convenientemente la mayor parte de los caracteres contenidos en el documento, suponiendo que la codificación esté [etiquetada correctamente](#). Algunos caracteres se saldrán fuera de esta codificación, pero aún pueden seguir siendo representados mediante [referencias de caracteres](#). Éstas siempre hacen referencia al conjunto de caracteres del documento, no a la codificación de caracteres.

Los servidores y los proxies pueden cambiar la codificación de caracteres (la llamada *transcodificación*) en tiempo real para atender a las demandas de los agentes de usuario (véase la sección 14.2 sobre [\[RFC2616\]](#), el encabezado de petición HTTP "Accept-Charset"). Los servidores y los proxies no tienen por qué servir un documento con una codificación de caracteres que cubra el conjunto completo de caracteres del documento.

Las codificaciones de caracteres usadas normalmente en la Web incluyen la ISO-8859-1 (también llamada "Latin-1"; utilizable para la mayor parte de los idiomas de Europa occidental), la ISO-8859-5 (que soporta el alfabeto cirílico), la SHIFT_JIS (una codificación japonesa), la EUC-JP (otra codificación japonesa), y la UTF-8 (una codificación de ISO 10646 que utiliza un número diferente de bytes para distintos caracteres). En los nombres de las codificaciones no se distingue entre mayúsculas y minúsculas, así que por ejemplo "SHIFT_JIS", "Shift_JIS", y "shift_jis" son equivalentes.

Esta especificación no dicta qué codificaciones de caracteres debe soportar un agente de usuario.

Los [agentes de usuario conformes](#) deben transformar correctamente a ISO 10646 todos los caracteres de todas las codificaciones de caracteres que reconozcan (o deben comportarse como si lo hicieran).

Notas sobre codificaciones específicas

Cuando un texto HTML se transmite en UTF-16 (charset=UTF-16), los datos de texto deberían transmitirse con el orden de bytes de la red ("big-endian", el byte de mayor peso primero) de acuerdo con [\[ISO10646\]](#), Sección 6.3 y [\[UNICODE\]](#), cláusula C3, página 3-1.

Además, para maximizar la probabilidad de una interpretación correcta, se recomienda que los documentos transmitidos como UTF-16 empiecen siempre con un carácter de ESPACIO NO SEPARABLE DE ANCHURA ZERO (ZERO-WIDTH NON-BREAKING SPACE, hexadecimal FEFF, también llamado Byte Order Mark, BOM) que, al invertirse los bytes, se convierte en el hexadecimal FFFE, un carácter que nunca será asignado. De este modo, un agente de usuario que reciba un hexadecimal FFFE como los primeros bytes de un texto sabrá que los bytes deben ser invertidos para el resto del texto.

El formato de transformación UTF-1 de [\[ISO10646\]](#) (registrado por IANA como ISO-10646-UTF-1), no debería utilizarse. Para información sobre ISO 8859-8 y el algoritmo bidireccional, consúltese la sección sobre [bidireccionalidad y codificación de caracteres](#).

5.2.2 Especificación de la codificación de caracteres

¿Cómo determina un servidor qué codificación de caracteres se aplica a un documento que está sirviendo? Algunos servidores examinan los primeros bytes del documento, o comprueban una base de datos de ficheros conocidos y sus codificaciones. Muchos servidores modernos dan a los administradores de páginas web más control sobre la configuración de la codificación que los servidores antiguos. Los administradores de páginas deberían usar estos mecanismos para enviar un parámetro "charset" siempre que sea posible, pero con cuidado de no identificar un documento con un valor del parámetro "charset" equivocado.

¿Cómo sabe un agente de usuario qué codificación de caracteres ha sido utilizada? El servidor debería proporcionar esta información. La manera más directa que tiene un servidor para informar al agente de usuario sobre la codificación de caracteres del documento es utilizando el parámetro "charset" del campo "Content-Type" de encabezado del protocolo HTTP ([\[RFC2616\]](#), secciones 3.4 y 14.17). Por ejemplo, el siguiente encabezado HTTP anuncia que la codificación de caracteres es EUC-JP:

```
Content-Type: text/html; charset=EUC-JP
```

Consulte la sección sobre [conformidad](#) para la definición de [text/html](#).

El protocolo HTTP ([\[RFC2616\]](#), sección 3.7.1) menciona ISO-8859-1 como la codificación de caracteres por defecto cuando el parámetro "charset" esté ausente del campo de encabezado "Content-Type". En la práctica, esta recomendación ha resultado ser inútil, ya que algunos servidores no permiten que se envíe un parámetro "charset", y otros pueden no estar configurados para enviar el parámetro. Por lo tanto, los agentes de usuario no deben suponer ningún valor por defecto para el parámetro "charset".

Para tener en cuenta las limitaciones de los servidores o de sus configuraciones, los documentos HTML pueden incluir información explícita sobre la codificación de caracteres del documento; el elemento [META](#) puede utilizarse para proporcionar esta información a los agentes de usuario.

Por ejemplo, para especificar que la codificación de caracteres del documento actual es "EUC-JP", un documento debería incluir la siguiente declaración [META](#):

```
<META http-equiv="Content-Type" content="text/html; charset=EUC-JP">
```

La declaración [META](#) sólo debe utilizarse cuando la codificación de caracteres se organice de modo que los bytes con valores ASCII correspondan a caracteres ASCII (al menos hasta que se interprete el elemento [META](#)). Las declaraciones [META](#) deberían aparecer lo antes posible dentro del elemento [HEAD](#).

Para aquellos casos en que ni el protocolo HTTP ni el elemento [META](#) proporcionen información sobre la codificación de caracteres de un documento, HTML también proporciona el atributo [charset](#) para varios elementos. Combinando todos estos mecanismos, un autor puede asegurarse bastante de que cuando el usuario reciba un recurso, el agente de usuario reconocerá la codificación de caracteres.

En resumen, los agentes de usuario conformes deben tener en cuenta las siguientes prioridades a la hora de determinar la codificación de caracteres de un documento (de prioridad más alta a más baja):

1. Un parámetro "charset" HTTP en un campo "Content-Type".

2. Una declaración [META](#) con "http-equiv" establecido en "Content-Type" y un valor dado para "charset".
3. El atributo [charset](#) establecido en un elemento que designe un recurso externo.

Además de esta lista de prioridades, el agente de usuario puede usar una heurística y las preferencias del usuario. Por ejemplo, un agente de usuario puede utilizar una heurística para distinguir entre las varias codificaciones utilizadas para textos en japonés. Además, los agentes de usuario tienen normalmente una codificación de caracteres local por defecto y definible por el usuario, que pueden aplicar en ausencia de otros indicadores.

Los agentes de usuario pueden proporcionar un mecanismo que permita a los usuarios invalidar información incorrecta sobre el "charset". Sin embargo, si un agente de usuario ofrece este mecanismo, sólo debería hacerlo para navegar y no para editar, para evitar la creación de páginas web con un parámetro "charset" incorrecto.

Nota. Si, para una aplicación específica, es absolutamente necesario referirse a caracteres que estén fuera de [\[ISO10646\]](#), estos caracteres deberían asignarse a una zona privada para evitar conflictos con versiones presentes o futuras del estándar. De todos modos, esto no se recomienda en absoluto, por motivos de portabilidad.

5.3 Referencias de caracteres

Una codificación de caracteres dada puede no ser capaz de expresar todos los caracteres del conjunto de caracteres del documento. Para tales codificaciones, o cuando las configuraciones de hardware o de software no permitan a los usuarios introducir algunos caracteres del documento directamente, los autores pueden usar referencias de caracteres SGML. Las referencias de caracteres son un mecanismo independiente de la codificación de caracteres para introducir cualquier carácter del conjunto de caracteres del documento.

Las referencias de caracteres en HTML pueden aparecer de dos maneras:

- Referencias numéricas de caracteres (bien decimales, bien hexadecimales).
- Referencias a entidades de caracteres.

Las referencias de caracteres dentro de comentarios no tienen significado especial; sólo son datos para comentar.

Nota. HTML proporciona otros modos de presentar datos de caracteres, en particular [imágenes en línea](#).

Nota. En SGML, es posible eliminar el ";" final de una referencia de caracteres en algunos casos (p.ej., en un salto de línea o inmediatamente antes de una etiqueta). En otras circunstancias no puede ser eliminado (p.ej., en medio de una palabra). Recomendamos encarecidamente utilizar el ";" en todos los casos para evitar problemas con agentes de usuario que requieran que este carácter esté presente.

5.3.1 Referencias numéricas de caracteres

Las referencias numéricas de caracteres especifican la [posición del código](#) de un carácter en el conjunto de caracteres del documento. Las referencias numéricas de caracteres pueden tener dos

formas:

- La sintaxis "&#D;", donde *D* es un número decimal, se refiere al carácter de ISO 10646 con el número decimal *D*.
- La sintaxis "&#xH;" o "&#XH;", donde *H* es un número hexadecimal, se refiere al carácter de ISO 10646 con el número hexadecimal *H*. Para los números hexadecimales de referencias de caracteres numéricas no se distingue entre mayúsculas y minúsculas.

A continuación se muestran algunos ejemplos de referencias numéricas de caracteres:

- å (en decimal) representa la letra "a" con un circulito encima (usada, por ejemplo, en noruego).
- å (en hexadecimal) representa el mismo carácter.
- å (en hexadecimal) representa también el mismo carácter.
- И (en decimal) representa la letra mayúscula cirílica "I".
- 水 (en hexadecimal) representa el carácter chino para "agua".

Nota. Aunque la representación hexadecimal no se define en [\[ISO8879\]](#), se espera que lo esté en la revisión, como se describe en [\[WEBSGML\]](#). Esta convención es particularmente útil ya que los estándares de caracteres suelen utilizar representaciones hexadecimales.

5.3.2 Referencias a entidades de caracteres

Para que los autores tengan una manera más intuitiva de referirse a caracteres del conjunto de caracteres del documento, HTML ofrece un conjunto de referencias a entidades de caracteres. Las referencias a entidades de caracteres utilizan nombres simbólicos para que los autores no tengan que recordar [posiciones de código](#). Por ejemplo, la referencia a la entidad de caracteres å se refiere a la letra minúscula "a" con un anillo ("ring") encima; "å" es más fácil de recordar que å.

HTML 4 no define una referencia a entidades de caracteres para cada carácter del conjunto de caracteres del documento. Por ejemplo, no hay referencia a entidades de caracteres para la letra mayúscula cirílica "I". Consulte la [lista completa de referencias de caracteres](#) definidas en HTML 4.

Las referencias a entidades de caracteres sí distinguen entre mayúsculas y minúsculas. Así, Å se refiere a un carácter diferente (A mayúscula con anillo) que å (a minúscula con anillo).

Hay cuatro referencias a entidades de caracteres que merecen mención especial, ya que se usan frecuentemente para transformar ciertos caracteres especiales en secuencias de escape:

- "<" representa el signo <.
- ">" representa el signo >.
- "&" representa el signo &.
- """ representa el signo ".

Los autores que quieran poner el carácter "<" en el texto deberían usar "<" (ASCII decimal 60) para evitar la posible confusión con el comienzo de una etiqueta (delimitador de apertura de etiqueta inicial). Análogamente, los autores deberían usar ">" (ASCII decimal 62) en el texto en lugar de ">" para evitar problemas con agentes de usuario antiguos que lo interpretan incorrectamente como el final de una etiqueta (delimitador de cierre de una etiqueta) cuando aparece dentro de valores de atributos entrecomillados.

Los autores deberían usar "&" (ASCII decimal 38) en lugar de "&" para evitar la confusión con el comienzo de una referencia de caracteres (delimitador de apertura de una referencia a entidades). Los autores deberían usar también "&" en valores de atributos, ya que las referencias de caracteres están permitidas dentro de valores de atributos [CDATA](#).

Algunos autores usan la referencia a entidades de caracteres """ para codificar las comillas dobles ("), ya que este carácter puede utilizarse para delimitar los valores de los atributos.

5.4 Caracteres no representables

Un agente de usuario puede no ser capaz de representar correctamente todos los caracteres de un documento, por ejemplo, porque el agente de usuario carezca de una fuente apropiada, porque un carácter tenga un valor que no pueda expresarse con la codificación de caracteres interna del agente de usuario, etc.

Al haber tantas cosas diferentes que pueden hacerse en estos casos, este documento no prescribe ningún comportamiento por defecto. Dependiendo de la implementación, los caracteres no representables pueden también ser tratados por el sistema de presentación subyacente y no por la propia aplicación. En ausencia de un comportamiento más sofisticado, por ejemplo a medida de las necesidades de un script o lenguaje particular, recomendamos el siguiente comportamiento a los agentes de usuario:

1. Adoptar un mecanismo claramente visible, pero no molesto, para alertar al usuario sobre los recursos ausentes.
2. Si los caracteres no disponibles se presentan usando su representación numérica, usar la forma hexadecimal (no la decimal), ya que ésta es la forma utilizada en los estándares de conjuntos de caracteres.

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

6 Tipos de datos básicos de HTML

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Mayúsculas y minúsculas](#)
2. [Tipos básicos de SGML](#)
3. [Cadenas de texto](#)
4. [URIs](#)
5. [Colores](#)
 1. [Notas sobre el uso de colores](#)
6. [Longitudes](#)
7. [Tipos de contenido \(tipos MIME\)](#)
8. [Códigos de idioma](#)
9. [Codificaciones de caracteres](#)
10. [Caracteres individuales](#)
11. [Fechas y horas](#)
12. [Tipos de vínculos](#)
13. [Descriptores de medios](#)
14. [Datos de scripts](#)
15. [Datos de hojas de estilo](#)
16. [Nombres de marcos destino](#)

Esta sección de la especificación describe los tipos básicos de datos que pueden aparecer como contenido de un elemento o valor de un atributo.

Para información preliminar sobre cómo leer el DTD HTML, consulte el [tutorial de SGML](#).

6.1 Mayúsculas y minúsculas

En la definición de cada atributo se incluye información sobre la importancia que tiene para sus valores la distinción entre mayúsculas y minúsculas. Esta información se presenta de acuerdo con la siguiente leyenda:

CS

El valor distingue entre mayúsculas y minúsculas (es decir, los agentes de usuario interpretan "a" y "A" de manera diferente).

CI

El valor no distingue entre mayúsculas y minúsculas (es decir, los agentes de usuario interpretan "a" igual que "A").

CN

No puede haber diferencias entre mayúsculas y minúsculas, p.ej., porque es un número o un carácter del conjunto de caracteres del documento.

CA

La definición del elemento o atributo ya da la información sobre la distinción entre mayúsculas y minúsculas.

CT

Consultar la definición del tipo para detalles sobre la distinción entre mayúsculas y minúsculas.

Si el valor de un atributo es una lista, estas claves se aplican a todos los valores de la lista, a menos que se indique lo contrario.

6.2 Tipos básicos de SGML

La [definición del tipo de documento](#) especifica la sintaxis de los contenidos de los elementos y los valores de los atributos de HTML por medio de palabras (tokens) SGML (p.ej., PCDATA, CDATA, NAME, ID, etc.). Véase [ISO8879](#) para sus definiciones completas. Lo que sigue es un resumen de la información fundamental:

- **CDATA** es una secuencia de caracteres tomados del conjunto de caracteres del documento y puede incluir entidades de caracteres. Los agentes de usuario deberían interpretar los valores de atributos como sigue:
 - Reemplazar las entidades de caracteres con caracteres,
 - Ignorar los avances de línea,
 - Reemplazar cada retorno de carro o tabulación con un espacio simple.

Los agentes de usuario pueden ignorar el espacio en blanco inicial o final de valores de atributos CDATA (p.ej., " mivalor " puede interpretarse como "mivalor"). Los autores no deberían declarar valores de atributos con espacios en blanco al principio o al final.

Para algunos atributos de HTML 4 con valores de atributo CDATA, la especificación impone restricciones más fuertes al conjunto de valores legales para el atributo, restricciones que no pueden expresarse por el DTD.

Aunque los elementos [STYLE](#) y [SCRIPT](#) usan CDATA para su modelo de datos, para estos elementos los agentes de usuario deben tratar CDATA de manera diferente. El código y las entidades deben tratarse como texto sin formato y pasarse a la aplicación tal y como están. La primera aparición de la secuencia de caracteres "<" (delimitador de apertura de etiqueta final) se trata como si finalizara el contenido del elemento. En documentos válidos, esta sería la etiqueta final del elemento.

- Las palabras **ID** y **NAME** deben comenzar con una letra ([A-Za-z]) que puede estar seguida por

un número cualquiera de letras, dígitos, ([0-9]), guiones ("-"), subguiones ("_"), dos puntos (":"), y puntos (".").

- **IDREF** e **IDREFS** son referencias a palabras ID definidas por otros atributos. IDREF es una palabra simple e IDREFS es una lista de palabras separadas por espacios.
- Las palabras **NUMBER** deben contener al menos un dígito ([0-9]).

6.3 Cadenas de texto

Ciertos atributos ([%Text;](#) en el DTD) toman por valor un texto que se supone debe ser "legible por humanos". Para información preliminar sobre atributos, consulte la [parte del tutorial sobre atributos](#).

6.4 URIs

Esta especificación utiliza el término URI tal y como se define en [\[URI\]](#) (ver también [\[RFC1630\]](#)).

Obsérvese que los URIs incluyen a los URLs (según se definen en [\[RFC1738\]](#) y [\[RFC1808\]](#)).

Los URIs relativos se convierten en URIs completos a partir de un URI base. La [\[RFC1808\]](#), sección 3, define el algoritmo normativo para este proceso. Para más información sobre URIs base, consulte la sección sobre [URIs base](#) en el capítulo sobre [vínculos](#).

Los URIs se representan en el DTD con la entidad paramétrica [%URI;](#).









En general los URIs hacen distinción entre [mayúsculas y minúsculas](#). Puede haber URIs, o partes de URIs, donde la diferencia no importe (p.ej., nombres de máquinas), pero la identificación de éstas puede no ser fácil. Los usuarios deberían considerar que en los URIs se distingue siempre entre mayúsculas y minúsculas (para quedarse del lado de la seguridad).









Consulte el apéndice para información sobre [caracteres no ASCII en valores de atributos URI](#).

6.5 Colores

El tipo de valor de atributo "color" ([%Color;](#)) se refiere a las definiciones de colores según se especifican en [\[SRGB\]](#). Un valor de color puede ser o bien un número hexadecimal (anteponiendo un signo "#") o uno de los siguientes dieciséis nombres de colores. En los nombres de colores no se distingue entre [mayúsculas y minúsculas](#).

Nombres de colores y valores sRGB

	Black = "#000000" (Negro)		Green = "#008000" (Verde)
	Silver = "#C0C0C0" (Plateado)		Lime = "#00FF00" (Verde lima)
	Gray = "#808080" (Gris)		Olive = "#808000" (Verde oliva)
	White = "#FFFFFF" (Blanco)		Yellow = "#FFFF00" (Amarillo)

	Maroon = "#800000" (Marrón)		Navy = "#000080" (Azul marino)
	Red = "#FF0000" (Rojo)		Blue = "#0000FF" (Azul)
	Purple = "#800080" (Púrpura)		Teal = "#008080" (Azul verdoso)
	Fuchsia = "#FF00FF" (Fucsia)		Aqua = "#00FFFF" (Celeste)

Así, los valores de color "#800080" y "Purple" se refieren ambos al color púrpura.

6.5.1 Notas sobre el uso de colores

Aunque los colores pueden añadir cantidades significativas de información a los documentos y hacerlos más legibles, considere las siguientes pautas cuando incluya color en sus documentos:

- El uso de elementos y atributos HTML para especificar colores está [desaprobado](#). Se aconseja utilizar [hojas de estilo](#) en su lugar.
- No utilice combinaciones de colores que puedan causar problemas a personas con dificultades para distinguir colores.
- Si usa una imagen de fondo o establece un color de fondo, asegúrese de que establece también los colores de los distintos textos.
- Los colores especificados con los elementos [BODY](#) y [FONT](#) y con [bgcolor](#) en las tablas tienen apariencias diferentes en plataformas diferentes (p.ej., estaciones de trabajo, Macs, Windows, y paneles LCD frente a CRTs), por lo que no debería basar un efecto específico completamente en ellos. En el futuro, el soporte del modelo de color [\[SRGB\]](#) junto con los perfiles de color ICC debería mitigar este problema.
- Cuando sea posible, adopte las convenciones comunes para evitar confusiones.

6.6 Longitudes

HTML especifica tres tipos de valores de longitud para los atributos:

1. **Píxeles (Pixels)**: El valor ([%Pixels](#); en el DTD) es un entero que representa un número de píxeles del lienzo (papel, pantalla). Así, el valor "50" significa cincuenta píxeles. Para información normativa sobre la definición de un píxel, consulte [\[CSS1\]](#).
2. **Longitud (Length)**: El valor ([%Length](#); en el DTD) puede ser o bien un %Pixel; o bien un porcentaje del espacio horizontal o vertical disponible. Así, el valor "50%" significa la mitad del espacio disponible.
3. **Multilongitud (MultiLength)**: El valor ([%MultiLength](#); en el DTD) puede ser un %Length; o una longitud relativa. Una longitud relativa tiene la forma "i*", donde "i" es un entero. Los agentes de usuario, cuando reparten espacio entre los elementos que compiten por ese espacio, adjudican primero las longitudes en píxeles y en porcentajes, y después dividen el espacio sobrante entre las longitudes relativas. Cada longitud relativa recibe una porción del espacio disponible que es proporcional al entero que precede al "*". El valor "*" es equivalente a "1*".

Así, si hay disponibles 60 píxeles de espacio después de haber adjudicado el agente de usuario el espacio en píxeles y en porcentajes, y las longitudes relativas que deben asignarse son "1*", "2*" y "3*", se asignarán 10 píxeles al 1*, 20 píxeles al 2* y 30 píxeles al 3*.

Los valores de longitud son neutros frente a la distinción entre [mayúsculas y minúsculas](#).

6.7 Tipos de contenido (tipos MIME)

Nota. Un "media type" o "tipo de medios" (definido en [\[RFC2045\]](#) y [\[RFC2046\]](#)) especifica la naturaleza de un recurso vinculado. Esta especificación emplea el término "content type" ("tipo de contenido") en lugar de "media type" (tipo de medios) de acuerdo con el uso actual. Además de esto, en esta especificación, "media type" se puede referir a los [medios](#) en que un agente de usuario representa un documento.

Este tipo está representado en el DTD por [%ContentType;](#)

Los tipos de contenido no hacen distinción entre [masyúsculas y minúsculas](#).

Ejemplos de tipos de contenido son "text/html", "image/png", "image/gif", "video/mpeg", "text/css", y "audio/basic". Para la lista de tipos MIME registrados, consulte [\[MIMETYPES\]](#).

6.8 Códigos de idioma

El valor de los atributos cuyo tipo es un código de idioma ([%LanguageCode;](#) en el DTD) se refiere a un código de idioma de los especificados en [\[RFC1766\]](#), sección 2. Para información sobre especificación de códigos de idioma en HTML, consulte la sección sobre [códigos de idioma](#). No se permite espacio en blanco dentro del código de idioma.

Los códigos de idioma no distinguen entre [mayúsculas y minúsculas](#).

6.9 Codificaciones de caracteres

Los atributos "charset" ([%Charset;](#) en el DTD) se refieren a una codificación de caracteres según se describe en la sección sobre [codificaciones de caracteres](#). Los valores deben ser cadenas (p.ej., "euc-jp") del registro IANA (ver [\[CHARSETS\]](#) para una lista completa).

Los nombres de codificaciones de caracteres no distinguen entre [mayúsculas y minúsculas](#).

Los agentes de usuario deben seguir los pasos descritos en la sección sobre [especificación de codificaciones de caracteres](#) para determinar la codificación de caracteres de un recurso externo.

6.10 Caracteres individuales

Ciertos atributos necesitan un carácter individual del [conjunto de caracteres del documento](#). Estos atributos son del tipo [%Character;](#) en el DTD.

Los caracteres individuales pueden especificarse mediante [referencias de caracteres](#) (p.ej., "&").

6.11 Fechas y horas

La [\[ISO8601\]](#) permite muchas opciones y variaciones en la representación de fechas y horas. La especificación actual usa uno de los formatos descritos en el perfil [\[DATETIME\]](#) para la definición de cadenas fecha/hora legales ([%Datetime](#); en el DTD).

El formato es:

```
AAAA-MM-DDThh:mm:ssDZH
```

donde:

```
AAAA = año con cuatro dígitos
MM    = mes con dos dígitos (01=enero, etc.)
DD    = día del mes con dos dígitos (de 01 a 31)
hh    = hora con dos dígitos (de 00 a 23) (NO se permite am/pm)
mm    = minuto con dos dígitos (00 a 59)
ss    = segundo con dos dígitos (00 a 59)
DZH   = designador de zona horaria
```

El designador de zona horaria es uno de los siguientes:

Z

indica UTC (Coordinated Universal Time, Tiempo Universal Coordinado). La "Z" debe ser mayúscula.

+hh:mm

indica que la hora es una hora local que está hh horas y mm minutos por delante del UTC.

-hh:mm

indica que la hora es una hora local que está hh horas y mm minutos por detrás del UTC.

Deben estar presentes exactamente los componentes mostrados aquí, y exactamente con esta puntuación. Obsérvese que la letra "T" aparece literalmente en la cadenas (debe ser mayúscula), para indicar el comienzo de la hora, según se especifica en [\[ISO8601\]](#)

Si una aplicación generadora no conoce la hora con precisión de segundos, puede usar el valor "00" para los segundos (y para los minutos y para las horas si es necesario).

Nota. [\[DATETIME\]](#) no trata la cuestión de segundos bisiestos.

6.12 Tipos de vínculos

Los autores pueden usar los siguientes tipos de vínculos reconocidos, enumerados aquí junto con sus interpretaciones convencionales. En el DTD, [%LinkTypes](#); se refiere a una lista de tipos de vínculos separados por espacios. No se permiten caracteres de espacio en blanco dentro de los tipos de vínculo.

Estos tipos de vínculo no hacen distinción entre [mayúsculas y minúsculas](#), es decir, "Alternate" tiene el mismo significado que "alternate".

Los agentes de usuario, los motores de búsqueda, etc. pueden interpretar estos tipos de vínculos de diferentes maneras. Por ejemplo, los agentes de usuario podrían proporcionar acceso a los documentos vinculados por medio de una barra de navegación.

Alternate

Designa una versión alternativa del documento en que aparece el vínculo. Cuando se usa con el atributo [hreflang](#), implica que hay una versión traducida del documento. Cuando se usa conjuntamente con el atributo [media](#), implica que hay una versión diseñada para un medio (o medios) diferentes.

Stylesheet

Se refiere a una hoja de estilo externa. Véase la sección sobre [hojas de estilo externas](#) para más detalles. Se usa junto al tipo de vínculo "Alternate" para ofrecer hojas de estilo alternativas seleccionables por el usuario.

Start

Se refiere al primer documento de un conjunto de documentos. Este tipo de vínculo dice a los motores de búsqueda qué documento es considerado por el autor como el punto de inicio de un conjunto.

Next

Se refiere al siguiente documento en una secuencia lineal de documentos. Los agentes de usuario pueden optar por precargar el documento marcado como "next", para reducir el tiempo de carga aparente.

Prev

Se refiere al documento anterior en una serie ordenada de documentos. Algunos agentes de usuario también soportan el sinónimo "Previous".

Contents

Se refiere a un documento que sirve como tabla de contenidos. Algunos agentes de usuario también soportan el sinónimo *ToC* (de "Table of Contents").

Index

Se refiere a un documento que proporciona un índice para el documento actual.

Glossary

Se refiere a un documento que proporciona un glosario de términos que pertenecen al documento actual.

Copyright

Se refiere al aviso de copyright del documento actual.

Chapter

Se refiere a un documento que actúa como capítulo en una colección de documentos.

Section

Se refiere a un documento que actúa como sección en una colección de documentos.

Subsection

Se refiere a un documento que actúa como subsección en una colección de documentos.

Appendix

Se refiere a un documento que actúa como apéndice en una colección de documentos.

Help

Se refiere a un documento que ofrece ayuda (más información, vínculos a otros recursos informativos, etc.)

Bookmark

Se refiere a una señal de lectura. Una señal de lectura (bookmark) es un vínculo a un punto de entrada importante dentro de un documento extenso. Se puede utilizar el atributo [title](#), por ejemplo, para dar un rótulo a la señal de lectura. Obsérvese que pueden definirse varias señales de lectura en cada documento.

Los autores pueden querer definir tipos de vínculos adicionales no descritos en esta especificación. Si hacen esto, deberían usar un [perfil \(profile\)](#) citando las convenciones usadas para definir los tipos de vínculos. Véase el atributo [profile](#) del elemento [HEAD](#) para más detalles.

Para más información sobre tipos de vínculos, consulte la sección sobre [vínculos en documentos HTML](#).

6.13 Descriptores de medios

La siguiente es una lista de los descriptores de medios reconocidos ([%MediaDesc](#); en el DTD).

screen

Para pantallas no paginadas de computadora.

tty

Para medios que utilicen una cuadrícula de caracteres de ancho fijo, como teletipos, terminales y dispositivos portátiles con posibilidades limitadas de representación.

tv

Para dispositivos tipo televisión (baja resolución, en color, desplazamiento limitado).

projection

Para proyectores.

handheld

Para dispositivos de mano (pantalla pequeña, monocromos, gráficos por mapas de bits, ancho de banda limitado).

print

Para material paginado, opaco, y para documentos que se ven en una pantalla en modo de presentación preliminar a la impresión.

braille

Para dispositivos táctiles braille.

aural

Para sintetizadores de voz.

all

Apropiado para todos los dispositivos.

En versiones futuras de HTML pueden introducirse nuevos valores, y podrían permitirse valores parametrizados. Para facilitar la introducción de estas extensiones, los agentes de usuario deben ser capaces de analizar el atributo [media](#) como sigue:

1. El valor es una lista de entradas separadas por comas. Por ejemplo,

```
media="screen, 3d-glasses, print and resolution > 90dpi"
```

se transforma en:

```
"screen"  
"3d-glasses"  
"print and resolution > 90dpi"
```

2. Cada entrada se trunca justo hasta antes del primer carácter que no sea una letra US ASCII [azA-Z] (ISO 10646 hex 41-5a, 61-7a), un dígito [0-9] (hex 30-39), o un guión (hex 2d). En el ejemplo, esto da:

```
"screen"  
"3d-glasses"  
"print"
```

3. Se hace entonces un emparejamiento con el conjunto de tipos de medios definidos arriba, teniendo en cuenta la diferencia entre [mayúsculas y minúsculas](#). Los agentes de usuario pueden ignorar las entradas que no resulten emparejadas. En el ejemplo nos quedamos con `screen` y con `print`.

Nota. Las hojas de estilo pueden incluir variaciones dependientes del medio en su interior (p.ej., la estructura CSS `@media`). En estos casos puede ser apropiado usar "[media=all](#)".

6.14 Datos de scripts

Los datos de scripts (`%Script`; en el [DTD](#)) pueden aparecer como contenido del elemento [SCRIPT](#) y como valor de los [atributos de eventos intrínsecos](#). Los agentes de usuario no deben evaluar los datos de script como código HTML, sino que deben pasarlos tal y como están como datos para un motor de scripts.

La importancia de la distinción entre mayúsculas y minúsculas en los datos de scripts depende del lenguaje de programación del script.

Obsérvese que los datos de scripts que estén contenidos en un elemento no pueden contener [referencias de caracteres](#), pero los datos de scripts que sean el valor de un atributo sí pueden contenerlas. El apéndice proporciona más información sobre la [especificación de datos no HTML](#).

6.15 Datos de hojas de estilo

Los datos de hojas de estilo (`%StyleSheet`; en el [DTD](#)) pueden aparecer como contenido de un elemento [STYLE](#) y como valor de un atributo [style](#). Los agentes de usuario no deben evaluar los datos de estilo como código HTML.

La importancia de la distinción entre mayúsculas y minúsculas en los datos de estilo depende del lenguaje de la hoja de estilo.

Obsérvese que los datos de hojas de estilo que estén contenidos en un elemento no pueden contener [referencias de caracteres](#), pero los datos de hojas de estilo que sean el valor de un atributo sí pueden contenerlas. El apéndice proporciona más información sobre la [especificación de datos no HTML](#).

6.16 Nombres de marcos destino

Excepto los nombres reservados enumerados a continuación, los nombres de marcos destino ([%FrameTarget](#); en el DTD) deben empezar con un carácter alfabético (a-zA-Z). Los agentes de usuario deberían ignorar otros nombres de destino.

Los siguientes nombres de destino están reservados y tienen significados especiales.

`_blank`

El agente de usuario debería cargar el documento designado en una ventana nueva y sin nombre.

`_self`

El agente de usuario debería cargar el documento en el mismo marco que el elemento que hace referencia a este destino.

`_parent`

El agente de usuario debería cargar el documento en el [FRAMESET](#) padre inmediato del marco actual. Este valor es equivalente a `_self` si el marco actual no tiene padre.

`_top`

El agente de usuario debería cargar el documento en la ventana original completa (cancelando así todos los demás marcos). Este valor es equivalente a `_self` si el marco actual no tiene padre.

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

7 La estructura global de un documento HTML

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Introducción a la estructura de un documento HTML](#)
2. [Información sobre la versión de HTML](#)
3. [El elemento HTML](#)
4. [La cabecera del documento](#)
 1. [El elemento HEAD](#)
 2. [El elemento TITLE](#)
 3. [El atributo title](#)
 4. [Metadatos](#)
 - [Especificación de metadatos](#)
 - [El elemento META](#)
 - [Perfiles de metadatos](#)
5. [El cuerpo del documento](#)
 1. [El elemento BODY](#)
 2. [Identificadores de elementos: los atributos id y class](#)
 3. [Elementos en bloque y elementos en línea](#)
 4. [Agrupación de elementos: los elementos DIV y SPAN](#)
 5. [Encabezados: los elementos H1, H2, H3, H4, H5, H6](#)
 6. [El elemento ADDRESS](#)

7.1 Introducción a la estructura de un documento HTML

Un documento HTML 4 se compone de tres partes:

1. una línea que contiene [información sobre la versión de HTML](#),
2. una sección de cabecera declarativa (delimitada por el elemento [HEAD](#)),
3. un cuerpo, que contiene el contenido real del documento. El cuerpo puede ser especificado mediante el elemento [BODY](#) o mediante el elemento [FRAMESET](#).

Puede aparecer espacio en blanco (espacios, saltos de línea, tabulaciones y comentarios) antes y después de cada sección. Las secciones 2 y 3 deberían estar delimitadas por el elemento [HTML](#).

Aquí tenemos un ejemplo de un documento HTML sencillo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Mi primer documento HTML</TITLE>
  </HEAD>
```

```
<BODY>
  <P>¡Hola mundo!
</BODY>
</HTML>
```

7.2 Información sobre la versión de HTML

Un documento HTML válido declara qué versión de HTML se utiliza en el documento. La *declaración del tipo de documento* especifica la definición del tipo de documento (DTD) que se usa en el documento (ver [\[ISO8879\]](#)).

HTML 4.01 especifica tres DTDs, de modo que los autores deben incluir una de las siguientes declaraciones del tipo de documento en sus documentos. Los DTDs varían en cuanto a los elementos que soportan.

- El [DTD HTML 4.01 Estricto](#) (Strict DTD) incluye todos los elementos y atributos que no han sido [desaprobados](#) o que no aparecen en documentos con marcos. Para los documentos que usen este DTD, utilice esta declaración del tipo de documento:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
```

- El [DTD HTML 4.01 Transicional](#) (Transitional DTD) incluye todo lo que incluye el DTD estricto más los elementos y atributos desaprobados (la mayoría de los cuales están relacionados con la presentación visual). Para los documentos que usen este DTD, utilice esta declaración del tipo de documento:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
```

- El [DTD HTML 4.01 para Documentos con Marcos](#) (Frameset DTD) incluye todo lo que incluye el DTD Transicional más los marcos. Para los documentos que usen este DTD, utilice esta declaración del tipo de documento:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
  "http://www.w3.org/TR/html4/frameset.dtd">
```

El URI que aparece en la declaración del tipo de documento permite a los agentes de usuario descargar el DTD y los [conjuntos de entidades](#) que sean necesarios. Los siguientes URIs (relativos) se refieren a los DTDs y conjuntos de entidades de HTML 4:

- "[strict.dtd](#)" -- DTD estricto por defecto
- "[loose.dtd](#)" -- DTD no estricto
- "[frameset.dtd](#)" -- DTD para documentos con marcos
- "[HTMLlat1.ent](#)" -- entidades Latin-1
- "[HTMLsymbol.ent](#)" -- entidades Symbol
- "[HTMLspecial.ent](#)" -- entidades especiales

La vinculación entre identificadores públicos y ficheros puede especificarse utilizando un fichero de catálogo según el formato recomendado por el Oasis Open Consortium (ver [\[OASISOPEN\]](#)). Al comienzo de la sección sobre la referencia SGML de HTML 4.01 se incluye un [fichero de catálogo de muestra para HTML 4.01](#). Las dos últimas letras de la declaración indican el idioma del DTD. Para HTML, éste es siempre inglés ("EN").

Nota. En lo que concierne a la versión de HTML 4.01 del 24 de diciembre, el Grupo de Trabajo HTML se compromete a la siguiente política:

- *Los posibles cambios en los DTDs futuros de HTML 4 no invalidarán los documentos que sean conformes con los DTDs de la presente especificación.* El Grupo de Trabajo HTML se reserva el derecho de corregir los errores conocidos.
- *Los programas que sean conformes con los DTDs de la presente especificación pueden no tener en cuenta las características de los DTDs futuros de HTML 4 que no reconozcan.*

Esto significa que en una declaración del tipo de documento, los autores pueden utilizar con seguridad un identificador de sistema que se refiera a la última versión de un DTD HTML 4. Los autores también pueden optar por usar un identificador

de sistema que se refiera a una versión específica (antigua) de un DTD HTML 4 cuando sea necesaria la validación con respecto a ese DTD en particular. El W3C hará todo lo posible para que los documentos archivados estén siempre disponibles en sus direcciones originales y en su forma original.

7.3 El elemento HTML

```
<!ENTITY % html.content "HEAD, BODY">
```

```
<!ELEMENT HTML O O (%html.content;) -- elemento raíz del documento -->
```

```
<!ATTLIST HTML
```

```
  %i18n; -- lang, dir --
```

```
>
```

Etiqueta inicial: *opcional*, Etiqueta final: *opcional*

Definiciones de atributos

version = [CDATA](#) [CN]

Desaprobado. El valor de este atributo especifica qué DTD HTML gobierna el documento actual. Este atributo ha sido desaprobado porque es redundante con la [información sobre la versión](#) proporcionada por la declaración del tipo de documento.

Atributos definidos en otros lugares

- [lang](#) ([información sobre el idioma](#)), [dir](#) ([dirección del texto](#))

Después de la declaración del tipo de documento, el resto de un documento HTML está contenido en el elemento [HTML](#).

Así, un documento HTML típico tiene esta estructura:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
```

```
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<HTML>
```

```
...La cabecera, el cuerpo, etc. van aquí...
```

```
</HTML>
```

7.4 La cabecera del documento

7.4.1 El elemento HEAD

```
<!-- %head.misc; definido previamente como "SCRIPT|STYLE|META|LINK|OBJECT" -->
```

```
<!ENTITY % head.content "TITLE & BASE?">
```

```
<!ELEMENT HEAD O O (%head.content;) +(%head.misc;) -- cabecera del documento -->
```

```
<!ATTLIST HEAD
```

```
  %i18n; -- lang, dir --
```

```
  profile %URI; #IMPLIED -- diccionario de metainformación con nombre
```

```
--
```

```
>
```

Etiqueta inicial: *opcional*, Etiqueta final: *opcional*

Definiciones de atributos

profile = [uri](#) [CT]

Este atributo especifica la localización de uno o más perfiles de metadatos, separados por espacio en blanco. Con vistas a extensiones futuras, los agentes de usuario deberían considerar este valor como una lista, si bien esta especificación sólo tiene en cuenta el primer URI. Se habla sobre los [perfiles](#) más adelante, en la sección sobre [metadatos](#).

Atributos definidos en otros lugares

- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)

El elemento [HEAD](#) contiene información sobre el documento actual, como el título, palabras clave que pueden ser de utilidad para motores de búsqueda, y otros datos que no se consideran parte del contenido del documento. En general, los agentes de usuario no representan los elementos que aparecen como contenido del [HEAD](#). Sin embargo, pueden poner la información del [HEAD](#) a disposición de los usuarios a través de otros mecanismos.

7.4.2 El elemento TITLE

```
<!-- El elemento TITLE no se considera parte del flujo de texto.
      Debería ser mostrado, por ejemplo, como el encabezado de la página
      o como el título de la ventana. Se requiere exactamente un título
      por documento.
-->
<!ELEMENT TITLE - - (#PCDATA) -(%head.misc;) -- título del documento -->
<!ATTLIST TITLE %i18n>
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria***

Atributos definidos en otros lugares

- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)

Todos los documentos HTML **deben** tener un elemento [TITLE](#) en la sección [HEAD](#).

Los autores deberían utilizar el elemento [TITLE](#) para identificar los contenidos de un documento. Debido a que los usuarios a menudo consultan documentos fuera de contexto, los autores deberían proporcionar títulos ricos en contexto. Así, en vez de usar un título como "Introducción", que no proporciona mucha información acerca del contexto, los autores deberían poner en su lugar un título del estilo "Introducción a la apicultura medieval".

Por razones de accesibilidad, los agentes de usuario siempre deben poner el contenido del elemento [TITLE](#) a disposición de los usuarios (incluyendo los elementos [TITLE](#) que aparezcan en marcos). El mecanismo para ello depende del agente de usuario (p.ej., como un título, hablado).

Los títulos pueden contener [entidades de caracteres](#) (para caracteres acentuados, caracteres especiales, etc.), pero no pueden contener código (incluyendo comentarios). Aquí tenemos un ejemplo de título de documento:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
      "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE>Un estudio sobre la dinámica de la población</TITLE>
... otros elementos de cabecera...
</HEAD>
<BODY>
... cuerpo del documento...
</BODY>
</HTML>
```

7.4.3 El atributo title

Definiciones de atributos

title = [texto](#) [CS]

Este atributo ofrece información consultiva sobre el elemento para el cual se establece.

A diferencia del elemento [TITLE](#), que proporciona información sobre un documento entero y que sólo puede aparecer una vez, el atributo [title](#) puede anotar cualquier número de elementos. Para saber si un elemento soporta este atributo consulte la

definición del elemento.

Los valores del atributo [title](#) pueden ser representados por los agentes de usuario de diferentes maneras. Por ejemplo, los navegadores visuales suelen representar el título como un "tool tip" (un mensaje corto que aparece cuando el dispositivo apuntador se detiene sobre un objeto). Los agentes de usuario de voz pueden pronunciar la información del título en un contexto similar. Por ejemplo, al establecer el atributo en un vínculo, los agentes de usuario (visuales y no visuales) pueden decir a los usuarios la naturaleza del recurso vinculado:

...texto...

Aquí hay una foto mía

```
<A href="http://algunsitio.com/cosascheveres.gif" title="Yo haciendo submarinismo">
  haciendo submarinismo el verano pasado
</A>
```

...más texto...

El atributo [title](#) tiene un papel adicional cuando se utiliza con el elemento [LINK](#) para designar una [hoja de estilo externa](#). Consulte la sección sobre [vínculos y hojas de estilo](#) para más detalles.

Nota. Para mejorar la calidad de la síntesis de voz en los casos en que las técnicas estándar logran malos resultados, las versiones futuras de HTML podrían incluir un atributo para codificar información fonémica y prosódica.

7.4.4 Metadatos

Nota. El Marco de Descripción de Recursos (Resource Description Framework) del W3C (ver [\[RDF10\]](#)) se convirtió en Recomendación del W3C en febrero de 1999. El RDF permite a los autores especificar metadatos legibles por máquina sobre documentos HTML y otros recursos accesibles por la red.

El HTML permite a los autores especificar metadatos -- información sobre un documento más que contenido del propio documento -- de diferentes maneras.

Por ejemplo, para especificar el autor de un documento, puede utilizarse el elemento [META](#) como sigue:

```
<META name="Author" content="Dave Raggett ">
```

El elemento [META](#) especifica una propiedad (aquí "Author") y le asigna un valor (aquí "Dave Raggett").

Esta especificación no define un conjunto de propiedades legales de metadatos. El significado de una propiedad y el conjunto de valores legales para esa propiedad debería estar definida en un diccionario de referencia llamado [perfil](#). Por ejemplo, un perfil diseñado para ayudar a los motores de búsqueda a indexar documentos podría definir propiedades tales como "author", "copyright", "keywords", etc.

Especificación de metadatos

En general, la especificación de metadatos implica dos pasos:

1. Declaración de una propiedad y de un valor para esta propiedad. Esto puede hacerse de dos maneras:
 1. Desde dentro de un documento, por medio del elemento [META](#).
 2. Desde fuera de un documento, vinculando los metadatos por medio del elemento [LINK](#) (véase la sección sobre [tipos de vínculos](#)).
2. Referencia a un [perfil](#) en el que se definen la propiedad y sus valores legales. Para designar un perfil, se usa el atributo [profile](#) del elemento [HEAD](#).

Obsérvese que al estar definido un perfil por el elemento [HEAD](#), se aplica el mismo perfil a todos los elementos [META](#) y [LINK](#) de la cabecera del documento.

Los agentes de usuario no necesitan soportar los mecanismos de metadatos. Para aquellos que opten por soportar metadatos, esta especificación no define cómo deberían interpretarse los metadatos.

El elemento META

```

<!ELEMENT META - O EMPTY -- metainformación genérica -->
<!ATTLIST META
  %i18n; -- lang, dir, para usar con content --
  http-equiv NAME #IMPLIED -- nombre de encabezado de respuesta HTTP --
  name NAME #IMPLIED -- nombre de la metainformación --
  content CDATA #REQUIRED -- información asociada --
  scheme CDATA #IMPLIED -- seleccionar forma de contenido --
>

```

Etiqueta inicial: **obligatoria**, Etiqueta final: **prohibida**

Definiciones de atributos

Para los siguientes atributos, los valores permitidos y su interpretación depende del [perfil](#):

name = [name](#) [CS]

Este atributo identifica un nombre de propiedad. Esta especificación no enumera los valores legales para este atributo.

content = [cdata](#) [CS]

Este atributo especifica el valor de una propiedad. Esta especificación no enumera los valores legales para este atributo.

scheme = [cdata](#) [CS]

Este atributo especifica un esquema que se usará para interpretar el valor de la propiedad (véase la sección sobre [perfiles](#) para más detalles).

http-equiv = [name](#) [CI]

Este atributo puede utilizarse en lugar del atributo name. Los servidores HTTP utilizan este atributo para obtener información sobre los encabezados del mensaje de respuesta HTTP.

Atributos definidos en otros lugares

- [lang](#) ([información sobre el idioma](#)), [dir](#) ([dirección del texto](#))

El elemento [META](#) puede utilizarse para identificar propiedades de un documento (p.ej., el autor, la fecha de caducidad, una lista de palabras clave, etc.) y para asignar valores a esas propiedades. Esta especificación no define un conjunto normativo de propiedades.

Cada elemento [META](#) especifica una pareja propiedad/valor. El atributo name identifica la propiedad y el atributo [content](#) especifica el valor de la propiedad.

Por ejemplo, la siguiente declaración establece un valor para la propiedad Author:

```
<META name="Author" content="Dave Raggett">
```

Puede utilizarse el atributo [lang](#) de [META](#) para especificar el idioma del valor del atributo [content](#). Esto permite a los sintetizadores de voz aplicar reglas de pronunciación dependientes del idioma.

En este ejemplo, se declara que el nombre del autor está en francés:

```
<META name="Author" lang="fr" content="Arnaud Le Hors">
```

Nota. El elemento [META](#) es un mecanismo genérico para la especificación de metadatos. Sin embargo, hay algunos elementos y atributos HTML que ya manejan determinados metadatos y que pueden ser utilizados por los autores en lugar de [META](#) para especificar dichos metadatos: el elemento [TITLE](#), el elemento [ADDRESS](#), los elementos [INS](#) y [DEL](#), el atributo [title](#), y el atributo [cite](#).

Nota. Cuando una propiedad especificada mediante un elemento [META](#) toma un valor que es un [URI](#), algunos autores prefieren especificar los metadatos mediante el elemento [LINK](#). Así, la siguiente declaración de metadatos:

```
<META name="DC.identifier"
  content="http://www.ietf.org/rfc/rfc1866.txt">
```

también podría haberse escrito así:

```
<LINK rel="DC.identifier"
      type="text/plain"
      href="http://www.ietf.org/rfc/rfc1866.txt">
```

META y encabezados HTTP

El atributo [http-equiv](#) puede utilizarse en lugar del atributo name, lo cual tiene un significado especial cuando los documentos se obtienen mediante el Protocolo de Transferencia de Hipertexto (HTTP). Los servidores HTTP pueden usar el nombre de la propiedad especificada por el atributo [http-equiv](#) para crear un encabezado al estilo [\[RFC822\]](#) en la respuesta HTTP. Vea la especificación HTTP ([\[RFC2616\]](#)) para más detalles sobre encabezados HTTP válidos.

La siguiente declaración [META](#) de ejemplo:

```
<META http-equiv="Expires" content="Tue, 20 Aug 1996 14:25:27 GMT">
```

resultará en el encabezado HTTP:

```
Expires: Tue, 20 Aug 1996 14:25:27 GMT
```

Esto lo pueden utilizar las cachés para determinar cuándo obtener una nueva copia del documento asociado.

***Nota.** Algunos agentes de usuario soportan el uso de [META](#) para refrescar la página actual después de un número especificado de segundos, con la opción de reemplazarla con un URI diferente. Los autores **no** deberían utilizar esta técnica para dirigir a los usuarios a páginas diferentes, ya que esto hace la página inaccesible para algunos usuarios. En lugar de eso, la redirección automática de páginas debería realizarse usando redirección en el lado del servidor.*

META y motores de búsqueda

Un uso común de [META](#) es especificar palabras clave que pueden usar los motores de búsqueda para mejorar la calidad de los resultados de una búsqueda. Cuando se proporcionen varios elementos [META](#) con información para varios idiomas, los motores de búsqueda pueden utilizar el atributo [lang](#) como filtro para mostrar los resultados de la búsqueda usando las preferencias de idioma del usuario. Por ejemplo,

```
<!-- Para hablantes de inglés americano -->
<META name="keywords" lang="en-us"
      content="vacation, Greece, sunshine">
<!-- Para hablantes de inglés británico -->
<META name="keywords" lang="en"
      content="holiday, Greece, sunshine">
<!-- Para hablantes de español -->
<META name="keywords" lang="es"
      content="vacaciones, Grecia, sol">
```

También puede incrementarse la efectividad de los motores de búsqueda usando el elemento [LINK](#) para especificar vínculos a traducciones del documento en otros idiomas, vínculos a versiones del documento en otros medios (p.ej., PDF), y, cuando el documento es parte de una colección, vínculos a un punto apropiado de partida para examinar la colección completa.

Se puede encontrar más ayuda en la sección sobre [cómo ayudar a los motores de búsqueda a indexar su sitio Web](#).

META y PICS

La Plataforma para la Selección de Contenido en Internet (PICS, especificada en [\[PICS\]](#)) es una infraestructura para asociar etiquetas (metadatos) con contenido de Internet. Diseñada originalmente para ayudar a los padres y a las escuelas a controlar los lugares a los que pueden acceder los niños en Internet, también facilita otros usos para las etiquetas, incluyendo firmas de código, privacidad, y gestión de los derechos de la propiedad intelectual.

Este ejemplo ilustra cómo puede usarse una declaración [META](#) para incluir una etiqueta PICS 1.1:


```
<HEAD>
<META http-equiv="PICS-Label" content='
(PICS-1.1 "http://www.gcf.org/v2.5"
  labels on "1994.11.05T08:15-0500"
  until "1995.12.31T23:59-0000"
  for "http://w3.org/PICS/Overview.html"
  ratings (suds 0.5 density 0 color/hue 1))
'>
<TITLE>... título del documento ...</TITLE>
</HEAD>
```

META e información por defecto

El elemento [META](#) puede utilizarse para especificar la información por defecto de un documento en los aspectos siguientes:

- El [lenguaje de scripts por defecto](#).
- El [lenguaje de hojas de estilo por defecto](#).
- La [codificación de caracteres del documento](#).

El siguiente ejemplo especifica que la [codificación de caracteres](#) de un documento es la ISO-8859-5

```
<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-5">
```

Perfiles de metadatos

El atributo [profile](#) de [HEAD](#) especifica la localización de un perfil de metadatos. El valor del atributo [profile](#) es un URI. Los agentes de usuario pueden utilizar este URI de dos maneras:

- Como un nombre único a nivel global. Los agentes de usuario pueden ser capaces de reconocer el nombre (sin necesidad de obtener el perfil) y realizar alguna acción según las convenciones conocidas relativas a ese perfil. Por ejemplo, los motores de búsqueda podrían proporcionar una interfaz para búsqueda en catálogos de documentos HTML, de modo que todos los documentos podrían usar el mismo perfil para representar entradas de un catálogo.
- Como un vínculo. Los agentes de usuario pueden seguir el URI y realizar alguna acción según las definiciones contenidas en el perfil (p.ej., autorizar el uso del perfil dentro del documento HTML actual). Esta especificación no define formatos de perfiles.

Este ejemplo hace referencia a un perfil hipotético que define propiedades útiles para indexar documentos. A las propiedades definidas en este perfil -- incluyendo "author", "copyright", "keywords" (palabras clave) y "date" (fecha) -- se les asignan valores mediante declaraciones [META](#) subsiguientes.

```
<HEAD profile="http://www.acme.com/profiles/core">
<TITLE>Cómo completar portadas de Memoranda</TITLE>
<META name="author" content="José Pérez">
<META name="copyright" content="&copy; 1997 Acme Corp.">
<META name="keywords" content="empresarial,instrucciones,catálogos">
<META name="date" content="1994-11-06T08:49:37+00:00">
</HEAD>
```

En el momento de escribir esta especificación, la práctica común es usar los formatos de fechas descritos en [\[RFC2616\]](#), sección 3.3. Como estos formatos son relativamente complicados de procesar, recomendamos que los autores utilicen el formato de fechas [\[ISO8601\]](#). Para más información, véanse las secciones sobre los elementos [INS](#) y [DEL](#).

El atributo [scheme](#) permite a los autores proporcionar a los agentes de usuario más contexto para la interpretación correcta de los metadatos. A veces, esta información adicional puede ser crítica, por ejemplo cuando los metadatos pueden ser especificados según formatos diferentes. Por ejemplo, un autor podría especificar una fecha en el formato (ambiguo) "10-9-97"; ¿significa esto 9 de octubre de 1997 o 10 de septiembre de 1997? El valor "Mes-Día-Año" para el atributo [scheme](#) eliminaría la ambigüedad de este valor de fecha.

En otras ocasiones, el atributo [scheme](#) puede proporcionar información útil aunque no crítica a los agentes de usuario.

Por ejemplo, la siguiente declaración [scheme](#) podría ayudar a un agente de usuario a determinar que el valor de la propiedad "identificador" es un número de código ISBN:

```
<META scheme="ISBN" name="identificador" content="0-8230-2355-9">
```

Los valores del atributo [scheme](#) dependen de la propiedad name y del [profile](#) asociado.

Nota. Un ejemplo de perfil es el Dublin Core (ver [\[DCORE\]](#)). Este perfil define un conjunto de propiedades recomendadas para descripciones bibliográficas electrónicas, y su objetivo es promover la interoperabilidad entre modelos descriptivos dispares.

7.5 El cuerpo del documento

7.5.1 El elemento BODY

```
<!ELEMENT BODY O O (%block;|SCRIPT)+ +(INS|DEL) -- cuerpo del documento -->
<!ATTLIST BODY
  %attrs;                -- %coreattrs, %i18n, %events --
  onload                 %Script;    #IMPLIED  -- el documento ha sido cargado --
  onunload              %Script;    #IMPLIED  -- el documento ha sido quitado --
  >
```

Etiqueta inicial: opcional, Etiqueta final: opcional

Definiciones de atributos

background = [uri](#) [CT]

Desaprobado. El valor de este atributo es un URI que designa un recurso de imagen. En general la imagen se repite para rellenar el fondo (en navegadores visuales).

text = [color](#) [CI]

Desaprobado. Este atributo establece el color de primer plano para el texto (en navegadores visuales).

link = [color](#) [CI]

Desaprobado. Este atributo establece el color del texto que marca los vínculos de hipertexto no visitados (en navegadores visuales)

vlink = [color](#) [CI]

Desaprobado. Este atributo especifica el color del texto que marca los vínculos de hipertexto visitados (en navegadores visuales).

alink = [color](#) [CI]

Desaprobado. Este atributo especifica el color del texto que marca los vínculos de hipertexto cuando son seleccionados por el usuario (en navegadores visuales).

Atributos definidos en otros lugares

- [id](#), [class](#) ([identificadores a nivel de documento](#))
- [lang](#) ([información sobre el idioma](#)), [dir](#) ([dirección del texto](#))
- [title](#) ([título del elemento](#))
- [style](#) ([información de estilo en línea](#))
- [bgcolor](#) ([color del fondo](#))
- [onload](#), [onunload](#) ([eventos intrínsecos](#))
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) ([eventos intrínsecos](#))

El cuerpo de un documento contiene el contenido del documento. El contenido puede ser presentado por un agente de usuario de distintas maneras. Por ejemplo, para los navegadores visuales, se puede imaginar el cuerpo como un lienzo sobre

el que aparece el contenido: texto, imágenes, colores, gráficos, etc. Para agentes de usuario por voz, el mismo contenido podría ser pronunciado. Debido a que ahora el método preferido de especificar la presentación de un documento son las [hojas de estilo](#), los atributos presentacionales del elemento [BODY](#) han sido [desaprobados](#).

EJEMPLO DESAPROBADO:

El siguiente fragmento HTML ilustra el uso de los atributos [desaprobados](#). Establece el color de fondo del lienzo en blanco, el color de primer plano del texto en negro, y el color de los hipervínculos en rojo inicialmente, fucsia cuando son activados y marrón una vez que han sido visitados.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
  <TITLE>Un estudio sobre la dinámica de la población</TITLE>
</HEAD>
<BODY bgcolor="white" text="black"
  link="red" alink="fuchsia" vlink="maroon">
  ... cuerpo del documento ...
</BODY>
</HTML>
```

Usando [hojas de estilo](#), se podría conseguir el mismo efecto de la siguiente manera:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
  <TITLE>Un estudio sobre la dinámica de la población</TITLE>
  <STYLE type="text/css">
    BODY { background: white; color: black }
    A:link { color: red }
    A:visited { color: maroon }
    A:active { color: fuchsia }
  </STYLE>
</HEAD>
<BODY>
  ... cuerpo del documento ...
</BODY>
</HTML>
```

El usar hojas de estilo externas (vinculadas) nos da flexibilidad para cambiar la presentación sin tener que revisar el documento fuente HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
  <TITLE>Un estudio sobre la dinámica de la población</TITLE>
  <LINK rel="stylesheet" type="text/css" href="estilolisto.css">
</HEAD>
<BODY>
  ... cuerpo del documento ...
</BODY>
</HTML>
```

Marcos y cuerpos HTML. Los documentos que contienen marcos reemplazan el elemento [BODY](#) con el elemento [FRAMESET](#). Consulte la sección sobre [marcos](#) para más información.

7.5.2 Identificadores de elementos: los elementos id y class

Definiciones de atributos

id = *name* [CS]

Este atributo asigna un nombre a un elemento. Este nombre debe ser único en un documento.

class = *lista de cdata* [CS]

Este atributo asigna un nombre de clase o un conjunto de nombres de clase a un elemento. Se puede asignar el mismo nombre o nombres de clase a cualquier número de elementos. Los nombres de clase múltiples deben estar separados por caracteres de espacio en blanco.

El atributo [id](#) asigna un identificador único a un elemento (lo cual puede ser verificado por un analizador SGML). Por ejemplo, los siguientes párrafos se distinguen por sus valores de [id](#):

```
<P id="miparrafo"> Esto es un párrafo con un nombre único.</P>
<P id="tuparrafo"> Esto también es un párrafo con un nombre único.</P>
```

El atributo [id](#) tiene varios papeles en HTML:

- Como selector para las [hojas de estilo](#).
- Como [vínculo](#) destino para vínculos de hipertexto.
- Como medio de hacer referencia a un elemento en particular desde un [script](#).
- Como nombre de un elemento [OBJECT](#) declarado.
- Para procesos generales por parte de agentes de usuario (p.ej., para identificar campos cuando se transfieren datos desde páginas HTML hasta una base de datos, para traducir documentos HTML a otros formatos, etc.).

El atributo [class](#), por otra parte, asigna uno o más nombres de clase a un elemento; se puede decir que el elemento pertenece a estas clases. Varios elementos pueden compartir el mismo nombre de clase. El atributo [class](#) tiene varios papeles en HTML:

- Como selector para [hojas de estilo](#) (cuando un autor desea asignar información de estilo a un conjunto de elementos).
- Para procesos generales por parte de agentes de usuario.

En el siguiente ejemplo, el elemento [SPAN](#) se utiliza junto con los atributos [id](#) y [class](#) para codificar mensajes informativos. Los mensajes aparecen tanto en inglés como en español.

```
<!-- Mensajes en inglés -->
<P><SPAN id="msg1" class="info" lang="en">Variable declared twice</SPAN>
<P><SPAN id="msg2" class="advertencia" lang="en">Undeclared variable</SPAN>
<P><SPAN id="msg3" class="error" lang="en">Bad syntax for variable name</SPAN>

<!-- Mensajes en español -->
<P><SPAN id="msg1" class="info" lang="es">Variable declarada dos veces</SPAN>
<P><SPAN id="msg2" class="advertencia" lang="es">Variable no declarada</SPAN>
<P><SPAN id="msg3" class="error" lang="es">Sintaxis incorrecta para el nombre
de la variable</SPAN>
```

Las siguientes reglas de estilo CSS dirían a los agentes de usuario visuales que representarían los mensajes informativos en verde, los mensajes de advertencia en amarillo, y los mensajes de error en rojo:

```
SPAN.info      { color: green }
SPAN.advertencia { color: yellow }
SPAN.error     { color: red }
```

Obsérvese que el "msg1" español y el "msg1" inglés no pueden aparecer en el mismo documento, ya que ambos comparten el mismo valor de [id](#). Los autores pueden hacer un uso mayor del atributo [id](#) para refinar la presentación de mensajes individuales, hacerlos vínculos destino, etc.

Se les puede asignar información de identificador y clase a casi todos los elementos HTML.

Supongamos, por ejemplo, que estamos escribiendo un documento sobre un lenguaje de programación. El documento debe incluir un número de ejemplos preformateados. Usamos el elemento [PRE](#) para formatear los ejemplos. También asignamos un color de fondo (verde) a todos los ejemplares del elemento [PRE](#) que pertenezcan a la clase "ejemplo".

```
<HEAD>
<TITLE>... título del documento ...</TITLE>
<STYLE type="text/css">
PRE.ejemplo { background : green }
</STYLE>
</HEAD>
<BODY>
<PRE class="ejemplo" id="ejemplo-1">
...código del ejemplo...
</PRE>
</BODY>
```

Al establecer el atributo [id](#) para este ejemplo, podemos (1) crear un hipervínculo a él y (2) sustituir la información de estilo de la clase por información de estilo propia.

Nota. El atributo [id](#) comparte el mismo espacio de nombres que el atributo [name](#) cuando se usa para nombres de vínculos. Consulte la sección sobre [vínculos con id](#) para más información.

7.5.3 Elementos en bloque y elementos en línea

Algunos de los elementos HTML que pueden aparecer en [BODY](#) se llaman elementos "en bloque" (o también "a nivel de bloque") mientras que otros son elementos "en línea" (o "a nivel de texto"). Esta distinción se basa en varios aspectos:

Modelo de contenido

Generalmente, los elementos en bloque pueden contener elementos en línea y a otros elementos en bloque.

Generalmente, los elementos en línea sólo pueden contener datos y a otros elementos en línea. Inherentemente a esta distinción estructural está la idea de que los elementos en bloque crean estructuras "más grandes" que los elementos en línea.

Formato

Los elementos en bloque tienen por defecto un formato diferente que el de los elementos en línea. Generalmente, los elementos en bloque comienzan en una nueva línea, y los elementos en línea no. Para información sobre espacio en blanco, saltos de línea, y formato de bloques, consulte la sección sobre [texto](#).

Direccionalidad

Por razones técnicas relacionadas con el algoritmo de texto bidireccional de [UNICODE](#), los elementos en bloque y en línea difieren en el modo de heredar la información de direccionalidad. Para más detalles, vea la sección sobre [herencia de la dirección del texto](#).

[Las hojas de estilo](#) proporcionan medios para especificar la representación de elementos arbitrarios, incluyendo la representación como elemento en bloque o en línea. En algunos casos, como por ejemplo para información en línea para objetos de lista, esto puede ser apropiado, pero en general no es aconsejable que los autores invaliden la interpretación convencional de los elementos HTML de este modo.

La alteración del estilo de presentación tradicional de los elementos en bloque y en línea también influye en el algoritmo de texto bidireccional. Vea la sección sobre [el efecto de las hojas de estilo en la bidireccionalidad](#) para más información.

7.5.4 Agrupación de elementos: los elementos DIV y SPAN

```
<!ELEMENT DIV - - (%flow;)* -- contenedor genérico de idioma/estilo -->
<!ATTLIST DIV
  %attrs; -- %coreattrs, %i18n, %events --
  >
<!ELEMENT SPAN - - (%inline;)* -- contenedor genérico de idioma/estilo -->
<!ATTLIST SPAN
```

```
%attrs; -- %coreattrs, %i18n, %events --
>
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [align](#) (alineación)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

Los elementos [DIV](#) y [SPAN](#), junto con los atributos [id](#) y [class](#), ofrecen un mecanismo genérico para añadir estructura a los documentos. Estos elementos especifican si su contenido es en línea ([SPAN](#)) o en bloque ([DIV](#)) pero no imponen ningún otro estilo de presentación al contenido. Así, los autores pueden usar estos elementos junto con [hojas de estilo](#), el atributo [lang](#), etc., para adaptar el HTML a sus propios gustos y necesidades.

Supongamos, por ejemplo, que quisiéramos generar un documento HTML basado en una base de datos de información sobre clientes. Como HTML no incluye elementos que identifiquen objetos tales como "cliente", "número de teléfono", "dirección de correo electrónico", etc., utilizamos [DIV](#) y [SPAN](#) para lograr los efectos estructurales y presentacionales deseados. Podríamos usar el elemento [TABLE](#) del modo siguiente para estructurar la información:

```
<!-- Ejemplo de registro de la base de datos de clientes: -->
<!-- Nombre: Stephane Boyera, Tel: (212) 555-1212, Email: sb@foo.org -->

<DIV id="cliente-boyera" class="cliente">
<P><SPAN class="cliente-titulo">Información sobre el cliente:</SPAN>
<TABLE class="cliente-datos">
<TR><TH>Apellido:<TD>Boyera</TR>
<TR><TH>Nombre:<TD>Stephane</TR>
<TR><TH>Tel:<TD>(212) 555-1212</TR>
<TR><TH>Email:<TD>sb@foo.org</TR>
</TABLE>
</DIV>

<DIV id="cliente-lafon" class="cliente">
<P><SPAN class="cliente-titulo">Información sobre el cliente:</SPAN>
<TABLE class="cliente-datos">
<TR><TH>Apellido:<TD>Lafon</TR>
<TR><TH>Nombre:<TD>Yves</TR>
<TR><TH>Tel:<TD>(617) 555-1212</TR>
<TR><TH>Email:<TD>yves@coucou.com</TR>
</TABLE>
</DIV>
```

A continuación, podemos añadir fácilmente declaraciones a la hoja de estilo para ajustar la presentación de estas entradas de la base de datos.

Para otro ejemplo de uso, consulte el ejemplo de la sección sobre [los atributos class e id](#).

En general los agentes de usuario visuales colocan un salto de línea antes y otro después de los elementos [DIV](#), por ejemplo:

```
<P>aaaaaaaaa<DIV>bbbbbbbbbb</DIV><DIV>cccc<P>cccc</DIV>
```

que normalmente se representa como:

aaaaaaaaa
bbbbbbbbb
cccc

cccc

7.5.5 Encabezados: los elementos H1, H2, H3, H4, H5, H6

```
<!ENTITY % heading "H1|H2|H3|H4|H5|H6">
<!--
  Hay seis niveles de encabezados, desde H1 (el más importante)
  hasta H6 (el menos importante).
-->

<!ELEMENT (%heading;) - - (%inline;)* -- encabezado -->
<!ATTLIST (%heading;)
  %attrs; -- %coreattrs, %il8n, %events --
  >
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria***

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [align](#) (alineación)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

Un encabezado describe brevemente el tema de la sección que introduce. La información de encabezado puede ser utilizada por los agentes de usuario, por ejemplo, para construir una tabla de contenidos de un documento automáticamente.

Hay seis niveles de encabezados en HTML, siendo [H1](#) el más importante y [H6](#) el menos importante. Los navegadores visuales pueden representar los encabezados más importantes con fuentes más grandes que los menos importantes.

El siguiente ejemplo muestra cómo usar el elemento [DIV](#) para asociar un encabezado con la sección del documento que le sigue. Esto nos permite definir un estilo para la sección (color del fondo, fuente del texto, etc.) con hojas de estilo.

```
<DIV class="seccion" id="elefantes-de-la-selva">
<H1>Elefantes de la selva</H1>
<P>En esta sección descubriremos a esos grandes desconocidos:
  los elefantes de la selva.
...la sección continúa...
<DIV class="subseccion" id="habitat-de-la-jungla">
<H2>Hábitat</H2>
<P>Los elefantes de la selva no viven en los árboles, sino entre ellos.
...la subsección continúa...
</DIV>
</DIV>
```

Podemos decorar esta estructura con información de estilo tal como ésta:

```
<HEAD>
<TITLE>... título del documento ...</TITLE>
<STYLE type="text/css">
```

```
DIV.seccion { text-align: justify; font-size: 12pt}
DIV.subseccion { text-indent: 2em }
H1 { font-style: italic; color: green }
H2 { color: green }
</STYLE>
</HEAD>
```

Secciones numeradas y referencias

HTML no genera por sí mismo números de sección a partir de los encabezados. Sin embargo esto podría ser ofrecido por los agentes de usuario. Pronto los lenguajes de hojas de estilo como CSS permitirán a los autores controlar la generación de números de sección (muy útiles para hacer referencias de documentos impresos, como en "Ver la sección 7.2").

Algunas personas consideran que saltarse niveles de encabezado es mala práctica. Aceptan H1 H2 H1 pero no aceptan H1 H3 H1 ya que se salta el nivel de encabezado H2.

7.5.6 El elemento ADDRESS

```
<!ELEMENT ADDRESS - - (%inline;)* -- información sobre el autor -->
<!ATTLIST ADDRESS
  %attrs; -- %coreattrs, %i18n, %events --
  >
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

El elemento [ADDRESS](#) puede ser utilizado por los autores para proporcionar información de contacto en un documento o en una parte de un documento, como por ejemplo un formulario. Este elemento suele aparecer el principio de un documento.

Por ejemplo, una página del sitio web del W3C relacionada con HTML podría incluir la siguiente información de contacto:

```
<ADDRESS>
<A href="../People/Raggett/">Dave Raggett</A>,
<A href="../People/Arnaud/">Arnaud Le Hors</A>,
personas de contacto del <A href="Activity">W3C HTML Activity</A><BR>
$Fecha: 1999/12/24 23:07:14 $
</ADDRESS>
```

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

8 Información sobre el idioma y la dirección del texto

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Especificación del idioma del contenido: el atributo lang](#)
 1. [Códigos de idioma](#)
 2. [Herencia de los códigos de idioma](#)
 3. [Interpretación de los códigos de idioma](#)
2. [Especificación de la dirección del texto y de las tablas: el atributo dir](#)
 1. [Introducción al algoritmo bidireccional](#)
 2. [Herencia de la información sobre la dirección del texto](#)
 3. [Especificación de la dirección del texto incluido](#)
 4. [Anulación del algoritmo bidireccional: el elemento BDO](#)
 5. [Referencias de caracteres para el control de la direccionalidad y de la unión](#)
 6. [Efecto de las hojas de estilo en la bidireccionalidad](#)

Esta sección del documento trata sobre dos temas importantes que afectan a la internacionalización del HTML: la especificación del idioma (el atributo [lang](#)) y de la dirección (el atributo [dir](#)) del texto de un documento.

8.1 Especificación del idioma del contenido: el atributo lang

Definiciones de atributos

lang = [código de idioma](#) [CI]

Este atributo especifica el idioma base de los valores de los atributos y del texto contenido en un elemento. El valor por defecto de este atributo es desconocido.

La información sobre el idioma especificada por medio del atributo [lang](#) puede ser utilizada por un agente de usuario de diferentes formas para controlar la representación. Algunas cosas para las que la información sobre el idioma proporcionada por el autor puede ser útil son las siguientes:

- Ayudar a los motores de búsqueda

- Ayudar a los sintetizadores de voz
- Ayudar al agente de usuario a elegir variaciones de un signo para tipografía de alta calidad
- Ayudar al agente de usuario a elegir un conjunto de caracteres para el entrecomillado de citas
- Ayudar al agente de usuario a hacer decisiones sobre [separación de palabras](#), ligaduras, y espaciado
- Ayudar a los verificadores de ortografía y gramática

El atributo [lang](#) especifica el idioma del contenido del elemento y de los valores de los atributos; el que sea relevante para un atributo dado dependerá de la sintaxis y de la semántica del atributo y de la operación involucrada.

El objetivo del atributo [lang](#) es permitir a los agentes de usuario representar el contenido de forma más significativa según la práctica cultural aceptada para un idioma dado. Esto no implica que los agentes de usuario deberían representar los caracteres que no son típicos de un idioma dado de maneras menos significativas; los agentes de usuario deben intentar representar todos los caracteres lo mejor posible, independientemente del valor especificado por [lang](#).

Por ejemplo, si aparecen caracteres del alfabeto griego dentro un texto escrito en español:

```
<P><Q lang="es">Sus superpoderes son el resultado de las radiaciones &gamma; ,</Q> dijo él.</P>
```

un agente de usuario (1) debería intentar representar el contenido en español de una manera apropiada (p.ej., en lo que respecta a la puntuación de la cita) y (2) debe intentar representar `γ` lo mejor posible aunque no sea un carácter español.

Consulte la sección sobre [caracteres no representables](#) para información relacionada.

8.1.1 Códigos de idioma

El valor del atributo [lang](#) es un código de idioma que identifica un lenguaje natural hablado, escrito o usado de cualquier modo para la comunicación de información entre personas. Los lenguajes de ordenador están explícitamente excluidos de los códigos de idiomas.

La [RFC1766](#) define y explica los códigos de idioma que deben ser usados en los documentos HTML.

En pocas palabras, los códigos de idioma consisten en un código principal y una serie posiblemente vacía de subcódigos:

código-de-idioma = código-principal ("-" subcódigo)*

Aquí tenemos algunos ejemplos de códigos de idioma:

- "en": inglés
- "en-US": la variación estadounidense del inglés.
- "en-cockney": la variación de Cockney del inglés.
- "i-navajo": el idioma Navajo hablado por algunos nativos americanos.
- "x-klíngon": la marca principal "x" indica un lenguaje experimental

En la [ISO639](#) se reservan códigos principales de dos letras para las abreviaturas de los idiomas. Entre

estos códigos de dos letras están fr (francés), de (alemán), it (italiano), nl (neerlandés), el (griego), es (español), pt (portugués), ar (árabe), he (hebreo), ru (ruso), zh (chino), ja (japonés), hi (hindi), ur (urdu), y sa (sánscrito).

Se sobreentiende que cualquier código de dos letras es un código de país de [\[ISO3166\]](#).

8.1.2 Herencia de los códigos de idioma

Un elemento hereda la información sobre el código de idioma de acuerdo con el siguiente orden de precedencia (de más alto a más bajo):

- El atributo [lang](#) establecido para el propio elemento.
- El elemento padre más cercano que tenga establecido el atributo [lang](#) (es decir, el atributo [lang](#) se hereda).
- El encabezado HTTP "Content-Language" (que puede ser configurado en un servidor). Por ejemplo:

```
Content-Language: en-cockney
```

- Valores por defecto del agente de usuario y preferencias del usuario.

En este ejemplo, el idioma principal del documento es el francés ("fr"). Hay un párrafo que se declara como escrito en español ("es"), después del cual el idioma principal vuelve a ser el francés. El párrafo siguiente incluye un frase en japonés ("ja"), después de la cual el idioma principal vuelve a ser el francés.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<HTML lang="fr">
<HEAD>
<TITLE>Un document multilingue</TITLE>
</HEAD>
<BODY>
...Interpretado como francés...
<P lang="es">...Interpretado como español...
<P>...Interpretado de nuevo como francés...
<P>...Texto en francés interrumpido por <EM lang="ja">algo
  en japonés</EM> Aquí vuelve a empezar en francés...
</BODY>
</HTML>
```

Nota. Las celdas de las tablas pueden heredar los valores los valores de [lang](#) no de su padre, sino de la primera celda de un tramo. Consulte la sección sobre [herencia de la alineación](#) para más detalles.

8.1.3 Interpretación de los códigos de idioma

En el contexto del HTML, los agentes de usuario deberían interpretar un código de idioma como una jerarquía de símbolos más que como un símbolo independiente. Cuando un agente de usuario ajusta la representación de acuerdo con la información sobre el idioma (por ejemplo, comparando los códigos de idioma de las hojas de estilos y los valores de [lang](#)), siempre debería dar prioridad a un emparejamiento exacto, pero también debería considerar suficiente el emparejamiento de códigos principales. Así, si se

establece un valor "en-US" para el atributo [lang](#) del elemento [HTML](#), un agente de usuario debería preferir en primer lugar la información de estilo correspondiente a "en-US", y a continuación la correspondiente al valor más general "en".

Nota. Las jerarquías de códigos de idioma no garantizan que todos los idiomas con un mismo prefijo serán entendidos por aquellos que dominen uno o más de esos idiomas. Lo que hacen es permitir a un usuario englobar varios idiomas bajo un prefijo común cuando sea el caso para ese usuario.

8.2 Especificación de la dirección del texto y de las tablas: el atributo dir

Definiciones de atributos

dir = LTR | RTL [\[CI\]](#)

Este atributo especifica la dirección base del texto direccionalmente neutro (es decir, texto que no tiene direccionalidad heredada según se define en [\[UNICODE\]](#)) de los contenidos o los valores de los atributos de un elemento. También especifica la [direccionalidad de las tablas](#). Valores posibles:

- LTR: Texto o tabla de izquierda a derecha (Left-to-right).
- RTL: Texto o tabla de derecha a izquierda (Right-to-left).

Los autores, además de especificar el idioma de un documento con el atributo [lang](#), pueden necesitar especificar la direccionalidad base (izquierda a derecha o derecha a izquierda) de porciones de texto de un documento, de la estructura de una tabla, etc. Esto se hace con el atributo [dir](#).

La especificación [\[UNICODE\]](#) asigna direccionalidad a los caracteres y define un algoritmo (complejo) para determinar la direccionalidad correcta del texto. Si un documento no contiene un carácter mostrable de derecha a izquierda, no se requiere a un agente de usuario conforme que aplique el algoritmo bidireccional [\[UNICODE\]](#). Si un documento contiene caracteres de derecha a izquierda, y si el agente de usuario muestra estos caracteres, el agente de usuario debe usar el algoritmo bidireccional.

Si bien Unicode especifica caracteres especiales que tratan con la dirección del texto, HTML ofrece construcciones de código de mayor nivel que hacen lo mismo: el atributo [dir](#) (no confundir con el elemento [DIR](#)) y el elemento [BDO](#). Así, para expresar un cita en hebreo, es más intuitivo escribir

```
<Q lang="he" dir="rtl">...una cita en hebreo...</Q>
```

que la referencias equivalentes en Unicode:

```
&#x202B;&#x05F4;...una cita en hebreo...&#x05F4;&#x202C;
```

Los agentes de usuario **no** deben usar el atributo [lang](#) para determinar la direccionalidad del texto.

El atributo [dir](#) se hereda y puede ser anulado. Consulte la sección sobre [información acerca de la herencia de la dirección del texto](#) para más detalles.

8.2.1 Introducción al algoritmo bidireccional

Los siguientes ejemplos ilustran el comportamiento esperado del algoritmo bidireccional. Intervienen dos idiomas: el inglés, con escritura de izquierda a derecha, y el hebreo, con escritura de derecha a izquierda.

Consideremos el siguiente texto de ejemplo:

```
inglés1 HEBREO2 inglés3 HEBREO4 inglés5 HEBREO6
```

Los caracteres de este ejemplo (y de todos los ejemplos relacionados) se almacenan en el ordenador del modo que se muestra aquí: el primer carácter de la fila es "i", el segundo es "n", y el último es "6".

Supongamos que el idioma predominante del documento que contiene a este párrafo es el inglés. Eso significa que la dirección base es de izquierda a derecha. La presentación correcta de esta línea sería:

```
inglés1 2OERBEH inglés3 4OERBEH inglés5 6OERBEH
  <-----          <-----          <-----
          H              H              H
----->
                    I
```

Las líneas a trazos indican la estructura de la frase: el inglés es el que predomina y hay algunos textos incluidos en hebreo. Para lograr la presentación correcta no se necesita código adicional, ya que los agentes de usuario que apliquen el algoritmo bidireccional invertirán correctamente los fragmentos en hebreo.

En cambio, si el idioma predominante del documento es el hebreo, la dirección base es de derecha a izquierda. La presentación correcta es por tanto:

```
6OERBEH inglés5 4OERBEH inglés3 2OERBEH inglés1
  ----->          ----->          ----->
          I              I              I
<-----
                    H
```

En este caso, se ha escrito la frase entera de derecha a izquierda, y las secuencias incluidas en inglés se han invertido apropiadamente con el algoritmo bidireccional.

8.2.2 Herencia de la información sobre la dirección del texto

El algoritmo bidireccional de Unicode obliga a que los bloques de texto tengan una dirección de texto base. Para especificar la dirección base de un elemento en bloque, se establece el atributo [dir](#) del elemento. El valor por defecto del atributo [dir](#) es "ltr" (*left-to-right*, texto de izquierda a derecha).

Cuando se establece el atributo [dir](#) de un elemento en bloque, éste tiene efecto hasta la finalización del bloque y en cualquier elemento en bloque anidado. Si se establece el atributo [dir](#) para algún elemento anidado, se anula en él el valor heredado.

Para establecer la dirección de texto base de un documento completo, se establece el atributo [dir](#) del elemento [HTML](#).

Por ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<HTML dir="RTL">
<HEAD>
<TITLE>...un título de derecha a izquierda...</TITLE>
</HEAD>
...texto de derecha a izquierda...
<P dir="ltr">...texto de izquierda a derecha...</P>
<P>...de nuevo texto de derecha a izquierda...</P>
</HTML>
```

Los elementos en línea, por su parte, no heredan el atributo [dir](#). Eso significa que un elemento en línea sin un atributo [dir](#) **no** abre un nivel adicional de inclusión con respecto al algoritmo bidireccional. (Aquí se considera que un elemento está en línea o en bloque según su presentación por defecto. Téngase en cuenta que los elementos [INS](#) y [DEL](#) pueden estar en bloque o en línea dependiendo de su contexto.)

8.2.3 Especificación de la dirección del texto incluido

El algoritmo bidireccional [\[UNICODE\]](#) invierte automáticamente las secuencias incluidas de caracteres de acuerdo con su direccionalidad inherente (como se ilustra en los ejemplos previos). Sin embargo, en general sólo se puede tener en cuenta un nivel de inclusión. Para lograr niveles adicionales de cambios de dirección por inclusión, debe hacerse uso del atributo [dir](#) de un elemento en línea.

Consideremos el mismo texto del ejemplo anterior:

```
inglés1 HEBREO2 inglés3 HEBREO4 inglés5 HEBREO6
```

Supongamos que el idioma predominante del documento que contiene a este párrafo es el inglés. Supongamos además que la frase en inglés anterior contiene una sección en hebreo que va desde HEBREO2 hasta HEBREO4, y que la sección en hebreo contiene una cita en inglés (inglés3). La presentación deseada de este texto es por lo tanto:

```
inglés1 4OERBEH inglés3 2OERBEH inglés5 6OERBEH
          ----->
              I
          <-----
              H
----->
              I
```

Para lograr dos cambios de dirección por inclusión, debemos proporcionar información adicional, para lo cual delimitamos la segunda inclusión explícitamente. En este ejemplo, utilizamos el elemento [SPAN](#) y el atributo [dir](#) para dar formato al texto:

```
inglés1 <SPAN dir="RTL">HEBREO2 inglés3 HEBREO4</SPAN> inglés5 HEBREO6
```

Los autores también pueden utilizar los caracteres especiales de Unicode para lograr cambios múltiples de dirección por inclusión. Para una inclusión de izquierda a derecha, se rodea el texto incluido con los

caracteres LEFT-TO-RIGHT EMBEDDING (inclusión de izquierda a derecha, "LRE", hexadecimal 202A) y POP DIRECTIONAL FORMATTING (fin de formato direccional, "PDF", hexadecimal 202C). Para una inclusión de derecha a izquierda, se rodea el texto incluido con los caracteres RIGHT-TO-LEFT EMBEDDING (inclusión de derecha a izquierda, "RTE", hexadecimal 202B) y PDF.

Uso conjunto del código de direccionalidad de HTML y de los caracteres Unicode. Los autores y los diseñadores de programas de creación deberían tener en cuenta que pueden surgir conflictos si se utilizan al mismo tiempo el atributo [dir](#) de los elementos en línea (incluyendo [BDO](#)) y los caracteres de formato [\[UNICODE\]](#) correspondientes. Se usará preferiblemente uno u otro sistema exclusivamente. La especificación del formato con HTML ofrece mayores garantías de integridad estructural del documento y evita los problemas que puedan surgir al editar texto bidireccional HTML con un editor de textos sencillo, pero algunos programas pueden ser más aptos para la utilización de los caracteres [\[UNICODE\]](#). Si se usan ambos métodos, deberían tomarse todas las precauciones para asegurar el anidamiento correcto del código y los caracteres de formato; de otro modo los resultados de la representación son imprevisibles.

8.2.4 Anulación del algoritmo bidireccional: el elemento BDO

```
<!ELEMENT BDO - - (%inline;)* -- anular BiDi I18N -->
<!ATTLIST BDO
  %coreattrs; -- id, class, style, title --
  lang %LanguageCode; #IMPLIED -- código de idioma --
  dir (ltr|rtl) #REQUIRED -- direccionalidad --
>
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria***

Definiciones de atributos

dir = LTR | RTL [\[CI\]](#)

Este atributo obligatorio especifica la dirección base del texto contenido en el elemento. Esta dirección prevalece sobre la direccionalidad inherente de los caracteres según se define en [\[UNICODE\]](#). Valores posibles:

- LTR: Texto de izquierda a derecha (Left-to-right).
- RTL: Texto de derecha a izquierda (Right-to-left).

Atributos definidos en otros lugares

- [lang \(información sobre el idioma\)](#)

En general, el algoritmo bidireccional y el atributo [dir](#) son suficientes para controlar los cambios de direccionalidad por inclusión. Sin embargo, a veces pueden darse casos en que el algoritmo bidireccional produce una presentación incorrecta. El elemento [BDO](#) permite a los autores desactivar el algoritmo bidireccional para fragmentos de texto específicos.

Consideremos un documento que contenga el mismo texto que antes:

inglés1 HEBREO2 inglés3 HEBREO4 inglés5 HEBREO6

pero supongamos que este texto ya ha sido puesto en su orden visual. Una razón para ello podría ser que

los estándares MIME ([\[RFC2045\]](#), [\[RFC1556\]](#)) recomiendan el orden visual, es decir, que las secuencias de caracteres de derecha a izquierda se inserten de derecha a izquierda en el flujo de caracteres. En un mensaje de correo electrónico, se podría dar el siguiente formato al texto anterior, incluyendo saltos de línea:

```
inglés1 2OERBEH inglés3
4OERBEH inglés5 6OERBEH
```

Esto generaría un conflicto con el algoritmo bidireccional [\[UNICODE\]](#), ya que el algoritmo invertiría 2OERBEH, 4OERBEH, y 6OERBEH una segunda vez, mostrando las palabras en hebreo de izquierda a derecha en lugar de derecha a izquierda.

La solución en este caso es anular el algoritmo de bidireccionalidad colocando el extracto del mensaje de correo en un elemento [PRE](#) (para conservar los saltos de línea) y cada línea en un elemento [BDO](#), cuyo atributo [dir](#) sea igual a LTR:

```
<PRE>
<BDO dir="LTR">inglés1 2OERBEH inglés3</BDO>
<BDO dir="LTR">4OERBEH inglés5 6OERBEH</BDO>
</PRE>
```

Esto le dice al algoritmo bidireccional "¡Déjame de izquierda a derecha!" y produce la presentación deseada:

```
inglés1 2OERBEH inglés3
4OERBEH inglés5 6OERBEH
```

El elemento [BDO](#) debería utilizarse en aquellos casos en que fuera preciso un control absoluto del orden de las secuencias (p.ej., números de referencia multi-lenguaje). El atributo [dir](#) es obligatorio para este elemento.

Los autores también pueden usar los caracteres especiales Unicode para anular el algoritmo bidireccional -- LEFT-TO-RIGHT OVERRIDE (anular izquierda a derecha, 202D) o RIGHT-TO-LEFT OVERRIDE (anular derecha a izquierda, hexadecimal 202E). El carácter POP DIRECTIONAL FORMATTING (hexadecimal 202C) finaliza ambas anulaciones de bidireccionalidad.

***Nota.** Recuérdese que pueden surgir conflictos si se usan simultáneamente el atributo [dir](#) de los elementos en línea (incluyendo [BDO](#)) y los caracteres de formato [\[UNICODE\]](#) correspondientes.*

***Bidireccionalidad y codificaciones de caracteres.** De acuerdo con [\[RFC1555\]](#) y [\[RFC1556\]](#), existen convenciones especiales sobre el uso del parámetro "charset" para indicar el tratamiento de la bidireccionalidad que se hace en el correo MIME, en particular para distinguir entre direccionalidad visual, implícita y explícita. El valor "ISO-8859-8" (para hebreo) indica codificación visual, "ISO-8859-8-i" indica bidireccionalidad implícita, e "ISO-8859-8-e" indica direccionalidad explícita.*

Como HTML usa el algoritmo de bidireccionalidad de Unicode, los documentos conformes codificados con la ISO 8859-8 deben ser marcados como "ISO-8859-8-i". Con HTML también es posible el control direccional explícito, pero no puede expresarse con ISO 8859-8, y por tanto no debería utilizarse "ISO-8859-8-e" .

El valor "ISO-8859-8" implica que se ha dado formato visual al documento, utilizando código

*incorrectamente (como por ejemplo [TABLE](#) con alineación a la derecha y sin ajuste automático de líneas) para asegurarse de que los agentes de usuario antiguos que no soportan bidireccionalidad puedan realizar una presentación razonable del documento. Tales documentos no son conformes con la especificación presente. Si es necesario, pueden hacerse conformes con esta especificación (y al mismo tiempo se mostrarán correctamente en agentes de usuario antiguos) añadiendo elementos [BDO](#) donde haga falta. Al contrario de lo que se dice en [\[RFC1555\]](#) y en [\[RFC1556\]](#), ISO-8859-6 (Árabe) **no** se ordena visualmente.*

8.2.5 Referencias de caracteres para el control de la direccionalidad y de la unión

Dado que a veces se dan ambigüedades en la direccionalidad de ciertos caracteres (p.ej., los de puntuación), la especificación [\[UNICODE\]](#) incluye caracteres que permiten la resolución apropiada de dichas ambigüedades. Además, Unicode incluye algunos caracteres para controlar el comportamiento de unión y separación de caracteres cuando esto sea necesario (p.ej., algunas situaciones con letras árabes). HTML 4 incluye [referencias de caracteres](#) para estos caracteres.

Los siguientes extractos del DTD presentan algunas de las entidades direccionales:

```
<!ENTITY zwnj CDATA "&#8204;"--=separador de anchura cero-->
<!ENTITY zwj CDATA "&#8205;"--=conector de anchura cero-->
<!ENTITY lrm CDATA "&#8206;"--=marca de izquierda a derecha-->
<!ENTITY rlm CDATA "&#8207;"--=marca de derecha a izquierda-->
```

La entidad `zwnj` (*zero width non-joiner*) se utiliza para evitar el comportamiento de unión en aquellos contextos en que la unión ocurre pero no debería. La entidad `zwj` (*zero width joiner*) hace lo contrario: fuerza la unión cuando no ocurre aunque debería. Por ejemplo, la letra árabe "HEH" se utiliza para abreviar "Hijri", el nombre del sistema de calendario islámico. Como la "HEH" aislada se parece a la escritura manual del dígito árabe cinco (basada en dígitos índicos), para evitar confundir "HEH" con un dígito cinco al final de un año, se usa la forma inicial de "HEH". Sin embargo, no sigue ningún contexto (es decir, una letra de unión) al que pueda unirse la "HEH". El carácter `zwj` proporciona ese contexto.

Análogamente, en textos persas, hay casos es que una letra que normalmente se uniría a la letra subsiguiente, en escritura cursiva no debería hacerlo. En estos casos se utiliza el carácter `zwj` para evitar la unión.

Los otros caracteres, `lrm` y `rlm`, se usan para forzar la direccionalidad de caracteres de direccionalidad neutral. Por ejemplo, si hay un signo de comillas dobles entre un texto árabe (derecha a izquierda) y otro en latín (izquierda a derecha), la dirección de las comillas no está clara (¿corresponden al texto árabe o al latín?). Los caracteres `lrm` y `rlm` tienen propiedad direccional, pero no propiedades de anchura ni de separación de palabras o líneas. Consulte [\[UNICODE\]](#) para más detalles.

Signos de caracteres reflejados. *En general, el algoritmo bidireccional no refleja signos de caracteres, sino que los deja intactos. Una excepción a esto son los caracteres tales como los paréntesis (ver [\[UNICODE\]](#), tabla 4-7). En aquellos casos en que se desee reflejar signos, por ejemplo para jeroglíficos egipcios, o en boustrophedon griego, o para efectos especiales de diseño, debería controlarse esto con hojas de estilo.*

8.2.6 Efecto de las hojas de estilo en la bidireccionalidad

En general, la utilización de hojas de estilo para cambiar la representación visual de un elemento en bloque a en línea o viceversa no ofrece ninguna dificultad. Sin embargo, debido a que el algoritmo bidireccional se basa en la [distinción en línea/en bloque](#), debe tenerse especial cuidado si se hace esta transformación.

Cuando un elemento en línea que no tiene un atributo [dir](#) se transforma mediante una hoja de estilo al estilo de un elemento en bloque, hereda el atributo [dir](#) del elemento padre en bloque más cercano que defina la dirección base del bloque.

Cuando un elemento en bloque que no tiene un atributo [dir](#) se transforma mediante una hoja de estilo al estilo de un elemento en línea, la presentación resultante debería ser equivalente, en términos de formato de la bidireccionalidad, al formato obtenido añadiendo explícitamente al elemento transformado un atributo [dir](#) (igual al valor heredado).

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

9 Texto

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Espacio en blanco](#)
2. [Texto estructurado](#)
 1. [Elementos de frase: EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, ABBR, y ACRONYM](#)
 2. [Citas: los elementos BLOCKQUOTE y Q](#)
 - [Representación de citas](#)
 3. [Subíndices y superíndices: los elementos SUB y SUP](#)
3. [Líneas y párrafos](#)
 1. [Párrafos: el elemento P](#)
 2. [Control de los saltos de línea](#)
 - [Forzar un salto de línea: el elemento BR](#)
 - [Impedir un salto de línea](#)
 3. [División de palabras](#)
 4. [Texto preformateado: el elemento PRE](#)
 5. [Representación visual de los párrafos](#)
4. [Anotación de cambios en los documentos: los elementos INS y DEL](#)

Las siguientes secciones tratan sobre temas relativos a la estructuración del texto. Los elementos relacionados con la [presentación del texto](#) (elementos de alineación, elementos de fuentes, hojas de estilo, etc.) se tratan en otros lugares de la especificación. Para información sobre caracteres, consulte la sección sobre [el conjunto de caracteres del documento](#).

9.1 Espacio en blanco

El [conjunto de caracteres del documento](#) incluye una gran variedad de caracteres de espacio en blanco. Gran parte de ellos son elementos tipográficos utilizados en algunas aplicaciones para producir ciertos efectos visuales de espaciado. En HTML, sólo se definen como caracteres de espacio en blanco los siguientes:

- Espacio ASCII ()
- Tabulador ASCII ()
- Avance de página ASCII (form feed, )
- Espacio de anchura cero (​)

Los [saltos de línea](#) también son caracteres de espacio en blanco. Obsérvese que si bien la [ISO10646](#) define   y   como separaciones inequívocas de líneas y párrafos respectivamente, estos caracteres no constituyen saltos de línea en HTML, y esta especificación no los incluye en la categoría general de caracteres de espacio en blanco.

Esta especificación no indica el comportamiento, en cuanto a su representación o a otros aspectos, de otros caracteres de espacio que no sean los aquí identificados explícitamente como caracteres de espacio en blanco. Por esta razón, los autores deberían utilizar los elementos y estilos apropiados, y no caracteres de espacio, para lograr efectos de formato visual relacionados con el espacio en blanco.

Para todos los elementos HTML, excepto para el elemento [PRE](#), las secuencias de espacio en blanco separan "palabras" (con el término "palabra" queremos decir "secuencias de caracteres que no son de espacio en blanco"). Los agentes de usuario, al dar formato al texto, deberían identificar estas palabras y mostrarlas de acuerdo con las convenciones del idioma escrito y del medio destino.

Esta presentación puede implicar la colocación de espacio entre palabras, pero las convenciones relativas al espacio varían de un idioma a otro. Por ejemplo, en escritura latina, el espacio entre palabras se suele representar mediante un espacio ASCII (), mientras que en tailandés se usa un separador de palabras de anchura cero (​). En japonés y en chino el espacio entre palabras normalmente no se representa de ninguna manera.

Obsérvese que una secuencia de espacios en blanco entre palabras en el documento fuente puede ser representada como un espacio entre palabras completamente diferente (excepto en el caso del elemento [PRE](#)). En particular, los agentes de usuario deberían colapsar las secuencias de espacios en blanco de la entrada al producir la salida del espacio entre palabras. Esto puede y debería hacerse incluso en ausencia de información sobre el idioma (proveniente del atributo [lang](#), el campo de encabezado HTTP "Content-Language" (ver [RFC2616](#), sección 14.12), la configuración del agente de usuario, etc.).

El elemento [PRE](#) se utiliza para [texto preformateado](#), en el cual el espacio en blanco es significativo.

Para evitar problemas con las [reglas de saltos de línea de SGML](#) e inconsistencias entre las implementaciones existentes, los autores no deberían suponer que los agentes de usuario representarán el espacio en blanco que esté inmediatamente después de una etiqueta inicial o inmediatamente antes de una etiqueta final. Así, los autores, y en particular las herramientas de creación, deberían escribir:

```
<P>Ofrecemos <A>soporte técnico</A> gratuito a nuestros suscriptores.</P>
```

y no:

```
<P>Ofrecemos<A> soporte técnico </A>gratuito para nuestros suscriptores.</P>
```

9.2 Texto estructurado

9.2.1 Elementos de frase: EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, ABBR, y ACRONYM

```
<!ENTITY % frase "EM | STRONG | DFN | CODE |
                SAMP | KBD | VAR | CITE | ABBR | ACRONYM" >
<!ELEMENT ( %fontstyle; | %phrase; ) - - ( %inline; ) * >
<!ATTLIST ( %fontstyle; | %phrase; )
  %attrs;                -- %coreattrs, %il8n, %events --
>
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria***

Atributos definidos en otros lugares

- [id](#), [class](#) ([identificadores a nivel de documento](#))
- [lang](#) ([información sobre el idioma](#)), [dir](#) ([dirección del texto](#))
- [title](#) ([título del elemento](#))
- [style](#) ([información de estilo en línea](#))
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) ([eventos intrínsecos](#))

Los elementos de frase añaden información estructural a fragmentos de texto. Los significados normales de los elementos de frase son los siguientes:

EM:

Párrafos, línea y frases

Indica énfasis (*emphasis*).

STRONG:

Indica un énfasis más fuerte.

CITE:

Contiene una cita o una referencia a otras fuentes.

DFN:

Indica que aquí es donde se define el término encerrado.

CODE:

Designa un fragmento de código de computadora.

SAMP:

Designa una muestra de la salida de un programa, script, etc.

KBD:

Indica texto que debe ser introducido por el usuario.

VAR:

Indica que el texto es una variable o un argumento de un programa.

ABBR:

Indica una forma abreviada (p.ej., WWW, HTTP, URI, Mass., etc.).

ACRONYM:

Indica un acrónimo (p.ej., WAC, radar, etc.).

[EM](#) y [STRONG](#) se usan para indicar énfasis. Los demás elementos de frase tienen significados particulares en documentos técnicos. Estos ejemplos ilustran algunos de los elementos de frase:

```
Como dijo <CITE>Harry S. Truman</CITE>,  
<Q lang="en-us">The buck stops here.</Q>
```

Se puede encontrar más información en <CITE>[ISO-0000]</CITE>.

Por favor, utilice el siguiente número de referencia en nuestra correspondencia futura: 1-234-55

La presentación de los elementos de frase depende del agente de usuario. En general, los agentes de usuario presentan el texto [EM](#) en itálica y el texto [STRONG](#) con una fuente negrita. Los sintetizadores de voz pueden cambiar los parámetros de síntesis, como el volumen, el tono y la velocidad.

Los elementos [ABBR](#) y [ACRONYM](#) permiten a los autores indicar claramente la aparición de abreviaturas y acrónimos. Los idiomas occidentales hacen uso extensivo de acrónimos tales como "GmbH", "NATO", y "F.B.I.", así como de abreviaturas como "M.", "Inc.", "et al.", "etc.". Tanto en chino como en japonés se utilizan mecanismos de abreviación análogos, por los cuales las referencias subsiguientes a un nombre largo se realizan con un subconjunto de los caracteres Han del nombre original. Al marcar estas estructuras se proporciona información útil a los agentes de usuario y a herramientas tales como correctores ortográficos, sintetizadores de voz, sistemas de traducción e indexadores de motores de búsqueda.

El contenido de los elementos [ABBR](#) y [ACRONYM](#) especifica la expresión abreviada, tal y como aparece en el texto. Puede utilizarse el atributo title de estos elementos para proporcionar la forma completa o expandida de la expresión.

Aquí tenemos algunos ejemplos de [ABBR](#):

```
<P>  
<ABBR title="World Wide Web">WWW</ABBR>  
<ABBR lang="fr"  
  title="Soci&eacute;t&eacute; Nationale des Chemins de Fer">  
  SNCF  
</ABBR>  
<ABBR lang="es" title="Do&ntilde;a">Do&ntilde;a</ABBR>
```

```
<ABBR title="Abbreviation">abbr.</ABBR>
```

Obsérvese que las abreviaturas y los acrónimos tienen a menudo pronunciaciones idiosincráticas. Por ejemplo, mientras que "IRS" y "BBC" se suelen pronunciar letra por letra, "OTAN" y "UNESCO" se pronuncian fonéticamente. Y hay otras formas abreviadas (p.ej., "URI" y "SCSI") que algunas personas deletrean y que otras pronuncian como palabras. Cuando sea necesario, los autores deberían usar hojas de estilo para especificar la pronunciación de una forma abreviada.

9.2.2 Citas: los elementos BLOCKQUOTE y Q

```
<!ELEMENT BLOCKQUOTE - - (%block;|SCRIPT)+ -- cita larga -->
<!ATTLIST BLOCKQUOTE
  %attrs;          -- %coreattrs, %i18n, %events --
  cite             %URI;          #IMPLIED -- URI del documento o mensaje original --
  >
<!ELEMENT Q - - (%inline;)* -- cita corta en línea -->
<!ATTLIST Q
  %attrs;          -- %coreattrs, %i18n, %events --
  cite             %URI;          #IMPLIED -- URI del documento o mensaje original --
  >
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria***

Definiciones de atributos

cite = [uri](#) [CT]

El valor de este atributo es un URI que designa un documento o mensaje original. Este atributo sirve para dar información sobre la fuente de la cual se tomó prestada la cita.

Atributos definidos en otros lugares

- [id](#), [class](#) ([identificadores a nivel de documento](#))
- [lang](#) ([información sobre el idioma](#)), [dir](#) ([dirección del texto](#))
- [title](#) ([título del elemento](#))
- [style](#) ([información de estilo en línea](#))
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) ([eventos intrínsecos](#))

Estos dos elementos designan citas de texto. [BLOCKQUOTE](#) es para citas largas (contenido en bloque) y [Q](#) sirve para citas cortas (contenido en línea) que no necesitan cambios de párrafo.

Este ejemplo da formato a un extracto de "The Two Towers", de J.R.R. Tolkien, como un cita en bloque.

```
<BLOCKQUOTE cite="http://www.micom.com/tolkien/twotowers.html">
<P>They went in single file, running like hounds on a strong scent,
and an eager light was in their eyes. Nearly due west the broad
swath of the marching Orcs tramped its ugly slot; the sweet grass
of Rohan had been bruised and blackened as they passed.</P>
</BLOCKQUOTE>
```

Representación de citas

Los agentes de usuario visuales representan generalmente un [BLOCKQUOTE](#) como un bloque con sangría.

Los agentes de usuario visuales deben asegurarse de que el contenido del elemento [Q](#) se representa con marcas de citación a su alrededor. Los autores no deberían colocar marcas de citación al principio y al final del contenido de un elemento [Q](#).

Los agentes de usuario deberían representar las marcas de citación en función del idioma (ver el atributo [lang](#)). Hay muchos idiomas que adoptan diferentes estilos de citación para las citas externas y las internas (anidadas), lo cual debería

ser respetado por los agentes de usuario.

El siguiente ejemplo ilustra las citas anidadas con el elemento [Q](#).

```
John said, <Q lang="en-us">I saw Lucy at lunch, she told me
<Q lang="en-us">Mary wants you to get some ice cream on your
way home.</Q> I think I will get some at Ben and Jerry's, on
Gloucester Road.</Q>
```

Como el idioma de ambas citas en inglés americano, los agentes de usuario deberían representarlos de forma apropiada, por ejemplo, con comillas simples alrededor de la cita interior y con comillas dobles alrededor de la cita exterior:

```
John said, "I saw Lucy at lunch, she told me 'Mary wants you
to get some ice cream on your way home.' I think I will get some
at Ben and Jerry's, on Gloucester Road."
```

Nota. Recomendamos que las implementaciones de hojas de estilo proporcionen un mecanismo para insertar signos de puntuación de citas antes y después de una cita delimitada por un [BLOCKQUOTE](#) de un modo apropiado según el contexto del idioma actual y el grado de anidamiento de las citas.

Sin embargo, debido a que algunos autores han usado [BLOCKQUOTE](#) simplemente como un mecanismo para dejar márgenes a los lados de un texto, y para preservar la intención de los autores, los agentes de usuario **no** deberían insertar signos de puntuación de citas en el estilo por defecto.

El uso de [BLOCKQUOTE](#) para dar márgenes al texto está [desaprobado](#) en favor de las hojas de estilo.

9.2.3 Subíndices y superíndices: los elementos SUB y SUP

```
<!ELEMENT (SUB|SUP) - - (%inline;)* -- subíndice, superíndice -->
<!ATTLIST (SUB|SUP)
  %attrs; -- %coreattrs, %il8n, %events --
>
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

En muchos idiomas (p.ej., en francés) se necesitan los subíndices y los superíndices para una representación apropiada. En estos casos deberían emplearse los elementos [SUB](#) y [SUP](#).

```
H<sub>2</sub>O
E = mc<sup>2</sup>
<SPAN lang="fr">M<sup>lle</sup> Dupont</SPAN>
```

9.3 Líneas y párrafos

Tradicionalmente los autores dividen sus pensamientos y argumentos en secuencias de párrafos. La organización de la información en párrafos no se ve afectada por el modo en que se presentan los párrafos: los párrafos que están doblemente justificados contienen los mismos pensamientos que aquellos que están justificados a la izquierda.

La forma en que HTML *define* un párrafo es directa: el elemento [P](#) define un párrafo.

La presentación visual de los párrafos no es tan simple. Hay que tener en cuenta varios aspectos, tanto estilísticos como técnicos:

- Tratamiento del espacio en blanco
- Saltos de línea y cambio automático de línea
- Justificación
- División de palabras
- Convenciones del idioma escrito y direccionalidad del texto
- Formateo de los párrafos con respecto al contenido circundante

Todas estas cuestiones las tratamos a continuación. [La alineación de párrafos y objetos flotantes](#) se trata más adelante en este documento.

9.3.1 Párrafos: el elemento P

```
<!ELEMENT P - O (%inline;)*           -- párrafo -->
<!ATTLIST P
  %attrs;                               -- %coreattrs, %i18n, %events --
>
```

Etiqueta inicial: obligatoria, Etiqueta final: opcional

Atributos definidos en otros lugares

- [id](#), [class](#) ([identificadores a nivel de documento](#))
- [lang](#) ([información sobre el idioma](#)), [dir](#) ([dirección del texto](#))
- [title](#) ([título del elemento](#))
- [style](#) ([información de estilo de línea](#))
- [align](#) ([alineación](#))
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) ([eventos intrínsecos](#))

El elemento [P](#) representa un párrafo. No puede contener [elementos en bloque](#) (incluyendo a otros [P](#)).

Desaconsejamos a los autores el uso de elementos [P](#) vacíos. Los agentes de usuario no deberían tener en cuenta los elementos [P](#) vacíos.

9.3.2 Control de los saltos de línea

Un salto de línea se define como un retorno de carro (), un avance de línea (
), o una pareja retorno de carro/avance de línea. Todos los saltos de línea constituyen [espacio en blanco](#).

Para más información sobre la especificación SGML en lo que se refiere a los saltos de línea, consulte las [notas sobre los saltos de línea](#) del apéndice.

Forzar un salto de línea: el elemento BR

```
<!ELEMENT BR - O EMPTY                 -- salto de línea forzado -->
<!ATTLIST BR
  %coreattrs;                           -- id, class, style, title --
>
```

Etiqueta inicial: obligatoria, Etiqueta final: prohibida

Atributos definidos en otros lugares

- [id, class](#) (identificadores a nivel de documento)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [clear](#) (alineación y objetos flotantes)

El elemento [BR](#) rompe (finaliza) forzosamente la línea actual de texto.

Para agentes de usuario visuales, puede utilizarse el atributo [clear](#) para determinar si el código que sigue al elemento [BR](#) fluye alrededor de las imágenes u otros objetos que estén flotando en el margen izquierdo o derecho, o si comenzará debajo de dichos objetos. Pueden encontrarse más detalles en la sección sobre [alineación y objetos flotantes](#). Se aconseja a los autores que usen hojas de estilo para controlar el flujo de texto alrededor de imágenes y otros objetos.

Con respecto al formato bidireccional, el elemento [BR](#) debería comportarse del mismo modo que se comporta el carácter de [ISO10646](#) LINE SEPARATOR en el algoritmo bidireccional.

Impedir un salto de línea

En algunas ocasiones los autores pueden querer evitar que se produzca un salto de línea entre dos palabras. La entidad ` ` (` `; o ` `) actúa como un espacio en el cual los agentes de usuario no deberían colocar un salto de línea.

9.3.3 Separación de palabras

En HTML hay dos tipos de guiones: el guión normal y el guión de separación. El guión normal debería ser interpretado por un agente de usuario exactamente igual que cualquier otro carácter. El guión de separación le dice al agente de usuario dónde puede producirse un salto de línea.

Aquellos navegadores que interpreten los guiones de separación deben respetar la siguiente semántica: Si una línea se rompe por un guión de separación, debe mostrarse un guión al final de la primera línea. Si no se produce un salto de línea en un guión de separación, el agente de usuario no debe mostrar el carácter de guión. Para operaciones tales como búsqueda y ordenación, el guión de separación no debería ser tenido nunca en cuenta.

En HTML el guión normal se representa mediante el carácter "-" (`-`; o `-`). El guión de separación se representa mediante la referencia a entidad de caracteres `­` (`­`; o `­`)

9.3.4 Texto preformateado: el elemento PRE

```
<!ENTITY % pre.exclusion "IMG|OBJECT|BIG|SMALL|SUB|SUP">
```

```
<!ELEMENT PRE - - (%inline;)* -(%pre.exclusion;) -- texto preformateado -->
```

```
<!ATTLIST PRE
```

```
  %attrs; -- %coreattrs, %i18n, %events --
```

```
>
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria***

Definiciones de atributos

width = [number](#) [CN]

Desaprobado. Este atributo proporciona a los agentes de usuario visuales información sobre la anchura deseada del bloque formateado. El agente de usuario puede usar esta información para seleccionar un tamaño de fuente apropiado o para dar un margen apropiado al contenido. La anchura deseada se expresa como un número de caracteres. Este atributo no está ampliamente soportado en la actualidad.

Atributos definidos en otros lugares

- [id, class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)

- [style](#) ([información de estilo en línea](#))
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) ([eventos intrínsecos](#))

El elemento [PRE](#) le dice a los agentes de usuario visuales que el texto contenido está "preformateado". Al tratar con texto preformateado, los agentes de usuario visuales:

- Pueden dejar el [espacio en blanco](#) intacto.
- Pueden representar el texto con una fuente de ancho fijo.
- Pueden desactivar el ajuste automático de líneas (*word wrap*).
- No deben desactivar el procesamiento bidireccional.

No es necesario que los agentes de usuario no visuales respeten el [espacio en blanco](#) adicional presente en el contenido de un elemento [PRE](#).

Para más información sobre la especificación SGML en lo que respecta a los saltos de línea, consulte las [notas sobre los saltos de línea](#) del apéndice.

El fragmento del DTD mostrado anteriormente indica qué elementos no pueden aparecer dentro de una declaración [PRE](#). Son los mismos que en HTML 3.2, y la intención es mantener constantes el espaciado de líneas y la alineación de columnas de texto representado con una fuente de ancho fijo. Se desaconseja a los autores alterar este comportamiento a través de hojas de estilo.

El siguiente ejemplo muestra la primera estrofa preformateada del poema de Gabriel Celaya, *A Pablo Neruda*:

```
<PRE>
    Te escribo desde un puerto.
    La mar salvaje llora.
Salvaje, y triste, y solo te escribo abandonado.
Las olas funerales redoblan el vacío.
Los megáfonos llaman a través de la niebla.
La pálida corola de la lluvia me envuelve.
    Te escribo desolado.
</PRE>
```

Así es como se representaría esto en un caso típico:

```
    Te escribo desde un puerto.
    La mar salvaje llora.
Salvaje, y triste, y solo te escribo abandonado.
Las olas funerales redoblan el vacío.
Los megáfonos llaman a través de la niebla.
La pálida corola de la lluvia me envuelve.
    Te escribo desolado.
```

El carácter de tabulación horizontal

Los agentes de usuario visuales suelen interpretar el carácter de tabulación horizontal (decimal 9 en [\[ISO10646\]](#) y [\[ISO88591\]](#)) como el número más pequeño distinto de cero de espacios necesarios para alinear los caracteres con las paradas del tabulador, colocadas cada 8 caracteres. Recomendamos encarecidamente no utilizar tabuladores horizontales en texto preformateado, ya que al editar es muy normal fijar el espacio de tabulación en valores diferentes, lo cual conduce a documentos mal alineados.

9.3.5 Representación visual de los párrafos

Nota. *La sección siguiente es una descripción informativa del comportamiento de algunos agentes de usuario visuales al dar formato a los párrafos. Las hojas de estilo tendrán un control mejor sobre el formato de los párrafos.*

El modo en que los párrafos se representan visualmente depende del agente de usuario. Normalmente los párrafos se representan alineados a la izquierda y sin alinear por la derecha. Para otras escrituras de derecha a izquierda son

apropiadas otras presentaciones por defecto.

Los agentes de usuario HTML han representado tradicionalmente los párrafos con espacio en blanco antes y después, p.ej.,

Seguíale Sancho a pie, llevando, como tenía de costumbre, del cabestro a su jumento, perpetuo compañero de sus prósperas y adversas fortunas; y habiendo andado una buena pieza por entre aquellos castaños y árboles sombríos, dieron en un pradillo que al pie de unas altas peñas se hacía, de las cuales se precipitaba un grandísimo golpe de agua.

Al pie de las peñas estaban unas casas mal hechas, que más parecían ruinas de edificios que casas, de entre las cuales advirtieron que salía el ruido y estruendo de aquel golpear, que aún no cesaba.

Esto contrasta con el estilo en las novelas, en las cuales la primera línea de cada párrafo tiene sangría, y entre la última línea del párrafo actual y la primera del siguiente se mantiene el espacio entre líneas normal, p.ej.,

Seguíale Sancho a pie, llevando, como tenía de costumbre, del cabestro a su jumento, perpetuo compañero de sus prósperas y adversas fortunas; y habiendo andado una buena pieza por entre aquellos castaños y árboles sombríos, dieron en un pradillo que al pie de unas altas peñas se hacía, de las cuales se precipitaba un grandísimo golpe de agua.

Al pie de las peñas estaban unas casas mal hechas, que más parecían ruinas de edificios que casas, de entre las cuales advirtieron que salía el ruido y estruendo de aquel golpear, que aún no cesaba.

Los agentes de usuario, siguiendo el precedente sentado por el navegador NCSA Mosaic en 1993, no suelen justificar ambos márgenes, en parte porque es difícil de hacer sin sofisticadas rutinas de división de palabras. La llegada de las hojas de estilo y de las fuentes suavizadas con posicionamiento más preciso que el píxel, promete a los autores HTML posibilidades más ricas de lo que lo han sido hasta ahora.

Las hojas de estilo proporcionan un mejor control sobre el tamaño y estilo de la fuente, los márgenes, los espacios antes y después de los párrafos, la sangría de la primera línea, la justificación, y otros muchos detalles. La hoja de estilo por defecto del agente de usuario representa los elementos [P](#) de una forma familiar, como se ilustra más arriba. Se podría, en principio, redefinir esto para representar los párrafos sin los espacios que separan convencionalmente los párrafos sucesivos. En general, como esto puede confundir a los lectores, desaconsejamos esta práctica.

Convencionalmente, los agentes de usuario de HTML visuales ajustan automáticamente las líneas de texto a la anchura disponible entre los márgenes. Los algoritmos de ajuste automático de línea dependen del tipo de escritura a que se esté dando formato.

En las escrituras occidentales, por ejemplo, el texto sólo debería partirse por un espacio en blanco. Los agentes de usuario antiguos partían las líneas incorrectamente justo después de la etiqueta inicial o justo antes de la etiqueta final de un elemento, lo cual dejaba algunos signos de puntuación colgados. Por ejemplo, considérese esta frase:

Una estatua de los [Cihuateteus](#), que son los patrones ...

Si se parte la línea justo antes de la etiqueta final del elemento [A](#), la coma saldrá el comienzo de la siguiente línea:

Una estatua de los Cihuateteus
, que son los patrones ...

Esto es un error, ya que no había espacio en blanco en ese punto del código.

9.4 Anotación de cambios en los documentos: los elementos INS y DEL

```
<!-- INS/DEL se utilizan incluyéndolos en BODY -->
<!ELEMENT (INS|DEL) - - (%flow;)* -- texto insertado, texto borrado -->
<!ATTLIST (INS|DEL)
  %attrs; -- %coreattrs, %i18n, %events --
  cite %URI; #IMPLIED -- información sobre la razón del cambio --
  datetime %Datetime; #IMPLIED -- fecha y hora del cambio --
>
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

Definiciones de atributos

cite = [uri](#) [CT]

El valor de este atributo es un URI que designa un documento o mensaje fuente. Este atributo sirve para hacer referencia a la información que explica por qué se modificó un documento.

datetime = [datetime](#) [CS]

El valor de este atributo especifica la fecha y la hora en que se hizo el cambio.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

[INS](#) y [DEL](#) se usan para marcar secciones del documento que han sido insertadas o borradas con respecto a una versión diferente de un documento (p.ej., en el borrador de una ley cuyos cambios necesitan ver los legisladores).

Estos dos elementos son especiales dentro de HTML, ya que pueden actuar o bien como elementos en bloque o bien como elementos en línea (pero no como ambos a la vez). Pueden contener una o más palabras dentro de un párrafo o uno o más elementos en bloque como párrafos, listas y tablas.

Este ejemplo podría estar sacado de una enmienda para cambiar la legislación relativa a cuántos tenientes de alcalde puede haber en un Ayuntamiento determinado, de 8 a 9.

```
<P>
  El alcalde nombrará a <DEL>8</DEL><INS>9</INS> tenientes de alcalde.
</P>
```

Los elementos [INS](#) y [DEL](#) no deben contener contenido en bloque cuando estos elementos se comporten como elementos en línea.

EJEMPLO ILEGAL:

Lo siguiente no es HTML legal.

```
<P>
<INS><DIV>...contenido en bloque...</DIV></INS>
</P>
```

Los agentes de usuario deberían representar el texto insertado y borrado de manera que los cambios sean obvios. Por ejemplo, el texto insertado pueden aparecer con una fuente especial, el texto borrado pueden no aparecer o aparecer tachado o con marcas especiales, etc.

Estos dos ejemplos corresponden al 5 de noviembre de 1994, a las 8:15:30 am, US Eastern Standard Time.

```
1994-11-05T13:15:30Z
1994-11-05T08:15:30-05:00
```

Con un elemento [INS](#), esto da:

```
<INS datetime="1994-11-05T08:15:30-05:00"
      cite="http://www.blabla.org/midoc/comentarios.html">
```

Además, las últimas estadísticas del departamento de publicidad sugieren que estas prácticas están en alza.

```
</INS>
```

El documento "http://www.blabla.org/midoc/comentarios.html" contendría comentarios sobre por qué se incluyó la información en el documento.

Los autores también pueden optar por comentar los textos insertados o borrados por medio del atributo [title](#) de los elementos [INS](#) y [DEL](#). Los agentes de usuario pueden presentar esta información a los usuarios (p.ej., con un nota flotante). Por ejemplo:

```
<INS datetime="1994-11-05T08:15:30-05:00"
      title="Cambiado tras los comentarios de Steve B en la reunión.">
```

Además, las últimas estadísticas del departamento de publicidad sugieren que estas prácticas están en alza.

```
</INS>
```

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

10 Listas

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Introducción a las listas](#)
2. [Listas no ordenadas \(UL\), listas ordenadas \(OL\) y objetos de lista \(LI\)](#)
3. [Listas de definiciones: los elementos DL, DT y DD](#)
 1. [Representación visual de las listas](#)
4. [Los elementos DIR y MENU](#)

10.1 Introducción a las listas

HTML ofrece a los autores varios mecanismos para especificar listas de información. Todas las listas deben contener uno o más objetos de lista. Las listas pueden contener:

- Información no ordenada.
- Información ordenada.
- Definiciones.

La lista anterior, por ejemplo, es una lista no ordenada, creada con el elemento [UL](#):

```
<UL>
<LI>Información no ordenada.
<LI>Información ordenada.
<LI>Definiciones.
</UL>
```

Una lista ordenada, creada por medio del elemento [OL](#), debería contener información para la cual debe ponerse énfasis en el orden, como en una receta de cocina:

1. Mezcle los ingredientes secos íntimamente.
2. Vierta los ingredientes líquidos.
3. Remueva durante 10 minutos.
4. Hornee durante una hora a 300 grados.

Las listas de definiciones, creadas por medio del elemento [DL](#), consisten generalmente en una serie de parejas término/definición (si bien las listas de definiciones pueden tener otras aplicaciones). Por ejemplo, se podría usar una lista de definiciones en la publicidad de un producto:

Menor coste

¡La nueva versión de este producto cuesta mucho menos que la versión anterior!

Más fácil de usar

¡Hemos cambiado el producto para que sea mucho más fácil de usar!

Seguro para los niños

Puede dejar a sus hijos solos en una habitación con este producto y no se harán daño (no lo garantizamos).

Esto se define en HTML como sigue:

```
<DL>
<DT><STRONG>Menor coste</STRONG>
```

```

<DD>¡La nueva versión de este producto cuesta mucho menos que la versión anterior!
<DT><STRONG>Más fácil de usar</STRONG>
<DD>¡Hemos cambiado el producto para que sea mucho más fácil de usar!
<DT><STRONG>Seguro para los niños</STRONG>
<DD>Puede dejar a sus hijos solos en una habitación con este producto
    y no se harán daño (no lo garantizamos).
</DL>

```

Las listas pueden además estar anidadas, pudiendo usarse al mismo tiempo tipos diferentes de listas, como en el ejemplo siguiente, que es una lista de definiciones que contiene una lista no ordenada (los ingredientes) y una lista ordenada (los pasos a seguir):

Ingredientes:

- 100 g de harina
- 10 g de azúcar
- 1 taza de agua
- 2 huevos
- sal, pimienta

Pasos a seguir:

1. Mezcle los ingredientes secos íntimamente.
2. Vierta los ingredientes líquidos.
3. Remueva durante 10 minutos.
4. Hornear durante una hora a 300 grados.

Notas:

Puede añadir uvas para mejorar la receta.

La presentación exacta de los tres tipos de listas depende del agente de usuario. Desaconsejamos a los autores el uso de listas únicamente con el fin de dar sangría al texto. Éste es un aspecto estilístico que se resuelve apropiadamente con hojas de estilo.

10.2 Listas no ordenadas (UL), listas ordenadas (OL) y objetos de lista (LI)

```

<!ELEMENT UL - - (LI)+                -- lista no ordenada -->
<!ATTLIST UL
  %attrs;                               -- %coreattrs, %il8n, %events --
>
<!ELEMENT OL - - (LI)+                -- lista ordenada -->
<!ATTLIST OL
  %attrs;                               -- %coreattrs, %il8n, %events --
>

```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria***

```

<!ELEMENT LI - O (%flow;)*            -- objeto de lista -->
<!ATTLIST LI
  %attrs;                               -- %coreattrs, %il8n, %events --
>

```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **opcional***

Definiciones de atributos

type = información de estilo [\[CI\]](#)

Desaprobado. Este atributo especifica el estilo de un objeto de lista. Los valores disponibles son para los agentes de usuario visuales. [Los valores posibles](#) se describen más abajo (junto con la información sobre las diferencias entre mayúsculas y minúsculas).

start = [number \[CN\]](#)

Desaprobado. Sólo para [OL](#). Este atributo especifica el número del primer objeto de una lista ordenada. El número inicial por defecto es el "1". Obsérvese que si bien el valor de este atributo es un entero, el rótulo correspondiente puede no ser numérico. Así, por ejemplo, cuando el estilo de objeto de lista es de letras latinas mayúsculas (A, B, C, ...), start=3 significa "C". Cuando el estilo es de numerales romanos en minúsculas, start=3 significa "iii", etc.

value = [number \[CN\]](#)

Desaprobado. Sólo para [LI](#). Este atributo establece el número del objeto de lista actual. Obsérvese que si bien el valor de este atributo es un entero, el rótulo correspondiente puede no ser numérico (véase el atributo [start](#)).

compact [\[CI\]](#)

Desaprobado. Si está establecido, este atributo booleano indica a los agentes de usuario visuales que representen la lista de un modo más compacto. La interpretación de este atributo depende del agente de usuario.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

Las listas ordenadas y las no ordenadas se representan de idéntica manera, excepto en que los agentes de usuario visuales numeran los objetos de las listas ordenadas. Los agentes de usuario pueden presentar estos números de diferentes formas. Las listas no ordenadas no se numeran.

Ambos tipos de lista se componen de secuencias de objetos de lista definidos por el elemento LI (cuya etiqueta final puede omitirse).

Este ejemplo ilustra la estructura básica de una lista.

```
<UL>
  <LI> ... primer objeto de lista...
  <LI> ... segundo objeto de lista...
  ...
</UL>
```

Las listas pueden además estar anidadas:

EJEMPLO DESAPROBADO:

```
<UL>
  <LI> ... Nivel uno, número uno...
  <OL>
    <LI> ... Nivel dos, número uno...
    <LI> ... Nivel dos, número dos...
    <OL start="10">
      <LI> ... Nivel tres, número uno...
    </OL>
    <LI> ... Nivel dos, número tres...
  </OL>
  <LI> ... Nivel uno, número dos...
</UL>
```

Detalles sobre el orden de los números. En las listas ordenadas, no es posible reanudar automáticamente la numeración de una lista donde acabó una lista anterior, así como tampoco ocultar la numeración para un objeto de la lista. Sin embargo, los autores pueden inicializar el número de un objeto de la lista estableciendo su atributo value. La numeración de los objetos de lista subsiguientes continúa a partir del nuevo valor. Por ejemplo:

```
<ol>
```



```
<li value="30"> hace que éste sea el objeto de lista número 30.
<li value="40"> hace que éste sea el objeto de lista número 40.
<li> hace que éste sea el objeto de lista número 41.
</ol>
```

10.3 Listas de definiciones: los elementos DL, DT y DD

```
<!-- listas de definiciones - DT para el término, DD para su definición -->
```

```
<!ELEMENT DL - - (DT|DD)+ -- lista de definiciones -->
<!ATTLIST DL
  %attrs; -- %coreattrs, %il8n, %events --
>
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria***

```
<!ELEMENT DT - O (%inline;)* -- término definido -->
<!ELEMENT DD - O (%flow;)* -- descripción de la definición -->
<!ATTLIST (DT|DD)
  %attrs; -- %coreattrs, %il8n, %events --
>
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **opcional***

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

Las listas de definiciones sólo difieren levemente de los otros tipos de listas, en que los objetos de lista consisten en dos partes: un término y una descripción. El término viene dado por el elemento [DT](#) y está restringido a contenido en línea. La descripción viene dada por un elemento [DD](#) que contiene contenido en bloque.

Aquí tenemos un ejemplo:

```
<DL>
  <DT>Dweeb
  <DD>persona joven y excitable que puede madurar y convertirse
    en un <EM>Nerd</EM> o en un <EM>Geek</EM>

  <DT>Hacker
  <DD>un programador inteligente

  <DT>Nerd
  <DD>técnicamente brillante pero socialmente inútil
</DL>
```

Aquí tenemos un ejemplo con términos y definiciones múltiples:

```
<DL>
  <DT>Center
  <DT>Centre
```

- <DD> A point equidistant from all points on the surface of a sphere.
- <DD> In some field sports, the player who holds the middle position on the field, court, or forward line.

</DL>

Otra aplicación de [DL](#) es, por ejemplo, dar formato a un diálogo, de modo que cada [DT](#) identifica al hablante, y cada [DD](#) contiene sus palabras.

10.3.1 Representación visual de las listas

Nota. Lo que sigue es una descripción informativa del comportamiento de algunos agentes de usuario visuales al dar formato a las listas. Las hojas de estilo permiten un mejor control sobre formato de las listas (p.ej., su numeración, convenciones dependientes del idioma, sangrías, etc.).

Los agentes de usuario visuales suelen sangrar la listas anidadas según su nivel de anidamiento.

Tanto para [OL](#) como para [UL](#), el atributo [type](#) especifica las opciones de representación para los agentes de usuario visuales.

Para el elemento [UL](#), los valores posibles del atributo [type](#) son disc (disco), square (cuadrado), y circle (círculo). El valor por defecto depende del nivel de anidamiento de la lista actual. Estos valores no distinguen entre mayúsculas y minúsculas.

El modo en que se presenta cada valor depende del agente de usuario. Los agentes de usuario deberían intentar presentar un "disc" como un pequeño círculo relleno, un "circle" como un pequeño círculo vacío, y un "square" como un pequeño cuadrado vacío.

Un agente de usuario gráfico podría hacer las siguientes representaciones:

- para el valor "disc"
- para el valor "circle"
- para el valor "square"

Para el elemento [OL](#), los valores posibles del atributo [type](#) se resumen en la siguiente tabla (la diferencia entre mayúsculas y minúsculas es importante):

Tipo	Estilo de numeración	
1	números arábigos	1, 2, 3, ...
a	alfabética en minúsculas	a, b, c, ...
A	alfabética en mayúsculas	A, B, C, ...
i	números romanos en minúsculas	i, ii, iii, ...
I	números romanos en mayúsculas	I, II, III, ...

Obsérvese que el atributo [type](#) está [desaprobado](#) y que los estilos de las listas deberían especificarse mediante hojas de estilo.

Por ejemplo, usando CSS se puede especificar que el estilo de los números de los objetos de lista de una lista numerada debería ser numerales romanos en minúsculas. En el extracto de a continuación, todos los elementos [OL](#) que pertenezcan a la clase "conromanos" tendrán numerales romanos delante de los objetos de la lista.

```
<STYLE type="text/css">
OL.conromanos { list-style-type: lower-roman }
</STYLE>
<BODY>
<OL class="conromanos">
<LI> Paso uno ...
<LI> Paso dos ...
</OL>
</BODY>
```

La representación de una lista de definiciones también depende del agente de usuario. El ejemplo:

<DL>

```
<DT>Dweeb  
<DD>persona joven y excitable que puede madurar y convertirse  
    en un <EM>Nerd</EM> o un <EM>Geek</EM>
```

```
<DT>Hacker  
<DD>un programador inteligente
```

```
<DT>Nerd  
<DD>técnicamente brillante pero socialmente inútil  
</DL>
```

podría representarse como sigue:

```
Dweeb      persona joven y excitable que puede madurar y convertirse en un Nerd o un Geek  
Hacker     un programador inteligente  
Nerd       técnicamente brillante pero socialmente inútil
```

10.4 Los elementos DIR y MENU

DIR y **MENU** están [desaprobados](#).

Véase el [DTD Transicional](#) para su definición formal.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

El elemento [DIR](#) fue diseñado para crear listas multicolumna de directorios. El elemento [MENU](#) fue diseñado para listas de menú de una sola columna. Ambos elementos tienen la misma estructura que [UL](#), y únicamente varía su representación. En la práctica, un agente de usuario representará una lista [DIR](#) o [MENU](#) exactamente igual que una lista [UL](#).

Recomendamos encarecidamente usar [UL](#) en lugar de estos elementos.

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

11 Tablas

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Introducción a las tablas](#)
2. [Elementos para construir tablas](#)
 1. [El elemento TABLE](#)
 - [Direccionalidad de tablas](#)
 2. [Títulos de tablas: el elemento CAPTION](#)
 3. [Grupos de filas: los elementos THEAD, TFOOT y TBODY](#)
 4. [Grupos de columnas: los elementos COLGROUP y COL](#)
 - [El elemento COLGROUP](#)
 - [El elemento COL](#)
 - [Cálculo del número de columnas de una tabla](#)
 - [Cálculo de la anchura de las columnas](#)
 5. [Filas de una tabla: el elemento TR](#)
 6. [Celdas de una tabla: los elementos TH y TD](#)
 - [Celdas que abarcan varias filas o columnas](#)
3. [Formato de tablas por agentes de usuario visuales](#)
 1. [Bordes y líneas de división](#)
 2. [Alineación horizontal y vertical](#)
 - [Herencia de las especificaciones de alineación](#)
 3. [Márgenes de las celdas](#)
4. [Representación de tablas por agentes de usuario no visuales](#)
 1. [Asociación de información de encabezado con celdas de datos](#)
 2. [Categorización de celdas](#)
 3. [Algoritmo para encontrar información de encabezado](#)
5. [Tabla de muestra](#)

11.1 Introducción a las tablas

El modelo de tablas de HTML permite a los autores organizar datos -- textos, texto preformateado, imágenes, vínculos, formularios, campos de formularios, otras tablas, etc. -- en filas y en columnas de celdas.

Cada tabla puede tener un título asociado (ver el elemento [CAPTION](#)) que proporciona una descripción corta del propósito de la tabla. Se puede proporcionar también una descripción más larga (por medio del atributo [summary](#)) para el provecho de las personas que utilicen agentes de usuario basados en Braille o en voz.

Las [filas de una tabla](#) pueden agruparse en secciones de encabezado, pie y cuerpo, (por medio de los elementos [THEAD](#), [TFOOT](#) y [TBODY](#), respectivamente). Los grupos de filas conllevan información estructural adicional y los agentes de usuario los pueden representar de modo que se enfatice esta estructura. Los agentes de usuario pueden sacar partido de la división entre encabezado, cuerpo y pie para permitir el desplazamiento de las secciones de cuerpo independientemente de las secciones de

cabecera y pie. Cuando se imprimen tablas largas, se puede repetir en cada página la información de cabecera y pie que contienen los datos de la tabla.

Los autores también pueden [agrupar columnas](#) para proporcionar información estructural adicional de la que pueden sacar partido los agentes de usuario. Además de esto, los autores pueden declarar propiedades de columnas al comienzo de la definición de la tabla (por medio de los elementos [COLGROUP](#) y [COL](#)) de tal modo que se permita a los agentes de usuario representar la tabla incrementalmente en lugar de tener que esperar a que lleguen todos los datos de la tabla antes de empezar a representarla.

Las [celdas de la tabla](#) puede contener o bien información de "encabezado" (ver el elemento [TH](#)) o de "datos" (ver el elemento [TD](#)). Las celdas pueden ocupar varias filas y columnas. El modelo de tablas de HTML 4 permite a los autores dar un rótulo a cada celda de modo que los [agentes de usuario no visuales](#) puedan comunicar más fácilmente información de encabezado de la celda al usuario. Estos mecanismos no sólo ayudan considerablemente a los usuarios con discapacidades visuales, sino que además permiten manejar tablas a navegadores multi-modo sin cables con posibilidades de representación limitadas (p.ej., teléfonos con soporte web).

No deberían usarse tablas con la única finalidad de organizar la presentación de los contenidos de un documento (es decir, de crear el "layout"), ya que esto puede ocasionar problemas cuando se represente en un medio no visual. Además, al incluir gráficos, estas tablas pueden forzar a los usuarios a hacer desplazar horizontalmente la pantalla para ver una tabla diseñada en un sistema con una pantalla más grande. Para minimizar estos problemas, los autores deberían usar [hojas de estilo](#) en lugar de tablas para organizar la presentación.

Nota. Esta especificación incluye información más detallada sobre tablas en las secciones sobre [fundamentos del diseño con tablas y cuestiones de implementación](#).

Aquí tenemos una tabla sencilla que ilustra algunas de las características del modelo de tablas de HTML. La siguiente definición de tabla:

```
<TABLE border="1"
  summary="Esta tabla da algunas estadísticas sobre la mosca
  de la fruta: altura y peso medio, y porcentaje
  con los ojos rojos (para machos y para hembras).">
<CAPTION><EM>Una tabla de prueba con celdas fusionadas</EM></CAPTION>
<TR><TH rowspan="2"><TH colspan="2">Media
  <TH rowspan="2">Ojos<BR>Rojos
<TR><TH>altura<TH>peso
<TR><TH>Machos<TD>1.9<TD>0.003<TD>40%
<TR><TH>Hembras<TD>1.7<TD>0.002<TD>43%
</TABLE>
```

podría representarse de una manera similar a ésta en un dispositivo tty:

```

Una tabla de muestra con celdas fusionadas
/-----\
|          |          Media          | Ojos |
|          |-----|          | rojos |
|          | altura | peso  |      |
|-----|-----|-----|-----|
| Machos  | 1.9   | 0.003 | 40%  |
|-----|-----|-----|-----|
| Hembras | 1.7   | 0.002 | 43%  |
\-----/

```

o a ésta por un agente de usuario gráfico:

Una tabla de muestra con celdas fusionadas

	Media		Ojos Rojos
	altura	peso	
Machos	1.9	0.003	40%
Hembras	1.7	0.002	43%

11.2 Elementos para construir tablas

11.2.1 El elemento TABLE

```
<!ELEMENT TABLE - -
      (CAPTION?, (COL*|COLGROUP*), THEAD?, TFOOT?, TBODY+)>
<!ATTLIST TABLE -- elemento tabla --
  %attrs; -- %coreattrs, %il8n, %events --
  summary %Text; #IMPLIED -- propósito/estructura para salida por voz --
  width %Length; #IMPLIED -- anchura de la tabla --
  border %Pixels; #IMPLIED -- controlar el borde que rodea a la tabla --
  frame %TFrame; #IMPLIED -- qué partes del borde representar --
  rules %TRules; #IMPLIED -- líneas de división entre filas y columnas --
--
  cellspacing %Length; #IMPLIED -- espaciado entre celdas --
  cellpadding %Length; #IMPLIED -- espacio dentro de las celdas --
>
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

Definiciones de atributos

summary = [texto](#) [CS]

Este atributo proporciona un resumen del propósito y la estructura de la tabla para agentes de usuario que representen en medios no visuales como voz o Braille.

align = left|center|right [CI]

Desaprobado. Este atributo especifica la posición de la tabla con respecto al documento. Valores permitidos:

- left: La tabla está a la izquierda del documento.
- center: La tabla está centrada con el documento.
- right: La tabla está a la derecha del documento.

width = [longitud](#) [CN]

Este atributo especifica la anchura deseada de la tabla entera y está destinado a agentes de usuario visuales. Cuando el valor es un valor porcentual, el valor es relativo al espacio disponible horizontal del agente de usuario. En ausencia de una especificación de anchura, la anchura de la tabla se determina por el agente de usuario.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)
- [bgcolor](#) (color de fondo)
- [frame](#), [rules](#), [border](#) (bordes y líneas de división)

- [cellspacing](#), [cellpadding](#) ([márgenes de celdas](#))

El elemento [TABLE](#) contiene a todos los demás elementos que especifican el título, las filas, el contenido, y el formato.

La siguiente lista informativa describe qué operaciones pueden llevar a cabo los agentes de usuario cuando representen una tabla:

- Poner a disposición del usuario el [resumen](#) (summary) de la tabla. Los autores deberían proporcionar un resumen del contenido y la estructura de la tabla de modo que las personas que utilicen agentes de usuario no visuales puedan entenderla mejor.
- Representar el título, si está definido.
- Representar el encabezado de la tabla, si está definido. Representar el pie de la tabla, si está definido. Los agentes de usuario deben saber dónde representar el encabezado y el pie. Por ejemplo, si el medio de salida está paginado, los agentes de usuario pueden colocar el encabezado en la parte superior de cada página y el pie en la inferior. Análogamente, si el agente de usuario proporciona un mecanismo para desplazar las filas, el encabezado puede aparecer en la parte superior del área de desplazamiento y el pie en la parte inferior.
- Calcular el [número de columnas](#) de la tabla. Obsérvese que el número de filas de una tabla es igual al número de elementos [TR](#) contenidos por el elemento [TABLE](#).
- Agrupar las columnas de acuerdo a las especificaciones de [agrupamiento de columnas](#) que haya.
- Representar las celdas, fila por fila y agrupadas en las columnas apropiadas, entre el encabezado y el pie. Los agentes de usuario visuales deberían [dar formato a la tabla](#) de acuerdo con los atributos HTML y las especificaciones de las hojas de estilo.

El modelo de tablas de HTML ha sido diseñado de modo que los agentes de usuario, con la ayuda de los autores, puedan representar las tablas *incrementalmente* (es decir, a medida que llegan las filas de la tabla), en lugar de tener que esperar a que lleguen todos los datos antes de empezar a representarlos.

Para que un agente de usuario pueda dar formato a una tabla en un solo paso, los autores deben decir al agente de usuario:

- El número de columnas de la tabla. Consulte la sección sobre el [cálculo del número de columnas de una tabla](#) para más detalles sobre cómo proporcionar esta información.
- Las anchuras de estas columnas. Consulte la sección sobre el [cálculo de la anchura de las columnas](#) para más detalles sobre cómo proporcionar esta información.

Más concretamente, un agente de usuario puede representar una tabla en un solo paso cuando las anchuras de las columnas se especifiquen usando una combinación de elementos [COLGROUP](#) y [COL](#). Si alguna de las columnas se especifica en términos relativos o porcentuales (véase la sección sobre el [cálculo de la anchura de las columnas](#)), los autores deben especificar también la anchura de la propia tabla.

Direccionalidad de tablas

La direccionalidad de una tabla es o bien la direccionalidad heredada (por defecto es de izquierda a derecha) o bien la especificada por el atributo [dir](#) del elemento [TABLE](#).

Para una tabla de izquierda a derecha, la columna cero es la del lado izquierdo y la fila cero es la superior. Para una columna de derecha a izquierda, la columna cero es la del lado derecho, y la fila cero es la superior.

Cuando un agente de usuario asigna celdas adicionales a una fila (véase la sección sobre el [cálculo del número de columnas de una tabla](#)), las celdas adicionales de la fila se añaden a la derecha de la tabla si la tabla es de izquierda a derecha, y a la izquierda si la tabla es de derecha a izquierda.

Obsérvese que [TABLE](#) es el único elemento en el que [dir](#) invierte el orden visual de las columnas; no puede invertirse independientemente una fila individual ([TR](#)) ni un grupo de columnas ([COLGROUP](#)).

El atributo [dir](#), cuando se establece para un elemento [TABLE](#), también afecta a la dirección del texto dentro de las celdas de la tabla (ya que el atributo [dir](#) es heredado por elementos en bloque).

Para especificar una tabla de derecha a izquierda, el atributo [dir](#) se establece como sigue:

```
<TABLE dir="RTL">  
...el resto de la tabla...  
</TABLE>
```

La dirección del texto en celdas individuales puede cambiarse estableciendo el atributo [dir](#) en un elemento que defina la celda. Consulte la sección sobre [texto bidireccional](#) para más información sobre cuestiones de dirección del texto.

11.2.2 Títulos de tablas: el elemento CAPTION

```
<!ELEMENT CAPTION - - (%inline;)* -- título de una tabla -->
<!ATTLIST CAPTION
  %attrs; -- %coreattrs, %i18n, %events --
>
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria***

Definiciones de atributos

align = top|bottom|left|right [CI]

Desaprobado. Para agentes de usuario visuales, este atributo especifica la posición del título con respecto a la tabla.

Valores posibles:

- top: El título está en la parte superior de la tabla. Este es el valor por defecto.
- bottom: El título está en la parte inferior de la tabla.
- left: El título está a la izquierda de la tabla.
- right: El título está a la derecha de la tabla.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

El texto del elemento [CAPTION](#), cuando esté presente, debería describir la naturaleza de la tabla. El elemento [CAPTION](#) sólo se permite inmediatamente después de la etiqueta inicial de [TABLE](#). Un elemento [TABLE](#) sólo puede contener un elemento [CAPTION](#).

Los agentes de usuario visuales permiten a las personas con visión percibir rápidamente la estructura de una tabla a partir de los encabezados así como a partir del título. Una consecuencia de esto es que la información sobre el propósito y la estructura de la tabla que dan los títulos normalmente no será la adecuada desde la perspectiva de las personas que utilicen agentes de usuario no visuales.

Los autores deberían por lo tanto esforzarse en proporcionar información adicional resumiendo el propósito y la estructura de la tabla usando el atributo [summary](#) del elemento [TABLE](#). Esto es especialmente importante para tablas sin título. Los ejemplos que figuran más adelante ilustran el uso del atributo [summary](#).

Los agentes de usuario visuales deberían evitar recortar cualquier parte de la tabla, incluyendo el título, a menos que se proporcionen medios para acceder a todo el contenido, p.ej., mediante un barra de desplazamiento horizontal o vertical. Recomendamos que el texto del título se ajuste a la misma anchura que la tabla. (Véase también la sección sobre [algoritmos recomendados de composición de la presentación](#).)

11.2.3 Grupos de filas: los elementos THEAD, TFOOT y TBODY

```
<!ELEMENT THEAD - O (TR)+ -- encabezado de tabla -->
<!ELEMENT TFOOT - O (TR)+ -- pie de tabla -->
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **opcional***


```
<!ELEMENT TBODY O O (TR)+ -- cuerpo de tabla -->
```

Etiqueta inicial: **opcional**, Etiqueta final: **opcional**

```
<!ATTLIST (THEAD|TBODY|TFOOT) -- sección de tabla --
  %attrs; -- %coreattrs, %il8n, %events --
  %cellhalign; -- alineación horizontal en las celdas --
  %cellvalign; -- alineación vertical en las celdas --
>
```

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)
- [align](#), [char](#), [charoff](#), [valign](#) (alineación en celdas)

Las filas de una tabla pueden agruparse en una cabecera de tabla, un pie de tabla, y una o más secciones de cuerpo de tabla, usando los elementos [THEAD](#), [TFOOT](#) y [TBODY](#), respectivamente. Esta división permite a los agentes de usuario ofrecer la posibilidad de desplazar la información de los cuerpos de la tabla independientemente de la cabecera y el pie. Cuando se imprimen tablas largas, la información de cabecera y pie de tabla puede repetirse en todas las páginas que contengan datos de la tabla.

La cabecera de tabla y el pie de tabla deberían contener información sobre las columnas de la tabla. El cuerpo de tabla debería contener filas de datos de tabla.

En caso de estar presentes, cada elemento [THEAD](#), [TFOOT](#) y [TBODY](#) contiene un *grupo de filas*. Cada grupo de filas debe contener al menos una fila, definida por el elemento [TR](#).

Este ejemplo ilustra el orden y la estructura de encabezados, pies, y cuerpos de tabla.

```
<TABLE>
<THEAD>
  <TR> ...información de cabecera...
</THEAD>
<TFOOT>
  <TR> ...información de pie...
</TFOOT>
<TBODY>
  <TR> ...primera fila de los datos del bloque uno...
  <TR> ...segunda fila de los datos del bloque uno...
</TBODY>
<TBODY>
  <TR> ...primera fila de los datos del bloque dos...
  <TR> ...segunda fila de los datos del bloque dos...
  <TR> ...tercera fila de los datos del bloque dos...
</TBODY>
</TABLE>
```

[TFOOT](#) debe aparecer antes de [TBODY](#) dentro de una definición de [TABLE](#) de modo que los agentes de usuario puedan representar el pie antes de recibir todas las (potencialmente numerosas) filas de datos. A continuación se resume qué etiquetas son obligatorias y cuáles pueden omitirse:

- La etiqueta inicial de [TBODY](#) es siempre obligatoria excepto cuando la tabla contenga sólo un cuerpo de tabla y ninguna sección de cabecera ni de pie. La etiqueta final de [TBODY](#) puede omitirse siempre sin problemas.
- Las etiquetas iniciales de [THEAD](#) y [TFOOT](#) son obligatorias cuando estén presentes las secciones de cabecera y de pie de tabla respectivamente, pero las etiquetas finales correspondientes pueden omitirse sin problemas.

Los analizadores de los agentes de usuario conformes deben obedecer estas reglas por razones de compatibilidad con versiones anteriores.

La tabla del ejemplo previo puede abreviarse quitando ciertas etiquetas iniciales:

```
<TABLE>
<THEAD>
  <TR> ...información de cabecera...
<TFOOT>
  <TR> ...información de pie...
<TBODY>
  <TR> ...primera fila de los datos del bloque uno...
  <TR> ...segunda fila de los datos del bloque uno...
<TBODY>
  <TR> ...primera fila de los datos del bloque dos...
  <TR> ...segunda fila de los datos del bloque dos...
  <TR> ...tercera fila de los datos del bloque dos...
</TABLE>
```

Las secciones [THEAD](#), [TFOOT](#) y [TBODY](#) deben contener el mismo número de columnas.

11.2.4 Grupos de columnas: los elementos [COLGROUP](#) y [COL](#)

Los grupos de columnas permiten a los autores crear divisiones estructurales dentro de una tabla. Los autores pueden resaltar esta estructura por medio de hojas de estilo o con atributos HTML (p.ej., el atributo [rules](#) del elemento [TABLE](#)). Para un ejemplo de presentación visual de grupos de columnas, consulte la [tabla de muestra](#).

Una tabla puede contener o bien un único grupo implícito de columnas (si no hay ningún elemento [COLGROUP](#) delimitando las columnas), o cualquier número de grupos explícitos de columnas (cada uno de ellos delimitado por un elemento [COLGROUP](#)).

El elemento [COL](#) permite a los autores compartir atributos entre varias columnas sin que ello implique ningún tipo de agrupamiento estructural. El "campo de acción" del elemento [COL](#) es el número de columnas que compartirán los atributos del elemento.

El elemento [COLGROUP](#)

```
<!ELEMENT COLGROUP - O (COL)* -- grupo de columnas de tabla -->
<!ATTLIST COLGROUP
  %attrs; -- %coreattrs, %il8n, %events --
  span NUMBER 1 -- número por defecto de columnas en el
grupo--
  width %MultiLength; #IMPLIED -- anchura por defecto para
las COLs contenidas --
  %cellhalign; -- alineación horizontal en las celdas --
  %cellvalign; -- alineación vertical en las celdas --
  >
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **opcional***

Definiciones de atributos

[span](#) = [number](#) [[CN](#)]

Este atributo, que debe ser un entero > 0, especifica el número de columnas de un grupo de columnas. Los diferentes valores significan lo siguiente:

- En ausencia de un atributo [span](#), cada [COLGROUP](#) define un grupo de columnas que contiene una columna.
- Si el atributo [span](#) se establece en $N > 0$, el elemento [COLGROUP](#) actual define un grupo de columnas que contiene N columnas.

Los agentes de usuario deben ignorar este atributo si el elemento [COLGROUP](#) contiene uno o más elementos [COL](#).

width = [multilongitud \[CN\]](#)

Este atributo especifica una anchura por defecto para cada columna del grupo de columnas actual. Además de los valores normales de píxeles, porcentajes y longitudes relativas, este atributo permite la forma especial "0*" (cero asterisco) que significa que la anchura de cada columna del grupo debería ser la anchura mínima necesaria para alojar los contenidos de la columna. Esto implica que antes de poder calcular correctamente la anchura de la columna deben conocerse primero todos los contenidos de la columna. Los autores deberían tener en cuenta que al especificar "0*" impedirán a los agentes de usuario visuales representar la tabla incrementalmente.

La anchura de cualquier columna del grupo de columnas cuya anchura ([width](#)) se especifique mediante un elemento [COL](#) prevalece sobre el valor de este atributo.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)
- [align](#), [char](#), [charoff](#), [valign](#) (alineación en celdas)

El elemento [COLGROUP](#) crea un grupo explícito de columnas. El número de columnas del grupo de columnas puede especificarse de dos maneras mutuamente exclusivas:

1. El atributo [span](#) del elemento (valor por defecto 1) especifica el número de columnas del grupo.
2. Cada elemento [COL](#) del [COLGROUP](#) representa una o más columnas en el grupo.

La ventaja de usar el atributo [span](#) es que los autores pueden agrupar información sobre las anchuras de las columnas. Así, si una tabla contiene cuarenta columnas, todas ellas de 20 píxeles de ancho, es más fácil escribir:

```
<COLGROUP span="40" width="20">
</COLGROUP>
```

que:

```
<COLGROUP>
  <COL width="20">
  <COL width="20">
  ...un total de cuarenta elementos COL...
</COLGROUP>
```

Cuando sea necesario distinguir una columna del resto del grupo (p.ej., para información de estilo, para especificar información de anchura, etc.), los autores deben identificar la anchura de esa columna con un elemento [COL](#). Así, para aplicar información de estilo especial a la última columna del ejemplo anterior, la singularizamos del siguiente modo:

```
<COLGROUP width="20">
  <COL span="39">
  <COL id="dame-un-formato-especial">
</COLGROUP>
```

El atributo [width](#) del elemento [COLGROUP](#) es heredado por las cuarenta columnas. El primer elemento [COL](#) se refiere a las primeras 39 columnas (y no hace nada especial con ellas) y el segundo asigna un valor de [id](#) a la columna número 40 de modo que una hoja de estilo pueda hacer referencia a ella.

La tabla del siguiente ejemplo contiene dos grupos de columnas. El primer grupo de columnas contiene 10 columnas y el segundo contiene 5 columnas. La anchura por defecto de las columnas del primer grupo de columnas es de 50 píxeles. La anchura de las columnas del segundo grupo de columnas será la mínima necesaria para cada columna.

```
<TABLE>
<COLGROUP span="10" width="50">
```

```
<COLGROUP span="5" width="0*">
<THEAD>
<TR><TD> ...
</TABLE>
```

El elemento COL

```
<!ELEMENT COL          - O EMPTY          -- columna de tabla -->
<!ATTLIST COL          -- grupos y propiedades de columnas --
  %attrs;              -- %coreattrs, %il8n, %events --
  span                 NUMBER             1   -- los atributos de COL afectan a N columnas
  --
  width                %MultiLength;     #IMPLIED -- especificación de la anchura
                                         de las columnas --
  %cellhalign;         -- alineación horizontal en las celdas --
  %cellvalign;         -- alineación vertical en las celdas --
  >
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **prohibida**

Definiciones de atributos

span = *number* [CN]

Este atributo, cuyo valor debe ser un entero > 0, especifica el número de columnas que "abarca" el elemento [COL](#); el elemento [COL](#) comparte sus atributos con todas las columnas que abarca. El valor por defecto de este atributo es 1 (es decir, el elemento [COL](#) se refiere a una sola columna). Si el atributo [span](#) se establece en $N > 1$, el elemento [COL](#) actual comparte sus atributos con las siguientes $N-1$ columnas.

width = *multilongitud* [CN]

Este atributo especifica una anchura por defecto para todas las columnas abarcadas por el elemento [COL](#) actual. Tiene el mismo significado que el atributo [width](#) del elemento [COLGROUP](#) y prevalece sobre él.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)
- [align](#), [char](#), [charoff](#), [valign](#) (alineación en celdas)

El elemento [COL](#) permite a los autores agrupar especificaciones de atributos para columnas de tablas. [COL](#) **no** agrupa columnas estructuralmente; ése es el papel del elemento [COLGROUP](#). Los elementos [COL](#) están vacíos y sólo sirven como soporte de atributos. Pueden aparecer dentro o fuera de un grupo explícito de columnas (es decir, de un elemento [COLGROUP](#)).

El atributo [width](#) de [COL](#) se refiere a la anchura de todas y cada una de las columnas que abarca el elemento.

Cálculo del número de columnas de una tabla

Hay dos maneras de determinar el número de columnas de una tabla (en orden de precedencia):

1. Si el elemento [TABLE](#) contiene algún elemento [COLGROUP](#) o [COL](#), los agentes de usuario deberían calcular el número de columnas sumando los valores siguientes:
 - Para cada elemento [COL](#), tomar el valor de su atributo [span](#) (valor por defecto 1).
 - Para cada elemento [COLGROUP](#) que contenga al menos un elemento [COL](#), ignorar el atributo [span](#) del elemento [COLGROUP](#). Para cada elemento [COL](#), realizar el cálculo del paso 1.
 - Para cada elemento vacío [COLGROUP](#), tomar el valor de su atributo [span](#) (valor por defecto 1).

2. De otra manera, si el elemento [TABLE](#) no contiene elementos [COLGROUP](#) ni [COL](#), los agentes de usuario deberían basar el número de columnas en lo que requieran las filas. El número de columnas es igual al número de columnas requerido por la fila con más columnas, incluyendo las celdas que abarcan varias columnas. Para cualquier fila que tenga menos columnas que este número de columnas, el final de la fila debería rellenarse con celdas vacías. El "final" de una fila depende de la [direccionalidad de la tabla](#).

Es un error que una tabla contenga elementos [COLGROUP](#) o [COL](#) y que los dos cálculos no resulten en el mismo número de columnas.

Una vez que el agente de usuario ha calculado el número de columnas de la tabla, puede agruparlas en [grupos de columnas](#).

Por ejemplo, para cada una de las tablas siguientes, los dos métodos de cálculo de columnas dan tres columnas. Las primeras tres tablas pueden representarse incrementalmente.

```
<TABLE>
<COLGROUP span=" 3 "></COLGROUP>
<TR><TD> ...
...filas...
</TABLE>
```

```
<TABLE>
<COLGROUP>
<COL>
<COL span=" 2 ">
</COLGROUP>
<TR><TD> ...
...filas...
</TABLE>
```

```
<TABLE>
<COLGROUP>
<COL>
</COLGROUP>
<COLGROUP span=" 2 ">
<TR><TD> ...
...filas...
</TABLE>
```

```
<TABLE>
<TR>
  <TD><TD><TD>
</TR>
</TABLE>
```

Cálculo de la anchura de las columnas

Los autores pueden especificar anchuras de columnas de tres maneras:

Fijas (Fixed)

Una especificación de anchura fija está dada en píxeles (p.ej., `width="30"`). Una especificación de anchura fija permite la representación incremental.

Porcentuales (Percentage)

Una especificación porcentual (p.ej., `width="20%"`) se basa en el porcentaje del espacio horizontal disponible para la tabla (entre los bordes actuales izquierdo y derecho, incluyendo elementos flotantes). Obsérvese que este espacio no depende de la propia tabla, y por lo tanto las especificaciones porcentuales permiten la representación incremental.

Proporcionales (Proportional)

Las especificaciones proporcionales (p.ej., `width="3*"`) se refieren a porciones de espacio horizontal *requerido* por una tabla. Si se da a la anchura de la tabla un valor fijo por medio del atributo `width` del elemento [TABLE](#), los agentes de usuario pueden representar la tabla incrementalmente incluso con columnas proporcionales.

Sin embargo, si la tabla no tiene una anchura fija, los agentes de usuario deben recibir todos los datos de la tabla antes de poder determinar el espacio horizontal requerido por la tabla. Sólo entonces puede asignarse este espacio a las columnas proporcionales.

Si un autor no especifica información de anchura para una columna, un agente de usuario no puede dar formato a la tabla incrementalmente, ya que debe esperar a que lleguen todos los datos de la columna para poder asignarle una anchura adecuada.

Si las anchuras de las columnas resultan ser demasiado estrechas para los contenidos de alguna de las celdas de la tabla, los agentes de usuario pueden optar por remodelar la tabla.

La tabla de este ejemplo contiene seis columnas. La primera no pertenece a un grupo explícito de columnas. Las tres siguientes pertenecen al primer grupo explícito de columnas y las dos últimas pertenecen al segundo grupo explícito de columnas. Esta tabla no puede ser formateada incrementalmente ya que contiene especificaciones de anchuras de columna proporcionales y no hay ningún valor para el atributo [width](#) del elemento [TABLE](#).

Una vez que el agente de usuario (visual) ha recibido los datos de la tabla, el agente de usuario asignará el espacio horizontal disponible como sigue: Primero el agente de usuario asignará 30 píxeles a las columnas uno y dos. Después, reservará el espacio mínimo requerido por la tercera columna. El resto del espacio horizontal lo dividirá en seis porciones iguales (ya que $2^* + 1^* + 3^* = 6$ porciones). La columna cuatro (2^*) recibirá dos de estas porciones, la columna cinco (1^*) recibirá una, y la columna seis (3^*) recibirá tres.

```
<TABLE>
<COLGROUP>
  <COL width="30">
<COLGROUP>
  <COL width="30">
  <COL width="0*">
  <COL width="2*">
<COLGROUP align="center">
  <COL width="1*">
  <COL width="3*" align="char" char=":">
<THEAD>
<TR><TD> ...
...filas...
</TABLE>
```

Hemos establecido el valor del atributo [align](#) del tercer grupo de columnas en "center". Todas las celdas de todas las columnas de este grupo heredarán este valor, pero pueden anularlo. De hecho, el [COL](#) hace justo eso, especificando que todas las celdas de la columna que gobierna serán alineadas con el carácter ":".

En la siguiente tabla, las especificaciones de anchuras de columna permiten al agente de usuario dar formato a la tabla incrementalmente:

```
<TABLE width="200">
<COLGROUP span="10" width="15">
<COLGROUP width="*">
  <COL id="penultima-columna">
  <COL id="ultima-columna">
<THEAD>
<TR><TD> ...
...filas...
</TABLE>
```

Las diez primeras columnas serán cada una de 15 píxeles de ancho. Las dos últimas columnas recibirán cada una la mitad de los 50 píxeles restantes. Obsérvese que los elementos [COL](#) sólo sirven para que pueda especificarse un valor [id](#) a las dos últimas columnas.

Nota. Aunque el atributo [width](#) del elemento [TABLE](#) no está desaprobado, se aconseja a los autores usar hojas de estilo para especificar anchuras de tabla.

11.2.5 Filas de una tabla: el elemento TR

```
<!ELEMENT TR          - O (TH|TD)+          -- fila de tabla -->
<!ATTLIST TR          -- fila de tabla --
  %attrs;             -- %coreattrs, %il8n, %events --
  %cellhalign;        -- alineación horizontal en las celdas --
  %cellvalign;        -- alineación vertical en las celdas --
  >
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **opcional**

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)
- [bgcolor](#) (color de fondo)
- [align](#), [char](#), [charoff](#), [valign](#) (alineación en celdas)

Los elementos [TR](#) actúan como contenedores de una fila de celdas de una tabla. La etiqueta final puede omitirse.

Esta tabla de ejemplo contiene tres filas, cada una iniciada con el elemento [TR](#):

```
<TABLE summary="Esta tabla muestra el número de tazas de
    café consumidas por cada senador, el tipo
    de café (descafeinado o normal), y si lo
    toma con azúcar.">
<CAPTION>Tazas de café consumidas por cada senador</CAPTION>
<TR> ...Una fila de encabezado...
<TR> ...Primera fila de datos...
<TR> ...Segunda fila de datos...
...el resto de la tabla...
</TABLE>
```

11.2.6 Celdas de una tabla: los elementos TH y TD

```
<!ELEMENT (TH|TD)    - O (%flow;)*          -- celda de encabezado de tabla,
                                                celda de datos de tabla -->

<!-- Para las tablas normales scope es más simple que el atributo headers -->
<!ENTITY % Scope "(row|col|rowgroup|colgroup)">

<!-- TH es para encabezados, TD es para datos,
     y para celdas que actúen como ambos use TD -->
<!ATTLIST (TH|TD)    -- celda de encabezado o de datos --
  %attrs;             -- %coreattrs, %il8n, %events --
  abbr                %Text;                 #IMPLIED -- abreviatura para la celda de encabezado --
  axis                CDATA                 #IMPLIED -- lista de encabezados relacionados
                                                separados por comas --
  headers            IDREFS                 #IMPLIED -- lista de id's de celdas de encabezado --
  scope              %Scope;               #IMPLIED -- campo de acción cubierto por las celdas
                                                de encabezado --
  rowspan            NUMBER                 1      -- número de filas abarcado por la celda --
```

```

colspan      NUMBER      1      -- número de columnas abarcado por la celda --
%cellhalign;      -- alineación horizontal en las celdas --
%cellvalign;      -- alineación vertical en las celdas --
>

```

Etiqueta inicial: **obligatoria**, Etiqueta final: **opcional**

Definiciones de atributos

headers = [idrefs](#) [CS]

Este atributo especifica la lista de celdas de encabezado que proporcionan información de encabezado para la celda de datos actual. El valor de este atributo es una lista de nombres de celdas separados por comas; a estas celdas se les debe dar nombre estableciendo su atributo [id](#). Generalmente los autores usan el atributo [headers](#) para ayudar a los agentes de usuario no visuales a representar información de encabezado sobre celdas de datos (p.ej., leyendo la información de encabezado antes de los datos de la celda), pero el atributo también puede emplearse junto con hojas de estilo. Véase también el atributo [scope](#).

scope = *scope-name* [CI]

Este atributo especifica el conjunto de celdas de datos para las cuales la celda de encabezado actual proporciona información de encabezado. Este atributo puede usarse en lugar del atributo [headers](#), en particular en tablas sencillas. Si se especifica, este atributo debe tener uno de los siguientes valores:

- **row** (*fila*): La celda actual proporciona información de encabezado para el resto de la fila que la contiene (véase también la sección sobre [direccionalidad de tablas](#)).
- **col** (*columna*): La celda actual proporciona información de encabezado para el resto de la columna que la contiene.
- **rowgroup** (*grupo de filas*): La celda de encabezado proporciona información de encabezado para el resto del [grupo de filas](#) que la contiene.
- **colgroup** (*grupo de columnas*): La celda de encabezado proporciona información de encabezado para el resto del [grupo de columnas](#) que la contiene.

abbr = *texto* [CS]

Este atributo debería usarse para proporcionar una forma abreviada del contenido de la celda; los agentes de usuario pueden representar esta forma abreviada en lugar del contenido de la celda cuando sea apropiado. Los nombres abreviados deberían ser cortos, ya que los agentes de usuario pueden representarlos repetidas veces. Por ejemplo, los sintetizadores de voz pueden representar los encabezados abreviados relacionados con una celda en particular antes de representar el contenido de esa celda.

axis = *cdata* [CI]

Este atributo puede utilizarse para situar una celda en categorías conceptuales, las cuales pueden considerarse como ejes de un espacio n-dimensional. Los agentes de usuario pueden dar a los usuarios acceso a estas categorías (p.ej., el usuario puede pedir al agente de usuario todas las celdas que pertenecen a ciertas categorías, el agente usuario puede presentar una tabla con la forma de una tabla de contenidos, etc.). Consulte la sección sobre [categorización de celdas](#) para más información. El valor de este atributo es una lista de nombres de categorías separados por comas.

rowspan = *number* [CN]

Este atributo especifica el número de filas abarcado por la celda actual. El valor por defecto de este atributo es uno ("1"). El valor cero ("0") significa que la celda abarca todas las filas desde la fila actual hasta la última fila de la sección de tabla ([THEAD](#), [TBODY](#), o [TFOOT](#)) en la que la celda está definida.

colspan = *number* [CN]

Este atributo especifica el número de columnas abarcado por la celda actual. El valor por defecto de este atributo es uno ("1"). El valor cero ("0") significa que la celda abarca todas las columnas desde la columna actual hasta la última columna del grupo de columnas ([COLGROUP](#)) en que la celda está definida.

nowrap [CI]

Desaprobado. Si está presente, este atributo booleano le dice a los agentes de usuario visuales que deshabiliten el ajuste automático de líneas (*text wrapping*) para esta celda. Para lograr efectos de ajuste de texto deberían usarse [hojas de estilo](#) en lugar de este atributo. **Nota.** Si se utiliza sin cuidado, este atributo puede resultar en celdas excesivamente anchas.

width = *longitud* [CN]

Desaprobado. Este atributo proporciona a los agentes de usuario una anchura de celda recomendada.

height = *longitud* [CN]

Desaprobado. Este atributo proporciona a los agentes de usuario una altura de celda recomendada.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)
- [bgcolor](#) (color de fondo)
- [align](#), [char](#), [charoff](#), [valign](#) (alineación en celdas)

Las celdas de una tabla pueden contener dos tipos de información: información de encabezado y datos. Esta distinción permite a los agentes de usuario representar celdas de encabezado y de datos de manera diferente, incluso en ausencia de hojas de estilo. Por ejemplo, los agentes de usuario visuales pueden presentar el texto de las celdas de encabezado con una fuente en negrita. Los sintetizadores de voz pueden representar la información de encabezado con una inflexión de voz diferente.

El elemento **TH** define una celda que contiene información de encabezado. Los agentes de usuario tienen a su disposición dos piezas de información de encabezado: los contenidos del elemento **TH** y el valor del atributo [abbr](#). Los agentes de usuario deben representar o bien los contenidos de la celda o bien el valor del atributo [abbr](#). En medios visuales, el último puede ser apropiado cuando no hay espacio suficiente para representar los contenidos completos de la celda. Para medios no visuales puede usarse [abbr](#) como abreviatura de los encabezados de la tabla cuando éstos se representen junto con los contenidos de las celdas a las que se aplican.

Los atributos [headers](#) y [scope](#) también permiten a los autores ayudar a los agentes de usuario no visuales a procesar información de encabezado. Consulte la sección sobre [rótulos de celdas para agentes de usuario no visuales](#) para más información y ejemplos.

El elemento **TD** define una celda que contiene datos.

Las celdas pueden estar vacías (es decir, pueden no contener datos).

Por ejemplo, la siguiente tabla contiene cuatro columnas de datos, cada una encabezada por una descripción de la columna.

```
<TABLE summary="Esta tabla muestra el número de tazas
de café consumidas por cada senador, el tipo
de café (descafeinado o normal), y si
lo toma con azúcar">
<CAPTION>Tazas de café consumidas por cada senador</CAPTION>
<TR>
  <TH>Nombre</TH>
  <TH>Tazas</TH>
  <TH>Tipo de Café</TH>
  <TH>¿Azúcar?</TH>
<TR>
  <TD>T. Sexton</TD>
  <TD>10</TD>
  <TD>Espresso</TD>
  <TD>No</TD>
<TR>
  <TD>J. Dinnen</TD>
  <TD>5</TD>
  <TD>Descafeinado</TD>
  <TD>Sí</TD>
</TABLE>
```

Un agente de usuario que represente la salida en un dispositivo tty podría mostrar esto como sigue:

Nombre	Tazas	Tipo de Café	¿Azúcar?
T. Sexton	10	Espresso	No
J. Dinnen	5	Descafeinado	Sí

Celdas que abarcan varias filas o columnas

Las celdas pueden abarcar varias filas o columnas. El número de filas o de columnas abarcado por una celda se establece con los atributos [rowspan](#) y [colspan](#) de los elementos [TH](#) y [TD](#).

En esta definición de tabla, especificamos que la celda de la fila cuatro, columna 2, debería abarcar un total de tres columnas, incluyendo la columna actual.

```
<TABLE border="1">
<CAPTION>Tazas de café consumidas por cada senador</CAPTION>
<TR><TH>Nombre<TH>Tazas<TH>Tipo de Café<TH>¿Azúcar?
<TR><TD>T. Sexton<TD>10<TD>Espresso<TD>No
<TR><TD>J. Dinnen<TD>5<TD>Descafeinado<TD>Sí
<TR><TD>A. Soria<TD colspan="3"><em>No disponible</em>
</TABLE>
```

Un agente visual podría representar esta tabla en un dispositivo tty como sigue:

```
Tazas de café consumidas por cada senador
-----
| Nombre |Tazas|Tipo de Café |¿Azúcar?|
-----
| T. Sexton|10   |Espresso       |No      |
-----
| J. Dinnen|5    |Descafeinado   |Yes     |
-----
| A. Soria |No disponible|                |        |
-----
```

El siguiente ejemplo ilustra (con la ayuda de los bordes de la tabla) cómo afectan las definiciones de celdas que abarcan más de una fila o de una columna a la definición de celdas posteriores. Consideremos la siguiente definición de tabla:

```
<TABLE border="1">
<TR><TD>1 <TD rowspan="2">2 <TD>3
<TR><TD>4 <TD>6
<TR><TD>7 <TD>8 <TD>9
</TABLE>
```

Al abarcar la celda "2" la primera y la segunda filas, la definición de la segunda fila la tendrá en cuenta. Así, el segundo [TD](#) de la fila dos en realidad define la tercera celda de la fila. Visualmente, la tabla podría representarse en un dispositivo tty como sigue:

```
-----
| 1 | 2 | 3 |
----|---|----
| 4 |   | 6 |
----|---|----
| 7 | 8 | 9 |
-----
```

mientras que un agente de usuario gráfico podría representarlo de esta forma:

1	2	3
4	6	
7	8	9

Obsérvese que si el [TD](#) que define la celda "6" hubiera sido omitido, el agente de usuario habría añadido una celda adicional vacía para completar la fila.

Análogamente, en la siguiente definición de tabla:

```
<TABLE border="1" >
<TR><TD>1 <TD>2 <TD>3
<TR><TD colspan="2">4 <TD>6
<TR><TD>7 <TD>8 <TD>9
</TABLE>
```

la celda "4" abarca dos columnas, de modo que el segundo [TD](#) de la fila en realidad define la tercera celda ("6"):

```
-----
| 1 | 2 | 3 |
-----
| 4 |   | 6 |
-----
| 7 | 8 | 9 |
-----
```

Un agente de usuario gráfico podría representarlo de esta forma:

1	2	3
4	6	
7	8	9

La definición de celdas superpuestas es un error. Los agentes de usuario pueden tratar este error de formas diferentes (p.ej., la representación puede variar).

El siguiente ejemplo ilegal ilustra cómo podrían crearse celdas superpuestas. En esta tabla, la celda "5" abarca dos filas y la celda "7" abarca dos columnas, de modo que hay una superposición en la celda que está entre "7" y "9":

```
<TABLE border="1" >
<TR><TD>1 <TD>2 <TD>3
<TR><TD>4 <TD rowspan="2">5 <TD>6
<TR><TD colspan="2">7 <TD>9
</TABLE>
```

11.3 Formato de tablas por agentes de usuario visuales

Nota. Las siguientes secciones describen los atributos HTML relacionados con el formato visual de tablas. Cuando esta especificación fue publicada por primera vez en 1997, [\[CSS1\]](#) no ofrecía mecanismos para controlar todos los aspectos del formato visual de tablas. Desde entonces, [\[CSS2\]](#) ha añadido propiedades para permitir el formato visual de tablas.

HTML 4 incluye mecanismos para controlar:

- [los estilos de los bordes](#)
- [el alineación horizontal y vertical](#) de los contenidos de las celdas
- y [los márgenes de las celdas](#)

11.3.1 Bordos y líneas de división

Los atributos siguientes afectan al marco externo (frame) y a las líneas de división interiores (rules) de una tabla.

Definiciones de atributos

frame = void|above|below|hsides|lhs|rhs|vsides|box|border [\[CI\]](#)

Este atributo especifica qué lados del marco que rodea a una tabla serán visibles. Valores posibles:

- void: Ningún lado. Este es el valor por defecto.
- above: Sólo el borde superior.
- below: Sólo el borde inferior.
- hsides: Sólo los bordes superior e inferior.
- vsides: Sólo los lados derecho e izquierdo.
- lhs: Sólo el lado izquierdo.
- rhs: Sólo el lado derecho.
- box: Los cuatro lados.
- border: Los cuatro lados.

rules = none|groups|rows|cols|all [\[CI\]](#)

Este atributo especifica qué líneas de división aparecerán entre las celdas de una tabla. La representación de las líneas de división depende del agente de usuario. Valores posibles:

- none: Ninguna línea de división. Este es el valor por defecto.
- groups: Sólo aparecerán líneas de división entre grupos de filas (ver [THEAD](#), [TFOOT](#) y [TBODY](#)) y grupos de columnas (ver [COLGROUP](#) y [COL](#)).
- rows: Sólo aparecerán líneas de división entre filas.
- cols: Sólo aparecerán líneas de división entre columnas.
- all: Aparecerán líneas de división entre todas las filas y columnas.

border = *píxeles* [\[CN\]](#)

Este atributo especifica la anchura (sólo en píxeles) del marco que rodea a una tabla (véase la Nota que se encuentra más abajo para más información sobre este atributo).

Para ayudar a distinguir las celdas de una tabla, podemos establecer el atributo [border](#) del elemento [TABLE](#). Consideremos uno de los ejemplos previos:

```
<TABLE border="1"
  summary="Esta tabla muestra el número de tazas
    de café consumidas por cada senador, el tipo
    de café (descafeinado o normal) y si lo
    toma con azúcar.">
<CAPTION>Tazas de café consumidas por cada senador</CAPTION>
<TR>
  <TH>Nombre</TH>
  <TH>Tazas</TH>
  <TH>Tipo de Café</TH>
  <TH>¿Azúcar?</TH>
<TR>
  <TD>T. Sexton</TD>
  <TD>10</TD>
  <TD>Espresso</TD>
  <TD>No</TD>
<TR>
  <TD>J. Dinnen</TD>
  <TD>5</TD>
  <TD>Descafeinado</TD>
  <TD>Sí</TD>
</TABLE>
```

En el siguiente ejemplo, el agente de usuario debería mostrar bordes de cinco píxeles de grosor en los lados izquierdo y derecho de la tabla, con líneas de división entre cada dos columnas.

```
<TABLE border="5" frame="vsides" rules="cols">
<TR> <TD>1 <TD>2 <TD>3
<TR> <TD>4 <TD>5 <TD>6
<TR> <TD>7 <TD>8 <TD>9
</TABLE>
```

Los agentes de usuario deberían obedecer las siguientes especificaciones para conservar la compatibilidad con versiones anteriores.

- Especificar `border="0"` implica `frame="void"` y, a menos que se especifique lo contrario, `rules="none"`.
- Otros valores de `border` implican `frame="border"` y, a menos que se especifique lo contrario, `rules="all"`.
- El valor "border" en la etiqueta inicial del elemento `TABLE` debería ser interpretado como el valor del atributo `frame`. Implica `rules="all"` y algún valor por defecto (distinto de cero) para el atributo `border`.

Por ejemplo, las siguientes definiciones son equivalentes:

```
<TABLE border="2">
<TABLE border="2" frame="border" rules="all">
```

así como las siguientes:

```
<TABLE border>
<TABLE frame="border" rules="all">
```

Nota. El atributo `border` también define el comportamiento de los bordes de los elementos `OBJECT` e `IMG`, pero toma valores diferentes para esos elementos.

11.3.2 Alineación horizontal y vertical

Los siguientes atributos pueden especificarse para diferentes elementos de una tabla (véanse sus definiciones).

```
<!-- atributos de alineación horizontal de los contenidos de las celdas -->
<!ENTITY % cellhalign
"align      (left|center|right|justify|char) #IMPLIED
 char       %Character;      #IMPLIED  -- carácter de alineación, p.ej. char=':' --
 charoff    %Length;        #IMPLIED  -- offset del carácter de alineación --"
>
<!-- atributos de alineación vertical para los contenidos de las celdas -->
<!ENTITY % cellvalign
"valign     (top|middle|bottom|baseline) #IMPLIED"
>
```

Definiciones de atributos

`align` = left|center|right|justify|char [\[CI\]](#)

Este atributo especifica la alineación de los datos y la justificación del texto de una celda. Valores posibles:

- left: Datos a la izquierda/Texto justificado a la izquierda.
- center: Datos centrados/Texto con justificación centrada. Este es el valor por defecto para los encabezados de las tablas.
- right: Datos a la derecha/Texto justificado a la derecha.
- justify: Texto doblemente justificado.
- char: Alinear el texto alrededor de un carácter específico. Si un agente de usuario no soporta alineación alrededor de un carácter, el comportamiento en presencia de este valor queda sin especificar.

`valign` = top|middle|bottom|baseline [\[CI\]](#)

Este atributo especifica la posición vertical de los datos dentro de una celda. Valores posibles:

- top: Los datos de la celda se alinean con la parte superior de la celda.

- middle: Los datos de la celda se centran verticalmente dentro de la celda. Este es el valor por defecto.
- bottom: Los datos de la celda se alinean con la parte inferior de la celda.
- baseline: Todas las celdas que estén en la misma fila que una celda cuyo atributo valign tenga este valor deberían tener sus datos textuales posicionados de tal modo que la primera línea de texto aparezca en una línea de base común para todas las celdas de la fila. Esta restricción no se aplica a las líneas subsiguientes de texto de estas celdas.

char = [carácter \[CN\]](#)

Este atributo especifica que un carácter individual dentro de un fragmento de texto actúe como eje de alineación. El valor por defecto para este atributo es el carácter de punto decimal para el idioma actual, definido por el atributo [lang](#) (p.ej., el punto (".") en inglés y la coma (",") en francés). Los agentes de usuario no necesitan soportar este atributo.

charoff = [longitud \[CN\]](#)

Si está presente, este atributo especifica la distancia (offset) entre el borde y la primera aparición del carácter de alineación en cada línea. Si una línea no incluye el carácter de alineación, debería ser desplazada horizontalmente hasta la posición de alineación.

Cuando se usa [charoff](#) para establecer el offset de un carácter de alineación, la dirección del desplazamiento está determinada por la dirección actual del texto (establecida con el atributo [dir](#)). En texto de izquierda a derecha (el valor por defecto), el desplazamiento es desde el margen izquierdo. En textos de derecha a izquierda, el desplazamiento es desde el margen derecho. Los agentes de usuario no necesitan soportar este atributo.

La tabla de este ejemplo alinea una fila de valores monetarios alrededor de un punto decimal. Hemos establecido explícitamente el carácter de alineación "."

```
<TABLE border="1">
<COLGROUP>
<COL><COL align="char" char=".">
<THEAD>
<TR><TH>Verdura <TH>Precio por kilo
<TBODY>
<TR><TD>Lechuga <TD>$1
<TR><TD>Zanahorias <TD>$10.50
<TR><TD>Nabos <TD>$100.30
</TABLE>
```

La tabla formateada puede parecerse a la siguiente:

Verdura	Precio por kilo
Lechuga	\$1
Zanahorias	\$10.50
Nabos	\$100.30

Cuando los contenidos de una celda contienen más de una aparición del carácter de alineación especificado por [char](#) y los contenidos no caben en una sola línea, el comportamiento del agente de usuario queda sin especificar. Los autores deberían por lo tanto tener cuidado cuando usen [char](#).

Nota. Los agentes de usuario visuales suelen representar los elementos [TH](#) horizontal y verticalmente centrados dentro de la celda y con una fuente en negrita.

Herencia de las especificaciones de alineación

La alineación de los contenidos de una celda puede especificarse independientemente para cada celda, o heredarse de los elementos que la contienen, como una fila, una columna, o la propia tabla.

El orden de precedencia (de más alta a más baja) de los atributos [align](#), [char](#) y [charoff](#) es el siguiente:

1. Un atributo de alineación establecido en un elemento dentro de los datos de la celda (p.ej., [P](#)).
2. Un atributo de alineación establecido en una celda ([TH](#) y [TD](#)).
3. Un atributo de alineación establecido en un elemento de grupo de columnas ([COL](#) y [COLGROUP](#)). Cuando la celda sea parte de un tramo que abarque varias columnas, la propiedad de alineación se hereda de la definición de la celda al comienzo del tramo.
4. Un atributo de alineación establecido en un elemento de fila o grupo de filas ([TR](#), [THEAD](#), [TFOOT](#) y [TBODY](#)). Cuando una celda sea parte de un tramo que abarque varias filas, la propiedad de alineación se hereda de la definición de la celda al comienzo del tramo.
5. Un atributo de alineación establecido en la tabla ([TABLE](#)).
6. El valor por defecto de la alineación.

El orden de precedencia (de más alta a más baja) para el atributo [valign](#) (así como para los otros atributos heredados [lang](#), [dir](#) y [style](#)) es el siguiente:

1. Un atributo establecido en un elemento dentro de los datos de la celda (p.ej., [P](#)).
2. Un atributo establecido en una celda ([TH](#) y [TD](#)).
3. Un atributo establecido en un elemento de fila o grupo de filas ([TR](#), [THEAD](#), [TFOOT](#) y [TBODY](#)). Cuando la celda forme parte de un tramo que abarque varias filas, el valor del atributo se hereda de la definición de la celda al comienzo del tramo.
4. Un atributo establecido en un elemento de grupo de columnas ([COL](#) y [COLGROUP](#)). Cuando la celda forme parte de un tramo que abarque varias columnas, el valor del atributo se hereda de la definición de la celda al comienzo del tramo.
5. Un atributo establecido en la tabla ([TABLE](#)).
6. El valor por defecto del atributo.

Además de eso, cuando se representen celdas, para determinar la alineación horizontal las columnas tienen preferencia sobre las filas, mientras que para la alineación vertical las filas tienen preferencia sobre las columnas.

La alineación por defecto de las celdas depende del agente de usuario. Sin embargo, los agentes de usuario deberían establecer el atributo por defecto según la direccionalidad actual (es decir, que no sea "left" en todos los casos).

Los agentes de usuario que no soporten el valor "justify" del atributo [align](#) deberían usar en su lugar el valor heredado de la direccionalidad.

***Nota.** Obsérvese que una celda puede heredar un atributo no de su padre sino de la primera celda de un tramo. Esta es una excepción a las reglas generales de herencia de atributos.*

11.3.3 Márgenes de las celdas

Definiciones de atributos

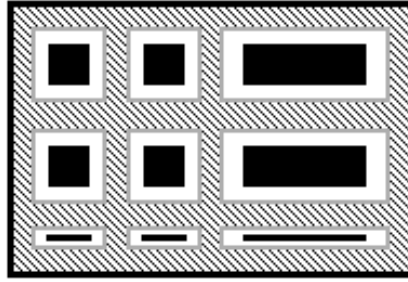


`cellspacing` = *longitud* [[CN](#)]

Este atributo especifica cuánto espacio debería dejar el agente de usuario entre el lado izquierdo de la tabla y el lado izquierdo de la columna que está más a la izquierda, entre la parte superior de la tabla y el lado superior de la fila que está más arriba, y lo mismo para los lados derecho e inferior. El atributo también especifica la cantidad de espacio entre celdas.

`cellpadding` = *longitud* [[CN](#)]

Este atributo especifica la cantidad de espacio entre el borde de la celda y sus contenidos. Si el valor de este atributo es una longitud en píxeles, los cuatro bordes deberían estar a esta distancia de los contenidos. Si el valor del atributo es una longitud porcentual, los bordes superior e inferior deberían estar igualmente separados del contenido según un porcentaje del espacio vertical disponible, y los bordes izquierdo y derecho deberían estar igualmente separados de los contenidos según un porcentaje del espacio horizontal disponible.

Estos dos atributos controlan el espacio entre y dentro de las celdas. La siguiente ilustración explica cuál es la relación entre ellos:

Borde de la tabla Cellspacing Cellpadding Contenido de la celda 

En el siguiente ejemplo, el atributo [cellspacing](#) especifica que las celdas deberían estar separadas entre sí y hasta el marco de la tabla por veinte píxeles. El atributo [cellpadding](#) especifica que el margen superior de la celda y el margen inferior de la celda estarán separados de los contenidos de la celda por el 10% del espacio vertical disponible (para un total del 20%). Análogamente, el borde izquierdo de la celda y el borde derecho de la celda estarán separados de los contenidos de la celda por el 10% del espacio horizontal disponible (para un total del 20%).

```
<TABLE cellspacing="20" cellpadding="20%">
<TR> <TD>Dato1 <TD>Dato2 <TD>Dato3
</TABLE>
```

Si una tabla o columna dada tiene una anchura fija, [cellspacing](#) y [cellpadding](#) pueden requerir más espacio que el asignado. Los agentes de usuario pueden dar a estos atributos precedencia sobre el atributo [width](#) cuando existan conflictos, pero no necesitan hacerlo.

11.4 Representación de tablas por agentes de usuario no visuales

11.4.1 Asociación de información de encabezado con celdas de datos

Los agentes de usuario no visuales tales como sintetizadores de voz o dispositivos Braille pueden usar los siguientes atributos de los elementos [TD](#) y [TH](#) para representar las celdas de las tablas más intuitivamente:

- Para una celda de datos dada, el atributo [headers](#) enumera las celdas que proporcionan información de encabezado pertinente. Para ello, se debe dar un nombre a cada celda de encabezado por medio del atributo [id](#). Obsérvese que no siempre es posible hacer una distinción clara entre las celdas que son de encabezado y las celdas de datos. Para aquellas celdas en que la diferencia no es clara, debería usar el elemento [TD](#) junto con los atributos [id](#) o [scope](#).
- Para una celda de encabezado dada, el atributo [scope](#) le dice al agente de usuario las celdas de datos para las cuales proporciona información esta celda de encabezado. Los autores pueden optar por usar este atributo en lugar de [headers](#) si lo encuentran más conveniente; los dos atributos realizan la misma función. El atributo [headers](#) es necesario generalmente cuando se colocan los encabezados en posiciones irregulares con respecto a los datos a los que se aplican.
- El atributo [abbr](#) especifica un encabezado abreviado para celdas de encabezado de modo que los agentes de usuario puedan representar la información de encabezado más rápidamente.

En el siguiente ejemplo, asignamos información de encabezado a celdas por medio del atributo [headers](#). Todas las celdas de una misma columna se refieren a la misma celda de encabezado (a través del atributo [id](#)).

```
<TABLE border="1"
  summary="Esta tabla muestra el número de tazas
  de café consumidas por cada senador, el tipo
  de café (descafeinado o normal), y si lo
  toma con azúcar.">
<CAPTION>Tazas de café consumidas por cada senador</CAPTION>
```



```

<TR>
  <TH id="t1">Nombre</TH>
  <TH id="t2">Tazas</TH>
  <TH id="t3" abbr="Tipo">Tipo de Café</TH>
  <TH id="t4">¿Azúcar?</TH>
<TR>
  <TD headers="t1">T. Sexton</TD>
  <TD headers="t2">10</TD>
  <TD headers="t3">Espresso</TD>
  <TD headers="t4">No</TD>
<TR>
  <TD headers="t1">J. Dinnen</TD>
  <TD headers="t2">5</TD>
  <TD headers="t3">Descafeinado</TD>
  <TD headers="t4">Sí</TD>
</TABLE>

```

Un sintetizador de voz podría representar esta tabla de la forma siguiente:

Título: Tazas de café consumidas por cada senador

Resumen: Esta tabla muestra el número de tazas de café consumidas por cada senador, el tipo de café (descafeinado o normal), y si lo toma con azúcar.

Nombre: T. Sexton, Tazas: 10, Tipo: Espresso, Azúcar: No

Nombre: J. Dinnen, Tazas: 5, Tipo: Decaf, Azúcar: Sí

Observe como se abrevia el encabezado "Tipo de Café" a "Tipo" usando el atributo [abbr](#).

Aquí tenemos el mismo ejemplo, utilizando el atributo [scope](#) en lugar del atributo [headers](#). Obsérvese el valor "col" del atributo [scope](#), que significa "todas las celdas de esta columna":

```

<TABLE border="1"
  summary="Esta tabla muestra el número de tazas
  de café consumidas por cada senador, el tipo
  de café (descafeinado o normal), y si lo
  toma con azúcar.">
<CAPTION>Tazas de café consumidas por cada senador</CAPTION>
<TR>
  <TH scope="col">Nombre</TH>
  <TH scope="col">Tazas</TH>
  <TH scope="col" abbr="Tipo">Tipo de Café</TH>
  <TH scope="col">¿Azúcar?</TH>
<TR>
  <TD>T. Sexton</TD>
  <TD>10</TD>
  <TD>Espresso</TD>
  <TD>No</TD>
<TR>
  <TD>J. Dinnen</TD>
  <TD>5</TD>
  <TD>Descafeinado</TD>
  <TD>Sí</TD>
</TABLE>

```

Aquí hay un ejemplo algo más complejo que ilustra otros valores del atributo [scope](#):

```

<TABLE border="1" cellpadding="5" cellspacing="2"
  summary="Cursos de historia ofrecidos en la comunidad de Bath
  ordenados por nombre del curso, tutor, resumen,
  código y tasas">

```

```

<TR>
  <TH colspan="5" scope="colgroup">
    Cursos de la Comunidad -- Bath Otoño de 1997
  </TH>
</TR>
<TR>
  <TH scope="col" abbr="Nombre">Nombre del Curso</TH>
  <TH scope="col" abbr="Tutor">Tutor del Curso</TH>
  <TH scope="col">Resumen</TH>
  <TH scope="col">Código</TH>
  <TH scope="col">Tasas</TH>
</TR>
<TR>
  <TD scope="row">Tras la Guerra Civil</TD>
  <TD>Dr. John Wroughton</TD>
  <TD>
    El curso examinará los años turbulentos que siguieron
    a 1646 en Inglaterra. <EM> Reuniones cada 6 semanas
    a partir del lunes 13 de octubre.</EM>
  </TD>
  <TD>H27</TD>
  <TD>&pound;32</TD>
</TR>
<TR>
  <TD scope="row">Una introducción a la Inglaterra anglosajona</TD>
  <TD>Mark Cottle</TD>
  <TD>
    Este curso de un día presenta una reconstrucción
    de los anglosajones y de su sociedad en el
    período medieval temprano. <EM>Sábado 18 de Octubre.</EM>
  </TD>
  <TD>H28</TD>
  <TD>&pound;18</TD>
</TR>
<TR>
  <TD scope="row">La gloriosa Grecia</TD>
  <TD>Valerie Lorenz</TD>
  <TD>
    Lugar de nacimiento de la democracia y la filosofía, patria del teatro,
    hogar de la discusión. Los romanos pueden haberlo hecho, pero los
    griegos lo hicieron primero. <EM>Clases matutinas del sábado 25
    de Octubre de 1997</EM>
  </TD>
  <TD>H30</TD>
  <TD>&pound;18</TD>
</TR>
</TABLE>

```

Un agente de usuario gráfico podría representar esto así:

Cursos de la Comunidad -- Bath Otoño de 1997

Nombre del curso	Tutor del Curso	Resumen	Código	Tasas
Tras la Guerra Civil	Dr. John Wroughton	El curso examinará los años turbulentos que siguieron a 1646 en Inglaterra. <i>Reuniones cada 6 semanas a partir del lunes 13 de octubre</i>	H27	£32
Una introducción a la Inglaterra anglosajona	Mark Cottle	Este curso de un día presenta una reconstrucción de los anglosajones y de su sociedad en el período medieval temprano. <i>Sábado 18 de octubre.</i>	H28	£18
La gloriosa Grecia	Valerie Lorenz	Lugar de nacimiento de la democracia y la filosofía, patria del teatro, hogar de la discusión. Los romanos pueden haberlo hecho, pero los griegos lo hicieron primero. <i>Clases matutinas del sábado 25 de octubre de 1997</i>	H30	£18

Obsérvese que se usa el atributo [scope](#) con el valor "row". Aunque la primera celda de cada fila contiene datos y no información de encabezado, el atributo [scope](#) hace que la celda de datos se comporte como una celda de encabezado de fila. Esto permite a los sintetizadores de voz proporcionar el nombre del curso en cuestión si se le pide, o pronunciarlo inmediatamente antes del contenido de cada celda.

11.4.2 Categorización de celdas

Los usuarios que navegan por una tabla con un agente de usuario basado en voz pueden querer oír una explicación de los contenidos de una celda además de los propios contenidos. Un modo en que el agente de usuario podría proporcionar una explicación es pronunciando la información de encabezado asociada antes de pronunciar los contenidos de la celda de datos (véase la sección sobre la [asociación de información de encabezado con celdas de datos](#)).

Los usuarios también pueden querer información sobre más de una celda, en cuyo caso proporcionar información de encabezado a nivel de celda (con [headers](#), [scope](#) y [abbr](#)) puede no ser lo más adecuado. Considérese la tabla siguiente, que clasifica los gastos de comida, hotel y transporte en dos localidades (San Jose y Seattle) a lo largo de varios días:

Informe de Gastos de Viaje

	Comidas	Hotel	Transporte	subtotales
San Jose				
25-Ago-97	37.74	112.00	45.00	
26-Ago-97	27.28	112.00	45.00	
subtotales	65.02	224.00	90.00	379.02
Seattle				
27-Ago-97	96.25	109.00	36.00	
28-Ago-97	35.00	109.00	36.00	
subtotales	131.25	218.00	72.00	421.25
Totales	196.27	442.00	162.00	800.27

Los usuarios podrían querer extraer información de la tabla por medio de preguntas:

- "¿Cuánto gasté en todas mis comidas?"
- "¿Cuánto gasté en comida el 25 de agosto?"
- "¿Cuánto gasté en total en San Jose?"

Cada pregunta implica un cálculo por parte del agente de usuario que puede implicar a cero o más celdas. Para determinar, por ejemplo, el coste de las comidas del 25 de agosto, el agente de usuario debe saber qué celdas de la tabla se refieren a "Comidas" (todas ellas), y cuáles se refieren a "Fechas" (en particular al 25 de agosto), y encontrar la intersección de los dos conjuntos.

Para permitir este tipo de preguntas, el modelo de tablas de HTML 4 permite a los autores colocar encabezados y datos de tabla en categorías. Por ejemplo, para la tabla de gastos de viaje, un autor podría agrupar las celdas de encabezado "San Jose" y "Seattle" en la categoría "Localidad", los encabezados "Comidas", "Hotel", y "Transporte" en la categoría "Gastos", y los

cuatro días en la categoría "Fecha". Las tres preguntas anteriores podrían tener entonces los significados siguientes:

- "¿Cuánto gasté en todas mis comidas?" significa "¿Cuáles son las celdas de datos de la categoría "Gastos=Comidas"?"
- "¿Cuánto gasté en comidas el 25 de agosto?" significa "¿Cuáles son las celdas de datos de las categorías "Gastos=Comidas" y "Fecha=25/08/1997"?"
- "¿Cuánto gasté en total en San Jose?" significa "¿Cuáles son las celdas de datos de las categorías "Gastos=Comidas, Hotel, Transporte" y "Localidad=San Jose"?"

Los autores categorizan una celda de encabezado o de datos estableciendo el atributo [axis](#) de la celda. Por ejemplo, en la tabla de gastos de viaje, la celda que contiene la información "San Jose" podría ser colocada en la categoría "Localidad" como sigue:

```
<TH id="a6" axis="localidad">San Jose</TH>
```

Cualquier celda que contenga información relacionada con "San Jose" debería referirse a esta celda de encabezado o bien por medio del atributo [headers](#) o bien por medio del atributo [scope](#). Así, los gastos en comidas del 25/08/1997 deberían codificarse para que se refirieran al atributo [id](#) de la celda de encabezado "San Jose" (cuyo valor aquí es "a6"):

```
<TD headers="a6">37.74</TD>
```

Cada atributo [headers](#) proporciona una lista de referencias a valores [id](#). Los autores pueden así categorizar una celda dada de diferentes maneras (o dicho de otra manera, por cualquier número de "headers", de ahí su nombre).

Abajo hemos codificado la tabla de gastos de viaje con información sobre categorías:

```
<TABLE border="1"
  summary="Esta tabla resume los gastos
    producidos durante los viajes a
    San Jose y a Seattle en agosto">
<CAPTION>
  Informe de Gastos de Viaje
</CAPTION>
<TR>
  <TH></TH>
  <TH id="a2" axis="gastos">Comidas</TH>
  <TH id="a3" axis="gastos">Hotel</TH>
  <TH id="a4" axis="gastos">Transporte</TH>
  <TD>subtotales</TD>
</TR>
<TR>
  <TH id="a6" axis="localidad">San Jose</TH>
  <TH></TH>
  <TH></TH>
  <TH></TH>
  <TD></TD>
</TR>
<TR>
  <TD id="a7" axis="fecha">25/08/1997</TD>
  <TD headers="a6 a7 a2">37.74</TD>
  <TD headers="a6 a7 a3">112.00</TD>
  <TD headers="a6 a7 a4">45.00</TD>
  <TD></TD>
</TR>
<TR>
  <TD id="a8" axis="fecha">26/08/1997</TD>
  <TD headers="a6 a8 a2">27.28</TD>
  <TD headers="a6 a8 a3">112.00</TD>
  <TD headers="a6 a8 a4">45.00</TD>
  <TD></TD>
</TR>
<TR>
```

```

<TD>subtotales</TD>
<TD>65.02</TD>
<TD>224.00</TD>
<TD>90.00</TD>
<TD>379.02</TD>
</TR>
<TR>
  <TH id="a10" axis="localidad">Seattle</TH>
  <TH></TH>
  <TH></TH>
  <TH></TH>
  <TD></TD>
</TR>
<TR>
  <TD id="a11" axis="fecha">27/08/1997</TD>
  <TD headers="a10 a11 a2">96.25</TD>
  <TD headers="a10 a11 a3">109.00</TD>
  <TD headers="a10 a11 a4">36.00</TD>
  <TD></TD>
</TR>
<TR>
  <TD id="a12" axis="fecha">28/08/1997</TD>
  <TD headers="a10 a12 a2">35.00</TD>
  <TD headers="a10 a12 a3">109.00</TD>
  <TD headers="a10 a12 a4">36.00</TD>
  <TD></TD>
</TR>
<TR>
  <TD>subtotales</TD>
  <TD>131.25</TD>
  <TD>218.00</TD>
  <TD>72.00</TD>
  <TD>421.25</TD>
</TR>
<TR>
  <TH>Totales</TH>
  <TD>196.27</TD>
  <TD>442.00</TD>
  <TD>162.00</TD>
  <TD>800.27</TD>
</TR>
</TABLE>

```

Obsérvese que al codificar la tabla de esta manera, los agentes de usuario pueden evitar confundir al usuario con información no deseada. Por ejemplo, si un sintetizador de voz tuviera que pronunciar todas las cifras de la columna "Comidas" de esta tabla en respuesta a la pregunta "¿Cuánto me gasté en comidas?", un usuario no podría distinguir los gastos diarios subtotales de los totales. Categorizando cuidadosamente los datos de las celdas, los autores permiten a los agentes de usuario hacer distinciones semánticas importantes en la representación.

Por supuesto, no hay límites en cuanto a la forma en que los autores pueden categorizar la información de una tabla. Por ejemplo, en la tabla de gastos de viaje, podríamos añadir las categorías adicionales "subtotales" y "totales".

Esta especificación no exige a los agentes de usuario que traten la información proporcionada por el atributo [axis](#), ni hace ninguna recomendación sobre cómo pueden presentar los agentes de usuario la información de [axis](#) a los usuarios, ni sobre cómo pueden preguntar los usuarios al agente de usuario sobre esta información.

Sin embargo, los agentes de usuario, en particular los sintetizadores de voz, podrían extraer la información que las distintas celdas resultado de una pregunta tienen en común. Por ejemplo, si el usuario pregunta "¿Cuánto me gasté en comidas en San Jose?", el agente de usuario podría determinar primero las celdas en cuestión (25/08/1997: 37.74, 26/08/1997: 27.28), y después representar esta información. Un agente de usuario que pronuncie esta información podría leerla así:

Localidad: San Jose. Fecha: 25/08/1997. Gastos, Comidas: 37.74

Localidad: San Jose. Fecha: 26/08/1997. Gastos, Comidas: 27.28

o más compacto:

San Jose, 25/08/1997, Comidas: 37.74

San Jose, 26/08/1997, Comidas: 27.28

Se obtendría una representación aún más económica sacando la información común y reordenándola:

San Jose, Comidas, 25/08/1997: 37.74

26/08/1997: 27.28

Los agentes de usuario que soporten este tipo de representación deberían permitir a los usuarios personalizar la representación (p.ej., con hojas de estilo).

11.4.3 Algoritmo para encontrar información de encabezado

En ausencia de información de encabezado de los atributos [scope](#) o [headers](#), los agentes de usuario pueden construir información de encabezado de acuerdo con el siguiente algoritmo. El fin del algoritmo es encontrar una lista ordenada de encabezados. (En la descripción del algoritmo que sigue, se supone que la [direccionalidad de la tabla](#) es de izquierda a derecha.)

- Primero, buscar hacia la izquierda de la posición de la celda para encontrar celdas de encabezado de fila. Después buscar hacia arriba para encontrar celdas de encabezado de columna. La búsqueda en una dirección dada se detiene cuando se alcanza el borde de la tabla o cuando se encuentra una celda de datos tras una celda de encabezado.
- Los encabezados de fila se insertan en la lista en el orden en que aparecen en la tabla. Para tablas de izquierda a derecha, los encabezados se insertan de izquierda a derecha.
- Los encabezados de columna se insertan después de los encabezados de fila, en el orden en que aparecen en la tabla, de arriba a abajo.
- Si una celda de encabezado tiene el atributo [headers](#) establecido, entonces los encabezados a que hace referencia este atributo se insertan en la lista y la búsqueda se detiene en la dirección actual.
- Las celdas [TD](#) que tengan el atributo [axis](#) establecido se tratan también como celdas de encabezado.

11.5 Tabla de muestra

Esta muestra ilustra el agrupamiento de filas y columnas. Este ejemplo está adaptado de "Developing International Software", por Nadine Kano.

En "arte ascii", la siguiente tabla:

```
<TABLE border="2" frame="hsides" rules="groups"
  summary="Páginas de códigos soportadas por las diferentes versiones
  de MS Windows.">
<CAPTION>PAGINAS DE CODIGOS SOPORTADAS POR MICROSOFT WINDOWS</CAPTION>
<COLGROUP align="center">
<COLGROUP align="left">
<COLGROUP align="center" span="2">
<COLGROUP align="center" span="3">
<THEAD valign="top">
<TR>
<TH>ID<BR>Code-Page
<TH>Nombre
<TH>ACP
<TH>OEMCP
<TH>Windows<BR>NT 3.1
<TH>Windows<BR>NT 3.51
<TH>Windows<BR>95
<TBODY>
<TR><TD>1200<TD>Unicode (BMP de ISO/IEC-10646)<TD><TD><TD>X<TD>X<TD>*
```

```

<TR><TD>1250<TD>Windows 3.1 Europa del Este<TD>X<TD><TD>X<TD>X<TD>X
<TR><TD>1251<TD>Windows 3.1 Cirílico<TD>X<TD><TD>X<TD>X<TD>X
<TR><TD>1252<TD>Windows 3.1 US (ANSI)<TD>X<TD><TD>X<TD>X<TD>X
<TR><TD>1253<TD>Windows 3.1 Griego<TD>X<TD><TD>X<TD>X<TD>X
<TR><TD>1254<TD>Windows 3.1 Turco<TD>X<TD><TD>X<TD>X<TD>X
<TR><TD>1255<TD>Hebreo<TD>X<TD><TD><TD><TD>X
<TR><TD>1256<TD>Árabe<TD>X<TD><TD><TD><TD>X
<TR><TD>1257<TD>Bálticp<TD>X<TD><TD><TD><TD>X
<TR><TD>1361<TD>Coreano (Johab)<TD>X<TD><TD><TD><TD>**<TD>X
<TBODY>
<TR><TD>437<TD>MS-DOS Estados Unidos<TD><TD>X<TD>X<TD>X<TD>X
<TR><TD>708<TD>Árabe (ASMO 708)<TD><TD>X<TD><TD><TD>X
<TR><TD>709<TD>Árabe (ASMO 449+, BCON V4)<TD><TD>X<TD><TD><TD>X
<TR><TD>710<TD>Árabe (Árabe Transparente)<TD><TD>X<TD><TD><TD>X
<TR><TD>720<TD>Árabe (ASMO Transparente)<TD><TD>X<TD><TD><TD>X
</TABLE>

```

se representaría de forma parecida a ésta:

PAGINAS DE CODIGOS SOPORTADAS POR MICROSOFT WINDOWS

```

=====
ID      | Nombre                                     | ACP  OEMCP | Windows Windows Windows
Code-Page |                                         |      | NT 3.1 NT 3.51  95
-----|-----|-----|-----|-----|-----
1200    | Unicode (BMP de ISO 10646)              |      |      |      |      | *
1250    | Windows 3.1 Europa del Este              | X    |      |      |      | X
1251    | Windows 3.1 Cirílico                     | X    |      |      |      | X
1252    | Windows 3.1 US (ANSI)                   | X    |      |      |      | X
1253    | Windows 3.1 Griego                       | X    |      |      |      | X
1254    | Windows 3.1 Turco                        | X    |      |      |      | X
1255    | Hebreo                                    | X    |      |      |      | X
1256    | Árabe                                     | X    |      |      |      | X
1257    | Báltico                                  | X    |      |      |      | X
1361    | Coreano (Johab)                          | X    |      |      |      | X
-----|-----|-----|-----|-----|-----
437     | MS-DOS Estados Unidos                    |      | X    |      | X    | X
708     | Árabe (ASMO 708)                         |      | X    |      |      | X
709     | Árabe (ASMO 449+, BCON V4)              |      | X    |      |      | X
710     | Árabe (Árabe Transparente)              |      | X    |      |      | X
720     | Árabe (ASMO Transparente)               |      | X    |      |      | X
=====

```

Un agente de usuario gráfico podría representarlo así:

PÁGINAS DE CÓDIGOS SOPORTADAS POR MICROSOFT WINDOWS

Code-Page ID	Nombre	ACP	OEMCP	Windows NT 3.1	Windows NT 3.51	Windows 95
1200	Unicode (BMP de ISO/IEC-10646)			X	X	*
1250	Windows 3.1 Europa del Este	X		X	X	X
1251	Windows 3.1 Cirílico	X		X	X	X
1252	Windows 3.1 US (ANSI)	X		X	X	X
1253	Windows 3.1 Griego	X		X	X	X
1254	Windows 3.1 Turco	X		X	X	X
1255	Hebreo	X				X
1256	Árabe	X				X
1257	Báltico	X				X
1361	Coreano(Johab)	X			**	X
437	MS-DOS Estados Unidos		X	X	X	X
708	Árabe (ASMO 708)		X			X
709	Árabe (ASMO 449+, BCON V4)		X			X
710	Árabe (Árabe Transparente)		X			X
720	Árabe (ASMO Transparente)		X			X

Este ejemplo ilustra cómo puede usarse [COLGROUP](#) para agrupar columnas y para establecer la alineación por defecto de las columnas. Análogamente se utiliza [TBODY](#) para agrupar filas. Los atributos [frame](#) y [rules](#) dicen al agente de usuario qué bordes y líneas de división representar.

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

12 Vínculos

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Introducción a los vínculos](#)
 1. [Visitar un recurso vinculado](#)
 2. [Otras relaciones de vínculo](#)
 3. [Especificación de vínculos](#)
 4. [Títulos de vínculos](#)
 5. [Internacionalización y vínculos](#)
2. [El elemento A](#)
 1. [Sintaxis de nombres de vínculos](#)
 2. [Los vínculos anidados son ilegales](#)
 3. [Vínculos con el atributo id](#)
 4. [Recursos no disponibles o no identificables](#)
3. [Relaciones entre documentos: el elemento LINK](#)
 1. [Vínculos directos e inversos](#)
 2. [Vínculos y hojas de estilo externas](#)
 3. [Vínculos y motores de búsqueda](#)
4. [Información sobre la ruta de acceso: el elemento BASE](#)
 1. [Cómo completar los URIs relativos](#)

12.1 Introducción a los vínculos

HTML ofrece muchas de las posibilidades de publicación convencionales para la creación de textos enriquecidos y documentos estructurados, pero lo que lo separa de la mayoría de los otros lenguajes para el formato de documentos son sus características para hipertexto y para documentos interactivos. Esta sección presenta el *vínculo* (o enlace, o hipervínculo), la estructura básica del hipertexto. Un vínculo es una conexión desde un recurso web a otro. Aunque es un concepto simple, el vínculo ha sido una de las principales fuerzas que ha hecho posible el éxito de la Web.

Un vínculo tiene dos extremos (llamados en inglés anchors, *anclas*), y una dirección. El vínculo comienza en el "ancla de origen" (*origen del vínculo*) y apunta al "ancla destino" (*destino del vínculo*), que puede ser cualquier recurso de la Web (p.ej., una imagen, un videoclip, un archivo de sonido, un programa, un documento HTML, un elemento dentro de un documento HTML, etc.).

12.1.1 Visitar un recurso vinculado

El comportamiento por defecto asociado a un vínculo es la obtención de otro recurso de la Web. Normalmente este comportamiento se logra implícitamente seleccionando el vínculo (p.ej., haciendo clic con el ratón, a través del teclado, etc.).

El siguiente extracto HTML contiene dos vínculos, uno cuyo destino es un documento HTML llamado "capitulo2.html" y otro cuyo destino es la imagen GIF del fichero "bosque.gif":

```
<BODY>
...texto...
```

```
<P>Encontrarás mucho más en el <A href="capitulo2.html">capítulo dos</A>.
Mira también este <A href=" ../imagenes/bosque.gif">mapa del bosque encantado.</A>
</BODY>
```

Al activar estos vínculos (haciendo clic con el ratón, a través del teclado, por comandos de voz, etc.), los usuarios pueden visitar estos recursos. Obsérvese que el atributo [href](#) de cada origen de vínculo especifica la dirección del destino por medio de un URI.

El destino de un vínculo puede ser un elemento contenido en un documento HTML. Se le debe asignar un nombre al destino, y cualquier URI que se refiera a este destino debe incluir el nombre como [identificador de fragmento](#).

Los destinos de vínculo en los documentos HTML pueden especificarse o bien mediante el elemento [A](#) (dándoles un nombre con el atributo [name](#)), o mediante cualquier otro elemento (dándole un nombre con el atributo [id](#)).

Así, por ejemplo, un autor podría crear una tabla de contenidos cuyas entradas estuvieran vinculadas con los elementos de encabezado [H2](#), [H3](#), etc., del mismo documento. Si usáramos el elemento [A](#) para crear los destinos de los vínculos, escribiríamos:

```
<H1>Tabla de Contenidos</H1>
<P><A href="#seccion1">Introducción</A><BR>
<A href="#seccion2">Antecedentes</A><BR>
<A href="#seccion2.1">Experiencias personales</A><BR>
...el resto de la tabla de contenidos...
...el cuerpo del documento...
<H2><A name="seccion1">Introducción</A></H2>
...sección 1...
<H2><A name="seccion2">Antecedentes</A></H2>
...sección 2...
<H3><A name="seccion2.1">Experiencias personales</A></H3>
...sección 2.1...
```

Podemos lograr el mismo efecto haciendo que los elementos de encabezado sean ellos mismos los destinos de los vínculos:

```
<H1>Tabla de Contenidos</H1>
<P><A href="#seccion1">Introducción</A><BR>
<A href="#seccion2">Antecedentes</A><BR>
<A href="#seccion2.1">Experiencias personales</A><BR>
...el resto de la tabla de contenidos...
...el cuerpo del documento...
<H2 id="seccion1">Introducción</H2>
...sección 1...
<H2 id="seccion2">Antecedentes</H2>
...sección 2...
<H3 id="seccion2.1">Experiencias personales</H3>
...sección 2.1...
```

12.1.2 Otras relaciones de vínculo

La *utilización* más común de un vínculo es con mucho la obtención de otro recurso de la Web, tal y como se ilustra en los ejemplos precedentes. Sin embargo, los autores pueden insertar en sus documentos vínculos que expresen otras relaciones entre recursos distintas a "activar este vínculo para visitar ese recurso relacionado". Para los vínculos que expresan otros tipos de relaciones, se especifican uno o más [tipos de vínculo](#) en el origen del vínculo.

Los *papeles* de un vínculo definido por [A](#) o [LINK](#) se especifican mediante los atributos [rel](#) y [rev](#).

Por ejemplo, los vínculos definidos por el elemento [LINK](#) pueden describir la posición de un documento dentro de una serie de documentos. En el extracto que sigue, los vínculos contenidos en el documento llamado "Capítulo 5" apuntan a los capítulos anterior y siguiente:

```
<HEAD>
```

```
...otras informaciones de cabecera...
<TITLE>Capítulo 5</TITLE>
<LINK rel="prev" href="capitulo4.html">
<LINK rel="next" href="capitulo6.html">
</HEAD>
```

El tipo de vínculo del primer vínculo es "prev" ("anterior"), y el del segundo es "next" ("siguiente"), dos de los diferentes [tipos de vínculos](#) reconocidos. Los vínculos especificados con [LINK](#) **no** son representados con los contenidos del documento, aunque los agentes de usuario pueden representarlos de otras maneras (p.ej., como herramientas de navegación).

Incluso aunque no se usen para navegar, estos vínculos pueden ser interpretados de formas interesantes. Por ejemplo, un agente de usuario que imprima una serie de documentos HTML como un documento único, puede usar la información de estos vínculos como la base para formar un documento lineal coherente. Más adelante se da más información sobre la utilización de [vínculos para el beneficio de los motores de búsqueda](#).

12.1.3 Especificación de vínculos

Aunque algunos elementos y atributos de HTML crean vínculos a otros recursos (p.ej., el elemento [IMG](#), el elemento [FORM](#), etc.), este capítulo se refiere a los vínculos creados con los elementos [LINK](#) y [A](#). El elemento [LINK](#) sólo puede aparecer en la cabecera del documento. El elemento [A](#) sólo puede aparecer en el cuerpo.

Si el atributo [href](#) del elemento [A](#) tiene algún valor, el elemento define un origen de vínculo que puede ser activado por el usuario para obtener un recurso de la Web. El origen del vínculo está en el lugar donde aparece el elemento [A](#) y el destino del vínculo es el recurso web.

El agente de usuario puede manipular el recurso obtenido de diferentes maneras: abriendo un nuevo documento HTML en la misma ventana del agente de usuario, abriendo un nuevo documento HTML en una ventana diferente, iniciando un nuevo programa que maneje el recurso, etc. Como el elemento [A](#) tiene contenido (texto, imágenes, etc.), los agentes de usuario pueden representar este contenido de modo tal que se indique la presencia de un vínculo (p.ej., subrayando el contenido).

Si el atributo [name](#) o el atributo [id](#) del elemento [A](#) toman algún valor, el elemento define un posible destino de otros vínculos.

Los autores pueden establecer los atributos [name](#) y [href](#) de un mismo elemento [A](#) simultáneamente.

El elemento [LINK](#) define una relación entre el documento actual y otro recurso. Aunque [LINK](#) no tiene contenido, la relación que define puede ser representada por algunos agentes de usuario.

12.1.4 Títulos de vínculos

Se puede especificar el atributo [title](#) tanto para [A](#) como para [LINK](#) para añadir información sobre la naturaleza de un vínculo. Esta información puede ser pronunciada por un agente de usuario, representada como una indicación visual ("tool-tip"), producir un cambio en la imagen del cursor, etc.

Así, podemos desarrollar uno de los [ejemplos anteriores](#) proporcionando un título para cada vínculo:

```
<BODY>
...texto...
<P>Encontrarás mucho más en el <A href="capitulo2.html"
    title="Ir al capítulo 2">capítulo dos</A>.
<A href="./capitulo2.html"
    title="Obtener capítulo dos.">capítulo dos</A>.
Mira también este <A href="./imagenes/bosque.gif"
    title="imagen GIF del bosque encantado">mapa del
bosque encantado.</A>
</BODY>
```

12.1.5 Internacionalización y vínculos

Debido a que los vínculos pueden apuntar a documentos codificados con diferentes [codificaciones de caracteres](#), los elementos [A](#) y [LINK](#) soportan el atributo [charset](#). Este atributo permite a los autores avisar a los agentes de usuario sobre la codificación de los datos del otro extremo del vínculo.

El atributo [hreflang](#) proporciona a los agentes de usuario información sobre el idioma del recurso del otro extremo del vínculo, igual que el atributo [lang](#) proporciona información sobre el idioma del contenido o de los valores de los atributos de un elemento.

Los agentes de usuario, provistos de esta información adicional, deberían ser capaces de evitar la presentación de "basura" al usuario. En su lugar, pueden o bien localizar los recursos necesarios para la presentación correcta del documento o, si no pueden localizar los recursos, deberían al menos advertir al usuario de que el documento será ilegible, explicando las causas.

12.2 El elemento A

```
<!ELEMENT A - - (%inline;)* -(A) -- ancla -->
<!ATTLIST A
  %attrs; -- %coreattrs, %il8n, %events --
  charset %Charset; #IMPLIED -- codificación de caracteres
  del recurso vinculado --
  type %ContentType; #IMPLIED -- tipo de contenido consultivo --
  name CDATA #IMPLIED -- destino de vínculo con nombre --
  href %URI; #IMPLIED -- URI del recurso vinculado --
  hreflang %LanguageCode; #IMPLIED -- código de idioma --
  rel %LinkTypes; #IMPLIED -- tipos de vínculos directos --
  rev %LinkTypes; #IMPLIED -- tipos de vínculos inversos --
  accesskey %Character; #IMPLIED -- carácter de la tecla de accesibilidad --
  shape %Shape; rect -- para mapas de imágenes en el cliente --
  coords %Coords; #IMPLIED -- para mapas de imágenes en el cliente --
  tabindex NUMBER #IMPLIED -- posición en el orden de tabulación --
  onfocus %Script; #IMPLIED -- el foco se dirigió hacia el elemento --
  onblur %Script; #IMPLIED -- el elemento perdió el foco --
>
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria***

Definiciones de atributos

name = [CDATA](#) [CS]

Este atributo asigna un nombre al elemento actual de modo que éste pueda actuar como destino de otro vínculo. El valor de este atributo debe ser un nombre único de destino. El campo de acción de este nombre es el documento actual. Obsérvese que este atributo comparte el mismo espacio de nombres que el atributo [id](#).

href = [uri](#) [CT]

Este atributo especifica la localización de un recurso de la Web, definiendo así un vínculo entre el elemento actual (el origen del vínculo) y el destino del vínculo definido por este atributo.

hreflang = [código de idioma](#) [CI]

Este atributo especifica el idioma base del recurso designado por [href](#) y sólo puede utilizarse si se especifica [href](#).

type = [tipo de contenido](#) [CI]

Este atributo da una indicación sobre el tipo de contenido de los contenidos disponibles en la dirección del destino del vínculo. Permite a los agentes de usuario optar por la no obtención del contenido si se les avisa de que van a obtener contenidos de un tipo de contenido que no soportan.

Los autores que usen este atributo afrontan la responsabilidad y el riesgo de que el valor especificado puede llegar a ser inconsistente con el contenido disponible en la dirección de destino del vínculo.

Para la lista actual de los tipos de contenido registrados, consulte [\[MIMETYPES\]](#).

rel = [tipos de vínculo](#) [CI]

Este atributo describe la relación entre el documento actual y el destino del vínculo, especificado por el atributo [href](#).

El valor de este atributo es una lista de tipos de vínculo separados por espacios.

rev = [tipos de vínculo](#) [CI]

Este atributo se utiliza para describir un [vínculo inverso](#) desde el origen del vínculo, especificado por el atributo [href](#), y el documento actual. El valor de este atributo es una lista de tipos de vínculo separados por espacios.

charset = [codificación de caracteres](#) [CI]

Este atributo especifica la codificación de caracteres del recurso designado por el elemento. Consulte la sección sobre [codificaciones de caracteres](#) para más detalles.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [shape](#) y [coords](#) (mapas de imágenes)
- [onfocus](#), [onblur](#), [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)
- [target](#) (información sobre el marco destino)
- [tabindex](#) (navegación con tabulador)
- [accesskey](#) (teclas de acceso)

Cada elemento [A](#) define un origen o destino de vínculo.

1. El contenido de un elemento [A](#) define la posición del origen o del destino del vínculo.
2. El atributo [name](#) da un nombre al elemento de modo que pueda ser el destino de cero o más vínculos (ver también [vínculos con id](#)).
3. El atributo [href](#) hace que este elemento sea el origen de exactamente un vínculo.

Los autores también pueden crear un elemento [A](#) que no sea origen ni destino de vínculo, es decir, que no especifica ni [href](#), ni [name](#), ni [id](#). Los valores de estos atributos pueden ser establecidos posteriormente a través de [scripts](#).

En el ejemplo que sigue, el elemento [A](#) define un vínculo. El origen del vínculo es el texto "Sitio web del W3C" y el destino del vínculo es "http://www.w3.org/":

```
Para más información sobre el W3C, consulte el
<A href="http://www.w3.org/">sitio web del W3C</A>.
```

Este vínculo designa la página principal del World Wide Web Consortium. Cuando un usuario active este vínculo en un agente de usuario, el agente de usuario obtendrá el recurso, en este caso un documento HTML.

Los agentes de usuario suelen representar los vínculos de modo tal que éstos son obvios para los usuarios (subrayándolos, invirtiendo los colores, etc.). La representación exacta depende del agente de usuario. La representación puede variar según que el usuario haya visitado ya el vínculo o no. Una posible representación visual del vínculo anterior podría ser ésta:

```
Para más información sobre el W3C, consulte el sitio web del W3C.
~~~~~
```

Para decir explícitamente a los agentes de usuario cuál es la codificación de caracteres de la página destino, se establece el atributo [charset](#):

```
Para más información sobre el W3C, consulte el
<A href="http://www.w3.org/" charset="ISO-8859-1">sitio web del W3C</A>
```

Supongamos que definimos un destino de vínculo llamado "destino-uno" en el fichero "uno.html".

```
...texto antes del destino de vínculo...
<A name="destino-uno">Esta es la localización del destino de vínculo número uno.</A>
...texto después del destino de vínculo...
```

Esto crea un destino de vínculo alrededor del texto "Esta es la localización del destino de vínculo número uno".

Normalmente, los contenidos de [A](#) no se representan de manera especial cuando [A](#) sólo define un destino de vínculo.

Una vez definido el destino, podemos crear un vínculo a él desde el mismo documento o desde otro diferente. Los URIs referidos a elementos [A](#) que son destinos de vínculo, contienen un carácter "#" seguido del nombre del elemento (el [identificador de fragmento](#)). Aquí tenemos varios ejemplos:

- Un URI absoluto: `http://www.miempresa.com/uno.html#destino-uno`
- Un URI relativo: `./uno.html#destino-uno` o `uno.html#destino-uno`
- Cuando el vínculo está definido en el mismo documento: `#destino-uno`

Así, un vínculo definido en el fichero "dos.html" contenido en el mismo directorio que "uno.html" se referiría a este destino del modo siguiente:

...texto antes del vínculo...

Para más información, consulte el `destino uno`.
...texto después del vínculo...

El elemento [A](#) del siguiente ejemplo especifica un vínculo (con [href](#)) y crea un destino de vínculo (con [name](#)) simultáneamente:

```
¡Acabo de volver de mis vacaciones! Aquí os pongo
<A name="destino-dos"
  href="http://www.algunaempresa.com/Gente/Juan/vacaciones/familia.png">
una foto de mi familia junto al lago.</A>
```

Este ejemplo contiene un vínculo a un tipo diferente de recurso web (una imagen PNG). Al activar el vínculo la imagen debería ser obtenida de la Web (y posiblemente mostrada si el sistema ha sido configurado para ello).

Nota. Los agentes de usuario deberían ser capaces de encontrar destinos de vínculo creados con elementos [A](#) vacíos, pero algunos de ellos no pueden. Por ejemplo, algunos agentes de usuario pueden no encontrar el "destino-vacio" del siguiente fragmento HTML:

```
<A name="destino-vacio"></A>
<EM>...código HTML...</EM>
<A href="#destino-vacio">Vínculo a destino vacío</A>
```

12.2.1 Sintaxis de nombres de vínculos

Un nombre de vínculo es el valor del atributo [name](#) o del atributo [id](#) cuando éste se use en el contexto de vínculos. Los nombres de vínculos deben seguir las siguientes reglas:

- **Unicidad:** Los nombres de vínculo deben ser únicos dentro de un documento. No pueden aparecer en un mismo documento nombres de vínculo que sólo se diferencien por las letras mayúsculas y minúsculas.
- **Emparejamiento de cadenas:** Las comparaciones entre [identificadores de fragmento](#) y nombres de vínculos deben dar resultados exactos (incluyendo mayúsculas y minúsculas).

Así, el ejemplo siguiente es correcto con respecto al emparejamiento de cadenas y los agentes de usuario deben considerarlo como tal:

```
<P><A href="#xxx">...</A>
...más texto...
<P><A name="xxx">...</A>
```

EJEMPLO ILEGAL:

El siguiente ejemplo es ilegal con respecto a la unicidad, ya que los dos nombres son iguales y sólo se diferencian por las mayúsculas:

```
<P><A name="xxx">...</A>
<P><A name="XXX">...</A>
```

Aunque el siguiente extracto es HTML legal, el comportamiento del agente de usuario es indefinido; algunos agentes de

usuario pueden considerar (incorrectamente) que hay un emparejamiento, y otros no.

```
<P><A href="#xxx">...</A>
...más texto...
<P><A name="XXX">...</A>
```

Los nombres de los vínculos deberían restringirse a los caracteres ASCII. Consulte el apéndice para más información sobre [caracteres no ASCII en valores de atributos URI](#).

12.2.2 Los vínculos anidados son ilegales

Los vínculos definidos mediante elementos [A](#) no deben ser anidados; un elemento [A](#) no debe contener ningún otro elemento [A](#).

Como el DTD define que el elemento [LINK](#) es vacío, los elementos [LINK](#) tampoco pueden estar anidados.

12.2.3 Vínculos con el atributo id

El atributo [id](#) puede utilizarse para crear un destino de vínculo en la etiqueta inicial de cualquier elemento (incluyendo al elemento [A](#)).

Este ejemplo ilustra la utilización del atributo [id](#) para poner un destino de vínculo en un elemento [H2](#). El destino es vinculado por medio del elemento [A](#).

```
Puedes leer más sobre esto en la <A href="#seccion2">Sección Dos</A>.
...más abajo en el mismo documento
<H2 id="seccion2">Sección Dos</H2>
...más abajo en el mismo documento
<P>Consulte la <A href="#seccion2">Sección Dos</A> más arriba
para más detalles.
```

El siguiente ejemplo le da un nombre a un destino de vínculo con el atributo [id](#):

```
¡Acabo de volver de mis vacaciones! Aquí os pongo una
<A id="destino-dos">foto de mi familia junto al lago.</A>
```

Los atributos [id](#) y [name](#) comparten el mismo espacio de nombres. Esto significa que no pueden definir cada uno un destino de vínculo con un mismo nombre en el mismo documento. Está permitido usar ambos atributos para especificar un identificador único para un mismo elemento en los siguientes elementos: [A](#), [APPLET](#), [FORM](#), [FRAME](#), [IFRAME](#), [IMG](#) y [MAP](#). Cuando se usen ambos atributos en un mismo elemento, sus valores deben ser idénticos.

EJEMPLO ILEGAL:

El siguiente extracto es HTML ilegal, ya que estos atributos declaran el mismo nombre dos veces en el mismo documento.

```
<A href="#a1">...</A>
...
<H1 id="a1">
...páginas y más páginas...
<A name="a1"></A>
```

El siguiente ejemplo ilustra el hecho de que [id](#) y [name](#) deben tener el mismo valor cuando los dos aparecen en la etiqueta inicial de un elemento:

```
<P><A name="a1" id="a1" href="#a1">...</A>
```

Debido a su especificación en el DTD de HTML, el atributo [name](#) puede contener [referencias de caracteres](#). Así, el valor `Dürst` es un valor válido para el atributo [name](#), como lo es `Dül;rst`. El atributo [id](#), por contra, no puede contener referencias de caracteres.

¿Es mejor utilizar [id](#), o [name](#)? Los autores deberían considerar las siguientes cuestiones cuando decidan entre usar [id](#) o

name para el nombre de un vínculo:

- El atributo *id* puede actuar como algo más que un nombre de vínculo (p.ej., selector en una hoja de estilo, identificador en un proceso, etc.).
- Algunos agentes de usuario antiguos no soportan los vínculos creados con el atributo *id*.
- El atributo *name* permite nombres de vínculos más ricos (con *entidades*).

12.2.4 Recursos no disponibles o no identificables

Una referencia a un recurso no disponible o no identificable es un error. Aunque los agentes de usuario puedan diferir en el modo de tratar tales errores, recomendamos el comportamiento siguiente:

- Si un agente de usuario no puede localizar un recurso vinculado, debería alertar al usuario.
- Si un agente de usuario no puede identificar el tipo de un recurso vinculado, debería intentar procesarlo de todos modos. Debería alertar al usuario y puede permitir al usuario intervenir para identificar el tipo de documento.

12.3 Relaciones entre documentos: el elemento LINK

```
<!ELEMENT LINK - O EMPTY -- un vínculo independiente del medio -->
<!ATTLIST LINK
  %attrs; -- %coreattrs, %i18n, %events --
  charset %Charset; #IMPLIED -- codificación de caracteres del
  recurso vinculado --
  href %URI; #IMPLIED -- URI del recurso vinculado --
  hreflang %LanguageCode; #IMPLIED -- código de idioma --
  type %ContentType; #IMPLIED -- tipo consultivo de contenido --
  rel %LinkTypes; #IMPLIED -- tipos de vínculos directos --
  rev %LinkTypes; #IMPLIED -- tipos de vínculos inversos --
  media %MediaDesc; #IMPLIED -- para representar en estos medios --
>
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **prohibida***

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)
- [href](#), [hreflang](#), [type](#), [rel](#), [rev](#) (vínculos)
- [target](#) (información sobre el marco destino)
- [media](#) (información de estilo en cabecera)
- [charset](#) (codificaciones de caracteres)

Este elemento define un vínculo. A diferencia de [A](#), sólo puede aparecer en la sección [HEAD](#) de un documento, aunque puede aparecer cualquier número de veces. Si bien [LINK](#) no tiene contenido, conlleva cierta información relacional que puede ser representada por los agentes de usuario de distintas maneras (p.ej., una barra de herramientas con un menú desplegable para los vínculos).

Este ejemplo ilustra cómo pueden aparecer varias definiciones [LINK](#) en la sección [HEAD](#) de un documento. El documento actual es "Capitulo2.html". El atributo [rel](#) especifica la relación del documento vinculado con el documento actual. Los valores "Index", "Next" y "Prev" se explican en la sección sobre [tipos de vínculos](#).


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
  <TITLE>Capítulo 2</TITLE>
  <LINK rel="Index" href="../indice.html">
  <LINK rel="Next" href="Capitulo3.html">
  <LINK rel="Prev" href="Capitulo1.html">
</HEAD>
...el resto del documento...
```

12.3.1 Vínculos directos e inversos

Los atributos [rel](#) y [rev](#) juegan papeles complementarios: el atributo [rel](#) especifica un vínculo directo y el atributo [rev](#) especifica un vínculo inverso.

Consideremos dos documentos A y B.

Documento A: <LINK href="docB" rel="blabla">

Esto significa exactamente lo mismo que:

Documento B: <LINK href="docA" rev="blabla">

Ambos atributos pueden especificarse simultáneamente.

12.3.2 Vínculos y hojas de estilo externas

Cuando el elemento [LINK](#) vincula una hoja de estilo externa a un documento, el atributo [type](#) especifica el lenguaje de la hoja de estilo, y el atributo [media](#) especifica el medio o medios de representación deseados. Los agentes de usuario pueden ahorrar tiempo obteniendo de la red sólo aquellas hojas de estilo que se apliquen al dispositivo actual.

Los [tipos de medios](#) se tratan con más profundidad en la sección sobre hojas de estilo.

12.3.3 Vínculos y motores de búsqueda

Los autores pueden usar el elemento [LINK](#) para proporcionar distintas informaciones a los motores de búsqueda, entre las que se incluyen:

- Vínculos a versiones alternativas de un documento, escritas en otro idioma.
- Vínculos a versiones alternativas de un documento, diseñadas para medios diferentes, por ejemplo, una versión especial para imprimir.
- Vínculos a la página inicial de un conjunto de documentos.

Los ejemplos que siguen ilustran cómo puede combinarse la información sobre el idioma, los tipos de medios y los tipos de vínculos para mejorar el procesamiento de los documentos por parte de los motores de búsqueda.

En el siguiente ejemplo, usamos el atributo [hreflang](#) para decir a los motores de búsqueda dónde encontrar las versiones en neerlandés, portugués y árabe de un documento. Obsérvese el uso del atributo [charset](#) para el manual en árabe. Obsérvese también el uso del atributo [lang](#) para indicar que el valor del atributo [title](#) del elemento [LINK](#) que designa al manual francés está en francés.

```
<HEAD>
<TITLE>El manual en castellano</TITLE>
<LINK title="El manual en neerlandés"
  type="text/html"
  rel="alternate"
  hreflang="nl"
  href="http://algunsitio.com/manual/neerlandes.html">
<LINK title="El manual en portugués"
```

```

    type="text/html "
    rel="alternate"
    hreflang="pt "
    href="http://algunsitio.com/manual/portugues.html ">
<LINK title="El manual en árabe"
    type="text/html "
    rel="alternate"
    charset="ISO-8859-6 "
    hreflang="ar "
    href="http://algunsitio.com/manual/arabe.html ">
<LINK lang="fr" title="La documentation en Fran&ccedil;ais"
    type="text/html "
    rel="alternate"
    hreflang="fr "
    href="http://algunsitio.com/manual/frances.html ">
</HEAD>

```

En el siguiente ejemplo, le decimos a los motores de búsqueda dónde encontrar la versión impresa de un manual.

```

<HEAD>
<TITLE>Manual de referencia</TITLE>
<LINK media="print" title="El manual en postscript "
    type="application/postscript "
    rel="alternate"
    href="http://algunsitio.com/manual/postscript.ps ">
</HEAD>

```

En el siguiente ejemplo, le decimos a los motores de búsqueda dónde encontrar la portada de un conjunto de documentos.

```

<HEAD>
<TITLE>Manual de referencia -- Página 5</TITLE>
<LINK rel="Start" title="La primera página del manual"
    type="text/html "
    href="http://algunsitio.com/manual/portada.html ">
</HEAD>

```

En las notas del apéndice se da más información sobre [cómo ayudar a los motores de búsqueda a indexar su sitio web](#).

12.4 Información sobre la ruta de acceso: el elemento BASE

```

<!ELEMENT BASE - O EMPTY          -- URI base del documento -->
<!ATTLIST BASE
  href          %URI;          #REQUIRED -- URI que actúa como URI base --
>

```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **prohibida***

Definiciones de atributos

`href = uri [CT]`

Este atributo especifica un URI absoluto que actúa como el URI base para completar URIs relativos.

Atributos definidos en otros lugares

- [target](#) ([información sobre el marco destino](#))

En HTML, los vínculos y las referencias a recursos externos como imágenes, aplicaciones, programas de procesamiento de formularios, hojas de estilo, etc., se especifican siempre mediante un URI. Los URIs relativos se [completan](#) de acuerdo con un URI base, el cual puede provenir de distintas fuentes. El elemento [BASE](#) permite a los autores especificar el URI base de un documento explícitamente.

Si está presente, el elemento [BASE](#) debe aparecer en la sección [HEAD](#) del documento HTML, antes de cualquier elemento que haga referencia a una fuente externa. La información de ruta especificada por el elemento [BASE](#) sólo afecta a los URIs del documento en el cual aparece el elemento.

Por ejemplo, dadas las siguientes declaraciones [BASE](#) y [A](#):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Nuestros productos</TITLE>
    <BASE href="http://www.pajareria.com/productos/intro.html">
  </HEAD>

  <BODY>
    <P>¿Conoce ya nuestras <A href=" ../jaulas/pajaros.gif">Jaulas para Pájaros</A>?
  </BODY>
</HTML>
```

el URI relativo "../jaulas/pajaros.gif" se completaría a:

<http://www.pajareria.com/jaulas/pajaros.gif>

12.4.1 Cómo completar los URIs relativos

Los agentes de usuario deben calcular el URI base para completar los URIs relativos de acuerdo con [\[RFC1808\]](#), sección 3. A continuación se describe cómo se aplica [\[RFC1808\]](#) al HTML específicamente.

Los agentes de usuario deben calcular el URI base de acuerdo con el siguiente orden de precedencia (de prioridad más alta a prioridad más baja):

1. El URI base se establece por medio del elemento [BASE](#).
2. El URI base viene dado por metadatos detectados durante una interacción del protocolo, por ejemplo en un encabezado HTTP (ver [\[RFC2616\]](#)).
3. Por defecto, el URI base es el del documento actual. No todos los documentos HTML tienen un URI base (p.ej., un documento HTML válido puede aparecer en un mensaje de correo electrónico y no ser designado con un URI). Tales documentos HTML se consideran erróneos si contienen URIs relativos que se basen en un URI base por defecto.

Por otra parte, los elementos [OBJECT](#) y [APPLET](#) definen atributos que tienen precedencia sobre el valor establecido por el elemento [BASE](#). Consulte las definiciones de estos elementos para más información sobre cuestiones específicas de estos elementos relacionadas con los URIs.

***Nota.** Para las versiones de HTTP que definan un encabezado Link, los agentes de usuario deberían tratar estos encabezados exactamente igual que los elementos [LINK](#) del documento. HTTP 1.1, tal y como está definido [\[RFC2616\]](#), no incluye un campo de encabezado Link (consulte la sección 19.6.3).*

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

13 Objetos, Imágenes y Aplicaciones

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Introducción a los objetos, imágenes y aplicaciones](#)
2. [Incluir una imagen: el elemento IMG](#)
3. [Inclusión genérica: el elemento OBJECT](#)
 1. [Reglas para representar objetos](#)
 2. [Inicialización de objetos: el elemento PARAM](#)
 3. [Esquemas globales de nombres para objetos](#)
 4. [Declaración y creación de un objeto](#)
4. [Incluir un applet: el elemento APPLET](#)
5. [Notas sobre la inclusión de documentos](#)
6. [Mapas de imágenes](#)
 1. [Mapas de imágenes en el lado del cliente: los elementos MAP y AREA](#)
 - [Ejemplos de mapas de imágenes en el lado del cliente](#)
 2. [Mapas de imágenes en el lado del servidor](#)
7. [Presentación visual de imágenes, objetos y aplicaciones](#)
 1. [Anchura y altura](#)
 2. [Espacio en blanco alrededor de imágenes y objetos](#)
 3. [Bordes](#)
 4. [Alineación](#)
8. [Cómo especificar texto alternativo](#)

13.1 Introducción a los objetos, imágenes y aplicaciones

Las características multimedia de HTML permiten a los autores incluir en sus páginas imágenes, aplicaciones (programas que se descargan automáticamente y se ejecutan en la máquina del usuario), videoclips, y otros documentos HTML en sus páginas.

Por ejemplo, para incluir una imagen PNG en un documento, los autores pueden escribir:

```
<BODY>
<P>Aquí tienen un primer plano del Gran Cañón:
<OBJECT data="canyon.png" type="image/png">
Esto es un <EM>primer plano</EM> del Gran Cañón.
</OBJECT>
</BODY>
```

Las versiones anteriores de HTML permitían a los autores incluir imágenes (por medio de [IMG](#)) y aplicaciones (por medio de [APPLET](#)). Estos elementos tienen varias limitaciones:

- No sirven para resolver el problema más general de cómo incluir tipos de medios nuevos y futuros.
- El elemento [APPLET](#) sólo funciona con aplicaciones basadas en Java. Este elemento está [desaprobado](#) en favor de [OBJECT](#).

- Plantean problemas de accesibilidad.

Para solucionar estos problemas, HTML 4 introduce el elemento [OBJECT](#), que ofrece una solución universal para la inclusión de objetos genéricos. El elemento [OBJECT](#) permite a los autores HTML especificar todo lo que necesita un agente de usuario para la representación de un objeto: código fuente, valores iniciales, y datos en tiempo de ejecución. En esta especificación, el término "objeto" se utiliza para describir las cosas que la gente quiere colocar en los documentos HTML; otros términos usados habitualmente son: applets, plug-ins, controladores de medios, etc.

El nuevo elemento [OBJECT](#) asume así algunas de las tareas realizadas por los elementos existentes. Considérese la siguiente tabla de funcionalidades:

Tipo de inclusión	Elemento específico	Elemento genérico
Imagen	IMG	OBJECT
Aplicación	APPLET (Desaprobado)	OBJECT
Otro documento HTML	IFRAME	OBJECT

La tabla indica que cada tipo de inclusión tiene una solución específica y una general. El elemento genérico [OBJECT](#) servirá como solución de implementación de los tipos de medios futuros.

Para incluir imágenes, los autores pueden usar el elemento [OBJECT](#) o el elemento [IMG](#).

Para incluir aplicaciones, los autores deberían usar el elemento [OBJECT](#) ya que el elemento [APPLET](#) está **desaprobado**.

Para incluir un documento HTML en otros, los autores pueden usar o bien el nuevo elemento [IFRAME](#) o bien el elemento [OBJECT](#). En ambos casos, el documento incluido sigue siendo independiente del documento principal. Los agentes de usuario visuales pueden presentar el documento incluido en una ventana diferenciada del documento principal. Consulte las notas [sobre documentos incluidos](#) para una comparación entre [OBJECT](#) e [IFRAME](#) en lo que a inclusión de documentos se refiere.

Las imágenes y otros objetos incluidos pueden tener hipervínculos asociados a ellos, tanto a través de los [mecanismos de vinculación](#) estándar, como también a través de [mapas de imágenes](#). Un mapa de imágenes especifica las regiones geométricas activas de un objeto incluido, y asigna un vínculo a cada región. Cuando se activan, estos vínculos pueden hacer que se abra un documento, que se ejecute un programa en el servidor, etc.

En las secciones siguientes, hablamos sobre los distintos mecanismos que están a disposición de los autores para inclusiones multimedia y para crear mapas de imágenes de dichas inclusiones.

13.2 Incluir una imagen: el elemento IMG

```
<!-- Para evitar problemas con AAUU de sólo texto así
como para hacer el contenido de las imágenes inteligibles
y navegables para los usuarios de AAUU no visuales, es
necesario proporcionar una descripción con ALT, y evitar
mapas de imágenes en el lado del servidor -->
<!ELEMENT IMG - O EMPTY -- Imagen incluida -->
<!ATTLIST IMG
  %attrs; -- %coreattrs, %i18n, %events --
  src %URI; #REQUIRED -- URI de la imagen a incluir --
  alt %Text; #REQUIRED -- descripción corta --
  longdesc %URI; #IMPLIED -- vínculo a descripción larga
  (complementa a alt) --
  name CDATA #IMPLIED -- nombre de la imagen para los scripts --
  height %Length; #IMPLIED -- especificar nueva altura --
  width %Length; #IMPLIED -- especificar nueva anchura --
  usemap %URI; #IMPLIED -- usar mapa de imágenes en el cliente --
  ismap (ismap) #IMPLIED -- usar mapa de imágenes en el servidor --
>
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **prohibida***

*Definiciones de atributos*src = [uri \[CT\]](#)

Este atributo especifica la localización del recurso de imagen. Ejemplos de formatos de imagen ampliamente reconocidos son GIF, JPEG y PNG.

longdesc = [uri \[CT\]](#)

Este atributo especifica un vínculo a una descripción larga de la imagen. Esta descripción debería ser un complemento de la descripción corta proporcionada mediante al atributo [alt](#). Cuando la imagen tenga asociado un [mapa de imágenes](#), este atributo debería proporcionar información sobre los contenidos del mapa de imágenes. Esto es particularmente importante para mapas de imágenes en el lado del servidor. Como un elemento [IMG](#) puede estar dentro del contenido de un elemento [A](#), el mecanismo de la interfaz del usuario del agente de usuario para acceder al recurso "longdesc" de aquél debe ser diferente del mecanismo para acceder al recurso [href](#) de éste.

name = [cdata \[CI\]](#)

Este atributo asigna un nombre al elemento de modo que se pueda hacer referencia a él desde hojas de estilo o scripts. **Nota.** Este atributo ha sido incluido por motivos de compatibilidad con versiones anteriores. Las aplicaciones deberían usar el atributo [id](#) para identificar a los elementos.

Atributos definidos en otros lugares

- [id](#), [class](#) ([identificadores a nivel de documento](#))
- [alt](#) ([texto alternativo](#))
- [lang](#) ([información sobre el idioma](#)), [dir](#) ([dirección del texto](#))
- [title](#) ([título del elemento](#))
- [style](#) ([información de estilo en línea](#))
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) ([eventos intrínsecos](#))
- [ismap](#), [usemap](#) ([mapas de imágenes en el lado del cliente](#))
- [align](#), [width](#), [height](#), [border](#), [hspace](#), [vspace](#) ([presentación visual de objetos, imágenes y aplicaciones](#))

El elemento [IMG](#) incluye una imagen en el documento actual en la posición de la definición del elemento. El elemento [IMG](#) no tiene contenido; normalmente se reemplaza en línea por la imagen designada por el atributo [src](#), con la única excepción de las imágenes alineadas a la izquierda o a la derecha y que están ["flotando"](#) fuera de línea.

En un ejemplo anterior, definimos un vínculo a una fotografía de familia. Aquí, insertamos la foto directamente en el documento actual:

```
<BODY>
<P>¡Acabo de volver de mis vacaciones! Aquí os pongo
una foto de mi familia en el lago:
<IMG src="http://www.algunsitio.com/Gente/Juan/vacaciones/familia.png"
alt="Una foto de mi familia en el lago.">
</BODY>
```

Esta inclusión también puede realizarse con el elemento [OBJECT](#) como sigue:

```
<BODY>
<P>¡Acabo de volver de mis vacaciones! Aquí os pongo una foto
de mi familia en el lago:
<OBJECT data="http://www.algunsitio.com/Gente/Juan/vacaciones/familia.png"
type="image/png">
Una foto de mi familia en el lago.
</OBJECT>
</BODY>
```

El atributo [alt](#) especifica un texto alternativo que se representa cuando la imagen no puede ser mostrada (ver abajo para más información sobre [cómo especificar texto alternativo](#)). Los agentes de usuario deben representar el texto alternativo cuando no puedan soportar imágenes, cuando no puedan soportar un tipo de imágenes, o cuando estén configurados para no mostrar

imágenes.

El siguiente ejemplo muestra cómo puede utilizarse el atributo `longdesc` para una descripción más rica:

```
<BODY>
<P>
<IMG src="mapadelsitio.gif"
      alt="Mapa del Sitio de Laboratorios HP"
      longdesc="mapadelsitio.html">
</BODY>
```

El atributo `alt` proporciona una descripción corta de la imagen. Esta información debería ser suficiente como para que los usuarios puedan decidir si quieren seguir el vínculo dado por el atributo `longdesc` hacia la descripción más larga, en este caso "mapadelsitio.html".

Consulte la sección sobre la [presentación visual de objetos, imágenes y aplicaciones](#) para información sobre tamaño de la imagen, alineación y bordes.

13.3 Inclusión genérica: el elemento OBJECT

```
<!ELEMENT OBJECT - - (PARAM | %flow;)*
-- objeto genérico incluido -->
<!ATTLIST OBJECT
  %attrs;          -- %coreattrs, %il8n, %events --
  declare          (declare)          #IMPLIED -- declarar pero no crear --
  classid          %URI;              #IMPLIED -- identifica una implementación --
  codebase         %URI;              #IMPLIED -- URI base para classid, data, archive--
  data             %URI;              #IMPLIED -- referencia a los datos del objeto --
  type             %ContentType;      #IMPLIED -- tipo de contenido de los datos --
  codetype         %ContentType;      #IMPLIED -- tipo de contenido del código --
  archive          CDATA              #IMPLIED -- lista de URIs separados por espacios --
  standby          %Text;             #IMPLIED -- mensaje a mostrar mientras se carga --
  height          %Length;           #IMPLIED -- especificar nueva altura --
  width           %Length;           #IMPLIED -- especificar nueva anchura --
  usemap          %URI;              #IMPLIED -- usar mapa de imágenes en el cliente --
  name            CDATA              #IMPLIED -- enviar como parte de un formulario --
  tabindex        NUMBER             #IMPLIED -- posición en el orden de tabulación --
>
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

Definiciones de atributos

`classid` = [uri](#) [CT]

Este atributo puede utilizarse para especificar la localización de la implementación de un objeto mediante un URI. Puede usarse junto con, o como alternativa al atributo `data`, dependiendo del tipo de objeto involucrado.

`codebase` = [uri](#) [CT]

Este atributo especifica la ruta de acceso base (*base path*) utilizado para completar los URIs relativos especificados por los atributos `classid`, `data` y `archive`. Si está ausente, su valor por defecto es el URI base del documento actual.

`codetype` = [tipo de contenido](#) [CI]

Este atributo especifica el tipo esperado de contenido de datos cuando se carga el objeto especificado por `classid`. Este atributo es opcional pero se recomienda cuando se especifica `classid` ya que permite que el agente de usuario evite la carga de información de tipos de contenido que no soporta. Cuando está ausente, su valor por defecto es el valor del atributo `type`.

`data` = [uri](#) [CT]

Este atributo puede utilizarse para especificar la localización de los datos del objeto, por ejemplo datos de imágenes para

objetos que definen imágenes, o más en general, una forma serializada de un objeto que puede usarse para recrearlo. Si se da como URI relativo, debería interpretarse relativamente al atributo codebase.

type = [tipo de contenido](#) [CI]

Este atributo especifica el tipo de contenido de los datos especificados por [data](#). Este atributo es opcional, pero se recomienda incluirlo cuando se especifica [data](#), ya que permite que el agente de usuario no cargue información de tipos de contenido que no soporta. Si el valor de este atributo es diferente del "Content-Type" HTTP devuelto por el servidor cuando se obtiene el objeto, el "Content-Type" HTTP tiene preferencia.

archive = [lista de uris](#) [CT]

Este atributo puede utilizarse para especificar una lista de URIs *separados por espacios* de archivos que contienen recursos relevantes para el objeto, los cuales pueden incluir los recursos especificados por los atributos [classid](#) y [data](#). La precarga de archivos resultará generalmente en tiempos menores de descarga para los objetos. Los archivos especificados como URIs relativos deberían interpretarse relativamente al atributo codebase.

declare [CI]

Cuando está presente, este atributo booleano hace que la definición actual de [OBJECT](#) sea solamente una declaración. El objeto debe ser creado por una definición [OBJECT](#) subsiguiente referida a esta declaración.

standby = [texto](#) [CS]

Este atributo especifica un mensaje que puede presentar un agente de usuario mientras carga la implementación y los datos del objeto.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)
- [tabindex](#) (navegación con tabulador)
- [usemap](#) (mapas de imágenes en el lado del cliente)
- [name](#) (envío de formularios)
- [align](#), [width](#), [height](#), [border](#), [hspace](#), [vspace](#) (presentación visual de objetos, imágenes y aplicaciones)

La mayoría de los agentes de usuario tienen mecanismos integrados para representar tipos de datos comunes como texto, imágenes GIF, colores, fuentes, y un puñado de elementos gráficos. Para representar tipos de datos para los cuales no tienen soporte nativo, los agentes de usuario suelen ejecutar aplicaciones externas. El elemento [OBJECT](#) permite a los autores controlar si los datos deberían ser representados externamente por algún programa, especificado por el autor, que represente los datos dentro del agente de usuario.

En el caso más general, puede ser necesario que el autor tenga que especificar tres tipos de información:

- La implementación del objeto incluido. Por ejemplo, si el objeto incluido es una aplicación de reloj, el autor debe indicar la localización del código ejecutable de la aplicación.
- Los datos que deben representarse. Por ejemplo, si el objeto incluido es un programa que representa datos de fuentes, el autor debe indicar la localización de estos datos.
- Otros valores que necesite el objeto en tiempo de ejecución. Por ejemplo, algunas aplicaciones pueden requerir valores iniciales para los parámetros.

El elemento [OBJECT](#) permite a los autores especificar estos tres tipos de datos, pero los autores no tienen que especificar necesariamente los tres a la vez. Por ejemplo, algunos objetos pueden no necesitar datos (p.ej., una aplicación autónoma que realice una pequeña animación). Otros pueden no necesitar una inicialización en tiempo de ejecución. Y otros pueden no necesitar información adicional sobre la implementación, es decir, el propio agente de usuario puede ya saber cómo representar ese tipo de datos (p.ej., imágenes GIF).

Los autores especifican la implementación de un objeto y la localización de los datos a representar por medio del objeto [OBJECT](#). Sin embargo, para especificar valores en tiempo de ejecución, los autores usan el elemento [PARAM](#), del cual se habla en la sección sobre [inicialización de objetos](#).

El elemento [OBJECT](#) también puede aparecer en el contenido del elemento [HEAD](#). Ya que los agentes de usuario en general no representan los elementos del [HEAD](#), los autores deberían asegurarse de que los elementos [OBJECT](#) en el [HEAD](#) no especifiquen contenido que pueda ser representado. Consulte la sección sobre [compartir datos de un marco](#) para un ejemplo de inclusión del elemento [OBJECT](#) en el elemento [HEAD](#).

Consulte la sección sobre [controles de formulario](#) para información sobre elementos [OBJECT](#) en formularios.

Este documento no especifica el comportamiento de elementos [OBJECT](#) que usen tanto el atributo [classid](#) para identificar una implementación como el atributo [data](#) para especificar los datos para esa implementación. Para asegurar la portabilidad, los autores deberían usar el elemento [PARAM](#) para decir a las implementaciones dónde obtener los datos adicionales.

13.3.1 Reglas para representar objetos

Un agente de usuario debe interpretar un elemento [OBJECT](#) de acuerdo con las siguientes reglas de precedencia:

1. El agente de usuario debe intentar en primer lugar representar el objeto. No debería representar los contenidos del elemento, pero debe examinarlos por si el elemento contiene hijos directos que sean elementos [PARAM](#) (ver [inicialización de objetos](#)) o elementos [MAP](#) (ver [mapas de imágenes en el lado del cliente](#)).
2. Si por cualquier razón el agente de usuario no es capaz de representar el objeto (no está configurado para ello, no hay suficientes recursos, la arquitectura no es la adecuada, etc.), debe intentar representar sus contenidos.

Los autores no deberían incluir contenido en los elementos [OBJECT](#) que aparezcan en el elemento [HEAD](#).

En el siguiente ejemplo, insertamos una aplicación de reloj analógico en un documento por medio del elemento [OBJECT](#). La aplicación, escrita en el lenguaje Python, no necesita datos adicionales ni valores en tiempo de ejecución. El atributo [classid](#) especifica la localización de la aplicación:

```
<P><OBJECT classid="http://www.miamachina.it/relojalogico.py">
</OBJECT>
```

Obsérvese que el reloj será representado tan pronto como el agente de usuario represente esta declaración [OBJECT](#). Es posible retrasar la representación del objeto *declarando* el objeto en primer lugar (según se describe más abajo).

Los autores deberían completar esta declaración con un texto alternativo como contenido de [OBJECT](#) por si el agente de usuario no puede representar el reloj.

```
<P><OBJECT classid="http://www.miamachina.it/relojalogico.py">
Un reloj animado.
</OBJECT>
```

Una consecuencia significativa del diseño del elemento [OBJECT](#) es que ofrece un mecanismo para especificar representaciones alternativas del objeto; cada declaración [OBJECT](#) anidada puede especificar tipos de contenido alternativos. Si un agente de usuario no puede representar el primer [OBJECT](#), intenta representar los contenidos, que a su vez pueden ser otro elemento [OBJECT](#), etc.

En el siguiente ejemplo, anidamos varias declaraciones [OBJECT](#) para ilustrar el funcionamiento de las representaciones alternativas. Un agente de usuario intentará representar el primer elemento [OBJECT](#) que pueda, en el orden siguiente: (1) una aplicación LaTierra escrita en el lenguaje Python, (2) una animación MPEG de la Tierra, (3) una imagen GIF de la Tierra, (4) texto alternativo.

```
<P>                <!-- Primero, intentarlo con la aplicación Python -->
<OBJECT title="La Tierra vista desde el espacio"
  classid="http://www.observer.mars/LaTierra.py">
  <!-- Si no, intentarlo con el vídeo MPEG -->
<OBJECT data="LaTierra.mpeg" type="application/mpeg">
  <!-- Si no, intentarlo con la imagen GIF -->
  <OBJECT data="LaTierra.gif" type="image/gif">
  <!-- Si no, representar el texto -->
```

```

    <STRONG>La Tierra</STRONG> vista desde el espacio.
  </OBJECT>
</OBJECT>
</OBJECT>

```

La declaración más externa especifica una aplicación que no necesita datos ni valores iniciales. La segunda declaración especifica una animación MPEG, y como no define la localización de ninguna implementación que procese MPEG, depende de que el agente de usuario represente la animación. También establecemos el atributo `type` para que los agentes de usuario que sepan que no pueden procesar MPEG no se molesten en obtener "LaTierra.mpeg" de la red. La tercera declaración especifica la localización de un fichero GIF y proporciona un texto alternativo por si todos los mecanismos anteriores fallaran.

Datos en línea frente a datos externos. Los datos a representar pueden proporcionarse de dos maneras: en línea y desde un recurso externo. Si bien el primer método resultará generalmente en una representación más rápida, no es conveniente cuando se representan grandes cantidades de datos.

Aquí tenemos un ejemplo que ilustra cómo se puede servir a un elemento [OBJECT](#) datos en línea:

```

<P>
<OBJECT id="reloj1"
  classid="clsid:663C8FEF-1EF9-11CF-A3DB-080036F12502"
  data="data:application/x-oleobject;base64, ...datos en base64...">
  Un reloj.
</OBJECT>

```

Consulte la sección sobre la [representación visual de objetos, imágenes y aplicaciones](#) para información sobre tamaño de objetos, alineación y bordes.

13.3.2 Inicialización de objetos: el elemento PARAM

```

<!ELEMENT PARAM - O EMPTY -- valor de propiedad con nombre -->
<!ATTLIST PARAM
  id          ID          #IMPLIED -- identificador único a nivel de documento --
  name        CDATA       #REQUIRED -- nombre de la propiedad --
  value       CDATA       #IMPLIED -- valor de la propiedad --
  valuetype   (DATA|REF|OBJECT) DATA -- cómo interpretar el valor --
  type        %ContentType; #IMPLIED -- tipo de contenido para el valor
                                     cuando valuetype=ref --
>

```

Etiqueta inicial: **obligatoria**, Etiqueta final: **prohibida**

Definiciones de atributos

`name` = [CDATA](#)

Este atributo define el nombre de un parámetro de ejecución, que se supone que el objeto insertado conoce.

Dependiendo de la implementación específica del objeto, se distinguirá o no entre mayúsculas y minúsculas.

`value` = [CDATA](#)

Este atributo especifica el valor del parámetro de ejecución especificado por [name](#). Los valores de las propiedades no tienen significado para HTML; su significado lo determina el objeto en cuestión.

`valuetype` = data|ref|object [\[CI\]](#)

Este atributo especifica el tipo de atributo value. Valores posibles:

- data: Este es el valor por defecto del atributo. Significa que el valor especificado por value se evaluará y pasará a la implementación del objeto como una cadena.
- ref: El valor especificado por value es un URI que designa un recurso donde se almacenan valores de tiempo de ejecución. Esto permite a las herramientas del soporte identificar los URIs dados como parámetros. El URI debe pasarse al objeto **tal y como está**, es decir, sin completar.
- object: El valor especificado por value es un identificador que se refiere a una declaración [OBJECT](#) del mismo documento. El identificador debe ser el valor del atributo [id](#) establecido para el objeto [OBJECT](#) declarado.

type = [tipo de contenido \[CI\]](#)

Este atributo especifica el tipo de contenido del recurso contenido por el atributo value **sólo** en el caso en que [valuetype](#) sea igual a "ref". Este atributo especifica así para el agente usuario el tipo de valores que encontrará en el URI designado por value.

Atributos definidos en otros lugares

- [id \(identificadores a nivel de documento\)](#)

Los elementos [PARAM](#) especifican un conjunto de valores que pueden ser necesarios para un objeto en tiempo de ejecución. Puede aparecer cualquier número de elementos [PARAM](#) en el contenido de un elemento [OBJECT](#) o [APPLET](#), y en cualquier orden, pero deben ser colocados al principio del contenido del elemento [OBJECT](#) o [APPLET](#) que los contienen.

Se supone que la sintaxis de nombres y valores es entendida por la implementación del objeto. Este documento no especifica cómo deberían obtener los agentes de usuario los pares nombre/valor ni cómo deberían interpretar los nombres de parámetros que aparezcan dos veces.

Volvemos al ejemplo del reloj para ilustrar la utilización de [PARAM](#): supongamos que la aplicación es capaz de manejar dos parámetros de ejecución que definen su altura y anchura iniciales. Podemos establecer las dimensiones iniciales en 40x40 píxeles con dos elementos [PARAM](#).

```
<P><OBJECT classid="http://www.miamachina.it/reloj analogico.py">
<PARAM name="altura" value="40" valuetype="data">
<PARAM name="anchura" value="40" valuetype="data">
Este agente de usuario no puede representar aplicaciones Python.
</OBJECT>
```

En el siguiente ejemplo, se especifican los datos de ejecución para el parámetro "Valores_inic" del objeto como recurso externo (un fichero GIF). El valor del atributo [valuetype](#) se establece por tanto a "ref" y value es un URI que designa el recurso.

```
<P><OBJECT classid="http://www.gifstuff.com/aplicgif"
standby="Cargando Elvis...">
<PARAM name="Valores_inic"
value="./imagenes/elvis.gif">
valuetype="ref">
</OBJECT>
```

Obsérvese que también hemos establecido el atributo [standby](#) de modo que el agente de usuario pueda mostrar un mensaje mientras se carga el mecanismo de representación.

Cuando representen un elemento [OBJECT](#), los agentes de usuario sólo deben buscar el contenido de aquellos elementos [PARAM](#) que sean hijos directos y dárselos al [OBJECT](#).

Así, en el ejemplo siguiente, si se representa "obj1", "param1" se aplica a "obj1" (y no a "obj2"). Si "obj1" no se representa y "obj2" sí, "param1" no se tiene en cuenta, y "param2" se aplica a "obj2". Si ninguno de los [OBJECT](#) se representa, ninguno de los [PARAM](#) se aplica.

```
<P>
<OBJECT id="obj1">
<PARAM name="param1">
<OBJECT id="obj2">
<PARAM name="param2">
</OBJECT>
</OBJECT>
```

13.3.3 Esquemas globales de nombres para objetos

La localización de la implementación de un objeto viene dada por un URI. Como vimos en la [introducción a los URIs](#), el primer segmento de un URI absoluto especifica el esquema de nombres usado para transferir los datos designados por el URI.

Para documentos HTML, este esquema es normalmente "http". Algunas aplicaciones podrían usar otros esquemas de nombres. Por ejemplo, cuando se especifica una aplicación Java, los autores pueden usar URIs que comiencen con java, y para aplicaciones ActiveX, los autores pueden usar "clsid".

En el ejemplo siguiente, insertamos una aplicación Java en un documento HTML.

```
<P><OBJECT classid="java:program.start">
</OBJECT>
```

Al establecer el atributo [codetype](#), un agente de usuario puede decidir si obtener la aplicación Java según su capacidad de hacerlo.

```
<OBJECT codetype="application/java-archive"
classid="java:program.start">
</OBJECT>
```

Algunos esquemas de representación necesitan información adicional para identificar su implementación, y hay que decirles dónde encontrar esa información. Los autores pueden dar la información de la ruta de acceso a la implementación del objeto a través del atributo [codebase](#).

```
<OBJECT codetype="application/java-archive"
classid="java:program.start">
codebase="http://bla.bla.com/java/miimplementacion/"
</OBJECT>
```

El siguiente ejemplo especifica (con el atributo [classid](#)) un objeto ActiveX a través de un URI que comienza con el esquema de nombres "clsid". El atributo [data](#) localiza los datos a representar (otro reloj).

```
<P><OBJECT classid="clsid:663C8FEF-1EF9-11CF-A3DB-080036F12502"
data="http://www.acme.com/ole/reloj.stm">
Esta aplicación no está soportada.
</OBJECT>
```

13.3.4 Declaración y creación de objetos

Los objetos precedentes sólo ilustran definiciones de objetos aislados. Cuando un documento debe contener más de un ejemplar del mismo objeto, es posible separar la declaración de un objeto de su creación o creaciones. Esto tiene varias ventajas:

- El agente de usuario puede obtener los datos de la red *una vez* (durante la declaración) y reutilizarlos para cada ejemplar.
- Es posible crear un objeto desde una localización diferente a la de la declaración del objeto, por ejemplo, desde un vínculo.
- Es posible especificar objetos como datos de ejecución de otros objetos.

Para declarar un objeto de modo que no sea ejecutado cuando lo lea el agente de usuario, se establece el atributo booleano [declare](#) del elemento [OBJECT](#). Al mismo tiempo, los autores deben identificar la declaración estableciendo el atributo [id](#) del elemento [OBJECT](#) en un valor único. Las creaciones posteriores del objeto se referirán a este identificador.

Un [OBJECT](#) declarado debe aparecer en un documento antes de la primera creación de ese [OBJECT](#).

Se crea un objeto definido con el atributo [declare](#) cada vez que un elemento que se refiera a ese objeto necesite que sea representado (p.ej., se activa un vínculo que se refiere a él, se activa un objeto que se refiere a él, etc.).

En el siguiente ejemplo, declaramos un [OBJECT](#) y hacemos que se cree haciendo referencia a él desde un vínculo. Así, el objeto puede activarse haciendo clic en un texto resaltado, por ejemplo.

```
<P><OBJECT declare
id="tierra.declaracion"
data="LaTierra.mpeg"
type="application/mpeg">
<STRONG>La Tierra</STRONG> vista desde el espacio.
```

```
</OBJECT>
```

...más abajo en el documento...

```
<P>¡Una preciosa <A href="#tierra.declaracion"> animación de la Tierra!</A>
```

Otra manera de crear un objeto declarado de esta forma es:

```
<OBJECT data="#tierra.declaracion"></OBJECT>
```

El siguiente ejemplo ilustra cómo especificar valores de ejecución que son otros objetos. En este ejemplo, enviamos texto (concretamente un poema) a un hipotético mecanismo para ver poemas. El objeto reconoce un parámetro llamado "fuente" (por ejemplo para representar el poema con una cierta fuente). El valor de este parámetro es a su vez un objeto que inserta (pero que no representa) el objeto fuente. La relación entre el objeto fuente y el objeto visor de poemas se consigue (1) asignando el [id](#) "tribuna" a la declaración del objeto y (2) refiriéndose a él desde el elemento [PARAM](#) del objeto visor de poemas (con [valuetype](#) y [value](#)).

```
<P><OBJECT declare
  id="tribuna"
  type="application/x-webfont"
  data="tribuna.gif">
</OBJECT>
...ver el problema de KublaKhan.txt aquí...
<P><OBJECT classid="http://bla.bla.com/visordepoemas"
  data="KublaKhan.txt">
<PARAM name="fuente" valuetype="object" value="#tribuna">
<P>Te estás perdiendo un visor de poemas realmente estupendo ...
</OBJECT>
```

Los agentes de usuario que no soporten el atributo [declare](#) deben representar los contenidos de la declaración [OBJECT](#).

13.4 Incluir un applet: el elemento APPLET

APPLET está [desaprobado \(con todos sus atributos\)](#) en favor de [OBJECT](#).

Ver el [DTD Transicional](#) para la definición formal.

Definiciones de atributos

codebase = [uri](#) [CT]

Este atributo especifica el URI base para el applet. Si este atributo no se especifica, entonces su valor por defecto es el mismo URI base que el del documento actual. Los valores de este atributo sólo pueden referirse a subdirectorios del directorio que contiene al documento actual. **Nota.** *Si bien la restricción a subdirectorios es una excepción a la práctica habitual y a la especificación HTML 3.2, el Grupo de Trabajo HTML ha decidido mantener esta restricción en esta versión de la especificación por motivos de seguridad.*

code = [cdata](#) [CS]

Este atributo especifica o bien el nombre del fichero class que contiene la subclase Applet compilada del applet, o bien la ruta para obtener la clase, incluyendo al propio fichero class. Se interpreta con respecto al URI base (codebase) del applet. Deben estar presentes o bien [code](#) o bien [object](#).

name = [cdata](#) [CS]

Este atributo especifica un nombre para la instancia del applet, lo cual hace posible que los applets de la misma página se encuentren (y se comuniquen) entre sí.

archive = [lista de uris](#) [CT]

Esta lista especifica una lista de URIs *separados por comas* de archivos que contienen clases y otros recursos que serán "precargados". Las clases se cargan usando una instancia de un AppletClassLoader con el [codebase](#) dado. Los URIs relativos se interpretan con respecto al URI base (codebase) del applet. La precarga de recursos puede mejorar significativamente el rendimiento de los applets.

object = [cdata](#) [CS]

Este atributo indica el nombre de un recurso que contiene una representación serializada del estado de un applet. Se interpreta relativamente al URI base (codebase) del applet. Los datos serializados contienen el nombre de la clase del

applet, pero no la implementación. El nombre de la clase se utiliza para obtener la implementación en un archivo o fichero class.

Cuando el applet es "deserializado", se invoca el método start(), pero no el método init(). Los atributos que eran válidos cuando el objeto original fue serializado **no** son restaurados. Cualquier atributo que se pase a esta instancia [APPLET](#) estará a disposición del applet. Los autores deberían usar esta característica con extrema cautela. Un applet debería ser detenido antes de ser serializado.

Deben estar presentes o bien [code](#) o bien [object](#). Si se dan tanto [code](#) como [object](#) y proporcionan nombres de clases diferentes, se produce un error.

width = [longitud \[CI\]](#)

Este atributo especifica la anchura inicial del área de presentación del applet (sin incluir las ventanas o cuadros de diálogo que cree el applet).

height = [longitud \[CI\]](#)

Este atributo especifica la altura inicial del área de presentación del applet (sin incluir las ventanas o cuadros de diálogo que cree el applet).

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [alt](#) (texto alternativo)
- [align](#), [hspace](#), [vspace](#) (presentación visual de objetos, imágenes y aplicaciones)

Este elemento, admitido por todos los browsers con soporte Java, permite a los diseñadores incluir un applet Java en un documento HTML. Ha sido [desaprobado](#) en favor del elemento [OBJECT](#).

El contenido de [APPLET](#) actúa como información alternativa para agentes de usuario que no soporten este elemento o que estén configurados para no soportar applets. En cualquier otro caso los agentes de usuario deben ignorar el contenido.

EJEMPLO DESAPROBADO:

En el siguiente ejemplo, el elemento [APPLET](#) incluye un applet Java en el documento. Al no haberse suministrado un [codebase](#), se supone que el applet está en el mismo directorio que el documento actual.

```
<APPLET code="Burbujas.class" width="500" height="500">
Applet Java que dibuja burbujas animadas.
</APPLET>
```

Este ejemplo puede reformularse con [OBJECT](#) de esta forma:

```
<P><OBJECT codetype="application/java"
  classid="java:Burbujas.class"
  width="500" height="500">
Applet Java que dibuja burbujas animadas.
</OBJECT>
```

Se pueden proporcionar valores iniciales al applet por medio del elemento [PARAM](#).

EJEMPLO DESAPROBADO:

El siguiente ejemplo de applet Java:

```
<APPLET code="AudioItem" width="15" height="15">
<PARAM name="snd" value="Hola.au|Bienvenido.au">
Applet Java que reproduce un sonido de bienvenida.
</APPLET>
```

puede reformularse de la siguiente manera con [OBJECT](#):

```
<OBJECT codetype="application/java "
```

```

classid="AudioItem"
width="15" height="15">
<PARAM name="snd" value="Hola.au|Bienvenido.au">
Applet Java que reproduce un sonido de bienvenida.
</OBJECT>

```

13.5 Notas sobre la inclusión de documentos

Algunas veces, más que crear un [vínculo](#) a un documento, lo que quiere un autor es incluirlo directamente en un documento HTML principal. Para esto, los autores pueden usar o bien el elemento [IFRAME](#) o bien el elemento [OBJECT](#), pero ambos elementos difieren en varios aspectos. No solamente tienen los dos elementos modelos de contenido diferentes, sino que el elemento [IFRAME](#) puede ser un marco destino (véase la sección sobre la [especificación de información sobre el marco destino](#) para más detalles) y puede ser "seleccionado" por un agente de usuario dirigiendo a él el foco para imprimir, ver el fuente HTML, etc. Los agentes de usuario pueden representar los marcos seleccionados de modo que se distingan de los marcos no seleccionados (p.ej., dibujando un borde alrededor del marco seleccionado).

Un documento incluido es completamente independiente del documento en el cual se incluye. Por ejemplo, los URIs relativos del documento incluido [se completan](#) de acuerdo con el URI base del documento incluido, no con el del documento principal. Un documento incluido sólo se representa dentro de otro documento (p.ej., en una subventana); por lo demás sigue siendo independiente.

Por ejemplo, la línea siguiente incluye los contenidos de `incluye_me.html` en el lugar en que aparece la definición [OBJECT](#).

```

...texto antes...
<OBJECT data="incluye_me.html">
Atención: no se pudo incluir incluye_me.html.
</OBJECT>
...texto después...

```

Recuérdese que los contenidos de [OBJECT](#) sólo deben ser representados si no se puede cargar el fichero especificado por el atributo [data](#).

El comportamiento de un agente de usuario en los casos en que un fichero se incluye a sí mismo queda sin definir.

13.6 Mapas de imágenes

Los mapas de imágenes permiten a los autores especificar regiones en una imagen u objeto y asignar una acción específica a cada región (p.ej., abrir un documento, ejecutar un programa, etc.). Cuando la región es activada por el usuario, se ejecuta la acción.

Un mapa de imágenes se crea asociando un objeto con una especificación de las áreas geométricas sensibles del objeto.

Hay dos tipos de mapas de imágenes:

- *En el lado del cliente.* Cuando un usuario activa una región de un mapa de imágenes en el lado del cliente con un ratón, las coordenadas en píxeles son interpretadas por el agente de usuario. El agente de usuario selecciona el vínculo especificado por la región activada y lo sigue.
- *En el lado del servidor.* Cuando un usuario activa una región de un mapa de imágenes en el lado del servidor, las coordenadas en píxeles son enviadas al agente del lado del servidor especificado por el atributo [href](#) del elemento [A](#). El agente del servidor interpreta las coordenadas y realiza alguna acción.

Se prefieren los mapas de imágenes en el cliente que los mapas de imágenes en el servidor por dos razones: son accesibles a las personas que utilizan agentes de usuario no gráficos y permiten saber en todo momento si el apuntador está sobre una región activa o no.

13.6.1 Mapas de imágenes en el lado del cliente: los elementos MAP y AREA

```

<!ELEMENT MAP - - ((%block;) | AREA)+ -- mapa de imágenes en el lado del cliente -->
<!ATTLIST MAP

```

```

%attrs; -- %coreattrs, %i18n, %events --
name CDATA #REQUIRED -- como referencia para usemap --
>

```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

```

<!ELEMENT AREA - O EMPTY -- área de un mapa de imágenes en el
cliente-->
<!ATTLIST AREA
%attrs; -- %coreattrs, %i18n, %events --
shape %Shape; rect -- controla la interpretación de las coords --
coords %Coords; #IMPLIED -- lista de longitudes separadas por comas --
href %URI; #IMPLIED -- URI del recurso vinculado --
nohref (nohref) #IMPLIED -- esta región no tiene acción --
alt %Text; #REQUIRED -- descripción corta --
tabindex NUMBER #IMPLIED -- posición en el orden de tabulación --
accesskey %Character; #IMPLIED -- carácter de la tecla de accesibilidad --
onfocus %Script; #IMPLIED -- el foco se dirigió hacia el elemento --
onblur %Script; #IMPLIED -- el elemento perdió el foco --
>

```

Etiqueta inicial: **obligatoria**, Etiqueta final: **prohibida**

Definiciones de atributos de MAP

name = [CDATA \[CI\]](#)

Este atributo asigna un nombre al mapa de imágenes definido por un elemento [MAP](#).

Definiciones de atributos de AREA

shape = default|rect|circle|poly [\[CI\]](#)

Este tributo especifica la forma de una región. Valores posibles:

- default: Especifica la región completa.
- rect: Define una región rectangular.
- circle: Define una región circular.
- poly: Define una región poligonal.

coords = *coordenadas* [\[CN\]](#)

Este atributo especifica la posición y la forma en la pantalla. El número y orden de estos valores depende de la forma que está siendo definida. Combinaciones posibles:

- rect: x-izquierda, y-superior, x-derecha, y-inferior.
- circle: x-centro, y-centro, radio. **Nota.** Cuando el valor del radio sea un valor porcentual, los agentes de usuario deberían calcular el valor final del radio basándose en la anchura y altura del objeto asociado. El radio debería ser el menor valor de los dos.
- poly: x1, y1, x2, y2, ..., xN, yN. El primer par de coordenadas x, y debería ser igual al último para cerrar el polígono. Cuando estas coordenadas no sean iguales, los agentes de usuario deberían añadir un par adicional de coordenadas para cerrar el polígono.

Las coordenadas son relativas a la esquina superior izquierda del objeto. Todos los valores son [longitudes](#). Todos los valores van separados por comas.

N.T.: [La especificación no aclara](#) si las coordenadas x,y se miden respecto al cero o al uno, es decir, si las coordenadas de la esquina superior izquierda son (0,0) o (1,1).

nohref [\[CI\]](#)

Si está establecido, este atributo booleano especifica que una región no tiene asociado ningún vínculo.

Atributo para asociar un mapa de imágenes con un elemento

usemap = [uri \[CT\]](#)

Este atributo asocia un mapa de imágenes con un elemento. El mapa de imágenes está definido por un elemento [MAP](#).

El valor de `usemap` debe ser igual al valor del atributo `name` del elemento `MAP` asociado.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [name](#) (envío de objetos con formularios)
- [alt](#) (texto alternativo)
- [href](#) (referencias de vínculos) [target](#) (información sobre el marco destino)
- [tabindex](#) (navegación con tabulador)
- [accesskey](#) (teclas de acceso)
- [shape](#) (mapas de imágenes)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#), [onfocus](#), [onblur](#) (eventos intrínsecos)

El elemento `MAP` especifica un mapa de imágenes en el lado del cliente (u otro mecanismo de navegación) que puede ser asociado con otros elementos (`IMG`, `OBJECT`, o `INPUT`). Un mapa de imágenes se asocia a un elemento a través del atributo `usemap` del elemento. El elemento `MAP` puede ser utilizado sin una imagen asociada por mecanismos generales de navegación.

La presencia del atributo `usemap` en un elemento `OBJECT` implica que el objeto que incluye es una imagen. Además, cuando el elemento `OBJECT` tenga asociado un mapa de imágenes en el lado del cliente, los agentes de usuario pueden permitir la interacción del usuario con el elemento `OBJECT` sólo en lo que se refiere al mapa de imágenes en el lado del cliente. Esto permite a los agentes de usuario (tales como un navegador de audio o un robot) interactuar con el `OBJECT` sin tener que procesarlo; el agente de usuario puede incluso optar por no obtener (o procesar) el objeto. Cuando un `OBJECT` tenga asociado un mapa de imágenes, los autores no deberían esperar que el objeto sea obtenido o procesado por todos los agentes de usuario.

El modelo de contenido del elemento `MAP` permite a los autores lo siguiente:

1. Uno o más elementos `AREA`. Estos elementos no tienen contenido, sino que especifican las regiones geométricas del mapa de imágenes y los vínculos asociados con cada región. Obsérvese que en general los agentes de usuario no representan los elementos `AREA`. Por tanto, los autores deben proporcionar texto alternativo para cada `AREA` con el atributo `alt` (ver abajo más información sobre [cómo especificar texto alternativo](#)).
2. Contenido en bloque. Este contenido debería incluir elementos `A` que especifiquen regiones geométricas del mapa de imágenes y el vínculo asociado con cada región. Obsérvese que el agente de usuario debería representar el contenido en bloque del elemento `MAP`. Los autores deberían usar este método para crear documentos más accesibles.

Cuando un elemento `MAP` contenga contenido mixto (tanto elementos `AREA` como contenido en bloque), los agentes de usuario deberían ignorar los elementos `AREA`.

Los autores deberían especificar la geometría de un mapa de imágenes completamente con elementos `AREA`, o completamente con elementos `A`, o completamente con ambos si el contenido es mixto. Los autores pueden querer usar contenido mixto para que los agentes de usuario antiguos utilicen la geometría del mapa especificada por los elementos `AREA` y que los agentes de usuario modernos saquen partido de la riqueza del contenido en bloque.

Si dos o más regiones se superponen, tiene prioridad la región definida por el elemento que aparece antes en el documento.

Los agentes de usuario y los autores deberían ofrecer alternativas textuales a los mapas de imágenes gráficos para los casos en que los gráficos no estén disponibles o en que el usuario no pueda acceder a ellos. Por ejemplo, los agentes de usuario pueden usar el texto `alt` para crear vínculos textuales en lugar de un mapa de imágenes gráfico. Estos vínculos pueden ser activados de diferentes maneras (con el teclado, activación por voz, etc.).

Nota. `MAP` no es compatible con los agentes de usuario HTML 2.0.

Ejemplos de mapas de imágenes en el lado del cliente

En el ejemplo siguiente, creamos un mapa de imágenes en el lado del cliente para el elemento [OBJECT](#). No queremos representar los contenidos del mapa de imágenes cuando se represente el [OBJECT](#), así que "ocultamos" el elemento [MAP](#) dentro del contenido del elemento [OBJECT](#). En consecuencia, los contenidos del elemento [MAP](#) sólo serán representados si el [OBJECT](#) no puede ser representado.

```
<HTML>
  <HEAD>
    <TITLE>;El sitio chévere!</TITLE>
  </HEAD>
  <BODY>
    <P><OBJECT data="barranavl.gif" type="image/gif" usemap="#map1">
    <MAP name="map1">
      <P>Navegar por este sitio:
      <A href="guia.html" shape="rect" coords="0,0,118,28">Acceder a la Guía</a> |
      <A href="atajo.html" shape="rect" coords="118,0,184,28">Ir</A> |
      <A href="buscar.html" shape="circle" coords="184,200,60">Buscar</A> |
      <A href="top10.html" shape="poly"
        coords="276,0,276,28,100,200,50,50,276,0">Top Ten</A>
    </MAP>
    </OBJECT>
  </BODY>
</HTML>
```

Podemos querer representar los contenidos del mapa de imágenes incluso si el agente de usuario puede representar el [OBJECT](#). Por ejemplo, podemos querer asociar un mapa de imágenes con un elemento [OBJECT](#) e incluir una barra de navegación textual en la parte inferior de la página. Para ello, definimos el elemento [MAP](#) fuera del [OBJECT](#):

```
<HTML>
  <HEAD>
    <TITLE>;El sitio chévere!</TITLE>
  </HEAD>
  <BODY>
    <P><OBJECT data="barranavl.gif" type="image/gif" usemap="#map1">
    </OBJECT>

    ...aquí el resto de la página...

    <MAP name="map1">
      <P>Navegar por este sitio:
      <A href="guia.html" shape="rect" coords="0,0,118,28">Acceder a la Guía</a> |
      <A href="atajo.html" shape="rect" coords="118,0,184,28">Ir</A> |
      <A href="busca.html" shape="circle" coords="184,200,60">Buscar</A> |
      <A href="top10.html" shape="poly"
        coords="276,0,276,28,100,200,50,50,276,0">Top Ten</A>
    </MAP>
  </BODY>
</HTML>
```

En el siguiente ejemplo, creamos un mapa de imágenes similar, esta vez utilizando el elemento [AREA](#). Obsérvese el uso del texto [alt](#):

```
<P><OBJECT data="barranavl.gif" type="image/gif" usemap="#map1">
  <P>Esto es una barra de navegación.
</OBJECT>

<MAP name="map1">
  <AREA href="guia.html"
    alt="Acceder a la Guía"
    shape="rect"
```

```

    coords="0,0,118,28">
<AREA href="buscar.html"
    alt="Buscar"
    shape="rect"
    coords="184,0,276,28">
<AREA href="atajo.html"
    alt="Ir"
    shape="circle"
    coords="184,200,60">
<AREA href="top10.html"
    alt="Top Ten"
    shape="poly"
    coords="276,0,276,28,100,200,50,50,276,0">
</MAP>

```

Aquí tenemos una versión similar usando el elemento [IMG](#) en lugar de [OBJECT](#) (con la misma declaración [MAP](#)):

```
<P><IMG src="barranavl.gif" usemap="#map1" alt="barra de navegación">
```

El siguiente ejemplo ilustra cómo pueden compartirse los mapas de imágenes.

Los elementos [OBJECT](#) anidados son útiles para proporcionar "redes de seguridad" en caso de que un agente de usuario no soporte ciertos formatos. Por ejemplo:

```

<P>
<OBJECT data="barranav.png" type="image/png">
  <OBJECT data="barranav.gif" type="image/gif">
    texto que describe la imagen...
  </OBJECT>
</OBJECT>

```

Si el agente de usuario no soporta el formato PNG, intenta representar la imagen GIF. Si no soporta GIF (es decir, es un agente de usuario por voz), utiliza la descripción textual proporcionada como contenido del elemento [OBJECT](#) interior. Cuando los elementos [OBJECT](#) se anidan de esta manera, los autores pueden compartir mapas de imágenes entre ellos:

```

<P>
<OBJECT data="barranav.png" type="image/png" usemap="#map1">
  <OBJECT data="barranav.gif" type="image/gif" usemap="#map1">
    <MAP name="map1">
      <P>Navegar por este sitio:
      <A href="guia.html" shape="rect" coords="0,0,118,28">Acceder a la Guía</A> |
      <A href="atajo.html" shape="rect" coords="118,0,184,28">Ir</A> |
      <A href="buscar.html" shape="circle" coords="184,200,60">Buscar</A> |
      <A href="top10.html" shape="poly"
        coords="276,0,276,28,100,200,50,50,276,0">Top Ten</A>
    </MAP>
  </OBJECT>
</OBJECT>

```

El siguiente ejemplo ilustra cómo pueden especificarse elementos [A](#) para crear zonas inactivas dentro de un mapa de imágenes. El primer vínculo especifica una pequeña región circular sin vínculo asociado. El segundo vínculo especifica una región circular más grande con el mismo centro. La combinación de ambos define un anillo cuyo centro es inactivo y cuya corona es activa. El orden de las definiciones de los vínculos es importante, ya que el círculo menor debe prevalecer sobre el círculo mayor.

```

<MAP name="map1">
<P>
<A shape="circle" coords="100,200,50">Yo soy inactivo.</A>
<A href="vinculo-circulo-externo.html" shape="circle"
  coords="100,200,250">Yo soy activo.</A>
</MAP>

```

Análogamente, el atributo [nohref](#) del elemento [AREA](#) declara que una región geométrica no tiene un vínculo asociado.

13.6.2 Mapas de imágenes en el lado del servidor

Los mapas de imágenes en el lado del servidor puede ser interesantes en aquellos casos en que el mapa de imágenes sea demasiado complicado para un mapa de imágenes en el lado del cliente.

Sólo es posible definir un mapa de imágenes el lado del servidor para los elementos [IMG](#) e [INPUT](#). En el caso de [IMG](#), el [IMG](#) debe estar dentro de un elemento [A](#) y debe establecerse el atributo booleano [ismap](#) ([ICI](#)). En el caso de [INPUT](#), el [INPUT](#) debe ser del tipo "image".

Cuando el usuario activa el vínculo haciendo clic sobre la imagen, las coordenadas de pantalla se envían directamente al servidor donde se aloja el documento. Las coordenadas de pantalla se expresan como píxeles de pantalla relativos a la imagen. Para información normativa sobre la definición de un píxel y cómo escalarlo, consulte [\[CSS1\]](#).

En el siguiente ejemplo, la región activa define un vínculo en el lado del servidor. Así, un clic en cualquier parte de la imagen hará que las coordenadas del clic sean enviadas al servidor.

```
<P><A href="http://www.acme.com/cgi-bin/competicion">
  <IMG src="juego.gif" ismap alt="diana"></A>
```

El lugar del clic se pasa al servidor como sigue. El agente de usuario crea un nuevo URI a partir del URI especificado por el atributo [href](#) del elemento [A](#), añadiendo un "?" seguido de las coordenadas x e y, separadas por una coma. A continuación se sigue el vínculo especificado por el nuevo URI. Por ejemplo, en el ejemplo dado, si el usuario hace clic en x=10, y=27, entonces el URI creado es "http://www.acme.com/cgi-bin/competicion?10,27".

Los agentes de usuario que no ofrezcan al usuario medios para especificar unas coordenadas específicas (p.ej., agentes de usuario no gráficos que reciban la entrada por teclado, agentes de usuario por voz, etc.) deberían enviar las coordenadas "0,0" al servidor cuando se activa el link.

13.7 Presentación visual de imágenes, objetos y aplicaciones

Todos los atributos de los objetos [IMG](#) y [OBJECT](#) que guarden relación con la alineación y la presentación han sido [desaprobados](#) en favor de las hojas de estilo.

13.7.1 Anchura y altura

Definiciones de atributos

`width = longitud [CN]`

Invalidar anchura de imagen u objeto.

`height = longitud [CN]`

Invalidar altura de imagen u objeto.

Si se especifican, los atributos [width](#) y [height](#) dicen a los agentes de usuario que invaliden el tamaño original de la imagen u objeto en favor de estos valores.

Cuando el objeto es una imagen, se escala. Los agentes de usuario deberían hacer lo posible para escalar el objeto o imagen de modo que se respeten la anchura y altura especificadas por el autor. Obsérvese que las longitudes expresadas como porcentajes se basan el espacio horizontal o vertical disponible actualmente, no en el tamaño original de la imagen, objeto o aplicación.

Los atributos [height](#) y [width](#) dan a los agentes de usuario una idea del tamaño de una imagen u objeto para que puedan reservar espacio para ellos y continuar la representación del documento mientras esperan a los datos de la imagen.

13.7.2 Espacio en blanco alrededor de imágenes y objetos

Definiciones de atributos

`hspace = píxeles [CN]`

[Desaprobado.](#) Este atributo especifica la cantidad de espacio en blanco a insertar a la izquierda y a la derecha de un [IMG](#), [APPLET](#), u [OBJECT](#). El valor por defecto no está especificado, pero generalmente es una longitud pequeña

distinta de cero.

`vspace = píxeles \[CN\]`

Desaprobado. Este atributo especifica la cantidad de espacio en blanco a insertar encima y debajo de un [IMG](#), [APPLET](#), u [OBJECT](#). El valor por defecto no está especificado, pero generalmente es una longitud pequeña distinta de cero.

13.7.3 Bordes

Una imagen u objeto puede estar rodeada de un borde (p.ej., cuando se especifica un borde por el usuario o cuando la imagen es el contenido de un elemento [A](#)).

Definiciones de atributos

`border = píxeles \[CN\]`

Desaprobado. Este atributo especifica la anchura del borde de un [IMG](#) u [OBJECT](#), en píxeles. El valor por defecto de este atributo depende del agente de usuario.

13.7.4 Alineación

Definiciones de atributos

`align = bottom|middle|top|left|right`

Desaprobado. Este atributo especifica la posición de un [IMG](#), [OBJECT](#), o [APPLET](#) con respecto a su contexto.

Los siguientes valores de [align](#) se refieren a la posición del objeto con respecto al texto que le rodea:

- **bottom:** significa que la parte inferior del objeto debería estar alineada verticalmente con la línea de base actual. Este es el valor por defecto.
- **middle:** significa que el centro del objeto debería estar alineado verticalmente con la línea de base actual.
- **top:** significa que la parte superior del objeto debería estar alineada verticalmente con la parte superior de la línea de texto actual.

Hay otros dos valores, `left` y `right`, que hacen que la imagen "flote" hacia el margen izquierdo o derecho actual. Se habla sobre ellos en la sección sobre [objetos flotantes](#).

Diferentes interpretaciones de align. Los agentes de usuario pueden interpretar de manera diferente el atributo [align](#).

Algunos sólo tienen en cuenta lo que haya ocurrido en la línea antes del elemento, algunos tienen en cuenta el texto que hay a ambos lados del elemento.

13.8 Cómo especificar texto alternativo

Definiciones de atributos

`alt = texto \[CS\]`

Este atributo especifica texto alternativo para agentes de usuario que no puedan mostrar imágenes, formularios o aplicaciones. El idioma de este texto alternativo está especificado por el atributo [lang](#).

Varios elementos no textuales ([IMG](#), [AREA](#), [APPLET](#) e [INPUT](#)) permiten a los autores especificar texto alternativo que sirva como contenido cuando el elemento no pueda ser representado normalmente. El especificar texto alternativo ayuda a los usuarios que no tengan terminales gráficas, a los usuarios cuyos navegadores no soporten formularios, a los usuarios con discapacidades visuales, a aquellos que utilicen sintetizadores de voz, a aquellos que hayan configurado sus agentes de usuario para no mostrar imágenes, etc.

El atributo [alt](#) debe especificarse para los elementos [IMG](#) y [AREA](#). Es opcional para los elementos [INPUT](#) y [APPLET](#).

Si bien el texto alternativo puede ser muy útil, hay que tratarlo con cuidado. Los autores deberían seguir las siguientes pautas:

- No especificar texto alternativo irrelevante cuando las imágenes incluidas sólo sirven para dar *formato* a una página, por ejemplo, `alt="bola roja"` sería inapropiado para una imagen que añade una bola roja para decorar un título o un párrafo. En tales casos, el texto alternativo debería ser la cadena vacía (`""`). En cualquier caso se aconseja a los autores que eviten usar imágenes para dar formato a las páginas, y que utilicen hojas de estilo en su lugar.
- No especificar texto alternativo sin significado (p.ej., "relleno"). Esto no solamente frustrará a los usuarios, sino que

ralentizará a los agentes de usuario que deban convertir texto a salida por voz o Braille.

Los implementadores deberían consultar la sección sobre [accessibilidad](#) para información sobre cómo tratar los casos en que se omite el texto alternativo.

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

14 Hojas de estilo

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Introducción a las hojas de estilo](#)
2. [Añadir estilos al HTML](#)
 1. [Especificación del lenguaje de hojas de estilo por defecto](#)
 2. [Información de estilo en línea](#)
 3. [Información de estilo en cabecera: el elemento STYLE](#)
 4. [Tipos de medios](#)
3. [Hojas de estilo externas](#)
 1. [Hojas de estilo preferentes y alternativas](#)
 2. [Especificación de hojas de estilo externas](#)
4. [Hojas de Estilo en Cascada](#)
 1. [Cascadas dependientes del medio](#)
 2. [Herencia y cascada](#)
5. [Ocultar datos de estilo a los agentes de usuario](#)
6. [Vinculación de hojas de estilo mediante encabezados HTTP](#)

14.1 Introducción a las hojas de estilo

Las hojas de estilo representan un avance importante para los diseñadores de páginas web, al darles un mayor rango de posibilidades para mejorar la apariencia de sus páginas. En los entornos científicos en que la Web fue concebida, la gente estaba más preocupada por el contenido de sus páginas que por su presentación. A medida que la Web era descubierta por un espectro mayor de personas de distintas procedencias, las limitaciones del HTML se convirtieron en fuente de continua frustración, y los autores se vieron forzados a superar las limitaciones estilísticas del HTML. Aunque las intenciones han sido buenas -- mejorar la presentación de las páginas web --, las técnicas para conseguirlo han tenido efectos secundarios negativos. Entre estas técnicas, que dan buenos resultados para algunas personas, algunas veces, pero no siempre ni para todas las personas, se incluyen:

- La utilización de extensiones propietarias del HTML
- Conversión del texto en imágenes
- Utilización de imágenes para controlar el espacio en blanco
- La utilización de tablas para la organización de las páginas
- Escribir programas en lugar de usar HTML

Estas técnicas incrementan considerablemente la complejidad de las páginas web, ofrecen una flexibilidad limitada, sufren de problemas de interoperabilidad, y crean dificultades para las personas con discapacidades.

Las hojas de estilo resuelven estos problemas al mismo tiempo que reemplazan al limitado rango de mecanismos de presentación del HTML. Con las hojas de estilo es más fácil especificar la cantidad de espacio entre líneas, el sangrado de las líneas, los colores a utilizar para el texto y el fondo, el tamaño y estilo de las fuentes, y otros muchos detalles.

Por ejemplo, la siguiente hoja de estilo CSS (almacenada en el fichero "especial.css") hace que el color del texto de un párrafo sea verde ("green") y lo rodea de un borde rojo ("red") continuo ("solid"):

```
P.especial {  
color : green;  
border: solid red;  
}
```

Los autores pueden vincular esta hoja de estilo a su documento fuente HTML con el elemento [LINK](#):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">  
<HTML>  
<HEAD>  
<TITLE>Un documento con una hoja de estilo externa<TITLE>  
<LINK href="especial.css" rel="stylesheet" type="text/css">  
</HEAD>  
<BODY>  
<P class="especial">Este párrafo debería tener texto especial verde.  
</BODY>  
</HTML>
```

HTML 4 soporta las siguientes características de hojas de estilo:

Colocación flexible de la información de estilo

Al colocar las hojas de estilo en ficheros separados es más fácil reutilizarlas. Algunas veces es útil incluir instrucciones de representación dentro del documento al que se aplican, ya sea agrupadas al comienzo del documento, o en atributos de los elementos a lo largo del cuerpo del documento. Para facilitar la gestión de estilos a nivel de sitio, esta especificación describe cómo utilizar los encabezados HTTP para especificar qué hojas de estilo se aplican a un documento.

Independencia de lenguajes de hojas de estilo específicos

Esta especificación no liga el HTML con ningún lenguaje de hojas de estilo en particular. Esto permite que se puedan usar otros lenguajes diferentes, desde los lenguajes más simples válidos para la mayoría de los usuarios, hasta los más complejos, útiles para una minoría de usuarios con necesidades muy especializadas. Todos los ejemplos que se incluyen en lo que sigue se sirven del lenguaje CSS (Hojas de Estilo en Cascada, Cascading Style Sheets, [\[CSS1\]](#)), pero también son posibles otros lenguajes de hojas de estilo.

Cascada

Se llama así a la capacidad que proporcionan algunos lenguajes de hojas de estilo tales como CSS que permiten que las informaciones de estilo provenientes de varias fuentes puedan combinarse. Éstas podrían ser, por ejemplo, las guías de estilo de una empresa, los estilos comunes a un grupo de documentos, y los estilos específicos de un documento en particular. Al almacenarse independientemente, las hojas de estilo pueden reutilizarse, lo cual simplifica las tareas de diseño y hace más efectiva la utilización de la memoria caché de la red. La cascada define una secuencia ordenada de hojas de estilo en la que las reglas de las últimas hojas tienen una prioridad mayor que las de las primeras. No todos los lenguajes de hojas de estilo soportan la cascada.

Dependencias de los medios

HTML permite a los autores especificar sus documentos de una manera independiente del medio. Esto permite a los usuarios acceder a las páginas web usando una amplia gama de dispositivos y medios, p.ej., pantallas gráficas para ordenadores bajo Windows, Macintosh OS y X11, dispositivos para aparatos de televisión, teléfonos adaptados y dispositivos portátiles PDA, navegadores basados en voz, y dispositivos táctiles Braille.

Las hojas de estilo, en contraste, se aplican a medios o grupos de medios específicos. Una hoja de estilo diseñada para una pantalla puede ser aplicable para una salida impresa, pero es de poca utilidad para los navegadores basados en voz. Esta especificación le permite definir categorías generales de medios a los que es aplicable una hoja de estilo dada. Esto permite a los agentes de usuario evitar la descarga de hojas de estilo que no sean apropiadas. Los lenguajes de hojas de estilo pueden incluir características para describir dependencias del medio dentro de una misma hoja de estilo.

Estilos alternativos

Los autores pueden querer ofrecer a los lectores varias maneras diferentes de ver un documento. Por ejemplo, una hoja de estilo para representar documentos compactos con fuentes pequeñas, o una que especifique fuentes más grandes para una lectura más fácil. Esta especificación permite a los autores especificar una hoja de estilo preferente así como hojas alternativas que se dirijan a medios o usuarios específicos. Los agentes de usuario deberían dar a los usuarios la oportunidad de seleccionar una de las hojas de estilo alternativas o incluso de desactivar las hojas de estilo completamente.

Consideraciones de rendimiento

Algunas personas han expresado su preocupación acerca de los posibles problemas de rendimiento relacionados con las hojas de estilo. Por ejemplo, la obtención de una hoja de estilo externa puede retrasar la presentación del documento al usuario. Algo parecido sucede si la cabecera del documento contiene un conjunto muy grande de reglas de estilo.

La propuesta actual resuelve estos problemas, pues permite a los autores incluir instrucciones de representación dentro de cada elemento HTML. Así, la información de representación siempre estará disponible en el momento en que el agente de usuario quiera representar cada elemento.

En muchos casos los autores sacarán partido de una hoja de estilo común a un grupo de documentos. En este caso, la distribución de reglas de estilo a lo largo del documento conducirá a un rendimiento peor que si se usara una hoja de estilo vinculada, ya que para la mayoría de los documentos la hoja de estilo ya estará almacenada en la caché local. La disponibilidad pública de hojas de estilo de calidad potenciará este efecto.

14.2 Añadir estilos al HTML

Nota. La hoja de estilo por defecto del HTML 4 que se incluye en [\[CSS2\]](#) expresa la información de estilo por defecto generalmente aceptada para cada elemento. Los autores, así como los implementadores, pueden encontrar en ella un recurso útil.

Los documentos HTML pueden contener reglas de hojas de estilo directamente, o bien pueden importar hojas de estilo.

Se puede usar cualquier lenguaje de hojas de estilo con HTML. Un lenguaje simple de hojas de estilo puede ser suficiente para la mayoría de los usuarios, pero para otras necesidades más especializadas pueden ser apropiados otros lenguajes. Esta especificación utiliza para los ejemplos el lenguaje de estilo llamado "Hojas de Estilo en Cascada" ("Cascading Style Sheets", [\[CSS1\]](#)), abreviado CSS.

La [sintaxis de los datos de estilo](#) depende del lenguaje de hojas de estilo.

14.2.1 Especificación del lenguaje de hojas de estilo por defecto

Los autores deben especificar el lenguaje de hojas de estilo de la información de estilo asociada con un documento HTML.

Los autores deberían usar el elemento [META](#) para especificar el lenguaje de hojas de estilo por defecto de un documento. Por ejemplo, para especificar que el valor por defecto es CSS, los autores deberían poner la siguiente declaración en la sección [HEAD](#) de sus documentos:

```
<META http-equiv="Content-Style-Type" content="text/css">
```

El lenguaje de hojas de estilo por defecto también puede ser establecido con encabezados HTTP. La declaración [META](#) de arriba es equivalente al encabezado HTTP siguiente:

```
Content-Style-Type: text/css
```

Los agentes de usuario deberían determinar el lenguaje de hojas de estilo por defecto de acuerdo con los siguientes pasos (ordenados de mayor a menor prioridad):

1. Si alguna declaración [META](#) especifica "Content-Style-Type", la última de ellas en el flujo de caracteres determina el lenguaje de hojas de estilo por defecto.
2. De otro modo, si algún encabezado HTTP especifica "Content-Style-Type", el último de ellos en el flujo de caracteres determina el lenguaje de hojas de estilo por defecto.
3. De otro modo, el lenguaje de hojas de estilo por defecto es "text/css".

Los documentos que incluyan elementos que especifiquen el atributo [style](#) pero que no definan un lenguaje de hojas de estilo por defecto son incorrectos. Las herramientas de creación deberían generar información de lenguaje de hojas de estilo por defecto (normalmente una declaración [META](#)) de modo que los agentes de usuario no tengan que atenerse al valor por defecto "text/css".

14.2.2 Información de estilo en línea

Definiciones de atributos

style = [estilo](#) [CN]

Este atributo especifica información de estilo para el elemento actual.

La [sintaxis](#) del valor del atributo [style](#) viene determinada por el [lenguaje de hojas de estilo por defecto](#). Por ejemplo, para los estilos [[CSS2]] en línea, se usa la sintaxis de bloque de declaraciones descrita en la sección 4.1.8 (sin las llaves).

Este ejemplo CSS establece información sobre el color y el tamaño de la fuente del texto de un párrafo específico.

```
<P style="font-size: 12pt; color: fuchsia">¿No son maravillosas las hojas de estilo?
```

En CSS, las declaraciones de propiedades son de la forma "nombre : valor" y están separadas por un punto y coma.

Para especificar información de estilo en línea para más de un elemento, los autores deberían usar el elemento STYLE. Para lograr una flexibilidad óptima, los autores deberían definir los estilos en hojas de estilo externas.

14.2.3 Información de estilo en cabecera: el elemento STYLE

```
<!ELEMENT STYLE - - %StyleSheet          -- información de estilo -->
<!ATTLIST STYLE
  %i18n;          -- lang, dir, para usar con title --
  type           %ContentType; #REQUIRED -- tipo de contenido del lenguaje de estilos
  --
  media         %MediaDesc;    #IMPLIED -- diseñado para usar con estos medios --
  title         %Text;        #IMPLIED -- título consultivo --
  >
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

Definiciones de atributos

type = [tipo de contenido](#) [CI]

Este atributo especifica el lenguaje de hojas de estilo de los contenidos del elemento y prevalece sobre el lenguaje de hojas de estilo por defecto. El lenguaje de hojas de estilo se especifica como un tipo de contenido (p.ej., "text/css").

Los autores deben proporcionar un valor para este atributo; no hay valor por defecto para este atributo.

media = [descriptores de medios](#) [CI]

Este atributo especifica el medio destino al que se dirige la información de estilo. Puede ser un solo descriptor de medios, o una lista de ellos separados por comas. El valor por defecto de este atributo es "screen" (pantalla).

Atributos definidos en otros lugares

- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)

El elemento [STYLE](#) permite a los autores poner reglas de hojas de estilo en la cabecera del documento. HTML permite cualquier número de elementos [STYLE](#) en la sección [HEAD](#) de un documento.

Los agentes de usuario que no soporten hojas de estilo, o que no soporten el lenguaje de hojas de estilo utilizado por un elemento [STYLE](#) en particular, deben ocultar los contenidos del elemento [STYLE](#). Es un error representar el contenido como parte del texto del documento. Algunos lenguajes de hojas de estilo soportan sintaxis para [ocultar el contenido](#) a agentes de usuarios no conformes.

La [sintaxis de los datos de estilo](#) depende del lenguaje de hojas de estilo.

Algunas implementaciones de hojas de estilo pueden permitir una gama más amplia de reglas en el elemento [STYLE](#) que en el atributo [style](#). Por ejemplo, con CSS, pueden declararse reglas dentro de un elemento [STYLE](#) para:

- Todas las apariciones de un elemento de HTML específico (p.ej., todos los elementos [P](#), todos los elementos [H1](#), etc.)

- Todas las apariciones de un elemento de HTML que pertenezcan a una clase específica (es decir, cuyo atributo [class](#) tome cierto valor).
- Las apariciones únicas de un elemento de HTML (es decir, cuyo atributo [id](#) tome cierto valor).

Las reglas de precedencia y herencia de las reglas de estilo dependen del lenguaje de hojas de estilo.

La siguiente declaración [STYLE](#) CSS pone un borde alrededor de todos los elementos [H1](#) del documento y los centra en la página.

```
<HEAD>
  <STYLE type="text/css">
    H1 {border-width: 1px; border: solid; text-align: center}
  </STYLE>
</HEAD>
```

Para especificar que esta información de estilo sólo debería aplicarse a los elementos [H1](#) de una clase específica, la modificamos como sigue:

```
<HEAD>
  <STYLE type="text/css">
    H1.miclase {border-width: 1px; border: solid; text-align: center}
  </STYLE>
</HEAD>
<BODY>
  <H1 class="miclase"> Este H1 está afectado por nuestro estilo </H1>
  <H1> Este no está afectado por nuestro estilo </H1>
</BODY>
```

Finalmente, para limitar el alcance de la información de estilo a una sola aparición de [H1](#), establecemos el atributo [id](#):

```
<HEAD>
  <STYLE type="text/css">
    #miid {border-width: 1px; border: solid; text-align: center}
  </STYLE>
</HEAD>
<BODY>
  <H1 class="miclase"> Este H1 no está afectado </H1>
  <H1 id="miid"> Este H1 está afectado por el estilo </H1>
  <H1> Este H1 no está afectado </H1>
</BODY>
```

Aunque se puede especificar información de estilo prácticamente para cualquier elemento, hay dos elementos, [DIV](#) y [SPAN](#), que son particularmente útiles en el sentido de que no imponen ningún significado presentacional (aparte de la distinción [en bloque/en línea](#)). Combinados con hojas de estilo, estos elementos permiten a los usuarios extender el HTML indefinidamente, en especial si se utilizan junto con los atributos [class](#) e [id](#).

En el siguiente ejemplo, utilizamos el elemento [SPAN](#) para especificar que el estilo de fuente de las primeras palabras de un párrafo sea versalitas ("small-caps").

```
<HEAD>
  <STYLE type="text/css">
    SPAN.ver-ex { font-variant: small-caps }
  </STYLE>
</HEAD>
<BODY>
  <P><SPAN class="ver-ex">Las primeras</SPAN> palabras
  de este párrafo van en versalitas.
</BODY>
```

En el siguiente ejemplo, utilizamos el elemento [DIV](#) y el atributo [class](#) para establecer la justificación del texto de una serie

de párrafos que constituyen el abstracto de un artículo científico. Esta información de estilo podría reutilizarse para otras secciones de abstracto especificando el valor correspondiente del atributo `class` en cualquier otro lugar del documento.

```
<HEAD>
  <STYLE type="text/css">
    DIV.Abstracto { text-align: justify }
  </STYLE>
</HEAD>
<BODY>
  <DIV class="Abstracto">
    <P>La gama de productos Gran Jefe es nuestra apuesta
      para el año que viene. Este informe sienta las bases
      para confrontar a Gran Jefe con los productos de la
      competencia.

    <P>Gran Jefe reemplaza a nuestra gama "Commander",
      si bien ésta seguirá de momento en catálogo.
  </DIV>
</BODY>
```

14.2.4 Tipos de medios

HTML permite a los autores diseñar documentos que saquen partido de las características de los medios en los que el documento vaya a ser representado (p.ej., pantallas gráficas, pantallas de televisión, dispositivos de mano, navegadores basados en voz, dispositivos táctiles Braille, etc.). Al especificar el atributo `media`, los autores permiten a los agentes de usuario cargar y aplicar las hojas de estilo de manera selectiva. Consulte la lista de [descriptores de medios](#) reconocidos.

Las siguientes declaraciones de ejemplo se aplican a elementos [H1](#). Cuando se proyecten en una reunión de negocios, todos ellos aparecerán en azul. Cuando se impriman, aparecerán centrados.

```
<HEAD>
  <STYLE type="text/css" media="projection">
    H1 { color: blue }
  </STYLE>

  <STYLE type="text/css" media="print">
    H1 { text-align: center }
  </STYLE>
```

Este ejemplo añade efectos de sonido a los vínculos para su uso en salidas por voz:

```
<STYLE type="text/css" media="aural">
  A { cue-before: uri(bell.aiff); cue-after: uri(dong.wav) }
</STYLE>
</HEAD>
```

El control de medios es particularmente interesante cuando se aplica a hojas de estilo externas, ya que los agentes de usuario pueden ahorrar tiempo si obtienen de la red únicamente aquellas hojas de estilo que se apliquen al dispositivo actual. Por ejemplo, los navegadores basados en voz pueden evitar la descarga de hojas de estilo diseñadas para la representación visual. Véase la sección sobre [cascadas dependientes del medio](#) para más información.

14.3 Hojas de estilo externas

Los autores pueden separar las hojas de estilo de los documentos HTML. Esto ofrece varias ventajas:

- Los autores y los administradores de sitios web pueden compartir hojas de estilo entre varios documentos (y sitios).
- Los autores pueden cambiar la hoja de estilo sin necesidad de hacer modificaciones en el documento.
- Los agentes de usuario pueden cargar hojas de estilo selectivamente (en función de los descriptores de medios).

14.3.1 Hojas de estilo preferentes y alternativas

HTML permite a los autores asociar cualquier número de hojas de estilo externas a un documento. El lenguaje de hojas de estilo define el modo en que interaccionan varias hojas de estilo (por ejemplo, las reglas de "cascada" de CSS).

Los autores pueden especificar cualquier número de hojas de estilo mutuamente excluyentes llamadas hojas de estilo alternativas. Los usuarios pueden seleccionar su hoja favorita según sus preferencias. Por ejemplo, un autor puede especificar una hoja de estilo diseñada para pantallas pequeñas y otra para usuarios con poca visión (p.ej., con fuentes grandes). Los agentes de usuario deberían permitir a los usuarios seleccionar las hojas de estilo alternativas.

El autor puede especificar que una de las alternativas es una hoja de estilo preferente. Los agentes de usuario deberían aplicar la hoja de estilo preferente del autor a menos que el usuario haya seleccionado una alternativa diferente.

Los autores pueden agrupar varias hojas de estilo alternativas (incluyendo las hojas de estilo preferentes del autor) bajo un nombre de estilo común. Cuando un usuario selecciona un nombre de estilo, el agente de usuario debe aplicar todas las hojas de estilo con ese nombre. Los agentes de usuario no deben aplicar hojas de estilo alternativas con un nombre de estilo diferente. La sección sobre [especificación de hojas de estilo externas](#) explica cómo dar un nombre a un grupo de hojas de estilo.

Los autores también pueden especificar hojas de estilo persistentes que los agentes de usuario deben aplicar además de cualquier hoja de estilo alternativa.

Los agentes de usuario deben respetar los [descriptores de medios](#) al aplicar cualquier hoja de estilo.

Los agentes de usuario también deberían permitir a los usuarios deshabilitar completamente las hojas de estilo del autor, en cuyo caso el agente de usuario no debería aplicar ninguna hoja de estilo persistente ni alternativa.

14.3.2 Especificación de hojas de estilo externas

Los autores especifican hojas de estilo externas mediante los siguientes atributos del elemento [LINK](#):

- Haciendo el valor de [href](#) igual a la localización del fichero de la hoja de estilo. El valor de [href](#) es un [URI](#).
- Haciendo que el valor del atributo [type](#) indique el lenguaje del recurso (hoja de estilo) vinculado. Esto permite al agente de usuario evitar la descarga de una hoja de estilo cuyo lenguaje no soporta.
- Especificando si la hoja de estilo es persistente, preferente, o alternativa:
 - Para hacer una hoja de estilo persistente, se hace el atributo [rel](#) igual a "stylesheet" y no se establece el atributo [title](#).
 - Para hacer una hoja de estilo preferente, se hace el atributo [rel](#) igual a "stylesheet" y se da un nombre a la hoja de estilo mediante el atributo [title](#).
 - Para especificar una hoja de estilo alternativa, se hace el atributo [rel](#) igual a "alternate stylesheet" y se da un nombre a la hoja de estilo mediante el atributo [title](#).

Los agentes de usuario deberían posibilitar a los usuarios ver la lista de estilos alternativos y escoger uno de ellos. Se recomienda que el nombre de cada opción sea el valor del atributo [title](#).

En este ejemplo, especificamos en primer lugar una hoja de estilo persistente localizada en el fichero `miestilo.css`:

```
<LINK href="miestilo.css" rel="stylesheet" type="text/css">
```

Si establecemos el atributo [title](#) la convertimos en la hoja de estilo preferente del autor:

```
<LINK href="miestilo.css" title="compacto" rel="stylesheet" type="text/css">
```

Si añadimos la palabra clave "alternate" al atributo [rel](#) la convertimos en una hoja de estilo alternativa:

```
<LINK href="miestilo.css" title="Medium" rel="alternate stylesheet" type="text/css">
```

Para más información sobre hojas de estilo externas, consulte la sección sobre [vínculos y hojas de estilo externas](#).

Los autores también pueden usar el elemento [META](#) para establecer la hoja de estilo preferente del documento. Por ejemplo, para especificar que la hoja de estilo preferente sea "compacto" (ver el ejemplo precedente), los autores pueden incluir la

siguiente línea en el [HEAD](#):

```
<META http-equiv="Default-Style" content="compacto">
```

La hoja de estilo preferente también se puede especificar mediante encabezados HTTP. La declaración [META](#) precedente es equivalente a este encabezado HTTP:

```
Default-Style: "compacto"
```

Si hay dos o más declaraciones [META](#) o encabezados HTTP que especifiquen la hoja de estilo preferente, la que prevalece es la última. A estos efectos, se considera que los encabezados HTTP aparecen antes que la sección [HEAD](#) del documento.

Si hay dos o más elementos [LINK](#) que especifiquen una hoja de estilo preferente, el que prevalece es el primero.

Las hojas de estilo preferentes especificadas con un elemento [META](#) o con encabezados HTTP prevalecen sobre las especificadas con el elemento [LINK](#).

14.4 Hojas de Estilo en Cascada

Los lenguajes de hojas de estilo en cascada, tales como CSS, permiten que se pueda combinar la información de varias fuentes. Sin embargo, no todos los lenguajes de hojas de estilo soportan la cascada. Para definir una cascada, los autores especifican una secuencia de elementos [LINK](#) y/o [STYLE](#). La información de estilo se combina en cascada según el orden en que aparecen los elementos en la sección [HEAD](#).

***Nota.** Esta especificación no especifica cómo se combinan en cascada hojas de estilo de lenguajes de hojas de estilo distintos. Los autores deberían evitar mezclar lenguajes de hojas de estilo.*

En el siguiente ejemplo, especificamos dos hojas de estilo alternativas llamadas "compacto". Si el usuario selecciona el estilo "compacto", el agente de usuario debe aplicar ambas hojas de estilo externas, así como la hoja de estilo persistente "comun.css". Si el usuario selecciona el estilo "tipos grandes", sólo se aplicarán la hoja de estilo alternativa "tgrandes.css" y "comun.css".

```
<LINK rel="alternate stylesheet" title="compacto"
      href="peq-base.css" type="text/css">
<LINK rel="alternate stylesheet" title="compacto"
      href="peq-extras.css" type="text/css">
<LINK rel="alternate stylesheet" title="tipos grandes"
      href="tgrandes.css" type="text/css">
<LINK rel="stylesheet" href="comun.css" type="text/css">
```

Aquí tenemos un ejemplo de cascada que incluye tanto al elemento [LINK](#) como al [STYLE](#).

```
<LINK rel="stylesheet" href="empresa.css" type="text/css">
<LINK rel="stylesheet" href="informe.css" type="text/css">
<STYLE type="text/css">
  p.especial { color: rgb(230, 100, 180) }
</STYLE>
```

14.4.1 Cascadas dependientes del medio

Una cascada puede incluir hojas de estilo aplicables a medios diferentes. Tanto [LINK](#) como [STYLE](#) pueden usarse con el atributo [media](#). El agente de usuario es por tanto responsable de filtrar aquellas hojas de estilo que no se apliquen al medio actual.

En el siguiente ejemplo, definimos un cascada en la cual se incluyen varias versiones de la hoja de estilo "empresa": una diseñada para su impresión, otra para la pantalla y otra para navegadores basados en voz (útil, por ejemplo, para leer correo electrónico en el coche). La hoja de estilo "informe" se aplica a todos los medios. La regla de color definida por el elemento [STYLE](#) se usa para impresoras y pantallas, pero no para la representación auditiva.

```
<LINK rel="stylesheet" media="aural" href="empresa-aural.css" type="text/css">
<LINK rel="stylesheet" media="screen" href="empresa-screen.css" type="text/css">
<LINK rel="stylesheet" media="print" href="empresa-print.css" type="text/css">
<LINK rel="stylesheet" href="informe.css" type="text/css">
<STYLE media="screen, print" type="text/css">
  p.especial { color: rgb(230, 100, 180) }
</STYLE>
```

14.4.2 Herencia y cascada

Cuando el agente de usuario quiere representar un documento, necesita encontrar valores para las propiedades de estilo, p.ej., la familia tipográfica, el tamaño de fuente, la altura de línea, el color del texto, etc. El mecanismo exacto depende del lenguaje de la hoja de estilo, pero en general es válida la descripción siguiente:

Se utiliza el mecanismo de cascada cuando varias reglas de estilo se aplican directamente a un mismo elemento. El mecanismo permite al agente de usuario ordenar las reglas según su especificidad para determinar la regla a aplicar. Si no encuentra ninguna regla, el siguiente paso depende de si la propiedad de estilo puede ser heredada o no. No todas las propiedades se heredan. Para estas propiedades el lenguaje de la hoja de estilo proporciona unos valores por defecto que son los que se usan cuando no hay reglas explícitas para un elemento en particular.

Si la propiedad puede ser heredada, el agente de usuario examina el elemento contenedor inmediato para ver si hay alguna regla que se aplique a él. Este proceso continúa hasta que se encuentra alguna regla aplicable. Este mecanismo permite que las hojas de estilo puedan definirse de una manera compacta. Por ejemplo, los autores pueden especificar la familia tipográfica de todos los elementos contenidos en [BODY](#) con una sola regla que se aplique al elemento [BODY](#).

14.5 Ocultar datos de estilo a los agentes de usuario

Algunos lenguajes de hojas de estilo soportan sintaxis pensada para permitir a los autores ocultar el contenido de los elementos [STYLE](#) a los agentes de usuario no conformes.

Este ejemplo ilustra cómo comentar en CSS el contenido de los elementos [STYLE](#) para asegurarse de que los agentes de usuario antiguos y no conformes no lo representarían como texto.

```
<STYLE type="text/css">
<!--
  H1 { color: red }
  P  { color: blue}
  -->
</STYLE>
```

14.6 Vinculación de hojas de estilo mediante encabezados HTTP

Esta sección sólo se aplica a los agentes de usuario conformes con las versiones de HTTP que definan un campo de encabezado Link. Obsérvese que HTTP 1.1, tal y como se define en [\[RFC2616\]](#), no incluye un campo de encabezado Link (ver sección 19.6.3).

Los administradores de servidores web pueden creer conveniente configurar un servidor de modo que una hoja de estilo se aplique a un grupo de páginas. El encabezado HTTP Link tiene el mismo efecto que un elemento [LINK](#) con los mismos atributos y valores. Varios encabezados Link corresponderían a varios elementos [LINK](#) que aparecieran en el mismo orden. Por ejemplo,

```
Link: <http://www.acme.com/empresa.css>; REL=stylesheet
```

corresponde a:

```
<LINK rel="stylesheet" href="http://www.acme.com/empresa.css">
```

Es posible especificar varias hojas de estilo alternativas especificando varios encabezados `Link`, y a continuación utilizar el atributo `rel` para determinar el estilo por defecto.

En el ejemplo siguiente, se aplica "compacto" por defecto, ya que omite la palabra clave "alternate" del atributo `rel`.

```
Link: <compacto.css>; rel="stylesheet"; title="compacto"
```

```
Link: <tgrandes.css>; rel="alternate stylesheet"; title="tipos grandes"
```

Esto debería funcionar cuando se envían documentos HTML por correo electrónico. Algunos agentes de correo pueden alterar el orden de los encabezados de [RFC822](#). Para evitar que esto afecte al orden de cascada de las hojas de estilo especificadas con encabezados `Link`, los autores pueden usar la concatenación de encabezados para combinar varias apariciones del mismo campo de encabezado. Las comillas sólo se necesitan si los valores de los atributos contienen algún espacio en blanco. Utilice entidades SGML para hacer referencia a caracteres que no se permitan en encabezados de HTML o de correo electrónico, o que sean susceptibles de resultar afectados por el tránsito a lo largo de la red.

Los elementos [LINK](#) y [META](#) declarados implícitamente por encabezados HTTP aparecen antes de cualquier elemento explícito [LINK](#) o [META](#) en el [HEAD](#) del documento.

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

15 Alineación, estilos de fuente y separadores horizontales

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Formato](#)
 1. [Color de fondo](#)
 2. [Alineación](#)
 3. [Objetos flotantes](#)
 - [Hacer flotar a un objeto](#)
 - [Texto flotante alrededor de un objeto](#)
2. [Fuentes](#)
 1. [Elementos de estilo de fuente: los elementos TT, I, B, BIG, SMALL, STRIKE, S y U](#)
 2. [Elementos de modificación de fuentes: FONT y BASEFONT](#)
3. [Separadores: el elemento HR](#)

Esta sección de la especificación trata sobre algunos elementos y atributos HTML que pueden utilizarse para el formato visual de elementos. Muchos de ellos están [desaprobados](#).

15.1 Formato

15.1.1 Color de fondo

Definiciones de atributos

bgcolor = [color](#) [[CI](#)]

[Desaprobado.](#) Este atributo establece el color de fondo del cuerpo del documento o de las celdas de una tabla.

Este atributo establece el color de fondo del lienzo del cuerpo del documento (el elemento [BODY](#)) o de las tablas (los elementos [TABLE](#), [TR](#), [TH](#) y [TD](#)). Con el elemento [BODY](#) pueden utilizarse atributos adicionales para especificar el color del texto.

Este atributo ha sido [desaprobado](#) en favor de las hojas de estilo para la especificación de información sobre el color de fondo.

15.1.2 Alineación

Es posible alinear elementos en bloque (tablas, imágenes, objetos, párrafos, etc.) en el lienzo con el atributo align. Aunque este atributo se puede establecer en muchos elementos HTML, su rango de valores posibles difiere en ocasiones de unos elementos a otros. Aquí sólo se considera el significado del atributo align para texto.

Definiciones de atributos

align = left|center|right|justify [[CI](#)]

[Desaprobado.](#) Este atributo especifica la alineación horizontal de su elemento con respecto al contexto circundante.

Valores posibles:

- left: las líneas de texto se representan con los bordes izquierdos alineados.
- center: las líneas de texto son centradas.
- right: las líneas de texto se representan con los bordes derechos alineados.
- justify: las líneas de texto se justifican en ambos márgenes.

El valor por defecto depende de la dirección base del texto. Para un texto de izquierda a derecha el valor por defecto es align=left, mientras que para un texto de derecha a izquierda el valor por defecto es align=right.

EJEMPLO DESAPROBADO:

Este ejemplo centra un encabezado en el lienzo.

```
<H1 align="center"> Cómo Tallar la Madera </H1>
```

Usando CSS, por ejemplo, podríamos conseguir el mismo efecto de la manera siguiente:

```
<HEAD>
<TITLE>Cómo Tallar la Madera</TITLE>
<STYLE type="text/css">
  H1 { text-align: center }
</STYLE>
<BODY>
<H1> Cómo Tallar la Madera </H1>
```

Obsérvese que esto centrará todas las declaraciones [H1](#). Podríamos reducir el alcance del estilo estableciendo el atributo [class](#) del elemento:

```
<HEAD>
<TITLE>Cómo Tallar la Madera</TITLE>
<STYLE type="text/css">
  H1.madera {text-align: center}
</STYLE>
<BODY>
<H1 class="madera"> Cómo Tallar la Madera </H1>
```

EJEMPLO DESAPROBADO:

Análogamente, para alinear a la derecha un párrafo en el lienzo con el atributo de HTML [align](#) podríamos tener:

```
<P align="right">...Montones de párrafos de texto...
```

lo cual, con CSS, se convertiría en:

```
<HEAD>
<TITLE>Cómo Tallar la Madera</TITLE>
<STYLE type="text/css">
  P.mipar {text-align: right}
</STYLE>
<BODY>
<P class="mipar">...Montones de párrafos de texto...
```

EJEMPLO DESAPROBADO:

Para alinear a la derecha una serie de párrafos, los agrupamos con el elemento [DIV](#):

```
<DIV align="right">
<P>...texto del primer párrafo...
<P>...texto del segundo párrafo...
<P>...texto del tercer párrafo...
</DIV>
```

Con CSS, la propiedad `text-align` se hereda del elemento padre a los hijos, y por tanto podemos decir:

```
<HEAD>
<TITLE>Cómo Tallar la Madera</TITLE>
<STYLE type="text/css">
  DIV.parrafos {text-align: right}
</STYLE>
<BODY>
<DIV class="parrafos">
  <P>...texto del primer párrafo...
  <P>...texto del segundo párrafo...
  <P>...texto del tercer párrafo...
</DIV>
```

Para centrar el documento completo con CSS:

```
<HEAD>
<TITLE>Como Tallar la Madera</TITLE>
<STYLE type="text/css">
  BODY {text-align: center}
</STYLE>
<BODY>
  ...el cuerpo está centrado...
</BODY>
```

El elemento `CENTER` es exactamente equivalente a especificar el elemento `DIV` con el atributo `align` igual a "center". El elemento `CENTER` está **desaprobado**.

15.1.3 Objetos flotantes

Las imágenes y los objetos pueden aparecer directamente "en línea" o pueden hacerse "flotar" a un lado de la página, alterando temporalmente los márgenes del texto, que puede fluir por los lados del objeto.

Hacer flotar a un objeto

El atributo `align` aplicado a objetos, imágenes, tablas, marcos, etc., hace que el objeto flote hasta el margen izquierdo o derecho. Los objetos flotantes en general comienzan una nueva línea. Este atributo acepta los siguiente valores:

- `left`: Hace flotar al objeto hasta el margen izquierdo actual. El texto subsiguiente fluye alrededor del borde derecho del objeto.
- `right`: Hace flotar al objeto hasta el margen derecho actual. El texto subsiguiente fluye alrededor del borde izquierdo del objeto.

EJEMPLO DESAPROBADO:

El siguiente ejemplo muestra cómo hacer flotar un elemento `IMG` hasta el margen izquierdo actual del lienzo.

```
<IMG align="left" src="http://blabla.com/unaimagen.gif" alt="mi barca">
```

Algunos atributos de alineación también permiten el valor "center", que no hace que el objeto flote, sino que lo centra con respecto a los márgenes actuales. Sin embargo, para `P` y `DIV`, el valor "center" hace que los contenidos del elemento estén centrados.

Texto flotante alrededor de un objeto

Otro atributo, definido para el elemento `BR`, controla el flujo del texto alrededor de objetos flotantes.

Definiciones de atributos

`clear = none|left|right|all` [\[CI\]](#)

Desaprobado. Especifica dónde debería aparecer la línea siguiente en un navegador visual después del salto de

línea provocado por este elemento. Este atributo tiene en cuenta los objetos flotantes (imágenes, tablas, etc.).

Valores posibles:

- none: La siguiente línea empezará normalmente. Este es el valor por defecto.
- left: La siguiente línea comenzará en la primera línea que quede por debajo de los objetos que estén flotando en el margen izquierdo.
- right: La siguiente línea comenzará en la primera línea que quede por debajo de los objetos que estén flotando en el margen derecho.
- all: La siguiente línea comenzará en la primera línea que quede por debajo de los objetos que estén flotando en cualquiera de los dos márgenes.

Consideremos el siguiente escenario visual, en el cual el texto fluye por la derecha de una imagen hasta que la línea se rompe con un [BR](#):

```
*****  -----
|          | -----
| imagen  | --<BR>
|          | -----
*****
```

Si el atributo [clear](#) se hace igual a none, la línea que siga al [BR](#) comenzará inmediatamente debajo de él en el margen derecho de la tabla:

```
*****  -----
|          | -----
| imagen  | --<BR>
|          | -----
*****
```

EJEMPLO DESAPROBADO:

Si el atributo [clear](#) se hace igual a left o all, la siguiente línea aparecerá como se muestra a continuación:

```
*****  -----
|          | -----
| imagen  | --<BR clear="left">
|          | -----
*****
```

Usando hojas de estilo, podríamos especificar que todos los saltos de línea deberían comportarse de esta manera para los objetos (imágenes, tablas, etc.) que estuvieran flotando en el margen izquierdo. Con CSS podríamos conseguirlo de esta manera:

```
<STYLE type="text/css">
BR { clear: left }
</STYLE>
```

Para especificar este comportamiento para un elemento [BR](#) específico, podríamos combinar la información de estilo y el atributo [id](#):

```
<HEAD>
...
<STYLE type="text/css">
BR#mibr { clear: left }
</STYLE>
</HEAD>
<BODY>
<P>...
```

```
*****
|          |          |
|  tabla  |          |
|          |          |
*****
-----
...
</BODY>
```

15.2 Fuentes

Los siguientes elementos HTML especifican información de fuentes. Aunque no todos están [desaprobados](#), se desaconseja su uso en favor de las hojas de estilo.

15.2.1 Elementos de estilo de fuente: los elementos TT, I, B, BIG, SMALL, STRIKE, S y U

```
<!ENTITY % fontstyle
"TT | I | B | BIG | SMALL">
<!ELEMENT (%fontstyle;|%phrase;) - - (%inline;)*>
<!ATTLIST (%fontstyle;|%phrase;)
%attrs; -- %coreattrs, %i18n, %events --
>
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el lenguaje), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#) [onkeyup](#) (eventos intrínsecos)

La representación de elementos de estilo de fuente depende del agente de usuario. Lo que sigue es solamente una descripción informativa.

- TT:** Representa como texto de teletipo o ancho fijo.
- I:** Representa como estilo de texto itálica.
- B:** Representa como estilo de texto negrita.
- BIG:** Representa el texto con una fuente "grande".
- SMALL:** Representa el texto con una fuente "pequeña".
- STRIKE** y **S:** [Desaprobados](#). Representan texto de estilo tachado.
- U:** [Desaprobado](#). Representa texto subrayado.

La siguiente frase muestra varios tipos de texto:

```
<P><b>negrita</b>,
<i>itálica</i>, <b><i>negrita itálica</i></b>, <tt>texto de teletipo</tt>, y texto
<big>grande</big> y <small>pequeño</small>.
```

Estas palabras podrían ser representadas como se muestra a continuación:



Es posible lograr una variedad mucho mayor de efectos de fuentes usando hojas de estilo. Para especificar texto azul en itálica en un párrafo con CSS:

```
<HEAD>
<STYLE type="text/css">
P#mipar {font-style: italic; color: blue}
</STYLE>
</HEAD>
<P id="mipar">...Montones de texto azul en itálica...
```

Los elementos de estilo de fuente deben ser anidados correctamente. La representación de elementos de estilo de fuente anidados depende del agente de usuario.

15.2.2 Elementos de modificación de fuentes: FONT y BASEFONT

FONT y **BASEFONT** están **desaprobados**.

Véase el [DTD Transicional](#) para la definición formal.

Definiciones de atributos

size = [CDATA](#) [CN]

Desaprobado. Este atributo especifica el tamaño de la fuente. Valores posibles:

- Un entero entre 1 y 7. Esto establece la fuente en algún tamaño fijo, cuya representación depende del agente de usuario. No todos los agentes de usuario pueden representar los siete tamaños.
- Un incremento relativo del tamaño de la fuente. El valor "+1" significa un tamaño más grande. El valor "-3" significa tres tamaños menor. Todos los tamaños pertenecen a la escala de 1 a 7.

color = [color](#) [CI]

Desaprobado. Este atributo especifica el color del texto.

face = [CDATA](#) [CI]

Desaprobado. Este atributo define una lista de nombres de fuentes separados por comas que el agente de usuario debería buscar en orden de preferencia.

Atributos definidos en otros lugares

- [id](#), [class](#) ([identificadores a nivel de documento](#))
- [lang](#) ([información sobre el idioma](#)), [dir](#) ([dirección del texto](#))
- [title](#) ([título del elemento](#))
- [style](#) ([información de estilo en línea](#))

El elemento **FONT** cambia el tamaño y color de la fuente para el texto de sus contenidos.

El elemento **BASEFONT** establece el tamaño base de fuente (usando el atributo size). Los cambios del tamaño de la fuente producidos con **FONT** son relativos al tamaño base de fuente establecido por **BASEFONT**. Si no se usa **BASEFONT**, el tamaño base de fuente es 3.

EJEMPLO DESAPROBADO:

El siguiente ejemplo mostrará la diferencia entre los siete tamaños de fuente disponibles con **FONT**:

```
<P><font size=1>size=1</font>
<font size=2>size=2</font>
<font size=3>size=3</font>
<font size=4>size=4</font>
<font size=5>size=5</font>
<font size=6>size=6</font>
<font size=7>size=7</font>
```

Esto podría representarse así:

size=1 size=2 size=3 size=4 size=5 size=6 size=7

A continuación mostramos un ejemplo del efecto de los tamaños de fuente relativos usando un tamaño base de fuente 3:

size=-4 size=-3 size=-2 size=-1 size=+1 size=+2 size=+3 size=+4

El tamaño base de fuente no se aplica a los encabezados, excepto cuando éstos sean modificados por medio del elemento [FONT](#) con cambios de tamaño de fuente relativos.

15.3 Separadores: el elemento HR

```
<!ELEMENT HR - O EMPTY -- separador horizontal -->
<!ATTLIST HR
  %attrs;                -- %coreattrs, %i18n, %events --
  >
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **prohibida***

Definiciones de atributos

align = left|center|right [\[CI\]](#)

Desaprobado. Este atributo especifica la alineación horizontal del separador con respecto al contexto circundante.

Valores posibles:

- left: el separador se representa alineado a la izquierda.
- center: el separador se representa centrado.
- right: el separador se representa alineado a la derecha.

El valor por defecto es align=center.

noshade [\[CI\]](#)

Desaprobado. Si se establece, este atributo booleano pide al agente de usuario que represente el separador con un color sólido en lugar del "relieve" tradicional de dos colores.

size = *píxeles* [\[CI\]](#)

Desaprobado. Este atributo especifica la altura del separador. El valor por defecto de este atributo depende del agente de usuario.

width = *longitud* [\[CI\]](#)

Desaprobado. Este atributo especifica la anchura del separador. El valor por defecto es del 100%, es decir, el separador se extiende por todo el ancho del lienzo.

Atributos definidos en otros lugares

- [id](#), [class](#) ([identificadores a nivel de documento](#))
- [lang](#) ([información sobre el idioma](#)), [dir](#) ([dirección del texto](#))
- [title](#) ([título del elemento](#))
- [style](#) ([información de estilo en línea](#))
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) ([eventos intrínsecos](#))

El elemento [HR](#) hace que el agente de usuario represente una línea separadora horizontal.

La cantidad de espacio vertical insertado entre el separador y el contenido que le rodea depende del agente de usuario.

EJEMPLO DESAPROBADO:

Este ejemplo centra los separadores, dándoles un tamaño igual a la mitad de la anchura disponible entre márgenes. El separador superior tiene el tamaño por defecto, mientras que los dos inferiores son de 5 píxeles. El separador inferior

debería representarse con un color sólido sin efecto de relieve:

```
<HR width="50%" align="center">  
<HR size="5" width="50%" align="center">  
<HR noshade size="5" width="50%" align="center">
```

Estos separadores podrían representarse como sigue:



[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

16 Marcos

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Introducción a los marcos](#)
2. [Disposición de los marcos](#)
 1. [El elemento FRAMESET](#)
 - [Filas y columnas](#)
 - [Anidamiento de grupos de marcos](#)
 - [Compartir datos entre marcos](#)
 2. [El elemento FRAME](#)
 - [Especificación de los contenidos iniciales de un marco](#)
 - [Representación visual de un marco](#)
3. [Especificación de información sobre el marco destino](#)
 1. [Especificación del destino de los vínculos por defecto](#)
 2. [Semántica de marcos destino](#)
4. [Contenido alternativo](#)
 1. [El elemento NOFRAMES](#)
 2. [Descripciones largas de marcos](#)
5. [Marcos en línea: el elemento IFRAME](#)

16.1 Introducción a los marcos

Los marcos HTML permiten a los autores presentar documentos con vistas múltiples, que pueden ser ventanas o subventanas independientes. Las vistas múltiples ofrecen a los autores una manera de mantener cierta información visible mientras otras vistas se desplazan o se sustituyen. Por ejemplo, dentro de una misma ventana, un marco podría mostrar un gráfico estático, un segundo marco un menú de navegación, y un tercero el documento principal que puede ser desplazado, o reemplazado al navegar por el segundo marco.

Aquí tenemos un documento simple con marcos:

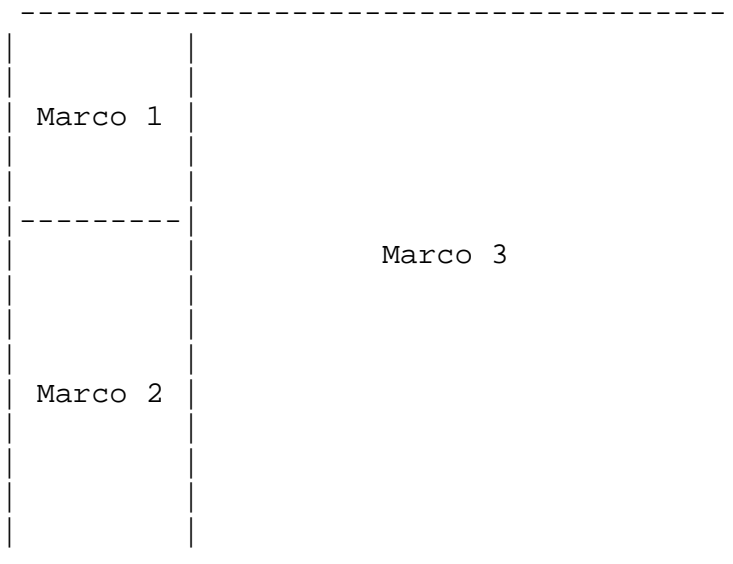
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
<HEAD>
<TITLE>Un documento simple con marcos</TITLE>
</HEAD>
<FRAMESET cols="20%, 80%">
  <FRAMESET rows="100, 200">
    <FRAME src="contenidos_del_marco1.html">
    <FRAME src="contenidos_del_marco2.gif">
  </FRAMESET>
<FRAME src="contenidos_del_marco3.html">
<NOFRAMES>
  <P>Este conjunto de marcos contiene:
```

```

<UL>
  <LI><A href="contenidos_del_marco1.html">Contenidos chéveres</A>
  <LI><IMG src="contenidos_del_marco2.gif" alt="Una imagen chévere">
  <LI><A href="contenidos_del_marco3.html">Más contenidos chéveres</A>
</UL>
</NOFRAMES>
</FRAMESET>
</HTML>

```

Esto podría crear una disposición de marcos como la siguiente:



Si el agente de usuario no puede mostrar marcos o está configurado para no mostrarlos, representará los contenidos del elemento [NOFRAMES](#).

16.2 Disposición de los marcos

Los documentos HTML que describen una disposición de marcos (llamados documentos con marcos) tienen una estructura diferente a la de los documentos HTML sin marcos. Un documento normal tiene una sección [HEAD](#) y una sección [BODY](#). Un documento con marcos tiene una sección [HEAD](#), y un [FRAMESET](#) en lugar del [BODY](#).

La sección [FRAMESET](#) de un documento especifica la disposición de las vistas en la ventana principal del agente de usuario. Además, la sección [FRAMESET](#) puede contener un elemento [NOFRAMES](#) que proporcione [contenido alternativo](#) para los agentes de usuario que no soporten marcos o que estén configurados para no mostrar marcos.

Los elementos que normalmente podrían colocarse en el elemento [BODY](#) no deben aparecer antes del primer elemento [FRAMESET](#) o el [FRAMESET](#) no será tenido en cuenta.

16.2.1 El elemento FRAMESET

```

<![ %HTML.Frameset; [
<!ELEMENT FRAMESET - - ((FRAMESET|FRAME)+ & NOFRAMES?) -- subdivisión en ventanas -->
<!ATTLIST FRAMESET
  %coreattrs;           -- id, class, style, title --
  rows                  %MultiLengths; #IMPLIED -- lista de longitudes,
                                     por defecto: 100% (1 fila) --
  cols                  %MultiLengths; #IMPLIED -- lista de longitudes,
                                     por defecto: 100% (1 columna) --
  onload                %Script;       #IMPLIED -- todos los marcos han sido cargados --
  onunload              %Script;       #IMPLIED -- todos los marcos han sido quitados --
>

```

```
] ]>
```

Definiciones de atributos

`rows` = [lista de multilongitudes \[CN\]](#)

Este atributo especifica la disposición de los marcos horizontales. Es una lista de longitudes en píxeles, porcentajes o longitudes relativas, separadas por comas. El valor por defecto es 100%, que significa una fila.

`cols` = [lista de multilongitudes \[CN\]](#)

Este atributo especifica la disposición de los marcos verticales. Es una lista de longitudes en píxeles, porcentajes o longitudes relativas, separadas por comas. El valor por defecto es 100%, que significa una columna.

Atributos definidos en otros lugares

- [id, class](#) (identificadores a nivel de documento)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onload, onunload](#) (eventos intrínsecos)

El elemento [FRAMESET](#) especifica la organización de la ventana principal del usuario en términos de subespacios rectangulares.

Filas y columnas

Cuando se establece el atributo [rows](#) (filas) se define el número de subespacios horizontales. Cuando se establece el atributo [cols](#) (columnas) se define el número de subespacios verticales. Ambos atributos se pueden especificar simultáneamente para crear una cuadrícula.

Si no se establece el atributo [rows](#), cada columna se extiende a lo largo de toda la longitud de la página. Si no se establece el atributo [cols](#), cada fila se extiende a lo largo de toda la anchura de la página. Si no se establece ninguno de los dos atributos, el marco tiene exactamente el mismo tamaño que la página.

Los marcos se crean de izquierda a derecha para las columnas y de arriba a abajo para las filas. Cuando se especifican ambos atributos, las vistas se crean de izquierda a derecha en la fila superior, de izquierda a derecha en la segunda fila, etc.

En este primer ejemplo dividimos la pantalla verticalmente en dos (es decir, creamos una mitad superior y una mitad inferior).

```
<FRAMESET rows="50%, 50%">
...el resto de la definición...
</FRAMESET>
```

El siguiente ejemplo crea tres columnas: la segunda tiene una anchura fija de 250 píxeles (lo cual es útil, por ejemplo, para incluir una imagen de tamaño conocido). La primera recibe el 25% del espacio restante, y la tercera el 75% del espacio restante.

```
<FRAMESET cols="1*,250,3*">
...el resto de la definición...
</FRAMESET>
```

El siguiente ejemplo crea una cuadrícula de 2x3 subespacios.

```
<FRAMESET rows="30%,70%" cols="33%,34%,33%">
...el resto de la definición...
</FRAMESET>
```

Para el siguiente ejemplo, supongamos que la ventana del navegador tiene actualmente una altura de 1000 píxeles. Para la primera vista se asigna el 30% de la altura total (300 píxeles). Para la segunda vista se especifica que tenga una altura de exactamente 400 píxeles. Esto deja 300 píxeles para repartir entre los otros dos marcos. La altura del cuarto marco se ha especificado como "2*", de modo que es el doble de alto que el tercer marco, cuya altura es sólo "*" (equivalente a 1*). Por tanto el tercer marco será de 100 píxeles de alto y el cuarto tendrá una altura de 200 píxeles..

```
<FRAMESET rows="30%,400,*,2*">
...el resto de la definición...
```

```
</FRAMESET>
```

Las longitudes absolutas que no sumen el 100% del espacio real disponible deberían ser ajustadas por los agentes de usuario. Cuando sobre espacio, el espacio sobrante debería repartirse proporcionalmente entre cada vista. Cuando falte espacio, debería reducirse cada vista en función de la relación entre el espacio especificado y el espacio total.

Anidamiento de grupos de marcos

Los grupos de marcos pueden anidarse hasta cualquier nivel.

En el siguiente ejemplo, el [FRAMESET](#) exterior divide el espacio disponible en tres columnas iguales. A continuación el [FRAMESET](#) interior divide la segunda área en dos filas de alturas diferentes.

```
<FRAMESET cols="33%, 33%, 34%">
  ...contenidos del primer marco...
  <FRAMESET rows="40%, 50%">
    ...contenidos del segundo marco, primera fila...
    ...contenidos del segundo marco, segunda fila...
  </FRAMESET>
  ...contenidos del tercer marco...
</FRAMESET>
```

Compartir datos entre marcos

Los autores pueden compartir datos entre varios marcos incluyendo estos datos a través de un elemento [OBJECT](#). Los autores deberían incluir el elemento [OBJECT](#) en el elemento [HEAD](#) del documento con marcos y darle un nombre con el atributo [id](#). Cualquier documento que sea el contenido de un marco del documento con marcos puede hacer referencia a este identificador.

El siguiente ejemplo ilustra cómo podría hacer referencia un script a un elemento [OBJECT](#) definido para todo un grupo de marcos:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
  "http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
<HEAD>
<TITLE>Esto es un documento con marcos con un OBJECT en el HEAD</TITLE>
<!-- ¡Este OBJECT no se representa! -->
<OBJECT id="miobjeto" data="datos.dat"></OBJECT>
</HEAD>
<FRAMESET>
  <FRAME src="blanca.html" name="blanca">
</FRAMESET>
</HTML>
```

```
<!-- En blanca.html -->
<HTML>
<HEAD>
<TITLE>Página de Blanca</TITLE>
</HEAD>
<BODY>
  ...comienzo del documento...
  <P>
  <SCRIPT type="text/javascript">
parent.miobjeto.mipropiedad
  </SCRIPT>
  ...el resto del documento...
</BODY>
</HTML>
```

16.2.2 El elemento FRAME

```

<![ %HTML.Frameset; [
<!-- los nombres de marcos reservados comienzan con "_", los demás con una letra -->
<!ELEMENT FRAME - O EMPTY -- subventana -->
<!ATTLIST FRAME
  %coreattrs; -- id, class, style, title --
  longdesc %URI; #IMPLIED -- vínculo a descripción larga
                                (complementa al título) --
  name CDATA #IMPLIED -- nombre destino del marco --
  src %URI; #IMPLIED -- origen del contenido del marco --
  frameborder (1|0) 1 -- ¿poner bordes al marco? --
  marginwidth %Pixels; #IMPLIED -- anchuras de los márgenes en píxeles --
  marginheight %Pixels; #IMPLIED -- alturas de los márgenes en píxeles --
  noresize (noresize) #IMPLIED -- ¿permitir a los usuarios redimensionar
                                los marcos? --
  scrolling (yes|no|auto) auto -- barra de desplazamiento o no --
>
]]>

```

Definiciones de atributos

name = [CDATA \[CI\]](#)

Este atributo asigna un nombre al marco actual. Este nombre puede utilizarse como el destino de vínculos subsiguientes.

longdesc = [URI \[CT\]](#)

Este atributo especifica un vínculo a una descripción larga del marco. Esta descripción debería complementar la descripción corta proporcionada por el atributo [title](#), y puede ser particularmente útil para agentes de usuario no visuales.

src = [URI \[CT\]](#)

Este atributo especifica la localización de los contenidos iniciales que contendrá el marco.

noresize [\[CI\]](#)

Si está presente, este atributo booleano le dice al agente de usuario que la ventana del marco no debe ser redimensionable.

scrolling = auto|yes|no [\[CI\]](#)

Este atributo especifica información sobre el desplazamiento de la ventana del marco. Valores posibles:

- auto: Este valor le dice al agente de usuario que proporcione mecanismos de desplazamiento en la ventana del marco cuando sea necesario. Este es el valor por defecto.
- yes: Este valor le dice al agente de usuario que siempre proporcione mecanismos de desplazamiento en la ventana del marco.
- no: Este valor le dice al agente de usuario que nunca proporcione mecanismos de desplazamiento en la ventana del marco.

frameborder = 1|0 [\[CN\]](#)

Este atributo proporciona información al agente de usuario sobre el borde del marco. Valores posibles:

- 1: Este valor le dice al agente de usuario que dibuje un separador entre este marco y todos los marcos adyacentes. Este es el valor por defecto.
- 0: Este valor le dice al agente de usuario que no dibuje un separador entre este marco y todos los marcos adyacentes. Obsérvese que aún se puede dibujar un separador junto a este marco si así se especifica para otros marcos.

marginwidth = [píxeles \[CN\]](#)

Este atributo especifica la cantidad de espacio que debe dejarse entre los contenidos del marco en sus márgenes izquierdo y derecho. El valor debe ser mayor o igual que cero (píxeles). El valor por defecto depende del agente de usuario.

marginheight = [píxeles \[CN\]](#)

Este atributo especifica la cantidad de espacio que debe dejarse entre los contenidos del marco en sus márgenes superior

e inferior. El valor debe ser mayor o igual que cero (píxeles). El valor por defecto depende al agente de usuario.

Atributos definidos en otros lugares

- [id, class \(identificadores a nivel de documento\)](#)
- [title \(título del elemento\)](#)
- [style \(información de estilo en línea\)](#)

El elemento [FRAME](#) define los contenidos y la apariencia de un marco dado.

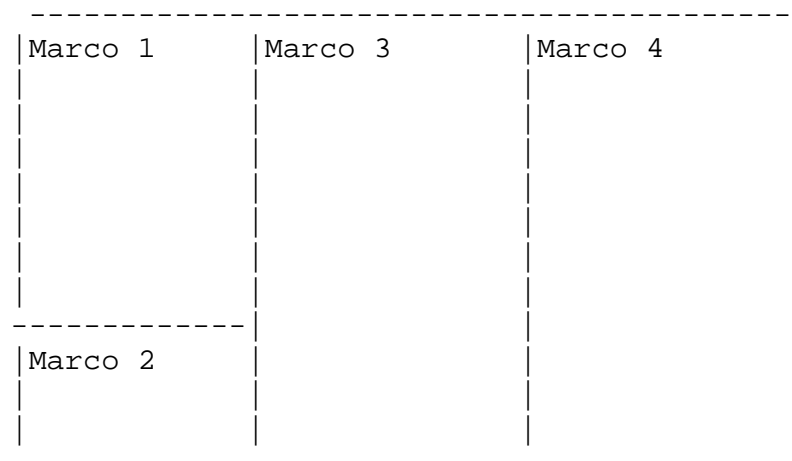
Especificación de los contenidos iniciales de un marco

El atributo [src](#) especifica el documento inicial que contendrá el marco.

El siguiente ejemplo de documento HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
  "http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
<HEAD>
<TITLE>Un documento con marcos</TITLE>
</HEAD>
<FRAMESET cols="33%,33%,33%">
  <FRAMESET rows="*,200">
    <FRAME src="contenidos_del_marco1.html">
    <FRAME src="contenidos_del_marco2.gif">
  </FRAMESET>
  <FRAME src="contenidos_del_marco3.html">
  <FRAME src="contenidos_del_marco4.html">
</FRAMESET>
</HTML>
```

debería crear una distribución de marcos parecida a ésta:



y hacer que el agente de usuario cargara cada fichero en una vista separada.

Los contenidos de un marco no deben estar en el mismo documento que la definición del marco.

EJEMPLO ILEGAL:

La siguiente definición de un grupo de marcos no es legal según HTML, ya que los contenidos del segundo marco están en el mismo documento que la definición del grupo de marcos.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
  "http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
<HEAD>
<TITLE>Un documento con marcos</TITLE>
```

```

</HEAD>
<FRAMESET cols="50%,50%">
  <FRAME src="contenidos_del_marco1.html">
  <FRAME src="#vinculo_al_mismo_documento">
  <NOFRAMES>
  ...texto...
  <H2><A name="vinculo_al_mismo_documento">Sección importante</A></H2>
  ...texto...
  </NOFRAMES>
</FRAMESET>
</HTML>

```

Representación visual de un marco

El siguiente ejemplo ilustra el uso de los atributos decorativos de [FRAME](#). Especificamos que el marco 1 no permitirá barras de desplazamiento. El marco 2 dejará espacio en blanco alrededor de sus contenidos (inicialmente, un fichero de imagen) y el marco no será redimensionable. No se dibujará ningún borde entre los marcos 3 y 4. Se dibujarán los bordes (por defecto) entre los marcos 1, 2 y 3.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
  "http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
<HEAD>
<TITLE>Un documento con marcos</TITLE>
</HEAD>
<FRAMESET cols="33%,33%,33%">
  <FRAMESET rows="*,200">
    <FRAME src="contenidos_del_marco1.html" scrolling="no">
    <FRAME src="contenidos_del_marco2.gif"
      marginwidth="10" marginheight="15"
      noresize>
  </FRAMESET>
  <FRAME src="contenidos_del_marco3.html" frameborder="0">
  <FRAME src="contenidos_del_marco4.html" frameborder="0">
</FRAMESET>
</HTML>

```

16.3 Especificación de información sobre el marco destino

Nota. Para información sobre las prácticas actuales referentes a la determinación del marco destino, consulte las [notas sobre marcos del apéndice](#).

Definiciones de atributos

`target = marco-destino` [\[CI\]](#)

Este atributo especifica el nombre de un marco en el que debe abrirse un documento.

Al asignar un nombre a un marco por medio del atributo [name](#), los autores pueden referirse a él como el "destino" de los vínculos definidos por otros elementos. Se pueden establecer el atributo [target](#) para los elementos que creen vínculos ([A](#), [LINK](#)), para los mapas de imágenes ([AREA](#)), y para los formularios ([FORM](#)).

Consulte la sección sobre [nombres de marcos destino](#) para información sobre nombres de marcos reconocidos.

Este ejemplo ilustra cómo es posible mediante la especificación de un destino la modificación dinámica de los contenidos de un marco. Primero definimos un grupo de marcos en el documento `frameset.html`, mostrado a continuación:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
  "http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
<HEAD>
<TITLE>Un documento con marcos</TITLE>

```

```

</HEAD>
<FRAMESET rows="50%,50%">
  <FRAME name="fijo" src="inicial_fijo.html">
  <FRAME name="dinamico" src="inicial_dinamico.html">
</FRAMESET>
</HTML>

```

Después, en `inicial_dinamico.html`, hacemos un vínculo al marco llamado "dinamico".

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
<TITLE>Un documento con vínculos con destinos específicos</TITLE>
</HEAD>
<BODY>
...comienzo del documento...
<P>Puede avanzar ahora a la
  <A href="diapo2.html" target="dinamico">diapositiva 2.</A>
...más documento...
<P>Lo está haciendo muy bien. Vaya ahora a la
  <A href="diapo3.html" target="dinamico">diapositiva 3.</A>
</BODY>
</HTML>

```

Si se activa cualquiera de los vínculos se abre un nuevo documento en el marco llamado "dinamico", mientras que el otro marco, "fijo", mantiene sus contenidos iniciales.

***Nota.** La definición de un grupo de marcos nunca cambia, pero los contenidos de uno de sus marcos sí pueden cambiar. Una vez que los contenidos de un marco cambian, la definición del grupo de marcos deja de reflejar el estado actual de sus marcos.*

Actualmente no hay ninguna manera de codificar con un URI el estado real de un grupo de marcos. Por tanto, muchos agentes de usuario no permiten a los usuarios asignar un marcador a un grupo de marcos.

Los grupos de marcos hacen más difícil para los usuarios la navegación hacia adelante y hacia atrás por la historia del agente de usuario.

16.3.1 Especificación del destino de los vínculos por defecto

Cuando muchos vínculos del mismo documento designan al mismo destino, es posible especificar el destino una sola vez para que no sea necesario incluir el atributo `target` en todos los elementos. Esto se hace estableciendo el atributo `target` del elemento `BASE`.

Volvemos al ejemplo anterior, esta vez definiendo la información sobre el destino en el elemento `BASE` para quitarla de los elementos `A`.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
<TITLE>Un documento con un destino específico en BASE</TITLE>
<BASE href="http://www.micom.com/Diapos" target="dinamico">
</HEAD>
<BODY>
...comienzo del documento...
<P>Puede avanzar ahora a la <A href="diapo2.html">diapositiva 2.</A>
...más documento...
<P>Lo está haciendo muy bien. Vaya ahora a la
  <A href="diapo3.html">diapositiva 3.</A>
</BODY>
</HTML>

```


16.3.2 Semántica de marcos destino

Los agentes de usuario deberían determinar el marco destino en el que cargar un recurso vinculado de acuerdo con las siguientes reglas de precedencia (ordenadas de mayor a menor prioridad):

1. Si un elemento especifica en su atributo [target](#) un marco conocido, cuando se activa el vínculo (p.ej., se sigue el vínculo o se procesa un formulario), el recurso designado por el elemento debería cargarse en el marco destino.
2. Si un elemento no tiene el atributo [target](#) establecido pero el elemento [BASE](#) sí lo tiene, el atributo [target](#) del elemento [BASE](#) determina el marco.
3. Si ni el elemento que vincula el recurso ni el elemento [BASE](#) hacen referencia a un destino, el recurso designado por el elemento debería cargarse en el marco que contiene al elemento.
4. Si alguno de los atributos [target](#) se refiere a un marco desconocido F, el agente de usuario debería crear una ventana y marco nuevos, asignar el nombre F al marco, y cargar el recurso designado por el elemento en el nuevo marco.

Los agentes de usuario pueden proporcionar a los usuarios un mecanismo para deshabilitar el atributo [target](#).

16.4 Contenido alternativo

Los autores deberían proporcionar contenido alternativo para aquellos agentes de usuario que no soporten marcos o que estén configurados para no mostrar marcos.

16.4.1 El elemento NOFRAMES

```
<![ %HTML.Frameset; [
<!ENTITY % noframes.content "(BODY) -(NOFRAMES)">
]]>

<!ENTITY % noframes.content "(%flow;)*">

<!ELEMENT NOFRAMES - - %noframes.content;
-- contenedor de contenido alternativo para representación no basada en marcos -->
<!ATTLIST NOFRAMES
  %attrs; -- %coreattrs, %il8n, %events --
>
```

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

El elemento [NOFRAMES](#) especifica un contenido que sólo deberían mostrar los agentes de usuario que no soporten marcos o que estén configurados para no mostrar marcos. Los agentes de usuario que soporten marcos sólo deben mostrar los contenidos de una declaración [NOFRAMES](#) cuando se configuren para no mostrar marcos. Los agentes de usuario que no soporten marcos deben mostrar los contenidos de [NOFRAMES](#) en cualquier caso.

El elemento [NOFRAMES](#) es parte tanto del [DTD transicional como del de documentos con marcos](#). En un documento que use el DTD de documentos con marcos, [NOFRAMES](#) se puede usar al final de la sección FRAMESET del documento.

Por ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
```

```
<HEAD>
<TITLE>Un documento con marcos con NOFRAMES</TITLE>
</HEAD>
<FRAMESET cols="50%, 50%">
  <FRAME src="principal.html">
  <FRAME src="tabla_de_contenidos.html">
  <NOFRAMES>
  <P>Aquí puede encontrar la <A href="principal-sinmarcos.html">
    version sin marcos del documento.</A>
  </NOFRAMES>
</FRAMESET>
</HTML>
```

[NOFRAMES](#) se puede usar, por ejemplo, en un documento que sea el origen de un marco y que use el DTD transicional. Esto permite a los autores explicar el propósito del documento en los casos en que éste se vea fuera del grupo de marcos o con un agente de usuario que no soporte marcos.

16.4.2 Descripciones largas de marcos

El atributo [longdesc](#) permite a los autores hacer los documentos con marcos más accesibles a las personas que utilizan agentes de usuario no visuales. Este atributo designa un recurso que proporciona una descripción larga del marco. Los autores deberían recordar que las descripciones largas asociadas con los marcos se refieren al *marco*, y no a los contenidos del marco. Como los contenidos pueden variar con el tiempo, la descripción larga inicial podría ser inapropiada para los contenidos posteriores del marco. En particular, los autores no deberían incluir una imagen como único contenido de un marco.

El siguiente documento con marcos describe dos marcos. El marco izquierdo contiene una tabla de contenidos y el marco derecho contiene inicialmente la imagen de un avestruz:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
<HEAD>
<TITLE>Un documento con marcos mal diseñado</TITLE>
</HEAD>
<FRAMESET cols="20%, 80%">
  <FRAME src="tabla_de_contenidos.html">
  <FRAME src="avestruz.gif" longdesc="desc-avestruz.html">
</FRAMESET>
</HTML>
```

Obsérvese que la imagen ha sido incluida en el marco independientemente de cualquier elemento HTML, de modo que el autor no tiene ninguna manera de especificar un texto alternativo aparte de usar el atributo [longdesc](#). Si los contenidos del marco derecho cambian (p.ej., el usuario selecciona una serpiente de cascabel en la tabla de contenidos), los usuarios no tendrán acceso textual a los nuevos contenidos del marco.

Por tanto, los autores no deberían poner una imagen en un marco directamente. En su lugar, la imagen debería especificarse en un documento HTML independiente, en el cual se podría adjuntar el texto alternativo apropiado:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
<HEAD>
<TITLE>Una documento con marcos bien diseñado</TITLE>
</HEAD>
<FRAMESET cols="20%, 80%">
  <FRAME src="tabla_de_contenidos.html">
  <FRAME src="contenedor_avestruz.html">
</FRAMESET>
</HTML>
```

```
<!-- En contenedor_avestruz.html: -->
```

```
<HTML>
<HEAD>
<TITLE>El avestruz, robusto y veloz</TITLE>
</HEAD>
<P>
<OBJECT data="avestruz.gif" type="image/gif">
¡Estos avestruces seguro que están ricos!
</OBJECT>
</HTML>
```

16.5 Marcos en línea: el elemento IFRAME

```
<!ELEMENT IFRAME - - (%flow;)* -- subventana en línea -->
<!ATTLIST IFRAME
  %coreattrs; -- id, class, style, title --
  longdesc %URI; #IMPLIED -- vínculo a descripción larga
                        (complementa al título) --
  name CDATA #IMPLIED -- nombre destino del marco --
  src %URI; #IMPLIED -- origen del contenido del marco --
  frameborder (1|0) 1 -- ¿poner bordes al marco? --
  marginwidth %Pixels; #IMPLIED -- anchuras de los márgenes en píxeles --
  marginheight %Pixels; #IMPLIED -- alturas de los márgenes en píxeles --
  scrolling (yes|no|auto) auto -- barra de desplazamiento o no --
  align %IAAlign; #IMPLIED -- alineación vertical u horizontal --
  height %Length; #IMPLIED -- altura del marco --
  width %Length; #IMPLIED -- anchura del marco --
>
```

Definiciones de atributos

longdesc = [uri](#) [CT]

Este atributo especifica un vínculo a una descripción larga del marco. Esta descripción debería servir como complemento de la descripción corta que proporciona el atributo [title](#), y es particularmente útil para los agentes de usuario no visuales.

name = [cdata](#) [CI]

Este atributo asigna un nombre al marco actual. Este nombre puede utilizarse como el destino de vínculos subsiguientes.

width = [longitud](#) [CN]

La anchura del marco en línea.

height = [longitud](#) [CN]

La altura del marco en línea.

Atributos definidos en otros lugares

- [id](#), [class](#) ([identificadores a nivel de documento](#))
- [title](#) ([título del elemento](#))
- [style](#) ([información de estilo en línea](#))
- [name](#), [src](#), [frameborder](#), [marginwidth](#), [marginheight](#), [scrolling](#) ([controles y decoraciones de los marcos](#))
- [align](#) ([alineación](#))

El elemento [IFRAME](#) permite a los autores insertar un marco dentro de un bloque de texto. Insertar un marco en línea dentro de una sección de texto es muy similar a insertar un objeto mediante un elemento [OBJECT](#): ambos permiten insertar un documento HTML en medio de otro, ambos pueden alinearse con el texto circundante, etc.

La información a insertar en línea se designa mediante el atributo [src](#) de este elemento. Los *contenidos* del elemento [IFRAME](#), por su parte, sólo deberían ser mostrados por los agentes de usuario que no soporten marcos o que estén configurados para no mostrar marcos.

Para aquellos agentes de usuario que soporten marcos, el siguiente ejemplo colocará un marco en línea rodeado por un borde en medio del texto.

```
<IFRAME src="blabla.html" width="400" height="500"
        scrolling="auto" frameborder="1">
[Su agente de usuario no soporta marcos o está actualmente configurado
para no mostrar marcos. Sin embargo, puede visitar
<A href="blabla.html">el documento relacionado.</A>]
</IFRAME>
```

Los marcos en línea no pueden ser redimensionados (y por lo tanto no tienen un atributo [noresize](#)).

Nota. También puede incluirse un documento HTML dentro de otro documento HTML por medio del elemento [OBJECT](#). Véase la sección sobre [documentos incluidos](#) para más detalles.

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

17 Formularios

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Introducción a los formularios](#)
2. [Controles](#)
 1. [Tipos de controles](#)
3. [El elemento FORM](#)
4. [El elemento INPUT](#)
 1. [Tipos de controles creados con INPUT](#)
 2. [Ejemplos de formularios con controles INPUT](#)
5. [El elemento BUTTON](#)
6. [Los elementos SELECT, OPTGROUP y OPTION](#)
 1. [Opciones preseleccionadas](#)
7. [El elemento TEXTAREA](#)
8. [El elemento ISINDEX](#)
9. [Rótulos](#)
 1. [El elemento LABEL](#)
10. [Añadir estructura a los formularios: los elementos FIELDSET y LEGEND](#)
11. [Dirigir el foco hacia un elemento](#)
 1. [Navegación con tabulador](#)
 2. [Teclas de acceso](#)
12. [Controles deshabilitados y de sólo lectura](#)
 1. [Controles deshabilitados](#)
 2. [Controles de sólo lectura](#)
13. [Envío de formularios](#)
 1. [Método de envío del formulario](#)
 2. [Controles con éxito](#)
 3. [Procesamiento de los datos del formulario](#)
 - [Paso uno: Identificación de los controles con éxito](#)
 - [Paso dos: Construcción del conjunto de datos del formulario](#)
 - [Paso tres: Codificación del conjunto de datos del formulario](#)
 - [Paso cuatro: Envío del conjunto de datos del formulario codificado](#)
 4. [Tipos de contenido de formularios](#)
 - [application/x-www-form-urlencoded](#)
 - [multipart/form-data](#)

17.1 Introducción a los formularios

Un formulario HTML es una sección de un documento que contiene contenido normal, código, elementos especiales llamados [controles](#) (casillas de verificación (checkboxes), radiobotones (radio buttons), menús, etc.), y rótulos (labels) en esos controles. Los usuarios normalmente "completan" un formulario modificando sus controles (introduciendo texto, seleccionando objetos de un menú, etc.), antes de enviar el formulario a un agente para que lo procese (p.ej., a un servidor web, a un servidor de correo, etc.)

Aquí se muestra un ejemplo de un formulario simple que incluye rótulos, radiobotones y botones para reinicializar el formulario o para enviarlo:

```
<FORM action="http://algunsitio.com/prog/usuarionuevo" method="post">
  <P>
    <LABEL for="nombre">Nombre: </LABEL>
      <INPUT type="text" id="nombre"><BR>
    <LABEL for="apellido">Apellido: </LABEL>
      <INPUT type="text" id="apellido"><BR>
    <LABEL for="email">email: </LABEL>
      <INPUT type="text" id="email"><BR>
    <INPUT type="radio" name="sexo" value="Varón"> Varón<BR>
    <INPUT type="radio" name="sexo" value="Mujer"> Mujer<BR>
    <INPUT type="submit" value="Enviar"> <INPUT type="reset">
  </P>
</FORM>
```

Nota. Esta especificación incluye información más detallada sobre formularios en las subsecciones sobre [representación de formularios](#).

17.2 Controles

Los usuarios interactúan con los formularios a través de los llamados controles.

El "nombre de control" de un control viene dado por su atributo name. El "campo de acción" o alcance del atributo name de un control contenido en un elemento [FORM](#) es el elemento [FORM](#).

Cada control tiene tanto un valor inicial como un valor actual, que son ambas cadenas de caracteres. Consulte la definición de cada control para obtener información sobre los valores iniciales y las posibles restricciones que puede imponer cada control sobre sus valores. En general, el "valor inicial" de un control puede especificarse con el atributo value del elemento de control. Sin embargo, el valor inicial de un elemento [TEXTAREA](#) viene dado por sus contenidos, y el valor inicial de un elemento [OBJECT](#) de un formulario está determinado por la implementación del objeto (es decir, se sale fuera del alcance de esta especificación).

El "valor actual" del control se hace en primer lugar igual al valor inicial. A partir de ese momento, el valor actual del control puede ser modificado a través de la interacción con el usuario y mediante [scripts](#).

El valor inicial de un control no cambia. Así, cuando se reinicializa el formulario, el valor actual de cada control se reinicializa a su valor inicial. Si el control no tiene un valor inicial, el efecto de una reinicialización del formulario sobre ese control es indefinido.

Cuando se envía un formulario para su procesamiento, para algunos controles se empareja su nombre con su valor actual, y estas parejas se [envían](#) con el formulario. Aquellos controles cuyas parejas nombre/valor se envían se llaman [controles con éxito](#).

17.2.1 Tipos de controles

HTML define los siguientes tipos de controles:

botones

Los autores pueden crear tres tipos de botones:

- botones de envío (submit buttons): Cuando se activa, un botón de envío [envía un formulario](#). Un formulario puede contener más de un botón de envío.

- botones de reinicialización (reset buttons): Cuando se activa, un botón de reinicialización reinicializa todos los controles a sus [valores iniciales](#).
- botones pulsadores (push buttons): Los botones pulsadores no tienen un comportamiento por defecto. Cada botón pulsador puede tener asociados [scripts en el lado del cliente](#) a través del atributo [event](#) del elemento. Cuando ocurre un evento (p.ej., el usuario aprieta el botón, lo suelta, etc.), se acciona el script asociado.

Los autores deberían especificar el lenguaje de programación del script de un botón pulsador a través de una [declaración de scripts por defecto](#) (con el elemento [META](#)).

Los autores crean botones con el elemento [BUTTON](#) o el elemento [INPUT](#). Consulte las definiciones de estos elementos para más detalles sobre cómo especificar diferentes tipos de botones.

Nota. Los autores deberían observar que el elemento [BUTTON](#) ofrece posibilidades más ricas de representación que el elemento [INPUT](#).

casillas de verificación (checkboxes)

Las casillas de verificación (y los radiobotones) son interruptores de encendido/apagado que pueden ser conmutados por el usuario. Una casilla de verificación está "marcada" cuando se establece el atributo [checked](#) del elemento de control. Cuando se envía un formulario, solamente pueden tener [éxito](#) los controles de casillas de verificación que estén marcadas.

Varias casillas de verificación de un formulario pueden compartir el mismo [nombre de control](#). Así, por ejemplo, las casillas de verificación permiten a los usuarios elegir varios valores para la misma propiedad. Para crear un control de casilla de verificación se utiliza el elemento [INPUT](#).

radiobotones (radio buttons)

Los radiobotones son como las casillas de verificación, excepto en que cuando varios comparten el mismo [nombre de control](#), son mutuamente exclusivos: cuando uno está "encendido", todos los demás con el mismo nombre se "apagan". Para crear un control de radiobotón se usa el elemento [INPUT](#).

Si ninguno de los radiobotones de un conjunto que comparten el mismo nombre tiene un valor inicial de "encendido", el comportamiento del agente de usuario al elegir cuál es el control que está inicialmente "encendido" queda sin definir.

Nota. Debido a que las implementaciones existentes tratan este caso de formas diferentes, la especificación actual difiere de RFC 1866 ([\[RFC1866\]](#) sección 8.1.2.4), que dice:

En cualquier momento, exactamente uno de los radiobotones de un conjunto está marcado. Si ninguno de los elementos <INPUT> de un conjunto de radiobotones especifica 'CHECKED', entonces el agente de usuario debe marcar el primer radiobotón del conjunto inicialmente.

Al diferir los comportamientos de los agentes de usuario, los autores deberían asegurarse de que en cada conjunto de radiobotones hay uno que inicialmente está "encendido".

menús (menus)

Los menús ofrecen al usuario opciones entre las cuales elegir. El elemento [SELECT](#) crea un menú, en combinación con los elementos [OPTGROUP](#) y [OPTION](#).

entrada de texto (text input)

Los autores pueden crear dos tipos de controles que permiten a los usuarios introducir textos. El elemento [INPUT](#) crea un control de entrada de una sola línea, y el elemento [TEXTAREA](#) crea un control de entrada de varias líneas. En ambos casos, el texto introducido se convierte en el [valor actual](#) del control.

selección de ficheros (file select)

Este tipo de control permite al usuario elegir ficheros de modo que sus contenidos puedan ser enviados con un formulario. Se usa el elemento [INPUT](#) para crear un control de selección de ficheros.

controles ocultos (hidden controls)

Los autores pueden crear controles que no se muestran pero cuyos valores se envían con un formulario. Los autores usan normalmente este tipo de controles para almacenar información entre intercambios cliente/servidor que de otro modo se perdería debido a la naturaleza no persistente del protocolo HTTP (ver [\[RFC2616\]](#)). Para crear un control oculto se usa el elemento [INPUT](#).

controles tipo objeto (object controls)

Los autores pueden insertar objetos genéricos en los formularios de modo que los valores asociados se envíen junto con los demás controles. Los autores crean controles tipo objeto con el elemento [OBJECT](#).

Los elementos utilizados para crear controles aparecen normalmente dentro de un elemento [FORM](#), pero también pueden aparecer fuera de la declaración de un elemento [FORM](#) cuando se utilizan para construir interfaces de usuario. Sobre esto se habla en la sección sobre [eventos intrínsecos](#). Obsérvese que los controles que estén fuera de un formulario no pueden ser [controles con éxito](#).

17.3 El elemento FORM

```
<!ELEMENT FORM - - (%block;|SCRIPT)+ -(FORM) -- formulario interactivo -->
<!ATTLIST FORM
  %attrs;                -- %coreattrs, %i18n, %events --
  action      %URI;      #REQUIRED -- procesador del formulario en el servidor --
  method      (GET|POST) GET      -- método HTTP usado para enviar formulario --
  enctype     %ContentType; "application/x-www-form-urlencoded"
  accept      %ContentTypes; #IMPLIED -- lista de tipos MIME para subir ficheros --
  name        CDATA      #IMPLIED -- nombre del formulario para los scripts --
  onsubmit    %Script;   #IMPLIED -- el formulario fue enviado --
  onreset     %Script;   #IMPLIED -- el formulario fue reinicializado --
  accept-charset %Charsets; #IMPLIED -- lista de codif. de caracteres soportadas --
>
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

Definiciones de atributos

action = [uri](#) [CT]

Este atributo especifica un agente procesador de formularios. El comportamiento del agente de usuario frente a un valor diferente de un URI HTTP es indefinido.

method = get|post [CI]

Este atributo especifica qué método HTTP se usará para enviar el [conjunto de datos del formulario](#). Los valores posibles (que no distinguen entre mayúsculas y minúsculas) son "get" (valor por defecto) y "post". Véase la sección sobre [envío de formularios](#) para información sobre su uso.

enctype = [tipo de contenido](#) [CI]

Este atributo especifica el [tipo de contenido](#) usado para enviar el formulario al servidor (cuando el valor del atributo [method](#) sea "post"). El valor por defecto de este atributo es "application/x-www-form-urlencoded". El valor "multipart/form-data" debería usarse en combinación con el elemento [INPUT](#), type="file".

accept-charset = [lista de codificaciones de caracteres](#) [CI]

Este atributo especifica la lista de [codificaciones de caracteres](#) para los datos introducidos que son aceptadas por el servidor que procesa este formulario. El valor es una lista de valores de [codificaciones de caracteres](#) separadas por espacios y/o comas. El cliente debe interpretar esta lista como una lista o-exclusiva, es decir, el servidor es capaz de aceptar cualquier codificación de caracteres individual por entidad recibida.

El valor por defecto de este atributo es la cadena reservada "UNKNOWN" ("desconocido"). Los agentes de usuario pueden interpretar este valor como la codificación de caracteres que fue usada para transmitir el documento que contiene este elemento [FORM](#).

accept = [lista de tipos de contenido](#) [CI]

Este atributo especifica una lista de tipos de contenido separados por comas que un servidor procesador de formularios manejará correctamente. Los agentes de usuario pueden utilizar esta información para filtrar ficheros no conformes cuando pidan al usuario seleccionar ficheros para enviar al servidor (véase el elemento [INPUT](#) cuando [type](#)="file").

name = [cdata](#) [CI]

Este atributo da nombre al elemento de modo que se pueda hacer referencia a él desde hojas de estilo o scripts. **Nota.** Este atributo ha sido incluido por motivos de compatibilidad con versiones anteriores. Las aplicaciones deberían usar el atributo [id](#) para identificar elementos.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [style](#) (información de estilo en línea)
- [title](#) (título del elemento)
- [target](#) (información del marco destino)
- [onsubmit](#), [onreset](#), [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

El elemento [FORM](#) actúa como contenedor de [controles](#). Especifica:

- La disposición ("layout") del formulario (dada por los contenidos del elemento).
- El programa que manejará el formulario completado y enviado (el atributo [action](#)). El programa receptor debe ser capaz de interpretar las parejas nombre/valor para poder hacer uso de ellas.
- El método por el cual se enviarán los datos del usuario al servidor (el atributo [method](#)).
- Una codificación de caracteres que debe ser aceptada por el servidor para poder manejar este formulario (el atributo [accept-charset](#)). Los agentes de usuario pueden avisar al usuario del valor del atributo [accept-charset](#) y/o restringir al usuario la posibilidad de introducir caracteres no reconocidos.

Un formulario puede contener texto y código (párrafos, listas, etc.) además de [controles de formulario](#).

El siguiente ejemplo muestra un formulario que va a ser procesado por el programa "usuariouno" cuando sea enviado. El formulario será enviado al programa usando el método HTTP "post".

```
<FORM action="http://algunsitio.com/prog/usuariouno" method="post">
...contenidos del formulario...
</FORM>
```

Consulte la sección sobre [envío de formularios](#) para información sobre cómo deben preparar los agentes de usuario los datos del formulario para los servidores y cómo deberían tratar los agentes de usuario las respuestas esperadas.

Nota. Quedan fuera del alcance de esta especificación las cuestiones sobre el comportamiento de los servidores que reciben datos de formularios.

17.4 El elemento INPUT

```
<!ENTITY % InputType
" (TEXT | PASSWORD | CHECKBOX |
  RADIO | SUBMIT | RESET |
  FILE | HIDDEN | IMAGE | BUTTON) "
>
```

```
<!-- se requiere el atributo name para todos excepto para submit y reset -->
<!ELEMENT INPUT - O EMPTY -- control de formulario -->
<!ATTLIST INPUT
  %attrs; -- %coreattrs, %i18n, %events --
  type %InputType; TEXT -- qué tipo de control se necesita --
  name CDATA #IMPLIED -- enviar como parte del formulario --
  value CDATA #IMPLIED -- especificar para radiobotones
  y casillas de verificación --
  checked (checked) #IMPLIED -- para radiobotones y casillas de verif. --
  disabled (disabled) #IMPLIED -- no disponible en este contexto --
  readonly (readonly) #IMPLIED -- para texto y contraseñas --
  size CDATA #IMPLIED -- específico de cada tipo de campo --
  maxlength NUMBER #IMPLIED -- máximo de caracteres para campos de texto
--
  src %URI; #IMPLIED -- para campos con imágenes --
```

alt	CDATA	#IMPLIED	-- descripción corta --
usemap	%URI;	#IMPLIED	-- usar mapa de imágenes en el cliente --
ismap	(ismap)	#IMPLIED	-- usar mapa de imágenes en el servidor --
tabindex	NUMBER	#IMPLIED	-- posición en el orden de tabulación --
accesskey	%Character;	#IMPLIED	-- carácter de la tecla de accesibilidad --
onfocus	%Script;	#IMPLIED	-- el foco se dirigió hacia el elemento --
onblur	%Script;	#IMPLIED	-- el elemento perdió el foco --
onselect	%Script;	#IMPLIED	-- se seleccionó parte del texto --
onchange	%Script;	#IMPLIED	-- el valor del elemento fue modificado --
accept	%ContentTypes;	#IMPLIED	-- lista de tipos MIME para subir ficheros --

>

*Etiqueta inicial: **obligatoria**, Etiqueta final: **prohibida***

Definiciones de atributos

type = text|password|checkbox|radio|submit|reset|file|hidden|image|button [\[CI\]](#)

Este atributo especifica el [tipo de control](#) a crear. El valor por defecto para este atributo es "text".

name = [cdata](#) [\[CI\]](#)

Este atributo asigna el [nombre de control](#).

value = [cdata](#) [\[CA\]](#)

Este atributo especifica el [valor inicial](#) del control. Es opcional excepto cuando el atributo [type](#) tenga el valor "radio" o "checkbox".

size = [cdata](#) [\[CN\]](#)

Este atributo le dice al agente de usuario la anchura inicial del control. La anchura viene dada en [píxeles](#) excepto cuando el atributo [type](#) tenga el valor "text" o "password". En estos casos, el valor se refiere al número (entero) de caracteres.

maxlength = [number](#) [\[CN\]](#)

Cuando el atributo [type](#) tiene el valor "text" o "password", este atributo especifica el número máximo de caracteres que puede introducir el usuario. Este número puede exceder del especificado por [size](#), en cuyo caso el agente de usuario debería ofrecer un mecanismo de desplazamiento. El valor por defecto para este atributo es un número ilimitado.

checked [\[CI\]](#)

Cuando el atributo [type](#) tiene el valor "radio" o "checkbox", este atributo booleano especifica que el botón está marcado ("on"). Los agentes de usuario no deben tener en cuenta este atributo para otros tipos de control.

src = [uri](#) [\[CT\]](#)

Cuando el atributo [type](#) tiene el valor "image", este atributo especifica la localización de la imagen que debe usarse para decorar el botón gráfico de envío.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [alt](#) (texto alternativo)
- [align](#) (alineación)
- [accept](#) (tipos de contenidos legales para un servidor)
- [readonly](#) (controles de entrada de sólo lectura)
- [disabled](#) (controles de entrada deshabilitados)
- [tabindex](#) (navegación con tabulador)
- [accesskey](#) (teclas de acceso)
- [usemap](#) (mapas de imágenes en el lado del cliente)
- [ismap](#) (mapas de imágenes en el lado del servidor)

- [onfocus](#), [onblur](#), [onselect](#), [onchange](#), [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

17.4.1 Tipos de controles creados con INPUT

El [tipo de control](#) definido por el elemento [INPUT](#) depende del valor del atributo [type](#):

text

Crea un control de [entrada de texto](#) de una línea.

password

Igual que "text", pero el texto introducido se presentan de tal modo que se oculten los caracteres (p.ej., mediante un serie de asteriscos). Este tipo de control suele utilizarse para datos importantes tales como contraseñas. Obsérvese que el [valor actual](#) es el texto *introducido* por el usuario, no el texto representado por el agente de usuario.

Nota. Los diseñadores de aplicaciones deberían tener en cuenta que este mecanismo ofrece en realidad un nivel muy bajo de seguridad. Aunque la contraseña se oculte a las personas que puedan estar mirando, se transmite al servidor como texto sin enmascarar, y puede ser leído por cualquiera que tenga acceso de bajo nivel a la red.

checkbox

Crea una [casilla de verificación o "checkbox"](#).

radio

Crea un [radiobotón o "radio button"](#).

submit

Crea un [botón de envío o "submit button"](#).

image

Crea un [botón de envío](#) gráfico. El valor del atributo [src](#) especifica el URI de la imagen que decorará el botón. Por razones de accesibilidad, los autores deberían proporcionar un [texto alternativo](#) para la imagen por medio del atributo [alt](#).

Cuando se utiliza un dispositivo apuntador para hacer clic sobre la imagen, se envían al servidor el formulario y la coordenadas en que se pulsó el dispositivo. El valor x se mide en [píxeles](#) desde la izquierda de la imagen, y el valor y en [píxeles](#) desde la parte superior de la imagen. Los datos enviados incluyen *name.x=x-value* y *name.y=y-value* donde "name" es el valor del atributo name, y *x-value* e *y-value* son las coordenadas x e y, respectivamente.

Si el servidor realiza acciones diferentes dependiendo del lugar en que se pulsó el dispositivo, los usuarios de navegadores no gráficos estarán en desventaja. Por esta razón, los autores deberían considerar otras alternativas:

- Usar varios botones de envío (cada uno con su imagen) en lugar de un solo botón gráfico de envío. Los autores pueden usar hojas de estilo para controlar la posición de estos botones.
- Usar un [mapa de imágenes en el lado del cliente](#) junto con scripts.

reset

Crea un [botón de reinicialización o "reset button"](#).

button

Crea un [botón pulsador o "push button"](#). Los agentes de usuario deberían usar el valor del atributo value como rótulo del botón.

hidden

Crea un [control oculto o "hidden control"](#).

file

Crea un control de [selección de fichero o "file select"](#). Los agentes de usuario pueden usar el valor del atributo [value](#) para el nombre de fichero inicial.

17.4.2 Ejemplos de formularios con controles INPUT

El siguiente fragmento HTML de ejemplo define un formulario simple que permite al usuario introducir su nombre, apellido, dirección de correo electrónico y sexo. Cuando se active el botón de envío, el formulario será enviado al programa especificado por el atributo [action](#).

```
<FORM action="http://algunsitio.com/prog/usuarionuevo" method="post">
  <P>
    Nombre: <INPUT type="text" name="nombre"><BR>
    Apellido: <INPUT type="text" name="apellido"><BR>
    email: <INPUT type="text" name="email"><BR>
    <INPUT type="radio" name="sexo" value="Varón"> Varón<BR>
    <INPUT type="radio" name="sexo" value="Mujer"> Mujer<BR>
    <INPUT type="submit" value="Enviar"> <INPUT type="reset">
  </P>
</FORM>
```

Este formulario podría ser representado de la siguiente manera:

En la sección sobre el elemento [LABEL](#), se habla sobre la codificación de rótulos tales como "Nombre".

En el ejemplo que sigue, se ejecuta la función Javascript llamada `verificar` cuando se da el evento "onclick":

```
<HEAD>
<META http-equiv="Content-Script-Type" content="text/javascript">
</HEAD>
<BODY>
  <FORM action="..." method="post">
    <P>
      <INPUT type="button" value="Pínchame" onclick="verificar()">
    </FORM>
  </BODY>
```

Consulte la sección sobre [eventos intrínsecos](#) para más información sobre scripts y eventos.

El siguiente ejemplo muestra cómo pueden enviarse los contenidos de un fichero especificado por el usuario con un formulario. Se le pide al usuario su nombre y una lista de nombres de ficheros cuyos contenidos deberían enviarse con el formulario. Al especificar para [enctype](#) el valor "multipart/form-data", los contenidos de cada fichero se empaquetarán para su envío en una sección separada de un documento multiparte.

```
<FORM action="http://servidor.com/cgi/procesar"
  enctype="multipart/form-data"
  method="post">
  <P>
    ¿Cómo se llama usted? <INPUT type="text" name="nombre_del_remitente">
    ¿Qué ficheros desea enviar? <INPUT type="file" name="nombre_de_los_ficheros">
  </P>
</FORM>
```

17.5 El elemento **BUTTON**

```

<!ELEMENT BUTTON - -
    (%flow;)* - (A|%formctrl;|FORM|FIELDSET)
    -- botón pulsador -->
<!ATTLIST BUTTON
    %attrs; -- %coreattrs, %i18n, %events --
    name          CDATA          #IMPLIED
    value         CDATA          #IMPLIED -- se envía al servidor junto al formulario --
    type         (button|submit|reset) submit -- para usar como botón de formulario --
    disabled     (disabled)      #IMPLIED -- no disponible en este contexto --
    tabindex     NUMBER         #IMPLIED -- posición en el orden de tabulación --
    accesskey    %Character;    #IMPLIED -- carácter de la tecla de accesibilidad --
    onfocus     %Script;        #IMPLIED -- el foco se dirigió hacia el elemento --
    onblur       %Script;        #IMPLIED -- el elemento perdió el foco --
    >

```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

Definiciones de atributos

name = [CDATA](#) [CI]

Este atributo asigna el [nombre de control](#).

value = [CDATA](#) [CS]

Este atributo asigna al botón su [valor inicial](#).

type = submit|button|reset [CI]

Este atributo declara el tipo del botón. Valores posibles:

- submit: Crea un [botón de envío \(submit button\)](#). Este es el valor por defecto.
- reset: Crea un [botón de reinicialización \(reset button\)](#).
- button: Crea un [botón pulsador \(push button\)](#).

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información sobre el estilo en línea)
- [disabled](#) (controles de entrada deshabilitados)
- [accesskey](#) (teclas de acceso)
- [tabindex](#) (navegación con tabulador)
- [onfocus](#), [onblur](#), [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

Los botones creados con el elemento [BUTTON](#) funcionan exactamente igual que los botones creados con el elemento [INPUT](#), pero ofrecen posibilidades más ricas de representación: el elemento [BUTTON](#) puede tener contenido. Por ejemplo, un elemento [BUTTON](#) que contenga una imagen se parece y funciona como un elemento [INPUT](#) cuyo atributo [type](#) sea igual a "image", pero el tipo de elemento [BUTTON](#) permite un contenido.

Los agentes de usuario visuales pueden representar los botones [BUTTON](#) con un relieve y un movimiento arriba/abajo al pulsarlos, mientras que pueden representar los botones [INPUT](#) como imágenes "planas".

El siguiente ejemplo extiende un ejemplo previo, pero creando los botones de [envío](#) y de [reinicialización](#) con [BUTTON](#) en lugar de con [INPUT](#). Los botones contienen imágenes sacadas de elementos [IMG](#).

```
<FORM action="http://algunsitio.com/prog/usuarioNuevo" method="post">
  <P>
    Nombre: <INPUT type="text" name="nombre"><BR>
    Apellido: <INPUT type="text" name="apellido"><BR>
    email: <INPUT type="text" name="email"><BR>
    <INPUT type="radio" name="sexo" value="Varón"> Varón<BR>
    <INPUT type="radio" name="sexo" value="Mujer"> Mujer<BR>
    <BUTTON name="enviar" value="enviar" type="submit">
    Enviar<IMG src="/iconos/ahivaeso.gif" alt="¡Ahí va eso!"></BUTTON>
    <BUTTON name="reiniciar" type="reset">
    Reinicializar<IMG src="/iconos/ayno.gif" alt="¡Ay, no!"></BUTTON>
  </P>
</FORM>
```

Recuérdese que los autores deben proporcionar [texto alternativo](#) para los elementos [IMG](#).

Es ilegal asociar un mapa de imágenes con un [IMG](#) que aparezca como el contenido de un elemento [BUTTON](#).

EJEMPLO ILEGAL:

Lo siguiente no es HTML legal.

```
<BUTTON>
<IMG src="blabla.gif" usemap="...">
</BUTTON>
```

17.6 Los elementos SELECT, OPTGROUP y OPTION

```
<!ELEMENT SELECT - - (OPTGROUP|OPTION)+ -- selector de opciones -->
<!ATTLIST SELECT
  %attrs; -- %coreattrs, %i18n, %events --
  name CDATA #IMPLIED -- nombre del campo --
  size NUMBER #IMPLIED -- filas visibles --
  multiple (multiple) #IMPLIED -- por defecto es selección simple --
  disabled (disabled) #IMPLIED -- no disponible en este contexto --
  tabindex NUMBER #IMPLIED -- posición en el orden de tabulación --
  onfocus %Script; #IMPLIED -- el foco se dirigió hacia el elemento --
  onblur %Script; #IMPLIED -- el elemento perdió el foco --
  onchange %Script; #IMPLIED -- el valor del elemento fue modificado --
>
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

Definiciones de atributos de [SELECT](#)

name = [cdata](#) [CI]

Este atributo asigna el [nombre de control](#).

size = [número](#) [CN]

Si un elemento [SELECT](#) se presenta como una lista con desplazamiento ("scrolled list box"), este atributo especifica el número de filas de la lista que deberían ser visibles al mismo tiempo. No es preciso que los agentes visuales presenten un elemento [SELECT](#) como una lista ("list box"); pueden usar cualquier otro mecanismo, como por ejemplo un menú desplegable ("drop-down menu").

multiple [CI]

Si está activado, este atributo booleano permite selecciones múltiples. Si no está activado, el elemento [SELECT](#) sólo permite selecciones simples.

Atributos definidos en otros lugares

- [id](#), [class](#) ([identificadores a nivel de documento](#))

- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [disabled](#) (controles de entrada deshabilitados)
- [tabindex](#) (navegación con tabulador)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

El elemento [SELECT](#) crea un [menú](#). Cada opción ofrecida por el menú se representa por un elemento [OPTION](#). Un elemento [SELECT](#) debe contener al menos un elemento [OPTION](#).

El elemento [OPTGROUP](#) permite a los autores agrupar opciones lógicamente. Esto es particularmente útil cuando el usuario debe elegir de entre una larga lista de opciones; es más fácil apreciar y recordar grupos de opciones relacionadas que una larga lista de opciones sueltas. En HTML 4, todos los elementos [OPTGROUP](#) deben especificarse directamente dentro de un elemento [SELECT](#) (es decir, no pueden anidarse unos grupos dentro de otros).

17.6.1 Opciones preseleccionadas

Pueden preseleccionarse para el usuario cero o más opciones. Los agentes de usuario deberían determinar qué opciones son preseleccionadas de acuerdo con lo siguiente:

- Si no hay ningún elemento [OPTION](#) con el atributo [selected](#) establecido, el comportamiento del agente de usuario en cuanto a la elección de la opción seleccionada inicialmente queda sin especificar. **Nota.** Debido a que las implementaciones existentes tratan de manera diferente este caso, la especificación actual difiere de RFC 1866 ([RFC1866](#) sección 8.1.3), que dice:

El estado inicial tiene la primera opción seleccionada, a menos que el atributo `SELECTED` esté presente en alguno de los elementos `<OPTION>`.

Al diferir los comportamientos de los agentes de usuario, los autores deberían asegurarse de que todos los menús incluyen una [OPTION](#) preseleccionada por defecto.

- Si un elemento [OPTION](#) tiene el atributo [selected](#) activado, debería estar preseleccionado.
- Si el elemento [SELECT](#) tiene el atributo [multiple](#) activado y hay más de un elemento [OPTION](#) que tenga el atributo [selected](#) activado, deberían estar todos preseleccionados.
- Se considera un error que haya más de un elemento [OPTION](#) con el atributo [selected](#) activado y que el elemento [SELECT](#) no tenga el atributo [multiple](#) establecido. Los agentes de usuario pueden tratar este caso de maneras diferentes, pero no deberían preseleccionar más de una opción.

```
<!ELEMENT OPTGROUP - - (OPTION)+ -- grupo de opciones -->
<!ATTLIST OPTGROUP
  %attrs;                -- %coreattrs, %i18n, %events --
  disabled (disabled)   #IMPLIED -- no disponible en este contexto --
  label %Text;          #REQUIRED -- para usar en menús jerárquicos --
>
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria***

Definiciones de atributos de OPTGROUP

label = *texto* [CS]

Este atributo especifica el rótulo del grupo de opciones.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)

- [disabled](#) (controles de entrada deshabilitados)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

Nota. Se avisa a los implementadores de que las versiones futuras de HTML pueden extender el mecanismo de agrupamiento para permitir grupos anidados (es decir, para que los elementos [OPTGROUP](#) puedan anidarse). Esto permitirá a los autores representar una jerarquía de opciones más rica.

```
<!ELEMENT OPTION - O (#PCDATA)          -- opción seleccionable -->
<!ATTLIST OPTION
  %attrs;                                -- %coreattrs, %i18n, %events --
  selected (selected) #IMPLIED          -- opción preseleccionada --
  disabled (disabled) #IMPLIED          -- no disponible en este contexto --
  label %Text; #IMPLIED                 -- para usar en menús jerárquicos --
  value CDATA #IMPLIED                 -- por defecto es el contenido del elemento --
>
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

Definiciones de atributos de [OPTION](#)

selected [\[CI\]](#)

Si está establecido, este atributo booleano especifica que esta opción está preseleccionada.

value = [cdata](#) [\[CS\]](#)

Este atributo especifica el [valor inicial](#) del control. Si este atributo no está establecido, el [valor inicial](#) es igual a los contenidos del elemento [OPTION](#).

label = [texto](#) [\[CS\]](#)

Este atributo permite a los autores especificar un rótulo para la opción más corto que el contenido del elemento [OPTION](#). Cuando esté especificado, los agentes de usuario deberían usar como rótulo de la opción el valor de este atributo en lugar del contenido del elemento [OPTION](#).

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [disabled](#) (controles de entrada deshabilitados)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

Cuando se represente una opción de un menú, los agentes de usuario deberían usar como opción el valor del atributo [label](#) del elemento [OPTION](#). Si este atributo no está especificado, los agentes de usuario deberían usar los contenidos del elemento [OPTION](#).

El atributo [label](#) del elemento [OPTGROUP](#) especifica el rótulo de un grupo de opciones.

En este ejemplo, creamos un menú que permite al usuario seleccionar cuál de los siete componentes de un programa instalar. Los dos primeros componentes están preseleccionados pero pueden ser deseleccionados por el usuario. El resto de los componentes no están preseleccionados. El atributo [size](#) dice que el menú sólo debería tener cuatro filas, aunque el usuario pueda elegir entre siete opciones. Las otras opciones deberían estar disponibles a través de un mecanismo de desplazamiento.

El elemento [SELECT](#) va seguido de botones de envío y de reinicialización.

```
<FORM action="http://algunsitio.com/prog/elegir-componente" method="post">
  <P>
    <SELECT multiple size="4" name="elegir-componente">
      <OPTION selected value="Componente_1_a">Componente_1</OPTION>
```



```

<OPTION selected value="Componente_1_b">Componente_2</OPTION>
<OPTION>Componente_3</OPTION>
<OPTION>Componente_4</OPTION>
<OPTION>Componente_5</OPTION>
<OPTION>Componente_6</OPTION>
<OPTION>Componente_7</OPTION>
</SELECT>
<INPUT type="submit" value="Enviar"><INPUT type="reset">
</P>
</FORM>

```

Solamente las opciones seleccionadas [tendrán éxito](#) (usando el [nombre de control](#) "elegir-componente"). Cuando no haya opciones seleccionadas, el control no tendrá éxito, y ni el nombre ni ninguno de los valores se enviarán al servidor cuando se envíe el formulario. Obsérvese que el atributo [value](#), cuando está establecido, determina el [valor inicial](#) del control, que de otro modo es el contenido del elemento.

En este ejemplo usamos el elemento [OPTGROUP](#) para agrupar opciones. El siguiente código:

```

<FORM action="http://algunsitio.com/prog/algunprograma" method="post">
<P>
<SELECT name="ComOS">
  <OPTION selected label="ninguno" value="ninguno">Ninguno</OPTION>
  <OPTGROUP label="PortMaster 3">
    <OPTION label="3.7.1" value="pm3_3.7.1">PortMaster 3 con ComOS 3.7.1</OPTION>
    <OPTION label="3.7" value="pm3_3.7">PortMaster 3 con ComOS 3.7</OPTION>
    <OPTION label="3.5" value="pm3_3.5">PortMaster 3 con ComOS 3.5</OPTION>
  </OPTGROUP>
  <OPTGROUP label="PortMaster 2">
    <OPTION label="3.7" value="pm2_3.7">PortMaster 2 con ComOS 3.7</OPTION>
    <OPTION label="3.5" value="pm2_3.5">PortMaster 2 con ComOS 3.5</OPTION>
  </OPTGROUP>
  <OPTGROUP label="IRX">
    <OPTION label="3.7R" value="IRX_3.7R">IRX con ComOS 3.7R</OPTION>
    <OPTION label="3.5R" value="IRX_3.5R">IRX con ComOS 3.5R</OPTION>
  </OPTGROUP>
</SELECT>
</FORM>

```

representa el siguiente agrupamiento:

```

Ninguno
PortMaster 3
  3.7.1
  3.7
  3.5
PortMaster 2
  3.7
  3.5
IRX
  3.7R
  3.5R

```

Los agentes de usuario visuales pueden permitir a los usuarios seleccionar de entre grupos de opciones a través de un menú jerárquico o de algún otro mecanismo que refleje la estructura de las opciones.

Un agente de usuario gráfico podría representarlo así::



Esta imagen muestra un elemento [SELECT](#) representado mediante menús en cascada. El rótulo superior del menú muestra el valor seleccionado actualmente (PortMaster 3, 3.7.1). El usuario ha desplegado dos menús en cascada, pero aún no ha seleccionado el nuevo valor (PortMaster 2, 3.7). Obsérvese que cada menú en cascada muestra el rótulo de un elemento [OPTGROUP](#) u [OPTION](#).

17.7 El elemento TEXTAREA

```
<!ELEMENT TEXTAREA - - (#PCDATA) -- campo de texto multilínea -->
<!ATTLIST TEXTAREA
  %attrs; -- %coreattrs, %il8n, %events --
  name CDATA #IMPLIED
  rows NUMBER #REQUIRED
  cols NUMBER #REQUIRED
  disabled (disabled) #IMPLIED -- no disponible en este contexto --
  readonly (readonly) #IMPLIED
  tabindex NUMBER #IMPLIED -- posición en el orden de tabulación --
  accesskey %Character; #IMPLIED -- carácter de la tecla de accesibilidad --
  onfocus %Script; #IMPLIED -- el foco se dirigió hacia el elemento --
  onblur %Script; #IMPLIED -- el elemento perdió el foco --
  onselect %Script; #IMPLIED -- se seleccionó parte del texto --
  onchange %Script; #IMPLIED -- el valor del elemento fue modificado --
>
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria***

Definiciones de atributos

name = [CDATA](#) [CI]

Este atributo asigna el [nombre de control](#).

rows = [número](#) [CN]

Este atributo especifica el número de líneas de texto visibles. Los usuarios deberían poder introducir un número mayor de líneas, por lo que los agentes de usuario deberían proporcionar medios para desplazar los contenidos del control cuando los contenidos se extiendan más allá del área visible.

cols = [número](#) [CN]

Este atributo especifica la anchura visible en caracteres de anchura media. Los usuarios deberían poder introducir líneas de mayor longitud, por lo que los agentes de usuario deberían proporcionar medios para desplazar los contenidos del control cuando los contenidos se extiendan más allá del área visible. Los agentes del usuario pueden partir las líneas de texto visible para que las líneas largas puedan verse sin necesidad de desplazarlas.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [readonly](#) (controles de entrada de sólo lectura)
- [disabled](#) (controles de entrada deshabilitados)
- [tabindex](#) (navegación con tabulador)

- [onfocus](#), [onblur](#), [onselect](#), [onchange](#), [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

El elemento [TEXTAREA](#) crea un control de [entrada de texto](#) multilinea. Los agentes de usuario deberían usar los contenidos de este elemento como [valor inicial](#) del control y representar este texto inicialmente.

Este ejemplo crea un control [TEXTAREA](#) de 20 filas por 80 columnas que contiene inicialmente dos líneas de texto. El elemento [TEXTAREA](#) va seguido de botones de envío y reinicialización.

```
<FORM action="http://algunsitio.com/prog/leer-texto" method="post">
  <P>
    <TEXTAREA name="eltexto" rows="20" cols="80">
      Primera línea del texto inicial.
      Segunda línea del texto inicial.
    </TEXTAREA>
    <INPUT type="submit" value="Enviar"><INPUT type="reset">
  </P>
</FORM>
```

Estableciendo el atributo [readonly](#) los autores pueden mostrar texto no modificable en un elemento [TEXTAREA](#). Esto no es lo mismo que usar código de texto estándar, ya que el valor del [TEXTAREA](#) se envía con el formulario.

17.8 El elemento ISINDEX

ISINDEX está [desaprobado](#). Este elemento crea un control de [entrada de texto](#) de una línea. Los autores debería usar el elemento [INPUT](#) para crear controles de [entrada de texto](#).

Véase el [DTD Transicional](#) para la definición formal.

Definiciones de atributos

prompt = *texto* [CS]

Desaprobado. Este atributo especifica una cadena indicadora ("prompt") para el campo de entrada.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)

El elemento [ISINDEX](#) crea un control de [entrada de texto](#) de una línea que permite cualquier número de caracteres. Los agentes de usuario pueden usar el valor del atributo [prompt](#) como cadena indicadora ("prompt").

EJEMPLO DESAPROBADO:

La siguiente declaración [ISINDEX](#):

```
<ISINDEX prompt="Escriba la frase a buscar: ">
```

podría reescribirse con [INPUT](#) del siguiente modo:

```
<FORM action="..." method="post">
<P>Entre la frase a buscar: <INPUT type="text"></P>
</FORM>
```

Semántica de ISINDEX. Actualmente, la semántica de [ISINDEX](#) sólo está bien definida cuando el URI base del documento que lo contiene es un URI HTTP. En la práctica, la cadena introducida está restringida a Latin-1 ya que no hay ningún mecanismo con el que el URI pueda especificar un juego de caracteres diferente.

17.9 Rótulos

A algunos controles de formulario se les asocian rótulos automáticamente (botones pulsadores) aunque a la mayoría no (campos de texto, casillas de verificación y radiobotones, y menús).

Para aquellos controles que tengan rótulos implícitos, los agentes de usuario deberían utilizar el valor del atributo `value` como texto del rótulo.

El elemento [LABEL](#) se utiliza para especificar rótulos de controles que no tienen rótulos implícitos.

17.9.1 El elemento LABEL

```
<!ELEMENT LABEL - - (%inline;)* -(LABEL) -- texto del rótulo de un
                                campo de formulario -->
<!ATTLIST LABEL
  %attrs;                -- %coreattrs, %il8n, %events --
  for                    IDREF          #IMPLIED -- concuerda con el valor ID del campo --
  accesskey              %Character;    #IMPLIED -- carácter de tecla de accesibilidad --
  onfocus               %Script;       #IMPLIED -- el foco se dirigió hacia el elemento --
  onblur                 %Script;       #IMPLIED -- el elemento perdió el foco --
  >
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria***

Definiciones de atributos

for = [idref](#) [CS]

Este atributo asocia explícitamente el rótulo definido con otro control. Cuando está presente, el valor de este atributo debe ser el mismo que el valor del atributo [id](#) de algún otro control del mismo documento. Cuando no está presente, el rótulo definido se asocia con los contenidos del elemento.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [accesskey](#) (teclas de acceso)
- [onfocus](#), [onblur](#), [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

El elemento [LABEL](#) puede utilizarse para adjuntar información a los controles. Cada elemento [LABEL](#) se asocia exactamente con un control de formulario.

El atributo [for](#) asocia explícitamente un rótulo con otro control: el valor del atributo [for](#) debe ser el mismo que el valor del atributo [id](#) del elemento de control asociado. Se puede asociar más de un [LABEL](#) con el mismo control creando múltiples referencias a través del atributo [for](#).

Este ejemplo crea una tabla que se utiliza para alinear dos controles de [entrada de texto](#) y sus rótulos asociados. Cada rótulo está explícitamente asociado a un control de [entrada de texto](#):

```
<FORM action="..." method="post">
<TABLE>
  <TR>
    <TD><LABEL for="nombre">Nombre</LABEL>
    <TD><INPUT type="text" name="nombre" id="nombre">
  <TR>
    <TD><LABEL for="apellido">Apellido</LABEL>
    <TD><INPUT type="text" name="apellido" id="apellido">
```

```
</TABLE>
</FORM>
```

Este ejemplo extiende el formulario de un ejemplo previo haciendo que incluya elementos [LABEL](#).

```
<FORM action="http://algunsitio.com/prog/usuarioNuevo" method="post">
  <P>
    <LABEL for="nombre">Nombre: </LABEL>
      <INPUT type="text" id="nombre"><BR>
    <LABEL for="apellido">Last name: </LABEL>
      <INPUT type="text" id="apellido"><BR>
    <LABEL for="email">email: </LABEL>
      <INPUT type="text" id="email"><BR>
    <INPUT type="radio" name="sexo" value="Varón"> Varón<BR>
    <INPUT type="radio" name="sexo" value="Mujer"> Mujer<BR>
    <INPUT type="submit" value="Enviar"> <INPUT type="reset">
  </P>
</FORM>
```

Para asociar implícitamente un rótulo con otro control, el elemento de control debe estar dentro de los contenidos del elemento [LABEL](#). En este caso, el [LABEL](#) sólo puede contener un elemento de control. El rótulo en sí puede colocarse antes o después del control asociado.

En este ejemplo, asociamos implícitamente dos rótulos a dos controles de [entrada de texto](#):

```
<FORM action="..." method="post">
<P>
<LABEL>
  Nombre
  <INPUT type="text" name="nombre">
</LABEL>
<LABEL>
  <INPUT type="text" name="apellido">
  Apellido
</LABEL>
</P>
</FORM>
```

Obsérvese que esta técnica no puede utilizarse cuando se usa una tabla para fijar la disposición de los elementos, con el rótulo en una celda y su control asociado en otra celda.

Cuando el [foco](#) se dirige hacia un elemento [LABEL](#), éste pasa el foco a su control asociado. Véanse más adelante en la sección sobre [teclas de acceso](#) algunos ejemplos.

Los rótulos pueden ser representados por los agentes de usuario de diferentes maneras (p.ej., visualmente, leídos por sintetizadores de voz, etc.)

17.10 Añadir estructura a los formularios: los elementos FIELDSET y LEGEND

```
<!--
  #PCDATA es para resolver el problema del contenido mixto,
  de acuerdo con la especificación aquí sólo se permite espacio en blanco!
-->
<!ELEMENT FIELDSET - - (#PCDATA,LEGEND,(%flowi)*) -- grupo de controles
                                     de formulario -->
<!ATTLIST FIELDSET
  %attrs;                               -- %coreattrs, %il8n, %events --
>
```

```
<!ELEMENT LEGEND - - (%inline;)* -- leyenda del grupo de campos -->
<!ATTLIST LEGEND
  %attrs; -- %coreattrs, %il8n, %events --
  accesskey %Character; #IMPLIED -- carácter de la tecla de accesibilidad --
  >
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

Definiciones de atributos de LEGEND

align = top|bottom|left|right [CI]

Desaprobado. Este atributo especifica la posición de la leyenda con respecto al grupo de campos (fieldset). Valores posibles:

- top: La leyenda está en la parte superior del grupo de campos. Este es el valor por defecto.
- bottom: La leyenda está en la parte inferior del grupo de campos.
- left: La leyenda está a la izquierda del grupo de campos.
- right: La leyenda está a la derecha del grupo de campos.

Atributos definidos en otros lugares

- [id](#), [class](#) (identificadores a nivel de documento)
- [lang](#) (información sobre el idioma), [dir](#) (dirección del texto)
- [title](#) (título del elemento)
- [style](#) (información de estilo en línea)
- [accesskey](#) (teclas de acceso)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (eventos intrínsecos)

El elemento [FIELDSET](#) (grupo de campos) permite a los autores agrupar temáticamente controles y rótulos relacionados. Gracias al agrupamiento de controles es más fácil para los usuarios entender su propósito y al mismo tiempo se facilita la navegación con agentes de usuario visuales y la navegación por voz para agentes de usuario basados en voz. El uso correcto de este elemento hace los documentos más accesibles.

El elemento [LEGEND](#) permite a los autores asignar un título a un [FIELDSET](#). La leyenda mejora la accesibilidad cuando el [FIELDSET](#) no se representa visualmente.

En este ejemplo, creamos un formulario que podría rellenarse en una consulta médica. Se divide en tres secciones: información personal, historial médico, y medicación actual. Cada sección contiene controles para introducir la información apropiada.

```
<FORM action="..." method="post">
  <P>
  <FIELDSET>
    <LEGEND>Información Personal</LEGEND>
    Apellido: <INPUT name="personal_apellido" type="text" tabindex="1">
    Nombre: <INPUT name="personal_nombre" type="text" tabindex="2">
    Dirección: <INPUT name="personal_dirección" type="text" tabindex="3">
    ...más información personal...
  </FIELDSET>
  <FIELDSET>
    <LEGEND>Historial Médico</LEGEND>
    <INPUT name="historial_enfermedades"
      type="checkbox"
      value="Viruela" tabindex="20"> Viruela
    <INPUT name="historial_enfermedades"
      type="checkbox"
      value="Paperas" tabindex="21"> Paperas
    <INPUT name="historial_enfermedades"
      type="checkbox"
```

```

        value="Mareos" tabindex="22"> Mareos
<INPUT name="historial_enfermedades"
        type="checkbox"
        value="Resfriado" tabindex="23"> Resfriado
...más historial médico...
</FIELDSET>
<FIELDSET>
<LEGEND>Medicación Actual</LEGEND>
¿Está tomando actualmente algún tipo de medicación?
<INPUT name="medicacion_ahora"
        type="radio"
        value="Sí" tabindex="35">Sí
<INPUT name="medicacion_ahora"
        type="radio"
        value="No" tabindex="35">No

Si está tomando actualmente algún tipo de medicación, indíquela
en el espacio proporcionado a continuación:
<TEXTAREA name="medicacion_actual"
        rows="20" cols="50"
        tabindex="40">
</TEXTAREA>
</FIELDSET>
</FORM>

```

Obsérvese que en este ejemplo podríamos mejorar la presentación visual del formulario alineando los elementos dentro de cada [FIELDSET](#) (con hojas de estilo), añadiendo información de fuentes y colores (con hojas de estilo), añadiendo scripts (por ejemplo, para abrir el área de texto de "medicación actual" sólo si el usuario indica que actualmente está tomando medicación), etc.

17.11 Dirigir el foco hacia un elemento

En un documento HTML, el usuario debe dirigir el foco hacia un elemento para que éste se active y realice sus funciones. Por ejemplo, los usuarios deben activar un vínculo especificado con el elemento [A](#) para seguir el vínculo especificado. Análogamente, los usuarios deben dirigir el foco hacia un [TEXTAREA](#) para poder introducir texto en su interior.

Hay varias maneras de dirigir el foco hacia un elemento:

- Designar el elemento con un dispositivo apuntador.
- Navegar de un elemento a otro con el teclado. El autor del documento puede definir un *orden de tabulación* que especifique el orden en que se dirige el foco hacia los elementos si el usuario navega por el documento usando el teclado (ver [navegación con tabulador](#)). Una vez seleccionado, el elemento puede activarse con alguna otra secuencia de teclas.
- Seleccionar un elemento por medio de una [tecla de acceso](#) (a veces llamada "acelerador de teclado").

17.11.1 Navegación con tabulador

Definiciones de atributos

tabindex = [número](#) [[CN](#)]

Este atributo especifica la posición del elemento actual dentro del orden de tabulación del documento actual. Este valor debe ser un número entre 0 y 32767. Los agentes de usuario no deberían tener en cuenta los ceros a la izquierda.

El orden de tabulación define el orden en que el foco se dirige hacia los elementos cuando se navega por medio del teclado. El orden de tabulación puede incluir elementos anidados en otros elementos.

Los agentes de usuario deberían navegar por los elementos a los que puede dirigirse el foco de acuerdo con las siguientes reglas:

1. Navegar en primer lugar por aquellos elementos que soporten el atributo [tabindex](#) y tengan asignado para éste un valor positivo. La navegación se produce desde el elemento con menor valor de [tabindex](#) hasta el elemento con el valor más alto. Los valores no necesitan ser secuenciales ni deben comenzar por un valor en particular. Si hay elementos que tengan valores idénticos de [tabindex](#) debería navegarse por ellos según el orden en que aparezcan en el flujo de

caracteres.

2. A continuación se navega por aquellos elementos que no soporten el atributo [tabindex](#) o por los que soportándolo tengan asignado para él un valor "0". Se navega por estos elementos según el orden en que aparezcan en el flujo de caracteres.
3. Los elementos que estén [deshabilitados](#) no participan en el orden de tabulación.

Los siguientes elementos soportan el atributo [tabindex](#): [A](#), [AREA](#), [BUTTON](#), [INPUT](#), [OBJECT](#), [SELECT](#) y [TEXTAREA](#).

En este ejemplo, el orden de tabulación será el [BUTTON](#), los elementos [INPUT](#) en orden (obsérvese que "campo1" y el botón comparten el mismo tabindex, pero "campo1" aparece después en el flujo de caracteres), y por último el vínculo creado por el elemento [A](#).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE>Un documento con FORM</TITLE>
</HEAD>
<BODY>
...texto...
<P>Ir al
<A tabindex="10" href="http://www.w3.org/">sitio web del W3C.</A>
...más texto...
<BUTTON type="button" name="obtener-base-de-datos"
        tabindex="1" onclick="obtener-base-de-datos">
Obtener la base de datos acutal.
</BUTTON>
...más texto...
<FORM action="..." method="post">
<P>
<INPUT tabindex="1" type="text" name="campo1">
<INPUT tabindex="2" type="text" name="campo2">
<INPUT tabindex="3" type="submit" name="enviar">
</P>
</FORM>
</BODY>
</HTML>
```

Teclas de tabulación. La secuencia real de teclas que permiten la navegación con tabulador o la activación de los elementos depende de la configuración del agente de usuario (p.ej., la tecla "tab" se usa para la navegación y la tecla "intro" se usa para activar el elemento seleccionado).

Los agentes de usuario pueden también definir secuencias de teclas para navegar en orden inverso al de tabulación. Cuando se alcanza el final (o el principio) del orden de tabulación, los agentes de usuario puede volver al principio (o al final).

17.11.2 Teclas de acceso

Definiciones de atributos

`accesskey` = [carácter](#) [CN]

Este atributo asigna una tecla de acceso a un elemento. Una tecla de acceso es un carácter único del conjunto de caracteres del documento. **Nota.** Los autores deberían considerar la forma que tienen los usuarios potenciales de interactuar a la hora de especificar la tecla de acceso.

Al pulsar la tecla de acceso asignada a un elemento, el foco se dirige hacia el elemento. La acción que tiene lugar cuando el foco se dirige hacia un elemento depende del elemento. Por ejemplo, cuando un usuario activa un vínculo definido por el elemento [A](#), el agente de usuario normalmente sigue el vínculo. Cuando un usuario activa un radiobotón, el agente de usuario cambia el valor del radiobotón. Cuando el usuario activa un campo de texto, éste permite la entrada de texto, etc.

Los siguientes elementos soportan el atributo [accesskey](#): [A](#), [AREA](#), [BUTTON](#), [INPUT](#), [LABEL](#), [LEGEND](#) y [TEXTAREA](#).

Este ejemplo asigna la tecla de acceso "U" a un rótulo asociado con un control [INPUT](#). Al pulsar la tecla de acceso, el foco se dirige hacia el rótulo, el cual a su vez lo dirige al control asociado. El usuario puede entonces introducir texto en el área

INPUT.

```
<FORM action="..." method="post">
<P>
<LABEL for="nombre-usuario" accesskey="U">
User Name
</LABEL>
<INPUT type="text" name="usuario" id="nombre-usuario">
</P>
</FORM>
```

En este ejemplo, asignamos una tecla de acceso a un vínculo definido por el elemento [A](#). Al pulsar esta tecla de acceso, el usuario es llevado a otro documento, en este caso una tabla de contenidos.

```
<P><A accesskey="C"
    rel="contents"
    href="http://algunsitio.com/especificacion/contenidos.html">
Tabla de Contenidos</A>
```

La invocación de teclas de acceso depende del sistema subyacente. Por ejemplo, en máquinas que ejecuten MS Windows, normalmente hay que pulsar la tecla "alt" además de la tecla de acceso. En sistemas Apple, normalmente hay que pulsar la tecla "cmd" además de la tecla de acceso.

La representación de teclas de acceso depende del agente de usuario. Recomendamos a los autores que incluyan la tecla de acceso en el texto del rótulo o dondequiera que se aplique la tecla de acceso. Los agentes de usuario deberían representar las teclas de acceso de tal modo que se enfatice su papel y se distinga de otros caracteres (p.ej., subrayándola).

17.12 Controles deshabilitados y de sólo lectura

En aquellos contextos en los que la entrada de datos por parte del usuario sea indeseable o irrelevante, es importante poder deshabilitar un control o convertirlo en un control de sólo lectura. Por ejemplo, podríamos querer deshabilitar el botón de envío de un formulario mientras que el usuario no haya introducido ciertos datos obligatorios. Análogamente, un autor podría querer incluir una sección de texto de sólo lectura que debería ser enviada como un valor junto con el formulario. Las siguientes secciones describen los controles deshabilitados y de sólo lectura.

17.12.1 Controles deshabilitados

Definiciones de atributos

disabled [\[CI\]](#)

Cuando se establece para un control de formulario, este atributo booleano deshabilita el control para la entrada de datos por parte del usuario.

Cuando está establecido, el atributo [disabled](#) tiene los siguientes efectos sobre un elemento:

- No se puede dirigir el [foco](#) hacia controles deshabilitados.
- En el [orden de tabulación](#), se salta por encima de los controles deshabilitados.
- Los controles deshabilitados no pueden [tener éxito](#).

Los siguientes elementos soportan el atributo [disabled](#): [BUTTON](#), [INPUT](#), [OPTGROUP](#), [OPTION](#), [SELECT](#) y [TEXTAREA](#).

Este atributo se hereda, pero las declaraciones locales prevalecen sobre el valor heredado.

El modo en que se representan los elementos deshabilitados depende del agente de usuario. Por ejemplo, algunos agentes de usuario dibujan en gris los objetos de menú deshabilitados, los rótulos de los botones, etc.

En este ejemplo, el elemento [INPUT](#) está deshabilitado. Por tanto, no puede recibir datos del usuario, y su valor no se enviará con el formulario.

```
<INPUT disabled name="pedro" value="piedra">
```

Nota. El único modo de modificar dinámicamente el valor del atributo [disabled](#) es por medio de un [script](#).

17.12.2 Controles de sólo lectura

Definiciones de atributos

readonly [\[CI\]](#)

Cuando está establecido para un control de formulario, este atributo booleano impide que haya cambios en el control.

El atributo [readonly](#) especifica si el control puede ser modificado por el usuario.

Cuando está establecido, el atributo [readonly](#) tiene los siguientes efectos sobre un elemento:

- El [foco](#) puede dirigirse hacia elementos de sólo lectura, pero éstos no pueden ser modificados por el usuario.
- Los elementos de sólo lectura están incluidos en la [navegación con tabulador](#).
- Los elementos de sólo lectura pueden [tener éxito](#).

Los siguientes elementos soportan el atributo [readonly](#): [INPUT](#) y [TEXTAREA](#).

El modo en que se representan los elementos de sólo lectura depende del agente de usuario.

Nota. La única manera de modificar dinámicamente el valor del atributo [readonly](#) es mediante un [script](#).

17.13 Envío de formularios

Las siguientes secciones explican cómo envían los agentes de usuario los datos de los formularios a los agentes procesadores de formularios.

17.13.1 Método de envío del formulario

El atributo [method](#) del elemento [FORM](#) especifica el método HTTP usado para enviar el formulario al agente procesador. Este atributo puede tener dos valores:

- [get](#): Con el método HTTP "get", el [conjunto de datos del formulario](#) se agrega al URI especificado por el atributo [action](#) (con un signo de interrogación ("?") como separador) y este nuevo URI se envía al agente procesador.
- [post](#): Con el método HTTP "post", el [conjunto de datos del formulario](#) se incluye en el cuerpo del formulario y se envía al agente procesador.

El método "get" debería usarse cuando el formulario es idempotente (es decir, cuando no tiene efectos secundarios). Muchas búsquedas en bases de datos no tienen efectos secundarios visibles y constituyen aplicaciones ideales del método "get".

Si el servicio asociado con el procesamiento de un formulario causa efectos secundario (por ejemplo, si el formulario modifica una base de datos o la suscripción a un servicio), debería usarse el método "post".

Nota. El método "get" restringe los valores del [conjunto de datos del formulario](#) a caracteres ASCII. Sólo el método "post" (con [enctype](#)="multipart/form-data") cubre el conjunto de caracteres [\[ISO10646\]](#) completo.

17.13.2 Controles con éxito

Un control con éxito es "válido" para su envío. Todos los controles con éxito tienen su [nombre de control](#) emparejado con su [valor actual](#) como parte del [conjunto de datos del formulario](#) enviado. Un control con éxito debe estar definido dentro de un elemento [FORM](#) y debe tener un [nombre de control](#).

Sin embargo:

- Los controles que están [deshabilitados](#) no pueden tener éxito.
- Si un formulario contiene más de un [botón de envío](#), solamente el botón de envío activado tiene éxito.
- Todas las [casillas de verificación \(checkboxes\)](#) marcadas pueden tener éxito.
- Para los [radiobotones \(radio buttons\)](#) que compartan el mismo valor del atributo [name](#), solamente el radiobotón marcado puede tener éxito.
- Para los [menús](#), el [nombre de control](#) viene dado por un elemento [SELECT](#) y los valores son proporcionados por elementos [OPTION](#). Solamente las opciones seleccionadas pueden tener éxito. Cuando no haya opciones seleccionadas, el control no tiene éxito, y ni el nombre ni los valores se envían al servidor cuando se envía el formulario.

- El [valor actual](#) de un [selector de ficheros](#) es una lista de uno o más nombres de ficheros. Al enviar el formulario, los *contenidos* de cada fichero se envían con el resto de los datos del formulario. Los contenidos de los ficheros se empaquetan de acuerdo con el [tipo de contenido](#) del formulario.
- El valor actual de un control tipo objeto está determinado por la implementación del objeto.

Si un control no tiene [valor actual](#) cuando se envía el formulario, los agentes de usuario no están obligados a tratarlo como un control con éxito.

Además de esto, los agentes de usuario no deberían considerar a los siguientes como controles con éxito:

- [Botones de reinicialización \(reset buttons\)](#).
- Elementos [OBJECT](#) cuyo atributo [declare](#) haya sido establecido.

Los [controles ocultos](#) y los controles que no sean representados debido a la configuración de una [hoja de estilo](#) pueden tener éxito. Por ejemplo:

```
<FORM action="..." method="post">
<P>
<INPUT type="password" style="display:none"
      name="password-invisible"
      value="mipassword">
</FORM>
```

En este ejemplo, seguirá habiendo un valor emparejado con el nombre "password-invisible" que será enviado con el formulario.

17.13.3 Procesamiento de los datos del formulario

Cuando el usuario envía un formulario (p.ej., activando un [botón de envío](#)), el agente de usuario lo procesa del siguiente modo.

Paso uno: Identificar los [controles con éxito](#)

Paso dos: Construir el conjunto de datos del formulario

Un conjunto de datos del formulario es una secuencia de parejas [nombre de control/valor actual](#) construida a partir de los [elementos con éxito](#).

Paso tres: Codificar el conjunto de datos del formulario

El conjunto de datos del formulario se codifica a continuación de acuerdo con el [tipo de contenido](#) especificado por el atributo [enctype](#) del elemento [FORM](#).

Paso cuatro: Enviar el conjunto de datos del formulario codificado

Finalmente, los datos codificados se envían al agente procesador designado por el atributo [action](#) usando el protocolo especificado por el atributo [method](#).

Esta especificación no especifica todos los métodos válidos de envío ni los [tipos de contenido](#) que pueden utilizarse con los formularios. Sin embargo, los agentes de usuario HTML 4 deben soportar las convenciones establecidas en los siguientes casos:

- Si [method](#) es "get" y [action](#) es un URI HTTP, el agente de usuario toma el valor de [action](#), le agrega un "?", y a continuación agrega el [conjunto de datos del formulario](#), codificado según el [tipo de contenido](#) "application/x-www-form-urlencoded". A continuación el agente de usuario sigue este URI. En este escenario, los datos del formulario se restringen a códigos ASCII.
- Si [method](#) es "post" y [action](#) es un URI HTTP, el agente de usuario conduce una transacción HTTP "post" usando el valor del atributo [action](#) y un mensaje creado de acuerdo con el [tipo de contenido](#) especificado por el atributo [enctype](#).

Para cualquier otro valor de [action](#) o [method](#), el comportamiento queda sin especificar.

Los agentes de usuario deberían representar la respuesta de las transacciones HTTP "get" y "post".

17.13.4 Tipos de contenido de formularios

El atributo [enctype](#) del elemento [FORM](#) especifica el [tipo de contenido](#) usado para codificar el [conjunto de datos del formulario](#) para su envío al servidor. Los agentes de usuario deben soportar los tipos de contenido enumerados más adelante. El comportamiento para otros tipos de contenido queda sin especificar.

Consulte también la sección sobre [transformación del signo & en secuencias de escape para valores de atributos URI](#).

application/x-www-form-urlencoded

Este es el tipo de contenido por defecto. Los formularios enviados con este tipo de contenido deben codificarse como sigue:

1. Los nombres de control y los valores se transforman en secuencias de escape. Los caracteres de espacio se sustituyen por '+', y los caracteres reservados se transforman en secuencias de escape según se describe en [\[RFC1738\]](#), sección 2.2. Los caracteres no alfanuméricos se reemplazan por '%HH', un signo de porcentaje y dos dígitos hexadecimales que representan el código ASCII del carácter. Los saltos de línea se representan como parejas "CR LF" (es decir, '%0D%0A').
2. Las parejas nombre de control/valor se enumeran según el orden en que aparecen en el documento. El nombre se separa del valor con un signo '=' y las parejas nombre/valor se separan entre sí con un signo '&'.

multipart/form-data

Nota. Consulte [\[RFC2388\]](#) para información adicional sobre carga de ficheros, incluyendo cuestiones de compatibilidad con versiones anteriores, la relación entre "multipart/form-data" y otros tipos de contenido, cuestiones de rendimiento, etc.

Consulte el apéndice para información sobre [cuestiones de seguridad relacionadas con los formularios](#).

El tipo de contenido "application/x-www-form-urlencoded" no es eficiente para enviar grandes cantidades de datos binarios o textos que contenga caracteres no ASCII. Para enviar formularios que contengan ficheros, datos no ASCII, y datos binarios debería utilizarse el tipo de contenido "multipart/form-data".

El contenido "multipart/form-data" sigue las reglas de todos los flujos de datos MIME multiparte, como se esboza en [\[RFC2045\]](#). La definición de "multipart/form-data" puede encontrarse en el registro [\[IANA\]](#).

Un mensaje "multipart/form-data" contiene una serie de partes, cada una de las cuales representa un [control con éxito](#). Las partes se envían al agente procesador en el mismo orden en que aparecen los controles correspondientes en el flujo del documento. No deberían aparecer límites entre partes en ninguno de los datos; el modo en que se consigue esto queda fuera del alcance de esta especificación.

Como con todos los tipos MIME multiparte, cada parte tiene un encabezado opcional "Content-Type" cuyo valor por defecto es "text/plain". Los agentes de usuario deberían proveer el encabezado "Content-Type", acompañado por un parámetro "charset".

Se espera que cada parte contenga:

1. un encabezado "Content-Disposition" cuyo valor es "form-data".
2. un nombre de atributo que especifique el [nombre de control](#) del control correspondiente. Los nombres de control codificados originalmente con [conjuntos de caracteres no ASCII](#) pueden codificarse usando el método explicado en [\[RFC2045\]](#).

Así, por ejemplo, para un control llamado "micontrol", la parte correspondiente se especificaría así:

```
Content-Disposition: form-data; name="micontrol"
```

Como con todas las transmisiones MIME, se utiliza "CR LF" (es decir, '%0D%0A') para separar líneas de datos.

Se puede codificar cada parte, pudiéndose proporcionar el encabezado "Content-Transfer-Encoding" si el valor de esa parte no es conforme con la codificación por defecto (7BIT) (ver [\[RFC2045\]](#), sección 6)

Si se envían los contenidos de un fichero con un formulario, el fichero introducido debería identificarse con el [tipo de contenido](#) apropiado (p.ej., "application/octet-stream"). Si deben enviarse varios ficheros como resultado de la entrada de un solo formulario, deberían ser enviados como "multipart/mixed" incluidos dentro del "multipart/form-data".

El agente de usuario debería intentar proporcionar un nombre de fichero para cada fichero enviado. El nombre del fichero

puede especificarse con el parámetro "filename" del encabezado 'Content-Disposition: form-data', o, en el caso de múltiples ficheros, en un encabezado 'Content-Disposition: attachment' de la subparte. Si el nombre del fichero del sistema operativo del cliente no está en US-ASCII, el nombre del fichero podría aproximarse o codificarse usando el método de [RFC2045](#). Esto es conveniente, por ejemplo, en aquellos casos en que los ficheros subidos pudieran contener referencias cruzadas (p.ej., un fichero TeX y su descripción de estilo auxiliar ".sty").

El siguiente ejemplo ilustra la codificación "multipart/form-data". Supongamos que tenemos el siguiente formulario:

```
<FORM action="http://servidor.com/cgi/procesar"
      enctype="multipart/form-data"
      method="post">
  <P>
  ¿Cómo se llama? <INPUT type="text" name="nombre-envio"><BR>
  ¿Qué ficheros va a enviar? <INPUT type="file" name="ficheros"><BR>
  <INPUT type="submit" value="Enviar"> <INPUT type="reset">
</FORM>
```

Si el usuario introduce "Alfredo" en la entrada de texto, y selecciona el fichero de texto "fichero1.txt", el agente de usuario podría enviar los datos siguientes:

```
Content-Type: multipart/form-data; boundary=AaB03x

--AaB03x
Content-Disposition: form-data; name="nombre-envio"

Alfredo
--AaB03x
Content-Disposition: form-data; name="ficheros"; filename="fichero1.txt"
Content-Type: text/plain

... contenidos de fichero1.txt ...
--AaB03x--
```

Si el usuario eligió un segundo fichero (una imagen) "fichero2.gif", el agente de usuario podría construir las partes de este modo:

```
Content-Type: multipart/form-data; boundary=AaB03x

--AaB03x
Content-Disposition: form-data; name="nombre-envio"

Alfredo
--AaB03x
Content-Disposition: form-data; name="ficheros"
Content-Type: multipart/mixed; boundary=BbC04y

--BbC04y
Content-Disposition: attachment; filename="fichero1.txt"
Content-Type: text/plain

... contenidos de fichero1.txt ...
--BbC04y
Content-Disposition: attachment; filename="fichero2.gif"
Content-Type: image/gif
Content-Transfer-Encoding: binary

...contenidos de fichero2.gif...
--BbC04y--
--AaB03x--
```

18 Scripts

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Introducción a los scripts](#)
2. [Diseño de documentos para agentes de usuario que soporten scripts](#)
 1. [El elemento SCRIPT](#)
 2. [Especificación del lenguaje de scripts](#)
 - [El lenguaje de scripts por defecto](#)
 - [Declaración local del lenguaje de un script](#)
 - [Referencias a elementos HTML desde un script](#)
 3. [Eventos intrínsecos](#)
 4. [Modificación dinámica de documentos](#)
3. [Diseño de documentos para agentes de usuario que no soporten scripts](#)
 1. [El elemento NOSCRIPT](#)
 2. [Ocultar datos de scripts a los agentes de usuario](#)

18.1 Introducción a los scripts

Un script en el lado del cliente es un programa que puede acompañar a un documento HTML o que puede estar incluido en él. El programa se ejecuta en la máquina del cliente cuando se carga el documento, o en algún otro instante, como por ejemplo cuando se activa un vínculo. El soporte de scripts de HTML es independiente del lenguaje de scripts.

Los scripts ofrecen a los autores la posibilidad de extender los documentos HTML de maneras activas e interactivas. Por ejemplo:

- Pueden evaluarse los scripts a medida que se carga el documento para modificar los contenidos del documento dinámicamente.
- Los scripts pueden acompañar a un formulario para procesar los datos a medida que éstos se introducen. Los diseñadores pueden rellenar dinámicamente partes de un formulario según los valores de los otros campos. También pueden asegurarse de que los datos introducidos concuerden con rangos de valores predeterminados, de que los campos sean consistentes entre sí, etc.
- Los scripts pueden ser llamados por eventos que afecten al documento, como la carga, la descarga, el movimiento del foco sobre los elementos, los movimientos del ratón, etc.
- Los scripts pueden ser vinculados a controles de formulario (p.ej., botones) para producir elementos gráficos para la interfaz del usuario.

Hay dos tipos de scripts que los autores pueden asociar a un documento HTML:

- Aquellos que se ejecutan una sola vez cuando el agente de usuario carga el documento. Los scripts que aparecen dentro de un elemento [SCRIPT](#) se ejecutan cuando el elemento es cargado. Para los agentes de usuario que no puedan o que no vayan a ejecutar scripts, los autores pueden incluir contenido alternativo por medio del elemento [NOSCRIPT](#).
- Aquellos que son ejecutados cada vez que ocurre un determinado evento. Estos scripts pueden ser asignados a varios elementos por medio de los atributos de [eventos intrínsecos](#).

Nota. Esta especificación incluye información más detallada sobre scripts en la sección sobre [macros de scripts](#).

18.2 Diseño de documentos para agentes de usuario que soporten scripts

Las siguientes secciones tratan sobre cuestiones que afectan a los agentes de usuario que soportan scripts.

18.2.1 El elemento SCRIPT

```
<!ELEMENT SCRIPT - - %Script;          -- sentencias de script -->
<!ATTLIST SCRIPT
  charset      %Charset;      #IMPLIED -- codif. de caracteres del recurso
vinculado--
  type         %ContentType; #REQUIRED -- tipo de contenido del lenguaje de scripts
--
  src          %URI;          #IMPLIED -- URI del script externo --
  defer       (defer)        #IMPLIED -- El AU puede retrasar la ejecución --
>
```

Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria**

Definiciones de atributos

src = [uri](#) [CT]

Este atributo especifica la localización de un script externo.

type = [tipo de contenido](#) [CI]

Este atributo especifica el lenguaje de scripts de los contenidos del elemento y prevalece sobre el lenguaje de scripts por defecto. El lenguaje de scripts se especifica como un tipo de contenido (p.ej., "text/javascript"). Los autores deben proporcionar un valor para este atributo. No hay valor por defecto para este atributo.

language = [cdata](#) [CI]

Desaprobado. Este atributo especifica el lenguaje de scripts de los contenidos de este elemento. Su valor es un identificador del lenguaje, pero debido a que estos identificadores no son estándar, este atributo ha sido [desaprobado](#) en favor de type.

defer [CI]

Si está establecido, este atributo booleano indica al agente de usuario que el script no va a generar ningún contenido en el documento (p.ej., en javascript, cuando no hubiera ningún "document.write") y por lo tanto el agente de usuario puede seguir analizando y representando.

Atributos definidos en otros lugares

- [charset](#) (codificaciones de caracteres)

El elemento [SCRIPT](#) coloca un script dentro de un documento. Este elemento puede aparecer cualquier número de veces en el [HEAD](#) o en el [BODY](#) de un documento HTML.

El script puede estar definido dentro de los contenidos del elemento [SCRIPT](#) o en un fichero externo. Si el atributo [src](#) no está establecido, los agentes de usuario deben interpretar que los contenidos del elemento son el script. Si [src](#) tiene un valor URI, los agentes de usuario no deben tener en cuenta los contenidos del elemento y deben obtener el script mediante el URI. Obsérvese que el atributo [charset](#) se refiere a la [codificación de caracteres](#) del script designado por el atributo [src](#); no afecta al contenido del elemento [SCRIPT](#).

Los scripts son evaluados por *motores de scripts*, con los cuales deben poder comunicarse los agentes de usuario.

La [sintaxis de los datos de scripts](#) depende del lenguaje de scripts.

18.2.2 Especificación del lenguaje de scripts

Al no estar ligado el HTML a un lenguaje de scripts específico, los autores de los documentos deben decir explícitamente a los agentes de usuario el lenguaje de cada script. Esto puede hacerse o bien mediante una declaración por defecto o bien mediante una declaración local.

El lenguaje de scripts por defecto

Los autores deberían especificar el lenguaje de scripts por defecto de todos los scripts de un documento incluyendo la siguiente declaración [META](#) en el [HEAD](#):

```
<META http-equiv="Content-Script-Type" content="type">
```

donde "type" es un [tipo de contenido](#) que se refiere al lenguaje de scripts. Como ejemplos de este valor tenemos "text/tcl", "text/javascript", "text/vbscript".

En ausencia de una declaración [META](#), el valor por defecto puede ser establecido con un encabezado HTTP "Content-Script-Type".

```
Content-Script-Type: type
```

donde "type" es nuevamente un [tipo de contenido](#) que se refiere al lenguaje de scripts.

Los agentes de usuario deberían determinar el lenguaje de scripts por defecto de un documento de acuerdo con los siguientes pasos (ordenados de prioridad más alta a más baja):

1. Si alguna declaración [META](#) especifica el "Content-Script-Type", la última de ellas en el flujo de caracteres determina el lenguaje de scripts por defecto.
2. En caso contrario, si algún encabezado HTTP especifica el "Content-Script-Type", el último de ellos en el flujo de caracteres determina el lenguaje de scripts por defecto.

Los documentos que no especifiquen información relativa al lenguaje de scripts por defecto y que contengan elementos que especifiquen un script de [evento intrínseco](#) son incorrectos. Los agentes de usuario aún pueden intentar interpretar scripts especificados incorrectamente, pero no se requiere que lo hagan. Las herramientas de creación deberían generar información sobre el lenguaje de scripts por defecto para ayudar a que los autores eviten la creación de documentos incorrectos.

Declaración local del lenguaje de un script

Se debe especificar el atributo type de todos los elementos [SCRIPT](#) de un documento. El valor del atributo type de un elemento [SCRIPT](#) prevalece sobre el lenguaje de scripts por defecto de ese elemento.

En este ejemplo, declaramos que el lenguaje de scripts por defecto es "text/tcl". Incluimos un [SCRIPT](#) en la cabecera, cuyo script se localiza en un fichero externo y que está en el lenguaje de scripts "text/vbscript". También incluimos un [SCRIPT](#) en el cuerpo, que contiene su propio script escrito en "text/javascript".

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE>Un documento con SCRIPT</TITLE>
<META http-equiv="Content-Script-Type" content="text/tcl">
<SCRIPT type="text/vbscript" src="http://algunsitio.com/progs/vbcalc">
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT type="text/javascript">
...algo en JavaScript...
</SCRIPT>
</BODY>
</HTML>
```

Referencias a elementos HTML desde un script

Cada lenguaje de scripts tiene sus propias convenciones para referirse a objetos HTML desde dentro del script. Esta especificación no define un mecanismo estándar para referirse a objetos HTML.

Sin embargo, los scripts deberían hacer referencia a un elemento de acuerdo con su nombre asignado. Los motores de scripts deberían seguir las siguientes reglas de precedencia cuando identifiquen un elemento: un atributo [name](#) prevalece sobre un

atributo [id](#) si ambos están establecidos. En caso contrario, se puede usar uno u otro.

18.2.3 Eventos intrínsecos

Nota. Se avisa a los autores de documentos HTML que se espera que haya cambios en el ámbito de los eventos intrínsecos (p.ej., cómo se ligan los scripts a los eventos). La investigación en este campo está siendo llevada a cabo por miembros del Grupo de Trabajo del Modelo de Objetos del Documento del W3C (ver el sitio web del W3C en <http://www.w3.org/> para más información).

Definiciones de atributos

onload = [script](#) [CT]

El evento onload ocurre cuando el agente de usuario finaliza la carga de una ventana o de todos los marcos de un [FRAMESET](#). Este atributo puede utilizarse con los elementos [BODY](#) y [FRAMESET](#).

onunload = [script](#) [CT]

El evento onunload ocurre cuando el agente de usuario elimina un documento de una ventana o marco. Este atributo puede utilizarse con los elementos [BODY](#) y [FRAMESET](#).

onclick = [script](#) [CT]

El evento onclick ocurre cuando se hace clic con el dispositivo apuntador sobre un elemento. Este atributo puede utilizarse con la mayoría de los elementos.

ondblclick = [script](#) [CT]

El evento ondblclick ocurre cuando se hace doble clic con el dispositivo apuntador sobre un elemento. Este atributo puede utilizarse con la mayoría de los elementos.

onmousedown = [script](#) [CT]

El evento onmousedown ocurre cuando el botón del dispositivo apuntador se pulsa cuando está encima de un elemento. Este atributo puede utilizarse con la mayoría de los elementos.

onmouseup = [script](#) [CT]

El evento onmouseup ocurre cuando el botón del dispositivo apuntador se suelta cuando está encima de un elemento. Este atributo puede utilizarse con la mayoría de los elementos.

onmouseover = [script](#) [CT]

El evento onmouseover ocurre cuando el dispositivo apuntador se sitúa sobre un elemento. Este atributo puede utilizarse con la mayoría de los elementos.

onmousemove = [script](#) [CT]

El evento onmousemove ocurre cuando el dispositivo apuntador se mueve mientras está sobre un elemento. Este atributo puede utilizarse con la mayoría de los elementos.

onmouseout = [script](#) [CT]

El evento onmouseout ocurre cuando el dispositivo apuntador se aparta de un elemento. Este atributo puede utilizarse con la mayoría de los elementos.

onfocus = [script](#) [CT]

El evento [onfocus](#) ocurre cuando el foco se dirige hacia un elemento, ya sea con el dispositivo apuntador o por navegación con tabulador. Este atributo puede utilizarse con los siguientes elementos: [A](#), [AREA](#), [LABEL](#), [INPUT](#), [SELECT](#), [TEXTAREA](#) y [BUTTON](#).

onblur = [script](#) [CT]

El evento onblur ocurre cuando el elemento pierde el foco ya sea con el dispositivo apuntador o por navegación con tabulador. Puede utilizarse con los mismos elementos que onfocus.

onkeypress = [script](#) [CT]

El evento onkeypress ocurre cuando se pulsa y se suelta una tecla encima de un elemento. Este atributo puede utilizarse con la mayoría de los elementos.

onkeydown = [script](#) [CT]

El evento onkeydown ocurre cuando se pulsa una tecla encima de un elemento. Este atributo puede utilizarse con la mayoría de los elementos.

onkeyup = [script](#) [CT]

El evento onkeyup ocurre cuando una tecla se suelta encima de un elemento. Este atributo puede utilizarse con la mayoría de los elementos.

onsubmit = [script \[CT\]](#)

El evento onsubmit ocurre cuando se envía un formulario. Sólo se aplica al elemento [FORM](#).

onreset = [script \[CT\]](#)

El evento onreset ocurre cuando se reinicializa un formulario. Sólo se aplica al elemento [FORM](#).

onselect = [script \[CT\]](#)

El evento onselect ocurre cuando un usuario selecciona texto de un campo de texto. Este atributo puede utilizarse con los elementos [INPUT](#) y [TEXTAREA](#).

onchange = [script \[CT\]](#)

El evento onchange ocurre cuando un control pierde el foco de entrada y su valor ha sido modificado después de que el foco se dirigió hacia él. Este atributo se aplica a los siguientes elementos: [INPUT](#), [SELECT](#) y [TEXTAREA](#).

Es posible asociar una acción con un cierto número de eventos que ocurren cuando un usuario interacciona con un agente de usuario. Cada uno de los "eventos intrínsecos" recién enumerados toma como valor un script. El script se ejecuta cada vez que el evento ocurre para ese elemento. La [sintaxis de los datos del script](#) depende del lenguaje de scripts.

Los elementos de control tales como [INPUT](#), [SELECT](#), [BUTTON](#), [TEXTAREA](#) y [LABEL](#) responden todos a ciertos eventos intrínsecos. Cuando estos elementos no aparecen dentro de un formulario, se pueden emplear para enriquecer la interfaz gráfica del usuario del documento.

Por ejemplo, los autores pueden querer incluir botones en sus documentos que no envíen un formulario pero que puedan comunicarse con un servidor cuando son activados.

Los siguientes ejemplos muestran posibles comportamientos de controles e interfaces de usuario basados en eventos intrínsecos.

En el siguiente ejemplo, nombreUsuario es un campo de texto obligatorio. Cuando un usuario intenta abandonar el campo, el evento onblur llama a una función JavaScript para confirmar que nombreUsuario tiene un valor aceptable.

```
<INPUT NAME="nombreUsuario" onblur="validarNombreUsuario(this.value)">
```

Aquí tenemos otro ejemplo en JavaScript:

```
<INPUT NAME="num"
  onchange="if (!checkNum(this.value, 1, 10))
    {this.focus();this.select();} else {thanks()}"
  VALUE="0">
```

Aquí tenemos un ejemplo en VBScript de un manejador de eventos para un campo de texto:

```
<INPUT name="edit1" size="50">
<SCRIPT type="text/vbscript">
  Sub edit1_cambiado()
    If edit1.value = "abc" Then
      button1.enabled = True
    Else
      button1.enabled = False
    End If
  End Sub
</SCRIPT>
```

Aquí tenemos el mismo ejemplo usando Tcl:

```
<INPUT name="edit1" size="50">
<SCRIPT type="text/tcl">
  proc edit1_cambiado {} {
    if {[edit value] == abc} {
      button1 enable 1
    }
  }
</SCRIPT>
```

Scripts en documentos HTML

```
    } else {  
        button1 enable 0  
    }  
}  
edit1 onChange edit1_cambiado  
</SCRIPT>
```

Aquí tenemos un ejemplo en JavaScript que asocia un evento con un script. En primer lugar vemos un manejador sencillo de clics:

```
<BUTTON type="button" name="miboton" value="10">  
<SCRIPT type="text/javascript">  
    function mi_onclick() {  
        . . .  
    }  
    document.form.miboton.onclick = mi_onclick  
</SCRIPT>  
</BUTTON>
```

Aquí tenemos un manejador de ventanas más interesante:

```
<SCRIPT type="text/javascript">  
    function mi_onload() {  
        . . .  
    }  
  
    var ventana = window.open("algun/otro/URI")  
    if (ventana) ventana.onload = mi_onload  
</SCRIPT>
```

En Tcl esto sería algo así:

```
<SCRIPT type="text/tcl">  
    proc mi_onload {} {  
        . . .  
    }  
    set ventana [window open "algun/otro/URI"]  
    if {$ventana != ""} {  
        $ventana onload mi_onload  
    }  
</SCRIPT>
```

Obsérvese que un "document.write" o sus sentencias equivalentes en un manejador de eventos intrínsecos lo que hacen es crear un nuevo documento y escribir en él, no modificar el documento actual.

18.2.4 Modificación dinámica de documentos

Los scripts que se ejecutan cuando un documento es cargado pueden modificar los contenidos del documento dinámicamente. La capacidad de hacer esto depende del lenguaje de scripts en sí (p.ej., la sentencia "document.write" en el modelo de objetos de HTML no está soportada por algunas marcas).

La modificación dinámica de un documento puede ser modelizada de la siguiente manera:

1. Todos los elementos [SCRIPT](#) se evalúan en orden a medida que el documento es cargado.
2. Todas las construcciones de scripts contenidas en un elemento [SCRIPT](#) dado que generen datos CDATA SGML son evaluados. Su texto generado combinado se inserta en el documento sustituyendo al documento [SCRIPT](#).
3. Los datos CDATA generados son evaluados nuevamente.

Los documentos HTML deben ser conformes con el DTD del HTML tanto antes como después del procesamiento de cualquiera de los elementos [SCRIPT](#).

El ejemplo siguiente ilustra cómo puede un script modificar un documento dinámicamente. El siguiente script:

```
<TITLE>Documento de prueba</TITLE>
<SCRIPT type="text/javascript">
    document.write("<p><b>¡Hola Mundo!\</b>")
</SCRIPT>
```

tiene el mismo efecto que este código HTML:

```
<TITLE>Documento de prueba</TITLE>
<P><B>¡Hola Mundo!\</B>
```

18.3 Diseño de documentos para agentes de usuario que no soporten scripts

Las siguientes secciones tratan sobre cómo pueden los autores crear documentos que funcionen para agentes de usuario que no soporten scripts.

18.3.1 El elemento NOSCRIPT

```
<!ELEMENT NOSCRIPT - - (%block;)+
  -- contenedor de contenido alternativo para representación no basada en scripts -->
<!ATTLIST NOSCRIPT
  %attrs;                -- %coreattrs, %i18n, %events --
  >
```

*Etiqueta inicial: **obligatoria**, Etiqueta final: **obligatoria***

El elemento [NOSCRIPT](#) permite a los autores proporcionar contenido alternativo cuando un script no es ejecutado. El contenido de un elemento [NOSCRIPT](#) sólo debería ser representado por un agente de usuario capaz de reconocer scripts en los casos siguientes:

- El agente de usuario está configurado para no evaluar scripts.
- El agente de usuario no soporta un lenguaje de scripts invocado por un elemento [SCRIPT](#) anterior en el documento.

Los agentes de usuario que no soporten scripts en el lado del cliente deben representar los contenidos de este elemento.

En el ejemplo siguiente, un agente de usuario que ejecute el [SCRIPT](#) incluirá en el documento algunos datos creados dinámicamente. Si el agente de usuario no soporta scripts, el usuario aún podrá obtener los datos por medio de un vínculo.

```
<SCRIPT type="text/tcl">
  ...scripts Tcl para insertar datos...
</SCRIPT>
<NOSCRIPT>
  <P>Acceder a los <A href="http://algunsitio.com/datos">datos.</A>
</NOSCRIPT>
```

18.3.2 Ocultar datos de scripts a agentes de usuario

Es probable que los agentes de usuario que no reconozcan el elemento [SCRIPT](#) representen los contenidos del elemento como texto. Algunos motores de scripts, incluyendo los de los lenguajes JavaScript, VBScript y Tcl, permiten que las sentencias de los scripts estén contenidas en un comentario SGML. Los agentes de usuario que no reconozcan el elemento [SCRIPT](#) ignorarán así el comentario, mientras que los motores de scripts que funcionen correctamente entenderán que los scripts de los comentarios deberían ser ejecutados.

Otra solución al problema es mantener los scripts en documentos externos y hacer referencia a ellos con el atributo [src](#).

Comentando scripts en Javascript

El motor de JavaScript permite que aparezca la cadena "`<!--`" al principio del elemento `SCRIPT`, e ignora el resto de los

caracteres hasta el final de la línea. JavaScript interpreta "/" como el inicio de un comentario que se extiende hasta el final de la línea actual. Esta cadena es necesaria para ocultar la cadena "-->" al analizador JavaScript.

```
<SCRIPT type="text/javascript">
<!-- para ocultar los contenidos del script a los navegadores viejos
function cuadrado(i) {
    document.write("La llamada pasó ", i , " a la función.", "<BR>")
    return i * i
}
document.write("La función devolvió ", cuadrado(5), ".")
// dejar de ocultar contenidos a los navegadores viejos -->
</SCRIPT>
```

Comentando scripts en VBScript

En VBScript, un carácter de comilla simple hace que el resto de la línea actual sea tratada como un comentario. Puede usarse por tanto para ocultar a VBScript la cadena "-->", por ejemplo:

```
<SCRIPT type="text/vbscript">
<!--
Sub blabla()
    ...
End Sub
' -->
</SCRIPT>
```

Comentando scripts en TCL

En Tcl, el carácter "#" comenta el resto de la línea:

```
<SCRIPT type="text/tcl">
<!-- para ocultar los contenidos del script a los navegadores viejos
proc cuadrado {i} {
    document write "La llamada pasó $i a la función.<BR>"
    return [expr $i * $i]
}
document write "La función devolvió [cuadrado 5]."
```

dejar de ocultar los contenidos a los navegadores viejos -->

```
</SCRIPT>
```

Nota. Algunos navegadores cierran los comentarios al encontrar el primer carácter ">", de modo que para ocultar el contenido de los scripts de estos navegadores, se pueden invertir los operandos de los operadores relacionales y de desplazamiento (p.ej., usar "y < x" en vez de "x > y") o se pueden usar caracteres de escape dependientes del lenguaje de scripts para ">".

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

19 Información de referencia de SGML para HTML

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Validación de documentos](#)
2. [Catálogo SGML de ejemplo](#)

Las secciones siguientes contienen la definición formal SGML de HTML 4. Incluye la [declaración SGML](#), la [Definición del Tipo de Documento](#) (Document Type Definition, DTD), y las [Referencias a entidades de caracteres](#), así como un [catálogo SGML de ejemplo](#).

Estos ficheros también están disponibles en formato ASCII según se indica a continuación:

N.T.: Estas son las versiones originales en inglés de los documentos correspondientes.

DTD por Defecto:

[strict.dtd](#)

DTD Transicional:

[loose.dtd](#)

DTD para Documentos con Marcos:

[frameset.dtd](#)

Declaración SGML:

[HTML4.decl](#)

Ficheros de definición de entidades:

[HTMLspecial.ent](#)

[HTMLsymbol.ent](#)

[HTMLlat1.ent](#)

Un catálogo de ejemplo:

[HTML4.cat](#)

19.1 Validación de documentos

Muchos autores se basan en un conjunto limitado de navegadores para comprobar los documentos que producen, y suponen que si los navegadores pueden representar sus documentos, es que éstos son válidos. Desafortunadamente, esto es una manera muy poco efectiva de verificar la validez de un documento, precisamente porque los navegadores están diseñados para que al encontrar documentos no válidos los representen lo mejor posible para evitar la frustración de los usuarios.

Para una mejor validación, debería comprobar su documento con un analizador SGML tal como nsgmls (ver [\[SP\]](#)), para verificar que los documentos HTML sean conformes con el DTD de HTML 4. Si la [declaración del tipo de documento](#) de su documento incluye un URI y si su analizador SGML soporta este tipo de identificador de sistema, obtendrá el DTD directamente. En otro caso puede usar el catálogo SGML de ejemplo que sigue. Éste asume que el DTD ha sido guardado como "strict.dtd" y que las entidades se encuentran en los ficheros "HTMLlat1.ent", "HTMLsymbol.ent" y "HTMLspecial.ent". En cualquier caso, asegúrese de que su analizador SGML es capaz de trabajar con [\[ISO10646\]](#). Vea la documentación de su herramienta de validación para más detalles.

Tenga muy en cuenta que esta validación, aunque es útil y muy recomendable, no garantiza que el documento sea completamente conforme con la especificación HTML 4. Esto se debe a que el analizador SGML se basa únicamente en el DTD de SGML dado, el cual no expresa todos los aspectos de un documento HTML 4 válido. Más concretamente, una analizador SGML asegura que la sintaxis, la estructura, la lista de elementos y sus atributos son válidos. Pero, por ejemplo, no puede detectar errores tales como dar al atributo [width](#) de un elemento [IMG](#) un valor no válido (p.ej., "blabla" o "12.5"). Aunque esta especificación restringe el valor de este atributo a "un entero que representa una longitud en píxeles", el DTD sólo lo define como [CDATA](#), que en realidad admite cualquier valor. Sólo un programa especializado podría englobar la totalidad de la especificación de HTML 4.

De todos modos, este tipo de validación sigue siendo altamente recomendable, ya que permite la detección de un gran conjunto de errores que hacen a los documentos no válidos.

19.2 Catálogo SGML de ejemplo

Este catálogo incluye la directriz *override* para asegurarse de que los programas procesadores tales como nsgmls utilicen identificadores públicos con preferencia sobre los identificadores de sistema. Esto significa que los usuarios no deben estar conectados a la Web para obtener los identificadores de sistema basados es URIs.

```

OVERRIDE YES

```

```

PUBLIC "-//W3C//DTD HTML 4.01//EN" strict.dtd
PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" loose.dtd
PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" frameset.dtd
PUBLIC "-//W3C//ENTITIES Latin1//EN//HTML" HTMLlat1.ent
PUBLIC "-//W3C//ENTITIES Special//EN//HTML" HTMLspecial.ent
PUBLIC "-//W3C//ENTITIES Symbols//EN//HTML" HTMLsymbol.ent

```

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

20 Declaración SGML de HTML 4

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Nota. *El número total de puntos de código permitidos en el conjunto de caracteres del documento de esta declaración SGML incluye los primeros 17 planos de [\[ISO10646\]](#) (17 por 65536). Se ha hecho esta limitación porque este número está limitado a una longitud de 8 dígitos en la versión actual del estándar SGML. Esto no implica que se vaya mantener a largo plazo la restricción del UCS (Conjunto universal de caracteres) a los primeros 17 planos. Es altamente probable que la superación de la limitación de 8 dígitos de SGML, y la actualización de esta especificación, se produzcan antes de la primera asignación de un carácter por encima de los 17 primeros planos.*

Nota. *En sentido estricto, el número de registro ISO 177 se refiere al estado original de [\[ISO10646\]](#) en 1993. Desde 1993 se han introducido cambios consistentes en la adición de caracteres y en una ocasión una operación de recolocación de un gran número de puntos de código para el Coreano Hangul (enmienda 5). Las revisiones de la especificación HTML 4 pueden actualizar la referencia a la ISO 10646 para incluir cambios adicionales.*

20.1 Declaración SGML

```
<!SGML "ISO 8879:1986 (WWW)"
--
    Declaración SGML para el Lenguaje de Formato de Documentos
    de Hipertexto, versión 4

    Con soporte para los primeros 17 planos de ISO 10646
    y límites incrementados para longitudes de etiquetas y
    literales, etc.
--

CHARSET
    BASESET "ISO Registration Number 177//CHARSET
            ISO/IEC 10646-1:1993 UCS-4 with
            implementation level 3//ESC 2/5 2/15 4/6"
    DESCSET 0          9          UNUSED
            9          2          9
            11         2          UNUSED
            13         1          13
            14         18         UNUSED
            32         95         32
            127        1          UNUSED
            128        32         UNUSED
            160        55136      160
```


55296 2048 UNUSED -- SURROGATES --
 57344 1056768 57344

CAPACITY SGMLREF
 TOTALCAP 150000
 GRPCAP 150000
 ENTCAP 150000

SCOPE DOCUMENT
 SYNTAX

SHUNCHAR CONTROLS 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 127

BASESET "ISO 646IRV:1991//CHARSET
 International Reference Version
 (IRV)//ESC 2/8 4/2"

DESCSET 0 128 0

FUNCTION

RE 13
 RS 10
 SPACE 32
 TAB SEPCHAR 9

NAMING LCNMSTRT ""
 UCNMSTRT ""
 LCNMCHAR ".-_: "
 UCNMCHAR ".-_: "
 NAMECASE GENERAL YES
 ENTITY NO

DELIM GENERAL SGMLREF
 HCRO "&#x" -- 38 es el número del ampersand --
 SHORTREF SGMLREF

NAMES SGMLREF

QUANTITY SGMLREF
 ATTCNT 60 -- incrementado --
 ATTSPLN 65536 -- Estos son los valores más grandes --
 LITLEN 65536 -- permitidos en la declaración --
 NAMELEN 65536 -- Evita los límites fijados en las --
 PILEN 65536 -- implementaciones actuales de los --
 TAGLVL 100 -- agentes de usuario HTML --
 TAGLEN 65536
 GRPGTCNT 150
 GRPCNT 64

FEATURES

MINIMIZE

DATATAG NO
 OMITTAG YES
 RANK NO
 SHORTTAG YES

LINK

SIMPLE NO
 IMPLICIT NO

EXPLICIT NO
OTHER
CONCUR NO
SUBDOC NO
FORMAL YES
APPINFO NONE

>

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

21 Definición del Tipo de Documento

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

```
<!--
Este es el DTD Estricto de HTML 4.01, que excluye los elementos
y atributos de presentación que el W3C espera que dejen de
utilizarse a medida que madure el soporte de las hojas de estilo.
Los autores deberían usar el DTD Estricto siempre que sea posible,
pero pueden usar el DTD Transicional si es necesario el soporte
de elementos y atributos de presentación.
```

```
HTML 4 incluye mecanismos para hojas de estilo, scripts, objetos
incluidos, soporte mejorado de texto de derecha a izquierda o
direcciones mezcladas, y mejoras en los formularios para facilitar
la accesibilidad por parte de personas con discapacidades.
```

```
Borrador: $Date: 1999/12/24 22:40:35 $
```

```
Autores:
```

```
  Dave Raggett <dsr@w3.org>
  Arnaud Le Hors <lehors@w3.org>
  Ian Jacobs <ij@w3.org>
```

```
Puede encontrar más información sobre HTML 4.01 en:
```

```
http://www.w3.org/TR/1999/REC-html401-19991224
```

```
La especificación de HTML 4.01 incluye restricciones
sintácticas adicionales que no pueden expresarse en
los términos de los DTDs.
```

```
-->
<!--
Uso normal:
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
...
</head>
<body>
...
</body>
</html>
```

```
El URI utilizado como identificador de sistema en el identificador
público permite al agente de usuario descargar el DTD y los conjuntos
de entidades necesarios.
```

El FPI (Formal Public Identifier) para el DTD Transicional de HTML 4.01 es:

```
"-//W3C//DTD HTML 4.01 Transitional//EN"
```

Esta versión del DTD Transicional es:

```
http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd
```

Si está escribiendo un documento que incluya marcos, use el siguiente FPI:

```
"-//W3C//DTD HTML 4.01 Frameset//EN"
```

Esta versión del DTD para Documentos con Marcos es:

```
http://www.w3.org/TR/1999/REC-html401-19991224/frameset.dtd
```

Use los siguientes URIs (relativos) para hacer referencia a los DTDs y definiciones de entidades de esta especificación:

```
"strict.dtd"  
"loose.dtd"  
"frameset.dtd"  
"HTMLlat1.ent"  
"HTMLsymbol.ent"  
"HTMLspecial.ent"
```

```
-->
```

```
<!--===== Nombres Importados =====-->  
<!-- Conmutador de Característica para Documentos con Marcos -->  
<!ENTITY % HTML.Frameset "IGNORE">  
  
<!ENTITY % ContentType "CDATA"  
  -- tipo de medio, según \[RFC2045\]  
  -->  
  
<!ENTITY % ContentTypes "CDATA"  
  -- lista de tipos de medios separados por comas, según \[RFC2045\]  
  -->  
  
<!ENTITY % Charset "CDATA"  
  -- una codificación de caracteres, según \[RFC2045\]  
  -->  
  
<!ENTITY % Charsets "CDATA"  
  -- una lista de codificaciones de caracteres separadas por espacios,  
  según \[RFC2045\]  
  -->  
  
<!ENTITY % LanguageCode "NAME"  
  -- un código de idioma, según \[RFC1766\]  
  -->  
  
<!ENTITY % Character "CDATA"  
  -- un carácter individual de \[ISO10646\]  
  -->
```

```

<!ENTITY % LinkTypes "CDATA"
  -- una lista de tipos de vínculos separados por espacios
  -->

<!ENTITY % MediaDesc "CDATA"
  -- un descriptor de medios, o una lista de ellos separados por comas
  -->

<!ENTITY % URI "CDATA"
  -- un Identificador Uniforme de Recursos,
  ver \[URI\]
  -->

<!ENTITY % Datetime "CDATA" -- información de fecha y hora. Formato de fechas ISO -->

<!ENTITY % Script "CDATA" -- expresión de script -->

<!ENTITY % StyleSheet "CDATA" -- datos de hoja de estilo -->

<!ENTITY % Text "CDATA">

<!-- Entidades Paramétricas -->

<!ENTITY % head.misc "SCRIPT | STYLE | META | LINK | OBJECT" -- elementos de cabecera
  repetibles -->

<!ENTITY % heading "H1 | H2 | H3 | H4 | H5 | H6">

<!ENTITY % list "UL | OL">

<!ENTITY % preformatted "PRE">

<!--===== Entidades nemónicas de caracteres =====>

<!ENTITY % HTMLlat1 PUBLIC
  "-//W3C//ENTITIES Latin1//EN//HTML"
  "HTMLlat1.ent">
%HTMLlat1;

<!ENTITY % HTMLsymbol PUBLIC
  "-//W3C//ENTITIES Symbols//EN//HTML"
  "HTMLsymbol.ent">
%HTMLsymbol;

<!ENTITY % HTMLspecial PUBLIC
  "-//W3C//ENTITIES Special//EN//HTML"
  "HTMLspecial.ent">
%HTMLspecial;

<!--===== Atributos Genéricos =====>

<!ENTITY % coreattrs
  "id          ID          #IMPLIED -- identificador único a nivel de documento --
  "class       CDATA       #IMPLIED -- lista de clases separadas por espacios --

```

```

  style      %StyleSheet; #IMPLIED -- información de estilo asociada --
  title     %Text;      #IMPLIED -- título consultivo --"
>

<!ENTITY % il8n
"lang      %LanguageCode; #IMPLIED -- código de idioma --
 dir      (ltr|rtl)      #IMPLIED -- dirección del texto débil/neutral --"
>

<!ENTITY % events
"onclick   %Script;      #IMPLIED -- se hizo clic con un botón del apuntador --
 ondblclick %Script;      #IMPLIED -- se hizo doble clic con un botón
 del apuntador --
 onmousedown %Script;    #IMPLIED -- se pulsó un botón del apuntador --
 onmouseup  %Script;    #IMPLIED -- se soltó un botón del apuntador --
 onmouseover %Script;    #IMPLIED -- se movió un apuntador al interior de --
 onmousemove %Script;  #IMPLIED -- se movió un apuntador en el interior de --
 onmouseout %Script;  #IMPLIED -- se alejó un apuntador --
 onkeypress %Script;    #IMPLIED -- se pulsó y soltó una tecla --
 onkeydown  %Script;    #IMPLIED -- se pulsó una tecla --
 onkeyup    %Script;    #IMPLIED -- se soltó una tecla --"
>

<!-- Conmutador de Característica Reservada -->
<!ENTITY % HTML.Reserved "IGNORE">

<!-- Los siguientes atributos están reservados para su posible uso futuro -->
<![ %HTML.Reserved; [
<!ENTITY % reserved
"datasrc   %URI;      #IMPLIED -- una fuente de datos individual o tabular --
 datafld   CDATA         #IMPLIED -- el nombre de la propiedad o de la columna
--
 dataformatas (plaintext|html) plaintext -- texto o html --"
>
]]>

<!ENTITY % reserved "">

<!ENTITY % attrs "%coreattrs; %il8n; %events;">

<!--===== Códigos para Texto =====>

<!ENTITY % fontstyle
"TT | I | B | BIG | SMALL">

<!ENTITY % phrase "EM | STRONG | DFN | CODE |
SAMP | KBD | VAR | CITE | ABBR | ACRONYM" >

<!ENTITY % special
"A | IMG | OBJECT | BR | SCRIPT | MAP | Q | SUB | SUP | SPAN | BDO">

<!ENTITY % formctrl "INPUT | SELECT | TEXTAREA | LABEL | BUTTON">

<!-- %inline; cubre elementos en línea o "a nivel de texto" -->
<!ENTITY % inline "#PCDATA | %fontstyle; | %phrase; | %special; | %formctrl;">

```

```

<!ELEMENT (%fontstyle;|%phrase;) - - (%inline;)*>
<!ATTLIST (%fontstyle;|%phrase;)
  %attrs;          -- %coreattrs, %i18n, %events --
>

<!ELEMENT (SUB|SUP) - - (%inline;)* -- subíndice, superíndice -->
<!ATTLIST (SUB|SUP)
  %attrs;          -- %coreattrs, %i18n, %events --
>

<!ELEMENT SPAN - - (%inline;)* -- contenedor genérico de idioma/estilo -->
<!ATTLIST SPAN
  %attrs;          -- %coreattrs, %i18n, %events --
  %reserved;      -- reservado para posibles usos futuros --
>

<!ELEMENT BDO - - (%inline;)* -- anular BiDi I18N -->
<!ATTLIST BDO
  %coreattrs;     -- id, class, style, title --
  lang            %LanguageCode; #IMPLIED -- código de idioma --
  dir            (ltr|rtl)      #REQUIRED -- direccionalidad --
>

<!ELEMENT BR - O EMPTY -- salto de línea forzado -->
<!ATTLIST BR
  %coreattrs;     -- id, class, style, title --
>

<!--===== Modelos de contenido de HTML =====>

<!--
  HTML tiene dos modelos de contenido básicos:

  %inline;      elementos a nivel de carácter y cadenas de texto
  %block;       elemento tipo bloque, p.ej. párrafos y listas
-->

<!ENTITY % block
  "P | %heading; | %list; | %preformatted; | DL | DIV | NOSCRIPT |
  BLOCKQUOTE | FORM | HR | TABLE | FIELDSET | ADDRESS">

<!ENTITY % flow "%block; | %inline;">

<!--===== Cuerpo del Documento =====>

<!ELEMENT BODY O O (%block;|SCRIPT)+ +(INS|DEL) -- cuerpo del documento -->
<!ATTLIST BODY
  %attrs;          -- %coreattrs, %i18n, %events --
  onload          %Script;    #IMPLIED -- el documento ha sido cargado --
  onunload        %Script;    #IMPLIED -- el documento ha sido quitado --
>

<!ELEMENT ADDRESS - - (%inline;)* -- información sobre el autor -->
<!ATTLIST ADDRESS
  %attrs;          -- %coreattrs, %i18n, %events --
>

```

```

<!ELEMENT DIV - - (%flow;)* -- contenedor genérico de idioma/estilo -->
<!ATTLIST DIV
  %attrs; -- %coreattrs, %il8n, %events --
  %reserved; -- reservado para posibles usos futuros --
>

<!--===== El Elemento Ancla (origen/destino de vínculo) =====>

<!ENTITY % Shape "(rect|circle|poly|default)">
<!ENTITY % Coords "CDATA" -- lista de longitudes separadas por comas -->

<!ELEMENT A - - (%inline;)* -(A) -- ancla -->
<!ATTLIST A
  %attrs; -- %coreattrs, %il8n, %events --
  charset %Charset; #IMPLIED -- codificación de caracteres
  del recurso vinculado --
  type %ContentType; #IMPLIED -- tipo de contenido consultivo --
  name CDATA #IMPLIED -- destino de vínculo con nombre --
  href %URI; #IMPLIED -- URI del recurso vinculado --
  hreflang %LanguageCode; #IMPLIED -- código de idioma --
  rel %LinkTypes; #IMPLIED -- tipos de vínculos directos --
  rev %LinkTypes; #IMPLIED -- tipos de vínculos inversos --
  accesskey %Character; #IMPLIED -- carácter de la tecla de accesibilidad --
  shape %Shape; rect -- para mapas de imágenes en el cliente --
  coords %Coords; #IMPLIED -- para mapas de imágenes en el cliente --
  tabindex NUMBER #IMPLIED -- posición en el orden de tabulación --
  onfocus %Script; #IMPLIED -- el foco se dirigió hacia el elemento --
  onblur %Script; #IMPLIED -- el elemento perdió el foco --
>

<!--===== Mapas de imágenes en el lado del cliente =====>

<!-- Éstos pueden colocarse en el mismo documento o agrupados en un
documento separado, aunque esto último no está todavía muy soportado -->

<!ELEMENT MAP - - ((%block;) | AREA)+ -- mapa de imágenes en el lado del cliente -->
<!ATTLIST MAP
  %attrs; -- %coreattrs, %il8n, %events --
  name CDATA #REQUIRED -- como referencia para usemap --
>

<!ELEMENT AREA - O EMPTY -- área de un mapa de imágenes en el
cliente-->
<!ATTLIST AREA
  %attrs; -- %coreattrs, %il8n, %events --
  shape %Shape; rect -- controla la interpretación de las coords --
  coords %Coords; #IMPLIED -- lista de longitudes separadas por comas --
  href %URI; #IMPLIED -- URI del recurso vinculado --
  nohref (nohref) #IMPLIED -- esta región no tiene acción --
  alt %Text; #REQUIRED -- descripción corta --
  tabindex NUMBER #IMPLIED -- posición en el orden de tabulación --
  accesskey %Character; #IMPLIED -- carácter de la tecla de accesibilidad --
  onfocus %Script; #IMPLIED -- el foco se dirigió hacia el elemento --
  onblur %Script; #IMPLIED -- el elemento perdió el foco --

```


>

<!--===== El Elemento LINK (vínculo) =====-->

<!--

En principio los valores de las relaciones se pueden usar:

- a) para menús/barras de herramientas específicas del documento que usen los elementos LINK de la cabecera del documento, p.ej., start (comienzo), contents (contenidos), previous (anterior), next (siguiente), index (índice), end (final), help (ayuda)
- b) para vincular una hoja de estilo externa (rel=stylesheet)
- c) para vincular un script (rel=script)
- d) por las hojas de estilo para controlar cómo se representan los conjuntos de nodos html en documentos impresos
- e) para crear un vínculo a una versión imprimible del documento, p.ej., una versión postscript o pdf (rel=alternate media=print)

-->

<!ELEMENT LINK - O EMPTY -- un vínculo independiente del medio -->

<!ATTLIST LINK

%attrs;			-- %coreattrs , %i18n , %events --
charset	%Charset;	#IMPLIED	-- codificación de caracteres del recurso vinculado --
href	%URI;	#IMPLIED	-- URI del recurso vinculado --
hreflang	%LanguageCode;	#IMPLIED	-- código de idioma --
type	%ContentType;	#IMPLIED	-- tipo consultivo de contenido --
rel	%LinkTypes;	#IMPLIED	-- tipos de vínculos directos --
rev	%LinkTypes;	#IMPLIED	-- tipos de vínculos inversos --
media	%MediaDesc;	#IMPLIED	-- para representar en estos medios --

>

<!--===== Imágenes =====-->

<!-- Length se define en el DTD Estricto para cellpadding/cellspacing -->

<!ENTITY % Length "[CDATA](#)" -- nn para píxeles o nn% para longitudes porcentuales --><!ENTITY % MultiLength "[CDATA](#)" -- píxeles, porcentaje, o relativa --><![[%HTML.Frameset;](#) [<!ENTITY % MultiLengths "[CDATA](#)" -- lista de Multilength separadas por coma -->

]]>

<!ENTITY % Pixels "[CDATA](#)" -- entero que representa una longitud en píxeles -->

<!-- Para evitar problemas con AAUU de sólo texto así como para hacer el contenido de las imágenes inteligibles y navegables para los usuarios de AAUU no visuales, es necesario proporcionar una descripción con ALT, y evitar mapas de imágenes en el lado del servidor -->

<!ELEMENT IMG - O EMPTY -- Imagen incluida -->

<!ATTLIST IMG

%attrs;			-- %coreattrs , %i18n , %events --
src	%URI;	#REQUIRED	-- URI de la imagen a incluir --
alt	%Text;	#REQUIRED	-- descripción corta --
longdesc	%URI;	#IMPLIED	-- vínculo a descripción larga

```

                                (complementa a alt) --
name          CDATA          #IMPLIED -- nombre de la imagen para los scripts --
height       %Length;      #IMPLIED -- especificar nueva altura --
width        %Length;      #IMPLIED -- especificar nueva anchura --
usemap       %URI;         #IMPLIED -- usar mapa de imágenes en el cliente --
ismap        (ismap)       #IMPLIED -- usar mapa de imágenes en el servidor --
>

<!-- USEMAP apunta a un elemento MAP que puede estar en este documento
o en un documento externo, aunque esto último no está muy soportado -->

<!--===== OBJECT =====>
<!--
Se usa OBJECT para incluir objetos como parte de una página HTML.
Los elementos PARAM deberían preceder a cualquier otro contenido.
Aspectos técnicos del modelo de contenido mixto de SGML impiden
que esto se especifique formalmente...
-->

<!ELEMENT OBJECT - - (PARAM | %flow;)*
-- objeto genérico incluido -->
<!ATTLIST OBJECT
%attrs;          -- %coreattrs, %i18n, %events --
declare         (declare)   #IMPLIED -- declarar pero no crear --
classid        %URI;       #IMPLIED -- identifica una implementación --
codebase       %URI;       #IMPLIED -- URI base para classid, data, archive--
data           %URI;       #IMPLIED -- referencia a los datos del objeto --
type           %ContentType; #IMPLIED -- tipo de contenido de los datos --
codetype       %ContentType; #IMPLIED -- tipo de contenido del código --
archive        CDATA       #IMPLIED -- lista de URIs separados por espacios --
standby        %Text;      #IMPLIED -- mensaje a mostrar mientras se carga --
height         %Length;    #IMPLIED -- especificar nueva altura --
width          %Length;    #IMPLIED -- especificar nueva anchura --
usemap         %URI;       #IMPLIED -- usar mapa de imágenes en el cliente --
name           CDATA       #IMPLIED -- enviar como parte de un formulario --
tabindex       NUMBER      #IMPLIED -- posición en el orden de tabulación --
%reserved;     -- reservado para posibles usos futuros --
>

<!ELEMENT PARAM - O EMPTY -- valor de propiedad con nombre -->
<!ATTLIST PARAM
id            ID          #IMPLIED -- identificador único a nivel de documento --
name         CDATA       #REQUIRED -- nombre de la propiedad --
value        CDATA       #IMPLIED -- valor de la propiedad --
valuetype    (DATA|REF|OBJECT) DATA -- cómo interpretar el valor --
type         %ContentType; #IMPLIED -- tipo de contenido para el valor
                                cuando valuetype=ref --
>

<!--===== Separador Horizontal =====>

<!ELEMENT HR - O EMPTY -- separador horizontal -->
<!ATTLIST HR
%attrs;          -- %coreattrs, %i18n, %events --
>

```

```

<!--===== Párrafos =====>
<!ELEMENT P - O (%inline;)* -- párrafo -->
<!ATTLIST P
  %attrs; -- %coreattrs, %il8n, %events --
>

<!--===== Encabezados =====>

<!--
  Hay seis niveles de encabezados, desde H1 (el más importante)
  hasta H6 (el menos importante).
-->

<!ELEMENT (%heading;) - - (%inline;)* -- encabezado -->
<!ATTLIST (%heading;)
  %attrs; -- %coreattrs, %il8n, %events --
>

<!--===== Texto preformateado =====>

<!-- excluye los códigos de imágenes y de cambios del tamaño de la fuente -->
<!ENTITY % pre.exclusion "IMG|OBJECT|BIG|SMALL|SUB|SUP">

<!ELEMENT PRE - - (%inline;)* -(%pre.exclusion;) -- texto preformateado -->
<!ATTLIST PRE
  %attrs; -- %coreattrs, %il8n, %events --
>

<!--===== Citas en Línea =====>

<!ELEMENT Q - - (%inline;)* -- cita corta en línea -->
<!ATTLIST Q
  %attrs; -- %coreattrs, %il8n, %events --
  cite %URI; #IMPLIED -- URI del documento o mensaje original --
>

<!--===== Citas en Bloque =====>

<!ELEMENT BLOCKQUOTE - - (%block;|SCRIPT)+ -- cita larga -->
<!ATTLIST BLOCKQUOTE
  %attrs; -- %coreattrs, %il8n, %events --
  cite %URI; #IMPLIED -- URI del documento o mensaje original --
>

<!--===== Texto Insertado/Borrado =====>

<!-- INS/DEL se utilizan incluyéndolos en BODY -->
<!ELEMENT (INS|DEL) - - (%flow;)* -- texto insertado, texto borrado -->
<!ATTLIST (INS|DEL)
  %attrs; -- %coreattrs, %il8n, %events --
  cite %URI; #IMPLIED -- información sobre la razón del cambio --
  datetime %Datetime; #IMPLIED -- fecha y hora del cambio --
>

<!--===== Listas =====>

```

```

<!-- listas de definiciones - DT para el término, DD para su definición -->
<!ELEMENT DL - - (DT|DD)+ -- lista de definiciones -->
<!ATTLIST DL
  %attrs; -- %coreattrs, %il8n, %events --
>

<!ELEMENT DT - O (%inline;)* -- término definido -->
<!ELEMENT DD - O (%flow;)* -- descripción de la definición -->
<!ATTLIST (DT|DD)
  %attrs; -- %coreattrs, %il8n, %events --
>

<!ELEMENT OL - - (LI)+ -- lista ordenada -->
<!ATTLIST OL
  %attrs; -- %coreattrs, %il8n, %events --
>

<!-- Estilos de marcador de lista no ordenadas (UL) -->
<!ELEMENT UL - - (LI)+ -- lista no ordenada -->
<!ATTLIST UL
  %attrs; -- %coreattrs, %il8n, %events --
>

<!ELEMENT LI - O (%flow;)* -- objeto de lista -->
<!ATTLIST LI
  %attrs; -- %coreattrs, %il8n, %events --
>

<!--===== Formularios =====>
<!ELEMENT FORM - - (%block;|SCRIPT)+ -(FORM) -- formulario interactivo -->
<!ATTLIST FORM
  %attrs; -- %coreattrs, %il8n, %events --
  action %URI; #REQUIRED -- procesador del formulario en el servidor --
  method (GET|POST) GET -- método HTTP usado para enviar formulario--
  enctype %ContentType; "application/x-www-form-urlencoded"
  accept %ContentTypes; #IMPLIED -- lista de tipos MIME para subir ficheros --
  name CDATA #IMPLIED -- nombre del formulario para los scripts --
  onsubmit %Script; #IMPLIED -- el formulario fue enviado --
  onreset %Script; #IMPLIED -- el formulario fue reinicializado --
  accept-charset %Charsets; #IMPLIED -- lista de codif. de caracteres soportadas --
>

<!-- Cada rótulo (label) no debe contener MÁS de un campo -->
<!ELEMENT LABEL - - (%inline;)* -(LABEL) -- texto del rótulo de un
  campo de formulario -->
<!ATTLIST LABEL
  %attrs; -- %coreattrs, %il8n, %events --
  for IDREF #IMPLIED -- concuerda con el valor ID del campo --
  accesskey %Character; #IMPLIED -- carácter de la tecla de accesibilidad --
  onfocus %Script; #IMPLIED -- el foco se dirigió hacia el elemento --
  onblur %Script; #IMPLIED -- el elemento perdió el foco --
>

```

```

<!ENTITY % InputType
  "(TEXT | PASSWORD | CHECKBOX |
   RADIO | SUBMIT | RESET |
   FILE | HIDDEN | IMAGE | BUTTON)"
>

<!-- se requiere el atributo name para todos excepto para submit y reset -->
<!ELEMENT INPUT - O EMPTY -- control de formulario -->
<!ATTLIST INPUT
  %attrs; -- %coreattrs, %i18n, %events --
  type %InputType; TEXT -- qué tipo de control se necesita --
  name CDATA #IMPLIED -- enviar como parte del formulario --
  value CDATA #IMPLIED -- especificar para radiobotones
  y casillas de verificación --
  checked (checked) #IMPLIED -- para radiobotones y casillas de verif. --
  disabled (disabled) #IMPLIED -- no disponible en este contexto --
  readonly (readonly) #IMPLIED -- para texto y contraseñas --
  size CDATA #IMPLIED -- específico de cada tipo de campo --
  maxlength NUMBER #IMPLIED -- máximo de caracteres para campos de texto
--
  src %URI; #IMPLIED -- para campos con imágenes --
  alt CDATA #IMPLIED -- descripción corta --
  usemap %URI; #IMPLIED -- usar mapa de imágenes en el cliente --
  ismap (ismap) #IMPLIED -- usar mapa de imágenes en el servidor --
  tabindex NUMBER #IMPLIED -- posición en el orden de tabulación --
  accesskey %Character; #IMPLIED -- carácter de la tecla de accesibilidad --
  onfocus %Script; #IMPLIED -- el foco se dirigió hacia el elemento --
  onblur %Script; #IMPLIED -- el elemento perdió el foco --
  onselect %Script; #IMPLIED -- se seleccionó parte del texto --
  onchange %Script; #IMPLIED -- el valor del elemento fue modificado --
  accept %ContentTypes; #IMPLIED -- lista de tipos MIME para subir ficheros --
  %reserved; -- reservado para posibles usos futuros --
>

<!ELEMENT SELECT - - (OPTGROUP|OPTION)+ -- selector de opciones -->
<!ATTLIST SELECT
  %attrs; -- %coreattrs, %i18n, %events --
  name CDATA #IMPLIED -- nombre del campo --
  size NUMBER #IMPLIED -- filas visibles --
  multiple (multiple) #IMPLIED -- por defecto es selección simple --
  disabled (disabled) #IMPLIED -- no disponible en este contexto --
  tabindex NUMBER #IMPLIED -- posición en el orden de tabulación --
  onfocus %Script; #IMPLIED -- el foco se dirigió hacia el elemento --
  onblur %Script; #IMPLIED -- el elemento perdió el foco --
  onchange %Script; #IMPLIED -- el valor del elemento fue modificado --
  %reserved; -- reservado para posibles usos futuros --
>

<!ELEMENT OPTGROUP - - (OPTION)+ -- grupo de opciones -->
<!ATTLIST OPTGROUP
  %attrs; -- %coreattrs, %i18n, %events --
  disabled (disabled) #IMPLIED -- no disponible en este contexto --
  label %Text; #REQUIRED -- para usar en menús jerárquicos --
>

<!ELEMENT OPTION - O (#PCDATA) -- opción seleccionable -->

```

```

<!ATTLIST OPTION
  %attrs;                -- %coreattrs, %i18n, %events --
  selected      (selected)      #IMPLIED  -- opción preseleccionada --
  disabled      (disabled)      #IMPLIED  -- no disponible en este contexto --
  label         %Text;          #IMPLIED  -- para usar en menús jerárquicos --
  value         CDATA           #IMPLIED  -- por defecto es el contenido del elemento --
>

<!ELEMENT TEXTAREA - - (#PCDATA) -- campo de texto multilínea -->
<!ATTLIST TEXTAREA
  %attrs;                -- %coreattrs, %i18n, %events --
  name           CDATA       #IMPLIED
  rows           NUMBER     #REQUIRED
  cols           NUMBER     #REQUIRED
  disabled      (disabled)  #IMPLIED  -- no disponible en este contexto --
  readonly      (readonly)  #IMPLIED
  tabindex      NUMBER     #IMPLIED  -- posición en el orden de tabulación --
  accesskey     %Character;  #IMPLIED  -- carácter de la tecla de accesibilidad --
  onfocus      %Script;    #IMPLIED  -- el foco se dirigió hacia el elemento --
  onblur       %Script;    #IMPLIED  -- el elemento perdió el foco --
  onselect      %Script;    #IMPLIED  -- se seleccionó parte del texto --
  onchange      %Script;    #IMPLIED  -- el valor del elemento fue modificado --
  %reserved;    -- reservado para posibles usos futuros --
>

<!--
#PCDATA es para resolver el problema del contenido mixto,
de acuerdo con la especificación aquí sólo se permite espacio en blanco!
-->
<!ELEMENT FIELDSET - - (#PCDATA,LEGEND,(%flow;)* -- grupo de controles
de formulario -->
<!ATTLIST FIELDSET
  %attrs;                -- %coreattrs, %i18n, %events --
>

<!ELEMENT LEGEND - - (%inline;)* -- leyenda del grupo de campos -->
<!ATTLIST LEGEND
  %attrs;                -- %coreattrs, %i18n, %events --
  accesskey     %Character;  #IMPLIED  -- carácter de la tecla de accesibilidad --
>

<!ELEMENT BUTTON - -
(%flow;)* -(A|%formctrl;|FORM|FIELDSET)
-- botón pulsador -->
<!ATTLIST BUTTON
  %attrs;                -- %coreattrs, %i18n, %events --
  name          CDATA     #IMPLIED
  value         CDATA     #IMPLIED  -- se envía al servidor junto al formulario --
  type         (button|submit|reset) submit -- para usar como botón de formulario --
  disabled     (disabled) #IMPLIED  -- no disponible en este contexto --
  tabindex     NUMBER    #IMPLIED  -- posición en el orden de tabulación --
  accesskey    %Character; #IMPLIED  -- carácter de la tecla de accesibilidad --
  onfocus     %Script;  #IMPLIED  -- el foco se dirigió hacia el elemento --
  onblur      %Script;  #IMPLIED  -- el elemento perdió el foco --
  %reserved;  -- reservado para posibles usos futuros --

```

>

<!--===== Tablas =====-->

<!-- Estándar de tablas HTML IETF, ver [\[RFC1942\]](#) -->

<!--

El atributo BORDER establece el grosor del marco que rodea a la tabla. Las unidades por defecto son píxeles de pantalla.

El atributo FRAME especifica qué partes del marco que rodea a la tabla deberían representarse. Los valores no coinciden con los de CALS para evitar un conflicto de nombres con el atributo VALIGN.

El valor "border" se ha incluido por compatibilidad con <TABLE BORDER>, que produce frame=border y border=implied. Para <TABLE BORDER=1> se obtiene border=1 y frame=implied. En este caso, es apropiado tratar esto como frame=border por compatibilidad con los navegadores existentes.

-->

<!ENTITY % TFrame "(void|above|below|hsides|lhs|rhs|vsides|box|border)">

<!--

El atributo RULES define qué líneas de división entre celdas se van a dibujar:

Si RULES no está presente entonces se hace igual a:

"none" si BORDER está ausente o BORDER=0, "all" en caso contrario

-->

<!ENTITY % TRules "(none | groups | rows | cols | all)">

<!-- colocación horizontal de la tabla en relación al documento -->

<!ENTITY % TAlign "(left|center|right)">

<!-- atributos de alineación horizontal de los contenidos de las celdas -->

<!ENTITY % cellhalign

"[align](#) (left|center|right|justify|char) #IMPLIED

[char](#) [%Character;](#) #IMPLIED -- carácter de alineación, p.ej. char=':' --

[charoff](#) [%Length;](#) #IMPLIED -- offset del carácter de alineación --"

>

<!-- atributos de alineación vertical para los contenidos de las celdas -->

<!ENTITY % cellvalign

"[valign](#) (top|middle|bottom|baseline) #IMPLIED"

>

<!ELEMENT [TABLE](#) - -

([CAPTION?](#), ([COL*](#)|[COLGROUP*](#)), [THEAD?](#), [TFOOT?](#), [TBODY+](#))>

<!ELEMENT [CAPTION](#) - - ([%inline;](#))* -- título de una tabla -->

<!ELEMENT [THEAD](#) - O (TR)+ -- cabecera de tabla -->

<!ELEMENT [TFOOT](#) - O (TR)+ -- pie de tabla -->

<!ELEMENT [TBODY](#) O O (TR)+ -- cuerpo de tabla -->

<!ELEMENT [COLGROUP](#) - O (COL)* -- grupo de columnas de tabla -->

<!ELEMENT [COL](#) - O EMPTY -- columna de tabla -->

<!ELEMENT [TR](#) - O (TH|TD)+ -- fila de tabla -->

<!ELEMENT ([TH](#)|[TD](#)) - O ([%flow;](#))* -- celda de encabezado de tabla, celda de datos de tabla -->

<!ATTLIST [TABLE](#) -- elemento tabla --

```

%attrs; -- %coreattrs, %i18n, %events --
summary %Text; #IMPLIED -- propósito/estructura para salida por voz --
width %Length; #IMPLIED -- anchura de la tabla --
border %Pixels; #IMPLIED -- controlar el borde que rodea a la tabla --
frame %TFrame; #IMPLIED -- qué partes del borde representar --
rules %TRules; #IMPLIED -- líneas de división entre filas y columnas
--
cellspacing %Length; #IMPLIED -- espaciado entre celdas --
cellpadding %Length; #IMPLIED -- espacio dentro de las celdas --
%reserved; -- reservado para posibles usos futuros --
datapagesize CDATA #IMPLIED -- reservado para posibles usos futuros --
>

```

```
<!ATTLIST CAPTION
```

```

%attrs; -- %coreattrs, %i18n, %events --
>

```

```
<!--
```

COLGROUP agrupa un conjunto de elementos COL. Permite agrupar varias columnas relacionadas semánticamente.

```
-->
```

```
<!ATTLIST COLGROUP
```

```

%attrs; -- %coreattrs, %i18n, %events --
span NUMBER 1 -- número por defecto de columnas en el
grupo--
width %MultiLength; #IMPLIED -- anchura por defecto para
las COLs contenidas --
%cellhalign; -- alineación horizontal en las celdas --
%cellvalign; -- alineación vertical en las celdas --
>

```

```
<!--
```

Los elementos COL definen las propiedades de alineación de las celdas de una o más columnas.

El atributo WIDTH especifica la anchura de las columnas, p.ej.

```

width=64      anchura en píxeles de pantalla
width=0.5*    anchura relativa de 0.5

```

El atributo SPAN hace que los atributos de un elemento COL se apliquen a más de una columna.

```
-->
```

```
<!ATTLIST COL
```

```

%attrs; -- %coreattrs, %i18n, %events --
span NUMBER 1 -- los atributos de COL afectan a N columnas
--
width %MultiLength; #IMPLIED -- especificación de la anchura
de las columnas --
%cellhalign; -- alineación horizontal en las celdas --
%cellvalign; -- alineación vertical en las celdas --
>

```

```
<!--
```

Se usa THEAD para duplicar los encabezados cuando la tabla se extiende a lo largo de varias páginas, o para encabezados estáticos cuando las secciones

TBODY se representan en un panel con desplazamiento.

Se usa THEAD para duplicar los pies de tabla cuando la tabla se extiende a lo largo de varias páginas, o para pies de tabla estáticos cuando las secciones TBODY se representan en un panel con desplazamiento.

Se usan varias secciones TBODY cuando se necesitan líneas de división entre grupos de filas de una tabla.

```
-->
<!ATTLIST (THEAD|TBODY|TFOOT)          -- sección de tabla --
  %attrs;                               -- %coreattrs, %il8n, %events --
  %cellhalign;                          -- alineación horizontal en las celdas --
  %cellvalign;                          -- alineación vertical en las celdas --
>

<!ATTLIST TR                          -- fila de tabla --
  %attrs;                               -- %coreattrs, %il8n, %events --
  %cellhalign;                          -- alineación horizontal en las celdas --
  %cellvalign;                          -- alineación vertical en las celdas --
>

<!-- Para las tablas normales scope es más simple que el atributo headers -->
<!ENTITY % Scope "(row|col|rowgroup|colgroup)">

<!-- TH es para encabezados, TD es para datos,
      y para celdas que actúen como ambos use TD -->
<!ATTLIST (TH|TD)                      -- celda de encabezado o de datos --
  %attrs;                               -- %coreattrs, %il8n, %events --
  abbr          %Text;                  #IMPLIED -- abreviatura para la celda de encabezado --
  axis          CDATA                   #IMPLIED -- lista de encabezados relacionados
                                                separados por comas --
  headers       IDREFS                  #IMPLIED -- lista de id's de celdas de encabezado --
  scope         %Scope;                 #IMPLIED -- campo de acción cubierto por las celdas
                                                de encabezado --
  rowspan       NUMBER                  1      -- número de filas abarcado por la celda --
  colspan       NUMBER                  1      -- número de columnas abarcado por la celda --
  %cellhalign;                          -- alineación horizontal en las celdas --
  %cellvalign;                          -- alineación vertical en las celdas --
>

<!--===== Cabecera del Documento =====>
<!-- %head.misc; definido previamente como "SCRIPT|STYLE|META|LINK|OBJECT" -->
<!ENTITY % head.content "TITLE & BASE?">

<!ELEMENT HEAD O O (%head.content;) +(%head.misc;) -- cabecera del documento -->
<!ATTLIST HEAD
  %il8n;                               -- lang, dir --
  profile      %URI;                    #IMPLIED -- diccionario de metainformación con nombre
--
>

<!-- El elemento TITLE no se considera parte del flujo de texto.
      Debería ser mostrado, por ejemplo, como el encabezado de la página
      o como el título de la ventana. Se requiere exactamente un título
```

por documento.

```
-->
<!ELEMENT TITLE - - (#PCDATA) -(%head.misc;) -- título del documento -->
<!ATTLIST TITLE %i18n>

<!ELEMENT BASE - O EMPTY -- URI base del documento -->
<!ATTLIST BASE
href %URI; #REQUIRED -- URI que actúa como URI base --
>

<!ELEMENT META - O EMPTY -- metainformación genérica -->
<!ATTLIST META
%i18n; -- lang, dir, para usar con content --
http-equiv NAME #IMPLIED -- nombre de encabezado de respuesta HTTP --
name NAME #IMPLIED -- nombre de la metainformación --
content CDATA #REQUIRED -- información asociada --
scheme CDATA #IMPLIED -- seleccionar forma de contenido --
>

<!ELEMENT STYLE - - %StyleSheet -- información de estilo -->
<!ATTLIST STYLE
%i18n; -- lang, dir, para usar con title --
type %ContentType; #REQUIRED -- tipo de contenido del lenguaje de estilo --
media %MediaDesc; #IMPLIED -- diseñado para usar con estos medios --
title %Text; #IMPLIED -- título consultivo --
>

<!ELEMENT SCRIPT - - %Script; -- sentencias de script -->
<!ATTLIST SCRIPT
charset %Charset; #IMPLIED -- codif. de caracteres del recurso
vinculado--
type %ContentType; #REQUIRED -- tipo de contenido del lenguaje de scripts
--
src %URI; #IMPLIED -- URI de un script externo --
defer (defer) #IMPLIED -- Los AAUU pueden aplazar
la ejecución del script --
event CDATA #IMPLIED -- reservado para posibles usos futuros --
for %URI; #IMPLIED -- reservado para posibles usos futuros --
>

<!ELEMENT NOSCRIPT - - (%block;) +
-- contenedor de contenido alternativo para representación no basada en scripts -->
<!ATTLIST NOSCRIPT
%attrs; -- %coreattrs, %i18n, %events --
>

<!--===== Estructura del Documento =====>
<!ENTITY % html.content "HEAD, BODY">

<!ELEMENT HTML O O (%html.content;) -- elemento raíz del documento -->
<!ATTLIST HTML
%i18n; -- lang, dir --
>
```

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

22 Definición del Tipo de Documento Transicional

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

```
<!--
Este es el DTD Transicional de HTML 4.01, que incluye los atributos
y elementos de presentación que el W3C espera que dejen de utilizarse
a medida que madure el soporte de las hojas de estilo. Los autores
deberían usar el DTD Estricto siempre que sea posible, pero pueden
usar el DTD Transicional si es necesario el soporte de atributos
y elementos de presentación.
```

```
HTML 4 incluye mecanismos para hojas de estilo, scripts, objetos
incluidos, soporte mejorado para textos de izquierda a derecha o
direcciones mezcladas, y mejoras en los formularios para facilitar
la accesibilidad por parte de personas con discapacidades.
```

```
Borrador: $Date: 1999/12/24 22:40:35 $
```

```
Autores:
```

```
  Dave Raggett <dsr@w3.org>
  Arnaud Le Hors <lehors@w3.org>
  Ian Jacobs <ij@w3.org>
```

```
Puede encontrar más información sobre HTML 4.01 en:
```

```
http://www.w3.org/TR/1999/REC-html401-19991224
```

```
La especificación de HTML 4.01 incluye restricciones
sintácticas adicionales que no pueden expresarse en
los términos de los DTDs.
```

```
-->
<!ENTITY % HTML.Version "-//W3C//DTD HTML 4.01 Transitional//EN"
-- Uso normal:
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
<head>
...
</head>
<body>
...
</body>
</html>
```

```
El URI utilizado como identificador del sistema en el identificador
público permite al agente de usuario descargar el DTD y los conjuntos
de entidades necesarios.
```

El FPI (Formal Public Identifier) para el DTD Estricto de HTML 4.01 es:

```
"-//W3C//DTD HTML 4.01//EN"
```

Esta versión del DTD Estricto es:

```
http://www.w3.org/TR/1999/REC-html401-19991224/strict.dtd
```

Los autores deberían usar el DTD Estricto a menos que necesiten los controles de presentación para los agentes de usuario que no soporten (adecuadamente) hojas de estilo.

Si va a escribir un documento que incluya marcos, use el siguiente FPI:

```
"-//W3C//DTD HTML 4.01 Frameset//EN"
```

Esta versión del DTD de Documentos con Marcos es:

```
http://www.w3.org/TR/1999/REC-html401-19991224/frameset.dtd
```

Utilice los siguientes URIs (relativos) para hacer referencia a los DTDs y a las definiciones de entidades de esta especificación:

```
"strict.dtd"  
"loose.dtd"  
"frameset.dtd"  
"HTMLlat1.ent"  
"HTMLsymbol.ent"  
"HTMLspecial.ent"
```

-->

```
<!--===== Nombres Importados =====-->
```

```
<!-- Conmutador de Característica para Documentos con Marcos -->
```

```
<!ENTITY % HTML.Frameset "IGNORE">
```

```
<!ENTITY % ContentType "CDATA"
```

```
-- tipo de medio, según \[RFC2045\]
```

```
-->
```

```
<!ENTITY % ContentTypes "CDATA"
```

```
-- lista de tipos de medios separados por comas, según \[RFC2045\]
```

```
-->
```

```
<!ENTITY % Charset "CDATA"
```

```
-- una codificación de caracteres, según \[RFC2045\]
```

```
-->
```

```
<!ENTITY % Charsets "CDATA"
```

```
-- una lista de codificaciones de caracteres separadas por espacios,  
según \[RFC2045\]
```

```
-->
```

```
<!ENTITY % LanguageCode "NAME"
```

```
-- un código de idioma, según \[RFC1766\]
```

```
-->
```

```

<!ENTITY % Character "CDATA"
    -- un carácter individual de \[ISO10646\]
    -->

<!ENTITY % LinkTypes "CDATA"
    -- una lista de tipos de vínculo separados por espacios
    -->

<!ENTITY % MediaDesc "CDATA"
    -- un descriptor de medios, o una lista de ellos separados por comas
    -->

<!ENTITY % URI "CDATA"
    -- un Identificador Uniforme de Recursos, ver \[URI\]
    -->

<!ENTITY % Datetime "CDATA" -- información de fecha y hora. Formato de fechas ISO -->

<!ENTITY % Script "CDATA" -- expresión de script -->

<!ENTITY % StyleSheet "CDATA" -- datos de hoja de estilo -->

<!ENTITY % FrameTarget "CDATA" -- representar en este marco -->

<!ENTITY % Text "CDATA">

<!-- Entidades Paramétricas -->

<!ENTITY % head.misc "SCRIPT | STYLE | META | LINK | OBJECT"
    -- elementos de cabecera repetibles
    -->

<!ENTITY % heading "H1 | H2 | H3 | H4 | H5 | H6">

<!ENTITY % list "UL | OL | DIR | MENU">

<!ENTITY % preformatted "PRE">

<!ENTITY % Color "CDATA" -- un color según sRGB: #RRGGBB con valores hexadecimales
-->

<!-- Además hay 16 nombres de colores muy extendidos con sus valores sRGB:

    Black   = #000000 (Negro)           Green    = #008000 (Verde)
    Silver  = #C0C0C0 (Plata)           Lime     = #00FF00 (Lima)
    Gray    = #808080 (Gris)            Olive    = #808000 (Verde oliva)
    White   = #FFFFFF (Blanco)          Yellow   = #FFFF00 (Amarillo)
    Maroon  = #800000 (Marrón)          Navy     = #000080 (Azul marino)
    Red     = #FF0000 (Rojo)            Blue     = #0000FF (Azul)
    Purple  = #800080 (Púrpura)         Teal     = #008080 (Verde azulado)
    Fuchsia = #FF00FF (Fucsia)          Aqua     = #00FFFF (Celeste)
-->

<!ENTITY % bodycolors "
    bgcolor      %Color;          #IMPLIED -- color de fondo del documento --

```

```

text          %Color;          #IMPLIED -- color de texto del documento --
link          %Color;          #IMPLIED -- color de los vínculos --
vlink        %Color;          #IMPLIED -- color de los vínculos visitados --
alink        %Color;          #IMPLIED -- color de los vínculos seleccionados --
">

```

```
<!--===== Entidades nemónicas de caracteres =====>
```

```

<!ENTITY % HTMLlat1 PUBLIC
    "-//W3C//ENTITIES Latin1//EN//HTML"
    "HTMLlat1.ent">

```

```
%HTMLlat1;
```

```

<!ENTITY % HTMLsymbol PUBLIC
    "-//W3C//ENTITIES Symbols//EN//HTML"
    "HTMLsymbol.ent">

```

```
%HTMLsymbol;
```

```

<!ENTITY % HTMLspecial PUBLIC
    "-//W3C//ENTITIES Special//EN//HTML"
    "HTMLspecial.ent">

```

```
%HTMLspecial;
```

```
<!--===== Atributos Genéricos =====>
```

```

<!ENTITY % coreattrs
    "id          ID          #IMPLIED -- identificador único a nivel de documento --
    class       CDATA       #IMPLIED -- lista de clases separadas por comas --
    style       %StyleSheet; #IMPLIED -- información de estilo asociada --
    title       %Text;       #IMPLIED -- título consultivo --"
    >

```

```

<!ENTITY % i18n
    "lang        %LanguageCode; #IMPLIED -- código de idioma --
    dir          (ltr|rtl)    #IMPLIED -- dirección del texto débil/neutral --"
    >

```

```

<!ENTITY % events
    "onclick     %Script;       #IMPLIED -- se hizo clic con un botón del apuntador --
    ondblclick  %Script;       #IMPLIED -- se hizo doble clic con un botón
    del apuntador --
    onmousedown %Script;       #IMPLIED -- se pulsó un botón del apuntador --
    onmouseup   %Script;       #IMPLIED -- se soltó un botón del apuntador --
    onmouseover %Script;       #IMPLIED -- se movió un apuntador al interior de --
    onmousemove %Script;       #IMPLIED -- se movió un apuntador en el interior de --
    onmouseout  %Script;       #IMPLIED -- se alejó un apuntador --
    onkeypress  %Script;       #IMPLIED -- se pulso y soltó una tecla --
    onkeydown   %Script;       #IMPLIED -- se pulsó una tecla --
    onkeyup     %Script;       #IMPLIED -- se soltó una tecla --"
    >

```

```
<!-- Conmutador de Característica Reservada -->
```

```
<!ENTITY % HTML.Reserved "IGNORE">
```

```
<!-- Los siguientes atributos están reservados para su posible uso futuro -->
```

```
<![ %HTML.Reserved; [
```

```

<!ENTITY % reserved
    "datasrc     %URI;          #IMPLIED -- una fuente de datos individual o tabular --

```

```

datafld      CDATA          #IMPLIED  -- el nombre de la propiedad o de la columna
--
dataformatas (plaintext|html) plaintext -- texto o html --"
>
]]>

<!ENTITY % reserved "">

<!ENTITY % attrs "%coreattrs; %il8n; %events;">

<!ENTITY % align "align (left|center|right|justify) #IMPLIED"
-- por defecto es left para párrafos de izquierda a derecha,
-- right para derecha a izquierda --
>

<!--===== Códigos para Texto =====>

<!ENTITY % fontstyle
"TT | I | B | U | S | STRIKE | BIG | SMALL">

<!ENTITY % phrase "EM | STRONG | DFN | CODE |
SAMP | KBD | VAR | CITE | ABBR | ACRONYM" >

<!ENTITY % special
"A | IMG | APPLET | OBJECT | FONT | BASEFONT | BR | SCRIPT |
MAP | Q | SUB | SUP | SPAN | BDO | IFRAME">

<!ENTITY % formctrl "INPUT | SELECT | TEXTAREA | LABEL | BUTTON">

<!-- %inline; cubre los elementos en línea o "a nivel de texto" -->
<!ENTITY % inline "#PCDATA | %fontstyle; | %phrase; | %special; | %formctrl;">

<!ELEMENT (%fontstyle;|%phrase;) - - (%inline;)*>
<!ATTLIST (%fontstyle;|%phrase;)
%attrs;          -- %coreattrs, %il8n, %events --
>

<!ELEMENT (SUB|SUP) - - (%inline;)*      -- subíndice, superíndice -->
<!ATTLIST (SUB|SUP)
%attrs;          -- %coreattrs, %il8n, %events --
>

<!ELEMENT SPAN - - (%inline;)*          -- contenedor genérico de idioma/estilo -->
<!ATTLIST SPAN
%attrs;          -- %coreattrs, %il8n, %events --
%reserved;      -- reservado para posibles usos futuros --
>

<!ELEMENT BDO - - (%inline;)*          -- Anular BiDi I18N -->
<!ATTLIST BDO
%coreattrs;      -- id, class, style, title --
lang             %LanguageCode; #IMPLIED -- código de idioma --
dir              (ltr|rtl)      #REQUIRED -- direccionalidad --
>

<!ELEMENT BASEFONT - O EMPTY          -- tamaño base de la fuente -->
<!ATTLIST BASEFONT
id               ID             #IMPLIED -- identificador único en el documento --

```

```

  size          CDATA          #REQUIRED -- tamaño base de fuente para elementos FONT
--
  color         %Color;        #IMPLIED  -- color del texto --
  face          CDATA          #IMPLIED  -- lista de nombres de fuente
                          separados por comas --
>

<!ELEMENT FONT - - (%inline;)* -- cambio local de fuente -->
<!ATTLIST FONT
  %coreattrs; -- id, class, style, title --
  %il8n;     -- lang, dir --
  size       CDATA       #IMPLIED  -- [+|-]nn p.ej. size="+1", size="4" --
  color     %Color;    #IMPLIED  -- color del texto --
  face     CDATA     #IMPLIED  -- lista de nombres de fuente
                          separados por comas --
>

<!ELEMENT BR - O EMPTY -- salto de línea forzado -->
<!ATTLIST BR
  %coreattrs; -- id, class, style, title --
  clear     (left|all|right|none) none -- control del flujo de texto --
>

<!--===== Modelos de contenido de HTML =====>

<!--
  HTML tiene dos modelos de contenido básicos:

  %inline;     elementos a nivel de carácter y cadenas de texto
  %block;     elementos tipo bloque, p.ej., párrafos y listas
-->

<!ENTITY % block
  "P | %heading; | %list; | %preformatted; | DL | DIV | CENTER |
  NOSCRIPT | NOFRAMES | BLOCKQUOTE | FORM | ISINDEX | HR |
  TABLE | FIELDSET | ADDRESS">

<!ENTITY % flow "%block; | %inline;">

<!--===== Cuerpo del Documento =====>

<!ELEMENT BODY O O (%flow;)* +(INS|DEL) -- cuerpo del documento -->
<!ATTLIST BODY
  %attrs; -- %coreattrs, %il8n, %events --
  onload   %Script; #IMPLIED  -- el documento ha sido cargado --
  onunload %Script; #IMPLIED  -- el documento ha sido quitado --
  background %URI; #IMPLIED  -- patrón de relleno para el fondo
                          del documento --
  %bodycolors; -- bgcolor, text, link, vlink, alink --
>

<!ELEMENT ADDRESS - - ((%inline;)|P)* -- información sobre al autor -->
<!ATTLIST ADDRESS
  %attrs; -- %coreattrs, %il8n, %events --
>

<!ELEMENT DIV - - (%flow;)* -- contenedor genérico de idioma/estilo -->

```



```

<!ATTLIST DIV
  %attrs;          -- %coreattrs, %il8n, %events --
  %align;          -- align, alineación del texto --
  %reserved;      -- reservado para posibles usos futuros --
>

<!ELEMENT CENTER - - (%flow;)*          -- forma abreviada de DIV align=center -->
<!ATTLIST CENTER
  %attrs;          -- %coreattrs, %il8n, %events --
>

<!--===== El Elemento Ancla (origen/destino de vínculo) =====>

<!ENTITY % Shape "(rect|circle|poly|default)">
<!ENTITY % Coords "CDATA" -- lista de longitudes separadas por comas -->

<!ELEMENT A - - (%inline;)* -(A)        -- origen o destino de vínculo -->
<!ATTLIST A
  %attrs;          -- %coreattrs, %il8n, %events --
  charset          %Charset;            #IMPLIED -- codificación de caracteres
                                     del recurso vinculado --
  type            %ContentType;         #IMPLIED -- tipo consultivo de contenido --
  name            CDATA                 #IMPLIED -- destino de vínculo con nombre --
  href            %URI;                 #IMPLIED -- URI del recurso vinculado --
  hreflang       %LanguageCode;        #IMPLIED -- código de idioma --
  target         %FrameTarget;         #IMPLIED -- representar en este marco --
  rel            %LinkTypes;           #IMPLIED -- tipos de vínculos directos --
  rev            %LinkTypes;           #IMPLIED -- tipos de vínculos inversos --
  accesskey      %Character;           #IMPLIED -- carácter de la tecla de accesibilidad --
  shape          %Shape;                rect    -- para mapas de imágenes en el cliente --
  coords         %Coords;               #IMPLIED -- para mapas de imágenes en el cliente --
  tabindex       NUMBER                 #IMPLIED -- posición en el orden de tabulación --
  onfocus       %Script;               #IMPLIED -- el foco se dirigió hacia el elemento --
  onblur        %Script;               #IMPLIED -- el elemento perdió el foco --
>

<!--===== Mapas de imágenes en el lado del cliente =====>

<!-- Éstos pueden colocarse en el mismo documento o agrupados en un
      documento separado, aunque esto último no está todavía muy soportado -->

<!ELEMENT MAP - - ((%block;) | AREA)+ -- mapa de imágenes en el lado del cliente -->
<!ATTLIST MAP
  %attrs;          -- %coreattrs, %il8n, %events --
  name            CDATA                 #REQUIRED -- como referencia para usemap --
>

<!ELEMENT AREA - O EMPTY                -- area de un mapa de imágenes
                                     en el lado del cliente -->
<!ATTLIST AREA
  %attrs;          -- %coreattrs, %il8n, %events --
  shape          %Shape;                rect    -- controla la interpretación de las coords --
  coords         %Coords;               #IMPLIED -- lista de longitudes separadas por comas --
  href           %URI;                  #IMPLIED -- URI del recurso vinculado --
  target        %FrameTarget;          #IMPLIED -- representar en este marco --
  nohref        (nohref)               #IMPLIED -- esta región no tiene acción --

```

```

alt           %Text;           #REQUIRED -- descripción corta --
tabindex      NUMBER          #IMPLIED  -- posición en el orden de tabulación --
accesskey     %Character;     #IMPLIED  -- carácter de la tecla de accesibilidad --
onfocus      %Script;       #IMPLIED  -- el foco se dirigió hacia el elemento --
onblur        %Script;       #IMPLIED  -- el elemento perdió el foco --
>

```

```
<!--===== El Elemento LINK (vínculo) =====>
```

```
<!--
En principio los valores de las relaciones se pueden usar:
```

- a) para menús/barras de herramientas específicas del documento que usen los elementos LINK de la cabecera del documento, p.ej., start (comienzo), contents (contenidos), previous (anterior), next (siguiente), index (índice), end (final), help (ayuda)
- b) para vincular una hoja de estilo externa (rel=stylesheet)
- c) para vincular un script (rel=script)
- d) por las hojas de estilo para controlar cómo se representan los conjuntos de nodos html en documentos impresos
- e) para crear un vínculo a una versión imprimible del documento, p.ej., una versión postscript o pdf (rel=alternate media=print)

```
-->
```

```
<!ELEMENT LINK - O EMPTY -- un vínculo independiente del medio -->
```

```
<!ATTLIST LINK
```

```

%attrs;           -- %coreattrs, %i18n, %events --
charset           %Charset;     #IMPLIED  -- codificación de caracteres
                 del recurso vinculado --
href              %URI;         #IMPLIED  -- URI del recurso vinculado --
hreflang          %LanguageCode; #IMPLIED  -- código de idioma --
type              %ContentType; #IMPLIED  -- tipo consultivo de contenido --
rel               %LinkTypes;   #IMPLIED  -- tipos de vínculos directos --
rev               %LinkTypes;   #IMPLIED  -- tipos de vínculos inversos --
media             %MediaDesc;   #IMPLIED  -- para representar en estos medios --
target            %FrameTarget; #IMPLIED  -- representar en este marco --
>

```

```
<!--===== Imágenes =====>
```

```

<!-- Length definido en el DTD estricto para cellpadding/cellspacing -->
<!ENTITY % Length "CDATA" -- nn para píxeles o nn% para longitudes porcentuales -->
<!ENTITY % MultiLength "CDATA" -- píxel, porcentaje, o relativa -->

```

```

<![ %HTML.Frameset; [
<!ENTITY % MultiLengths "CDATA" -- lista de MultiLength separadas por comas -->
]]>

```

```
<!ENTITY % Pixels "CDATA" -- entero que representa una longitud en píxeles -->
```

```
<!ENTITY % IAlign "(top|middle|bottom|left|right)" -- ¿centrar? -->
```

```

<!-- Para evitar problemas con AAUU de sólo texto así
como para hacer el contenido de las imágenes inteligibles
y navegables para los usuarios de AAUU no visuales, es
necesario proporcionar una descripción con ALT, y evitar

```

mapas de imágenes en el lado del servidor -->

```

<!ELEMENT IMG - O EMPTY -- Imagen incluida -->
<!ATTLIST IMG
  %attrs; -- %coreattrs, %i18n, %events --
  src %URI; #REQUIRED -- URI de la imagen a incluir --
  alt %Text; #REQUIRED -- descripción corta --
  longdesc %URI; #IMPLIED -- vínculo a descripción larga
    (complementa a alt) --
  name CDATA #IMPLIED -- nombre de la imagen para los scripts --
  height %Length; #IMPLIED -- especificar nueva altura --
  width %Length; #IMPLIED -- especificar nueva anchura --
  usemap %URI; #IMPLIED -- usar mapa de imágenes en el cliente --
  ismap (ismap) #IMPLIED -- usar mapa de imágenes en el servidor --
  align %IAAlign; #IMPLIED -- alineación vertical u horizontal --
  border %Pixels; #IMPLIED -- grosor del borde del vínculo --
  hspace %Pixels; #IMPLIED -- espacio horizontal --
  vspace %Pixels; #IMPLIED -- espacio vertical --
>

<!-- USEMAP apunta a un elemento MAP que puede estar en este documento
o en un documento externo, aunque esto último no está muy soportado -->

<!--===== OBJECT =====>
<!--
Se usa OBJECT para incluir objetos como parte de una página HTML.
Los elementos PARAM deberían preceder a cualquier otro contenido.
Aspectos técnicos del modelo de contenido mixto de SGML impiden
que esto se especifique formalmente...
-->

<!ELEMENT OBJECT - - (PARAM | %flow;)*
-- objeto genérico incluido -->
<!ATTLIST OBJECT
  %attrs; -- %coreattrs, %i18n, %events --
  declare (declare) #IMPLIED -- declarar pero no crear --
  classid %URI; #IMPLIED -- identifica una implementación --
  codebase %URI; #IMPLIED -- URI base para classid, datos, archivo --
  data %URI; #IMPLIED -- referencia a los datos del objeto --
  type %ContentType; #IMPLIED -- tipo de contenido de los datos --
  codetype %ContentType; #IMPLIED -- tipo de contenido del código --
  archive CDATA #IMPLIED -- lista de URIs separados por espacios --
  standby %Text; #IMPLIED -- mensaje a mostrar mientras se carga --
  height %Length; #IMPLIED -- especificar nueva altura --
  width %Length; #IMPLIED -- especificar nueva anchura --
  usemap %URI; #IMPLIED -- usar mapa de imágenes en el cliente --
  name CDATA #IMPLIED -- enviar como parte de un formulario --
  tabindex NUMBER #IMPLIED -- posición en el orden de tabulación --
  align %IAAlign; #IMPLIED -- alineación vertical u horizontal --
  border %Pixels; #IMPLIED -- grosor del borde del vínculo --
  hspace %Pixels; #IMPLIED -- espacio horizontal --
  vspace %Pixels; #IMPLIED -- espacio vertical --
  %reserved; -- reservado para posibles usos futuros --
>

<!ELEMENT PARAM - O EMPTY -- valor de propiedad con nombre -->
<!ATTLIST PARAM

```

Definición del Tipo de Documento HTML 4 Transicional

```

id           ID           #IMPLIED -- identificador único a nivel de documento --
name        CDATA        #REQUIRED -- nombre de la propiedad --
value       CDATA        #IMPLIED -- valor de la propiedad --
valuetype   (DATA|REF|OBJECT) DATA -- cómo interpretar el valor --
type        %ContentType; #IMPLIED -- tipo de contenido para el valor
                                     cuando valuetype=ref --
>

<!--===== APPLLET Java =====>
<!--
  Debe existir o el atributo code o el atributo object.
  Los elementos PARAM se colocan antes de cualquier otro contenido.
-->
<!ELEMENT APPLET - - (PARAM | %flow;)* -- Applet Java -->
<!ATTLIST APPLET
  %coreattrs;           -- id, class, style, title --
  codebase       %URI;           #IMPLIED -- URI base opcional para el applet --
  archive        CDATA          #IMPLIED -- lista de archivos separados por comas --
  code           CDATA          #IMPLIED -- fichero de la clase del applet --
  object         CDATA          #IMPLIED -- fichero applet serializado --
  alt            %Text;         #IMPLIED -- descripción corta --
  name           CDATA          #IMPLIED -- permite que los applets
                                     se encuentren entre sí --
  width          %Length;       #REQUIRED -- anchura inicial --
  height         %Length;       #REQUIRED -- altura inicial --
  align          %IAalign;      #IMPLIED -- alineación vertical u horizontal --
  hspace         %Pixels;      #IMPLIED -- espacio horizontal --
  vspace        %Pixels;      #IMPLIED -- espacio vertical --
>

<!--===== Separador Horizontal =====>
<!ELEMENT HR - O EMPTY -- separador horizontal -->
<!ATTLIST HR
  %attrs;           -- %coreattrs, %il8n, %events --
  align            (left|center|right) #IMPLIED
  noshade          (noshade)         #IMPLIED
  size             %Pixels;       #IMPLIED
  width            %Length;       #IMPLIED
>

<!--===== Párrafos =====>
<!ELEMENT P - O (%inline;)* -- párrafo -->
<!ATTLIST P
  %attrs;           -- %coreattrs, %il8n, %events --
  %align;          -- align, alineación del texto --
>

<!--===== Encabezados =====>
<!--
  Hay seis niveles de encabezados, desde H1 (el más importante)
  hasta H6 (el menos importante).
-->
<!ELEMENT (%heading;) - - (%inline;)* -- encabezado -->

```

```

<!ATTLIST (%heading;)
  %attrs;          -- %coreattrs, %il8n, %events --
  %align;          -- align, alineación del texto --
  >

<!--===== Texto Preformateado =====>

<!-- excluye los códigos de imágenes y de cambios del tamaño de la fuente -->
<!ENTITY % pre.exclusion "IMG|OBJECT|APPLET|BIG|SMALL|SUB|SUP|FONT|BASEFONT">

<!ELEMENT PRE - - (%inline;)* -(%pre.exclusion;) -- texto preformateado -->
<!ATTLIST PRE
  %attrs;          -- %coreattrs, %il8n, %events --
  width            NUMBER            #IMPLIED
  >

<!--===== Citas en Línea =====>

<!ELEMENT Q - - (%inline;)*          -- cita corta en línea -->
<!ATTLIST Q
  %attrs;          -- %coreattrs, %il8n, %events --
  cite             %URI;            #IMPLIED -- URI del documento o mensaje original --
  >

<!--===== Citas en Bloque =====>

<!ELEMENT BLOCKQUOTE - - (%flow;)*  -- cita larga -->
<!ATTLIST BLOCKQUOTE
  %attrs;          -- %coreattrs, %il8n, %events --
  cite             %URI;            #IMPLIED -- URI del documento o mensaje original --
  >

<!--===== Texto Insertado/Borrado =====>

<!-- INS/DEL se utilizan incluyéndolos en BODY -->
<!ELEMENT (INS|DEL) - - (%flow;)*      -- texto insertado, texto borrado -->
<!ATTLIST (INS|DEL)
  %attrs;          -- %coreattrs, %il8n, %events --
  cite             %URI;            #IMPLIED -- información sobre la razón del cambio --
  datetime        %Datetime;      #IMPLIED -- fecha y hora del cambio --
  >

<!--===== Listas =====>

<!-- listas de definiciones - DT para el término, DD para su definición -->

<!ELEMENT DL - - (DT|DD)+            -- lista de definiciones -->
<!ATTLIST DL
  %attrs;          -- %coreattrs, %il8n, %events --
  compact         (compact)          #IMPLIED -- espacio entre objetos reducido --
  >

<!ELEMENT DT - O (%inline;)*        -- término definido -->
<!ELEMENT DD - O (%flow;)*          -- descripción de la definición -->
<!ATTLIST (DT|DD)
  %attrs;          -- %coreattrs, %il8n, %events --
  >

```

```
<!-- Estilo de numeración de listas ordenadas (OL)
```

1	números arábigos	1, 2, 3, ...
a	alfabético minúscula	a, b, c, ...
A	alfabético mayúscula	A, B, C, ...
i	romanos minúscula	i, ii, iii, ...
I	romanos mayúscula	I, II, III, ...

Se aplica el estilo al número de la secuencia, que por defecto se inicializa a 1 para el primer objeto de una lista ordenada

Esto no puede expresarse directamente en SGML debido a la multiplicidad de casos.

```
-->
```

```
<!ENTITY % OLStyle "CDATA" -- restringido a: "(1|a|A|i|I)" -->
```

```
<!ELEMENT OL - - (LI)+ -- lista ordenada -->
<!ATTLIST OL
  %attrs; -- %coreattrs, %il8n, %events --
  type %OLStyle; #IMPLIED -- estilo de numeración --
  compact (compact) #IMPLIED -- espacio entre objetos reducido --
  start NUMBER #IMPLIED -- número inicial de la secuencia --
>
```

```
<!-- Estilos de marcador de listas no ordenadas (UL) -->
```

```
<!ENTITY % ULStyle "(disc|square|circle)">
```

```
<!ELEMENT UL - - (LI)+ -- lista no ordenada -->
<!ATTLIST UL
  %attrs; -- %coreattrs, %il8n, %events --
  type %ULStyle; #IMPLIED -- estilo del marcador --
  compact (compact) #IMPLIED -- espacio entre objetos reducido --
>
```

```
<!ELEMENT (DIR|MENU) - - (LI)+ -(%block;) -- lista tipo directorio, tipo menú -->
```

```
<!ATTLIST DIR
  %attrs; -- %coreattrs, %il8n, %events --
  compact (compact) #IMPLIED -- espacio entre objetos reducido --
>
<!ATTLIST MENU
  %attrs; -- %coreattrs, %il8n, %events --
  compact (compact) #IMPLIED -- espacio entre objetos reducido --
>
```

```
<!ENTITY % LIStyle "CDATA" -- restringido a: "(%ULStyle;|%OLStyle;)" -->
```

```
<!ELEMENT LI - O (%flow;)* -- objeto de lista -->
<!ATTLIST LI
  %attrs; -- %coreattrs, %il8n, %events --
  type %LIStyle; #IMPLIED -- estilo de objeto de lista --
  value NUMBER #IMPLIED -- reinicializar número de secuencia --
>
```

```
<!--===== Formularios =====>
```

```
<!ELEMENT FORM - - (%flow;)* -(FORM) -- formulario interactivo -->
```

```
<!ATTLIST FORM
```

```

%attrs; -- %coreattrs, %i18n, %events --
action      %URI;          #REQUIRED -- procesador del formulario en el servidor --
method      (GET|POST)     GET         -- método HTTP usado para enviar formulario --
enctype     %ContentType;  "application/x-www-form-urlencoded"
accept      %ContentTypes; #IMPLIED   -- lista de tipos MIME para subir ficheros --
name        CDATA         #IMPLIED   -- nombre del formulario para los scripts --
onsubmit    %Script;      #IMPLIED   -- el formulario fue enviado --
onreset     %Script;      #IMPLIED   -- el formulario fue reinicializado --
target      %FrameTarget; #IMPLIED   -- representar en este marco --
accept-charset %Charsets; #IMPLIED   -- lista de codificaciones de caracteres
soportadas --

>

<!-- Cada rótulo (label) no debe contener MÁS de un campo -->
<!ELEMENT LABEL - - (%inline;)* -(LABEL) -- texto del rótulo de un
                                campo de formulario -->
<!ATTLIST LABEL
%attrs; -- %coreattrs, %i18n, %events --
for         IDREF         #IMPLIED   -- concuerda con el valor ID del campo --
accesskey   %Character;   #IMPLIED   -- carácter de la tecla de accesibilidad --
onfocus    %Script;      #IMPLIED   -- el foco se dirigió hacia el elemento --
onblur      %Script;      #IMPLIED   -- el elemento perdió el foco --

>

<!ENTITY % InputType
"(TEXT | PASSWORD | CHECKBOX |
RADIO | SUBMIT | RESET |
FILE | HIDDEN | IMAGE | BUTTON)"
>

<!-- se requiere el atributo name para todos excepto para submit y reset -->
<!ELEMENT INPUT - O EMPTY -- control de formulario -->
<!ATTLIST INPUT
%attrs; -- %coreattrs, %i18n, %events --
type      %InputType;     TEXT        -- qué tipo de control se necesita --
name      CDATA           #IMPLIED   -- enviar como parte del formulario --
value     CDATA           #IMPLIED   -- especificar para radiobotones
                                y casillas de verificación --
checked   (checked)       #IMPLIED   -- para radiobotones
                                y casillas de verificación --
disabled  (disabled)     #IMPLIED   -- no disponible en este contexto --
readonly  (readonly)     #IMPLIED   -- para texto y contraseñas --
size      CDATA           #IMPLIED   -- específico de cada tipo de campo --
maxlength NUMBER         #IMPLIED   -- máximo de caracteres para campos de texto

--
src       %URI;           #IMPLIED   -- para campos con imágenes --
alt       CDATA           #IMPLIED   -- descripción corta --
usemap    %URI;           #IMPLIED   -- usar mapa de imágenes en el cliente --
ismap     (ismap)        #IMPLIED   -- usar mapa de imágenes en el servidor --
tabindex  NUMBER         #IMPLIED   -- posición en el orden de tabulación --
accesskey %Character;     #IMPLIED   -- carácter de la tecla de accesibilidad --
onfocus  %Script;       #IMPLIED   -- el foco se dirigió hacia el elemento --
onblur    %Script;       #IMPLIED   -- el elemento perdió el foco --
onselect  %Script;       #IMPLIED   -- se seleccionó parte del texto --
onchange  %Script;       #IMPLIED   -- el valor del elemento fue modificado --
accept    %ContentTypes; #IMPLIED   -- lista de tipos MIME para subir ficheros --

```

```

align          %IAalign;          #IMPLIED    -- alineación vertical u horizontal --
%reserved;    -- reservado para posibles usos futuros --
>

<!ELEMENT SELECT - - (OPTGROUP|OPTION)+ -- selector de opciones -->
<!ATTLIST SELECT
  %attrs;      -- %coreattrs, %i18n, %events --
  name         CDATA              #IMPLIED    -- nombre del campo --
  size         NUMBER             #IMPLIED    -- filas visibles --
  multiple     (multiple)        #IMPLIED    -- por defecto es selección simple --
  disabled     (disabled)        #IMPLIED    -- no disponible en este contexto --
  tabindex     NUMBER             #IMPLIED    -- posición en el orden de tabulación --
  onfocus     %Script;          #IMPLIED    -- el foco se dirigió hacia el elemento --
  onblur      %Script;          #IMPLIED    -- el elemento perdió el foco --
  onchange     %Script;          #IMPLIED    -- el valor del elemento fue modificado --
  %reserved;   -- reservado para posibles usos futuros --
>

<!ELEMENT OPTGROUP - - (OPTION)+ -- grupo de opciones -->
<!ATTLIST OPTGROUP
  %attrs;      -- %coreattrs, %i18n, %events --
  disabled     (disabled)        #IMPLIED    -- no disponible en este contexto --
  label       %Text;            #REQUIRED   -- para usar en menús jerárquicos --
>

<!ELEMENT OPTION - 0 (#PCDATA) -- opción seleccionable -->
<!ATTLIST OPTION
  %attrs;      -- %coreattrs, %i18n, %events --
  selected     (selected)        #IMPLIED    -- opción preseleccionada --
  disabled     (disabled)        #IMPLIED    -- no disponible en este contexto --
  label       %Text;            #IMPLIED    -- para usar en menús jerárquicos --
  value       CDATA              #IMPLIED    -- por defecto es el contenido del elemento --
>

<!ELEMENT TEXTAREA - - (#PCDATA) -- campo de texto multilínea -->
<!ATTLIST TEXTAREA
  %attrs;      -- %coreattrs, %i18n, %events --
  name         CDATA              #IMPLIED
  rows         NUMBER             #REQUIRED
  cols         NUMBER             #REQUIRED
  disabled     (disabled)        #IMPLIED    -- no disponible en este contexto --
  readonly     (readonly)        #IMPLIED
  tabindex     NUMBER             #IMPLIED    -- posición en el orden de tabulación --
  accesskey    %Character;       #IMPLIED    -- carácter de la tecla de accesibilidad --
  onfocus     %Script;          #IMPLIED    -- el foco se dirigió hacia el elemento --
  onblur      %Script;          #IMPLIED    -- el elemento perdió el foco --
  onselect     %Script;          #IMPLIED    -- se seleccionó parte del texto --
  onchange     %Script;          #IMPLIED    -- el valor del elemento fue modificado --
  %reserved;   -- reservado para posibles usos futuros --
>

<!--
#PCDATA es para resolver el problema del contenido mixto,
de acuerdo con la especificación aquí sólo se permite espacio en blanco!
-->
<!ELEMENT FIELDSET - - (#PCDATA,LEGEND,(%flow;)* ) -- grupo de controles

```


de formulario -->

```

<!ATTLIST FIELDSET
  %attrs;                -- %coreattrs, %il8n, %events --
>

<!ELEMENT LEGEND - - (%inline;)*      -- leyenda del grupo de campos -->
<!ENTITY % LAlign "(top|bottom|left|right)">

<!ATTLIST LEGEND
  %attrs;                -- %coreattrs, %il8n, %events --
  accesskey  %Character;  #IMPLIED  -- carácter de la tecla de accesibilidad --
  align      %LAlign;     #IMPLIED  -- con relación al grupo de campos --
>

<!ELEMENT BUTTON - -
  (%flow;)* - (A|%formctrl;|FORM|ISINDEX|FIELDSET|IFRAME)
  -- botón (pulsador) -->
<!ATTLIST BUTTON
  %attrs;                -- %coreattrs, %il8n, %events --
  name          CDATA      #IMPLIED
  value         CDATA      #IMPLIED  -- se envía al servidor junto al formulario --
  type         (button|submit|reset) submit -- para usarlo como botón de formulario --
  disabled     (disabled)  #IMPLIED  -- no disponible en este contexto --
  tabindex     NUMBER      #IMPLIED  -- posición en el orden de tabulación --
  accesskey    %Character;  #IMPLIED  -- carácter de la tecla de accesibilidad --
  onfocus     %Script;    #IMPLIED  -- el foco se dirigió hacia el elemento --
  onblur       %Script;    #IMPLIED  -- el elemento perdió el foco --
  %reserved;   -- reservado para posibles usos futuros --
>

<!--===== Tablas =====>

<!-- Estándar de tablas HTML IETF, ver \[RFC1942\] -->

<!--
El atributo BORDER establece el grosor del marco que rodea a la
tabla. Las unidades por defecto son píxeles de pantalla.

El atributo FRAME especifica qué partes del marco que rodea a la
tabla deberían representarse. Los valores no coinciden con los de
CALs para evitar un conflicto de nombres con el atributo VALIGN.

El valor "border" se ha incluido por compatibilidad con
<TABLE BORDER>, que produce frame=border y border=implied.
Para <TABLE BORDER=1> se obtiene border=1 y frame=implied.
En este caso, es apropiado tratar esto como frame=border por
compatibilidad con los navegadores existentes.
-->
<!ENTITY % TFrame "(void|above|below|hsides|lhs|rhs|vsides|box|border)">

<!--
El atributo RULES define qué líneas de división entre celdas se van a dibujar:

Si RULES no está presente entonces se hace igual a:
"none" si BORDER está ausente o BORDER=0, "all" en caso contrario
-->
<!ENTITY % TRules "(none | groups | rows | cols | all)">

```

```

<!-- colocación horizontal de la tabla en relación al documento -->
<!ENTITY % TAlign "(left|center|right)">

<!-- atributos de alineación horizontal de los contenidos de las celdas -->
<!ENTITY % cellhalign
"align      (left|center|right|justify|char) #IMPLIED
 char       %Character;      #IMPLIED  -- carácter de alineación, p.ej. char=':' --
 charoff    %Length;        #IMPLIED  -- offset del carácter de alineación --"
>

<!-- atributos de alineación vertical de los contenidos de las celdas -->
<!ENTITY % cellvalign
"valign     (top|middle|bottom|baseline) #IMPLIED"
>

<!ELEMENT TABLE - -
      (CAPTION?, (COL*|COLGROUP*), THEAD?, TFOOT?, TBODY+)>
<!ELEMENT CAPTION - - (%inline;)*      -- título de la tabla -->
<!ELEMENT THEAD - 0 (TR)+              -- cabecera de tabla -->
<!ELEMENT TFOOT - 0 (TR)+             -- pie de tabla -->
<!ELEMENT TBODY 0 0 (TR)+             -- cuerpo de tabla -->
<!ELEMENT COLGROUP - 0 (COL)*         -- grupo de columnas de tabla -->
<!ELEMENT COL - 0 EMPTY               -- columna de tabla -->
<!ELEMENT TR - 0 (TH|TD)+            -- fila de tabla -->
<!ELEMENT (TH|TD) - 0 (%flow;)*      -- celda de encabezado de tabla,
                                     celda de datos de tabla -->

<!ATTLIST TABLE -- elemento tabla --
  %attrs; -- %coreattrs, %il8n, %events --
  summary %Text; #IMPLIED -- propósito/estructura para salida por voz --
  width %Length; #IMPLIED -- anchura de tabla --
  border %Pixels; #IMPLIED -- controla el marco que rodea a la tabla --
  frame %TFrame; #IMPLIED -- qué partes del marco representar --
  rules %TRules; #IMPLIED -- líneas de división entre filas y columnas
--
  cellspacing %Length; #IMPLIED -- espaciado entre celdas --
  cellpadding %Length; #IMPLIED -- espacio dentro de las celdas --
  align %TAlign; #IMPLIED -- posición de la tabla respecto a la
ventana--
  bgcolor %Color; #IMPLIED -- color de fondo de las celdas --
  %reserved; -- reservado para posibles usos futuros --
  datapagesize CDATA #IMPLIED -- reservado para posibles usos futuros --
>

<!ENTITY % CAlign "(top|bottom|left|right)">

<!ATTLIST CAPTION
  %attrs; -- %coreattrs, %il8n, %events --
  align %CAlign; #IMPLIED -- relativo a la tabla --
>

<!--
COLGROUP agrupa un conjunto de elementos COL. Permite agrupar
varias columnas relacionadas semánticamente.
-->
<!ATTLIST COLGROUP

```

```

%attrs; -- %coreattrs, %i18n, %events --
span      NUMBER      1      -- número por defecto de columnas en el
grupo--
width     %MultiLength; #IMPLIED -- anchura por defecto para
las COLs contenidas --
%cellhalign; -- alineación horizontal en las celdas --
%cellvalign; -- alineación vertical en las celdas --
>

```

```
<!--
```

Los elementos COL definen las propiedades de alineación de las celdas de una o más columnas.

El atributo WIDTH especifica la anchura de las columnas, p.ej.

```

width=64      anchura en píxeles de pantalla
width=0.5*    anchura relativa de 0.5

```

El atributo SPAN hace que los atributos de un elemento COL se apliquen a más de una columna.

```
-->
```

```

<!ATTLIST COL -- grupos y propiedades de columnas --
%attrs; -- %coreattrs, %i18n, %events --
span      NUMBER      1      -- Los atributos de COL afectan a N columnas
--
width     %MultiLength; #IMPLIED -- especificación de la anchura
de las columnas --
%cellhalign; -- alineación horizontal en las celdas --
%cellvalign; -- alineación vertical en las celdas --
>

```

```
<!--
```

Se usa THEAD para duplicar los encabezados cuando la tabla se extiende a lo largo de varias páginas, o para encabezados estáticos cuando las secciones TBODY se representan en un panel con desplazamiento.

Se usa THEAD para duplicar los pies de tabla cuando la tabla se extiende a lo largo de varias páginas, o para pies de tabla estáticos cuando las secciones TBODY se representan en un panel con desplazamiento.

Se usan varias secciones TBODY cuando se necesitan líneas de división entre grupos de filas de una tabla.

```
-->
```

```

<!ATTLIST (THEAD|TBODY|TFOOT) -- sección de tabla --
%attrs; -- %coreattrs, %i18n, %events --
%cellhalign; -- alineación horizontal en las celdas --
%cellvalign; -- alineación vertical en las celdas --
>

```

```
<!ATTLIST TR
```

```

%attrs; -- %coreattrs, %i18n, %events --
%cellhalign; -- alineación horizontal en celdas --
%cellvalign; -- alineación vertical en celdas --
bgcolor   %Color;     #IMPLIED -- color de fondo de la celda --
>

```

```

<!-- Para las tablas normales scope es más simple que el atributo headers -->
<!ENTITY % Scope "(row|col|rowgroup|colgroup)">

<!-- TH es para encabezados, TD es para datos,
      y para celdas que actúen como ambos use TD -->
<!ATTLIST (TH|TD)
  %attrs;                -- celda de encabezado o de datos --
  %coreattrs, %il8n, %events --
  abbr      %Text;      #IMPLIED -- abreviatura para la celda de encabezado --
  axis      CDATA      #IMPLIED -- lista de encabezados relacionados
                                     separados por comas --
  headers   IDREFS     #IMPLIED -- lista de id's de celdas de encabezado --
  scope     %Scope;    #IMPLIED -- campo de acción cubierto por
                                     las celdas de encabezado --
  rowspan   NUMBER     1      -- número de filas abarcado por la celda --
  colspan   NUMBER     1      -- número de columnas abarcado por la celda --
  %cellhalign;          -- alineación horizontal en celdas --
  %cellvalign;         -- alineación vertical en celdas --
  nowrap    (nowrap)   #IMPLIED -- suprimir ajuste automático de líneas --
  bgcolor   %Color;    #IMPLIED -- color de fondo de la celda --
  width     %Length;   #IMPLIED -- anchura de la celda --
  height    %Length;   #IMPLIED -- altura de la celda --
>

<!--===== Marcos de Documento =====>

<!--
El modelo de contenido de los documentos HTML depende de si el HEAD
va seguido de un elemento FRAMESET o BODY. La omisión generalizada de
la etiqueta inicial de BODY hace impracticable definir el modelo
de contenido sin el uso de una sección marcada.
-->

<![ %HTML.Frameset; [
<!ELEMENT FRAMESET - - ((FRAMESET|FRAME)+ & NOFRAMES?) -- subdivisión en ventanas -->
<!ATTLIST FRAMESET
  %coreattrs;          -- id, class, style, title --
  rows      %MultiLengths; #IMPLIED -- lista de longitudes,
                                     por defecto: 100% (1 fila) --
  cols      %MultiLengths; #IMPLIED -- lista de longitudes,
                                     por defecto: 100% (1 columna) --
  onload    %Script;    #IMPLIED -- todos los marcos han sido cargados --
  onunload  %Script;    #IMPLIED -- todos los marcos han sido quitados --
>
]]>

<![ %HTML.Frameset; [
<!-- los nombres reservados de marcos comienzan con "_",
      los demás comienzan con una letra -->
<!ELEMENT FRAME - O EMPTY -- subventana -->
<!ATTLIST FRAME
  %coreattrs;          -- id, class, style, title --
  longdesc   %URI;     #IMPLIED -- vínculo a descripción larga
                                     (complementa al título) --
  name       CDATA     #IMPLIED -- nombre destino del marco --
  src        %URI;     #IMPLIED -- origen del contenido del marco --

```

```

frameborder (1|0)          1          -- ¿poner bordes al marco? --
marginwidth %Pixels;      #IMPLIED  -- anchuras de los márgenes en píxeles --
marginheight %Pixels;    #IMPLIED  -- alturas de los márgenes en píxeles --
noresize      (noresize)    #IMPLIED  -- ¿permitir a los usuarios redimensionar
                                     los marcos? --
scrolling    (yes|no|auto)  auto      -- barra de desplazamiento o no --
>
]]>

<!ELEMENT IFRAME - - (%flow;)*          -- subventana en línea -->
<!ATTLIST IFRAME
  %coreattrs;                -- id, class, style, title --
  longdesc      %URI;          #IMPLIED  -- vínculo a descripción larga
                                     (complementa al título) --
  name          CDATA          #IMPLIED  -- nombre destino del marco --
  src           %URI;          #IMPLIED  -- origen del contenido del marco --
  frameborder (1|0)          1          -- ¿poner bordes al marco? --
  marginwidth %Pixels;      #IMPLIED  -- anchuras de los márgenes en píxeles --
  marginheight %Pixels;    #IMPLIED  -- alturas de los márgenes en píxeles --
  scrolling    (yes|no|auto)  auto      -- barra de desplazamiento o no --
  align        %IAalign;      #IMPLIED  -- alineación vertical u horizontal --
  height       %Length;     #IMPLIED  -- altura del marco --
  width        %Length;     #IMPLIED  -- anchura del marco --
>

<![ %HTML.Frameset; [
<!ENTITY %noframes.content "(BODY) -(NOFRAMES)">
]]>

<!ENTITY %noframes.content "%flow;*">

<!ELEMENT NOFRAMES - - %noframes.content;
-- contenedor de contenido alternativo para representación no basada en marcos -->
<!ATTLIST NOFRAMES
  %attrs;                -- %coreattrs, %il8n, %events --
>

<!--===== Cabecera del Documento =====>
<!-- %head.misc; definido previamente como "SCRIPT|STYLE|META|LINK|OBJECT" -->
<!ENTITY %head.content "TITLE & ISINDEX? & BASE?">

<!ELEMENT HEAD O O (%head.content;) +(%head.misc;) -- cabecera del documento -->
<!ATTLIST HEAD
  %il8n;                -- lang, dir --
  profile      %URI;          #IMPLIED  -- diccionario de metainformación con nombre
--
>

<!-- El elemento TITLE no se considera parte del flujo de texto.
Debería ser mostrado, por ejemplo como el encabezado de la página
o como el título de la ventana. Se requiere exactamente un título
por documento.
-->
<!ELEMENT TITLE - - (#PCDATA) -(%head.misc;) -- título del documento -->
<!ATTLIST TITLE %il8n;>

<!ELEMENT ISINDEX - O EMPTY          -- entrada de texto de una línea

```

```

con indicador -->
<!ATTLIST ISINDEX
  %coreattrs;          -- id, class, style, title --
  %il8n;              -- lang, dir --
  prompt              %Text;          #IMPLIED -- mensaje indicador -->

<!ELEMENT BASE - O EMPTY -- URI base del documento -->
<!ATTLIST BASE
  href                %URI;          #IMPLIED -- URI que actúa como URI base --
  target              %FrameTarget; #IMPLIED -- representar en este marco --
  >

<!ELEMENT META - O EMPTY -- metainformación genérica -->
<!ATTLIST META
  %il8n;              -- lang, dir, para usar con content --
  http-equiv          NAME           #IMPLIED -- nombre de encabezado de respuesta HTTP --
  name                NAME           #IMPLIED -- nombre de la metainformación --
  content             CDATA          #REQUIRED -- información asociada --
  scheme              CDATA          #IMPLIED -- seleccionar forma de contenido --
  >

<!ELEMENT STYLE - - %StyleSheet -- información de estilo -->
<!ATTLIST STYLE
  %il8n;              -- lang, dir, para usar con title --
  type                %ContentType; #REQUIRED -- tipo de contenido del lenguaje de estilo --
  media               %MediaDesc;    #IMPLIED -- diseñado para usar con estos medios --
  title               %Text;         #IMPLIED -- título consultivo --
  >

<!ELEMENT SCRIPT - - %Script; -- sentencias de script -->
<!ATTLIST SCRIPT
  charset             %Charset;      #IMPLIED -- codificación de caracteres
                                     del recurso vinculado --
  type                %ContentType; #REQUIRED -- tipo de contenido del lenguaje de scripts
--
  language            CDATA          #IMPLIED -- nombre predefinido del lenguaje de
scripts--
  src                 %URI;          #IMPLIED -- URI de un script externo --
  defer               (defer)       #IMPLIED -- Los AAUU pueden aplazar
                                     la ejecución del script --
  event               CDATA          #IMPLIED -- reservado para posibles usos futuros --
  for                 %URI;          #IMPLIED -- reservado para posibles usos futuros --
  >

<!ELEMENT NOSCRIPT - - (%flow;)*
  -- contenedor de contenido alternativo para representación no basada en scripts -->
<!ATTLIST NOSCRIPT
  %attrs;              -- %coreattrs, %il8n, %events --
  >

<!--===== Estructura del Documento =====>
<!ENTITY % version "version CDATA #FIXED '%HTML.Version;'">

<![ %HTML.FrameSet; [
<!ENTITY % html.content "HEAD, FRAMESET">
]]>

```

```
<!ENTITY % html.content "HEAD, BODY">
```

```
<!ELEMENT HTML O O (%html.content;) -- elemento raíz del documento -->
```

```
<!ATTLIST HTML
```

```
  %i18n;
```

```
  -- lang, dir --
```

```
  %version;
```

```
>
```

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

23 Definición del Tipo de Documento con Marcos

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

```
<!--
Este el DTD para Documentos con Marcos de HTML 4.01,
que debería emplearse para documentos con marcos. Este
DTD es idéntico al DTD Transicional de HTML 4.01, excepto
por el modelo de contenido del elemento "HTML": en
documentos con marcos, el elemento "FRAMESET" reemplaza
al elemento "BODY".

        Borrador: $Date: 1999/12/24 23:20:08 $

        Autores:
            Dave Raggett <dsr@w3.org>
            Arnaud Le Hors <lehors@w3.org>
            Ian Jacobs <ij@w3.org>

Puede encontrar más información sobre HTML 4.01 en:

http://www.w3.org/TR/1999/REC-html401-19991224.

-->
<!ENTITY % HTML.Version "-//W3C//DTD HTML 4.01 Frameset//EN"
-- Uso normal:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
        "http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
...
</head>
<frameset>
...
</frameset>
</html>
-->
```



```
<!ENTITY % HTML.Frameset "INCLUDE">  
<!ENTITY % HTML4.dtd PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
%HTML4.dtd;
```

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

24 Referencias a entidades de caracteres en HTML 4

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Introducción a las referencias a entidades de caracteres](#)
2. [Referencias a entidades de caracteres para caracteres de la ISO 8859-1](#)
 1. [Lista de caracteres](#)
3. [Referencias a entidades de caracteres para símbolos, símbolos matemáticos y letras griegas](#)
 1. [Lista de caracteres](#)
4. [Referencias a entidades de caracteres para caracteres con significado en el código y caracteres de internacionalización](#)
 1. [Lista de caracteres](#)

24.1 Introducción a las referencias a entidades de caracteres

Una [referencia a una entidad de caracteres](#) es una estructura SGML que hace referencia a un carácter del [conjunto de caracteres del documento](#).

Esta versión de HTML soporta varios conjuntos de referencias a entidades de caracteres:

- [caracteres de la ISO 8859-1 \(Latin-1\)](#). De acuerdo con la sección 14 de [RFC1866](#), se ha extendido a esta especificación el conjunto de entidades Latin-1 para cubrir toda la parte derecha de la ISO-8859-1 (todas las posiciones de código con el bit más significativo establecido), incluyendo los ya habitualmente utilizados , © y ®. Los nombres de las entidades se han sacado de los apéndices de SGML (definido en [ISO8879](#)).
- [símbolos, símbolos matemáticos y letras griegas](#). Estos caracteres pueden representarse mediante signos de la fuente de Adobe "Symbol".
- [caracteres con significado en el código y caracteres de internacionalización](#) (p.ej., para texto bidireccional).

Las secciones que siguen presentan las listas completas de referencias a entidades de caracteres. Aunque por convención ([ISO10646](#)) los comentarios que siguen a cada entrada se suelen escribir con letras mayúsculas, en esta especificación los hemos convertido a minúsculas por razones de legibilidad.

24.2 Referencias a entidades de caracteres para caracteres de la ISO 8859-1

Las referencias a entidades de caracteres de esta sección producen caracteres cuyos equivalentes numéricos ya deberían estar soportados por los agentes de usuario conformes con HTML 2.0. Así, la referencia a entidad de caracteres ÷ es una forma de obtener el signo de división (÷) más conveniente que ÷.

Para soportar estas entidades con nombre, los agentes de usuario no tienen más que reconocer los nombres de las entidades y convertirlos a caracteres que caigan dentro del repertorio de la [ISO88591](#).

El carácter 65533 (hexadecimal FFFD) es el último carácter válido de UCS-2. El 65534 (hexadecimal FFFE) está sin asignar y se reserva como la versión con los bytes traspuestos del ESPACIO DE NO SEPARACIÓN DE ANCHURA ZERO con el propósito de detectar el orden de los bytes. El 65535 (hexadecimal FFFF) está sin asignar.

24.2.1 Lista de caracteres

```

<!-- © International Organization for Standardization 1986
      Se concede permiso para copiar en cualquier formato para su
      uso conforme con los sistemas y aplicaciones SGML tal y como se
      definen en ISO 8879, siempre y cuando se incluya este aviso en
      todas las copias.
-->
<!-- Conjunto de entidades de caracteres. Invocación típica:
      <!ENTITY % HTMLlat1 PUBLIC
           "-//W3C//ENTITIES Latin 1//EN//HTML">
      %HTMLlat1;
-->

<!ENTITY nbsp      CDATA "&#160;" -- espacio de no separación, U+00A0 ISOnum -->
<!ENTITY iexcl    CDATA "&#161;" -- signo de exclamación invertido, U+00A1 ISOnum -->
<!ENTITY cent     CDATA "&#162;" -- signo de centavo, U+00A2 ISOnum -->
<!ENTITY pound    CDATA "&#163;" -- signo de libra, U+00A3 ISOnum -->
<!ENTITY curren   CDATA "&#164;" -- signo de moneda, U+00A4 ISOnum -->
<!ENTITY yen      CDATA "&#165;" -- signo de yen, U+00A5 ISOnum -->
<!ENTITY brvbar   CDATA "&#166;" -- barra partida = barra vertical partida,
                               U+00A6 ISOnum -->
<!ENTITY sect     CDATA "&#167;" -- signo de sección, U+00A7 ISOnum -->
<!ENTITY uml      CDATA "&#168;" -- diéresis, U+00A8 ISODia -->
<!ENTITY copy     CDATA "&#169;" -- signo de copyright, U+00A9 ISOnum -->
<!ENTITY ordf     CDATA "&#170;" -- indicador ordinal femenino, U+00AA ISOnum -->
<!ENTITY laquo    CDATA "&#171;" -- comillas dobles angulares abiertas,
                               U+00AB ISOnum -->
<!ENTITY not      CDATA "&#172;" -- signo de negación, U+00AC ISOnum -->
<!ENTITY shy      CDATA "&#173;" -- guión blando = guión discrecional,
                               U+00AD ISOnum -->
<!ENTITY reg      CDATA "&#174;" -- signo de registrado = signo de marca comercial
                               registrada, U+00AE ISOnum -->
<!ENTITY macr     CDATA "&#175;" -- macron (marca larga) = superguión = guión alto,
                               U+00AF ISODia -->
<!ENTITY deg      CDATA "&#176;" -- signo de grados, U+00B0 ISOnum -->
<!ENTITY plusmn   CDATA "&#177;" -- signo más-menos = signo más-o-menos,
                               U+00B1 ISOnum -->
<!ENTITY sup2     CDATA "&#178;" -- dos superíndice = al cuadrado,
                               U+00B2 ISOnum -->
<!ENTITY sup3     CDATA "&#179;" -- tres superíndice = al cubo,
                               U+00B3 ISOnum -->
<!ENTITY acute    CDATA "&#180;" -- acento agudo, U+00B4 ISODia -->
<!ENTITY micro    CDATA "&#181;" -- signo de micras, U+00B5 ISOnum -->
<!ENTITY para     CDATA "&#182;" -- signo de párrafo, U+00B6 ISOnum -->
<!ENTITY middot   CDATA "&#183;" -- punto medio = coma georgiana, U+00B7 ISOnum -->
<!ENTITY cedil    CDATA "&#184;" -- cedilla, U+00B8 ISODia -->
<!ENTITY sup1     CDATA "&#185;" -- uno superíndice, U+00B9 ISOnum -->
<!ENTITY ordm     CDATA "&#186;" -- indicador ordinal masculino,
                               U+00BA ISOnum -->
<!ENTITY raquo    CDATA "&#187;" -- comillas dobles angulares cerradas,
                               U+00BB ISOnum -->
<!ENTITY frac14   CDATA "&#188;" -- fracción común un cuarto
                               = fracción de un cuarto, U+00BC ISOnum -->
<!ENTITY frac12   CDATA "&#189;" -- fracción común un medio
                               = fracción de un medio, U+00BD ISOnum -->
<!ENTITY frac34   CDATA "&#190;" -- fracción común tres cuartos
                               = fracción de tres cuartos, U+00BE ISOnum -->
<!ENTITY iquest   CDATA "&#191;" -- signo de interrogación invertido,

```

```

U+00BF ISONum -->
<!ENTITY Agrave CDATA "&#192;" -- letra mayúscula latina A con acento grave,
U+00C0 ISOLat1 -->
<!ENTITY Aacute CDATA "&#193;" -- letra mayúscula latina A con acento agudo,
U+00C1 ISOLat1 -->
<!ENTITY Acirc CDATA "&#194;" -- letra mayúscula latina A con acento circunflejo,
U+00C2 ISOLat1 -->
<!ENTITY Atilde CDATA "&#195;" -- letra mayúscula latina A con tilde,
U+00C3 ISOLat1 -->
<!ENTITY Auml CDATA "&#196;" -- letra mayúscula latina A con diéresis,
U+00C4 ISOLat1 -->
<!ENTITY Aring CDATA "&#197;" -- letra mayúscula latina A con anillo,
U+00C5 ISOLat1 -->
<!ENTITY AElig CDATA "&#198;" -- letra mayúscula latina AE,
= ligadura mayúscula latina AE,
U+00C6 ISOLat1 -->
<!ENTITY Ccedil CDATA "&#199;" -- letra mayúscula latina C con cedilla,
U+00C7 ISOLat1 -->
<!ENTITY Egrave CDATA "&#200;" -- letra mayúscula latina E con acento grave,
U+00C8 ISOLat1 -->
<!ENTITY Eacute CDATA "&#201;" -- letra mayúscula latina E con acento agudo,
U+00C9 ISOLat1 -->
<!ENTITY Ecirc CDATA "&#202;" -- letra mayúscula latina E con acento circunflejo,
U+00CA ISOLat1 -->
<!ENTITY Euml CDATA "&#203;" -- letra mayúscula latina E con diéresis,
U+00CB ISOLat1 -->
<!ENTITY Igrave CDATA "&#204;" -- letra mayúscula latina I con acento grave,
U+00CC ISOLat1 -->
<!ENTITY Iacute CDATA "&#205;" -- letra mayúscula latina I con acento agudo,
U+00CD ISOLat1 -->
<!ENTITY Icirc CDATA "&#206;" -- letra mayúscula latina I con acento circunflejo,
U+00CE ISOLat1 -->
<!ENTITY Iuml CDATA "&#207;" -- letra mayúscula latina I con diéresis,
U+00CF ISOLat1 -->
<!ENTITY ETH CDATA "&#208;" -- letra mayúscula latina ETH, U+00D0 ISOLat1 -->
<!ENTITY Ntilde CDATA "&#209;" -- letra mayúscula latina N con tilde
= eñe mayúscula, U+00D1 ISOLat1 -->
<!ENTITY Ograve CDATA "&#210;" -- letra mayúscula latina O con acento grave,
U+00D2 ISOLat1 -->
<!ENTITY Oacute CDATA "&#211;" -- letra mayúscula latina O con acento agudo,
U+00D3 ISOLat1 -->
<!ENTITY Ocirc CDATA "&#212;" -- letra mayúscula latina O con acento circunflejo,
U+00D4 ISOLat1 -->
<!ENTITY Otilde CDATA "&#213;" -- letra mayúscula latina O con tilde,
U+00D5 ISOLat1 -->
<!ENTITY Ouml CDATA "&#214;" -- letra mayúscula latina O con diéresis,
U+00D6 ISOLat1 -->
<!ENTITY times CDATA "&#215;" -- signo de multiplicación, U+00D7 ISONum -->
<!ENTITY Oslash CDATA "&#216;" -- letra mayúscula latina O tachada,
letra mayúscula latina O con barra inclinada,
U+00D8 ISOLat1 -->
<!ENTITY Ugrave CDATA "&#217;" -- letra mayúscula latina U con acento grave,
U+00D9 ISOLat1 -->
<!ENTITY Uacute CDATA "&#218;" -- letra mayúscula latina U con acento agudo,
U+00DA ISOLat1 -->
<!ENTITY Ucirc CDATA "&#219;" -- letra mayúscula latina U con acento circunflejo,
U+00DB ISOLat1 -->
<!ENTITY Uuml CDATA "&#220;" -- letra mayúscula latina U con diéresis,
U+00DC ISOLat1 -->
<!ENTITY Yacute CDATA "&#221;" -- letra mayúscula latina Y con acento agudo,

```

```

U+00DD ISolat1 -->
<!ENTITY THORN CDATA "&#222;" -- letra mayúscula latina THORN,
U+00DE ISolat1 -->
<!ENTITY szlig CDATA "&#223;" -- letra minúscula latina s muda = ess-zed,
U+00DF ISolat1 -->
<!ENTITY agrave CDATA "&#224;" -- letra minúscula latina a con acento grave,
U+00E0 ISolat1 -->
<!ENTITY aacute CDATA "&#225;" -- letra minúscula latina a con acento agudo,
U+00E1 ISolat1 -->
<!ENTITY acirc CDATA "&#226;" -- letra minúscula latina a con acento circunflejo,
U+00E2 ISolat1 -->
<!ENTITY atilde CDATA "&#227;" -- letra minúscula latina a con tilde,
U+00E3 ISolat1 -->
<!ENTITY auml CDATA "&#228;" -- letra minúscula latina a con diéresis,
U+00E4 ISolat1 -->
<!ENTITY aring CDATA "&#229;" -- letra minúscula latina a con anillo,
U+00E5 ISolat1 -->
<!ENTITY aelig CDATA "&#230;" -- letra minúscula latina ae,
= ligadura minúscula latina ae, U+00E6 ISolat1 -->
<!ENTITY ccedil CDATA "&#231;" -- letra minúscula latina c con cedilla,
U+00E7 ISolat1 -->
<!ENTITY egrave CDATA "&#232;" -- letra minúscula latina e con acento grave,
U+00E8 ISolat1 -->
<!ENTITY eacute CDATA "&#233;" -- letra minúscula latina e con acento agudo,
U+00E9 ISolat1 -->
<!ENTITY ecirc CDATA "&#234;" -- letra minúscula latina e con acento circunflejo,
U+00EA ISolat1 -->
<!ENTITY euml CDATA "&#235;" -- letra minúscula latina e con diéresis,
U+00EB ISolat1 -->
<!ENTITY igrave CDATA "&#236;" -- letra minúscula latina i con acento grave,
U+00EC ISolat1 -->
<!ENTITY iacute CDATA "&#237;" -- letra minúscula latina i con acento agudo,
U+00ED ISolat1 -->
<!ENTITY icirc CDATA "&#238;" -- letra minúscula latina i con acento circunflejo,
U+00EE ISolat1 -->
<!ENTITY iuml CDATA "&#239;" -- letra minúscula latina i con diéresis,
U+00EF ISolat1 -->
<!ENTITY eth CDATA "&#240;" -- letra minúscula latina eth, U+00F0 ISolat1 -->
<!ENTITY ntilde CDATA "&#241;" -- letra minúscula latina n con tilde
= eñe minúscula, U+00F1 ISolat1 -->
<!ENTITY ograve CDATA "&#242;" -- letra minúscula latina o con acento grave,
U+00F2 ISolat1 -->
<!ENTITY oacute CDATA "&#243;" -- letra minúscula latina o con acento agudo,
U+00F3 ISolat1 -->
<!ENTITY ocirc CDATA "&#244;" -- letra minúscula latina o con acento circunflejo,
U+00F4 ISolat1 -->
<!ENTITY otilde CDATA "&#245;" -- letra minúscula latina o con tilde,
U+00F5 ISolat1 -->
<!ENTITY ouml CDATA "&#246;" -- letra minúscula latina o con diéresis,
U+00F6 ISolat1 -->
<!ENTITY divide CDATA "&#247;" -- signo de división, U+00F7 ISOnum -->
<!ENTITY oslash CDATA "&#248;" -- letra minúscula latina o tachada,
= letra minúscula latina o con barra inclinada,
U+00F8 ISolat1 -->
<!ENTITY ugrave CDATA "&#249;" -- letra minúscula latina u con acento grave,
U+00F9 ISolat1 -->
<!ENTITY uacute CDATA "&#250;" -- letra minúscula latina u con acento agudo,
U+00FA ISolat1 -->
<!ENTITY ucirc CDATA "&#251;" -- letra minúscula latina u con acento circunflejo,
U+00FB ISolat1 -->

```

```

<!ENTITY uuml    CDATA "&#252;" -- letra minúscula latina u con diéresis,
                                U+00FC ISolat1 -->
<!ENTITY yacute  CDATA "&#253;" -- letra minúscula latina y con acento agudo,
                                U+00FD ISolat1 -->
<!ENTITY thorn   CDATA "&#254;" -- letra minúscula latina thorn,
                                U+00FE ISolat1 -->
<!ENTITY yuml    CDATA "&#255;" -- letra minúscula latina y con diéresis,
                                U+00FF ISolat1 -->

```

24.3 Referencias a entidades de caracteres para símbolos, símbolos matemáticos y letras griegas

Las referencias a entidades de caracteres de esta sección producen caracteres que pueden ser representados por los signos de la fuente de Adobe Symbol, ampliamente extendida, que incluye caracteres griegos, varios símbolos de agrupamiento, y una selección de operadores matemáticos como símbolos de gradiente, productorio y sumatorio.

Para soportar estas entidades, los agentes de usuario pueden soportar la totalidad de la [\[ISO10646\]](#) o usar otros medios. La representación de signos para estos caracteres puede lograrse o bien mostrando los caracteres relevantes de la [\[ISO10646\]](#) o bien por otros medios, como por ejemplo transformando internamente las entidades enumeradas, las referencias numéricas de caracteres, y los caracteres, en las posiciones adecuadas de alguna fuente que contenga los signos necesarios.

***Cuándo usar entidades griegas.** Este conjunto de entidades contiene todas las letras usadas en el griego moderno. Sin embargo, no contiene la puntuación griega, caracteres acentuados precompuestos ni los acentos sin espaciado (tonos, dialytika) necesarios para componerlos. No hay letras arcaicas, letras exclusivas del copto, ni letras precompuestas para el griego politónico. Las entidades aquí definidas no están pensadas para la representación de textos en griego moderno y como tales no serían una buena representación; están pensadas, más bien, para las letras griegas ocasionales que aparecen en obras técnicas y matemáticas.*

24.3.1 Lista de caracteres

```

<!-- Caracteres matemáticos, griegos y simbólicos para HTML -->

<!-- Conjunto de entidades de caracteres. Invocación típica:
<!ENTITY % HTMLsymbol PUBLIC
    "-//W3C//ENTITIES Symbols//EN//HTML">
%HTMLsymbol; -->

<!-- © International Organization for Standardization 1986:
Se concede permiso para copiar en cualquier formato para su
uso conforme con los sistemas y aplicaciones SGML tal y como se
definen en ISO 8879, siempre y cuando se incluya este aviso en
todas las copias.
-->

<!-- Se da el nombre del conjunto de entidades ISO relacionado a
menos que se trate de un carácter nuevo. Los nombres nuevos (es
decir, que no están en la lista ISO 8879) no están en conflicto
con ningún nombre de entidad existente en ISO 8879. Se da para
cada carácter el número de carácter ISO 10646, en hexadecimal.
Los valores CDATA son conversiones decimales de los valores de
la ISO 10646 y se refieren al conjunto de caracteres del
documento. Los nombres son nombres de la ISO 10646.
-->

<!-- Latin Extended-B -->
<!ENTITY fnof    CDATA "&#402;" -- f minúscula latina con gancho = función
                                = florín, U+0192 ISOtech -->

```

```

<!-- Griego -->
<!ENTITY Alpha    CDATA "&#913;" -- letra griega mayúscula alpha, U+0391 -->
<!ENTITY Beta     CDATA "&#914;" -- letra griega mayúscula beta, U+0392 -->
<!ENTITY Gamma    CDATA "&#915;" -- letra griega mayúscula gamma,
                    U+0393 ISOgrk3 -->
<!ENTITY Delta    CDATA "&#916;" -- letra griega mayúscula delta,
                    U+0394 ISOgrk3 -->
<!ENTITY Epsilon  CDATA "&#917;" -- letra griega mayúscula epsilon, U+0395 -->
<!ENTITY Zeta     CDATA "&#918;" -- letra griega mayúscula zeta, U+0396 -->
<!ENTITY Eta      CDATA "&#919;" -- letra griega mayúscula eta, U+0397 -->
<!ENTITY Theta    CDATA "&#920;" -- letra griega mayúscula theta,
                    U+0398 ISOgrk3 -->
<!ENTITY Iota     CDATA "&#921;" -- letra griega mayúscula iota, U+0399 -->
<!ENTITY Kappa    CDATA "&#922;" -- letra griega mayúscula kappa, U+039A -->
<!ENTITY Lambda   CDATA "&#923;" -- letra griega mayúscula lambda,
                    U+039B ISOgrk3 -->
<!ENTITY Mu       CDATA "&#924;" -- letra griega mayúscula mu, U+039C -->
<!ENTITY Nu       CDATA "&#925;" -- letra griega mayúscula nu, U+039D -->
<!ENTITY Xi       CDATA "&#926;" -- letra griega mayúscula xi, U+039E ISOgrk3 -->
<!ENTITY Omicron  CDATA "&#927;" -- letra griega mayúscula omicron, U+039F -->
<!ENTITY Pi       CDATA "&#928;" -- letra griega mayúscula pi, U+03A0 ISOgrk3 -->
<!ENTITY Rho      CDATA "&#929;" -- letra griega mayúscula rho, U+03A1 -->
<!-- no hay Sigmaf, ni tampoco carácter U+03A2 -->
<!ENTITY Sigma    CDATA "&#931;" -- letra griega mayúscula sigma,
                    U+03A3 ISOgrk3 -->
<!ENTITY Tau      CDATA "&#932;" -- letra griega mayúscula tau, U+03A4 -->
<!ENTITY Upsilon  CDATA "&#933;" -- letra griega mayúscula upsilon,
                    U+03A5 ISOgrk3 -->
<!ENTITY Phi      CDATA "&#934;" -- letra griega mayúscula phi,
                    U+03A6 ISOgrk3 -->
<!ENTITY Chi      CDATA "&#935;" -- letra griega mayúscula chi, U+03A7 -->
<!ENTITY Psi      CDATA "&#936;" -- letra griega mayúscula psi,
                    U+03A8 ISOgrk3 -->
<!ENTITY Omega    CDATA "&#937;" -- letra griega mayúscula omega,
                    U+03A9 ISOgrk3 -->
<!ENTITY alpha    CDATA "&#945;" -- letra griega minúscula alpha,
                    U+03B1 ISOgrk3 -->
<!ENTITY beta     CDATA "&#946;" -- letra griega minúscula beta, U+03B2 ISOgrk3 -->
<!ENTITY gamma    CDATA "&#947;" -- letra griega minúscula gamma,
                    U+03B3 ISOgrk3 -->
<!ENTITY delta    CDATA "&#948;" -- letra griega minúscula delta,
                    U+03B4 ISOgrk3 -->
<!ENTITY epsilon  CDATA "&#949;" -- letra griega minúscula epsilon,
                    U+03B5 ISOgrk3 -->
<!ENTITY zeta     CDATA "&#950;" -- letra griega minúscula zeta, U+03B6 ISOgrk3 -->
<!ENTITY eta      CDATA "&#951;" -- letra griega minúscula eta, U+03B7 ISOgrk3 -->
<!ENTITY theta    CDATA "&#952;" -- letra griega minúscula theta,
                    U+03B8 ISOgrk3 -->
<!ENTITY iota     CDATA "&#953;" -- letra griega minúscula iota, U+03B9 ISOgrk3 -->
<!ENTITY kappa    CDATA "&#954;" -- letra griega minúscula kappa,
                    U+03BA ISOgrk3 -->
<!ENTITY lambda   CDATA "&#955;" -- letra griega minúscula lambda,
                    U+03BB ISOgrk3 -->
<!ENTITY mu       CDATA "&#956;" -- letra griega minúscula mu, U+03BC ISOgrk3 -->
<!ENTITY nu       CDATA "&#957;" -- letra griega minúscula nu, U+03BD ISOgrk3 -->
<!ENTITY xi       CDATA "&#958;" -- letra griega minúscula xi, U+03BE ISOgrk3 -->
<!ENTITY omicron  CDATA "&#959;" -- letra griega minúscula omicron, U+03BF NEW -->
<!ENTITY pi       CDATA "&#960;" -- letra griega minúscula pi, U+03C0 ISOgrk3 -->
<!ENTITY rho      CDATA "&#961;" -- letra griega minúscula rho, U+03C1 ISOgrk3 -->

```

```

<!ENTITY sigmaf CDATA "&#962;" -- letra griega minúscula final sigma,
    U+03C2 ISOgrk3 -->
<!ENTITY sigma CDATA "&#963;" -- letra griega minúscula sigma,
    U+03C3 ISOgrk3 -->
<!ENTITY tau CDATA "&#964;" -- letra griega minúscula tau, U+03C4 ISOgrk3 -->
<!ENTITY upsilon CDATA "&#965;" -- letra griega minúscula upsilon,
    U+03C5 ISOgrk3 -->
<!ENTITY phi CDATA "&#966;" -- letra griega minúscula phi, U+03C6 ISOgrk3 -->
<!ENTITY chi CDATA "&#967;" -- letra griega minúscula chi, U+03C7 ISOgrk3 -->
<!ENTITY psi CDATA "&#968;" -- letra griega minúscula psi, U+03C8 ISOgrk3 -->
<!ENTITY omega CDATA "&#969;" -- letra griega minúscula omega,
    U+03C9 ISOgrk3 -->
<!ENTITY thetasym CDATA "&#977;" -- símbolo letra griega minúscula theta,
    U+03D1 NEW -->
<!ENTITY upsih CDATA "&#978;" -- símbolo letra griega mayúscula upsilon con
    gancho, U+03D2 NEW -->
<!ENTITY piv CDATA "&#982;" -- símbolos griego pi, U+03D6 ISOgrk3 -->

<!-- Puntuación general -->
<!ENTITY bull CDATA "&#8226;" -- marcador (bullet) = círculo negro pequeño,
    U+2022 ISOpub -->
<!-- "bullet" NO es lo mismo que el operador "bullet", U+2219 -->
<!ENTITY hellip CDATA "&#8230;" -- elipsis horizontal = puntos suspensivos,
    U+2026 ISOpub -->
<!ENTITY prime CDATA "&#8242;" -- prima = minutos = pies, U+2032 ISOftech -->
<!ENTITY Prime CDATA "&#8243;" -- doble prima = segundos = pulgadas,
    U+2033 ISOftech -->
<!ENTITY oline CDATA "&#8254;" -- línea superior,
    U+203E NEW -->
<!ENTITY frasl CDATA "&#8260;" -- línea de fracción, U+2044 NEW -->

<!-- Símbolos tipo letras -->
<!ENTITY weierp CDATA "&#8472;" -- P mayúscula cursiva = conjunto potencia
    = p de Weierstrass, U+2118 ISOamso -->
<!ENTITY image CDATA "&#8465;" -- I negra mayúscula = parte imaginaria,
    U+2111 ISOamso -->
<!ENTITY real CDATA "&#8476;" -- R negra mayúscula = símbolo de parte real,
    U+211C ISOamso -->
<!ENTITY trade CDATA "&#8482;" -- signo de marca comercial, U+2122 ISOnum -->
<!ENTITY alefsym CDATA "&#8501;" -- símbolo alef = primer cardinal transfinito,
    U+2135 NEW -->
<!-- el símbolo alef NO es el mismo que la letra hebrea alef, U+05D0
    aunque puede usarse el mismo signo para representar a ambos caracteres -->

<!-- Flechas -->
<!ENTITY larr CDATA "&#8592;" -- flecha hacia la izquierda, U+2190 ISOnum -->
<!ENTITY uarr CDATA "&#8593;" -- flecha hacia arriba, U+2191 ISOnum-->
<!ENTITY rarr CDATA "&#8594;" -- flecha hacia la derecha, U+2192 ISOnum -->
<!ENTITY darr CDATA "&#8595;" -- flecha hacia abajo, U+2193 ISOnum -->
<!ENTITY harr CDATA "&#8596;" -- flecha izquierda derecha, U+2194 ISOamsa -->
<!ENTITY crarr CDATA "&#8629;" -- flecha doblada hacia abajo
    = retorno de carro, U+21B5 NEW -->
<!ENTITY lArr CDATA "&#8656;" -- flecha doble hacia la izquierda,
    U+21D0 ISOftech -->
<!-- ISO 10646 no dice que lArr sea lo mismo que la flecha 'implicado por' pero
    tampoco tiene ningún otro carácter para esa función. Por tanto ? puede usarse
    lArr para 'implicado por' tal y como sugiere ISOftech -->
<!ENTITY uArr CDATA "&#8657;" -- flecha doble hacia arriba, U+21D1 ISOamsa -->
<!ENTITY rArr CDATA "&#8658;" -- flecha doble hacia la derecha,
    U+21D2 ISOftech -->

```



```

<!-- ISO 10646 no dice que este sea el carácter 'implica' pero no tiene ningún
      otro caracter para esta función, por tanto ?
      puede usarse rArr para 'implica' tal y como sugiere ISOTech -->
<!ENTITY dArr      CDATA "&#8659;" -- flecha doble hacia abajo, U+21D3 ISOamsa -->
<!ENTITY hArr      CDATA "&#8660;" -- flecha doble izquierda derecha,
                        U+21D4 ISOamsa -->

<!-- Operadores Matemáticos -->
<!ENTITY forall    CDATA "&#8704;" -- para todo, U+2200 ISOTech -->
<!ENTITY part      CDATA "&#8706;" -- diferencial parcial, U+2202 ISOTech -->
<!ENTITY exist     CDATA "&#8707;" -- existe, U+2203 ISOTech -->
<!ENTITY empty     CDATA "&#8709;" -- conjunto vacío = diámetro,
                        U+2205 ISOamsa -->
<!ENTITY nabla     CDATA "&#8711;" -- nabla, U+2207 ISOTech -->
<!ENTITY isin      CDATA "&#8712;" -- pertenece a, U+2208 ISOTech -->
<!ENTITY notin     CDATA "&#8713;" -- no pertenece a, U+2209 ISOTech -->
<!ENTITY ni        CDATA "&#8715;" -- contiene a, U+220B ISOTech -->
<!-- no debería haber un nombre más fácil de recordar que 'ni'? -->
<!ENTITY prod      CDATA "&#8719;" -- producto de n elementos = signo de productorio,
                        U+220F ISOamsb -->

<!-- prod NO es lo mismo que el carácter U+03A0 'letra griega mayúscula pi' aunque
      puede usarse el mismo signo para ambos -->
<!ENTITY sum       CDATA "&#8721;" -- sumatorio, U+2211 ISOamsb -->
<!-- sum NO es lo mismo que el carácter U+03A3 'letra griega mayúscula sigma'
      aunque puede usarse el mismo signo para ambos -->
<!ENTITY minus     CDATA "&#8722;" -- signo menos, U+2212 ISOTech -->
<!ENTITY lowast    CDATA "&#8727;" -- operador asterisco, U+2217 ISOTech -->
<!ENTITY radic     CDATA "&#8730;" -- raíz cuadrada = signo radical,
                        U+221A ISOTech -->
<!ENTITY prop      CDATA "&#8733;" -- proporcional a, U+221D ISOTech -->
<!ENTITY infin     CDATA "&#8734;" -- infinito, U+221E ISOTech -->
<!ENTITY ang       CDATA "&#8736;" -- ángulo, U+2220 ISOamsa -->
<!ENTITY and       CDATA "&#8743;" -- y lógico, U+2227 ISOTech -->
<!ENTITY or        CDATA "&#8744;" -- o lógico, U+2228 ISOTech -->
<!ENTITY cap       CDATA "&#8745;" -- intersección, U+2229 ISOTech -->
<!ENTITY cup       CDATA "&#8746;" -- unión, U+222A ISOTech -->
<!ENTITY int       CDATA "&#8747;" -- integral, U+222B ISOTech -->
<!ENTITY there4    CDATA "&#8756;" -- de aquí, U+2234 ISOTech -->
<!ENTITY sim       CDATA "&#8764;" -- operador tilde = varía con = similar a,
                        U+223C ISOTech -->

<!-- el operador tilde NO es el mismo carácter que la tilde, U+007E,
      aunque puede usarse el mismo signo para representar a ambos -->
<!ENTITY cong      CDATA "&#8773;" -- aproximadamente igual a, U+2245 ISOTech -->
<!ENTITY asymp     CDATA "&#8776;" -- casi igual que = asintótico a,
                        U+2248 ISOamsr -->
<!ENTITY ne        CDATA "&#8800;" -- distinto de, U+2260 ISOTech -->
<!ENTITY equiv     CDATA "&#8801;" -- idéntico a, U+2261 ISOTech -->
<!ENTITY le        CDATA "&#8804;" -- menor o igual que, U+2264 ISOTech -->
<!ENTITY ge        CDATA "&#8805;" -- mayor o igual que,
                        U+2265 ISOTech -->
<!ENTITY sub       CDATA "&#8834;" -- subconjunto de, U+2282 ISOTech -->
<!ENTITY sup       CDATA "&#8835;" -- superconjunto de, U+2283 ISOTech -->
<!-- obsérvese que nsup, 'no es superconjunto de, U+2283' no está cubierto por
      la codificación de la fuente Symbol y no se han incluido. ¿Debería estarlo,
      por simetría? Está es ISOamsn -->
<!ENTITY nsub      CDATA "&#8836;" -- no es subconjunto de, U+2284 ISOamsn -->
<!ENTITY sube      CDATA "&#8838;" -- subconjunto de o igual que, U+2286 ISOTech -->
<!ENTITY supe      CDATA "&#8839;" -- superconjunto de o igual que,
                        U+2287 ISOTech -->
<!ENTITY oplus     CDATA "&#8853;" -- círculo más = suma directa,

```

```

                                U+2295 ISOamsb -->
<!ENTITY otimes    CDATA "&#8855;" -- círculo por = producto vectorial,
                                U+2297 ISOamsb -->
<!ENTITY perp      CDATA "&#8869;" -- ortogonal a = perpendicular,
                                U+22A5 ISotech -->
<!ENTITY sdot      CDATA "&#8901;" -- operador punto, U+22C5 ISOamsb -->
<!-- el operador punto NO es el mismo carácter que U+00B7 punto medio -->

<!-- Otros Símbolos Técnicos -->
<!ENTITY lceil     CDATA "&#8968;" -- parte superior de corchete abierto,
                                U+2308 ISOamsc -->
<!ENTITY rceil     CDATA "&#8969;" -- parte superior de corchete cerrado,
                                U+2309 ISOamsc -->
<!ENTITY lfloor    CDATA "&#8970;" -- parte inferior de corchete abierto,
                                U+230A ISOamsc -->
<!ENTITY rfloor    CDATA "&#8971;" -- parte inferior de corchete cerrado,
                                U+230B ISOamsc -->
<!ENTITY lang      CDATA "&#9001;" -- corchete angular hacia la izquierda,
                                U+2329 ISotech -->
<!-- lang NO es el mismo carácter que U+003C 'menor que'
      ni U+2039 'comilla simple angular abierta' -->
<!ENTITY rang      CDATA "&#9002;" -- corchete angular hacia la derecha,
                                U+232A ISotech -->
<!-- rang NO es el mismo carácter que U+003E 'mayor que'
      ni U+203A 'comilla simple angular cerrada' -->

<!-- Formas geométricas -->
<!ENTITY loz       CDATA "&#9674;" -- rombo, U+25CA ISOpub -->

<!-- Otros símbolos -->
<!ENTITY spades    CDATA "&#9824;" -- palo negro de picas, U+2660 ISOpub -->
<!-- aquí al parecer negras quiere decir relleno -->
<!ENTITY clubs     CDATA "&#9827;" -- palo negro de tréboles = trébol,
                                U+2663 ISOpub -->
<!ENTITY hearts    CDATA "&#9829;" -- palo negro de corazones = corazón,
                                U+2665 ISOpub -->
<!ENTITY diams     CDATA "&#9830;" -- palo negro de diamantes, U+2666 ISOpub -->

```

24.4 Referencias a entidades de caracteres para caracteres con significado en el código y para caracteres de internacionalización

Las referencias a entidades de caracteres de esta sección son para convertir caracteres significativos para el código en secuencias de escape (estos caracteres son los mismos que en HTML 2.0 y 3.2), y para denotar espacios y guiones. Otros caracteres de esta sección se aplican a cuestiones de internacionalización como la eliminación de las ambigüedades del texto bidireccional (ver la sección sobre [texto bidireccional](#) para más detalles).

También se han añadido entidades para el resto de los caracteres que se dan en CP-1252 y que no aparecen en los conjuntos de entidades HTMLlat1 o HTMLsymbol. Todos estos aparecen en el rango 128-159 del juego de caracteres CP-1252. Estas entidades permiten que se denoten los caracteres independientemente de la plataforma.

Para soportar estas entidades, los agentes de usuario pueden o bien soportar la totalidad de [ISO10646](#) o usar otros medios. La representación de signos para estos caracteres puede lograrse o bien pudiendo mostrar los caracteres correspondientes de la [ISO10646](#) o bien por otros medios, como por ejemplo transformando internamente las entidades enumeradas, las referencias numéricas de caracteres y los caracteres, en las posiciones adecuadas de alguna fuente que contenga los signos necesarios.

24.4.1 Lista de caracteres

```

<!-- Caracteres especiales para HTML -->

<!-- Conjunto de entidades de caracteres. Invocación típica:
<!ENTITY % HTMLspecial PUBLIC
    "-//W3C//ENTITIES Special//EN//HTML">
%HTMLspecial; -->

<!-- © International Organization for Standardization 1986:
Se concede permiso para copiar en cualquier formato para su
uso conforme con los sistemas y aplicaciones SGML tal y como se
definen en ISO 8879, siempre y cuando se incluya este aviso en
todas las copias.
-->

<!-- Se da el nombre del conjunto de entidades ISO relacionado a
menos que se trate de un carácter nuevo. Los nombres nuevos (es
decir, que no están en la lista ISO 8879) no están en conflicto
con ningún nombre de entidad existente en ISO 8879. Se da para
cada carácter el número de carácter ISO 10646, en hexadecimal.
Los valores CDATA son conversiones decimales de los valores de
la ISO 10646 y se refieren al conjunto de caracteres del
documento. Los nombres son nombres de la ISO 10646.
-->

<!-- Controles de código y Latín Básico -->
<!ENTITY quot    CDATA "&#34;"    -- comillas, U+0022 ISOnum -->
<!ENTITY amp    CDATA "&#38;"    -- ampersand, U+0026 ISOnum -->
<!ENTITY lt     CDATA "&#60;"    -- signo menor, U+003C ISOnum -->
<!ENTITY gt     CDATA "&#62;"    -- signo mayor, U+003E ISOnum -->

<!-- Latin Extended-A -->
<!ENTITY OElig  CDATA "&#338;"    -- ligadura mayúscula latina OE, U+0152 ISolat2 -->
<!ENTITY oelig  CDATA "&#339;"    -- ligadura minúscula latina oe, U+0153 ISolat2 -->
<!-- ligadura es equívoco, éste es un carácter independiente en algunos idiomas -->
<!ENTITY Scaron CDATA "&#352;"    -- letra mayúscula latina S con anticircunflejo,
U+0160 ISolat2 -->
<!ENTITY scaron CDATA "&#353;"    -- letra minúscula latina S con anticircunflejo,
U+0161 ISolat2 -->
<!ENTITY Yuml   CDATA "&#376;"    -- letra mayúscula latina Y con diéresis,
U+0178 ISolat2 -->

<!-- Letras Modificadoras Espaciadoras -->
<!ENTITY circ   CDATA "&#710;"    -- acento circunflejo, U+02C6 ISOpub -->
<!ENTITY tilde   CDATA "&#732;"    -- tilde pequeña, U+02DC ISODia -->

<!-- Puntuación General -->
<!ENTITY ensp   CDATA "&#8194;"    -- espacio en, U+2002 ISOpub -->
<!ENTITY emsp   CDATA "&#8195;"    -- espacio em, U+2003 ISOpub -->
<!ENTITY thinsp CDATA "&#8201;"    -- espacio fino, U+2009 ISOpub -->
<!ENTITY zwnj   CDATA "&#8204;"    -- separador de anchura cero,
U+200C NEW RFC 2070 -->
<!ENTITY zwj    CDATA "&#8205;"    -- conector de anchura cero,
U+200D NEW RFC 2070 -->
<!ENTITY lrm    CDATA "&#8206;"    -- marca de izquierda a derecha,
U+200E NEW RFC 2070 -->
<!ENTITY rlm    CDATA "&#8207;"    -- marca de derecha a izquierda,
U+200F NEW RFC 2070 -->

```

```
<!ENTITY ndash    CDATA "&#8211;" -- gui3n en, U+2013 ISOpub -->
<!ENTITY mdash    CDATA "&#8212;" -- gui3n em, U+2014 ISOpub -->
<!ENTITY lsquo    CDATA "&#8216;" -- comilla simple abierta, U+2018 ISOnum -->
<!ENTITY rsquo    CDATA "&#8217;" -- comilla simple cerrada, U+2019 ISOnum -->
<!ENTITY sbquo    CDATA "&#8218;" -- comilla simple inferior, U+201A NEW -->
<!ENTITY ldquo    CDATA "&#8220;" -- comillas dobles abiertas, U+201C ISOnum -->
<!ENTITY rdquo    CDATA "&#8221;" -- comillas dobles cerradas, U+201D ISOnum -->
<!ENTITY bdquo    CDATA "&#8222;" -- comillas dobles inferiores, U+201E NEW -->
<!ENTITY dagger   CDATA "&#8224;" -- daga, U+2020 ISOpub -->
<!ENTITY Dagger   CDATA "&#8225;" -- daga doble, U+2021 ISOpub -->
<!ENTITY permil   CDATA "&#8240;" -- por mil, U+2030 ISOtech -->
<!ENTITY lsaquo   CDATA "&#8249;" -- comilla simple angular abierta,
                    U+2039 ISO proposed -->
<!-- lsaquo est1a propuesto pero a3n no es est1andar ISO -->
<!ENTITY rsaquo   CDATA "&#8250;" -- comilla simple angular cerrada,
                    U+203A ISO proposed -->
<!-- rsaquo est1a propuesto pero a3n no es est1andar ISO -->
<!ENTITY euro     CDATA "&#8364;" -- s3mbolo del euro, U+20AC NEW -->
```

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [3ndice](#)

Apéndice A: Cambios

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Cambios entre la versión de HTML 4.0 del 24 de abril de 1998 y la versión de HTML 4.01 del 24 de diciembre de 1999](#)
 1. [Cambios en la especificación](#)
 - [Cambios generales](#)
 - [Sobre SGML y HTML](#)
 - [Representación de Documentos HTML](#)
 - [Tipos de datos básicos de HTML](#)
 - [La estructura global de un documento HTML](#)
 - [Información sobre el idioma y la dirección del texto](#)
 - [Tablas](#)
 - [Vínculos](#)
 - [Objetos, Imágenes y Aplicaciones](#)
 - [Hojas de estilo en documentos HTML](#)
 - [Marcos](#)
 - [Formularios](#)
 - [Declaración SGML](#)
 - [DTD Estricto](#)
 - [Notas](#)
 - [Referencias](#)
 2. [Erratas corregidas](#)
 3. [Pequeños errores tipográficos corregidos](#)
 4. [Clarificaciones](#)
 5. [Problemas conocidos de los navegadores](#)
2. [Cambios entre las versiones del 18 de diciembre de 1997 y del 24 de abril de 1998](#)
 1. [Erratas corregidas](#)
 2. [Pequeños errores tipográficos corregidos](#)
3. [Cambios entre HTML 3.2 y HTML 4.0 \(18 de diciembre de 1997\)](#)
 1. [Cambios en los elementos](#)
 - [Elementos nuevos](#)
 - [Elementos desaprobados](#)
 - [Elementos obsoletos](#)
 2. [Cambios en los atributos](#)
 3. [Cambios relacionados con la accesibilidad](#)
 4. [Cambios relacionados con los metadatos](#)

5. [Cambios relacionados con el texto](#)
6. [Cambios relacionados con los vínculos](#)
7. [Cambios relacionados con las tablas](#)
8. [Cambios relacionados con las imágenes, objetos y mapas de imágenes](#)
9. [Cambios relacionados con los formularios](#)
10. [Cambios relacionados con las hojas de estilo](#)
11. [Cambios relacionados con los marcos](#)
12. [Cambios relacionados con los scripts](#)
13. [Cambios relacionados con la internacionalización](#)

A.1 Cambios entre la versión de HTML 4.0 del 24 de abril de 1998 y la versión de HTML 4.01 del 24 de diciembre de 1999

Esta sección describe en qué se diferencian la versión del 24 de diciembre de 1999 de la especificación de HTML 4.01 y la versión del 24 de abril de 1998 de la especificación de HTML 4.0.

A.1.1 Cambios en la especificación

Cambios generales

- Nuevas hojas de estilo para el documento basadas en los estilos de los informes técnicos del W3C.
- Se añadió una [tabla de contenidos abreviada](#).
- Se actualizó el copyright.
- Se eliminaron partes de código que hacían que algunos navegadores se colgaran.
- Se añadió a Shane McCarron a la sección de [agradecimientos](#).
- En la [sección 1.4](#) se eliminaron los detalles del copyright para hacer referencia al sitio del W3C en su lugar.
- Las referencias al conjunto de caracteres del documento son todas a la ISO 10646 (y una vez a UNICODE para señalar su equivalencia). Las referencias a UNICODE se refieren sólo al algoritmo bidireccional.
- Los ejemplos usan ahora FPIs con fecha.

[Sobre SGML y HTML](#)

- [Sección 3.2.2](#): Los valores de los atributos también pueden contener el carácter dos puntos y subguiones.

[Representación de Documentos HTML](#)

- [El Conjunto de Caracteres del Documento](#): Ahora sólo se usa [ISO10646] para las referencias al conjunto de caracteres del documento. [UNICODE] se reserva para las referencias relacionadas con el algoritmo bidireccional.

[Tipos de datos básicos de HTML](#)

- [Descriptores de medios](#): Todos los caracteres de los ejemplos se describen ahora usando notación hexadecimal (y haciendo referencia a la ISO 10646 en lugar de a Unicode).

[La estructura global de un documento HTML](#)

- [7.2 Información sobre la versión de HTML](#): Se ha añadido una nota sobre el compromiso del Grupo de Trabajo HTML de que:
 - Los posibles cambios en los DTDs futuros de HTML 4 no invalidarán los documentos que sean conformes con los DTDs de la presente especificación. El Grupo de Trabajo HTML se reserva el derecho de corregir los errores conocidos.
 - Los programas que sean conformes con los DTDs de la presente especificación pueden no tener en cuenta las

características de los DTDs futuros de HTML 4 que no reconozcan.

- [7.2 Información sobre la versión de HTML](#): Se usan como identificadores de sistema URIs de HTML sin fecha. Estos URIs son también los que se usan globalmente en todos los ejemplos.
- [7.4.4 Metadatos](#): Se ha eliminado la nota relativa al trabajo del W3C sobre metadatos y se ha reemplazado con una nota sobre el RDF.
- [7.4.4.2 Metadatos](#): Al final de la sección sobre encabezados HTTP se ha eliminado el ejemplo de auto-actualización (ya que no es parte de la Recomendación) y se ha añadido una nota para usar redirecciones del lado del servidor.

[Información sobre el idioma y la dirección del texto](#)

- [El atributo dir](#): Se clarifica que el atributo [dir](#) se aplica al contenido del elemento, a los valores de los atributos y la dirección de las tablas.

[Tablas](#)

- [11.2.6 Celdas de una tabla](#): Se han cambiado las definiciones de rowspan y colspan. Ahora los "spans" están limitados por grupos (rowgroups o colgroups).
- [11.3.2 Celdas de una tabla](#): Cuando "char=align" no está soportado por el agente de usuario, el comportamiento queda sin definir.

[Vínculos](#)

- [12.2 El elemento A](#): Se ha modificado la descripción del atributo [type](#) de los elementos [A](#) y ([LINK](#)) para enfatizar su naturaleza consultiva.
- [12.2.3 Vínculos con el atributo id](#): Es legal que "name" e "id" aparezcan en la misma etiqueta inicial cuando se definen ambos para un elemento. Deben tener valores idénticos.
- [12.3.3 Vínculos y motores de búsqueda](#): Se ha eliminado la referencia al atributo [dir](#) del ejemplo ya que no se aplica a los recursos vinculados (solamente al contenido del elemento y a los valores textuales de los atributos).
- [12.4.1 Cómo completar los URIs relativos](#): Como RFC 2616 no incluye un campo de encabezado Link, la siguiente frase sólo es válida para versiones anteriores a HTTP 1.1: "Los elementos Link especificados mediante encabezados HTTP se tratan exactamente igual que los elementos LINK que aparezcan explícitamente en un documento."

[Objetos, Imágenes y Aplicaciones](#)

- [13.2 El elemento IMG](#): Se ha añadido el atributo [name](#) por compatibilidad con versiones anteriores.
- [13.2 El elemento IMG](#): Se ha añadido una nota sobre la obligación de los agentes de usuario de proporcionar mecanismos diferentes para acceder al URI del "longdesc" (de IMG) y al URI del "src" (de A) cuando IMG sea parte del contenido de un elemento A.
- [13.3 El elemento OBJECT](#): Se ha añadido una nota para indicar que cuando el valor del tipo "type" de OBJECT y el encabezado HTTP difieren, el último prevalece.
- [13.3 El elemento OBJECT](#): Se ha añadido una frase para usar PARAM en lugar de usar al mismo tiempo los atributos "data" y "classid" de OBJECT.
- [13.4 El elemento APPLET](#): Se ha añadido una nota para indicar que, por razones de seguridad, el atributo "codebase" de APPLET sólo se busca en los subdirectorios del directorio actual.
- [13.6.1 Mapas de imágenes en el lado del cliente](#): La definición del atributo "poly" se ha hecho más clara. Una nota en el atributo "coords" de AREA indica que si los autores no cierran un polígono los agentes de usuario deberían cerrarlo.
- [13.6.1 Mapas de imágenes en el lado del cliente](#):
 - El modelo de contenidos del elemento [MAP](#) permite ahora a los autores *mezclar* el contenido [AREA](#) con contenido en bloque.
 - Los agentes de usuario "**deberían**" representar el contenido en bloque (antes era "pueden").
 - El elemento MAP se puede usar sin una imagen por mecanismos generales de navegación.
 - Los agentes de usuario no deben tener en cuenta los elementos [AREA](#) cuando su contenido sea mixto ([AREA](#) y

bloques).

- Los autores deberían especificar completamente las geometrías o bien con elementos [AREA](#), o con elementos [A](#) a nivel de bloque, o con ambos.
- [13.7.2](#) y [13.7.3](#) : Las definiciones de los atributos vspace y hspace son ahora similares a las definiciones de los demás atributos.
- [13.7.2](#) y [13.7.3](#) : El tipo de los valores de los atributos vspace, hspace y border se ha cambiado de "longitud" a "píxeles".
- [13.8 Texto alternativo](#): La última frase de la sección se enlaza con una notas para los desarrolladores de agentes de usuario relativas al tratamiento de textos vacíos para el atributo "alt".

[Hojas de estilo en documentos HTML](#)

- [14.6 Vinculación de hojas de estilo mediante encabezados HTTP](#): Debido a que RFC 2616 no incluye un campo de encabezado Link, toda esta sección se aplica sólo a las versiones anteriores a la HTTP 1.1.

[Marcos](#)

- [16.4.1 NOFRAMES](#): Se ha añadido un texto a la descripción de NOFRAMES sobre la representación cuando (1) los marcos están desactivados y (2) los marcos no están soportados.
- [16.4.1 NOFRAMES](#): Se ha añadido un texto sobre qué DTDs pueden tener NOFRAMES (con marcos, transicional).

[Formularios](#)

- [17.2.1 Tipos de controles](#): En la descripción de los radiobotones, si ningún radiobotón está seleccionado inicialmente, el comportamiento del agente de usuario a la hora de seleccionar uno queda sin definir. Esto difiere de RFC 1866.
- [17.3 El elemento FORM](#): Se ha añadido el atributo [name](#) por compatibilidad con versiones anteriores.
- [17.3 El elemento FORM](#): Se ha eliminado la referencia al URI "mailto" en la definición del atributo "action".
- [17.3 El elemento FORM](#): Se ha eliminado el ejemplo "mailto" del final de la sección ya que su comportamiento no está definido en esta especificación.
- [17.3 El elemento FORM](#): Se ha añadido el atributo [accept](#) al fragmento del DTD. Además, se ha corregido la descripción del atributo [accept-charset](#).
- [17.4 El elemento INPUT](#): Se ha añadido "ismap" al elemento INPUT. Además, en la definición de [value](#), se ha añadido "checkbox" a los valores de [type](#) que requieren un valor.
- [17.6.1](#): Cuando no hay ninguna opción preseleccionada, el comportamiento del agente de usuario queda sin definir. Los autores deberían proporcionar una opción explícita para cubrir este caso. Este comportamiento difiere de RFC 1866.

[Declaración SGML](#)

- [Declaración SGML de HTML 4](#): Se ha eliminado el texto sobre las referencias actualizadas a la ISO 10646. Se ha reemplazado por: "Las revisiones de la especificación HTML 4 pueden actualizar la referencia a ISO 10646 para incluir cambios adicionales."

[DTD Estricto](#)

- Atributos vspace/hspace/border de IMG, OBJECT, APPLET en píxeles.
- Se ha cambiado el modelo de contenido de MAP a ((%block;)| AREA)+
- Se ha añadido el atributo "ismap" a INPUT
- Se ha añadido el atributo [accept](#) al fragmento del DTD del elemento [FORM](#).
- Se ha cambiado el comentario del atributo [axis](#) para referirse a una lista separada por comas.
- El atributo [archive](#) del elemento [OBJECT](#) toma un valor de tipo CDATA en lugar de tipo %URI, ya que el valor es una lista de URIs separados por espacios.

Notas

- [Notas](#): Se han actualizado las notas sobre accesibilidad para apuntar a las Guías de Accesibilidad al Contenido de la Web.

Referencias

- Vínculos actualizados a los RFCs, ahora se usa <http://www.ietf.org/rfc>
- Se han puesto los vínculos en títulos.
- Se ha añadido la fecha de la revisión del 27 de agosto de 1998 de [DATETIME]
- Se ha añadido la fecha de la revisión del 11 de enero de 1999 de [CSS1].
- Se ha corregido la fecha de publicación de [CSS2].
- Se ha actualizado [UNICODE] a la versión 3.0
- Se ha actualizado [ISO10646] para permitir nuevas asignaciones de caracteres. Obsérvese que la enmienda cinco se tiene en cuenta implícitamente.
- Se espera que [RFC1766] vaya a actualizarse.
- [RFC2279] declara obsoleto a [RFC2044].
- [RFC2616] declara obsoleto a [RFC2068].
- Se ha añadido [RFC2388] además de [RFC1867].
- Actualizada la dirección de [LEXHTML], se ha añadido la fecha.
- Se ha actualizado la dirección de [DCORE].
- Se ha actualizado [WEBSGML]
- Se ha actualizado la dirección de [HTML3STYLE].
- Se ha añadido [RDF10] (reemplaza al antiguo RDF)
- Se ha cambiado [WAIGUIDE] -> [WAI]
- Se han añadido las referencias informativas [WCGL], [UAGL] y [ATGL]
- Se actualizado la referencia a [URI] (RFC 2396).

A.1.2 Erratas corregidas

Sección 13.6.1

Los ejemplos de mapas de imágenes que usaban "poly" se han corregido para formar un polígono cerrado. Además, el último par de coordenadas es el mismo que el primer par para cerrar el polígono.

Sección 14.4.1

En el ejemplo final, al elemento [STYLE](#) le falta la asignación del atributo "media=screen, print".

Sección 15.2.1

En el ejemplo con "mipar", la regla CSS debería decir:

```
P#mipar {font-style: italic; color: blue}
```

En CSS, "#" se refiere al nombre ID, mientras que "." se refiere a un nombre de clase. Este ejemplo utiliza el atributo "id".

Sección 16.2.2

Los valores de [marginwidth](#) y [marginheight](#) deben ser 0 píxeles o más, no 1 píxel o más.

Sección 16.2.2

El elemento [FRAME](#) no tiene un atributo [target](#).

Sección 16.5

El elemento [IFRAME](#) no tiene un atributo [target](#).

Sección 17.2.1

En la descripción de "checkboxes", se ha cambiado "selected" por "checked" en "cuando se establece el atributo

selected del elemento de control".

[Sección 17.6.1](#)

En la sección "Atributos definidos en otros lugares" del elemento [OPTGROUP](#), no deberían aparecer los atributos [onfocus](#), [onblur](#) y [onchange](#).

[Sección 18.2.3](#)

A la lista de elementos que toman [onfocus](#) y [onblur](#), sumar [A](#) y [AREA](#).

[Sección 20](#)

La declaración SGML de HTML 4 debe modificarse ligeramente para soportar las referencias numéricas hexadecimales de caracteres. Las líneas:

```
DELIM
    GENERAL SGMLREF
    SHORTREF SGMLREF
```

deben cambiarse por:

```
DELIM
    GENERAL SGMLREF
    HCRO "&#38;#x" -- 38 es el número del ampersand --
    SHORTREF SGMLREF
```

Y el `<!SGML "ISO 8879:1986"` inicial debe cambiarse por `<!SGML "ISO 8879:1986 (WWW)"`.

[Sección 21](#)

El elemento [HR](#) también debería tener los atributos [lang](#) y [dir](#). En la definición del elemento se dice que éstos están definidos en otros lugares, pero no se incluyeron en los DTDs.

[Sección 21](#)

El atributo `archive` del elemento [OBJECT](#), según se define en el DTD, toma un valor de tipo `%URI`. Esto es incorrecto: el valor puede ser una lista de URIs separados por espacios (como se indica en la definición del atributo en el comentario del DTD).

El fragmento de DTD del elemento [FORM](#) debería incluir una definición del atributo [accept](#), que se incluye en la definición del elemento. La definición debería ser la siguiente:

```
accept %ContentTypes; #IMPLIED -- lista de tipos MIME para subir ficheros --
```

[Sección B.4.1](#)

Al final de la sección, las frases siguientes son incorrectas: "La lista de términos en el contenido es ALL, INDEX, NOFOLLOW, NOINDEX. Los valores de los atributos `name` y `content` no hacen distinción entre mayúsculas y minúsculas". En realidad, la definición de [META](#) especifica que los valores de los atributos `name` y [content](#) sí distinguen entre mayúsculas y minúsculas.

[Sección B.4.1.1](#)

La especificación dice: "No se permiten líneas en blanco". Las líneas en blanco sí se permiten en el fichero `robots.txt`, pero no dentro de un "registro" dado. Obsérvese que la especificación no define registro.

Más adelante en la misma página, la especificación dice: "Debe haber exactamente un 'User-agent' por registro". En realidad, puede haber más de un campo `User-Agent` en el fichero `robots.txt`, pero no más de uno por registro.

Para más información sobre robots de búsqueda, consulte, por ejemplo:

- <http://www.kollar.com/robots.html>
- <http://info.webcrawler.com/mak/projects/robots/norobots-rfc.html>
- <http://info.webcrawler.com/mak/projects/robots/robots.html>

[Referencias](#)

- La referencia [\[URI\]](#) debería actualizarse a RFC 2396 de agosto 1998. "Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee, R. Fielding, L. Masinter, de agosto de 1998. El RFC 2396 actualiza [\[RFC1738\]](#) y [\[RFC1808\]](#).

A.1.3 Pequeños errores tipográficos corregidos

Nota del traductor: se ha omitido la relación de pequeños errores tipográficos corregidos porque no tiene especial interés en esta versión traducida.

A.1.4 Clarificaciones

[Sección 3.2.1](#)

En el séptimo párrafo se ha añadido "hasta su etiqueta inicial correspondiente" a "(p.ej., deben anidarse correctamente, una etiqueta final cierra hasta su etiqueta inicial correspondiente todas las etiquetas iniciales intermedias abiertas (sección 7.5.1.), etc.)."

[Sección 3.2.4](#)

Se ha añadido una frase que dice que los comentarios forman parte del código.

[Sección 3.3.3](#)

En el segundo objeto de la lista se ha cambiado "Si la etiqueta final del elemento" por "Si las etiquetas del elemento".

[Sección 3.3.3.1](#)

En la definición de un modelo de contenido, "A" quiere decir que "A" debe ocurrir una y sólo una vez. Además se han añadido "+(A)" y "-(A)" a la sección sobre la sintaxis del modelo de contenido.

[Sección 7.4.2](#)

Se ha clarificado que [TITLE](#) no puede incluir comentarios.

[Sección 10.3](#)

Todas las apariciones de "cracker" en esta sección y sus subsecciones se han sustituido por "hacker". Las definiciones de "hacker" y "nerd" se han tomado de "The Hacker's Dictionary".

[Sección 13.7.2](#)

Los atributos [hspace](#) y [vspace](#) están desaprobados.

[Sección 13.7.4](#)

El atributo [align](#) está desaprobadado para los elementos [IMG](#), [OBJECT](#) y [APPLET](#).

A.1.5 Problemas conocidos de los navegadores

- Algunas versiones de Netscape Navigator 4.0X se cuelgan al leer el Capítulo 3 de las versiones anteriores de esta especificación. Netscape es consciente de este fallo y lo ha corregido en la versión 4.5. Para eludir este problema vaya al submenú Editar/Preferencias/Avanzado y desactive las hojas de estilo (y posiblemente también el JavaScript).

A.2 Cambios entre las versiones del 18 de diciembre de 1997 y del 24 de abril de 1998

Esta sección describe las diferencias entre la versión de HTML 4.0 del 24 de abril de 1998 y la versión del 18 de diciembre de 1997.

A.2.1 Erratas corregidas

[Sección 2.1.1](#)

Se decía que "http://www.w3.org/TR/PR-html4/cover.html" designaba la especificación actual de HTML. La especificación actual de HTML está en realidad en <http://www.w3.org/TR/REC-html40>.

[Sección 7.5.2](#)

El hipervínculo de name estaba mal. Ahora apunta a <types.html#type-name>.

[Sección 7.5.4](#)

[href](#) figuraba como atributo de los elementos [DIV](#) y [SPAN](#). No lo es.

[Sección 7.5.6](#)

Se usaba un elemento [P](#) en el ejemplo. Es inválido en [ADDRESS](#).

[Sección 8.1](#)

En el primer ejemplo, que decía "Her super-powers were the result...", había unas comillas de más antes de la palabra "Her".

[Sección 9.3.4](#)

El atributo [width](#) no estaba marcado como [desaprobado](#).

[Sección 11.2.4, "Cálculo de la anchura de las columnas"](#)

La frase "We have set the value of the align attribute in the third column group to 'center'" decía "second" en lugar de "third".

[Sección 11.2.6, "Celdas que abarcan varias filas o columnas"](#)

El segundo párrafo decía "In this table definition, we specify that the cell in row four, column two should span a total of three columns, including the current row." Ahora acaba con "...including the current column."

[Sección 13.2](#)

La frase que comienza por "User agents must render alternate text when they cannot support ..." decía "next" en lugar de "text".

[Sección 13.6.2](#)

La última frase del segundo párrafo se aplica tanto al elemento [IMG](#) como al elemento [INPUT](#). Sin embargo, el atributo [ismap](#) no está definido para [INPUT](#). Ahora la frase sólo se aplica a [IMG](#).

[Sección 14.2.3](#)

El atributo [title](#) del elemento [STYLE](#) no figuraba en la lista de atributos definidos en otros lugares.

[Sección 14.3.2](#)

El segundo ejemplo hacía title="Compact". Ahora hace title="compact".

[Sección 15.1.2](#)

La sentencia que acaba por "the [align](#) attribute." decía "the [align](#) element."

[Sección 15.1.3.2](#)

La regla de estilo CSS "BR.mybr { clear: left }" estaba mal, ya que ésta se refiere a la clase "mybr" y no al valor del id. La sintaxis correcta es: "BR#mybr { clear: left }".

[Sección 16](#)

Todos los ejemplos que contenían una Declaración del Tipo de Documento usaban como identificador del sistema algo como "THE_LATEST_VERSION_/frameset.dtd" o "THE_LATEST_VERSION_" para el identificador del DTD de Documentos con Marcos. Ahora usan la declaración del tipo de documento correcta de acuerdo con la [Sección 7.2](#)

[Sección 16.3](#) y [Sección 16.3.1](#)

El segundo ejemplo de 16.3 y el ejemplo de 16.3.1 usaban el DTD incorrecto; ahora usan el DTD Transicional.

[Sección 17.5](#)

En los "atributos definidos en otros lugares" del elemento [BUTTON](#), faltaban [id](#), [class](#), [lang](#), [dir](#), [title](#), [style](#) y [tabindex](#). Además, se ha quitado [usemap](#).

[Sección 17.6/17.6.1](#)

En los "atributos definidos en otros lugares" de [OPTION](#) y [OPTGROUP](#) figuraban por error [onfocus](#), [onblur](#) y [onchange](#). Faltaba la sección de "atributos definidos en otros lugares" del elemento [SELECT](#) (consulte el DTD para una lista completa de atributos).

[Sección 17.9.1](#)

Se decía que el atributo [tabindex](#) estaba definido para el elemento [LABEL](#). No lo está.

[Sección 17.12.2](#)

La frase "Los siguientes elementos soportan el atributo [readonly](#): [INPUT](#) y [TEXTAREA](#)." decía "Los siguientes elementos soportan el atributo [readonly](#): [INPUT](#), [TEXT](#), [PASSWORD](#) y [TEXTAREA](#)."

[Sección 18.2.2, "Declaración local del lenguaje de un script"](#)

El primer párrafo decía: "También es posible especificar el lenguaje de scripts de cada elemento [SCRIPT](#) por medio del atributo type. Si no se especifica un lenguaje de scripts por defecto, debe especificarse este atributo para todos los elementos [SCRIPT](#)." Como el atributo type es obligatorio para el elemento [SCRIPT](#), ahora este párrafo dice: "Se debe especificar el atributo type de todos los elementos [SCRIPT](#) de un documento. El valor del atributo type de un elemento [SCRIPT](#) prevalece sobre el lenguaje de scripts por defecto de ese elemento."

[Sección 21](#)

Se añadió una nota que comentaba que la especificación incluía algunas restricciones sintácticas que no podían expresarse en el DTD.

[Sección 24.2.1](#) y fichero [HTMLlat1.ent](#)

El comentario de la referencia de caracteres "not" decía "= discretionary hyphen". Esto se ha eliminado. El FPI del comentario decía "-//W3C//ENTITIES Full Latin 1//EN//HTML", en lugar de "-//W3C//ENTITIES Latin1//EN//HTML".

[Sección 24.3.1](#) y fichero [HTMLsymbol.ent](#)

El FPI del comentario decía "-//W3C//ENTITIES Symbolic//EN//HTML", en lugar de "-//W3C//ENTITIES Symbols//EN//HTML".

[Sección A.1.1, "Elementos nuevos"](#) (previamente A.1.1) y [Sección A.1.1, "Elementos desaprobados"](#) (previamente A.1.2)

El elemento [S](#) que está [desaprobado](#) figuraba como parte de los cambios entre HTML 3.2 y HTML 4.0. Este elemento en realidad no estaba definido en [HTML 3.2](#). Ahora está en la lista de elementos nuevos.

[Sección A.1.3 \(previamente A.3\)](#)

Se decía que el atributo [longdesc](#) está especificado para las tablas. No lo está. En su lugar es el atributo [summary](#) el que permite a los autores dar descripciones más largas de las tablas.

[Sección B.4](#)

La frase "You may help search engines by using the [LINK](#) element with **rel="start"** along with the [title](#) attribute, ..." decía "You may help search engines by using the [LINK](#) element with **rel="begin"** along with a [TITLE](#), ..." Lo mismo para el ejemplo que acompaña.

[Sección B.5.1](#)

La frase "This can be altered by setting the [width](#) attribute of the [TABLE](#) element." decía "This can be altered by setting the width-TABLE attribute of the [TABLE](#) element."

[Sección B.5.2](#)

La frase "Rules for handling objects too large for a column apply when the explicit or implied alignment results in a situation where the data exceeds the assigned width of the column." decía "too large for column". El significado de la frase no estaba claro ya que se refería a "reglas" que gobernaban una condición de error; el comportamiento del agente de usuario en condiciones de error se sale fuera del alcance de esta especificación.

[Índice de atributos](#)

El atributo [href](#) del elemento [BASE](#) estaba marcado como [desaprobado](#). No lo está. Sin embargo, tampoco está definido en el DTD Estricto.

El atributo [language](#) del elemento [SCRIPT](#) no estaba marcado como [desaprobado](#). Ahora lo está, y ya no está definido en el DTD Estricto.

A.2.2 Pequeños errores tipográficos corregidos

Nota del traductor: se ha omitido la relación de pequeños errores tipográficos corregidos porque no tiene especial interés en esta versión traducida.

A.3 Cambios entre HTML 3.2 y HTML 4.0 (18 de diciembre de 1997)

Esta sección describe las diferencias entre la especificación de HTML 4.0 del 18 de diciembre de 1997 y HTML 3.2 ([\[HTML32\]](#)).

A.3.1 Cambios en los elementos

Elementos nuevos

Los elementos nuevos en HTML 4.0 son: [ABBR](#), [ACRONYM](#), [BDO](#), [BUTTON](#), [COL](#), [COLGROUP](#), [DEL](#), [FIELDSET](#), [FRAME](#), [FRAMESET](#), [IFRAME](#), [INS](#), [LABEL](#), [LEGEND](#), [NOFRAMES](#), [NOSCRIPT](#), [OBJECT](#), [OPTGROUP](#), [PARAM](#), [S](#) (desaprobado), [SPAN](#), [TBODY](#), [TFOOT](#), [THEAD](#) y [Q](#).

Elementos desaprobados

Los siguientes elementos están [desaprobados](#): [APPLET](#), [BASEFONT](#), [CENTER](#), [DIR](#), [FONT](#), [ISINDEX](#), [MENU](#), [S](#), [STRIKE](#) y [U](#).

Elementos obsoletos

Los siguientes elementos están obsoletos: [LISTING](#), [PLAINTEXT](#) y [XMP](#). Para todos ellos, los autores deberían usar [PRE](#) en su lugar.

A.3.2 Cambios en los atributos

- Casi todos los atributos que especifican la presentación de un documento HTML (p.ej., colores, alineación, fuentes, gráficos, etc.) han sido [desaprobados](#) en favor de las hojas de estilo. La [lista de atributos](#) del apéndice indica qué atributos han sido [desaprobados](#).
- Los atributos [id](#) y [class](#) permiten a los autores asignar [información de nombre y clase](#) a los elementos para las hojas de estilo, como identificadores de vínculo, para los scripts, para declaraciones de objetos, para tratamientos genéricos del documento, etc.

A.3.3 Cambios relacionados con la accesibilidad

HTML 4.0 introduce muchos cambios para promover la [accesibilidad](#), entre los que se incluyen:

- El atributo [title](#) puede establecerse ahora en prácticamente todos los elementos.
- Los autores pueden proporcionar descripciones largas para las tablas (ver el atributo [summary](#)), las imágenes y los marcos (ver el atributo [longdesc](#)).

A.3.4 Cambios relacionados con los metadatos

Ahora los autores pueden especificar [perfiles](#) que proporcionen explicaciones sobre los metadatos especificados con los elementos [META](#) o [LINK](#).

A.3.5 Cambios relacionados con el texto

- Las nuevas características de [internacionalización](#) permiten a los autores especificar la dirección del texto y el idioma.
- Los elementos [INS](#) y [DEL](#) permiten a los autores marcar los cambios de sus documentos.
- Los elementos [ABBR](#) y [ACRONYM](#) permiten a los autores marcar abreviaturas y acrónimos en sus documentos.

A.3.6 Cambios relacionados con los vínculos

- El atributo [id](#) hace a cualquier elemento el destino de un vínculo.

A.3.7 Cambios relacionados con las tablas

El modelo de tablas de HTML 4.0 se ha desarrollado a partir de los trabajos anteriores en HTML+ y en el borrador inicial de [HTML3.0](#). Se ha extendido el modelo antiguo en respuesta a las demandas de los suministradores de información en los siguientes aspectos:

- Los autores pueden especificar tablas que se representen incrementalmente a medida que el agente de usuario recibe los datos.

- Los autores pueden especificar tablas que sean más accesibles a los usuarios con agentes de usuario no visuales.
- Los autores pueden especificar tablas con cabeceras y pies fijos. Los agentes de usuario pueden sacar partido de esto al desplazar tablas grandes o cuando representen las tablas en medios paginados.

El modelo de tablas de HTML 4.0 satisface las demandas de valores por defecto opcionales a nivel de columna para las propiedades de alineación, más flexibilidad en la especificación de los bordes y las líneas de división de las tablas, y la posibilidad de alinear alrededor de caracteres designados. Sin embargo, se espera que en el futuro sean las [hojas de estilo](#) las encargadas de realizar la labor de la representación de las tablas.

Se ha logrado además un objetivo importante al proporcionar compatibilidad con la implementación de tablas de Netscape, ampliamente extendida. Otro fin ha sido la simplificación de la importación de tablas conformes con el modelo SGML CALS. El último borrador hace el atributo [align](#) compatible con las últimas versiones de los navegadores más populares. Se han hecho algunas clarificaciones acerca del papel del atributo [dir](#) y sobre el comportamiento recomendado cuando se mezclan anchuras de columnas absolutas y relativas.

Se ha introducido un nuevo elemento, [COLGROUP](#), que permite que se agrupen conjuntos de columnas con anchuras y propiedades de alineación diferentes especificadas con uno o más elementos [COL](#). Se ha clarificado la semántica de [COLGROUP](#) con respecto a los borradores anteriores, y se ha sustituido `rules="basic"` por `rules="groups"`.

Se ha introducido el atributo [style](#) para extender las propiedades asociadas con los bordes y las partes interiores de un grupo de celdas. Por ejemplo, el estilo de línea: punteada, doble, fina/gruesa, etc.; el color/patrón de relleno del interior; los márgenes de las celdas e información sobre la fuente. Esto será objeto de una especificación complementaria de hojas de estilo.

Se han modificado los atributos [frame](#) y [rules](#) para evitar conflictos de nombres SGML entre ellos, y para evitar conflictos con los atributos [align](#) y [valign](#). Estos cambios fueron motivados, además, por el deseo de evitar problemas en el futuro si se extiende esta especificación para permitir los atributos [frame](#) y [rules](#) en otros elementos de tabla.

A.3.8 Cambios relacionados con las imágenes, objetos y mapas de imágenes

- El elemento [OBJECT](#) permite la inclusión genérica de objetos.
- Los elementos [IFRAME](#) y [OBJECT](#) permiten a los autores crear documentos incluidos.
- El atributo [alt](#) es obligatorio para los elementos [IMG](#) y [AREA](#).
- El mecanismo para crear [mapas de imágenes](#) permite ahora a los autores crear mapas de imágenes más accesibles. El modelo de contenidos del elemento [MAP](#) ha cambiado por esta razón.

A.3.9 Cambios relacionados con los formularios

Esta especificación introduce varios atributos y elementos nuevos que afectan a los formularios:

- El atributo [accesskey](#) permite a los autores especificar teclas para el acceso directo a los controles del formulario.
- El atributo [disabled](#) permite a los autores hacer un control inicialmente insensible.
- El atributo [readonly](#) permite a los autores prohibir cambios en un control de formulario.
- El elemento [LABEL](#) asocia un rótulo con un control de formulario específico.
- El elemento [FIELDSET](#) agrupa campos relacionados y, junto al elemento [LEGEND](#), se puede utilizar como el nombre del grupo. Estos dos nuevos elementos permiten una mejor representación y mayor interactividad. Los navegadores basados en voz pueden describir mejor el formulario y los navegadores gráficos pueden hacer los rótulos sensibles.
- Un nuevo conjunto de atributos, en combinación con [scripts](#), permite a los suministradores de formularios verificar los datos introducidos por el usuario.
- El elemento [BUTTON](#) y el elemento [INPUT](#) con `type` igual a "button" se pueden utilizar junto con [scripts](#) para crear formularios más ricos.
- El elemento [OPTGROUP](#) permite a los autores agrupar opciones de menú en un [SELECT](#), lo cual es particularmente importante desde el punto de vista de la accesibilidad de los formularios.
- Cambios adicionales relacionados con la [internacionalización](#).

A.3.10 Cambios relacionados con las hojas de estilo

HTML 4.0 soporta un conjunto mayor de [descriptores de medios](#) para que los autores puedan escribir hojas de estilo dependientes del dispositivo.

A.3.11 Cambios relacionados con los marcos

HTML 4.0 soporta documentos con marcos y marcos en línea.

A.3.12 Cambios relacionados con los scripts

Ahora muchos elementos tienen [atributos de eventos](#) que pueden combinarse con scripts; el script se ejecuta cuando el evento ocurre (p.ej., cuando se carga el documento, cuando se hace clic con el ratón, etc.).

A.3.13 Cambios relacionados con la internacionalización

HTML 4.0 integra las recomendaciones de [\[RFC2070\]](#) para la internacionalización de HTML.

Sin embargo, esta especificación difiere de [\[RFC2070\]](#) en lo siguiente:

- El atributo [accept-charset](#) se ha especificado para el elemento [FORM](#) y no para los elementos [TEXTAREA](#) e [INPUT](#).
- La especificación HTML 4.0 hace clarificaciones adicionales con respecto al [algoritmo bidireccional](#).
- La utilización de [CDATA](#) para definir los elementos [SCRIPT](#) y [STYLE](#) no conserva la capacidad de transcodificar los documentos, como se describe en la sección 2.1 de [\[RFC2070\]](#).

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

Apéndice B: Notas sobre Aplicación, Implementación y Diseño

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Notas sobre documentos no válidos](#)
2. [Caracteres especiales en valores de atributos URI](#)
 1. [Caracteres no ASCII en valores de atributos URI](#)
 2. [El signo & en valores de atributos URI](#)
3. [Notas de implementación de SGML](#)
 1. [Saltos de línea](#)
 2. [Especificación de datos no HTML](#)
 - [Contenido de elementos](#)
 - [Valores de atributos](#)
 3. [Características de SGML con soporte limitado](#)
 4. [Atributos booleanos](#)
 5. [Secciones marcadas](#)
 6. [Instrucciones de procesamiento](#)
 7. [Código abreviado](#)
4. [Cómo ayudar a los motores de búsqueda a indexar su sitio web](#)
 1. [Motores de búsqueda](#)
 - [El fichero robots.txt](#)
 - [Los robots y el elemento META](#)
5. [Notas sobre tablas](#)
 1. [Criterios de diseño](#)
 - [Reformateo dinámico](#)
 - [Representación incremental](#)
 - [Estructura y presentación](#)
 - [Grupos de filas y de columnas](#)
 - [Accesibilidad](#)
 2. [Algoritmos recomendados de composición](#)
 - [Algoritmo de Composición Fija](#)
 - [Algoritmo de Autocomposición](#)
6. [Notas sobre formularios](#)
 1. [Representación incremental](#)
 2. [Proyectos futuros](#)
7. [Notas sobre scripts](#)

1. [Sintaxis reservada para macros de scripts futuras](#)

- [Práctica actual en macros de scripts](#)

8. [Notas sobre marcos](#)

9. [Notas sobre accesibilidad](#)

10. [Notas sobre seguridad](#)

1. [Cuestiones de seguridad relacionadas con los formularios](#)

Las siguientes notas son informativas, no normativas. A pesar de la aparición de palabras tales como "debe" y "debería", todos las exigencias de esta sección aparecen en otros lugares de la especificación.

B.1 Notas sobre documentos no válidos

Esta especificación no define el modo en que los agentes de usuario conformes tratan las condiciones generales de error, incluyendo el comportamiento de los agentes de usuario cuando encuentran elementos, atributos, valores de atributos, o entidades no especificadas en este documento.

Sin embargo, para facilitar la experimentación y la interoperabilidad entre las implementaciones de las diferentes versiones de HTML, recomendamos el comportamiento siguiente:

- Si un agente de usuario encuentra un elemento que no reconoce, debería intentar representar el contenido del elemento.
- Si un agente de usuario encuentra un atributo que no reconoce, debería pasar por alto la especificación completa del atributo (es decir, no tener en cuenta ni el atributo ni su valor).
- Si un agente de usuario encuentra un valor de atributo que no reconoce, debería usar el valor por defecto del atributo.
- Si encuentra una entidad no declarada, la entidad debería tratarse como datos de caracteres.

También recomendamos que los agentes de usuario proporcionen soporte para notificar tales errores al usuario.

Como los agentes de usuario pueden tratar las condiciones de error de maneras diferentes, los autores y los usuarios no deberían tomar como norma ningún comportamiento de recuperación de errores en particular.

La especificación HTML 2.0 ([RFC1866](#)) observa que muchos agentes de usuario HTML 2.0 suponen que un documento que no comience con una declaración del tipo de documento se refiere a la especificación HTML 2.0. Como la experiencia ha demostrado que esta es una mala suposición, la especificación actual no recomienda este comportamiento.

Por razones de interoperabilidad, los autores no deben "extender" el HTML por medio de mecanismos SGML disponibles (p.ej., extendiendo los DTDs, añadiendo un nuevo conjunto de definiciones de entidades, etc.).

B.2 Caracteres especiales en valores de atributos URI

B.2.1 Caracteres no ASCII en valores de atributos URI

Aunque los URIs no contienen valores no ASCII (ver [URI](#), sección 2.1) a veces los autores los especifican en valores de atributos que esperan URIs (es decir, definidos con [%URI](#); en el [DTD](#)). Por ejemplo, el siguiente valor de [href](#) es **ilegal**:

```
<A href="http://blabla.org/Håkon">...</A>
```

Recomendamos que los agentes de usuario adopten la siguiente convención para el tratamiento de caracteres no ASCII en estos casos:

1. Representar cada carácter en UTF-8 (ver [RFC2279](#)) con uno o más bytes.
2. Transformar estos bytes en secuencias de escape con el mecanismo de escape URI (es decir, convirtiendo cada byte a %HH, donde HH es la notación hexadecimal del valor del byte).

Este procedimiento resulta en un URI sintácticamente legal (según se define en [RFC1738](#), sección 2.2 o [RFC2141](#), sección 2) que es independiente de la [codificación de caracteres](#) a la que pudiera haber sido transcodificado el documento HTML que llevaba el URI.

Nota. Algunos agentes de usuario antiguos procesan trivialmente los URIs de HTML usando los bytes de la [codificación de caracteres](#) en que fue recibido el documento. Algunos documentos HTML antiguos dependen de esta práctica y no funcionan cuando son transcodificados. Los agentes de usuario que deseen manipular estos documentos antiguos deberían, al recibir un URI que contenga caracteres fuera del conjunto legal, usar en primer lugar la convención basada en UTF-8. Únicamente en el caso de que el URI resultante no funcione deberían intentar construir el URI basándose en los bytes de la [codificación de caracteres](#) en que fue recibido el documento.

Nota. La misma conversión basada en UTF-8 debería aplicarse a los valores del atributo [name](#) del elemento [A](#).

B.2.2 El signo & en valores de atributos URI

El URI que se construye cuando se [envía un formulario](#) puede utilizarse como un vínculo "estilo ancla" (p.ej., en el atributo [href](#) del elemento [A](#)). Desafortunadamente, la utilización del carácter "&" para separar los campos del formulario interfiere con su uso en los valores de los atributos SGML para delimitar [referencias a entidades de caracteres](#). Por ejemplo, para usar el URI "http://host/?x=1&y=2" como URI de un destino de vínculo, debe escribirse `` o ``.

Recomendamos a los implementadores de servidores HTTP, y en particular a los implementadores de CGI, que soporten el uso de ";" en lugar de "&" para ahorrar a los autores las molestias de convertir los caracteres "&" de este modo.

B.3 Notas de implementación de SGML

B.3.1 Saltos de línea

SGML (ver [ISO8879](#), sección 7.6.1) especifica que un salto de línea que siga inmediatamente a una etiqueta inicial debe ser ignorado, así como también debe serlo un salto de línea inmediatamente antes de una etiqueta final. Esto se aplica a todos los elementos HTML sin excepción.

Los siguientes dos ejemplos de HTML deben ser representados de manera idéntica:

```
<P>Tomás está viendo la tele.</P>
```

```
<P>
Tomás está viendo la tele.
</P>
```

También deben serlo los siguientes dos ejemplos:

```
<A>Mi página web favorita</A>
```

```
<A>
Mi página web favorita
</A>
```

B.3.2 Especificación de datos no HTML

Los datos de [scripts](#) y [estilos](#) pueden aparecer como contenidos de elementos o como valores de atributos. Las siguientes secciones describen los límites entre el código HTML y los datos extraños a él.

Nota. El [DTD](#) define los datos de [scripts](#) y [estilo](#) como CDATA para los contenidos de ambos elementos y para los valores de los atributos. Las reglas de SGML no permiten [referencias de caracteres](#) en contenidos CDATA de elementos, pero sí las permiten en valores CDATA de atributos. Los autores deberían prestar atención especial cuando copien datos de [scripts](#) y [estilos](#) entre contenidos de elementos y valores de atributos.

Esta asimetría también implica que cuando se transcodifique desde una codificación más rica a una más pobre, el transcodificador no puede simplemente reemplazar los caracteres inconvertibles de datos de [scripts](#) o [estilo](#) a las referencias numéricas de caracteres correspondientes; debe analizar el documento HTML y conocer la sintaxis de los lenguajes de [scripts](#) y [hojas de estilo](#) para poder procesar los datos correctamente.

Contenido de elementos

Cuando los datos de scripts o estilo sean el contenido de un elemento ([SCRIPT](#) y [STYLE](#)), los datos comienzan inmediatamente después de la etiqueta inicial del elemento y terminan en el primer delimitador ETAGO ("`</`") seguido por un carácter inicial de nombre (`[a-zA-Z]`); obsérvese que esto puede no ser la etiqueta final del elemento. Los autores deberían por tanto transformar el carácter "`</`" en su secuencia de escape cuando aparezca dentro del contenido. Los mecanismos de escape son específicos de cada lenguaje de scripts o de hojas de estilo.

EJEMPLO ILEGAL:

Los siguientes datos de script contienen, incorrectamente, la secuencia "`<`" (como parte de "``") antes de la etiqueta final de [SCRIPT](#):

```
<SCRIPT type="text/javascript">
  document.write ("<EM>Esto no funcionará</EM>")
</SCRIPT>
```

En JavaScript, este código puede expresarse legalmente ocultando el delimitador ETAGO antes del carácter inicial del nombre SGML:

```
<SCRIPT type="text/javascript">
  document.write ("<EM>Esto funcionará<\/EM>")
</SCRIPT>
```

En Tcl, esto puede hacerse de la siguiente manera:

```
<SCRIPT type="text/tcl">
  document write "<EM>Esto funcionará<\/EM>"
</SCRIPT>
```

En VBScript puede evitarse el problema con el uso de la función `Chr ()`:

```
"<EM>Esto funcionará<" & Chr(47) & "EM>"
```

Valores de atributos

Cuando los datos de scripts o estilo sean el valor de un atributo (o bien [style](#) o bien los atributos de [eventos intrínsecos](#)), los autores deberían transformar en secuencias de escape todas las apariciones de comillas delimitadoras dobles o simples que aparezcan dentro del valor de acuerdo con la convención del lenguaje de scripts o de hojas de estilo. Los autores deberían también transformar las apariciones del carácter "&" si éste no se utiliza como comienzo de una [referencia de caracteres](#).

- "" debería escribirse como """ o """;
- '&' debería escribirse como "&" o "&";

Así, por ejemplo, se podría escribir:

```
<INPUT name="num" value="0"
onchange="if (comparar(this.value, &quot;ayuda&quot;)) {obtenerayuda()} ">
```

B.3.3 Características de SGML con soporte limitado

Los sistemas SGML conformes con [ISO8879](#) reconocen, en teoría, un cierto número de características que no están ampliamente soportadas por los agentes de usuario HTML. Recomendamos a los autores que eviten el uso de todas estas características.

B.3.4 Atributos booleanos

Los autores deberían saber que muchos agentes de usuario sólo reconocen la forma minimizada de los atributos booleanos, y no la forma completa.

Por ejemplo, es posible que los autores quieran especificar:

```
<OPTION selected>
```

en lugar de

```
<OPTION selected="selected">
```

B.3.5 Secciones marcadas

Las secciones marcadas juegan un papel similar a la estructura `#ifdef` reconocida por los preprocesadores C.

```
<![INCLUDE[
  <!-- esto será incluido -->
]]>
```

```
<![IGNORE[
  <!-- esto será ignorado -->
]]>
```

SGML también define el uso de secciones marcadas para contenido CDATA, dentro de las cuales "<" no se considera como el comienzo de una etiqueta, p.ej.,

```
<![CDATA[
  <un> ejemplo de código <sgml> que no es
  <difícil> de escribir con < y similares.
]]>
```

Lo que delata que un agente de usuario no reconoce una sección marcada es la aparición del "]]>", que sólo se ve cuando el agente de usuario considera por error al primer carácter ">" como el final de la etiqueta que comienza por "<![".

B.3.6 Instrucciones de procesamiento

Las instrucciones de procesamiento son un mecanismo para capturar órdenes específicas de la plataforma. Una instrucción de procesamiento comienza con `<?` y termina con `>`

```
<?instruccion >
```

Por ejemplo:

```
<?>
<?style tt = font courier>
<?page break>
<?experiment> ... <?/experiment>
```

Los autores deberían saber que muchos agentes de usuario representan las instrucciones de procesamiento como parte del texto del documento.

B.3.7 Código abreviado

Algunas estructuras SGML SHORTTAG permiten escribir menos pero no añaden capacidad expresiva a la aplicación SGML. Aunque estas estructuras no introducen ambigüedades desde el punto de vista técnico, reducen la robustez de los documentos, especialmente cuando se amplía el lenguaje para incluir nuevos elementos. Así, si bien las construcciones SHORTTAG de SGML relacionadas con los atributos son ampliamente usadas y soportadas, no sucede lo mismo con aquéllas relacionadas con los elementos. Los documentos que las usen serán documentos conformes con SGML, pero es poco probable que funcionen con muchas herramientas HTML existentes.

Las estructuras SHORTTAG en cuestión son las siguientes:

- Etiquetas NET:

```
<name/.../
```

- Etiqueta Inicial cerrada:

```
<name1<name2>
```

- Etiqueta Inicial vacía:

```
<>
```

- Etiqueta Final vacía:

```
</>
```

B.4 Cómo ayudar a los motores de búsqueda a indexar su sitio web

Esta sección proporciona algunas sugerencias simples que harán sus documentos más accesibles a los motores de búsqueda.

Defina el idioma del documento

En el contexto global de la Web es importante saber en qué idioma está escrita una página. Se habla sobre esto en la sección sobre [información sobre el idioma](#).

Especifique las versiones del documento en otros idiomas

Si ha preparado traducciones del documento en otros idiomas, debería usar el elemento [LINK](#) para hacer referencia a ellas. Esto permitirá al motor indexador ofrecer a sus usuarios resultados de búsqueda en el idioma preferido del usuario, independientemente de cómo se escribió la petición de búsqueda. Por ejemplo, los siguientes vínculos ofrecen versiones alternativas en francés y alemán a los motores de búsqueda:

```
<LINK rel="alternate"
      type="text/html"
      href="midoc-fr.html" hreflang="fr"
      lang="fr" title="La vie souterraine">
<LINK rel="alternate"
      type="text/html"
      href="midoc-de.html" hreflang="de"
      lang="de" title="Das Leben im Untergrund">
```

Proporcione palabras clave y descripciones

Algunos motores indexadores buscan elementos [META](#) que definen una lista de palabras clave o frases separadas por comas, o que den una descripción corta. El valor del atributo [name](#) buscado por el motor de búsqueda no está definido por esta especificación. Considere estos ejemplos,

```
<META name="keywords" content="vacaciones,Grecia,sol">
<META name="description" content="Vacaciones idílicas en Europa">
```

Indique el punto inicial de un conjunto

Muchas veces los conjuntos de documentos de procesadores de textos o de presentaciones se convierten a documentos HTML. Es útil para los resultados de las búsquedas hacer referencia al comienzo del conjunto además de la página obtenida tras la búsqueda. Puede ayudar a los motores de búsqueda usando el elemento [LINK](#) con **rel="start"** junto con el atributo [title](#), como aquí:

```
<LINK rel="start"
      type="text/html"
      href="paginal.html"
      title="Teoría General de la Relatividad">
```

Proporcione instrucciones de indexado a los robots

A algunas personas les puede sorprender que su sitio haya sido indexado por un robot indexador y que el robot no

haya pedido permiso para visitar una parte sensible del sitio. Muchos robots web ofrecen facilidades a los administradores de sitios web y a los proveedores de contenidos para limitar la actuación del robot. Esto se consigue por medio de dos mecanismos: un fichero "robots.txt" y el elemento [META](#) en los documentos HTML, según se describe más abajo.

B.4.1 Motores de búsqueda

El fichero robots.txt

Cuando un robot visita un sitio web, digamos `http://www.blabla.com/`, lo primero que hace es comprobar y analizar sus contenidos para ver si le está permitido obtener el documento. Se puede editar el fichero `robots.txt` para que se aplique sólo a robots específicos, y para impedir el acceso a ficheros o directorios específicos.

Aquí se muestra un fichero `robots.txt` de ejemplo que impide a todos los robots visitar todo el sitio

```
User-agent: *      # se aplica a todos los robots
Disallow: /       # impedir el indexado de todas las páginas
```

El robot simplemente buscará un URI `/robots.txt` en su sitio, estando el sitio definido por un servidor HTTP ejecutándose en una máquina determinada y en un número de puerto determinado. Aquí tenemos algunas localizaciones de ejemplo para `robots.txt`:

URI del Sitio	URI de robots.txt
<code>http://www.w3.org/</code>	<code>http://www.w3.org/robots.txt</code>
<code>http://www.w3.org:80/</code>	<code>http://www.w3.org:80/robots.txt</code>
<code>http://www.w3.org:1234/</code>	<code>http://www.w3.org:1234/robots.txt</code>
<code>http://w3.org/</code>	<code>http://w3.org/robots.txt</code>

Sólo puede haber un único fichero `/robots.txt` en un sitio. Más concretamente, no debería poner ficheros `"robots.txt"` en los subdirectorios de los usuarios, ya que los robots no los mirarán nunca. Si quiere que sus usuarios puedan crear sus propios ficheros `"robots.txt"`, tendrá que mezclarlos todos en un `/robots.txt` único. Si no quiere hacer esto, es posible que sus usuarios quieran utilizar el elemento META Robots en su lugar.

Un par de consejos: Los URI's distinguen entre mayúsculas y minúsculas, y todas las cadenas de `/robots.txt` deben estar en minúsculas. No se permiten líneas en blanco dentro de un registro simple del fichero `"robots.txt"`.

Debe haber exactamente un campo `"User-agent"` por registro. El robot debería ser liberal al interpretar este campo. Se recomienda un emparejamiento, sin distinguir entre mayúsculas y minúsculas, entre una subcadena y el nombre sin la información de la versión.

Si el valor es `"*"`, el registro describe la política de acceso por defecto para cualquier robot que no se haya emparejado con ninguno de los demás registros. No está permitido tener más de uno de estos registros en el fichero `"robots.txt"`.

El campo `"Disallow"` especifica un URI parcial que no debe ser visitado. Éste puede ser un ruta de acceso completa, o una ruta parcial; cualquier URI que comience con este valor no será accedido. Por ejemplo,

```
Disallow: /help  impide tanto /help.html como /help/index.html, mientras que
Disallow: /help/ impediría /help/index.html pero permitiría /help.html.
```

Un valor vacío para `"Disallow"` indica que se puede acceder a todos los URIs. Debe haber al menos un campo `"Disallow"` en el fichero `robots.txt`.

Los robots y el elemento META

El elemento [META](#) permite a los autores HTML decir a los robots que les visiten si un documento puede ser indexado, o usado para obtener más vínculos. No se requiere ninguna acción por parte del administrador.

En el siguiente ejemplo, el robot no debería ni indexar este documento, ni analizarlo para encontrar vínculos.

```
<META name="ROBOTS" content="NOINDEX, NOFOLLOW" >
```

La lista de términos en el contenido es ALL, INDEX, NOFOLLOW, NOINDEX.

Nota. A principios de 1997 eran muy pocos los robots que implementaban esto, pero se espera que esta situación cambie a medida que se preste más atención pública al control de los robots de indexado.

B.5 Notas sobre tablas

B.5.1 Criterios de diseño

El modelo de tablas de HTML ha evolucionado a partir de estudios de modelos de tablas SGML existentes, del tratamiento de tablas en varios paquetes comunes de procesamiento de textos, y de un amplio rango de técnicas de composición tabular en revistas, libros y otros documentos basados en papel. El modelo fue elegido para permitir que las tablas simples se pudieran expresar con simplicidad, con complejidad adicional cuando fuera necesaria. Gracias a ello puede crearse fácilmente el código para tablas sencillas con editores de texto normales y se reduce el proceso inicial de aprendizaje. Esta característica ha sido muy importante para el éxito actual del HTML.

Cada vez son más las personas que crean tablas convirtiéndolas desde otros formatos de documento, o creándolas directamente con editores WYSIWYG. Es importante el hecho de que el modelo de tablas de HTML se acomode bien a estas herramientas de creación, incluyendo el modo en que se presentan las celdas que abarcan varias filas o columnas, y cómo se asocian las propiedades de alineación y otras propiedades de presentación a grupos de celdas.

Reformateo dinámico

Una de las consideraciones principales respecto al modelo de tablas de HTML es que el autor no controla qué tamaño dará un usuario a la tabla, qué fuentes usará, etc. Esto hace que sea arriesgado basarse en anchuras de columna especificadas en términos de unidades absolutas en píxeles. En lugar de ello, las tablas deben poder modificar su tamaño dinámicamente para ajustarse al tamaño actual de la ventana y a las fuentes disponibles. Los autores pueden proporcionar pautas para los tamaños relativos de las columnas, pero los agentes de usuario deberían asegurarse de que las columnas sean lo suficientemente anchas para representar la anchura del mayor elemento contenido en la celda. Si la especificación del autor debe ser anulada, no deberían cambiar drásticamente las anchuras relativas de las columnas.

Representación incremental

Para tablas grandes o conexiones lentas a la red, la representación incremental de las tablas es importante para la satisfacción del usuario. Los agentes de usuario deberían empezar a representar una tabla antes de que se hayan recibido todos los datos. La anchura de ventana por defecto de la mayoría de los agentes de usuario muestra unos 80 caracteres, y los gráficos de muchas páginas HTML están diseñadas con estos valores por defecto en mente. Si se especifica el número de columnas, y si se incluyen medios para controlar la anchura de la tabla y las anchuras de las diferentes columnas, los autores pueden dar a los agentes de usuario indicaciones que les permitan representar incrementalmente los contenidos de la tabla.

Para la representación incremental, el navegador necesita el número de columnas y sus anchuras. La anchura por defecto de la tabla es el tamaño actual de la ventana (`width="100%"`). Esto puede alterarse estableciendo el atributo [width](#) del elemento [TABLE](#). Por defecto todas las columnas tienen la misma anchura, pero se pueden especificar las anchuras de las columnas con uno o más elementos [COL](#) antes de que comiencen los datos de la tabla.

El otro problema a resolver es el número de columnas. Algunas personas han sugerido esperar hasta que se haya recibido la primera fila de la tabla, pero esto podría llevar bastante tiempo si las celdas tienen mucho contenido. Por lo general, si lo que se pretende es la representación incremental, tiene más sentido que los autores especifiquen explícitamente el número de columnas en el elemento [TABLE](#).

Los autores aún necesitan un modo de decir a los agentes de usuario si usar la representación incremental o si dimensionar automáticamente la tabla para acomodar los contenidos de las celdas. En el modo de auto-dimensionado de dos pasadas, el número de columnas se determina en la primera pasada. En el modo incremental, el número de columnas debe declararse al principio (con elementos [COL](#) o [COLGROUP](#)).

Estructura y presentación

HTML distingue entre código estructural, como párrafos y citas, e instrucciones de presentación, como márgenes, fuentes, colores, etc. ¿Cómo afecta esta distinción a las tablas? Desde el punto de vista del purista, la alineación del texto dentro de las celdas de una tabla y las líneas entre las celdas son cuestiones de representación, y no de estructura. En la práctica, sin embargo, es útil agrupar éstas con la información estructural, ya que estas características son muy transportables de una aplicación a la siguiente. El modelo de tablas de HTML deja la mayor parte de la información de presentación a las hojas de estilo asociadas. El modelo presentado en esta especificación está diseñado para sacar provecho de dichas hojas de estilo, pero no obliga a usarlas.

Los paquetes actuales de publicación electrónica proporcionan gran control sobre la representación de las tablas, y sería impracticable reproducir esto en HTML sin convertir a HTML en un formato de texto enriquecido muy abultado como RTF o MIF. Sin embargo, esta especificación ofrece a los autores la posibilidad de elegir entre un conjunto de clases o estilos de bordes usados normalmente. El atributo [frame](#) controla la apariencia del borde que rodea a la tabla, mientras que el atributo [rules](#) determina la elección de las divisiones que habrá en el interior de la tabla. Por medio de anotaciones de representación se logrará un nivel de control más preciso. El atributo [style](#) puede utilizarse para especificar información de representación de elementos individuales. Y se puede dar más información de representación con el elemento [STYLE](#) de la cabecera del documento o por medio de hojas de estilo vinculadas.

Durante el desarrollo de esta especificación, se investigaron diferentes posibilidades para la especificación de los patrones de líneas de división en tablas. Una de las cuestiones se refiere a las clases de declaraciones que pueden hacerse. Si se incluye soporte para sustracción de bordes así como para adición de bordes, se obtienen algoritmos relativamente complejos. Por ejemplo, para incluir en todo el conjunto de elementos de tabla los atributos [frame](#) y [rules](#) fue necesario definir un algoritmo de 24 pasos para determinar si en un borde dado de una celda dada debería haber una línea de división o no. Ni siquiera esta complejidad adicional proporciona suficiente control sobre la representación como para satisfacer el espectro completo de necesidades relacionadas con las tablas. La especificación actual se atiene deliberadamente a un modelo simple e intuitivo, suficiente para la mayoría de los propósitos. Se necesita más trabajo experimental antes de que se estandarice una aproximación más compleja.

Grupos de filas y de columnas

Esta especificación proporciona un superconjunto del modelo más simple presentado en el trabajo anterior en HTML+. Se considera que las tablas están formadas por un título opcional con una secuencia de filas, las cuales consisten a su vez en una secuencia de celdas de tabla. El modelo distingue además entre celdas de encabezado y celdas de datos, y permite que las celdas abarquen varias filas y columnas.

Siguiendo el modelo de tablas de CALS (ver [\[CALS\]](#)), esta especificación permite que las filas de una tabla se agrupen en secciones de cabecera, cuerpo y pie. Esto simplifica la representación y puede usarse para repetir la cabecera y el pie de la tabla cuando ésta se extienda a lo largo de varias páginas, o para proporcionar cabeceras fijas sobre un panel desplazable para el cuerpo. En el código, la sección del pie se coloca antes de las secciones de cuerpo. Esto es una optimización compartida con CALS para trabajar con tablas muy largas. Permite que se represente el pie sin tener que esperar a que se procese la totalidad de la tabla.

Accesibilidad

Para aquéllos con discapacidades visuales, HTML ofrece la esperanza de enmendar el daño causado por la adopción de interfaces gráficas de usuario basadas en ventanas. El modelo de tablas de HTML incluye atributos para anotar cada celda, para permitir una conversión de texto a voz de mayor calidad. Los mismos atributos pueden usarse para permitir la importación y exportación automatizada de datos de tablas a bases de datos o a hojas de cálculo.

B.5.2 Algoritmos recomendados de composición

Si hay elementos [COL](#) o [COLGROUP](#) presentes, éstos especifican el número de columnas, y la tabla puede ser representada usando una composición fija. En caso contrario debería usarse el algoritmo de autocomposición descrito más adelante.

Si el atributo `width` no está especificado, los agentes de usuario visuales deberían suponer un valor por defecto del 100% a la hora de dar formato.

Se recomienda a los agentes de usuario que incrementen las anchuras de las columnas por encima del valor especificado por `width` en aquellos casos en que los contenidos de las celdas se desbordaran si no lo hicieran. Los agentes de usuario que redefinan la anchura especificada deberían hacerlo de manera razonable. Los agentes de usuario pueden optar por dividir las palabras entre líneas para evitar la necesidad del desplazamiento horizontal o cuando dicho desplazamiento sea poco

práctico o no deseable.

En lo que a la composición se refiere, los agentes de usuario deberían considerar que los títulos de las tablas (especificados con el elemento [CAPTION](#)) se comportan como las celdas. Cada título es una celda que abarca todas las columnas de la tabla si está en la parte superior o inferior de la tabla, o todas las filas si está a la izquierda o a la derecha de la tabla.

Algoritmo de Composición Fija

Para este algoritmo, se supone que el número de columnas es conocido. Las anchuras de las columnas deberían establecerse por defecto al mismo tamaño. Los autores pueden cancelar esto especificando anchuras de columnas relativas o absolutas, usando los elementos [COLGROUP](#) o [COL](#). La anchura por defecto de la tabla es el espacio entre los márgenes izquierdo y derecho actuales, pero puede redefinirse con el atributo [width](#) del elemento [TABLE](#), o determinarse a partir de las anchuras absolutas de las columnas. Para operar con mezclas de anchuras de columnas absolutas y relativas, el primer paso es reservar espacio de la anchura de la tabla para las columnas con anchuras absolutas. Después de eso, el espacio restante se divide entre las columnas con anchuras relativas.

La sintaxis de tablas es de por sí insuficiente para garantizar la consistencia de los valores de los atributos. Por ejemplo, el número de elementos [COL](#) y [COLGROUP](#) puede ser inconsistente con el número de columnas que se deriva de las celdas de la tabla. Otro problema se produce cuando las columnas son demasiado estrechas y los contenidos de las celdas se desbordan. La anchura de la tabla especificada con el elemento [TABLE](#) o con elementos [COL](#) puede hacer que los contenidos de las celdas se desborden. Se recomienda que los agentes de usuario intenten recuperarse elegantemente de estas situaciones, p.ej., [dividiendo palabras](#) o, como último recurso, partiendo palabras si los puntos de separación son desconocidos.

En el caso en que un elemento indivisible cause desbordamiento en una celda, el agente de usuario puede considerar ajustar las anchuras de las columnas y volver a representar la tabla. En el peor de los casos, puede optar por recortar el elemento si no es posible ajustar la anchura de la columna o desplazar el contenido de la celda. En cualquier caso, si el contenido de la celda se parte o se recorta, debería indicarse al usuario de manera apropiada.

Algoritmo de Autocomposición

Si el número de columnas no está especificado con elementos [COL](#) y [COLGROUP](#), entonces el agente de usuario debería usar el algoritmo de autocomposición. Éste hace dos pasadas por los datos de la tabla y aumenta linealmente con el tamaño de la tabla.

En la primera pasada se deshabilita el ajuste automático de líneas, y el agente de usuario lleva la cuenta de la anchura mínima y máxima de cada celda. La anchura máxima viene dada por la línea más ancha. Como el ajuste de líneas ha sido deshabilitado, los párrafos se consideran como líneas muy largas a menos que estén partidos por elementos [BR](#). La anchura mínima viene dada por el elemento de texto más ancho (palabra, imagen, etc.) teniendo en cuenta la sangría, los marcadores de lista, etc. En otras palabras, es necesario determinar la anchura mínima que necesita una celda antes de que la celda comience a desbordarse. Si se permite a los agentes de usuario dividir palabras se minimizará la necesidad de desplazamiento horizontal o, en el peor de los casos, el recorte de los contenidos de la celda.

Este proceso también se aplica a cualquier tabla anidada que aparezca en el contenido de las celdas. La anchuras mínimas y máximas de las celdas de las tablas anidadas se usan para determinar las anchuras mínimas y máximas de estas tablas, y por tanto de la propia celda de la tabla padre. El algoritmo es lineal con el contenido agregado en celdas, y en términos generales, independiente del nivel de anidamiento.

Para hacer frente a la alineación de caracteres en los contenidos de las celdas, el algoritmo mantiene tres valores min/max para cada columna: a la izquierda del carácter de alineación, a la derecha del carácter de alineación, y sin alineación. La anchura mínima de una columna es entonces: $\max(\text{min_left} + \text{min_right}, \text{min_non-aligned})$.

La anchuras de celda mínimas y máximas se usan entonces para determinar las anchuras mínimas y máximas correspondientes de las columnas. Éstas, a su vez, se usan para hallar la anchura mínima y máxima de la tabla. Obsérvese que las celdas pueden contener tablas anidadas, pero esto no complica el código significativamente. El siguiente paso es asignar anchuras a las columnas según el espacio disponible (es decir, el espacio entre los márgenes izquierdo y derecho actuales).

Para las celdas que abarquen varias columnas, una aproximación simple consiste en dividir uniformemente las anchuras min/max entre todas las columnas abarcadas. Una aproximación ligeramente más compleja es usar las anchuras min/max de las celdas que no abarcan más de una columna para ponderar las proporciones en que se dividen las anchuras. Los

experimentos sugieren que una mezcla de ambas aproximaciones da buenos resultados para un amplio espectro de tablas.

Es necesario incluir los bordes de las tablas y los márgenes entre celdas cuando se asignen las anchuras de las columnas. Hay tres casos:

1. **La anchura mínima de la tabla es igual o más ancha que el espacio disponible.** En este caso, se asignan las anchuras mínimas y se permite al usuario desplazarse horizontalmente. Para la conversión a Braille será necesario reemplazar las celdas por referencias a notas que contengan el contenido completo. Por convención éstas aparecen antes de la tabla.
2. **La anchura máxima de la tabla cabe en el espacio disponible.** En este caso se establecen las columnas en sus anchuras máximas.
3. **La anchura máxima de la tabla es mayor que el espacio disponible, pero la anchura mínima de la tabla es menor.** En este caso, se halla la diferencia **W** entre el espacio disponible y la anchura mínima de la tabla. Sea **D** la diferencia entre la anchura máxima y la anchura mínima de la tabla.

Para cada columna, sea **d** la diferencia entre las anchuras máxima y mínima de esa columna. Ahora se hace la anchura de la columna igual a la mínima más **d** por **W** partido por **D**. Esto hace que las columnas cuya diferencia entre las anchuras mínima y máxima sea grande sean más anchas que las columnas en que esta diferencia sea menor.

Este paso de asignación se repite entonces para las tablas anidadas usando las anchuras mínimas y máximas obtenidas para estas tablas en la primera pasada. En este caso, la anchura de la celda de la tabla padre hace el papel del tamaño actual de la ventana en la descripción precedente. Este proceso se repite recursivamente para todas las tablas anidadas. La tabla más exterior se representa entonces usando las anchuras asignadas. Las tablas se representan a continuación como parte de los contenidos de las celdas de la tabla padre.

Si la anchura de la tabla se especifica con el atributo [width](#), el agente de usuario intenta establecer las anchuras de las columnas para respetar este valor. El atributo [width](#) no es obligatorio si hace que haya columnas que tengan anchuras menores que su anchura mínima (es decir, indivisible).

Si se especifican anchuras relativas con el elemento [COL](#), se modifica el algoritmo para incrementar las anchuras de las columnas sobre la anchura mínima para cumplir con las restricciones de las anchuras relativas. Los elementos [COL](#) deberían tomarse como simples indicaciones, de modo que las columnas no deberían establecerse en una anchura menor que la mínima. Análogamente, las columnas no deberían hacerse tan anchas que la tabla se extendiera más allá de los límites de la ventana. Si un elemento [COL](#) especifica una anchura relativa de cero, la columna debería establecerse siempre en su anchura mínima.

Cuando se use el algoritmo de dos pasadas, la posición de alineación por defecto, en ausencia de un atributo [charoff](#) explícito o heredado, puede determinarse eligiendo la posición que centraría las líneas para las cuales las anchuras antes y después del carácter de alineación tengan los valores máximos para cualquiera de las líneas de la columna en que `align=char`. Para la composición incremental de tablas el valor por defecto sugerido es `charoff="50%"`. Si hay varias celdas en diferentes filas de una misma columna que usen un carácter de alineación, entonces por defecto todas estas celdas deberían alinearse, independientemente del carácter usado para la alineación. Se aplicarán reglas para el tratamiento de objetos demasiado grandes para una columna cuando la alineación explícita o implícita conduzca a una situación en que los datos excedan la anchura asignada para la columna.

***Elección de nombres de atributos.** Habría sido preferible elegir valores para el atributo [frame](#) consistentes con el atributo [rules](#) y con los valores usados para la alineación. Por ejemplo: `none`, `top`, `bottom`, `topbot`, `left`, `right`, `leftright`, `all`. Desafortunadamente, SGML requiere que los valores de atributos enumerados sean únicos para cada elemento, independientemente del nombre del atributo. Esto provoca problemas inmediatos para "none", "left", "right" y "all". Los valores del atributo [frame](#) se han elegido para evitar conflictos con los atributos [rules](#), [align](#) y [valign](#). Esto sirve como aviso para el futuro, ya que se anticipa que los atributos [frame](#) y [rules](#) se añadirán a otros elementos de tabla en revisiones futuras de esta especificación. Una alternativa habría sido hacer [frame](#) un atributo CDATA. El consenso del Grupo de Trabajo HTML del W3C fue que los beneficios de poder usar herramientas de validación SGML para comprobar los atributos basándose en valores enumerados compensaban la necesidad de nombres consistentes.*

B.6 Notas sobre formularios

B.6.1 Representación incremental

La representación incremental de los documentos que se reciben de la red da lugar a ciertos problemas relacionados con los formularios. Los agentes de usuario deberían evitar que los formularios fueran enviados antes de que se hayan recibido todos los elementos del formulario.

La representación incremental de los documentos plantea algunas cuestiones relacionadas con la navegación con tabulador. La heurística de dirigir el foco hacia el [tabindex](#) de menor valor del documento parece razonable a primera vista. Sin embargo, esto implica que se debe esperar a que se haya recibido todo el texto del documento, ya que hasta ese momento el [tabindex](#) de menor valor aún puede cambiar. Si el usuario pulsa el tabulador antes, es razonable que los agentes de usuario dirijan el foco al [tabindex](#) de menor valor disponible actualmente.

Si se asocian formularios con scripts en el lado del cliente, hay más problemas potenciales. Por ejemplo, un procesador de scripts para un cierto campo puede hacer referencia a un campo que aún no exista.

B.6.2 Proyectos futuros

Esta especificación define un conjunto de elementos y atributos lo suficientemente potentes como para cubrir las necesidades generales de producción de formularios. Sin embargo aún hay sitio para muchas posibles mejoras. Por ejemplo, en el futuro podrían intentar resolverse los siguientes problemas:

- La variedad de tipos de campos de formulario es demasiado limitada en comparación con las modernas interfaces de usuario. Por ejemplo, no hay posibilidad de introducir datos tabulares, deslizadores, o controles de páginas múltiples.
- Los servidores no pueden actualizar los campos de un formulario enviado, y en lugar de eso deben enviar un documento HTML completo, haciendo que la pantalla parpadee.
- Plantean problemas a los navegadores basados por voz, dificultando a las personas con discapacidades visuales la interacción con formularios HTML.

Otra posible extensión sería añadir el atributo [usemap](#) a [INPUT](#) para usar mapas de imágenes en el lado del cliente cuando "[type=image](#)". El elemento [AREA](#) correspondiente al punto pulsado contribuiría al valor pasado al servidor. Para evitar la necesidad de modificar los scripts del servidor, sería apropiado extender [AREA](#) para que proporcionara los valores x e y para su uso con el elemento [INPUT](#).

B.7 Notas sobre scripts

B.7.1 Sintaxis reservada para macros de scripts futuras

Esta especificación reserva sintaxis para el soporte futuro de macros de scripts en atributos CDATA HTML. La intención es permitir que se especifiquen atributos dependiendo de las propiedades de los objetos que aparezcan anteriormente en la página. La sintaxis es:

```
atributo = "... &{ cuerpo de la macro }; ... "
```

Práctica actual en macros de scripts

El cuerpo de la macro se compone de una o más sentencias en el lenguaje de scripts por defecto (como los atributos de eventos intrínsecos). El punto y coma que sigue a la llave derecha es siempre necesario, ya que de otro modo el carácter de llave cerrada "}" se considera como parte del cuerpo de la macro. También vale la pena recordar que las comillas siempre son necesarias para los atributos que contengan macros de scripts.

El procesamiento de los atributos CDATA es el siguiente:

1. El analizador SGML evalúa todas las entidades SGML (p.ej., ">").
2. A continuación el motor de scripts evalúa las macros de scripts.
3. Por último la cadena de caracteres resultante se pasa a la aplicación para su procesamiento subsiguiente.

El procesamiento de macros tiene lugar cuando el documento es cargado (o recargado) pero no vuelve a tener lugar cuando el documento se redimensiona, se redibuja, etc.

EJEMPLO DESAPROBADO:

Aquí tenemos algunos ejemplos en JavaScript. El primero de ellos elige un color de fondo al azar para el documento:

```
<BODY bgcolor='&{randomrgb}';'>
```

Tal vez lo que queramos es oscurecer el fondo por las tardes:

```
<BODY bgcolor='&{if(Date.getHours > 18)...}';'>
```

El siguiente ejemplo usa JavaScript para establecer las coordenadas de un mapa de imágenes en el lado del cliente:

```
<MAP NAME=blabla>
  <AREA shape="rect" coords="&{mirect(uri-imagen)};" href="&{miuri};" alt="">
</MAP>
```

Este ejemplo establece el tamaño de una imagen según las propiedades del documento:

```
<IMG src="barra.gif" width='&{document.banner.width/2};' height='50%' alt="banner">
```

Podemos establecer el URI de un vínculo o imagen con un script:

```
<SCRIPT type="text/javascript">
  function fabricante(herramienta) {
    ...
  }
  function localidad(fabricante) {
    ...
  }
  function logo(fabricante) {
    ...
  }
</SCRIPT>
<A href='&{localidad(fabricante("herramienta"))};'>herramienta</A>
<IMG src='&{logo(fabricante("herramienta"))};' alt="logo">
```

Este último ejemplo muestra cómo los atributos CDATA de SGML pueden entrecomillarse con comillas simples o con comillas dobles. Si se usan comillas simples alrededor de la cadena del atributo entonces pueden incluirse comillas dobles como parte de la cadena del atributo. Otra aproximación es usar " para las comillas dobles:

```
<IMG src="&{logo(fabricante(&quot;herramienta&quot;))};" alt="logo">
```

B.8 Notas sobre marcos

Al no haber garantía de que un nombre de destino de un marco sea único, es apropiado describir la práctica actual para encontrar un marco con un nombre de destino dado:

1. Si el nombre de destino es una palabra reservada de las descritas en el texto normativo, se aplica según se describe.
2. En caso contrario, se realiza una búsqueda en la jerarquía de marcos de la ventana que contenía el vínculo. Se usa el primer marco cuyo nombre concuerde exactamente.
3. Si no se encuentra ningún marco en (2), se aplica el paso 2 a todas las ventanas, desde el primer plano hasta el fondo. Se detiene la búsqueda en cuanto se encuentre un marco con exactamente el mismo nombre.
4. Si no se encuentra ningún marco en (3), se crea una ventana nueva y se le asigna el nombre de destino.

B.9 Notas sobre accesibilidad

La Iniciativa por la Accesibilidad en la Web del W3C (W3C Web Accessibility Initiative, [\[WAI\]](#)) está produciendo una serie de guías para mejorar la accesibilidad a la Web por parte de personas con discapacidades. Hay tres conjuntos de guías:

- Guías para la Accesibilidad de los Contenidos de la Web ([\[WCGL\]](#)), para autores y administradores de sitios. Consulte las Guías para la Accesibilidad del Contenido de la Web para información sobre el suministro de textos alternativos para las imágenes, aplicaciones, scripts, etc.
- Guías para la Accesibilidad de Agentes de Usuario ([\[UAGL\]](#)), para desarrolladores de agentes de usuario (navegadores, reproductores multimedia, tecnologías auxiliares). Consulte estas guías para encontrar pautas sobre el tratamiento de texto alternativo.
- Guías de Accesibilidad para Herramientas de Creación ([\[ATGL\]](#)), para desarrolladores de herramientas de creación.

B.10 Notas sobre seguridad

Los vínculos, las imágenes incluidas, y todos los demás elementos que contengan [URIs](#) como parámetros pueden provocar que el URI sea derreferenciado en respuesta a la entrada directa por el usuario. En este caso, deberían considerarse las cuestiones sobre seguridad de [\[RFC1738\]](#), sección 6. Los métodos ampliamente utilizados para el envío de peticiones de formularios -- HTTP y SMTP -- proporcionan poca garantía de confidencialidad. Los suministradores de información que soliciten información sensible por medio de formularios -- especialmente con el elemento [INPUT](#), [type="password"](#) -- deberían ser conscientes de ello y alertar a sus usuarios de la falta de confidencialidad.

B.10.1 Cuestiones de seguridad relacionadas con los formularios

Un agente de usuario no debería enviar ningún fichero que el usuario no haya pedido explícitamente que se envíe. Así, se espera que los agentes de usuario confirmen cualquier nombre de fichero por defecto que pudiera ser sugerido por el atributo [value](#) del elemento [INPUT](#). Los controles ocultos no deben especificar ficheros.

Esta especificación no contiene ningún mecanismo para el cifrado de los datos; esto debería controlarse con cualquier otro mecanismo que sea apropiado para la transmisión segura de datos.

Una vez que se ha subido un fichero, el agente procesador debería procesarlo y almacenarlo correctamente.

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

Referencias

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Contenidos

1. [Referencias normativas](#)
2. [Referencias informativas](#)

Referencias normativas

[CSS1]

["Cascading Style Sheets, level 1"](#) (Hojas de Estilo en Cascada, Nivel 1), H. W. Lie y B. Bos, 17 de diciembre de 1996. Revisada el 11 de enero de 1999. Este documento se encuentra en <http://www.w3.org/TR/1999/REC-CSS1-19990111>

[DATETIME]

["Date and Time Formats"](#) ("Formatos para Fechas y Horas"), Nota del W3C, M. Wolf y C. Wicksteed, 15 de septiembre de 1997. Revisada el 27 de agosto de 1998. Este documento se encuentra en <http://www.w3.org/TR/1998/NOTE-datetime-19980827>

[HTML40]

["HTML 4.0 Specification"](#) ("Especificación de HTML 4), D. Raggett, A. Le Hors, I. Jacobs. La versión del 24 de abril de 1998 se encuentra en <http://www.w3.org/TR/1998/REC-html40-19980424>. La versión del 24 de abril incluía los cambios editoriales habidos desde la Revisión [original del 18 de diciembre de 1997](#).

[IANA]

["Assigned Numbers"](#) (Números Asignados), STD 2, RFC 1700, USC/ISI, J. Reynolds y J. Postel, octubre de 1994.

[ISO639]

"Codes for the representation of names of languages" ("Códigos para la representación de nombres de idiomas"), ISO 639:1988. Para más información consulte <http://www.iso.ch/cate/d4766.html>. Puede consultar también <http://www.oasis-open.org/cover/iso639a.html>.

[ISO3166]

"Codes for the representation of names of countries" ("Códigos para la representación de nombres de países"), ISO 3166:1993.

[ISO8601]

"Data elements and interchange formats -- Information interchange -- Representation of dates and times" ("Elementos de datos y formatos de intercambio -- Intercambio de información -- Representación de fechas y horas"), ISO 8601:1988.

[ISO8879]

"Information Processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML)" ("Tratamiento de la Información -- Sistemas de Texto y Oficinas -- Lenguaje Generalizado Estándar para el Formato de Documentos"), ISO 8879:1986. Consulte <http://www.iso.ch/cate/d16387.html> para más información sobre el estándar.

[ISO10646]

"Information Technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane" ("Tecnología de la Información -- Juego Universal de Caracteres Codificados con Varios Octetos (UCS) -- Primera Parte: Arquitectura y Plano Multilingüe Básico"), ISO/IEC 10646-1:1993. Esta referencia se refiere a un conjunto de puntos de códigos (*codepoints*) que puede evolucionar a medida que se suman a ellos nuevos caracteres. Esta referencia incluye por lo tanto enmiendas futuras siempre y cuando **no** cambien los caracteres asignados en las primeras cinco enmiendas a la ISO/IEC 10646-1:1993. Asimismo, esta referencia supone que los conjuntos de caracteres definidos por ISO 10646 y Unicode mantienen una equivalencia carácter por carácter. Esta referencia también incluye las publicaciones futuras de otras partes de la 10646 (es decir, partes posteriores a la primera) que definan caracteres en los planos 1-16.

[ISO88591]

"Information Processing -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1" ("Tratamiento de la Información -- Juegos de caracteres gráficos codificados con un solo *byte* de 8 bits -- Primera Parte: Alfabeto Latino Número 1"), ISO 8859-1:1987.

[MIMETYPES]

Lista de tipos de contenido registrados (tipos MIME). Puede bajar una lista de los tipos de contenido registrados de <ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/>.

[RFC1555]

["Hebrew Character Encoding for Internet Messages"](#) ("Codificación de Caracteres Hebreos para Mensajes de Internet"), H. Nussbacher e Y. Bourvine, diciembre de 1993.

[RFC1556]

["Handling of Bi-directional Texts in MIME"](#) ("Tratamiento de Textos Bidireccionales en MIME"), H. Nussbacher, diciembre de 1993.

[RFC1738]

["Uniform Resource Locators"](#) ("Localizadores Uniformes de Recursos"), T. Berners-Lee, L. Masinter, y M. McCahill, diciembre de 1994.

[RFC1766]

["Tags for the Identification of Languages"](#) ("Etiquetas para la Identificación de Idiomas"), H. Alvestrand, marzo de 1995. Se espera que la RFC1766 sea actualizada por <http://www.ietf.org/internet-drafts/draft-alvestrand-lang-tags-v2-00.txt>, trabajo que se está llevando a cabo actualmente.

[RFC1808]

["Relative Uniform Resource Locators"](#) ("Localizadores Relativos Uniformes de Recursos"), R.

Fielding, junio de 1995.

[RFC2045]

["Multipurpose Internet Mail Extensions \(MIME\) Part One: Format of Internet Message Bodies"](#)

(Extensiones Multi-propósito para el Correo por Internet, Primera Parte: Formato de los Cuerpos de los Mensajes de Internet"), N. Freed y N. Borenstein, noviembre de 1996.

Obsérvese que este RFC declara obsoletos a RFC1521, RFC1522, y RFC1590.

[RFC2046]

["Multipurpose Internet Mail Extensions \(MIME\) Part Two: Media Types"](#)

("Extensiones Multi-propósito para el Correo por Internet, Segunda Parte: Tipos de Medios"), N. Freed y N. Borenstein, noviembre de 1996. Obsérvese que este RFC declara obsoletos a RFC1521,

RFC1522, y RFC1590.

[RFC2119]

["Key words for use in RFCs to Indicate Requirement Levels"](#)

("Palabras clave para indicar niveles de exigencia en los RFC"), S. Bradner, marzo de 1997.

[RFC2141]

["URN Syntax"](#) ("Sintaxis de URN"), R. Moats, mayo de 1997.

[RFC2279]

["UTF-8, a transformation format of ISO 10646"](#)

("UTF-8, un formato de transformación de ISO 10646"), F. Yergeau, enero de 1998. Este RFC declara obsoleto al [RFC 2044](#).

[RFC2616]

["Hypertext Transfer Protocol -- HTTP/1.1"](#)

("Protocolo de Transferencia de Hipertexto -- HTTP/1.1"), R. Fielding, J. Gettys, J. Mogul, H. Frystyk Nielsen, L. Masinter, P. Leach y T. Berners-Lee, junio de 1999. Este RFC declara obsoleto al RFC 2068.

[SRGB]

["A Standard Default color Space for the Internet"](#)

("Un espacio estándar de colores por defecto para Internet"), versión 1.10, M. Stokes, M. Anderson, S.Chandrasekar, y R. Motta, 5 de noviembre de 1996. Este documento se encuentra en <http://www.w3.org/Graphics/Color/sRGB>

[UNICODE]

The Unicode Consortium. "The Unicode Standard, Version 3.0" ("El estándar Unicode, Versión 3.0"), Reading, MA, Addison-Wesley Developers Press, 2000. ISBN 0-201-61633-5. Consulte también <http://www.unicode.org/unicode/standard/versions/>.

[URI]

["Uniform Resource Identifiers \(URI\): Generic Syntax"](#)

("Identificadores Uniformes de Recursos: Sintaxis Genérica), T. Berners-Lee, R. Fielding, L. Masinter, agosto de 1998.

Obsérvese que el RFC 2396 actualiza [\[RFC1738\]](#) y [\[RFC1808\]](#).

[WEBSGML]

["Final text of revised TC2 to ISO 8879:1986"](#)

(Texto final de la fe de erratas técnicas número 2 de la ISO 8879:1986"), C. F. Goldfarb, ed., 6 de diciembre de 1998.

Referencias informativas

[ATGL]

"Authoring Tool Accessibility Guidelines" ("Guías de Accesibilidad para Herramientas de Creación"), J. Treviranus, J. Richards, I. Jacobs, C. McCathieNevile, eds. El último Borrador de estas guías para el diseño de herramientas de creación accesibles está disponible en <http://www.w3.org/TR/WAI-AUTOOLS/>

[BRYAN88]

"SGML: An Author's Guide to the Standard Generalized Markup Language" ("Una guía del Lenguaje Generalizado Estándar para el Formato de Documentos"), M. Bryan, Addison-Wesley Publishing Co., 1988.

[CALs]

Continuous Acquisition and Life-Cycle Support (CALs). CALs es una estrategia del Departamento de Estrategia de los Estados Unidos para conseguir la creación, intercambio y uso efectivos de datos digitales para sistemas y equipamientos armamentísticos. Se puede encontrar más información en la [página del CALs](#).

[CHARSETS]

Valores registrados de juegos de caracteres. Puede descargar una lista de los valores registrados de juegos de caracteres de <ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets>.

[CSS2]

["Cascading Style Sheets, level 2"](#) ("Hojas de Estilo en Cascada, Nivel 2"), B. Bos, H. W. Lie, C. Lilley, e I. Jacobs, 12 de mayo 1998. Este documento se encuentra en <http://www.w3.org/TR/1998/REC-CSS2-19980512>

[DCORE]

The Dublin Core. Para más información consulte <http://purl.org/dc>

[ETHNO]

"Ethnologue, Languages of the World", Decimosegunda Edición, Barbara F. Grimes, editora, Summer Institute of Linguistics, octubre de 1992.

[GOLD90]

"The SGML Handbook" ("Guía de referencia del SGML"), C. F. Goldfarb, Clarendon Press, 1991.

[HTML30]

["HyperText Markup Language Specification Version 3.0"](#) ("Especificación del Lenguaje para el Formato de Documentos de Hipertexto, versión 3.0"), D. Raggett, septiembre de 1995. Este documento se encuentra en <http://www.w3.org/MarkUp/html3/CoverPage>

[HTML32]

["HTML 3.2 Reference Specification"](#) ("Especificación de la Referencia HTML 3.2"), D. Raggett, 14 de enero de 1997. Este documento se encuentra en <http://www.w3.org/TR/REC-html32>

[HTML3STYLE]

["HTML and Style Sheets"](#) ("HTML y Hojas de Estilo"), B. Bos, D. Raggett, y H. Lie, 24 de marzo de 1997. Este documento se encuentra en <http://www.w3.org/TR/WD-style-970324>

[LEXHTML]

["A Lexical Analyzer for HTML and Basic SGML"](#) ("Un Analizador Léxico para SGML básico y HTML), D. Connolly, 15 de junio de 1996. Este documento se encuentra en <http://www.w3.org/TR/WD-sgml-lex-960615>

[OASISOPEN]

The Organization for the Advancement of Structured Information Standards (OASIS, la Organización para el Progreso de los Estándares de Información Estructurada): <http://www.oasis-open.org/>.

[PICS]

Platform for Internet Content (PICS, Plataforma para el Contenido de Internet). Para más información consulte <http://www.w3.org/PICS/>

[RDF10]

["Resource Description Framework \(RDF\) Model and Syntax Specification"](#) (Modelo y Especificación Técnica del Marco de Descripción de Recursos), O. Lassila, R. Swick, eds., 22 de febrero de 1999. Este documento se encuentra en <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>

[RFC822]

["Standard for the Format of ARPA Internet Text Messages"](#) ("Estándar para el formato de mensajes de texto de la internet ARPA"), Revisado por David H. Crocker, agosto de 1982.

[RFC850]

["Standard for Interchange of USENET Messages"](#) ("Estándar para el intercambio de mensajes en USENET"), M. Horton, junio de 1983.

[RFC1468]

["Japanese Character Encoding for Internet Messages"](#) ("Codificación de caracteres japoneses para mensajes de Internet"), J. Murai, M. Crispin, y E. van der Poel, junio de 1993.

[RFC1630]

["Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web"](#) ("Identificadores Universales de Recursos en la WWW: Una sintaxis unificada para la expresión de nombres y direcciones de objetos de la Red cuando se usan en la World-Wide Web"), T. Berners-Lee, junio de 1994.

[RFC1866]

["HyperText Markup Language 2.0"](#) ("Lenguaje para el Formato de Documentos de Hipertexto 2.0"), T. Berners-Lee y D. Connolly, noviembre de 1995.

[RFC1942]

["HTML Tables"](#) ("Tablas HTML"), Dave Raggett, mayo de 1996.

[RFC2048]

["Multipurpose Internet Mail Extensions \(MIME\) Part Four: Registration Procedures"](#) ("Extensiones Multi-propósito para el Correo por Internet, Cuarta Parte: Procedimientos de Registro"), N. Freed, J. Klensin, y J. Postel, noviembre de 1996. Obsérvese que este RFC declara obsoletos a RFC1521, RFC1522, y RFC1590.

[RFC2070]

["Internationalization of the HyperText Markup Language"](#) ("Internacionalización del Lenguaje para el Formato de Documentos de Hipertexto"), F. Yergeau, G. Nicol, G. Adams, y M. Dürst, enero de 1997.

[RFC2388]

["Returning Values from Forms: multipart/form-data"](#) ("Retorno de valores de formularios: multipart/form-data"), L. Masinter, agosto de 1998. Consulte también RFC 1867, ["Form-based File Upload in HTML"](#) ("Carga de ficheros basada en formularios HTML"), E. Nebel y L. Masinter, noviembre de 1995.

[SP]

[SP](#) es un analizador SGML de dominio público. Puede encontrar más información en <http://www.jclark.com/sp/index.htm>.

[SQ91]

"The SGML Primer" ("Guía elemental del SGML"), Tercera Edición, SoftQuad Inc., 1991.

[TAKADA]

"Multilingual Information Exchange through the World-Wide Web" ("Intercambio de Información Multilingüe en la World-Wide Web", Toshihiro Takada, Computer Networks and ISDN Systems, Vol. 27, No. 2, pp. 235-241, noviembre de 1994.

[UAGL]

"User Agent Accessibility Guidelines" ("Guías de Accesibilidad para Agentes de Usuario"), J. Gunderson e I. Jacobs, eds. El último borrador de estas guías para el diseño de agentes de usuario accesibles está disponible en <http://www.w3.org/TR/WAI-USERAGENT>.

[WAI]

En el sitio Web de la Iniciativa por la Accesibilidad en la Web (Web Accessibility Initiative, WAI), pueden encontrarse guías para el diseño de documentos HTML accesibles: <http://www.w3.org/WAI/>.

[WCGL]

"Web Content Accessibility Guidelines 1.0" ("Guías de Accesibilidad al Contenido de la Web 1.0"), W. Chisholm, G. Vanderheiden, e I. Jacobs, eds., 5 de mayo de 1999. Este documento se encuentra en <http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505>.

[VANH90]

"Practical SGML" ("SGML Práctico"), E. van Herwijnen, Kluwer Academic Publishers Group, Norwell and Dordrecht, 1990.

[XHTML]

["XHTML\[tm\] 1.0: The Extensible HyperText Markup Language"](#) ("XHTML[tm] 1.0: El Lenguaje Extensible para el Formato de Documentos de Hipertexto"), S. Pemberton et al. La última versión de esta especificación está disponible en <http://www.w3.org/TR/xhtml1>. En la fecha de publicación del presente documento, XHTML 1.0 es una Recomendación Propuesta por el W3C.

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [atributos](#) [índice](#)

Índice de elementos

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Legenda: Opcional, Prohibido, Vacío, Desaprobado, Loose DTD (DTD Transicional), Frameset DTD (DTD de Documentos con Marcos)

Nombre	Etiqueta inicial	Etiqueta final	Vacío	Desapr.	DTD	Descripción
A						origen o destino de vínculo
ABBR						abreviatura (p.ej., WWW, HTTP, etc.)
ACRONYM						
ADDRESS						información sobre el autor
APPLET				D	L	applet Java
AREA		P	V			área de un mapa de imágenes en el lado del cliente
B						estilo de texto en negrita
BASE		P	V			URI base del documento
BASEFONT		P	V	D	L	tamaño base de fuente
BDO						anular algoritmo BiDi I18N
BIG						estilo de texto grande
BLOCKQUOTE						cita larga
BODY	O	O				cuerpo del documento
BR		P	V			salto de línea forzado
BUTTON						botón
CAPTION						título de tabla
CENTER				D	L	forma abreviada de DIV align=center
CITE						cita
CODE						fragmento de código de computadora
COL		P	V			columna de una tabla

COLGROUP		O				grupo de columnas de una tabla
DD		O				descripción de una definición
DEL						texto borrado
DFN						definición
DIR				D	L	lista tipo directorio
DIV						contenedor genérico de idioma/estilo
DL						lista de definiciones
DT		O				término definido
EM						énfasis
FIELDSET						grupo de controles de un formulario
FONT				D	L	cambio local de la fuente
FORM						formulario interactivo
FRAME		P	V		F	subventana
FRAMESET					F	subdivisión en ventanas
H1						encabezado
H2						encabezado
H3						encabezado
H4						encabezado
H5						encabezado
H6						encabezado
HEAD	O	O				cabecera del documento
HR		P	V			separador horizontal
HTML	O	O				elemento raíz del documento
I						estilo de texto en itálica
IFRAME					L	subventana en línea
IMG		P	V			imagen incluida
INPUT		P	V			control de formulario
INS						texto insertado
ISINDEX		P	V	D	L	entrada de texto en una sola línea con indicador
KBD						texto que debe introducir el usuario
LABEL						texto del rótulo de un campo de formulario
LEGEND						leyenda de un grupo de campos

LI		O				objeto de lista
LINK		P	V			un vínculo independiente del medio
MAP						mapa de imágenes en el lado del cliente
MENU				D	L	lista tipo menú
META		P	V			metainformación genérica
NOFRAMES					F	contenedor de contenido alternativo para la representación no basada en marcos
NOSCRIPT						contenedor de contenido alternativo para la representación no basada en scripts
OBJECT						objeto incluido genérico
OL						lista ordenada
OPTGROUP						grupo de opciones
OPTION		O				opción seleccionable
P		O				párrafo
PARAM		P	V			valor de propiedad con nombre
PRE						texto preformateado
Q						cita corta en línea
S				D	L	estilo de texto tachado
SAMP						ejemplo de salida de programas, scripts, etc.
SCRIPT						sentencias de script
SELECT						selector de opciones
SMALL						estilo de texto pequeño
SPAN						contenedor genérico de idioma/estilo
STRIKE				D	L	estilo de texto tachado
STRONG						énfasis fuerte
STYLE						información de estilo
SUB						subíndice
SUP						superíndice
TABLE						
TBODY	O	O				cuerpo de tabla
TD		O				celda de datos de una tabla

TEXTAREA						campo de texto multilínea
TFOOT		O				pie de tabla
TH		O				celda de encabezado de tabla
THEAD		O				cabecera de tabla
TITLE						título del documento
TR		O				fila de una tabla
TT						estilo de texto de teletipo o monoespacio
U				D	L	estilo de texto subrayado
UL						lista no ordenada
VAR						variable o argumento de un programa

[anterior](#) [siguiente](#) [contenidos](#) [atributos](#) [índice](#)

Índice de Atributos

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en jrpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

Leyenda: Desaprobado, Loose DTD (DTD Transicional), Frameset DTD (DTD de Documentos con Marcos)

Nombre	Elementos Relacionados	Tipo	Por Defecto	Desapr.	DTD	Comentario
abbr	TD , TH	%Text;	#IMPLIED			abreviatura de celda de cabecera
accept-charset	FORM	%Charsets;	#IMPLIED			lista de codificaciones de caracteres soportadas
accept	FORM , INPUT	%ContentTypes;	#IMPLIED			lista de tipos MIME para subir ficheros
accesskey	A , AREA , BUTTON , INPUT , LABEL , LEGEND , TEXTAREA	%Character;	#IMPLIED			carácter de la tecla de accesibilidad
action	FORM	%URI;	#REQUIRED			procesador de formulario en servidor
align	CAPTION	%CAlign;	#IMPLIED	D	L	título de una tabla
align	APPLET , IFRAME , IMG , INPUT , OBJECT	%IAAlign;	#IMPLIED	D	L	alineación vertical u horizontal
align	LEGEND	%LAlign;	#IMPLIED	D	L	leyenda de un grupo de campos (fieldset)
align	TABLE	%TAlign;	#IMPLIED	D	L	posición de la tabla respecto a la ventana
align	HR	(left center right)	#IMPLIED	D	L	
align	DIV , H1 , H2 , H3 , H4 , H5 , H6 , P	(left center right justify)	#IMPLIED	D	L	alineación de texto

align	COL , COLGROUP , TBODY , TD , TFOOT , TH , THEAD , TR	(left center right justify char)	#IMPLIED			
alink	BODY	%Color;	#IMPLIED	D	L	color de vínculos seleccionados
alt	APPLET	%Text;	#IMPLIED	D	L	descripción corta
alt	AREA , IMG	%Text;	#REQUIRED			descripción corta
alt	INPUT	CDATA	#IMPLIED			descripción corta
archive	APPLET	CDATA	#IMPLIED	D	L	lista de archivos separados por comas
archive	OBJECT	CDATA	#IMPLIED			lista de URIs separados por espacios
axis	TD , TH	CDATA	#IMPLIED			lista de cabeceras relacionadas separadas por comas
background	BODY	%URI;	#IMPLIED	D	L	fichero de textura de fondo del documento
bgcolor	TABLE	%Color;	#IMPLIED	D	L	color de fondo de las celdas
bgcolor	TR	%Color;	#IMPLIED	D	L	color de fondo de una fila
bgcolor	TD , TH	%Color;	#IMPLIED	D	L	color de fondo de una celda
bgcolor	BODY	%Color;	#IMPLIED	D	L	color de fondo del documento
border	TABLE	%Pixels;	#IMPLIED			controla la anchura del marco que rodea una tabla
border	IMG , OBJECT	%Pixels;	#IMPLIED	D	L	anchura del borde de un vínculo
cellpadding	TABLE	%Length;	#IMPLIED			espacio dentro de celdas
cellspacing	TABLE	%Length;	#IMPLIED			espacio entre celdas
char	COL , COLGROUP , TBODY , TD , TFOOT , TH , THEAD , TR	%Character;	#IMPLIED			carácter de alineación, p.ej. char=':'

charoff	COL , COLGROUP , TBODY , TD , TFOOT , TH , THEAD , TR	%Length;	#IMPLIED			offset para carácter de alineación
charset	A , LINK , SCRIPT	%Charset;	#IMPLIED			codificación de caracteres del recurso vinculado
checked	INPUT	(checked)	#IMPLIED			para radiobotones y casillas de verificación
cite	BLOCKQUOTE , Q	%URI;	#IMPLIED			URI del documento o mensaje fuente
cite	DEL , INS	%URI;	#IMPLIED			información sobre la razón del cambio
class	Todos los elementos excepto BASE , BASEFONT , HEAD , HTML , META , PARAM , SCRIPT , STYLE , TITLE	CDATA	#IMPLIED			lista de clases separadas por espacios
classid	OBJECT	%URI;	#IMPLIED			identifica una implementación
clear	BR	(left all right none)	none	D	L	control del flujo de texto
code	APPLET	CDATA	#IMPLIED	D	L	fichero de clase applet
codebase	OBJECT	%URI;	#IMPLIED			URI base para classid, data, archive
codebase	APPLET	%URI;	#IMPLIED	D	L	URI base opcional para applet
codetype	OBJECT	%ContentType;	#IMPLIED			tipo de contenido para code
color	BASEFONT , FONT	%Color;	#IMPLIED	D	L	color del texto
cols	FRAMESET	%MultiLengths;	#IMPLIED		F	lista de longitudes, por defecto: 100% (1 col)
cols	TEXTAREA	NUMBER	#REQUIRED			
colspan	TD , TH	NUMBER	1			número de columnas abarcado por celda

compact	DIR , DL , MENU , OL , UL	(compact)	#IMPLIED	D	L	espacio reducido entre objetos
content	META	CDATA	#REQUIRED			información asociada
coords	AREA	%Coords;	#IMPLIED			lista de longitudes separadas por coma
coords	A	%Coords;	#IMPLIED			para usar con mapas de imágenes en el cliente
data	OBJECT	%URI;	#IMPLIED			referencia a datos del objeto
datetime	DEL , INS	%Datetime;	#IMPLIED			fecha y hora del cambio
declare	OBJECT	(declare)	#IMPLIED			declara el objeto pero no lo crea
defer	SCRIPT	(defer)	#IMPLIED			El AU puede retrasar la ejecución del script
dir	Todos los elementos excepto APPLET , BASE , BASEFONT , BDO , BR , FRAME , FRAMESET , IFRAME , PARAM , SCRIPT	(ltr rtl)	#IMPLIED			dirección de texto débil/neutral
dir	BDO	(ltr rtl)	#REQUIRED			direccionalidad
disabled	BUTTON , INPUT , OPTGROUP , OPTION , SELECT , TEXTAREA	(disabled)	#IMPLIED			no disponible en este contexto
enctype	FORM	%ContentType;	"application/x-www-form-urlencoded"			
face	BASEFONT , FONT	CDATA	#IMPLIED	D	L	lista de nombres de fuentes separados por coma
for	LABEL	IDREF	#IMPLIED			empareja según valor de campo ID
frame	TABLE	%TFrame;	#IMPLIED			qué partes del marco representar
frameborder	FRAME , IFRAME	(1 0)	1		F	¿pintar bordes del marco?

headers	TD , TH	IDREFS	#IMPLIED			lista de id's de celdas de encabezado
height	IFRAME	%Length;	#IMPLIED		L	altura del marco
height	TD , TH	%Length;	#IMPLIED	D	L	altura de una celda
height	IMG , OBJECT	%Length;	#IMPLIED			nueva altura
height	APPLET	%Length;	#REQUIRED	D	L	altura inicial
href	A , AREA , LINK	%URI;	#IMPLIED			URI del recurso vinculado
href	BASE	%URI;	#IMPLIED			URI que actúa como URI base
hreflang	A , LINK	%LanguageCode;	#IMPLIED			código de idioma
hspace	APPLET , IMG , OBJECT	%Pixels;	#IMPLIED	D	L	espacio de relleno horizontal
http-equiv	META	NAME	#IMPLIED			nombre de encabezado HTTP de respuesta
id	Todos los elementos excepto BASE , HEAD , HTML , META , SCRIPT , STYLE , TITLE	ID	#IMPLIED			id único en todo el documento
ismap	IMG , INPUT	(ismap)	#IMPLIED			usar mapa de imágenes en servidor
label	OPTION	%Text;	#IMPLIED			para usar en menús jerárquicos
label	OPTGROUP	%Text;	#REQUIRED			para usar en menús jerárquicos
lang	Todos los elementos excepto APPLET , BASE , BASEFONT , BR , FRAME , FRAMESET , IFRAME , PARAM , SCRIPT	%LanguageCode;	#IMPLIED			código de idioma
language	SCRIPT	CDATA	#IMPLIED	D	L	nombre del lenguaje predefinido de scripts
link	BODY	%Color;	#IMPLIED	D	L	color de los vínculos

longdesc	IMG	%URI;	#IMPLIED			vínculo a descripción larga (complementa a alt)
longdesc	FRAME, IFRAME	%URI;	#IMPLIED		F	vínculo a descripción larga (complementa a title)
marginheight	FRAME, IFRAME	%Pixels;	#IMPLIED		F	altura del margen en píxeles
marginwidth	FRAME, IFRAME	%Pixels;	#IMPLIED		F	anchura del margen en píxeles
maxlength	INPUT	NUMBER	#IMPLIED			máximo de caracteres para campos de texto
media	STYLE	%MediaDesc;	#IMPLIED			diseñado para usar con estos medios
media	LINK	%MediaDesc;	#IMPLIED			para representar en estos medios
method	FORM	(GET POST)	GET			método HTTP usado para enviar el formulario
multiple	SELECT	(multiple)	#IMPLIED			por defecto es selección simple
name	BUTTON, TEXTAREA	CDATA	#IMPLIED			
name	APPLET	CDATA	#IMPLIED	D	L	permite a los applets encontrarse entre sí
name	SELECT	CDATA	#IMPLIED			nombre del campo
name	FORM	CDATA	#IMPLIED			nombre del formulario, para los scripts
name	FRAME, IFRAME	CDATA	#IMPLIED		F	nombre del marco, para designarlo como destino
name	IMG	CDATA	#IMPLIED			nombre de la imagen, para los scripts
name	A	CDATA	#IMPLIED			vínculo destino con nombre
name	INPUT, OBJECT	CDATA	#IMPLIED			enviar como parte del formulario
name	MAP	CDATA	#REQUIRED			para su referencia por usemap
name	PARAM	CDATA	#REQUIRED			nombre de propiedad
name	META	NAME	#IMPLIED			nombre de metainformación

nohref	AREA	(nohref)	#IMPLIED			esta región no tiene acción
noresize	FRAME	(noresize)	#IMPLIED		F	¿permitir a los usuarios redimensionar marcos?
noshade	HR	(noshade)	#IMPLIED	D	L	
nowrap	TD , TH	(nowrap)	#IMPLIED	D	L	suprimir ajuste automático de líneas
object	APPLET	CDATA	#IMPLIED	D	L	fichero applet serializado
onblur	A , AREA , BUTTON , INPUT , LABEL , SELECT , TEXTAREA	%Script;	#IMPLIED			el elemento perdió el foco
onchange	INPUT , SELECT , TEXTAREA	%Script;	#IMPLIED			el valor del elemento fue modificado
onclick	Todos los elementos excepto APPLET , BASE , BASEFONT , BDO , BR , FONT , FRAME , FRAMESET , HEAD , HTML , IFRAME , ISINDEX , META , PARAM , SCRIPT , STYLE , TITLE	%Script;	#IMPLIED			se hizo clic con un botón del apuntador
ondblclick	Todos los elementos excepto APPLET , BASE , BASEFONT , BDO , BR , FONT , FRAME , FRAMESET , HEAD , HTML , IFRAME , ISINDEX , META ,	%Script;	#IMPLIED			se hizo doble clic con un botón del apuntador

	PARAM , SCRIPT , STYLE , TITLE				
onfocus	A , AREA , BUTTON , INPUT , LABEL , SELECT , TEXTAREA	%Script;	#IMPLIED		el foco se dirigió hacia el elemento
onkeydown	Todos los elementos excepto APPLET , BASE , BASEFONT , BDO , BR , FONT , FRAME , FRAMESET , HEAD , HTML , IFRAME , ISINDEX , META , PARAM , SCRIPT , STYLE , TITLE	%Script;	#IMPLIED		se pulsó una tecla
onkeypress	Todos los elementos excepto APPLET , BASE , BASEFONT , BDO , BR , FONT , FRAME , FRAMESET , HEAD , HTML , IFRAME , ISINDEX , META , PARAM , SCRIPT , STYLE , TITLE	%Script;	#IMPLIED		una tecla fue pulsada y soltada

onkeyup	<p>Todos los elementos excepto APPLET, BASE, BASEFONT, BDO, BR, FONT, FRAME, FRAMESET, HEAD, HTML, IFRAME, ISINDEX, META, PARAM, SCRIPT, STYLE, TITLE</p>	%Script;	#IMPLIED			una tecla fue soltada
onload	FRAMESET	%Script;	#IMPLIED		F	todos los marcos fueron cargados
onload	BODY	%Script;	#IMPLIED			el documento fue cargado
onmousedown	<p>Todos los elementos excepto APPLET, BASE, BASEFONT, BDO, BR, FONT, FRAME, FRAMESET, HEAD, HTML, IFRAME, ISINDEX, META, PARAM, SCRIPT, STYLE, TITLE</p>	%Script;	#IMPLIED			se pulsó un botón del apuntador
onmousemove	<p>Todos los elementos excepto APPLET, BASE, BASEFONT, BDO, BR, FONT, FRAME, FRAMESET, HEAD, HTML, IFRAME, ISINDEX, META, PARAM,</p>	%Script;	#IMPLIED			un apuntador se movió al interior del elemento

	SCRIPT , STYLE , TITLE					
onmouseout	Todos los elementos except APPLET , BASE , BASEFONT , BDO , BR , FONT , FRAME , FRAMESET , HEAD , HTML , IFRAME , ISINDEX , META , PARAM , SCRIPT , STYLE , TITLE	%Script;	#IMPLIED			un apuntador se quitó de encima del elemento
onmouseover	Todos los elementos excepto APPLET , BASE , BASEFONT , BDO , BR , FONT , FRAME , FRAMESET , HEAD , HTML , IFRAME , ISINDEX , META , PARAM , SCRIPT , STYLE , TITLE	%Script;	#IMPLIED			un apuntador se movió encima del elemento
onmouseup	Todos los elementos excepto APPLET , BASE , BASEFONT , BDO , BR , FONT , FRAME , FRAMESET , HEAD , HTML , IFRAME , ISINDEX , META , PARAM , SCRIPT , STYLE , TITLE	%Script;	#IMPLIED			se soltó un botón de un apuntador

onreset	FORM	%Script;	#IMPLIED			el formulario fue reinicializado
onselect	INPUT , TEXTAREA	%Script;	#IMPLIED			se seleccionó parte de un texto
onsubmit	FORM	%Script;	#IMPLIED			el formulario fue enviado
onunload	FRAMESET	%Script;	#IMPLIED		F	se quitaron todos los marcos
onunload	BODY	%Script;	#IMPLIED			el documento ha sido quitado
profile	HEAD	%URI;	#IMPLIED			diccionario con nombres de metainformación
prompt	ISINDEX	%Text;	#IMPLIED	D	L	mensaje indicador
readonly	TEXTAREA	(readonly)	#IMPLIED			
readonly	INPUT	(readonly)	#IMPLIED			para texto y contraseñas
rel	A , LINK	%LinkTypes;	#IMPLIED			tipos de vínculos directos
rev	A , LINK	%LinkTypes;	#IMPLIED			tipos de vínculos inversos
rows	FRAMESET	%MultiLengths;	#IMPLIED		F	lista de longitudes, por defecto: 100% (1 fila)
rows	TEXTAREA	NUMBER	#REQUIRED			
rowspan	TD , TH	NUMBER	1			número de filas abarcado por la celda
rules	TABLE	%TRules;	#IMPLIED			líneas de división entre filas y columnas
scheme	META	CDATA	#IMPLIED			seleccionar forma de contenido
scope	TD , TH	%Scope;	#IMPLIED			campo de aplicación de una celda de cabecera
scrolling	FRAME , IFRAME	(yes no auto)	auto		F	barra de desplazamiento o no
selected	OPTION	(selected)	#IMPLIED			
shape	AREA	%Shape;	rect			controla la interpretación de las coordenadas
shape	A	%Shape;	rect			para usar con mapas de imágenes en el cliente
size	HR	%Pixels;	#IMPLIED	D	L	

size	FONT	CDATA	#IMPLIED	D	L	[+ -]nn p.ej. size="+1", size="4"
size	INPUT	CDATA	#IMPLIED			específico de cada tipo de campo
size	BASEFONT	CDATA	#REQUIRED	D	L	tamaño de fuente base para elementos FONT
size	SELECT	NUMBER	#IMPLIED			filas visibles
span	COL	NUMBER	1			los atributos de COL afectan a N columnas
span	COLGROUP	NUMBER	1			número de columnas por defecto en el grupo
src	SCRIPT	%URI;	#IMPLIED			URI del script externo
src	INPUT	%URI;	#IMPLIED			para campos con imágenes
src	FRAME , IFRAME	%URI;	#IMPLIED		F	fuentes del contenido del marco
src	IMG	%URI;	#REQUIRED			URI de la imagen a incluir
standby	OBJECT	%Text;	#IMPLIED			mensaje a mostrar mientras se carga
start	OL	NUMBER	#IMPLIED	D	L	número inicial de la secuencia
style	Todos los elementos excepto BASE , BASEFONT , HEAD , HTML , META , PARAM , SCRIPT , STYLE , TITLE	%StyleSheet;	#IMPLIED			información de estilo asociada
summary	TABLE	%Text;	#IMPLIED			propósito/estructura para salida por voz
tabindex	A , AREA , BUTTON , INPUT , OBJECT , SELECT , TEXTAREA	NUMBER	#IMPLIED			posición en el orden de tabulación
target	A , AREA , BASE , FORM , LINK	%FrameTarget;	#IMPLIED		L	representar en este marco

text	BODY	%Color;	#IMPLIED	D	L	color del texto del documento
title	Todos los elementos excepto BASE , BASEFONT , HEAD , HTML , META , PARAM , SCRIPT , TITLE	%Text;	#IMPLIED			título consultivo
type	A , LINK	%ContentType;	#IMPLIED			tipo de contenido consultivo
type	OBJECT	%ContentType;	#IMPLIED			tipo de contenido para los datos
type	PARAM	%ContentType;	#IMPLIED			tipo de contenido para el valor cuando <code>valuetype=ref</code>
type	SCRIPT	%ContentType;	#REQUIRED			tipo de contenido para lenguaje de scripts
type	STYLE	%ContentType;	#REQUIRED			tipo de contenido para lenguaje de estilos
type	INPUT	%InputType;	TEXT			qué tipo de control hace falta
type	LI	%LIStyle;	#IMPLIED	D	L	estilo de objeto de lista
type	OL	%OLStyle;	#IMPLIED	D	L	estilo de numeración
type	UL	%ULStyle;	#IMPLIED	D	L	estilo de gráfico de lista
type	BUTTON	(button submit reset)	submit			para usar como botón de formulario
usemap	IMG , INPUT , OBJECT	%URI;	#IMPLIED			usar mapa de imágenes en el cliente
valign	COL , COLGROUP , TBODY , TD , TFOOT , TH , THEAD , TR	(top middle bottom baseline)	#IMPLIED			alineación vertical en celdas
value	INPUT	CDATA	#IMPLIED			especificar para radiobotones y casillas de verificación

value	OPTION	CDATA	#IMPLIED			por defecto el contenido del elemento
value	PARAM	CDATA	#IMPLIED			valor de propiedad
value	BUTTON	CDATA	#IMPLIED			se manda al servidor cuando se envía
value	LI	NUMBER	#IMPLIED	D	L	reinicializar número de secuencia
valuetype	PARAM	(DATA REF OBJECT)	DATA			cómo interpretar el valor
version	HTML	CDATA	%HTML.Version;	D	L	constante
vlink	BODY	%Color;	#IMPLIED	D	L	color de los vínculos visitados
vspace	APPLET , IMG , OBJECT	%Pixels;	#IMPLIED	D	L	espacio vertical
width	HR	%Length;	#IMPLIED	D	L	
width	IFRAME	%Length;	#IMPLIED		L	anchura del marco
width	IMG , OBJECT	%Length;	#IMPLIED			nueva anchura
width	TABLE	%Length;	#IMPLIED			anchura de la tabla
width	TD , TH	%Length;	#IMPLIED	D	L	anchura de la celda
width	APPLET	%Length;	#REQUIRED	D	L	anchura inicial
width	COL	%MultiLength;	#IMPLIED			especificación de la anchura de la columna
width	COLGROUP	%MultiLength;	#IMPLIED			anchura por defecto de los COLs contenidos
width	PRE	NUMBER	#IMPLIED	D	L	

[anterior](#) [siguiente](#) [contenidos](#) [elementos](#) [índice](#)

Índice

Nota: Este documento es parte de una traducción al castellano de la Recomendación del W3C "HTML 4.01 Specification" ([más información](#)). Puede consultar la [versión original](#) del mismo. Para cualquier comentario o corrección acerca de la traducción póngase en contacto con el traductor en irpozo@conclase.net. Gracias por su colaboración.

Véase el [Aviso de copyright](#) de la traducción.

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#)

abreviaturas y acrónimos [1](#)

accesibilidad

características de HTML 4 [1](#)

contenido alternativo de un objeto [1](#)

de mapas de imágenes [1](#), [2](#)

descripción larga de una imagen [1](#)

teclas de acceso [1](#)

texto alternativo [1](#)

y contenido alternativo de marcos [1](#)

y descripciones largas de marcos [1](#)

y hojas de estilo [1](#)

agente de usuario [1](#)

conforme [1](#)

proceso de datos de scripts y estilos [1](#)

tratamiento de mapas de imágenes [1](#)

y condiciones de error [1](#), [2](#)

y datos de estilo [1](#)

y datos de scripts [1](#)

ajuste automático de líneas [1](#)

algoritmo bidireccional de Unicode [1](#)

alineación

de elementos en bloque [1](#)

de imágenes [1](#)

de los contenidos de una tabla [1](#)

de objetos [1](#)

objetos flotantes [1](#)

texto flotante [1](#)

aplicaciones y applets

formas de incluirlas [1](#)

application/x-www-form-urlencoded [1](#), [2](#)

atributo [1](#)

booleano [1](#), [2](#)

booleano minimizado [1](#)

comillas alrededor del valor [1](#)

declaración en el DTD [1](#)

distinción entre mayúsculas y minúsculas [1](#)

mayúsculas y minúsculas en los valores [1](#)

valor #FIXED [1](#)

valor #IMPLIED [1](#)

valor #REQUIRED [1](#)

atributo class

funciones [1](#)

atributo id

funciones [1](#)

mismo espacio de nombres que name [1](#)

atributo lang

aplicabilidad [1](#)

no para dirección [1](#)

%attrs; [1](#)

autor [1](#)

bidirección

algoritmo Unicode [1](#)

anulación [1](#)

y codificación de caracteres [1](#)

y hojas de estilo [1](#)

%block; [1](#)

bloques

elementos [1](#)

y bidirección [1](#), [2](#)

BODY

no presente en documentos con marcos [1](#)

borde

alrededor de un marco [1](#)

alrededor de un objeto [1](#)

alrededor de una imagen [1](#)

alrededor de una tabla [1](#)

botón [1](#)

botón de envío [1](#)

botón de reinicialización [1](#)

campo de acción

de una celda de encabezado de una tabla [1](#)

caracteres

abstractos [1](#)

esfuerzos para representarlos [1](#)

representación de los no representables [1](#)

tecla de acceso [1](#)

tratamiento de los no representables [1](#)

casilla de verificación [1](#)

catálogo para HTML [1](#)

CDATA [1](#), [2](#)

datos de scripts y estilos [1](#)

celda de datos

en una tabla [1](#)

celda de encabezado

abreviatura [1](#)

campo de acción [1](#)

en una tabla [1](#)

CERN [1](#)

%Character; [1](#)

%Charset; [1](#)

citas [1](#)

representación [1](#)

codificación de caracteres [1](#), [2](#)

de vínculos [1](#)

determinación por agentes de usuario [1](#)

ejemplos comunes [1](#)

elección [1](#)

especificación [1](#)

nombre [1](#)

para enviar formularios [1](#)

por defecto [1](#)

UTF-1 [1](#)

UTF-16 [1](#)

y bidirección [1](#)

código (markup) [1](#)

color

de fondo [1](#)

nombre [1](#)

%Color; [1](#)

columna

anchura en una tabla [1](#)

numero de columnas en una tabla [1](#)

comentarios

en DTD [1](#)

en HTML [1](#)

no representados [1](#)

referencias de caracteres en comentarios [1](#)

sólo informativos [1](#)

usados para ocultar datos de hojas de estilo [1](#)

usados para ocultar datos de scripts [1](#)

conformidad [1](#)

conjunto de caracteres del documento [1](#)

equivalencia entre ISO10646 y UNICODE [1](#)

ISO10646 [1](#), [2](#)

conjunto de datos del formulario [1](#)

codificación [1](#)

conjunto universal de caracteres [1](#)

conjuntos de entidades

URIs para HTML 4.01 [1](#)

%ContentType; [1](#)

control [1](#)

con éxito [1](#)

de sólo lectura [1](#)

dirigiendo el foco hacia un control [1](#)

eventos [1](#)

navegación con tabulador [1](#)

nombre del control [1](#)

tecla de acceso [1](#)

tipos [1](#)

valor del control [1](#)

valor inicial [1](#)

control de entrada de contraseña [1](#)

control de entrada de texto [1](#)

de una sola línea [1](#)

multilínea [1](#)

control oculto [1](#), [2](#)

control tipo objeto [1](#), [2](#)

control de selección de ficheros [1](#)

envío [1](#)

controles deshabilitados [1](#)

sin éxito [1](#)

coordenadas

de mapas de imágenes en el lado del servidor [1](#)

del punto pulsado al enviar un formulario [1](#)

%Datetime; [1](#)

declaración del tipo de documento [1](#)

para el DTD de documentos con marcos [1](#)

para el DTD estricto [1](#)

para el DTD transicional [1](#)

definición de entidades paramétricas [1](#)

definición del tipo de documento [1](#)

comentarios [1](#)

cómo leerla [1](#)

de documentos con marcos (frameset) [1](#)

estricta (strict) [1](#)

los ejemplos son conformes [1](#)

los fragmentos de DTD son conformes [1](#)

transicional (transitional) [1](#)

desaprobado [1](#)

elementos [1](#)

descripción larga de una imagen

relación con el texto alt [1](#)

descriptor de medios

análisis [1](#)

lista de descriptores reconocidos [1](#)

mayúsculas y minúsculas [1](#)

dirección

de la información en tablas [1](#)

del texto [1](#)

herencia entre elementos anidados [1](#)

documento

formas de incluir [1](#), [2](#), [3](#)

modificación dinámica con scripts [1](#)

validación SGML [1](#)

documento HTML [1](#)

documentos con marcos [1](#)

anidamiento [1](#)

compartición de datos entre documentos [1](#)

contenido alternativo [1](#)

DTD, declaración [1](#)

DTD, definición [1](#)

especificación de la distribución de los marcos [1](#)

problemas de navegación [1](#)

utilización de NOFRAMES [1](#)

DTD estricto

declaración [1](#)

definición [1](#)

DTD transicional

declaración [1](#)

definición [1](#)

Dublin Core [1](#)

elemento

declaración del tipo [1](#), [2](#)

distinción mayúsculas-minúsculas [1](#)

en bloque [1](#)

en línea [1](#)

- etiqueta final [1](#)
- etiqueta final omitida [1](#)
- etiqueta inicial [1](#)
- etiqueta inicial omitida [1](#)
- identificador único [1](#)
- lista de los desaprobados [1](#)
- lista de los obsoletos [1](#)
- modelo de contenido [1](#)
- referencias desde scripts [1](#)
- soporte de los desaprobados [1](#)
- soporte de los obsoletos [1](#)
- tipos [1](#)
- vacío [1](#), [2](#)

en línea

- elemento [1](#)

encabezado Content-Language [1](#)

encabezado Content-Script-Type [1](#)

encabezado Content-Style-Type [1](#)

encabezado Content-Type [1](#)

encabezados

- correctamente anidados [1](#)

encabezados numerados

- numeración [1](#)

entidad de mensaje [1](#)

entidad paramétrica

- %Character; [1](#)

- %Charset; [1](#)

- %Color; [1](#)

- %ContentType; [1](#)

- %Datetime; [1](#)

- %FrameTarget; [1](#)

- %LanguageCode; [1](#)

- %Length; [1](#)

- %MediaDesc; [1](#)

- %MultiLength; [1](#)

- %Pixels; [1](#)

[%Script; 1](#)

[%Text; 1](#)

[%URI; 1](#)

[%attrs; 1](#)

[%block; 1](#)

[%inline; 1](#)

error

[mapa de imágenes con IMG en BUTTON 1](#)

[recursos no disponibles 1](#)

[representación de las reglas de estilo de STYLE 1](#)

[tratamiento por agentes de usuarios 1, 2](#)

espacio en blanco 1

[alrededor de imágenes y objetos 1](#)

[alrededor de los contenidos de una tabla 1](#)

[alrededor de un marco 1](#)

[carácter 1](#)

[colapsado 1](#)

[conservado en PRE 1](#)

espacio entre palabras 1

esquema 1

etiqueta inicial 1

[omitida 1](#)

etiqueta final 1

[declarada como opcional 1](#)

[omitida 1](#)

eventos

[eventos intrínsecos 1](#)

fecha

[formato 1](#)

[del texto insertado y borrado 1](#)

fila

[número de filas en una tabla 1](#)

foco 1

[el rótulo lo dirige hacia el control 1](#)

[y tecla de acceso 1](#)

%FrameTarget; 1

fuelle

estilo con HTML [1](#)

formulario

añadiendo r6tulos [1](#)

codificaci6n de los datos [1](#)

controles [1](#)

envío [1](#)

estructuraci6n de los controles [1](#)

m6todo de envío [1](#)

m6todos y acciones [1](#)

navegaci6n por los controles [1](#)

notas de presentaci6n [1](#)

orden de tabulaci6n de los controles [1](#)

procesamiento de los controles [1](#)

reinicializaci6n [1](#)

tipos de contenido para codificar [1](#)

tipos de controles [1](#)

valores enviados [1](#)

GET

y envío de formularios [1](#)

Grupo de Trabajo HTML

miembros [1](#)

grupo de columnas [1](#)

grupo de filas [1](#)

gui6n de separaci6n [1](#)

herramienta de creaci6n [1](#), [2](#)

y lenguaje de hojas de estilo por defecto [1](#)

hoja de estilo

alternativa [1](#)

comentarios para ocultarla [1](#)

datos [1](#)

en cascada [1](#)

especificaci6n de una hoja externa [1](#)

especificaci6n de una hoja preferente [1](#)

externa [1](#)

externa a trav6s de v6nculos [1](#)

- [introducción](#) 1
- [medios destino](#) 1
- [persistente](#) 1
- [preferente](#) 1
- [reglas en HEAD](#) 1
- [reglas en línea](#) 1
- [usada con DIV y SPAN](#) 1
- [y bidirección](#) 1

hora

- [formato](#) 1

HTML

- [comentarios](#) 1
- [como aplicación SGML](#) 1
- [consejos para autores](#) 1
- [desarrollo](#) 1
- [especificación de datos no HTML](#) 1
- [versión 2.0](#) 1
- [versión 3.0](#) 1
- [versión 3.2](#) 1
- [versión HTML+](#) 1

HTTP

- [encabezado Content-Language](#) 1
- [encabezado Content-Script-Type](#) 1
- [encabezado Content-Style-Type](#) 1
- [encabezado Content-Type](#) 1
- [encabezado Default-Style](#) 1
- [GET y POST con formularios](#) 1
- [usado para vincular hojas de estilo externas](#) 1

ID 1

identificador de fragmento [1](#), [2](#)

Identificador Universal de Recursos (ver URI) 1

idioma

- [códigos para especificarlo](#) 1
- [de un recurso vinculado](#) 1
- [del texto](#) 1

IDREF 1

[IDREFS 1](#)

imagen

[alineación 1](#)

[anchura y altura 1](#)

[borde alrededor 1](#)

[descripción larga 1](#)

[espacio en blanco alrededor 1](#)

[maneras de incluirlas 1](#)

[no directamente en marcos 1](#)

[representación visual 1](#)

[inclusión de un objeto 1](#)

[%inline; 1](#)

[Internet Engineering Task Force \(IETF\) 1](#)

[%LanguageCode; 1](#)

[%Length; 1](#)

[lenguaje de formato de documentos \(markup language\) 1](#)

lenguaje de hojas de estilo

[por defecto 1](#)

lenguaje de scripts

[declaración local 1](#)

[especificación 1](#)

[por defecto 1](#)

líneas de división

[entre elementos en bloque 1](#)

[entre las celdas de una tabla 1](#)

lista

[anidamiento 1](#)

[lista de definiciones 1](#)

[numeración 1](#)

[no ordenada 1](#)

[ordenada 1](#)

[representación visual 1](#)

[y hojas de estilo 1](#)

[longitud relativa 1](#)

[mapa de imágenes 1, 2](#)

[accesibilidad 1](#)

con OBJECT [1](#)

en el lado del cliente [1](#)

en el lado del servidor [1](#), [2](#)

ilegal para IMG en BUTTON [1](#)

regiones superpuestas [1](#)

mapa de imágenes en el lado del cliente [1](#)

creación [1](#)

mapa de imágenes en el lado del servidor [1](#), [2](#)

coordenadas del clic [1](#)

marco

algoritmo para hallar el destino [1](#)

borde [1](#)

contenidos iniciales [1](#)

descripción larga [1](#)

en línea [1](#)

espacio en blanco alrededor [1](#)

especificación del marco destino [1](#)

introducción [1](#)

lista de nombres destino reservados [1](#)

problemas con los URIs [1](#)

Marco de Descripción de Recursos (Resource Description Framework, RDF) [1](#), [2](#)

marco destino

algoritmo [1](#)

especificación [1](#)

nombres reservados [1](#)

por defecto [1](#)

semántica [1](#)

mayúsculas y minúsculas

en codificaciones de caracteres [1](#)

en códigos de idioma [1](#)

en datos de estilo [1](#)

en datos de scripts [1](#)

en descriptores de medios [1](#)

en nombres de atributos [1](#)

en nombres de colores [1](#)

en nombres de elementos [1](#)

- en nombres de vínculos [1](#)
- en referencias a entidades de caracteres [1](#)
- en referencias numéricas de caracteres [1](#)
- en tipos de contenido [1](#)
- en tipos de vínculos [1](#)
- en URIs [1](#)
- en valores de atributos [1](#), [2](#), [3](#)
- en valores de longitud [1](#)

`%MediaDesc;` [1](#)

medios

- usados con hojas de estilo [1](#)
- y hojas de estilo externas [1](#)

menú [1](#)

- agrupamiento de opciones [1](#)
- opciones preseleccionadas [1](#)
- representación de opciones [1](#)
- representación visual de opciones agrupadas [1](#)

metadatos [1](#)

- comparación entre LINK y META [1](#)
- perfiles [1](#)
- scheme [1](#)

modelo de contenido [1](#)

- elementos excluidos [1](#)
- sintaxis en el DTD [1](#)

motor de búsqueda

- colaboración [1](#), [2](#)
- y vínculos [1](#)

`%MultiLength;` [1](#)

multipart/form-data [1](#), [2](#)

NAME [1](#)

navegación con tabulador [1](#)

notas sobre notación minimizada [1](#)

NUMBER [1](#)

objeto

- alineación [1](#)
- anchura y altura [1](#)

borde alrededor [1](#)

declarado estáticamente [1](#)

en HEAD [1](#), [2](#), [3](#)

en un formulario [1](#)

espacio en blanco alrededor [1](#)

inclusión genérica [1](#)

inicialización [1](#)

localización de la implementación y los datos [1](#)

nombre del esquema [1](#)

reglas para objetos incluidos [1](#)

representación alternativa [1](#)

representación visual [1](#)

objetos flotantes [1](#)

obsoleto [1](#)

elementos [1](#)

orden de tabulación [1](#)

orígenes y destinos de vínculos (anclas) [1](#)

caracteres ASCII en el nombre [1](#)

caracteres no ASCII en el nombre [1](#)

comparación entre A e id [1](#)

creación con el atributo id [1](#)

creación con un A [1](#)

definidos por scripts [1](#)

espacio de nombres [1](#)

mayúsculas y minúsculas en el nombre [1](#)

referencias de caracteres en el nombre [1](#)

sintaxis del nombre [1](#)

unicidad del nombre [1](#)

párrafo

representación visual [1](#)

paso de un URI relativo a uno absoluto [1](#)

perfil [1](#)

píxel [1](#)

%Pixels; [1](#)

Platform for Internet Content Selection (PICS) [1](#)

posición de código [1](#)

POST

para datos no ASCII de formulario [1](#)

y envío de formularios [1](#)

radiobotón [1](#)

recorte

texto de tablas [1](#)

referencias a entidades de caracteres [1](#)

referencia de caracteres [1](#), [2](#)

para direccionalidad [1](#)

referencias numéricas de caracteres [1](#)

reinicialización de un formulario [1](#)

repertorio de caracteres [1](#)

resumen

de los contenidos de una tabla [1](#)

robot de búsqueda

colaboración [1](#)

rótulo

asociación explícita a un control [1](#)

asociación implícita a un control [1](#)

y el foco [1](#)

salto de línea [1](#)

forzado [1](#)

prohibido [1](#)

y texto bidireccional [1](#)

y texto flotante [1](#)

script

comentarios para ocultar scripts [1](#)

cuando no son soportados [1](#)

datos [1](#)

ejecutados tras un evento [1](#)

ejecutados cuando se carga el documento [1](#)

introducción [1](#)

notas de implementación [1](#)

referencias a elementos [1](#)

sintaxis reservada para scripts [1](#)

usados para especificar un vínculo [1](#)

usados para modificar el documento [1](#)

usos [1](#)

%Script; [1](#)

seguridad

de los controles de contraseña [1](#)

notas [1](#)

separación con guiones [1](#)

SGML

aplicación [1](#)

características con soporte limitado [1](#), [2](#)

catálogo para HTML [1](#)

conjunto de caracteres del documento [1](#)

declaración [1](#)

declaración de HTML 4 [1](#)

definición del tipo de documento (DTD) [1](#)

declaración del tipo de elemento [1](#)

introducción [1](#)

notas de implementación [1](#)

tratamiento de los saltos de línea [1](#)

validación de documentos [1](#)

tabla

algoritmo para encontrar información de cabecera [1](#)

algoritmos de composición [1](#)

alineación de contenidos [1](#)

anchura de las columnas [1](#)

bordes y líneas de división [1](#)

categorización de celdas [1](#)

celdas de datos [1](#)

celdas de encabezado [1](#)

celdas que abarcan varias filas/columnas [1](#)

direccionalidad [1](#)

grupo de columnas [1](#)

grupo de filas [1](#)

márgenes de celdas [1](#)

no para organizar la página [1](#)

notas sobre representación incremental [1](#)

número de columnas [1](#)

número de filas [1](#)

pronunciación de datos de celdas [1](#)

representación incremental [1](#)

representación no visual [1](#)

representación visual [1](#)

resumen de los contenidos [1](#)

título [1](#)

tecla de acceso [1](#)

%Text; [1](#)

text/html [1](#)

texto

citado [1](#)

con ajuste automático de líneas [1](#)

dirección [1](#)

flotante [1](#)

insertado y borrado [1](#)

preformateado [1](#)

texto alternativo

especificación [1](#)

tipo de contenido

application/x-www-form-urlencoded [1](#)

multipart/form-data [1](#)

text/html [1](#)

tipo de vínculo

lista de tipos reconocidos [1](#)

mayúsculas y minúsculas [1](#)

perfiles para tipos nuevos [1](#)

tipos de contenido

para codificar datos de formularios [1](#)

tipos de datos

CDATA [1](#)

ID [1](#)

IDREF [1](#)

IDREFS [1](#)

NAME [1](#)

NUMBER [1](#)

título

a disposición del usuario [1](#)

de un documento [1](#)

usado para anotar elementos [1](#)

URI

caracteres no ASCII en valores de atributos [1](#)

especificación del URI base [1](#)

mayúsculas y minúsculas [1](#)

paso de relativo a absoluto [1](#), [2](#)

relativo [1](#)

usos en HTML [1](#)

%URI; [1](#)

URL

relación con URI [1](#)

usuario [1](#)

UTF-1 [1](#)

UTF-16 [1](#)

valor de atributo #FIXED [1](#)

valor de atributo #IMPLIED [1](#)

valor de atributo #REQUIRED [1](#)

valores por defecto

codificación de caracteres [1](#)

lenguaje de hojas de estilo [1](#)

lenguaje de scripts [1](#)

marco destino [1](#)

vínculo

definición [1](#)

destino por defecto [1](#)

directos e inversos [1](#)

el anidamiento es ilegal [1](#)

representación [1](#)

semántica con marcos destino [1](#)

tipo [1](#)

título [1](#)

usado para definir relaciones [1](#)

usado para obtener recursos [1](#)

y codificación de caracteres [1](#)

y hojas de estilo dependientes del medio [1](#)

y hojas de estilo externas [1](#), [2](#)

World Wide Web (Web) [1](#)

[anterior](#) [contenidos](#) [elementos](#) [atributos](#)

<!--

This is HTML 4.01 Strict DTD, which excludes the presentation attributes and elements that W3C expects to phase out as support for style sheets matures. Authors should use the Strict DTD when possible, but may use the Transitional DTD when support for presentation attribute and elements is required.

HTML 4 includes mechanisms for style sheets, scripting, embedding objects, improved support for right to left and mixed direction text, and enhancements to forms for improved accessibility for people with disabilities.

Draft: \$Date: 1999/12/24 22:40:35 \$

Authors:

Dave Raggett <dsr@w3.org>
Arnaud Le Hors <lehors@w3.org>
Ian Jacobs <ij@w3.org>

Further information about HTML 4.01 is available at:

<http://www.w3.org/TR/1999/REC-html401-19991224>

The HTML 4.01 specification includes additional syntactic constraints that cannot be expressed within the DTDs.

-->

<!--

Typical usage:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
...
</head>
<body>
...
</body>
</html>
```

The URI used as a system identifier with the public identifier allows the user agent to download the DTD and entity sets as needed.

The FPI for the Transitional HTML 4.01 DTD is:

```
"-//W3C//DTD HTML 4.01 Transitional//EN"
```

This version of the transitional DTD is:

<http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd>

If you are writing a document that includes frames, use the following FPI:

```
"-//W3C//DTD HTML 4.01 Frameset//EN"
```

This version of the frameset DTD is:

<http://www.w3.org/TR/1999/REC-html401-19991224/frameset.dtd>

Use the following (relative) URIs to refer to the DTDs and entity definitions of this specification:

```
"strict.dtd"  
"loose.dtd"  
"frameset.dtd"  
"HTMLlat1.ent"  
"HTMLsymbol.ent"  
"HTMLspecial.ent"
```

-->

<!--===== Imported Names =====>

<!-- Feature Switch for frameset documents -->

<!ENTITY % HTML.Frameset "IGNORE">

<!ENTITY % ContentType "CDATA"
-- media type, as per [RFC2045]
-->

<!ENTITY % ContentTypes "CDATA"
-- comma-separated list of media types, as per [RFC2045]
-->

<!ENTITY % Charset "CDATA"
-- a character encoding, as per [RFC2045]
-->

<!ENTITY % Charsets "CDATA"
-- a space-separated list of character encodings, as per [RFC2045]
-->

<!ENTITY % LanguageCode "NAME"
-- a language code, as per [RFC1766]
-->

<!ENTITY % Character "CDATA"
-- a single character from [ISO10646]
-->

<!ENTITY % LinkTypes "CDATA"
-- space-separated list of link types
-->

<!ENTITY % MediaDesc "CDATA"
-- single or comma-separated list of media descriptors
-->

<!ENTITY % URI "CDATA"
-- a Uniform Resource Identifier,
see [URI]
-->

<!ENTITY % Datetime "CDATA" -- date and time information. ISO date format -->

<!ENTITY % Script "CDATA" -- script expression -->

<!ENTITY % StyleSheet "CDATA" -- style sheet data -->

<!ENTITY % Text "CDATA">

<!-- Parameter Entities -->

<!ENTITY % head.misc "SCRIPT|STYLE|META|LINK|OBJECT" -- repeatable head elements -->

<!ENTITY % heading "H1|H2|H3|H4|H5|H6">

<!ENTITY % list "UL | OL">

<!ENTITY % preformatted "PRE">

<!--===== Character mnemonic entities =====-->

<!ENTITY % HTMLlat1 PUBLIC
"-//W3C//ENTITIES Latin1//EN//HTML"
"HTMLlat1.ent">
%HTMLlat1;

<!ENTITY % HTMLsymbol PUBLIC
"-//W3C//ENTITIES Symbols//EN//HTML"
"HTMLsymbol.ent">
%HTMLsymbol;

<!ENTITY % HTMLspecial PUBLIC
"-//W3C//ENTITIES Special//EN//HTML"
"HTMLspecial.ent">
%HTMLspecial;

<!--===== Generic Attributes =====-->

<!ENTITY % coreattrs
"id ID #IMPLIED -- document-wide unique id --
class CDATA #IMPLIED -- space-separated list of classes --
style %StyleSheet; #IMPLIED -- associated style info --
title %Text; #IMPLIED -- advisory title --"
>

<!ENTITY % i18n
"lang %LanguageCode; #IMPLIED -- language code --
dir (ltr|rtl) #IMPLIED -- direction for weak/neutral text --"
>

<!ENTITY % events
"onclick %Script; #IMPLIED -- a pointer button was clicked --
ondblclick %Script; #IMPLIED -- a pointer button was double clicked--
onmousedown %Script; #IMPLIED -- a pointer button was pressed down --
onmouseup %Script; #IMPLIED -- a pointer button was released --
onmouseover %Script; #IMPLIED -- a pointer was moved onto --
onmousemove %Script; #IMPLIED -- a pointer was moved within --
onmouseout %Script; #IMPLIED -- a pointer was moved away --
onkeypress %Script; #IMPLIED -- a key was pressed and released --
onkeydown %Script; #IMPLIED -- a key was pressed down --
onkeyup %Script; #IMPLIED -- a key was released --"
>

<!-- Reserved Feature Switch -->
<!ENTITY % HTML.Reserved "IGNORE">

<!-- The following attributes are reserved for possible future use -->

<![%HTML.Reserved; [
<!ENTITY % reserved
"datasrc %URI; #IMPLIED -- a single or tabular Data Source --
datafld CDATA #IMPLIED -- the property or column name --
dataformatas (plaintext|html) plaintext -- text or html --"
>
]]>

<!ENTITY % reserved "">

<!ENTITY % attrs "%coreattrs; %il8n; %events;">

<!--===== Text Markup =====>

<!ENTITY % fontstyle
"TT | I | B | BIG | SMALL">

<!ENTITY % phrase "EM | STRONG | DFN | CODE |
SAMP | KBD | VAR | CITE | ABBR | ACRONYM" >

<!ENTITY % special
"A | IMG | OBJECT | BR | SCRIPT | MAP | Q | SUB | SUP | SPAN | BDO">

<!ENTITY % formctrl "INPUT | SELECT | TEXTAREA | LABEL | BUTTON">

<!-- %inline; covers inline or "text-level" elements -->
<!ENTITY % inline "#PCDATA | %fontstyle; | %phrase; | %special; | %formctrl;">

<!ELEMENT (%fontstyle;|%phrase;) - - (%inline;)*>
<!ATTLIST (%fontstyle;|%phrase;)
%attrs; -- %coreattrs, %il8n, %events --
>

<!ELEMENT (SUB|SUP) - - (%inline;)* -- subscript, superscript -->
<!ATTLIST (SUB|SUP)
%attrs; -- %coreattrs, %il8n, %events --
>

<!ELEMENT SPAN - - (%inline;)* -- generic language/style container -->
<!ATTLIST SPAN
%attrs; -- %coreattrs, %il8n, %events --
%reserved; -- reserved for possible future use --
>

<!ELEMENT BDO - - (%inline;)* -- I18N BiDi over-ride -->
<!ATTLIST BDO
%coreattrs; -- id, class, style, title --
lang %LanguageCode; #IMPLIED -- language code --
dir (ltr|rtl) #REQUIRED -- directionality --
>

<!ELEMENT BR - O EMPTY -- forced line break -->
<!ATTLIST BR
%coreattrs; -- id, class, style, title --
>

<!--===== HTML content models =====>

<!--
HTML has two basic content models:

%inline; character level elements and text strings
%block; block-like elements e.g. paragraphs and lists
-->

<!ENTITY % block
"P | %heading; | %list; | %preformatted; | DL | DIV | NOSCRIPT |
BLOCKQUOTE | FORM | HR | TABLE | FIELDSET | ADDRESS">

<!ENTITY % flow "%block; | %inline;">

<!--===== Document Body =====>

```
<!ELEMENT BODY O O (%block;|SCRIPT)+ +(INS|DEL) -- document body -->
<!ATTLIST BODY
  %attrs; -- %coreattrs, %i18n, %events --
  onload %Script; #IMPLIED -- the document has been loaded --
  onunload %Script; #IMPLIED -- the document has been removed --
  >

<!ELEMENT ADDRESS - - (%inline;)* -- information on author -->
<!ATTLIST ADDRESS
  %attrs; -- %coreattrs, %i18n, %events --
  >

<!ELEMENT DIV - - (%flow;)* -- generic language/style container -->
<!ATTLIST DIV
  %attrs; -- %coreattrs, %i18n, %events --
  %reserved; -- reserved for possible future use --
  >

<!--===== The Anchor Element =====>

<!ENTITY % Shape "(rect|circle|poly|default)">
<!ENTITY % Coords "CDATA" -- comma-separated list of lengths -->

<!ELEMENT A - - (%inline;)* -(A) -- anchor -->
<!ATTLIST A
  %attrs; -- %coreattrs, %i18n, %events --
  charset %Charset; #IMPLIED -- char encoding of linked resource --
  type %ContentType; #IMPLIED -- advisory content type --
  name CDATA #IMPLIED -- named link end --
  href %URI; #IMPLIED -- URI for linked resource --
  hreflang %LanguageCode; #IMPLIED -- language code --
  rel %LinkTypes; #IMPLIED -- forward link types --
  rev %LinkTypes; #IMPLIED -- reverse link types --
  accesskey %Character; #IMPLIED -- accessibility key character --
  shape %Shape; rect -- for use with client-side image maps --
  coords %Coords; #IMPLIED -- for use with client-side image maps --
  tabindex NUMBER #IMPLIED -- position in tabbing order --
  onfocus %Script; #IMPLIED -- the element got the focus --
  onblur %Script; #IMPLIED -- the element lost the focus --
  >

<!--===== Client-side image maps =====>

<!-- These can be placed in the same document or grouped in a
      separate document although this isn't yet widely supported -->

<!ELEMENT MAP - - ((%block;) | AREA)+ -- client-side image map -->
<!ATTLIST MAP
  %attrs; -- %coreattrs, %i18n, %events --
  name CDATA #REQUIRED -- for reference by usemap --
  >

<!ELEMENT AREA - O EMPTY -- client-side image map area -->
<!ATTLIST AREA
  %attrs; -- %coreattrs, %i18n, %events --
  shape %Shape; rect -- controls interpretation of coords --
  coords %Coords; #IMPLIED -- comma-separated list of lengths --
  href %URI; #IMPLIED -- URI for linked resource --
  nohref (nohref) #IMPLIED -- this region has no action --
  alt %Text; #REQUIRED -- short description --
  tabindex NUMBER #IMPLIED -- position in tabbing order --
  accesskey %Character; #IMPLIED -- accessibility key character --
  onfocus %Script; #IMPLIED -- the element got the focus --
```

onblur %Script; #IMPLIED -- the element lost the focus --
>

<!--===== The LINK Element =====>

<!--
Relationship values can be used in principle:

- a) for document specific toolbars/menus when used with the LINK element in document head e.g. start, contents, previous, next, index, end, help
- b) to link to a separate style sheet (rel=stylesheet)
- c) to make a link to a script (rel=script)
- d) by stylesheets to control how collections of html nodes are rendered into printed documents
- e) to make a link to a printable version of this document e.g. a postscript or pdf version (rel=alternate media=print)

-->

<!ELEMENT LINK - O EMPTY -- a media-independent link -->
<!ATTLIST LINK
%attrs; -- %coreattrs, %i18n, %events --
charset %Charset; #IMPLIED -- char encoding of linked resource --
href %URI; #IMPLIED -- URI for linked resource --
hreflang %LanguageCode; #IMPLIED -- language code --
type %ContentType; #IMPLIED -- advisory content type --
rel %LinkTypes; #IMPLIED -- forward link types --
rev %LinkTypes; #IMPLIED -- reverse link types --
media %MediaDesc; #IMPLIED -- for rendering on these media --
>

<!--===== Images =====>

<!-- Length defined in strict DTD for cellpadding/cellspacing -->
<!ENTITY % Length "CDATA" -- nn for pixels or nn% for percentage length -->
<!ENTITY % MultiLength "CDATA" -- pixel, percentage, or relative -->

<![%HTML.Frameset; [
<!ENTITY % MultiLengths "CDATA" -- comma-separated list of MultiLength -->
]]>

<!ENTITY % Pixels "CDATA" -- integer representing length in pixels -->

<!-- To avoid problems with text-only UAs as well as to make image content understandable and navigable to users of non-visual UAs, you need to provide a description with ALT, and avoid server-side image maps -->

<!ELEMENT IMG - O EMPTY -- Embedded image -->
<!ATTLIST IMG
%attrs; -- %coreattrs, %i18n, %events --
src %URI; #REQUIRED -- URI of image to embed --
alt %Text; #REQUIRED -- short description --
longdesc %URI; #IMPLIED -- link to long description (complements alt) --
name CDATA #IMPLIED -- name of image for scripting --
height %Length; #IMPLIED -- override height --
width %Length; #IMPLIED -- override width --
usemap %URI; #IMPLIED -- use client-side image map --
ismap (ismap) #IMPLIED -- use server-side image map --
>

<!-- USEMAP points to a MAP element which may be in this document or an external document, although the latter is not widely supported -->

<!--===== OBJECT =====>

<!--
OBJECT is used to embed objects as part of HTML pages
PARAM elements should precede other content. SGML mixed content
model technicality precludes specifying this formally ...
-->

<!ELEMENT OBJECT - - (PARAM | %flow;)*
-- generic embedded object -->

<!ATTLIST OBJECT
%attrs; -- %coreattrs, %i18n, %events --
declare (declare) #IMPLIED -- declare but don't instantiate flag --
classid %URI; #IMPLIED -- identifies an implementation --
codebase %URI; #IMPLIED -- base URI for classid, data, archive--
data %URI; #IMPLIED -- reference to object's data --
type %ContentType; #IMPLIED -- content type for data --
codetype %ContentType; #IMPLIED -- content type for code --
archive CDATA #IMPLIED -- space-separated list of URIs --
standby %Text; #IMPLIED -- message to show while loading --
height %Length; #IMPLIED -- override height --
width %Length; #IMPLIED -- override width --
usemap %URI; #IMPLIED -- use client-side image map --
name CDATA #IMPLIED -- submit as part of form --
tabindex NUMBER #IMPLIED -- position in tabbing order --
%reserved; -- reserved for possible future use --
>

<!ELEMENT PARAM - O EMPTY -- named property value -->

<!ATTLIST PARAM
id ID #IMPLIED -- document-wide unique id --
name CDATA #REQUIRED -- property name --
value CDATA #IMPLIED -- property value --
valuetype (DATA|REF|OBJECT) DATA -- How to interpret value --
type %ContentType; #IMPLIED -- content type for value
when valuetype=ref --
>

<!--===== Horizontal Rule =====>

<!ELEMENT HR - O EMPTY -- horizontal rule -->

<!ATTLIST HR
%attrs; -- %coreattrs, %i18n, %events --
>

<!--===== Paragraphs =====>

<!ELEMENT P - O (%inline;)* -- paragraph -->

<!ATTLIST P
%attrs; -- %coreattrs, %i18n, %events --
>

<!--===== Headings =====>

<!--
There are six levels of headings from H1 (the most important)
to H6 (the least important).
-->

<!ELEMENT (%heading;) - - (%inline;)* -- heading -->

<!ATTLIST (%heading;)
%attrs; -- %coreattrs, %i18n, %events --
>

<!--===== Preformatted Text =====>


```

<!-- excludes markup for images and changes in font size -->
<!ENTITY % pre.exclusion "IMG|OBJECT|BIG|SMALL|SUB|SUP">

<!ELEMENT PRE - - (%inline;)* -(%pre.exclusion;) -- preformatted text -->
<!ATTLIST PRE
  %attrs; -- %coreattrs, %i18n, %events --
>

<!--===== Inline Quotes =====>

<!ELEMENT Q - - (%inline;)* -- short inline quotation -->
<!ATTLIST Q
  %attrs; -- %coreattrs, %i18n, %events --
  cite %URI; #IMPLIED -- URI for source document or msg --
>

<!--===== Block-like Quotes =====>

<!ELEMENT BLOCKQUOTE - - (%block;|SCRIPT)+ -- long quotation -->
<!ATTLIST BLOCKQUOTE
  %attrs; -- %coreattrs, %i18n, %events --
  cite %URI; #IMPLIED -- URI for source document or msg --
>

<!--===== Inserted/Deleted Text =====>

<!-- INS/DEL are handled by inclusion on BODY -->
<!ELEMENT (INS|DEL) - - (%flow;)* -- inserted text, deleted text -->
<!ATTLIST (INS|DEL)
  %attrs; -- %coreattrs, %i18n, %events --
  cite %URI; #IMPLIED -- info on reason for change --
  datetime %Datetime; #IMPLIED -- date and time of change --
>

<!--===== Lists =====>

<!-- definition lists - DT for term, DD for its definition -->

<!ELEMENT DL - - (DT|DD)+ -- definition list -->
<!ATTLIST DL
  %attrs; -- %coreattrs, %i18n, %events --
>

<!ELEMENT DT - O (%inline;)* -- definition term -->
<!ELEMENT DD - O (%flow;)* -- definition description -->
<!ATTLIST (DT|DD)
  %attrs; -- %coreattrs, %i18n, %events --
>

<!ELEMENT OL - - (LI)+ -- ordered list -->
<!ATTLIST OL
  %attrs; -- %coreattrs, %i18n, %events --
>

<!-- Unordered Lists (UL) bullet styles -->
<!ELEMENT UL - - (LI)+ -- unordered list -->
<!ATTLIST UL
  %attrs; -- %coreattrs, %i18n, %events --
>

```

```

<!ELEMENT LI - O (%flow;)* -- list item -->
<!ATTLIST LI
  %attrs; -- %coreattrs, %i18n, %events --
>

<!--===== Forms =====>
<!ELEMENT FORM - - (%block;|SCRIPT)+ -(FORM) -- interactive form -->
<!ATTLIST FORM
  %attrs; -- %coreattrs, %i18n, %events --
  action %URI; #REQUIRED -- server-side form handler --
  method (GET|POST) GET -- HTTP method used to submit the form--
  enctype %ContentType; "application/x-www-form-urlencoded"
  accept %ContentTypes; #IMPLIED -- list of MIME types for file upload --
  name CDATA #IMPLIED -- name of form for scripting --
  onsubmit %Script; #IMPLIED -- the form was submitted --
  onreset %Script; #IMPLIED -- the form was reset --
  accept-charset %Charsets; #IMPLIED -- list of supported charsets --
>

<!-- Each label must not contain more than ONE field -->
<!ELEMENT LABEL - - (%inline;)* -(LABEL) -- form field label text -->
<!ATTLIST LABEL
  %attrs; -- %coreattrs, %i18n, %events --
  for IDREF #IMPLIED -- matches field ID value --
  accesskey %Character; #IMPLIED -- accessibility key character --
  onfocus %Script; #IMPLIED -- the element got the focus --
  onblur %Script; #IMPLIED -- the element lost the focus --
>

<!ENTITY % InputType
  "(TEXT | PASSWORD | CHECKBOX |
  RADIO | SUBMIT | RESET |
  FILE | HIDDEN | IMAGE | BUTTON)"
>

<!-- attribute name required for all but submit and reset -->
<!ELEMENT INPUT - O EMPTY -- form control -->
<!ATTLIST INPUT
  %attrs; -- %coreattrs, %i18n, %events --
  type %InputType; TEXT -- what kind of widget is needed --
  name CDATA #IMPLIED -- submit as part of form --
  value CDATA #IMPLIED -- Specify for radio buttons and checkboxes --
  checked (checked) #IMPLIED -- for radio buttons and check boxes --
  disabled (disabled) #IMPLIED -- unavailable in this context --
  readonly (readonly) #IMPLIED -- for text and passwd --
  size CDATA #IMPLIED -- specific to each type of field --
  maxlength NUMBER #IMPLIED -- max chars for text fields --
  src %URI; #IMPLIED -- for fields with images --
  alt CDATA #IMPLIED -- short description --
  usemap %URI; #IMPLIED -- use client-side image map --
  ismap (ismap) #IMPLIED -- use server-side image map --
  tabindex NUMBER #IMPLIED -- position in tabbing order --
  accesskey %Character; #IMPLIED -- accessibility key character --
  onfocus %Script; #IMPLIED -- the element got the focus --
  onblur %Script; #IMPLIED -- the element lost the focus --
  onselect %Script; #IMPLIED -- some text was selected --
  onchange %Script; #IMPLIED -- the element value was changed --
  accept %ContentTypes; #IMPLIED -- list of MIME types for file upload --
  %reserved; -- reserved for possible future use --
>

<!ELEMENT SELECT - - (OPTGROUP|OPTION)+ -- option selector -->
<!ATTLIST SELECT
  %attrs; -- %coreattrs, %i18n, %events --
  name CDATA #IMPLIED -- field name --

```

```
size NUMBER #IMPLIED -- rows visible --
multiple (multiple) #IMPLIED -- default is single selection --
disabled (disabled) #IMPLIED -- unavailable in this context --
tabindex NUMBER #IMPLIED -- position in tabbing order --
onfocus %Script; #IMPLIED -- the element got the focus --
onblur %Script; #IMPLIED -- the element lost the focus --
onchange %Script; #IMPLIED -- the element value was changed --
%reserved; -- reserved for possible future use --
>
```

<!ELEMENT OPTGROUP - - (OPTION)+ -- option group -->

<!ATTLIST OPTGROUP

```
%attrs; -- %coreattrs, %i18n, %events --
disabled (disabled) #IMPLIED -- unavailable in this context --
label %Text; #REQUIRED -- for use in hierarchical menus --
>
```

<!ELEMENT OPTION - O (#PCDATA) -- selectable choice -->

<!ATTLIST OPTION

```
%attrs; -- %coreattrs, %i18n, %events --
selected (selected) #IMPLIED
disabled (disabled) #IMPLIED -- unavailable in this context --
label %Text; #IMPLIED -- for use in hierarchical menus --
value CDATA #IMPLIED -- defaults to element content --
>
```

<!ELEMENT TEXTAREA - - (#PCDATA) -- multi-line text field -->

<!ATTLIST TEXTAREA

```
%attrs; -- %coreattrs, %i18n, %events --
name CDATA #IMPLIED
rows NUMBER #REQUIRED
cols NUMBER #REQUIRED
disabled (disabled) #IMPLIED -- unavailable in this context --
readonly (readonly) #IMPLIED
tabindex NUMBER #IMPLIED -- position in tabbing order --
accesskey %Character; #IMPLIED -- accessibility key character --
onfocus %Script; #IMPLIED -- the element got the focus --
onblur %Script; #IMPLIED -- the element lost the focus --
onselect %Script; #IMPLIED -- some text was selected --
onchange %Script; #IMPLIED -- the element value was changed --
%reserved; -- reserved for possible future use --
>
```

<!--
#PCDATA is to solve the mixed content problem,
per specification only whitespace is allowed there!
-->

<!ELEMENT FIELDSET - - (#PCDATA,LEGEND,(%flow;)* -- form control group -->

<!ATTLIST FIELDSET

```
%attrs; -- %coreattrs, %i18n, %events --
>
```

<!ELEMENT LEGEND - - (%inline;)* -- fieldset legend -->

<!ATTLIST LEGEND

```
%attrs; -- %coreattrs, %i18n, %events --
accesskey %Character; #IMPLIED -- accessibility key character --
>
```

<!ELEMENT BUTTON - -
(%flow;)* -(A|%formctrl;|FORM|FIELDSET)
-- push button -->

<!ATTLIST BUTTON

```
%attrs; -- %coreattrs, %i18n, %events --
name CDATA #IMPLIED
```

```

value          CDATA          #IMPLIED  -- sent to server when submitted --
type           (button|submit|reset) submit -- for use as form button --
disabled       (disabled)     #IMPLIED  -- unavailable in this context --
tabindex       NUMBER         #IMPLIED  -- position in tabbing order --
accesskey      %Character;    #IMPLIED  -- accessibility key character --
onfocus       %Script;       #IMPLIED  -- the element got the focus --
onblur         %Script;       #IMPLIED  -- the element lost the focus --
%reserved;    -- reserved for possible future use --
>

```

<!--===== Tables =====-->

<!-- IETF HTML table standard, see [RFC1942] -->

<!--

The BORDER attribute sets the thickness of the frame around the table. The default units are screen pixels.

The FRAME attribute specifies which parts of the frame around the table should be rendered. The values are not the same as CALS to avoid a name clash with the VALIGN attribute.

The value "border" is included for backwards compatibility with <TABLE BORDER> which yields frame=border and border=implied For <TABLE BORDER=1> you get border=1 and frame=implied. In this case, it is appropriate to treat this as frame=border for backwards compatibility with deployed browsers.

-->

<!ENTITY % TFrame "(void|above|below|hsides|lhs|rhs|vsides|box|border)">

<!--

The RULES attribute defines which rules to draw between cells:

If RULES is absent then assume:

"none" if BORDER is absent or BORDER=0 otherwise "all"

-->

<!ENTITY % TRules "(none | groups | rows | cols | all)">

<!-- horizontal placement of table relative to document -->

<!ENTITY % TAlign "(left|center|right)">

<!-- horizontal alignment attributes for cell contents -->

<!ENTITY % cellhalign

```

"align          (left|center|right|justify|char) #IMPLIED
char            %Character;    #IMPLIED  -- alignment char, e.g. char=':' --
charoff         %Length;       #IMPLIED  -- offset for alignment char --"
>

```

<!-- vertical alignment attributes for cell contents -->

<!ENTITY % cellvalign

```

"valign        (top|middle|bottom|baseline) #IMPLIED"
>

```

<!ELEMENT TABLE - -

(CAPTION?, (COL*|COLGROUP*), THEAD?, TFOOT?, TBODY+)>

<!ELEMENT CAPTION - - (%inline;)* -- table caption -->

<!ELEMENT THEAD - O (TR)+ -- table header -->

<!ELEMENT TFOOT - O (TR)+ -- table footer -->

<!ELEMENT TBODY O O (TR)+ -- table body -->

<!ELEMENT COLGROUP - O (COL)* -- table column group -->

<!ELEMENT COL - O EMPTY -- table column -->

<!ELEMENT TR - O (TH|TD)+ -- table row -->

<!ELEMENT (TH|TD) - O (%flow;)* -- table header cell, table data cell-->

```

<!ATTLIST TABLE
  %attrs;
  summary      %Text;          #IMPLIED -- purpose/structure for speech output--
  width        %Length;       #IMPLIED -- table width --
  border       %Pixels;       #IMPLIED -- controls frame width around table --
  frame        %TFrame;       #IMPLIED -- which parts of frame to render --
  rules        %TRules;       #IMPLIED -- rulings between rows and cols --
  cellspacing  %Length;       #IMPLIED -- spacing between cells --
  cellpadding  %Length;       #IMPLIED -- spacing within cells --
  %reserved;
  datapagesize CDATA          #IMPLIED -- reserved for possible future use --
>

```

```

<!ATTLIST CAPTION
  %attrs;
  %coreattrs, %i18n, %events --
>

```

<!--
COLGROUP groups a set of COL elements. It allows you to group several semantically related columns together.

```

-->
<!ATTLIST COLGROUP
  %attrs;
  span          NUMBER          1 -- default number of columns in group --
  width         %MultiLength;   #IMPLIED -- default width for enclosed COLs --
  %cellhalign;
  %cellvalign;
  %coreattrs, %i18n, %events --
  horizontal alignment in cells --
  vertical alignment in cells --
>

```

<!--
COL elements define the alignment properties for cells in one or more columns.

The WIDTH attribute specifies the width of the columns, e.g.

```

width=64          width in screen pixels
width=0.5*        relative width of 0.5

```

The SPAN attribute causes the attributes of one COL element to apply to more than one column.

```

-->
<!ATTLIST COL
  %attrs;
  span          NUMBER          1 -- COL attributes affect N columns --
  width         %MultiLength;   #IMPLIED -- column width specification --
  %cellhalign;
  %cellvalign;
  %coreattrs, %i18n, %events --
  horizontal alignment in cells --
  vertical alignment in cells --
>

```

<!--
Use THEAD to duplicate headers when breaking table across page boundaries, or for static headers when TBODY sections are rendered in scrolling panel.

Use TFOOT to duplicate footers when breaking table across page boundaries, or for static footers when TBODY sections are rendered in scrolling panel.

Use multiple TBODY sections when rules are needed between groups of table rows.

```

-->
<!ATTLIST (THEAD|TBODY|TFOOT)
  %attrs;
  %cellhalign;
  %coreattrs, %i18n, %events --
  horizontal alignment in cells --

```

```

%cellvalign;          -- vertical alignment in cells --
>

<!-- table row --
<!ATTLIST TR
  %attrs;              -- %coreattrs, %i18n, %events --
  %cellhalign;        -- horizontal alignment in cells --
  %cellvalign;        -- vertical alignment in cells --
>

<!-- Scope is simpler than headers attribute for common tables -->
<!ENTITY % Scope "(row|col|rowgroup|colgroup)">

<!-- TH is for headers, TD for data, but for cells acting as both use TD -->
<!ATTLIST (TH|TD)
  %attrs;              -- %coreattrs, %i18n, %events --
  abbr %Text;          #IMPLIED -- abbreviation for header cell --
  axis CDATA           #IMPLIED -- comma-separated list of related headers--
  headers IDREFS       #IMPLIED -- list of id's for header cells --
  scope %Scope;       #IMPLIED -- scope covered by header cells --
  rowspan NUMBER       1      -- number of rows spanned by cell --
  colspan NUMBER       1      -- number of cols spanned by cell --
  %cellhalign;        -- horizontal alignment in cells --
  %cellvalign;        -- vertical alignment in cells --
>

<!--===== Document Head =====>
<!-- %head.misc; defined earlier on as "SCRIPT|STYLE|META|LINK|OBJECT" -->
<!ENTITY % head.content "TITLE & BASE?">

<!ELEMENT HEAD O O (%head.content;) +(%head.misc;) -- document head -->
<!ATTLIST HEAD
  %i18n;              -- lang, dir --
  profile %URI;       #IMPLIED -- named dictionary of meta info --
>

<!-- The TITLE element is not considered part of the flow of text.
      It should be displayed, for example as the page header or
      window title. Exactly one title is required per document.
-->
<!ELEMENT TITLE - - (#PCDATA) -(%head.misc;) -- document title -->
<!ATTLIST TITLE %i18n>

<!ELEMENT BASE - O EMPTY -- document base URI -->
<!ATTLIST BASE
  href %URI;         #REQUIRED -- URI that acts as base URI --
>

<!ELEMENT META - O EMPTY -- generic metainformation -->
<!ATTLIST META
  %i18n;              -- lang, dir, for use with content --
  http-equiv NAME     #IMPLIED -- HTTP response header name --
  name NAME          #IMPLIED -- metainformation name --
  content CDATA      #REQUIRED -- associated information --
  scheme CDATA       #IMPLIED -- select form of content --
>

<!ELEMENT STYLE - - %StyleSheet -- style info -->
<!ATTLIST STYLE
  %i18n;              -- lang, dir, for use with title --
  type %ContentType; #REQUIRED -- content type of style language --
  media %MediaDesc;  #IMPLIED -- designed for use with these media --
  title %Text;       #IMPLIED -- advisory title --

```

>

```
<!ELEMENT SCRIPT - - %Script;          -- script statements -->
<!ATTLIST SCRIPT
  charset      %Charset;      #IMPLIED -- char encoding of linked resource --
  type         %ContentType;  #REQUIRED -- content type of script language --
  src          %URI;          #IMPLIED -- URI for an external script --
  defer       (defer)        #IMPLIED -- UA may defer execution of script --
  event       CDATA          #IMPLIED -- reserved for possible future use --
  for         %URI;          #IMPLIED -- reserved for possible future use --
>
```

```
<!ELEMENT NOSCRIPT - - (%block;)+
  -- alternate content container for non script-based rendering -->
<!ATTLIST NOSCRIPT
  %attrs;          -- %coreattrs, %i18n, %events --
>
```

```
<!--===== Document Structure =====>
<!ENTITY % html.content "HEAD, BODY">
```

```
<!ELEMENT HTML O O (%html.content;)    -- document root element -->
<!ATTLIST HTML
  %i18n;          -- lang, dir --
>
```

<!--

This is the HTML 4.01 Transitional DTD, which includes presentation attributes and elements that W3C expects to phase out as support for style sheets matures. Authors should use the Strict DTD when possible, but may use the Transitional DTD when support for presentation attribute and elements is required.

HTML 4 includes mechanisms for style sheets, scripting, embedding objects, improved support for right to left and mixed direction text, and enhancements to forms for improved accessibility for people with disabilities.

Draft: \$Date: 1999/12/24 22:40:35 \$

Authors:

Dave Raggett <dsr@w3.org>
Arnaud Le Hors <lehors@w3.org>
Ian Jacobs <ij@w3.org>

Further information about HTML 4.01 is available at:

<http://www.w3.org/TR/1999/REC-html401-19991224>

The HTML 4.01 specification includes additional syntactic constraints that cannot be expressed within the DTDs.

-->

<!ENTITY % HTML.Version "-//W3C//DTD HTML 4.01 Transitional//EN"

-- Typical usage:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
...
</head>
<body>
...
</body>
</html>
```

The URI used as a system identifier with the public identifier allows the user agent to download the DTD and entity sets as needed.

The FPI for the Strict HTML 4.01 DTD is:

"-//W3C//DTD HTML 4.01//EN"

This version of the strict DTD is:

<http://www.w3.org/TR/1999/REC-html401-19991224/strict.dtd>

Authors should use the Strict DTD unless they need the presentation control for user agents that don't (adequately) support style sheets.

If you are writing a document that includes frames, use the following FPI:

"-//W3C//DTD HTML 4.01 Frameset//EN"

This version of the frameset DTD is:

Use the following (relative) URIs to refer to the DTDs and entity definitions of this specification:

"strict.dtd"
"loose.dtd"
"frameset.dtd"
"HTMLlat1.ent"
"HTMLsymbol.ent"
"HTMLspecial.ent"

-->

<!--===== Imported Names =====>

<!-- Feature Switch for frameset documents -->

<!ENTITY % HTML.Frameset "IGNORE">

<!ENTITY % ContentType "CDATA"
-- media type, as per [RFC2045]
-->

<!ENTITY % ContentTypes "CDATA"
-- comma-separated list of media types, as per [RFC2045]
-->

<!ENTITY % Charset "CDATA"
-- a character encoding, as per [RFC2045]
-->

<!ENTITY % Charsets "CDATA"
-- a space-separated list of character encodings, as per [RFC2045]
-->

<!ENTITY % LanguageCode "NAME"
-- a language code, as per [RFC1766]
-->

<!ENTITY % Character "CDATA"
-- a single character from [ISO10646]
-->

<!ENTITY % LinkTypes "CDATA"
-- space-separated list of link types
-->

<!ENTITY % MediaDesc "CDATA"
-- single or comma-separated list of media descriptors
-->

<!ENTITY % URI "CDATA"
-- a Uniform Resource Identifier,
see [URI]
-->

<!ENTITY % Datetime "CDATA" -- date and time information. ISO date format -->

<!ENTITY % Script "CDATA" -- script expression -->

<!ENTITY % StyleSheet "CDATA" -- style sheet data -->

<!ENTITY % FrameTarget "CDATA" -- render in this frame -->

<!ENTITY % Text "CDATA">

<!-- Parameter Entities -->

<!ENTITY % head.misc "SCRIPT|STYLE|META|LINK|OBJECT" -- repeatable head elements -->

<!ENTITY % heading "H1|H2|H3|H4|H5|H6">

<!ENTITY % list "UL | OL | DIR | MENU">

<!ENTITY % preformatted "PRE">

<!ENTITY % Color "CDATA" -- a color using sRGB: #RRGGBB as Hex values -->

<!-- There are also 16 widely known color names with their sRGB values:

| | | | | | |
|---------|---|---------|--------|---|---------|
| Black | = | #000000 | Green | = | #008000 |
| Silver | = | #C0C0C0 | Lime | = | #00FF00 |
| Gray | = | #808080 | Olive | = | #808000 |
| White | = | #FFFFFF | Yellow | = | #FFFF00 |
| Maroon | = | #800000 | Navy | = | #000080 |
| Red | = | #FF0000 | Blue | = | #0000FF |
| Purple | = | #800080 | Teal | = | #008080 |
| Fuchsia | = | #FF00FF | Aqua | = | #00FFFF |

-->

<!ENTITY % bodycolors "

| | | | |
|---------|---------|----------|---------------------------------|
| bgcolor | %Color; | #IMPLIED | -- document background color -- |
| text | %Color; | #IMPLIED | -- document text color -- |
| link | %Color; | #IMPLIED | -- color of links -- |
| vlink | %Color; | #IMPLIED | -- color of visited links -- |
| alink | %Color; | #IMPLIED | -- color of selected links -- |

">

<!--===== Character mnemonic entities =====>

<!ENTITY % HTMLlat1 PUBLIC
"-//W3C//ENTITIES Latin1//EN//HTML"
"HTMLlat1.ent">

%HTMLlat1;

<!ENTITY % HTMLsymbol PUBLIC
"-//W3C//ENTITIES Symbols//EN//HTML"
"HTMLsymbol.ent">

%HTMLsymbol;

<!ENTITY % HTMLspecial PUBLIC
"-//W3C//ENTITIES Special//EN//HTML"
"HTMLspecial.ent">

%HTMLspecial;

<!--===== Generic Attributes =====>

<!ENTITY % coreattrs

| | | | |
|-------|--------------|----------|---------------------------------------|
| "id | ID | #IMPLIED | -- document-wide unique id -- |
| class | CDATA | #IMPLIED | -- space-separated list of classes -- |
| style | %StyleSheet; | #IMPLIED | -- associated style info -- |
| title | %Text; | #IMPLIED | -- advisory title --" |

>

<!ENTITY % i18n

| | | | |
|-------|----------------|----------|--|
| "lang | %LanguageCode; | #IMPLIED | -- language code -- |
| dir | (ltr rtl) | #IMPLIED | -- direction for weak/neutral text --" |

>

```

<!ENTITY % events
"onclick      %Script;      #IMPLIED -- a pointer button was clicked --
ondblclick    %Script;      #IMPLIED -- a pointer button was double clicked--
onmousedown   %Script;      #IMPLIED -- a pointer button was pressed down --
onmouseup     %Script;      #IMPLIED -- a pointer button was released --
onmouseover   %Script;      #IMPLIED -- a pointer was moved onto --
onmousemove   %Script;      #IMPLIED -- a pointer was moved within --
onmouseout    %Script;      #IMPLIED -- a pointer was moved away --
onkeypress    %Script;      #IMPLIED -- a key was pressed and released --
onkeydown     %Script;      #IMPLIED -- a key was pressed down --
onkeyup       %Script;      #IMPLIED -- a key was released --"
>

```

```

<!-- Reserved Feature Switch -->
<!ENTITY % HTML.Reserved "IGNORE">

```

```

<!-- The following attributes are reserved for possible future use -->

```

```

<![ %HTML.Reserved; [
<!ENTITY % reserved
"datasrc      %URI;          #IMPLIED -- a single or tabular Data Source --
datafld       CDATA          #IMPLIED -- the property or column name --
dataformatas (plaintext|html) plaintext -- text or html --"
>
]]>

```

```

<!ENTITY % reserved "">

```

```

<!ENTITY % attrs "%coreattrs; %il8n; %events;">

```

```

<!ENTITY % align "align (left|center|right|justify) #IMPLIED"
-- default is left for ltr paragraphs, right for rtl --
>

```

```

<!--===== Text Markup =====>

```

```

<!ENTITY % fontstyle
"TT | I | B | U | S | STRIKE | BIG | SMALL">

```

```

<!ENTITY % phrase "EM | STRONG | DFN | CODE |
SAMP | KBD | VAR | CITE | ABBR | ACRONYM" >

```

```

<!ENTITY % special
"A | IMG | APPLET | OBJECT | FONT | BASEFONT | BR | SCRIPT |
MAP | Q | SUB | SUP | SPAN | BDO | IFRAME">

```

```

<!ENTITY % formctrl "INPUT | SELECT | TEXTAREA | LABEL | BUTTON">

```

```

<!-- %inline; covers inline or "text-level" elements -->
<!ENTITY % inline "#PCDATA | %fontstyle; | %phrase; | %special; | %formctrl;">

```

```

<!ELEMENT (%fontstyle;|%phrase;) - - (%inline;)*>
<!ATTLIST (%fontstyle;|%phrase;)
%attrs; -- %coreattrs, %il8n, %events --
>

```

```

<!ELEMENT (SUB|SUP) - - (%inline;)* -- subscript, superscript -->
<!ATTLIST (SUB|SUP)
%attrs; -- %coreattrs, %il8n, %events --
>

```

```

<!ELEMENT SPAN - - (%inline;)* -- generic language/style container -->
<!ATTLIST SPAN
%attrs; -- %coreattrs, %il8n, %events --
%reserved; -- reserved for possible future use --
>

```

```

<!ELEMENT BDO - - (%inline;)* -- I18N BiDi over-ride -->
<!ATTLIST BDO
  %coreattrs; -- id, class, style, title --
  lang %LanguageCode; #IMPLIED -- language code --
  dir (ltr|rtl) #REQUIRED -- directionality --
>

<!ELEMENT BASEFONT - O EMPTY -- base font size -->
<!ATTLIST BASEFONT
  id ID #IMPLIED -- document-wide unique id --
  size CDATA #REQUIRED -- base font size for FONT elements --
  color %Color; #IMPLIED -- text color --
  face CDATA #IMPLIED -- comma-separated list of font names --
>

<!ELEMENT FONT - - (%inline;)* -- local change to font -->
<!ATTLIST FONT
  %coreattrs; -- id, class, style, title --
  %i18n; -- lang, dir --
  size CDATA #IMPLIED -- [+|-]nn e.g. size="+1", size="4" --
  color %Color; #IMPLIED -- text color --
  face CDATA #IMPLIED -- comma-separated list of font names --
>

<!ELEMENT BR - O EMPTY -- forced line break -->
<!ATTLIST BR
  %coreattrs; -- id, class, style, title --
  clear (left|all|right|none) none -- control of text flow --
>

<!--===== HTML content models =====>

<!--
  HTML has two basic content models:

  %inline; character level elements and text strings
  %block; block-like elements e.g. paragraphs and lists
-->

<!ENTITY % block
  "P | %heading; | %list; | %preformatted; | DL | DIV | CENTER |
  NOSCRIPT | NOFRAMES | BLOCKQUOTE | FORM | ISINDEX | HR |
  TABLE | FIELDSET | ADDRESS">

<!ENTITY % flow "%block; | %inline;">

<!--===== Document Body =====>

<!ELEMENT BODY O O (%flow;)* +(INS|DEL) -- document body -->
<!ATTLIST BODY
  %attrs; -- %coreattrs, %i18n, %events --
  onload %Script; #IMPLIED -- the document has been loaded --
  onunload %Script; #IMPLIED -- the document has been removed --
  background %URI; #IMPLIED -- texture tile for document
  background --
  %bodycolors; -- bgcolor, text, link, vlink, alink --
>

<!ELEMENT ADDRESS - - ((%inline;)|P)* -- information on author -->
<!ATTLIST ADDRESS
  %attrs; -- %coreattrs, %i18n, %events --
>

<!ELEMENT DIV - - (%flow;)* -- generic language/style container -->

```

```

<!ATTLIST DIV
  %attrs;           -- %coreattrs, %i18n, %events --
  %align;           -- align, text alignment --
  %reserved;       -- reserved for possible future use --
>

```

```

<!ELEMENT CENTER - - (%flow;)*           -- shorthand for DIV align=center -->
<!ATTLIST CENTER
  %attrs;           -- %coreattrs, %i18n, %events --
>

```

<!--===== The Anchor Element =====>

```

<!ENTITY % Shape "(rect|circle|poly|default)">
<!ENTITY % Coords "CDATA" -- comma-separated list of lengths -->

```

```

<!ELEMENT A - - (%inline;)* -(A)        -- anchor -->
<!ATTLIST A
  %attrs;           -- %coreattrs, %i18n, %events --
  charset          %Charset;             #IMPLIED -- char encoding of linked resource --
  type             %ContentType;         #IMPLIED -- advisory content type --
  name             CDATA                  #IMPLIED -- named link end --
  href            %URI;                  #IMPLIED -- URI for linked resource --
  hreflang        %LanguageCode;        #IMPLIED -- language code --
  target          %FrameTarget;         #IMPLIED -- render in this frame --
  rel             %LinkTypes;           #IMPLIED -- forward link types --
  rev            %LinkTypes;           #IMPLIED -- reverse link types --
  accesskey      %Character;            #IMPLIED -- accessibility key character --
  shape          %Shape;                rect    -- for use with client-side image maps --
  coords         %Coords;               #IMPLIED -- for use with client-side image maps --
  tabindex       NUMBER                 #IMPLIED -- position in tabbing order --
  onfocus       %Script;               #IMPLIED -- the element got the focus --
  onblur        %Script;               #IMPLIED -- the element lost the focus --
>

```

<!--===== Client-side image maps =====>

<!-- These can be placed in the same document or grouped in a separate document although this isn't yet widely supported -->

```

<!ELEMENT MAP - - ((%block;) | AREA)+ -- client-side image map -->
<!ATTLIST MAP
  %attrs;           -- %coreattrs, %i18n, %events --
  name             CDATA                  #REQUIRED -- for reference by usemap --
>

```

```

<!ELEMENT AREA - O EMPTY                -- client-side image map area -->
<!ATTLIST AREA
  %attrs;           -- %coreattrs, %i18n, %events --
  shape          %Shape;                rect    -- controls interpretation of coords --
  coords         %Coords;               #IMPLIED -- comma-separated list of lengths --
  href          %URI;                   #IMPLIED -- URI for linked resource --
  target        %FrameTarget;          #IMPLIED -- render in this frame --
  nohref       (nohref)                #IMPLIED -- this region has no action --
  alt          %Text;                   #REQUIRED -- short description --
  tabindex     NUMBER                   #IMPLIED -- position in tabbing order --
  accesskey    %Character;              #IMPLIED -- accessibility key character --
  onfocus     %Script;                 #IMPLIED -- the element got the focus --
  onblur      %Script;                 #IMPLIED -- the element lost the focus --
>

```

<!--===== The LINK Element =====>

<!-- Relationship values can be used in principle:

- a) for document specific toolbars/menus when used with the LINK element in document head e.g. start, contents, previous, next, index, end, help
- b) to link to a separate style sheet (rel=stylesheet)
- c) to make a link to a script (rel=script)
- d) by stylesheets to control how collections of html nodes are rendered into printed documents
- e) to make a link to a printable version of this document e.g. a postscript or pdf version (rel=alternate media=print)

-->

```
<!ELEMENT LINK - O EMPTY -- a media-independent link -->
<!ATTLIST LINK
  %attrs; -- %coreattrs, %i18n, %events --
  charset %Charset; #IMPLIED -- char encoding of linked resource --
  href %URI; #IMPLIED -- URI for linked resource --
  hreflang %LanguageCode; #IMPLIED -- language code --
  type %ContentType; #IMPLIED -- advisory content type --
  rel %LinkTypes; #IMPLIED -- forward link types --
  rev %LinkTypes; #IMPLIED -- reverse link types --
  media %MediaDesc; #IMPLIED -- for rendering on these media --
  target %FrameTarget; #IMPLIED -- render in this frame --
>
```

<!--===== Images =====-->

```
<!-- Length defined in strict DTD for cellpadding/cellspacing -->
<!ENTITY % Length "CDATA" -- nn for pixels or nn% for percentage length -->
<!ENTITY % MultiLength "CDATA" -- pixel, percentage, or relative -->
```

```
<![ %HTML.Frameset; [
<!ENTITY % MultiLengths "CDATA" -- comma-separated list of MultiLength -->
]]>
```

```
<!ENTITY % Pixels "CDATA" -- integer representing length in pixels -->
```

```
<!ENTITY % IAlign "(top|middle|bottom|left|right)" -- center? -->
```

```
<!-- To avoid problems with text-only UAs as well as
to make image content understandable and navigable
to users of non-visual UAs, you need to provide
a description with ALT, and avoid server-side image maps -->
```

```
<!ELEMENT IMG - O EMPTY -- Embedded image -->
<!ATTLIST IMG
  %attrs; -- %coreattrs, %i18n, %events --
  src %URI; #REQUIRED -- URI of image to embed --
  alt %Text; #REQUIRED -- short description --
  longdesc %URI; #IMPLIED -- link to long description
  (complements alt) --
  name CDATA #IMPLIED -- name of image for scripting --
  height %Length; #IMPLIED -- override height --
  width %Length; #IMPLIED -- override width --
  usemap %URI; #IMPLIED -- use client-side image map --
  ismap (ismap) #IMPLIED -- use server-side image map --
  align %IAlign; #IMPLIED -- vertical or horizontal alignment --
  border %Pixels; #IMPLIED -- link border width --
  hspace %Pixels; #IMPLIED -- horizontal gutter --
  vspace %Pixels; #IMPLIED -- vertical gutter --
>
```

```
<!-- USEMAP points to a MAP element which may be in this document
or an external document, although the latter is not widely supported -->
```

<!--===== OBJECT =====-->

```

<!--
  OBJECT is used to embed objects as part of HTML pages
  PARAM elements should precede other content. SGML mixed content
  model technicality precludes specifying this formally ...
-->

<!ELEMENT OBJECT - - (PARAM | %flow;)*
  -- generic embedded object -->
<!ATTLIST OBJECT
  %attrs;                -- %coreattrs, %i18n, %events --
  declare      (declare)      #IMPLIED  -- declare but don't instantiate flag --
  classid      %URI;          #IMPLIED  -- identifies an implementation --
  codebase     %URI;          #IMPLIED  -- base URI for classid, data, archive--
  data         %URI;          #IMPLIED  -- reference to object's data --
  type         %ContentType;  #IMPLIED  -- content type for data --
  codetype     %ContentType;  #IMPLIED  -- content type for code --
  archive      CDATA          #IMPLIED  -- space-separated list of URIs --
  standby      %Text;        #IMPLIED  -- message to show while loading --
  height       %Length;      #IMPLIED  -- override height --
  width        %Length;      #IMPLIED  -- override width --
  usemap       %URI;          #IMPLIED  -- use client-side image map --
  name         CDATA          #IMPLIED  -- submit as part of form --
  tabindex     NUMBER        #IMPLIED  -- position in tabbing order --
  align        %IAAlign;     #IMPLIED  -- vertical or horizontal alignment --
  border       %Pixels;      #IMPLIED  -- link border width --
  hspace       %Pixels;      #IMPLIED  -- horizontal gutter --
  vspace       %Pixels;      #IMPLIED  -- vertical gutter --
  %reserved;   -- reserved for possible future use --
>

```

```

<!ELEMENT PARAM - O EMPTY                -- named property value -->
<!ATTLIST PARAM
  id      ID      #IMPLIED  -- document-wide unique id --
  name    CDATA   #REQUIRED  -- property name --
  value   CDATA   #IMPLIED  -- property value --
  valuetype (DATA|REF|OBJECT) DATA -- How to interpret value --
  type    %ContentType; #IMPLIED  -- content type for value
                                     when valuetype=ref --
>

```

```

<!--===== Java APPLETT =====>
<!--
  One of code or object attributes must be present.
  Place PARAM elements before other content.
-->

```

```

<!ELEMENT APPLETT - - (PARAM | %flow;)* -- Java applet -->
<!ATTLIST APPLETT
  %coreattrs;                -- id, class, style, title --
  codebase     %URI;          #IMPLIED  -- optional base URI for applet --
  archive      CDATA          #IMPLIED  -- comma-separated archive list --
  code         CDATA          #IMPLIED  -- applet class file --
  object       CDATA          #IMPLIED  -- serialized applet file --
  alt          %Text;        #IMPLIED  -- short description --
  name         CDATA          #IMPLIED  -- allows applets to find each other --
  width        %Length;      #REQUIRED  -- initial width --
  height       %Length;      #REQUIRED  -- initial height --
  align        %IAAlign;     #IMPLIED  -- vertical or horizontal alignment --
  hspace       %Pixels;      #IMPLIED  -- horizontal gutter --
  vspace       %Pixels;      #IMPLIED  -- vertical gutter --
>

```

```

<!--===== Horizontal Rule =====>

```

```

<!ELEMENT HR - O EMPTY -- horizontal rule -->
<!ATTLIST HR

```

```

%attrs;                                -- %coreattrs, %i18n, %events --
align      (left|center|right) #IMPLIED
noshade    (noshade)           #IMPLIED
size       %Pixels;            #IMPLIED
width      %Length;            #IMPLIED
>

```

<!--===== Paragraphs =====>

```

<!ELEMENT P - O (%inline;)*           -- paragraph -->
<!ATTLIST P
  %attrs;                                -- %coreattrs, %i18n, %events --
  %align;                                -- align, text alignment --
>

```

<!--===== Headings =====>

```

<!--
  There are six levels of headings from H1 (the most important)
  to H6 (the least important).
-->

```

```

<!ELEMENT (%heading;) - - (%inline;)* -- heading -->
<!ATTLIST (%heading;)
  %attrs;                                -- %coreattrs, %i18n, %events --
  %align;                                -- align, text alignment --
>

```

<!--===== Preformatted Text =====>

```

<!-- excludes markup for images and changes in font size -->
<!ENTITY % pre.exclusion "IMG|OBJECT|APPLET|BIG|SMALL|SUB|SUP|FONT|BASEFONT">

<!ELEMENT PRE - - (%inline;)* -(%pre.exclusion;) -- preformatted text -->
<!ATTLIST PRE
  %attrs;                                -- %coreattrs, %i18n, %events --
  width      NUMBER                      #IMPLIED
>

```

<!--===== Inline Quotes =====>

```

<!ELEMENT Q - - (%inline;)*           -- short inline quotation -->
<!ATTLIST Q
  %attrs;                                -- %coreattrs, %i18n, %events --
  cite       %URI;                      #IMPLIED -- URI for source document or msg --
>

```

<!--===== Block-like Quotes =====>

```

<!ELEMENT BLOCKQUOTE - - (%flow;)*    -- long quotation -->
<!ATTLIST BLOCKQUOTE
  %attrs;                                -- %coreattrs, %i18n, %events --
  cite       %URI;                      #IMPLIED -- URI for source document or msg --
>

```

<!--===== Inserted/Deleted Text =====>

```

<!-- INS/DEL are handled by inclusion on BODY -->
<!ELEMENT (INS|DEL) - - (%flow;)*     -- inserted text, deleted text -->
<!ATTLIST (INS|DEL)
  %attrs;                                -- %coreattrs, %i18n, %events --
  cite       %URI;                      #IMPLIED -- info on reason for change --
  datetime   %Datetime;                 #IMPLIED -- date and time of change --
>

```


<!--===== Lists =====>

<!-- definition lists - DT for term, DD for its definition -->

```
<!ELEMENT DL - - (DT|DD)+ -- definition list -->
<!ATTLIST DL
  %attrs; -- %coreattrs, %i18n, %events --
  compact (compact) #IMPLIED -- reduced interitem spacing --
>
```

```
<!ELEMENT DT - O (%inline;)* -- definition term -->
<!ELEMENT DD - O (%flow;)* -- definition description -->
<!ATTLIST (DT|DD)
  %attrs; -- %coreattrs, %i18n, %events --
>
```

<!-- Ordered lists (OL) Numbering style

| | | |
|---|----------------|-----------------|
| 1 | arabic numbers | 1, 2, 3, ... |
| a | lower alpha | a, b, c, ... |
| A | upper alpha | A, B, C, ... |
| i | lower roman | i, ii, iii, ... |
| I | upper roman | I, II, III, ... |

The style is applied to the sequence number which by default is reset to 1 for the first list item in an ordered list.

This can't be expressed directly in SGML due to case folding.

-->

```
<!ENTITY % OLStyle "CDATA" -- constrained to: "(1|a|A|i|I)" -->
```

```
<!ELEMENT OL - - (LI)+ -- ordered list -->
<!ATTLIST OL
  %attrs; -- %coreattrs, %i18n, %events --
  type %OLStyle; #IMPLIED -- numbering style --
  compact (compact) #IMPLIED -- reduced interitem spacing --
  start NUMBER #IMPLIED -- starting sequence number --
>
```

<!-- Unordered Lists (UL) bullet styles -->

```
<!ENTITY % ULStyle "(disc|square|circle)">
```

```
<!ELEMENT UL - - (LI)+ -- unordered list -->
<!ATTLIST UL
  %attrs; -- %coreattrs, %i18n, %events --
  type %ULStyle; #IMPLIED -- bullet style --
  compact (compact) #IMPLIED -- reduced interitem spacing --
>
```

```
<!ELEMENT (DIR|MENU) - - (LI)+ -(%block;) -- directory list, menu list -->
```

```
<!ATTLIST DIR
  %attrs; -- %coreattrs, %i18n, %events --
  compact (compact) #IMPLIED -- reduced interitem spacing --
>
```

```
<!ATTLIST MENU
  %attrs; -- %coreattrs, %i18n, %events --
  compact (compact) #IMPLIED -- reduced interitem spacing --
>
```

```
<!ENTITY % LISTyle "CDATA" -- constrained to: "(%ULStyle;|%OLStyle;)" -->
```

```
<!ELEMENT LI - O (%flow;)* -- list item -->
<!ATTLIST LI
```

```

%attrs; -- %coreattrs, %i18n, %events --
type      %LIStyle;      #IMPLIED -- list item style --
value     NUMBER         #IMPLIED -- reset sequence number --
>

```

<!--===== Forms =====>

<!ELEMENT FORM - - (%flow;)* -(FORM) -- interactive form -->

<!ATTLIST FORM

```

%attrs; -- %coreattrs, %i18n, %events --
action    %URI;          #REQUIRED -- server-side form handler --
method    (GET|POST)     GET         -- HTTP method used to submit the form--
enctype   %ContentType; "application/x-www-form-urlencoded"
accept    %ContentTypes; #IMPLIED -- list of MIME types for file upload --
name      CDATA          #IMPLIED -- name of form for scripting --
onsubmit  %Script;       #IMPLIED -- the form was submitted --
onreset   %Script;       #IMPLIED -- the form was reset --
target    %FrameTarget; #IMPLIED -- render in this frame --
accept-charset %Charsets; #IMPLIED -- list of supported charsets --
>

```

<!-- Each label must not contain more than ONE field -->

<!ELEMENT LABEL - - (%inline;)* -(LABEL) -- form field label text -->

<!ATTLIST LABEL

```

%attrs; -- %coreattrs, %i18n, %events --
for      IDREF           #IMPLIED -- matches field ID value --
accesskey %Character;    #IMPLIED -- accessibility key character --
onfocus %Script;       #IMPLIED -- the element got the focus --
onblur   %Script;       #IMPLIED -- the element lost the focus --
>

```

<!ENTITY % InputType

```

"(TEXT | PASSWORD | CHECKBOX |
  RADIO | SUBMIT | RESET |
  FILE | HIDDEN | IMAGE | BUTTON)"
>

```

<!-- attribute name required for all but submit and reset -->

<!ELEMENT INPUT - O EMPTY -- form control -->

<!ATTLIST INPUT

```

%attrs; -- %coreattrs, %i18n, %events --
type      %InputType;    TEXT      -- what kind of widget is needed --
name      CDATA          #IMPLIED -- submit as part of form --
value     CDATA          #IMPLIED -- Specify for radio buttons and checkboxes --
checked   (checked)     #IMPLIED -- for radio buttons and check boxes --
disabled  (disabled)    #IMPLIED -- unavailable in this context --
readonly  (readonly)    #IMPLIED -- for text and passwd --
size      CDATA          #IMPLIED -- specific to each type of field --
maxlength NUMBER        #IMPLIED -- max chars for text fields --
src       %URI;          #IMPLIED -- for fields with images --
alt       CDATA          #IMPLIED -- short description --
usemap    %URI;          #IMPLIED -- use client-side image map --
ismap     (ismap)       #IMPLIED -- use server-side image map --
tabindex  NUMBER        #IMPLIED -- position in tabbing order --
accesskey %Character;    #IMPLIED -- accessibility key character --
onfocus  %Script;       #IMPLIED -- the element got the focus --
onblur    %Script;       #IMPLIED -- the element lost the focus --
onselect  %Script;       #IMPLIED -- some text was selected --
onchange  %Script;       #IMPLIED -- the element value was changed --
accept    %ContentTypes; #IMPLIED -- list of MIME types for file upload --
align     %IAAlign;     #IMPLIED -- vertical or horizontal alignment --
%reserved; -- reserved for possible future use --
>

```

<!ELEMENT SELECT - - (OPTGROUP|OPTION)+ -- option selector -->

<!ATTLIST SELECT

```

%attrs; -- %coreattrs, %i18n, %events --
name CDATA #IMPLIED -- field name --
size NUMBER #IMPLIED -- rows visible --
multiple (multiple) #IMPLIED -- default is single selection --
disabled (disabled) #IMPLIED -- unavailable in this context --
tabindex NUMBER #IMPLIED -- position in tabbing order --
onfocus %Script; #IMPLIED -- the element got the focus --
onblur %Script; #IMPLIED -- the element lost the focus --
onchange %Script; #IMPLIED -- the element value was changed --
%reserved; -- reserved for possible future use --
>

```

<!ELEMENT OPTGROUP - - (OPTION)+ -- option group -->

<!ATTLIST OPTGROUP

```

%attrs; -- %coreattrs, %i18n, %events --
disabled (disabled) #IMPLIED -- unavailable in this context --
label %Text; #REQUIRED -- for use in hierarchical menus --
>

```

<!ELEMENT OPTION - O (#PCDATA) -- selectable choice -->

<!ATTLIST OPTION

```

%attrs; -- %coreattrs, %i18n, %events --
selected (selected) #IMPLIED
disabled (disabled) #IMPLIED -- unavailable in this context --
label %Text; #IMPLIED -- for use in hierarchical menus --
value CDATA #IMPLIED -- defaults to element content --
>

```

<!ELEMENT TEXTAREA - - (#PCDATA) -- multi-line text field -->

<!ATTLIST TEXTAREA

```

%attrs; -- %coreattrs, %i18n, %events --
name CDATA #IMPLIED
rows NUMBER #REQUIRED
cols NUMBER #REQUIRED
disabled (disabled) #IMPLIED -- unavailable in this context --
readonly (readonly) #IMPLIED
tabindex NUMBER #IMPLIED -- position in tabbing order --
accesskey %Character; #IMPLIED -- accessibility key character --
onfocus %Script; #IMPLIED -- the element got the focus --
onblur %Script; #IMPLIED -- the element lost the focus --
onselect %Script; #IMPLIED -- some text was selected --
onchange %Script; #IMPLIED -- the element value was changed --
%reserved; -- reserved for possible future use --
>

```

```

<!--
#PCDATA is to solve the mixed content problem,
per specification only whitespace is allowed there!
-->

```

<!ELEMENT FIELDSET - - (#PCDATA,LEGEND,(%flow;)* -- form control group -->

<!ATTLIST FIELDSET

```

%attrs; -- %coreattrs, %i18n, %events --
>

```

<!ELEMENT LEGEND - - (%inline;)* -- fieldset legend -->

<!ENTITY % LAlign "(top|bottom|left|right)">

<!ATTLIST LEGEND

```

%attrs; -- %coreattrs, %i18n, %events --
accesskey %Character; #IMPLIED -- accessibility key character --
align %LAlign; #IMPLIED -- relative to fieldset --
>

```

<!ELEMENT BUTTON - -

(%flow;)* -(A|%formctrl;|FORM|ISINDEX|FIELDSET|IFRAME)

```

-- push button -->
<!ATTLIST BUTTON
  %attrs;                -- %coreattrs, %i18n, %events --
  name                    CDATA                #IMPLIED
  value                   CDATA                #IMPLIED -- sent to server when submitted --
  type                    (button|submit|reset) submit -- for use as form button --
  disabled               (disabled)          #IMPLIED -- unavailable in this context --
  tabindex               NUMBER              #IMPLIED -- position in tabbing order --
  accesskey              %Character;         #IMPLIED -- accessibility key character --
  onfocus               %Script;           #IMPLIED -- the element got the focus --
  onblur                 %Script;           #IMPLIED -- the element lost the focus --
  %reserved;            -- reserved for possible future use --
>

```

<!--===== Tables =====-->

<!-- IETF HTML table standard, see [RFC1942] -->

<!-- The BORDER attribute sets the thickness of the frame around the table. The default units are screen pixels.

The FRAME attribute specifies which parts of the frame around the table should be rendered. The values are not the same as CALS to avoid a name clash with the VALIGN attribute.

The value "border" is included for backwards compatibility with <TABLE BORDER> which yields frame=border and border=implied For <TABLE BORDER=1> you get border=1 and frame=implied. In this case, it is appropriate to treat this as frame=border for backwards compatibility with deployed browsers.

```
-->
<!ENTITY % TFrame "(void|above|below|hsides|lhs|rhs|vsides|box|border)">
```

<!-- The RULES attribute defines which rules to draw between cells:

If RULES is absent then assume:
"none" if BORDER is absent or BORDER=0 otherwise "all"

```
-->
<!ENTITY % TRules "(none | groups | rows | cols | all)">
```

<!-- horizontal placement of table relative to document -->
<!ENTITY % TAlign "(left|center|right)">

<!-- horizontal alignment attributes for cell contents -->
<!ENTITY % cellhalign
"align (left|center|right|justify|char) #IMPLIED
char %Character; #IMPLIED -- alignment char, e.g. char=':' --
charoff %Length; #IMPLIED -- offset for alignment char --"
>

<!-- vertical alignment attributes for cell contents -->
<!ENTITY % cellvalign
"valign (top|middle|bottom|baseline) #IMPLIED"
>

```

<!ELEMENT TABLE - -
  (CAPTION?, (COL*|COLGROUP*), THEAD?, TFOOT?, TBODY+)>
<!ELEMENT CAPTION - - (%inline;)* -- table caption -->
<!ELEMENT THEAD - O (TR)+ -- table header -->
<!ELEMENT TFOOT - O (TR)+ -- table footer -->
<!ELEMENT TBODY O O (TR)+ -- table body -->
<!ELEMENT COLGROUP - O (COL)* -- table column group -->

```

```

<!ELEMENT COL          - O EMPTY          -- table column -->
<!ELEMENT TR          - O (TH|TD)+       -- table row -->
<!ELEMENT (TH|TD)     - O (%flow;)*      -- table header cell, table data cell-->

<!ATTLIST TABLE      -- table element --
  %attrs;              -- %coreattrs, %il8n, %events --
  summary              %Text;             #IMPLIED -- purpose/structure for speech output--
  width                %Length;           #IMPLIED -- table width --
  border               %Pixels;           #IMPLIED -- controls frame width around table --
  frame                %TFrame;           #IMPLIED -- which parts of frame to render --
  rules                %TRules;           #IMPLIED -- rulings between rows and cols --
  cellspacing          %Length;           #IMPLIED -- spacing between cells --
  cellpadding          %Length;           #IMPLIED -- spacing within cells --
  align                %TAlign;           #IMPLIED -- table position relative to window --
  bgcolor              %Color;            #IMPLIED -- background color for cells --
  %reserved;           -- reserved for possible future use --
  datapagesize         CDATA              #IMPLIED -- reserved for possible future use --
>

```

```

<!ENTITY % CAlign "(top|bottom|left|right)">

```

```

<!ATTLIST CAPTION     -- %coreattrs, %il8n, %events --
  %attrs;
  align                %CAlign;           #IMPLIED -- relative to table --
>

```

<!--
COLGROUP groups a set of COL elements. It allows you to group several semantically related columns together.
-->

```

<!ATTLIST COLGROUP   -- %coreattrs, %il8n, %events --
  %attrs;
  span                 NUMBER              1      -- default number of columns in group --
  width                %MultiLength;      #IMPLIED -- default width for enclosed COLs --
  %cellhalign;        -- horizontal alignment in cells --
  %cellvalign;        -- vertical alignment in cells --
>

```

<!--
COL elements define the alignment properties for cells in one or more columns.

The WIDTH attribute specifies the width of the columns, e.g.

```

width=64           width in screen pixels
width=0.5*         relative width of 0.5

```

The SPAN attribute causes the attributes of one COL element to apply to more than one column.

```

-->
<!ATTLIST COL        -- column groups and properties --
  %attrs;            -- %coreattrs, %il8n, %events --
  span               NUMBER              1      -- COL attributes affect N columns --
  width              %MultiLength;      #IMPLIED -- column width specification --
  %cellhalign;      -- horizontal alignment in cells --
  %cellvalign;      -- vertical alignment in cells --
>

```

<!--
Use THEAD to duplicate headers when breaking table across page boundaries, or for static headers when TBODY sections are rendered in scrolling panel.

Use TFOOT to duplicate footers when breaking table across page boundaries, or for static footers when

TBODY sections are rendered in scrolling panel.

Use multiple TBODY sections when rules are needed between groups of table rows.

```
-->
<!ATTLIST (THEAD|TBODY|TFOOT)          -- table section --
  %attrs;                               -- %coreattrs, %i18n, %events --
  %cellhalign;                          -- horizontal alignment in cells --
  %cellvalign;                           -- vertical alignment in cells --
  >

<!ATTLIST TR                            -- table row --
  %attrs;                               -- %coreattrs, %i18n, %events --
  %cellhalign;                          -- horizontal alignment in cells --
  %cellvalign;                           -- vertical alignment in cells --
  bgcolor      %Color;                   #IMPLIED -- background color for row --
  >

<!-- Scope is simpler than headers attribute for common tables -->
<!ENTITY % Scope "(row|col|rowgroup|colgroup)">

<!-- TH is for headers, TD for data, but for cells acting as both use TD -->
<!ATTLIST (TH|TD)                       -- header or data cell --
  %attrs;                               -- %coreattrs, %i18n, %events --
  abbr        %Text;                     #IMPLIED -- abbreviation for header cell --
  axis        CDATA                      #IMPLIED -- comma-separated list of related headers--
  headers     IDREFS                     #IMPLIED -- list of id's for header cells --
  scope       %Scope;                   #IMPLIED -- scope covered by header cells --
  rowspan     NUMBER                     1       -- number of rows spanned by cell --
  colspan     NUMBER                     1       -- number of cols spanned by cell --
  %cellhalign;                          -- horizontal alignment in cells --
  %cellvalign;                          -- vertical alignment in cells --
  nowrap      (nowrap)                   #IMPLIED -- suppress word wrap --
  bgcolor     %Color;                     #IMPLIED -- cell background color --
  width       %Length;                   #IMPLIED -- width for cell --
  height      %Length;                   #IMPLIED -- height for cell --
  >
```

<!--===== Document Frames =====-->

<!-- The content model for HTML documents depends on whether the HEAD is followed by a FRAMESET or BODY element. The widespread omission of the BODY start tag makes it impractical to define the content model without the use of a marked section.

```
-->

<![ %HTML.Frameset; [
<!ELEMENT FRAMESET - - ((FRAMESET|FRAME)+ & NOFRAMES?) -- window subdivision-->
<!ATTLIST FRAMESET
  %coreattrs;                            -- id, class, style, title --
  rows      %MultiLengths; #IMPLIED     -- list of lengths,
                                                default: 100% (1 row) --
  cols      %MultiLengths; #IMPLIED     -- list of lengths,
                                                default: 100% (1 col) --
  onload    %Script;                     #IMPLIED -- all the frames have been loaded --
  onunload  %Script;                     #IMPLIED -- all the frames have been removed --
  >
]]>
```

```
<![ %HTML.Frameset; [
<!-- reserved frame names start with "_" otherwise starts with letter -->
<!ELEMENT FRAME - O EMPTY                -- subwindow -->
<!ATTLIST FRAME
```

```

%coreattrs; -- id, class, style, title --
longdesc %URI; #IMPLIED -- link to long description
-- (complements title) --
name CDATA #IMPLIED -- name of frame for targetting --
src %URI; #IMPLIED -- source of frame content --
frameborder (1|0) 1 -- request frame borders? --
marginwidth %Pixels; #IMPLIED -- margin widths in pixels --
marginheight %Pixels; #IMPLIED -- margin height in pixels --
noresize (noresize) #IMPLIED -- allow users to resize frames? --
scrolling (yes|no|auto) auto -- scrollbar or none --
>
]]>

<!ELEMENT IFRAME - - (%flow;)* -- inline subwindow -->
<!ATTLIST IFRAME
%coreattrs; -- id, class, style, title --
longdesc %URI; #IMPLIED -- link to long description
-- (complements title) --
name CDATA #IMPLIED -- name of frame for targetting --
src %URI; #IMPLIED -- source of frame content --
frameborder (1|0) 1 -- request frame borders? --
marginwidth %Pixels; #IMPLIED -- margin widths in pixels --
marginheight %Pixels; #IMPLIED -- margin height in pixels --
scrolling (yes|no|auto) auto -- scrollbar or none --
align %IAAlign; #IMPLIED -- vertical or horizontal alignment --
height %Length; #IMPLIED -- frame height --
width %Length; #IMPLIED -- frame width --
>

<![ %HTML.Frameset; [
<!ENTITY % noframes.content "(BODY) -(NOFRAMES)">
]]>

<!ENTITY % noframes.content "(%flow;)*">

<!ELEMENT NOFRAMES - - %noframes.content;
-- alternate content container for non frame-based rendering -->
<!ATTLIST NOFRAMES
%attrs; -- %coreattrs, %il8n, %events --
>

<!--===== Document Head =====>
<!-- %head.misc; defined earlier on as "SCRIPT|STYLE|META|LINK|OBJECT" -->
<!ENTITY % head.content "TITLE & ISINDEX? & BASE?">

<!ELEMENT HEAD O O (%head.content;) +(%head.misc;) -- document head -->
<!ATTLIST HEAD
%il8n; -- lang, dir --
profile %URI; #IMPLIED -- named dictionary of meta info --
>

<!-- The TITLE element is not considered part of the flow of text.
It should be displayed, for example as the page header or
window title. Exactly one title is required per document.
-->
<!ELEMENT TITLE - - (#PCDATA) -(%head.misc;) -- document title -->
<!ATTLIST TITLE %il8n>

<!ELEMENT ISINDEX - O EMPTY -- single line prompt -->
<!ATTLIST ISINDEX
%coreattrs; -- id, class, style, title --
%il8n; -- lang, dir --
prompt %Text; #IMPLIED -- prompt message -->

<!ELEMENT BASE - O EMPTY -- document base URI -->

```

```
<!ATTLIST BASE
  href      %URI;          #IMPLIED  -- URI that acts as base URI --
  target    %FrameTarget; #IMPLIED  -- render in this frame --
  >

<!ELEMENT META - O EMPTY -- generic metainformation -->
<!ATTLIST META
  %il8n; -- lang, dir, for use with content --
  http-equiv NAME          #IMPLIED  -- HTTP response header name --
  name       NAME          #IMPLIED  -- metainformation name --
  content    CDATA         #REQUIRED -- associated information --
  scheme     CDATA         #IMPLIED  -- select form of content --
  >

<!ELEMENT STYLE - - %StyleSheet -- style info -->
<!ATTLIST STYLE
  %il8n; -- lang, dir, for use with title --
  type     %ContentType;  #REQUIRED -- content type of style language --
  media    %MediaDesc;   #IMPLIED  -- designed for use with these media --
  title    %Text;        #IMPLIED  -- advisory title --
  >

<!ELEMENT SCRIPT - - %Script; -- script statements -->
<!ATTLIST SCRIPT
  charset   %Charset;    #IMPLIED  -- char encoding of linked resource --
  type      %ContentType; #REQUIRED  -- content type of script language --
  language  CDATA        #IMPLIED  -- predefined script language name --
  src       %URI;        #IMPLIED  -- URI for an external script --
  defer     (defer)      #IMPLIED  -- UA may defer execution of script --
  event     CDATA        #IMPLIED  -- reserved for possible future use --
  for       %URI;        #IMPLIED  -- reserved for possible future use --
  >

<!ELEMENT NOSCRIPT - - (%flow;)*
  -- alternate content container for non script-based rendering -->
<!ATTLIST NOSCRIPT
  %attrs; -- %coreattrs, %il8n, %events --
  >

<!--===== Document Structure =====>
<!ENTITY % version "version CDATA #FIXED '%HTML.Version;'">

<![ %HTML.Frameset; [
<!ENTITY % html.content "HEAD, FRAMESET">
]]>

<!ENTITY % html.content "HEAD, BODY">

<!ELEMENT HTML O O (%html.content;) -- document root element -->
<!ATTLIST HTML
  %il8n; -- lang, dir --
  %version;
  >
```


<!--

This is the HTML 4.01 Frameset DTD, which should be used for documents with frames. This DTD is identical to the HTML 4.01 Transitional DTD except for the content model of the "HTML" element: in frameset documents, the "FRAMESET" element replaces the "BODY" element.

Draft: \$Date: 1999/12/24 23:20:08 \$

Authors:

Dave Raggett <dsr@w3.org>
Arnaud Le Hors <lehors@w3.org>
Ian Jacobs <ij@w3.org>

Further information about HTML 4.01 is available at:

<http://www.w3.org/TR/1999/REC-html401-19991224>.

-->

<!ENTITY % HTML.Version "-//W3C//DTD HTML 4.01 Frameset//EN"

-- Typical usage:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
    "http://www.w3.org/TR/html4/frameset.dtd">
```

```
<html>
```

```
<head>
```

```
...
```

```
</head>
```

```
<frameset>
```

```
...
```

```
</frameset>
```

```
</html>
```

-->

<!ENTITY % HTML.Frameset "INCLUDE">

<!ENTITY % HTML4.dtd PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

%HTML4.dtd;

<!SGML "ISO 8879:1986 (WWW)"

--

SGML Declaration for HyperText Markup Language version HTML 4

With support for the first 17 planes of ISO 10646 and increased limits for tag and literal lengths etc.

--

CHARSET

BASESET "ISO Registration Number 177//CHARSET ISO/IEC 10646-1:1993 UCS-4 with implementation level 3//ESC 2/5 2/15 4/6"

DESCSET 0 9 UNUSED
9 2 9
11 2 UNUSED
13 1 13
14 18 UNUSED
32 95 32
127 1 UNUSED
128 32 UNUSED
160 55136 160
55296 2048 UNUSED -- SURROGATES --
57344 1056768 57344

CAPACITY

SGMLREF
TOTALCAP 150000
GRPCAP 150000
ENTCAP 150000

SCOPE DOCUMENT

SYNTAX

SHUNCHAR CONTROLS 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 127

BASESET "ISO 646IRV:1991//CHARSET International Reference Version (IRV)//ESC 2/8 4/2"

DESCSET 0 128 0

FUNCTION

RE 13
RS 10
SPACE 32
TAB SEPCHAR 9

NAMING

LCNMSTRT ""
UCNMSTRT ""
LCNMCHAR ".-_: "
UCNMCHAR ".-_: "
NAMECASE GENERAL YES
ENTITY NO

DELIM

GENERAL SGMLREF
HCRO "&#x" -- 38 is the number for ampersand --
SHORTREF SGMLREF

NAMES

SGMLREF

QUANTITY

SGMLREF
ATTCNT 60 -- increased --
ATTSPLEN 65536 -- These are the largest values --
LITLEN 65536 -- permitted in the declaration --
NAMELEN 65536 -- Avoid fixed limits in actual --
PILEN 65536 -- implementations of HTML UA's --
TAGLVL 100
TAGLEN 65536
GRPGTCNT 150
GRPCNT 64

FEATURES

MINIMIZE

DATATAG NO
OMITTAG YES
RANK NO
SHORTTAG YES

LINK

SIMPLE NO
IMPLICIT NO
EXPLICIT NO

OTHER

CONCUR NO
SUBDOC NO
FORMAL YES
APPINFO NONE

>

<!-- Special characters for HTML -->

<!-- Character entity set. Typical invocation:

```
<!ENTITY % HTMLspecial PUBLIC
  "-//W3C//ENTITIES Special//EN//HTML">
%HTMLspecial; -->
```

<!-- Portions (C) International Organization for Standardization 1986:

Permission to copy in any form is granted for use with conforming SGML systems and applications as defined in ISO 8879, provided this notice is included in all copies.

-->

<!-- Relevant ISO entity set is given unless names are newly introduced.

New names (i.e., not in ISO 8879 list) do not clash with any existing ISO 8879 entity names. ISO 10646 character numbers are given for each character, in hex. CDATA values are decimal conversions of the ISO 10646 values and refer to the document character set. Names are ISO 10646 names.

-->

<!-- C0 Controls and Basic Latin -->

```
<!ENTITY quot      CDATA "&#34;"      -- quotation mark = APL quote,
                                     U+0022 ISOnum -->
<!ENTITY amp       CDATA "&#38;"      -- ampersand, U+0026 ISOnum -->
<!ENTITY lt        CDATA "&#60;"      -- less-than sign, U+003C ISOnum -->
<!ENTITY gt        CDATA "&#62;"      -- greater-than sign, U+003E ISOnum -->
```

<!-- Latin Extended-A -->

```
<!ENTITY OElig    CDATA "&#338;"      -- latin capital ligature OE,
                                     U+0152 ISolat2 -->
<!ENTITY oelig    CDATA "&#339;"      -- latin small ligature oe, U+0153 ISolat2 -->
<!-- ligature is a misnomer, this is a separate character in some languages -->
<!ENTITY Scaron   CDATA "&#352;"      -- latin capital letter S with caron,
                                     U+0160 ISolat2 -->
<!ENTITY scaron   CDATA "&#353;"      -- latin small letter s with caron,
                                     U+0161 ISolat2 -->
<!ENTITY Yuml     CDATA "&#376;"      -- latin capital letter Y with diaeresis,
                                     U+0178 ISolat2 -->
```

<!-- Spacing Modifier Letters -->

```
<!ENTITY circ     CDATA "&#710;"      -- modifier letter circumflex accent,
                                     U+02C6 ISOpub -->
<!ENTITY tilde    CDATA "&#732;"      -- small tilde, U+02DC ISodia -->
```

<!-- General Punctuation -->

```
<!ENTITY ensp     CDATA "&#8194;"      -- en space, U+2002 ISOpub -->
<!ENTITY emsp     CDATA "&#8195;"      -- em space, U+2003 ISOpub -->
<!ENTITY thinsp   CDATA "&#8201;"      -- thin space, U+2009 ISOpub -->
<!ENTITY zwnj     CDATA "&#8204;"      -- zero width non-joiner,
                                     U+200C NEW RFC 2070 -->
<!ENTITY zwj      CDATA "&#8205;"      -- zero width joiner, U+200D NEW RFC 2070 -->
<!ENTITY lrm      CDATA "&#8206;"      -- left-to-right mark, U+200E NEW RFC 2070 -->
<!ENTITY rlm      CDATA "&#8207;"      -- right-to-left mark, U+200F NEW RFC 2070 -->
<!ENTITY ndash    CDATA "&#8211;"      -- en dash, U+2013 ISOpub -->
<!ENTITY mdash    CDATA "&#8212;"      -- em dash, U+2014 ISOpub -->
<!ENTITY lsquo    CDATA "&#8216;"      -- left single quotation mark,
                                     U+2018 ISOnum -->
<!ENTITY rsquo    CDATA "&#8217;"      -- right single quotation mark,
                                     U+2019 ISOnum -->
<!ENTITY sbquo    CDATA "&#8218;"      -- single low-9 quotation mark, U+201A NEW -->
<!ENTITY ldquo    CDATA "&#8220;"      -- left double quotation mark,
                                     U+201C ISOnum -->
<!ENTITY rdquo    CDATA "&#8221;"      -- right double quotation mark,
```

```
U+201D ISOnum -->
<!ENTITY bdquo    CDATA "&#8222;" -- double low-9 quotation mark, U+201E NEW -->
<!ENTITY dagger   CDATA "&#8224;" -- dagger, U+2020 ISOpub -->
<!ENTITY Dagger   CDATA "&#8225;" -- double dagger, U+2021 ISOpub -->
<!ENTITY permil   CDATA "&#8240;" -- per mille sign, U+2030 ISOtech -->
<!ENTITY lsaquo   CDATA "&#8249;" -- single left-pointing angle quotation mark,
U+2039 ISO proposed -->
<!-- lsaquo is proposed but not yet ISO standardized -->
<!ENTITY rsaquo   CDATA "&#8250;" -- single right-pointing angle quotation mark,
U+203A ISO proposed -->
<!-- rsaquo is proposed but not yet ISO standardized -->
<!ENTITY euro     CDATA "&#8364;" -- euro sign, U+20AC NEW -->
```

<!-- Mathematical, Greek and Symbolic characters for HTML -->

<!-- Character entity set. Typical invocation:

```
<!ENTITY % HTMLsymbol PUBLIC
    "-//W3C//ENTITIES Symbols//EN//HTML">
%HTMLsymbol; -->
```

<!-- Portions (C) International Organization for Standardization 1986:

Permission to copy in any form is granted for use with conforming SGML systems and applications as defined in ISO 8879, provided this notice is included in all copies.

-->

<!-- Relevant ISO entity set is given unless names are newly introduced.

New names (i.e., not in ISO 8879 list) do not clash with any existing ISO 8879 entity names. ISO 10646 character numbers are given for each character, in hex. CDATA values are decimal conversions of the ISO 10646 values and refer to the document character set. Names are ISO 10646 names.

-->

<!-- Latin Extended-B -->

```
<!ENTITY fnof      CDATA "&#402;" -- latin small f with hook = function
                    = florin, U+0192 ISOTech -->
```

<!-- Greek -->

```
<!ENTITY Alpha    CDATA "&#913;" -- greek capital letter alpha, U+0391 -->
```

```
<!ENTITY Beta     CDATA "&#914;" -- greek capital letter beta, U+0392 -->
```

```
<!ENTITY Gamma    CDATA "&#915;" -- greek capital letter gamma,
                    U+0393 ISOgrk3 -->
```

```
<!ENTITY Delta    CDATA "&#916;" -- greek capital letter delta,
                    U+0394 ISOgrk3 -->
```

```
<!ENTITY Epsilon  CDATA "&#917;" -- greek capital letter epsilon, U+0395 -->
```

```
<!ENTITY Zeta     CDATA "&#918;" -- greek capital letter zeta, U+0396 -->
```

```
<!ENTITY Eta      CDATA "&#919;" -- greek capital letter eta, U+0397 -->
```

```
<!ENTITY Theta    CDATA "&#920;" -- greek capital letter theta,
                    U+0398 ISOgrk3 -->
```

```
<!ENTITY Iota     CDATA "&#921;" -- greek capital letter iota, U+0399 -->
```

```
<!ENTITY Kappa    CDATA "&#922;" -- greek capital letter kappa, U+039A -->
```

```
<!ENTITY Lambda   CDATA "&#923;" -- greek capital letter lambda,
                    U+039B ISOgrk3 -->
```

```
<!ENTITY Mu       CDATA "&#924;" -- greek capital letter mu, U+039C -->
```

```
<!ENTITY Nu       CDATA "&#925;" -- greek capital letter nu, U+039D -->
```

```
<!ENTITY Xi       CDATA "&#926;" -- greek capital letter xi, U+039E ISOgrk3 -->
```

```
<!ENTITY Omicron  CDATA "&#927;" -- greek capital letter omicron, U+039F -->
```

```
<!ENTITY Pi       CDATA "&#928;" -- greek capital letter pi, U+03A0 ISOgrk3 -->
```

```
<!ENTITY Rho      CDATA "&#929;" -- greek capital letter rho, U+03A1 -->
```

<!-- there is no Sigmaf, and no U+03A2 character either -->

```
<!ENTITY Sigma    CDATA "&#931;" -- greek capital letter sigma,
                    U+03A3 ISOgrk3 -->
```

```
<!ENTITY Tau      CDATA "&#932;" -- greek capital letter tau, U+03A4 -->
```

```
<!ENTITY Upsilon  CDATA "&#933;" -- greek capital letter upsilon,
                    U+03A5 ISOgrk3 -->
```

```
<!ENTITY Phi      CDATA "&#934;" -- greek capital letter phi,
                    U+03A6 ISOgrk3 -->
```

```
<!ENTITY Chi      CDATA "&#935;" -- greek capital letter chi, U+03A7 -->
```

```
<!ENTITY Psi      CDATA "&#936;" -- greek capital letter psi,
                    U+03A8 ISOgrk3 -->
```

```
<!ENTITY Omega    CDATA "&#937;" -- greek capital letter omega,
                    U+03A9 ISOgrk3 -->
```

```
<!ENTITY alpha    CDATA "&#945;" -- greek small letter alpha,
                    U+03B1 ISOgrk3 -->
```

```
<!ENTITY beta     CDATA "&#946;" -- greek small letter beta, U+03B2 ISOgrk3 -->
```

```
<!ENTITY gamma      CDATA "&#947;" -- greek small letter gamma,
                        U+03B3 ISOgrk3 -->
<!ENTITY delta      CDATA "&#948;" -- greek small letter delta,
                        U+03B4 ISOgrk3 -->
<!ENTITY epsilon    CDATA "&#949;" -- greek small letter epsilon,
                        U+03B5 ISOgrk3 -->
<!ENTITY zeta       CDATA "&#950;" -- greek small letter zeta, U+03B6 ISOgrk3 -->
<!ENTITY eta        CDATA "&#951;" -- greek small letter eta, U+03B7 ISOgrk3 -->
<!ENTITY theta      CDATA "&#952;" -- greek small letter theta,
                        U+03B8 ISOgrk3 -->
<!ENTITY iota       CDATA "&#953;" -- greek small letter iota, U+03B9 ISOgrk3 -->
<!ENTITY kappa      CDATA "&#954;" -- greek small letter kappa,
                        U+03BA ISOgrk3 -->
<!ENTITY lambda     CDATA "&#955;" -- greek small letter lambda,
                        U+03BB ISOgrk3 -->
<!ENTITY mu         CDATA "&#956;" -- greek small letter mu, U+03BC ISOgrk3 -->
<!ENTITY nu         CDATA "&#957;" -- greek small letter nu, U+03BD ISOgrk3 -->
<!ENTITY xi         CDATA "&#958;" -- greek small letter xi, U+03BE ISOgrk3 -->
<!ENTITY omicron    CDATA "&#959;" -- greek small letter omicron, U+03BF NEW -->
<!ENTITY pi         CDATA "&#960;" -- greek small letter pi, U+03C0 ISOgrk3 -->
<!ENTITY rho        CDATA "&#961;" -- greek small letter rho, U+03C1 ISOgrk3 -->
<!ENTITY sigmaf     CDATA "&#962;" -- greek small letter final sigma,
                        U+03C2 ISOgrk3 -->
<!ENTITY sigma      CDATA "&#963;" -- greek small letter sigma,
                        U+03C3 ISOgrk3 -->
<!ENTITY tau        CDATA "&#964;" -- greek small letter tau, U+03C4 ISOgrk3 -->
<!ENTITY upsilon    CDATA "&#965;" -- greek small letter upsilon,
                        U+03C5 ISOgrk3 -->
<!ENTITY phi        CDATA "&#966;" -- greek small letter phi, U+03C6 ISOgrk3 -->
<!ENTITY chi        CDATA "&#967;" -- greek small letter chi, U+03C7 ISOgrk3 -->
<!ENTITY psi        CDATA "&#968;" -- greek small letter psi, U+03C8 ISOgrk3 -->
<!ENTITY omega      CDATA "&#969;" -- greek small letter omega,
                        U+03C9 ISOgrk3 -->
<!ENTITY thetasym   CDATA "&#977;" -- greek small letter theta symbol,
                        U+03D1 NEW -->
<!ENTITY upsiah     CDATA "&#978;" -- greek upsilon with hook symbol,
                        U+03D2 NEW -->
<!ENTITY piv        CDATA "&#982;" -- greek pi symbol, U+03D6 ISOgrk3 -->

<!-- General Punctuation -->
<!ENTITY bull       CDATA "&#8226;" -- bullet = black small circle,
                        U+2022 ISOpub -->
<!-- bullet is NOT the same as bullet operator, U+2219 -->
<!ENTITY hellip    CDATA "&#8230;" -- horizontal ellipsis = three dot leader,
                        U+2026 ISOpub -->
<!ENTITY prime     CDATA "&#8242;" -- prime = minutes = feet, U+2032 ISOftech -->
<!ENTITY Prime     CDATA "&#8243;" -- double prime = seconds = inches,
                        U+2033 ISOftech -->
<!ENTITY oline     CDATA "&#8254;" -- overline = spacing overscore,
                        U+203E NEW -->
<!ENTITY frasl     CDATA "&#8260;" -- fraction slash, U+2044 NEW -->

<!-- Letterlike Symbols -->
<!ENTITY weierp    CDATA "&#8472;" -- script capital P = power set
                        = Weierstrass p, U+2118 ISOamso -->
<!ENTITY image     CDATA "&#8465;" -- blackletter capital I = imaginary part,
                        U+2111 ISOamso -->
<!ENTITY real      CDATA "&#8476;" -- blackletter capital R = real part symbol,
                        U+211C ISOamso -->
<!ENTITY trade     CDATA "&#8482;" -- trade mark sign, U+2122 ISOnum -->
<!ENTITY alefsym   CDATA "&#8501;" -- alef symbol = first transfinite cardinal,
                        U+2135 NEW -->
<!-- alef symbol is NOT the same as hebrew letter alef,
      U+05D0 although the same glyph could be used to depict both characters -->
```

```
<!-- Arrows -->
<!ENTITY larr      CDATA "&#8592;" -- leftwards arrow, U+2190 ISOnum -->
<!ENTITY uarr      CDATA "&#8593;" -- upwards arrow, U+2191 ISOnum-->
<!ENTITY rarr      CDATA "&#8594;" -- rightwards arrow, U+2192 ISOnum -->
<!ENTITY darr      CDATA "&#8595;" -- downwards arrow, U+2193 ISOnum -->
<!ENTITY harr      CDATA "&#8596;" -- left right arrow, U+2194 ISOamsa -->
<!ENTITY crarr     CDATA "&#8629;" -- downwards arrow with corner leftwards
                        = carriage return, U+21B5 NEW -->
<!ENTITY lArr      CDATA "&#8656;" -- leftwards double arrow, U+21D0 ISOTECH -->
<!-- ISO 10646 does not say that lArr is the same as the 'is implied by' arrow
      but also does not have any other character for that function. So ? lArr can
      be used for 'is implied by' as ISOTECH suggests -->
<!ENTITY uArr      CDATA "&#8657;" -- upwards double arrow, U+21D1 ISOamsa -->
<!ENTITY rArr      CDATA "&#8658;" -- rightwards double arrow,
                        U+21D2 ISOTECH -->
<!-- ISO 10646 does not say this is the 'implies' character but does not have
      another character with this function so ?
      rArr can be used for 'implies' as ISOTECH suggests -->
<!ENTITY dArr      CDATA "&#8659;" -- downwards double arrow, U+21D3 ISOamsa -->
<!ENTITY hArr      CDATA "&#8660;" -- left right double arrow,
                        U+21D4 ISOamsa -->

<!-- Mathematical Operators -->
<!ENTITY forall    CDATA "&#8704;" -- for all, U+2200 ISOTECH -->
<!ENTITY part      CDATA "&#8706;" -- partial differential, U+2202 ISOTECH -->
<!ENTITY exist     CDATA "&#8707;" -- there exists, U+2203 ISOTECH -->
<!ENTITY empty     CDATA "&#8709;" -- empty set = null set = diameter,
                        U+2205 ISOamsa -->
<!ENTITY nabla     CDATA "&#8711;" -- nabla = backward difference,
                        U+2207 ISOTECH -->
<!ENTITY isin      CDATA "&#8712;" -- element of, U+2208 ISOTECH -->
<!ENTITY notin     CDATA "&#8713;" -- not an element of, U+2209 ISOTECH -->
<!ENTITY ni        CDATA "&#8715;" -- contains as member, U+220B ISOTECH -->
<!-- should there be a more memorable name than 'ni'? -->
<!ENTITY prod      CDATA "&#8719;" -- n-ary product = product sign,
                        U+220F ISOamsb -->
<!-- prod is NOT the same character as U+03A0 'greek capital letter pi' though
      the same glyph might be used for both -->
<!ENTITY sum       CDATA "&#8721;" -- n-ary summation, U+2211 ISOamsb -->
<!-- sum is NOT the same character as U+03A3 'greek capital letter sigma'
      though the same glyph might be used for both -->
<!ENTITY minus     CDATA "&#8722;" -- minus sign, U+2212 ISOTECH -->
<!ENTITY lowast    CDATA "&#8727;" -- asterisk operator, U+2217 ISOTECH -->
<!ENTITY radic     CDATA "&#8730;" -- square root = radical sign,
                        U+221A ISOTECH -->
<!ENTITY prop      CDATA "&#8733;" -- proportional to, U+221D ISOTECH -->
<!ENTITY infin     CDATA "&#8734;" -- infinity, U+221E ISOTECH -->
<!ENTITY ang       CDATA "&#8736;" -- angle, U+2220 ISOamsa -->
<!ENTITY and       CDATA "&#8743;" -- logical and = wedge, U+2227 ISOTECH -->
<!ENTITY or        CDATA "&#8744;" -- logical or = vee, U+2228 ISOTECH -->
<!ENTITY cap       CDATA "&#8745;" -- intersection = cap, U+2229 ISOTECH -->
<!ENTITY cup       CDATA "&#8746;" -- union = cup, U+222A ISOTECH -->
<!ENTITY int       CDATA "&#8747;" -- integral, U+222B ISOTECH -->
<!ENTITY there4    CDATA "&#8756;" -- therefore, U+2234 ISOTECH -->
<!ENTITY sim       CDATA "&#8764;" -- tilde operator = varies with = similar to,
                        U+223C ISOTECH -->
<!-- tilde operator is NOT the same character as the tilde, U+007E,
      although the same glyph might be used to represent both -->
<!ENTITY cong      CDATA "&#8773;" -- approximately equal to, U+2245 ISOTECH -->
<!ENTITY asymp     CDATA "&#8776;" -- almost equal to = asymptotic to,
                        U+2248 ISOamsr -->
<!ENTITY ne        CDATA "&#8800;" -- not equal to, U+2260 ISOTECH -->
<!ENTITY equiv     CDATA "&#8801;" -- identical to, U+2261 ISOTECH -->
<!ENTITY le        CDATA "&#8804;" -- less-than or equal to, U+2264 ISOTECH -->
<!ENTITY ge        CDATA "&#8805;" -- greater-than or equal to,
```



```
U+2265 ISotech -->
<!ENTITY sub      CDATA "&#8834;" -- subset of, U+2282 ISotech -->
<!ENTITY sup      CDATA "&#8835;" -- superset of, U+2283 ISotech -->
<!-- note that nsup, 'not a superset of, U+2283' is not covered by the Symbol
font encoding and is not included. Should it be, for symmetry?
It is in ISOamsn -->
<!ENTITY nsub     CDATA "&#8836;" -- not a subset of, U+2284 ISOamsn -->
<!ENTITY sube     CDATA "&#8838;" -- subset of or equal to, U+2286 ISotech -->
<!ENTITY supe     CDATA "&#8839;" -- superset of or equal to,
U+2287 ISotech -->
<!ENTITY oplus    CDATA "&#8853;" -- circled plus = direct sum,
U+2295 ISOamsb -->
<!ENTITY otimes   CDATA "&#8855;" -- circled times = vector product,
U+2297 ISOamsb -->
<!ENTITY perp     CDATA "&#8869;" -- up tack = orthogonal to = perpendicular,
U+22A5 ISotech -->
<!ENTITY sdot     CDATA "&#8901;" -- dot operator, U+22C5 ISOamsb -->
<!-- dot operator is NOT the same character as U+00B7 middle dot -->

<!-- Miscellaneous Technical -->
<!ENTITY lceil    CDATA "&#8968;" -- left ceiling = apl upstile,
U+2308 ISOamsc -->
<!ENTITY rceil    CDATA "&#8969;" -- right ceiling, U+2309 ISOamsc -->
<!ENTITY lfloor   CDATA "&#8970;" -- left floor = apl downstile,
U+230A ISOamsc -->
<!ENTITY rfloor   CDATA "&#8971;" -- right floor, U+230B ISOamsc -->
<!ENTITY lang     CDATA "&#9001;" -- left-pointing angle bracket = bra,
U+2329 ISotech -->
<!-- lang is NOT the same character as U+003C 'less than'
or U+2039 'single left-pointing angle quotation mark' -->
<!ENTITY rang     CDATA "&#9002;" -- right-pointing angle bracket = ket,
U+232A ISotech -->
<!-- rang is NOT the same character as U+003E 'greater than'
or U+203A 'single right-pointing angle quotation mark' -->

<!-- Geometric Shapes -->
<!ENTITY loz      CDATA "&#9674;" -- lozenge, U+25CA ISOpub -->

<!-- Miscellaneous Symbols -->
<!ENTITY spades   CDATA "&#9824;" -- black spade suit, U+2660 ISOpub -->
<!-- black here seems to mean filled as opposed to hollow -->
<!ENTITY clubs    CDATA "&#9827;" -- black club suit = shamrock,
U+2663 ISOpub -->
<!ENTITY hearts   CDATA "&#9829;" -- black heart suit = valentine,
U+2665 ISOpub -->
<!ENTITY diams    CDATA "&#9830;" -- black diamond suit, U+2666 ISOpub -->
```

```
<!-- Portions (C) International Organization for Standardization 1986
Permission to copy in any form is granted for use with
conforming SGML systems and applications as defined in
ISO 8879, provided this notice is included in all copies.
-->
<!-- Character entity set. Typical invocation:
<!ENTITY % HTMLlat1 PUBLIC
"-//W3C//ENTITIES Latin 1//EN//HTML">
%HTMLlat1;
-->

<!ENTITY nbsp CDATA "&#160;" -- no-break space = non-breaking space,
U+00A0 ISOnum -->
<!ENTITY iexcl CDATA "&#161;" -- inverted exclamation mark, U+00A1 ISOnum -->
<!ENTITY cent CDATA "&#162;" -- cent sign, U+00A2 ISOnum -->
<!ENTITY pound CDATA "&#163;" -- pound sign, U+00A3 ISOnum -->
<!ENTITY curren CDATA "&#164;" -- currency sign, U+00A4 ISOnum -->
<!ENTITY yen CDATA "&#165;" -- yen sign = yuan sign, U+00A5 ISOnum -->
<!ENTITY brvbar CDATA "&#166;" -- broken bar = broken vertical bar,
U+00A6 ISOnum -->
<!ENTITY sect CDATA "&#167;" -- section sign, U+00A7 ISOnum -->
<!ENTITY uml CDATA "&#168;" -- diaeresis = spacing diaeresis,
U+00A8 ISODia -->
<!ENTITY copy CDATA "&#169;" -- copyright sign, U+00A9 ISOnum -->
<!ENTITY ordf CDATA "&#170;" -- feminine ordinal indicator, U+00AA ISOnum -->
<!ENTITY laquo CDATA "&#171;" -- left-pointing double angle quotation mark
= left pointing guillemet, U+00AB ISOnum -->
<!ENTITY not CDATA "&#172;" -- not sign, U+00AC ISOnum -->
<!ENTITY shy CDATA "&#173;" -- soft hyphen = discretionary hyphen,
U+00AD ISOnum -->
<!ENTITY reg CDATA "&#174;" -- registered sign = registered trade mark sign,
U+00AE ISOnum -->
<!ENTITY macr CDATA "&#175;" -- macron = spacing macron = overline
= APL overbar, U+00AF ISODia -->
<!ENTITY deg CDATA "&#176;" -- degree sign, U+00B0 ISOnum -->
<!ENTITY plusmn CDATA "&#177;" -- plus-minus sign = plus-or-minus sign,
U+00B1 ISOnum -->
<!ENTITY sup2 CDATA "&#178;" -- superscript two = superscript digit two
= squared, U+00B2 ISOnum -->
<!ENTITY sup3 CDATA "&#179;" -- superscript three = superscript digit three
= cubed, U+00B3 ISOnum -->
<!ENTITY acute CDATA "&#180;" -- acute accent = spacing acute,
U+00B4 ISODia -->
<!ENTITY micro CDATA "&#181;" -- micro sign, U+00B5 ISOnum -->
<!ENTITY para CDATA "&#182;" -- pilcrow sign = paragraph sign,
U+00B6 ISOnum -->
<!ENTITY middot CDATA "&#183;" -- middle dot = Georgian comma
= Greek middle dot, U+00B7 ISOnum -->
<!ENTITY cedil CDATA "&#184;" -- cedilla = spacing cedilla, U+00B8 ISODia -->
<!ENTITY sup1 CDATA "&#185;" -- superscript one = superscript digit one,
U+00B9 ISOnum -->
<!ENTITY ordm CDATA "&#186;" -- masculine ordinal indicator,
U+00BA ISOnum -->
<!ENTITY raquo CDATA "&#187;" -- right-pointing double angle quotation mark
= right pointing guillemet, U+00BB ISOnum -->
<!ENTITY frac14 CDATA "&#188;" -- vulgar fraction one quarter
= fraction one quarter, U+00BC ISOnum -->
<!ENTITY frac12 CDATA "&#189;" -- vulgar fraction one half
= fraction one half, U+00BD ISOnum -->
<!ENTITY frac34 CDATA "&#190;" -- vulgar fraction three quarters
= fraction three quarters, U+00BE ISOnum -->
<!ENTITY iquest CDATA "&#191;" -- inverted question mark
= turned question mark, U+00BF ISOnum -->
<!ENTITY Agrave CDATA "&#192;" -- latin capital letter A with grave
= latin capital letter A grave,
```

U+00C0 ISolat1 -->
<!ENTITY Aacute CDATA "Á" -- latin capital letter A with acute,
U+00C1 ISolat1 -->
<!ENTITY Acirc CDATA "Â" -- latin capital letter A with circumflex,
U+00C2 ISolat1 -->
<!ENTITY Atilde CDATA "Ã" -- latin capital letter A with tilde,
U+00C3 ISolat1 -->
<!ENTITY Auml CDATA "Ä" -- latin capital letter A with diaeresis,
U+00C4 ISolat1 -->
<!ENTITY Aring CDATA "Å" -- latin capital letter A with ring above
= latin capital letter A ring,
U+00C5 ISolat1 -->
<!ENTITY AElig CDATA "Æ" -- latin capital letter AE
= latin capital ligature AE,
U+00C6 ISolat1 -->
<!ENTITY Ccedil CDATA "Ç" -- latin capital letter C with cedilla,
U+00C7 ISolat1 -->
<!ENTITY Egrave CDATA "È" -- latin capital letter E with grave,
U+00C8 ISolat1 -->
<!ENTITY Eacute CDATA "É" -- latin capital letter E with acute,
U+00C9 ISolat1 -->
<!ENTITY Ecirc CDATA "Ê" -- latin capital letter E with circumflex,
U+00CA ISolat1 -->
<!ENTITY Euml CDATA "Ë" -- latin capital letter E with diaeresis,
U+00CB ISolat1 -->
<!ENTITY Igrave CDATA "Ì" -- latin capital letter I with grave,
U+00CC ISolat1 -->
<!ENTITY Iacute CDATA "Í" -- latin capital letter I with acute,
U+00CD ISolat1 -->
<!ENTITY Icirc CDATA "Î" -- latin capital letter I with circumflex,
U+00CE ISolat1 -->
<!ENTITY Iuml CDATA "Ï" -- latin capital letter I with diaeresis,
U+00CF ISolat1 -->
<!ENTITY ETH CDATA "Ð" -- latin capital letter ETH, U+00D0 ISolat1 -->
<!ENTITY Ntilde CDATA "Ñ" -- latin capital letter N with tilde,
U+00D1 ISolat1 -->
<!ENTITY Ograve CDATA "Ò" -- latin capital letter O with grave,
U+00D2 ISolat1 -->
<!ENTITY Oacute CDATA "Ó" -- latin capital letter O with acute,
U+00D3 ISolat1 -->
<!ENTITY Ocirc CDATA "Ô" -- latin capital letter O with circumflex,
U+00D4 ISolat1 -->
<!ENTITY Otilde CDATA "Õ" -- latin capital letter O with tilde,
U+00D5 ISolat1 -->
<!ENTITY Ouml CDATA "Ö" -- latin capital letter O with diaeresis,
U+00D6 ISolat1 -->
<!ENTITY times CDATA "×" -- multiplication sign, U+00D7 ISOnum -->
<!ENTITY Oslash CDATA "Ø" -- latin capital letter O with stroke
= latin capital letter O slash,
U+00D8 ISolat1 -->
<!ENTITY Ugrave CDATA "Ù" -- latin capital letter U with grave,
U+00D9 ISolat1 -->
<!ENTITY Uacute CDATA "Ú" -- latin capital letter U with acute,
U+00DA ISolat1 -->
<!ENTITY Ucirc CDATA "Û" -- latin capital letter U with circumflex,
U+00DB ISolat1 -->
<!ENTITY Uuml CDATA "Ü" -- latin capital letter U with diaeresis,
U+00DC ISolat1 -->
<!ENTITY Yacute CDATA "Ý" -- latin capital letter Y with acute,
U+00DD ISolat1 -->
<!ENTITY THORN CDATA "Þ" -- latin capital letter THORN,
U+00DE ISolat1 -->
<!ENTITY szlig CDATA "ß" -- latin small letter sharp s = ess-zed,
U+00DF ISolat1 -->
<!ENTITY agrave CDATA "à" -- latin small letter a with grave

= latin small letter a grave,
U+00E0 ISolat1 -->
<!ENTITY aacute CDATA "á" -- latin small letter a with acute,
U+00E1 ISolat1 -->
<!ENTITY acirc CDATA "â" -- latin small letter a with circumflex,
U+00E2 ISolat1 -->
<!ENTITY atilde CDATA "ã" -- latin small letter a with tilde,
U+00E3 ISolat1 -->
<!ENTITY auml CDATA "ä" -- latin small letter a with diaeresis,
U+00E4 ISolat1 -->
<!ENTITY aring CDATA "å" -- latin small letter a with ring above
= latin small letter a ring,
U+00E5 ISolat1 -->
<!ENTITY aelig CDATA "æ" -- latin small letter ae
= latin small ligature ae, U+00E6 ISolat1 -->
<!ENTITY ccedil CDATA "ç" -- latin small letter c with cedilla,
U+00E7 ISolat1 -->
<!ENTITY egrave CDATA "è" -- latin small letter e with grave,
U+00E8 ISolat1 -->
<!ENTITY eacute CDATA "é" -- latin small letter e with acute,
U+00E9 ISolat1 -->
<!ENTITY ecirc CDATA "ê" -- latin small letter e with circumflex,
U+00EA ISolat1 -->
<!ENTITY euml CDATA "ë" -- latin small letter e with diaeresis,
U+00EB ISolat1 -->
<!ENTITY igrave CDATA "ì" -- latin small letter i with grave,
U+00EC ISolat1 -->
<!ENTITY iacute CDATA "í" -- latin small letter i with acute,
U+00ED ISolat1 -->
<!ENTITY icirc CDATA "î" -- latin small letter i with circumflex,
U+00EE ISolat1 -->
<!ENTITY iuml CDATA "ï" -- latin small letter i with diaeresis,
U+00EF ISolat1 -->
<!ENTITY eth CDATA "ð" -- latin small letter eth, U+00F0 ISolat1 -->
<!ENTITY ntilde CDATA "ñ" -- latin small letter n with tilde,
U+00F1 ISolat1 -->
<!ENTITY ograve CDATA "ò" -- latin small letter o with grave,
U+00F2 ISolat1 -->
<!ENTITY oacute CDATA "ó" -- latin small letter o with acute,
U+00F3 ISolat1 -->
<!ENTITY ocirc CDATA "ô" -- latin small letter o with circumflex,
U+00F4 ISolat1 -->
<!ENTITY otilde CDATA "õ" -- latin small letter o with tilde,
U+00F5 ISolat1 -->
<!ENTITY ouml CDATA "ö" -- latin small letter o with diaeresis,
U+00F6 ISolat1 -->
<!ENTITY divide CDATA "÷" -- division sign, U+00F7 ISOnum -->
<!ENTITY oslash CDATA "ø" -- latin small letter o with stroke,
= latin small letter o slash,
U+00F8 ISolat1 -->
<!ENTITY ugrave CDATA "ù" -- latin small letter u with grave,
U+00F9 ISolat1 -->
<!ENTITY uacute CDATA "ú" -- latin small letter u with acute,
U+00FA ISolat1 -->
<!ENTITY ucirc CDATA "û" -- latin small letter u with circumflex,
U+00FB ISolat1 -->
<!ENTITY uuml CDATA "ü" -- latin small letter u with diaeresis,
U+00FC ISolat1 -->
<!ENTITY yacute CDATA "ý" -- latin small letter y with acute,
U+00FD ISolat1 -->
<!ENTITY thorn CDATA "þ" -- latin small letter thorn,
U+00FE ISolat1 -->
<!ENTITY yuml CDATA "ÿ" -- latin small letter y with diaeresis,
U+00FF ISolat1 -->

OVERRIDE YES

```
PUBLIC "-//W3C//DTD HTML 4.01//EN" strict.dtd
PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" loose.dtd
PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" frameset.dtd
PUBLIC "-//W3C//ENTITIES Latin1//EN//HTML" HTMLlat1.ent
PUBLIC "-//W3C//ENTITIES Special//EN//HTML" HTMLspecial.ent
PUBLIC "-//W3C//ENTITIES Symbols//EN//HTML" HTMLsymbol.ent
```