

HTML

The World Wide Web Consortium (W3C)

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

Oficina Española del W3C

<http://www.w3c.es/>

Traducciones

Especificación HTML 4.01

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

Introducción a HTML 4

¿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).

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".

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, más gral.

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:

`José`.

¿Qué es 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.

Breve Historia del HTML

El HTML fue desarrollado originalmente por Tim Berners-Lee y fue popularizado por el navegador Mosaic. 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. Los documentos HTML deberían funcionar bien en diferentes navegadores y plataformas. Gracias a la interoperabilidad se reducen gastos, ya que sólo se debe desarrollar una versión de cada documento.

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 no, etc.

HTML 4

El HTML 4.01, revisión de HTML 4.0, desarrolla el lenguaje 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.

Los documentos pueden 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. 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 texto-voz y separación de palabras, etc.

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.

Tablas

El nuevo modelo de tablas de HTML se basa en [\[RFC1942\]](#). La posibilidad 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.

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.

Hojas de estilo

Las hojas de estilo simplifican el código HTML y lo liberan 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 utilizaban las tablas y las imágenes como medio de organizar la presentación de sus páginas. Al ofrecer las hojas de estilo mecanismos de presentación más potentes, el W3C declarará obsoletos en el futuro muchos de los elementos y atributos de presentación del HTML, anteriores. 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.

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.

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 [\[RDF10\]](#)).


Recomendaciones para creación de documentos con HTML 4: seguir los siguientes principios.

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 la presentación se reduce el coste de servir a un amplio espectro de plataformas, medios, etc. y se facilitan las revisiones del documento.

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.

3. Ayudar con la representación incremental

Mediante un diseño cuidadoso de las tablas y haciendo uso de las nuevas características de representación incremental, los autores pueden ayudar a los agentes de usuario a representar los documentos más rápidamente.



Conformidad: requisitos y recomendaciones

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

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.

DEFINICIONES

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

Autor: es una persona o programa que escribe o genera documentos HTML. Una *herramienta de creación* es un caso especial de autor.

Recomendamos a los autores que escriban documentos que sean conformes con el [DTD estricto](#) mejor que con otros DTDs definidos por esta especificación.

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: es cualquier dispositivo que interprete documentos HTML. Los agentes de usuario incluyen navegadores visuales (de texto o gráficos), 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. Por razones de compatibilidad, recomendamos que las herramientas que interpreten HTML 4 sigan soportando HTML 3.2 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. Para un comportamiento recomendado de manejo de errores, consultar 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 pueden declararse obsoletos en el futuro.

Los agentes de usuario deberían seguir dando soporte a los elementos desaprobados por razones de compatibilidad. 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, [\[CSS1\]](#)).

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.

Estructura de un documento HTML

Un documento HTML 4 se compone de tres partes:

1. una línea que contiene [información sobre la versión](#),
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>
```


1. Información sobre la versión de HTML

Un documento HTML válido declara qué versión de HTML se utiliza en él. La *declaración del tipo de documento* especifica la definición del DTD que se usa en él [ISO8879](#). HTML4.01 especifica tres DTDs, que 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.
- 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).
- El [DTD HTML 4.01 para Documentos con Marcos](#) (Frameset DTD) incluye todo lo que incluye el DTD Transicional más los marcos.

El elemento HTML

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>
```

2. La cabecera del documento

El elemento HEAD: Contiene información sobre el documento actual, como el título, palabras clave que pueden ser útiles para motores de búsqueda, y otros datos que no se consideran parte del contenido del documento.

El elemento TITLE: Todos los documentos HTML **deben** tener un elemento [TITLE](#) en la sección [HEAD](#). Los autores deberían utilizar este elemento 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".

Metadatos: *Marco de Descripción de Recursos(ResourceDescription Framework) del W3C (ver [\[RDF10\]](#)) se convirtió en Recomendación del W3C en febrero de 1999. 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 de maneras. Por ejemplo, para especificar el autor:

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

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. 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.

Ejemplo: declara que el nombre del autor está en francés:

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

El siguiente ejemplo especifica la [codificación de caracteres](#)

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

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:

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

2. El cuerpo del documento

El elemento BODY: El cuerpo de un documento contiene el contenido del documento. Este 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:

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

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

Las hojas de estilo externas (vinculadas) dan flexibilidad para cambiar la presentación sin tener que revisar el fuente HTML:

```
<HEAD>
  <TITLE>Estudio sobre la población</TITLE>
  <LINK rel="stylesheet" type="text/css" href="el.css"
</HEAD>
```

Marcos y cuerpos HTML. Los documentos que contienen marcos reemplazan el elemento BODY con el elemento FRAMESET.

Identificadores de elementos: los elementos `id` y `class`

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 `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.

Agrupación de elementos: los elementos `DIV` y `SPAN`

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"). 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.

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

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. 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...</P>
<DIV class="subseccion" id="habitat-jungla">
<H2>Hábitat</H2>
<P>Los elefantes de la selva no viven en... </P>
</DIV>
</DIV>
```

Podemos decorar esta estructura con información de estilo:

```
<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>
```

Información sobre el idioma y dirección del texto

Estos son 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.

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. 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>  
<LI>Información ordenada.</LI>  
<LI>Definiciones.</LI>  
</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.

Las listas de definiciones, creadas por medio del elemento [DL](#), consisten generalmente en una serie de parejas término/definición. Las listas pueden estar anidadas.

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. Este aspecto estilístico se resuelve con hojas de estilo.

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 larga (atributo [summary](#)) para el provecho de 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). 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 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 permita a los agentes de usuario representar la tabla incrementalmente (es decir, a medida que llegan las filas de la tabla) 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" (elemento [TH](#)) o de "datos" (elemento [TD](#)).

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 posibilidad de representación limitada (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 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](#) y no tablas, para organizar la presentación.

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">
<CAPTION><EM>Una tabla de prueba con celdas fusionadas</EM></CAPTION>
<TR><TH rowspan="2"></TH>
      <TH colspan="2">Media</TH>
      <TH rowspan="2">Ojos<BR>Rojos</TH>
<TR><TH>altura</TH><TH>peso</TH>
<TR><TH>Machos</TH>
      <TD>1.9</TD><TD>0.003</TD><TD>40%</TD>
<TR><TH>Hembras</TH>
      <TD>1.7</TD><TD>0.002</TD><TD>43%</TD>
</TABLE>
```

podría representarse de una manera similar 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%

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.).

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. 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, 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..
```

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 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...
```

Introducción a 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.

Ejemplo: incluir una imagen PNG en un documento

```
<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. 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

El elemento genérico [OBJECT](#) servirá como solución de implementación de los tipos de medios futuros.

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.

Ejemplo: insertar una foto directamente en el documento actual

```
<BODY>
<P>Foto de mi familia en el lago:</P>
<IMG
src="http://www.xx.com/vacaciones/familia.png"
alt="Una foto de mi familia en el lago.">
</BODY>
```

Esta inclusión también puede realizarse con el elemento [OBJECT](#):

```
<BODY>
<P>Foto de mi familia en el lago:<P>
<OBJECT
data="http://www.xx.com/vacaciones/familia.png"
type="image/png">
Una foto de mi familia en el lago.
</OBJECT>
</BODY>
```

Presentación visual de imágenes, objetos, 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 (ej. [border](#) y [align](#))

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.

El atributo [alt](#) especifica un texto alternativo que se representa cuando la imagen no puede ser mostrada. Los agentes de usuario deben representar el texto alternativo cuando no puedan soportar 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"></P>
```

```
</BODY>
```

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 evitar usar imágenes para dar formato a las páginas, y utilizar 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.*

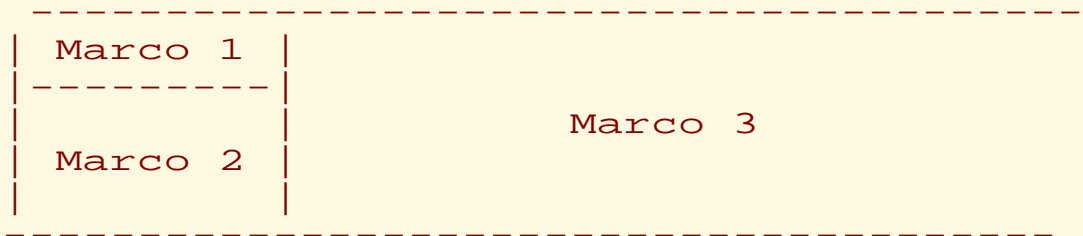
Introducción a los marcos

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.

Ejemplo de 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="cont_marco1.html">xxxx</A>
        <LI><IMG src="cont_marco2.gif"
            alt="Una imagen"></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](#).

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 el elemento [NOFRAMES](#) para proporcionar [contenido alternativo](#).

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.

Marcos en línea: el elemento [IFRAME](#)

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 mostrarlos.

Introducción a los formularios

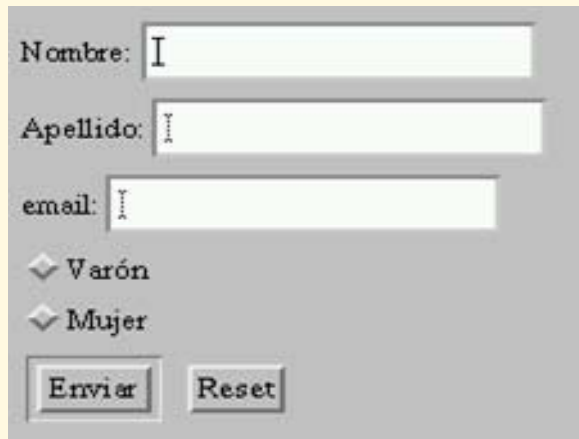
Un formulario HTML es una sección de un documento que contiene contenido normal, código, elementos especiales llamados controles (casillas de verificación (checkboxes), radiobotones (radio buttons), menús, etc.), y rótulos (labels) en esos controles. Los usuarios normalmente "completan" un formulario 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.)

Ejemplo:

```
<FORM action="http://algunsitio.com/prog/nuevo"
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>
```

Cuando se active el botón de envío, el formulario será enviado al programa especificado por el atributo action.

Este formulario podría ser representado de la siguiente manera:



A screenshot of a web form. It contains three text input fields labeled 'Nombre:', 'Apellido:', and 'email:'. Below these are two radio button options: 'Varón' and 'Mujer'. At the bottom are two buttons: 'Enviar' and 'Reset'.

Un formulario puede contener texto y código (párrafos, listas, etc.) además de [controles de formulario](#).

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="Presionar"
onclick="verificar()"></P>
    </FORM>
  </BODY>
```

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").

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.

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.

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 se pueden asociar a un documento:

- 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. Éstos pueden ser asignados a varios elementos por medio de los atributos de [eventos intrínsecos](#).