

Introducción a **XHTML**

AVISO IMPORTANTE

El contenido de este libro es **provisional**, ya que sus contenidos se siguen revisando y actualizando con frecuencia.

En el sitio web **www.librosweb.es** siempre se encuentra disponible la versión más reciente de los contenidos y de este archivo PDF.

Sobre este libro.....	5
1. Introducción	6
1.1. ¿Qué es HTML?.....	6
1.2. Breve historia de HTML.....	6
1.3. Especificación oficial.....	8
1.4. HTML y XHTML.....	8
1.5. HTML y CSS.....	9
2. Características básicas	10
2.1. Estructura de un documento HTML.....	10
2.2. Lenguajes de etiquetas	10
2.3. El primer documento HTML.....	12
2.4. Etiquetas definidas por HTML	15
2.5. Atributos	15
2.6. Elementos HTML.....	16
2.7. Elementos en línea y elementos de bloque	17
2.8. Sintaxis de las etiquetas XHTML.....	19
2.9. Atributos comunes de las etiquetas HTML	21
3. Texto	24
3.1. Texto básico, estructurado y marcado	24
3.2. Estructurar	25
3.2.1. Párrafos.....	25
3.2.2. Secciones.....	27
3.3. Espacios en blanco y nuevas líneas.....	30
3.3.1. Nuevas líneas.....	31
3.3.2. Espacios en blanco	33
3.3.3. Texto preformateado.....	35
3.4. Marcado básico de texto.....	40
3.5. Marcado avanzado de texto	47
4. Enlaces.....	52
4.1. URL.....	52
4.2. Enlaces relativos y absolutos	56
4.3. Enlaces básicos.....	60
4.4. Enlaces avanzados.....	64
4.4.1. Idioma del enlace (hreflang).....	65
4.4.2. Tipo de contenido (type)	66
4.4.3. Tipo de relación (rel y rev).....	66
4.4.4. Codificación de caracteres (charset).....	67
4.5. Otros tipos de enlaces.....	67
4.6. Ejemplos de enlaces habituales	70
4.6.1. Enlace al inicio del sitio web.....	70
4.6.2. Enlace a un email	70

4.6.3. Enlace a un archivo FTP	71
4.6.4. Enlazar varias hojas de estilos CSS.....	72
4.6.5. Enlazar hojas de estilos CSS para diferentes medios.....	72
4.6.6. Enlazar el favicon	72
4.6.7. Enlazar un archivo RSS.....	72
4.6.8. Enlazar hojas de estilos, favicon y RSS	72
4.6.9. Indicar que existe una versión de la página en otro idioma.....	73
4.6.10. Indicar que existe una versión de la página preparada para imprimir.....	73
4.6.11. Indicar que existe una página que es índice de la página actual.....	74
5. Listas.....	75
5.1. Listas no ordenadas.....	75
5.2. Listas ordenadas	77
5.3. Listas de definición	79
6. Imágenes y objetos.....	83
6.1. Imágenes	83
6.2. Mapas de imagen.....	86
6.3. Objetos	89
7. Tablas	94
7.1. Tablas básicas	95
7.2. Tablas avanzadas.....	105
8. Formularios.....	114
8.1. Formularios básicos.....	115
8.2. Formularios avanzados.....	122
8.3. Formularios completos.....	126
9. Estructura y layout.....	133
10. Metainformación	137
10.1. Estructura de la cabecera.....	137
10.2. Metainformación del documento.....	138
10.3. DOCTYPE	140
11. Otras etiquetas y atributos importantes.....	143
11.1. JavaScript.....	143
11.2. Comentarios.....	145
11.3. CSS.....	146
11.4. Iframes.....	147
11.5. Otras etiquetas.....	149
12. Otros conceptos relacionados	153
12.1. Accesibilidad	153
12.1.1. Requisitos del nivel A de accesibilidad	154
12.2. Validación.....	155
12.3. Codificación de caracteres	158
13. Fragmentos de código	161

13.1. Documento XHTML.....	166
13.2. Cabecera XHTML.....	167
13.3. Tabla.....	168
13.4. Formulario.....	169

Sobre este libro...

- Los contenidos de este libro están bajo una licencia Creative Commons Reconocimiento - No Comercial - Sin Obra Derivada 3.0 (<http://creativecommons.org/licenses/by-nc-nd/3.0/deed.es>)
- Esta versión impresa se creó en Junio de 2007 y todavía está **incompleta**. La versión más actualizada de los contenidos de este libro se encuentra disponible en <http://www.librosweb.es/xhtml1>
- Si quieres aportar sugerencias, comentarios, críticas o informar sobre errores, puedes contactarnos en contacto@librosweb.es

Capítulo 1. Introducción

1.1. ¿Qué es HTML?

La definición más simple de HTML:

“HTML es lo que se utiliza para crear todas las páginas web de Internet.”

HTML es un lenguaje que permite definir las partes que forman cada página web. Los diseñadores de páginas web utilizan el lenguaje HTML para crear sus páginas, los programas que utilizan los diseñadores crean páginas escritas con HTML y los navegadores que utilizamos los usuarios muestran las páginas web después de leer su contenido HTML.

Aunque HTML es un lenguaje que utilizan los ordenadores y los programas de diseño, es muy fácil de aprender y escribir por parte de las personas. El nombre HTML está formado por las siglas de *HyperText Markup Language* y más adelante se verá el significado de cada sigla.

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado **W3C** (*World Wide Web Consortium*). Al ser un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de la misma manera en cualquier navegador de cualquier sistema operativo.

El propio **W3C** define el lenguaje HTML como “*un lenguaje reconocido universalmente y que permite publicar información de forma global*”. Desde su primera definición, el lenguaje HTML pasó de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en la mayoría de aplicaciones electrónicas (buscadores, tiendas online, banca electrónica, etc.).

1.2. Breve historia de HTML

El lenguaje HTML fue desarrollado originalmente por **Tim Berners-Lee** mientras trabajaba en el **CERN** (*Organización Europea para la Investigación Nuclear*) a finales de los 80. La explosión en su popularidad se produjo en los años 90, gracias al boom de Internet.

A partir de ese momento, las diferentes versiones de HTML fueron definidas por el organismo **IETF** (*Internet Engineering Task Force*) y el organismo **W3C** (*World Wide Web Consortium*). La versión HTML 2.0 fue publicada por el *Grupo de Trabajo de HTML* del IETF el 22 de Septiembre de 1995 y estandarizaba todas las prácticas comunes en el desarrollo de páginas web hasta el año 1994.

La versión HTML 3.2 se publicó el 14 de Enero de 1997 y es la primera recomendación de HTML publicada por el W3C. Esta revisión incorpora las características principales de las páginas web desarrolladas hasta 1996, como tablas, applets, texto que fluye alrededor de las imágenes, etc.

HTML 4.0 se publicó el 24 de Abril de 1998 (siendo una versión corregida de la publicación original del 18 de Diciembre de 1997) y supone un gran salto desde las versiones anteriores. Entre sus novedades más destacadas se encuentran las hojas de estilos, la posibilidad de incluir pequeños programas o *scripts* en las páginas web, mejora de la accesibilidad de las páginas diseñadas, tablas complejas, mejoras en los formularios, etc.

La última especificación oficial de HTML se publicó el 24 de Diciembre de 1999 y se denomina HTML 4.01. Se trata de una revisión y actualización de la versión HTML 4.0 y por tanto, no incluye novedades significativas.

A partir de la versión HTML 4.01, el W3C no publicará ninguna especificación oficial de HTML. A partir de esa versión, HTML se adapta a las características del lenguaje XML y pasa a denominarse XHTML. La primera versión de XHTML se denomina XHTML 1.0 y se publicó el 26 de Enero de 2000 (y posteriormente se revisó el 1 de Agosto de 2006).

XHTML 1.0 consiste en adaptar HTML 4.01 al lenguaje XML, por lo que mantiene casi todas sus etiquetas y características, pero añade restricciones en su sintaxis y algunos otros elementos propios de XML. XHTML puede considerarse un HTML mejor hecho, más profesional y con unas normas más estrictas.

A partir de ahora, el W3C solo trabajará en la definición de XHTML, por lo que nunca se publicará una recomendación de HTML 5.0, sino que se publicará la recomendación XHTML 2.0. De hecho, la versión XHTML 1.1 ya ha sido publicada y se utiliza como preparación de la futura versión XHTML 2.0 sobre la que se está trabajando activamente y que todavía se encuentra en estado de borrador.

1.3. Especificación oficial

El organismo **W3C** (*World Wide Web Consortium*) elabora las normas que deben seguir los diseñadores de páginas web para crear las páginas HTML. Las normas oficiales están escritas en inglés y se pueden consultar de forma gratuita en las siguientes direcciones:

- Especificación oficial de HTML 4.01: <http://www.w3.org/TR/html401/>
- Especificación oficial de XHTML 1.0: <http://www.w3.org/TR/xhtml1/>

Afortunadamente, no es necesario leer las especificaciones y recomendaciones oficiales de HTML para aprender a diseñar páginas mediante HTML o XHTML. Las normas oficiales están escritas con un lenguaje bastante formal y algunas secciones son difíciles de comprender. Por ello, en los próximos capítulos se explica de forma sencilla y con muchos ejemplos la norma oficial de XHTML.

1.4. HTML y XHTML

El lenguaje XHTML es muy similar al lenguaje HTML. De hecho, XHTML no es más que una adaptación de HTML al lenguaje XML. HTML es descendiente directo del lenguaje SGML, mientras que XHTML lo es del XML (que a su vez también es descendiente de SGML).

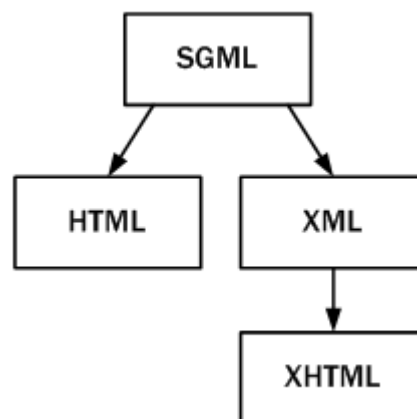


Figura 1.1. Esquema de la evolución de HTML y XHTML

Los documentos XHTML son muy similares a los documentos HTML. Sin embargo, al ser las normas de XHTML más estrictas, son mucho más fáciles de procesar por los diferentes sistemas (ordenadores, móviles, televisores, etc.).

1.5. HTML y CSS

Al principio, las páginas creadas mediante HTML incluían, además de los contenidos propios de la página, información sobre el diseño y aspecto de los contenidos (color de la letra, separación entre párrafos, márgenes laterales de los contenidos, etc.) y también incluían todos los programas (también llamados *scripts*) que manipulaban los contenidos de la página.

Incluir en una misma página HTML los contenidos, su diseño y su programación solo presenta inconvenientes, sobre todo de mantenimiento y actualización de la página. Normalmente, los contenidos y el diseño de la página son tareas que realizan diferentes personas, por lo que siempre que sea posible es conveniente separar los contenidos y su presentación.

CSS es el mecanismo que permite separar completamente los contenidos definidos mediante XHTML y la presentación de esos contenidos en la pantalla del usuario. Una ventaja añadida de la separación de contenidos y presentación es que los documentos XHTML definidos son más flexibles, ya que se adaptan mejor a las diferentes plataformas (pantallas de ordenador, pantallas de móviles, impresoras, dispositivos utilizados por personas discapacitadas, etc.).

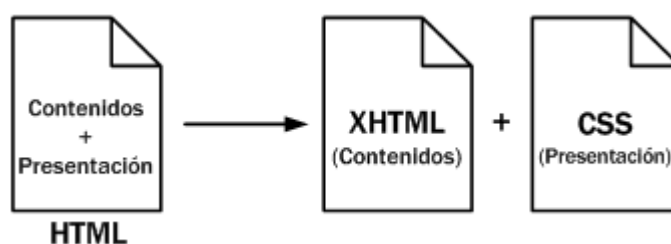


Figura 1.2. Esquema de la separación de los contenidos y su presentación

Capítulo 2. Características básicas

2.1. Estructura de un documento HTML

Todos los documentos HTML están divididos en 2 partes: la cabecera y el cuerpo. En la cabecera se incluye información sobre el propio documento o página web (como por ejemplo el idioma de la página y el título del documento). El cuerpo del documento agrupa todos los contenidos del propio documento (párrafos de texto, imágenes, titulares, etc.).

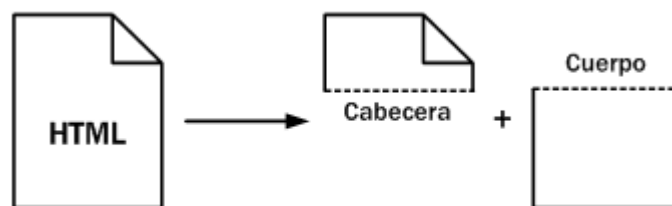


Figura 2.1. Esquema de las partes que forman un documento HTML

El cuerpo (llamado *body* en inglés) contiene todo lo que el usuario ve en su pantalla y la cabecera (llamada *head* en inglés) contiene todo lo que no se ve (con una única excepción que se verá más adelante: el título de la página).

2.2. Lenguajes de etiquetas

Uno de los problemas iniciales a los que se tuvieron que enfrentar los informáticos fue el de cómo almacenar y cómo transmitir archivos con información. Como los primeros archivos solo contenían texto, encontraron una solución muy sencilla: codificar las letras del alfabeto y transformarlas en números.

De esta forma, para almacenar un texto en un archivo electrónico, se utiliza una tabla de conversión que transforma cada carácter en un número. Una vez almacenada la secuencia de números, el contenido del archivo se puede recuperar realizando el proceso inverso.

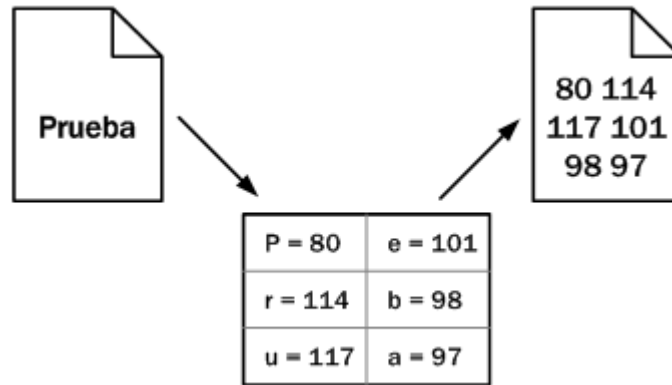


Figura 2.2. Ejemplo sencillo de codificación de caracteres

Este proceso de transformación de caracteres en secuencias de números se llama **codificación de caracteres** y cada una de las tablas que se han definido para realizar la transformación se llaman **páginas de código**. Una de las codificaciones más conocidas (y una de las primeras que se publicaron) es la codificación ASCII. La importancia de las codificaciones en HTML se verá más adelante.

Una vez resuelto el problema de almacenar la información sencilla basada en texto, se presenta el siguiente problema: ¿cómo almacenar la información compleja que incluye diferente formato para diferentes partes del texto? En otras palabras, ¿cómo se almacena un texto en negrita? ¿y un texto de color rojo? ¿y otro texto azul, en negrita, subrayado y con un tamaño de letra mayor?

Utilizar una tabla de conversión similar a las que se utilizan para textos sencillos no es posible, ya que existen infinitos posibles estilos. Una solución técnicamente posible sería la de almacenar la información sobre los diferentes elementos que componen la información en una zona reservada del archivo.

A pesar de estas posibles soluciones (y muchas otras que se puedan imaginar) la solución que se emplea es mucho más sencilla. La solución consiste en combinar en el mismo archivo los contenidos y la información sobre esos contenidos. Por ejemplo, si se quiere dividir el texto en párrafos y se desea indicar que algunas palabras del párrafo son de especial importancia, se puede indicar de la siguiente manera:

```
<parrafo>
```

```
Contenido de texto marcado mediante <importante>etiquetas</importante>.
```

```
</parrafo>
```

El principio de un párrafo se indica mediante los caracteres `<parrafo>` y el final de un párrafo se indica mediante los caracteres `</parrafo>`. De la misma manera, para asignar más importancia a ciertas palabras del texto, se indica el comienzo y el final de esa importancia mediante `<importante>` y `</importante>` respectivamente.

El proceso de indicar las diferentes partes que componen la información se denomina **marcar** (*markup* en inglés). Cada una de las estructuras que se emplean para marcar el inicio y el final de un elemento se denominan **etiquetas**. Aunque existen algunas excepciones, las etiquetas se indican por pares y se forman de la siguiente manera:

- **Etiqueta de apertura:** carácter `<`, seguido del nombre de la etiqueta (sin espacios en blanco) y seguido del carácter `>`.
- **Etiqueta de cierre:** carácter `<`, seguido del carácter `/`, seguido del nombre de la etiqueta (sin espacios en blanco) y seguido del carácter `>`.

```
<nombre_etiqueta> ... </nombre_etiqueta>
```

Las páginas HTML habituales están formadas por decenas de pares de etiquetas y por ese motivo HTML se considera un **lenguaje de etiquetas** (también llamado **lenguaje de marcado**). Además de HTML existen muchos otros lenguajes de etiquetas muy conocidos como XML, SGML, DocBook, MathML, etc.

La principal ventaja de los lenguajes de etiquetas es que es muy sencillo de leer y escribir por parte de las personas y de los ordenadores. La principal desventaja es que puede aumentar mucho el tamaño del documento, por lo que en general se definen etiquetas con nombres muy cortos.

2.3. El primer documento HTML

A continuación se muestra el código HTML de una página web muy sencilla:

```
<html>

<head>
<title>El primer documento HTML</title>
</head>
```

```
<body>
<p>El lenguaje HTML es <strong>tan sencillo</strong> que
prácticamente se entiende sin estudiar el significado
de sus etiquetas principales.</p>
</body>

</html>
```

Si quieres probar este primer ejemplo, debes hacer lo siguiente:

- Abre un editor de archivos de texto (si usas *Windows* puedes utilizar el *Bloc de notas*, el *Wordpad*, el *EmEditor*, el *UltraEdit*, etc. pero no utilices un editor complejo como *Word*) y crea un archivo nuevo
- Copia el código HTML mostrado anteriormente y pégalo en el archivo que estás creando
- Guarda el archivo con el nombre que quieras, pero con la extensión `.html`

Ahora ya puedes abrir el archivo creado en cualquier navegador de Internet y se mostrará con el siguiente aspecto:

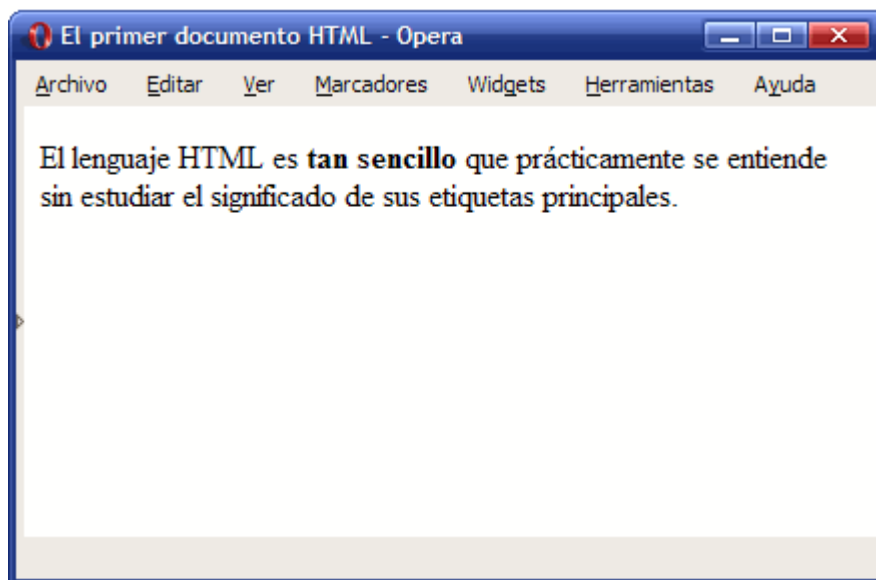


Figura 2.3. Aspecto que muestra el primer documento HTML en cualquier navegador

Si ya estás viendo tu primera página HTML en el navegador, prueba a pulsar sobre el menú `Ver > Código fuente` y podrás ver el código HTML de la página que está visualizando el navegador. De hecho, puedes ver el código HTML de cualquier página de Internet mediante el menú `Ver > Código fuente`. Prueba a ver el código HTML de tu página preferida y verás cuantas etiquetas puede llegar a tener una página compleja.

Volviendo al código HTML del primer ejemplo, es importante conocer las 3 etiquetas principales de un documento HTML (`<html>`, `<head>`, `<body>`):

- `<html>`: indica el comienzo y el final de un documento HTML. Ninguna etiqueta o contenido puede colocarse antes o después de la etiqueta `<html>` (con una sola excepción que se verá más adelante). En el interior de la etiqueta `<html>` se define la cabecera y el cuerpo del documento HTML y todo lo que se coloque fuera de la etiqueta `<html>` se ignora.
- `<head>`: delimita la parte del documento que se considera su cabecera. La cabecera contiene información sobre el propio documento HTML, como por ejemplo su título y el idioma de la página. Los contenidos indicados en la cabecera no son visibles para el usuario, con la excepción de la etiqueta `<title>`, que se utiliza para indicar el título del documento y que los navegadores lo visualizan en la parte superior izquierda de la ventana del navegador (si no te has fijado anteriormente, vuelve a abrir el primer ejemplo en cualquier navegador y observa donde se muestra el título de la página).
- `<body>`: define el cuerpo del documento HTML. El cuerpo encierra todos los contenidos que se muestran al usuario (párrafos de texto, imágenes, etc.). En general, el `<body>` de un documento contiene cientos de etiquetas HTML, mientras que el `<head>` no contiene más que unas pocas.

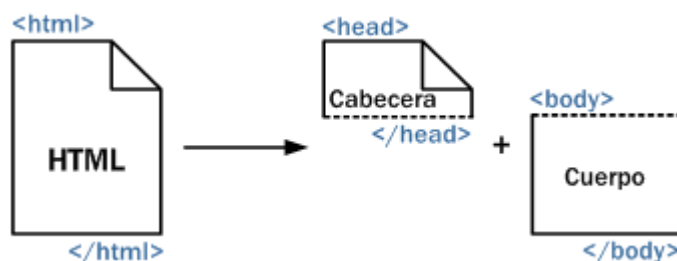


Figura 2.4. Esquema de las etiquetas principales que contiene un documento HTML

2.4. Etiquetas definidas por HTML

HTML define 90 etiquetas que los diseñadores pueden utilizar para marcar los diferentes elementos que componen una página:

a, abbr, acronym, address, applet, area, b, base, basefont, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, frame, frameset, h1, h2, h3, h4, h5, h6, head, hr, html, i, iframe, img, input, ins, isindex, kbd, label, legend, li, link, map, menu, meta, noframes, noscript, object, ol, optgroup, option, p, param, pre, q, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var.

De todas las etiquetas disponibles, las siguientes se consideran **obsoletas** y no se pueden utilizar: applet, basefont, center, dir, font, isindex, menu, s, strike, u.

2.5. Atributos

A pesar de las 90 etiquetas que define el lenguaje HTML, algunos elementos como las imágenes y los enlaces requieren cierta información adicional para ser útiles. La etiqueta `<a>` se emplea para insertar un enlace en una página, pero ¿cómo se puede indicar en cada caso la dirección a la que apunta el enlace? Evidentemente, no es posible crear una etiqueta por cada enlace diferente que se quiera definir.

La solución consiste en personalizar las etiquetas HTML mediante cierta información adicional llamada **atributos**. De esta forma, una misma etiqueta (`<a>`) utilizada para los enlaces se puede personalizar para que cada vez apunte a una dirección diferente.

```
<html>
```

```
<head>
```

```
<title>Ejemplo de atributos en las etiquetas</title>
```

```
</head>
```

```
<body>
```

```
<p>
```


Los enlaces son muy fáciles de indicar:

```
<a>Soy un enlace incompleto, porque no tengo dirección de destino</a>.
```

```
<a href="http://www.google.com">Este otro enlace apunta a la página de  
Google</a>.
```

```
</p>
```

```
</body>
```

```
</html>
```

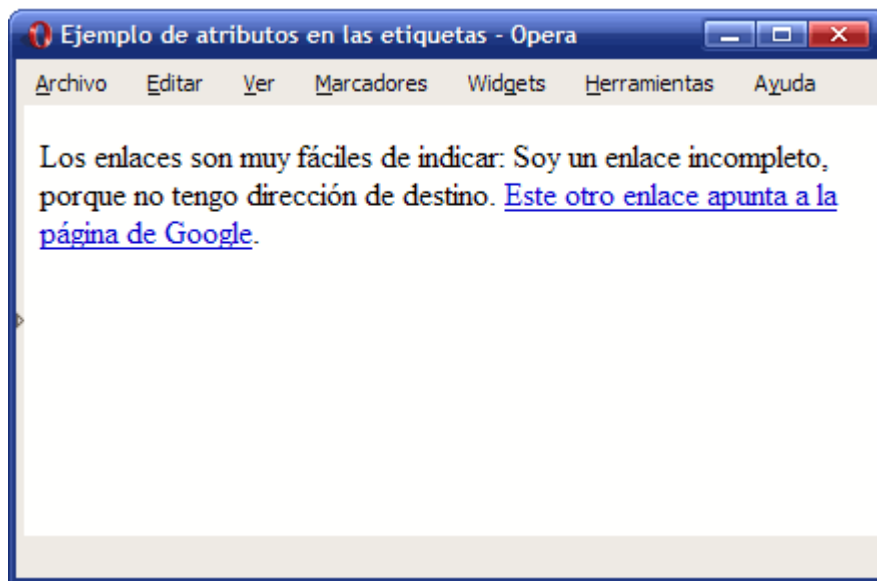


Figura 2.5. Los atributos permiten personalizar las etiquetas HTML

Cada una de las etiquetas tiene su propia lista de atributos disponibles. Además, por cada atributo también se indica el tipo de valor que se le puede asignar. Si el valor de un atributo no es válido, se ignora completamente ese atributo.

2.6. Elementos HTML

Además de etiquetas y atributos, HTML define el término “**elementos**” para referirse a las partes que componen los documentos HTML.

Los **elementos** HTML están formados por cada una de las estructuras que cumplan las siguientes condiciones:

- Empieza por una etiqueta de apertura.

- Contiene cero o más atributos.
- Contiene opcionalmente cualquier tipo de contenido de texto.
- Finaliza con una etiqueta de cierre.

Por ejemplo, un párrafo con atributos y contenidos se considera un elemento HTML, como se puede ver en el siguiente esquema:

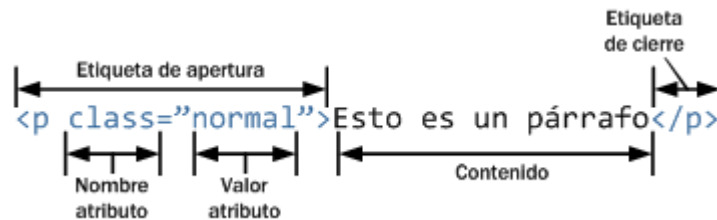


Figura 2.6. Esquema de las partes que componen un elemento HTML

La estructura mostrada es un elemento HTML ya que comienza con una etiqueta de apertura (`<p>`), contiene cero o más atributos (`class="normal"`), contiene un contenido de texto (Esto es un párrafo) y finaliza con una etiqueta de cierre (`</p>`).

Por tanto, si una página web tiene 2 párrafos definidos mediante la etiqueta `<p>`, la página contiene 2 elementos y 4 etiquetas (2 etiquetas `<p>` de apertura y 2 etiquetas `</p>` de cierre). De todas formas, es habitual intercambiar las palabras “elemento” y “etiqueta”, aunque estrictamente no son lo mismo.

2.7. Elementos en línea y elementos de bloque

HTML clasifica a todos los elementos en dos grandes grupos: elementos **en línea** (*inline elements* en inglés) y elementos de **bloque** (*block elements* en inglés).

La principal diferencia entre los dos tipos de elementos es la forma en la que ocupan el espacio disponible en la página. Los elementos de bloque siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea, aunque sus contenidos no lleguen hasta el final de la línea.

Los elementos en línea sin embargo no empiezan en nueva línea y solo ocupan el espacio necesario para mostrar sus contenidos. Ejemplo:

```
<html>

<head><title>Ejemplo de elementos en línea y elementos de bloque</title>
</head>

<body>
<p>Los párrafos son elementos de bloque.</p>
<a href="http://www.google.com">Los enlaces son elementos en línea</a>
<p>Dentro de un párrafo, <a href="http://www.google.com">los enlaces</a>
siguen siendo elementos en línea.</p>
</body>

</html>
```

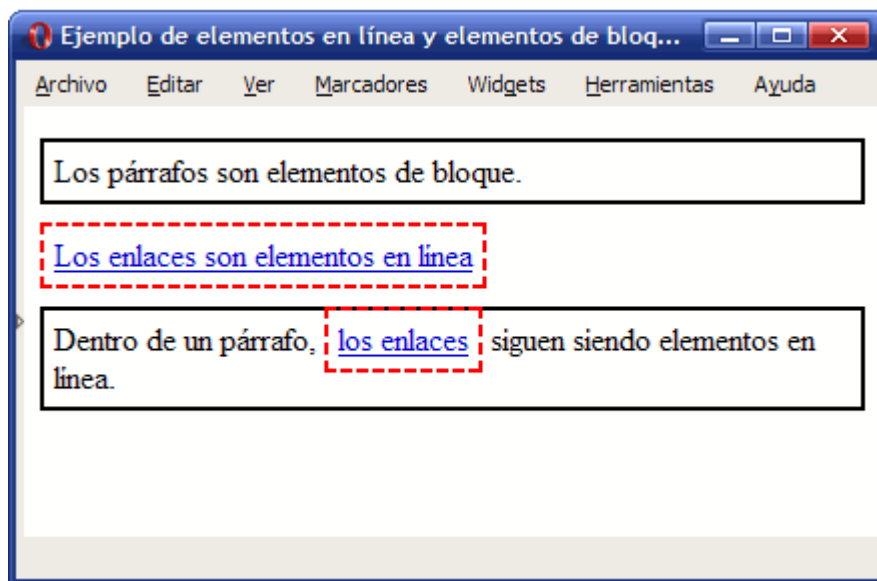


Figura 2.7. Diferencias entre elementos en línea y elementos de bloque

En el ejemplo anterior se ha añadido un borde diferente a los elementos `<p>` y los elementos `<a>`. De esta forma, es posible visualizar el espacio que ocupa cada elemento.

El primer párrafo contiene un texto que solo ocupa la mitad de la ventana del navegador. No obstante, el espacio reservado por el primer párrafo llega hasta el final de esa línea, por lo que está claro que los elementos `<p>` son elementos de bloque.

El primer enlace del ejemplo anterior también tiene un texto que ocupa solamente la mitad de la ventana del navegador. En este caso, el enlace solo ocupa el sitio necesario para mostrar sus contenidos. Si se añade otro enlace en esa misma línea, se mostrará a continuación del primer enlace. Por tanto, los elementos `<a>` son elementos en línea.

Por último, el segundo párrafo sigue ocupando todo el espacio disponible hasta el final de cada línea de texto del párrafo y los enlaces que se encuentran dentro del párrafo solo ocupan el sitio necesario para mostrar sus contenidos.

La mayoría de elementos de bloque pueden contener elementos en línea y otros elementos de bloque. Los elementos en línea solo pueden contener texto u otros elementos en línea. En otras palabras, un elemento de bloque no puede aparecer dentro de un elemento en línea. En cambio, un elemento en línea puede aparecer dentro de un elemento de bloque y dentro de otro elemento en línea.

Los elementos en línea definidos por HTML son: `a`, `abbr`, `acronym`, `b`, `basefont`, `bdo`, `big`, `br`, `cite`, `code`, `dfn`, `em`, `font`, `i`, `img`, `input`, `kbd`, `label`, `q`, `s`, `samp`, `select`, `small`, `span`, `strike`, `strong`, `sub`, `sup`, `textarea`, `tt`, `u`, `var`.

Los elementos de bloque definidos por HTML son: `address`, `blockquote`, `center`, `dir`, `div`, `dl`, `fieldset`, `form`, `h1`, `h2`, `h3`, `h4`, `h5`, `h6`, `hr`, `isindex`, `menu`, `noframes`, `noscript`, `ol`, `p`, `pre`, `table`, `ul`.

Los siguientes elementos también se considera que son de bloque: `dd`, `dt`, `frame-set`, `li`, `tbody`, `td`, `tfoot`, `th`, `thead`, `tr`. Los siguientes elementos pueden ser en línea y de bloque según las circunstancias: `button`, `del`, `iframe`, `ins`, `map`, `object`, `script`.

2.8. Sintaxis de las etiquetas XHTML

El lenguaje HTML original era muy permisivo en su sintaxis, es decir, permitía escribir sus etiquetas y atributos de muchas formas diferentes. Las etiquetas por ejemplo podían escribirse en mayúsculas, en minúsculas, combinando mayúsculas y minúsculas, etc. Los atributos de las etiquetas se podían indicar directamente o encerrados con comillas ("). Además, el orden en el que se abrían y cerraban las etiquetas no era importante.

La flexibilidad de HTML puede parecer un aspecto positivo, pero el resultado final son páginas con un código HTML desordenado, difícil de mantener y muy poco profesional.

Afortunadamente, XHTML soluciona estos problemas de HTML añadiendo ciertas normas en la forma de escribir etiquetas y atributos.

Las normas que introduce XHTML son solo 5, muy sencillas de aprender y casi todas son de sentido común. Las 5 normas definidas por XHTML son las siguientes:

Las etiquetas se tienen que cerrar de acuerdo a como se abren:

Ejemplo correcto (XHTML):

```
<p>Este es un párrafo con <a>un enlace</a></p>
```

Ejemplo incorrecto (HTML):

```
<p>Este es un párrafo con <a>un enlace</p></a>
```

Los nombres de las etiquetas y atributos siempre se escriben en minúsculas:

Ejemplo correcto (XHTML):

```
<p>Este es un párrafo con <a href="http://www.google.com">un enlace</a></p>
```

Ejemplo incorrecto (HTML):

```
<P>Este es un párrafo con <A HREF="http://www.google.com">un enlace</A></P>
```

El valor de los atributos siempre se encierra con comillas:

Ejemplo correcto (XHTML):

```
<p>Este es un párrafo con <a href="http://www.google.com">un enlace</a></p>
```

Ejemplo incorrecto (HTML):

```
<p>Este es un párrafo con <a href=http://www.google.com>un enlace</a></p>
```

Los atributos no se pueden comprimir:

Este tipo de atributos no es muy habitual.

Ejemplo correcto (XHTML):

```
<dl compact="compact">...</dl>
```

Ejemplo incorrecto (HTML):

```
<dl compact>...</dl>
```

Todas las etiquetas deben cerrarse siempre:

Algunas etiquetas no encierran ningún tipo de contenido, por lo que deben cerrarse justo después de abrirse. XHTML permite escribir de forma compacta una etiqueta que se abre y se cierra a la vez. Forma habitual: `
</br>`. Forma compacta: `
`. En la forma compacta es habitual equivocarse con la posición del carácter /.

Ejemplo correcto (XHTML):

```
<br/>
```

Ejemplo incorrecto (HTML):

```
<br>
```

2.9. Atributos comunes de las etiquetas HTML

Cada una de las etiquetas HTML define sus propios atributos, aunque algunos de los atributos son comunes a casi todas las etiquetas. De esta forma, es habitual explicar por separado los atributos comunes de las etiquetas para no tener que volver a hacerlo cada vez que se explica una nueva etiqueta.

Además, los atributos comunes se dividen en 4 grupos según su funcionalidad:

1) Los atributos básicos se utilizan prácticamente en todas las etiquetas de las páginas HTML habituales, aunque solo son realmente útiles cuando se utiliza CSS y/o JavaScript.

Atributos básicos

Atributo	Descripción
id = "texto"	Identifica de forma única a cada elemento de un documento HTML

<code>class = "texto"</code>	Permite establecer la clase CSS que se aplicará al elemento
<code>style = "estilos CSS"</code>	Permite establecer directamente el valor de las propiedades CSS del elemento
<code>title = "texto"</code>	Se emplea para establecer el título a un elemento (mejora la accesibilidad y los navegadores lo muestran cuando el usuario pasa el ratón por encima del elemento)

2) Algunas páginas muestran contenidos en varios idiomas e incluso en alfabetos con una escritura muy diferente a nuestro alfabeto.

Atributos para internacionalización (i18n)

Atributo	Descripción
<code>lang = "codigo de idioma"</code>	Indica el idioma del elemento
<code>xml:lang = "codigo de idioma"</code>	Indica el idioma del elemento. Tiene más prioridad que <code>lang</code> y es obligatorio incluirlo cuando se define el atributo <code>lang</code>
<code>dir</code>	Permite indicar la dirección del texto (útil para los idiomas que escriben de derecha a izquierda)

Como la palabra internacionalización es muy larga, se suele sustituir por la *palabra i18n* (el número 18 se refiere al número de letras que existen entre la letra i y la letra n de la palabra internacionalización).

3) Las páginas HTML pueden ser dinámicas y mostrar efectos cuando se producen algunos eventos (pinchar con el ratón, pulsar una tecla, etc.). Se utilizan exclusivamente por parte de JavaScript.

Atributos de eventos

Atributo	Descripción
----------	-------------

<p>onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup</p>	<p>Solamente se utilizan con JavaScript y permiten realizar operaciones cuando se producen eventos en un elemento (por ejemplo cuando el usuario pincha sobre un elemento o cuando pulsa una tecla)</p>
--	---

4) Los elementos que forman una página HTML se pueden seleccionar (pulsando la tecla TAB repetidas veces, seleccionándolo con el ratón, etc.). HTML define algunos atributos que controlan la selección de los elementos.

Atributos para los elementos que pueden obtener el foco

Atributo	Descripción
accesskey = "letra"	Permite definir una tecla de acceso rápido a un elemento HTML. Si el usuario pulsa la tecla ALT y esa tecla, se selecciona el elemento
tabindex = "numero"	Establece la posición del elemento en el orden de tabulación del documento (el orden en el que el elemento se selecciona cuando el usuario navega por el documento con la tecla TAB). Debe tener un valor entre 0 y 32.767
onfocus, onblur	Se emplean para definir las funciones JavaScript que se ejecutan cuando el elemento obtiene o pierde el foco

En el resto del documento, se emplearán las palabras "básicos", "input", "eventos" y "foco" respectivamente para referirse a cada uno de los grupos de atributos definidos anteriormente.

Capítulo 3. Texto

3.1. Texto básico, estructurado y marcado

La mayor parte del contenido de las páginas HTML habituales está formado por texto, llegando a ser más del 90% del código de la página. Por este motivo, es muy importante conocer los elementos y etiquetas que define HTML para el manejo del texto.

Para el tratamiento del texto, el lenguaje HTML sigue las mismas ideas y normas establecidas en otros entornos de publicación de contenidos. De esta forma, HTML define etiquetas para **estructurar** el contenido en secciones y párrafos y define otras etiquetas para **marcar** elementos importantes dentro del texto.

Por tanto, la tarea inicial del editor de contenidos HTML es estructurar el texto original definiendo sus párrafos, titulares y títulos de sección.

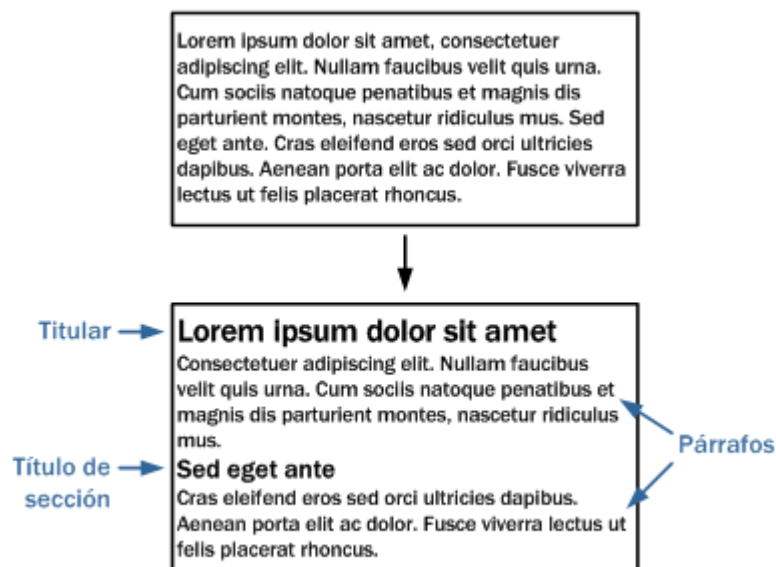


Figura 3.1. Resultado de estructurar un texto sencillo

El anterior ejemplo muestra la transformación de un texto simple en un texto estructurado con 1 titular, 1 título de sección y 2 párrafos.

Una vez definida la estructura básica de los contenidos de la página, el editor marca los elementos del texto: textos importantes, definiciones, abreviaturas, textos modificados, citas a otras referencias, etc.

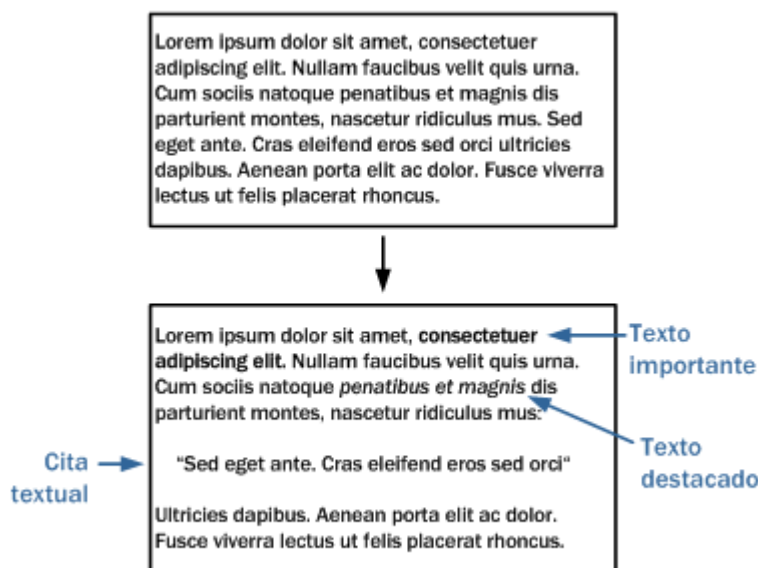


Figura 3.2. Resultado de marcar un texto sencillo

El anterior ejemplo muestra la transformación de un texto simple en un texto marcado que especifica las partes importantes (texto en negrita), las partes destacadas (texto en cursiva) y las partes que constituyen una cita textual de otro contenido (texto tabulado).

Por último, HTML hace un uso muy especial de los espacios en blanco y de las nuevas líneas, por lo que es necesario conocer ese comportamiento y las alternativas disponibles.

3.2. Estructurar

3.2.1. Párrafos

Una de las etiquetas de HTML más utilizadas es la etiqueta <p>, que permite definir los párrafos de una página. Para definir un párrafo, se encierra el texto que forma el párrafo con la etiqueta <p>, como muestra el siguiente ejemplo:

```
<html>

<head>
<title>Ejemplo de texto estructurado con párrafos</title>
</head>

<body>
<p>Este es el texto que forma el primer párrafo de la página.
Los párrafos pueden ocupar varias líneas y el navegador se encarga
de ajustar su longitud al tamaño de la ventana.</p>

<p>El segundo párrafo de la página también se define encerrando
su texto con la etiqueta p. El navegador también se encarga de
separar automáticamente cada párrafo.</p>
</body>

</html>
```

El ejemplo anterior se visualiza de la siguiente manera en cualquier navegador:

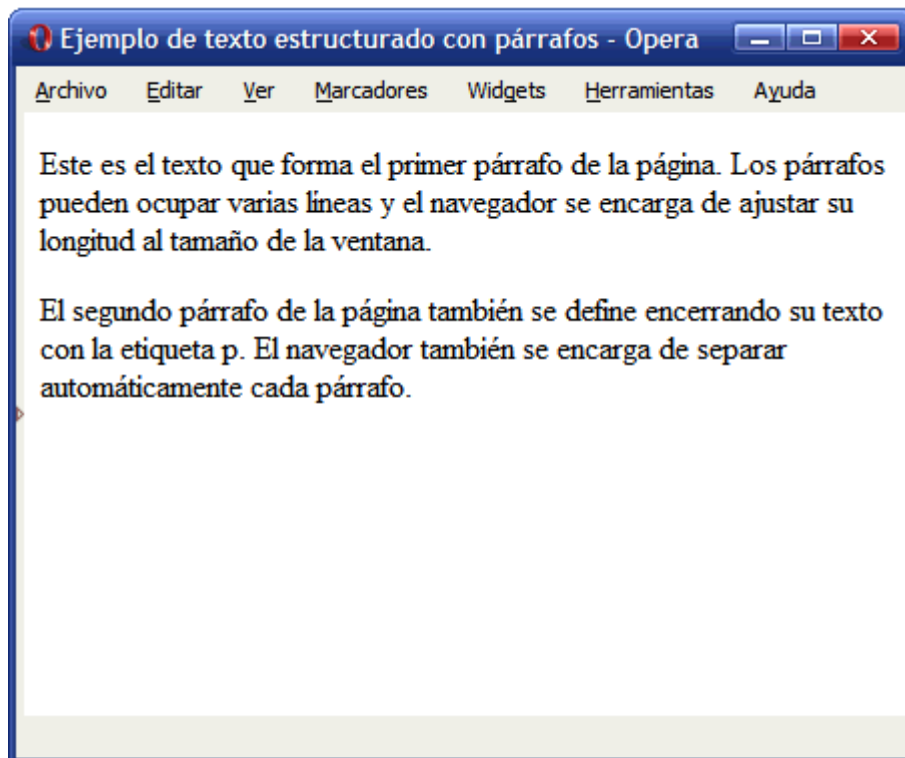


Figura 3.3. Ejemplo de texto HTML estructurado con párrafos

A continuación, se indica la definición formal de la etiqueta <p>:

Tabla 3.1. Etiqueta p

p	Párrafos
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	Bloque
Descripción	Define un párrafo del documento

Los párrafos en HTML son elementos de bloque y por ese motivo se ajustan al tamaño de la ventana ocupando todo el sitio disponible horizontalmente. Además, no tienen atributos específicos, pero sí que se les pueden asignar los atributos comunes de HTML (básicos, i18n y de eventos).

3.2.2. Secciones

Definir los párrafos que forman el texto de la página no es suficiente para estructurar los documentos HTML habituales. Los titulares y los títulos de sección permiten delimitar las diferentes secciones que forman el texto. HTML define las etiquetas necesarias para crear secciones de 6 niveles diferentes.

Las etiquetas que definen los títulos de sección son <h1>, <h2>, <h3>, <h4>, <h5> y <h6>. La etiqueta <h1> es la de mayor importancia y por tanto se utiliza para definir los titulares de la página. El resto de etiquetas es de importancia descendiente, hasta llegar a la etiqueta <h6>, que permite definir los titulares de las secciones menos importantes de la página.

Tabla 3.2. Etiqueta h1

h1	Sección (titular) de nivel 1
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	Bloque
Descripción	Se emplea para definir las diferentes secciones (titulares) de un documento. Los navegadores les asignan un tamaño de letra en función de su importancia

Al igual que la etiqueta `<p>`, las etiquetas de título de sección son elementos de bloque y no tienen atributos específicos.

Las etiquetas `<h1>`, ..., `<h6>` definen **títulos de sección**, no secciones completas, por lo que no es necesario encerrar los contenidos de una sección con la etiqueta correspondiente. Solamente se debe encerrar con las etiquetas `<h1>`, ..., `<h6>` los titulares de cada sección.

El siguiente ejemplo muestra el uso de las etiquetas de título de sección:

```
<html>

<head>
<title>Ejemplo de texto estructurado con secciones</title>
</head>

<body>
<h1>Titular de la página</h1>

<p>Párrafo de introducción...</p>

<h2>La primera sub-sección</h2>

<p>Párrafo de contenido...</p>
```

```
<h2>Otra subsección</h2>
```

```
<p>Más párrafos de contenido...</p>
```

```
</body>
```

```
</html>
```

Los navegadores muestran el ejemplo anterior de la siguiente manera:

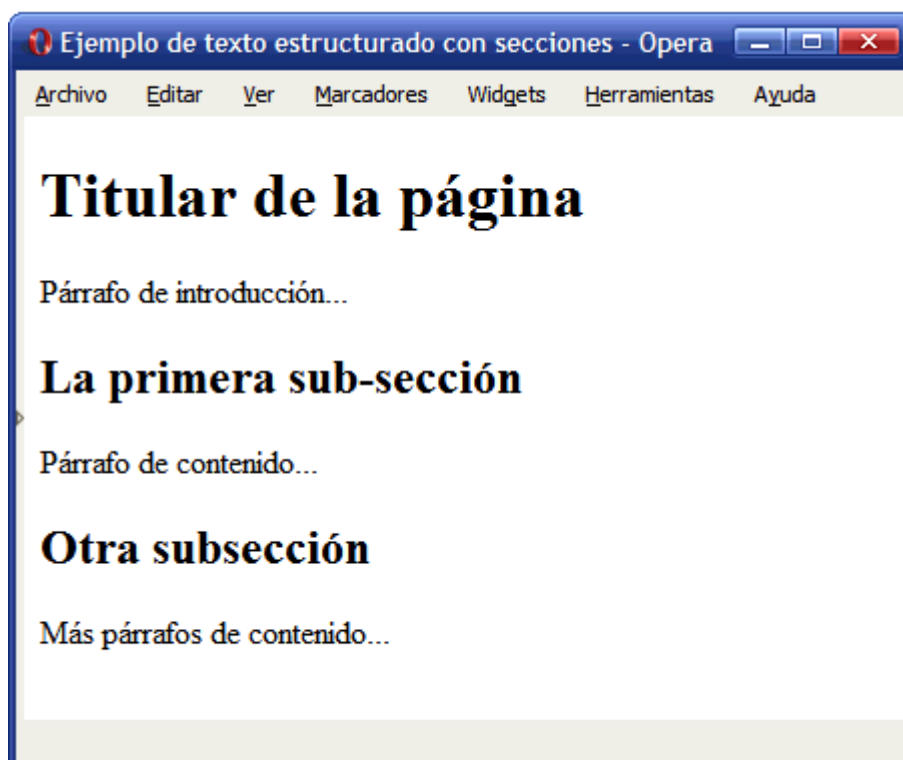


Figura 3.4. Ejemplo de texto HTML estructurado con párrafos y secciones

Los navegadores asignan de forma automáticamente el tamaño del titular de cada sección en función de la importancia de la sección. Así, los títulos de sección `<h1>` se muestran con el tamaño de letra más grande y los títulos de sección `<h6>` se visualizan con un tamaño de letra muy pequeño. Mediante CSS se puede modificar el tamaño y aspecto que los navegadores asignan por defecto a cada título de sección.

La siguiente imagen muestra el tamaño por defecto con el que los navegadores muestran cada titular:

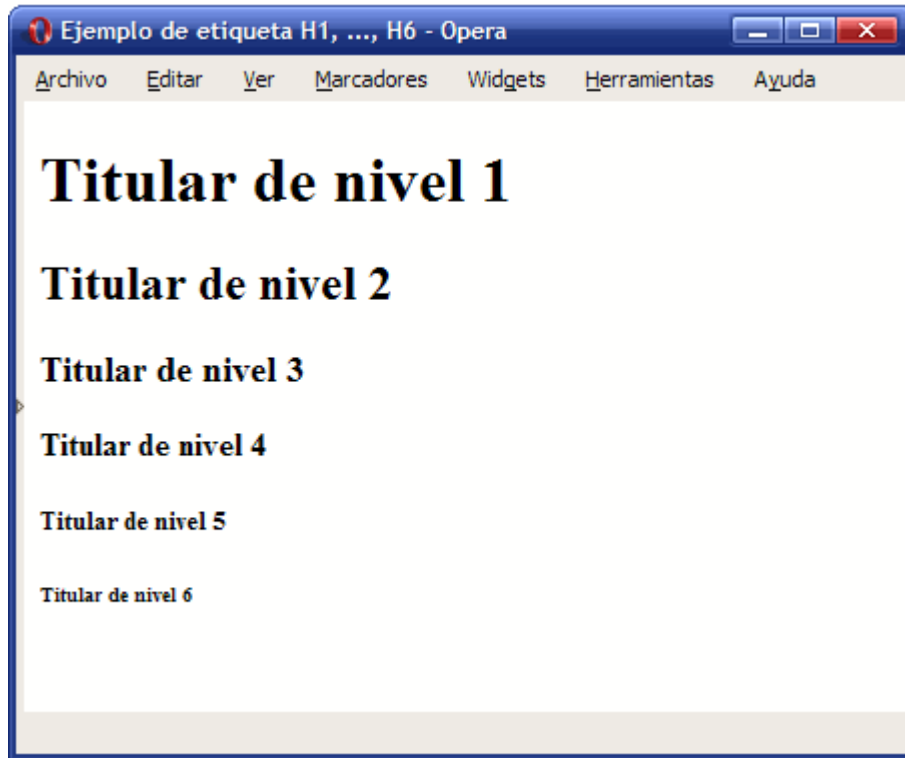


Figura 3.5. Ejemplo de uso de las etiquetas h1, h2, h3, h4, h5 y h6

3.3. Espacios en blanco y nuevas líneas

El aspecto más sorprendente del lenguaje HTML cuando se desarrollan los primeros documentos es el tratamiento especial de los “*espacios en blanco*” del texto. HTML considera *espacio en blanco* a: los espacios en blanco, los tabuladores, los retornos de carro y el carácter de nueva línea (ENTER o Intro).

El siguiente ejemplo ilustra este comportamiento:

```
<html>
<head><title>Ejemplo de etiqueta p</title></head>
<body>
<p>Este primer párrafo no contiene saltos de línea ni otro tipo de espaciado.</p>

<p>Este segundo párrafo sí que contiene saltos
de
línea
```

```
y otro tipo de espaciado.</p>  
</body>  
</html>
```

El anterior código HTML se visualiza en cualquier navegador de la siguiente manera:

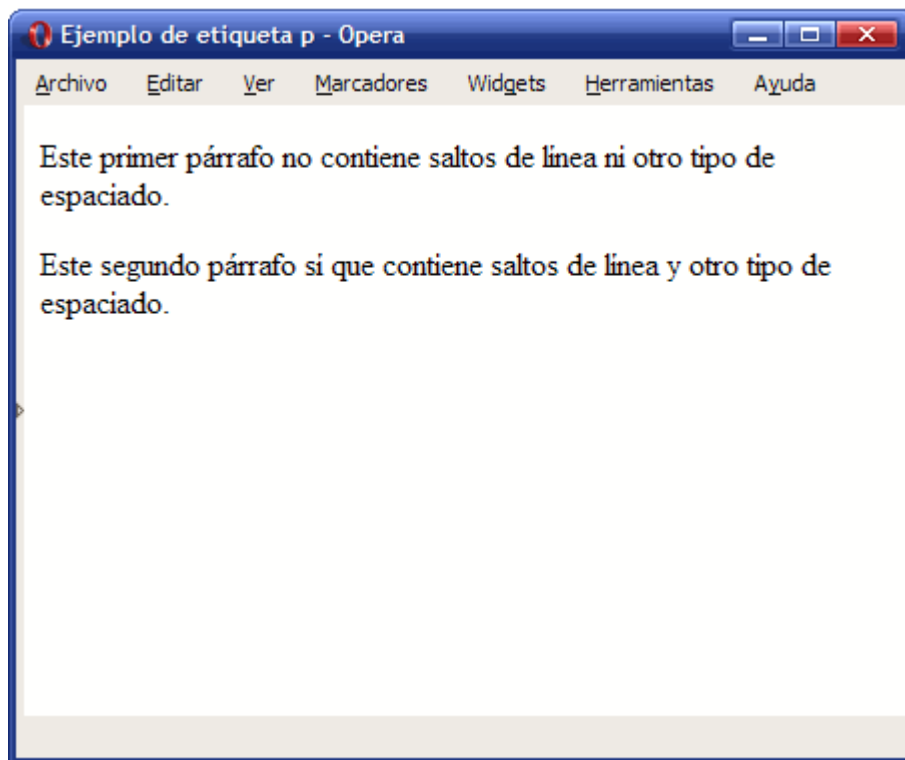


Figura 3.6. Ejemplo de comportamiento de HTML con los espacios en blanco

Los dos párrafos de la imagen anterior se ven idénticos, aunque el segundo párrafo incluye varios espacios en blanco y está escrito en varias líneas diferentes. La razón de este comportamiento es que HTML ignora todos los espacios en blanco *sobrantes*, es decir, todos los espacios en blanco que no son el espacio en blanco que separa las palabras.

HTML proporciona varias alternativas para poder incluir tantos espacios en blanco y tantas nuevas líneas como sean necesarias.

3.3.1. Nuevas líneas

La etiqueta `
` permite indicar el lugar en el que se va a incluir un carácter de nueva línea. En otras palabras, incluir la etiqueta `
` en un determinado punto del texto equivale a presionar la tecla ENTER (o Intro) en ese mismo punto.

La definición formal de `
` se muestra a continuación:

Tabla 3.3. Etiqueta `br`

br	Nueva línea
Atributos comunes	básicos
Atributos específicos	-
Tipo	En línea y etiqueta vacía
Descripción	Fuerza a que los navegadores inserten una nueva línea

La etiqueta `
` es una de las pocas etiquetas especiales de HTML. La particularidad de `
` es que es una etiqueta vacía, es decir, no debe encerrar ningún texto para que funcione correctamente. De esta forma, lo correcto es abrir y cerrar la etiqueta de forma seguida (`
</br>`). HTML permite abrir y cerrar a la vez una etiqueta de forma rápida: `
` (también se puede escribir como `
`).

Utilizando la etiqueta `
` se puede rehacer el ejemplo anterior para que respete las líneas que forman el segundo párrafo:

```
<html>

<head>
<title>Ejemplo de etiqueta br</title>
</head>

<body>
<p>Este primer párrafo no contiene saltos de línea ni otro tipo de espaciado.</p>

<p>Este segundo párrafo sí que contiene saltos <br/>
de <br/>
línea <br/>
```

```
y otro tipo de espaciado.</p>  
</body>
```

```
</html>
```

El navegador ahora sí que muestra correctamente las nuevas líneas que se querían insertar:

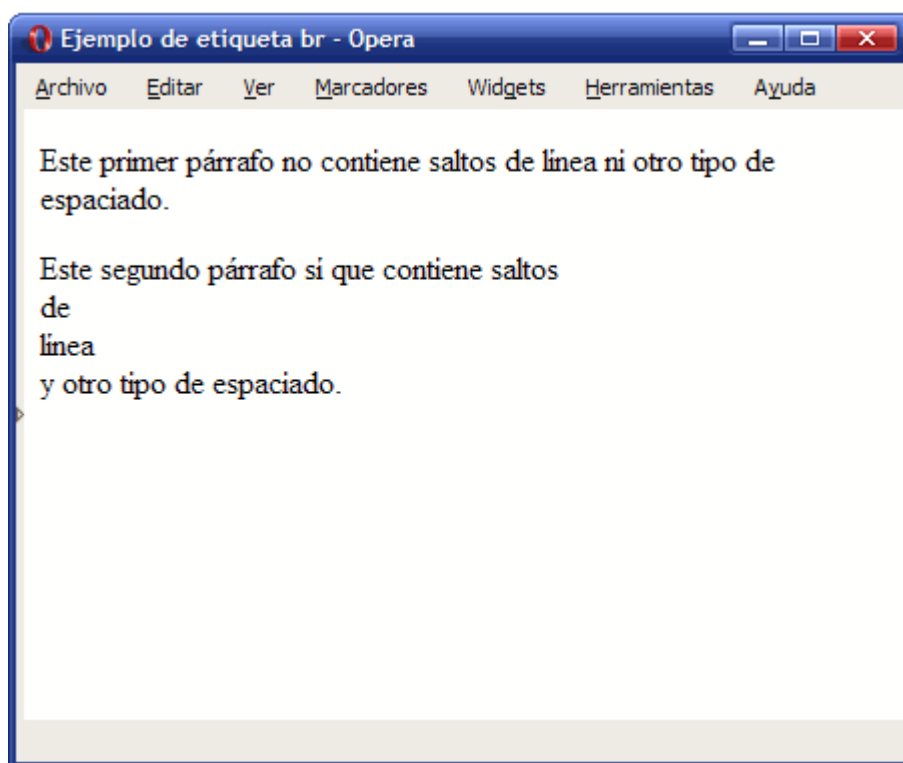


Figura 3.7. Ejemplo de uso de la etiqueta br

3.3.2. Espacios en blanco

La solución al problema de los espacios en blanco no es tan sencilla como el de las nuevas líneas. Para incluir espacios en blanco adicionales, se debe sustituir cada nuevo espacio en blanco por el texto ` ` (es importante incluir el símbolo `&` al principio y el símbolo `;` al final).

Así, el código HTML del ejemplo anterior se debe rehacer para incluir los espacios en blanco adicionales:

```
<html>
```

```
<head>
```

```
<title>Ejemplo de entidad &nbsp;</title>
</head>

<body>
<p>Este primer párrafo no contiene saltos de línea ni otro tipo de espaciado.</p>

<p>Este segundo párrafo sí que contiene saltos <br/>
de <br/>
línea <br/>
y &nbsp;&nbsp;  otro &nbsp;   tipo &nbsp;   de &nbsp;   espaciado.</p>
</body>

</html>
```

Ahora el navegador sí que muestra correctamente los espacios en blanco (y las nuevas líneas) del segundo párrafo:

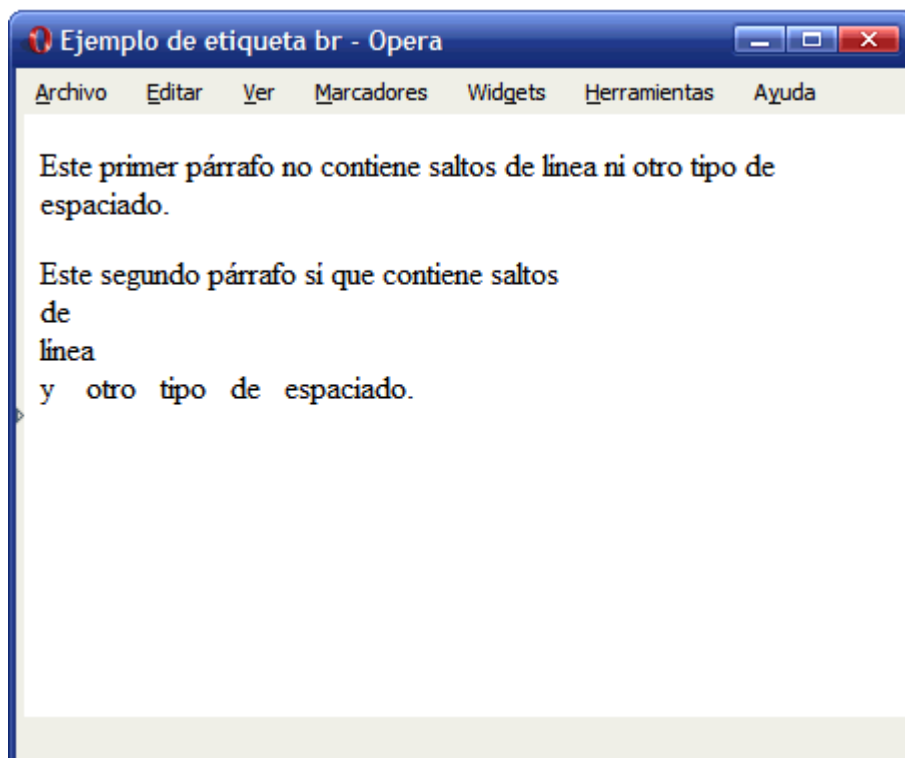


Figura 3.8. Ejemplo de uso de espacios en blanco en HTML

Cada texto ` ` solamente equivale a un espacio en blanco, por lo que se deben escribir tantos ` ` seguidos como espacios en blanco seguidos existan en el texto.

Más adelante se profundiza en el origen de ` `; y en el de algunos códigos similares imprescindibles para el diseño de páginas web.

3.3.3. Texto preformateado

En ocasiones, no se puede modificar un texto para el que se tienen que mostrar los espacios en blanco y las líneas originales. En este caso, se debe utilizar la etiqueta `<pre>` de HTML que muestra el texto tal y como se ha escrito. La definición formal de la etiqueta se muestra a continuación:

Tabla 3.4. Etiqueta `pre`

pre	Texto preformateado
Atributos comunes	básicos, <code>18n</code> y eventos
Atributos específicos	-
Tipo	Bloque
Descripción	Muestra el texto que se le indica tal y como está escrito (respetando los espacios en blanco)

El siguiente ejemplo muestra el uso de la etiqueta `<pre>`:

```
<html>
<head><title>Ejemplo de etiqueta pre</title></head>

<body>
<pre>
    La etiqueta pre
    respeta los espacios en blanco
    y
```

```
muestra el texto
  tal y como
  está escrito
</pre>

<p>
  La etiqueta pre
  respeta los espacios en blanco
  y
  muestra el texto
  tal y como
  está escrito
</p>

</body>
</html>
```

El ejemplo incluye el mismo texto (con espacios en blanco y varias líneas) dentro de una etiqueta `<pre>` y dentro de una etiqueta `<p>`. Las diferencias visuales en un navegador son muy evidentes:

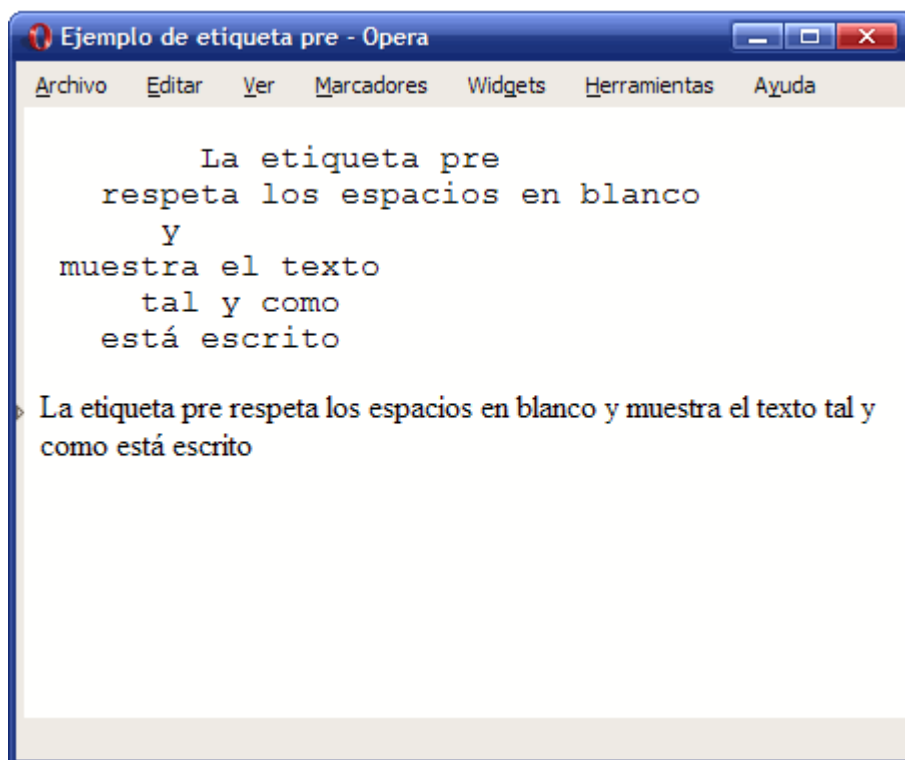


Figura 3.9. Ejemplo de uso de la etiqueta pre

El primer texto se ve en pantalla tal y como se ha escrito, respetando todos los espacios en blanco y todas las nuevas líneas. El segundo texto se ve como un párrafo normal para el que HTML ha eliminado todos los espacios en blanco sobrantes. Los elementos `<pre>` son especiales, ya que los navegadores les aplican las siguientes reglas:

- Mantienen todos los espacios en blanco (tabuladores, espacios y nuevas líneas)
- Muestra el texto en una fuente de ancho fijo
- No ajusta la longitud de las líneas (las líneas largas producen *scroll* horizontal)

La segunda característica significa que, por defecto, el texto incluido dentro de un elemento `<pre>` se muestra con un tipo de letra especial en el que todas las letras tienen la misma anchura. Si el sistema operativo del usuario es Windows, la letra que se utiliza es *Courier New*.

La tercera característica diferencia por completo a los párrafos de los elementos `<pre>`. Como se ha visto, los navegadores ajustan continuamente la anchura de los párrafos de texto para que ocupen todo el tamaño de la ventana. Sin embargo, los elementos `<pre>` se muestran tal y como

son originalmente, por lo que una línea muy larga dentro de un elemento `<pre>` provoca que la anchura de la página sea superior a la anchura de la ventana.

Si en el ejemplo anterior se añade más texto al final de la segunda línea (para producir una línea larga), el navegador muestra un *scroll* horizontal ya que el texto completo no cabe en el tamaño de la ventana y las líneas de los elementos `<pre>` nunca se ajustan.

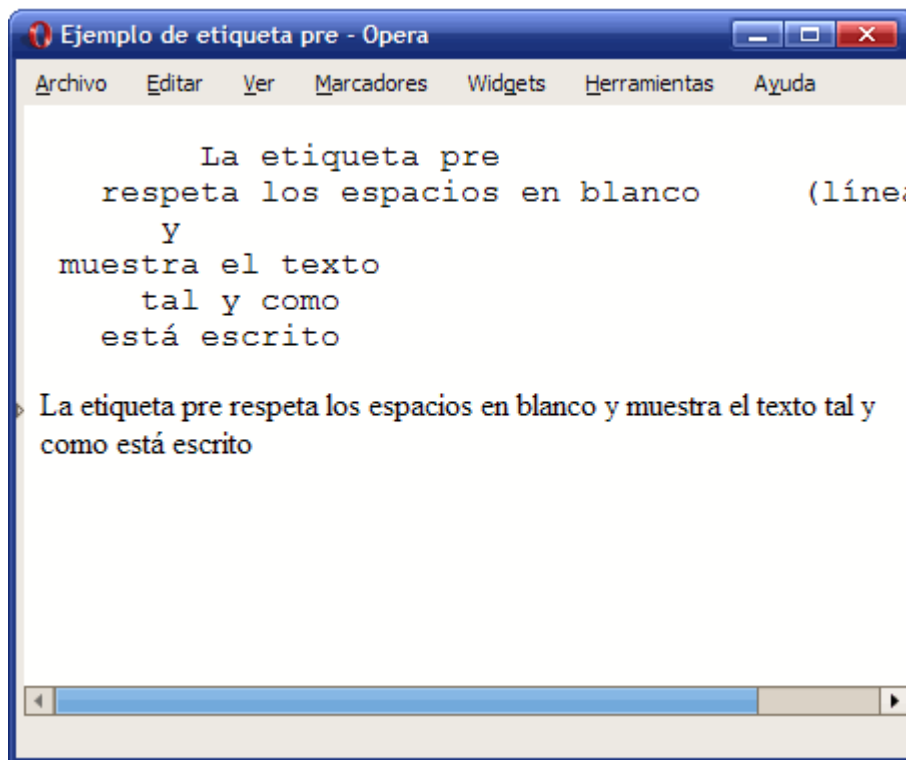


Figura 3.10. Ejemplo de aparición de scroll horizontal con la etiqueta pre

Otra etiqueta relacionada con `<pre>` es la etiqueta `<code>`, que se utiliza para mostrar código fuente de cualquier lenguaje de programación. La definición formal de `<code>` es la siguiente:

Tabla 3.5. Etiqueta code

code	Código fuente
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-

code	Código fuente
Tipo	En línea
Descripción	Considera el texto como si fuera un fragmento de código fuente

En la mayoría de páginas web, no tiene sentido utilizar la etiqueta `<code>`. Sin embargo, cuando se quiere incluir el listado de un programa, un trozo de código o incluso un determinado código HTML, lo correcto es emplear la etiqueta `<code>`.

Ejemplo:

```
<html>
<head><title>Ejemplo de etiqueta code</title></head>

<body>

< code>
    La etiqueta code
    no respeta los espacios en blanco
</ code>

<p>La etiqueta code es similar a la
etiqueta pre, sobre todo en el formato
del texto.</p>

</body>
</html>
```

El navegador muestra claramente el comportamiento de `<code>` y sus diferencias con `<pre>`:

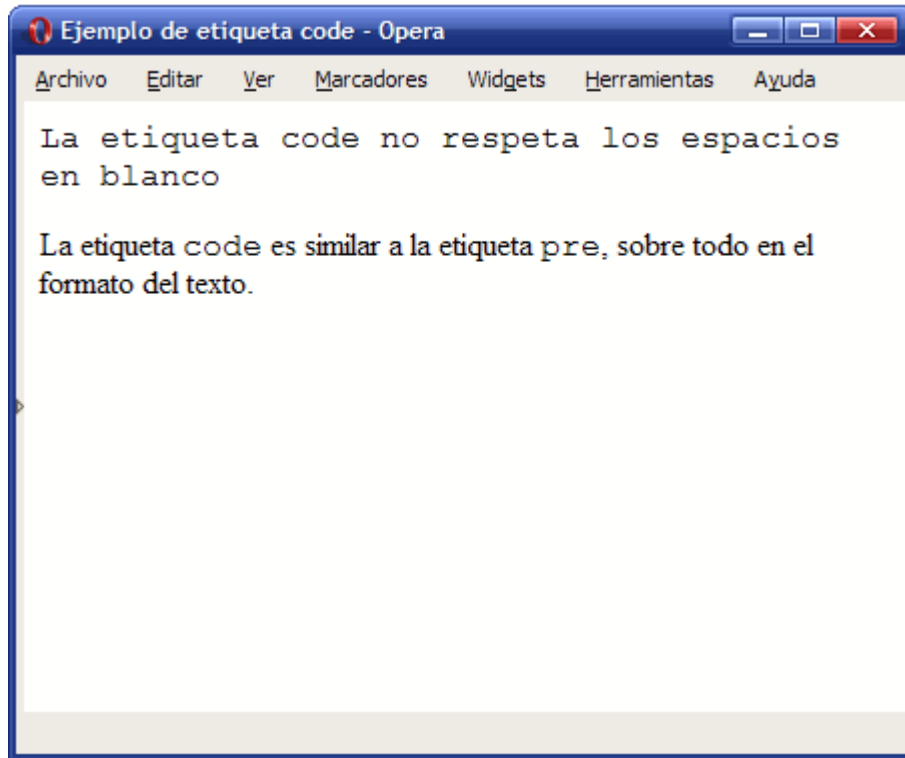


Figura 3.11. Ejemplo de uso de la etiqueta code

El texto incluido dentro de un elemento `<code>` se muestra con una fuente de ancho fijo. Al contrario que sucedía con `<pre>`, el elemento `<code>` no respeta los espacios en blanco ni las líneas. Por último, `<code>` es un elemento en línea, mientras que `<pre>` era un elemento de bloque.

3.4. Marcado básico de texto

Una vez estructurado el texto en párrafos y secciones, el siguiente paso es el marcado de los elementos que componen el texto. Los textos habituales están formados por elementos como palabras en negrita o cursiva, anotaciones y correcciones, citas a otros documentos externos, etc. HTML proporciona varias etiquetas para el marcado de cada uno de los diferentes tipos de texto.

Entre las etiquetas más utilizadas para marcar texto se encuentran `` y ``. La definición formal de las 2 etiquetas se muestra a continuación:

Tabla 3.6. Etiqueta em

em	Énfasis
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	En línea
Descripción	Marca el énfasis de un determinado texto para que los navegadores lo resalten de forma adecuada

Tabla 3.7. Etiqueta strong

strong	Énfasis más acentuado
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	En línea
Descripción	Marca el énfasis de un determinado texto de forma más acentuada que la etiqueta em

La etiqueta marca un texto indicando que su importancia es mayor que la del resto del texto. La etiqueta indica que un determinado texto es de la mayor importancia.

Ejemplo:

```
<html>

<head>
<title>Ejemplo de etiqueta em y strong</title>
</head>
```

```
<body>
<p>El lenguaje HTML permite marcar algunos segmentos de texto
como <em>muy importantes</em> y otros segmentos como <strong>los
más importantes</strong>.</p>
</body>

</html>
```

Por defecto, los navegadores muestran los elementos `` en cursiva para hacer evidente su importancia y muestran los elementos `` en negrita, para indicar que son los más importantes:

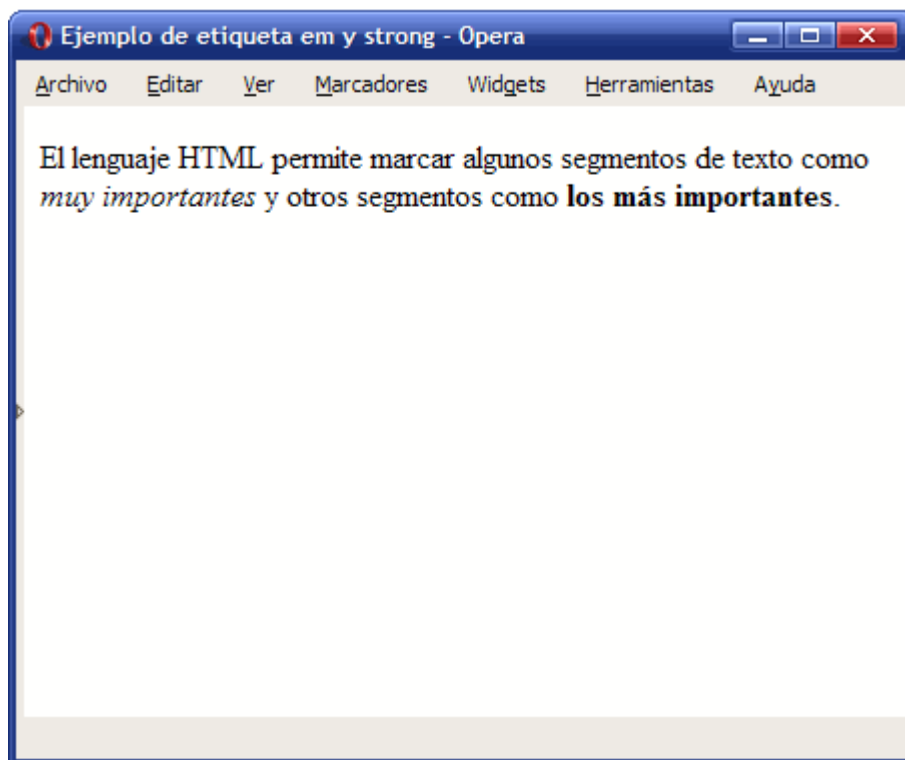


Figura 3.12. Ejemplo de uso de las etiquetas em y strong

HTML también permite marcar de forma adecuada las modificaciones realizadas en el contenido de una página. En otras palabras, HTML permite indicar de forma clara el texto eliminado y el texto añadido a un determinado texto original. Las etiquetas utilizadas son `<ins>` y ``, cuya definición formal es la siguiente:

Tabla 3.8. Etiqueta ins

ins	Inserción
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • <code>cite = "url"</code> - Permite indicar la URL de un documento en el que se puede obtener la información sobre el motivo por el que se realizó la modificación. • <code>datetime = "fecha"</code> - Especifica la fecha y hora en la que se realizó el cambio
Tipo	Bloque y en línea
Descripción	Se emplea para marcar una modificación en los contenidos consistente en la inserción de un nuevo contenido

Tabla 3.9. Etiqueta del

del	Borrado
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • <code>cite = "url"</code> - Permite indicar la URL de un documento en el que se puede obtener la información sobre el motivo por el que se realizó la modificación. • <code>datetime = "fecha"</code> - Especifica la fecha y hora en la que se realizó el cambio
Tipo	Bloque y en línea
Descripción	Se emplea para marcar una modificación en los contenidos consistente en el borrado de cierto contenido

Las 2 etiquetas cuentan con los mismos atributos específicos, que opcionalmente se pueden añadir para proporcionar más información. El atributo `cite` se emplea para indicar la dirección de un documento externo en el que se puede encontrar más información relacionada con la inserción o el borrado de texto. El atributo `datetime` puede utilizarse para indicar la fecha y la hora en la que se realizó cada cambio.

Ejemplo:

```
<html>
<head><title>Ejemplo de etiqueta ins y del</title></head>
<body>

<h3>Ejemplo de etiqueta ins y del</h3>

<p>El HTML, acrónimo inglés de Hyper Text Markup Language (lenguaje de
< del datetime="20061025" cite="http://www.librosweb.es/
mas_informacion.html">mercado de
hipertexto</ del> < ins datetime="20061026" cite="http://www.librosweb.es/
mas_informacion.html">
marcas hipertextuales</ ins>) es un lenguaje de marcación diseñado para
estructurar textos y
presentarlos en forma de hipertexto.</p>

</body>
</html>
```

Los navegadores muestran el ejemplo anterior de la siguiente manera:

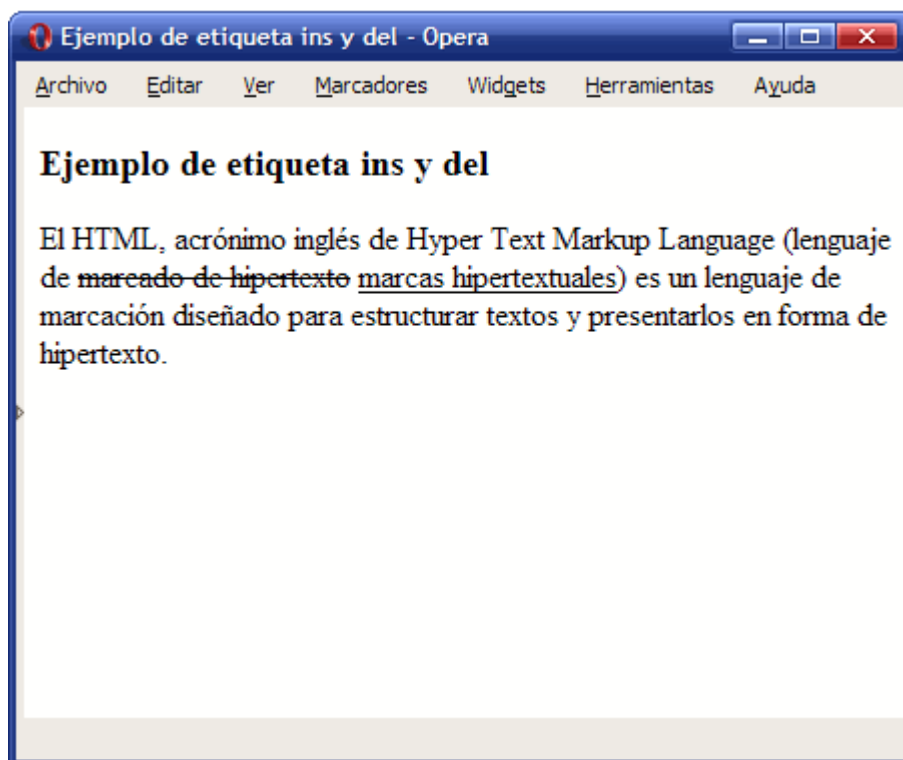


Figura 3.13. Ejemplo de uso de las etiquetas ins y del

Por defecto, el texto eliminado (marcado con la etiqueta ``) se muestra tachado de forma que el usuario pueda identificarlo fácilmente como un texto que formaba parte del texto original y que ya no tiene validez. El texto insertado (marcado con la etiqueta `<ins>`) se muestra subrayado, de forma que el usuario pueda identificarlo como un texto nuevo que no formaba parte del texto original.

En muchos tipos de páginas (artículos, noticias) es habitual citar textualmente un texto externo. HTML define la etiqueta `<blockquote>` para incluir citas textuales en los documentos. La definición de la etiqueta HTML con el nombre más largo se muestra a continuación:

Tabla 3.10. Etiqueta `blockquote`

blockquote	Citas
Atributos comunes	básicos, i18n y eventos

blockquote	Citas
Atributos específicos	<ul style="list-style-type: none"> • <code>cite = "url"</code> - Permite indicar la dirección del documento original del que se extrae la cita
Tipo	Bloque
Descripción	Se emplea para indicar que el texto que contiene es una cita de otro texto externo

Al igual que `<ins>` y ``, la etiqueta `<blockquote>` permite indicar la dirección de un documento del que se ha extraído la cita. Ejemplo:

```
<html>
<head><title>Ejemplo de etiqueta blockquote</title></head>

<body>
<p>Según el W3C, el valor del
atributo <em>cite</em> en las etiquetas <strong>blockquote</strong> tiene el
siguiente significado:</p>

<blockquote cite="http://www.w3.org/TR/html401/struct/text.html">"El valor de este
atributo
es una dirección URL que indica el documento original de la cita."</blockquote>
</body>

</html>
```

El aspecto que muestra el ejemplo anterior en cualquier navegador es el siguiente:

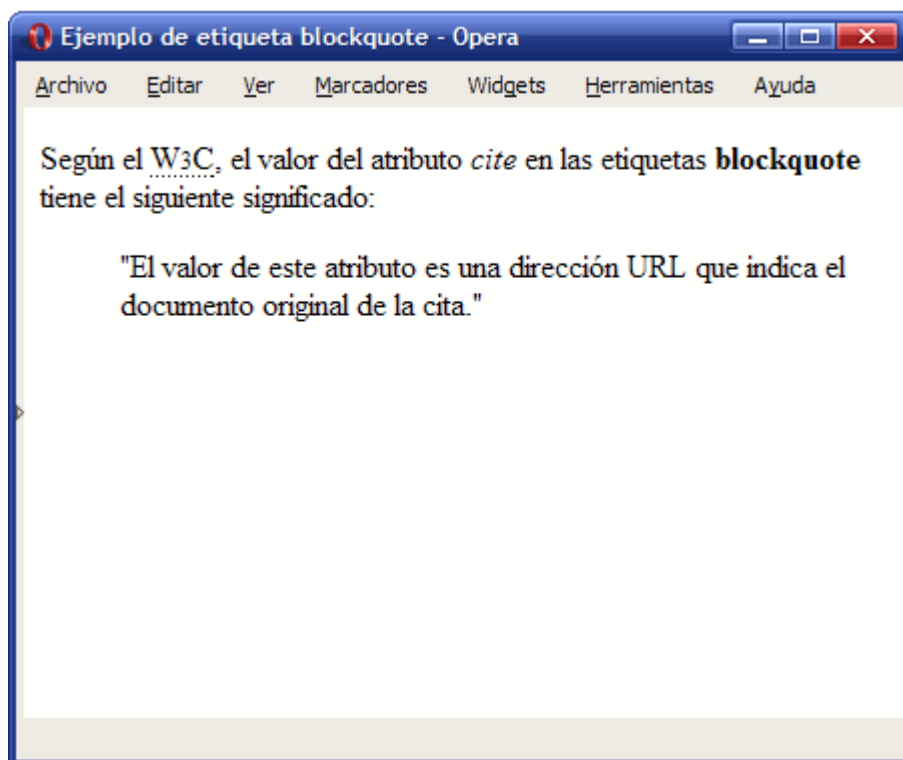


Figura 3.14. Ejemplo de uso de la etiqueta blockquote

Para indicar de forma clara que el texto es una cita externa, los navegadores muestran por defecto el texto del elemento `<blockquote>` con un gran margen en la parte izquierda.

3.5. Marcado avanzado de texto

Para elaborar documentos más correctos y completos, HTML incluye otras etiquetas que permiten marcar más elementos del texto.

La etiqueta `<abbr>` permite marcar las abreviaturas de un texto y la etiqueta `<acronym>` se emplea para marcar las siglas o acrónimos del texto. Su definición se la siguiente:

Tabla 3.11. Etiqueta `abbr`

abbr	Abreviaturas
Atributos comunes	básicos, <code>i18n</code> y eventos

abbr	Abreviaturas
Atributos específicos	<ul style="list-style-type: none"> • <code>title = "texto"</code> - Permite indicar el significado completo de la abreviatura
Tipo	En línea
Descripción	Se emplea para marcar las abreviaturas del texto y proporcionar el significado de esas abreviaturas

Tabla 3.12. Etiqueta acronym

acronym	Acrónimos o siglas
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • <code>title = "texto"</code> - Permite indicar el significado completo del acrónimo o sigla
Tipo	En línea
Descripción	Se emplea para marcar las siglas o acrónimos del texto y proporcionar el significado de esas siglas

El atributo `title` se puede utilizar para incluir el significado completo de la abreviatura o sigla.
Ejemplo:

```
<html>

<head>
<title>Ejemplo de etiqueta acronym</title>
</head>

<body>
```

```
<p>El lenguaje <acronym title="HyperText Markup Language">HTML</acronym> es
estandarizado
por el <acronym title="World Wide Web Consortium">W3C</acronym>.</p>
</body>

</html>
```

Los navegadores muestran por defecto un borde inferior punteado para todos los elementos `<abbr>` y `<acronym>`. Al posicionar el puntero sobre la palabra subrayada, el navegador muestra un pequeño recuadro (llamado *tooltip* en inglés) con el valor del atributo `title`:

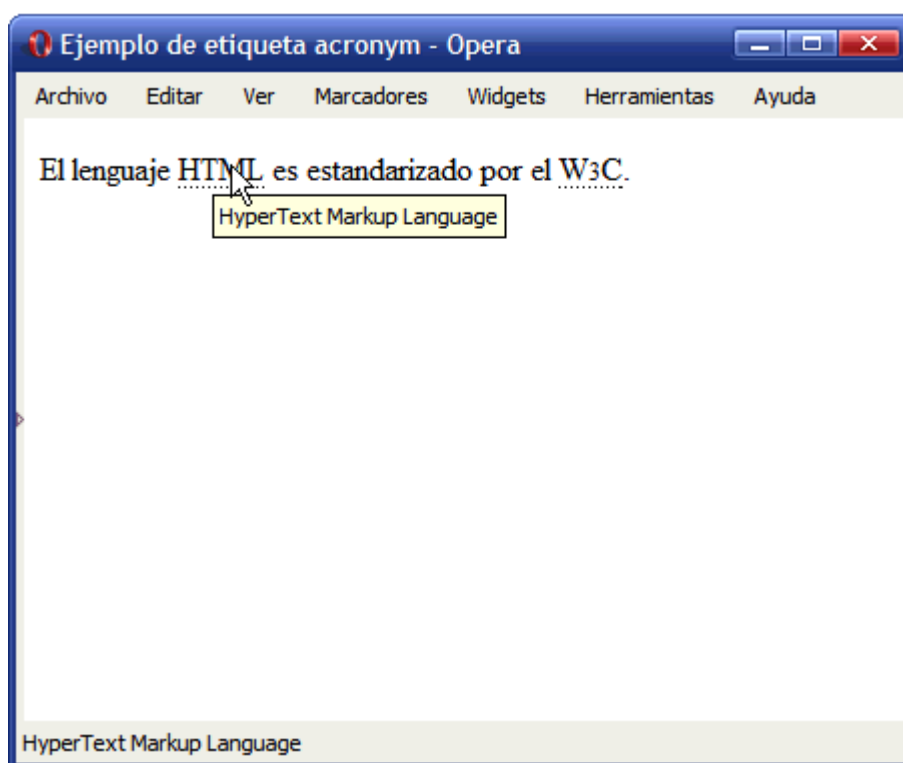


Figura 3.15. Ejemplo de uso de la etiqueta acronym

En ocasiones, puede resultar útil incluir la definición de una palabra extraña o cuyo uso está restringido a un entorno muy determinado. HTML incluye la etiqueta `<dfn>` para proporcionar al usuario la definición de todas las palabras para las que se considere apropiado. La definición formal de esta etiqueta se muestra a continuación:

Tabla 3.13. Etiqueta dfn

dfn	Definición
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • title = "texto" - Permite indicar el significado completo del término
Tipo	En línea
Descripción	Se emplea para marcar las definiciones de ciertos términos y proporcionar el significado de esos términos

El siguiente ejemplo muestra la definición de una palabra extraña fuera de los ámbitos médicos y psicológicos:

```
<p>Con estos síntomas, podría tratarse de un caso de <dfn title="Imagen o sensación subjetiva, propia de un sentido, determinada por otra sensación que afecta a un sentido diferente">sinestesia</dfn></p>
```

Por último, HTML incluye una etiqueta que se puede utilizar para marcar un texto como una citación:

Tabla 3.14. Etiqueta cite

cite	Cita
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	En línea
Descripción	Se emplea para marcar una cita o una referencia a otras fuentes

En ocasiones, no está clara la diferencia entre `<cite>` y `<blockquote>`. Una buena regla para recordar la diferencia entre ambas es que `<cite>` hace referencia a quien se cita y `<blockquote>` contiene la propia cita. Ejemplo:

Como dijo `<cite>Mahatma Gandhi</cite>`:

`<blockquote>`

Vive como si fueras a morir mañana y aprende como si fueras a vivir para siempre.

`</blockquote>`

Capítulo 4. Enlaces

El lenguaje de marcado HTML se definió teniendo en cuenta algunas de las características más habituales que existían en ese momento para la publicación de contenidos. Además, HTML incorporó una novedad respecto a la mayoría de lenguajes de marcado existentes y que supuso una revolución y el rotundo éxito de HTML.

El concepto incorporado por HTML es el “*hipertexto*” y la posibilidad de crear documentos interactivos. El elemento principal del hipertexto es el “*hiperenlace*”, también llamado “enlace web” o simplemente “enlace”.

Los enlaces se utilizan para establecer relaciones entre 2 recursos web. Aunque la mayoría de enlaces relacionan páginas web, también es posible enlazar otros recursos como imágenes, documentos, archivos, programas, etc.

Una característica que no se suele tener en cuenta en los enlaces es que están formados por 2 extremos y un sentido. En otras palabras, el enlace comienza en un extremo y apunta hacia el otro extremo. Cada uno de los dos extremos se llaman “*anchors*” en inglés, que se puede traducir literalmente como “anclas”.

4.1. URL

Antes de empezar a incluir enlaces en las páginas HTML creadas, es necesario comprender y dominar el concepto de URL. El acrónimo URL (del inglés *Uniform Resource Locator*) hace referencia al identificador único de cada recurso disponible en Internet. Las URL son esenciales para crear los enlaces, pero también se utilizan en otros elementos HTML como las imágenes y los formularios.

Cada página web que está publicada en Internet tiene un “nombre” único que permite diferenciarla de las demás. Ese “nombre único” es la URL de la página y coincide con la dirección que muestra el navegador para esa página. Si se accede a Google, la dirección que muestra el navegador es:

`http://www.google.com`

La cadena de texto `http://www.google.com` es la URL de la página principal de Google. La URL de una página es muy importante, ya que para crear un enlace que apunte a una determinada página, es imprescindible conocer su URL. Las partes que forman una URL sencilla son siempre las mismas. Si por ejemplo se considera la siguiente URL:

```
http://www.librosweb.es/xhtml/capitulo4.html
```

Las partes que componen la URL anterior son:

- Protocolo (`http://`): el mecanismo que debe utilizar el navegador para acceder a ese recurso. Todas las páginas web utilizan `http://`. Las páginas web *seguras* (por ejemplo las de los bancos, las de los servicios de email, etc.) utilizan `https://` (se añade una letra *s*).
- Servidor (`www.librosweb.es`): simplificando mucho su explicación, se puede decir que se trata del ordenador en el que se encuentra guardada la página que se quiere acceder.
- Ruta (`/xhtml/capitulo4.html`): “camino” que se debe seguir, una vez que se ha llegado al ordenador, para localizar la página concreta que se quiere acceder.

Aunque la mayoría de URL son similares a la mostrada anteriormente, las URL pueden llegar a ser muy complejas y estar formadas por más elementos:

```
http://www.456bereastreet.com/archive/200606/  
?title=standards_compliant_websites#comments
```

Las partes que forman la URL anterior son:

- Protocolo (`http://`)
- Servidor (`www.456bereastreet.com`)
- Ruta (`/archive/200606`)
- Consulta (`?title=standards_compliant_websites`): información adicional necesaria para que el servidor localice correctamente la página web que se quiere acceder
- Sección (`#comments`): además de acceder a una página web completa, también es posible hacer referencia a una sección específica dentro de una página web. En este ejemplo, se quiere acceder directamente a la sección `#comments` de la página web

Para construir las URL, se utilizan algunos caracteres como : y /. Por este motivo, algunos caracteres están reservados para uso exclusivo por parte de las URL. Además, algunos caracteres no están reservados por las URL, pero pueden ser problemáticos si se utilizan en la propia URL (como por ejemplo las comillas dobles).

De esta forma, si una URL incluye caracteres reservados y/o especiales, es necesario que lo indique de una forma especial. El proceso que se utiliza es la codificación y consiste en sustituir esos caracteres problemáticos por otros caracteres seguros. El servidor hará el proceso inverso (decodificación) cuando le llegue una URL con caracteres codificados.

A continuación se muestra la tabla para codificar los caracteres más comunes:

Carácter original	Carácter codificado
/	%2F
?	%3F
:	%3A
@	%40
=	%3D
&	%26
“	%22
\	%5C
'	%60
~	%7E
(espacio en blanco)	%20

Además, de momento las URL solo pueden contener caracteres válidos en el idioma inglés, por lo que no es posible utilizar caracteres como ñ, á, ç, etc. La codificación de este otro grupo de caracteres se muestra a continuación:

Carácter original	Carácter codificado
ñ	%F1

Ñ	%D1
á	%E1
é	%E9
í	%ED
ó	%F3
ú	%FA
Á	%C1
É	%C9
Í	%CD
Ó	%D3
Ú	%DA
ç	%E7
Ç	%C7

Teniendo en cuenta las 2 tablas de codificación de caracteres, es fácil crear las URL correctas sin caracteres problemáticos:

<!-- URL problemática -->

`http://www.ejemplo.com/ruta/español.html`

<!-- URL correcta -->

`http://www.ejemplo.com/ruta/espa%F1o1.html`

<!-- URL problemática -->

`http://www.ejemplo.com/ruta/nombre pagina.html`

<!-- URL correcta -->

`http://www.ejemplo.com/ruta/nombre%20pagina.html`

4.2. Enlaces relativos y absolutos

Las páginas web habituales suelen contener decenas de enlaces de diferentes tipos. La siguiente imagen muestra algunos de los tipos de enlaces de una página del sitio 456BereaStreet.com:



Figura 4.1. Ejemplo de diferentes tipos de enlaces en la página 456BereaStreet.com

En esta página, cuando se pincha sobre algunos enlaces, el navegador abandona el sitio web para acceder a una página que se encuentra en otro sitio. Estos enlaces se conocen como “enlaces externos”. Por otra parte, la mayoría de enlaces de un sitio web apuntan a páginas del propio sitio web, por lo que se denominan “enlaces internos”.

Además de interno/externo, la característica más importante de un enlace (y por tanto, de una URL) es si el enlace es absoluto o relativo. Las URL relativas prescinden del nombre del servidor y, opcionalmente, de la ruta hasta el documento. Aunque las URL relativas pueden ser difíciles de entender para los que comienzan con HTML, son tan útiles que todos los sitios web las utilizan.

Imagina que se dispone de una página cuya URL es `http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html` y en la que se quiere incluir un enlace a la página `http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina2.html`. Como se ha comentado, la URL identifica de forma única a cada página, por lo que para definir el enlace, se debería utilizar la URL completa de la segunda página.

Aunque el razonamiento es correcto, el resultado es poco eficiente, ya que la URL que se debe utilizar es demasiado larga. Como la URL de la página a la que se enlaza está en el mismo servidor, se puede obviar por completo el nombre del servidor y utilizar una URL más sencilla:

```
/ruta1/ruta2/ruta3/pagina2.html
```

La URL completa que incluye el nombre del servidor se llama **URL absoluta**. Cuando los 2 extremos de un enlace se encuentran en el mismo servidor, se puede obviar el nombre del servidor en las URL. Este tipo de URL abreviadas se llaman **URL relativas**. Cuando el navegador encuentra una URL que no tiene el nombre del servidor, supone que es un enlace interno y se lo añade automáticamente:

```
<!-- URL relativa -->
```

```
/ruta1/ruta2/ruta3/pagina2.html
```

```
<!-- Como no tiene nombre de servidor, el navegador
se lo añade -->
```

```
http://www.ejemplo.com
```

```
<!-- URL absoluta construida por el navegador -->
```

```
http://www.ejemplo.com
```

```
+
```

```
/ruta1/ruta2/ruta3/pagina2.html
```

```
=
```

```
http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina2.html
```

Cuando la URL a la que se enlaza no sólo está en el mismo servidor, sino que está en la misma ruta, la URL relativa se puede simplificar todavía más. En el ejemplo anterior, las 2 páginas se encuentran en la misma ruta `http://www.ejemplo.com/ruta1/ruta2/ruta3/`, por lo que la URL relativa se puede indicar simplemente como `pagina2.html`.

```
<!-- URL relativa -->  
pagina2.html
```

```
<!-- Como no tiene ni nombre de servidor ni ruta, el navegador  
  Las tiene que añadir -->
```

```
<!-- El nombre del servidor se añade igual que antes -->  
http://www.ejemplo.com
```

```
<!-- La ruta que se añade es la misma que la de la página en  
  la que se encuentra el enlace -->  
/ruta1/ruta2/ruta3/
```

```
<!-- URL absoluta construida por el navegador -->  
http://www.ejemplo.com  
+  
/ruta1/ruta2/ruta3/  
+  
pagina2.html  
=  
http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina2.html
```

El último tipo de URL relativa es el que más cuesta entender al principio. Siguiendo el mismo ejemplo anterior, se dispone de una página cuya URL es `http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html` y que quiere enlazar a la página que se encuentra en `http://www.ejemplo.com/ruta1/ruta2/ruta4/pagina2.html`. Como las rutas de las 2 páginas no son exactamente iguales, no se puede utilizar la misma estrategia que el ejemplo anterior y la URL relativa debería contener la ruta completa hasta la segunda página:

```
/ruta1/ruta2/ruta4/pagina2.html
```

Sin embargo, aunque las 2 rutas no son idénticas, son muy parecidas. De hecho, para ir desde la página 1 a la página 2 se puede pensar en “*sube un nivel desde la ruta de la página1 y luego entra en ruta4 para acceder a la pagina2*”. Las URL permiten indicar de forma sencilla la

acción de “subir un nivel” para después entrar en otra ruta. El símbolo que se emplea para indicar que se debe subir un nivel en la ruta es ../ (2 puntos y una barra).

De esta forma, la URL relativa en la página 1 podría ser ../ruta4/pagina2.html:

```
<!-- URL relativa -->
```

```
../ruta4/pagina2.html
```

```
<!-- Como no tiene ni nombre de servidor ni ruta, el navegador
```

```
las tiene que añadir -->
```

```
<!-- El nombre del servidor se añade igual que antes -->
```

```
http://www.ejemplo.com
```

```
<!-- La ruta que se añade se construye a partir de la URL
```

```
relativa (../ruta4/pagina2.html) -->
```

```
Ruta de la página 1 = /ruta1/ruta2/ruta3/
```

```
+
```

```
../ (subir un nivel)
```

```
=
```

```
/ruta1/ruta2
```

```
+
```

```
/ruta4/pagina2.html
```

```
=
```

```
/ruta1/ruta2/ruta4/pagina2.html
```

```
<!-- URL absoluta construida por el navegador -->
```

```
http://www.ejemplo.com
```

```
+
```

```
/ruta1/ruta2/ruta4/pagina2.html
```

```
=
```

```
http://www.ejemplo.com/ruta1/ruta2/ruta4/pagina2.html
```

Se pueden utilizar tantos ../ como niveles se quieran subir en la ruta. Si en el ejemplo anterior la ruta relativa fuera ../../ruta4/pagina2.html, la URL final construida es:

```
<!-- URL relativa -->
../../ruta4/pagina2.html

<!-- Como no tiene ni nombre de servidor ni ruta, el navegador
     Las tiene que añadir -->

<!-- El nombre del servidor se añade igual que antes -->
http://www.ejemplo.com

<!-- La ruta que se añade se construye a partir de La URL
     relativa (../../ruta4/pagina2.html) -->
Ruta de la página 1 = /ruta1/ruta2/ruta3/
+
../../ (subir dos niveles)
=
/ruta1
+
/ruta4/pagina2.html
=
/ruta1/ruta4/pagina2.html

<!-- URL absoluta construida por el navegador -->
http://www.ejemplo.com
+
/ruta1/ruta4/pagina2.html
=
http://www.ejemplo.com/ruta1/ruta4/pagina2.html
```

4.3. Enlaces básicos

Los enlaces en HTML se crean mediante la etiqueta `<a>` (su nombre viene del inglés “*anchor*”, literalmente traducido como “ancla”). La definición simplificada de `<a>` es la siguiente (en la siguiente sección se ve su definición completa):

Tabla 4.1. Etiqueta a

a	Enlaces
Atributos comunes	básicos, i18n, eventos y foco
Atributos específicos	<ul style="list-style-type: none"> • name = "texto" - Permite nombrar al enlace para que se pueda acceder desde otros enlaces • href = "url" - Indica la URL del recurso que se quiere enlazar
Tipo	En línea
Descripción	Se emplea para enlazar todo tipo de recursos

El atributo más importante de la etiqueta `<a>` es `href`, que se utiliza para indicar la URL a la que apunta el enlace. Cuando el usuario pincha sobre un enlace, el navegador se dirige a la URL del recurso indicado mediante `href`. Las URL de los enlaces pueden ser absolutas, relativas, internas y externas.

Con la definición anterior, para crear un enlace que apunte a la página principal de Google solamente habría que incluir lo siguiente en un documento HTML:

```
<a href="http://www.google.com">Página principal de Google</a>
```

Como el atributo `href` indica una URL, un enlace puede apuntar a cualquier recurso al que pueda acceder el navegador. El siguiente enlace apunta a una imagen, que se mostrará en el navegador cuando el usuario pinche sobre el enlace:

```
<a href="http://www.ejemplo.com/fondo_escritorio.jpg">Imagen interesante para un fondo de escritorio</a>
```

De la misma forma, los enlaces pueden apuntar directamente a documentos PDF, Word, etc.

```
<a href="http://www.ejemplo.com/informe.pdf">Descargar informe completo [PDF]</a>
```

```
<a href="http://www.ejemplo.com/informe.doc">Descargar informe completo [DOC]</a>
```

Un truco muy útil con los enlaces es el uso de URL relativas para poder volver al inicio del sitio web desde cualquier página web interior:

```
<a href="/">Volver al inicio</a>
```

El enlace anterior utiliza una URL relativa con una ruta que apunta directamente al nivel más alto del servidor. El navegador añadirá el nombre del servidor a la ruta y de esta forma, el enlace anterior siempre vuelve al inicio del sitio web independientemente de la página desde la que se pinche el enlace.

El otro atributo básico de la etiqueta `<a>` es `name`, que permite definir enlaces dentro de una misma página web. Si una página es muy larga, puede ser útil mostrar enlaces tipo `Saltar hasta la segunda sección`, `Volver al principio de la página`, etc.

Este tipo de enlaces son especiales, ya que la URL de la página siempre es la misma para todas las secciones y por tanto, debe añadirse algo a las URL para identificar cada sección.

```
<a name="primera_seccion"></a>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id ligula eu felis adipiscing ultrices. Duis gravida leo ut lectus. Praesent condimentum mattis ligula.</p>
```

...

```
<a name="segunda_seccion"></a>
```

```
<p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia a, convallis eu, nonummy et, odio. Aenean urna elit, ultrices id, placerat varius, facilisis eget, dolor.</p>
```

...

El atributo `name` permite crear “enlaces vacíos” que hacen referencia a secciones dentro de una misma página. Una vez definidos los “enlaces vacíos”, es posible crear un enlace que apunte directamente a una sección concreta de una página:

```
<!-- Enlace normal a la página -->
```

```
<a href="http://www.ejemplo.com/pagina1.html">Enlace a la página 1</a>
```

```
<!-- Enlace directo a la segunda sección de la página -->
```

```
<a href="http://www.ejemplo.com/pagina1.html#segunda_seccion">Enlace a la sección
2 de la página 1</a>
```

La sintaxis que se utiliza con estos enlaces es la misma que con los enlaces normales, salvo que se añade el símbolo # seguido del nombre de la sección a la que se apunta. Cuando el usuario pincha sobre uno de estos enlaces, el navegador accede a la página apuntada por la URL y baja directamente a la sección cuyo nombre se indica después del símbolo #.

También es posible utilizar este tipo de enlaces con URL relativas en una misma página. El siguiente ejemplo añade enlaces de tipo *Volver al inicio* de la página en varias secciones:

```
<a name="inicio"></a>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id ligula eu
felis adipiscing ultrices. Duis gravida leo ut lectus. Praesent condimentum mattis
ligula.</p>
```

```
<a href="#inicio">Volver al inicio de la página</a>
```

```
...
```

```
<p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia a, convallis
eu, nonummy et, odio. Aenean urna elit, ultrices id, placerat varius, facilisis
eget, dolor.</p>
```

```
<a href="#inicio">Volver al inicio de la página</a>
```

```
...
```

El uso de los enlaces directos a secciones también es posible sin utilizar el atributo `name` de las etiquetas `<a>`, ya que también funcionan con el atributo `id` de cualquier elemento. El siguiente ejemplo es equivalente al ejemplo anterior:

```
<h1 id="inicio">Título de la página</h1>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id ligula eu
felis adipiscing ultrices. Duis gravida leo ut lectus. Praesent condimentum mattis
ligula.</p>
```



```
<a href="#inicio">Volver al inicio de la página</a>
```

...

```
<p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia a, convallis eu, nonummy et, odio. Aenean urna elit, ultrices id, placerat varius, facilisis eget, dolor.</p>
```

```
<a href="#inicio">Volver al inicio de la página</a>
```

...

El nombre de la sección indicado después de # en el atributo href puede utilizar el valor de los atributos id de cualquier elemento. De hecho, se recomienda utilizar los atributos id de los elementos ya existentes en la página en vez de crear “enlaces vacíos” de tipo `...`.

4.4. Enlaces avanzados

Aunque la etiqueta `<a>` para los enlaces básicos es muy sencilla, su definición completa tiene muchos otros atributos que pueden ser útiles para los enlaces más avanzados:

Tabla 4.2. Etiqueta a

a	Enlaces
Atributos comunes	básicos, i18n, eventos y foco
Atributos específicos	<ul style="list-style-type: none"> • name = "texto" - Permite nombrar al enlace para que se pueda acceder desde otros enlaces • href = "url" - Indica la URL del recurso que se quiere enlazar • hreflang = "codigo_idioma" - Idioma del recurso enlazado

a	Enlaces
	<ul style="list-style-type: none"> • type = "tipo_de_contenido" - Permite "avisar" al navegador sobre el tipo de contenido que se enlaza (imágenes, archivos, etc.) para que pueda prepararse en caso de que no entienda ese contenido • rel = "tipo_de_relacion" - Describe la relación del documento actual con el recurso enlazado • rev = "tipo_de_relacion" - Describe la relación del recurso enlazado con el documento actual • charset = "tipo_de_charset" - Describe la codificación del recurso enlazado
Tipo	En línea
Descripción	Se emplea para enlazar todo tipo de recursos

4.4.1. Idioma del enlace (hreflang)

El enlace puede indicar al navegador el idioma del recurso que se enlaza. Para establecer el valor del idioma, se utiliza un código de 2 letras estandarizado. Además del idioma genérico, también se puede indicar la variación idiomática específica. Ejemplo de códigos de idioma más utilizados:

Código	Idioma	Variación idiomática
en	Inglés	-
en-US	Inglés	Estados Unidos
es	Español	-
es-ES	Español	España
es-AR	Español	Argentina

Otros códigos utilizados son: `fr` (francés), `de` (alemán), `it` (italiano), `nl` (holandés), `el` (griego), `es` (español), `pt` (portugués), `ar` (árabe), `he` (hebreo), `ru` (ruso), `zh` (chino), `ja` (japonés).

La lista completa de códigos de idioma está definida en el estándar ISO 639.

4.4.2. Tipo de contenido (`type`)

Se utiliza para “dar una pista” al navegador sobre el tipo de contenido que se enlaza. Se indica mediante una cadena de texto cuyos valores también están estandarizados. Los valores de los contenidos más utilizados son los siguientes: “`text/html`” (páginas HTML), “`image/png`” (imágenes con formato PNG), “`image/gif`” (imágenes con formato GIF), “`text/css`” (hojas de estilo CSS), “`application/rss+xml`” (archivos RSS).

La lista completa de tipos de contenido se define en los estándares RFC 2045 y RFC 2046.

4.4.3. Tipo de relación (`rel` y `rev`)

Los enlaces pueden proporcionar información adicional muy útil para los navegadores y para los motores de búsqueda como Google. Los atributos `rel` y `rev` permiten indicar la relación que la página actual tiene con la página a la que se enlaza (atributo `rel`) y la relación que tiene la página enlazada con la página actual (atributo `rev`).

Los tipos de relación definidos son los siguientes:

- `alternate` – Indica que es una versión alternativa al documento actual (puede ser una versión en otro idioma o una versión preparada para otro medio)
- `stylesheet` – Indica que se ha enlazado una hoja de estilos
- `start` – Indica que se trata del primer documento de una colección de documentos (por ejemplo el primer capítulo de un libro)
- `next` – Indica que es el documento que sigue al actual dentro de una secuencia lógica de documentos (por ejemplo, los capítulos de un libro)
- `prev` – Indica que es el documento que precede al actual dentro de una secuencia lógica de documentos (por ejemplo, los capítulos de un libro)

- `contents` – Indica que el recurso enlazado es el documento que contiene la tabla de contenidos de la colección de documentos (por ejemplo, el índice de un libro).
- `bookmark` – Establece el enlace actual como un “marcador” o “favorito”. Un marcador es un enlace que constituye un punto de entrada muy importante dentro del documento.

La especificación oficial de HTML define la lista completa de tipos de relaciones que se pueden utilizar.

4.4.4. Codificación de caracteres (charset)

Además del idioma, tipo de contenido y relación del recurso que se enlaza, los enlaces también pueden indicar la codificación de caracteres que utiliza la página web enlazada.

Los valores que se pueden utilizar también están estandarizados y los valores de las codificaciones más utilizadas son UTF-8 y ISO-8859-1, aunque existen decenas de códigos definidos (ISO-10646-UCS-2, IBM852, Big5-HKSCS, windows-1252, HZ-GB-2312).

El organismo IANA publica la lista completa de codificaciones de caracteres disponibles.

Si se utilizan todos estos atributos en los enlaces, los ejemplos de enlaces básicos se pueden rehacer de la siguiente manera:

```
<a href="http://www.google.com" hreflang="en" type="text/html"
charset="UTF-8">Página principal de Google</a> <a href="http://www.ejemplo.com/
fondo_escritorio.jpg" type="image/jpg">Imagen interesante para un fondo de
escritorio</a>
```

4.5. Otros tipos de enlaces

Además de `<a>`, HTML define otras 2 etiquetas para enlazar recursos desde una página web. La primera es la etiqueta `<script>` que sirve tanto para insertar un bloque de código JavaScript en la página como para enlazar un archivo JavaScript externo.

Tabla 4.3. Etiqueta script

script	Código ejecutable
Atributos comunes	-
Atributos específicos	<ul style="list-style-type: none"> • <code>src = "url"</code> - Indica la dirección del archivo que contiene el código • <code>type = "tipo_de_contenido"</code> - Permite "avisar" al navegador sobre el tipo de código que se incluye (normalmente JavaScript) • <code>defer = "defer"</code> - El código no va a modificar el contenido del documento • <code>charset = "tipo_de_charset"</code> - Describe la codificación del código enlazado
Tipo	Bloque y en línea (también puede ser una etiqueta vacía)
Descripción	Se emplea para enlazar o definir un bloque de código (normalmente JavaScript)

El uso más sencillo de la etiqueta `<script>` es el de enlazar un archivo JavaScript externo:

```
<head>
  <script type="text/javascript" src="http://www.ejemplo.com/js/inicializar.js" />
</head>
```

El atributo `type` siempre es el mismo para los archivos JavaScript (`"text/javascript"`). El atributo `src` es el equivalente del atributo `href` de los enlaces normales. La URL indicada en el atributo `src` puede ser absoluta o relativa.

Además de enlazar un archivo JavaScript externo, también es posible incluir en la página web un bloque de código JavaScript:

```
<html>
<head>
  <script type="text/javascript">
```

```

<![CDATA[
  window.onload = function() { alert("La página se ha cargado completamente"); }
]]>
</script>
</head>
<body>
...
</body>
</html>

```

La etiqueta `<script>` (tanto cuando enlaza, como cuando incluye directamente el código) puede aparecer en cualquier parte del documento HTML, aunque normalmente se incluye dentro de la cabecera de la página (`<head>...</head>`).

La segunda etiqueta para enlazar recursos es `<link>`, que permite enlazar y relacionar unos recursos con otros y es muy similar a la etiqueta `<a>`.

Tabla 4.4. Etiqueta link

link	Enlazar recursos
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> - Los siguientes con el mismo significado que para la etiqueta "a": charset, href, hreflang, type, rel y rev • <code>media = "tipo_de_medio"</code> - Indica el medio para el que debe aplicarse la relación
Tipo	Etiqueta vacía
Descripción	Se emplea para enlazar y establecer relaciones entre el documento y otros recursos

Al contrario que `<a>` y `<script>`, la etiqueta `<link>` solamente se puede incluir dentro de la cabecera del documento. Se pueden añadir tantas etiquetas `<link>` como hagan falta, pero siempre dentro de `<head>...</head>`.

El uso habitual de la etiqueta `<link>` es el de enlazar las hojas de estilos CSS utilizadas por las páginas web:

```
<head>
  ...
  <link rel="stylesheet" type="text/css" href="/css/comun.css" />
</head>
```

En este caso, es habitual establecer los atributos `rel` y `type` para indicar el tipo de recurso enlazado y su relación con la página web. La URL del recurso enlazado se indica en el atributo `href`, que admite tanto URL absolutas como relativas.

4.6. Ejemplos de enlaces habituales

4.6.1. Enlace al inicio del sitio web

```
<a href="/">Inicio</a>
```

Al pulsar el enlace anterior desde cualquier página web, se vuelve directamente a la página de inicio del sitio web.

4.6.2. Enlace a un email

```
<a href="mailto://nombre@direccion.com" title="Dirección de email para solicitar
más información">
Solicita más información
</a>
```

Al pinchar sobre el enlace anterior, se abre automáticamente el programa de correo electrónico del ordenador del usuario y se establece la dirección de envío al valor indicado después de `mailto://`. La sintaxis es la misma que la de un enlace normal, salvo que se cambia el prefijo `http://` por `mailto://`.

La sintaxis de `mailto://` permite utilizarlo para otros ejemplos más complejos:

```
<!-- Envío del correo electrónico a varias direcciones a la vez -->
<a href="mailto://nombre@direccion.com,otro_nombre@direccion.com">Solicita más
información</a>
```

```
<!-- Añadir un "asunto" inicial al correo electrónico -->
<a href="mailto://nombre@direccion.com?subject=Solicitud de más
información">Solicita más información</a>
```

```
<!-- Añadir un texto inicial en el cuerpo del correo electrónico -->
<a href="mailto://nombre@direccion.com?body=Estaría interesado en solicitar más
información sobre sus productos">Solicita más información</a>
```

Todas las opciones anteriores se pueden combinar entre sí para realizar ejemplos más avanzados. Aunque el uso de `mailto://` puede parecer una ventaja, su uso está desaconsejado. Si se incluye una dirección de correo electrónico directamente en una página web, es muy probable que en poco tiempo esa dirección de email se encuentre llena de correo electrónico basura, ya que existen programas automáticos encargados de rastrear sistemáticamente todas las páginas web de Internet para encontrar direcciones de correo electrónico válidas.

La forma de mostrar las direcciones de correo electrónico en las páginas web consiste en incluir la dirección en una imagen o indicarla de forma que solamente los usuarios puedan entenderlo:

```
<p>La dirección de correo es <strong>nombre (arroba) direccion.com</strong></p>
<p>La dirección de correo es <strong>nombre_arroba_direccion.com</strong></p>
<p>La dirección de correo es <strong>nombreQUITAESTOdireccion.com</strong></p>
<p>La dirección de correo es <strong>nombre(ARROBA)direccion.com</strong></p>
<p>La dirección de correo es <strong>nombre @ direccion . com</strong></p>
```

4.6.3. Enlace a un archivo FTP

Para enlazar un archivo almacenado en un servidor FTP, solamente es necesario cambiar el prefijo `http://` por `ftp://`:

```
<a href="ftp://ftp.ejemplo.com/ruta/archivo.zip" title="Archivo comprimido de los
contenidos">
```


Descarga un ZIP con todos los contenidos

4.6.4. Enlazar varias hojas de estilos CSS

```
<link rel="stylesheet" type="text/css" href="/css/comun.css" />
```

```
<link rel="stylesheet" type="text/css" href="/css/secciones.css" />
```

4.6.5. Enlazar hojas de estilos CSS para diferentes medios

```
<link rel="stylesheet" type="text/css" href="/css/comun.css" media="screen,  
projection" />
```

```
<link rel="stylesheet" type="text/css" href="/css/impresora.css" media="print" />
```

```
<link rel="stylesheet" type="text/css" href="/css/movil.css" media="handheld" />
```

4.6.6. Enlazar el favicon

El *favicon* o icono para favoritos es el pequeño icono que muestran las páginas en varias partes del navegador. Dependiendo del navegador que se utilice, este icono se muestra en la barra de direcciones, en la barra de título del navegador y/o en el menú de favoritos/marcadores.

```
<link rel="shortcut icon" href="/favicon.ico" type="image/ico" />
```

Aunque en principio la imagen debería ser de tipo .ico (formato gráfico de los iconos), algunos navegadores soportan favicons en otros formatos gráficos más habituales (como por ejemplo .PNG).

4.6.7. Enlazar un archivo RSS

```
<link rel="alternate" type="application/rss+xml" title="Resumen de todos los  
artículos del blog" href="/feed.xml" />
```

4.6.8. Enlazar hojas de estilos, favicon y RSS

En una misma página se pueden incluir varias etiquetas <link>, por lo que es habitual que las páginas enlacen hojas de estilos, favicon y archivos RSS de forma conjunta:

```
<head>
```

```
...
```

```

<link rel="stylesheet" type="text/css" href="/css/impresora.css" media="print" />
<link rel="stylesheet" type="text/css" href="/css/movil.css" media="handheld" />
<style type="text/css" media="screen,projection">
  @import '/css/main.css';
</style>
<link rel="shortcut icon" href="/favicon.ico" type="image/ico" />
<link rel="alternate" type="application/rss+xml" title="Resumen de todos los
artículos del blog" href="/feed.xml" />
...
</head>

```

4.6.9. Indicar que existe una versión de la página en otro idioma

```

<head>
<title>The manual in English</title>
<link lang="es" xml:lang="es" title="El manual en español"
  type="text/html"
  rel="alternate"
  hreflang="es"
  href="http://www.ejemplo.com/manual/espanol.html" />
</head>

```

4.6.10. Indicar que existe una versión de la página preparada para imprimir

```

<head>
<title>Manual</title>
<link media="print" title="El manual en PDF"
  type="application/pdf"
  rel="alternate"
  href="http://www.ejemplo.com/manual/manualcompleto.pdf" />
</head>

```

4.6.11. Indicar que existe una página que es índice de la página actual

```
<head>
<title>Manual - Capítulo 5</title>
<link rel="start" title="El índice del manual"
      type="text/html"
      href="http://www.ejemplo.com/manual/indice.html" />
</head>
```

Capítulo 5. Listas

En ocasiones, es necesario agrupar determinadas palabras o frases en un conjunto de elementos que tienen significado de forma conjunta. Un menú por ejemplo se compone de un grupo de categorías o secciones, una serie de componentes pueden constituir las partes de un producto, las instrucciones mostradas al usuario pueden estar formadas por una serie de pasos ordenados, etc.

El lenguaje HTML define 3 tipos diferentes de listas para agrupar los elementos: listas no ordenadas (se trata de una colección simple de elementos en la que no importa su orden), listas ordenadas (similar a la anterior, pero los elementos están numerados y por tanto, importa su orden) y listas de definición (un conjunto de términos y definiciones similar a un diccionario).

5.1. Listas no ordenadas

Se trata del tipo más simple de lista y el que se emplea mayoritariamente. La lista no ordenada es un conjunto de elementos relacionados entre sí pero para los que no se indica un orden o secuencia determinados. La etiqueta `` define la lista y la etiqueta `` cada uno de los elementos de la lista.

Tabla 5.1. Etiqueta ul

ul	Lista no ordenada
Atributos comunes	básicos, <code>i18n</code> y eventos
Atributos específicos	-
Tipo	Bloque
Descripción	Se emplea para definir listas no ordenadas

Tabla 5.2. Etiqueta li

li	Elemento de una lista
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	Bloque
Descripción	Se emplea para definir los elementos de las listas (ordenadas y no ordenadas)

Ejemplo

```
<html>
<head><title>Ejemplo de etiqueta UL</title></head>
<body>

<h1>Menú</h1>

<ul>
  <li>Inicio</li>
  <li>Noticias</li>
  <li>Artículos</li>
  <li>Contacto</li>
</ul>

</body>
</html>
```



Figura 5.1. Ejemplo de uso de la etiqueta ul

El navegador por defecto muestra los elementos de la lista tabulados y con una pequeña viñeta formada por un círculo negro. La viñeta de la lista se puede modificar mediante CSS seleccionándola entre unos valores predefinidos (cuadrado, círculo vacío, etc) o creando una imagen específica para utilizarla como viñeta personalizada.

5.2. Listas ordenadas

Las listas ordenadas son casi idénticas a las listas no ordenadas, salvo que en este caso los elementos relacionados se muestran siguiendo un orden o secuencia determinados. Cuando se crea por ejemplo una lista con las instrucciones de un producto, es importante el orden en el que se realizan las instrucciones. Cuando se muestra un índice o tabla de contenidos en un libro, es importante el orden de cada elemento del índice. En todos estos casos, la lista más adecuada es la lista ordenada, que se define mediante la etiqueta ``. Los elementos de la lista se definen mediante la etiqueta ``, la misma que se utiliza en las listas no ordenadas.

Tabla 5.3. Etiqueta ol

ol	Lista ordenada
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	Bloque
Descripción	Se emplea para definir listas ordenadas

Ejemplo

```
<html>
<head><title>Ejemplo de etiqueta OL</title></head>
<body>

<h1>Instrucciones</h1>

<ol>
  <li>Enchufar correctamente</li>
  <li>Comprobar conexiones</li>
  <li>Encender el aparato</li>
</ol>

</body>
</html>
```

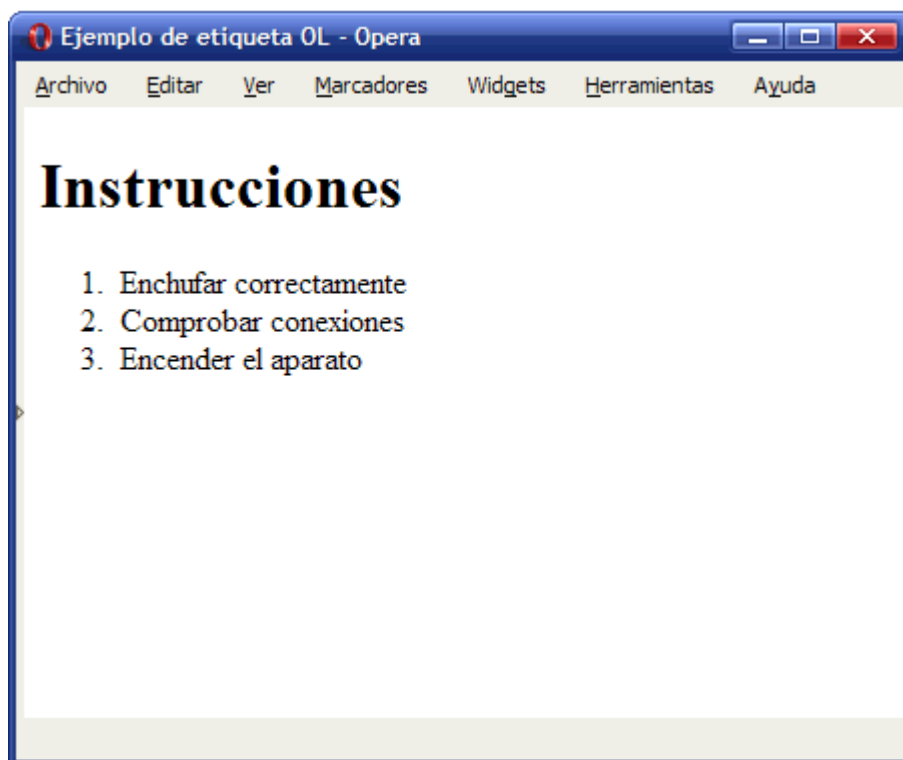


Figura 5.2. Ejemplo de uso de la etiqueta ol

El navegador muestra la lista de forma muy parecida a las listas no ordenadas, salvo que en este caso no se emplean viñetas genéricas en los elementos, sino que se numeran de forma consecutiva. Mediante CSS se puede seleccionar el tipo de numeración empleada (i, ii, iii, ... ; a, b, c; etc.)

5.3. Listas de definición

Se trata del tipo de lista menos empleado, e incluso muchos creadores de páginas web desconocen que existe este tipo de lista. Su funcionamiento es similar al de un diccionario, ya que cada elemento de la lista está formado por términos y definiciones. La etiqueta `<dl>` crea la lista de definición y las etiquetas `<dt>` y `<dd>` definen respectivamente el término y la descripción de cada elemento de la lista.

Tabla 5.4. Etiqueta dl

dl	Lista de definición
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	Bloque
Descripción	Se emplea para definir listas de definición

Tabla 5.5. Etiqueta dt

dt	Término de una definición
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	Bloque
Descripción	Se emplea para definir los términos de los elementos de una lista de definición

Tabla 5.6. Etiqueta dd

dd	Lista de definición
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	Bloque

dd	Lista de definición
Descripción	Se emplea para indicar las definiciones de los elementos de una lista de definición

Ejemplo

```
<html>
<head><title>Ejemplo de etiqueta DL</title></head>
<body>
<h1>Metalenguajes</h1>

<dl>
  <dt>SGML</dt>
  <dd>Metalenguaje para la definición de otros lenguajes de marcado</dd>

  <dt>XML</dt>
  <dd>Lenguaje basado en SGML y que se emplea para describir datos</dd>

  <dt>RSS</dt>
  <dt>GML</dt>
  <dt>XHTML</dt>
  <dt>SVG</dt>
  <dt>XUL</dt>
  <dd>Lenguajes derivados de XML para determinadas aplicaciones</dd>
</dl>

</body>
</html>
```

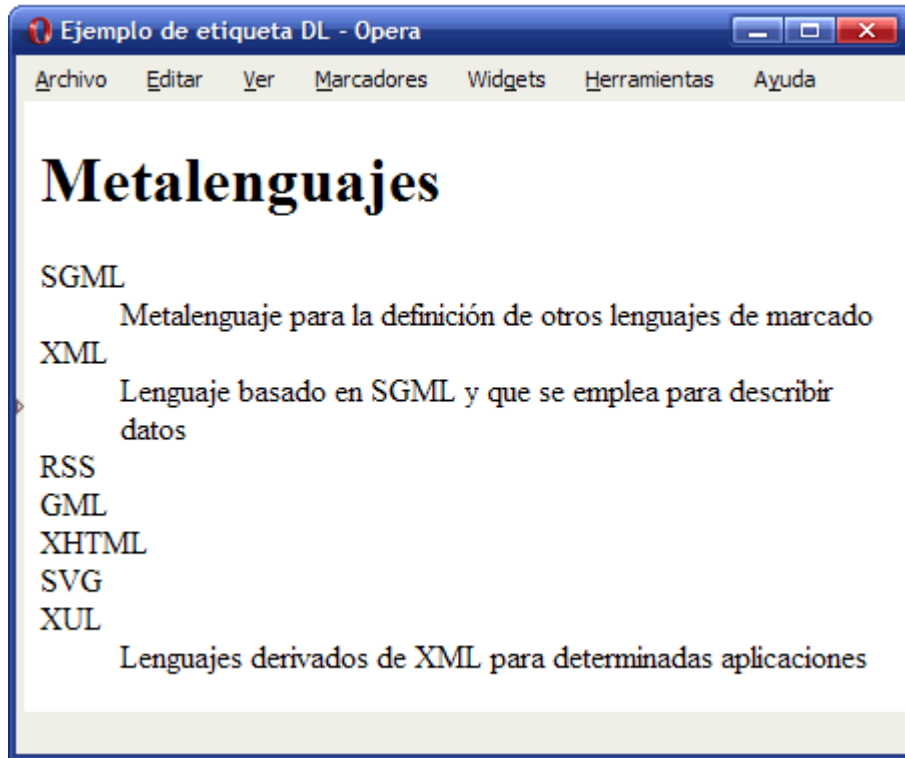


Figura 5.3. Ejemplo de uso de la etiqueta dl

Los navegadores formatean las listas de definición de forma similar a las otras listas, tabulando la definición y alineando a la izquierda los términos. Aunque no es habitual, cada término puede tener asociada más de una definición y cada definición puede tener asociada varios términos.

Capítulo 6. Imágenes y objetos

6.1. Imágenes

Las imágenes son uno de los elementos más importantes de las páginas web. De hecho, prácticamente todas las páginas web contienen alguna imagen y la mayoría incluyen decenas de imágenes. Dentro de las imágenes que se pueden incluir en una página HTML se deben distinguir 2 tipos: las imágenes de contenido y las imágenes de adorno.

Las imágenes de contenido son las que proporcionan información, complementan la información contextual, etc. Las imágenes de adorno son las que se utilizan para hacer bordes redondeados, para mostrarlas en una lista de elementos, las que se emplean como fondo de página, etc. Las imágenes de contenido se incluyen directamente en el código HTML mediante la etiqueta `` y las imágenes de adorno no se incluyen en el código HTML, sino que se emplean hojas de estilos CSS para mostrarlas.

La etiqueta `` es la que se utiliza para incluir las imágenes directamente en las páginas HTML:

Tabla 6.1. Etiqueta `img`

img	Imagen
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • <code>src = "url"</code> - Indica la URL de la imagen que se incluye • <code>alt = "texto"</code> - Descripción corta de la imagen • <code>longdesc = "url"</code> - Indica una URL en la que puede encontrarse una descripción más detallada de la imagen • <code>name = "texto"</code> - Nombre del elemento imagen

img	Imagen
	<ul style="list-style-type: none"> • <code>height = "unidad_de_medida"</code> - Indica la altura con la que se debe mostrar la imagen (no es obligatorio que coincida con la altura original de la imagen) • <code>width = "unidad_de_medida"</code> - Indica la anchura con la que se debe mostrar la imagen (no es obligatorio que coincida con la anchura original de la imagen)
Tipo	En línea y etiqueta vacía
Descripción	Se emplea para incluir imágenes en los documentos

Los 2 atributos requeridos son `src` y `alt`. El atributo `src` es similar al atributo `href` de los enlaces, ya que establece la URL de la imagen que se va a mostrar en el documento. Las URL indicadas pueden ser absolutas o relativas. El atributo `alt` proporciona un texto alternativo a la imagen. La finalidad del atributo `alt` es describir muy brevemente la imagen a la que hace referencia (la descripción debe ser menor que 1.024 caracteres).

Ejemplo de inclusión simple de una imagen:

```

```

Como es una etiqueta vacía, `` no tiene etiqueta de cierre, pero debe cerrarse para que el documento HTML sea válido. Como ya se vio anteriormente, la forma de cerrar una etiqueta vacía es mediante el símbolo `/>` al final de la etiqueta.

HTML no impone ninguna restricción sobre el formato gráfico que se puede utilizar en las imágenes, por lo que en principio la etiqueta `` puede incluir cualquier formato gráfico existente. Sin embargo, si la imagen utiliza un formato poco habitual, todos o algunos navegadores no serán capaces de mostrar esa imagen.

La recomendación es utilizar uno de los 3 siguientes formatos gráficos que entienden todos los navegadores modernos: GIF, JPG y PNG. El único inconveniente con estos formatos es que Internet Explorer 6 y las versiones anteriores no muestran correctamente las imágenes en formato PNG con transparencia de 24 bits (lo que se conoce como "canal alpha").

El atributo `longdesc` no se utiliza de forma habitual, pero permite indicar una URL en la que se puede mostrar más información sobre la imagen. Como la información adicional suele ser en forma de texto, es algo muy útil cuando la imagen es compleja para describirla y la persona que accede al contenido tiene algún tipo de discapacidad visual:

```

```

```

```

En el ejemplo anterior, las 2 imágenes se encuentran en el mismo directorio del servidor (`/imagenes/`). Se trata de una estrategia habitual en la mayoría de sitios web: agrupar todas las imágenes de contenido en un directorio especial independiente del resto de contenidos HTML. Además, el directorio siempre suele llamarse de la misma manera: `imagenes` o `images` en inglés.

Los atributos `width` y `height` se utilizan para indicar la anchura y altura con la que se muestran las imágenes, por lo que son los más contradictorios. Como ya se ha comentado, HTML estructura de forma correcta los contenidos de la página y CSS define el aspecto gráfico con el que se muestran los contenidos. En principio, la anchura y la altura con la que se muestra una imagen es parte de su aspecto gráfico, por lo que debería ser propio de CSS y no de XHTML.

Sin embargo, en la práctica no es viable establecer la anchura y altura de las imágenes de contenidos mediante CSS. Si el sitio web dispone de muchas imágenes, la sobrecarga de estilos diferentes para CSS sería contraproducente. Por este motivo, los atributos `width` y `height` son la excepción a la norma de que el código HTML no haga referencia al aspecto de los contenidos.

```

```

```

```

Si el valor del atributo `width` o `height` se indica mediante un número entero, se sobreentiende que hace referencia a la unidad de medida píxel. Por tanto, en el ejemplo anterior, la primera foto se muestra con una anchura de 200 píxel y una altura de 350 píxel.

También es posible indicar la anchura y altura en forma de porcentaje. En este caso, el porcentaje hace referencia a la altura/anchura del elemento en el que está contenida la imagen. Si la imagen no se encuentra dentro de ningún otro elemento, hace referencia a la anchura/altura total de la página.

```
<div>
  
</div>
```

El ejemplo anterior mezcla los 2 tipos de medidas que se pueden utilizar, para indicar que la foto tiene una anchura igual al 30% de la anchura del `<div>` que la contiene y una altura de 350 píxel.

La anchura/altura con la que se muestra una imagen no tiene que coincidir obligatoriamente con la anchura/altura real de la imagen. Sin embargo, cuando estos valores no coinciden, las imágenes se muestran deformadas y el aspecto final es muy desagradable.

6.2. Mapas de imagen

Aunque el uso de los mapas de imagen se ha reducido drásticamente en los últimos años, aun se utilizan en algunos sitios especializados. Muchas agencias de viaje y sitios relacionados utilizan mapas geográficos para seleccionar el destino del viaje. La mayoría de mapas se realiza hoy en día mediante Flash, aunque algunos siguen recurriendo a los mapas de imagen.

Un mapa de imagen permite definir diferentes zonas “*pinchables*” dentro de una imagen. El usuario puede pinchar sobre cada una de las zonas definidas y cada una puede establecer su propia URL. Siguiendo el ejemplo anterior, una sola imagen que muestre un mapa de todos los continentes puede definir una zona para cada continente. De esta forma, el usuario puede pinchar sobre la zona correspondiente a cada continente para ver los viajes disponibles a ese destino.

Las zonas o regiones que se pueden definir en una imagen se crean mediante rectángulos, círculos y polígonos. Cada mapa de imagen se define mediante una imagen (etiqueta ``) y mediante la etiqueta `<map>`. Dentro de un mapa de imagen, las regiones se definen mediante la etiqueta `<area>`.

Tabla 6.2. Etiqueta map

map	Mapa de imagen
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • name = "texto" - Nombre que identifica de forma única al mapa definido (es obligatorio indicar un nombre único)
Tipo	Bloque y en línea
Descripción	Se emplea para definir mapas de imagen

Tabla 6.3. Etiqueta area

area	Area de un mapa de imagen
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • href = "url" - URL a la que se accede al pinchar sobre el área • nohref = "nohref" - Se emplea para las áreas que no son seleccionables • shape = "default rect circle poly" - Indica el tipo de área que se define (toda la imagen, rectangular, circular o poligonal) • coords = "lista de números" - Se trata de una lista de números separados por comas que representan las coordenadas del área. Rectangular = X1,Y1,X2,Y2 (coordenadas X e Y del vértice superior izquierdo y coordenadas X e Y del vértice inferior derecho). Circular = X1,Y1,R (coordenadas X e Y del centro y radio del círculo). Poligonal = X1,Y1,X2,Y2,...,XnYn (coordenadas de los vértices del polígono. Si las últimas coordenadas no son iguales

area	Area de un mapa de imagen
	<p>que las primeras, se cierra automáticamente el polígono uniendo ambos vértices)</p> <ul style="list-style-type: none"> • Otros: <code>accesskey</code>, <code>tabindex</code> -
Tipo	Etiqueta vacía
Descripción	Se emplea para definir las distintas áreas que forman un mapa de imagen

Si una imagen utiliza un mapa de imagen, debe indicarlo mediante el atributo `usemap`. El valor del atributo debe ser el nombre del mapa de imagen definido en otra parte del mismo documento HTML:

```

...
<map name="continentes">
  ...
</map>
```

Las áreas se definen mediante el atributo `shape` que indica el tipo de área y `coords` que es una lista de coordenadas cuyo significado depende del tipo de área definido. El enlace de cada área se define mediante el atributo `href`, con la misma sintaxis y significado que para los enlaces normales.

El siguiente ejemplo muestra una imagen sencilla en la que se han definido 4 regiones correspondientes a las 4 figuras geométricas que muestra:

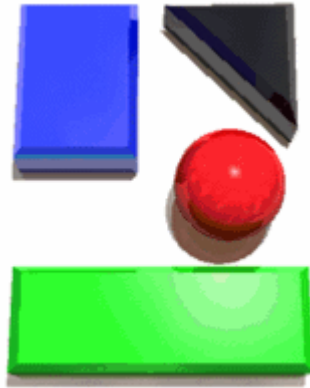


Figura 6.1. Ejemplo de imagen que incluye un mapa de imagen

```


<map name="mapa_zonas">
  <area shape="rect" coords="20,25,84,113" href="rectangulo.html" />
  <area shape="polygon" coords="90,25,162,26,163,96,89,25,90,24"
href="triangulo.html" />
  <area shape="circle" coords="130,114,29" href="circulo.html" />
  <area shape="rect" coords="19,156,170,211"
href="mailto:rectangulo@direccion.com" />
  <area shape="default" nohref="nohref" />
</map>
```

6.3. Objetos

Además de las imágenes, HTML permite incluir en las páginas otros elementos mucho más complejos, como applets de Java, vídeos en formato QuickTime o Flash, etc. Normalmente, este tipo de contenidos no los interpreta el navegador directamente, sino que hace uso de pequeños programas llamados plugins y que se encargan de tratar con este tipo de elementos complejos.

La etiqueta `<object>` es la que permite “*embeber*” o incluir en las páginas HTML cualquier tipo de contenido complejo:

Tabla 6.4. Etiqueta object

object	Objeto
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • data = "url" - Indica la URL de los datos que utiliza el objeto • classid, codebase, codetype - Información específica dependiente del tipo de objeto • height = "unidad_de_medida" - Indica la altura con la que se debe mostrar el objeto • width = "unidad_de_medida" - Indica la anchura con la que se debe mostrar el objeto
Tipo	Bloque y en línea
Descripción	Se emplea para embeber objetos en los documentos

El atributo `data` se emplea para establecer la URL del recurso que se va a incluir. El atributo `type` indica el tipo de contenido incluido. El propio estándar de HTML incluye ejemplos de uso de esta etiqueta. Incluir un vídeo en formato MPEG:

```
<object data="TheEarth.mpeg" type="application/mpeg" />
```

También se pueden incluir varias versiones alternativas de un mismo contenido. Así, si el navegador no es capaz de interpretar el formato por defecto, puede optar por cualquiera de los otros formatos alternativos:

```
<object title="La Tierra vista desde el espacio" classid="http://www.observer.mars/TheEarth.py">
```

```
  <!-- Formato alternativo en forma de vídeo -->
```

```
<object data="TheEarth.mpeg" type="application/mpeg">
```

```
  <!-- Otro formato alternativo mediante una imagen GIF -->
```

```
<object data="TheEarth.gif" type="image/gif">
```

```
  <!-- Si el navegador no soporta ningún formato, se muestra el siguiente texto -->
```

```

La <strong>Tierra</strong> vista desde el espacio.
</object>
</object>
</object>

```

A los objetos también se les puede pasar información adicional en forma de parámetros mediante la etiqueta <param>:

Tabla 6.5. Etiqueta param

param	Parámetros de un objeto
Atributos comunes	id
Atributos específicos	<ul style="list-style-type: none"> • name = "texto" - Indica el nombre del parámetro • value = "texto" - Indica el valor del parámetro
Tipo	Etiqueta vacía
Descripción	Se emplea para indicar el valor de los parámetros del objeto

Las etiquetas <param> siempre se incluyen en el interior de las etiquetas <object>:

```

<object data="..." type="...">
  <param name="valor1" value="40" />
  <param name="valor2" value="20" />
  <param name="valor3" value="texto de prueba" />
</object>

```

Uno de los principales inconvenientes de <object> es la forma de incluir vídeos en formato Flash en las páginas HTML. Si se utiliza el siguiente código:

```

<object data="nombre_video.swf" type="application/x-shockwave-flash"></object>

```

El elemento anterior es correcto desde el punto de vista técnico, pero provoca que algunos navegadores como Internet Explorer no visualicen el vídeo hasta que se ha descargado completamente. Si se trata de un vídeo largo, esta solución no es válida para el usuario.

Por este motivo, se utiliza una solución alternativa para incluir vídeos Flash en las páginas HTML: el uso de la etiqueta `<embed>`. Aunque esta solución funciona correctamente, no se trata de una solución válida desde el punto de vista del estándar de XHTML, por lo que las páginas que incluyan esta solución no pasarán correctamente el proceso de validación.

Tabla 6.6. Etiqueta embed

embed	Embeber objetos
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • <code>src = "url"</code> - Indica la URL del archivo u objeto que se incluye en la página • <code>type = "tipo_de_contenido"</code> - Indica el tipo de contenido del objeto (flash, quicktime, java, etc.) • <code>height = "unidad_de_medida"</code> - Indica la altura con la que se debe mostrar el objeto • <code>width = "unidad_de_medida"</code> - Indica la anchura con la que se debe mostrar el objeto
Tipo	Bloque
Descripción	Se emplea para embeber objetos en los documentos

Este es el motivo por el que los sitios web más populares de vídeos en formato Flash proporcionan un código similar al siguiente para incluir sus vídeos en las páginas HTML:

```
<object width="425" height="350">
  <param name="movie" value="http://www.youtube.com/v/MsH0rBWCYjs" ></param>
  <param name="wmode" value="transparent" ></param>
```

```
<embed src="http://www.youtube.com/v/MsH0rBWCYjs" type="application/  
x-shockwave-flash" wmode="transparent" width="425" height="350"></embed>  
</object>
```

Una vez más, se debe tener en cuenta que la solución anterior de utilizar la etiqueta `<embed>` es correcta desde el punto de vista del usuario (no tiene que esperar a que el vídeo se descargue completamente) pero no es una solución técnicamente válida, ya que la etiqueta `<embed>` no es parte del estándar XHTML.

Capítulo 7. Tablas

HTML definió desde sus primeras versiones un completo modelo para la creación de tablas en sus documentos. El modelo propuesto por HTML es sencillo, flexible y completo.

Sin embargo, las tablas HTML siempre han sido objeto de polémica al ser utilizadas para definir el layout de los documentos, en vez de emplearlas para la presentación de datos tabulares.

Utilizar tablas HTML para definir la estructura de un documento es algo obsoleto y poco recomendable, ya que se complica el código HTML y su mantenimiento es mucho más complejo en comparación con una solución CSS.

HTML utiliza los mismos conceptos de filas, columnas, cabeceras y títulos que los que se utilizan en cualquier otro entorno de publicación de documentos:

El diagrama muestra una tabla con las siguientes partes etiquetadas:

- Título:** Área de Diseño y Comunicación
- Columna:** Señala la estructura de columnas: Monográfico, Horas, Fechas, Horario, Dias.
- Cabecera de tabla:** Señala la fila superior que define las columnas.
- Cabecera de fila:** Señala la fila de un curso específico.
- Fila:** Señala una fila completa de un curso.

Área de Diseño y Comunicación					
Monográfico	Horas	Fechas	Horario	Dias	
Grafía e identidad corporativa Recogida de solicitudes: 28/08/06 - Publicación de lista de admitidos: 18/09/06	20	25/09/06	0:00	L-V	
Freehand MX Recogida de solicitudes: 04/09/06 - 22/09/06 Publicación de lista de admitidos: 25/09/06	60	02/10/06 - 24/10/06	16:00-20:00	L-V	
Photoshop CS 2 Recogida de solicitudes: 26/09/06 - 13/10/06 Publicación de lista de admitidos: 18/10/06	60	25/10/06 - 15/11/06	16:00-20:00	L-V	
Quarkxpress 6 Recogida de solicitudes: 16/10/06 - 08/11/06 Publicación de lista de admitidos: 09/11/06	40	16/11/06 - 29/11/06	16:00-20:00	L-V	
Indesign CS 2 Recogida de solicitudes: 30/10/06 - 22/11/06 Publicación de lista de admitidos: 23/11/06	36	30/11/06 - 15/12/06	16:00-20:00	L-V	
Edición digital con Adobe y creación de PDFs para imprenta Recogida de solicitudes: 13/11/06 - 22/11/06 Publicación de lista de admitidos: 11/12/06	20	18/12/06 - 22/12/06	16:00-20:00	L-V	

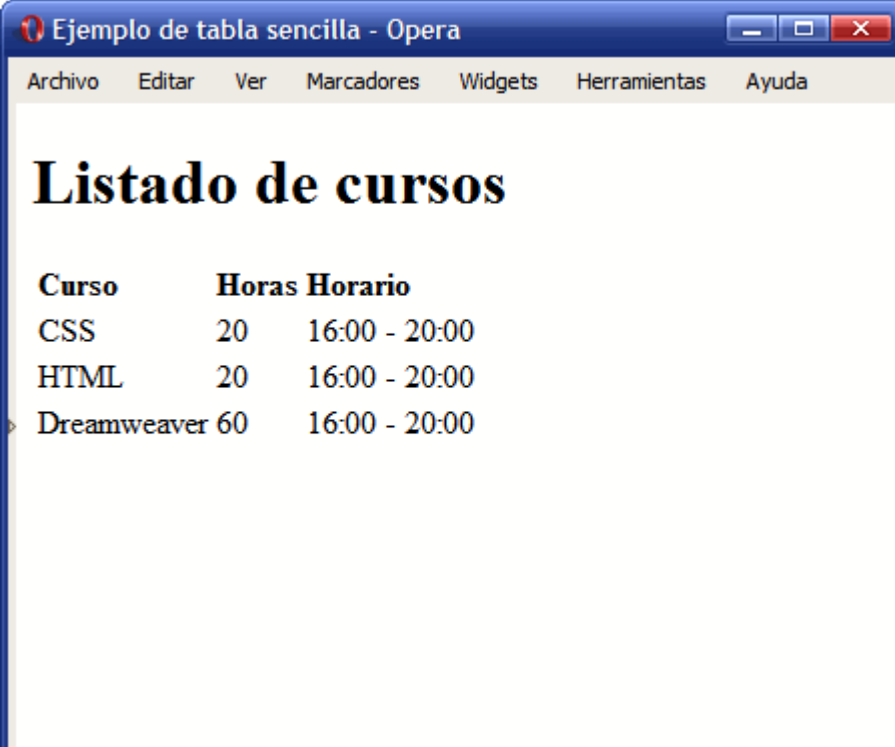
Figura 7.1. Partes que componen y definen una tabla compleja

HTML permite diseñar tablas muy complejas con agrupación de filas y de columnas, cabeceras y pies de tabla, subdivisiones en la tabla, múltiples cabeceras, etc.

7.1. Tablas básicas

Las tablas más sencillas de HTML se definen con 3 etiquetas: `<table>`, `<tr>` y `<td>`.

Ejemplo:



The screenshot shows a web browser window with the title "Ejemplo de tabla sencilla - Opera". The browser's menu bar includes "Archivo", "Editar", "Ver", "Marcadores", "Widgets", "Herramientas", and "Ayuda". The main content area displays a heading "Listado de cursos" followed by a table with three columns: "Curso", "Horas", and "Horario". The table contains three rows of data.

Curso	Horas	Horario
CSS	20	16:00 - 20:00
HTML	20	16:00 - 20:00
Dreamweaver	60	16:00 - 20:00

Figura 7.2. Ejemplo de tabla sencilla creada con las etiquetas `table`, `tr` y `td`

El código HTML necesario para crear esa tabla es el siguiente:

```
<html>
<head><title>Ejemplo de tabla sencilla</title></head>
<body>

<h1>Listado de cursos</h1>

<table>
```



```
<tr>
  <td><strong>Curso</strong></td>
  <td><strong>Horas</strong></td>
  <td><strong>Horario</strong></td>
</tr>
```

```
<tr>
  <td>CSS</td>
  <td>20</td>
  <td>16:00 - 20:00</td>
</tr>
```

```
<tr>
  <td>HTML</td>
  <td>20</td>
  <td>16:00 - 20:00</td>
</tr>
```

```
<tr>
  <td>Dreamweaver</td>
  <td>60</td>
  <td>16:00 - 20:00</td>
</tr>
</table>
```

```
</body>
</html>
```

La etiqueta `<table>` delimita la tabla de datos, la etiqueta `<tr>` define cada fila de datos y la etiqueta `<td>` define cada una de las columnas de las filas de la tabla.

Tabla 7.1. Etiqueta table

table	Tabla
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • <code>summary = "texto"</code> - Permite describir el contenido de la tabla de forma completa (para los buscadores y las personas discapacitadas)
Tipo	Bloque
Descripción	Se emplea para definir tablas de datos

Tabla 7.2. Etiqueta tr

tr	Fila de tabla
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	Bloque
Descripción	Se emplea para definir cada fila de las tablas de datos

Tabla 7.3. Etiqueta td

td	Celda de tabla
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • <code>abbr = "texto"</code> - Permite definir el contenido de la celda de forma abreviada (se emplea sobre todo con los navegadores de voz)

td	Celda de tabla
	<ul style="list-style-type: none"> • <code>headers = "lista_de_id"</code> - Indica las celdas que actúan como celdas de cabecera para esa celda (los títulos de las columnas y filas). Se indica como una lista de ID de celdas • <code>scope = "col, row, colgroup, rowgroup"</code> - Indica las celdas para las que esta celda será su cabecera. Ej: <code>scope="col"</code> indica que esta celda es la cabecera de todas las demás celdas que están en la misma columna • <code>colspan = "numero"</code> - Número de columnas que ocupa esta celda • <code>rowspan = "numero"</code> - Número de filas que ocupa esta celda
Tipo	Bloque
Descripción	Se emplea para definir cada celda que forman las filas de una tabla, es decir, las columnas de la tabla

HTML define la etiqueta `<caption>` para establecer la leyenda o título de una tabla. La etiqueta debe colocarse inmediatamente después de la etiqueta `<table>` y cada tabla solo puede tener 1 `<caption>`.

Tabla 7.4. Etiqueta `caption`

caption	Leyenda o título de tabla
Atributos comunes	básicos, <code>i18n</code> y eventos
Atributos específicos	-
Tipo	En línea
Descripción	Se emplea para definir la leyenda o título de una tabla

Habitualmente, algunas de las celdas de la tabla se utilizan como cabecera de las demás celdas de la fila o de la columna. En este caso, HTML define la etiqueta <th> que tiene las mismas propiedades que la etiqueta <td>.

Tabla 7.5. Etiqueta th

th	Celda cabecera de tabla
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • abbr = "texto" - Permite definir el contenido de la celda de forma abreviada (se emplea sobre todo con los navegadores de voz) • headers = "lista_de_id" - Indica las celdas que actúan como celdas de cabecera para esa celda (los títulos de las columnas y filas). Se indica como una lista de ID de celdas • scope = "col, row, colgroup, rowgroup" - Indica las celdas para las que esta celda será su cabecera. Ej: scope="col" indica que esta celda es la cabecera de todas las demás celdas que están en la misma columna • colspan = "numero" - Número de columnas que ocupa esta celda • rowspan = "numero" - Número de filas que ocupa esta celda
Tipo	Bloque
Descripción	Se emplea para definir las celdas que son cabecera de una fila o de una columna de la tabla

Las tablas complejas suelen disponer de una estructura irregular que en algunas filas fusiona varias columnas y en algunas columnas fusiona varias filas. De esta forma, los atributos más utilizados en las tablas avanzadas son rowspan y colspan.

Ejemplo sencillo de fusión de columnas en HTML

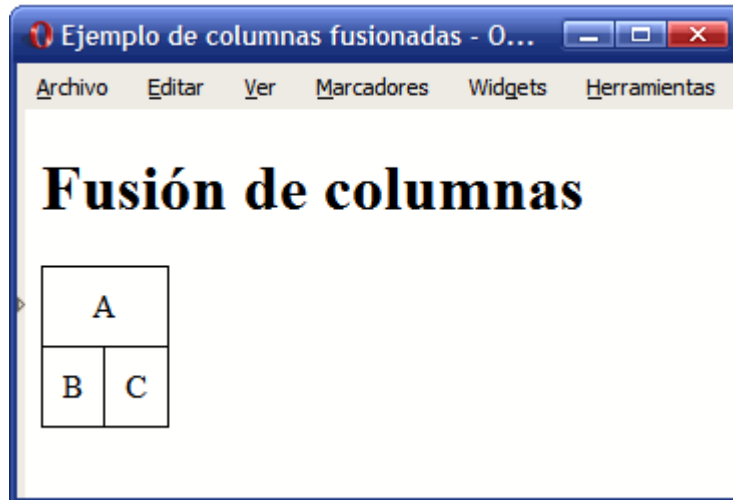


Figura 7.3. Ejemplo sencillo de fusión de columnas

El código HTML necesario para combinar las 2 columnas superiores se muestra a continuación:

```
<table>
<tr>
  <td colspan="2">A</td>
</tr>

<tr>
  <td>B</td>
  <td>C</td>
</tr>
</table>
```

Ejemplo sencillo de fusión de filas en HTML



Figura 7.4. Ejemplo sencillo de fusión de filas

El código HTML correspondiente se muestra a continuación:

```
<table>
<tr>
  <td>A</td>
  <td rowspan="2">B</td>
</tr>

<tr>
  <td>C</td>
</tr>
</table>
```

Ejemplo de fusión de columnas en HTML

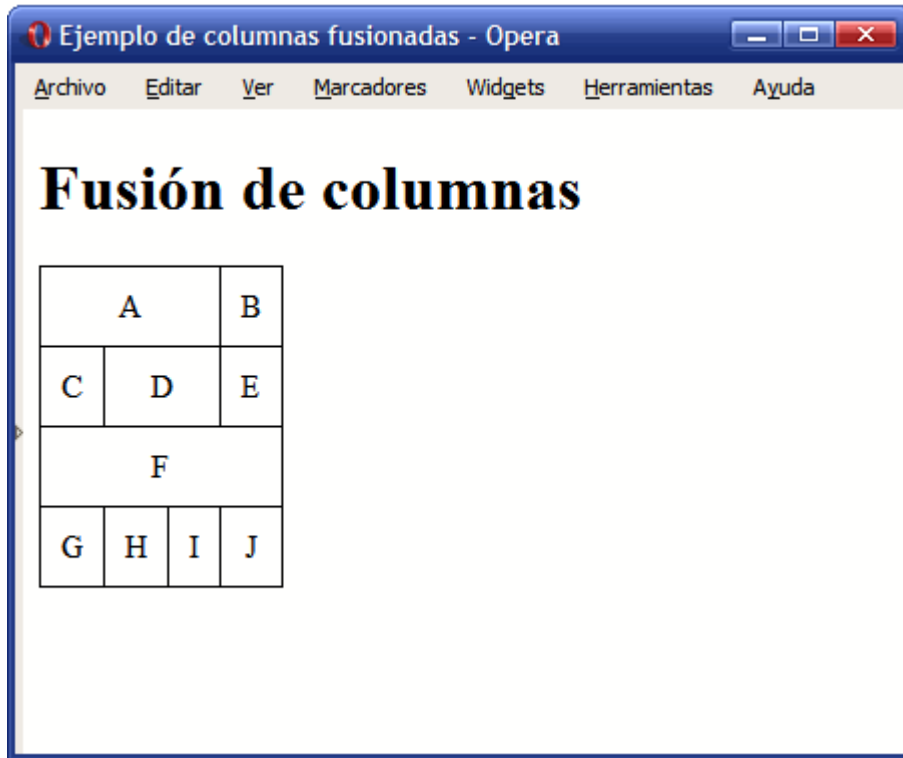


Figura 7.5. Ejemplo complejo de fusión de columnas

El código HTML necesario para fusionar las columnas de la tabla anterior se muestra a continuación:

```
<html>
<head><title>Ejemplo de columnas fusionadas</title></head>
<body>

<h1>Fusión de columnas</h1>

<table>
<tr>
  <td colspan="3">A</td>
  <td>B</td>
</tr>

<tr>
  <td>C</td>
```

```
<td colspan="2">D</td>
<td>E</td>
</tr>

<tr>
  <td colspan="4">F</td>
</tr>

<tr>
  <td>G</td>
  <td>H</td>
  <td>I</td>
  <td>J</td>
</tr>
</table>

</body>
</html>
```

Ejemplo de fusión de filas en HTML

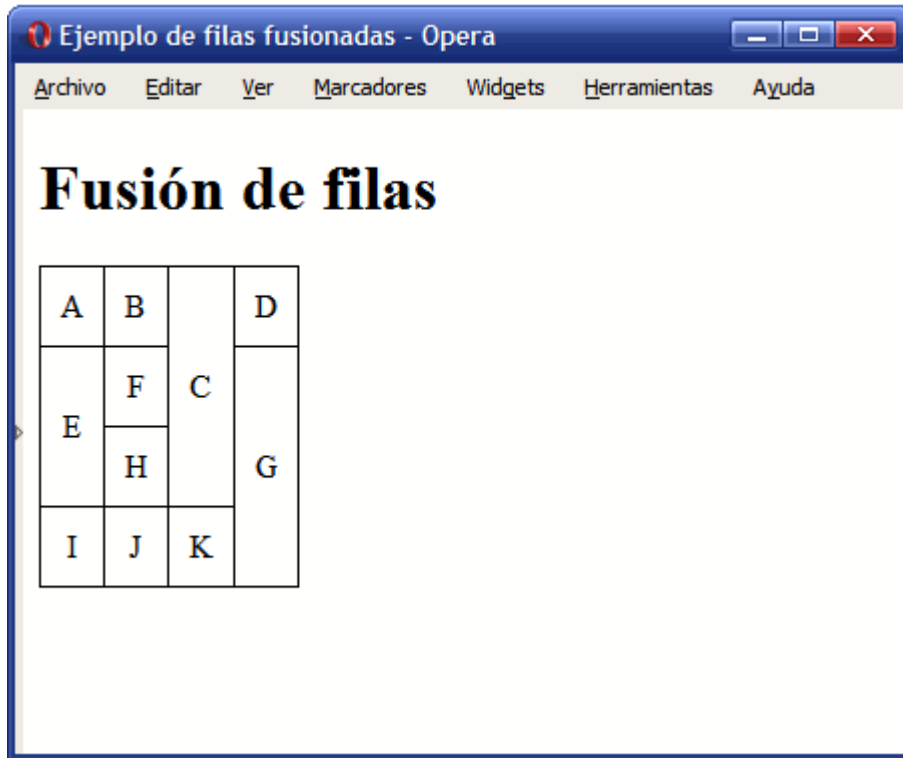


Figura 7.6. Ejemplo complejo de fusión de filas

El código HTML necesario para fusionar las filas de la tabla anterior se muestra a continuación:

```
<html>
<head><title>Ejemplo de filas fusionadas</title></head>
<body>

<h1>Fusión de filas</h1>

<table>
  <tr>
    <td>A</td>
    <td>B</td>
    <td rowspan="3">C</td>
    <td>D</td>
  </tr>
  <tr>
```

```
<td rowspan="2">E</td>
<td>F</td>
<td rowspan="3">G</td>
</tr>
<tr>
<td>H</td>
</tr>
<tr>
<td>I</td>
<td>J</td>
<td>K</td>
</tr>
</table>

</body>
</html>
```

7.2. Tablas avanzadas

Las tablas más complejas están formadas por una sección de cabecera, una sección de pie y varias secciones de datos. Además, varias columnas se pueden agrupar de forma lógica para poder aplicar estilos similares a un determinado grupo de columnas. Un ejemplo clásico de esta clase de tablas complejas es el de los balances de las empresas:

TELEFÓNICA GROUP CONSOLIDATED BALANCE SHEET <i>Unaudited figures (Euros in millions)</i>	2006			
	March	June	September	December
Non-current assets	49.725,7	57.457,3	57.982,2	59.545,0
Intangible assets	5.914,9	6.849,4	6.872,5	7.877,1
Goodwill	6.656,4	8.961,4	9.394,4	8.910,2
Property, plant and equipment and Investment property	23.416,2	27.787,2	27.961,5	28.027,4
Long-term financial assets and other non-current assets	4.959,4	5.353,0	5.546,5	6.345,6
Deferred tax assets	8.778,8	8.506,3	8.207,3	8.384,7
Current assets	11.362,3	12.625,5	13.905,5	13.628,8
Inventories	718,1	870,4	903,8	919,5
Trade and other receivables	6.311,5	7.390,9	7.459,1	7.515,7
Current tax receivable	1.208,9	1.358,0	1.561,5	1.448,3
Short-term financial investments	2.063,5	1.413,6	1.415,0	1.517,8
Cash and cash equivalents	1.048,8	1.579,1	2.548,9	2.213,2
Non-current assets classified as held for sale	11,4	13,6	17,1	14,3
Total Assets = Total Equity and Liabilities	61.088,0	70.082,8	71.887,6	73.173,8
Equity	13.000,2	13.961,7	14.924,8	16.158,4
Equity attributable to equity holders of the parent	11.313,5	10.637,8	12.265,0	12.733,3
Minority interest	1.686,7	3.323,9	2.659,8	3.425,1
Non-current liabilities	28.800,0	31.225,4	34.750,8	35.126,5
Long-term financial debt	18.113,2	19.667,5	23.884,2	25.167,6
Deferred tax liabilities	1.871,5	2.468,6	1.999,9	2.477,4
Long-term provisions	7.687,9	7.834,3	7.632,4	6.353,2
Other long-term liabilities	1.127,5	1.255,0	1.234,3	1.128,2
Current liabilities	19.287,7	24.895,7	22.212,1	21.888,9
Short-term financial debt	9.455,1	11.689,5	9.094,8	9.235,9
Trade and other payables	5.488,4	6.426,3	6.061,9	6.932,9
Current tax payable	1.997,6	2.089,1	2.345,8	2.191,6
Short-term provisions and other liabilities	2.341,2	4.690,7	4.709,6	3.528,5
Liabilities associated with non-current assets classified as held for sale	5,4	0,0	0,0	0,0
Financial Data				
Net Financial Debt (1)	23.948,1	27.990,4	28.676,1	30.067,0

(1) Net Financial Debt = Long term financial debt + Other long term liabilities + Short term financial debt - Short term financial investments - current assets.

Figura 7.7. Ejemplo de tabla compleja correspondiente al balance de una empresa

Tabla 7.6. Etiqueta thead, tbody, tfoot

thead	Cabecera de tabla
tbody	Sección de una tabla
tfoot	Pie de tabla
Atributos comunes	básicos, i18n y eventos

thead	Cabecera de tabla
tbody	Sección de una tabla
tfoot	Pie de tabla
Atributos específicos	-
Tipo	Bloque
Descripción	Se emplean para agrupar varias filas en una cabecera (thead) un pie (tfoot) o una sección (tbody) de una tabla

Cada tabla puede tener una cabecera, un pie y una o más secciones. Las etiquetas <thead> y <tfoot> deben colocarse inmediatamente antes que cualquier etiqueta <tbody>.

Ejemplo de tabla avanzada

Análisis de ventas

Análisis de ventas anuales

AÑO	Expansión de ventas			
	Producto A	Producto B	Producto C	Producto D
N-3	-	-	-	-
N-2	3	5	8	4
N-1	4	4	7	3
N	5	7	6	2
AÑO	Expansión de ventas			
	Producto A	Producto B	Producto C	Producto D

Figura 7.8. Ejemplo de tabla avanzada con cabecera, pie y secciones

```

<html>
<head><title>Ejemplo de tabla avanzada</title></head>
<body>

<h3>Análisis de ventas</h3>

<table summary="Análisis de ventas anuales">
<caption>Análisis de ventas anuales</caption>
<thead>
<tr>
<th rowspan="2" scope="col">AÑO</th>

```

```

<th colspan="4" scope="col">Expansión de ventas</th>
</tr>
<tr>
  <th scope="col">Producto A</th>
  <th scope="col">Producto B</th>
  <th scope="col">Producto C</th>
  <th scope="col">Producto D</th>
</tr>
</thead>

<tfoot>
<tr>
  <th rowspan="2" scope="col">AÑO</th>
  <th scope="col">Producto A</th>
  <th scope="col">Producto B</th>
  <th scope="col">Producto C</th>
  <th scope="col">Producto D</th>
</tr>

<tr>
<th colspan="4" scope="col">Expansión de ventas</th>
</tr>
</tfoot>

<tbody>
<tr>
  <th scope="row">N-3</th><td>-</td><td>-</td><td>-</td><td>-</td>
</tr>
<tr>
  <th scope="row">N-2</th><td>3</td><td>5</td><td>8</td><td>4</td>
</tr>
<tr>
  <th scope="row">N-1</th><td>4</td><td>4</td><td>7</td><td>3</td>

```

```
</tr>
<tr>
  <th scope="row">N</th><td>5</td><td>7</td><td>6</td><td>2</td>
</tr>
</tbody>

</table>

</body>

</html>
```

HTML también permite agrupar varias columnas o aplicar distintas propiedades a las columnas de forma muy sencilla. Las 2 etiquetas que se emplean son `<col>` y `<colgroup>`.

Ejemplo de definición de anchuras con la etiqueta `<col>`



The screenshot shows a browser window with the title "Ejemplo de tabla avanzada - Opera". The menu bar includes "Archivo", "Editar", "Ver", "Marcadores", "Widgets", "Herramientas", and "Ayuda". The main content area displays the heading "Análisis de ventas" followed by a sub-heading "Análisis de ventas anuales". Below this is a table with 5 columns and 5 rows. The columns are labeled "AÑO", "Producto A", "Producto B", "Producto C", and "Producto D". The rows represent years: "N-3", "N-2", "N-1", and "N". The data values are as follows:

AÑO	Producto A	Producto B	Producto C	Producto D
N-3	-	-	-	-
N-2	3	5	8	4
N-1	4	4	7	3
N	5	7	6	2

Figura 7.9. Ejemplo de tabla avanzada que usa la etiqueta `col`

```

<table summary="Análisis de ventas anuales">
<caption>Análisis de ventas anuales</caption>
<col style="width:10%;" />
<col style="width:30%;" />
<thead>
<tr>
  <th scope="col">AÑO</th>
  <th scope="col">Producto A</th>
  <th scope="col">Producto B</th>
  <th scope="col">Producto C</th>
  <th scope="col">Producto D</th>
</tr>
</thead>

<tbody>
<tr>
  <th scope="row">N-3</th><td>-</td><td>-</td><td>-</td><td>-</td>
</tr>
<tr>
  <th scope="row">N-2</th><td>3</td><td>5</td><td>8</td><td>4</td>
</tr>
<tr>
  <th scope="row">N-1</th><td>4</td><td>4</td><td>7</td><td>3</td>
</tr>
<tr>
  <th scope="row">N</th><td>5</td><td>7</td><td>6</td><td>2</td>
</tr>
</tbody>

</table>

```

Ejemplo de uso de la etiqueta <colgroup>

Análisis de ventas

Análisis de ventas anuales

AÑO	Producto A	Producto B	Producto C	Producto D
N-3	-	-	-	-
N-2	3	5	8	4
N-1	4	4	7	3
N	5	7	6	2

Figura 7.10. Ejemplo de tabla avanzada que usa la etiqueta colgroup

```

<table summary="Análisis de ventas anuales">
<caption>Análisis de ventas anuales</caption>
<colgroup span="1" style="color:red;" />
<colgroup span="3" style="color:blue;" />
<thead>
<tr>
  <th scope="col">AÑO</th>
  <th scope="col">Producto A</th>
  <th scope="col">Producto B</th>
  <th scope="col">Producto C</th>
  <th scope="col">Producto D</th>
</tr>
</thead>

<tbody>
<tr>

```

```
<th scope="row">N-3</th><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</tr>
<tr>
  <th scope="row">N-2</th><td>3</td><td>5</td><td>8</td><td>4</td>
</tr>
</tr>
<tr>
  <th scope="row">N-1</th><td>4</td><td>4</td><td>7</td><td>3</td>
</tr>
</tr>
<tr>
  <th scope="row">N</th><td>5</td><td>7</td><td>6</td><td>2</td>
</tr>
</tbody>

</table>
```

El principal inconveniente de `<col>` y `<colgroup>` es que la mayoría de sus funcionalidades solamente están disponibles en el navegador Internet Explorer.

Capítulo 8. Formularios

El aspecto más diferenciado del lenguaje HTML respecto de otros lenguajes de marcado es la posibilidad de definir elementos interactivos como formularios que permiten crear aplicaciones web con la que los usuarios pueden interactuar.

Aunque presenta varias carencias importantes (sliders, spinners, combobox, barras de progreso, etc.) el lenguaje HTML define suficientes tipos de controles como para crear formularios completos:

Bug Tracker
You find em, we'll squich em.

Where did you find it? *

Describe in detail what happened *

Operating System

Browser

Upload a Screenshot

Assign To:

Severity

Fixed?

Yes

No

Figura 8.1. Ejemplo de formulario HTML complejo

8.1. Formularios básicos

El formulario más sencillo se puede crear con solo 2 etiquetas: `<form>` y `<input>`:

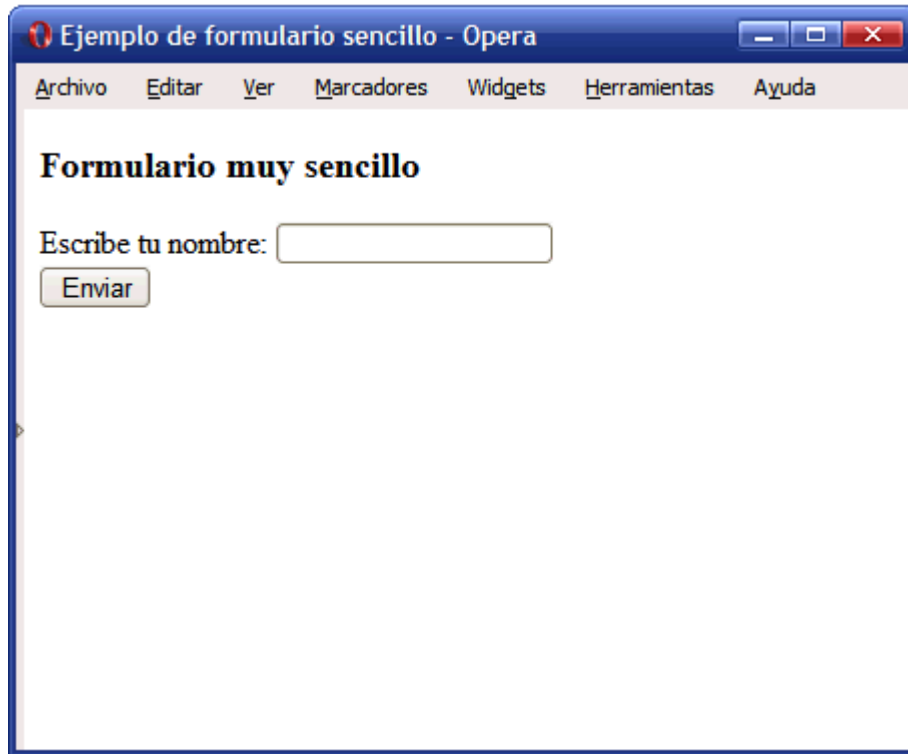


Figura 8.2. Formulario sencillo definido con las etiquetas form e input

El código HTML necesario para definir el formulario anterior se muestra a continuación:

```
<html>
<head><title>Ejemplo de formulario sencillo</title></head>
<body>

<h3>Formulario muy sencillo</h3>

<form action="http://www.misitio.com/maneja_formulario.php" method="post">
Escribe tu nombre: <input type="text" name="nombre" value="" />
<br/>
<input type="submit" value="Enviar" />
</form>

</body>
</html>
```

Tabla 8.1. Etiqueta form

form	Formulario
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • <code>action = "url"</code> - Indica la URL que se encarga de procesar la respuesta al formulario • <code>method = "POST o GET"</code> - Método HTTP empleado al enviar el formulario • <code>enctype = "application/x-www-form-urlencoded o multipart/form-data"</code> - Tipo de codificación empleada al enviar el formulario al servidor (solo para los formularios que permiten incluir archivos) • <code>accept = "tipo_de_contenido"</code> - Lista separada por comas de todos los tipos de archivos aceptados por el servidor (solo para los formularios que permiten incluir archivos) • Otros: <code>accept-charset, onsubmit, onreset</code> -
Tipo	Bloque
Descripción	Se emplea para insertar un formulario en la página

La etiqueta `<form>` permite definir un “contenedor” de los controles que forman el formulario. Los atributos más importantes (y que se emplean en todos los formularios) son `action` y `method`.

Tabla 8.2. Etiqueta `input`

input	Control de un formulario
Atributos comunes	básicos, i18n, eventos y foco

input	Control de un formulario
Atributos específicos	<ul style="list-style-type: none"> • <code>type = "text password checkbox radio submit reset file hidden image button"</code> - Indica el tipo de control que se incluye en el formulario • <code>name = "texto"</code> - Asigna un nombre al control (es imprescindible para poder procesar el formulario) • <code>value = "texto"</code> - Valor inicial del control • <code>size = "unidad_de_medida"</code> - Tamaño inicial del control (para los campos de texto y de password se refiere al número de caracteres, en el resto de controles se refiere a su tamaño en píxel) • <code>maxlength = "numero"</code> - Máximo número de caracteres para los controles de texto y de password • <code>checked = "checked"</code> - Permite seleccionar una opción de los controles checkbox y radiobutton • <code>disabled = "disabled"</code> - El control no se tiene en cuenta en el formulario (ni siquiera se envía al servidor). Los navegadores muestran los controles visualmente deshabilitados (de color gris) • <code>readonly = "readonly"</code> - El contenido del control no se puede modificar • <code>src = "url"</code> - Indica la URL de la imagen que se emplea como botón de formulario • <code>alt = "texto"</code> - Texto alternativo del control
Tipo	En línea y etiqueta vacía
Descripción	Se emplean para insertar un control en un formulario

La etiqueta `<input>` permite definir 10 controles diferentes de formulario, por lo que es la más utilizada al crear formularios:

Texto:

Nombre

Figura 8.3. Ejemplo de etiqueta input (type=text)

```
Nombre <br/>
<input type="text" name="nombre" value="" />
```

Contraseña

Contraseña

Figura 8.4. Ejemplo de etiqueta input (type=password)

```
Contraseña <br/>
<input type="password" name="contrasena" value="" />
```

Checkbox

Puestos de trabajo buscados
 Dirección
 Técnico
 Empleado

Figura 8.5. Ejemplo de etiqueta input (type=checkbox)

```
Puestos de trabajo buscados <br/>
<input name="puesto_directivo" type="checkbox" value="dirección"/> Dirección <br/>
```



```
<input name="puesto_tecnico" type="checkbox" value="tecnico"/> Técnico <br/>  
<input name="puesto_empleado" type="checkbox" value="empleado"/> Empleado <br/>
```

Radiobutton

Sexo
 Hombre
 Mujer

Figura 8.6. Ejemplo de etiqueta input (type=radio)

```
Sexo <br/>  
<input type="radio" name="sexo" value="hombre" checked="checked" /> Hombre <br/>  
<input type="radio" name="sexo" value="mujer" /> Mujer <br/>
```

Botón de envío de formulario

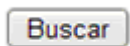


Figura 8.7. Ejemplo de etiqueta input (type=submit)

```
<input type="submit" name="buscar" value="Buscar" />
```

Botón de reseteo del formulario



Figura 8.8. Ejemplo de etiqueta input (type=reset)

```
<input type="reset" name="limpiar" value="Borrar datos del formulario" />
```

Adjuntar ficheros al formulario

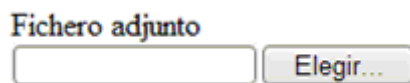


Figura 8.9. Ejemplo de etiqueta input (type=file)

```
Fichero adjunto <br/>
<input type="file" name="adjunto" />
```

Campo oculto

*Los campos ocultos
no se ven en pantalla

Figura 8.10. Ejemplo de etiqueta input (type=hidden)

```
<input type="hidden" name="url_previa" value="/articulo/primer.html" />
```

Botón de imagen



Figura 8.11. Ejemplo de etiqueta input (type=image)

```
<input type="image" name="enviar" src="accept.png" />
```

Botón

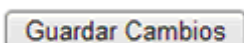
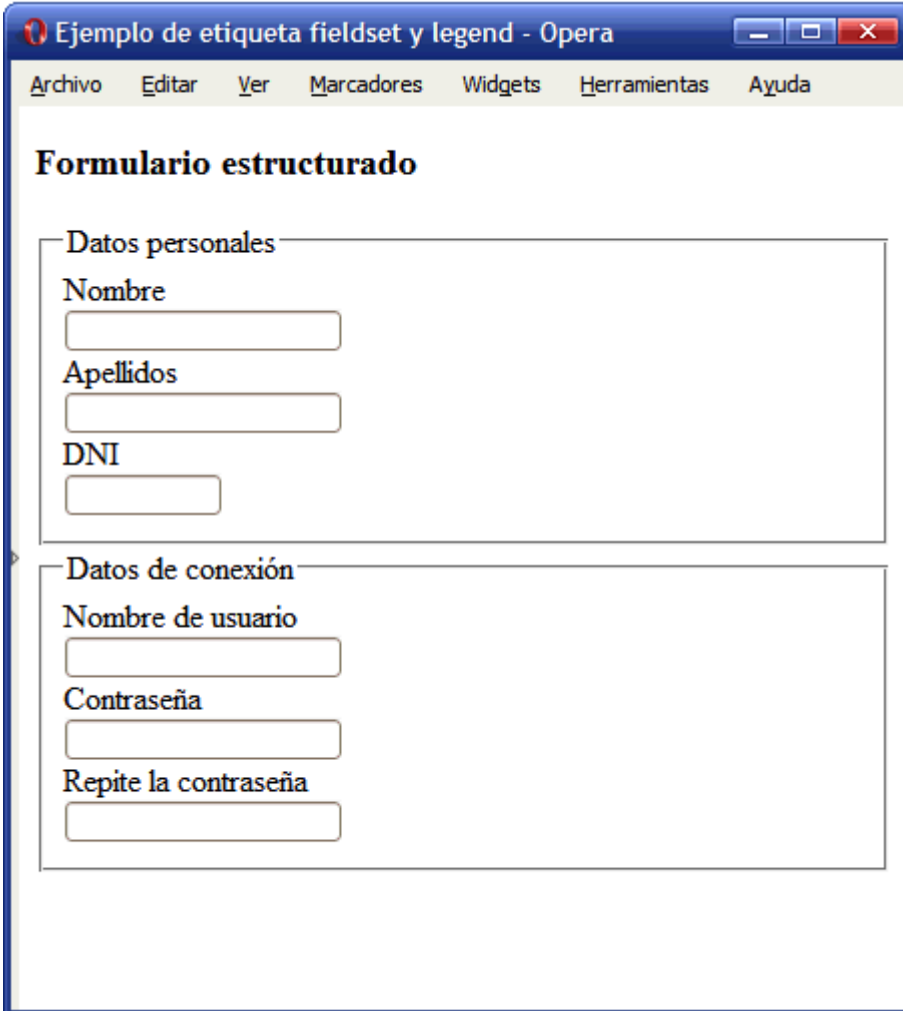


Figura 8.12. Ejemplo de etiqueta input (type=button)

```
<input type="button" name="guardar" value="Guardar Cambios" />
```

8.2. Formularios avanzados

Los formularios creados hasta ahora son sencillos y muy simples. HTML permite incluir algunos elementos adicionales para mejorar los formularios: agrupación de campos y etiquetas para los campos.



The screenshot shows a browser window titled "Ejemplo de etiqueta fieldset y legend - Opera". The browser's menu bar includes "Archivo", "Editar", "Ver", "Marcadores", "Widgets", "Herramientas", and "Ayuda". The main content area is titled "Formulario estructurado" and contains two fieldsets. The first fieldset, titled "Datos personales", contains three input fields labeled "Nombre", "Apellidos", and "DNI". The second fieldset, titled "Datos de conexión", contains three input fields labeled "Nombre de usuario", "Contraseña", and "Repite la contraseña".

Figura 8.13. Ejemplo de uso de las etiquetas fieldset y legend

Las etiquetas `<fieldset>` y `<legend>` permiten estructurar los formularios mediante la agrupación lógica de los campos.

Tabla 8.3. Etiqueta fieldset

fieldset	Agrupación de campos
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	Bloque
Descripción	Se emplea para agrupar de forma lógica varios campos de un formulario

Tabla 8.4. Etiqueta legend

legend	Título o leyenda de un fieldset
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • <code>accesskey = "letra"</code> - Permite definir una tecla de acceso rápido a un elemento HTML. Si el usuario pulsa la tecla ALT y esa tecla, se selecciona el elemento
Tipo	En línea
Descripción	Se emplea para definir el título o leyenda de un conjunto de campos de formulario agrupados con la etiqueta fieldset

El código fuente del anterior ejemplo se muestra a continuación:

```
<form action="maneja_formulario.php" method="post">
```

```
<fieldset>
```

```
<legend>Datos personales</legend>
```

```
Nombre <br/>
```

```

<input type="text" name="nombre" value="" />
<br/>
Apellidos <br/>
<input type="text" name="apellidos" value="" />
<br/>
DNI <br/>
<input type="text" name="dni" value="" size="10" maxlength="9" />
</fieldset>

```

```

<fieldset>
<legend>Datos de conexión</legend>
Nombre de usuario<br/>
<input type="text" name="nombre" value="" maxlength="10" />

<br/>
Contraseña<br/>
<input type="password" name="password" value="" maxlength="10" />
<br/>
Repita la contraseña<br/>
<input type="password" name="password2" value="" maxlength="10" />
</fieldset>

```

Los controles básicos (salvo los botones) presentan una carencia muy importante: no disponen de la opción de incluir un pequeño título o texto de ayuda junto al control.

A pesar de que se puede incluir cualquier texto al lado de los controles mediante párrafos y otros mecanismos, HTML incluye una etiqueta específica para la definición de los títulos de cada campo: la etiqueta <label>.

Tabla 8.5. Etiqueta label

label	Título o leyenda de un campo de formulario
Atributos comunes	básicos, i18n y eventos

label	Título o leyenda de un campo de formulario
Atributos específicos	<ul style="list-style-type: none"> • for = "id_de_elemento" - Indica el ID del campo del formulario para el que este "label" es su título • Otros: accesskey, onfocus y onblur -
Tipo	En línea
Descripción	Se emplea para definir el título o leyenda de los campos definidos en un formulario

Campos de formulario sin <label>

Nombre


```
<input type="text" name="nombre" value="" />
```

Apellidos


```
<input type="text" name="apellidos" value="" />
```

DNI


```
<input type="text" name="dni" value="" size="10" maxlength="9" />
```

Campos de formulario con <label>

```
<label for="nombre">Nombre</label> <br/>
```

```
<input type="text" id="nombre" name="nombre" value="" />
```

```
<label for="apellidos">Apellidos</label> <br/>
```

```
<input type="text" id="apellidos" name="apellidos" value="" />
```

```
<label for="dni">DNI</label> <br/>
```

```
<input type="text" id="dni" name="dni" value="" size="10" maxlength="9" />
```

La principal ventaja es que el código HTML está mejor estructurado, se mejora la accesibilidad y la semántica del documento. Además, al pinchar sobre el texto del `<label>`, se posiciona el puntero para poder introducir información. Esto último es especialmente útil para los campos `radiobutton` y `checkbox`.

8.3. Formularios completos

HTML define otros controles avanzados que no se pueden incluir mediante la etiqueta `<input>`. Se trata de las listas desplegables (`<select>`) y las áreas de texto (`<textarea>`).

Las áreas de texto son útiles cuando se debe introducir una gran cantidad de texto, ya que es mucho más cómodo de introducir que en un campo de texto normal:

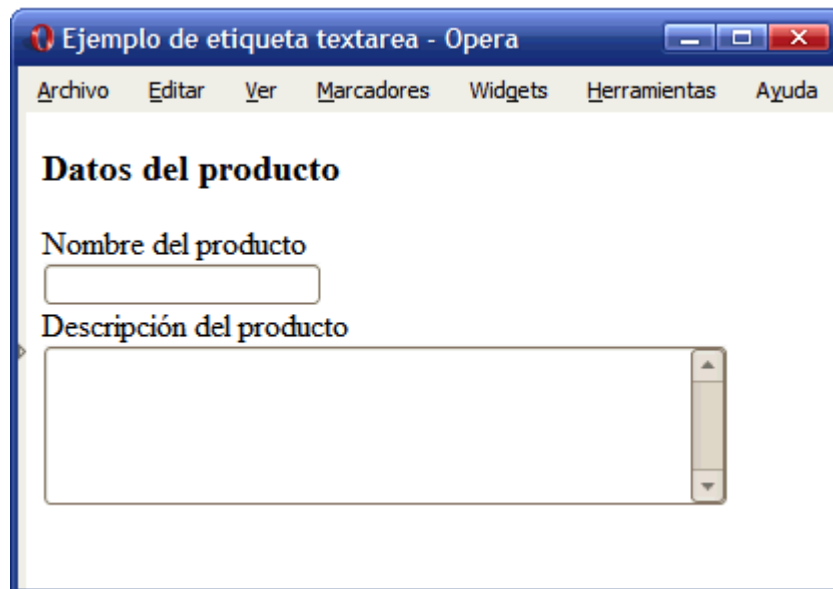


Figura 8.14. Ejemplo de uso de la etiqueta `textarea`

El código HTML del ejemplo anterior se muestra a continuación:

```
<form action="insertar_producto.php" method="post">  
  
<label for="nombre">Nombre del producto</label> <br/>  
<input type="text" id="nombre" name="nombre" value="" />  
  
<label for="descripcion">Descripción del producto</label> <br/>
```

```
<textarea id="descripcion" name="descripcion" cols="40" rows="5"></textarea>

</form>
```

Tabla 8.6. Etiqueta textarea

textarea	Área de texto
Atributos comunes	básicos, i18n, eventos y foco
Atributos específicos	<ul style="list-style-type: none"> • rows = "numero" - Número de filas de texto que mostrará el textarea • cols = "numero" - Número de caracteres que se muestran en cada fila del textarea • Otros: name, disabled, readonly, onselect, onchange, onfocus, onblur -
Tipo	En línea
Descripción	Se emplea para incluir un área de texto en un formulario

El principal inconveniente de los elementos <textarea> es que el lenguaje HTML no permite indicar un tamaño máximo al texto introducido, y habitualmente se debe recurrir a soluciones JavaScript o soluciones remotas en el servidor.

El otro control de formulario que define HTML son las listas desplegables:



Figura 8.15. Ejemplo de uso de la etiqueta select

HTML permite definir listas desplegables simples en las que solamente se puede seleccionar una opción y listas desplegadas en las que se puede seleccionar una o más opciones.

El código HTML del ejemplo anterior se muestra a continuación:

```
<label for="provincia">Provincia</label> <br/>
<select id="provincia" name="provincia">
  <option value="" selected="selected">- selecciona -</option>
  <option value="01">Alava</option>
  <option value="20">Guipuzcoa</option>
  <option value="48">Vizcaya</option>
</select>

<br/><br/>

<label for="provincia2">Provincia</label> <br/>
```

```

<select id="provincia2" name="provincia2" size="4">
  <option value="01" selected="selected">Alava</option>
  <option value="20">Guipuzcoa</option>
  <option value="48">Vizcaya</option>
</select>

<br/><br/>

<label for="provincia3">Provincia</label> <br/>
<select id="provincia3" name="provincia3" size="4" multiple="multiple">
  <option value="01" selected="selected">Alava</option>
  <option value="20">Guipuzcoa</option>
  <option value="48">Vizcaya</option>
</select>

```

Tabla 8.7. Etiqueta select

select	Lista desplegable
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • <code>size = "numero"</code> - Número de filas que se muestran de la lista (por defecto solo se muestra 1) • <code>multiple = "multiple"</code> - Si se incluye, se permite seleccionar más de 1 elemento • Otros: <code>name</code>, <code>disabled</code>, <code>onchange</code>, <code>onfocus</code>, <code>onblur</code> -
Tipo	En línea
Descripción	Se emplea para incluir una lista desplegable en un formulario

Tabla 8.8. Etiqueta option

option	Elemento de una lista desplegable
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<ul style="list-style-type: none"> • selected = "selected" - Indica si el elemento aparece seleccionado por defecto al cargarse la página • value = "texto" - El valor que se envía al servidor cuando el usuario elige esa opción • Otros: label, disabled -
Tipo	-
Descripción	Se emplea para definir cada elemento de una lista desplegable

Las listas desplegables permiten agrupar sus opciones de forma que sea más fácil al usuario seleccionar una opción entre varias cuando la lista es muy larga:

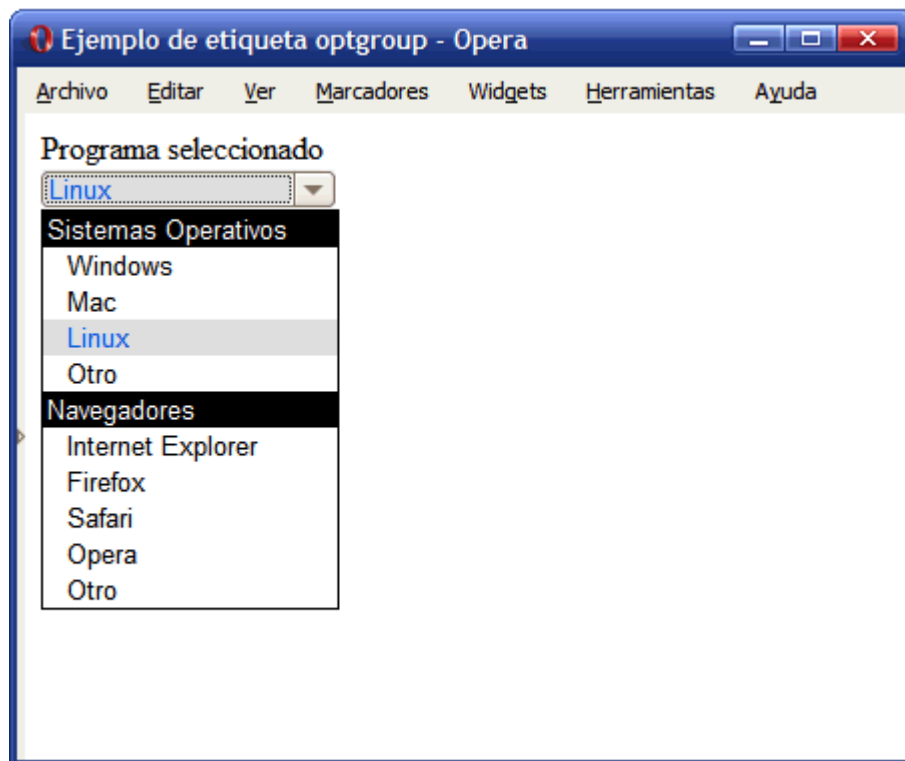


Figura 8.16. Ejemplo de uso de la etiqueta optgroup

La etiqueta `<optgroup>` permite agrupar los elementos de las listas desplegables de forma que se simplifique la selección del elemento deseado. El código HTML del ejemplo anterior se muestra a continuación:

```
<form id="formulario" method="post" action="" enctype="multipart/form-data">

<label for="programa">Programa seleccionado</label> <br/>
<select id="programa" name="programa">
  <optgroup label="Sistemas Operativos">
    <option value="Windows" selected="selected">Windows</option>
    <option value="Mac">Mac</option>
    <option value="Linux">Linux</option>
    <option value="Other">Otro</option>
  </optgroup>
  <optgroup label="Navegadores">
    <option value="Internet Explorer" selected="selected">Internet
Explorer</option>
    <option value="Firefox">Firefox</option>
    <option value="Safari">Safari</option>
    <option value="Opera">Opera</option>
    <option value="Other">Otro</option>
  </optgroup>
</select>

</form>
```

Tabla 8.9. Etiqueta `optgroup`

optgroup	Agrupación de elementos de una lista desplegable
Atributos comunes	básicos, i18n y eventos

optgroup	Agrupación de elementos de una lista desplegable
Atributos específicos	<ul style="list-style-type: none">• label = "texto" - Texto que se muestra como título de la agrupación de opciones• Otros: disabled, selected -
Tipo	-
Descripción	Se emplea para definir una agrupación lógica de opciones de una lista desplegable

Capítulo 9. Estructura y layout

Hasta ahora, se ha visto como estructurar y *marcar* mediante HTML diferentes elementos individuales como tablas, listas, enlaces, párrafos de texto, etc. Las páginas web normales, sin embargo, suelen incluir en una misma página decenas de elementos de este tipo.

Por este motivo, es necesaria una estructuración adicional de los contenidos de la página. La estrategia que se sigue es la de dividir la página en zonas en función de su finalidad: la zona de la cabecera de la página, la zona de contenidos, una zona lateral para el menú y otras secciones menores, la zona del pie de página, etc.

La siguiente imagen muestra las zonas definidas en una página web real:

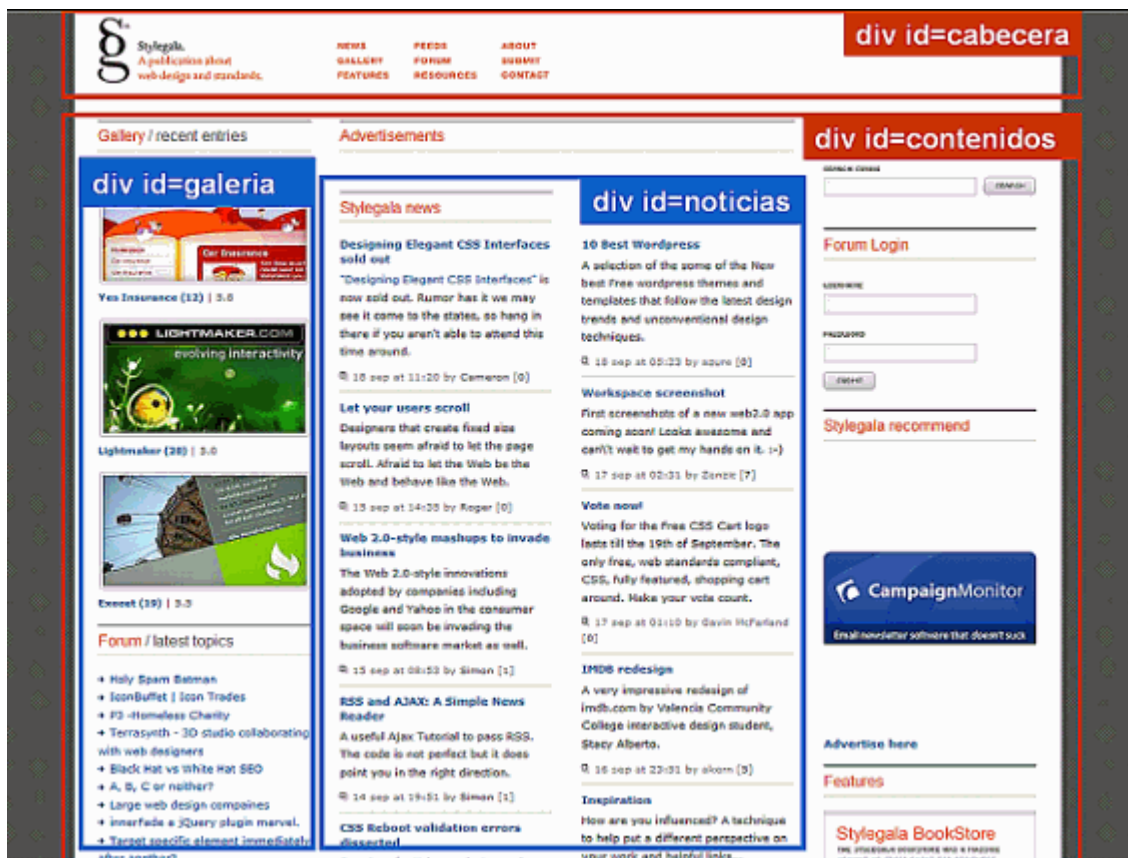


Figura 9.1. Ejemplo de página compleja estructurada con etiquetas div

Para definir las zonas de una página, se utiliza la etiqueta `<div>`:

Tabla 9.1. Etiqueta div

div	Divisiones
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	Bloque
Descripción	Agrupar elementos de bloque y mejorar la semántica del documento

El nombre `div` viene de *división*, ya que esta etiqueta define zonas o divisiones dentro de una página HTML. En cualquier caso, casi todos los diseñadores utilizan la palabra “capas” en vez de “divisiones”. Aunque se trata de un error importante (las capas se crean mediante una propiedad de CSS llamada `z-index`) es un error tan común que es preferible seguir llamando “capas” a las zonas definidas con la etiqueta `<div>`.

Todas las páginas web complejas que están bien creadas utilizan decenas de etiquetas `<div>`. Los atributos más utilizados con esta etiqueta son `id` (para identificar de forma única al `<div>`) y `class` (para aplicarle estilos mediante CSS).

No obstante, como no es posible estructurar una página web utilizando `<div>` pero sin utilizar hojas de estilos CSS, no se va a profundizar en la forma de diseñar una página web mediante `<div>`.

Por último, si se observa el código HTML de algunas páginas web complejas, se llega a la conclusión de que la mayoría utilizan los mismos nombres para identificar sus zonas. Los nombres más comunes, y sus equivalente en inglés, se muestran a continuación:

- contenedor (*wrapper*) suele encerrar la mayor parte de los contenidos de la página y se emplea para definir las características básicas de la página: su anchura, sus bordes, imágenes laterales, si se centra o no, etc.
- cabecera (*header*) que incluye todos los elementos invariantes de la parte superior de la página (logotipo, imagen o banner, cuadro de búsqueda superior, etc.)
- contenido (*content*) engloba el contenido principal del sitio (la zona de noticias, la zona de artículos, la zona de productos, etc. dependiendo del tipo de sitio web)

- `menu` (`menu`) se emplea para agrupar todos los elementos del menú lateral de la página
- `pie` (`footer`) que incluye todos los elementos invariantes de la parte inferior de la página (aviso de copyright, política de privacidad, términos de uso, etc.)
- `lateral` (`sidebar`) se emplea para agrupar los elementos de las columnas laterales y secundarias de la página.

De esta forma, el esqueleto de una página HTML compleja suele ser similar al siguiente:

```
...
<div id="contenedor">
  <div id="cabecera">
    ...
  </div>

  <div id="contenido">
    <div id="menu">
      ..
    </div>
    ...
  </div>

  <div id="pie">
    ...
  </div>
</div>
...
```

El equivalente para las páginas en inglés sería el siguiente:

```
...
<div id="wrapper">
  <div id="header">
    ...
  </div>
```



```
<div id="content">  
  <div id="menu">  
    ..  
  </div>  
  ...  
</div>  
  
<div id="footer">  
  ...  
</div>  
</div>
```

Capítulo 10. Metainformación

10.1. Estructura de la cabecera

Tabla 10.1. Etiqueta head

head	Cabecera
Atributos comunes	i18n
Atributos específicos	<ul style="list-style-type: none"> • <code>profile = "url"</code> - Especifica la URL que contiene más información sobre el documento • <code>id = "texto"</code> - Identifica de forma única a cada elemento de un documento HTML
Tipo	-
Descripción	Define la cabecera del documento HTML

La cabecera típica de un documento HTML presenta la siguiente estructura:

```
<head>
<!-- Zona de etiquetas META -->
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<meta name="generator" content="WordPress 2.0.4" />

<!-- Zona de título -->
<title>El título del documento</title>

<!-- Zona de enlaces (CSS, RSS, JavaScript) -->
<link rel="stylesheet" href="#" type="text/css" media="screen" />
<link rel="stylesheet" href="#" type="text/css" media="print" />
```

```
<link rel="alternate" type="application/rss+xml" title="RSS 2.0" href="#" />
```

```
<script src="#" type="text/javascript"></script>
```

```
</head>
```

La etiqueta <title> permite indicar el título del documento. La mayoría de sitios web eligen un título genérico e idéntico para todas sus páginas web. Sin embargo, los buscadores consideran que el contenido de la etiqueta <title> es de vital importancia para determinar el contenido de una página.

De esta forma, uno de los aspectos más importantes del SEO (*Search Engine Optimization*) es la elección de títulos apropiados y diferentes para cada página.

Tabla 10.2. Etiqueta title

title	Título del documento
Atributos comunes	i18n
Atributos específicos	<ul style="list-style-type: none"> • id = "texto" - Identifica de forma única a cada elemento de un documento HTML
Tipo	-
Descripción	Define el título del documento HTML

Todos los documentos HTML deben tener definido un título (y solo uno).

10.2. Metainformación del documento

La especificación oficial de HTML permite incluir en los documentos cualquier metainformación, ya que no define una lista de metadatos disponibles. La etiqueta empleada para la definición de la metainformación es <meta>.

Tabla 10.3. Etiqueta meta

meta	Metadatos
Atributos comunes	i18n
Atributos específicos	<ul style="list-style-type: none"> • name = "texto" - El nombre de la propiedad que se define (no existe una lista oficial de propiedades) • content = "texto" - El valor de la propiedad definida (no existe una lista de valores permitidos) • http-equiv = "texto" - En ocasiones reemplaza al atributo "name" y lo emplean los servidores para adaptar sus respuestas al documento • scheme = "texto" - Indica el esquema que se debe emplear para interpretar el valor de la propiedad
Tipo	-
Descripción	Permite definir el valor de los metadatos que forman la metainformación del documento

Ejemplos de las etiquetas <meta> más comunes:

Definir el autor del documento

```
<meta name="author" content="Juan Pérez" />
```

Definir el programa con el que se ha creado el documento

```
<meta name="generator" content="WordPress 2.0.4" />
```

Definir la codificación de caracteres del documento

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
```

Definir el copyright del documento

```
<meta name="copyright" content="misitio" />
```

Definir el comportamiento de los motores de búsqueda

```
<meta name="robots" content="index, follow" />
```

Definir las palabras clave que definen el contenido del documento

```
<meta name="keywords" content="design, css, cascading style sheet, style sheet,
stylesheet,
css, web standards, webstandards, standards, inspiration, daily inspiration,
resource, guide,
showcase, promotion, gallery, web, webdesign, web design, web site design" />
```

Definir una breve descripción del sitio

```
<meta name="description" content="Articles and news on web standards,
accessibility, usability,
and other things related to web development and web design." />
```

La etiqueta que define la codificación de los caracteres se emplea prácticamente en todos los documentos, y las etiquetas que definen la descripción y las palabras claves también son muy utilizadas.

10.3. DOCTYPE

Los documentos XHTML siguen una estructura similar al resto de documentos creados con XML o alguno de sus lenguajes derivados. Uno de los conceptos fundamentales de XML es la creación de DTD o Document Type Definition (Definición del Tipo de Documento).

Un DTD es un conjunto de normas y restricciones que se definen para fijar la sintaxis que deben cumplir los documentos de ese tipo.

Si se define un DTD para la creación de documentos relacionados con libros, se puede fijar que cada libro debe tener un título y solo uno, que puede tener 1 o más autores, que la información sobre el número de páginas puede ser opcional, etc.

El conjunto de normas, obligaciones, restricciones y de todas las posibles combinaciones que se pueden utilizar al crear un documento de un determinado tipo, se recogen en el correspondiente DTD.

Los documentos XHTML tienen su propio DTD definido, por lo que no es necesario crear un DTD específico en el que se definan las etiquetas disponibles, los atributos que pueden tener, el tipo de valores que puede tener cada atributo, etc.

En realidad, existen numerosos DTD diferentes definidos que se pueden emplear para crear documentos HTML y XHTML. Para la versión XHTML 1.0 existen 3 DTD definidos y se indican mediante la etiqueta <doctype>.

XHTML 1.0 Estricto

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

No permite ningún atributo HTML que se utilice para presentación (por lo que requiere una total separación de código HTML y presentación CSS).

XHTML 1.0 Transitorio

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Se permite el uso de algunos atributos HTML que son de presentación.

XHTML 1.0 Frameset

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Junto con el DOCTYPE apropiado, es necesario indicar el namespace asociado. Un namespace de un documento XML permite diferenciar las etiquetas y atributos que pertenecen a cada lenguaje derivado de XML.

Si en un mismo documento se mezclan etiquetas de 2 o más lenguajes derivados de XML (XHTML y SVG por ejemplo) y que tienen el mismo nombre, no se podría determinar a qué

lenguaje pertenecen y por tanto, no se podría interpretar esa etiqueta o ese atributo. Los namespaces se indican mediante una URL.

El namespace que utilizan todos los documentos XHTML (independientemente de la versión y del DOCTYPE) es `http://www.w3.org/1999/xhtml` y se indica de la siguiente manera en los documentos HTML:

```
<html xmlns="http://www.w3.org/1999/xhtml">
...
</html>
```

De esta forma, la cabecera habitual de un documento XHTML es la siguiente:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="es" xml:lang="es">
```

Capítulo 11. Otras etiquetas y atributos importantes

11.1. JavaScript

Algunos navegadores no disponen de soporte completo de JavaScript, otros navegadores permiten bloquear JavaScript parcialmente e incluso algunos usuarios bloquean completamente el uso de JavaScript porque creen que así navegan de forma más segura.

En estos casos, es habitual que si la página web requiere JavaScript para su correcto funcionamiento, se incluya un mensaje de aviso al usuario indicándole que debería activar JavaScript para disfrutar completamente de la página. El siguiente ejemplo muestra una página web basada en JavaScript cuando se accede con JavaScript activado y cuando se accede con JavaScript completamente desactivado.

Imagen de www.Netvibes.com con JavaScript activado

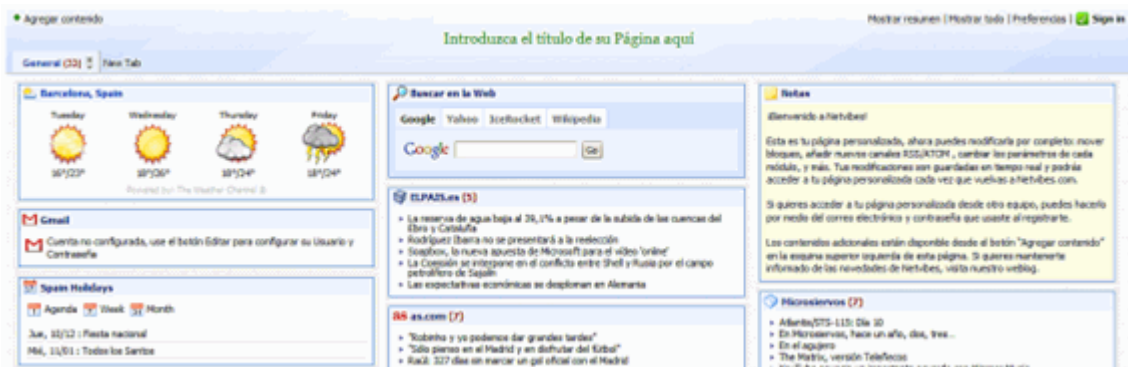


Figura 11.1. Ejemplo de página compleja con JavaScript activado

Imagen de www.Netvibes.com con JavaScript deshabilitado

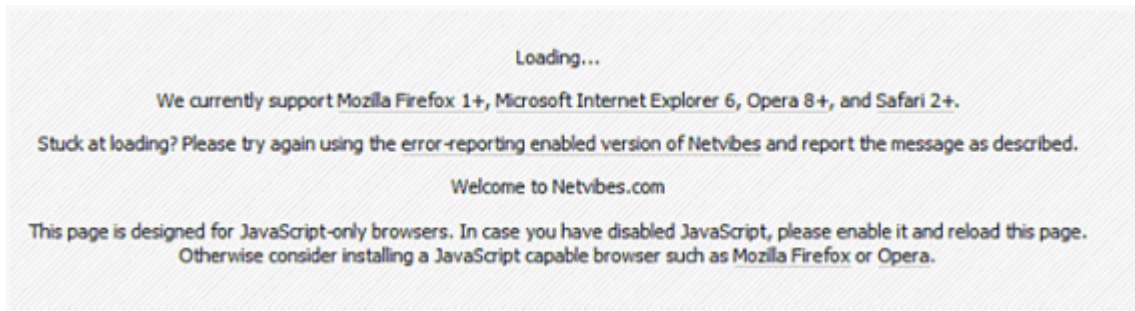


Figura 11.2. Ejemplo de página compleja con JavaScript desactivado

HTML define la etiqueta `<noscript>` para incluir un mensaje que se muestre al usuario cuando no se puede ejecutar JavaScript.

Tabla 11.1. Etiqueta noscript

noscript	Sin soporte de scripts
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	Bloque
Descripción	Define un mensaje alternativo que se muestra al usuario cuando su navegador no soporta la ejecución de scripts

Por tanto, incluir un mensaje de aviso que solamente sea visible en los navegadores que tienen bloqueado JavaScript es tan sencillo como utilizar la etiqueta `<noscript>`. **Ejemplo**

```
<head> ... </head>
<body>
<noscript>
  <p>Bienvenido a Mi Sitio</p>
  <p>La página que estás viendo requiere para su funcionamiento el uso de
JavaScript.
Si lo has deshabilitado intencionadamente, por favor vuelve a activarlo.</p>
```

```
</noscript>  
</body>
```

11.2. Comentarios

Al igual que la mayoría de lenguajes, HTML permite incluir comentarios dentro de su código. Los comentarios no se muestran por pantalla, pero sí que aparecen en el código HTML del documento. Por este motivo, nunca debe incluirse información sensible o confidencial en los comentarios.

La principal utilidad de los comentarios es la de ayudar a los diseñadores y programadores a determinar el comienzo y el final de las secciones de código, sobre todo en los documentos complejos.

La sintaxis de los comentarios es la siguiente:

- Apertura del comentario: `<!--`
- Contenido del comentario: (cualquier texto)
- Cierre del comentario: `-->`

Ejemplo de uso de los comentarios

```
<!-- Inicio de menú -->  
<div id="menu">  
<ul>  
  <li>...</li>  
  <li>...</li>  
  <li>...</li>  
  <li>...</li>  
</ul>  
<!-- Fin de menú -->  
  
<!-- Inicio de publicidad -->  
<div id="ad"> ... </div>  
<!-- Fin de publicidad -->
```

El inicio y el final de un comentario pueden estar en diferentes líneas, ya que un comentario de HTML puede ocupar tantas líneas como sea necesario. Además, los comentarios no se pueden anidar, es decir, no se puede incluir un comentario dentro de otro comentario.

11.3. CSS

Los atributos más utilizados (a excepción del atributo `id`) en la creación del código HTML de los sitios web actuales, están relacionados con CSS: `style` y `class`.

El atributo `class` permite definir la clase CSS que se aplicará a un elemento. Ejemplo:

```
<p class="resumen">Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Maecenas at diam id enim viverra semper. Nulla id urna. Donec sodales.</p>
```

El uso del atributo `class` es habitual para aplicar estilos CSS de forma semántica:

```
<div class="caja_destacada">
  <h3>Titular</h3>
  <p class="resumen">Lorem ipsum dolor sit amet, <span
class="destacado">consectetur
adipiscing elit</span>. Maecenas at diam id enim viverra semper. Nulla id urna.
Donec sodales.</p>
</div>
```

El `<div>` anterior define una caja destacada de texto y por eso su estilo se ha denominado “caja_destacada” y NO “caja_con_borde_negro_2_pixel”.

En cuanto a `style`, HTML define un atributo llamado `style` y una etiqueta llamada `<style>`. El atributo `style` se emplea para definir estilos directamente sobre los elementos HTML. La etiqueta `<style>` se utiliza para incluir bloques de código CSS.

Ejemplo de estilos definidos mediante el atributo `style`

```
<p>Algunas palabras de esta frase se muestran de <span style="color:red">color
rojo</span></p>
```

Ejemplo de estilos definidos mediante un bloque de código CSS

```
<head>
...
<style type="text/css">
  span {color:red;}
</style>
</head>
```

Evidentemente, esta sección es solo un esbozo de las posibilidades de CSS. De hecho, pronto publicaremos un libro de CSS de más de 200 páginas.

11.4. Iframes

La etiqueta `<iframe>` es una de las etiquetas más extrañas de HTML. Su única utilidad es la de insertar un documento HTML dentro de otro documento HTML. Un `iframe` puede considerarse como un *agujero* que se abre en una página web y en el que se muestra otra página web.

La utilidad principal es la de mostrar contenidos externos al sitio web como si fueran parte del mismo sitio. Otra vez se emplea para incluir una aplicación común a varios sitios de una misma empresa.

La página de **Google Analytics** emplea un `<iframe>` para incluir la página correspondiente a la validación de usuario.



Figura 11.3. Ejemplo de página con un iframe

Tabla 11.2. Etiqueta iframe

iframe	Marco (frame) en línea
Atributos comunes	básicos
Atributos específicos	<ul style="list-style-type: none"> • src = "ur1" - Localización del documento HTML que se visualiza en el iframe • height = "longitud" - Altura que ocupará el iframe en el documento • width = "longitud" - Anchura que ocupará el iframe en el documento

iframe	Marco (frame) en línea
	<ul style="list-style-type: none"> • name = "texto" - Nombre que identifica al iframe • longdesc = "url" - Dirección en la que puede encontrarse una descripción larga del contenido del iframe • scrolling = "yes no auto" - Indica si el iframe debe mostrar barras de scroll (horizontal y vertical) cuando el contenido incluido no cabe en el iframe
Tipo	Bloque y en línea
Descripción	Se emplea para incluir en línea un marco que muestre otro documento HTML

Ejemplo sencillo de inclusión de un <iframe>

```
<iframe src="/ruta/documento.html" width="250" height="250" scrolling="no" />
```

Ejemplo real de uso de <iframe>

```
<iframe src="https://www.google.com/accounts/ServiceLoginBox?service=analytics&nui=1&hl=en-US" marginwidth="0" marginheight="0" frameborder="0" scrolling="no" width="220" height="250" />
```

11.5. Otras etiquetas

La etiqueta <address> es una de las etiquetas más desconocidas y por ese motivo no se utiliza mucho. Su única misión es la de definir la información de contacto que puede incluir un documento.

Tabla 11.3. Etiqueta address

address	Direcciones
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	Bloque
Descripción	Identifica la información de contacto de un documento

Ejemplo sencillo

```
<address>
  Nombre de la empresa
  Dirección completa
  Teléfono y Fax
</address>
```

Ejemplo completo

```
<address>
  <a href="../../People/Raggett/">Dave Raggett</a>,
  <a href="../../People/Arnaud/">Arnaud Le Hors</a>,
  contact persons for the <a href="Activity">W3C HTML Activity</a><br/>
  $Date: 1999/12/24 23:37:50 $
</address>
```

Hace unos años, la etiqueta `<hr>` era una de las más utilizadas. Sin embargo, con el paso del tiempo su uso ha disminuido ya que se trata de un elemento que se considera de presentación y CSS incluye alternativas más adecuadas. La etiqueta `<hr>` permite mostrar una línea horizontal de separación.

Tabla 11.4. Etiqueta hr

hr	Línea horizontal
Atributos comunes	básicos, i18n y eventos
Atributos específicos	-
Tipo	Bloque
Descripción	Permite incluir una línea horizontal de separación

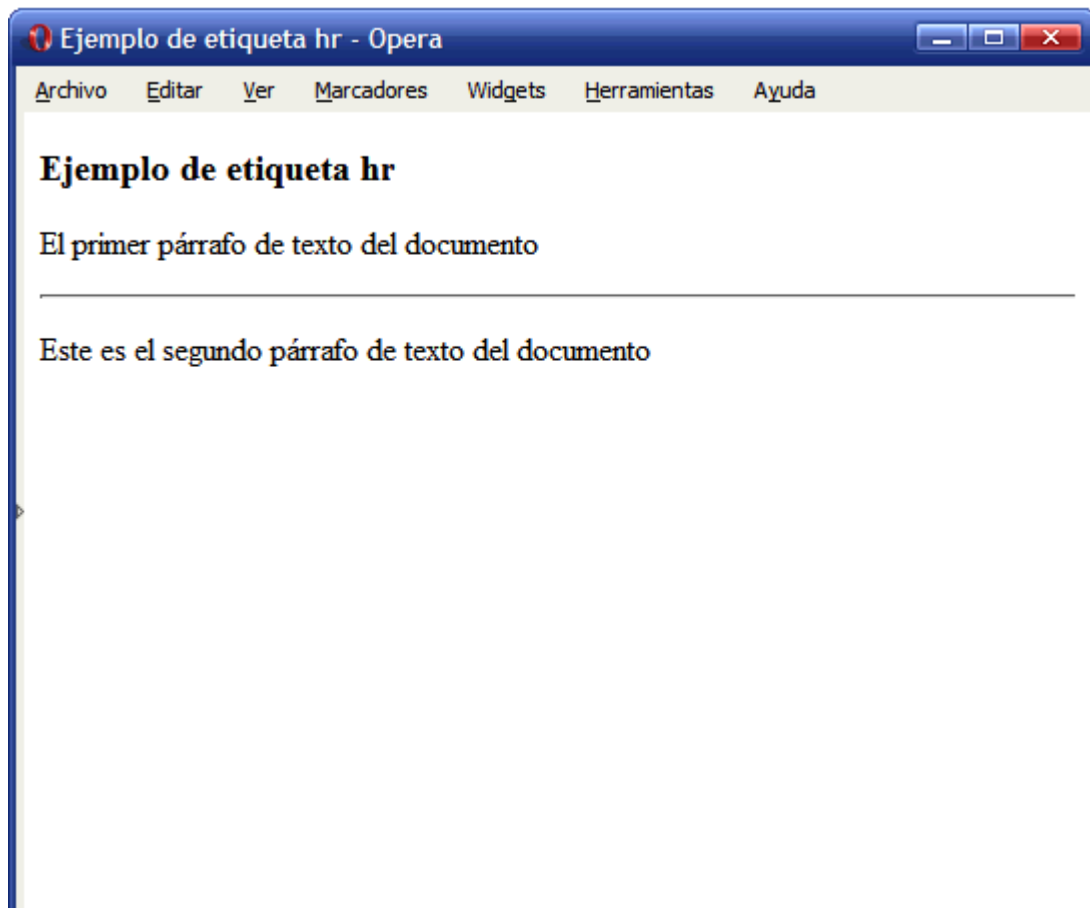


Figura 11.4. Ejemplo de uso de la etiqueta hr

El código HTML del ejemplo anterior se muestra a continuación:

```
<html>
<head><title>Ejemplo de etiqueta hr</title></head>
```



```
<body>
```

```
<h3>Ejemplo de etiqueta hr</h3>
```

```
<p>El primer párrafo de texto del documento</p>
```

```
<hr/>
```

```
<p>Este es el segundo párrafo de texto del documento</p>
```

```
</body>
```

```
</html>
```

Capítulo 12. Otros conceptos relacionados

12.1. Accesibilidad

El principal objetivo de la accesibilidad web es el de permitir a cualquier usuario (independientemente del tipo de discapacidad que sufra) el acceso a los contenidos del sitio y que permita la navegación necesaria para realizar los procesos deseados.

Los sitios web accesibles no solamente facilitan el acceso de sus contenidos a los usuarios discapacitados, sino que también permiten ofrecer la misma funcionalidad con dispositivos muy limitados (dispositivos no visuales, dispositivos con pantallas minúsculas, etc.).

Las 4 principales ventajas de diseñar un sitio web completamente accesible son las siguientes:

- Los sitios accesibles separan completamente diseño y contenido.
- Un sitio accesible puede ser accedido por multitud de dispositivos diferentes sin necesidad de reescribir el código HTML.
- Los sitios accesibles son los únicos con una audiencia potencial global, ya que permiten el acceso a todos los usuarios y a todos los dispositivos.
- Generalmente, los sitios accesibles son más fáciles de utilizar también para los usuarios sin discapacidades.

La creación de sitios accesibles puede realizarse siguiendo las recomendaciones establecidas por el W3C y que se recogen en el WCAG (Web Content Accessibility Guidelines).

En el momento de escribir estas líneas, la especificación oficial es la versión 1.0 que se publicó en 1999 y se puede acceder en <http://www.w3.org/TR/WCAG10/>

La versión 2.0 aun se encuentra en borrador, siendo su última versión la publicada el día 27 de Abril de 2006.

Las recomendaciones del WCAG 1.0 están formadas por 65 requisitos que un sitio debe cumplir para considerarse accesible. Los requerimientos se agrupan en prioridades.

Los requisitos de prioridad 1 son de obligado cumplimiento, los de prioridad 2 son recomendables y los de prioridad 3 son deseables. Si un sitio cumple con todos los requisitos de prioridad 1, se considera que el sitio es conforme al nivel A de accesibilidad.

El nivel AA de accesibilidad está reservado para los sitios que cumplan todos los requisitos de prioridad 1 y prioridad 2. Finalmente, los sitios que cumplen los 65 requisitos, son conformes al nivel AAA de accesibilidad.

12.1.1. Requisitos del nivel A de accesibilidad

Los requisitos de accesibilidad que exige el nivel A son los siguientes:

En general

1.1 Proporcionar un texto alternativo para todas las imágenes, objetos y otros elementos no textuales (mediante los atributos `alt` y `longdesc`).

2.1 Asegurar que toda la información que utilice el color como elemento informativo pueda ser entendida por las personas o dispositivos que no pueden distinguir los colores.

4.1 Marcar claramente (mediante los atributos `lang` y `xml:lang`) las variaciones del idioma del texto o de los elementos textuales (`<caption>`) respecto del idioma principal de la página.

6.1 El documento debe poder leerse completamente cuando no se utilicen hojas de estilos.

6.2 La información equivalente para los contenidos dinámicos debe adaptarse a los cambios de los contenidos dinámicos.

7.1 Ningún elemento debe parpadear en la pantalla.

14.1 El contenido del sitio se debe escribir con un lenguaje sencillo y limpio.

Si se utilizan mapas de imagen

1.2 Proporcionar un enlace textual por cada una de las regiones del mapa de imagen.

9.1 Utilizar mapas de imagen en el cliente, en vez de mapas de imagen de servidor.

Si se utilizan tablas

5.1 Utilizar cabeceras de fila y de columna.

5.2 Si la tabla tiene varios niveles de cabeceras, utilizar las agrupaciones disponibles (thead, tfoot).

Si se utilizan frames

12.1 Indicar un título a cada frame para su identificación y facilitar la navegación.

Si se utilizan applets y scripts

6.3 Asegurar que la página sea igualmente utilizable cuando no se ejecutan los applets y los scripts. Si no es posible, proporcionar informaciones equivalente o páginas alternativas que sean accesibles.

Si se utilizan contenidos multimedia (audio y vídeo)

1.3 Incluir una descripción textual del contenido multimedia.

1.4 Para los contenidos basados en vídeo o animaciones, sincronizar las alternativas textuales con la presentación.

Si no se pueden cumplir los anteriores requisitos

11.4 Proporcionar una página alternativa con la mayor cantidad posible de contenidos y que cumpla con los requisitos anteriores.

La lista completa con los 65 requisitos de los 3 niveles de accesibilidad se puede consultar en <http://www.w3.org/TR/WCAG10/full-checklist.html>

12.2. Validación

La validación es el proceso mediante el que asegura que un documento escrito en un determinado lenguaje (por ejemplo XHTML) cumple con las normas y restricciones de ese lenguaje.

Las normas y restricciones de los documentos escritos en XML (y en sus lenguajes derivados, como XHTML) se definen en el DTD o Document Type Definition (Definición del Tipo de Documento).

Existen varios validadores de documentos XHTML, tanto online como offline. Dreamweaver incluye una herramienta de validación que se puede acceder desde la opción **Edición > Preferencias > Validador**:

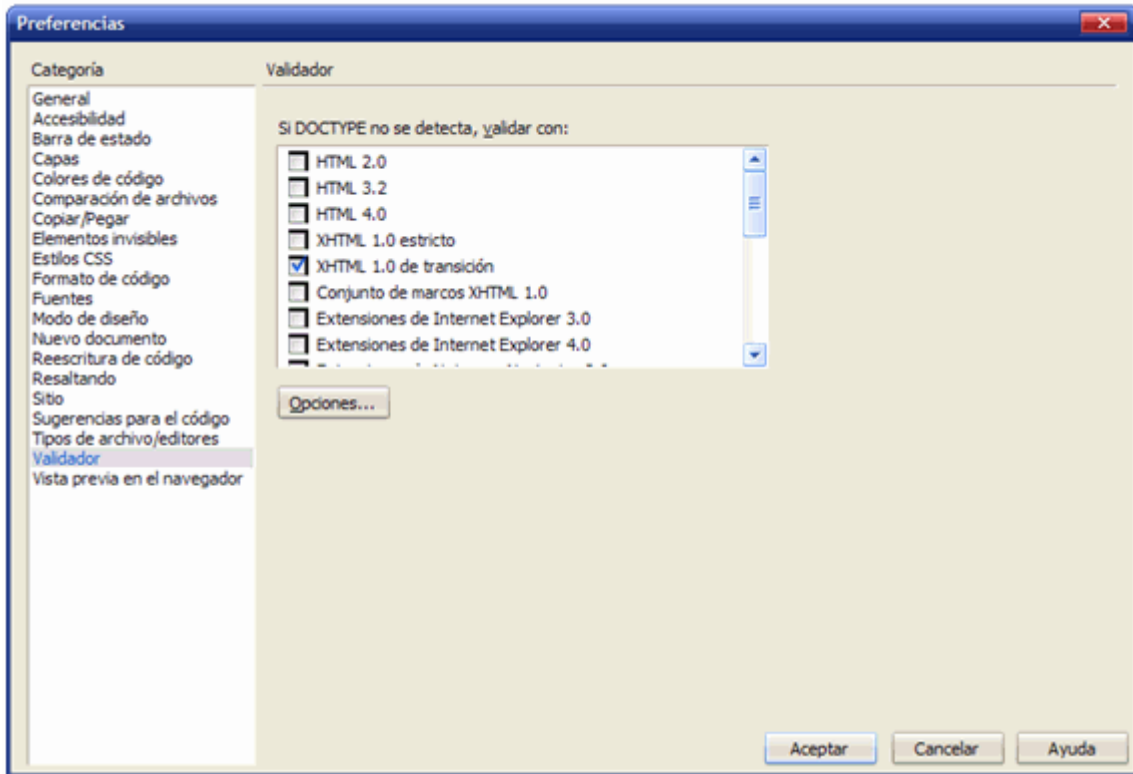


Figura 12.1. Ventana principal del validador de Dreamweaver

Para validar los documentos se debe seleccionar primero el DTD a utilizar (en la imagen anterior, se ha elegido el DTD de XHTML 1.0 de transición). Una vez seleccionado el DTD, Dreamweaver muestra la lista de todos los errores que ha encontrado:

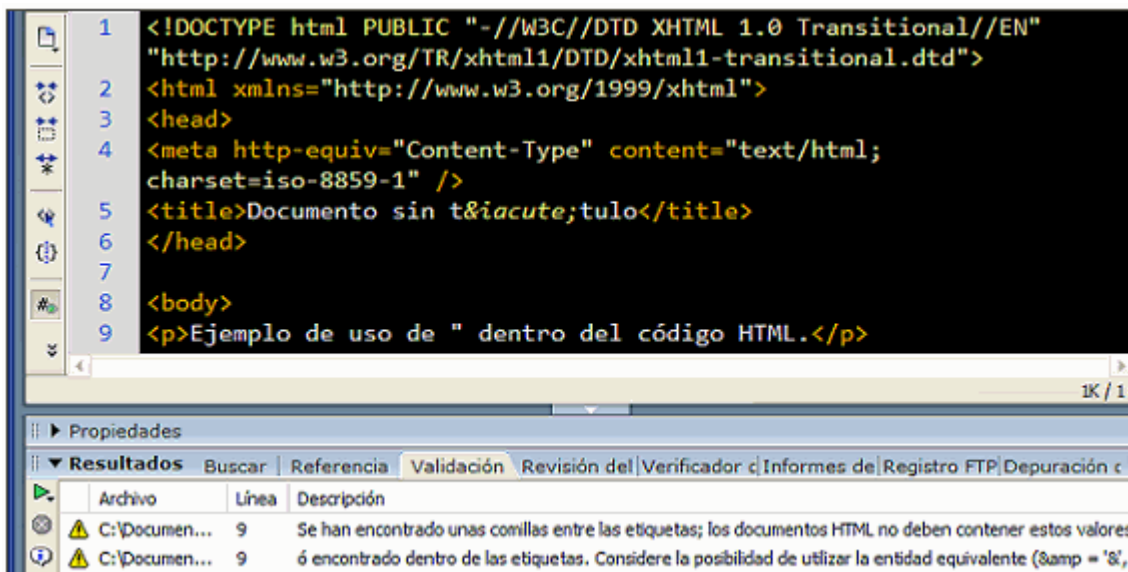


Figura 12.2. Resultado de validar una página con Dreamweaver

Otro método común de validación es mediante la herramienta online dispuesta por el W3C (<http://validator.w3.org/>):

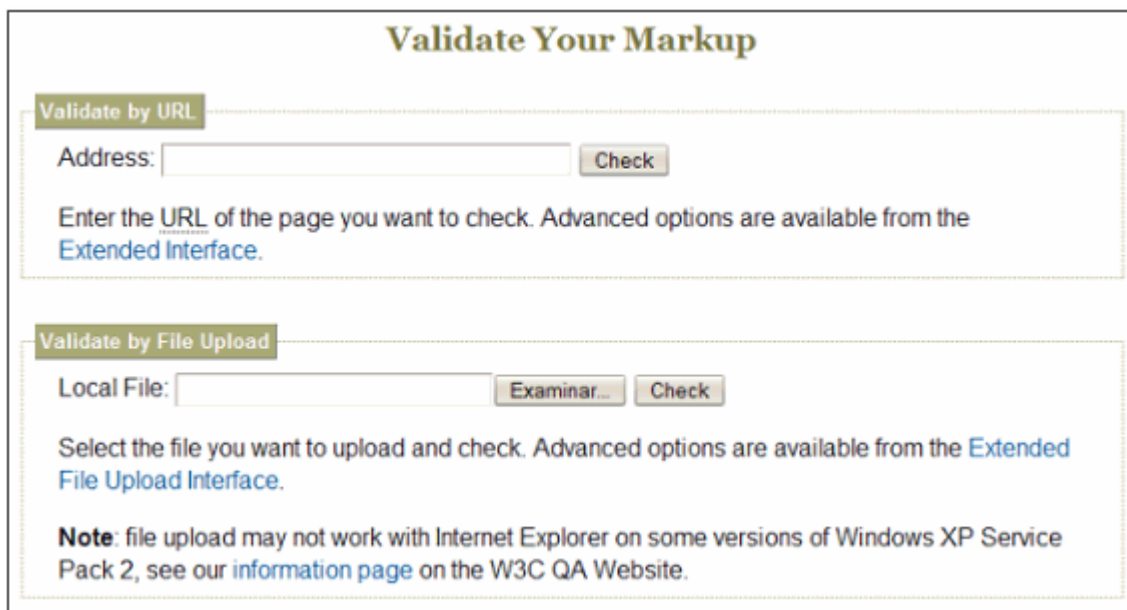


Figura 12.3. Página principal del validador del W3C

Se pueden validar documentos y páginas publicadas en Internet, archivos locales e incluso trozos de código. Una vez validado, la herramienta muestra el resultado indicando los errores encontrados o mostrando el DTD para el que el documento es válido.

Ejemplo de validación de la propia página del validador de W3C

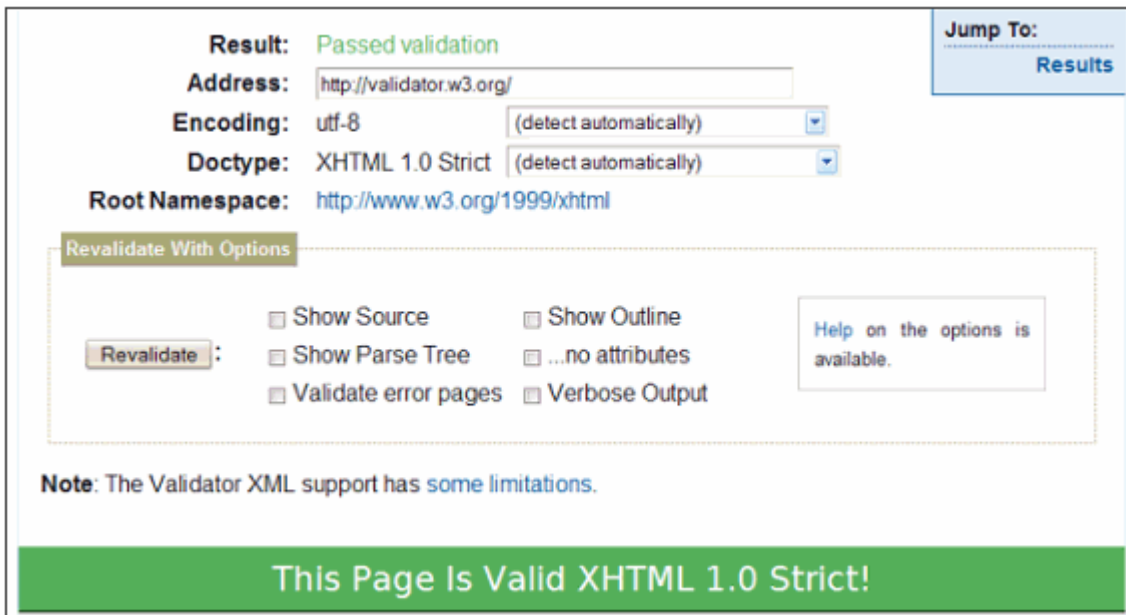


Figura 12.4. Resultado de validar una página con el validador de W3C

12.3. Codificación de caracteres

Algunos de los caracteres disponibles en el teclado no se pueden incorporar directamente a un documento HTML. Los caracteres *especiales* como <, > y “ requieren una notación especial para que los navegadores los muestren y no intenten interpretarlos.

Por otra parte, algunos caracteres *normales* como la ñ, las letras acentuadas (á, ..., ú) y algunas letras propias de algunos idiomas (ç, ì, ï) requieren de un tratamiento especial en el código HTML.

Las diferentes aplicaciones que intervienen en el proceso de creación y distribución de contenidos HTML (editor de diseñador web, editor del programador web, servidor que aloja los contenidos y navegador del usuario) almacenan y distribuyen los contenidos con una determinada codificación de caracteres.

La codificación de caracteres consiste en asignar un código numérico a cada carácter para almacenarlo de forma electrónica. Los códigos más antiguos como ASCII (publicado en 1967) solamente permitían el uso de 127 caracteres definidos (todos los caracteres utilizados por el idioma inglés).

Los códigos más modernos como el ISO-8859-1 y UTF-8 soportan muchos más caracteres. El código ISO-8859-1 define 191 caracteres que abarcan todos los idiomas que utilizan el alfabeto latino. El código UTF-8 soporta miles de caracteres de casi todos los alfabetos conocidos.

Si todos los elementos que intervienen en el proceso de creación y distribución de contenidos HTML utilizan la misma codificación (ISO-8859-1 o UTF-8) los caracteres *especiales* se pueden escribir directamente en el código:

```
<p>Este párrafo contiene caracteres acentuados y se almacena en formato UTF-8</p>
```

Sin embargo, en entornos distribuidos entre diferentes aplicaciones, diferentes responsables e incluso diferentes sistemas operativos, suele ser muy difícil poder asegurar que todos emplean y soportan la misma codificación.

De esta forma, HTML define un nuevo método de incluir caracteres especiales en los documentos HTML de forma rápida y sencilla (y compatible con cualquier codificación).

De esta forma, el anterior ejemplo se puede reescribir de la siguiente forma:

```
<p>Este párrafo contiene caracteres acentuados y se almacena en formato UTF-8</p>
```

El carácter á se ha reemplazado por la entidad ´ (a acute = a acentuada). HTML define decenas de entidades que permiten escribir de forma segura todos los caracteres especiales:

```
<p>&iquest;Las e&ntilde;es y los acentos est&aacute;n incluidos? S&iacute;, y tambi&eacute;n los caracteres de las etiquetas HTML &lt; y &gt;. </p>
```

Las entidades HTML que más se utilizan son las siguientes:

Entidad	Carácter	Descripción	Traducción
<	<	less than	signo de menor que
>	>	more than	signo de mayor que
&	&	ampersand	ampersand
"	"	quotation mark	comillas
 	(espacio en blanco)	no-break space	espacio en blanco

'	'	apostrophe	apóstrofo
ñ	ñ	latin letter n with tilde	ñ
Ñ	Ñ	latin capital n letter with tilde	Ñ
á	á	a acute	á
é	é	e acute	é
í	í	i acute	í
ó	ó	o acute	ó
ú	ú	u acute	ú
Á	Á	A acute	Á
É	É	E acute	É
Í	Í	I acute	Í
Ó	Ó	O acute	Ó
Ú	Ú	U acute	Ú
€	€	euro	signo del euro

La lista completa con las 252 entidades definidas se puede consultar en:

http://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references

Capítulo 13. Fragmentos de código

En el desarrollo de páginas web, es habitual repetir una y otra vez algunos trozos de código HTML como por ejemplo las tablas, los formularios y la cabecera de las páginas. Para evitar tener que reescribir continuamente el mismo código, se utilizan lo que se conoce como *fragmentos de código*, también llamados “*snippets*” por su nombre en inglés.

Si por ejemplo se quiere crear un fragmento de código de una tabla completa de XHTML, se deben realizar los siguientes pasos:

- Se escribe el código XHTML completo de una tabla vacía (con sus etiquetas <caption>, <thead>, <tbody>, <tfoot>, sus atributos summary, scope, etc.)
- Se guarda el código anterior en un archivo de texto con un nombre fácil de identificar (por ejemplo, “Tabla XHTML”).
- Cuando se necesite insertar una tabla en una página XHTML, se copia y se pega todo el código creado anteriormente y se completa con los datos necesarios.

El método descrito anteriormente no es muy eficiente si se debe realizar de forma manual. Afortunadamente, muchos programas utilizados para el diseño web disponen de la posibilidad de gestionar de forma más cómoda los fragmentos de código o *snippets*.

A continuación se muestra la forma de trabajar con los fragmentos de código en el conocido programa Dreamweaver:

1) Si no está visible, se debe mostrar la ventana correspondiente a los fragmentos de código mediante la opción de menú Ventana > Fragmentos o pulsando Mayúsc. + F9. Una vez abierta esta ventana, el aspecto que muestra se ve en la parte inferior izquierda de la siguiente imagen:

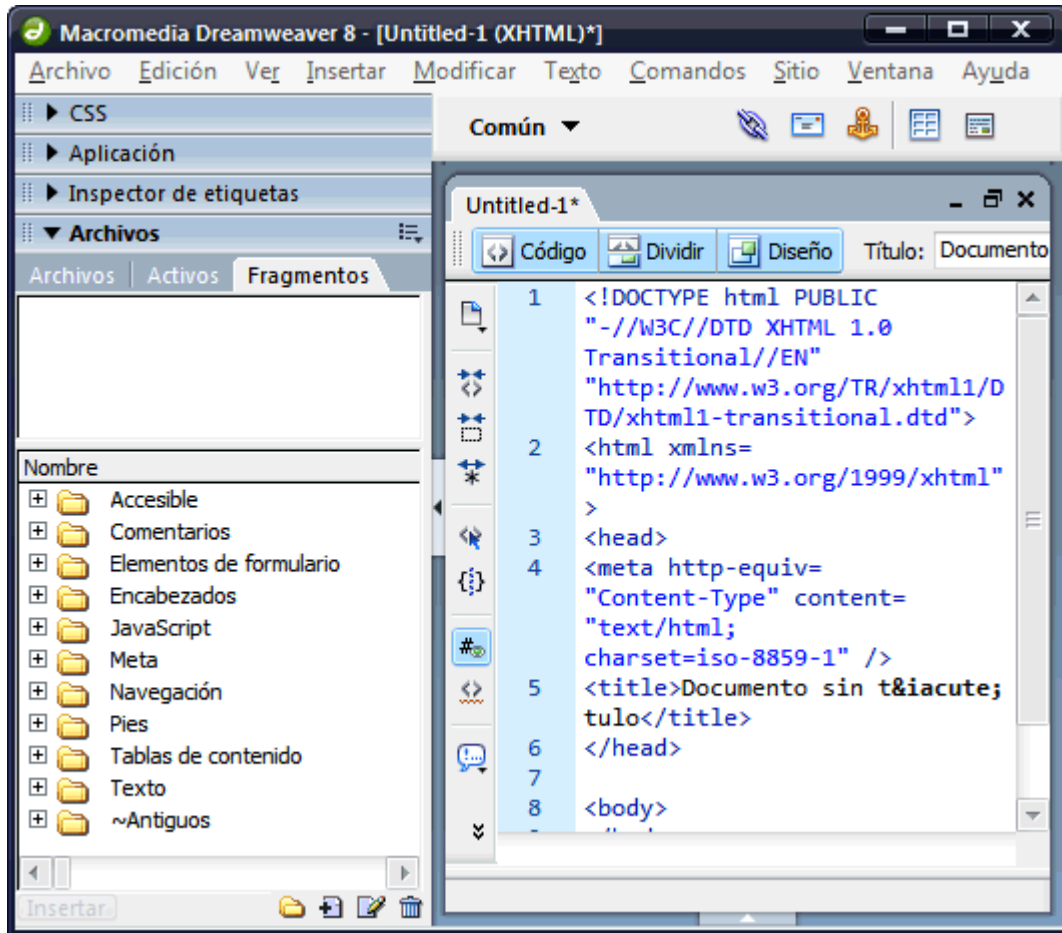


Figura 13.1. Visualizando la ventana de fragmentos de código en Dreamweaver

2) Dreamweaver ya dispone por defecto de muchos fragmentos de código. Si por ejemplo se pulsa sobre la carpeta *Meta*, Dreamweaver muestra un *snippet* llamado *No crear caché*. Al pinchar 2 veces sobre el nombre del fragmento, se inserta su contenido en el lugar en el que se encuentre el cursor dentro de la página, tal y como muestra la siguiente imagen:

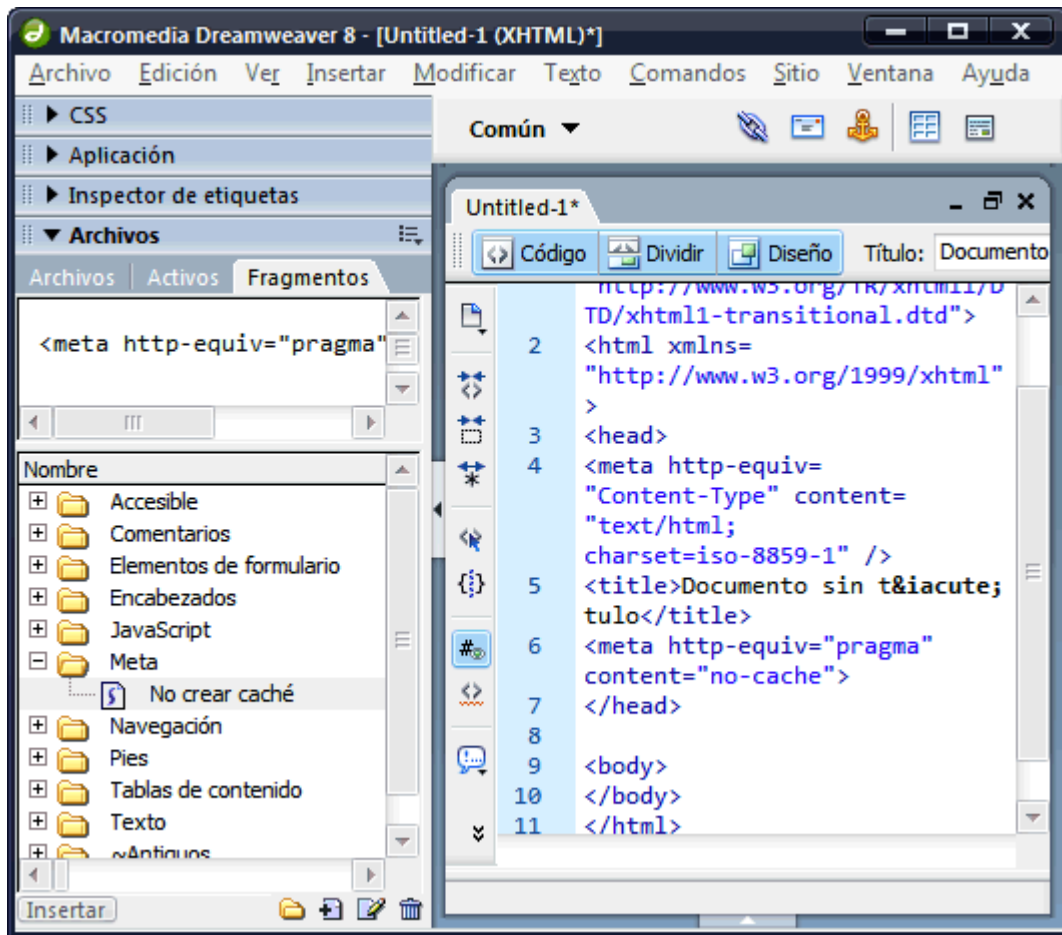


Figura 13.2. Insertando un fragmento de código de Dreamweaver en una página web

3) Para crear un fragmento de código propio, se crea en primer lugar una carpeta en la que se guardarán todos los *snippets* propios. El nombre elegido para esta carpeta es Propios y la forma de crearla es pinchando sobre el pequeño icono que simboliza una carpeta:

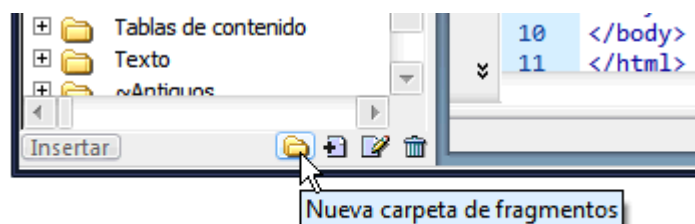


Figura 13.3. Creando una nueva carpeta para guardar los fragmentos de código propios

Una vez escrito el nombre de la nueva carpeta, el aspecto que muestra la ventana de *snippets* es el siguiente:

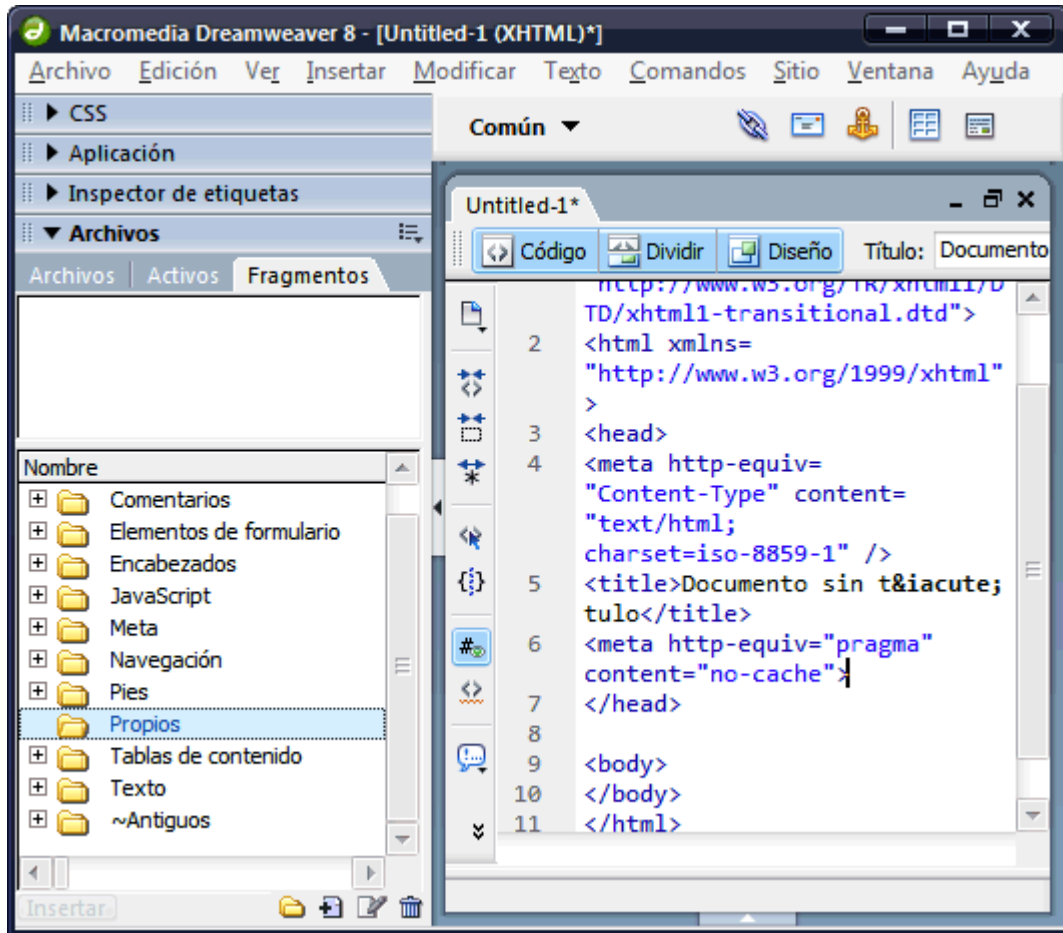


Figura 13.4. Nueva carpeta creada para guardar los fragmentos de código propios

4) Para crear un fragmento de código propio, se pulsa sobre el pequeño icono que simboliza un nuevo documento:

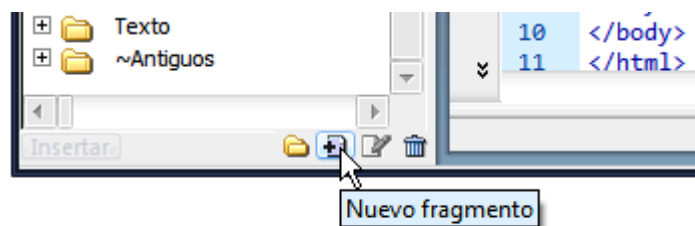


Figura 13.5. Creando un nuevo fragmento de código

Después de pulsar sobre ese icono, se muestra la ventana en la que se puede indicar el nombre y el contenido del fragmento:

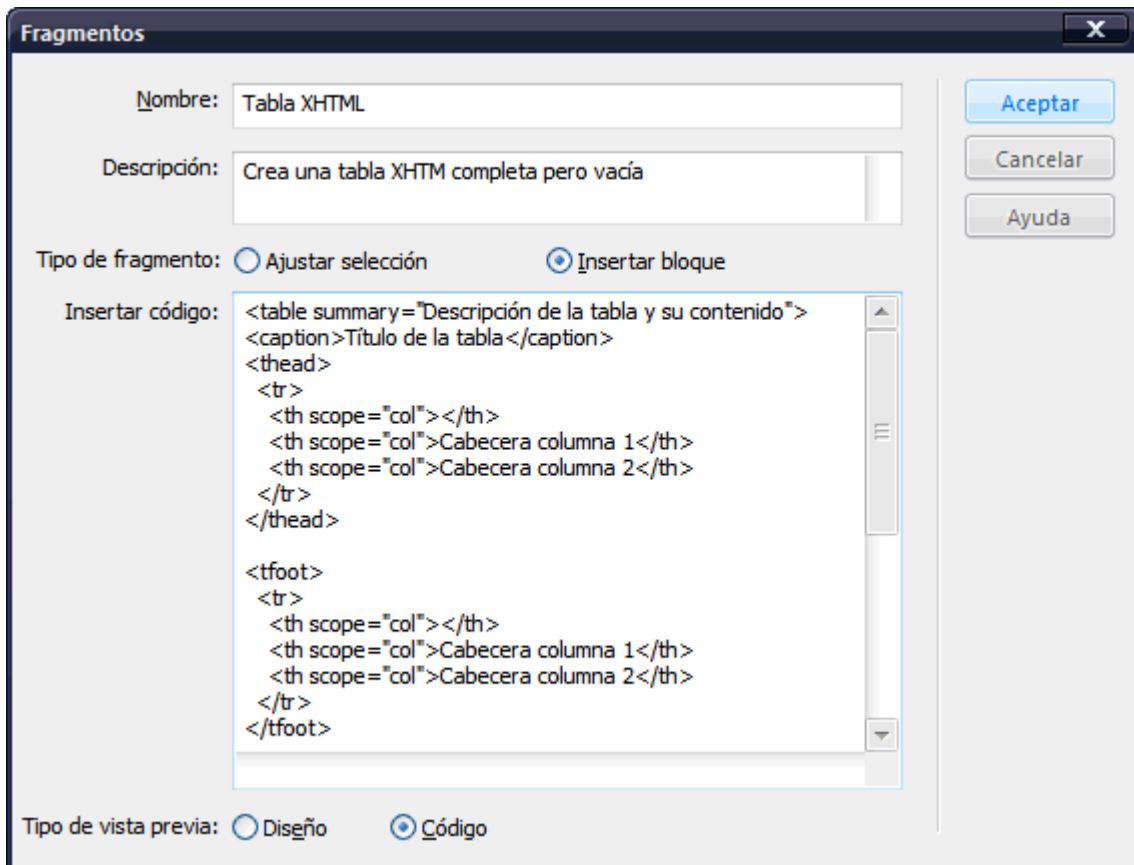


Figura 13.6. Propiedades (nombre, descripción y contenido) del fragmento de código propio

En este caso, se trata de un fragmento de código que inserta un bloque completo de código XHTML. También es posible crear fragmentos que añaden código XHTML antes y después del texto que ha sido seleccionado previamente.

Una vez creado el *snippet*, ya se puede insertar en cualquier zona de la página XHTML simplemente pulsando 2 veces sobre su nombre:

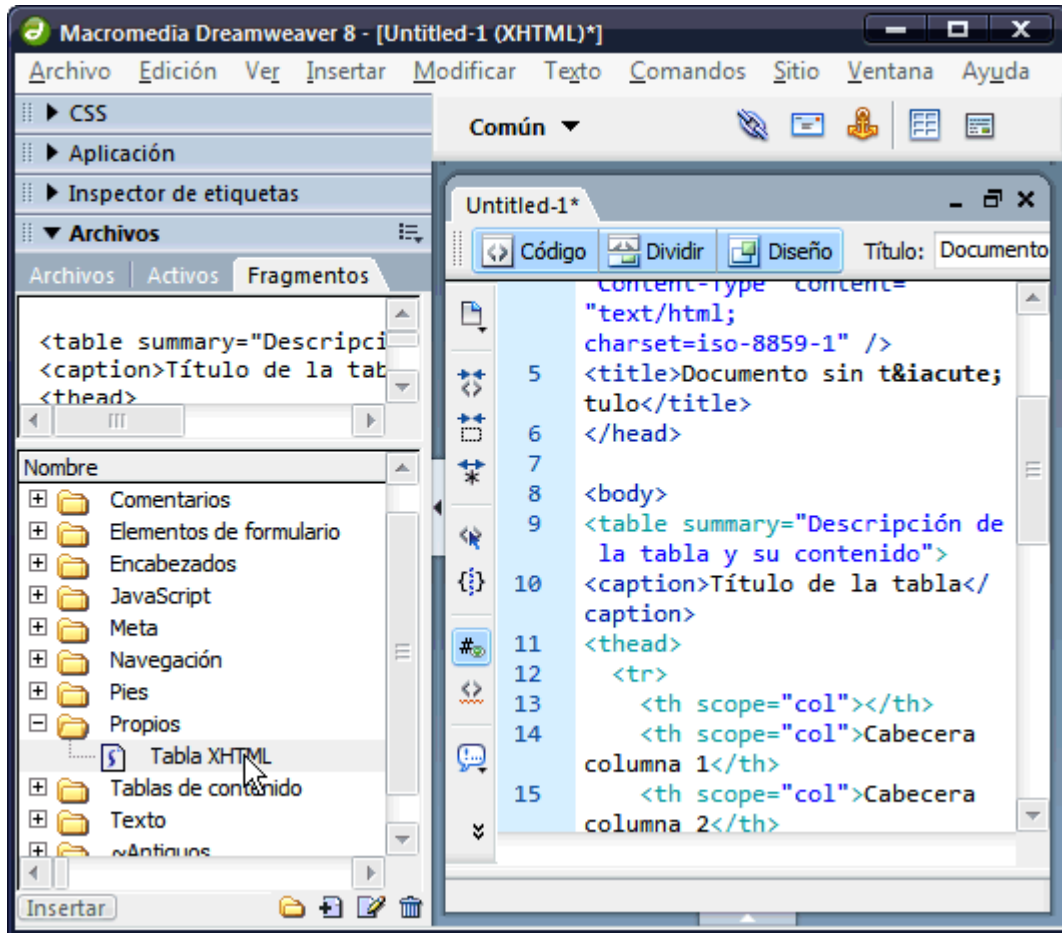


Figura 13.7. Insertando un fragmento de código propio en una página XHTML

13.1. Documento XHTML

Los elementos y etiquetas imprescindibles de un documento XHTML son los siguientes:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="es" xml:lang="es">
<head>
...
</head>
<body>
...

```

```
</body>
</html>
```

El DOCTYPE también puede ser:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

13.2. Cabecera XHTML

Las cabeceras HTML pueden llegar a ser muy complejas, ya que incluyen metainformación de la página, enlaces a hojas de estilos, archivos JavaScript y archivos RSS.

```
<head>
<title>Título de la página</title>

<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
<meta name="description" content="..." />
...
<meta name="author" content="..." />

<style type="text/css" media="screen,projection">
    @import '/css/archivo.css';
</style>
<link rel="stylesheet" type="text/css" href="/css/impresora.css" media="print" />
<link rel="stylesheet" type="text/css" href="/css/movil.css" media="handheld" />

<link rel="shortcut icon" href="/favicon.ico" type="image/ico" />

<link rel="alternate" type="application/rss+xml" title="..."
href="/archivo_rss.xml" />

<script type="text/javascript" src="/js/archivo.js" />

<style type="text/css">
```



```
/* Código CSS */
</style>

<script type="text/javascript">
  <![CDATA[
    // Código JavaScript
  ]]>
</script>
</head>
```

13.3. Tabla

Reescribir el código XHTML de una tabla avanzada es una tarea tediosa, ya que incluye secciones de cabecera y de pie, celdas que pueden ser cabecera de columna o de fila e información para mejorar la accesibilidad.

```
<table summary="Descripción de la tabla y su contenido">
<caption>Título de la tabla</caption>
<thead>
  <tr>
    <th scope="col"></th>
    <th scope="col">Cabecera columna 1</th>
    <th scope="col">Cabecera columna 2</th>
  </tr>
</thead>

<tfoot>
  <tr>
    <th scope="col"></th>
    <th scope="col">Cabecera columna 1</th>
    <th scope="col">Cabecera columna 2</th>
  </tr>
</tfoot>
```

```

<tbody>
  <tr>
    <th scope="row">Cabecera fila 1</th>
    <td>...</td>
    <td>...</td>
  </tr>
  <tr>
    <th scope="row">Cabecera fila 2</th>
    <td>...</td>
    <td>...</td>
  </tr>
</tbody>

</table>

```

13.4. Formulario

La clave de un formulario XHTML correcto reside en incluir la agrupación de campos de formulario (etiquetas `<fieldset>` y `<legend>`) y la definición de las etiquetas de cada campo (etiqueta `<label>`).

```

<form id="identificador" method="post" action="" enctype="multipart/form-data">

<fieldset>
<legend>Título del formulario</legend>

<label for="campo_texto">Campo de texto</label>
<input id="campo_texto" name="campo_texto" type="text" maxlength="255" value=""/>

<label for="campo_areatexto">Area de texto</label>
<textarea id="campo_areatexto" name="campo_areatexto" rows="10"
cols="50"></textarea>

```

```
<label for="campo_desplegable">Lista desplegable</label>
<select id="campo_desplegable" name="campo_desplegable">
  <option value="0" selected="selected">- selecciona -</option>
  <option value="1">Valor 1</option>
  <option value="2">Valor 2</option>
  <option value="3">Valor 3</option>
</select>

<label for="campo_fichero">Subir un fichero</label>
<input id="campo_fichero" name="campo_fichero" type="file" value=""/>

<input type="radio" id="valor1" name="campo_radio" value="valor1"
checked="checked" />
<label for="valor1">Valor 1</label>
<input type="radio" id="valor2" name="campo_radio" value="valor2" />
<label for="valor2">Valor 2</label>

<input id="campo_check" name="campo_check" type="checkbox" value="valor1"
checked="checked"/>
<label for="campo_check">Campo check 1</label>

<input id="boton_enviar" type="submit" value="Enviar formulario" />

</fieldset>
</form>
```