

CAPITULO 1: Presentación de HTML5

1-Breve historia de HTML5

Comenzamos esta historia en 1998 con la finalización de la especificación Html 4.0. Dejando aparte una ligera revisión en 1999 con Html 4.01, no ha habido nada nuevo en el horizonte de este primer lenguaje Web. El W3C (*World WideWeb Consortium*), organismo que gestiona los estándares de escritura en la red, ¿no anunció en su momento que Html, en su versión 4.0, no conocería ningún desarrollo posterior y, en breve, que Html estaba muerto?

El W3C editó, también en 1998, el estándar XML (*eXtensible Markup Language* o, en castellano, lenguaje de marcas extensible) para gestionar de forma estructurada los datos de tipo texto, donde el aspecto extensible del lenguaje permitía al usuario definir su propio lenguaje (comportándose como un metalenguaje) con sus distintos elementos. El W3C recomendaba encarecidamente, en la época, XML para expresar lenguajes de marcado específico.

Una de las primeras aplicaciones concretas de XML fue la reformulación de Html 4.0 según la sintaxis estricta y formalista de XML. Nació así Xhtml 1.0 en el año 2000. Si bien los desarrolladores saludaron unánimemente el rigor que Xhtml aportaba a la escritura de código, es justo señalar que la migración a Xhtml no ha supuesto el éxito esperado e incluso una gran parte de los diseñadores han permanecido fieles a Html 4.0, más laxo.

Muy rápidamente, el W3C puso en marcha el ambicioso proyecto de Xhtml 2.0, que debía suponer un punto y aparte respecto al pasado en materia de publicación en la red. Esta posición indujo a que Xhtml 2.0 no fuera compatible hacia atrás con todo lo que existía hasta la época, y por consiguiente con Html.

Esta ausencia de compatibilidad supuso un cierto descontento por parte de los fabricantes de software que desarrollaban los navegadores.

Todo ello tuvo como consecuencia la creación de un grupo disidente, el WHATWG (*Web Hypertext Application Technology Working Group*). Este grupo de trabajo nació como respuesta a la lentitud del desarrollo de estándares por parte de W3C y al carácter demasiado cerrado de su proceso interno de elaboración de especificaciones. En sus orígenes estaba compuesto por representantes de firmas como Mozilla, Opera y Apple con sus conocidos navegadores Firefox, Opera y Safari. Google no tardó en unirse a ellos. Su posición era diametralmente opuesta a la de Xhtml 2.0, trabajando de forma pragmática sobre la base de las implementaciones actuales y, por tanto, sobre la base de Html4.0. Sus primeros esfuerzos se centraron en los Web Forms 2.0 para renovar los formularios y en el concepto de publicación en la web basado en Web Apps 1.0, es decir aplicaciones Web.

Paralelamente Xhtml 2.0, incluso tras un borrador (*working draft*) que aparece en julio de 2006, conoce un desarrollo particularmente laborioso y los rumores muestran numerosas disensiones en el seno de este grupo de trabajo. Por otro lado, los navegadores antes citados ignoraron deliberadamente a Xhtml 2.0.

Parecía claro que Xhtml 2.0 tenía graves problemas. Sir Tim Berners Lee, inventor de la Web y presidente del W3C, decide, a finales de 2006, reabrir un grupo de trabajo sobre Html que, de forma pragmática, retome la labor realizada por WHATWG. De este modo ha existido durante todo este tiempo un equipo de trabajo sobre Xhtml 2.0 y otro sobre Html5. Este último publicó el 23 de enero de 2008 un primer borrador de trabajo que después ha sufrido numerosas evoluciones.

Pero los días de Xhtml 2.0 estaban contados. A finales de 2009, el mismo Sir Tim Berners Lee anuncia la disolución del grupo de trabajo sobre Xhtml 2.0 y el abandono definitivo de este último.

Html5 pasó a *candidate recommendation* (recomendación candidata) en diciembre de 2012 y se debería finalizar en el último trimestre de 2014.

El anuncio de Html5 ha causado un gran revuelo (un buzz, en el vocabulario moderno). Este interés se ha concretado rápidamente a diversos niveles. Google ha incorporado de forma temprana, desde la versión 5 de Chrome, elementos de Html5. También lo han hecho sus compañeros en el WHATWG con las últimas versiones de Firefox, Safari y Opera. Incluso Microsoft, a menudo rezagado en materia de adopción de novedades y estándares, ha incorporado Html5 en las versiones 9 y 10 de Internet Explorer. Por su lado, los teléfonos smartphone también están adoptando el estándar Html5: Apple con el iPhone, RIM con BlackBerry, Google con Google Phone, etc. ¿Se trata de una evolución o de una revolución?

De forma paralela está en curso toda una reformulación de Html5 según las reglas y la sintaxis de XML, bajo el nombre de Xhtml5.

De toda esta agitada actividad en torno a la gestación de Html5 nos quedaremos con:

- Que ya no se hablará de páginas Web sino del concepto más amplio de aplicaciones Web.
- Que Html5 es una evolución (en efecto muy importante) de Html 4.0.
- Que Html5 ha sido concebido directamente por los navegadores más innovadores del mercado como Firefox, Safari Opera y Google. Esto garantiza una adopción rápida de los estándares de Html5.
- Que cabe esperar de Html5 grandes cambios en la elaboración de formularios (véase los *Web Forms 2.0*).
- Que, después de una década sin novedades, Html5 responde a la necesidad real de los diseñadores de renovar la interfaz de las aplicaciones Web.

2-HTML5 es un lenguaje de etiquetas

2.1. Etiquetas

Las etiquetas, también llamadas elementos, son comandos interpretados por los navegadores e informados entre los signos menor que (<) y mayor que (>). De este modo una etiqueta se escribe <etiqueta>.

Por regla general, a toda etiqueta de apertura le corresponde una etiqueta de cierre que marca el final del comando indicado por la etiqueta de apertura. La etiqueta de cierre retoma el mismo enunciado que la etiqueta de apertura, aunque precedida por una barra oblicua (/). De este modo a la etiqueta de apertura <etiqueta> le corresponde la etiqueta de cierre </etiqueta>. La sintaxis de una etiqueta es, por tanto:

<etiqueta> ... </etiqueta>

Para comprender el modo de funcionamiento de las etiquetas se propone el texto siguiente:

¡Es <negrita>importante</negrita> aprender el lenguaje <cursiva>Html5</cursiva>!

Esto puede interpretarse de la siguiente manera:

- Escribir "¡Es" de forma normal, puesto que no se ha especificado nada.
- A continuación escribir la palabra (únicamente esta palabra) "importante" y ponerla en negrita.
- Retomar la escritura normal para "aprender el lenguaje".
- Escribir esta vez la palabra "Html5" en cursiva.
- Y terminar por "!" en escritura normal.

Esto daría el resultado siguiente:

¡Es **importante** aprender el lenguaje *Html5*!

El lenguaje Html no es ni más ni menos que esto. Cada vez que se indica una instrucción al navegador, por ejemplo insertar un título, empezar a escribir una tabla o realizar un vínculo a otra página, se aplica una etiqueta de apertura. La etiqueta de cierre señala al navegador que la instrucción se ha terminado.

Existe no obstante una excepción, pues Html5 ha heredado de Html las etiquetas únicas, también llamadas etiquetas vacías, que no tienen etiqueta de cierre.

Por ejemplo, la etiqueta imagen .

Para asegurar la compatibilidad hacia atrás con Xhtml, estas etiquetas también pueden escribirse en Html5 con un signo de cierre.

De este modo, la etiqueta imagen también puede escribirse en Xhtml .

El espacio situado antes de la barra oblicua de cierre es importante por motivos de compatibilidad con los navegadores más antiguos.

Algo que nos limita en cierto modo, es el hecho de que las etiquetas hacen referencia a términos o abreviaturas de términos anglosajones volviéndolas (en un primer momento) abstractas y complejas.

Ejemplos

	b por <i>bold</i> que significa negrita
<i>	i por <i>italic</i> que significa cursiva
<p>	p por <i>paragraph</i> que significa párrafo
<div>	div por <i>division</i> que significa... división
<table>	<i>table</i> significa tabla
<form>	<i>form</i> significa formulario
	<i>img</i> para imagen

2.2. Atributos de etiqueta

En ocasiones es preciso completar una instrucción con especificaciones más precisas en uno u otro dominio. Para hacer esto, el lenguaje Html5 dispone de los atributos de etiqueta. El atributo se inserta en la etiqueta, entre la palabra de instrucción y el signo > final.

La sintaxis completa de una etiqueta con un atributo es: <etiqueta atributo="valor"> ...</etiqueta>

El atributo contiene siempre un valor, que se indica de forma complementaria al atributo mediante el signo igual (=) seguido del valor escrito entre comillas. El respeto estricto a la sintaxis exige que no exista espacio antes o después del signo igual.

Es posible usar varios atributos, separados por espacio, en una misma etiqueta: <etiqueta atributo1="valor" atributo2="valor"> ...</etiqueta>

3-Uso correcto de las etiquetas

He aquí algunas reglas simples que será preciso respetar a la hora de escribir etiquetas Html5.

- En Html5, las etiquetas no son sensibles a las mayúsculas o minúsculas. De este modo es indiferente escribir <ETIQUETA>, <Etiqueta> o <etiqueta>. Algunos verán en la escritura con mayúsculas un medio eficaz para distinguir el código Html del contenido del documento. No obstante, el uso generalizado escribe las etiquetas con minúsculas (como en Xhtml).
- La regla general exige que toda etiqueta abierta <etiqueta> se cierre </etiqueta>. La libertad que se adquirió en la escritura de Html 3.2 a causa del funcionamiento más o menos permisivo de los navegadores ya no está de actualidad. El rigor aportado por Html 4.0 estricto y Xhtml 1.0 debe seguir guiando su escritura.
- Las etiquetas deben enlazarse correctamente. Cuando se aplican varias etiquetas a un elemento, el orden de cierre de éstas es esencial. La primera etiqueta de cierre debe corresponderse con la última etiqueta de apertura que no haya sido cerrada. El siguiente ejemplo resultará más claro:

Es correcto: <a><c>contenido</c>

Es incorrecto: <a><c>contenido</c>

- Los valores de los atributos deben informarse siempre entre comillas. Aquí también sigue siendo necesario el rigor en el código.

4-Validación de código HTML5

Si bien Html5 es menos formalista que Html 4.0 estricto o que Xhtml 1.0, sigue siendo indispensable validar el código fuente.

Las ventajas de un código fuente perfecto y por tanto de la validación son:

- Comprensión y verificación en profundidad de Html5 por parte de programadores iniciados (y avanzados).
- Asegurar que la página se mostrará siempre de forma correcta en la mayoría de los navegadores. Una página inválida puede llevar a dos navegadores distintos a interpretarla de forma muy diferente.
- Demostración de su propia capacidad y profesionalidad a la hora de realizar un código de calidad, conforme a los estándares de Html5.
- Los programas de síntesis vocal y otras ayudas técnicas dirigidas a personas con problemas de accesibilidad se basan en un código válido a la hora de procesar las páginas de la red.

Si el documento, sometido a un validador, no es correcto, se le proveerá un análisis de los problemas que le permitirá corregir su código. Las primeras validaciones pueden ser muy frustrantes pero suponen un excelente aprendizaje del código fuente. No obstante el informe de errores es, con frecuencia, muy técnico.

Para realizar su función de verificación y validación, los validadores precisan que se defina el doctype del documento

Los validadores Html5 (en línea) disponibles son:

- El propio validador de W3C.
- El validador validator.nu.

5-Etiquetas y atributos HTML 4.0 desaparecidos

Las diferencias, tanto a nivel de etiquetas como de atributos, respecto a Html 4.0 son numerosas e importantes.

5.1. A nivel de etiquetas

Destacamos en primer lugar la desaparición por completo de los cuadros. Las etiquetas `<frame>`, `<frameset>` y `<noframes>` han desaparecido definitivamente de las herramientas a disposición de los diseñadores. Esto no es ninguna sorpresa puesto que estaban deprecados desde hace varios años. Los programadores profesionales los evitaban como la peste después de que Google anunciara a los Webmasters que no garantizaba una indexación correcta de los sitios web que incluyeran cuadros. La etiqueta `<iframe>` como tal subsiste en Html5.

También se produce la desaparición de ciertas etiquetas de presentación como `<big>`, `<center>`, ``, `<strike>`, `<tt>` y `<u>`. También en este caso no hay grandes sorpresas puesto que el principio de separación entre contenido y presentación ya está bien establecido en el presente. Estas etiquetas de presentación ahora se tienen en cuenta en las hojas de estilo CSS.

5.2. A nivel de atributos

Html5 apuesta firmemente por el principio de separación entre contenido y presentación. Lo que asombra es la cantidad de atributos afectados. En resumen, todos los atributos relativos a la alineación, la anchura, los fondos (con color o con imagen), los bordes y la numeración de listas desaparecen.

Todos estos atributos deben tenerse en cuenta en las hojas de estilo, que se convierten así en un elemento inseparable del código Html5.

De forma más detallada:

- align en las etiquetas `<caption>`, `<iframe>`, ``, `<input>`, `<object>`, `<legend>`, `<table>`, `<hr>`, `<div>`, `<hx>`, `<p>`, `<col>`, `<colgroup>`, `<tbody>`, `<td>`, `<tfoot>`, `<th>`, `<thead>` y `<tr>`.
- alink, link, text y vlink en la etiqueta `<body>`.
- background en la etiqueta `<body>`.
- bgcolor en las etiquetas `<table>`, `<tr>`, `<td>`, `<th>` y `<body>`.
- border en las etiquetas `<table>` y `<object>`.
- cellpadding y cellspacing en la etiqueta `<table>`.
- frameborder en la etiqueta `<iframe>`.
- height en las etiquetas `<td>` y `<th>`.
- hspace y vspace en las etiquetas `` y `<object>`.
- marginheight y marginwidth en la etiqueta `<iframe>`.
- noshade en la etiqueta `<hr>`.
- nowrap en las etiquetas `<td>` y `<th>`.
- rules en la etiqueta `<table>`.
- size en la etiqueta `<hr>`.
- type en las etiquetas ``, `` y ``.
- valign en las etiquetas `<col>`, `<colgroup>`, `<tbody>`, `<td>`, `<tfoot>`, `<th>`, `<thead>` y `<tr>`.
- width en las etiquetas `<hr>`, `<table>`, `<td>`, `<th>`, `<col>`, `<colgroup>` y `<pre>`.

De este modo, ya no es posible fijar, en Html5, los bordes, la alineación, los fondos o el tamaño de las tablas sin recurrir a hojas de estilo CSS.

Con la desaparición de los atributos color o bgcolor, Html5 se define en blanco y negro.

Esta ausencia completa de atributos de presentación nos obliga a revisar nuestra estructura de aprendizaje tradicional de Html en la que se estudian por un lado las etiquetas Html y por otro las hojas de estilo CSS. Ahora, ya desde nuestros primeros pasos con Html, nos veremos obligados a incluir nociones elementales de hojas de estilo para la presentación.

6-El documento HTML5 mínimo

6.1. El doctype y su importancia

Todo documento Html debe comenzar por un doctype. Html5 propone un doctype único y simplificado.

```
<!DOCTYPEhtml>
```

Para darse cuenta de la situación, basta con compararlo con un doctype de Html 4.0.

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

El doctype, también llamado DTD, sirve para indicar al navegador a qué reglas de escritura obedece el código fuente de la página Html o Xhtml. Lo usará para mostrar la página según los estándares de W3C. Todos los navegadores aplican la misma normativa, por lo que cabe esperar una visualización idéntica en los distintos navegadores.

En ausencia de un doctype el navegador no sabe según qué reglas debe procesar la página. Se conforma con acogerse a reglas genéricas para renderizar a toda costa la página Html. Esto puede suponer diferencias respecto a la visualización en los distintos navegadores. Este modo, incompleto, se llama modo de compatibilidad o *quirksmode*.

El doctype debe situarse en la primera línea del archivo Html. Si hubiera cualquier otra cosa, incluso un simple espacio o una línea en blanco, ciertos navegadores consideran que la página no tiene el doctype y la mostrará en *quirksmode*.

6.2. La etiqueta HTML

La etiqueta `<html>` indica al navegador que se trata de un documento Html.

La etiqueta `<html>` es el elemento más alto o el elemento raíz del documento. Se declara justo debajo de la declaración del doctype.

```
<!DOCTYPE html>  
<html lang="es">  
...  
</html>
```

La etiqueta `<html>` puede incluir el atributo `lang="es"` que especifica que el documento está en lengua española.

Esta información es muy valiosa para los motores de búsqueda como Google y para los programas de síntesis de voz usados por personas con dificultades visuales.

En la dirección www.w3schools.com/tags/ref_language_codes.asp podrá encontrar una lista completa de las demás lenguas referenciadas.

6.3. El encabezado del documento

La etiqueta `<head> ...</head>` que se posiciona justo después del doctype y de la etiqueta `<html>` contiene lo que llamamos el encabezado del documento.

En este encabezado del documento se sitúa toda una serie de información relativa al propio documento, independientemente de su contenido.

Esta información puede ser de naturaleza muy diversa:

- El título del documento. La etiqueta <title> ...</title> que se explica en el siguiente punto.
- Declaraciones o llamadas a hojas de estilo CSS.
- Funciones o vínculos a archivos JavaScript.
- Información dirigida a los motores de búsqueda para mejorar la indexación y el posicionamiento de la página.
- Información dirigida a los navegadores.
- Meta-etiquetas que retoman la codificación, la descripción de la página, las palabras clave asociadas a la página, el nombre del autor, menciones de copyright, etc.

Con intención de simplificar, Html5 especifica valores por defecto para el atributo tipo de estos scripts, estilos y elementos link. Salvo si necesita un valor diferente al definido por defecto, podrá omitir en Html5 este atributo *type*.

JavaScript es el lenguaje de script predefinido.

Así, en Html 4.0, era preciso escribir:

```
<script type="text/javascript">
...
</script>
```

Y

```
<script type="text/javascript" src="archivo.js"></script>
```

En Html5, basta con escribir:

```
<script>
...
</script>
```

Y

```
<script src="archivo"></script>
```

CSS es ahora el lenguaje predefinido para las hojas de estilo.

Así, en Html 4.0, era preciso escribir:

```
<style type="text/css">
...
</style>
```

Y

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

En Html5, basta con escribir:

```
<style>
...
</style>
```

Y

```
<link rel="stylesheet" href="archivo.css">
```

Nuestro documento Html5 mínimo queda así:

```
<!DOCTYPE html>
```

```
<html lang="es">
<head>
...
</head>
...
</html>
```

Dejando aparte el título de la página (véase a continuación), ningún elemento comprendido entre las etiquetas `<head> ...</head>` se muestra en el navegador.

6.4. El título del documento

La etiqueta `<title>`, incluso en el encabezado del documento, es la única etiqueta obligatoria del mismo.

La etiqueta `<title> ...</title>` le asigna un título a su página.

El título se visualiza en la barra de título, situada en la parte superior izquierda de cualquier navegador y/o en las pestañas abiertas por el mismo.

La etiqueta `<title>` tiene especial importancia en lo que respecta al algoritmo de posicionamiento de Google.

Velaremos por tener, para los documentos Web, un título atractivo y sintético. Ya no es inútil incluir una o varias palabras claves relativas a la página.

El documento Html5 se completa de la siguiente forma:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
</head>
...
</html>
```

Si accidentalmente hubiera un error en las etiquetas `<title> ...</title>`, se mostraría una simple página en blanco.

6.5. La codificación (charset) del documento

La noción del juego de caracteres usado (*charset*) es una noción esencial en el desarrollo de páginas Web.

El juego de caracteres determina la forma en que los caracteres de un alfabeto se convierten en bytes en un archivo informático (y viceversa). Ciertos métodos de codificación son específicos de un entorno informático en un idioma o alfabeto dados, otros son multiplataforma y multiidioma. Ejemplos de codificación: ASCII, ISO88591, ISO885915, UTF8, etc.

ASCII	Juego de caracteres básico, aunque relativamente antiguo, que permite el intercambio de textos en inglés a nivel mundial. Se escogió en el origen de la web pero como el inglés no tiene acentos, había que codificar los caracteres acentuados y especiales con entidades del tipo <code>&acute;</code> (para la <code>é</code>) o <code>&euro;</code> (para el <code>€</code>).
ISO88591	Extensión de ASCII, ISO88591, también llamado Latin1o Europa occidental, es la codificación que se utiliza con frecuencia en nuestra zona mundial. Añade numerosos caracteres latinos así como otros signos de puntuación o símbolos. Actualmente, su utilización tiende a decrecer en favor de Unicode

ISO885915	Versión actualizada de ISO88591 que incluye, entre otros, el símbolo €.
UTF8	UTF8 es un formato de codificación de caracteres (charset) creado para codificar el conjunto de caracteres internacionales de Unicode de diferentes idiomas, a diferencia de los formatos más antiguos como iso88591 que requiere una forma de codificación para aceptar, por ejemplo, letras acentuadas o caracteres especiales. Su principal ventaja para la web es que permite representar los miles de caracteres de los diferentes idiomas y permite crear sitios Web que puedan ser multilingües (español, chino o árabe...) sin tener que crear múltiples versiones del sitio según la codificación de cada país.

Desde finales de 2011, el W3C, siguiendo en esto a la IETF (*Internet Engineering Task Force*) que edita los estándares de Internet, recomienda el charset UTF8 para la edición de páginas Web en gran medida por su universalidad.

Los navegadores incluyen una función de detección automática del juego de caracteres. Aun así, dejar la codificaciónal criterio del navegador supone en ocasiones una puerta abierta a errores de interpretación. Por ello es importante definir el charset que se utiliza en el encabezado del documento.

```
<meta charset="UTF-8">
```

La siguiente sintaxis, que recuerda a Xhtml 1.0, también está aceptada:

```
<meta charset="UTF-8" />
```

De este modo no sólo se ha simplificado el doctype en Html5. Recordemos la forma que tomaba la declaración de la codificación en Html 4.0:

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

Nuestro documento Html5 mínimo es, ahora:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="iso-8859-1">
</head>
...
</html>
```

La especificación Html5 recomienda (lógicamente) a los autores el uso de UTF8. Pero nada impide mantener el formato iso88591 o iso885915 más europeo. No obstante se arriesga a parecer algo anticuado...

6.6. El cuerpo del documento

Las etiquetas <body> ...</body> definen lo que llamamos el cuerpo del documento. Es entre estas etiquetas donde se situará el código Html5 que se usará para elaborar el contenido de la página.

Esta parte del documento Html se visualizará en la ventana del navegador y por tanto la que se presentará al internauta.

En Html5, los atributos dedicados a la presentación del cuerpo del documento en la etiqueta <body> se abandonan dejando paso al uso de hojas de estilo.

Estos atributos que determinaban el color de fondo (bgcolor), la imagen de fondo (background) o el color del texto(text) ya no tienen lugar en Html5.

Para finalizar, nuestro documento Html5 mínimo tiene el siguiente aspecto:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
.....
</body>
</html>
```

CAPITULO 2: Primeros elementos de las hojas de estilo

1-Propiedades de estilo

Las hojas de estilo abarcan todo aquello relacionado con la presentación del documento Html como el color, la alineación, el tamaño, los bordes, los fondos o el interlineado.

La declaración de un estilo se hace mediante el binomio propiedad: valor;.

Ejemplo

text-align: center;

Lo que puede interpretarse como: "alinear el texto de forma centrada".

Detallemos esta notación:

- propiedad indica el elemento de presentación que se ve implicado; por ejemplo color para el color, text-align para la alineación, border para los bordes, font-size para el tamaño de la fuente.
- la propiedad está separada por su valor mediante dos puntos.
- valor especifica una palabra clave, un porcentaje o un tamaño en función de la propiedad a la que está asignado.
- el punto y coma final indica que se ha terminado la declaración de estilo, y es obligatorio.
- se permite el uso de espacios. Así, ciertos autores tienen la costumbre de incluir un espacio entre los dos puntos y el valor para mejorar la legibilidad del código.

2-Selectores

A continuación hay que determinar los elementos del código Html que se verán afectados por el efecto de presentación. Estos elementos se llaman selectores.

2.1. Selectores básicos

El primer selector que nos viene a la mente es la etiqueta Html. De este modo podemos por ejemplo "engancha" la declaración de estilo anterior (alinear el texto de forma centrada) a la etiqueta <h1>. La sintaxis es: selector{ propiedad: valor;}

Ejemplo

h1 { text-align: center;}

2.2. Comentarios

- Se usa simplemente la etiqueta, sin los signos menor que (<) y mayor que (>). Es decir, únicamente el texto de la etiqueta.
- La declaración de estilo se efectúa entre dos llaves.
- No hay que olvidar la llave de cierre, en caso contrario el navegador no tendrá en cuenta la declaración de estilo.

De este modo, todas las etiquetas <h1> se alinearán de forma centrada.

2.3. Selectores de clase

En ciertas situaciones, no queremos agregar un efecto de presentación a todas las etiquetas de un tipo determinado, sino a ciertas etiquetas elegidas libremente. Las hojas de estilo cuentan, para ello, con la noción de clase (*class*) que permite al diseñador definir sus propios selectores. De ahí el nombre de selector universal.

La etiqueta a la que se quiere aplicar la declaración de estilo se distinguirá de las demás por el atributo `class="nombre_clase"` siendo, por ejemplo: `<h1 class="efecto1"> ...</h1>`

La sintaxis es: `.nombre_clase { propiedad: valor; }`

El nombre de la clase, precedido de un punto y seguido de la declaración de estilo.

Ejemplo: `.efecto1 { text-align: center; }`

Aquí únicamente la etiqueta `<h1>` con el atributo `class="efecto1"` estará centrada. Además, esta etiqueta de presentación podrá aplicarse a otras etiquetas del código como, por ejemplo, un párrafo o una imagen.

Su concepto es muy cercano a los selectores de clase pero, aquí, no se desea aplicar el efecto del estilo más que a un único elemento de la página. Este elemento, siendo único, podrá tratarse como un objeto que se podrá manipular por JavaScript.

Este elemento único se identifica mediante el atributo `id="nombre_identificador"` siendo, por ejemplo, `<h2 id="titulo2"> ...</h2>`.

La sintaxis en este caso es: `#identificador { propiedad: valor; }`

Ejemplo: `#titulo2 { text-align: center; }`

El nombre del identificador, precedido por el signo de almohadilla (#) y seguido de la declaración de estilo.

3-Incorporación del estilo

Existen múltiples formas de incorporar las declaraciones de estilo en una página Html. En este punto del estudio de CSS vamos a explicar las elementales.

3.1. Estilo en línea

La declaración de estilo se agrega directamente a una etiqueta concreta del código Html mediante el atributo `style`.

Ejemplo

`<h1 style="text-align: center;"> ... </h1>`

Esta forma básica no respeta la regla de separación de contenido y presentación. Su uso será, por tanto, excepcional.

3.2. Estilo interno

Las declaraciones de estilo se agrupan en el encabezado del documento entre las etiquetas `<head> ...</head>`. La sintaxis es la siguiente:

```
<style type="text/css">
Declaraciones de estilo
</style>
```

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
h1 { color: red;}
.efecto1 { padding-left: 20px;
font-style: italic;}
</style>
</head>
<body>
<h1>Titulo 1 en rojo</h1>
<h2 class="efecto1">Subtítulo en cursiva</h2>
<h1>Título 2 en rojo</h1>
</body>
</html>
```

Titulo 1 en rojo

Subtítulo en cursiva

Título 2 en rojo

CAPITULO 3: El texto

1-Texto simple

Todo el contenido de su página Web se sitúa en el cuerpo del documento Html5, es decir entre las etiquetas<body> ...</body>.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
Un texto en      Html5
</body>
</html>
```

Un texto en Html5

Observe que en Html se ignoran los espacios múltiples. El navegador considera un único espacio.

2-Salto de línea

En Html, los retornos de carro o saltos de línea que aparecen en el código se ignoran y se sustituyen por un espacio.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
Html5 es el sucesor de
Html 4.0
</body>
</html>
```

Html5 es el sucesor de Html 4.0

Vemos que el texto Html ocupa una sola línea.

Para forzar el salto de línea es preciso usar la etiqueta
.

Nuestro ejemplo queda así:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
Html5 es el sucesor de<br>
Html 4.0
</body>
</html>
```

Html5 es el sucesor de
Html 4.0

- La etiqueta
 es una etiqueta llamada única pues no tiene etiqueta de cierre.
- La notación
, usada en Xhtml 1.0, también está aceptada en Html5.

3-Negrita

W3C anuncia la separación estricta entre contenido y presentación. Html se encarga solamente de la estructura del documento, y las hojas de estilo CSS se encargan de todo el aspecto relacionado con el formato visual. Puede parecer, no obstante, paradójico volver a encontrarse etiquetas de presentación. Esté tranquilo sabiendo que se trata únicamente de algunos formatos básicos del texto como la negrita, la cursiva, los exponentes y subíndices, etc.

Un estudio detallado de las hojas de estilo CSS le permitirá realizar mayores fantasías en la presentación.

Para escribir un texto en negrita (*bold* en inglés), basta con incluirlo entre las etiquetas

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<b>Texto en negrita</b>
</body>
</html>
```

Texto en negrita

De forma similar, la etiqueta ` ...` permite igualmente escribir el texto en negrita en los navegadores visuales.

La diferencia entre las etiquetas `` y `` es, no obstante, sensible en Html. La etiqueta `` provoca un efecto puramente tipográfico y visual. La etiqueta `` tiene como objetivo principal reforzar el texto. Si bien los navegadores visuales han adoptado la misma representación que para la negrita, la interpretación es muy distinta en los programas de síntesis vocal que leen las páginas Web a personas con deficiencias visuales. La etiqueta `` refuerza el contenido mediante una entonación de voz distinta o un nivel sonoro más elevado. El sentido que se da al texto con la etiqueta `` es por tanto muy diferente.

4-Cursiva

Para escribir un texto en cursiva (*italic* en inglés), hay que incluirlo entre las etiquetas `<i> ...</i>`.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<i>Texto en cursiva</i>
</body>
</html>
```

Texto en cursiva

- Es posible escribir un texto en negrita y en cursiva enlazando las etiquetas `` e `<i>`. No obstante sí importa el orden en que se enlazan estas etiquetas. Esto significa que el orden de cierre de las etiquetas ha de ser inverso al orden de apertura de las mismas. De este modo `<i>texto</i>` está enlazado correctamente. Por el contrario con `<i>texto</i>` las etiquetas están mezcladas y no se ha respetado la forma de enlazarlas.
- Existe también la etiqueta `` (*énfasis*). Esta etiqueta también escribe el texto en cursiva en los navegadores. La diferencia con la etiqueta `<i>` está en el sentido que se da a ambas etiquetas. La etiqueta `<i>` es puramente tipográfica y visual. La etiqueta `` tiene como objetivo poner de relieve el texto que la contiene. Este significado visual puede interpretarse de otro modo en los navegadores no visuales.

5-Exponentes y subíndices

En ocasiones es necesario escribir algunos caracteres como exponentes o subíndices, por ejemplo en fórmulas matemáticas o químicas.

La etiqueta ^{...} escribe el contenido en exponente y la etiqueta _{...} en subíndice.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
(a + b)<sup>2</sup> = a<sup>2</sup> + 2ab + b<sup>2</sup><br>
H<sub>2</sub>O
</body>
</html>
```

$$(a + b)^2 = a^2 + 2ab + b^2$$
$$H_2O$$

6-Otras etiquetas de texto

Hay otras etiquetas de texto cuyo uso es menos frecuente, incluso raro.

6.1. Texto tachado

La etiqueta ... permite marcar un elemento de texto como suprimido (*delete*) o caducado, por ejemplo en la actualización de un precio en un sitio comercial. El texto aparece tachado en pantalla.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
La etiqueta <del>Html 4.0</del>Html5 ...<br>
Llave USB al precio de <del>29.99</del> 24.99 €
</body>
</html>
```

La etiqueta <strike> de Html 4.0, que permitía tachar el texto, no se ha mantenido en Html5. La etiqueta , que presenta el texto como tachado, puede ser una alternativa.

6.2. Texto preformateado

La etiqueta `<pre> ...</pre>` permite visualizar el texto tal y como se ha codificado en el editor de texto. Los espacios, tabulaciones y retornos de carro se respetan tal cual en la pantalla. Si incluye texto, se visualizará con un tipo de letra de paso fijo.

Ejemplo

```
<body>
<pre>
El texto incluido en una etiqueta pre
se visualiza con un tipo de letra de paso fijo
y preserva los espacios
Así como los saltos de línea.
</pre>
</body>
```

6.3. Dirección del texto

Algunos sistemas de escritura, tales como los alfabetos árabe y hebreo, se escriben de derecha a izquierda al contrario que el sentido de escritura convencional de izquierda a derecha de las lenguas que usan el alfabeto latino (como por ejemplo el castellano). Html, lenguaje universal, debe tener en cuenta de algún modo esta especificación.

La etiqueta `<bdo> ...</bdo>` indica el sentido de visualización del texto (de izquierda a derecha o de derecha a izquierda).

Los atributos son:

- `dir="ltr"` (*left to right*) para el sentido de lectura de izquierda a derecha (por defecto).
- `dir="rtl"` (*right to left*) para el sentido de derecha a izquierda.

Ejemplo

```
<body>
<bdo dir="ltr">dabale arroz a la zorra el abad</bdo><br>
<bdo dir="rtl">dabale arroz a la zorra el abad</bdo>
</body>
```

7-Comentarios

En ocasiones es útil comentar el código Html (como cualquier otro código de programación), para facilitar la comprensión a la hora de hacer alguna modificación.

En Html5, igual que en Html 4.0, los comentarios están precedidos por la etiqueta `<!--` y se cierran con la etiqueta `-->`.

Todo lo que escribamos entre estas etiquetas no se visualizará en la pantalla del navegador. Destaquemos, no obstante, que estos comentarios sí están visibles cuando se consulta el código fuente de la página.

Añadiremos que los comentarios pueden ocupar varias líneas y pueden situarse de forma indistinta en el `<head> ...</head>` o en el `<body> ... </body>`.

Ejemplo

```
<body>
<b><i>Texto en negrita y cursiva</i></b>
<!-- comentario particular -->
</body>
```

8-Caracteres especiales

Existe toda una maraña de caracteres especiales que es preciso codificar, en ocasiones, de forma especial.

Pensemos en un primer momento en los caracteres que se usan en la codificación Html5. Por ejemplo, el signo menor que (<) que marca la apertura de una etiqueta. A la hora de leer este código el navegador puede encontrarse con alguna dificultad. El validador de W3C le advierte de esta ambigüedad a la hora de verificar su página.

Un poco como para las palabras reservadas en un lenguaje de programación, existe una lista de elementos a evitar en la codificación del texto. Estos elementos son, por ejemplo:

- El signo < que indica la apertura de una etiqueta.
- El signo > que marca el cierre de una etiqueta.
- El signo " usado en los atributos de la etiqueta.
- El signo & que marca el inicio de una referencia de carácter.

Los distintos signos deben codificarse usando las entidades siguientes:

<	<
>	>
"	"
&	&

Preste atención al punto y coma final, pues es obligatorio.

Por otro lado, existe toda una serie de caracteres que no existen en el teclado normal como el símbolo © de copyright, el símbolo ® de marca registrada. Hay también otros caracteres específicos de las matemáticas como el signo ∫ para las integrales. Aquellos lectores interesados podrán encontrar toda una larga lista de caracteres especiales usados en Html en la dirección <http://www.ascii.cl/es/codigoshtml.htm>

Ejemplo

```
<body>
La etiqueta <b> ... </b> escribe el texto en
<b>negrita</b>.<br>
&spades; &clubs; &hearts; &diamonds;
</body>
```

En principio, si el carácter existe en el juego de caracteres (*charset*) declarado en el documento Html5, no es preciso escribir los caracteres especiales mediante entidades Html.

Así, en las codificaciones iso88591, iso885915 y en UTF8, la é existe y no es necesario recurrir a la entidad ´. Esto mejora sin duda la legibilidad del código.

Por el contrario, con la codificación iso88591, el signo € no existe puesto que es demasiado antigua para soportar el carácter en su juego de caracteres. En este caso hay que pasar obligatoriamente por el uso de su entidad Html €. Remarquemos que el símbolo € existe en iso885915 y UTF8.

Para saber más acerca de este tema, consulte las indicaciones de W3C en la dirección www.lagrange.net/w3c/html4.01/charset.html.

9-Dar formato al texto con las hojas de estilo

Lo importante ahora es comprender el mecanismo de las hojas de estilo para realizar tareas básicas de presentación.

9.1. Negrita

Es posible escribir un texto en negrita con las hojas de estilo CSS sin tener que recurrir a la etiqueta ``.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
span { font-weight: bold;}
</style>
</head>
<body>
Texto en <span class="negrita">negrita</span>
</body>
</html>
```

La propiedad de estilo `font-weight` (textualmente, el tamaño o la importancia de la fuente) se define a `bold` (es decir en negrita). Esta propiedad CSS aplica a las etiquetas ``. La propiedad completa `span { font-weight:bold;}` escribirá todas las etiquetas `` del documento en negrita.

9.2.Cursiva

Es posible escribir un texto en cursiva con las hojas de estilo CSS sin tener que recurrir a la etiqueta `<i>`.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.cursiva { font-style: italic;}
</style>
</head>
<body>
Texto en <span class="cursiva">cursiva</span>
</body>
</html>
```

Aquí la propiedad de estilo `font-style` (el estilo de la fuente) se pone a `italic` (en cursiva). Esta propiedad CSS aplica a las etiquetas con un atributo de clase que hemos definido `class="cursiva"`. La propiedad completa `.cursiva { font-style: italic;}` escribirá todas las etiquetas del documento con el atributo `class="cursiva"` en cursiva.

9.3. Tamaño de letra

Con la eliminación de la etiqueta y de su atributo size, el tamaño de la fuente se define únicamente mediante la propiedad de estilo CSS font-size.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
#grande { font-size: 36px;}
</style>
</head>
<body>
<div id="grande">Texto en grande</div>
</body>
</html>
```

El tamaño de la fuente (font-size) se fija a 36 píxeles (36px). No aplicará más que a la etiqueta <div> identificada por el id="grande".

9.4. Color del texto

Con la eliminación del atributo color de la etiqueta , el color del texto se define mediante la propiedad de estilo CSS color.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
h1 {color: red;}
</style>
</head>
<body>
<h1>Título en rojo</h1>
</body>
</html>
```

Todas las etiquetas <h1> escribirán su texto en color (color) rojo (red).

9.5. Alineación

La alineación del texto se realiza mediante la propiedad de estilo CSS text-align.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
```

```
<style type="text/css">
#centrado { text-align: center;}
</style>
</head>
<body>
<div id="centrado">Texto alineado al centro</div>
</body>
</html>
```

La propiedad text-align indica que la alineación del texto es centrada (center).

9.6. Subrayado

En la red, el subrayado es la convención adoptada para señalar un enlace. Cualquier otro subrayado del texto no puede sino inducir a error a su visitante. Por ello debería evitarse, incluso prohibirse, a la hora de escribir código Html.

La eliminación de la etiqueta de subrayado del texto <s> es un buen ejemplo.

No obstante sigue siendo posible subrayar un texto mediante una propiedad de estilo.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.subrayado { text-decoration: underline;}
</style>
</head>
<body>
Texto <span class="subrayado">subrayado</span>
</body>
</html>
```

La propiedad de estilo indica que la decoración del texto (text-decoration) es un subrayado (underline).

CAPITULO 4: Estructura del documento

1-Títulos

¿Qué puede resultar más útil que prever distintos tamaños de título para estructurar el contenido? Basta con pensar en los títulos que identifican una gran sección o una tabla de contenidos.

El lenguaje Html5 proporciona seis tamaños de letra para los títulos. Cuando se sabe que título en inglés se dice *heading*, la creación de la etiqueta de título se hace evidente: <h por *heading* seguido de un número 1 (para el más grande), 2, 3, 4, 5, 6 (para el más pequeño) y el símbolo > de cierre. En adelante, usaremos la notación <hx> ...</hx> donde x es un nivel de 1 a 6.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<h1>Título de nivel 1</h1>
<h2>Título de nivel 2</h2>
<h3>Título de nivel 3</h3>
<h4>Título de nivel 4</h4>
<h5>Título de nivel 5</h5>
<h6>Título de nivel 6</h6>
</body>
</html>
```

- Esta etiqueta existe desde los comienzos de Html y es, por tanto, perfectamente compatible con todos los navegadores.
- Por defecto, con la etiqueta <hx> ...</hx>, el texto se escribe en negrita.
- También por defecto, como para todos los elementos de bloque también llamados elementos de división (véase la sección Divisiones de página del presente capítulo), la etiqueta incluye una línea vacía entre el contenido de la etiqueta y el resto del documento.
- Para obtener caracteres más grandes o más pequeños que los proporcionados por defecto, o bien un tipo de letra diferente al proporcionado por defecto por el navegador, es preciso usar una hoja de estilo CSS.
- Las etiquetas del título al principio de la página se tienen muy en cuenta en los motores de búsqueda como Google, e influyen de forma importante en la clasificación y recomendación de los sitios Web, en particular de aquellos que contienen las palabras clave adecuadas.
- El W3C insiste para que, en Html5, las etiquetas de título <hx> encuentren su objetivo de calidad a la hora de estructurar las páginas Web. El uso de las etiquetas de título debería reservarse a su función original. Es importante no desviarlas de su objetivo principal para, por ejemplo, escribir un texto en negrita o con un tamaño de letra determinado. Las hojas de estilo CSS están ahí para realizar estas tareas de presentación.
- El atributo align, disponible en Html 4.0, no se ha mantenido en Html 5. La alineación del párrafo se realiza mediante una propiedad de estilo CSS.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style>
```

```
#centrado { text-align: center;}
#derecha { text-align: right;}
</style>
</head>
<body>
<h3>Título de nivel 3</h3>
<h3 id="centrado">Título de nivel 3</h3>
<h3 id="derecha">Título de nivel 3</h3>
</body>
</html>
```

2-Párrafos

Un contenido de tipo texto gana en legibilidad y comprensión cuando está dividido en distintos párrafos.

Para definir un párrafo, se utiliza la etiqueta <p> ...</p>.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor.</p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor.</p>
</body>
</html>
```

- Esta etiqueta es perfectamente compatible con los navegadores antiguos y modernos.
- Por defecto, cada párrafo está precedido y seguido del espacio correspondiente a un salto de línea para marcar la separación con el contenido anterior y posterior. Este espacio puede modificarse mediante las hojas de estilo.
- El atributo align de Html 4.0 ha desaparecido en la especificación Html5. La alineación (izquierda, centrado, derecha y justificado) debe realizarse mediante una propiedad de estilo CSS.

3-Citas

Las citas se escriben por delante del contenido de tipo texto mediante las etiquetas <blockquote> ...</blockquote>.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
Séneca afirma:<br>
```



```
<blockquote>Admira a quien lo intenta,<br>
aunque fracase.</blockquote>
</body>
</html>
```

Séneca afirma:

Admira a quien lo intenta,
Aunque fracase.

- La cita se visualiza con un salto de línea antes y otro después, igual que con los párrafos.
- Se muestra, a su vez, con un ligero desplazamiento respecto al margen izquierdo. En ocasiones se usa la etiqueta `<blockquote>` para obtener este desplazamiento. Dé preferencia a la propiedad de estilo `margin-left` para conseguir este efecto.
- Existe también la etiqueta `<q> ...</q>` para las citas. Ésta no incluye espacio antes ni después, ni tampoco desplazamiento a la derecha. Se visualiza entre comillas.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<p>Séneca afirma:
<q>Admira a quien lo intenta,
aunque fracase.</q>
</p>
</body>
</html>
```

Séneca afirma: “Admira a quien lo intenta, aunque fracase”

4-Separadores horizontales

El trazo horizontal puede ser muy útil para definir un cambio en el contenido. La etiqueta <hr> realiza esta función.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
Capítulo 1
<hr>
Capítulo 2
</body>
</html>
```

- No existen problemas de compatibilidad con esta etiqueta pues es bien antigua.
- La etiqueta <hr> es una etiqueta única que no tiene etiqueta de cierre. Html5 admite también la notación <hr />, derivada de Xhtml 1.0.
- Por defecto, esta línea horizontal ocupa toda la longitud de la ventana del navegador.
- Los diseñadores prefieren, a menudo, reemplazar este trazo, muy básico en su aspecto estético, por una imagen gráficamente más elaborada.
- Todos los atributos align, noshade, size, width, deprecated (*deprecated*) en Html 4.0, ya no existen en Html5. Es preciso usar hojas de estilo para determinar la alineación, el tamaño y la longitud.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style>
hr { text-align: center;
height: 3px;
background-color: black;
width: 100px;}
</style>
</head>
<body>
Capítulo 1
<hr>
Capítulo 2
</body>
</html>
```

5-Listas

Las listas, muy conocidas por los usuarios de procesadores de texto, son una forma muy eficaz de estructurar un contenido de tipo texto.

5.1. Listas ordenadas

La etiqueta ... escribe una lista ordenada.

Se definen a continuación, en el interior de estas etiquetas, los elementos de la lista, mediante las etiquetas

La estructura general de una lista ordenada o numerada es:

```
<ol>
<li>Capítulo 1</li>
<li>Capítulo 2</li>
<li>Capítulo 3</li>
<li>Capítulo 4</li>
</ol>
```

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
Listas en Html5:
<ol>
<li>Listas ordenadas</li>
<li>Listas no ordenadas</li>
<li>Listas anidadas</li>
<li>Listas de definiciones</li>
<li>Menú de listas</li>
</ol>
</body>
</html>
```

Listas en Html5:

1. Listas ordenadas
2. Listas no ordenadas
3. Listas anidadas
4. Listas de definiciones
5. Menú de listas

- La etiqueta es un clásico de Html. Se reconoce perfectamente en todos los navegadores.
- El atributo type, que permitía reemplazar las cifras por letras (mayúsculas o minúsculas) o números romanos, ya no forma parte de Html5. Veremos en la parte dedicada a las hojas de estilo CSS cómo modificar la numeración.
- El atributo start permite empezar la numeración en una cifra distinta al 1. Este atributo, para los amantes de la precisión, que había sido deprecado (*deprecated*) en Html 4.0 se ha retomado en Html5.
- El nuevo atributo Html5 reversed permite realizar una numeración descendente (5, 4, 3, 2, 1) en vez de la numeración ascendente habitual (1, 2, 3, 4, 5).

5.2. Listas no ordenadas

Las listas no ordenadas, también llamadas listas no numeradas, se escriben mediante la etiqueta

Se definen a continuación, en el interior de esta etiqueta, los elementos de la lista mediante las etiquetas ` ...`.

La estructura general de una lista no ordenada es:

```
<ul>
<li>Capítulo 1</li>
<li>Capítulo 2</li>
<li>Capítulo 3</li>
<li>Capítulo 4</li>
</ul>
```

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
Listas en Html5:
<ul>
<li>Listas ordenadas</li>
<li>Listas no ordenadas</li>
<li>Listas anidadas</li>
<li>Listas de definiciones</li>
<li>Menú de listas</li>
</ul>
</body>
</html>
```

- La etiqueta `` es un clásico de Html. Se reconoce perfectamente en todos los navegadores.
- El atributo `type`, que permitía cambiar de algún modo el tipo de marca (redonda llena, redonda vacía, cuadrada), ya no existe en Html5. Veremos en la parte dedicada a las hojas de estilo CSS cómo modificar la forma de las marcas o cómo reemplazarlas por imágenes.
- La forma de las marcas (disco lleno o disco vacío) la determina el navegador.

5.3. Listas anidadas

Nada impide elaborar listas ordenadas y listas no ordenadas anidadas. La codificación es, simplemente, un poco más compleja.

Ejemplo

Una lista numerada en la que se anida una lista no ordenada.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<ol>
<li>Títulos</li>
<li>Listas en Html5
<ul>
<li>Listas ordenadas</li>
```

```

<li>Listas no ordenadas</li>
<li>Listas anidadas</li>
<li>Listas de definiciones</li>
<li>Menú de listas</li>
</ul></li>
<li>Tablas en Html5</li>
<li>Formularios en Html5</li>
</ol>
</body>
</html>

```

5.4. Listas de definiciones

Html proporciona un último tipo de lista, particularmente útil para presentar, por ejemplo, glosarios. Esta lista, llamada de definición, presenta una lista de términos, cada uno seguido de su descripción.

La lista se construye en primer lugar mediante la declaración de una lista de definición (*definition list*), es decir<dl> ...</dl>.

Entre estas etiquetas, se declara lo que se llama el término mediante la etiqueta <dt> ...</dt>, seguida de su descripción en la etiqueta <dd> ... </dd>.

La estructura general de una lista de definición es la siguiente:

```

<dl>
<dt>Término 1</dt>
  <dd>Definición 1</dd>
<dt>Término 2</dt>
  <dd>Definición 2</dd>
<dt>Término 3</dt>
  <dd>Definición 3</dd>
</dl>

```

Ejemplo

```

<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<p>Etiquetas de lista disponibles:</p>
<dl>
<dt>ol</dt>
<dd>Lista ordenada</dd>
<dt>ul</dt>
<dd>Lista no ordenada</dd>
<dt>dl</dt>
<dd>Lista de definiciones</dd>
</dl>
</body>
</html>

```

6-Divisiones de página

6.1. Elementos de bloque

Tal y como hemos adelantado en nuestro estudio de Html, nos parece oportuno hablar de la división en bloque y en línea de los elementos.

El navegador interpreta cada etiqueta Html y determina la visualización a través de las hojas de estilo CSS aplicadas por defecto.

Existen dos grandes grupos de etiquetas: las etiquetas de renderizado CSS en bloque (*block*) y las etiquetas de renderizado CSS en línea (*inline*).

Las etiquetas de renderizado CSS en bloque ocupan por defecto toda la longitud de la ventana del navegador. Se sitúan, así, unas debajo de las otras. Generalmente, el navegador inserta automáticamente un espacio encima y debajo del contenido de la etiqueta.

Este es el caso, por ejemplo, de la etiqueta de título `<h>` y de todas las etiquetas de este capítulo. Todas ellas son etiquetas de tipo bloque.

Una etiqueta de título o de párrafo ocupa toda la longitud de la página y está separada por un espacio más o menos importante.

El código `<p>Párrafo 1</p><p>Párrafo 2</p>` se visualiza en dos líneas separadas puesto que la etiqueta `<p>` es una etiqueta de tipo bloque. Cada párrafo va a ocupar una línea.

Los principales elementos de división de tipo bloque, que se detallarán a lo largo de nuestro estudio, son:

- Etiquetas de título `<h> ...</h>`.
- Etiquetas de párrafo `<p> ...</p>`.
- Etiquetas de cita `<blockquote> ...</blockquote>`.
- La etiqueta `<hr>` que inserta una línea horizontal.
- Las etiquetas ` ...` de las listas ordenadas.
- Las etiquetas ` ...` de las listas no numeradas.
- Las etiquetas `<dl> ...</dl>` de las listas de definiciones.
- Etiquetas de tabla `<table> ...</table>`.
- Etiquetas de declaración de formularios `<form> ...</form>`.
- La etiqueta `<div> ...</div>` concebida especialmente para introducir una división (div de división) sin tener que usar alguna de las etiquetas mencionadas antes. Esta etiqueta es particularmente útil para aplicar una declaración de hojas de estilo CSS.

6.2. Elementos en línea

A diferencia de los elementos de bloque, los elementos en línea se sitúan siempre uno a continuación del otro permaneciendo en el mismo flujo de texto.

De este modo, el código texto en `negrita` y en `<i>cursiva</i>` se escribe en una sola línea, sin interrumpir el flujo del texto: texto en **negrita** y en *cursiva*.

Las principales etiquetas en línea son:

- La etiqueta ` ...` para escribir un texto en negrita.
- La etiqueta `<i> ...</i>` para escribir un texto en cursiva.
- La etiqueta `
` de salto de línea.
- La etiqueta `<a>` para los enlaces.
- La etiqueta `` para las imágenes.
- Las etiquetas de campos de formulario `<input>`, `<select> ...</select>` y `<textarea> ... </textarea>`.
- La etiqueta `` para introducir una división en línea en el texto. Esta etiqueta es particularmente útil para aplicar una declaración de hojas de estilo CSS.

CAPITULO 5: Enlaces

5-Insertar un enlace

Los enlaces constituyen la esencia del lenguaje Html y de las páginas Web. La riqueza del hipertexto es la que permite tejer esta red gigantesca que compone la Web.

La etiqueta <a> ... inserta un enlace. Su sintaxis básica es:

```
<a href="destino_del_enlace">Texto del enlace</a>
```

El destino del enlace puede ser:

- Un lugar de la página en curso.
- Otra página del sitio Web.
- Algún lugar de otra página del sitio web.
- Una página de otro sitio existente en la Web.
- Una dirección de correo electrónico.
- Un archivo para descargar.

La etiqueta de enlace tiene varios atributos:

href

El atributo href define la dirección (*url*) del destino del enlace.

hreflang

Indica el idioma del documento de destino si es diferente al del documento de origen.

ping

Es nuevo en Html5. El atributo ping puede contener una lista de direcciones url (separadas por espacio) que reciben una notificación cuando el usuario sigue el enlace.

```
<a href="html5.htm" ping="http://www.misitioweb.com/data/visitas.php">Html5</a>
```

El navegador nos enviará una petición POST a la dirección url especificada en el atributo. Este nuevo atributo va a ser particularmente útil para elaborar las estadísticas de un sitio Web. En el caso anterior, nos permitirá conocer el número de visitantes de la sección dedicada a Html5.

rel

Es nuevo en Html5. Especifica la relación entre el documento de origen y el archivo destino del enlace.

Aquí los posibles valores son abundantes: alternate, archives, author, bookmark, contact, external, first, help, icon, index, last, license, next, nofollow, noreferrer, pingback, prefetch, prev, search, stylesheet, sidebar, tag, up.

Este atributo de momento está implementado muy parcialmente en los navegadores más recientes.

target

Especifica al navegador la forma de visualizar el destino del enlace. Puede ser en una nueva instancia o en una nueva pestaña del navegador (target="_blank"), en la misma ventana de la página de origen del enlace (target="_self") o en la misma ventana pero ocupando toda la ventana del navegador.

type

Indica al navegador el tipo MIME del destino si no se trata de un documento Html5, por ejemplo un archivo de sonido o una imagen.

2-Enlaces a otra página

Los primeros enlaces a la hora de comprender la publicación en la Web apuntan a otra página situada en el sitio Web (dirección relativa).

Para crear un sitio Web, se aconseja agrupar todos los archivos relativos al sitio en una única carpeta. Siéntase libre a la hora de crear subcarpetas en esta carpeta.

2.1. Enlaces a una página situada en la misma carpeta

La dirección relativa nos causará problemas en el caso de tener una arborescencia de archivos compleja.

Comencemos nuestro estudio por el caso más sencillo. Considere dos archivos (pagina1.htm y pagina2.htm) situados en la misma carpeta.

Insertemos en la página 1 un enlace hacia la página 2.

El código del enlace en la página 1 es:

```
<a href="pagina2.htm">Enlace hacia la página 2</a>
```

2.2. Enlaces a una página situada en otra carpeta

En los sitios de cierta importancia, no es posible alojar todos los archivos en una única carpeta. Para estructurar el sitio, no es raro tener en la misma carpeta de origen varias subcarpetas.

El diseño de la ruta de acceso al archivo puede volverse un poco desconcertante, pues el método de direccionamiento está inspirado en el adoptado por Unix, poco familiar para los diseñadores habituados a Windows.

Veamos la situación siguiente: la carpeta de origen correspondiente a la raíz contiene un archivo indice.htm, una subcarpeta carpeta1 con el archivo pagina1.htm y otra subcarpeta carpeta2 con el archivo pagina2.htm. Esta arborescencia nos permitirá abordar varios casos.

a. Enlace desde indice.htm hacia pagina1.htm

El código de indice.htm es:

```
<a href="carpeta1/pagina1.htm">Texto del enlace</a>
```

- El archivo indice.htm y la carpeta carpeta1 se sitúan en el mismo nivel de la arborescencia. Una vez en carpeta1, basta con "descender" hasta pagina1.htm.
- Observe que es preciso usar la barra oblicua / y no la barra inversa \ que encontramos en ocasiones en Windows.

b. Enlace desde pagina1.htm hacia pagina2.htm

El código en pagina1.htm es:

```
<a href="../carpeta2/pagina2.htm">Texto del enlace</a>
```

- Para ir desde pagina1.htm hacia pagina2.htm es preciso, en primer lugar, subir un nivel para salir de carpeta1.
- Para subir un nivel se utiliza la notación "../". Una vez hemos subido un nivel, hay que descender a carpeta2 y, una vez en carpeta2, descender para llegar a pagina2.htm.

c. Enlace desde pagina2.htm hacia indice.htm

El código en pagina2.htm es:

```
<a href="../../indice.htm">Texto del enlace</a>
```


- Para ir de pagina2.htm hacia indice.htm es preciso en primer lugar subir un nivel en la arborescencia para salir de carpeta2, de ahí la notación "../". Descendiendo llegamos directamente a indice.htm.

3-Enlaces al interior de una página

En una página de cierta importancia, generalmente de tipo texto, es interesante poder llevar al usuario a un lugar concreto del documento. Se utiliza la técnica de las anclas (*anchor*) o puntos de fijación.

Este enlace al interior de la página se realiza en dos tiempos:

- La declaración del ancla.
- El enlace hacia el ancla.

3.1. Declaración del ancla

Las anclas van a definir un lugar en la página, lo que permitirá después hacer un enlace hacia ella.

La sintaxis de creación de un ancla es:

```
<a id="nombre del ancla"></a>
```

Con la desaparición del atributo name en Html5, la declaración de anclaje se realiza mediante un identificador id. Este ya era el caso en Xhtml 1.0.

Observe que el ancla no se visualiza en el navegador.

3.2. Enlace hacia un ancla situada en la misma página

Tras haber definido el ancla, basta con efectuar un enlace hacia ella.

Para crear el enlace, basta con escribir en el atributo href el nombre del ancla directamente precedido del signo dealmohadilla (#).

```
<a href="#nombre del ancla">Texto del enlace</a>
```

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<a href="#capitulo2">Ir al capítulo 2</a>
<h3>Capítulo 1</h3>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam...</p>
<h3>Capítulo 2</h3>
<a id="capitulo2"></a>
<p>Contenido del capítulo 2</p>
</body>
</html>
```

3.3. Enlace hacia un ancla situada en otra página

El principio es el mismo aunque habrá que definir también la dirección (relativa o absoluta) de la página.

Ejemplo

A partir del archivo indice.htm, efectuamos un enlace hacia un ancla en el archivo pagina1.htm de la subcarpeta carpeta1.

Tras haber incluido el ancla, ``, en el archivo pagina1.htm, el enlace en el archivo indice.htm queda:

```
<a href="carpeta1/pagina1.htm#ancla">Texto del enlace</a>
```

4-Enlaces a otro sitio

Un enlace puede hacer referencia a páginas de otros sitios situados en otra dirección en la Web. El destino indicado en la etiqueta de enlace será la dirección completa del sitio o de la página. Hablamos de direccionamiento absoluto.

Por ejemplo:

```
<a href="http://www.w3.org/standards/index.htm">Enlace</a>
```

Precisemos que se trata de la dirección completa, con la mención explícita del protocolo http://. En efecto, con los navegadores recientes, esta mención se ha vuelto facultativa y la ignoran gran parte de los usuarios.

5-Enlace a una dirección email

Para agregar cierta componente de interacción, es interesante permitir a los visitantes de su sitio la posibilidad de contactar por correo electrónico.

El destino en la etiqueta de enlace será entonces una dirección de correo electrónico, precedida del protocolo de correo electrónico (*mail*), es decir mailto: (con dos puntos solamente).

```
<a href="mailto:ediciones@eni.com">El autor</a>
```

La activación del enlace abrirá una ventana de la aplicación de correo electrónico configurada por defecto en el puesto del visitante. Por ejemplo, Microsoft Outlook.

Si bien no se trata de código Html propiamente dicho, es posible predefinir el asunto del mensaje o incluir el envío de una copia a otro destinatario.

- Para predefinir el asunto del email:
` Ediciones ENI` donde el contenido de subject es el asunto (predefinido). En nuestro ejemplo es "Formación Html5".
- Para incluir el envío de una copia a otro destinatario:
` Ediciones ENI` donde el contenido de cc es la dirección de correo electrónico del destinatario de la copia.
- Para combinar las dos posibilidades:
`Ediciones ENI`

6-Enlaces para descargar un archivo

Para permitir la descarga de un archivo, basta con especificar el nombre del mismo (con su extensión). Este modo de proceder es válido siempre que el archivo Html y el archivo ofrecido para su descarga estén situados en la misma carpeta o subcarpeta.

Para un archivo "formacion.pdf" situado en la misma carpeta, el código es:

```
<a href="formacion.pdf">Versión en PDF</a>
```

Si en el ordenador del visitante no hay ninguna aplicación instalada para leer los archivos pdf (Adobe Acrobat Reader por ejemplo), el navegador abre una ventana que invita a descargar el archivo. Por el contrario, si la aplicación Adobe Acrobat Reader está presente en el puesto del visitante, el navegador abre la aplicación y muestra el archivo.

Pasará igual para el resto de tipos de archivo. Cuando no existe una aplicación definida por defecto para la extensión del archivo, el navegador descarga el archivo, tras haber mostrado la ventana que invita a la descarga.

7-Enlaces hacia una ventana específica

Las páginas de destino de un enlace pueden abrirse en ventanas específicas gracias al atributo target.

Veamos las distintas posibilidades:

- target="_self" (por defecto). La página de destino se abre en la misma ventana que la página de origen del enlace.
- target="_top". La página de destino del enlace se abre en la misma ventana pero ocupará la totalidad de la ventana mostrada. En este punto de nuestro estudio, self y top tienen un efecto equivalente.
- target="_blank". La página de destino del enlace se abre en una nueva instancia o en una nueva pestaña del navegador.

8-Color y subrayado en los enlaces

Como herencia de los comienzos de Html, los enlaces se visualizan, por defecto, subrayados y en color azul para el enlace no visitado, en color púrpura para el enlace visitado y en color rojo para el enlace activo.

El subrayado es una convención general de la Web para indicar los enlaces de hipertexto. Para no desconcertar al internauta no experimentado, trataremos de ceñirnos a esta norma dentro de lo posible. No obstante, desde un punto de vista estético, el subrayado automático en una página que contenga muchos enlaces puede afear la presentación de la página.

Es posible quitar el subrayado usando una hoja de estilo (text-decoration: none).

Del mismo modo, para uniformar el color de los enlaces, es posible usar la propiedad de estilo color: valor;.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<style type="text/css">
a { text-decoration: none;
```

```

color: black;}
</style>
<meta charset="UTF-8">
</head>
<body>
Ir a <a href="pagina1.htm">Página 1</a><br>
Ir a <a href="pagina2.htm">Página 2</a><br>
Ir a <a href="pagina3.htm">Página 3</a><br>
</body>
</html>

```

9-Descripción acerca del enlace

El atributo `title="texto"` (que no es específico a la etiqueta del enlace) permite mostrar una descripción explicativa sobre el enlace cuando se pasa por encima el cursor del ratón. Este pequeño "extra" en ergonomía se aprecia especialmente, en términos de accesibilidad, por las interfaces vocales que guían a los internautas invidentes.

Implementada inicialmente por Internet Explorer, esta descripción aparece en todos los navegadores de nuestro estudio.

```
<a href="pagina2.htm" title="Enlace al capítulo 2">Sigue</a>
```

10-Nuevas posibilidades con los enlaces

Los enlaces ya pueden contener cualquier elemento de tipo bloque como divisiones (`<div>`) y objetos de lista (``).

De este modo, el siguiente código es válido en Html5, mientras que no lo era en Html 4.0.

Ejemplo

```

<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<style>
div { width: 280px;
height: 100px;
border: 1px solid black;
padding-left: 3px;}
a { text-decoration: none;
color: black;}
</style>
<meta charset="UTF-8">
</head>
<body>
<p>Haga clic en el extracto para leer el texto completo.</p>
<a href="pagina2.htm">
<div>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam...</div>
</a>
</body>
</html>

```

CAPITULO 6: Tablas

El movimiento se inició con Html 4.0 y Xhtml 1.0, y se ha visto reforzado con Html5. Las tablas se reservan exclusivamente para la presentación de datos. Ya no se considera, en una aplicación Web moderna, el uso de tablas para realizar la disposición de la página. Estas tablas de presentación, en ocasiones anidadas dentro de otras tablas, eran el estándar de diseño de páginas Web en lo que podemos considerar como Web 1.0. La presencia de múltiples tablas recargaba el código fuente y hacía de su lectura algo casi imposible. Por otro lado, las tablas complejas complican en gran medida la lectura a las ayudas vocales usadas por personas invidentes. La disposición de la página se asegura, ahora, mediante las etiquetas <div>.

No hay nada nuevo en Html5 respecto a Html 4.0. Simplemente hay que destacar la desaparición de muchos atributos de presentación que ahora deben incluirse obligatoriamente en las hojas de estilo CSS.

Todas las etiquetas y todos los atributos de Html5 relativos a las tablas son perfectamente compatibles con los navegadores del mercado.

1-Creación de una tabla

Una tabla (<table>) se compone de filas <tr>. Estas filas contienen celdas <td>. La distribución de las distintas etiquetas para mostrar una tabla sigue la lógica que se ilustra en el siguiente ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<table>
<tr>
<td>1</td><td>2</td>
</tr>
<tr>
<td>3</td><td>4</td>
</tr>
</table>
</body>
</html>
```

La tabla en Html se construye así:

- Se declara la apertura de la tabla con la etiqueta <table>.
- Esta tabla se compone de una primera fila, introducida por la etiqueta <tr>.
- Esta fila se compone de una primera celda, abierta con la etiqueta <td>.
- Tenemos un 1 como contenido único de esta celda.
- Se cierra la primera celda </td>.
- Pasamos a la segunda celda <td>2</td>.
- Termina la primera fila con </tr>.
- Comienza la segunda fila con una nueva etiqueta <tr>.
- Esta fila contiene las celdas <td>3</td><td>4</td>.
- Termina la segunda fila con </tr>.
- Aparece por último la etiqueta de cierre de la tabla </table>.

- Esta etiqueta, básica en Html, es compatible con todos los navegadores.
- La etiqueta <table> debe cerrarse obligatoriamente. En caso de que nos olvidemos la etiqueta de cierre </table>, corremos el riesgo de que la tabla no muestre todo su contenido o de que no se muestre correctamente.

Pasemos a los atributos de la etiqueta <table>. Para nuestra sorpresa, ¡esta etiqueta ya no tiene atributos en Html5!

Html 4.0 proporcionaba los atributos width, border, cellpadding, cellspacing, frame y rules. Todos estos atributos han desaparecido en Html5 y deben incluirse en las hojas de estilo CSS.

Las etiquetas <tr> y <td> siguen esta aproximación con la desaparición de los atributos align y valign.

1.1. Ancho de la tabla

El ancho de la tabla se define mediante la propiedad de estilo CSS width (ancho).

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style>table { width: 250px;}
</style>
</head>
<body>
<table>
<tr>
<td>1</td><td>2</td>
</tr>
<tr>
<td>3</td><td>4</td>
</tr>
</table>
</body>
</html>
```

También es posible determinar la altura de la tabla mediante la propiedad de estilo height.

```
<style>
table { height: 150px;}
</style>
```

1.2. Bordes de la tabla

Los bordes de una tabla y otros elementos Html se introducen mediante la propiedad de estilo CSS border (que significa borde).

Para obtener los bordes clásicos de una tabla, la declaración es un poco particular.

Nuestro objetivo es agregar un borde de 1 píxel (1px), de color negro (black) con un trazo sólido (solid).

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
```

```

<head>
<title>Html5</title>
<meta charset="UTF-8">
<style>table { border: 1px solid black;
width: 250px;}
</style>
</head>
<body>
<table>
<tr>
<td>1</td><td>2</td>
</tr>
<tr>
<td>3</td><td>4</td>
</tr>
</table>
</body>
</html>

```

También hay que agregar el borde a las celdas <td> que componen la tabla (td).

```

<style>
table { border: 1px solid black;
width: 250px;}
td { border: 1px solid black;
width: 33%;}
</style>

```

Subsiste un espacio entre el borde de la tabla y los bordes de las celdas (vestigio del atributo cellpadding para los iniciados). Para reducir ambos bordes a uno solo, se usa la propiedad de estilo border-collapse.

```

<style>
table { border: 1px solid black;
border-collapse: collapse;
width: 250px;}
td { border: 1px solid black;}
</style>

```

1.3. Alineación de la tabla

La tabla está, en nuestro caso, alineada a la izquierda. Para obtener una alineación centrada, hay que usar un truco. El hecho de incluir un margen (margin) automático a izquierda (left) y derecha (right) de la tabla provoca la alineación centrada.

```

<style>
table { margin-left: auto;
margin-right: auto;
border: 1px solid black;
border-collapse: collapse;
width: 250px;}
td { border: 1px solid black;}
</style>

```

1.4. Alineación de las celdas de la tabla

El contenido de las celdas está, en nuestro caso, alineado a la izquierda. Será más estético centrarlo (center) en las celdas. Para ello se usa la propiedad de estilo text-align.

```
<style>
table { text-align: center;
border: 1px solid black;
border-collapse: collapse;
width: 250px;}
td { border: 1px solid black;}
</style>
```

- Con text-align: center aplicado a la tabla (<table>), todas las celdas de la tabla mostrarán el contenido centrado dentro de ellas.

2-Celdas de la tabla

Las celdas pueden contener cualquier elemento definido en Html, bien sea un texto, imágenes, un enlace, un fondo e incluso otra tabla.

En Html5 han desaparecido los atributos width, height, align, valign, abbr, axis y scope de Html 4.0.

2.1. Ancho de las celdas

Por defecto, el navegador adapta el ancho de las celdas según su contenido.

Habrá que recurrir a la propiedad de estilo width, pero aplicada en esta ocasión a la etiqueta <td>, para uniformar el ancho de las celdas.

```
<style>
table { border: 1px solid black;
border-collapse: collapse;
width: 250px;}
td { border: 1px solid black;
width: 33%;}
</style>
```

2.2. Alineación horizontal

El contenido de una celda puede alinearse a la izquierda, centrado o a la derecha. El resultado se obtiene mediante la propiedad de estilo text-align aplicada a una celda. La propiedad text-align puede tomar los valores left (izquierda), center (centrado) o right (derecha). El valor por defecto es left.

Ejemplo

```
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style>
table { border: 1px solid black;
border-collapse: collapse;
width: 290px;}
td { border: 1px solid black;
```



```
width: 33%;}
</style>
</head>
<body>
<table>
<tr>
<td style="text-align: left;">Producto 1</td>
<td style="text-align: center;">En stock</td>
<td style="text-align: right;">3.43 €</td>
</tr>
<tr>
<td style="text-align: left;">Producto 2</td>
<td style="text-align: center;">En pedido</td>
<td style="text-align: right;">6.72 €</td>
</tr>
</table>
</body>
</html>
```

Aquí, la propiedad de estilo se ha aplicado directamente en el código de la etiqueta `<td>` mediante `style="text-align:xxx;"` (estilo en línea).

2.3. Alineación vertical

Por defecto, la alineación vertical del contenido de una celda se realiza en la mitad de la celda. Esto puede provocar situaciones poco estéticas a la hora de elaborar su tabla.

Es prudente prever una alineación vertical hacia la parte superior mediante la propiedad de estilo `vertical-align:top` aplicada a las celdas `<td>`.

```
<style>
table { border: 1px solid black;
border-collapse: collapse;
width: 140px;}
td { border: 1px solid black;
vertical-align:top;}
</style>
```

2.4. Margen interior de las celdas

El contenido de la celda está pegado al borde de la misma. Esto no resulta muy estético y ensucia la legibilidad de los datos. La propiedad de estilo `padding` le permite agregar un espacio entre el contenido de la celda y el borde de la misma.

Es importante distinguir el margen exterior, llamado `margin`, del margen interior designado por el término `padding`.

```
<style>
table { text-align: center;
border: 1px solid black;
border-collapse: collapse;
width: 250px;}
td { border: 1px solid black;
padding: 15px;}
</style>
```

Destacamos el hecho de que la propiedad de estilo `padding` agrega espacio por todo alrededor del contenido de la celda. Es posible agregar solamente un espacio entre el borde izquierdo y el contenido mediante la propiedad `padding-left`.

```

<style>
table { text-align: center;
border: 1px solid black;
border-collapse: collapse;
width: 250px;}
td { border: 1px solid black;
padding-left: 5px;}
</style>

```

2.5. Color de fondo

Tras la desaparición del atributo bgcolor, es posible agregar un color de fondo mediante la propiedad de estilo background-color.

```

<style>
table { text-align: center;
border: 1px solid black;
border-collapse: collapse;
width: 250px;}
td { border: 1px solid black;}
.color { background-color: #99ccff;}
</style>

```

Esta propiedad de estilo de color de fondo background-color también puede aplicarse a las etiquetas <table> y <tr>.

3-Fusión de celdas

Es posible fusionar horizontalmente o verticalmente las celdas.

Partiendo de una tabla con dos filas y tres columnas:

3.1. Fusión de columnas

Para fusionar columnas, Html5 proporciona el atributo de celda colspan="x" donde x se corresponde con el número de columnas que se desea fusionar.

```

<table>
<tr>
<td colspan="3" style="text-align: center;">Título de la tabla</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</table>

```

3.2. Fusión de filas

Para fusionar filas, Html5 proporciona el atributo de celda rowspan="x" donde x se corresponde con el número de filas que se desea fusionar verticalmente.

```

<table>
<tr>
<td rowspan="2">1</td>

```

```

<td>2</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
</table>

```

- Se ha fusionado la primera columna de dos filas.

4-Celdas de encabezado

Las celdas de encabezado de columnas o de filas se definen mediante la etiqueta `<th> ...</th>`. Estas celdas proporcionan, de algún modo, título a los datos de la columna o de la fila.

En los navegadores visuales, el contenido de la etiqueta `<th>` aparece en gris y centrado. Puede ser por este motivo que los diseñadores las dejan de lado. No obstante forma parte, históricamente, de las etiquetas de codificación de tablas en Html. Destaquemos que su apariencia siempre puede modificarse mediante las hojas de estilo CSS.

La etiqueta `<th>` se ha mantenido en la especificación de Html5 pues resulta de gran utilidad para los lectores de pantalla usados por personas invidentes. Contribuye así a la accesibilidad de los sitios web y, particularmente, de las tablas que, cuando tienen cierta amplitud, se convierten en un verdadero problema a la hora de describirlas auditivamente.

```

<table>
<tr>
<th>Atributo</th>
<th>Función</th>
</tr>
<tr>
<td><i>colspan</i></td>
<td>Para fusionar columnas</td>
</tr>
<tr>
<td><i>rowspan</i></td>
<td>Para fusionar filas</td>
</tr>
</table>

```

5-Leyenda de la tabla

La etiqueta `<caption>` permite asociar un título o una leyenda a la tabla.

La etiqueta `<caption>` debe situarse justo después de la etiqueta de apertura `<table>` y debe aparecer una sola vez en la tabla.

Por defecto, la mayoría de navegadores visualizan el contenido de la etiqueta `<caption>` de forma centrada y sobre la tabla. Es posible situar esta leyenda debajo de la tabla con ayuda de la propiedad de estilo CSS `caption {caption-side:bottom;}`.

6-Agrupar columnas

La etiqueta <colgroup> se usa para agrupar columnas con el objetivo de aplicar el mismo formato al conjunto de etiquetas del grupo mediante una hoja de estilo CSS.

El atributo span="x" determina el número de columnas así agrupadas. Por defecto, el valor de x es 1. El atributo span es el único atributo de la etiqueta <colgroup>. Los atributos align, valign, width de Html 4.0 han desaparecido.

Por otro lado, la etiqueta <colgroup> se posiciona justo a continuación de la etiqueta <table> (o justo a continuación de la etiqueta <caption> si se está usando) y delante de cualquier etiqueta <tr> o <td>.

Por definición, la etiqueta <colgroup> solamente se aplica para las tablas.

Ejemplo 1

He aquí una tabla de tres filas y tres columnas. Se aplica un color de fondo y una alineación centrada a las dos primeras columnas.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style>
table { border: 1px solid black;
border-collapse: collapse;
width: 250px;}
td { border: 1px solid black;}
colgroup { background-color : #99ccff;
text-align: center;}
</style>
</head>
<body>
<table>
<colgroup span="2"></colgroup>
<tr>
<td>1</td><td>2</td><td>3</td>
</tr>
<tr>
<td>4</td><td>5</td><td>6</td>
</tr>
<tr>
<td>7</td><td>8</td><td>9</td>
</tr>
</table>
</body>
</html>
```

1	2	3
4	5	6
7	8	9

Ejemplo 2

Si deseamos agrupar las dos últimas columnas, es preciso declarar varias etiquetas <colgroup>.

```
<!DOCTYPE html>
```

```

<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style>
table { border: 1px solid black;
border-collapse: collapse;
width: 250px;}
td { border: 1px solid black;}
#columnas { background-color : #99ccff;
text-align: center;}
</style>
</head>
<body>
<table>
<colgroup span="1"></colgroup>
<colgroup id="columnas" span="2"></colgroup>
<tr>
<td>1</td><td>2</td><td>3</td>
</tr>
<tr>
<td>4</td><td>5</td><td>6</td>
</tr>
<tr>
<td>7</td><td>8</td><td>9</td>
</tr>
</table>
</body>
</html>

```

1	2	3
4	5	6
7	8	9

7-Estructuración de la tabla

Html5 proporciona etiquetas para estructurar de forma lógica el contenido de una tabla.

Estas etiquetas son:

- <thead> para agrupar la información referente al encabezado de la tabla, como por ejemplo el título y el subtítulo de las columnas.
- <tbody> para el cuerpo de la tabla, es decir el conjunto de datos de la misma.
- <tfoot> para el contenido del pie de página, como por ejemplo anotaciones o una leyenda.

Esta división lógica no afecta en nada a la presentación por defecto de la tabla, aunque puede retomarse en las propiedades de estilo para adornar la presentación de la tabla.

Destaquemos que la etiqueta <tfoot> debe situarse antes de la etiqueta <tbody> de modo que el navegador pueda prever el pie de página antes de la recepción de las filas con los datos.

CAPITULO 7: Imágenes y fondos

1-Formatos de imagen

Para reducir al máximo el tiempo de descarga, la Web exige formatos de archivo particulares: los formatos GIF, JPEG y PNG. Estos tres tipos de formato implican el uso de un algoritmo de compresión.

1.1.El formato GIF

El formato GIF (de *Graphics Interchange Format*) vivió su momento de gloria con los inicios de la Web. Entonces estaba perfectamente adaptado a las tarjetas gráficas del momento (a menudo con un máximo de 256 colores) y a la pobre capacidad de la línea telefónica analógica (con una elevada tasa de compresión).

El formato GIF sólo permite codificar imágenes con un máximo de 256 colores. Esto ya no resulta práctico con los desarrollos y posibilidades de las imágenes digitales actuales.

- Es un formato de compresión eficaz y rápido. Las imágenes GIF tienen un peso muy reducido en número de bytes.
- La compresión se realiza sin pérdida de datos y sin pérdida de calidad.
- La función de entrelazado permite una visualización rápida de estas imágenes, que hace que se vuelvan progresivamente más nítidas.
- Existe la posibilidad, según la especificación GIF89a, de definir uno de los 256 colores como transparente. A la hora de visualizar esta imagen aparecerán los elementos que estén situados por detrás, allí donde la imagen sea transparente.
- Las imágenes son fácilmente editables y modificables dado el número limitado de colores.
- Es el único formato que permite construir animaciones, en forma de pequeños dibujos animados, según lo que se conoce como GIFs animados.
- El formato GIF está patentado por UNISYS.

Debido a su límite de 256 colores, el formato GIF no es una buena aproximación para los gráficos de alta resolución como las fotos, las imágenes sombreadas y los degradados. Por el contrario, es excelente para la elaboración de logotipos, botones, puntos, barras, símbolos, y otros elementos de clipart frecuentes en ciertos gráficos de la Web que no precisan de una paleta de colores muy extensa. Está siendo, no obstante, sustituido por el formato PNG.

1.2.El formato JPEG

El formato JPEG significa *Joint Photographic Expert Group*, no por el nombre del formato gráfico sino por el nombre de la compañía que ha desarrollado este formato. Es el tipo de formato más extendido en el universo de la fotografía digital.

- El formato JPEG permite salvar hasta 16,7 millones de colores por imagen.
- Su tasa de compresión es eficaz en lo que respecta al tamaño de la imagen resultante.
- Permite modificar la tasa de compresión según las necesidades del diseñador. Una compresión de entre el 15% y el 20% generará una imagen de calidad satisfactoria con un tamaño de archivo reducido.
- La otra cara de la moneda es que existe cierta pérdida de calidad en función de la tasa de compresión. A mayor factor de compresión, mayor pérdida de calidad de la imagen. Es, por tanto, una compresión destructiva.
- Se producen deformaciones (*artifacts* o artefactos), a menudo en los degradados de colores y en las esquinas y los bordes.
- Después de varios años se ha implementado la función de entrelazado, permitiendo una visualización progresiva y más rápida en la Web.
- El formato JPEG no ofrece la posibilidad de elaborar transparencias ni animaciones.

JPEG es un formato excelente para las fotografías, especialmente aquellas donde existen tonalidades diversas y degradados de color. No obstante, no está exenta de defectos, con sus consiguientes pérdidas de calidad y sus deformaciones.

A causa de estas pérdidas de datos y de las deformaciones, es imprescindible respetar la regla de guardar la imagen original y trabajar únicamente sobre copias de la misma.

1.3.El formato PNG

El formato PNG, de *Portable Network Graphic*, se presenta como el formato del futuro. Se trata de un formato digital concebido especialmente para el uso de imágenes en páginas Web. La recomendación de W3C respecto a PNG se remonta a 1996. PNG debería, en teoría, reunir todas las ventajas de GIF y JPEG. En la actualidad, está empezando a ser considerado seriamente por parte de los diseñadores de aplicaciones Web.

- El formato PNG soporta 16,7 millones de colores, igual que el formato JPEG.
- El formato PNG es una especificación concebida especialmente para Internet y es objeto de una recomendación por parte de W3C.
- Su compresión tiene muy buen rendimiento. La tasa de compresión del formato PNG es de entre el 5% y el 25%, superior a la del formato GIF.
- Esta compresión se realiza sin pérdida de datos ni de calidad.
- Permite realizar la transparencia de hasta 256 colores.
- El formato PNG es un formato abierto y no está patentado.
- Su función de entrelazado permite una visualización progresiva.
- No es capaz de generar imágenes animadas.

El formato PNG está llamado a ser el más utilizado en los próximos años, pues representa un excelente compromiso entre los formatos GIF y JPEG. Los profesionales lo están adoptando mayoritariamente en lugar del formato GIF para fotografías de tamaño pequeño.

1.4.El formato WebP

WebP es un nuevo formato de imagen que seguramente gane importancia en los próximos años. Este formato está desarrollado por Google a partir de uno de los formatos de la nueva etiqueta <video> de Html5 (el WebM). Según los primeros ensayos, WebP reducirá el tamaño de los archivos en un 39% respecto a los formatos JPEG, PNG y GIF, con una pérdida de calidad imperceptible.

2-Insertar una imagen

Es posible insertar imágenes mediante la etiqueta:

```

```

La etiqueta es una etiqueta única, no tiene etiqueta de cierre. La notación Xhtml 1.0 también está aceptada en Html5.

En Html5, la etiqueta imagen sólo tiene un atributo obligatorio: el atributo src="ruta del archivo de la imagen". En Html, la imagen no se inserta directamente en el documento sino que proviene de un archivo externo, designado por el atributo src, que precisa la dirección relativa del archivo de la imagen que tiene que mostrarse en la página.

De este modo, para una imagen situada en la misma carpeta que la página Html (ruta relativa), se escribe:

```

```

o

```

```

Para referenciar una imagen de un sitio Web (ruta absoluta), se escribe:

```
  
o  
<img src=http://www.html5.com/images/imagen.png />
```

Las reglas de direccionamiento son idénticas a las que se abordaron en el capítulo Enlaces.

Los atributos opcionales de la etiqueta imagen son:

- Los atributos height y width. Estos atributos permiten definir respectivamente, en píxeles, el alto y el ancho de la imagen en pantalla. La definición de estos atributos hace que el navegador reserve un lugar para la imagen antes de descargarla completamente, permitiéndole así continuar con la composición de la página y la visualización del texto de la misma. Definiendo las dimensiones de la imagen se acelera la carga de la página y permite ir visualizando algún contenido a los internautas impacientes. Si bien son opcionales, estos atributos están considerados como esenciales por parte de los diseñadores profesionales. Si el archivo no está disponible o si la ruta es incorrecta, se muestra un icono en lugar de la imagen deseada.
- El atributo alt="texto alternativo", que contiene una breve descripción de la imagen. Este texto está pensado principalmente para las personas invidentes, pues será leído por las interfaces vocales. Más recientemente, el contenido del atributo alt se usa en los motores de búsqueda (Google por ejemplo) para indexar y clasificar el contenido de un sitio Web y alimentar su base de datos de imágenes. Obligatorio en Html 4.0 y en Xhtml 1.0, se ha vuelto opcional en Html5. Este carácter opcional alimenta muchas reacciones, discusiones y controversias en los foros especializados.

Los atributos de presentación align, border, hspace y vspace no se han mantenido en Html5. Será preciso usar hojas de estilo CSS para reemplazar estos atributos desaparecidos.

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

3-Insertar un enlace sobre una imagen

Los enlaces desde una imagen se realizan simplemente insertándola entre las etiquetas de enlace <a>

Algunos navegadores continúan dibujando un borde alrededor de la imagen para señalar el enlace.

Para quitar este borde, es preciso utilizar una declaración de estilo. Antes de realizar un estudio detallado de las hojas de estilo, basta con agregar a la etiqueta la declaración style="border: none;".

```
<body>  
<a href=globo.htm>  
  
</a>  
</body>
```


4-Insertar un color de fondo

Tras la desaparición del atributo bgcolor, ya no es posible agregar un fondo de color a un elemento. Es preciso usar de forma obligatoria propiedades de estilo CSS, background-color en este caso.

```
<style>
body {background-color: rgb(200,215,230);}
</style>
```

5-Insertar una imagen de fondo

El atributo background se ha declarado obsoleto en Html5, por lo que ya no existe otra solución que utilizar la propiedad de estilo background-image para insertar una imagen de fondo.

```
<style>
body {background-image: url('bg.gif');}
</style>
```

CAPITULO 8: Etiquetas semánticas y de organización

A menudo es difícil percibir el porqué de estas etiquetas semánticas. Sobre todo porque su implementación no se corresponde en nada con la definición académica de la semántica como relación entre el significado y el significante.

Estas etiquetas semánticas describen la naturaleza de ciertos elementos del contenido de la página. Estos últimos pueden, de este modo, localizarse y procesarse más fácilmente mediante programas autómatas como, por ejemplo, los motores de búsqueda.

Ejemplo

La nueva etiqueta semántica de organización de la página `<nav> ... </nav>` simplemente indica que su contenido es un menú de navegación.

```
<nav>
<a href="#">Inicio</a> |
<a href="#.htm">Punto 1</a> |
<a href="#.htm">Punto 2</a> |
<a href="#.htm">Punto 3</a> |
<a href="#.htm">Contacto</a>
</nav>
```

Por tanto es aún más complicado comprender la utilidad de estas etiquetas Html si no tienen ninguna influencia sobre la estructura del contenido y, a menudo, ninguna influencia en la presentación y visualización de la página.

Este concepto de la Web semántica es una idea maestra de Tim Berners Lee (el inventor de la Web), quien la defiende con convicción desde hace muchos años.

1-Las antiguas etiquetas semánticas

Html5 hereda las antiguas etiquetas semánticas de Html 4.0. Estas etiquetas tienen un formato específico.

Estas etiquetas semánticas son:

- `<abbr> ... </abbr>` que indica una abreviatura, por ejemplo SA o IVA. Además de su función semántica, esta etiqueta sirve para indicar a los programas de síntesis vocal que no traten de leer la palabra tal y como está escrita, sino deletreándola.
- `<acronym> ... </acronym>` que indica un acrónimo, es decir una palabra formada por sílabas de varias palabras como ONU (Organización de las Naciones Unidas), ya no existe en Html5. Usaremos en adelante la etiqueta `<abbr>`.
- `<address> ... </address>` que indica una dirección de contacto. Su contenido se visualiza por defecto en cursiva y con letra más pequeña. Precisemos que no se realiza ningún enlace hacia la dirección mediante esta etiqueta.
- `<cite> ... </cite>` que indica una cita. Por defecto, esta cita se visualiza en cursiva.
- `<code> ... </code>` que señala una sintaxis o un código informático. Por defecto, su contenido se visualiza con un tipo de letra de paso fijo.
- `<samp> ... </samp>` que pone de relieve un texto de ejemplo. Por defecto, su contenido se visualiza con un tipo de letra de paso fijo.
- `<dfn> ... </dfn>` que indica la definición de un término. Por defecto, esta definición se visualiza en cursiva.
- `<kbd> ... </kbd>` que indica al usuario las letras del teclado que tiene que pulsar. Por defecto, su contenido se visualiza con un tipo de letra de paso fijo.
- `<var> ... </var>` que contiene una variable. Por defecto, el texto de la variable se representa con letra cursiva.
- ` ... ` define un texto importante. Por defecto, su contenido se visualiza en negrita.

- ` ... ` indica un texto sobre el que se pretende insistir o hacer énfasis. Por defecto, su contenido se visualiza en cursiva.

2-Las nuevas etiquetas de organización

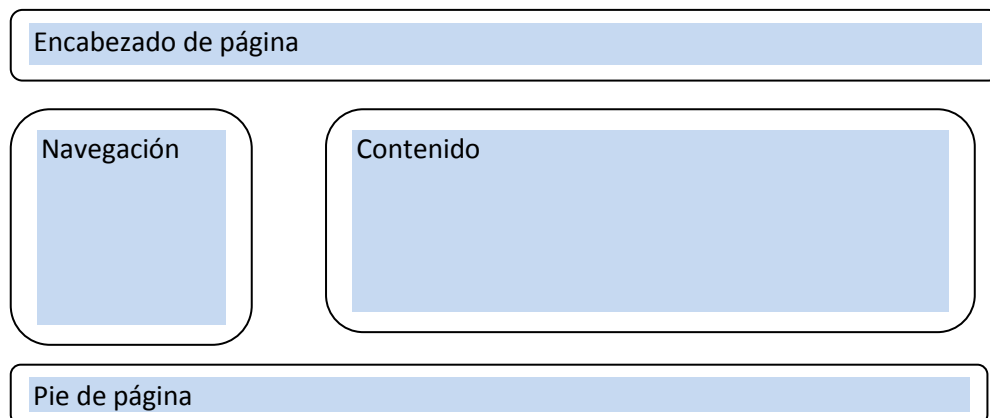
Estas etiquetas de organización están destinadas principalmente al diseño de la página con el objetivo de poder distinguir con mayor facilidad las secciones de código.

2.1. Etiquetas `<header>`, `<nav>`, `<footer>` y `<aside>`

Si hiciéramos una síntesis de las páginas Web existentes en la red, verificaríamos que todas ellas comparten, la mayoría de las veces, en totalidad o en parte, los siguientes elementos:

- Un encabezado de página con, por ejemplo, un logotipo, un banner, el nombre del sitio, un eslogan o una caja de búsqueda.
- Herramientas de navegación, útiles e indispensables para consultar las diferentes secciones del sitio.
- Una zona dedicada al contenido.
- Una zona anexa que permite agregar aquellos elementos accesorios al contenido propiamente dicho, como, por ejemplo, la publicidad.
- Un pie de página con la mención, por ejemplo, del copyright, el mapa del sitio, indicaciones legales, reglas de accesibilidad, etc.

Elementos que, de modo visual, pueden tomar distintas formas como por ejemplo:



La idea de Html5 consiste en crear nuevas etiquetas para poder identificar de manera más clara y más rápidamente los grandes bloques en el diseño de la página. De este modo:

- `<header> ... </header>` agrupa los elementos del encabezado de página. Esta etiqueta puede reemplazar, de forma natural, los `<div id="header">` que encontramos con frecuencia.
- `<nav> ... </nav>` indica los elementos de un menú de navegación.
- `<footer> ... </footer>` señala los elementos del pie de página. Esta etiqueta puede reemplazar, de forma natural, los `<div id="footer">` que encontramos con frecuencia.
- `<aside> ... </aside>` indica que se trata de elementos anexos al contenido.

Tomemos el siguiente ejemplo de código:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
```

```

<meta charset="UTF-8">
</head>
<body>

<h1>Nombre del sitio Web</h1>
<form action="http://www.google.com/search">
<input type="text" size="15" value="">
<input type="submit" value="Buscar">
</form>
<div>
<a href="index.htm">Inicio</a> |
<a href="item1.htm">Item 1</a> |
<a href="item2.htm">Item 2</a> |
<a href="item3.htm">Item 3</a> |
<a href="item4.htm">Item 4</a> |
<a href="contacto.htm">Contacto</a>
</div>
<h2>Nombre de la página</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam.</p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam.</p>
<hr>
<p>Copyright 201x</p>
</body>
</html>

```

Se trata de una página bastante corriente con:

- Un logotipo.
- El nombre del sitio Web.
- Un formulario de búsqueda.
- Enlaces de navegación dentro del sitio.
- El título de la página.
- Contenido.
- Una línea horizontal.
- Una mención de copyright.

Se trata de un código bastante corriente, que es válido en Html5, aunque no incorpora ningún elemento que permita organizar la página y distinguir las principales secciones del código relativas al encabezado, los menús de navegación y el pie de página. Esta información la aportan las nuevas etiquetas Html5 <header>, <nav> y <footer>.

Con las nuevas etiquetas de organización de Html5, este código podría presentar el aspecto siguiente:

```

<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<b><header>

<h1>Nombre del sitio Web</h1>
<form action="http://www.google.com/search">
<input type="text" size="15" value="">
<input type="submit" value="Buscar">
</form>
</header>
<b><nav>
<a href="index.htm">Inicio</a> |
<a href="item1.htm">Item 1</a> |
<a href="item2.htm">Item 2</a> |

```

```

<a href="item3.htm">Item 3</a> |
<a href="item4.htm">Item 4</a> |
<a href="contacto.htm">Contacto</a>
</nav>
<h2>Nombre de la página</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam.</p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam.</p>
<footer>
<hr>
<p>Copyright 201x</p>
</footer>
</body>
</html>

```

El código resulta mucho más legible, pudiendo diferenciar de un vistazo aquellos elementos de código relativos al encabezado de página, a las herramientas de navegación y al pie de página.

Esta es la única función que tienen las nuevas etiquetas semánticas en Html5. No cabe buscar en ellas funciones para dar formato a la página o de presentación. No sirven más que para organizar la página y mejorar la lectura del código por parte de los diseñadores y programadores.

Queda al criterio del diseñador la posibilidad de asociar hojas de estilo CSS a estas etiquetas para modificar la presentación.

2.2. Etiquetas <section> y <article>

El contenido principal de la página puede organizarse en distintas partes:

- La etiqueta <section> indica que una parte del contenido de la página se refiere a un tema concreto.
- La etiqueta <article> define un contenido del documento que posee una identidad independiente dentro de la página, como puede ser el artículo de un blog, un post en un foro o un producto en un sitio comercial.

Tomemos como ejemplo un blog con distintos temas que se abordan de forma diaria.

```

<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<h1>Día 1</h1>
<h2>Artículo 1</h2>
<div>Presentación del artículo 1</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam.</p>
<h2>Artículo 2</h2>
<div>Presentación del artículo 2</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam.</p>
<hr>

```

```

<h1>Día 2</h1>
<h2>Artículo 1</h2>
<div>Presentación del artículo 1</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam.</p>
</body>
</html>

```

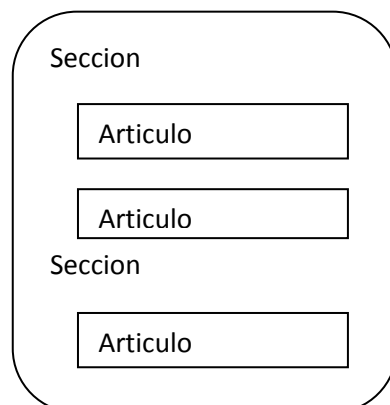
Apliquemos las etiquetas `<section>` y `<article>`.

```

<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<section>
<h1>Día 1</h1>
<article>
<h2>Artículo 1</h2>
<div>Presentación del artículo 1</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam.</p>
</article>
<article>
<h2>Artículo 2</h2>
<div>Presentación del artículo 2</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam.</p>
</article>
<hr>
</section>
<section>
<h1>Día 2</h1>
<article>
<h2>Artículo 1</h2>
<div>Presentación del artículo 1</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam.</p>
</article>
<hr>
</section>
</body>
</html>

```

Esta organización del código se puede visualizar así:



3-Nuevas etiquetas semánticas

Html5 introduce también toda una cantidad de etiquetas semánticas.

Etiqueta <hgroup>

La etiqueta <hgroup> pretende contener un grupo de etiquetas <hx> (al menos dos). Es una fórmula derivada de la etiqueta <header>.

```
<header>
<hgroup>
<h1>Título del artículo</h1>
<h2>Subtítulo del artículo</h2>
</hgroup>
</header>
```

Esta etiqueta podría retirarse de la versión definitiva de Html5.

Etiqueta <time>

La etiqueta <time> define una fecha y/o una hora. El atributo opcional datetime indica la fecha y la hora siguiendo la representación numérica internacional de la norma ISO8601.

```
<time>Mañana</time> será otro día.
<time datetime="2014-10-05">El 5 de octubre</time>
<time datetime="2014-04-13T00:19:00+02:00">Sábado 13 de abril
a las 19h</time>
```

Esta etiqueta no tiene representación visual en los navegadores.

Etiqueta <mark>

La etiqueta <mark> remarca parte del texto.

El contenido de esta etiqueta se subraya con un fondo amarillo.

Como el efecto de resaltado es nuevo en HTML, esta etiqueta a menudo pierde su cometido original para utilizarse como un efecto de presentación.

Etiqueta <meter>

La etiqueta <meter> define una medida. Se usa únicamente para medidas con un valor mínimo conocido y un valor máximo. Se utiliza para indicar un estado estable o evolucionando de manera puntual en el tiempo.

Esta etiqueta posee 6 atributos:

value:el valor del dato sobre la escala.
min:el valor mínimo posible.
low:el valor mínimo esperado.
high:el valor máximo esperado.
max:el valor máximo posible.
optimum:el valor máximo ideal.

Etiqueta <progress>

La etiqueta <progress> indica la progresión de una tarea, como las barras de progreso, que le resultarán familiares en la instalación de un programa. Solo tiene sentido si se utiliza junto con JavaScript para indicar el estado de progreso.

Esta etiqueta tiene dos atributos:

- value: el valor del progreso.
- max: el valor máximo posible.

Etiqueta <figure>

La etiqueta <figure> puede usarse para agrupar elementos tales como imágenes, vídeos o incluso texto que forma parte de una ilustración del contenido principal.

La etiqueta <figcaption>, usada de forma conjunta con la etiqueta <figure>, proporciona una leyenda a los elementos así agrupados, por ejemplo para añadir una leyenda a un grupo de imágenes.

Etiquetas <details> y <summary>

La etiqueta <details> indica contenido accesorio, que el usuario puede mostrar u ocultar. La etiqueta <summary> proporciona el título visible de estos elementos, sean un texto, una lista, una tabla, una imagen, etc.

Este espectacular efecto es similar a los menús desplegables u otros efectos acordeón y de manera nativa, sin hacer ninguna llamada a JavaScript. La etiqueta <details> puede contener cualquier elemento, sea texto, una lista, una tabla, una imagen, etc.

CAPITULO 9: Formularios

Los formularios ocupan un lugar predominante en el diseño y explotación de una aplicación o un sitio Web. Es, en efecto, la única forma de recibir de vuelta la información que proviene directamente del usuario final, estructurada, además, según las necesidades del diseñador. Basta con pensar en todos los sitios de carácter comercial en los que los formularios resultan indispensables.

La primera preocupación de los desarrolladores de una aplicación Web es recuperar datos válidos. Tomemos, por ejemplo, el dato de un código postal español, de cinco caracteres numéricos, o una dirección de correo electrónico en la que la sintaxis sea válida. Una de las primeras aplicaciones de JavaScript ha sido (y sigue siendo) la validación directa (del lado cliente) de los datos introducidos por el usuario. De este modo, todos los formularios de cierto nivel incluyen, además del código Html, una buena dosis de JavaScript para realizar la validación de los datos. Una de las ideas principales de Html5 consiste en que el lenguaje Html pueda realizar (y, por tanto, que los navegadores puedan realizar) la validación de los campos de los formularios, liberando así al código fuente de scripts redundantes. Todo esto incluso en el caso en que se haya desactivado el código JavaScript en el navegador.

Los formularios forman parte del lenguaje Html desde hace ya unos quince años y no se han modificado desde entonces. Reconozcamos que gráficamente, incluso después de dotarlos con propiedades de estilo, se ha instalado cierto hastío tanto en el lado de los diseñadores como en el de los usuarios. Por otro lado, con la Web 2.0, el usuario se ha habituado a una presentación más cercana a las aplicaciones de escritorio que a las páginas Web clásicas. Citemos los calendarios, habituales en los sitios de reserva en línea, o los cursores, usados en multitud de programas.

Integrando en el lenguaje Html las especificaciones de Xforms, Html5 muestra un importante interés agregando a los formularios heredados de Html 4.0 una cantidad impresionante de funciones inéditas, aproximándose en gran medida a lo que serían las aplicaciones de escritorio. La integración de estos nuevos campos de formulario en los navegadores se está haciendo muy progresivamente y habrá que esperar todavía algunos años antes de que todos los navegadores los gestionen por completo. Pero es innegable que Html5 ha abierto, para todos los diseñadores y desarrolladores, considerables perspectivas al respecto.

El propósito último de los formularios es el procesamiento automático de los datos devueltos. Por ello es preciso mencionar, del lado del servidor, los lenguajes de programación de gestión de bases de datos como, por ejemplo, PHP o MySQL. Entramos aquí en un terreno reservado a los programadores amateurs más avanzados, o profesionales. Afortunadamente para los debutantes existe el protocolo mailto, que permite enviar datos de formularios a una dirección de correo electrónico. Si bien este sistema presenta la ventaja de que puede usarlo todo el mundo, sin tener que recurrir a recursos externos, con lleva no obstante ciertas limitaciones. Además, el procesamiento posterior de los datos tendrá que hacerse de forma manual.

1-Declarar el formulario

La etiqueta <form> ... </form> tiene como única función la de declarar al navegador que debe incluir un formulario.

Esta etiqueta englobará los elementos o campos de formulario como, por ejemplo, una línea de texto, casillas a marcar, listas desplegadas, etc.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
```

```
<body>
<form action="">
Elementos del formulario
</form>
</body>
</html>
```

En sí, la etiqueta `<form>` no muestra nada en la ventana del navegador.

Los atributos usuales de la etiqueta `<form>` son:

name

Para asignar un nombre (*name*) al formulario.

action

Cuando demos la orden al navegador de enviar los datos del formulario, aquél precisa conocer la acción que debe realizar.

Esta acción será:

- Bien la llamada a un programa de procesamiento de datos, alojado en el servidor, en CGI, Perl, PHP, ASP... Por ejemplo: `action = "http://www.servidor/procesamiento.php"`
- Bien una dirección de correo electrónico para recuperar simplemente los datos. Se utiliza el protocolo mailto. Por ejemplo: `action="mailto:mi_correo@servidor"`
- Bien el procesamiento de los datos del formulario de forma interna (de lado cliente) mediante JavaScript, en cuyo caso el atributo `action` está vacío. Por ejemplo: `action=""`

Si usa el validador de W3C (<http://validator.w3.org>), verá que la presencia del atributo `action` es obligatoria.

enctype

El atributo `enctype` especifica en qué formato informático (*mime type*) se transmitirán los datos del formulario. El valor por defecto es `application/x-www-form-urlencoded`. Para enviar archivos este campo debe valer `multipart/form-data` y, por último, para el envío de datos a una dirección de correo electrónico mediante el protocolo mailto, el valor será `text/plain`.

method

La transmisión de datos de un formulario se realiza mediante el método GET o el método POST. El método GET realiza la transferencia de los datos en caracteres ASCII y los datos no pueden exceder los 100 caracteres. El método POST gestiona los caracteres no ASCII y tiene una capacidad de caracteres ilimitada. En la práctica se ha impuesto el método POST por motivos de facilidad y de eficacia.

Cuando los datos del formulario se procesan de forma interna (`action=""`), los atributos `method` y `enctype` son inútiles puesto que no se hace llamada al servidor.

- Esta etiqueta, básica en Html, es perfectamente compatible en los distintos navegadores.
- La etiqueta `<form>` debe, obligatoriamente, cerrarse. En caso de olvidar la etiqueta de cierre `</form>` el navegador no mostrará ningún campo del formulario.

2-Campo de texto

Este campo de texto permite captar la introducción de datos tanto alfabéticos como numéricos.

Ejemplo

Propongamos al usuario un campo de texto para introducir su nombre.

Nombre:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Nombre: <input type="text">
</form>
</body>
</html>
```

- Esta etiqueta, básica en Html, es perfectamente compatible en los distintos navegadores.
- La etiqueta `<input>` no tiene etiqueta de cierre. La sintaxis `<input type="text" />`, inspirada en Xhtml, está aceptada igualmente en Html5.

Los posibles atributos son:

name

Define un nombre (*name*) único para este elemento. Este atributo se utiliza para recoger el contenido del campo de texto cuando se envía el formulario.

size

Define el número de caracteres visibles del campo de texto y, por tanto, la longitud de la zona de texto. El usuario puede, no obstante, introducir tantos caracteres como quiera, incluso aunque sobrepasen la zona visible. El valor por defecto de este campo es 20.

maxlength

Determina el número máximo de caracteres que el usuario puede introducir en el campo de texto. Este atributo es particularmente útil para datos con un número concreto de caracteres como, por ejemplo, cinco cifras para introducir un código postal español.

value

Define el valor por defecto del campo de texto. Aparece como valor del campo de texto cuando se carga la página. Por ejemplo: `<input type="text" value="¡Escriba su nombre aquí!">`

readonly

Indica que el usuario no puede modificar el valor por defecto del campo de texto.

En Html5 han aparecido nuevos atributos. Actualmente no todos los navegadores los soportan.

placeholder

Permite incluir una sugerencia acerca del valor que se debe introducir en el campo de texto. Esta sugerencia aparece de color gris en el campo de formulario tras cargar la página y desaparece cuando el usuario sitúa el foco en el elemento afectado.

Hay quien podrá señalar que el atributo `value` realiza la misma función. La diferencia es sutil. Con `value`, el usuario tiene que borrar el contenido antes de introducir su propio texto. Con `placeholder`, el texto se borra automáticamente cuando se sitúa el foco sobre el elemento afectado. Además, con el atributo `value`, en caso de que se envíe el formulario sin haber modificado el valor de esta propiedad por defecto, será éste el valor que se envíe.

autofocus

Sitúa el foco sobre el elemento una vez cargada la página. Este atributo ahorra algunas líneas de JavaScript que realizan la misma función.

required

Convierte un elemento en obligatorio a la hora de enviar el formulario para su procesamiento posterior. Es muy útil para definir aquellos elementos obligatorios como el nombre o la dirección de correo electrónico del usuario.

pattern

Define una expresión regular JavaScript que se usará para realizar la validación del valor del campo. Por ejemplo `pattern="[0-9]"` indica que el valor del elemento debe ser un número entero comprendido entre 0 y 9.

Tomemos como ejemplo un campo de texto que va a contener un código postal (5 dígitos).

```
<form action="">
Código postal : <input type="text" pattern="[0-9]{5}"><br>
<input type="submit">
</form>
```

height

Determina, en píxeles o en porcentaje, la altura del campo de texto. Puede reemplazarse por la propiedad CSS `height`.

width

Determina, en píxeles o en porcentaje, la anchura del campo de texto. Puede reemplazarse por la propiedad CSS `width`.

3-Área de texto

En ciertas situaciones, es preciso prever algo más de espacio para que el usuario pueda expresarse. Es el caso, por ejemplo, de los comentarios o sugerencias. Se utiliza en este caso la etiqueta `<textarea> ... </textarea>` que incluye un área de texto de varias líneas.

Ejemplo

Un área de texto dedicada para los comentarios del usuario:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
<p>Comentarios: </p>
<textarea rows="4" cols="30"></textarea>
</form>
</body>
</html>
```

Comentarios:

- Esta etiqueta, básica en Html, es perfectamente compatible hacia atrás en los distintos navegadores.
- La etiqueta <textarea> precisa una etiqueta de cierre </textarea>.
- Es posible indicar un valor por defecto en el área de texto, no mediante el atributo value si no de la forma siguiente: <textarea rows="4" cols="25">¡Escriba aquí sus sugerencias!</textarea>

Los posibles atributos son:

name

Define un nombre para el elemento.

cols

Define el número de caracteres visibles en anchura para el área de texto. Puede reemplazarse por la propiedad CSS width.

rows

Define el número de líneas visibles en altura para el área de texto. Puede reemplazarse por la propiedad CSS height.

readonly

Indica que el valor informado por defecto en el área de texto no puede modificarse.

En Html5 han aparecido nuevos atributos. Actualmente no todos los navegadores los soportan.

He aquí algunos ejemplos:

autofocus

Sitúa el foco sobre el elemento una vez cargada la página.

maxlength

Determina el número máximo de caracteres que el usuario puede introducir en el área de texto.

required

Hace que sea obligatorio informar este elemento.

wrap

Especifica la forma en que se gestionan los saltos automáticos de línea en el texto a la hora de enviar el contenido. Con wrap="hard", se inserta un carácter de salto de línea junto con el texto. Con wrap="soft" (por defecto), no se inserta ningún carácter de salto de línea a la hora de transmitir la información.

Para los puristas, este atributo ya no formaba parte de la especificación Html 4.0. Se ha vuelto a introducir en la especificación Html5, no obstante bajo un aspecto modificado.

placeholder

Inserta un texto en el campo del formulario tras la carga de la página. Este texto se borra automáticamente cuando el usuario sitúa el foco en el área de texto.

4-Lista desplegable

La lista desplegable proporciona distintas opciones al usuario.

Ejemplo

Preguntemos al usuario cuál es su navegador preferido:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
<p>Su navegador preferido:
<select>
<option value="1">Internet Explorer</option>
<option value="2">Firefox</option>
<option value="3">Safari</option>
<option value="4">Opera</option>
<option value="5">Google Chrome</option>
</select>
</p>
</form>
</body>
</html>
```

- La etiqueta `<select> ... </select>` indica al navegador el uso de una lista desplegable. Los elementos de la lista se introducen mediante las etiquetas `<option> ... </option>`.
- Esta etiqueta es perfectamente compatible entre los distintos navegadores.

Los posibles atributos son:

name

Define un nombre para la lista desplegable, para un posible procesamiento posterior.

size

Por defecto, el atributo `size` de la etiqueta `<select>` es igual a 1, lo que resulta muy práctico dado que permite ganar bastante espacio en la disposición de la página. No obstante, mediante el atributo `size="x"` puede definir el número de elementos del menú que será visible.

multiple

Por defecto, el usuario no puede seleccionar más de un elemento del menú desplegable. Con el atributo múltiple de la etiqueta `<select>`, es posible seleccionar varios elementos a la vez. Para hacer esto, el usuario debe mantener pulsada la tecla [Ctrl] en el teclado y hacer clic sobre los elementos con el ratón. En tal caso es aconsejable recordar al usuario en la página esta forma de proceder, poco común para un internauta intermedio. El atributo `size` debe especificarse y ser igual al número de etiquetas `<option>`.

selected

Por defecto aparece seleccionado el primer elemento de la lista. No obstante es posible preseleccionar algún otro elemento mediante el atributo `selected` de la etiqueta `<option>`.

value

En principio es el texto del elemento seleccionado, situado tras la etiqueta `<option>`, el que se transmite a la hora de enviar el formulario. No obstante es posible especificar un valor diferente (generalmente un valor numérico) para transmitirlo, mediante el atributo `value="valor"`.

5-Botones de selección única (radio)

Los botones de selección única, también llamados botones radio, tienen la particularidad de que sólo permiten seleccionar una opción a la vez (el "o" exclusivo).

Ejemplo

Preguntemos el sexo del interlocutor:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Sexo:<br>
<input type="radio" name="sexo">Masculino
<input type="radio" name="sexo">Femenino
</form>
</body>
</html>
```

- Esta etiqueta es perfectamente compatible entre los distintos navegadores.
- La etiqueta <input> no tiene etiqueta de cierre. La sintaxis <input type ="radio"/>, inspirada en Xhtml, está igualmente aceptada en Html5.

Los posibles atributos son:

name

Aquí el atributo name es obligatorio. Además, en el caso de los botones radio, el nombre debe ser idéntico para todos los botones.

checked

Permite preseleccionar un botón radio.

value

En vista de un procesamiento posterior, es posible atribuir un valor a cada botón radio mediante el atributo value="valor".

6-Botones de selección múltiple (checkbox)

El uso de estos botones de selección múltiple, también llamados checkbox, es similar a los botones de selección única aunque, en este caso, es posible seleccionar varias opciones.

Ejemplo

Pidamos al usuario que precise los ingredientes de una pizza.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Su pizza con:<br>
<input type="checkbox" name="n1"> Mozzarella<br>
<input type="checkbox" name="n2"> Jamón<br>
```

```

<input type="checkbox" name="n3"> Salsa de tomate<br>
<input type="checkbox" name="n4"> Pimientos<br>
<input type="checkbox" name="n5"> Champiñones<br>
<input type="checkbox" name="n6"> Olivas negras<br>
<input type="checkbox" name="n7"> Salsa picante
</form>
</body>
</html>

```

- Esta etiqueta es perfectamente compatible entre los diferentes navegadores.
- La etiqueta `<input>` no tiene etiqueta de cierre. La sintaxis `<input type="checkbox" />`, inspirada en Xhtml, está igualmente aceptada en Html5.

Los posibles atributos son:

name

El atributo `name="nombre"` es obligatorio. Las reglas son menos precisas que para los botones de opción. Es posible usar nombres idénticos o nombres diferentes para cada casilla. No obstante, es preciso indicar nombres diferentes para tratar los elementos dentro de un script.

checked

El atributo `checked` permite preseleccionar una casilla.

value

Para poder procesarlos posteriormente es preciso indicar un valor a cada botón checkbox mediante el atributo `value="valor"`.

7-Botón de envío

```

<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
<input type="submit">
</form>
</body>
</html>

```

Enviar

- Esta etiqueta es perfectamente compatible entre los diferentes navegadores.
- La etiqueta `<input>` no tiene etiqueta de cierre. La sintaxis `<input type="submit" />`, inspirada en Xhtml, está igualmente aceptada en Html5.
- El texto del botón lo selecciona el navegador. En efecto, varía de un navegador a otro: **Enviar consulta** en Internet Explorer y Firefox, **Enviar** en Opera, Safari y Google Chrome. Es posible modificar el texto por defecto del botón mediante el atributo `value`. La sintaxis es, por ejemplo, `<input type="submit" value="Enviar el formulario">`.
- Es posible reemplazar el botón de envío por una imagen gráfica gracias a la etiqueta `<input type="image">`.

8-Botón de anulación

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
<input type="reset">
</form>
</body>
</html>
```

Restablecer

- Esta etiqueta es perfectamente compatible entre los diferentes navegadores.
- La etiqueta `<input>` no tiene etiqueta de cierre. La sintaxis `<input type="reset" />`, inspirada en Xhtml, está igualmente aceptada en Html5.
- El título del botón lo selecciona el navegador. La mayoría de ellos optan por **Restablecer**, salvo Safari que prefiere **Restaurar**. Es posible modificar el texto por defecto del botón mediante el atributo `value`. La sintaxis es, por ejemplo, `<input type="reset" value="Reiniciar">`.

9-Botón de acción

En materia de botones (*button*), existe también la etiqueta `<button> ... </button>`. Esta etiqueta permite ejecutar, al hacer clic sobre el botón, una acción específica definida por el diseñador del sitio Web, generalmente mediante algún código de JavaScript. Ofrece igualmente la posibilidad de realizar el envío y la anulación del formulario (`submit` y `reset`).

Por otro lado, esta etiqueta `<button>` tiene la ventaja de tener una etiqueta de apertura y otra de cierre. De este modo es posible incluir texto, imágenes o incluso contenido Html. Destaquemos también que esta etiqueta `<button>` no tiene por qué estar anidada obligatoriamente en un formulario `<form>` y puede usarse en múltiples contextos.

Todo esto la convierte en una etiqueta polivalente y justifica su éxito entre los desarrolladores.

Ejemplo

Al hacer clic en el botón, se abre una ventana de alerta:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<button onclick="alert('Acción JavaScript')">
Haga clic aquí</button>
</body>
</html>
```

- Esta etiqueta es perfectamente compatible entre los diferentes navegadores.
- La etiqueta `<button>` tiene una etiqueta de cierre: `</button>`.
- El texto del botón se declara entre ambas etiquetas, y no en el atributo `value`. Por ejemplo `<button>Haga clic aquí</button>`.

Los posibles atributos son:

name

Especifica un nombre para el botón.

type

El atributo type admite tres argumentos:

- button: se utiliza cuando el botón inicia un script (es el valor por defecto del atributo en la mayoría de navegadores).
- submit: para enviar el formulario.
- reset: para reinicializar el formulario.

disabled

Permite desactivar el botón cuando se carga la página. En este caso el botón aparece sombreado en gris. Es posible activarlo (enabled) con un script mediante una acción del usuario, por ejemplo después de haber aceptado las condiciones de uso.

10-Campos ocultos

Estos campos permiten almacenar datos que no estarán visibles al visitante de la página. Este tipo de elemento lo usan a menudo los programadores experimentados (especialmente en JavaScript) porque les permite almacenar datos de formularios anteriores, datos en bruto, o datos que provienen de algún otro script.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<input type="hidden" name="c" value="3.1416">
</body>
</html>
```

- Esta etiqueta es perfectamente compatible entre los diferentes navegadores.
- La etiqueta <input> no tiene etiqueta de cierre. La sintaxis <input type="hidden" />, inspirada en Xhtml, está igualmente aceptada en Html5.

11-Campos de transferencia de archivos

La etiqueta <input type="file"> permite transferir un archivo (*file*) desde el puesto del usuario a otro ordenador de tipo servidor.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="" method="post" enctype="multipart/form-data">
<input type="file">
</form>
```

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

Seleccionar archivo

Un clic en el botón **Examinar** (de nombre variable según el navegador) abre el explorador de carpetas del ordenador del usuario para seleccionar el archivo que se quiere transferir.

- Esta etiqueta es perfectamente compatible entre los diferentes navegadores.
- La etiqueta `<input>` no tiene etiqueta de cierre. La sintaxis `<input type="file" />`, inspirada en Xhtml, está igualmente aceptada en Html5.
- Los atributos habituales de la etiqueta `<input>` pueden utilizarse igualmente. Destacamos, no obstante, el atributo `maxlength` que permite fijar el tamaño máximo del archivo a transferir.
- En la declaración de la etiqueta `<form>`, es preciso usar los atributos `method="post"` y `enctype="multipart/formdata"` para realizar la transferencia según el formato correcto del archivo.
- Existe también el atributo `accept` que permite limitar la transferencia a ciertos tipos de archivos concretos. Por ejemplo archivos de texto (txt), Word (doc), Excel (xls), pdf, archivos de imagen (jpeg, gif o png), etc. En la descripción de los archivos es posible usar el carácter `*` como comodín.
- Destacamos que esta etiqueta Html no sirve más que para seleccionar el archivo que se quiere transferir. La transferencia en sí debe tenerse en cuenta en las aplicaciones del lado servidor, como por ejemplo PHP.

12-Campos de contraseña

Este tipo de campo es, de hecho, un simple campo de texto donde la codificación se sustituye, en la visualización, por puntos o asteriscos.

Este campo de contraseña no protege en ningún caso los datos, que se transmitirán sin cifrar. Únicamente nos protegen de las personas que pudieran estar mirando la pantalla en el momento de la introducción de la contraseña.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Login:
<input type="password">
</form>
</body>
</html>
```

- Esta etiqueta es perfectamente compatible entre los diferentes navegadores.
- La etiqueta `<input>` no tiene etiqueta de cierre. La sintaxis `<input type="password" />`, inspirada en Xhtml, está igualmente aceptada en Html5.
- Los atributos del campo de texto también pueden usarse aquí: `name`, `size`, `maxlength`, etc.

13-Organización de los campos de formulario

En el caso de formularios largos y complejos, en ocasiones resulta útil agrupar gráficamente ciertos elementos para organizar la página de forma lógica. Las etiquetas `<fieldset>` y `<legend>` permiten mejorar sensiblemente la ergonomía y la usabilidad de los formularios.

La etiqueta `<fieldset> ... </fieldset>` engloba los campos de formulario que usted determine. Estos campos se muestran en la pantalla rodeados por un borde.

La etiqueta `<legend> ... </legend>`, situada justo después de la etiqueta `<fieldset>`, agrega una leyenda relacionada con aquellos campos que hayamos agrupado mediante la etiqueta `<fieldset>`

- Esta etiqueta es perfectamente compatible entre los diferentes navegadores.
- No hay que olvidar incluir las etiquetas de cierre.
- Ya no existe una alineación de la leyenda a la derecha o a la izquierda en HTML5. Esta operación debe realizarse mediante una hoja de estilos CSS.

14-Etiquetado de los campos de formulario

La etiqueta `<label>` asocia explícitamente el título a un campo de formulario particular. Es como si pegáramos una etiqueta (*label*) delante de un elemento del formulario.

En un primer momento, la etiqueta `<label>` mejora en gran medida la ergonomía de los formularios permitiendo activar un elemento del grupo, por ejemplo un botón radio, haciendo clic sobre el botón radio o sobre el título, indiferentemente.

No obstante las etiquetas `<label>` son particularmente útiles en el dominio de la accesibilidad de los sitios Web por parte de personas invidentes. Estas "etiquetas" se tienen en cuenta en las ayudas técnicas (intérpretes de braille o programas de síntesis vocal) y facilitan en gran medida el uso de los formularios por parte de personas visualmente deficientes.

En primer lugar el texto asignado a un elemento de formulario debe situarse entre las etiquetas `<label> ...</label>`.

```
<label>Nombre</label>:
<input type="text"><br>
```

A continuación, es preciso asociar esta etiqueta `label` al control del formulario correspondiente. Para ello, el elemento de formulario tiene que definirse mediante un identificador de tipo `id`.

```
<label>Nombre</label>:
<input type="text" id="f1"><br>
```

El atributo `for="..."` dentro de la etiqueta `<label>` permite asociar directamente la etiqueta al campo haciendo referencia a este identificador.

```
<label for="f1">Nombre</label>:
<input type="text" id="f1"><br>
```

15-Campo de texto con lista de sugerencias

La etiqueta `<datalist>`, agregada a un campo de texto, abre una lista de sugerencias para introducir el dato sobre el que se ha situado el foco. Este comportamiento es similar al de Google Suggest a la hora de introducir alguna palabra clave en la barra de búsqueda.

Ejemplo

```
<!DOCTYPE html>
```

```

<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Escoja una fruta:
<input type="text" list="frutas">
<datalist id="frutas">
<option value="Naranja"></option>
<option value="Manzana"></option>
<option value="Pera"></option>
<option value="Ciruela"></option>
</datalist>
</form>
</body>
</html>

```

La etiqueta `<datalist>` contiene una lista de etiquetas `<option>` con las sugerencias. Estas etiquetas `<option>` deben poseer un atributo `value`.

La etiqueta `<datalist>` está ligada a un campo de formulario mediante un identificador `id` que se asocia con el atributo `list` de aquél.

16-Campo de texto para la dirección de correo electrónico

¿Cuántas veces le han pedido su dirección de correo electrónico navegando por la red?

Html5 responde a esta necesidad con la etiqueta `<input type="email">`.

Se convierte en una etiqueta particularmente interesante a la hora de realizar la validación directa en el navegador, sin que el diseñador tenga que agregar JavaScript.

Ejemplo

```

<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
e-mail: <input type="email" name="mail" required>
<input type="submit" value="OK">
</form>
</body>
</html>

```

17-Campo de texto url

Con un comportamiento similar a la etiqueta anterior `<input type="email">`, la etiqueta `<input type="url">` se define para introducir direcciones Web (*url*).

En este caso también se realiza una validación de forma directa (sin JavaScript) en el propio navegador.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Url de su sitio Web: <input type="url" name="web">
<input type="submit" value="OK">
</form>
</body>
</html>
```

18-Campo de texto con formato numérico

Otro campo nuevo incluido en Html5 son los contadores numéricos. Estos contadores, que se encuentran frecuentemente en aplicaciones como los procesadores de texto, se usan también en aplicaciones Web.

Esta nueva etiqueta `<input type="number">` tiene atributos específicos:

min

Indica el valor mínimo del contador.

max

Indica el valor máximo del contador.

step

Determina el incremento del contador cada vez que se hace clic en las flechitas.

value

Indica el valor inicial del contador.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Numero de artículos (grupos de 2):<br>
<input type="number" name="q" min="2" max="10" step="2" value="2">
</form>
</body>
</html>
```

- El código es compatible con el resto de navegadores que no tienen en cuenta esta nueva etiqueta y simplemente incluyen un campo de texto que debe completarse manualmente.

19-Campo de texto con formato de fecha

Los calendarios han cobrado especial importancia en las aplicaciones de la Web 2.0. Su uso es indiscutible en todos los sitios de reservas on line, bien sea para reservar un billete de avión o una habitación de hotel.

Para los desarrolladores, los formatos de fecha suponen un verdadero problema, sobre todo en sitios Web internacional (dd/mm/aa o mm/dd/aa).

Html5 proporcionará (en un futuro) campos de formulario especialmente adaptados a la codificación de fechas que permitirán visualizar directamente en el navegador, sin el uso de plugins particulares, un calendario donde el usuario podrá escoger la fecha en el formato deseado por el diseñador.

Se trata de uno de los avances más importantes de Html5 en cuanto al diseño de sitios Web.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Fecha de llegada: <input type="date" name="in"><br>
Fecha de partida: <input type="date" name="out">
</form>
</body>
</html>
```

Esta etiqueta <input> con formato de fecha se declina en los siguientes formatos.

<input type="date">

La más general, ilustrada anteriormente. Permite seleccionar el año, el mes y el día.

<input type="month">

Para seleccionar el mes y el año. Es útil, por ejemplo, para introducir la fecha de expiración de una tarjeta de crédito.

<input type="week">

Para una semana determinada.

No obstante Html5 no se queda aquí. Proporciona también una herramienta para el formato horario:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Entrega de la pizza a las:
<input type="time" min="11:00" max="23:00" step="900"
value="11:00">
</form>
</body>
</html>
```

El atributo min permite fijar una hora de inicio del contador, max una hora de fin y step el paso de avance, en este ejemplo de 15 en 15 minutos (900 segundos) cada vez que se hace clic en una flechita.

Por último, es posible combinar el calendario con el contador horario usando las etiquetas `<input type="datetime">` e `<input type="datetime-local">`.

```
<form action="">
Día y hora de llegada:
<input type="datetime" name="in">
</form>
```

20-Campo de texto de color

Html5 prevé un campo de texto destinado a recibir la codificación de un color (en formato hexadecimal) con la etiqueta `<input type="color">`.

Cuando se sitúa el foco en este campo de texto, se abre automáticamente una paleta de colores para realizar la selección del color, cuyo valor se reporta.

21-Cursores

Los cursores son herramientas intuitivas de fácil manejo que permiten fijar un valor concreto en muchas aplicaciones de software, generalmente de procesamiento numérico. Están presentes, por ejemplo, en programas de procesamiento de texto u otros programas como Photoshop.

Los cursores se introducen en Html5 mediante la etiqueta `<input type="range">`.

Los cursores están operativos en numerosos navegadores. Sólo Firefox no los implementa. Los demás navegadores los procesan como un campo de texto simple.

```
<form action="">
Cantidad solicitada<br>
<input name="cursor" type="range">
</form>
```

Esta nueva etiqueta `<input type="range">` tiene atributos específicos, idénticos a los de `<input type="number">`.

min

Indica el valor mínimo del contador.

max

Indica el valor máximo del contador.

step

Determina el paso de avance del contador cada vez que se hace clic con el ratón en la flecha para aumentar o disminuir (véase la siguiente captura de pantalla).

value

Determina el valor inicial del contador.

CAPITULO 10: Multimedia

1-Insertar un archivo de audio

1.1.Etiqueta <audio>

Html5 proporciona a día de hoy una etiqueta nueva para reproducir archivos de audio, independientemente de los plugins instalados en el puesto del usuario.

Insertar un archivo de audio puede realizarse de forma muy sencilla mediante la etiqueta:

```
<audio src="archivo_sonido"></audio>
```

Aunque la etiqueta <audio> está ampliamente implementada, no está de más prever un mensaje para aquellos navegadores que no puedan procesar esta etiqueta.

```
<audio src="archivo_sonido">
Su navegador no soporta la etiqueta audio.
</audio>
```

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<audio src="piano_ogg.ogg" controls>
Su navegador no soporta la etiqueta audio.
</audio>
</body>
</html>
```

Los atributos de la etiqueta <audio> son:

src

Obligatorio. Define la ruta del archivo de sonido.

controls

Muestra los controles del reproductor de audio (véase la captura de pantalla). Incluye las funcionalidades de reproducción, pausa, avance y volumen.

```
<audio src="sonido.ogg" controls>
```

En ausencia del atributo controls, el navegador no mostrará los controles del reproductor de audio.

La sintaxis Xhtml para los atributos controls="controls" también es válida.

autoplay

Define la reproducción automática del archivo de sonido desde el momento en que esté disponible, tras la carga de la página.

```
<audio src="sonido.ogg" autoplay>
```

La sintaxis Xhtml del atributo autoplay="autoplay" también es válida.

loop

Especifica que el archivo de sonido se reproduzca en bucle. De este modo, el sonido comienza de nuevo una vez termina.

```
<audio src="sonido.ogg" loop>
```

La sintaxis Xhtml del atributo loop="loop" también es válida.

preload

Indica al navegador que debe descargar el archivo de audio durante la carga de la página de modo que esté disponible para una reproducción inmediata una vez la solicite el usuario.

```
<audio src="sonido.ogg" preload>
```

Este atributo puede tomar los valores:

- preload="none". Sin carga previa.
- preload="metadata". Carga previa de los metadatos (metadata) asociados al archivo.
- preload="auto". Carga previa automática.

Este atributo preload se ignorará si el atributo autoplay está activado.

1.2. Formatos de archivo de audio

Formato ogg

Esta ha sido la opción de Firefox, primer navegador en implementar la etiqueta <audio>. El formato ogg se ha mantenido.

Ogg es el nombre del principal proyecto de la fundación Xiph.org que tiene como objetivo proporcionar a la comunidad de usuarios formatos y codecs multimedia abiertos, libres y desprovistos de cualquier tipo de patente. La extensión .ogg es una de las extensiones posibles para los archivos de formato ogg. Por abuso del lenguaje, se denomina de forma corriente "archivo Ogg" de audio a un archivo de formato ogg que contenga datos de audio comprimidos con Vorbis, uno de los codecs del proyecto ogg (fuente Wikipedia).

Los archivos ogg suponen una alternativa libre al formato mp3, mucho más conocido y expandido.

El formato ogg no lo reconocen los navegadores Safari ni Internet Explorer.

Formato mp3

MPEG1/2 Audio Layer 3, más conocido por su abreviatura MP3, es la especificación sonora del estándar MPEG1/MPEG2, de *Moving Picture Experts Group* (MPEG). Se trata de un algoritmo de compresión de audio capaz de reducir drásticamente la cantidad de datos necesaria para restituir el sonido, pero que, para el oyente, se asemeja a una reproducción del sonido original sin compresión, es decir con pérdida de calidad significativa pero aceptable para el oído humano (fuente Wikipedia).

Mp3 se percibe a menudo por parte del usuario como una tecnología gratuita. Esto permite que pueda codificar y decodificar su música de forma totalmente legal siempre y cuando el archivo original le pertenezca o se trate de una copia de uso privado. No obstante, esta tecnología propietaria forma parte de una patente y una licencia comercial.

El formato mp3 no se reconoce en Firefox.

Formato acc

El formato ACC (*Advanced Audio Coding*) es un algoritmo de compresión de audio con pérdida de datos pero que tiene mejor rendimiento en términos de compresión que el formato más antiguo MPEG1/2 Audio Layer 3 (más conocido como mp3). Por este motivo ha sido escogido por algunas firmas como Apple o RealNetworks.

El formato acc es el formato de los archivos de audio soportados por Apple en su reproductor portátil iPod y en su software iTunes.

Se trata también de un formato propietario.

El formato acc no lo reconoce el navegador Firefox.

Formato wav

Citamos el clásico formato wav que, por su ausencia de compresión, no parece uno de los más adaptados para competir en la red. En efecto, el tamaño de los archivos wav es muy (o demasiado) grande.

El formato wav no se reconoce en InternetExplorer 9 ni en iOS Safari.

- La etiqueta Html5 <audio> ha sido implementada en los navegadores más recientes.
- Google Chrome es, con diferencia, el navegador más polivalente en lo referente a formatos de audio.
- Para que su archivo de audio sea compatible con todos los navegadores Html5, basta que prevea tenerlo al menos en dos formatos, por ejemplo en ogg y en mp3.
- El formato mp4, dedicado al vídeo, también se puede utilizar para datos de audio.

1.3. Etiqueta <source>

La etiqueta <source> nos va a permitir resolver la problemática de los distintos formatos de archivo.

La etiqueta <source> se utiliza para especificar varios tipos de archivos de audio. Cada navegador escogerá el formato que mejor le convenga o el que pueda reproducir.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<audio controls>
<source src="piano_ogg.ogg">
<source src="piano_mp3.mp3">
<source src="piano_acc.acc">
Su navegador no soporta la etiqueta audio.
</audio>
</body>
</html>
```

Supongamos que leemos esta página en el navegador Safari que, recordemos, no soporta el formato ogg. Safari ignorará la primera etiqueta <source> que redirige a un archivo de tipo ogg. Pasa a la segunda etiqueta <source> que propone un archivo mp3. Como el formato mp3 sí está soportado en Safari, será este recurso el que se usará para reproducir el archivo de sonido.

Por el contrario, Firefox, que soporta los archivos ogg, utilizará el primer recurso. Las demás etiquetas <source> serán ignoradas.

Las etiquetas <source> se leen en el orden de aparición en el código fuente. El navegador usará el primer recurso que soporte.

Las múltiples etiquetas <source> aseguran una perfecta reproducción del archivo de sonido en sus distintos formatos por parte de los navegadores. No obstante, el diseñador del sitio o de la aplicación Web tiene que codificar el mismo archivo de sonido en varios formatos para asegurar la compatibilidad entre los distintos formatos.

Los atributos de la etiqueta <audio> son:

src

Obligatorio. Define la ruta del archivo de sonido.

type

Define el tipo MIME del contenido. Puede valer:

- type="audio/ogg"
- type="audio/mpeg"
- type="audio/acc"

2-Insertar un archivo de video

2.1. Etiqueta <video>

Html5 proporciona, a día de hoy, una etiqueta nueva para reproducir los archivos de vídeo, independientemente de los plugins instalados en el puesto del usuario.

Insertar un archivo de vídeo se realiza de forma muy sencilla mediante la etiqueta:

```
<video src="archivo_video"></video>
```

Aunque la implementación de la etiqueta <video> es importante, no es descabellado prever un mensaje para los usuarios de los navegadores antiguos que todavía no tienen en cuenta esta etiqueta.

```
<video src="archivo_video">
Su navegador no soporta la etiqueta video.
</video>
```

Veamos un primer ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<video src="video_ogv.ogv" controls>
Su navegador no soporta la etiqueta video.
</video>
</body>
</html>
```

Los atributos de la etiqueta <video> son:

src

Obligatorio. Define la ruta del archivo de vídeo.

width

Determina la anchura del vídeo.

height

Determina la altura del vídeo.

Igual que con la etiqueta , puede definir de forma explícita las dimensiones del vídeo. En otro caso, el elemento se muestra por defecto con la altura y la anchura propias del vídeo. Si especifica una dimensión, pero no la otra, el navegador ajustará automáticamente el tamaño de la dimensión que no se haya especificado para preservar la proporción de aspecto del vídeo.

poster

El atributo poster permite especificar una imagen que el navegador usará mientras se está descargando el vídeo o hasta que el usuario inicie su reproducción. Si no se especifica este atributo, se inserta la primera imagen del vídeo en su lugar.

```
<video src="video_ogv.ogv" poster="imagen.png">
```

controls

Muestra los controles del reproductor del vídeo. Se incluyen las funciones de reproducción, pausa, avance y volumen.

En ausencia del atributo controls, no se mostrarán los controles del reproductor en el navegador.

La sintaxis Xhtml del atributo controls="controls" también es válida.

autoplay

Define la lectura automática del archivo de vídeo una vez esté disponible, tras la carga de la página.

```
<video src="video_ogv.ogv" autoplay>
```

Recordemos que esta reproducción automática no siempre es apreciada por los usuarios, especialmente si la página contiene algo de texto.

La sintaxis Xhtml del atributo autoplay="autoplay" también es válida.

loop

Especifica que el archivo de vídeo se reproducirá en bucle. De este modo, el vídeo se reproduce de nuevo una vez ha llegado a su fin.

```
<video src="video_ogv.ogv" loop>
```

La sintaxis Xhtml del atributo loop="loop" también es válida.

preload

Indica al navegador que debe descargar el archivo de vídeo durante la carga de la página de modo que esté disponible para una reproducción inmediata una vez lo solicite el usuario.

```
<video src="video_ogv.ogv" preload>
```

Este atributo puede tomar los valores:

- preload="none". No existe carga previa.
- preload="metadata". Carga previa de los metadatos asociados al archivo.
- preload="auto". Carga previa automática.

Este atributo preload se ignora si está presente el atributo autoplay.

2.2. Formatos de archivo de vídeo

Formato ogv

El formato ogv para los archivos de vídeo es el equivalente a la extensión ogg para los archivos de audio.

Formato H.264

H.264 es un codec de compresión de vídeo digital y de vídeo en alta resolución según la norma MPEG4, desarrollada por el VCRG (*Video Coding Experts Group*) en asociación con el MPEG (*Moving Picture Experts Group*), también conocido con el nombre de AVC (*Advanced Video Coding*). El formato H.264 también se conoce con la nomenclatura MPEG4 AVC o incluso MPEG4 Part 10.

El formato H.264 no es un formato libre ni gratuito. Se trata de un formato de vídeo propietario que forma parte de acuerdos comerciales y royalties.

El formato H.264 se ha convertido en un estándar (no oficial) de vídeo en la web.

Formato WebM

Google adquirió el codec de vídeo VP8 tras la compra de la empresa On2 Technologies.

Para contrarrestar el ascenso fulgurante del formato H.264, Google publicó en mayo de 2010 el código fuente del codec VP8, libre de licencia. Está asociado a Mozilla y Opera y proporciona un formato de vídeo abierto, WebM, que utiliza VP8 para el flujo de vídeo y Vorbis para el flujo de audio.

- La etiqueta html <video> está implementada en los navegadores más recientes.
- Google Chrome es, con diferencia, el navegador más polivalente en lo que a formatos de vídeo se refiere.

2.3. Etiqueta <source>

La etiqueta <source> nos va a permitir resolver la problemática de los distintos formatos de archivo.

La etiqueta <source> se utiliza para especificar varios tipos de archivos de vídeo. Cada navegador escogerá el formato que mejor le convenga o el que pueda reproducir.

Ejemplo

```
<body>
<video controls>
<source src="video_ogv.ogv">
<source src="video_mp4.mp4">
<source src="video_webm.webm">
Su navegador no soporta la etiqueta video.
</video>
</body>
```

Es importante comprender el funcionamiento de las distintas etiquetas <source>.

Supongamos que leemos esta página en el navegador Safari que, recordemos, no soporta el formato ogv. Safari ignorará la primera etiqueta <source> que remite a un archivo de tipo ogv. Pasa a la segunda etiqueta <source> que propone un archivo mp4. Como el formato mp4 sí está soportado en Safari, será este recurso el que se usará para reproducir el archivo de vídeo.

Por el contrario, Firefox, que soporta los archivos ogv, utilizará el primer recurso. Las demás etiquetas <source> serán ignoradas.

Las etiquetas <source> se leen en el orden de aparición en el código fuente. El navegador usará el primer recurso que soporte.

Las múltiples etiquetas <source> aseguran una perfecta reproducción del archivo de vídeo en sus distintos formatos por parte de los navegadores. No obstante, el diseñador del sitio o de la aplicación Web tiene que codificar el mismo archivo de vídeo en varios formatos para asegurar la compatibilidad entre los distintos formatos.

Los atributos de la etiqueta <source> son:

src

Obligatorio. Define la ruta del archivo de sonido.

type

Define el tipo MIME del contenido. Puede valer:

- type="video/ogg"
- type="video/mp4"
- type="video/webm"

Especificando el atributo type acelerará el proceso de carga de la etiqueta <source> más adecuada para el navegador.