

LENGUAJES DE MARCAS Y SISTEMAS DE GESTIÓN DE INFORMACIÓN

1º Desarrollo de Aplicaciones Multiplataforma.

UND 2: HTML.



Índice de Contenidos

INTRODUCCIÓN	7
ESTRUCTURA DE UN DOCUMENTO	7
Elementos de la cabecera	8
<base>.....	8
<link>	8
<title>.....	8
<meta>.....	8
<script>	9
<style>	9
Elementos de cuerpo de documento	9
<main>	13
<header>.....	13
<nav>	14
<section>	15
<aside>.....	16
<footer>	17
Dentro del contenido.....	18
<article>	19
<hgroup>	23
<figure> y <figcaption>	25
COLOR.....	26
Codificación de colores.....	26
TEXTO	28
Encabezados, jerarquía y estructura del contenido de un documento.....	29
Párrafos.....	31
<p>	31
Alineación, espaciado y sangrado de texto	32
<pre>.....	33
Separadores de texto.....	33
<hr>.....	33
Etiquetas propias de bloques de texto.	35

<abbr>.....	35
<acronym>	35
<cite>	35
<var>	36
<code>	36
<samp>	36
	36
<ins>	36
<dfn>.....	37
.....	37
<kbd>	37
.....	37
Etiquetas de estilos lógicos aplicadas a un conjunto de caracteres	38
	38
<bdo>.....	38
<big>	38
<i>	38
<q>	39
<small>.....	39
<sub>	39
<sup>	39
<tt>	39
ENLACES DE HIPERTEXTO	40
<a>	40
<script> ...</script>	40
<link>	40
Estructura de un enlace: la dirección de internet o url	41
Enlaces internos.....	42
Enlaces especiales: correo electrónico. Enlaces de descarga	43
Enlaces de correo electrónico.	43
Enlaces de descarga.....	44
IMÁGENES	44

	45
<map>	45
<area>	45
Formatos de imágenes	46
Características de imágenes: tamaño, título, textos alternativos	46
Tamaño	46
Título	47
Texto alternativo	48
Enlaces en imágenes	48
Imágenes de fondo	48
LISTAS	49
	49
	49
	49
Ordenación de listas	50
Anidamiento en listas	51
TABLAS	51
Estructura básica	51
<table>	52
<tr>	52
<td>	52
<th>	52
<caption>	52
<thead>	52
Formatos de tablas: bordes, alineación, tamaño, etc.	53
Bordes	53
Alineación.	54
Formato de contenido de celdas.	56
Tablas anidadas.	59
MARCOS (FRAMES)	61
Creación de marcos	61
Marcos incrustados (iframes)	63

FORMULARIOS.....	64
<input>.....	65
type="..."	65
value="..."	65
size="..."	65
maxlength="..."	65
checked="checked"	66
disable="disable"	66
readonly="readonly"	66
src="url"	66
alt="..."	66
<fieldset>	66
<legend>	66
<label>	66
<textarea>.....	66
cols="..."	66
rows="..."	66
disabled="disabled"	66
readonly="readonly"	66
<select>.....	67
size="..."	67
multiple="multiple"	67
disabled="disabled"	67
<option>	67
selected="selected"	67
value="..."	67
disabled="disabled"	67
<optgroup>	67
label="..."	67
disabled="disabled"	67
Nuevos atributos en HTML5	68
Placeholder	68

Autofocus.....	68
Autocomplete	68
Height y width	68
List.....	68
Min y Max	69
Multiple	69
Novalidate.....	69
Pattern	69
Required	69
Nuevos tipos de input en HTML5	70
Búsqueda (search)	70
Teléfono (tel)	71
Dirección url (url).....	71
Correo electrónico (email).....	71
Fecha y hora (datetime).....	72
Fecha (date)	72
Mes (month)	72
Semana (week)	72
Hora (time)	73
Número (number).....	73
Rango (range)	73
Color (color)	74

INTRODUCCIÓN.

HTML (Hyper Text Markup Language) o **lenguaje de marcas de hipertexto**, es el lenguaje que permite la generación de hipertextos en la World Wide Web. Podemos definir el hipertexto como una manera de organizar la información de forma no secuencial. Mediante el modelo de hipertexto un sistema puede organizar y presentar los documentos en un medio informático, basado en la vinculación de documentos o fragmentos documentales digitales (textuales o gráficos) a otros fragmentos o documentos, lo que permite acceder a la información no necesariamente de forma secuencial.

ESTRUCTURA DE UN DOCUMENTO

Todo documento HTML tiene la siguiente estructura básica:

```
<html>
  <head> ...
</head>
  <body> ...
</body>
</html>
```

Las etiquetas `<html>` y `</html>` nos indican que todo el texto contenido entre ambas es código HTML. Por tanto, la primera debe abrir este tipo de documentos y la segunda, cerrarlos. La cabecera del documento, delimitada por `<head>` y `</head>`, proporciona información acerca de él.

El contenido, propiamente dicho, se aloja dentro del cuerpo del documento y viene señalado con las etiquetas `<body>` y `</body>`.

Esa estructura debe ser antecedita por una declaración de tipo de documento. Por ejemplo:

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

Esta declaración indica que las definiciones que describen la estructura de un documento XHTML se exponen en un documento DTD en la Red cuyo URL se especifica. De esta forma, los clientes web conocerán sin ambigüedad con qué reglas deben procesar el documento.

O simplemente:

```
<!DOCTYPE html> (Esta es para HTML5)
```

Elementos de la cabecera

Lo primero que diseñamos en la página web, es la cabecera. En la cabecera de un documento HTML se incluyen las siguientes etiquetas que dan información de la página o soporte para ella:

<base> Permite indicar la URL por defecto para las URL relativas, así como el lugar donde se abrirán las páginas enlazadas.

<link> Sirve para relacionar un documento con los recursos externos que utiliza, como por ejemplo la hoja de estilos aplicable al documento. Ejemplo:

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

<title> Como ya hemos visto sirve especificar el texto que aparecerá en la barra de título de la ventana, es importante no confundir con que al hacer una página web la información que presentemos lleve un título, el cual deberemos poner con una etiqueta de encabezado **<h1>** antes de los párrafos de información.

<meta> Permite proporcionar metainformación al navegador sobre la página. Se escriben tantas etiquetas meta, como valores de **name** queramos utilizar.

```
<meta name="..." content="..." />
```

Algunos de los valores de name son:

- **author**, permite indicar el autor de la página.
- **keywords**, lista de palabras clave separadas por comas.
- **description**, permite incluir una breve descripción de la página.

Por ejemplo:

```
<head>  
  <meta name="author" content="Francisco" />  
  <meta name="keywords" lang="es" content="instituto, granada, clase,  
  ciclo formativo" />  
  <meta name="description" lang="es" content="Página de ejemplo para la  
  explicación" />  
</head>
```


Cosas a tener en cuenta:

El atributo de [scheme](#) no es compatible con HTML5.

HTML5 tiene un nuevo atributo, [charset](#), que hace más fácil definir el conjunto de caracteres:

HTML 4.01:

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

HTML5:

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

Valores más comunes:

[UTF-8](#) – Conjunto de caracteres Unicode. o [ISO-8859-1](#) – Conjunto de caracteres para el alfabeto latino.

[<script>](#) Con esta etiqueta podemos incluir código en algún lenguaje cliente, por ejemplo, JavaScript. Esta etiqueta también puede aparecer una o más veces en la sección body.

Por ejemplo:

```
<script type="text/javascript"> alert("hola") </script>
```

[<style>](#) Al igual que con [<link>](#) podemos especificar los estilos para etiquetas dentro del mismo documento, por ejemplo:

```
<style type="text/css">
body {background: black; color:white;} div {background: red; width:300px;}
</style>
```

Elementos de cuerpo de documento

La estructura del cuerpo (el código entre las etiquetas [<body>](#)) generará la parte visible del documento. Este es el código que producirá nuestra página web.

HTML ofrece diferentes formas de construir y organizar la información dentro del cuerpo de un documento. Uno de los primeros elementos provistos para este propósito fue [<table>](#). Las tablas permitían a los diseñadores acomodar datos, texto, imágenes y herramientas dentro de filas y columnas de celdas, incluso sin que hayan sido concebidas para este propósito.

En los primeros días de la web, las tablas fueron una revolución, un gran paso hacia adelante con respecto a la visualización de los documentos y la experiencia ofrecida a los usuarios. Más adelante, gradualmente, otros elementos reemplazaron su función, permitiendo lograr lo mismo con menos código, facilitando de este modo la creación, permitiendo portabilidad y ayudando al mantenimiento de los sitios web.

El elemento `<div>` comenzó a dominar la escena. Con el surgimiento de webs más interactivas y la integración de **HTML**, **CSS** y **JavaScript**, el uso de `<div>` se volvió una práctica común. Pero este elemento, así como `<table>`, no provee demasiada información acerca de las partes del cuerpo que está representando. Desde imágenes a menús, textos, enlaces, códigos, formularios, cualquier cosa puede ir entre las etiquetas de apertura y cierre de un elemento `<div>`. En otras palabras, la palabra clave `div` solo especifica una división en el cuerpo, como la celda de una tabla, pero no ofrece indicio alguno sobre qué clase de división es, cuál es su propósito o qué contiene.

Para los usuarios estas claves o indicios no son importantes, pero para los navegadores la correcta interpretación de qué hay dentro del documento que se está procesando puede ser crucial en muchos casos. Luego de la revolución de los dispositivos móviles y el surgimiento de diferentes formas en que la gente accede a la web, la identificación de cada parte del documento es una tarea que se ha vuelto más relevante que nunca.

Considerando todo lo expuesto, HTML5 incorpora nuevos elementos que ayudan a identificar cada sección del documento y organizar el cuerpo del mismo. En HTML5 las secciones más importantes son diferenciadas y la estructura principal ya no depende más de los elementos `<div>` o `<table>`.

Cómo usamos estos nuevos elementos depende de nosotros, pero las palabras clave otorgadas a cada uno de ellos nos ayudan a entender sus funciones. Normalmente una página o aplicación web está dividida entre varias áreas visuales para mejorar la experiencia del usuario y facilitar la interactividad. Las palabras claves que representan cada nuevo elemento de HTML5 están íntimamente relacionadas con estas áreas, como veremos pronto. La siguiente imagen representa un diseño común encontrado en la mayoría de los sitios webs estos días.

A pesar del hecho de que cada diseñador crea sus propios diseños, en general podremos identificar las siguientes secciones en cada sitio web estudiado:



Representación visual de un clásico diseño web.

En la parte superior, descrito como **Cabecera**, se encuentra el espacio donde usualmente se ubica el logo, título, subtítulos y una corta descripción del sitio web o la página.

Inmediatamente debajo, podemos ver la **Barra de Navegación** en la cual casi todos los desarrolladores ofrecen un menú o lista de enlaces con el propósito de facilitar la navegación a través del sitio. Los usuarios son guiados desde esta barra hacia las diferentes páginas o documentos, normalmente pertenecientes al mismo sitio web.

El contenido más relevante de una página web se encuentra, en casi todo diseño, ubicado en su centro. Esta sección presenta información y enlaces valiosos. La mayoría de las veces es dividida en varias filas y columnas. En el ejemplo de la figura se utilizaron solo dos columnas: **Información Principal** y **Barra Lateral**, pero esta sección es extremadamente flexible y normalmente diseñadores la adaptan acorde a sus necesidades insertando más columnas, dividiendo cada columna entre bloques más pequeños o generando diferentes distribuciones y combinaciones. El contenido presentado en esta parte del diseño es usualmente de alta prioridad. En el diseño de ejemplo, Información Principal podría contener una lista de artículos, descripción de productos, entradas de un blog o cualquier otra información importante, y la Barra Lateral podría mostrar una lista de enlaces apuntando hacia cada uno de esos ítems. En un blog, por ejemplo, esta última columna ofrecerá una lista de enlaces apuntando a cada entrada del blog, información acerca del autor, etc.

En la base de un diseño web clásico siempre nos encontramos con una barra más que aquí llamamos **Institucional**. La nombramos de esta manera porque esta es el área en donde normalmente se muestra información acerca del sitio web, el autor o la empresa, además de algunos enlaces con respecto a reglas, términos y condiciones y toda información adicional que el desarrollador considere importante compartir. La barra Institucional es un complemento de la Cabecera y es parte de lo que se considera estos días la estructura esencial de una página web, como podemos apreciar en el siguiente ejemplo:



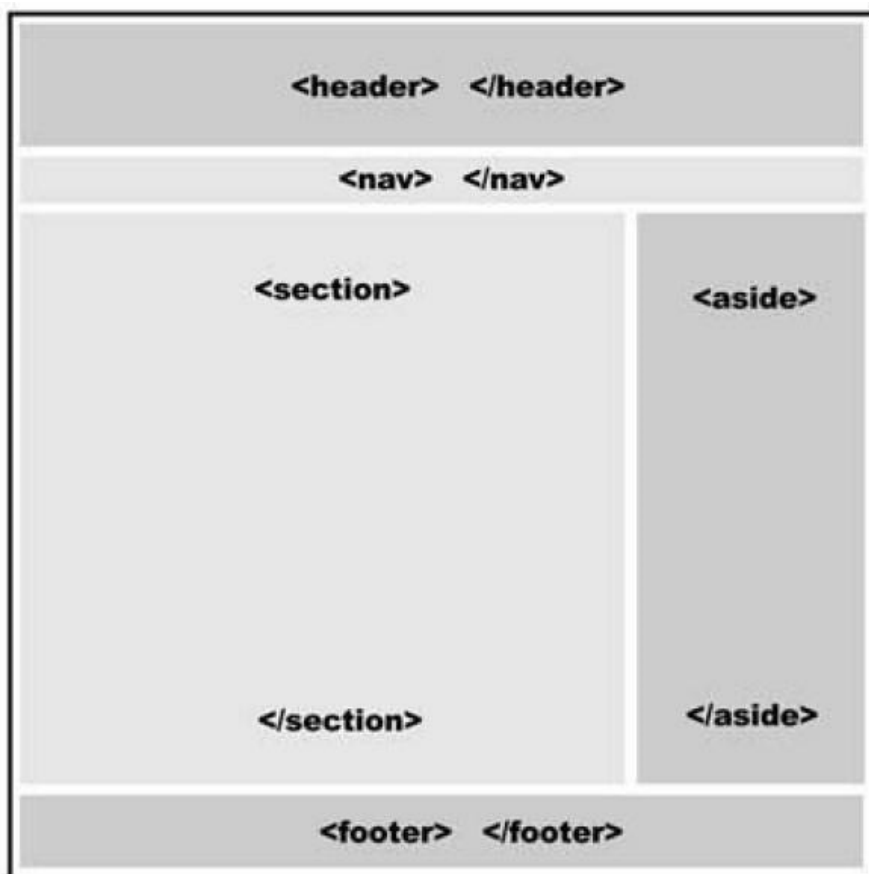
Representación visual de un clásico diseño para blogs.

La figura es una representación de un blog normal. En este ejemplo se puede claramente identificar cada parte del diseño considerado anteriormente:

- Cabecera
- Barra de Navegación
- Sección de Información Principal
- Barra Lateral
- El pie o la barra Institucional

Esta simple representación de un blog nos puede ayudar a entender que cada sección definida en un sitio web tiene un propósito.

HTML5 considera esta estructura básica y provee nuevos elementos para diferenciar y declarar cada una de sus partes. A partir de ahora podemos decir al navegador para qué es cada sección:



Representación visual de un diseño utilizando elementos HTML5.

La figura muestra el típico diseño presentado anteriormente, pero esta vez con los correspondientes elementos HTML5 para cada sección (incluyendo etiquetas de apertura y cierre).

`<main>` Engloba el contenido principal de la página o de la aplicación. Solo puede haber una etiqueta `<main>` en el documento.

`<header>` Uno de los nuevos elementos incorporados en HTML5 es `<header>`. El elemento `<header>` no debe ser confundido con `<head>` usado antes para construir la cabecera del documento. Del mismo modo que `<head>`, la intención de `<header>` es proveer información introductoria (títulos, subtítulos, logos), pero difiere con respecto a `<head>` en su alcance.

Mientras que el elemento <head> tiene el propósito de proveer información acerca de todo el documento, <header> es usado solo para el cuerpo o secciones específicas dentro del cuerpo:

```
<!DOCTYPE html>

<html lang="es">
  <head>
    <meta charset="iso-8859-1">
    <meta name="description" content="Ejemplo de HTML5">
    <meta name="keywords" content="HTML5, CSS3, JavaScript">
    <title>Este texto es el título del documento</title>
    <link rel="stylesheet" href="misestilos.css">
  </head>
  <body>
    <header>
      <h1>Este es el título principal del sitio web</h1>
    </header>
  </body>
</html>
```

En el código definimos el título de la página web utilizando el elemento <header>. Recuerde que esta cabecera no es la misma que la utilizada previamente para definir el título del documento. La inserción del elemento <header> representa el comienzo del cuerpo y por lo tanto de la parte visible del documento. A partir de aquí será posible ver los resultados de nuestro código en la ventana del navegador.

Conceptos básicos: Entre las etiquetas <header> en el código hay un elemento que probablemente no conoce. El elemento <h1> es un viejo elemento HTML usado para definir títulos. El número indica la importancia del título. **El elemento <h1> es el más importante y <h6> el de menor importancia**, por lo tanto <h1> será utilizado para mostrar el título principal y los demás para subtítulos o subtítulos internos. Más adelante veremos cómo estos elementos trabajan en HTML5.

<nav> Siguiendo con nuestro ejemplo, la siguiente sección es la Barra de Navegación. Esta barra es generada en HTML5 con el elemento <nav>:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="iso-8859-1">
    <meta name="description" content="Ejemplo de HTML5">
    <meta name="keywords" content="HTML5, CSS3, JavaScript">
    <title>Este texto es el título del documento</title>
    <link rel="stylesheet" href="misestilos.css">
  </head>
  <body>
    <header>
      <h1>Este es el título principal del sitio web</h1>
    </header>
```

```
        <nav>
            <ul>
                <li>principal</li>
                <li>fotos</li>
                <li>videos</li>
                <li>contacto</li>
            </ul>
        </nav>
    </body>
</html>
```

Como se puede apreciar en este código, el elemento `<nav>` se encuentra dentro de las etiquetas `<body>` pero es ubicado después de la etiqueta de cierre de la cabecera (`</header>`), no dentro de las etiquetas `<header>`. Esto es porque `<nav>` no es parte de la cabecera sino una nueva sección.

Anteriormente dijimos que la estructura y el orden que elegimos para colocar los elementos HTML5 dependen de nosotros. Esto significa que HTML5 es versátil y solo nos otorga los parámetros y elementos básicos con los que trabajar, pero cómo usarlos será exclusivamente decisión nuestra. Un ejemplo de esta versatilidad es que el elemento `<nav>` podría ser insertado dentro del elemento `<header>` o en cualquier otra parte del cuerpo. Sin embargo, siempre se debe considerar que estas etiquetas fueron creadas para brindar información a los navegadores y ayudar a cada nuevo programa y dispositivo en el mercado a identificar las partes más relevantes del documento. Para conservar nuestro código portable y comprensible, recomendamos como buena práctica seguir lo que marcan los estándares y mantener todo tan claro como sea posible. El elemento `<nav>` fue creado para ofrecer ayuda para la navegación, como en menús principales o grandes bloques de enlaces, y debería ser utilizado de esa manera.

Conceptos básicos: En el ejemplo generamos las opciones del menú para nuestra página web. Entre las etiquetas `<nav>` hay dos elementos que son utilizados para crear una lista. El propósito del elemento `` es definir la lista. Anidado entre las etiquetas `` encontramos varias etiquetas `` con diferentes textos representando las opciones del menú. Las etiquetas ``, como probablemente ya se ha dado cuenta, son usadas para definir cada ítem de la lista.

`<section>` Siguiendo nuestro diseño estándar nos encontramos con las columnas que llamamos **Información Principal** y **Barra Lateral**. Como explicamos anteriormente, la columna **Información Principal** contiene la información más relevante del documento y puede ser

encontrada en diferentes formas (por ejemplo, dividida en varios bloques o columnas). Debido a que el propósito de estas columnas es más general, el elemento en HTML5 que especifica estas secciones se llama simplemente `<section>`:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="iso-8859-1">
    <meta name="description" content="Ejemplo de HTML5">
    <meta name="keywords" content="HTML5, CSS3, JavaScript">
    <title>Este texto es el título del documento</title>
    <link rel="stylesheet" href="misestilos.css">
  </head>
  <body>
    <header>
      <h1>Este es el título principal del sitio web</h1>
    </header>
    <nav>
      <ul>
        <li>principal</li>
        <li>fotos</li>
        <li>videos</li>
        <li>contacto</li>
      </ul>
    </nav>
    <section>
    </section>
  </body>
</html>
```

Al igual que la Barra de Navegación, la columna Información Principal es una sección aparte. Por este motivo, la sección para Información Principal va debajo de la etiqueta de cierre `</nav>`.

`<aside>` En un típico diseño web la columna llamada **Barra Lateral** se ubica al lado de la columna Información Principal. Esta es una columna o sección que normalmente contiene datos relacionados con la información principal pero que no son relevantes o igual de importantes.

En el diseño de un blog, por ejemplo, la Barra Lateral contendrá una lista de enlaces. En el ejemplo, los enlaces apuntan a cada una de las entradas del blog y ofrecen información adicional sobre el autor. La información dentro de esta barra está relacionada con la información principal pero no es relevante por sí misma. Siguiendo el mismo ejemplo podemos decir que las entradas del blog son relevantes pero los enlaces y las pequeñas reseñas sobre esas entradas son solo una ayuda para la navegación, pero no lo que al lector realmente le interesa.

En HTML5 podemos diferenciar esta clase secundaria de información utilizando el elemento `<aside>`:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="iso-8859-1">
    <meta name="description" content="Ejemplo de HTML5">
    <meta name="keywords" content="HTML5, CSS3, JavaScript">
    <title>Este texto es el título del documento</title>
    <link rel="stylesheet" href="misestilos.css">
  </head>
  <body>
    <header>
      <h1>Este es el título principal del sitio web</h1>
    </header>
    <nav>
      <ul>
        <li>principal</li>
        <li>fotos</li>
        <li>videos</li>
        <li>contacto</li>
      </ul>
    </nav>
    <section>
    </section>
    <aside>
      <blockquote>Mensaje número uno</blockquote>
      <blockquote>Mensaje número dos</blockquote>
    </aside>
  </body>
</html>
```

El elemento `<aside>` podría estar ubicado del lado derecho o izquierdo de nuestra página de ejemplo, la etiqueta no tiene una posición predefinida. El elemento `<aside>` solo describe la información que contiene, no el lugar dentro de la estructura. Este elemento puede estar ubicado en cualquier parte del diseño y ser usado siempre y cuando su contenido no sea considerado como el contenido principal del documento. Por ejemplo, podemos usar `<aside>` dentro del elemento `<section>` o incluso insertado entre la información relevante, como en el caso de una cita.

`<footer>` Para finalizar la construcción de la plantilla o estructura elemental de nuestro documento HTML5, solo necesitamos un elemento más. Ya contamos con la cabecera del cuerpo, secciones con ayuda para la navegación, información importante y hasta una barra lateral con datos adicionales, por lo tanto, lo único que nos queda por hacer es cerrar nuestro diseño para otorgarle un final al cuerpo del documento. HTML5 provee un elemento específico para este propósito llamado `<footer>`:

```

<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="iso-8859-1">
    <meta name="description" content="Ejemplo de HTML5">
    <meta name="keywords" content="HTML5, CSS3, JavaScript">
    <title>Este texto es el título del documento</title>
    <link rel="stylesheet" href="misestilos.css">
  </head>
  <body>
    <header>
      <h1>Este es el título principal del sitio web</h1>
    </header>
    <nav>
      <ul>
        <li>principal</li>
        <li>fotos</li>
        <li>videos</li>
        <li>contacto</li>
      </ul>
    </nav>
    <section>
    </section>
    <aside>
      <blockquote>Mensaje número uno</blockquote>
      <blockquote>Mensaje número dos</blockquote>
    </aside>
    <footer>
      Derechos Reservados &copy; 2018-2019
    </footer>
  </body>
</html>

```

En el típico diseño de una página web la sección llamada Institucional será definida por etiquetas <footer>.

Esto es debido a que la barra representa el final (o pie) del documento y esta parte de la página web es normalmente usada para compartir información general sobre el autor o la organización detrás del proyecto.

Generalmente, el elemento <footer> representará el final del cuerpo de nuestro documento y tendrá el propósito descrito anteriormente. Sin embargo, <footer> puede ser usado múltiples veces dentro del cuerpo para representar también el final de diferentes secciones (del mismo modo que la etiqueta <header>).

Dentro del contenido

El cuerpo de nuestro documento está listo. La estructura básica de nuestro sitio web fue finalizada, pero aún tenemos que trabajar en el contenido. Los elementos HTML5 estudiados hasta el momento nos ayudan a identificar cada sección del diseño y asignar un propósito

intrínseco a cada una de ellas, pero lo que es realmente importante para nuestro sitio web se encuentra en el interior de estas secciones.

La mayoría de los elementos ya estudiados fueron creados para construir una estructura para el documento HTML que pueda ser identificada y reconocida por los navegadores y nuevos dispositivos. Aprendimos acerca de la etiqueta <body> usada para declarar el cuerpo o parte visible del documento, la etiqueta <header> con la que agrupamos información importante para el cuerpo, la etiqueta <nav> que provee ayuda para la navegación del sitio web, la etiqueta <section> necesaria para contener la información más relevante, y también <aside> y <footer> para ofrecer información adicional de cada sección y del documento mismo. Pero ninguno de estos elementos declara algo acerca del contenido. Todos tienen un específico propósito estructural.

Cuanto más nos introducimos dentro del documento más cerca nos encontramos de la definición del contenido. Esta información estará compuesta por diferentes elementos visuales como títulos, textos, imágenes, videos y aplicaciones interactivas, entre otros. Necesitamos poder diferenciar estos elementos y establecer una relación entre ellos dentro de la estructura.

<article> El diseño considerado anteriormente es el más común y representa una estructura esencial para los sitios web estos días, pero es además ejemplo de cómo el contenido clave es mostrado en pantalla. Del mismo modo que los blogs están divididos en entradas, los sitios web normalmente presentan información relevante dividida en partes que comparten similares características.

El elemento <article> nos permite identificar cada una de estas partes:

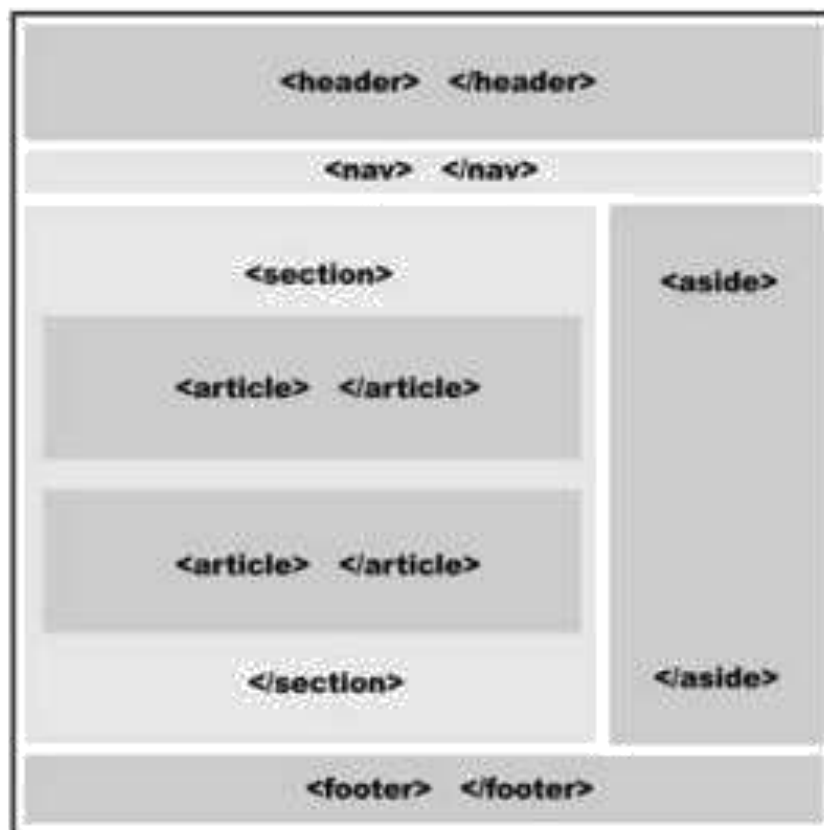
```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="iso-8859-1">
    <meta name="description" content="Ejemplo de HTML5">
    <meta name="keywords" content="HTML5, CSS3, JavaScript">
    <title>Este texto es el título del documento</title>
    <link rel="stylesheet" href="misestilos.css">
  </head>
  <body>
    <header>
      <h1>Este es el título principal del sitio web</h1>
    </header>
    <nav>
      <ul>
        <li>principal</li>
```

```
        <li>fotos</li>
        <li>videos</li>
        <li>contacto</li>
    </ul>
</nav>
<section>
    <article>
        Este es el texto de mi primer mensaje
    </article>
    <article>
        Este es el texto de mi segundo mensaje
    </article>
</section>
<aside>
    <blockquote>Mensaje número uno</blockquote>
    <blockquote>Mensaje número dos</blockquote>
</aside>
<footer>
    Derechos Reservados &copy; 2018-2019
</footer>
</body>
</html>
```

Como puede observarse en el código las etiquetas **<article>** se encuentran ubicadas dentro del elemento **<section>**. Las etiquetas **<article>** en nuestro ejemplo pertenecen a esta sección, son sus hijos, del mismo modo que cada elemento dentro de las etiquetas **<body>** es hijo del cuerpo. Y al igual que cada elemento hijo del cuerpo, las etiquetas **<article>** son ubicadas una sobre otra, como es mostrado en la siguiente figura.

Conceptos básicos: La estructura de un documento HTML puede ser descripta como un árbol, con el elemento **<html>** como su raíz. Otra forma de describir la relación entre elementos es nombrarlos como padres, hijos y hermanos, de acuerdo a la posición que ocupan dentro de esa misma estructura. Por ejemplo, en un típico documento HTML el elemento **<body>** es hijo del elemento **<html>** y hermano del elemento **<head>**. Ambos, **<body>** y **<head>**, tienen al elemento **<html>** como su padre.

En la imagen vemos la representación visual de las etiquetas `<article>` que fueron incluidas para contener información relevante de la página web.



El elemento `<article>` no está limitado por su nombre (no se limita, por ejemplo, a artículos de noticias). Este elemento fue creado con la intención de contener unidades independientes de contenido, por lo que puede incluir mensajes de foros, artículos de una revista digital, entradas de blog, comentarios de usuarios, etc... Lo que hace es agrupar porciones de información que están relacionadas entre sí independientemente de su naturaleza.

Como una parte independiente del documento, el contenido de cada elemento `<article>` tendrá su propia estructura.

Para definir esta estructura, podemos aprovechar la versatilidad de los elementos `<header>` y `<footer>` estudiados anteriormente. Estos elementos son portables y pueden ser usados no solo para definir los límites del cuerpo sino también en cualquier sección de nuestro documento:

```

<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="iso-8859-1">
    <meta name="description" content="Ejemplo de HTML5">
    <meta name="keywords" content="HTML5, CSS3, JavaScript">
    <title>Este texto es el título del documento</title>
    <link rel="stylesheet" href="misestilos.css">
  </head>
  <body>
    <header>
      <h1>Este es el título principal del sitio web</h1>
    </header>
    <nav>
      <ul>
        <li>principal</li>
        <li>fotos</li>
        <li>videos</li>
        <li>contacto</li>
      </ul>
    </nav>
    <section>
      <article>
        <header>
          <h1>Título del mensaje uno</h1>
        </header>
        Este es el texto del primer mensaje
        <footer>
          <p>comentarios (0)</p>
        </footer>
      </article>
      <article>
        <header>
          <h1>Titulo del mensaje dos</h1>
        </header>
        Este es el texto de mi segundo mensaje
        <footer>
          <p>comentarios (0)</p>
        </footer>
      </article>
    </section>
    <aside>
      <blockquote>Mensaje número uno</blockquote>
      <blockquote>Mensaje número dos</blockquote>
    </aside>
    <footer>
      Derechos Reservados &copy; 2018-2019
    </footer>
  </body>
</html>

```

Los dos mensajes insertados en el código han sido contruidos con el elemento <article> y tienen una estructura específica. En la parte superior de esta estructura incluimos las etiquetas <header> conteniendo el título definido con el elemento <h1>, debajo se encuentra el contenido mismo del mensaje y sobre el final, luego del texto, vienen las etiquetas <footer> especificando la cantidad de comentarios recibidos.

`<hgroup>` Dentro de cada elemento `<header>`, en la parte superior del cuerpo o al comienzo de cada `<article>`, incorporamos elementos `<h1>` para declarar un título. Básicamente, las etiquetas `<h1>` son todo lo que necesitamos para crear una línea de cabecera para cada parte del documento, pero es normal que necesitemos también agregar subtítulos o más información que especifique de qué se trata la página web o una sección en particular. De hecho, el elemento `<header>` fue creado para contener también otros elementos como tablas de contenido, formularios de búsqueda o textos cortos y logos.

Para construir este tipo de cabeceras, podemos aprovechar el resto de las etiquetas H, como `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` y `<h6>`, pero siempre considerando que por propósitos de procesamiento interno, y para evitar generar múltiples secciones durante la interpretación del documento por parte del navegador, estas etiquetas deben ser agrupadas juntas. Por esta razón, HTML5 provee el elemento `<hgroup>`:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="iso-8859-1">
    <meta name="description" content="Ejemplo de HTML5">
    <meta name="keywords" content="HTML5, CSS3, JavaScript">
    <title>Este texto es el título del documento</title>
    <link rel="stylesheet" href="misestilos.css">
  </head>
  <body>
    <header>
      <h1>Este es el título principal del sitio web</h1>
    </header>
    <nav>
      <ul>
        <li>principal</li>
        <li>fotos</li>
        <li>videos</li>
        <li>contacto</li>
      </ul>
    </nav>
    <section>
      <article>
        <header>
          <hgroup>
            <h1>Título del mensaje uno</h1>
            <h2>Subtítulo mensaje uno</h2>
          </hgroup>
          <p>publicado 10-12-2011</p>
        </header>
        Este es el texto de mi primer mensaje
        <footer>
          <p>comentarios (0)</p>
        </footer>
      </article>
      <article>
        <header>
          <hgroup>
            <h1>Título del mensaje dos</h1>
            <h2>Subtítulo mensaje dos</h2>
          </hgroup>
          <p>publicado 15-12-2017</p>
        </header>
      </article>
    </section>
  </body>
</html>
```

```

        </header>
        Este es el texto de mi segundo mensaje
    <footer>
        <p>comentarios (0)</p>
    </footer>
</article>
</section>
<aside>
    <blockquote>Mensaje número uno</blockquote>
    <blockquote>Mensaje número dos</blockquote>
</aside>
<footer>
    Derechos Reservados &copy; 2018-2019
</footer>
</body>
</html>

```

Las etiquetas H deben conservar su jerarquía, lo que significa que debemos primero declarar la etiqueta <h1>, luego usar <h2> para subtítulos y así sucesivamente. Sin embargo, a diferencia de anteriores versiones de HTML, HTML5 nos deja reusar las etiquetas H y construir esta jerarquía una y otra vez en cada sección del documento. En el ejemplo agregamos un subtítulo y datos adicionales a cada mensaje. Los títulos y subtítulos fueron agrupados juntos utilizando <hgroup>, recreando de este modo la jerarquía <h1> y <h2> en cada elemento <article>.

IMPORTANTE: El elemento <hgroup> es necesario cuando tenemos un título y subtítulo o más etiquetas H juntas en la misma cabecera. Este elemento puede contener solo etiquetas H y esta fue la razón por la que en nuestro ejemplo dejamos los datos adicionales afuera. **Si solo dispone de una etiqueta <h1> o la etiqueta <h1> junto con datos adicionales, no tiene que agrupar estos elementos juntos.** Por ejemplo, en la cabecera del cuerpo (<header>) no usamos este elemento porque solo tenemos una etiqueta H en su interior. Siempre recuerde que <hgroup> fue creado solo con la intención de agrupar etiquetas H, exactamente como su nombre lo indica.

Navegadores y programas que ejecutan y presentan en la pantalla sitios webs leen el código HTML y crean su propia estructura interna para interpretar y procesar cada elemento. Esta estructura interna está dividida en secciones que no tienen nada que ver con las divisiones en el diseño o el elemento <section>. Estas son secciones conceptuales generadas durante la interpretación del código. El elemento <header> no crea una de estas secciones por sí mismo, lo que significa que los elementos dentro de <header> representarán diferentes niveles e internamente pueden generar diferentes secciones. El elemento <hgroup> fue creado con el propósito de agrupar las etiquetas H y evitar interpretaciones incorrectas por parte de los navegadores.

Conceptos básicos: lo que llamamos “**información adicional**” dentro de la cabecera en nuestra descripción previa es conocido como **Metadata**. Metadata es un conjunto de datos que describen y proveen información acerca de otro grupo de datos. En nuestro ejemplo, Metadata es la fecha en la cual cada mensaje fue publicado.

<figure> y <figcaption> La etiqueta **<figure>** fue creada para ayudarnos a ser aún más específicos a la hora de declarar el contenido del documento. Antes de que este elemento fuera introducido, no podíamos identificar el contenido que era parte de la información pero a la vez independiente, como ilustraciones, fotos, videos, etc... Normalmente estos elementos son parte del contenido relevante pero pueden ser extraídos o movidos a otra parte sin afectar o interrumpir el flujo del documento. Cuando nos encontramos con esta clase de información, las etiquetas <figure> pueden ser usadas para identificarla:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="iso-8859-1">
    <meta name="description" content="Ejemplo de HTML5">
    <meta name="keywords" content="HTML5, CSS3, JavaScript">
    <title>Este texto es el título del documento</title>
    <link rel="stylesheet" href="misestilos.css">
  </head>
  <body>
    <header>
      <h1>Este es el título principal del sitio web</h1>
    </header>
    <nav>
      <ul>
        <li>principal</li>
        <li>fotos</li>
        <li>videos</li>
        <li>contacto</li>
      </ul>
    </nav>
    <section>
      <article>
        <header>
          <hgroup>
            <h1>Título del mensaje uno</h1>
            <h2>Subtítulo del mensaje uno</h2>
          </hgroup>
          <p>publicado 10-12-2017</p>
        </header>
        Este es el texto de mi primer mensaje
        <figure>
          
          <figcaption>
            Esta es la imagen del primer mensaje
          </figcaption>
        </figure>
        <footer>
          <p>comentarios (0)</p>
        </footer>
      </article>
    </section>
  </body>
</html>
```

```

        <header>
            <hgroup>
                <h1>Título del mensaje dos</h1>
                <h2>Subtítulo del mensaje dos</h2>
            </hgroup>
            <p>publicado 15-12-2017</p>
        </header>
        Este es el texto de mi segundo mensaje <footer>
        <p>comentarios (0)</p>
    </footer>
</article>
</section>
<aside>
    <blockquote>Mensaje número uno</blockquote>
    <blockquote>Mensaje número dos</blockquote>
</aside>
<footer>
    Derechos Reservados &copy; 2017-2018 </footer>
</body>
</html>

```

En el código, en el primer mensaje, luego del texto insertamos una imagen (``). Esta es una práctica común, a menudo el texto es enriquecido con imágenes o videos. Las etiquetas `<figure>` nos permiten envolver estos complementos visuales y diferenciarlos así de la información más relevante. También en el código se puede observar un elemento extra dentro de `<figure>`. Normalmente, unidades de información como imágenes o videos son descriptas con un corto texto debajo. HTML5 provee un elemento para ubicar e identificar estos títulos descriptivos. Las etiquetas `<figcaption>` encierran el texto relacionado con `<figure>` y establecen una relación entre ambos elementos y su contenido.

COLOR

Codificación de colores

Los colores en HTML son contruidos utilizando una combinación de tres colores básicos: **rojo**, **verde** y **azul**. Cada uno de estos colores puede tomar un valor **desde 0 a 255** y son representados **en notación hexadecimal (00 a FF)**. Dicho esto, un color está compuesto por seis caracteres donde los dos primeros representan al color rojo, los dos segundos al verde y los dos terceros al azul. Por ejemplo, el color rojo está formado en HTML con "FF" para el rojo (los dos primeros caracteres) y "00" para el resto: "FF0000".

Al mezclar estos valores se mezclarán los colores resultantes, de modo que puedes crear un amarillo al mezclar el rojo y el verde (por ejemplo, "FFFF00"), violeta al mezclar rojo y azul (por ejemplo, "FF00FF") ó cyan al mezclar verde y azul (por ejemplo, "00FFFF"). Esto funciona exactamente como la paleta de un pintor.

La escala cromática de intensidad de color es:

- Mínima (nulo = 00)
- Media (mediano = 80)
- Máxima (saturado = FF)

El tono del color puede también ser cambiado incrementando (por ejemplo, violeta claro "FF66FF") o disminuyendo (por ejemplo, violeta oscuro "AA00AA") los tres valores proporcionalmente.

Un valor de color puede ser o bien un número hexadecimal (anteponiendo un signo "#") o uno de los siguientes dieciséis nombres de colores. En los nombres de colores no se distingue entre mayúsculas y minúsculas. A continuación se listan los colores básicos en HTML:

			
Aqua #00FFFF	Black #000000	Blue #0000FF	Fuchsia #FF00FF
			
Gray #808080	Green #008000	Lime #00FF00	Maroon #800000
			
Navy #000080	Olive #808000	Purple #800080	Red #FF0000
			
Silver #C0C0C0	Teal #008080	White #FFFFFF	Yellow #FFFF00

TEXTO

La mayor parte del contenido de las páginas HTML habituales está formado por texto, llegando a ser más del 90% del código de la página.

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

La tarea inicial del editor de contenidos HTML consiste en estructurar el texto original definiendo sus párrafos, titulares y títulos de sección, como se muestra en la siguiente imagen:



El proceso de estructurar un texto simple consiste en indicar las diferentes zonas o secciones que componen el texto. De esta forma, los textos estructurados utilizan etiquetas para delimitar cada párrafo y títulos de sección para delimitar cada una de las secciones que forman el texto.

Una vez definida la estructura básica de los contenidos de la página, el siguiente paso consiste en marcar los diferentes elementos dentro del propio texto: definiciones, abreviaturas, textos importantes, textos modificados, citas a otras referencias, etc.



El anterior ejemplo muestra la transformación de un párrafo con un texto simple en un párrafo cuyo texto contiene elementos marcados de forma especial. Así, algunas palabras del texto se muestran en negrita porque se consideran importantes; otras palabras aparecen en cursiva, ya que se han marcado como destacadas e incluso una frase aparece tabulada y entre comillas, indicando que es una cita textual de otro contenido.

En las secciones siguientes se muestran todas las etiquetas que define HTML para estructurar y marcar el texto.

Encabezados, jerarquía y estructura del contenido de un documento.

Las páginas HTML habituales suelen tener una estructura más compleja que la que se puede crear solamente mediante párrafos. De hecho, es habitual que las páginas se dividan en diferentes secciones jerárquicas.

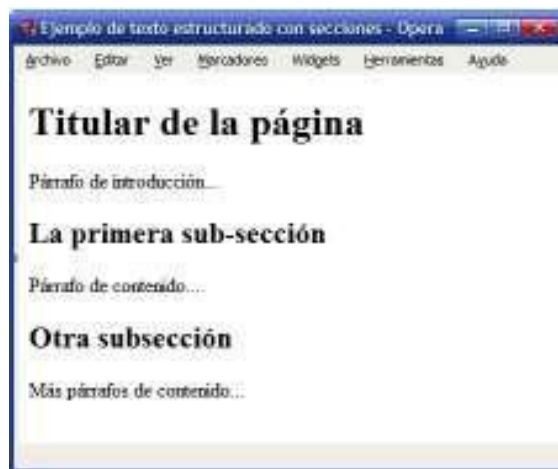
Los títulos de sección se utilizan para delimitar el comienzo de cada sección de la página. HTML permite crear secciones de hasta seis niveles de importancia. De esta forma, aunque una página puede definir cualquier número de secciones, sólo puede incluir seis niveles jerárquicos.

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

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

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

```
<html>
  <head>
    <title>Ejemplo de texto estructurado con secciones</title>
  </head>
  <body>
    <h1>Titular de la página</h1>
    <p>Párrafo de introducción...</p>
    <h2>La primera sub-sección</h2>
    <p>Párrafo de contenido...</p>
    <h2>Otra subsección</h2>
    <p>Más párrafos de contenido...</p>
  </body>
</html>
```

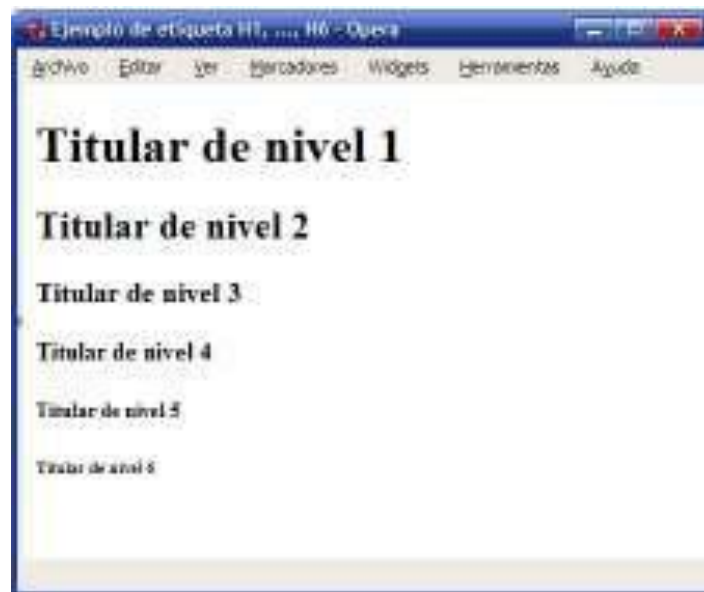


Los navegadores muestran el ejemplo anterior de la siguiente manera:

Los navegadores asignan de forma automáticamente el tamaño del título de cada sección en función de su importancia. Así, los títulos de sección <h1> se muestran con el tamaño de letra más grande, ya que son el nivel jerárquico superior, mientras que los títulos de sección <h6> se visualizan con un tamaño de letra muy pequeño, adecuado para el nivel jerárquico de menor importancia.

Evidentemente, el aspecto que los navegadores aplican por defecto a los títulos de sección se puede modificar utilizando las hojas de estilos de CSS.

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



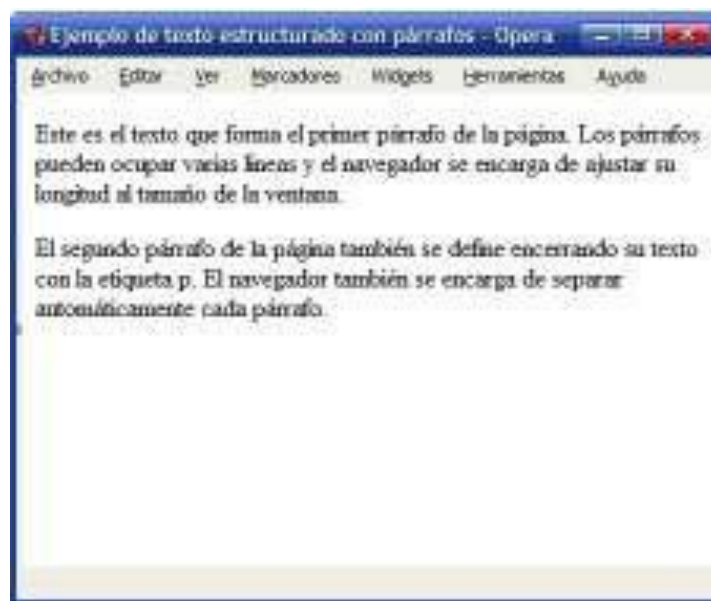
Ejemplo de uso de las etiquetas h1, h2, h3, h4, h5 y h6

Párrafos

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

```
<html>
  <head>
    <title>Ejemplo de texto estructurado con párrafos</title>
  </head>
  <body>
    <p>Este es el texto que forma el primer párrafo de la página.
    Los párrafos pueden ocupar varias líneas y el navegador se
    encarga de ajustar su longitud al tamaño de la ventana.</p>
    <p>El segundo párrafo de la página también se define encerrando
    su texto con la etiqueta p. El navegador también se encarga de
    separar automáticamente cada párrafo.</p>
  </body>
</html>
```

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



Los párrafos creados con HTML son elementos de bloque, por lo que siempre ocupan toda la anchura de la ventana del navegador. Además, no tienen atributos específicos, pero sí que se les pueden asignar los atributos comunes de HTML básicos, de internacionalización y de eventos.

Alineación, espaciado y sangrado de texto

La etiqueta <p> admite el atributo "ALIGN=" con los valores "left", "right", "center" y "justify" para alinearlo a la izquierda, a la derecha, centrarlo o justificarlo totalmente. Por defecto está alineado a la izquierda.

<p> Se utiliza para que los párrafos queden separados por una línea en blanco. **Si solo quieres escribir un punto y aparte debes usar
**. Se pueden emplear tantas etiquetas
 como saltos de línea se quieran añadir.

`<pre>` La etiqueta `<pre>` la puedes utilizar para que el navegador interprete el texto escrito tal y como está.

Como ejemplo es mostraremos este texto:

```
Escribo esta línea así
```

```
Dejo dos líneas de separación
```

```
y escribo tres más.
```

Sin poner ninguna etiqueta, el navegador respondería así:

```
"Escribo esta línea así Dejo dos líneas de separación y escribo tres más."
```

En cambio **utilizando la etiqueta `<pre>` y cerrándola con su correspondiente etiqueta, el texto se vería como queremos.**

Hoy en día lo más usual es utilizar otras etiquetas como `div` o `span` para aplicar formato al texto usando hojas de estilo que permiten un ajuste mucho más fino y mayores posibilidades de formateo.

Separadores de texto

`<hr>` Para separar un texto de otro o un párrafo de otro podemos utilizar una línea horizontal de un tamaño o un grosor determinado por nosotros. Esa franja la escribimos con la etiqueta `<hr>`. La contrario que muchas etiquetas html, ésta no necesita ser cerrada. Sólo con escribir la etiqueta anterior ya la escribimos.

La etiqueta `<hr>`, como muchas otras etiquetas, puede variar de aspecto dependiendo de una serie de caracteres o parámetros que podemos predefinir. Por ejemplo, podemos definir su grosor mediante el parámetro `size`.

Para escribir este parámetro en la etiqueta escribiremos “size=x”, siendo “x” el número del grosor de la franja. A continuación te vamos a ofrecer dos ejemplos de franjas con diferentes grosores, siendo la primera 10 y la segunda dos. La diferencia es abismal

```
<hr size=20 /> <hr size=2 />
```

Y el resultado sería el siguiente:



Otro parámetro que podemos definir es la anchura de la franja mediante el parámetro `width`. El parámetro será “width=x %”, siendo “x” el tanto por cien que queremos que ocupe nuestra franja.

En el caso de no escribir nada (como en los ejemplos anteriores), **la franja será predeterminada del 100%**. Si queremos que ocupe más o menos tendremos que escribirlo con el parámetro `width`.

A continuación vamos a escribir una franja de 75% de ancho:

```
<hr width=75% />
```

Que quedaría de la siguiente manera:



Muchos navegadores hacen esta franja con una sombra exterior que transforma la franja en tres dimensiones. Si quieres que la franja sea simple, **sin sombrita deberás escribir el parámetro “noshade”**.

También podemos predefinir el color que le queremos dar a la franja con el parámetro “`color`”. Es muy sencillo. Por ejemplo, si queremos que nuestra franja sea de color rojo sólo debemos ponerle la etiqueta:

```
<hr color=#FF000 />
```

Por último, puedes hacer que la franja quede alineada a un lado, a otro, o al centro del párrafo mediante el parámetro que ya vimos con anterioridad: “`align`”. “align = center” para el centro, “align = right” para la derecha y “align = left” para la izquierda.

Como ejemplos finales vamos a hacer dos franjas. La primera va a ser una franja de grosor 3, de un ancho del 50% y alineada al centro.

```
<hr size=3 width=50% align=center />
```

La siguiente va a ser una franja de grosor 2, de ancho 80%, sin sombra y alineada a la derecha.

```
<hr size=2 width=80% noshade=noshade align=right />
```

Etiquetas propias de bloques de texto.

<abbr> Pensada para etiquetar abreviaturas. El significado de la abreviatura debe escribirse mediante el atributo **title**.

```
<p>Sitúa el cursor del ratón sobre la abreviatura <abbr title="cónfer (compárese, véase)">cf</abbr> para ver su significado.</p>
```

Sitúa el cursor del ratón sobre la abreviatura **cf** para ver su significado.

cónfer (compárese, véase)

<acronym> Pensada para etiquetar acrónimos (siglas que se pronuncian como palabras). El significado del acrónimo debe escribirse mediante el atributo **title**.

```
<p>Sitúa el cursor del ratón sobre el acrónimo <acronym title="Rogamos Empujen Nuestros Ferrocarriles Estropeados (es broma)">RENFE</acronym> para ver significado.</p>
```

Sitúa el cursor del ratón sobre el acrónimo **RENFE** para ver su significado.

Rogamos Empujen Nuestros Ferrocarriles Estropeados (esnbroma)

<cite> Pensada para identificar una cita o referencia a otras fuentes. Ahora las etiquetas **<cite>** encierran el título de un trabajo, como un libro, una película, una canción, etc... Los navegadores suelen mostrar la etiqueta **<cite>** en cursiva.

```
<p>Como escribió Robert Browning, <cite>The best is yet to be</cite>.</p>
```

Como escribió Robert Browning, *The best is yet to be*.

<var> Pensada para identificar una instancia de una variable o de un argumento de programa. Los navegadores suelen mostrar la etiqueta **<var>** en cursiva.

```
<p><var>$saludo</var> = "Hola, mundo";</p>
```

\$saludo = "Hola, mundo";

<code> Pensada para etiquetar fragmento de código de ordenador. Los navegadores suelen mostrar la etiqueta **<code>** en tipo de letra monospace (normalmente Courier).

```
<p>El primer programa que se enseña en un lenguaje de programación  
suele ser algo así como: <br> <code>main() { printf ("Hola, mundo");  
}</code></p>
```

El primer programa que se enseña en un lenguaje de programación suele ser algo así como:
main() { printf ("Hola, mundo"); }

<samp> Pensada para identificar un ejemplo de la salida de un programa o de un script. Los navegadores suelen destacar la etiqueta **<samp>** cambiando el tipo de letra.

```
<p>Al ejecutar tu primer programa, en la pantalla podrás leer  
<samp>Hola, mundo</samp>.</p>
```

Al ejecutar tu primer programa, en la pantalla podrás leer Hola, mundo.

**** Pensada, junto con la etiqueta **<ins>**, para etiquetar modificaciones en un texto. Esta etiqueta debería etiquetar el texto que se ha eliminado de una página al revisarlo (si se quiere que se sepa que se ha eliminado, claro). Los navegadores suelen mostrar la etiqueta **** tachada.

```
<p>Internet Explorer es un navegador bastante <del>malo</del>  
mejorable.</p>
```

Internet Explorer es un navegador bastante ~~malo~~ mejorable.

<ins> Pensada, junto con la etiqueta ****, para etiquetar modificaciones en un texto. Esta etiqueta debería etiquetar el texto que se ha añadido a una página al revisarlo (si se quiere que se sepa que se ha añadido, claro). Los navegadores suelen mostrar la etiqueta **<ins>** subrayada.

<p>La verdad es que Microsoft podría mejorar su navegador<ins>, respetando las recomendaciones del W3C</ins>.</p>

La verdad es que Microsoft podría mejorar su navegador, respetando las recomendaciones del W3C.

<dfn> Pensada para identificar la primera aparición de un término en un texto. Los navegadores suelen mostrar la etiqueta <dfn> en cursiva.

<p>En 2005 se empezaron a comercializar discos duros que empleaban la <dfn>grabación magnética perpendicular</dfn> para aumentar la densidad de información.</p>

En 2005 se empezaron a comercializar discos duros que empleaban la *grabación magnética perpendicular* para aumentar la densidad de información.

 Pensada para resaltar una porción de texto dándole énfasis, aunque no tanto como con la etiqueta . Los navegadores suelen mostrar la etiqueta en cursiva.

<p>Es la última vez que te lo digo.</p>

Es la *última* vez que te lo digo.

<kbd> Pensada para identificar el texto que debe teclear el usuario. Los navegadores suelen destacar la etiqueta <kbd> cambiando el tipo de letra.

<p>En Amaya, para seleccionar un elemento, debes pulsar la tecla <kbd>F2</kbd>.</p>

En Amaya, para seleccionar un elemento, debes pulsar la tecla F2.

 Pensada para resaltar una porción de texto dándole énfasis, aún más que con la etiqueta . Los navegadores suelen mostrar la etiqueta en negrita.

<p>Es la última vez que te lo digo.</p>

Es la **última** vez que te lo digo.

Etiquetas de estilos lógicos aplicadas a un conjunto de caracteres

Estas etiquetas están en su mayoría desaconsejadas, ya que pueden conseguirse sus efectos utilizando las propiedades correspondientes de las hojas de estilo.

** Desaconsejada.** Originalmente pensada para etiquetar texto en negrita. Los navegadores suelen mostrar la etiqueta en negrita.

```
<p>Esta palabra está <b>resaltada</b>.</p>
```

Esta palabra está **resaltada**.

<bdo> Pensada para elegir la dirección del texto (de izquierda a derecha o de derecha a izquierda). Es obligatorio especificar el atributo **dir**, con el valor **rtl** (de derecha a izquierda) o **ltr** (de izquierda a derecha).

```
<p><bdo dir="rtl">Esta frase está escrita al derecho, pero debería leerse al revés</bdo>.</p>
```

séver la esreel aÍrebed orep ,ohcered la atircse átse esarf atsE.

<big> Desaconsejada. Originalmente pensada para etiquetar texto de mayor tamaño. Los navegadores suelen mostrar la etiqueta <big> con la propiedad font-size: larger.

```
<p>Esta palabra está <big>aumentada</big>.</p>
```

Esta palabra está **aumentada**.

La etiqueta <big> no está incluida en HTML5. En su lugar se recomienda utilizar propiedades CSS.

<i> Desaconsejada. Originalmente pensada para etiquetar texto en itálica. Los navegadores suelen mostrar la etiqueta <i> con la propiedad font-style: italic.

```
<p>Esta palabra está en <i>itálica</i>.</p>
```

Esta palabra está en *itálica*.

<q> Pensada para identificar una cita o referencia a otras fuentes. La recomendación HTML 4.0 especifica que los navegadores deben añadir automáticamente comillas al texto marcado.

<p>Como escribió Robert Browning, <q>The best is yet to be</q>.</p>

Como escribió Robert Browning, "The best is yet to be".

<small> Desaconsejada. Originalmente pensada para etiquetar texto de menor tamaño. Los navegadores suelen mostrar la etiqueta **<small>** con la propiedad font-size: smaller.

<p>Esta palabra está <small>reducida</small>.</p>

Esta palabra está reducida.

<sub> Pensada para identificar texto en subíndice.

<p>2H₂ + O₂ → 2H₂O</p>

$2\text{H}_2 + \text{O}_2 \rightarrow 2\text{H}_2\text{O}$

<sup> Pensada para identificar texto en superíndice.

<p>El último teorema de Fermat dice que la ecuación $x^{\text{ⁿ}} + y^{\text{ⁿ}} = z^{\text{ⁿ}}$ no tiene soluciones enteras para <var>x</var> , <var>y</var> y <var>z</var> cuando $\text{<var>n</var>} > 2$.</p>

El último teorema de Fermat dice que la ecuación $x^n + y^n = z^n$ no tiene soluciones enteras para x, y y z cuando $n > 2$.

<tt> Desaconsejada. Originalmente pensada para etiquetar texto de fuente de espaciado fijo. Los navegadores suelen mostrar la etiqueta **<tt>** en tipo de letra monospace (normalmente Courier).

<p>Estas palabras tienen un <tt>espaciado fijo</tt>.</p>

Estas palabras tienen un espaciado fijo.

La etiqueta **<tt>** no está incluida en HTML5. En su lugar se recomienda utilizar la etiqueta **<code>** o propiedades CSS.

ENLACES DE HIPERTEXTO

Las principales etiquetas para enlaces son:

`<a> ...`, con atributos básicos, de internacionalización, de eventos y de foco, Además, cuenta con los siguientes atributos específicos:

- `href="url"` indica la dirección a la que apunta el enlace. (Esta dirección puede ser relativa o absoluta si es del servidor, si no, en general se puede escribir mediante el protocolo y la dirección completa.)
- `name="valor"`, pone un nombre a la etiqueta y crea anclas a las que otro enlace puede hacer referencia usando `#valor` como valor del atributo `href`.
- `rel="..."`, indica la relación de la página actual con la página enlazada.
- `rev="..."`, indica la relación de la página enlazada con la actual.
- `target="..."`, señala el nombre de la ventana o el marco donde se mostrará el documento apuntado por el enlace.
- `hreflang="..."`, especifica el código de internacionalización de la página enlazada.

Los atributos `rel` y `rev` pueden tomar los valores: `alternate`, `start`, `next`, `prev`, `chapter`, `help`, etc.

El valor de `target` puede ser el nombre de una ventana o marco, o bien, `_blank`, si se quiere abrir una ventana nueva o `_self` (valor por defecto), si queremos visualizar el documento en el mismo marco o ventana. Si se pone un nombre que no existe abre una ventana nueva.

La etiqueta `<a>` es de tipo en línea.

`<script> ...</script>`, sólo tiene atributos específicos, que son:

- `src="..."`, indica la URL de un archivo externo que contiene el código del script.
- `type="text/javascript"`, especifica el tipo MIME (tipo de archivo/contenido).
- `defer = "defer"`, indica al navegador que no pare la carga de la página por la ejecución del script.

`<link>` establece una relación entre el documento y otros recursos que descarga. Esta etiqueta sólo puede aparecer en la cabecera del documento. Tiene atributos básicos, de

internacionalización, de eventos (aunque no se pueden usar al estar en la cabecera) y los específicos siguientes:

- `href="url"` del documento a enlazar.
- `type = "..."` indica el tipo (tipo MIME) del documento enlazado. (Ver la lista de tipos MIME en <http://www.htmlquick.com/es/reference/mime-types.html>)
- `rel="..."` indica la relación entre el documento actual y el enlazado
- `rev="..."` indica la relación entre el documento enlazado y el actual
- `target="..."` indica dónde se presentará el documento enlazado
- `hreflang="..."` especifica el código de internacionalización del documento enlazado.

Ejemplos:

```
<link rel="stylesheet" type="text/css" href="estilos/llamativo.css"/>
<link rel="shortcut icon" type="image/ico" href="favicon.ico" />
```

Estructura de un enlace: la dirección de internet o url

Un enlace se puede definir sobre un texto, una imagen, un elemento de una lista, etc. y hace referencia a una URL concreta. Ésta se compone de las siguientes partes:

Protocolo	Servidor	[Ruta]	[Archivo]	[Consulta]	[Secciones]
http:// https:// ftp:// file://	nombre.subdominio... dominio	Camino de directorios	nombre.extensión	?variable=valor	#ID (ID es el lugar concreto dentro de la página web)

Los caracteres que aparecen en los enlaces deben poder expresarse en [ASCII](#). Sin embargo, en ocasiones, las URL contienen caracteres que no se pueden representar mediante este código y, por tanto, éstos deben ser convertidos. La llamada codificación URL convierte una URL en un formato ASCII válido.

La codificación URL sustituye estos caracteres especiales por un "%" seguido por los dos [dígitos hexadecimales correspondientes al código ISO-8859-1](#). Como las URL no pueden contener [espacios en blanco](#), se sustituyen por el signo +.

He aquí la codificación de algunos de ellos:

/	?	@	~	ñ	Ñ	á	Ç
%2F	%3F	%40	%7E	%F1	%D1	%E1	%E7

Enlaces internos

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

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

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

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

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

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

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

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

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

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

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

```
<a href="#inicio">Volver al inicio de la página</a>...  
<p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia a, convallis eu, nonummy et, odio. Aenean urna elit, ultrices id, placerat varius, facilisis eget, dolor.</p>
```

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

Los enlaces directos a secciones también funcionan con el atributo `id` de cualquier elemento.

El siguiente ejemplo es equivalente al ejemplo anterior:

```
<h1 id="inicio">Título de la página</h1>  
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id ligula eu felis adipiscing ultrices. Duis gravida leo ut lectus. Praesent condimentum mattis ligula.</p>
```

```
<a href="#inicio">Volver al inicio de la página</a>...  
<p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia a, convallis eu, nonummy et, odio. Aenean urna elit, ultrices id, placerat varius, facilisis eget, dolor.</p>
```

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

El nombre de la sección que se indica después del símbolo `#` puede utilizar el valor de los atributos `id` de cualquier elemento. De hecho, **se recomienda utilizar los atributos `id` de los elementos ya existentes en la página en vez de crear "enlaces vacíos" de tipo ``.**

Enlaces especiales: correo electrónico. Enlaces de descarga.

Enlaces de correo electrónico.

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

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

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

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

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

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

Todas las opciones anteriores se pueden combinar entre sí para realizar ejemplos más avanzados. **Aunque el uso de mailto: puede parecer una ventaja, su uso está desaconsejado.**

Si se incluye una dirección de correo electrónico directamente en una página web, es muy probable que en poco tiempo esa dirección de email se encuentre llena de correo electrónico basura o "spam", ya que existen programas automáticos encargados de rastrear sistemáticamente todas las páginas web de Internet para encontrar direcciones de correo electrónico válidas.

Enlaces de descarga

Para enlazar un archivo almacenado en un servidor FTP, la parte del protocolo de la URL debe cambiar de http:// a ftp://.

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

IMÁGENES

En un sitio web, se pueden usar imágenes de contenido e imágenes de adorno y diseño. Las primeras deben incluirse mediante HTML y las segundas, en la medida de lo posible, con hojas de estilo CSS.

Los formatos de imagen más estandarizados en páginas web son JPG, PNG y GIF.

Existen diferentes etiquetas que permiten insertar imágenes y multimedia:

``, para insertar una imagen. Posee atributos básicos, de internacionalización, de eventos y los siguientes específicos:

- `src="url"`, para indicar la localización de la imagen.
- `alt="texto descriptivo"`, texto que aparece si no se carga la imagen.
- `title="texto descriptivo"`, título de la imagen.
- `width="valor"`, especifica la anchura de la imagen en píxeles o porcentaje.
- `height="valor"`, especifica la altura de la imagen en píxeles o porcentaje.
- `usemap="#nedelmap"`, mapa de imágenes para la imagen.

Es importante tener en cuenta que las imágenes tienen un comportamiento en línea y que **es obligatorio incluir los atributos `src` y `alt` cuando se crea una imagen**.

`<map>`, para crear un mapa de imágenes. Cuenta con atributos básicos, de internacionalización y de eventos y puede ser en línea o de bloque. Se usa junto a la etiqueta `<area>`.

`<area>`, para definir los distintos trozos del mapa. Cuenta con atributos básicos, de internacionalización, de eventos y los siguientes atributos específicos:

- `shape="..."`, para indicar la forma geométrica del trozo (`rect`, `circle`, `poly`).
- `coords="x..."`, para señalar las coordenadas del trozo de imagen.
- `href="url"`, especifica la dirección enlazada desde el trozo de imagen.
- `target="..."`, indica dónde se presentará el contenido apuntado en la url.

Ejemplo:

```

  <map id="dibujo" name="dibujo">
    <area shape="rect" coords="x1, y1, x2, y2"/>
    <area shape="circle" coords="x, y, r"/>
    <area shape="poly" coords="x1, y1, x2, y2, x3, y3,..."/>
  </map>
```

Aparte de éstas, en HTML existían, para multimedia, diversos tipos de etiquetas que, posteriormente, se estandarizaron en una sola, llamada `<object>`. Se trata de una etiqueta contenedora de objetos (video, audio, applets Java, Activex, PDF y Flash), la cual incluye otra, llamada `<param/>`, que permite pasar los parámetros adecuados al objeto correspondiente. Es de tipo bloque.

Ejemplo:

```
<object classid="CLSID:xxxxxxxxxx" width=800 height=600 type="video/wmv" />
  <param name="FileName" value="archive.wmv">
  <param name="autoStart" value="true">
</object>
```

La etiqueta <object> tenía la intención de sustituir los elementos y <applet>. Sin embargo, a causa de errores y la falta de soporte de navegadores, esto no ha sucedido. El soporte para la etiqueta en los navegadores depende del tipo de objeto, y lamentablemente, los principales navegadores utilizan códigos diferentes para cargar el mismo tipo de objeto.

Formatos de imágenes

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

La recomendación es utilizar uno de los tres siguientes formatos gráficos que entienden todos los navegadores modernos: **GIF, JPG y PNG**. El formato PNG presenta el inconveniente de que los navegadores obsoletos como Internet Explorer 6 no muestran correctamente las imágenes con transparencias de 24 bits.

Características de imágenes: tamaño, título, textos alternativos

Tamaño

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

Sin embargo, en la práctica no es viable establecer la anchura y altura de todas las imágenes de contenidos mediante CSS. Si el sitio web dispone de muchas imágenes, la sobrecarga de estilos diferentes que debería definir CSS sería contraproducente. Por este motivo, los

atributos `width` y `height` son la excepción a la norma de que el código HTML no haga referencia al aspecto de los contenidos.

Ejemplo:

```



```

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

También **es posible indicar la anchura y altura en forma de porcentaje.** En este caso, el porcentaje hace referencia a la altura/anchura del elemento en el que está contenida la imagen.

Si la imagen no se encuentra dentro de ningún otro elemento, hace referencia a la anchura/altura total de la página.

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

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

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

Si solamente se establece la altura de la imagen, el navegador calcula la anchura necesaria para que se mantenga la proporción de la imagen. De la misma forma, si sólo se establece la anchura de la imagen, el navegador calcula la altura que hace que la imagen se siga viendo con las mismas proporciones.

Titulo

EL atributo `title` permite añadir información a la imagen. Si se pasa el ratón por encima de la imagen y se deja un momento se podrá leer en un cuadro de texto el contenido de este atributo.

Texto alternativo

El atributo `alt` permite describir el contenido de la imagen mediante un texto breve. Las descripciones deben tener una **longitud inferior a 1024 caracteres** y son útiles para las personas y dispositivos discapacitados que no pueden acceder a las imágenes.

Enlaces en imágenes

Con frecuencia se emplean imágenes como enlaces. A veces son enlaces a las mismas imágenes con más resolución, como en el caso de fotos. Otras veces nos llevan a una localización cualquiera.

Para que una imagen sea un enlace hay que colocarla en el lugar del texto del enlace, si además queremos que vaya a otra imagen, ésta la debemos poner en el lugar de la URL.

Veamos unos ejemplos Supongamos para ello, que la estructura de nuestras páginas es la indicada en la imagen. Si escribo en la página actual localizada en Aprendiendo Html:



```
<a href="#inicio">  
      
</a>  
obtengo una imagen que al pulsarla me lleva al inicio de esta página.  
Para que una imagen sea un enlace a otra imagen se debe escribir:  
<a href="gifs/sky.jpg">  
      
</a>
```

Imágenes de fondo

Se puede poner una imagen de fondo en una página web usando tan solo elementos HTML, aunque están desaprobadados en favor de las hojas de estilo. Para ello basta con dotar de un valor al atributo `background` de la etiqueta `body`.

```
<body background=URI>
```

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

LISTAS

HTML ofrece a los autores varios mecanismos para especificar listas de información.

Todas las listas deben contener uno o más objetos de lista. Las listas pueden contener:

- Información no ordenada.
- Información ordenada.
- Definiciones.

La lista siguiente, por ejemplo, es una lista no ordenada, creada con el elemento ``:

```
<ul>  
  <li>información no ordenada.</li>  
  <li>información ordenada. </li>  
  <li>definiciones. </li>  
</ul>
```

Las listas son estructuras que permiten enumerar elementos por lo que, habitualmente, se usan en clasificaciones, instrucciones y barras de navegación (menús).

Las etiquetas que se utilizan para trabajar con listas son:

``: Lista sin indicador de orden.

``: Lista ordenadas (numeradas).

``: Elemento de una lista

Los tres son elementos de bloque.

He aquí un ejemplo de lista formada por dos elementos:

```
<ul>  
  <li>Elemento1</li>  
  <li>Elemento2</li>  
</ul>
```

El atributo `type`, especifica las opciones de representación en el navegador.

Para el elemento ``, los valores posibles del atributo `type` son `disc` (disco), `square` (cuadrado), y `circle` (círculo). El valor por defecto depende del nivel de anidamiento de la lista actual. Estos valores no distinguen entre mayúsculas y minúsculas.

Ordenación de listas

Una lista ordenada, creada por medio del elemento ``, debería contener información para la cual debe ponerse énfasis en el orden, como en una receta de cocina:

- Mezcle los ingredientes secos íntimamente.
- Vierta los ingredientes líquidos.
- Remueva durante 10 minutos.
- Hornee durante una hora a 300 grados.

El anterior ejemplo se traduce a html como:

```
<ol>
  <li>Mezcle los ingredientes secos íntimamente.</li>
  <li>Vierta los ingredientes líquidos.</li>
  <li>Remueva durante 10 minutos.</li>
  <li>Hornee durante una hora a 300 grados.</li>
</ol>
```

Para el elemento ``, los valores posibles del atributo `type` se resumen en la siguiente tabla (la diferencia entre mayúsculas y minúsculas es importante):

Tipo	Estilo de numeración	
1	números arábigos	1, 2, 3, ...
a	alfabética en minúsculas	a, b, c, ...
A	alfabética en mayúsculas	A, B, C, ...
i	números romanos en minúsculas	i, ii, iii, ...
I	números romanos en mayúsculas	I, II, III, ...

Obsérvese que **el atributo `type` está desaprobadado** y que los estilos de las listas deberían especificarse mediante hojas de estilo.

Anidamiento en listas

Ingredientes:

- 100 g de harina
- 10 g de azúcar
- 1 taza de agua
- 2 huevos
- sal, pimienta

Pasos a seguir:

1. Mezcle los ingredientes secos íntimamente.
2. Vierta los ingredientes líquidos.
3. Remueva durante 10 minutos.
4. Hornear durante una hora a 300 grados.

Notas:

Puede añadir uvas para mejorar la receta.

Las listas se pueden anidar, es decir, un elemento de una lista puede ser, a su vez, otra lista. En este caso, habrá que poner especial cuidado al colocar las etiquetas de cierre y apertura para que no se entrecrucen.

Las listas además de estar anidadas, pueden usar al mismo tiempo tipos diferentes de

listas, como en el ejemplo de la imagen, que es una lista de definiciones que contiene una lista no ordenada (los ingredientes) y una lista ordenada (los pasos a seguir).

TABLAS

Las tablas permiten representar datos en filas y columnas y, aunque durante cierto tiempo se usaron para maquetar información, esta práctica está totalmente desaconsejada.

Es responsabilidad de quien escribe la tabla asegurarse de que el número de datos se ajusta al de filas y columnas. Si no se hace así, el navegador no informará de los errores, sino que, tan sólo, mostrará huecos en determinados lugares o presentará un comportamiento extraño.

Estructura básica

Una tabla se organiza en filas y columnas y se representa en html definiendo, en primer lugar, la tabla en sí, a continuación, las filas de la tabla y para cada fila los elementos que contiene, que son las celdas que forman la tabla. Además, se pueden añadir elementos de encabezado y pie de tabla para usar estilos.

Las etiquetas para tablas son:

`<table>`, que posee atributos básicos, de internacionalización y de eventos. Además, posee el atributo específico `summary="..."`, para escribir un breve resumen de los contenidos de la tabla. Es de tipo bloque.

`<tr>`, señala cada fila de la tabla. Tiene atributos básicos, de internacionalización y de eventos.

`<td>`, señala cada elemento de una fila. Tiene atributos básicos, de internacionalización, de eventos y los atributos específicos:

Los atributos:

`abbr="..."`, para escribir un resumen de la celda.

`colspan="número"`, expande el número de celdas.

`rowspan="número"`, expande el número de filas.

Se puede poner una tabla dentro de otra, siempre que se coloque dentro de una celda.

`<th>`, posee atributos básicos, de internacionalización, de eventos y los mismos atributos específicos que `<td>`. Permite indicar las celdas que forman la cabecera de la tabla.

`<caption>`, posee atributos básicos, de internacionalización y de eventos. Sólo puede aparecer una vez y se escribe, justo, después de la etiqueta `<table>`. Se utiliza para poner un título o leyenda a la tabla.

`<thead>`, `<tfoot>`, `<tbody>` son etiquetas para crear tablas de forma avanzada. Su principal utilidad es agrupar bloques de la tabla y aplicar estilos a cada bloque, sin tener que ir poniendo el estilo en cada fila o celda. Las tres poseen los atributos básicos, de internacionalización y de eventos. Además, son de tipo bloque.

Una tabla tiene tres secciones distintas: encabezado, cuerpo y pie. La sección (o celda) de encabezado se define mediante un elemento `<thead>`, el cual contiene información correspondiente al encabezado, como los nombres de las columnas. Cada elemento `<tr>` define una sola fila de la tabla. Las columnas en la sección de encabezado se definen mediante los elementos `<th>`. La mayoría de los exploradores centran el texto que tiene el formato de los elementos `<th>` (columna de encabezado de la tabla) y lo muestran en negritas. Los

elementos de encabezado de una tabla se anidan dentro de los elementos de fila. La sección del pie se define mediante un elemento `<tfoot>` (pie de la tabla). Por lo general, el texto que se coloca en el pie incluye los resultados de cálculos y las notas al pie. Al igual que las demás secciones, el pie de la tabla puede contener filas, y cada fila puede contener columnas.

La sección del cuerpo, o cuerpo de la tabla, contiene los datos principales de la misma. El cuerpo de la tabla se define en un elemento `<tbody>`. En el cuerpo, cada elemento `<tr>` especifica una fila. Las celdas de datos contienen piezas de datos individuales, y se definen con elementos `<td>` (datos de tabla) dentro de cada fila.

Formatos de tablas: bordes, alineación, tamaño, etc.

XHTML incluye mecanismos para controlar:

- Los estilos de los bordes.
- El alineación horizontal y vertical de los contenidos de las celdas.
- Los márgenes de las celdas.
- Bordes y líneas de división.

Bordes.

Los atributos siguientes afectan al marco externo (`frame`) y a las líneas de división interiores (`rules`) de una tabla.

Definiciones de atributos:

`frame = void|above|below|hsides|lhs|rhs|vsides|box|border`. Este atributo especifica qué lados del marco que rodea a una tabla serán visibles. Valores posibles:

- `void`: Ningún lado. Este es el valor por defecto.
- `above`: Sólo el borde superior.
- `below`: Sólo el borde inferior.
- `hsides`: Sólo los bordes superior e inferior.
- `vsides`: Sólo los lados derecho e izquierdo.
- `lhs`: Sólo el lado izquierdo.
- `rhs`: Sólo el lado derecho.
- `box`: Los cuatro lados.
- `border`: Los cuatro lados.

rules = none|groups|rows|cols|all. Este atributo especifica qué líneas de división aparecerán entre las celdas de una tabla. Valores posibles:

- **none:** Ninguna línea de división. Este es el valor por defecto.
- **groups:** Sólo aparecerán líneas de división entre grupos de filas y grupos de columnas.
- **rows:** Sólo aparecerán líneas de división entre filas.
- **cols:** Sólo aparecerán líneas de división entre columnas.
- **all:** Aparecerán líneas de división entre todas las filas y columnas.

border = píxeles. Este atributo especifica la anchura (sólo en píxeles) del marco que rodea a una tabla.

Alineación.

Los siguientes atributos pueden especificarse para diferentes elementos de una tabla:

align = left|center|right|justify|char. Este atributo especifica la alineación de los datos y la justificación del texto de una celda. Valores posibles:

- **left:** Datos a la izquierda/Texto justificado a la izquierda.
- **center:** Datos centrados/Texto con justificación centrada. Este es el valor por defecto para los encabezados de las tablas.
- **right:** Datos a la derecha/Texto justificado a la derecha.
- **justify:** Texto doblemente justificado.
- **char:** Alinear el texto alrededor de un carácter específico.

valign = top|middle|bottom|baseline. Este atributo especifica la posición vertical de los datos dentro de una celda. Valores posibles:

- **top:** Los datos de la celda se alinean con la parte superior de la celda.
- **middle:** Los datos de la celda se centran verticalmente dentro de la celda. Este es el valor por defecto.
- **bottom:** Los datos de la celda se alinean con la parte inferior de la celda.
- **baseline:** Todas las celdas que estén en la misma fila que una celda cuyo atributo **valign** tenga este valor deberían tener sus datos textuales posicionados de tal modo que la

primera línea de texto aparezca en una línea de base común para todas las celdas de la fila. Esta restricción no se aplica a las líneas subsiguientes de texto de estas celdas.

char = carácter. Este atributo especifica que un carácter individual dentro de un fragmento de texto actúe como eje de alineación. El valor por defecto para este atributo es el carácter de punto decimal para el idioma actual, definido por el atributo **lang** (p.ej., el punto (".") en inglés y la coma (",") en francés). Los agentes de usuario no necesitan soportar este atributo.

charoff = longitud. Si está presente, este atributo especifica la distancia (**offset**) entre el borde y la primera aparición del carácter de alineación en cada línea. Si una línea no incluye el carácter de alineación, debería ser desplazada horizontalmente hasta la posición de alineación.

Cuando se usa **charoff** para establecer el **offset** de un carácter de alineación, la dirección del desplazamiento está determinada por la dirección actual del texto (establecida con el atributo **dir**). En texto de izquierda a derecha (el valor por defecto), el desplazamiento es desde el margen izquierdo. En textos de derecha a izquierda, el desplazamiento es desde el margen derecho.

La tabla de este ejemplo alinea una fila de valores monetarios alrededor de un punto decimal. Hemos establecido explícitamente el carácter de alineación ".".

```
<table border="1">
  <colgroup>
    <col align="char" char=".">
    <thead>
      <tr><th>verdura</th><th>precio por kilo</th></tr>
    </thead>
    <tbody>
      <tr><td>lechuga</td><td>$1</td></tr>
      <tr><td>zanahorias</td><td>$10.50</td></tr>
      <tr><td>nabos</td><td>$100.30</td></tr>
    </tbody>
  </colgroup>
</table>
```

La tabla formateada puede parecerse a la siguiente:

Verdura	Precio por kilo
Lechuga	\$1
Zanahorias	\$10.50
Nabos	\$100.30


Cuando los contenidos de una celda contienen más de una aparición del carácter de alineación especificado por char y los contenidos no caben en una sola línea, el comportamiento queda sin especificar. Los autores deberían por lo tanto tener cuidado cuando usen char.

Formato de contenido de celdas.

La etiqueta <col> se utiliza para asignar los mismos atributos a varias columnas de forma simultánea.

De esta forma, la etiqueta <col> no agrupa columnas, sino que sólo asigna atributos comunes a varias columnas.

La siguiente imagen muestra una tabla que hace uso de la etiqueta <col>:



Análisis de ventas

Análisis de ventas anuales

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

El código HTML necesario para crear la tabla anterior se muestra a continuación:

```
<table summary="Análisis de ventas anuales">
  <caption>Análisis de ventas anuales</caption>
  <col style="width:10%;" />
  <col style="width:30%;" />
  <thead>
    <tr>
      <th scope="col">AÑO</th>
      <th scope="col">Producto A</th>
      <th scope="col">Producto B</th>
      <th scope="col">Producto C</th>
      <th scope="col">Producto D</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">N-3</th>
      <td>-</td>
      <td>-</td>
      <td>-</td>
      <td>-</td>
    </tr>
    <tr>
      <th scope="row">N-2</th>
      <td>3</td>
      <td>5</td>
      <td>8</td>
      <td>4</td>
    </tr>
    <tr>
      <th scope="row">N-1</th>
      <td>4</td>
      <td>4</td>
      <td>7</td>
      <td>3</td>
    </tr>
    <tr>
      <th scope="row">N</th>
      <td>5</td>
      <td>7</td>
      <td>6</td>
      <td>2</td>
    </tr>
  </tbody>
</table>
```

Por otra parte, la etiqueta `<colgroup>` se emplea para agrupar de forma estructural varias columnas de la tabla. La forma habitual de indicar el número de columnas que abarca la agrupación es utilizar el atributo `span`, que establece el número de columnas de cada agrupación.

La siguiente imagen muestra una tabla avanzada con una agrupación de columnas realizada con la etiqueta <colgroup>:



AÑO	Producto A	Producto B	Producto C	Producto D
N-3	-	-	-	-
N-2	3	5	8	4
N-1	1	1	7	3
S	5	7	9	2

El código HTML necesario para crear la tabla anterior se muestra a continuación:

```
<table summary="Análisis de ventas anuales">
  <caption>Análisis de ventas anuales</caption>
  <colgroup span="1" style="color:red;" />
  <colgroup span="3" style="color:blue;" />
  <thead>
    <tr>
      <th scope="col">AÑO</th>
      <th scope="col">Producto A</th>
      <th scope="col">Producto B</th>
      <th scope="col">Producto C</th>
      <th scope="col">Producto D</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">N-3</th>
      <td>-</td>
      <td>-</td>
      <td>-</td>
      <td>-</td>
    </tr>
    <tr>
      <th scope="row">N-2</th>
      <td>3</td>
      <td>5</td>
      <td>8</td>
      <td>4</td>
    </tr>
```

```

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

```

El uso de las etiquetas <col> y <colgroup> no está muy extendido, debido a que la mayoría de navegadores no soportan muchas de sus funcionalidades.

AGRUPAMIENTO DE FILAS Y COLUMNAS

El tamaño de las celdas de la tabla se cambia para ajustarse a los datos que contienen. Los autores de documentos pueden crear celdas de datos más grandes mediante el uso de los atributos `rowspan` y `colspan`. Los valores que se asignan a estos atributos especifican el número de filas o columnas que ocupa una celda.

Tablas anidadas.

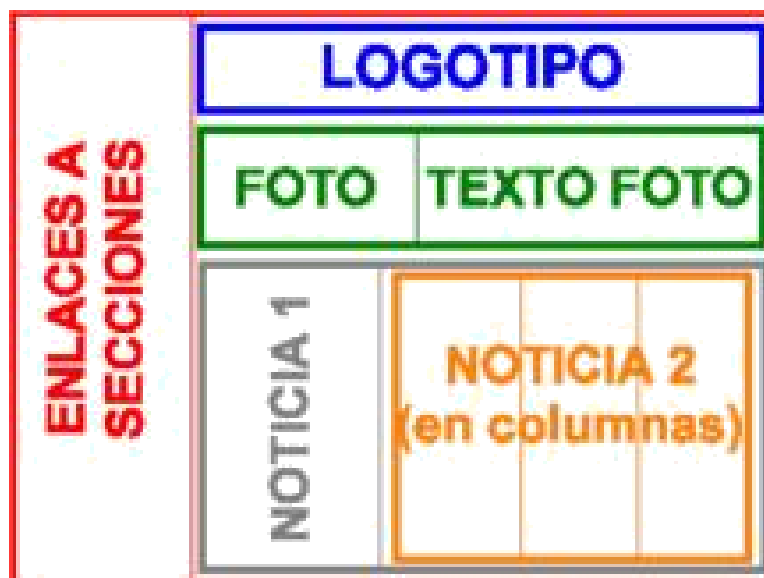
En ocasiones, especialmente cuando utilizamos las tablas como recurso de formato para distribuir la información dentro de la página, nos interesa introducir una tabla dentro de una celda perteneciente a una tabla creada anteriormente. Esto nos permitirá, por ejemplo, hacer que convivan tablas con borde junto a tablas sin borde.

Supongamos, por ejemplo, que estamos diseñando una página para introducir en ella los contenidos de una revista escolar. En la siguiente imagen podemos contemplar una posible estructura en la que vamos anidando sucesivas tablas.

Cada color corresponde a una tabla que incluye en su interior las que son de otro color diferente.

Como ya conocemos los códigos básicos que constituyen una tabla entenderemos rápidamente la estructura general. Para facilitar la interpretación se han recogido las etiquetas correspondientes a cada tabla con el mismo color con el que se presentan en el gráfico y se han ido sangrando las etiquetas correspondientes a cada tabla, de forma que se pueda ver con facilidad en qué celda se encuentra insertada cada una de ellas.

Introduciendo el texto correspondiente a cada zona de la hipotética pantalla obtendríamos



una presentación coincidente con el ejemplo gráfico.

```
<table border=0>
  <tr>
    <td>Enlaces a secciones</td>
    <td>
      <table border=0>
        <tr>
          <td>Logotipo </td>
        </tr>
      </table>
      <table border=0>
        <tr>
          <td>Foto </td>
          <td>Texto foto </td>
        </tr>
      </table>
      <table border=0>
        <tr>
          <td>Noticia 1 </td>
          <td>
            <table border=0>
```

```

        <tr>
            <td>Noticia en columnas
            </td>
            <td>Columna 2 </td>
            <td>Columna 3 </td>
        </tr>
    </table>
</td>
</tr>
</table>
</td>
</tr>
</table>

```

MARCOS (FRAMES)

Podríamos decir que una estructura de marcos es una artimaña que hace posible dividir la pantalla en varias zonas que nos permiten presentar en cada una de ellas una página independiente. Los marcos permiten organizar la presentación de varios documentos HTML simultáneamente en la ventana del navegador. La forma de conseguirlo es dividiendo ésta en diferentes áreas y asignando, a cada una de ellas, uno de los documentos a mostrar.

Creación de marcos

Su declaración tiene esta estructura:

```

<frameset [cols="..."| rows="..."]>
<frame src="url1" name="nombre_marco1"/>
<frame src="url2" name="nombre_marco2"/>
...
<noframes> Texto que aparece si el navegador no acepta marcos </noframes>
</frameset>

```

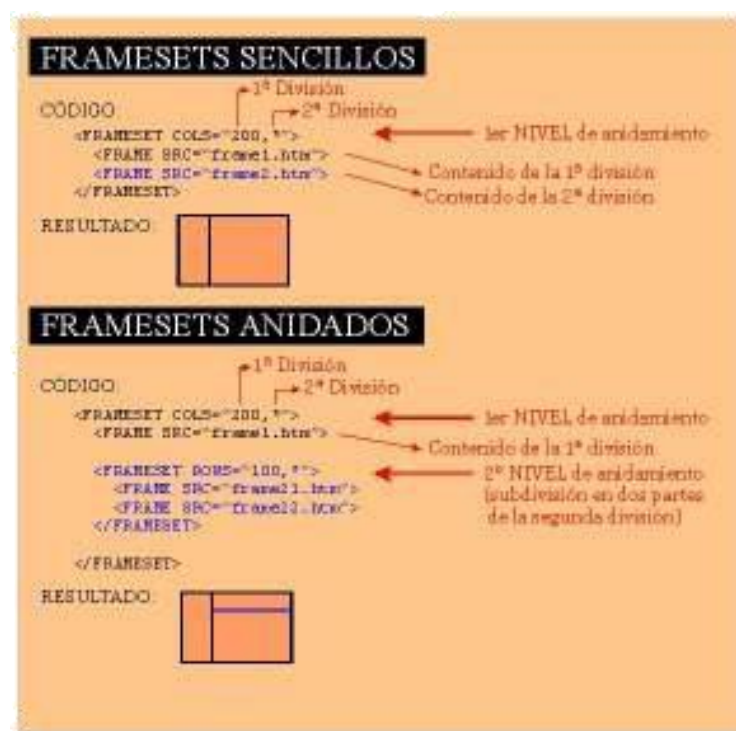
El tamaño de una fila o columna se puede indicar por el tanto por ciento que ocupa en el espacio total de las ventanas o por el número de píxeles. Por ejemplo **rows="20%,*,40%"**.

La etiqueta **<frame>** que sustituye a **<body>**, aparecerá tantas veces como filas o columnas se hayan indicado y, en ella, se especificará toda la información relativa a un marco. El atributo **src** se usa para indicar la página que se va a cargar en él y el atributo **name** para nombrar o identificar el marco.

Los marcos no son muy utilizados hoy en día, además, sólo son válidos en los documentos especificados como tipo frameset en su DTD. En la actualidad se usa otro tipo de marco que permite insertar un documento dentro de otro. El marco insertado se denomina “flotante” y

puede considerarse como un agujero que se abre en una página web para mostrar otra. Admite atributos básicos, de internacionalización, de eventos y los siguientes específicos:

- `src="url"`, página que se carga en el marco.
- `width="número"`, anchura en píxeles o en %.
- `height="número"`, altura en píxeles o en %.
- `scrolling="yes" | "no" | "auto"`. Indica la aparición o no de barras de desplazamiento laterales.
- `frameborder="1" | "0"`. Indica si el marco tiene borde o no.
- `name="..."`, nombre para el marco.
- `marginwidth="número"`, anchura en píxeles de los márgenes superior e inferior.
- `marginheight="número"`, altura en píxeles de los márgenes izquierdo y derecho



Marcos incrustados (iframes)

Además de poder crear una estructura de marcos convencional disponemos también de una posibilidad adicional que consiste en insertar dentro del flujo de la página una ventana en la cual podremos hacer que se muestren los contenidos de una página web. Es lo que se denomina marco en línea, in line **frame** o, utilizando el nombre de la etiqueta que sirve para crearlo, **iframe**.

Para crear un marco en línea utilizaremos la etiqueta **<iframe>** y su correspondiente cierre **</iframe>**

Lógicamente lo primero que tendremos que hacer será adjudicarle unas medidas al marco e indicar cuál es la página que se mostrará en su interior.

- **width**: será el atributo que nos indicará la anchura del marco.
- **height**: indicará la altura
- **src**: será el atributo con el que indiquemos cuál es la página que se mostrará en el hueco que hemos definido
- **name**: Servirá para identificar el marco en línea, de forma que podamos indicar que un enlace se cargue en dicho espacio.
- **frameborder**: Indicará si el marco lleva o no lleva borde. Sus valores pueden ser 0 y 1, siendo este último el valor por defecto
- **marginwidth** y **marginheight**: indican, mediante un entero positivo la distancia que separará el contenido del marco de los bordes internos del mismo.
- **scrolling**: utilizando el valor no haríamos que no se presentaran barras de desplazamiento, por lo que habría que haber comprobado que el contenido se visualiza en cualquier plataforma sin necesidad de ellas.

Para hacer que el destino de un vínculo se cargue dentro de un iframe lo único que tendremos que hacer al crear el vínculo es especificar el valor target con el nombre que le hayamos adjudicado al iframe. Lógicamente es imprescindible que el iframe lleve el atributo name.

Cuando se crea un vínculo desde una página para cargar un contenido en un iframe insertado en ella, el atributo target deberá llevar como valor el nombre adjudicado al iframe mediante el atributo name.

Cuando quieras utilizar vínculos desde dentro de un iframe de forma que se cargue otra página dentro del propio iframe el atributo target deberá indicar esta situación adoptando el valor target="_self"

Como el iframe es un elemento que, independientemente del tamaño que ocupe, se inserta como un elemento en línea dentro del flujo de la página se comporta de forma muy similar a cualquier otro elemento en línea, tal como una palabra.

Si algún navegador, no fuera capaz de mostrar el contenido del iframe podríamos paliar fácilmente este error. El texto que se introduzca antes del cierre </iframe> se ignorará siempre que se pueda mostrar el contenido del marco en línea, pero se mostrará en caso contrario, así que bastaría con que ese texto fuera el enlace a la página que mostramos en el iframe como veremos en el ejemplo.

Lo que tienes a continuación es, exactamente, el código que hace que se muestre el iframe que se puso como ejemplo al principio de la página.

```
<iframe src="mueetraiframe.html" name="muestra" width=200 height=100>  
    Ver un <a href="mueetraiframe.html"> ejemplo </a>  
</iframe>
```

El texto que aparece dentro del iframe sólo lo habrás visto en el caso de que tu navegador no soporte esta etiqueta y está ahí para que todos los visitantes pudieran acceder al contenido del archivo que pretendemos que se muestre en ese espacio.

FORMULARIOS

HTML es un lenguaje de marcado cuyo propósito principal consiste en estructurar los contenidos de los documentos y páginas web. Sin embargo, HTML también incluye elementos para crear aplicaciones web. El estándar HTML/XHTML permite crear formularios para que los usuarios interactúen con las aplicaciones web.

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

usuarios normalmente "completan" un formulario modificando sus controles (introduciendo texto, seleccionando objetos de un menú, etc.), antes de enviar el formulario a un agente para que lo procese (p.ej., a un servidor web, a un servidor de correo, etc.)

Por lo general, los datos que introducen los usuarios en una página Web se envían a un servidor Web, el cual proporciona acceso a los recursos de un sitio (por ejemplo, documentos en XHTML, imágenes). Estos recursos se ubican en el mismo equipo que el servidor Web, o en un equipo al que el servidor Web puede acceder a través de la red. Cuando un explorador solicita una página Web o un archivo que se encuentra en un servidor, este procesa la solicitud y devuelve el recurso solicitado. Una solicitud contiene el nombre y la ruta del recurso deseado, junto con el método de comunicación (conocido como protocolo). Los documentos en XHTML utilizan el Protocolo de transferencia de hipertexto (HTTP).

Dentro de la etiqueta `<form>`, se ponen los controles, cuyas etiquetas se nombran a continuación.

`<input>`, con atributos básicos, de internacionalización, de eventos, de foco y los siguientes:

`type="..."`, con diez posibles valores (text, password, checkbox, radio, submit, reset, file, hidden, image, button) y es de tipo en línea. Dependiendo del valor, se pueden utilizar otros atributos.

`value="..."`, indica el valor del control.

`size="..."`, especifica el ancho del control en número de caracteres.

`maxlength="..."`, en los controles text y password indica la longitud máxima de caracteres que se puede teclear.

`checked="checked"` indica si el control está marcado en los controles radio y checkbox.

`disable="disable"` desactiva el control y, por tanto, no se envía el valor al servidor.

`readonly="readonly"` pone el control en sólo lectura. (Sólo para los controles text o password).

`src="url"`, para indicar la ruta de la imagen que actuará como un botón de envío.

`alt=""`, descripción o texto alternativo que aparece cuando no se carga la imagen.

`<fieldset>` sirve para hacer agrupaciones lógicas de controles para lo cual dibuja un marco que engloba varios de ellos. Es de tipo bloque y tiene atributos básicos, de internacionalización y de eventos.

`<legend>`, se escribe justo debajo de la apertura del fieldset y le pone un título al grupo.

`<label>`, tiene atributos básicos, de internacionalización y de eventos. Define una etiqueta que puede asociarse con un control de formulario mediante el atributo específico `for="id"`, siendo "id" el identificador del control del formulario al que se asocia. Mejora la entrada de usuario, ya que si se hace un clic de ratón en el texto que muestra `<label>` conmuta el estado del control asociado.

`<textarea>`, de tipo línea, con atributos básicos, de internacionalización, de eventos, de foco y los siguientes específicos:

`cols="..."`, número de columnas.

`rows="..."`, número de filas.

`disabled="disabled"`, desactiva el textarea con lo que no se envía información.

`readonly="readonly"`, de sólo lectura por lo que no se puede escribir.

`<select>`, muestra listas desplegables o desplegadas. Es de tipo línea y admite atributos básicos, de internacionalización, de eventos, de foco y los siguientes específicos:

`size="..."`, fija el tamaño inicial de la lista. Si es mayor que uno la lista aparece desplegada.

`multiple="multiple"` permite elegir más de una opción.

`disabled="disabled"`, desactiva el control.

`<option>`, indica cada una de las opciones de la lista. Admite los siguientes atributos específicos:

`selected="selected"`, opción marcada por defecto.

`value="..."`, valor de la variable de la opción correspondiente que se envía al servidor.

`disabled="disabled"`, desactiva esa opción de la lista.

`<optgroup>`, sirve para englobar etiquetas `<option>` y hacer subgrupos. Tiene atributos básicos, de internacionalización, de eventos y los siguientes específicos:

`label="..."`, texto para la cabecera del grupo.

`disabled="disabled"`, desactiva el grupo.

Ejemplo:

```
<select>
  <optgroup label="vehículos suecos">
    <option value="Volvo">Volvo</option>
    <option value="Saab">Saab</option>
  </optgroup>
  <optgroup label="vehículos alemanes">
    <option value="Opel">Opel</option>
    <option value="Audi">Audi</option>
  </optgroup>
</select>
```

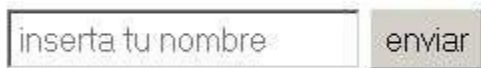
Nuevos atributos en HTML5

Placeholder

Este nuevo atributo permite al usuario ver una especie de leyenda o label dentro del input. Cuando el usuario comience a escribir sobre él este placeholder desaparecerá.

Es muy útil para guiar al usuario sobre el tipo de contenido a rellenar en el formulario.

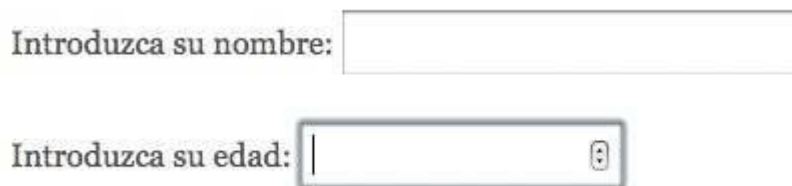
Ejemplo:

Un formulario con un campo de entrada de texto que contiene el texto "inserta tu nombre" en un color grisáceo, lo que indica que es un placeholder. A la derecha del campo hay un botón rectangular con el texto "enviar".

Autofocus

La función de este atributo es poner el cursor de manera activa en un input del formulario sin necesidad de hacer click en él.

Ejemplo:

Un formulario con dos campos de entrada. El primer campo está precedido por el texto "Introduzca su nombre:" y el cursor del ratón está activo dentro del campo. El segundo campo está precedido por el texto "Introduzca su edad:" y también tiene el cursor activo dentro del campo.

Autocomplete

Este nuevo elemento nos va ayudar a recordar, autocompletar y/o sugerir los valores insertados anteriormente en el mismo formulario.

Height y width

Estos atributos solo se pueden aplicar a los input de tipo 'image', así de esta manera podemos jugar con las dimensiones de la imagen que sustituye al botón 'submit' para enviar formularios.

List

Este atributo sirve para asociar el elemento de tipo 'datalist' con un input, de esta manera el input tendrá establecido por defecto una serie de valores para facilitar la inserción de datos.

Min y Max

Sirven para fijar los valores mínimos y máximos de los input con valor numérico o de fecha.

Multiple

Con el atributo multiple HTML5 se pueden seleccionar varios ficheros o emails en el envío de formularios, es compatible con los input de tipo file o email.

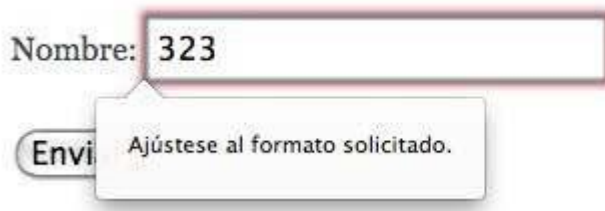
Novalidate

El atributo novalidate HTML5 hace que el elemento en cuestión no se valide al enviar el formulario, es decir, si el formulario tiene varios input que disponen de validación automática por parte del navegador podemos anular esta validación por cada input.

Pattern

Con el atributo pattern HTML5 podremos delimitar mediante expresiones regulares el contenido insertado por el usuario en los inputs de los formularios.

De no cumplirse la expresión regular sale un mensaje similar a este:



Required

Con el atributo required HTML5 podemos establecer que un input de formulario sea de rellenado obligatorio.

En caso de no rellenar el campo obligatorio sale un mensaje automático de aviso.



Nuevos tipos de input en HTML5

Dentro de la gran cantidad de novedades que nos ofrece el HTML5, una muy interesante son los nuevos valores para el atributo `type` del elemento `input`.

Estos nuevos tipos de campos hacen que los navegadores adopten distintos comportamientos que, sin dudas, nos van a hacer la vida más fácil a los desarrolladores de sitios web.

Así como actualmente tenemos el conocido `password` que oculta la contraseña con asteriscos o círculos (dependiendo del navegador); ahora contamos con el nuevo `search` que presenta una pequeña cruz para poder borrar su contenido; o también el nuevo campo numérico `number` que muestra dos flechas (hacia arriba y hacia abajo) para aumentar o disminuir el valor del número.

Algo muy importante a tener en cuenta es que si bien estos elementos todavía no son soportados por todos los navegadores modernos, el uso de los mismos no afectará de ningún modo en los navegadores que no los soporten, actuarán simplemente como si fuesen del tipo `text`.

Ejemplos de nuevos atributos

A continuación, los distintos valores, recuerda que solo verás los nuevos comportamientos si estás utilizando algún navegador moderno (últimas versiones de Chrome, Firefox, Safari, Opera o IE).

Búsqueda (`search`):

```
<input type="search" name="busqueda">
```

Al ingresar texto en el campo, el navegador muestra una cruz a la derecha para borrar todo lo que hemos escrito.



Teléfono (tel):

```
<input type="tel" name="telefono">
```

A la hora de completar un input de tipo tel, un smartphone como el iPhone convierte su teclado a números de teléfono.



Dirección url (url):

```
<input type="url" name="url">
```

En este campo, el teclado del iPhone es qwerty pero en modo "url", ya que ofrece teclas fundamentales para escribir una dirección web como son el punto, la barra "/" o la tecla ".com".



Correo electrónico (email):

```
<input type="email" name="correo">
```

Esta vez, el teclado del smartphone es qwerty pero también tenemos la tecla "@"



Fecha y hora (datetime):

```
<input type="datetime" name="fechahora">
```

Si estás viendo esta página con la última versión de Opera, al clicar en el campo verás un calendario muy completo que el navegador dispone de forma totalmente nativa.



Fecha (date):

```
<input type="date" name="fecha">
```

Si estamos usando Opera, el calendario es el mismo que el de la imagen anterior.

Mes (month):

```
<input type="month" name="mes">
```

El calendario aquí permite seleccionar el número de mes.

Semana (week):

```
<input type="week" name="semana">
```

El calendario que nos muestra Opera para el campo de semana, nos permite elegir el número de semana del año.



Hora (time):

```
<input type="time" name="hora">
```

Aquí el input está con el formato de hora, con los dos puntos ":" correspondientes y las flechas para subir o bajar el horario.



Número (number):

```
<input type="number" name="num" min="0" max="50">
```

Para este input tenemos los atributos min y max para establecer el máximo y el mínimo que acepta el campo.

En un smartphone vemos el teclado numérico:



Rango (range):

```
<input type="range" name="rango" min="0" max="50">
```

El input de tipo range se presenta como un control para arrastrar con el mouse (o con el dedo en un móvil con pantalla táctil). Este campo también acepta los atributos min y max.



Color (color):

```
<input type="color" name="color">
```

Si estás viendo este campo con la última versión de Opera, el navegador presenta de forma nativa un selector de color... otra funcionalidad que comúnmente tendríamos que hacer con JavaScript.



Al clicar en "otros..." de la ventana anterior, Opera muestra una ventana con un selector de color mucho más avanzado como el de la siguiente captura de pantalla:

