

# MF0950\_2-Contrucción de páginas web

UF1302: Creación de páginas web con el lenguaje de marcas

## Contenido

|  |    |
|--|----|
| UF1302: Creación de páginas web con el lenguaje de marcas.....       | 1  |
| UF1302: Creación de páginas web con el lenguaje de marcas.....       | 4  |
| HTML .....   | 4  |
| Estructura básica de una página web .....                            | 4  |
| Etiqueta "html" .....  | 4  |
| Etiqueta "head" .....  | 4  |
| Etiqueta "title" .....   | 5  |
| Etiqueta "body" .....  | 5  |
| Etiqueta "p".....  | 5  |
| Comentarios en HTML .....  | 6  |
| Introducción a los Estilos en HTML.....                              | 7  |
| Estilo en línea - Atributo <b>style</b> .....                        | 7  |
| Estilo interno - Etiqueta "style" .....                              | 7  |
| Estilo externo - Etiqueta "link".....                                | 8  |
| Atributos Globales.....  | 9  |
| Etiqueta "span" y atributo de tipo <b>class</b> y <b>CSS</b> . ....  | 9  |
| Etiquetas para dar formato al texto.....                             | 9  |
| Etiqueta "br".....   | 9  |
| Etiqueta "hr" .....  | 9  |
| Elemento "b" .....   | 10 |
| Elemento "i" .....   | 10 |
| Elementos "strong" y "em".....                                       | 11 |
| Elemento "small" .....   | 11 |
| Elementos "sub" y "sup".....   | 12 |
| Elemento "mark".....   | 12 |
| Cabeceras en HTML.....   | 13 |
| Enlaces en HTML .....  | 14 |
| Atributo <b>target</b> .....   | 14 |
| Enlace a otra página del mismo sitio web .....                       | 15 |
| Listas en HTML.....  | 16 |
| Lista desordenada - Elementos "ul" y "li" .....                      | 16 |
| Lista ordenada - Elementos "ol" y "li" .....                         | 17 |
| Lista de descripción de términos - Elementos "dl", "dt" y "dd" ..... | 17 |
| Imágenes en HTML .....   | 18 |
| Atributo <b>src</b> .....  | 18 |

|  |    |
|--|----|
| Atributos <b>width</b> y <b>height</b> .....   | 19 |
| Atributo <b>alt</b> .....  | 19 |
| Tablas en HTML.....  | 19 |
| Tabla con bordes - Propiedad CSS <b>border</b> .....                                 | 20 |
| Tabla con celdas separadas - Propiedad CSS <b>border-spacing</b> .....               | 20 |
| Tabla con borde junto - Propiedad CSS <b>border-collapse</b> .....                   | 21 |
| Ancho de una tabla - Propiedad CSS <b>width</b> .....                                | 22 |
| Título de una tabla - Elemento "caption" .....                                       | 23 |
| Una celda puede abarcar varias columnas de una tabla - Atributo <b>colspan</b> ..... | 24 |
| Una celda puede abarcar varias filas de una tabla - Atributo <b>rowspan</b> .....    | 24 |
| Cabeceras de una tabla - Elemento "th".....  | 25 |
| Estructura de una tabla - Elementos "thead", "tbody" y "tfoot" .....                 | 26 |
| Divisiones en HTML .....   | 28 |
| Elemento "div" y atributo <b>class</b> .....   | 29 |
| Elemento "div" y atributo <b>id</b> .....  | 30 |
| HTML5-Componentes básicos .....  | 33 |
| Estructura del cuerpo .....  | 34 |
| Organización .....   | 35 |
| <header> .....   | 37 |
| <nav> .....  | 38 |
| <section>.....   | 39 |
| <aside> .....  | 39 |
| <footer> .....   | 40 |
| Dentro del cuerpo.....   | 42 |
| Etiqueta HTML5 video: Embeber vídeos con su propio player .....                      | 43 |
| ¿Qué navegadores soportan el tag de HTML5 video?.....                                | 43 |
| Formatos de vídeo compatibles .....  | 43 |
| Sintaxis básica de la etiqueta de HTML5 video .....                                  | 44 |
| Atributos de la etiqueta de HTML5 video .....  | 44 |

## UF1302: Creación de páginas web con el lenguaje de marcas

### HMTL

**HTML** (*HyperText Markup Language*, Lenguaje de Marcado de HiperTexto) es un lenguaje utilizado para la creación de páginas web.

HTML es un estándar desarrollado por **W3C** (*World Wide Web Consortium*). En este enlace "<http://www.w3.org/TR/html/>" se puede consultar su especificación actual (**HTML5.1**).

Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por **etiquetas**, que marcan el inicio y el fin de cada Etiqueta del documento.

Un documento hipertexto no sólo se compone de texto, puede contener imágenes, sonido, vídeos, etc., por lo que el resultado puede considerarse como un documento multimedia.

Los documentos HTML deben tener la extensión **html** o **htm**, para que puedan ser visualizados en los navegadores (programas que permiten visualizar las páginas web).

Los **navegadores** se encargan de interpretar el código HTML de los documentos, y de mostrar a los usuarios las páginas web resultantes del código interpretado.

### Estructura básica de una página web

#### Etiqueta "html"

Un documento HTML contiene marcas (etiquetas), las cuales se escriben empleando los caracteres *menor que* "<", *mayor que* ">" y *barra inclinada* "/".

Por ejemplo, las etiquetas de inicio (<html>) y fin (</html>) representan al Etiqueta raíz ("html") que en todo documento HTML hay que escribir.

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

#### Etiqueta "head"

Dentro del Etiqueta "html", es decir, entre <html> y </html>, se debe escribir el Etiqueta "head" que, como iremos viendo a lo largo del curso, puede contener diversa información sobre el documento **que no se publica como contenido de la página**:

```
<html>  
  <head>  
  </head>  
</html>
```

## Etiqueta "title"

Entre `<head>` y `</head>` se declaran Etiquetas como el título de la página, una llamada a una hoja de estilos CSS externa y librerías JavaScript principalmente. Por ejemplo, el Etiqueta "title", sirve para indicar el título del documento.

```
<html>
  <head>
    <title>Ejemplo de 2 párrafos</title>
  </head>
</html>
```

Véase en el ejemplo "dos-parrafos.html" que, el texto escrito entre `<title>` y `</title>`, es decir "*Ejemplo de 2 párrafos*", se visualiza en la parte superior de la pestaña donde se muestra la página web en el navegador.

## Etiqueta "body"

En un documento HTML, después del "head" y siempre dentro de la etiqueta `<html></html>`, hay que declarar el Etiqueta "body":

```
<html>
  <head>
    <title>Ejemplo de 2 párrafos</title>
  </head>
  <body>
  </body>
</html>
```

El Etiqueta "body" alberga todo el contenido (párrafos, imágenes, vídeos...) del documento, los cuales se mostrarán en el navegador.

## Etiqueta "p"

En este caso, entre `<body>` y `</body>` se incluyen dos Etiquetas "p" delimitados por la etiqueta de inicio `<p>` y la de cierre `</p>`:

```
<html>
  <head>
    <title>Ejemplo de 2 párrafos</title>
  </head>
  <body>
    <p>Esto es un párrafo.</p>
    <p>Esto es otro párrafo.</p>
  </body>
</html>
```

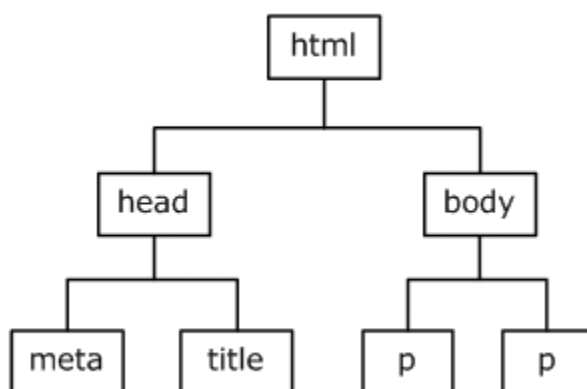
## Estructura básica de un documento HTML

Finalmente, completaremos el código de este primer ejemplo añadiendo:

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo de 2 párrafos</title>
  </head>
  <body>
    <p>Esto es un párrafo.</p>
    <p>Esto es otro párrafo.</p>
  </body>
</html>
```

- **<!DOCTYPE html>** en la primera línea. Esta es la forma de indicar al navegador donde se visualice el documento que este está escrito en la versión HTML5.
- **lang="es-ES"** en la etiqueta de inicio **<html>**. Esto sirve para especificar, mediante el atributo **lang**, el idioma del contenido de la etiqueta "html". En este caso español (España). Otros posibles valores del atributo **lang** son: **"es-AR"** para español (Argentina), **"es-CO"** para español (Colombia), **"es-MX"** para español (México), etc.
- **<meta charset="utf-8">** dentro de la etiqueta "head". Así, se indica la codificación de caracteres del documento, en esta ocasión **UTF-8** (*8-bit Unicode Transformation Format*, Formato de Transformación Unicode de 8 bits). Obsérvese que, la etiqueta "meta" está vacío (no tiene contenido) y, en consecuencia, no necesita ser cerrado escribiendo **</meta>**.

De esta forma, hemos escrito la **estructura básica de un documento HTML que muestra por pantalla dos párrafos**. Gráficamente, los elementos utilizados se pueden representar del siguiente modo:



## Comentarios en HTML

En un documento HTML, los comentarios pueden escribirse entre los caracteres "**<!--**" y "**-->**".

Los comentarios no se mostrarán en el navegador donde se visualice el documento HTML.

```
<!-- En el body hay 2 párrafos -->
```

## Introducción a los Estilos en HTML

Para cambiar el estilo en el que por defecto se visualizan los elementos de un documento HTML en un navegador web, existen varios métodos:

- **Estilo en línea (*Inline*):** utilizando el atributo **style**.
- **Estilo interno (*Internal*):** usando un elemento "style".
- **Estilo externo (*External*):** empleando un archivo CSS (*Cascading Style Sheets*) externo al documento HTML.

**EJEMPLO** A continuación se explica cómo hacer uso de estos métodos para mostrar un párrafo de color rojo.

### Estilo en línea - Atributo **style**

Para cambiar el color de un párrafo, se puede escribir el siguiente código ("**estilo-en-línea.html**"):

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo de estilo en línea</title>
  </head>
  <body>
    <p style="color:red">Esto es un párrafo de color rojo.</p>
  </body>
</html>
```

### Estilo interno - Etiqueta "style"

En este caso, se usa la etiqueta "style" dentro del "head".

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo de estilo interno</title>
    <style>
      p {color:red;}
    </style>
  </head>
  <body>
    <p>Esto es un párrafo de color rojo.</p>
  </body>
</html>
```

## Estilo externo - Etiqueta "link"

Para aplicar un estilo externo, hay que hacer uso de la etiqueta "link", el cual permite enlazar el documento HTML con un archivo externo:

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo de estilo externo</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <p>Esto es un párrafo de color rojo.</p>
  </body>
</html>
```

Véase que, en el atributo **rel** se indica la relación que existe entre el archivo HTML y el archivo CSS. Por otro lado, por medio del atributo **href** se ha enlazado el documento HTML con el archivo **"styles.css"**, que es donde está definido el estilo del párrafo:

```
p {color:red;}
```

Por norma general, utilizar archivos CSS externos es la forma más habitual y adecuada de aplicar estilos a documentos HTML.

**EJEMPLO** Además de **color**, existen otras muchas propiedades CSS que pueden utilizarse. Por ejemplo, si además de rojo, también se quiere indicar que el párrafo se muestre centrado en la pantalla del navegador, esto se puede hacer usando la propiedad **text-align** ("**parrafo-rojo-y-centrado.html**"):

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo de párrafo rojo y centrado</title>
    <style>
      p {
        color:red;
        text-align:center;
      }
    </style>
  </head>
  <body>
    <p>Esto es un párrafo de color rojo y centrado.</p>
  </body>
</html>
```



## Atributos Globales

Todas las etiquetas html disponen de una serie de atributos para poder marcar elementos para posteriormente ser usados en programación Javascript y/o en hojas de estilos CSS.

Atributo id: Pensado para que solo se use una vez por página (id="content")

Atributo class: Pensado para ser usado en varios elementos de una página (class="destacado")

Atributo title: Especifica información extra sobre un elemento que se muestra al pasar el ratón por encima en caja con fondo amarillo (tooltip).

Atributo style: Se usa para declarar estilos CSS que solo afectan a la etiqueta.

## Etiqueta "span" y atributo de tipo class y CSS.

Se puede decir que la etiqueta span sirve de comodín, ya que es una etiqueta que no tiene una acción directa de formato con el código pero sirve para establecer marcas dentro de otra etiqueta. El ejemplo más claro es cuando es cuando dentro de un párrafo, se quiere dar otro formato a una palabra o frase.

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo uso dla etiqueta span y del atributo class</title>
    <style>
      .rojo {color:red;}
    </style>
  </head>
  <body>
    <p>Párrafo con <span class="rojo">dos palabras</span> de color rojo.</p>
  </body>
</html>
```

El atributo class, si hablamos de estilos CSS, sirve para declarar un estilo para que afecte a varios elementos de una página. En html se declara con el nombre class="nombre\_clase" y en lenguaje CSS con el punto, .nombre\_clase.

## Etiquetas para dar formato al texto

### Etiqueta "br"

Declara un salto de línea (punto y aparte) sin margen.

```
<p>Texto uno.<br>Texto dos.<br>Texto tres.</p>
```

### Etiqueta "hr"

Para mostrar una línea horizontal en la pantalla, se puede usar la etiqueta "hr".

```
<p>Párrafo uno.</p>
<hr>
<p>Párrafo dos.</p>
<hr>
<p>Párrafo tres.</p>
```

## Elemento "b"

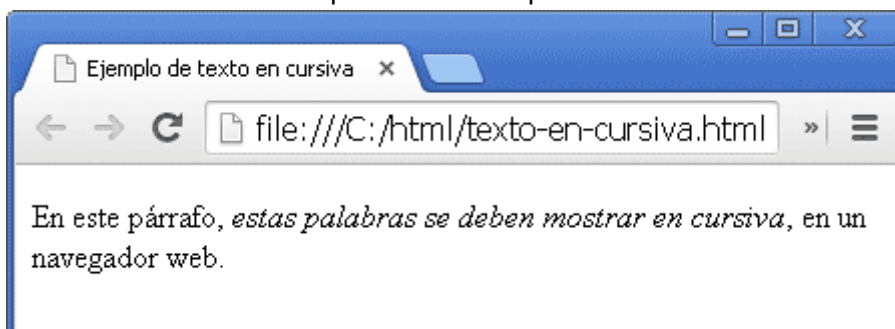
El elemento "b" sirve para indicar que un texto se muestre en negrita. Así, para visualizar:



```
<p>En este párrafo, <b>estas palabras se deben mostrar en negrita</b>, en un navegador web.</p>
```

## Elemento "i"

El elemento "i" sirve para indicar que un texto se muestre en itálica (cursiva).



```
<p>En este párrafo, <i>estas palabras se deben mostrar en cursiva</i>, en un navegador web.</p>
```

A un mismo texto se puede indicar más de un formato. Por ejemplo, para ver:



```
<p>En este párrafo, <i>de estas palabras en cursiva <b>estas también se ven en negrita</b></i>, en un navegador web.</p>
```

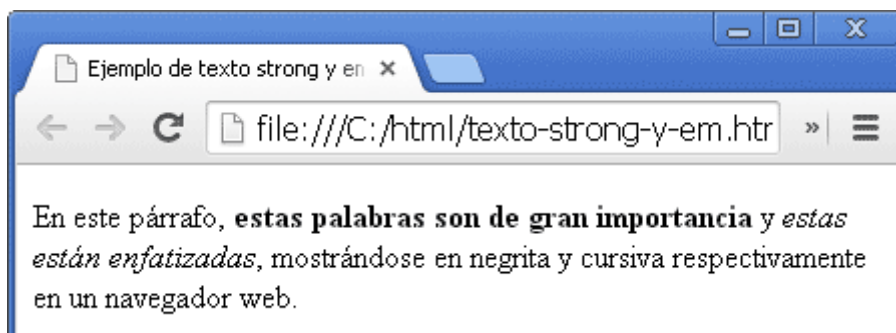
Obsérvese en el código, que las etiquetas `<b>` y `</b>` están anidadas dentro de las etiquetas `<i>` e `</i>`.

## Elementos "strong" y "em"

Por otra parte, el elemento "strong" (que alberga texto de gran importancia) y el elemento "em" (que contiene texto enfatizado), también muestran texto en negrita y cursiva, respectivamente.

<p>En este párrafo, <strong>estas palabras son de gran importancia</strong> y <em>estas están enfatizadas</em>, mostrándose en negrita y cursiva respectivamente en un navegador web.</p>

En un navegador se visualizará:



**Nota:** los elementos "b" e "i", no proporcionan la importancia semántica que sí tienen "strong" y "em".

## Elemento "small"

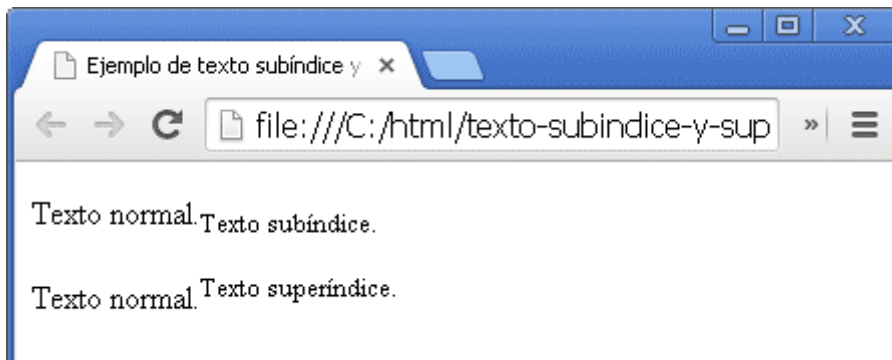
Con el elemento "small" se indica que un texto se muestre más pequeño en un navegador. Por ejemplo, para ver:



<p>En este párrafo, <small>estas palabras se muestran más pequeñas</small>, en un navegador web.</p>

## Elementos "sub" y "sup"

Los elementos "sub" y "sup" permiten indicar que un texto se muestre como un subíndice o como un superíndice, respectivamente. Por ejemplo, para ver en un navegador:

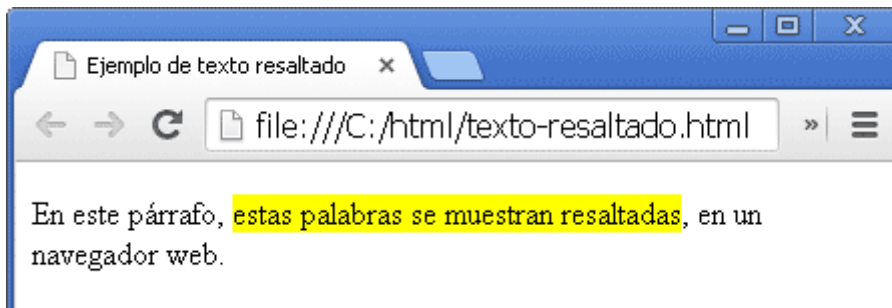


```
<p>Texto normal.<sub>Texto subíndice.</sub></p>
```

```
<p>Texto normal.<sup>Texto superíndice.</sup></p>
```

## Elemento "mark"

El elemento "mark" permite indicar que un texto se muestre resaltado en un navegador, como por ejemplo:



```
<p>En este párrafo, <mark>estas palabras se muestran resaltadas</mark>, en un navegador web.</p>
```

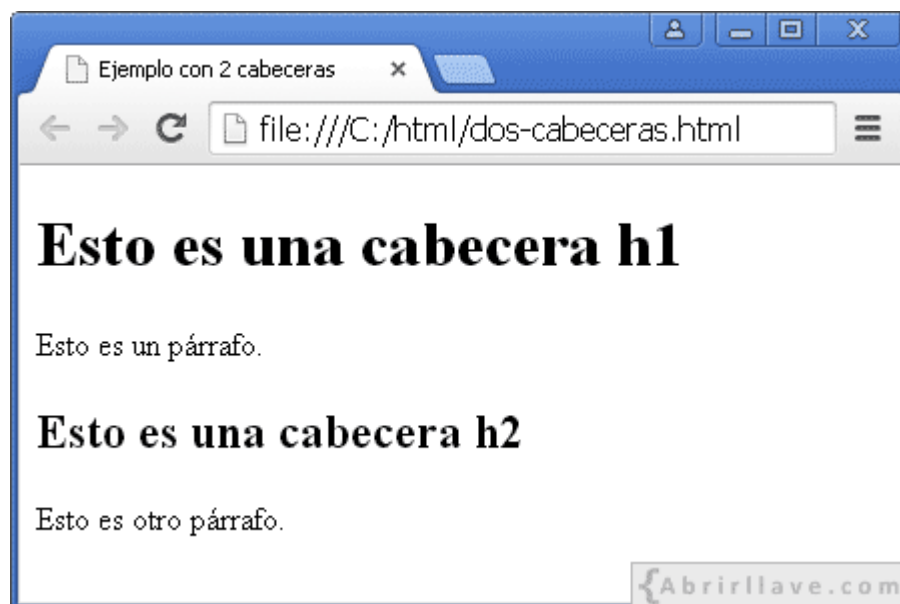
## Cabeceras en HTML

En HTML existen seis elementos que pueden ser utilizados para escribir cabeceras o encabezados: "h1", "h2", "h3", "h4", "h5" y "h6".

En el siguiente documento HTML se han escrito dos cabeceras:

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo con 2 cabeceras</title>
  </head>
  <body>
    <h1>Esto es una cabecera h1</h1>
    <p>Esto es un párrafo.</p>
    <h2>Esto es una cabecera h2</h2>
    <p>Esto es otro párrafo.</p>
  </body>
</html>
```

Al abrir el archivo "*dos-cabeceras.html*" en *Google Chrome* se verá:



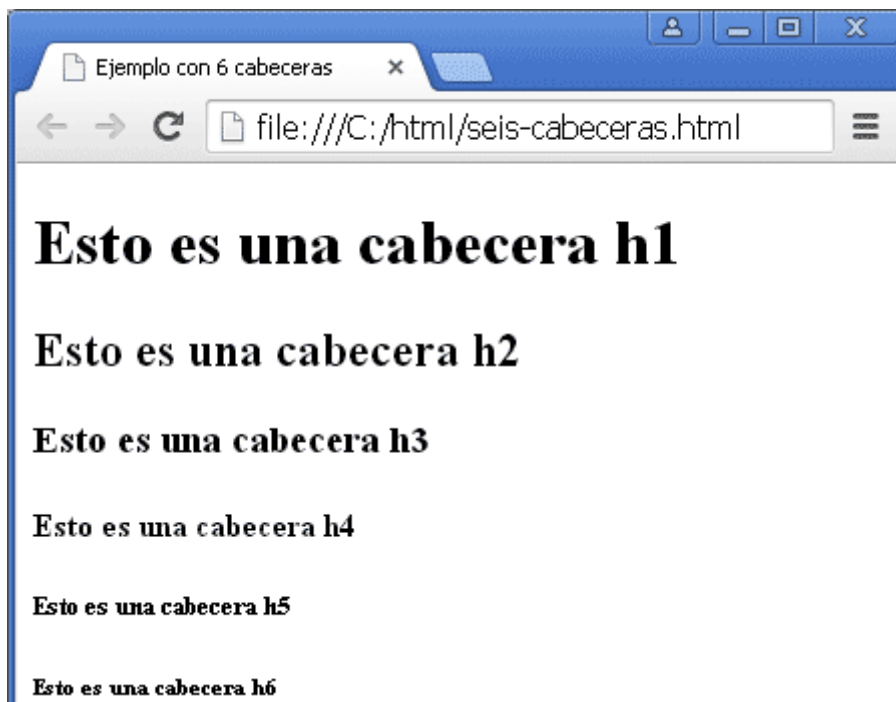
Las cabeceras se utilizan para estructurar el contenido de una página web y, a la hora de utilizarlas, hay que tener en cuenta que, la cabecera "h1" es la de mayor importancia, después "h2", "h3", etc. Siendo "h6" la menos importante.

### EJEMPLO

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo con 6 cabeceras</title>
  </head>
  <body>
    <h1>Esto es una cabecera h1</h1>
    <h2>Esto es una cabecera h2</h2>
```

```
<h3>Esto es una cabecera h3</h3>  
<h4>Esto es una cabecera h4</h4>  
<h5>Esto es una cabecera h5</h5>  
<h6>Esto es una cabecera h6</h6>  
</body>  
</html>
```

que en pantalla las cabeceras se muestran de mayor a menor tamaño, según su importancia:



En muchas páginas web es habitual que exista una única cabecera "h1" y varias cabeceras "h2", varias más "h3", etc.

## Enlaces en HTML

La etiqueta <a> define un hipervínculo, que se utiliza para vincular de una página a otra.

El atributo más importante del elemento <a> es el atributo href, que indica el destino del enlace.

De forma predeterminada, los vínculos aparecerán de la siguiente manera en todos los navegadores:

- Un enlace no visitado está subrayado y azul
- Un enlace visitado está subrayado y púrpura
- Un enlace activo está subrayado y rojo

<p><a href="http://www.w3schools.com/">Ir a página W3schools</a> para consultar información sobre programación.</p>

### Atributo target

Los enlaces por defecto se abren en la misma ventana del navegador, si el objetivo es abrir el enlace en una nueva ventana/pestaña del navegador, se debe declarar el atributo target con el valor \_blank.

<p><a href="http://www.abrirllave.com/" target="\_blank">Abrirllave</a> es un sitio web de tutoriales de informática.</p>

## Enlace a otra página del mismo sitio web

Si en un documento HTML se quiere escribir un enlace a otro documento ubicado en el mismo sitio web, por ejemplo a **"otra-pagina.html"**, visualizándose en pantalla:



Se podría escribir ("**pagina-con-un-enlace-a-otra-pagina.html**"):

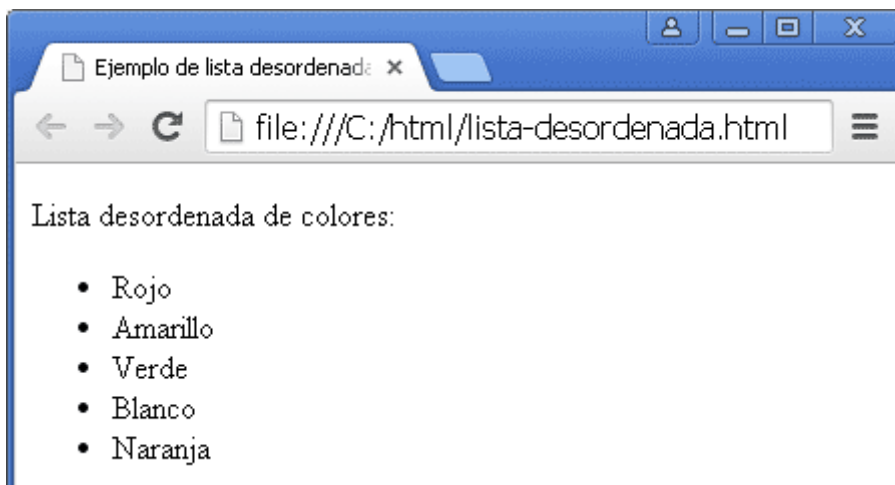
```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo de enlace a otra página</title>
  </head>
  <body>
    <p>Enlace a <a href="otra-pagina.html">otra página</a> ubicada en el mismo sitio web.</p>
  </body>
</html>
```

## Listas en HTML

En HTML, los elementos "ul", "ol", "li", "dl", "dt" y "dd", permiten representar listas ordenadas (*ordered lists*), listas desordenadas (*unordered lists*) y listas de descripción de términos (*description lists*). En el 98% de las páginas web, las barras o menús de navegación están declarados con listas, ideal para declarar menús desplegable con padres e hijos.

### Lista desordenada - Elementos "ul" y "li"

**EJEMPLO** Para que en un navegador se muestre la siguiente lista desordenada de colores:



Se puede hacer uso de los elementos "ul" y "li" del siguiente modo ("*lista-desordenada.html*"):

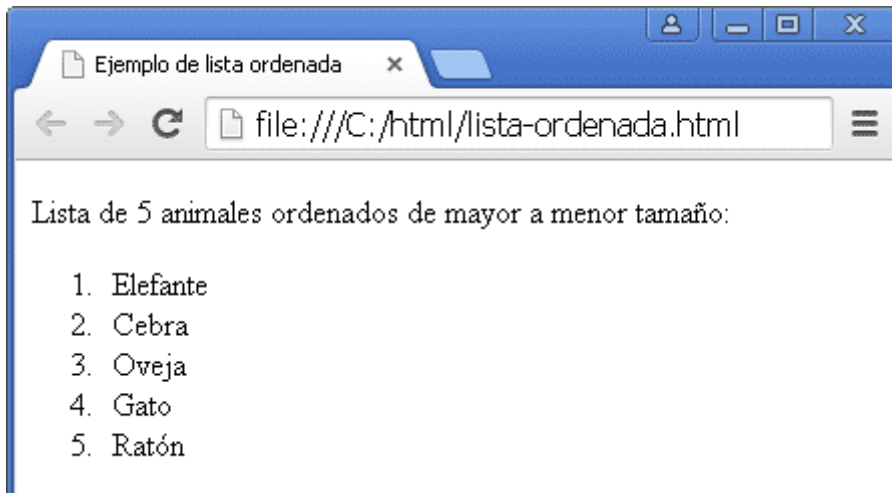
```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo de lista desordenada</title>
  </head>
  <body>
    <p>Lista desordenada de colores:</p>
    <ul>
      <li>Rojo</li>
      <li>Amarillo</li>
      <li>Verde</li>
      <li>Blanco</li>
      <li>Naranja</li>
    </ul>
  </body>
</html>
```

Obsérvese que, cada color está contenido en un elemento "li", y todos ellos se escriben dentro del elemento "ul".



## Lista ordenada - Elementos "ol" y "li"

**EJEMPLO** Si se quiere visualizar una lista ordenada como la siguiente:



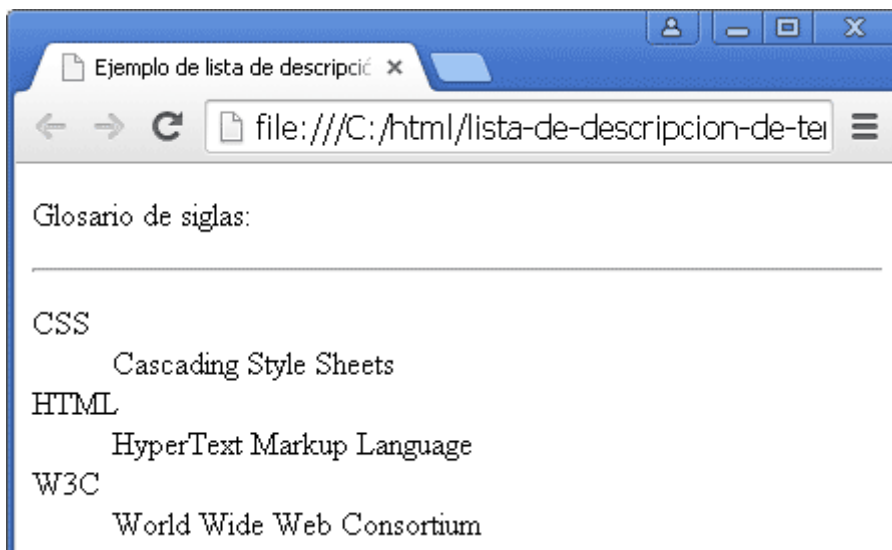
Para ello, se pueden utilizar los elementos "ol" y "li" ("*lista-ordenada.html*"):

```
<p>Lista de 5 animales ordenados de mayor a menor tamaño:</p>
<ol>
  <li>Elefante</li>
  <li>Cebra</li>
  <li>Oveja</li>
  <li>Gato</li>
  <li>Ratón</li>
</ol>
```

En una lista ordenada, los elementos están ubicados de forma que, si se cambiasen de lugar, cambiaría también el significado de la lista.

## Lista de descripción de términos - Elementos "dl", "dt" y "dd"

**EJEMPLO** Para mostrar un glosario de siglas:



Se puede escribir el siguiente código ("*lista-de-descripcion-de-terminos.html*"):

```
<!DOCTYPE html>
<html lang="es-ES">
<head>
  <meta charset="utf-8">
  <title>Ejemplo de lista de descripción de términos</title>
</head>
<body>
  <p>Glosario de siglas:</p>
  <hr>
  <dl>
    <dt>CSS</dt>
    <dd>Cascading Style Sheets</dd>
    <dt>HTML</dt>
    <dd>HyperText Markup Language</dd>
    <dt>W3C</dt>
    <dd>World Wide Web Consortium</dd>
  </dl>
</body>
</html>
```

Fíjese que, el elemento "dl" alberga a los grupos "término-descripción" (elemento "dt" - elemento "dd").

## Imágenes en HTML

Para visualizar una imagen en un documento HTML se puede utilizar el elemento "img".

**Nota:** Las imágenes no se insertan técnicamente en una página HTML, las imágenes están vinculadas a las páginas HTML. La etiqueta <img> crea un espacio de espera para la imagen referenciada.

**Consejo:** Para enlazar una imagen a otro documento, simplemente nido de la etiqueta <img> dentro <a> etiquetas.

Elemento "img"

Se puede escribir ("*una-imagen.html*"):

```
<!DOCTYPE html>
<html lang="es-ES">
<head>
  <meta charset="utf-8">
  <title>Ejemplo de una imagen</title>
</head>
<body>
  
</body>
</html>
```

Como se puede observar, el elemento "img" no necesita la etiqueta de cierre (</img>) ya que no tiene contenido. Ahora bien, véase que en "img" se han escrito varios atributos.

### Atributo src

El atributo src sirve para indicar la ubicación de la imagen que se quiere mostrar en el navegador web. En este caso, la imagen debe estar en la misma carpeta que el archivo "*una-imagen.html*".

## Atributos **width** y **height**

**width** y **height** permiten especificar, respectivamente, la anchura y la altura de la imagen en píxeles, esto provoca una transformación “forzosa” de la imagen, con lo que se aconseja solo usar estos atributos en casos excepcionales.

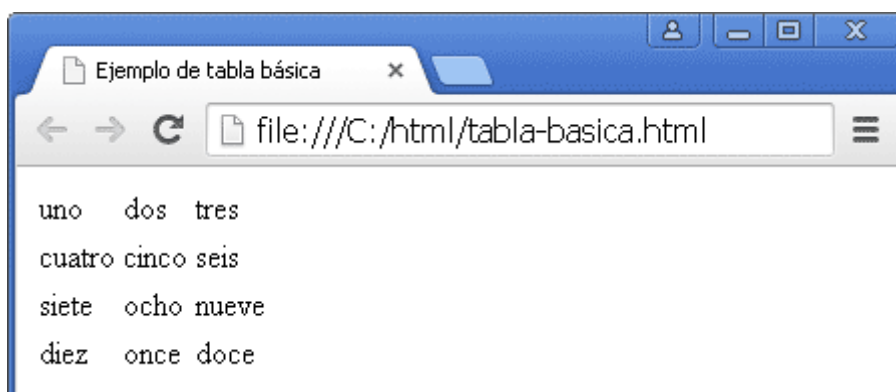
## Atributo **alt**

El atributo **alt** sirve para indicar un texto que se mostrará en pantalla en el caso de que la imagen no se encontrara. También es muy importante para tener una buena accesibilidad en la página, ya que el valor del atributo alt es el que se escucha por los altavoces para aquellas personas con dificultades en la vista y tiene instalado un lector de contenidos por voz.

## Tablas en HTML

Tabla básica - Elementos "table", "tr" y "td"

**EJEMPLO** En la siguiente tabla se muestran 12 datos:



|        |       |       |
|--------|-------|-------|
| uno    | dos   | tres  |
| cuatro | cinco | seis  |
| siete  | ocho  | nueve |
| diez   | once  | doce  |

Para ello, el código puede ser ("*tabla-basica.html*"):

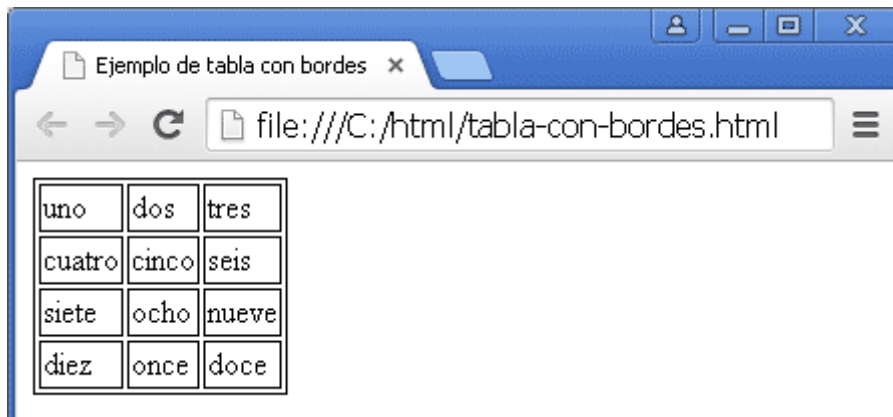
```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo de tabla básica</title>
  </head>
  <body>
    <table>
      <tr>
        <td>uno</td>
        <td>dos</td>
        <td>tres</td>
      </tr>
      <tr>
        <td>cuatro</td>
        <td>cinco</td>
        <td>seis</td>
      </tr>
      <tr>
        <td>siete</td>
        <td>ocho</td>
        <td>nueve</td>
      </tr>
      <tr>
        <td>diez</td>
        <td>once</td>
        <td>doce</td>
      </tr>
    </table>
  </body>
</html>
```

```
<td>diez</td>
<td>once</td>
<td>doce</td>
</tr>
</table>
</body>
</html>
```

Obsérvese que, cada fila de la tabla se representa por medio de un elemento "tr" y, dentro de cada fila, se ha escrito cada uno de los datos dentro de un "td".

### Tabla con bordes - Propiedad CSS border

**EJEMPLO** Por defecto, las tablas se muestran sin bordes. No obstante, para ver por ejemplo:



En "*tabla-con-bordes.html*" se ha definido la siguiente regla de estilo:

```
table, td {
    border:1px solid black;
}
```

### Tabla con celdas separadas - Propiedad CSS border-spacing

**EJEMPLO** Si se quiere separar el espacio que existe entre celdas, por ejemplo 10 píxeles:



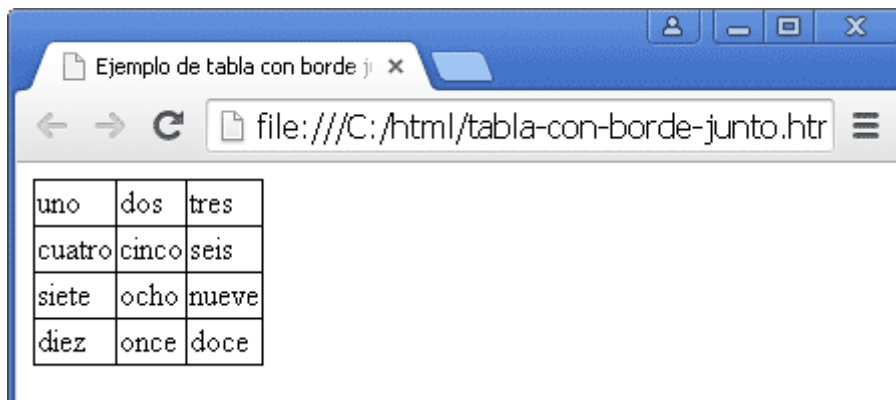
Se puede utilizar la propiedad CSS **border-spacing** ("*tabla-con-celdas-separadas.html*"):

```
table, td {
```

```
border:1px solid black;
}
table {
border-spacing:10px;
}
```

## Tabla con borde junto - Propiedad CSS **border-collapse**

**EJEMPLO** Para juntar las líneas del borde de una tabla, viéndose en pantalla:

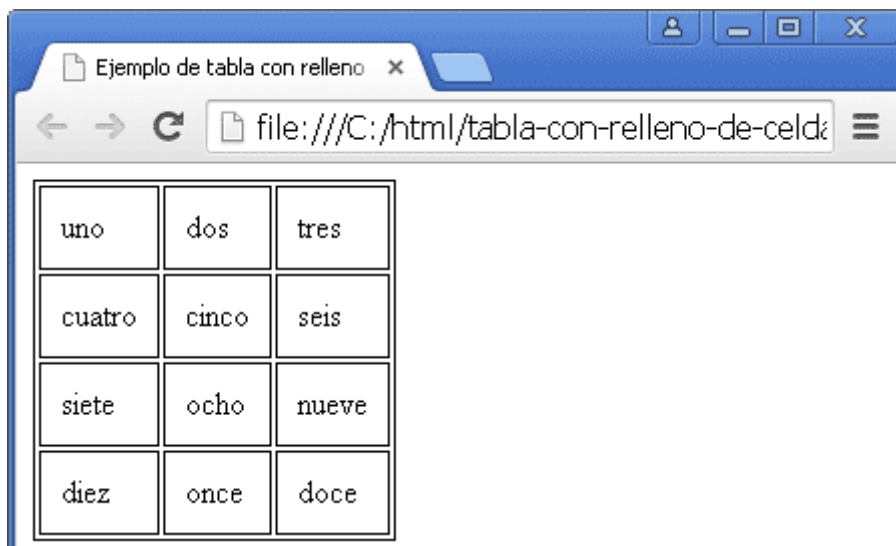


Se puede escribir ("*tabla-con-borde-junto.html*"):

```
table, td {
border:1px solid black;
}
table {
border-collapse:collapse;
}
```

## Tabla con relleno de celdas - Propiedad CSS **padding**

**EJEMPLO** Si se desea separar el contenido de las celdas de una tabla con sus respectivos bordes, por ejemplo 10 píxeles:



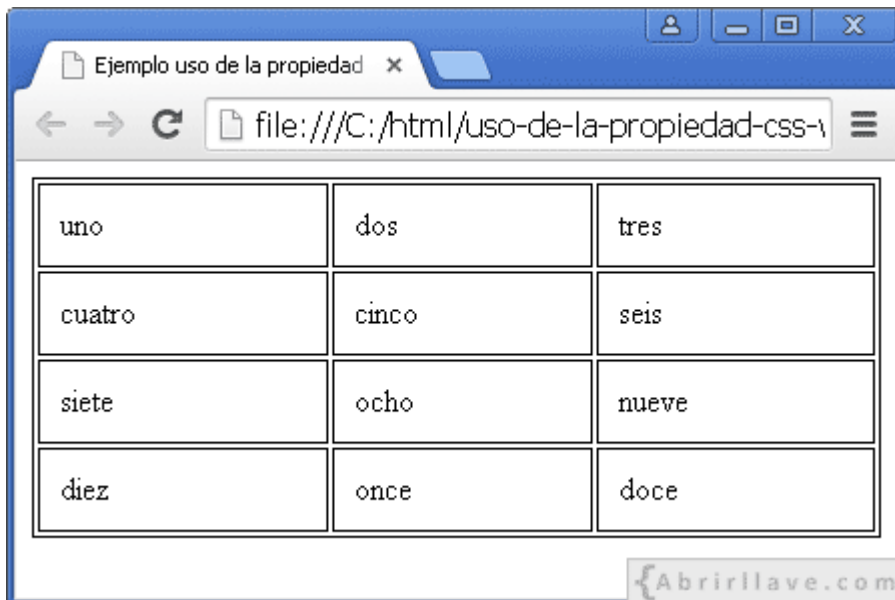
Se puede utilizar la propiedad CSS **padding** ("*tabla-con-relleno-de-celdas.html*"):

```
table, td {
```

```
border:1px solid black;  
}  
td {  
padding:10px;  
}
```

## Ancho de una tabla - Propiedad CSS **width**

**EJEMPLO** La propiedad CSS **width** permite definir el ancho de una tabla. Por ejemplo, para ver:

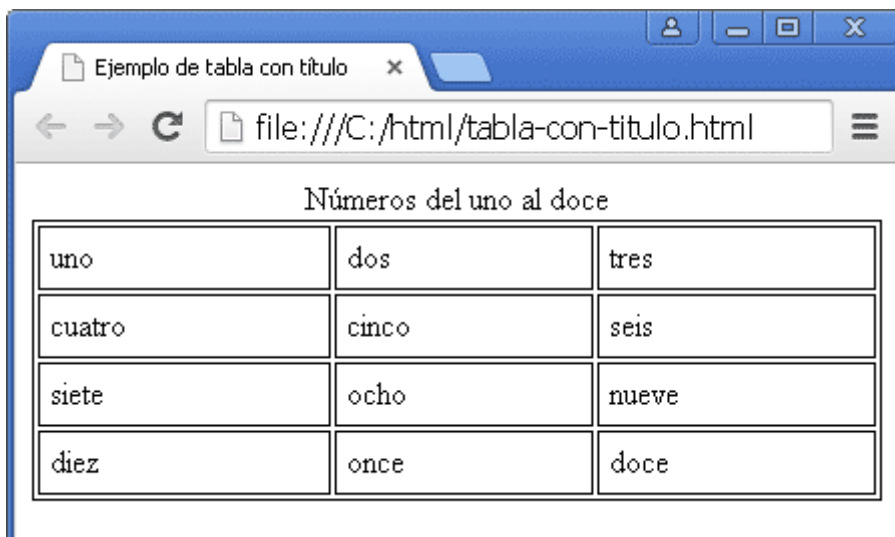


Se puede escribir ("*uso-de-la-propiedad-css-width-en-una-tabla.html*"):

```
table, td {  
border:1px solid black;  
}  
table {  
width:100%;  
}  
td {  
padding:10px;  
}
```

## Título de una tabla - Elemento "caption"

**EJEMPLO A** a una tabla se le puede indicar un título (o leyenda). Por ejemplo, la siguiente tabla lleva por título "Números del uno al doce":



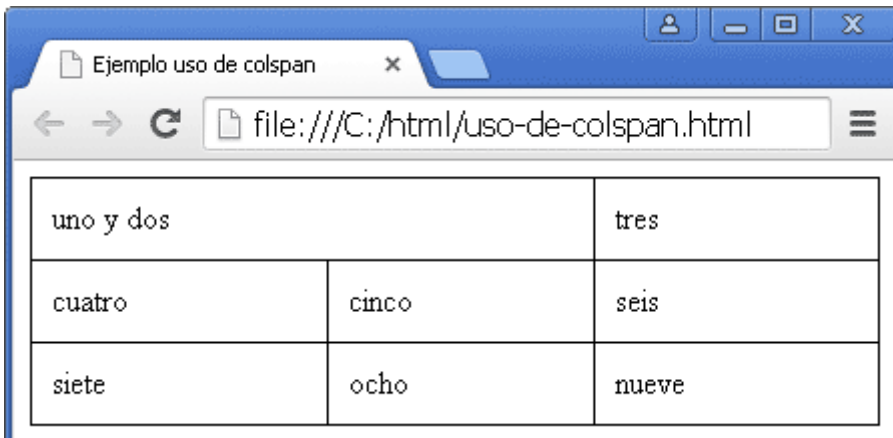
|        |       |       |
|--------|-------|-------|
| uno    | dos   | tres  |
| cuatro | cinco | seis  |
| siete  | ocho  | nueve |
| diez   | once  | doce  |

Para ello, se ha hecho uso del elemento "caption". Véase "*tabla-con-titulo.html*":

```
<table>
  <caption>Números del uno al doce</caption>
  <tr>
    <td>uno</td>
    <td>dos</td>
    <td>tres</td>
  </tr>
  <tr>
    <td>cuatro</td>
    <td>cinco</td>
    <td>seis</td>
  </tr>
  <tr>
    <td>siete</td>
    <td>ocho</td>
    <td>nueve</td>
  </tr>
  <tr>
    <td>diez</td>
    <td>once</td>
    <td>doce</td>
  </tr>
</table>
```

El elemento "caption" tiene que escribirse justo después de la etiqueta <table>.

Una celda puede abarcar varias columnas de una tabla - Atributo **colspan**  
**EJEMPLO** Con el atributo **colspan** se puede indicar que una celda abarque más de una columna:

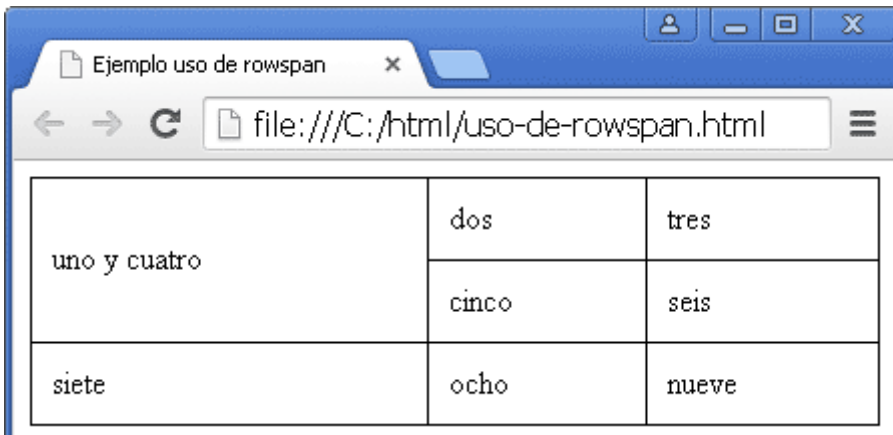


|           |       |       |
|-----------|-------|-------|
| uno y dos |       | tres  |
| cuatro    | cinco | seis  |
| siete     | ocho  | nueve |

Véase en el código del archivo "*uso-de-colspan.html*" que, al primer elemento "td" de la primera fila, se le ha indicado mediante **colspan="2"** que debe abarcar dos columnas de la tabla:

```
<table>
<tr>
  <td colspan="2">uno y dos</td>
  <td>tres</td>
</tr>
<tr>
  <td>cuatro</td>
  <td>cinco</td>
  <td>seis</td>
</tr>
<tr>
  <td>siete</td>
  <td>ocho</td>
  <td>nueve</td>
</tr>
</table>
```

Una celda puede abarcar varias filas de una tabla - Atributo **rowspan**  
**EJEMPLO** Con el atributo **rowspan** se puede indicar que una celda abarque más de una fila:



|              |       |       |
|--------------|-------|-------|
| uno y cuatro | dos   | tres  |
|              | cinco | seis  |
| siete        | ocho  | nueve |



Fíjese en el archivo "*uso-de-rowspan.html*" que, al primer elemento "td" de la primera fila, se le ha indicado mediante `rowspan="2"` que debe abarcar dos filas de la tabla:

```
<table>
<tr>
  <td rowspan="2">uno y cuatro</td>
  <td>dos</td>
  <td>tres</td>
</tr>
<tr>
  <td>cinco</td>
  <td>seis</td>
</tr>
<tr>
  <td>siete</td>
  <td>ocho</td>
  <td>nueve</td>
</tr>
</table>
```

### Cabeceras de una tabla - Elemento "th"

**EJEMPLO** La siguiente tabla tiene tres celdas cabecera, las cuales han sido definidas con el elemento "th":



| Ciudad   | Pais   | Continente |
|----------|--------|------------|
| Roma     | Italia | Europa     |
| El Cairo | Egipto | África     |
| Tokio    | Japón  | Asia       |

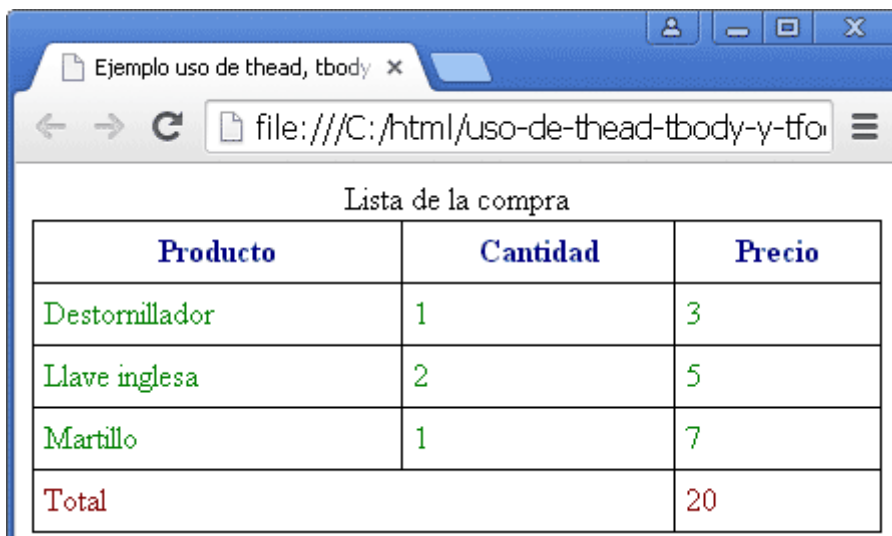
Obsérvese que, por defecto, el contenido de un elemento "th" se visualiza en negrita y centrado ("*tabla-con-cabeceras.html*").

```
<!DOCTYPE html>
<html lang="es-ES">
<head>
  <meta charset="utf-8">
  <title>Ejemplo de tabla con cabeceras</title>
  <style>
    table, td, th {
      border:1px solid black;
    }
    table {
      border-collapse:collapse;
      width:100%;
    }
    td, th {
```

```
padding:5px;
}
</style>
</head>
<body>
<table>
<tr>
<th>Ciudad</th>
<th>País</th>
<th>Continente</th>
</tr>
<tr>
<td>Roma</td>
<td>Italia</td>
<td>Europa</td>
</tr>
<tr>
<td>El Cairo</td>
<td>Egipto</td>
<td>Àfrica</td>
</tr>
<tr>
<td>Tokio</td>
<td>Japón</td>
<td>Asia</td>
</tr>
</table>
</body>
</html>
```

### Estructura de una tabla - Elementos "thead", "tbody" y "tfoot"

**EJEMPLO** Los elementos "thead", "tbody" y "tfoot" permiten estructurar los datos de una tabla y que sobre ellos se apliquen estilos distintos. Véase por ejemplo:



Lista de la compra

| Producto       | Cantidad | Precio    |
|----------------|----------|-----------|
| Destornillador | 1        | 3         |
| Llave inglesa  | 2        | 5         |
| Martillo       | 1        | 7         |
| <b>Total</b>   |          | <b>20</b> |

Para ello, se ha escrito ("*uso-de-thead-tbody-y-tfoot.html*"):

```
<!DOCTYPE html>
<html lang="es-ES">
<head>
  <meta charset="utf-8">
  <title>Ejemplo uso de thead, tbody y tfoot</title>
  <style>
    table, td, th {
      border:1px solid black;
    }
    table {
      border-collapse:collapse;
      width:100%;
    }
    td, th {
      padding:5px;
    }
    thead {
      color:navy;
    }
    tbody {
      color:green;
    }
    tfoot {
      color:maroon;
    }
  </style>
</head>
<body>
  <table>
    <caption>Lista de la compra</caption>
    <thead>
      <tr>
        <th>Producto</th>
        <th>Cantidad</th>
        <th>Precio</th>
      </tr>
    </thead>
    <tfoot>
      <tr>
        <td colspan="2">Total</td>
        <td>20</td>
      </tr>
    </tfoot>
    <tbody>
      <tr>
        <td>Destornillador</td>
        <td>1</td>
        <td>3</td>
      </tr>
      <tr>
        <td>Llave inglesa</td>
        <td>2</td>
        <td>5</td>
      </tr>
    </tbody>
  </table>
```

```
<tr>
  <td>Martillo</td>
  <td>1</td>
  <td>7</td>
</tr>
</tbody>
</table>
</body>
</html>
```

Los elementos "thead", "tbody" y "tfoot" pueden agrupar, respectivamente:

- una o más filas de cabeceras,
- una o más filas de datos del cuerpo de la tabla y
- una o más filas de datos del pie de la tabla.

Por otro lado, hay que saber que, en una tabla puede aparecer más de un elemento "tbody".

Además, hay que tener en cuenta que, el elemento "tfoot" debe escribirse después del "thead" y antes de todos los "tbody" presentes en la tabla.

## Divisiones en HTML

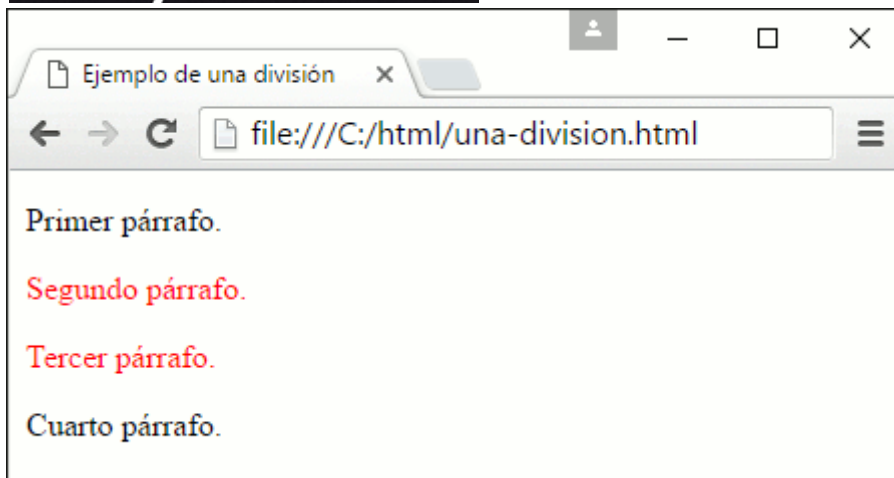
En un documento HTML el elemento "div" permite crear divisiones, también llamadas secciones o zonas. Las divisiones se utilizan para agrupar elementos y aplicarles estilos.

Elemento "div"

**EJEMPLO** En el siguiente documento ("*una-division.html*") el segundo y tercer párrafo están agrupados dentro de un elemento "div" donde se ha establecido que dichos párrafos deben mostrarse de color rojo:

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo de una división</title>
  </head>
  <body>
    <p>Primer párrafo.</p>
    <div style="color:red">
      <p>Segundo párrafo.</p>
      <p>Tercer párrafo.</p>
    </div>
    <p>Cuarto párrafo.</p>
  </body>
</html>
```

Al visualizarlo en pantalla, se verá algo similar a:



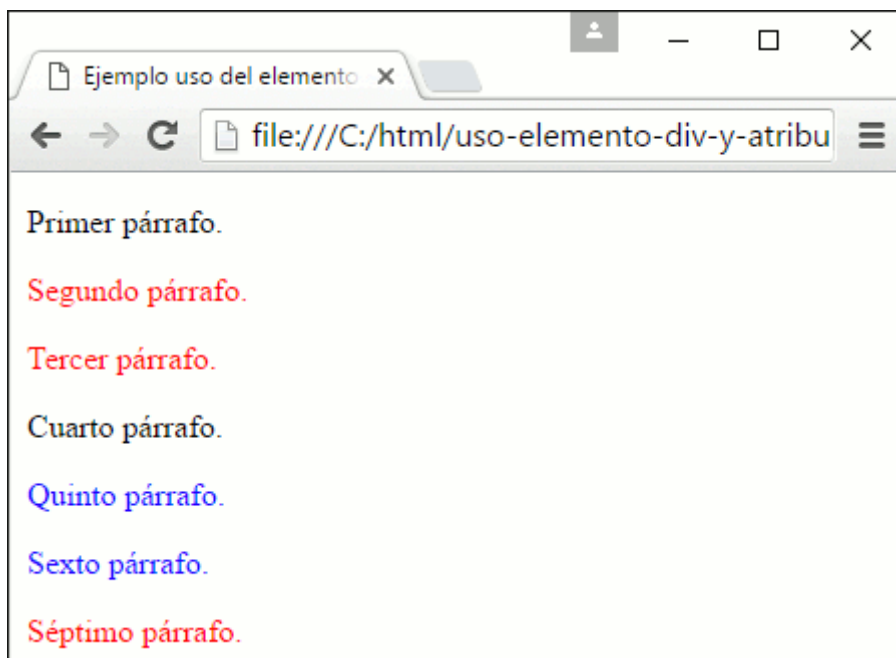
## Elemento "div" y atributo class

**EJEMPLO** En el siguiente documento HTML ("*uso-elemento-div-y-atributo-class.html*") se han escrito tres divisiones haciendo uso del atributo class. Obsérvese que, en dos de dichas divisiones el valor del atributo class es "rojo" y en la otra "azul".

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo uso del elemento div y del atributo class</title>
    <style>
      .rojo {color:red;}
      .azul {color:blue;}
    </style>
  </head>
  <body>
    <p>Primer párrafo.</p>
    <div class="rojo">
      <p>Segundo párrafo.</p>
      <p>Tercer párrafo.</p>
    </div>
    <p>Cuarto párrafo.</p>
    <div class="azul">
      <p>Quinto párrafo.</p>
      <p>Sexto párrafo.</p>
    </div>
```

```
<div class="rojo">
  <p>Séptimo párrafo.</p>
</div>
</body>
</html>
```

En un navegador se visualizará:



## Elemento "div" y atributo id

**EJEMPLO** En el código del siguiente documento ("*uso-elemento-div-y-atributo-id.html*") se han especificado cuatro divisiones (**cabecera**, **contenido**, **menu** y **pie**) contenidas dentro de otra división llamada **contendor**.

```
<!DOCTYPE html>
<html lang="es-ES">
<head>
  <meta charset="utf-8">
  <title>Ejemplo uso del elemento div y del atributo id</title>
  <style>
    *{
      margin:0;
      padding:0;
    }
    a:link, a:visited, a:hover, a:active {
      color:#0f0;
      font-size:16px;
    }
    body {
      background:#eee;
      font-family:verdana;
    }
    h1 {
```

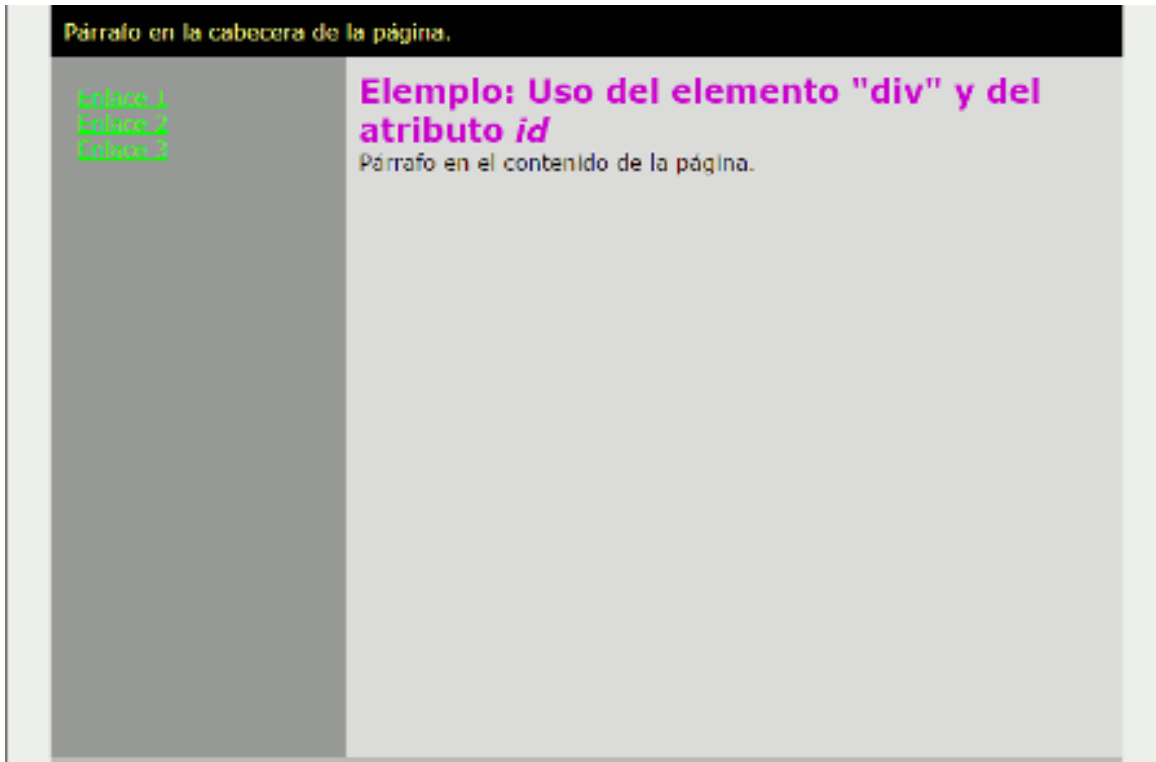
```

    color:#c0c;
    font-size:24px;
  }
  p {
    font-size:16px;
  }
  ul {
    list-style-type:none;
  }
  #cabecera {
    color:#ff9;
    background-color:#000;
    padding:10px;
  }
  #contenedor {
    margin:0 auto;
    width:800px;
  }
  #contenido {
    background-color:#ddd;
    float:left;
    height:500px;
    padding:10px;
    width:560px;
  }
  #menu {
    background-color:#999;
    float:left;
    height:480px;
    padding:20px;
    width:180px;
  }
  #pie {
    background-color:#bbb;
    clear:both;
    color:#900;
    padding:10px;
    text-align:center;
  }
</style>
</head>
<body>
<div id="contenedor">
  <div id="cabecera">
    <p>Pàrrafo en la cabecera de la pàgina.</p>
  </div>
  <div id="menu">
    <ul>
      <li><a href="enlace-1.html">Enlace 1</a></li>
      <li><a href="enlace-2.html">Enlace 2</a></li>
      <li><a href="enlace-3.html">Enlace 3</a></li>
    </ul>
  </div>
  <div id="contenido">
    <h1>Ejemplo: Uso del elemento &quot;div&quot; y del atributo <i>id</i></h1>

```

```
<p>Pàrrafo en el contenido de la página.</p>
</div>
<div id="pie">
  <p>Pàrrafo en el pie de la página.</p>
</div>
</div>
</body>
</html>
```

En pantalla se mostrará algo parecido a:





## HTML5-Componentes básicos

HTML5 no es una nueva versión del antiguo lenguaje de etiquetas, ni siquiera una mejora de esta ya antigua tecnología, sino un nuevo concepto para la construcción de sitios web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red.

Todo comenzó mucho tiempo atrás con una simple versión de HTML propuesta para crear la estructura básica de páginas web, organizar su contenido y compartir información. El lenguaje y la web misma nacieron principalmente con la intención de comunicar información por medio de texto.

El limitado objetivo de HTML motivó a varias compañías a desarrollar nuevos lenguajes y programas para agregar características a la web nunca antes implementadas. Estos desarrollos iniciales crecieron hasta convertirse en populares y poderosos accesos. Simples juegos y bromas animadas pronto se transformaron en sofisticadas aplicaciones, ofreciendo nuevas experiencias que cambiaron el concepto de la web para siempre.

De las opciones propuestas, Java y Flash fueron las más exitosas; ambas fueron masivamente adoptadas y ampliamente consideradas como el futuro de Internet. Sin embargo, tan pronto como el número de usuarios se incrementó e Internet pasó de ser una forma de conectar amantes de los ordenadores a un campo estratégico para los negocios y la interacción social, limitaciones presentes en estas dos tecnologías probaron ser una sentencia de muerte.

El mayor inconveniente de Java y Flash puede describirse como una falta de integración. Ambos fueron concebidos desde el principio como complementos (plug-ins), algo que se inserta dentro de una estructura pero que comparte con la misma solo espacio en la pantalla. No existía comunicación e integración alguna entre aplicaciones y documentos. La falta de integración resultó ser crítica y preparó el camino para la evolución de un lenguaje que comparte espacio en el documento con HTML y no está afectado por las limitaciones de los plug-ins.

Javascript, un lenguaje interpretado incluido en navegadores, claramente era la manera de mejorar la experiencia de los usuarios y proveer funcionalidad para la web. Sin embargo, después de algunos años de intentos fallidos para promoverlo y algunos malos usos, el mercado nunca lo adoptó plenamente y pronto su popularidad declinó. Los detractores tenían buenas razones para oponerse a su adopción. En ese momento, Javascript no era capaz de reemplazar la funcionalidad de Flash o Java.

A pesar de ser evidente que ambos limitaban el alcance de las aplicaciones y aislaban el contenido web, populares funciones como la reproducción de video se estaban convirtiendo en una parte esencial de la web y solo eran efectivamente ofrecidas a través de estas tecnologías.

A pesar del suceso inicial, el uso de Java comenzó a declinar. La naturaleza compleja del lenguaje, su evolución lenta y la falta de integración disminuyeron su importancia hasta el punto en el que hoy día no es más usado en aplicaciones web de importancia. Sin Java, el mercado volcó su atención a Flash. Pero el hecho de que Flash comparte las mismas características básicas que su competidor en la web lo hace también susceptible de correr el mismo destino.

Mientras esta competencia silenciosa se llevaba a cabo, el software para acceder a la web continuaba evolucionando. Junto con nuevas funciones y técnicas rápidas de acceso a la red, los navegadores también mejoraron gradualmente sus intérpretes Javascript. Más potencia trajo más oportunidades y este lenguaje estaba listo para aprovecharlas.

En cierto punto durante este proceso, se hizo evidente para algunos desarrolladores que ni Java o Flash podrían proveer las herramientas que ellos necesitaban para crear las aplicaciones demandadas por un número creciente de usuarios.

Estos desarrolladores, impulsados por las mejoras otorgadas por los navegadores, comenzaron a aplicar Javascript en sus aplicaciones de un modo nunca visto. La innovación y los increíbles resultados obtenidos llamaron la atención de más programadores. Pronto lo que fue llamado la “Web 2.0” nació y la percepción de Javascript en la comunidad de programadores cambió radicalmente. Javascript era claramente el lenguaje que permitía a los desarrolladores innovar y hacer cosas que nadie había podido hacer antes en la web. En los últimos años, programadores y diseñadores web alrededor del mundo surgieron con los más increíbles trucos para superar las limitaciones de esta tecnología y sus iniciales deficiencias en portabilidad.

Gracias a estas nuevas implementaciones, Javascript, HTML y CSS se convirtieron pronto en la más perfecta combinación para la necesaria evolución de la web. HTML5 es, de hecho, una mejora de esta combinación, el pegamento que une todo.

HTML5 propone estándares para cada aspecto de la web y también un propósito claro para cada una de las tecnologías involucradas. A partir de ahora, HTML provee los elementos estructurales, CSS se encuentra concentrado en cómo volver esa estructura utilizable y atractiva a la vista, y Javascript tiene todo el poder necesario para proveer dinamismo y construir aplicaciones web completamente funcionales. Las barreras entre sitios webs y aplicaciones finalmente han desaparecido.

Las tecnologías requeridas para el proceso de integración están listas. El futuro de la web es prometedor y la evolución y combinación de estas tres tecnologías (HTML, CSS y Javascript) en una poderosa especificación está volviendo a Internet la plataforma líder de desarrollo. HTML5 indica claramente el camino.

## Estructura del cuerpo

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

HTML siempre ofreció 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 la parte 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.

## Organización

La Figura 1-1 representa un diseño común encontrado en la mayoría de los sitios webs estos días. Apesar del hecho de que cada diseñador crea sus propios diseños, en general podremos identificar las siguientes secciones en cada sitio web estudiado:



**Figura 1-1.** 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 1-1 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...

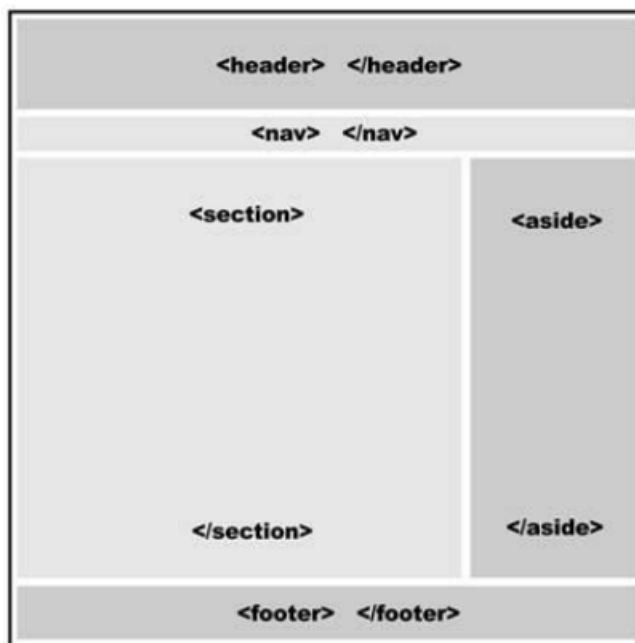
En la base de un diseño web clásico siempre nos encontramos con una barra más que aquí llamamos **Institucional**. La nombramos de esta manera porque esta es el área en donde normalmente se muestra información acerca del sitio web, el autor o la empresa, además de algunos enlaces con respecto a reglas, 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:



**Figura 1-2.** Representación visual de un clásico diseño para blogs.

La Figura 1-2 es una representación de un blog normal. En este ejemplo se puede claramente identificar cada parte del diseño considerado anteriormente. 1. Cabecera 2. Barra de Navegación 3. Sección de Información Principal 4. Barra Lateral 5. 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. A veces este propósito no es claro pero en esencia se encuentra siempre allí, ayudándonos a reconocer cualquiera de las secciones descritas anteriormente en todo diseño. 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:



**Figura 1-3.** Representación visual de un diseño utilizando elementos HTML5

La Figura 1-3 muestra el típico diseño presentado anteriormente, pero esta vez con los correspondientes elementos HTML5 para cada sección (incluyendo etiquetas de apertura y cierre).

### <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>
```

<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 el 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, se recomienda 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.



## <section>

Siguiendo nuestro diseño estándar nos encontramos con las columnas que en la Figura 1-1 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>
```

## <aside>

En un típico diseño web (Figura 1-1) 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 de la Figura 1-2, los enlaces apuntan a cada una de las entradas del blog y ofrecen información adicional sobre el autor (número 4).

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 podría estar ubicado del lado derecho o izquierdo de nuestra página de ejemplo, la etiqueta no tiene una posición predefinida. El elemento 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 dentro del elemento 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; 2010-2011
</footer>
</body>
</html>
```

En el típico diseño de una página web (Figura 1-1) 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 cuerpo

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.

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.

En la especificación HTML5, HTML está a cargo de la estructura del documento y provee un grupo completo de nuevos elementos para este propósito. La especificación también incluye algunos elementos con la única tarea de proveer estilos.

Esta es una lista de los que consideramos más relevantes:

**IMPORTANTE:** Para una completa referencia de los elementos HTML incluidos en la especificación, visite nuestro sitio web y siga los enlaces correspondientes a este capítulo.

**<header>** Este elemento presenta información introductoria y puede ser aplicado en diferentes secciones del documento. Tiene el propósito de contener la cabecera de una sección pero también puede ser utilizado para agrupar índices, formularios de búsqueda, logos, etc...

**<nav>** Este elemento indica una sección de enlaces con propósitos de navegación, como menús o índices. No todos los enlaces dentro de una página web tienen que estar dentro de un elemento **<nav>**, solo aquellos que forman partes de bloques de navegación.

**<section>** Este elemento representa una sección general del documento. Es usualmente utilizado para construir varios bloques de contenido (por ejemplo, columnas) con el propósito de ordenar el contenido que comparte una característica específica, como capítulos o páginas de un libro, grupo de noticias, artículos, etc...

**<aside>** Este elemento representa contenido que está relacionado con el contenido principal pero no es parte del mismo. Ejemplos pueden ser citas, información en barras laterales, publicidad, etc...

**<footer>** Este elemento representa información adicional sobre su elemento padre. Por ejemplo, un elemento **<footer>** insertado al final del cuerpo proveerá información adicional sobre el cuerpo del documento, como el pie normal de una página web. Puede ser usado no solo para el cuerpo sino también para diferentes secciones dentro del cuerpo, otorgando información adicional sobre estas secciones específicas.

**<article>** Este elemento representa una porción independiente de información relevante (por ejemplo, cada artículo de un periódico o cada entrada de un blog). El elemento **<article>** puede ser anidado y usado para mostrar una lista dentro de otra lista de ítems relacionados, como comentarios de usuarios en entradas de blogs, por ejemplo.

**<hgroup>** Este elemento es usado para agrupar elementos H cuando la cabecera tiene múltiples niveles (por ejemplo, una cabecera con título y subtítulo).

**<figure>** Este elemento representa una porción independiente de contenido (por ejemplo, imágenes, diagramas o videos) que son referenciadas desde el contenido principal. Esta es información que puede ser removida sin afectar el fluido del resto del contenido.

<figcaption> Este elemento es utilizado para mostrar una leyenda o pequeño texto relacionado con el contenido de un elemento <figure>, como la descripción de una imagen.

<mark> Este elemento resalta un texto que tiene relevancia en una situación en particular o que ha sido mostrado en respuesta de la actividad del usuario.

<small> Este elemento representa contenido al margen, como letra pequeña (por ejemplo, descargos, restricciones legales, declaración de derechos, etc...).

<cite> Este elemento es usado para mostrar el título de un trabajo (libro, película, poema, etc...).

<address> Este elemento encierra información de contacto para un elemento <article> o el documento completo. Es recomendable que sea insertado dentro de un elemento <footer>.

<time> Este elemento se utiliza para mostrar fecha y hora en formatos comprensibles por los usuarios y el navegador. El valor para los usuarios es ubicado entre las etiquetas mientras que el específico para programas y navegadores es incluido como el valor del atributo datetime. Un segundo atributo optativo llamado pubdate es usado para indicar que el valor de datetime es la fecha de publicación.

## Etiqueta HTML5 video: Embeber vídeos con su propio player

Hasta la llegada de **HTML5** no había ningún estandar para la **reproducción de vídeos ni audios** en la web, por lo que la mayoría de webs optaban por usar el plugin de flash, QuicTime, Windows Media...

Con la **etiqueta video** es muy sencillo crear el player para reproducir vídeo, os muestro una imagen en la que podemos apreciar un vídeo con su player con tan solo 3 líneas de código.

Otras etiquetas multimedia interesantes son:

- **SVG** para realizar gráficos vectoriales.
- **audio** que es similar que video pero para reproducir audios.

## ¿Qué navegadores soportan el tag de HTML5 video?

La compatibilidad de los principales navegadores con el **vídeo en HTML5** es la siguiente:

- **Internet Explorer:** Desde la versión 9.0 y posteriores.
- **Mozilla Firefox:** Desde la versión 3.5 y posteriores.
- **Google Chrome:** Desde la versión 4.0 y posteriores.
- **Safari:** Desde la versión 4.0 y posteriores.

## Formatos de vídeo compatibles

Tenemos 3 formatos compatibles siendo el MP4 el más soportado por los navegadores:

- **MP4:** Compatible con Google Chrome, Mozilla Firefox, Safari e Internet Explorer.
- **Ogg:** Compatible con Google Chrome y Mozilla Firefox.
- **WebM:** Compatible con Google Chrome y Mozilla Firefox.

Los media type para referenciar cada uno de estos 3 formatos son:

- **MP4:** video/mp4
- **Ogg:** video/ogg
- **WebM:** video/webm

Os aconsejo usar este [free vídeo converter](#)



## Sintaxis básica de la etiqueta de HTML5 video

Veamos la sintaxis básica:

```
<video width="600" height="400">
  <source src="video.mp4" type="video/mp4">
</video>
```

Tal y como se aprecia en el código superior con tan solo 2 etiquetas podemos crear el player:

- Con **<video>** indicamos que vamos que se va a embeber un vídeo, podemos asignar un alto y ancho al reproductor.
- Con **<source>** especificamos la ruta del vídeo a reproducir.

## Atributos de la etiqueta de HTML5 video

Os listo los atributos disponibles,

- **autoplay:** Con este atributo se especifica que el vídeo comience automáticamente a reproducirse. Se puede especificar de estas 2 maneras **autoplay="autoplay" o autoplay**.
- **preload:** En caso de omisión su valor es "auto". Sirve para especificar si queremos que el vídeo se vaya cargando independientemente si se da al play o no. Aumentaría el consumo de ancho de banda pero el usuario tendría menos cortes en la reproducción. Tenemos 3 posibles valores para este atributo:
  - preload = "none": El vídeo no comienza a descargarse hasta que el usuario haga play.
  - preload = "metadata": Solo se cargan los metadatos del vídeo (dimensiones, primer frame, duración...).
  - preload = "auto": El vídeo se descarga independientemente de la ejecución del play.
- **controls:** Con este atributo especificamos si se muestran los típicos controles del vídeo (play, stop, volumen...). Tiene 2 posibles sintaxis: **controls="controls" o controls**.
- **loop:** Especifica la reproducción continua del vídeo, es decir, una vez acabada la reproducción comienza de nuevo a reproducirse. Tiene 2 posibles sintaxis: **loop="loop" o loop**.
- **poster:** URL de la imagen a mostrar cuando el vídeo no está disponible, está cargando y no ha comenzado su reproducción. Por defecto es el primer frame del vídeo.
- **height y width:** El alto y ancho del vídeo para mostrarse en la web.
- **muted:** Funcionalidad para quitar el volumen al vídeo.
- **mediagroup:** Establece un grupo de reproducción al que pertenece el vídeo.
- **'src'** de la etiqueta source: URL/path del vídeo a reproducir.