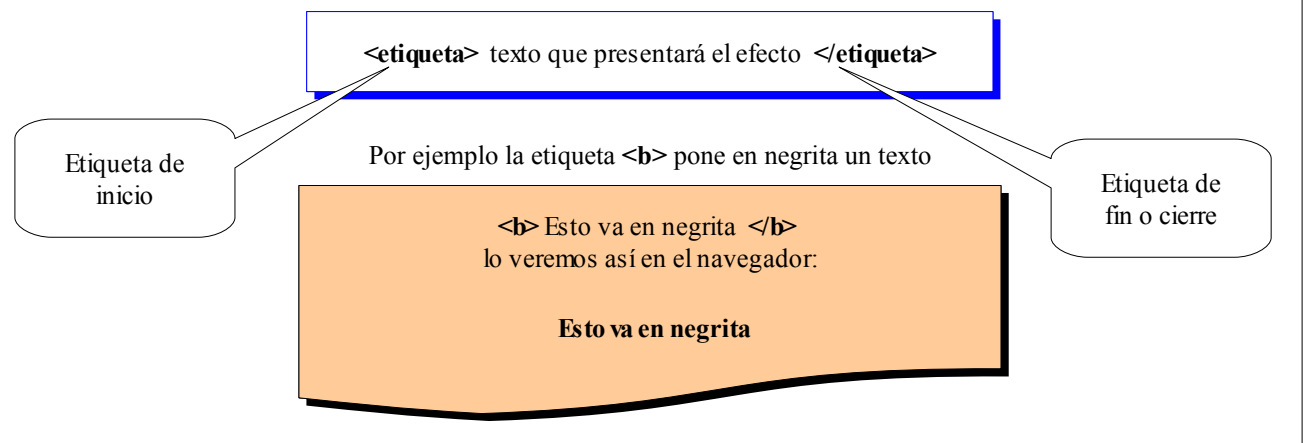


**HTML** Son las iniciales de la expresión en inglés **HyperText Markup Language**. Traducido al castellano sería **Lenguaje de Etiquetas de Hipertexto**. Se trata de un conjunto de etiquetas que se van intercalando entre el texto de forma que los programas que utilizamos para navegar por la Red sepan qué es lo que tienen que mostrar cuando accedemos a una página y cómo deben presentarlo en la pantalla. El W3C (World Wide Web Consortium) es el fórum internacional que se encarga desarrollar nuevas tecnologías relacionadas con la WEB dictando las normas que constituyen el estándar HTML entre otros. Con HTML podemos utilizar textos, sonidos, imágenes, y lo más importante, enlaces a otras páginas. Este lenguaje surge en principio con fines divulgativos y no se tuvo en cuenta que la web llegara a ser un área de ocio con carácter multimedia, debido a esto, el HTML ha ido necesitando actualizaciones con el paso del tiempo y la introducción de otras tecnologías complementarias como es el caso de **CSS** y **JavaScript**.

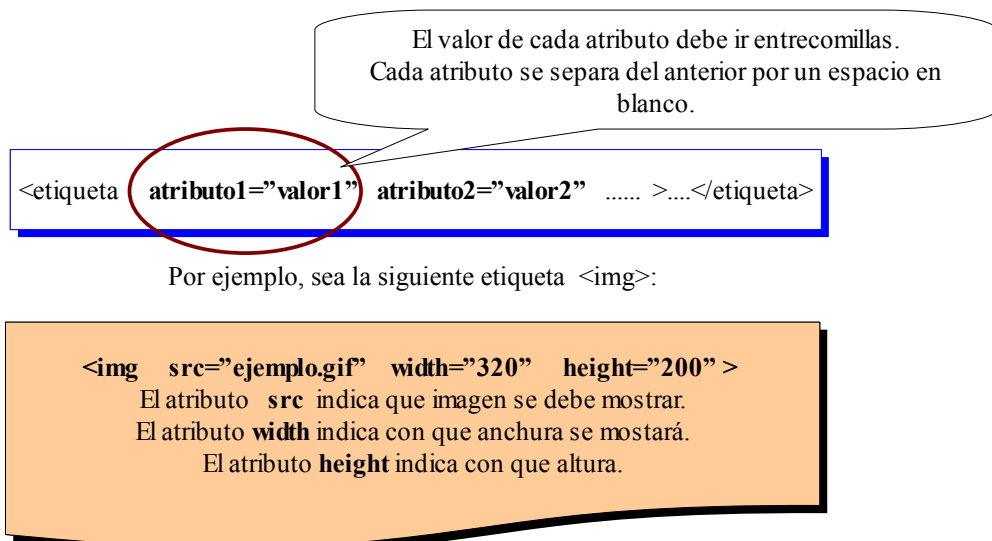
Una página web es un archivo que contiene código HTML en forma de texto. Estos archivos tienen la extensión “.html” o “.htm”. Para poder crear un archivo .html (pagina web) podemos hacerlo desde:

- Un simple editor de texto como el Bloc de notas de Windows , Wordpad , gedit , etc...
- Una suite ofimática que exporte documentos en html como Microsoft Office , OpenOffice , etc...
- Un editor de de páginas web como NVU , Dreamweaver , Frontpage , etc..

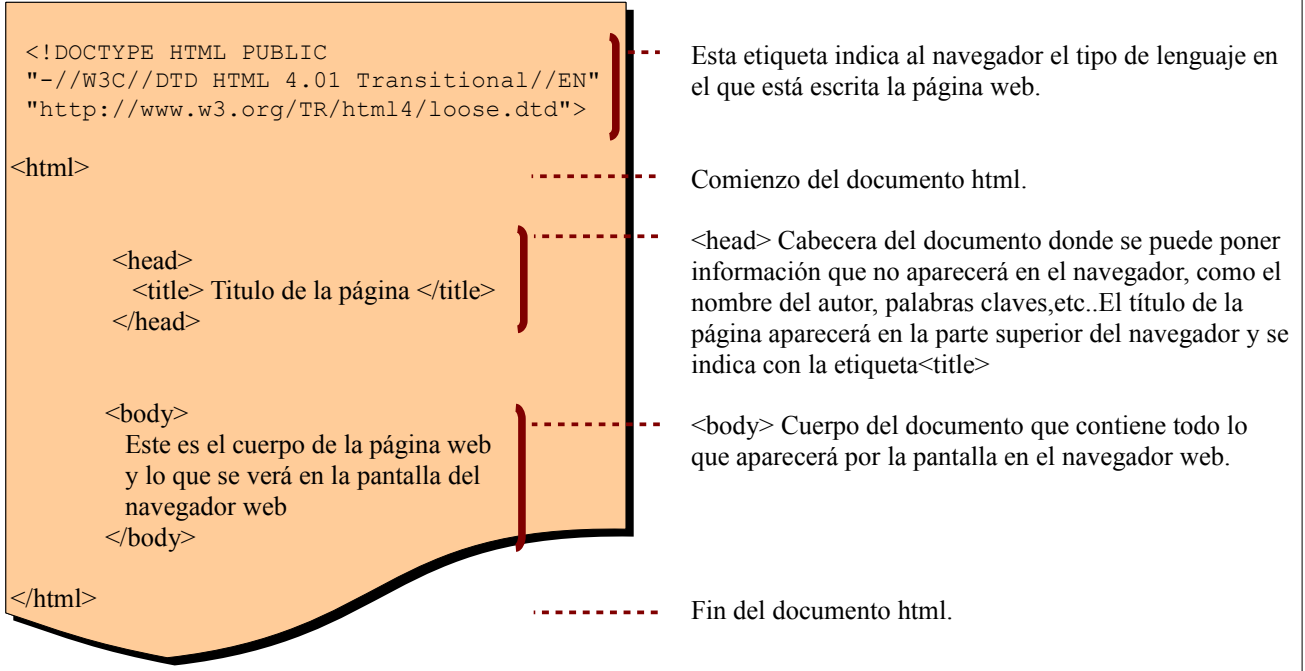
**<etiquetas>** Las etiquetas HTML se escriben encerradas entre ángulos y en **minúsculas** para que sean compatibles con el nuevo lenguaje XHTML . Normalmente se utilizan dos etiquetas: una de inicio y otra de fin para indicar que ha terminado el efecto que queríamos presentar. La única diferencia entre ambas es que la de cierre lleva una barra inclinada "/" antes del código.



**Atributos de etiquetas** Las etiquetas necesitan especificar información exacta como, por ejemplo, dónde está ubicado un archivo. Los atributos se utilizan con etiquetas para proporcionar más información sobre cómo debe implementarse la etiqueta.



## Estructura básica de un archivo HTML



## Comentarios

Un comentario es información que se incluye en el código html de una página web que no será visualizado en el navegador web, pero que es de mucha utilidad para el programador durante el desarrollo de la misma o en futuras revisiones. Se indican entre los símbolos <!-- y -->.

```
<!-- esto es un comentario y no se verá -->
```

## Caracteres especiales

Cuando dentro del cuerpo <body> del documento html tiene que aparecer algún carácter especial (como ñ, <, >, &, á, é, ...), para que se visualice correctamente en el navegador web debe de sustituirse en el código html por los siguientes valores:

Símbolo	Código
<	&lt;
>	&gt;
á	&aacute;
Á	&Aacute;
é	&eacute;
É	&Eacute;
&	&amp;
“	&quot;
ñ	&ntilde;
Ñ	&Ntilde;
;	&#191;

estos son sólo algunos. existen cientos.

```
<body> Esto es el cuerpo de la p&aacute;gina web </body>
```

## Metainformación

La metainformación es información adicional insertada en el código html que puede tener algunos usos como por ejemplo ser usada en aplicaciones externas, buscadores, índices, etc.. Se indican en la cabecera <head> con la etiqueta <meta> y puede tener tres atributos:

- **name**: identifica la información que se va a exportar.
- **content**: contenido de la información.
- **http-equiv**: es el nombre de un comando HTTP.

```
<meta name="author" content="Su nombre aquí">
<meta name="copyright" content="ingrese aquí el nombre de su compañía">
<meta name="contact_addr" content="aquí su dirección email">
<meta name="rating" content="14 years">
  <!-- rating: Para indicar hacia qué tipo de audiencia
    está orientado su sitio web disponemos de
    cuatro posibles valores para el content:
    *general: para todas las edades.
    *14 years: para mayores de 14 años.
    *mature: para adultos.
    *restricted: para adultos con restricciones. -->
<meta name="expires" content="never">
<meta name="language" content="es">
<meta name="charset" content="iso-8859-1">
<meta name="distribution" content="global">
<meta name="robot" content="Ingrese el valor aquí">
  <!-- robot: Cuando un motor de búsqueda visite su
    página tratará de seguir sus links e indexará cada
    página que encuentre. Posibles valores:
    *all: dirá al robot del motor de búsqueda que
    siga todos los links sobre su página web,
    e indexe todos los que encuentre.
    *noindex: para que el robot no indexe esa página.
    Pero puede seguir los links sobre ella.
    *nofollow: que indexe esta página pero que no
    siga los links sobre ella.
    *none: que no indexe esta página ni siga los
    links sobre ella. -->
<meta name="revisit-after" content="ingrese el valor aquí">
  <!-- revisit-after: para indicar a un robot cuando debe
    regresar e indexar el sitio de nuevo.
    Los posibles valores pueden ser la cantidad de tiempo
    que usted quiera: "10 days", "2 months", "1 week",... -->
<meta name="keywords" content="holistic, massage, reflexology, shiatsu,
manipulation, stress, stressbuster, relax, relaxation, ease, pain reduction, pain ">
  <!-- keywords=palabras clave. sirven para las búsquedas
    por clave de los buscadores. -->
<meta name="generator" content="dreamweaver 4.0 macromedia">
  <!-- indicamos el software utilizado para la creación de la página. -->
<meta http-equiv="refresh" content="15;url=http://www.mas.com">
  <!-- pasados 15 segundos se accederá automáticamente a la web indicada-->
<meta http-equiv="content-type" content="text/html; charset="iso-8859-1">
  <!-- definimos el tipo de contenido: text/html y los caracteres especiales
    utilizados, en este caso los occidentales (iso) -->
```

## Texto

Existen diversas etiquetas para poder indicar la distribución , forma , tamaño , color o tipo de fuente del texto.

```
<b> negrita </b>
<i> cursiva inclinada </i>
<tt> maquina </tt>
<strong> texto resaltado </strong>
<u> subrayado </u>
<strike> tachada</strike>
<sub> subíndice baja 1 nivel </sub>
<sup> superíndice sube 1 nivel </sup>
<big> aumenta el tamaño en 1 </big>
<small> disminuye el tamaño en 1 </small>
```

Los saltos de línea se indican con la etiqueta <br> y para diferenciar un párrafo de otro se usa la etiqueta <p> y </p>

```
<p>
Esto es un párrafo en una sólo línea.
</p>
<p>
Y esto es un párrafo que ocupa <br>
exactamente <br>
tres líneas.
</p>
```

Se puede alinear un texto con el atributo **align** a la izquierda , derecha o en el centro de la página.

```
<p align="left">
Este párrafo está a la izquierda
</p>
<p align="center">
Este otro está centrado
</p>
<p align="right">
Y este a la derecha
</p>
```

Para indentar un texto se puede usar <blockquote>

```
<blockquote>
Esto está indentado<br>
como bien podemos apreciar.<br>
</blockquote>
y esto no está indentado.
```

Títulos o encabezamientos (Headers).

Los encabezamientos se pueden indicar en seis tamaños distintos desde <h1> hasta <h6>.

```
<h1> Este es un titular pequeño </h1>
<h3> Este titular es mediano </h3>
<h6> Este es un titular grande</h6>
```

Mediante la etiqueta <font> y sus atributos podemos indicar distinto tipos de fuente , tamaño , colores , etc..

```
<font face="Arial" size="5" color="red">
Este texto se verá con el tipo de letra Arial ,
tamaño 5 y color rojo.
</font>
```

El color del texto se puede indicar con el valor del atributo en ingles o con el código hexadecimal RGB

(Red,Green,Blue) del color:

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

R G B  
# FF 00 00  
↑ ↑ ↑  
Indicador de número en base 16  
Cantidad de color Rojo (Red)  
Cantidad de color Verde (Green)  
Cantidad de color Azul (Blue)

```
<font color="blue">
Este texto es de color azul.
</font>
```

```
<font color="#0000FF">
Y este también es azul.
</font>
```

## Linea horizontal

Con la etiqueta <hr> se puede dibujar una línea horizontal en la pantalla. Los atributos más usuales para ella son: color (para ponerle un color) , size(para cambiar su altura) y width(donde indicaremos su anchura , por ejemplo si ponemos 300 indica una anchura de 300 pixeles ,o si ponemos 50% la línea ocuparía el 50% del ancho de la pantalla).

```
<hr color="red" size="6" width="90%" >
```



## Listas

Podemos crear tres tipos de listas: Desordenadas , ordenadas y de definición.

### Lista desordenada (unordered list) <ul>

```
<ul type="circle">
  <li>La etiqueta <ul> nos permite <br>
  presentar listas de <br>
  elementos sin orden alguno.
</li>
  <li>Cada elemento de la lista
  ir normalmente precedido
  por un círculo.
</li>
</ul>
```

- La etiqueta <ul> nos permite presentar listas de elementos sin orden alguno.
- Cada elemento de la lista irá normalmente precedido por un círculo.

Dentro de la etiqueta de apertura de la lista, <ul>, podemos utilizar el atributo **type** para cambiar el tipo de viñeta de todos los elementos de la lista. En el caso de que queramos cambiar solamente el de un elemento en especial utilizamos el atributo **type** sobre la etiqueta de elemento, es decir, sobre <li>. Los valores de **type** pueden ser:

- **circle**: la viñeta es un círculo negro.
- **disc**: la viñeta es una circunferencia.
- **square**: la viñeta es un cuadrado.

También disponemos del atributo **compact**, el cual nos permite compactar el espacio existente entre el texto y la viñeta. Este atributo no requiere valores.

### Lista ordenada (ordered list) <ol>

```
<ol type="1">
  <li>Primer elemento.</li>
  <li>Segundo elemento.</li>
  <li>Tercer elemento.</li>
</ol>
```

1. Primer elemento.
2. Segundo elemento.
3. Tercer elemento.

Utilizando el atributo **type** podemos especificar el tipo de numeración empleado eligiendo entre números (1, 2, 3...), letras (a, b, c...) y sus mayúsculas (A, B, C...) y números romanos en mayúsculas (I, II, III...) y minúsculas (i, ii, iii...). Los **valores** son:

- **1**: para ordenar por números.
- **a**: por letras.
- **A**: por letras mayúsculas.
- **i**: por números romanos en min.
- **I**: por números romanos en mayús.

Para indicar el número a partir del cual queremos empezar a contar utilizamos el atributo **start** de la etiqueta <ol>. El valor que se le ha de asignar es un número entero tanto para los números como para las letras o los números romanos

### Lista de definición (definition list) <dl>

```
<dl>
  <dt>Primer elemento
  <dd>Esta es la
  definición del primer elemento.
  <dt>Segundo elemento
  <dd>Esta es la
  definición del segundo elemento.
</dl>
```

Primer elemento  
Esta es la definición  
del primer elemento.  
Segundo elemento  
Esta es la definición  
del segundo elemento.

En este tipo de lista se muestra cada uno de los elementos seguido de su definición correspondiente. La etiqueta principal es <dl> (definition list), y las etiquetas de elemento y su definición son <dt> (definition term) y <dd> (definition definition) respectivamente.

La etiqueta <dd> desplaza la línea hacia la izquierda, por este motivo este tipo de etiqueta es utilizado para desplazar texto.

Podemos crear listas dentro de otras listas, es decir, anidar listas.

```
<dl>
  <dt>España:
  <ul>
    <li>Andalucía:
    <ol>
      <li>Granada
      <li>Sevilla
    </ol>
    <li>Extremadura
  </ul>
</dl>
```

España  
Comunidades Aut.  
\* Andalucía  
1. Granada  
2. Sevilla  
\*Extremadura

## Tablas

Una tabla es un conjunto de celdas organizadas dentro de las cuales podemos alojar distintos tipos de contenidos. Para crearlas utilizamos la etiqueta principal `<table>` junto con su cierre correspondiente `</table>`. Dentro de esta etiqueta vamos creando las distintas **filas** por medio de la etiqueta `<tr>` (table row) y `</tr>`, y en cada fila las **celdas** a través de la etiqueta `<td>` y `</td>`. Si queremos poner un título a la tabla utilizamos la etiqueta `<caption>` inmediatamente después de la etiqueta principal `<table>`.

### ATRIBUTOS DE `<table>`

- **border:** Donde se establece el valor en pixels del borde de la tabla. Si le asignamos el valor cero no habrá borde.
- **bordercolor:** color del borde.
- **background:** permite colocar un fondo para la tabla a partir de un enlace a una imagen.
- **bgcolor:** color del fondo de tabla.
- **cellspacing:** número de pixels entre celdas.
- **cellpadding:** pixels entre el borde de la celda y su contenido.
- **align:** alinea la tabla. Sus valores son: center, right o left.
- **width:** anchura de la tabla en pixels o en porcentaje (el 100% es el máximo del que dispone el navegador).
- **height:** altura de la tabla en pixels o en porcentaje.

### ATRIBUTOS DE `<tr>`

Los utilizamos para que los cambios afecten a toda las celdas de la fila.

- **align:** alineación que afecta a todas las celdas de la fila. Sus valores son: center, right o left.
- **valign:** alinea el contenido de las celdas verticalmente arriba (top), abajo (bottom) o centro (middle).
- **bgcolor:** color del interior de la línea de celdas.
- **bordercolor:** color del borde de la línea de celdas.

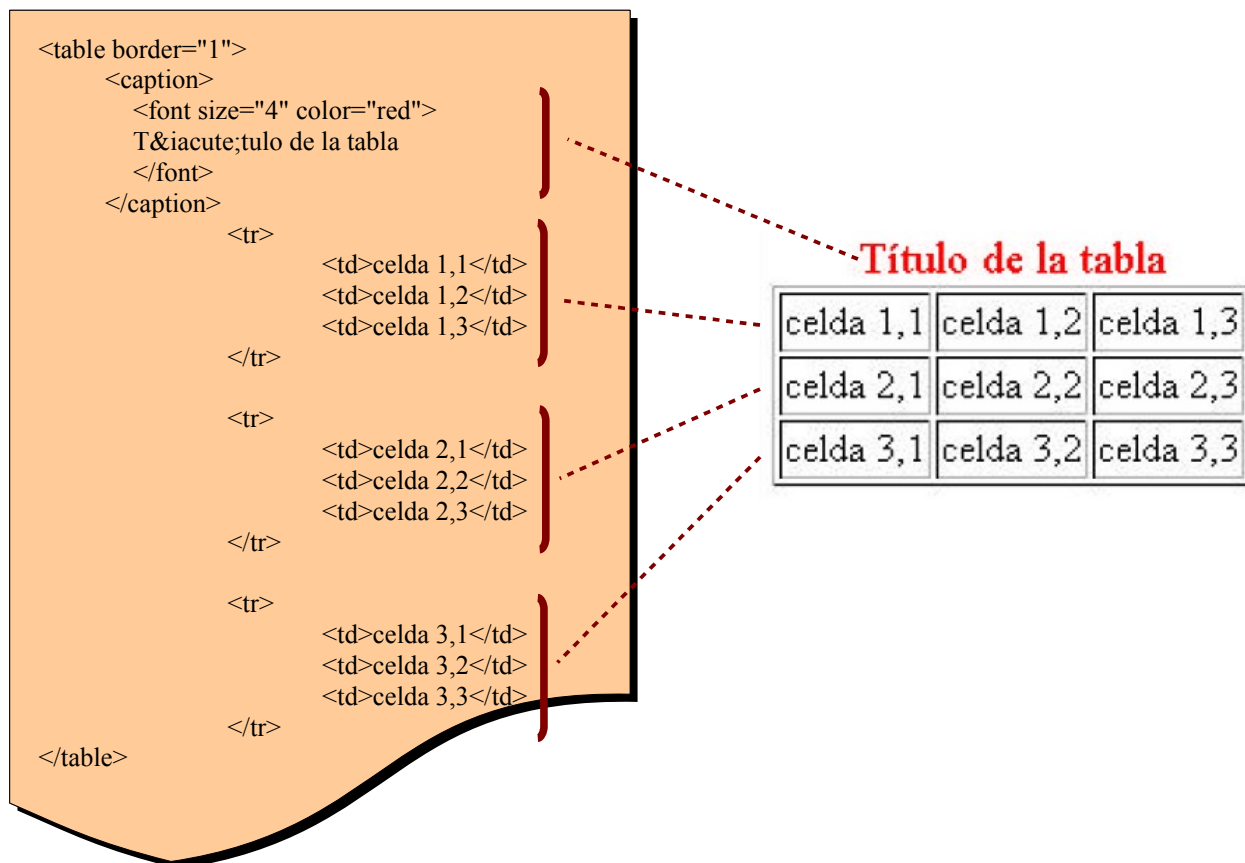
### ANIDAR TABLAS

Las tablas al igual que otros objetos de html permiten la anidación.

### ATRIBUTOS DE `<td>`

Nos permite cambiar características a una celda en concreto.

- **align:** alinea horizontalmente el contenido de la celda respecto a sus bordes. Sus valores son: center, right o left.
- **valign:** alinea el contenido de las celdas verticalmente arriba (top), abajo (bottom) o centro (middle).
- **bgcolor:** color del fondo de celda..
- **bordercolor:** color del borde.
- **background:** permite colocar un fondo para la celda a partir de un enlace a una imagen.
- **height:** altura de la celda en pixels o en porcentaje.
- **width:** anchura de la celda en pixels o en porcentaje.
- **colspan:** expande una celda horizontalmente .
- **rowspan:** expande una celda verticalmente.



## Imágenes

Para hacer que aparezcan imágenes en nuestro documento utilizamos la etiqueta `<img>` (image) y el atributo `src` (source) para indicar la ruta en la que está el archivo imagen. La etiqueta `<img>` no tiene cierre. Es decir:

```

```

### ATRIBUTOS DE `<img>`:

- **alt:** Dentro de las comillas de este atributo colocaremos una breve descripción de la imagen, de tal forma que este comentario se mostrará en lugar de la imagen en el caso de que el navegador no pueda cargar la imagen o durante el tiempo que tarde en cargarla. Una vez cargada la imagen si colocamos el puntero del ratón sobre ella también se nos mostrará la descripción escrita en este atributo.
- **height y width:** Definen la altura (height) y anchura (width) en pixels de la imagen.
- **border:** Define el tamaño en pixels del cuadro que rodea a la imagen. En caso de que no queramos que aparezca el borde, damos el valor cero ("0"). UNA TABLA CON BORDES INVISIBLES RESULTA MUY UTIL PARA DISTRIBUIR Y COLOCAR TEXTOS E IMÁGENES EN CUALQUIER PARTE DE LA PANTALLA.
- **vspace y hspace:** Sirven para indicar el espacio libre vertical (vspace) y horizontal (hspace), en pixeles, que tiene que colocarse entre la imagen y los otros elementos que la rodean, como texto, imágenes...
- **lowsrc:** Con este atributo indicamos un archivo de la imagen que está en baja resolución, de tal forma que cuando el navegador detecta este atributo primero descarga y muestra la imagen de baja resolución y luego descarga la imagen con la resolución correcta.
- **align:** Mediante este atributo alineamos la imagen con respecto al texto. Este atributo puede tener estos valores: **top,middle,bottom,left y right.**

### Ejemplo de código de una imagen insertada

```

```

Este texto está aquí arriba  
porque align="top"

Se vería así en el navegador web:



Este texto está aquí arriba  
porque align="top"

## RUTAS DE ACCESO A IMÁGENES O ARCHIVOS

### **Ruta absoluta:**

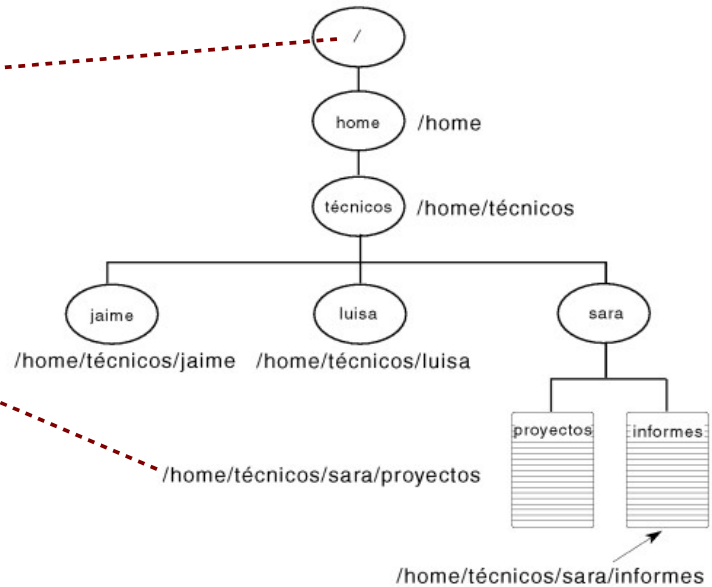
Llamamos de esta forma a la ruta que recoge la ubicación de un determinado archivo sin tener en cuenta el punto desde el que se solicita la referencia. Por ejemplo un archivo desde Internet: <http://platea.pntic.mec.es/~rluna/index.html>  
También sería una ruta absoluta una referencia a un archivo dentro de nuestro disco duro del tipo:  
`file:///F:/WEBS/HTML/imagenes/lateralespiral.gif`.

### **Ruta relativa:**

En este caso tenemos en cuenta el punto desde el que damos la referencia del tipo `imagenes/lateralespiral.gif` donde hemos eliminado la referencia inicial e indicamos que, desde el punto donde estamos, encontraremos una subcarpeta en la que estará la imagen `lateralespiral.gif`. Cuando creamos un sitio WEB es importante que todas las rutas sean relativas para que al hacer una réplica de nuestro sitio en el servidor todas las referencias sigan siendo válidas igual que en nuestro disco duro. Por cierto, supongo que ya te habrás dado cuenta que todos los archivos que formen parte del sitio WEB deben estar en la carpeta principal o en subcarpetas de la misma.

La ruta absoluta se expresa a partir del directorio raíz.

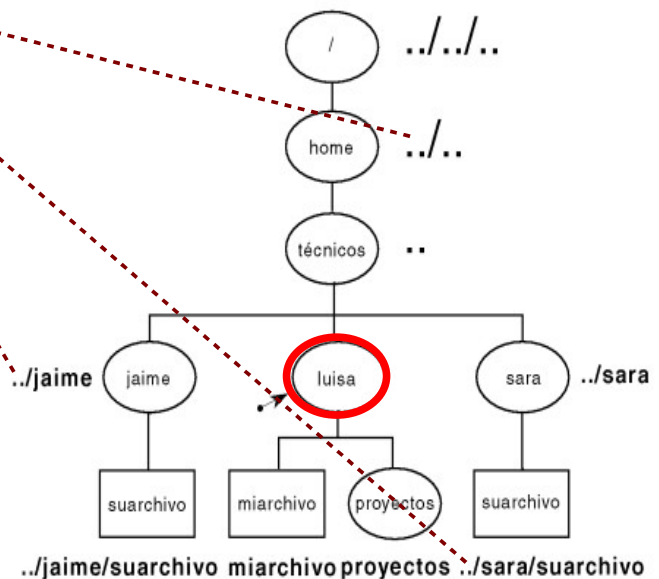
Estos son ejemplos de rutas absolutas.



Las rutas relativas se expresan a partir del directorio de trabajo actual ( en este ejemplo el actual es “**luisa**”).

Estas son rutas relativas a “**luisa**”:

Nombre de ruta relativa	Significado
.	Directorio o carpeta actual
..	Carpeta o directorio padre (el directorio que está por encima del directorio actual).
../..	Dos directorios por encima del directorio actual.
nombre de directorio	Nombre de un directorio o carpeta por debajo del directorio actual.





## ENLACES

HTML es un lenguaje para **hipertexto**, es decir, enlaces. Un **enlace** o **link** es una zona de texto o imagen que si son seleccionados nos trasladan a otro documento de hipertexto o a otra posición dentro del documento actual. Para crear un enlace hay que utilizar la etiqueta `<a>`. Todo lo que encerremos entre `<a>` y `</a>`, ya sea texto o imágenes, será considerado como enlace y sufrirá dos modificaciones:

1. Se visualizará de manera distinta en el navegador: el texto aparecerá subrayado y de un color distinto y las imágenes estarán rodeadas por un borde del mismo color que el del texto del enlace. Este color dependerá de la configuración que tenga el usuario en su navegador.
2. Al pulsar sobre el enlace, seremos enviados hacia donde apunte el enlace. El enlace en la página normalmente aparecerá subrayado y en color azul, y al deslizar el puntero del ratón sobre él cambiará su forma original transformándose por regla general en una mano con el dedo índice extendido. Para indicarle la dirección utilizamos el atributo **href** y le asignamos la dirección entre comillas dobles (“”). La dirección estará en formato **URL (Uniform Resource Locator)**. Cuando queremos acceder a un fichero situado en la misma máquina que la página web que estamos creando podemos utilizar **rutas relativas**: *ruta\_relativa/fichero*

La sintaxis general de un enlace es:

```
<a href= "dirección"> contenido </a>
```

Siendo el contenido un texto o imagen que será la parte de la página donde deberemos pulsar para acceder al enlace.

**Dirección en formato URL:** A través de una **URL** podemos indicar tanto una **dirección** de Internet como un **servicio** que tenga el servidor al que corresponde la dirección. El formato general es:

*servicio://maquina:puerto/ruta/fichero/fichero@usuario*

El **servicio** puede ser:

- **http:** para páginas **web**.
- **https:** servidores que nos ofrecen el uso de técnicas de **encriptación** para proteger los datos .
- **ftp:** para transmitir ficheros desde servidores de ftp.
- **mailto:** para poder mandar un mensaje **e-mail**.
- **news:** para acceder a **foros** de discusión. Para ello se ha de indicar el servidor y el grupo.
- **telnet:** nos permite conectarnos a otro ordenador y entrar en ellos como si nuestro ordenador fuese una terminal. La dirección de la máquina puede ser indicada también a través de su **dirección IP**.

## Ejemplos de enlaces

```
<a href="otrapagina.html"> texto enlace donde hacer clic </a>

<a href="otrapagina.html">  ( una imagen sirve
de zona activa donde hacer clic)

<a href="http://www.google.es" > enlace a otra pagina web en internet </a>

<a href="ftp://servidor ftp" > enlace a un servidor ftp </a>

<a href="mailto:correo@ono.com">abre el gestor de emails para mandar un correo</a>

<a name="nombreacla" > define un ancla (lugar concreto) en la página

<a href="#nombreacla" > salta a un ancla situada en la misma página </a>

<a href="otrapagina.html#nombreacla">enlaza con un lugar concreto de otra pagina.html.</a>
```

El atributo **name** sirve para darle nombre a un lugar concreto dentro de la página web conocido como ancla.

El carácter **#** indica que es un enlace a un ancla o lugar concreto dentro de una página.

## MAPA DE IMAGENES

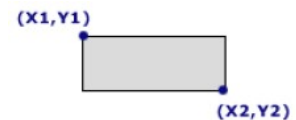
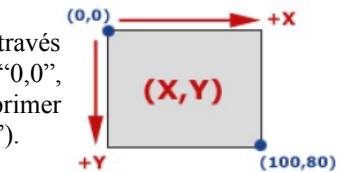
Consisten en **incluir varios enlaces dentro de una misma imagen**. Para ello definimos una serie de figuras geométricas dentro de la imagen, de forma que cada una de ellas servirá de zona activa donde haremos clic para saltar a algún documento o archivo. Para hacer un mapa de imagen tenemos primero que incluir la imagen que vamos a utilizar dentro del `<body>` de nuestro código a través de la etiqueta `<img>`. Posteriormente debemos de delimitar la imagen en figuras geométricas dentro de la etiqueta `<map>`. En la etiqueta `<map>` podemos utilizar el atributo **name** para definir el nombre de la definición del mapa. Dentro de la directiva `<map>` hemos de introducir cada una de las áreas las cuales se indican a través de la etiqueta `<area>`.

### Atributos de `<area>`:

- **alt**: para indicar el texto que se mostrará cuando situemos el ratón en el área. También se mostrará ese texto en el caso de que el navegador no pueda cargar la imagen en cuestión.
- **coords**: para definir las coordenadas del área. Cada punto de la imagen se define a través de su altura (X) y anchura (Y). Siendo la esquina superior izquierda la posición "0,0", y la esquina inferior derecha la posición "X,Y". Es decir, las coordenadas del primer pixel son "0,0" y el último pixel de una imagen, por ejemplo, de 10x10 será "9,9".
- **href**: para indicar el destino del enlace correspondiente al área.
- **nohref**: indica que no se tomará ninguna acción cuando se haga clic sobre el área definida.
- **shape**: tipo de área. Pudiendo ser:

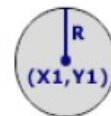
**rect**: área **rectangular**. Para definirla se utilizan las coordenadas de los puntos de la esquina superior izquierda y la esquina inferior derecha.

```
<area shape="rect" coords="x1,y1,x2,y2">
```



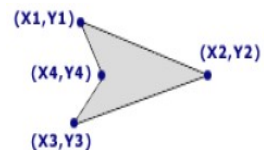
**circle**: área **circular** que se indica con la coordenada del centro del círculo y el radio.

```
<area shape="circle" coords="x1,y1,r">
```



**poly**: es un área **poligonal**. Para definirlo hay que indicar todos sus puntos en orden siguiendo el camino marcado por el perímetro del polígono.

```
<area shape="poly" coords="x1,y1,x2,y2,x3,y3,x4,y4">
```

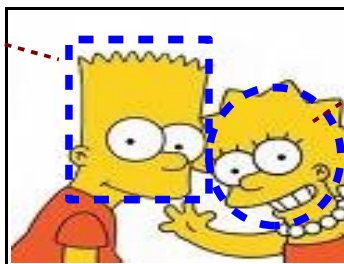


```
<html>
  <head><title>mapa</title></head>
  <body>

    <map name="mi_mapa">
      <area shape="rect" coords="20,15,60,70" href="Bart.html">
      <area shape="circle" coords="80,60,40" href="Lisa.html">
    </map>
    

  </body>
</html>
```

Al hacer clic sobre la cara de Bart se abrirá el archivo Bart.html que nos hablará sobre Bart.



Al hacer clic sobre la cara de Lisa se abrirá el archivo Lisa.html que nos hablará sobre Lisa.

## FRAMES

Los frames (o marcos) nos sirven para dividir la ventana del navegador en diferentes áreas, cada una de las cuales es un archivo html distinto e independiente. Cada marco o área tendrá sus propias barras de desplazamiento y propiedades. Para crear frames utilizamos la etiqueta `<frameset>` la cual va a sustituir al habitual `<body>`. Dentro de esta etiqueta primero definimos cada uno de los marcos poniéndoles un nombre y especificando qué fichero html le corresponde mediante la etiqueta `<frame>`, y por último indicamos lo que debe de aparecer al usuario en el caso de que su navegador no soporte frames utilizando la etiqueta `<noframes>`. Las divisiones que se pueden hacer con un `<frameset>` son en filas o en columnas. Para indicarlo utilizamos el atributo `cols` si queremos una partición en columnas o el atributo `rows` si la queremos en filas. En el atributo `cols` o `rows` colocamos entre comillas el número de particiones que deseamos realizar, indicando de paso el tamaño que va a asignarse a cada una. Los formatos de tamaño son los siguientes:

- **Porcentajes(%)**: porcentaje referido al espacio total disponible en la pantalla.
- **Absolutos**: mediante un número especificando el tamaño en pixels.
- **Sobre el espacio sobrante**: colocando un asterisco (\*) indicamos que queremos todo el espacio sobrante. Si ponemos el asterisco en varios marcos se repartirán el espacio equitativamente y si queremos que uno de los marcos sea mayor lo hacemos poniendo un número delante del asterisco, de tal forma que si ponemos `3*` ese marco será tres veces mayor que el que tenga tan sólo el asterisco. Estos tres formatos se pueden combinar.

Mediante la etiqueta `<frame>` y el atributo `src` definimos la procedencia de cada una de las filas o columnas. El valor del atributo `src` es la ruta y nombre del archivo html que va a mostrar el frame:

```
<frame src="pagina1.html">
```

En este caso estamos indicando que el frame que estamos definiendo debe mostrar la página `pagina1.html` en su interior. Habrá tantas etiquetas `<frame>` como particiones hayamos especificado en el atributo `cols` o `rows`.

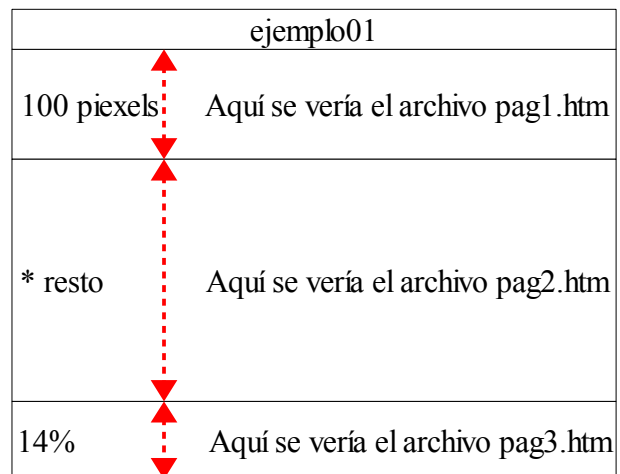
Ejemplos de frames ( observar que no hace falta `<body>`):

```
<html>
<head> <title>ejemplo01</title> </head>
<frameset rows="100,*,14%">

    <frame src="pag1.htm">
    <frame src="pag2.htm">
    <frame src="pag3.htm">

    <noframes><p>No podrás ver
esta página si el navegador
no soporta frames</p>
</noframes>

</frameset>
</html>
```

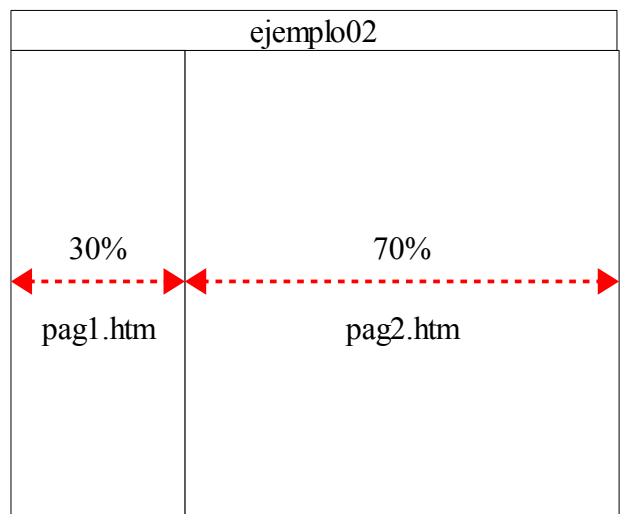


```
<html>
<head> <title>ejemplo02</title> </head>
<frameset cols="30%,70%">

    <frame src="pag1.htm">
    <frame src="pag2.htm">

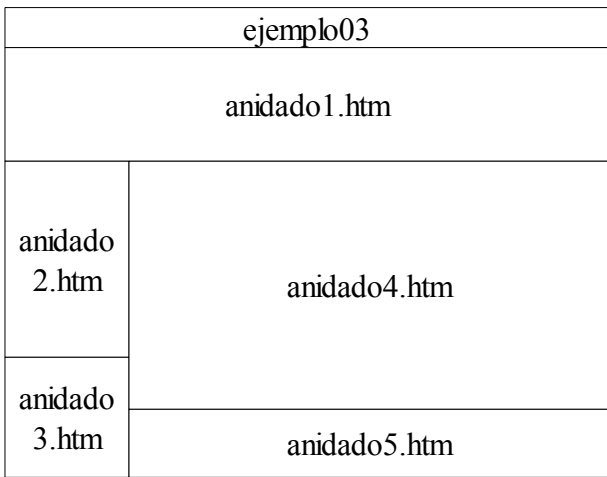
    <noframes><p>No podrás ver
esta página si el navegador
no soporta frames</p>
</noframes>

</frameset>
</html>
```



Este es un ejemplo de frame anidado para poder hacer divisiones en filas y columnas a la vez.

```
<html>
<head> <title>ejemplo03</title></head>
<frameset rows="20%,*">
  <frame src="anidado1.htm">
  <frameset cols="20%,*">
    <frameset rows="60%,*">
      <frame src="anidado2.htm">
      <frame src="anidado3.htm">
    </frameset>
    <frameset rows="80%,*">
      <frame src="anidado4.htm">
      <frame src="anidado5.htm">
    </frameset>
  </frameset>
</frameset>
</html>
```



**Los bordes de los frames:** Los navegadores cuando trabajan con marcos automáticamente dibujan un borde de separación entre los marcos.

En el caso de que no queramos que aparezca utilizaremos el atributo **frameborder="0"** en la etiqueta **<frameset>** (para que afecte a todos los frames). Aunque eliminemos el borde el navegador deja también un hueco entre los frames, para eliminarlo utilizamos los atributos **framespacing="0"** y **border="0"** en **<frameset>**.

**Atributos de <frameset>:**

- **cols:** distribución de marcos verticalmente.
- **rows:** distribución de marcos horizontalmente.
- **border:** tamaño en pixels del borde de los marcos.
- **bordercolor:** color del borde.
- **frameborder:** para indicar si se debe de mostrar el borde o no. Sus valores pueden ser:  
 yes: para que se vean los bordes.  
 no: para que no se vean.  
 0: para que no se vean.
- **framespacing:** número de pixels de anchura de la línea de separación de los frames.

**Atributos de <frame>:**

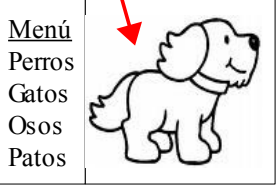
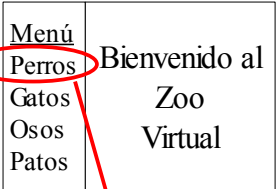
- **src:** para indicar el archivo que contiene el marco.
- **name:** para dar nombre al marco y así puedas dirigir los enlaces hacia él.
- **marginwidth:** número de pixels de separación entre los bordes verticales del marco y su contenido.
- **marginheight:** igual que el anterior pero en vertical.
- **scrolling:** para que aparezca o no una barra de desplazamiento (scrollbar) en el marco. Sus valores son:  
 yes: siempre aparecerán las barras.  
 no: no saldrán nunca.  
 auto: sólo saldrán si son necesarias.  
 noresize: atributo sin valores que sirve para indicar que el marco no se puede redimensionar.

**Uso del atributo target de los enlaces para redirigir su visualización en el frame adecuado:**

El atributo **target** puede tener los siguientes valores:

- **\_blank:** muestra la nueva página en una ventana nueva.
- **\_parent:** el enlace se actualiza sobre su padre o sobre la ventana que estamos trabajando.
- **\_top:** elimina todos los marcos existentes y carga la página a pantalla completa.
- **\_self:** se actualiza el marco donde está situado el enlace.
- **nombre de un frame previamente definido con su name:** el enlace se abrirá en el marco indicado.

**Ejemplo:** Dividimos la pantalla en dos columnas, la izquierda para colocar un menú y la derecha como ventana principal, donde se verán los contenidos de cada item del menú. Para que cuando pulsemos sobre el enlace "Perros" éste se abra en el marco de la derecha tenemos que darle un nombre al marco derecho (por ejemplo **name="derecha"**). En el enlace "Perros" hay que incluir el atributo **target="derecha"**. Tiene que haber un archivo **definición.html** que defina los frames, otro **menú.html** que se verá en la columna de la izquierda, y luego tantos .html como queramos que se vean en la columna de la derecha (como **Perros.html**, **Gatos.html**, etc.).



```
<html>
<head> <title>definicion frames</title></head>
<frameset cols="30%,*">
  <frame name="izquierda" src="menu.html">
  <frame name="derecha" src="zoo.html">
</frameset>
</html>
```

```
<html>
<head> <title>menú</title></head>
<body><u>Men&uacute;</u><br>
<a href="Perros.html" target="derecha" > Perros <a/><br>
<a href="Gatos.html" target="derecha" > Gatos <a/><br>
....
....
</body>
</html>
```

## IFRAMES

La etiqueta `<iframe>` permite insertar un marco dentro del documento. Funciona de manera similar a la etiqueta `<frame>` y `<frameset>` pero la etiqueta `<iframe>` puede ser insertada entre el contenido de cualquier documento html. En este caso `<iframe>` es insertado en medio del texto tal como una tabla, o una imagen, y es mostrado como una ventana conteniendo el documento html con el que se enlaza.

```
<html>
<head><title>IFRAME</title></head>
<body> texto texto <br>
  <center>
    <iframe scrolling="auto" src="http://www.google.es" frameborder="1"
      height="200" width="400">Si ves este mensaje, significa que tu navegador
      no soporta esta característica o está deshabilitada. Pero puedes acceder
      a esta información aquí;
    <a href="http://www.htmlquick.com/reference/tags/a.html">html</a>
    </iframe>
  </center>
  <br> texto texto <br>
</body>
</html>
```

## ejemplo iframe

texto texto



Google

text texto

## BODY

Imagen de fondo y color de fondo con la etiqueta `<body>`

**Atributo background:** Si queremos diseñar sitios más atractivos debemos utilizar imágenes o colores para el fondo de nuestras páginas. El fondo se define dentro de la etiqueta `<body>`. Para insertar una imagen de fondo debemos utilizar el atributo **background** dentro de la etiqueta `<body>`. El valor de este atributo es la URL o ruta de la imagen que deseamos colocar. Si la imagen es más pequeña que la pantalla del navegador, esta se repetirá tantas veces como sea necesario hasta cubrir toda la pantalla. **Ejemplos :**

```
<body background="paisaje.jpg">
<body background="http://www.virtualnauta.com/imagenes/punteado.gif">
```

**Atributo bgcolor :** Especifica el color de fondo de una página HTML. El valor del mismo puede expresarse con un número hexadecimal, valor RGB, o nombre de color (en inglés). Este atributo también debe ir dentro de la etiqueta `<body>`. **Ejemplos :**

```
<body bgcolor="#FF0000">
<body bgcolor="rgb(255,0,0)">
<body bgcolor="red">
```

**¡Importante!** Los atributos `background` y `bgcolor` están desaprobadados en la última versión de HTML (HTML4 y XHTML). En su lugar debemos usar [las propiedades de fondo de CSS](#) para escribir un código correcto de nuestro sitio.

## Sonido de fondo

`<bgsound>` Los formatos de sonido pueden ser : wav, mid, ram, ra, aif, aifc, aiff, au, snd, mp3, Es recomendable usar el que menos espacio utilice Para la configuración: `loop="-1"` o `loop="infinite"` Se reproducirá indefinidamente. `loop="1"` Se reproducirá una sola vez. `loop="2"` Se reproducirá dos veces. y así consecutivamente.

**Ejemplo:**

```
<html>
<head>
  <bgsound src="tu_sonido.mp3" loop="-1">
</head>
<body>
</body>
</html>
```

- **src="ruta\_fichero"** fija la ruta en la que se encuentra el fichero de audio a reproducir. La ruta puede ser relativa a nuestro sistema de carpetas local, absoluta respecto al sistema de carpetas del servidor web o una URL completa que localice el fichero en Internet.
- **loop="n"** determina el número de veces (n) que se debe ejecutar el fichero de audio.
- **balance="b"** determina el balance del sonido entre los dos altavoces del equipo, es decir, la potencia o intensidad con que se oír en cada uno de ellos (derecho e izquierdo). Sus valores pueden estar entre -10,000 y +10,000, correspondiendo el valor 0 a un balance equilibrado.
- **volume="v"** fija el volumen al que se oír el sonido, y sus valores pueden variar entre -10,000 (mínimo) y 0 (máximo). No es soportado por los equipos MAC.

Otra forma de incluir sonidos es dejando al usuario la decisión de escucharlos o no. Para hacerlo incluiremos el sonido en el parámetro **href** de un enlace, como si fuera una página HTML:

```
<a href="archivo.wav"> si pulsas aquí oirás el sonido </a>
```

**EMBED** Otra forma de incorporar sonido a una página web es utilizando la etiqueta `<embed>` .

```
<embed src="../../sonidos/canyon.mid" width=100 height=100 hidden="true" autostart="true" loop="true">
```

#### Atributos de `<embed>`:

- **src**: ruta y nombre del archivo a reproducir.
- **width y height**: para indicar el tamaño del reproductor. Son anchura y altura respectivamente.
- **autostart**: para indicar si se arranca automáticamente la reproducción. Los valores son "true" (para que arranque automáticamente) o "false" (para que no).
- **loop**: para que reproduzca ininterrumpidamente el archivo de sonido (valor "true"), o no (valor "false").
- **hidden**: para ocultar (valor "true"), o no (valor "false") el reproductor.

Además de sonido ,en el navegador web se pueden reproducir otros tipos de ficheros . Los **plugins** son programas auxiliares capaces de manejar formatos especiales de ficheros, que se incorporan al navegador y se ejecutan dentro de la ventana de visualización. Se incluyen en la página a través de la directiva `<embed>`. **Más atributos de `<embed>`:**

- **name**: nombre del objeto contenido en el plugin.
- **src**: localización del fichero a reproducir.
- **pluginspage**: URL donde el usuario encontrará instrucciones sobre cómo instalar el plugin.
- **units**: unidad de medida para los atributos **height** y **width**. Pueden ser **pixels** o en **%**.

**APPLET** Permite introducir programas Java (concretamente applets) dentro de una página web. **Un applet es una aplicación escrita en Java y desarrollada para ser ejecutada desde un navegador.** Las utilidades típicas en las que son empleados los applets son efectos gráficos en páginas Web, interfaces de usuario avanzadas, funcionalidad cliente en aplicaciones cliente-servidor y juegos. La sintaxis es la siguiente:

Los navegadores que no admitan applets, ignoran todo lo que aparece entre la directiva `<applet>` y su cierre, a excepción de los parámetros `<param>` que permiten el paso de parámetros. Java es compatible con todos los sistemas porque basa su funcionamiento en los **Byte Codes**, que no es más que una precompilación del código fuente de Java. Estos Byte Codes no son el programa en Java propiamente dicho, sino un archivo que contiene un código intermedio que puede manejar la **Máquina Virtual de Java**. Cada sistema operativo dispone de una Máquina Virtual de Java que puede interpretar los Byte Codes y transformarlos a sentencias ejecutables en el sistema en cuestión. Los applets tienen la extensión **".class"**.

```
<applet atributo1=valor1 atributo2=valor2 ...atributoN=valorN>
  <param ...>
  <param ...>
  <!-- en este espacio escribimos el código para
        navegadores que no reconozcan applets -->
  .....
  .....
</applet>
```

#### Atributos de `<applet>`:

- **codebase**: especifica la URL donde está localizado tanto el applet como todos los recursos (imágenes, clases Java...).
- **code**: especifica el nombre de la clase Java que contiene el applet.
- **name**: indica un nombre para la instancia concreta del applet.
- **archive**: lista de archivos separados por coma, conteniendo clases Java y recursos necesarios para la ejecución del applet, que son precargados por el navegador. Al precargar se acelera el proceso de ejecución de los applets. También tenemos atributos que ya hemos visto en otras ocasiones como: **width**, **height**, **align**, **hspace** y **vspace**.

**PARAM** Es una directiva para definir parámetros y es utilizada por applets Y por objetos. Sus atributos son:

- **name**: nombre con el cual será identificado el parámetro dentro del applet.
- **value**: valor concreto del parámetro.
- **valuetype**: tipo de atributo **value**. Su uso es optativo. Valores:
  - ? **data**: hace que value se mande al applet u objeto como un string. Es el valor por defecto.
  - ? **ref**: value contiene la URL donde se pueden localizar los valores en tiempo de ejecución.
  - ? **object**: indica que value contiene un identificador correspondiente a una declaración de `<object>` en el mismo documento.
- **type**: indica el tipo de contenido del recurso apuntado por **value** en el caso de que **valuetype** sea **ref**. Los únicos atributos soportados por todos los navegadores son los dos primeros.

### Ejemplo de applet:

```
<html>
<head><title>ejemplo de applet</title></head>
<body>
  <p align="center">
    <applet codebase="../applets" code="sinetext.class" width=470 height=75>
      <param name=text value="curso de html y css">
      <param name=rate value="4">
      <param name=font value="verdana">
      <param name=background value="#000000">
      <param name=foreground value="#ffffff">
      lo siento, pero no puedes ver este applet funcionando.
    </applet>
  </p>
</body>
</html>
```

### OBJECT

La etiqueta **<object>** sirve de solución genérica para la incluir "objetos" en los documentos. Estos "objetos" pueden ser: imágenes, applets, videos , sonido , otros documentos HTML,etc...

La estructura es muy similar a la de la etiqueta **<applet>**, con la inclusión entre **<object>** y su cierre de directivas **<param>** y contenido alternativo que se mostrará en navegadores que no soporten el objeto.

#### Atributos de **<object>**:

- **classid**: Sirve para especificar la localización de los datos del objeto. Este atributo puede utilizarse conjuntamente con el atributo **data** o como alternativa a éste. Dependerá del tipo de objeto. Su valor es una dirección URL.
  - **codebase**: Especifica el "camino" a utilizar para resolver los URL relativos especificados en otros atributos. Si no se especifica, el valor por defecto es la base del URL del documento activo. Su Valor es un URL.
  - **codetype**: Sirve para especificar el tipo de medio o datos esperados, cuando se descarga el objeto especificado por el atributo **classid**, por ejemplo: **"text/html"**, **"text/css"**, **"image/gif"**, etc. Si no se incluye este atributo, se toma por defecto el valor del atributo **type**.
  - **data**: Puede utilizarse para especificar la localización de los datos del objeto. Si se utiliza como dirección relativa, se interpretará en base al atributo **codebase**.
  - **type**: Especifica el tipo de contenido especificado por el atributo **data**. Es opcional, pero se recomienda su uso si se va a utilizar el atributo **data**, ya que evita la carga de tipos de contenido que no sean soportados por el navegador.
  - **archive**: Puede utilizarse para relacionar una serie de URLs de archivos que contengan recursos relativos al objeto en cuestión. Puede incluir los recursos especificados en los atributos **classid** y **data**. La lista debe ir separada por espacios en blanco. Si se utilizan direcciones relativas, se resolverán en base al atributo **codebase**.
  - **declare**: Cuando se utiliza este atributo, convierte la definición del objeto en una simple "declaración". El objeto queda incrustado, pero inactivo. Para activar el objeto, deberá utilizarse, con posterioridad, otra definición de objeto que haga referencia a esta declaración.
  - **stanby**: Este atributo especifica un mensaje que será mostrado mientras se cargan los datos del objeto.
  - **usemap**: Asocia un mapa de imagen con un elemento. La gran ventaja con respecto a **<img>** es que al colocar la definición de **<map>** (la lista de enlaces) entre **<object>** y su cierre, cabe la posibilidad de que un navegador sin imágenes vea, sin embargo, los enlaces del mapa.
- Y atributos que ya conocemos como **height**, **width**, **border**, **hspace**, **vspace** y **align**.

```
<html>
<head><title>ejemplo con object</title></head>
<body>
  <object data="musica.mov" width="200" height="150" border="2">
  </object>
  <object data="../imagenes/478-alienp3.gif" width="200" height="150" border="2">
  <!-- Si el objeto anterior no funciona se verá el siguiente texto -->
  Lo siento pero no se puede cargar ni el vídeo, ni la imagen.
  </object>
</body>
</html>
```

## DIV

La etiqueta `<div>` (división) nos permite dividir la página web en secciones, con lo que conseguiremos una cierta estructuración dentro de la página. Dicho de otra forma, `<div>` es un contenedor, una caja, en la que se pueden introducir otras cajas. De esta forma se les puede dar formato más tarde con CSS (hojas de estilo en cascada).

Para usar `<div>` es de vital importancia el atributo "id". Con `id` asignaremos a cada sección `<div>` un nombre único para poder distinguir cada una de ellas en el código fuente. También se puede asignar un "id" a `<body>`.

Ejemplo de división de una página en secciones con `<div>`:

```
<html>
  <head><title> secciones </title></head>
  <body id="Inicio">
    <div id="Sobrecubierta">
      <div id="Logotipo">
        Este es el sitio para nuestro logo.
      </div> <!-- Fin de Logotipo-->
      <div id="Navegación">
        Aquí pondremos enlaces para navegar.
      </div> <!-- Fin de Navegación-->
      <div id="Zona texto">
        Contenidos de la página.
      </div> <!-- Fin de Zona texto-->
      <div id="Pié de página">
        Esto es el pié de página
      </div> <!-- Fin de Pié de página-->
    </div> <!-- Fin de Sobrecubierta-->
  </body>
</html>
```

```
<body id="Inicio">
  <div id="Sobrecubierta">
    <div id="Logotipo">
      Este es el sitio para nuestro logo.
    </div>
    <div id="Navegación">
      Aquí pondremos enlaces para navegar.
    </div>
    <div id="Zona texto">
      Contenidos de la página.
    </div>
    <div id="Pié de página">
      Esto es el pié de página
    </div>
  </div>
</body>
```

### Representación esquemática de las secciones.

**Skip-Link:** Un skip-link es un salto o hipervínculo dentro de la misma página. Estos saltos son fáciles de realizar cuando se utilizan etiquetas `<div>`. Si estamos navegando por el principio de la página (suponiendo una página muy larga) y queremos saltar a una sección del final lo haríamos así:

```
<a href="#Pié de página"> Ir al pié </a>
```

(Observar que se antepone # al nombre definido en "id")

Si estamos en el final y queremos saltar el inicio:

```
<a href="#Logotipo"> Ir al Logo </a>
```



## HOJAS DE ESTILO EN CASCADA “CSS” (CASCADING STYLE SHEETS)

- CSS es un lenguaje creado especialmente para dar formato a los elementos html como fuentes , colores ,hipervínculos , secciones ,...
- Las posibilidades de dar formato de CSS se dividen en tres áreas:
  - 1) Fuentes y colores.
  - 2) Distancias y bordes.
  - 3) Layout ( posición de secciones).
- Se puede incorporar el lenguaje CSS en una página web html de tres formas distintas:
  - 1) Insertando código CSS en la cabecera mediante la etiqueta <style>:

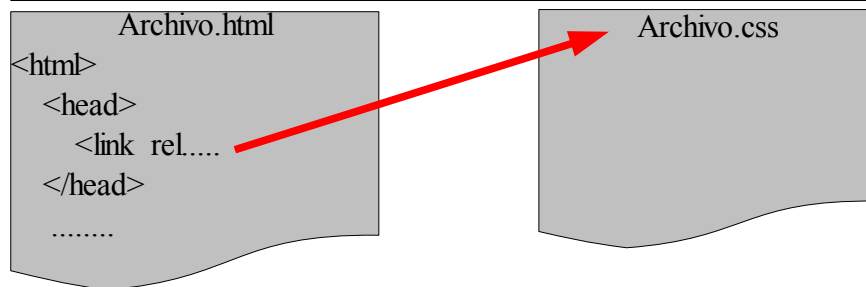
```
<head>
  <style type="text/css">
    <!--
      p {text-align:right}
    -->
  </style>
</head>
```

- 2) Como atributo **style** en cualquier elemento:

```
<p style="text-align: right">
  Párrafo con estilo propio
</p>
```

- 3) Creando un archivo CSS externo y enlazándolo con el archivo html. Los archivos CSS tienen la extensión .css . Dentro del archivo .html tenemos que poner el enlace a el archivo .css mediante la etiqueta <link>.

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



- Los comentarios en CSS se hacen así: `/* esto es un comentario */`
- La sintaxis básica de CSS está formada por un SELECTOR seguido de una DECLARACIÓN encerrada entre llaves { } la cual se compone de PROPIEDADES Y VALORES.

```
selector { propiedad: valor ; propiedad: valor ; propiedad: valor }
```

Los selectores pueden ser elementos usuales de html como `body` , `p` , `h1` , `h2` , .....o selectores usando los “id” asignados a `<div>` u otros elementos.En este último caso se antepone la almohadilla # al nombre del selector.

- **Ejemplo de posible fichero CSS:**

```
#Sobrecubierta {
    color: black;
    background-color: white;
    width: 720px;
    margin-top: 10px;
    margin-right: auto;
    margin-bottom: 10px;
    margin-left: auto;
}

#Logotipo {
    background-color: #f3c600;
    background-image: url( fondo.jpg );
    background-attachment: fixed;
}

h1 { font-size: 150%; }
h2 { font-size: 120%; }
h1 , h2 { text-align: center;
          letter-spacing: 4px;
}

p {
    font-size : 12pt;
    font-family : arial, helvetica;
    font-weight : normal;
}
```

### **Formación de grupos: class**

Para poder dar formato a varios párrafos de la página, un selector sencillo como “p” es demasiado indeterminado y una “id” no es recomendable porque sólo se puede utilizar una vez para un solo párrafo. Si lo que queremos es tener grupos de párrafos cada uno con un estilo lo que debemos usar son las clases de CSS. Por ejemplo si queremos que todos los párrafos informativos destaquen con un fondo gris y un borde sólido haríamos lo siguiente:

- En el fichero html pondríamos el atributo “class” a los párrafos deseados:  
`<p class="cuadro_info"> Esto es un párrafo informativo </p>`
- Y en el fichero CSS introducimos la clase con el nombre del atributo class asignado en el html precedido por un punto:

```
.cuadro_info {
    color: black;
    background-color: #8c8c8c;
    border: 3px solid #ecf7dd;
}
```

**Pseudoclasses:** Una pseudoclase acompaña al selector separado por dos puntos:” (las pseudoclasses usuales son link, visited, hover, focus, active, lang, first-line, first-letter, before y after ). Se utiliza frecuentemente en el selector correspondiente a la etiqueta <a> para los enlaces. Ejemplo:

- a: link {color: blue} ( El color de los enlaces normales será azul).
- a: visited {color: purple} (Color de los enlaces ya visitados)
- a: hover {color: green} ( Color del enlace cuando el ratón está o pasa por encima del enlace)
- a: focus {color: green} (Color del enlace cuando se selecciona con el teclado, tabulando)
- a: active {color: black} (Color del enlace cuando en el momento de hacer click con el ratón)

### **Los sectores se pueden agrupar separándolos por comas , Ejemplo:**

```
p , h1 , h2 { font-color: blue; }
```

### **Pequeñas partes de la página**

Para definir estilos en secciones reducidas de una página se utiliza la etiqueta <span>. Con su atributo **style** indicamos en sintaxis CSS las características de estilos. Lo vemos con un ejemplo, pondremos un párrafo en el que determinadas palabras las vamos a visualizar en color verde.

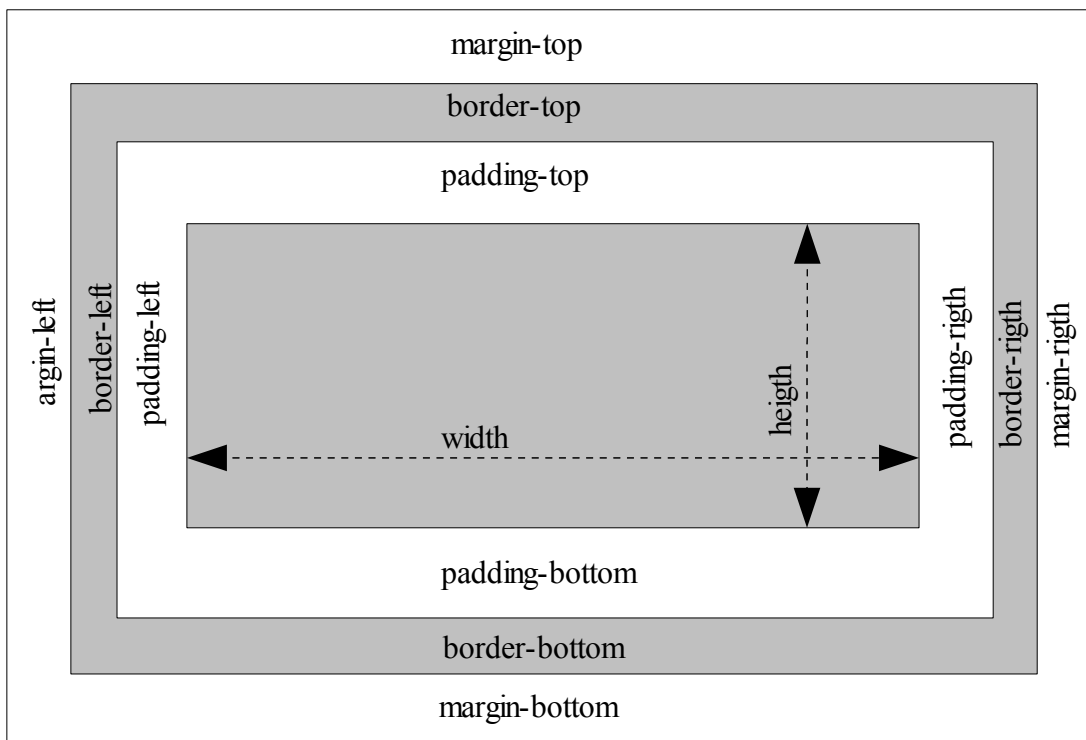
```
<p> esto es un párrafo en varias palabras
  <span style="color:green">en color verde</span>
  resulta muy fácil. </p>
```

## Unidades de medida usadas en CSS

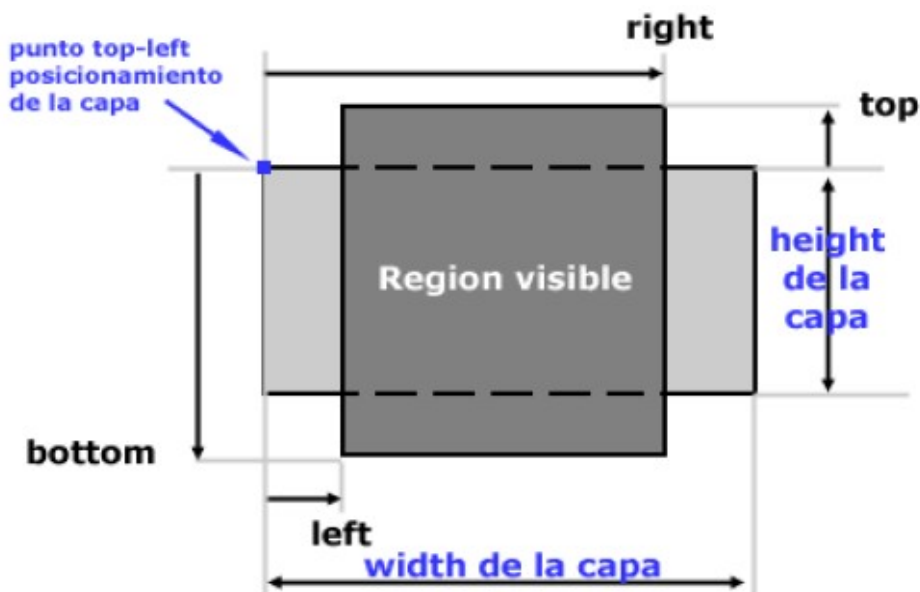
px: pixels  
in : inches (pulgadas)  
mm: milímetros  
cm: centímetros  
pc: picas  
pt: puntos  
ex: x-heights (tipo medio, la altura de una letra minúscula que no tiene trazos ascendentes ni descendentes)  
em: m-lengths (tipo máximo, la altura de una letra mayúscula que no tiene trazos ascendentes ni descendentes)

ejemplo: `<div style="margin: 10px;">texto cualquiera</div>`

## El modelo de cajas de CSS



## Posicionamiento de capas y clipping



## GUÍA DE REFERENCIA CSS (Propiedades, valores, selectores)

### MODELO DE CAJAS

#### MÁRGENES

Propiedad	Descripción	Valores
<code>margin-top</code> <code>margin-right</code> <code>margin-bottom</code> <code>margin-left</code>	Tamaño del margen superior, derecho, inferior e izquierdo	[ <longitud>   <porcentaje>   auto ]
<code>margin</code>	Ancho para varios márgenes individuales	[ <longitud>   <porcentaje>   auto ] {1,4}

#### RELLENO

Propiedad	Descripción	Valores
<code>padding-top</code> <code>padding-right</code> <code>padding-bottom</code> <code>padding-left</code>	Ancho del relleno superior, derecho, inferior e izquierdo	[ <longitud>   <porcentaje> ]
<code>padding</code>	Tamaños para varios rellenos individuales	[ <longitud>   <porcentaje> ] {1,4}

#### BORDES

Propiedad	Descripción	Valores
<code>border-top-width</code> <code>border-right-width</code> <code>border-bottom-width</code> <code>border-left-width</code>	Anchura del borde superior, derecho, inferior o izquierdo	[ thin   medium   thick   <longitud> ]
<code>border-width</code>	Anchos de varios bordes individuales	[ thin   medium   thick   <longitud> ] {1,4}
<code>border-top-color</code> <code>border-right-color</code> <code>border-bottom-color</code> <code>border-left-color</code>	Color del borde superior, derecho, inferior o izquierdo	[ <color>   transparent ]
<code>border-color</code>	Colores de varios bordes individuales	[ <color>   transparent ] {1,4}
<code>border-top-style</code> <code>border-right-style</code> <code>border-bottom-style</code> <code>border-left-style</code>	Estilo del borde superior, derecho, inferior o izquierdo	[ none   hidden   dotted   dashed   solid   double   groove   ridge   inset   outset ]
<code>border-style</code>	Estilos de varios bordes individuales	[ none   hidden   dotted   dashed   solid   double   groove   ridge   inset   outset ] {1,4}
<code>border-top</code> <code>border-right</code> <code>border-bottom</code> <code>border-left</code>	Ancho, estilo y el color para el borde superior, derecho, inferior o izquierdo	[ <border-top-width>    <border-top-style>    <border-top-color> ]
<code>border</code>	Ancho, el estilo y el color para los 4 bordes	[ <border-top-width>    <border-top-style>    <border-top-color> ]

### MODELO DE FORMATO VISUAL

Propiedad	Descripción	Valores
<code>display</code>	Comportamiento del contenedor	[ inline   block   list-item   run-in   inline-block   table   inline-table   table-row-group   table-header-group   table-footer-group   table-row   table-column-group   table-column   table-cell   table-caption   none ]
<code>position</code>	Esquema de posicionamiento	[ static   relative   absolute   fixed ]
<code>top</code> <code>right</code> <code>bottom</code> <code>left</code>	Desplazamiento de la caja (respecto al límite superior, derecho, inferior o izquierdo del contenedor)	[ <longitud>   <porcentaje>   auto ]
<code>float</code>	Posicionamiento flotante	[ left   right   none ]
<code>clear</code>	Control de cajas adyacentes a los float	[ none   left   right   both ]
<code>z-index</code>	Solapamiento de niveles de capas	[ auto   <entero_con_signo> ]
<code>direction</code>	Sentido direccional de la escritura	[ ltr   rtl ]
<code>unicode-bidi</code>	Sentido direccional de la escritura	[ normal   embed   bidi-override ]

### DETALLES DEL MODELO DE FORMATO VISUAL

Propiedad	Descripción	Valores
<code>width</code>	Ancho	[ <longitud>   <porcentaje>   auto ]
<code>min-width</code>	Ancho mínimo	[ <longitud>   <porcentaje> ]
<code>max-width</code>	Ancho máximo	[ <longitud>   <porcentaje>   none ]
<code>height</code>	Alto	[ <longitud>   <porcentaje>   auto ]
<code>min-height</code>	Alto mínimo	[ <longitud>   <porcentaje> ]
<code>max-height</code>	Alto máximo	[ <longitud>   <porcentaje>   none ]
<code>line-height</code>	Altura entre las bases del texto	[ normal   <número>   <longitud>   <porcentaje> ]
<code>vertical-align</code>	Alineación vertical del texto	[ baseline   sub   super   top   text-top   middle   bottom   text-bottom   <porcentaje>   <longitud> ]

## EFECTOS VISUALES

Propiedad	Descripción	Valores
overflow	Comportamiento del contenido si se desborda en la caja	[ visible   hidden   scroll   auto ]
clip	Especifica la región visible del elemento	[ rect (<longitud>, <longitud>, <longitud>, <longitud>)   auto ]
visibility	Visibilidad de las cajas	[ visible   hidden   collapse ]

## CONTENIDO GENERADO, NUMERACIÓN AUTOMÁTICA Y LISTAS

Propiedad	Descripción	Valores
content	Agregador de contenido para <a href="#">:after</a> y <a href="#">:before</a>	[ normal   none   [ <texto>   <uri>   <contador>   attr(X)   open-quote   close-quote   no-open-quote   no-close-quote]+ ]
quotes	Especifica las marcas para indicar las citas	[ [ <texto> <texto> ]+   none ]
counter-reset	Inicializa un contador	[ [ <identificador> <entero?> ]+   none ]
counter-increment	Incrementa un contador	[ [ <identificador> <entero?> ]+   none ]
list-style-type	Estilo aplicable a los marcadores visuales de las listas	[ disc   circle   square   decimal   decimal-leading-zero   lower-roman   upper-roman   lower-greek   lower-latin   upper-latin   armenian   georgian   lower-alpha   upper-alpha   none ]
list-style-image	Imagen aplicable a los elementos de las listas	[ url("http://...")   none ]
list-style-position	Posición dentro de la lista de los elementos marcadores de las listas	[ inside   outside ]
list-style	Permite establecer el estilo de la lista, la imagen y/o la posición	[ <list-style-type>    <list-style-position>    <list-style-image> ]

## COLORES Y FONDO

Propiedad	Descripción	Valores
color	Color del primer plano	<color>
background-color	Color de fondo	[ <color>   transparent ]
background-image	Imagen de fondo	[ url(...)   none ]
background-repeat	Repetición de la imagen de fondo	[ repeat   repeat-x   repeat-y   no-repeat ]
background-attachment	Desplazamiento de la imagen de fondo	[ scroll   fixed ]
background-position	Posición de la imagen de fondo	[ [ <porcentaje>   <longitud>   left   center   right ] [ <porcentaje>   <longitud>   top   center   bottom]? ]   [ [ left   center   right ]    [ top   center   bottom ] ]
background	Propiedades individuales relacionadas con el fondo	[ <background-color>    <background-image>    <background-repeat>    <background-attachment>    <background-position> ]

## MEDIOS PAGINADOS

Propiedad	Descripción	Valores
page-break-before	Gestionar saltos de página antes del elemento	[ auto   always   avoid   left   right ]
page-break-after	Gestionar saltos de página posterior al elemento	[ auto   always   avoid   left   right ]
page-break-inside	Evita los saltos de línea en el interior del elemento	[ avoid   auto ]
orphans	Mínimo número de líneas de un párrafo que deben ser dejadas como mínimo al final de una página	<entero>
widows	Mínimo número de líneas de un párrafo que deben ser dejadas como mínimo al principio de una página	<entero>

## FUENTES

Propiedad	Descripción	Valores
font-family	Familias de fuentes	[ [ <nombre-familia>   <familia-genérica> ] [, <nombre-familia>   <familia-genérica> ]* ]
font-style	Estilo de la fuente	[ normal   italic   oblique ]
font-variant	Convierte las minúsculas a mayúsculas pero mantienen un tamaño inferior a las mayúsculas	[ normal   small-caps ]
font-weight	Intensidad de la fuente	[ normal   bold   bolder   lighter   100   200   300   400   500   600   700   800   900 ]
font-size	Tamaño de la fuente	[ [ xx-small   x-small   small   medium   large   x-large   xx-large ]   [larger   smaller]   <longitud>   <porcentaje> ]
font	Atajo para establecer el resto de propiedades sobre las fuentes a la vez	[ [ [ <font-style>    <font-variant>    <font-weight> ]? <font-size> [ / <line-height> ]? <font-family>   caption   icon   menu   message-box   small-caption   status-bar ]

## TEXTO

Propiedad	Descripción	Valores
text-indent	Desplazamiento de la primera línea del texto	[ <longitud>   <porcentaje> ]
text-align	Alineamiento del texto	[ left   right   center   justify ]
text-decoration	Efectos de subrayado, tachado, parpadeo	[ none   [ underline    overline    line-through    blink ] ]
letter-spacing	Espacio entre caracteres	[ normal   <longitud> ]
word-spacing	Espacio entre palabras	[ normal   <longitud> ]
text-transform	Transformaciones del texto a mayúsculas/minúsculas	[ capitalize   uppercase   lowercase   none ]
white-space	Comportamiento de los espacios dentro de los elementos	[ normal   pre   nowrap   pre-wrap   pre-line ]

## TABLAS

Propiedad	Descripción	Valores
caption-side	Posición del título de respecto la tabla	[ top   bottom ]
table-layout	Control del algoritmo usado para el formato de las celdas, filas y columnas	[ auto   fixed ]
border-collapse	Selección del modelo de los bordes	[ collapse   separate ]
border-spacing	Espaciado entre los bordes de celdas adyacentes	<longitud> <longitud>?
empty-cells	Visibilidad de los bordes de celdas sin contenido	[ show   hide ]

## INTERFAZ DE USUARIO

Propiedad	Descripción	Valores
cursor	Especifica el cursor	[ [http://.../cursor.ico,]* [ auto   crosshair   default   pointer   move   e-resize   ne-resize   nw-resize   n-resize   se-resize   sw-resize   s-resize   w-resize   text   wait   help   progress ] ]
outline-width	Ancho de la línea exterior	<border-width>
outline-style	Estilo de la línea exterior	<border-style>
outline-color	Color de la línea exterior	[ <color>   invert ]
outline	Propiedades individuales de la línea exterior.	[ <outline-color>    <outline-style>    <outline-width> ]
speak-header	(Aural) Indica si las cabeceras de la tabla se leen antes de cada celda	[ once   always ]

## SELECTORES

Patrón	Aplicado a:	Ejemplo
*	Cualquier elemento	N/D
e	Elemento <e>	<e>
e f	Cualquier <f> que esté incluido en <e>	<e>...<d>...<f>...</d>...</e>
e > f	Cualquier <f> que es hijo de <e>	<e>...<f>...</f>...</e>
e:first-child	<e> cuando es el primer hijo de su elemento padre	<f><e></e>...</f>
e + f	<f> si está precedido inmediatamente por un <e>	...</e> <f>...<f>
e[miAtr]	<e> si posee el atributo "miAtr"	<e miAtr="xxx".../>
e[miAtr="v"]	<e> si posee el atributo "miAtr" y este tiene el valor "v"	<e miAtr="v".../>
e[miAtr~="v"]	<e> si su atributo "miAtr" es una lista de valores separados por espacios y uno es "v"	<e miAtr="x y z v".../>
e[lang="es"]	<e> si su atributo "lang" es una lista de valores separados por guiones y comienza con "es"	<e lang="es-ES" .../>
e.miClase	<e> cuya clase es "miClase"	<e class="miClase".../>
e#miId	<e> cuyo identificador es miId	<e id="miId".../>
a:link	Enlaces no visitados	N/D
a:visited	Enlaces visitados	N/D
e:active	<e> cuando es activado (tiempo entre que se pulsa un botón sobre él y se suelta)	N/D
e:hover	<e> cuando se posiciona el cursor sobre él pero no se activa	N/D
e:focus	<e> cuando tiene el foco posicionado en él	N/D
e:lang(c)	<e> si está marcado con el idioma c	N/D
e:first-line	Primera línea de <e>	N/D
e:first-letter	Primera letra de <e>	N/D
e:before	Aplica contenido antes de <e>	N/D
e:after	Aplica contenido después del elemento <e>	N/D
e , f	Aplica el mismo contenido a los elementos <e> y <f>	N/D

## JAVASCRIPT

*JavaScript* es un lenguaje interpretado que permite incluir macros o pequeños programas en páginas Web. Estas macros se ejecutan en el ordenador del visitante de nuestras páginas, y no en el servidor (algo muy interesante, los servidores Web suelen estar sobrecargados, mientras que los PC's de los usuarios no suelen estarlo).

*JavaScript* proporciona los medios para:

- Controlar las ventanas del navegador y el contenido que muestran.
- Programar páginas dinámicas simples sin tener que matar moscas a cañonazos de Java.
- Evitar depender del servidor Web para cálculos sencillos.
- Capturar los eventos generados por el usuario y responder a ellos sin salir a Internet.
- Simular el comportamiento de las macros *CGI* cuando no es posible usarlas.
- Comprobar los datos que el usuario introduce en un formulario antes de enviarlos.
- Comunicarse con el usuario mediante diversos métodos.

El lenguaje JavaScript se inserta en documentos **HTML** (*HyperText Markup Languaje*), de forma que su código queda reflejado en la propia página y no es llamado o cargado de ninguna fuente externa (por ejemplo un archivo). Se trata de un lenguaje interpretado puro (ni compilación, ni generación de intermedios codificados de ningún tipo).

Todo el código de JavaScript debe explicitarse en la página HTML según el formato:

```
<script
Language="JavaScript">
<!--
// Aquí irá su código
// -->
</script>
```

Observe que el *tag* (etiqueta) utilizado es **<script language=...>.....</script>**. En “**language**” se indica qué lenguaje *script* se utilizará, aquí indicamos JavaScript, ya que es lo que nos ocupa en estos momentos. Note también las acotaciones de comentario **<!--** al comienzo y **//-->** al final del área de escritura de código. Esto es para ocultarlo a los navegadores más antiguos, que no dan soporte a lenguajes script.

La etiqueta **<script>** se inserta en el encabezado entre **<head>** y **</head>** para definir funciones que serán utilizadas posteriormente en el cuerpo entre **<body>** y **</body>**.

En el cuerpo **<body>** también puede aparecer la etiqueta **<script>** para llamar a las funciones definidas en el encabezado. Las funciones definidas en **<head>** normalmente son ejecutadas por los llamados Manejadores de Eventos. Los eventos permiten al usuario interactuar con la página web. Los eventos más comunes son:

- onAbort** Se origina cuando se cancela la carga de una imagen.
- onBlur** Se produce al deseleccionar un campo de edición de un formulario.
- onClick** Se produce cuando se hace clic en un elemento del formulario o un enlace hipertexto.
- onChange** Se origina al cambiar en contenido de un campo o selección.
- onError** Se produce cuando se obtiene un error al cargar el documento HTML.
- onFocus** Se produce cuando se Activa un campo de edición.
- onLoad** Se origina cuando se carga el documento HTML.
- onMouseOut** Se produce cuando alejamos el mouse de un enlace de hipertexto.
- onMouseOver** Se produce cuando desplaza el mouse sobre los enlaces de hipertexto.
- onReset** Se produce cuando se resetea el formulario.
- onSelect** Se produce al seleccionar un campo de edición.
- onSubmit** Se origina al enviar el formulario.
- onUnload** Se origina al abandonar el documento HTML.
- onDbClick** Se produce cuando se hace doble clic en un formulario o enlace hipertexto.
- onDragDrop** Se produce cuando se vuelca (drop) un objeto sobre la ventana del navegador.
- onKeyDown** Se produce cuando se pulsa una tecla.
- onKeyPress** Se produce cuando se tiene una tecla pulsada.
- onMouseDown** Se origina al pulsar algún botón del mouse.
- OnMouseMove** Se origina al mover el cursor.
- onMove** Se produce cuando se mueve una ventana o frame.
- onResize** Se produce al cambiar el tamaño de una ventana o frame.

En Internet puedes encontrar páginas que suministran ejemplos de códigos en Javascript que puedes copiar y pegar en tu página web, como por ejemplo:

[www.elcodigo.net](http://www.elcodigo.net)

[www.lawebdelprogramador.com](http://www.lawebdelprogramador.com)

[www.mundojavascript.com](http://www.mundojavascript.com)

[www.losrecursosgratis.com](http://www.losrecursosgratis.com)

[www.javascript.com.mx/](http://www.javascript.com.mx/)

[www.codigogratias.com.ar](http://www.codigogratias.com.ar)

