

Diseño Web

Prof. Netz Romero

Los enlaces

Probablemente la característica que más ha influido, junto con las imágenes, en el espectacular desarrollo de la Web hayan sido los enlaces (links). Un enlace aparece generalmente como un texto azul subrayado y cuando situamos el cursor sobre él se transforma en una mano con el dedo índice extendido. Si pulsamos sobre el enlace saltamos a otra parte del documento, a otro documento situado en cualquier lugar, o incluso se abre el programa de correo para enviar un mensaje a la dirección indicada.

Los documentos HTML son hipertextos cuyo funcionamiento se debe, en su mayor parte, a los hiperenlaces o anclajes, cuya marca específica es `<a>`. El elemento `a` es la abreviatura de "anchor" (ancla, en español). Y el atributo `href` es la abreviatura de "hypertext reference" (es decir, referencia hipertextual), que especifica hacia dónde nos dirige el enlace; normalmente, una dirección de internet o un nombre de fichero.

En general los enlaces tienen la siguiente estructura

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

Los enlaces se utilizan para establecer relaciones entre dos recursos. Aunque la mayoría de enlaces relacionan páginas web, también es posible enlazar otros recursos como imágenes, documentos y archivos. El enlace puede ser de los siguientes tipos:

Enlace a una dirección de Internet

El "valor" declarado en la estructura del enlace hará referencia a una URL. El acrónimo URL (del inglés Uniform Resource Locator) hace referencia al identificador único de cada recurso disponible en Internet. Las URL permiten que cada página HTML publicada en Internet tenga un nombre único que permita diferenciarla de las demás. De esta forma es posible crear enlaces que apunten de forma inequívoca a una determinada página.

Si se accede a la página principal de Google, la dirección que muestra el navegador es:

```
http://www.google.com
```

La cadena de texto `http://www.google.com` es la URL completa de la página principal de Google. La URL de las páginas es imprescindible para crear los enlaces, ya que permite distinguir una página de otra.

El segundo objetivo de las URL es el de permitir la localización eficiente de cada recurso de Internet. Para ello es necesario comprender las diferentes partes que forman las URL. Una URL sencilla siempre está formada por las mismas tres partes. Si por ejemplo se considera la siguiente URL:

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

Las partes que componen la URL anterior son:

- **Protocolo** (`http://`): el mecanismo que debe utilizar el navegador para acceder a ese recurso. Todas las páginas web utilizan `http://`. Las páginas web seguras (por ejemplo las de los bancos y las de los servicios de email) utilizan `https://` (se añade una letra s).
- **Servidor** (`www.librosweb.es`): simplificando mucho su explicación, se trata del ordenador en el que se encuentra guardada la página que se quiere acceder. Los navegadores son capaces de obtener la dirección de cada servidor a partir de su nombre.
- **Ruta** (`/libro/xhtml/capitulo4.html`): camino que se debe seguir, una vez que se ha llegado al servidor, para localizar el recurso específico que se quiere acceder.



Veamos ahora mismo un ejemplo de cómo se codificaría un enlace:

```
<a href="http://www.google.com">Desde aquí enlazamos con HTML.net</a>
```

Direcciones absolutas y relativas:

Las *direcciones absolutas* son aquellas que contienen la URL completa. En estas direcciones podemos ver todos los elementos de la dirección. Un ejemplo muy claro es el siguiente:

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

En las *direcciones relativas* vemos que no está la URL completa. En este tipo de direcciones, las partes que faltan de la dirección el navegador las genera de la propia página en la que está el enlace. Por ejemplo:

URL relativa: `/ruta1/ruta2/pagina2.html`

El navegador entiende que dentro de la página y de la carpeta en la que se encuentra, debe dirigirse al archivo "pagina2.html". En el ejemplo anterior, las dos URL son equivalentes porque cuando no se indica el protocolo y el servidor de una URL, los navegadores suponen que son los mismos que los de la página origen.

Enlace a otra página local

En este caso la URL se dará de forma relativa (igual que para poner un fondo a la página). Por ejemplo si tenemos dos páginas en el mismo directorio `pagina1.html` y `pagina2.html` y queremos poner un enlace desde la `pagina1.html` a la `pagina2.html` debemos escribir:

```
<a href="pagina2.html">Haz clic para ir a la página 2</a>
```

Con esto es suficiente, no hay que crear ningún marcador a no ser que quieras ir a un sitio concreto de la `pagina2.html`. Este caso se muestra en la siguiente explicación.

Enlace a otro lugar del mismo documento

Hasta ahora hemos analizado los enlaces con recursos externos o con páginas diversas de un mismo sitio. Sin embargo, podemos asimismo crear enlaces con puntos específicos de un mismo documento gracias a la siguiente estructura:

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

`Primera parte`: Utilizamos el atributo `name` para dar nombre a una sección de nuestro documento. Posteriormente, cuando en nuestro documento queramos incluir un vínculo a dicha sección escribiremos: `Ir a la primera parte`. El siguiente ejemplo muestra lo mencionado anteriormente, por favor el nombre de este archivo deberá ser `pagina1.html`.

```
<html>
<head> <title>Ejemplo de enlaces</title> </head>
<body>
  <a name="arriba"><h1>Página de enlaces</h1></a>
  <a href="#abajo">Ir abajo</a><br><br><br><br>
  <a href="http://www.abc.es/ciencia/ciencia.asp">Ir a ciencia</a>
<br>...<br>...<br>...<br>...<br>...<br>...<br>...<br>...<br>
<p> Un equipo internacional de astrónomos se acaba de topar con este agujero negro
supermasivo situado en la galaxia CID-947, casi por casualidad. Se trata de un
fenómeno extraordinario ya que la masa de la galaxia es de un tamaño «normal», pero
su agujero negro es gigantesco y no solo eso, crece a un ritmo superior al de su
anfitriona.</p> <br>...<br>...<br>...<br>...<br>...<br>
<p> Todas estas anomalías dan como resultado un comportamiento desconocido hasta
ahora por los astrónomos y es que CID-947 sigue formando estrellas y haciéndose más
grande, el crecimiento de su agujero negro central es mucho más rápido que el de la
galaxia, lo que contradice de lleno lo que se pensaba hasta ahora. <a href="
http://www.abc.es/ciencia/20150724/abci-agujero-negro-videoblog-201507211328.html
">Ver nota original</a></p>
  <br>...<br>...<br>...<br>...<br>...<br>...<br>...<br>
  <br>...<br>...<br>...<br>...<br>...<br>...<br>...<br>
  <br>...<br>...<br>...<br>...<br>...<br>...<br>...<br>
<a href="http://www.abc.es/ciencia/20150711/abci-espacio-agujero-negro-galaxia-
201507102112.html"> Ver otra nota interesante</a>
  <a name="abajo">Fin de nota</a>
  <a href="#arriba">Ir arriba</a>
</body>
</html>
```

Enlaces usando imágenes

Anteriormente vimos enlaces con textos, también podemos extender la capacidad de hacer enlaces con imágenes. De esta forma, cuando pulsemos sobre una imagen, esta actuará como enlace y nos llevará al destino demarcado por el enlace. Para poder hacer esto en html, lo primero será poner una imagen. Para ello nos basamos en la etiqueta `img`:

Para que la imagen acceda a una página web:

```
<a href="http://www.google.com"> </a>
```

Para que la imagen acceda a otro documento:

```
<a href="paginal.html"> </a>
```

Para que la imagen dirija a otra parte del documento:

```
<a href="#partel">Ir a la primera parte  </a>
```

Ejemplo:

```
<html>
<head> <title>Enlaces con imagenes</title> </head>
<body>
  <br>...<br>...<br>
  <a name="inicio"><h1>Primera parte</h1></a>
  <a href="http://www.google.com"> </a>
  <br>...<br>...<br>...<br>
  <br>...<br>...<br>...<br>
  <br>...<br>...<br>...<br>
  <a href="paginal.html"> </a>
  <br>...<br>...<br>...<br>
  <br>...<br>...<br>...<br>
  <br>...<br>...<br>...<br>
  <a href="#inicio"></a>
  <br>...<br>...<br>...<br>
  <br>...<br>...<br>...<br>
  <br>...<br>...<br>...<br>
  <p>Fin de documento</p>
</body>
</html>
```

Enlace con una dirección de correo

Para crear un link de la dirección de un correo electrónico o email en html, por ejemplo del correo electrónico `prueba@prueba.com` tenemos que introducir el siguiente código:

```
<a href="mailto:prueba@prueba.com">Enviar email con html</a>
```

Listas

En ocasiones, es posible agrupar determinadas palabras o frases en un conjunto de elementos que tienen más significado de forma conjunta. El menú de navegación de un sitio web por ejemplo está formado por un grupo de palabras. Aunque cada palabra por separado tiene sentido, de forma conjunta constituyen el menú de navegación de la página, por lo que su significado conjunto es mayor que por separado.

El lenguaje html define tres tipos diferentes de listas para agrupar los elementos:

- **Listas no ordenadas** (se trata de una colección simple de elementos en la que no importa su orden)
- **Listas ordenadas** (similar a la anterior, pero los elementos están numerados y por tanto, importa su orden).
- **Listas de definición** (un conjunto de términos y definiciones similar a un diccionario).

Listas no ordenadas

Las listas no ordenadas van dentro de la etiqueta `` y de su cierre ``. Cada punto que queramos añadir a la lista, lo haremos dentro de la etiqueta `` y su cierre correspondiente. Si queremos definir nosotros mismo el símbolo del punto, podemos gracias al atributo `“type”`.

Así podemos hacer que la lista esté definida por puntos negros (`li type="disc"`), por puntos con el fondo blanco (`li type="circle"`) o por cuadrados (`li type="square"`). Aunque esta apariencia dependerá del navegador. Hay algunos navegadores que dan otras apariencias a estos mismos atributos.

Vamos con un pequeño ejemplo en código:

```
<ul>
<li type="circle">Esto es un tipo de punto.</li>
<li type="square">Este es otro.</li>
<li type="disc">Y este es otro diferente.</li>
</ul>
```

Listas ordenadas

Las listas ordenadas van enmarcadas dentro de las etiquetas `` ``. Cada punto de la lista se escribe con la misma etiqueta que en las no numeradas: ``. Pero al ser listas ordenadas los símbolos serán números y éstos se irán generando automáticamente por orden, conforme escribamos nuevos puntos. En las listas ordenadas podemos hacer que el primer punto comience con el número que nosotros queramos. Lo conseguiremos gracias al atributo `“value”`. Los siguientes puntos que escribamos se generarán automáticamente por orden, partiendo de ese número.

Por ejemplo, si queremos que nuestra lista empiece por el número 20, sólo deberemos escribir lo siguiente:

```
<ol>
<li value="20">Este será el número 20. </li>
<li>Este será el 21. </li>
<li> Este será el 22. Y así sucesivamente. </li>
</ol>
```

Listas de definiciones

Si lo que vamos a hacer es un listado de definiciones, podemos usar las etiquetas `<dl>`, `<dt>` y `<dd>`. Vamos a explicarlas por partes:

La etiqueta `<dl>` viene de los términos ingleses “Definiton list” y nos indica que dentro de ella, entre ella y su cierre, va a ir una definición.

La etiqueta `<dt>` viene de los términos “Definition term” y dentro de ella irá el término que vamos a definir. Para entendernos mejor, dentro de `<dt>` iría el título de la definición.

La etiqueta `<dd>` viene de los términos “Definition description” y nos dice que dentro de ésta irá la definición.

Si escribimos varios listados de definición, éstas se separarán automáticamente entre ellas para facilitar su diferenciación. Aquí podemos ver un ejemplo de código de dos listado de definición:

```
<dl>
<dt>Aquí va el término que definiremos</dt>
<dd>Y aquí dentro irá la definición propiamente dicha.</dd>
</dl>
<dl>
<dt>Aquí va la segunda definición</dt>
<dd>Y aquí dentro irá la segunda definición, separada de la anterior.</dd>
</dl>
```

Ejemplo:

```
<html>
<head> <title>Listas</title> </head>
<body>
  <p><h1>Menu</h1></p>
  <ul>
    <li>Inicio</li>
    <li>Noticias</li>
    <li>Artículos</li>
    <li>Contacto</li>
  </ul><br>
  <p><h1>Instrucciones</h1></p>
  <ol>
    <li>Enchufar correctamente</li>
    <li>Comprobar conexiones</li>
    <li>Encender el aparato</li>
  </ol><br>
  <p><h1>Dulces</h1></p>
  <dl>
    <dt>Chocolate</dt>
    <dd>Elaborado a base de cacao</dd>
    <dt>Caramelo</dt>
    <dd>Elaborado a base de azúcar</dd>
    <dt>Helado</dt>
    <dd>Elaborado a base de crema</dd>
  </dl><br>
</body>
</html>
```

Para el elemento ``, los valores posibles del atributo `type` se resumen en la siguiente tabla:

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

A continuación tenemos una lista ordenada con letras mayúsculas:

```
<ol type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Y aquí esta ordenada la lista con números romanos en minúsculas:

```
<ol type="i">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Ejemplo de una lista, que incluye anidamiento:

```
<html
<head><title>Listas anidadas</title></head>
<body>
  <ul>
    <li>Introducción</li>
    <li>Capítulo 1. Breve repaso de XHTML<ol>
      <li>XHTML y las Hojas de Estilo</li>
      <li>Introducción</li>
      <li>Compatibilidad entre navegadores</li>
      <li>Firefox</li>
      <li>Safari</li>
    </ol></li>
    <li>Apéndices<ol>
      <li>Referencia CSS2</li>
      <li>Glosario</li>
    </ol></li>
  </ul>
  <dl>
    <dt>Firefox</dt>
    <dd>El navegador desarrollado por coporativo Mozilla</dd>
    <dt>Safari</dt>
    <dd>El navegador desarrollado por Apple Inc</dd>
  </dl>
</body>
</html>
```

Tablas

En HTML, una poderosa herramienta a la hora de mostrar cierto tipo de información son las tablas. Las tablas nos permiten representar y ordenar cualquier elemento de nuestra presentación en diferentes filas y columnas de modo que podamos resumir grandes cantidades de información de una manera que puede representarse rápida y fácilmente.

Las tablas más sencillas de html se definen con tres etiquetas: `<table>` para crear la tabla (estructura contenedora principal), `<tr>` para crear cada fila y `<td>` para crear cada columna. La tabla se va definiendo declarando una fila y a continuación las celdas que contiene esa fila, luego otra fila y sus celdas, etc. No es necesario que todas las filas contengan el mismo número de celdas. Veamos un ejemplo sencillo:

html	Salida						
<pre><table> <tr> <td>Celda 1</td> <td>Celda 2</td> <td>Celda 3</td> </tr> <tr> <td>Celda 4</td> <td>Celda 5</td> <td>Celda 6</td> </tr> </table></pre>	<table><tr><td>Celda 1</td><td>Celda 2</td><td>Celda 3</td></tr><tr><td>Celda 4</td><td>Celda 5</td><td>Celda 6</td></tr></table>	Celda 1	Celda 2	Celda 3	Celda 4	Celda 5	Celda 6
Celda 1	Celda 2	Celda 3					
Celda 4	Celda 5	Celda 6					

El tag `<th>` indica que se trata de celdas que sirven como encabezado de tabla y suelen visualizarse en negrita. Y la opción `border` le agrega borde a nuestra tabla.

html	Salida									
<pre><table border="1"> <tr> <th>Cabereca 1</th> <th>Cabereca 2</th> <th>Cabereca 3</th> </tr> <tr> <td>Dato 1</td> <td>Dato 2</td> <td>Dato 3</td> </tr> <tr> <td>Dato 4</td> <td>Dato 5</td> <td>Dato 6</td> </tr> </table></pre>	<table><tr><th>Cabereca 1</th><th>Cabereca 2</th><th>Cabereca 3</th></tr><tr><td>Dato 1</td><td>Dato 2</td><td>Dato 3</td></tr><tr><td>Dato 4</td><td>Dato 5</td><td>Dato 6</td></tr></table>	Cabereca 1	Cabereca 2	Cabereca 3	Dato 1	Dato 2	Dato 3	Dato 4	Dato 5	Dato 6
Cabereca 1	Cabereca 2	Cabereca 3								
Dato 1	Dato 2	Dato 3								
Dato 4	Dato 5	Dato 6								

Celdas extendidas

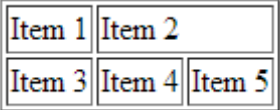
Para la unificación horizontal de celdas:

colspan= más un valor para indicar el número de columnas que se quiere abarcar.

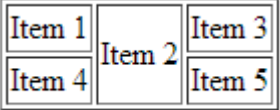
Para la unificación vertical de celdas:

rowspan= más un valor para indicar el número de filas que se quiere abarcar.

Ejemplo de columna expandida <colspan>

html	Salida
<pre><table border> <tr> <td>Item 1</td> <td colspan=2>Item 2</td> </tr> <tr> <td>Item 3</td> <td>Item 4</td> <td>Item 5</td> </tr> </table></pre>	

Ejemplo de linea expandida <rowspan>

html	Salida
<pre><table border> <tr> <td>Item 1</td> <td rowspan=2>Item 2</td> <td>Item 3</td> </tr> <tr> <td>Item 4</td> <td>Item 5</td> </tr> </table></pre>	

A continuación mostraremos varias combinaciones de tablas con las propiedades mencionadas anteriormente.

Combinación de columna expandida y cabecera

html	Salida										
<pre><table border> <tr> <th colspan=2>Head1</th> <th colspan=2>Head2</th> </tr> <tr> <td>A</td> <td>B</td> <td>C</td> <td>D</td> </tr> <tr> <td>E</td> <td>F</td> <td>G</td> <td>H</td> </tr> </table></pre>	<table><tr><td>Head1</td><td>Head2</td></tr><tr><td>A</td><td>B</td><td>C</td><td>D</td></tr><tr><td>E</td><td>F</td><td>G</td><td>H</td></tr></table>	Head1	Head2	A	B	C	D	E	F	G	H
Head1	Head2										
A	B	C	D								
E	F	G	H								

Cabeceras laterales

html	Salida												
<pre><table border> <tr><th>Head1</th> <td>C 1</td> <td>C 2</td> <td>C 3</td> </tr> <tr><th>Head2</th> <td>C 4</td> <td>C 5</td> <td>C 6</td> </tr> <tr><th>Head3</th> <td>C 7</td> <td>C 8</td> <td>C 9</td> </tr> </table></pre>	<table><tr><td>Head1</td><td>C 1</td><td>C 2</td><td>C 3</td></tr><tr><td>Head2</td><td>C 4</td><td>C 5</td><td>C 6</td></tr><tr><td>Head3</td><td>C 7</td><td>C 8</td><td>C 9</td></tr></table>	Head1	C 1	C 2	C 3	Head2	C 4	C 5	C 6	Head3	C 7	C 8	C 9
Head1	C 1	C 2	C 3										
Head2	C 4	C 5	C 6										
Head3	C 7	C 8	C 9										

Combinación de cabeceras laterales y líneas expandidas

html	Salida												
<pre><table border> <tr><th rowspan=2>Head1</th> <td>C 1</td> <td>C 2</td> <td>C 3</td> </tr> <tr><td>C 4</td> <td>C 5</td> <td>C 6</td> </tr> <tr><th>Head2</th> <td>C 7</td> <td>C 8</td> <td>C 9</td> </tr> </table></pre>	<table><tr><td>Head1</td><td>C 1</td><td>C 2</td><td>C 3</td></tr><tr><td></td><td>C 4</td><td>C 5</td><td>C 6</td></tr><tr><td>Head2</td><td>C 7</td><td>C 8</td><td>C 9</td></tr></table>	Head1	C 1	C 2	C 3		C 4	C 5	C 6	Head2	C 7	C 8	C 9
Head1	C 1	C 2	C 3										
	C 4	C 5	C 6										
Head2	C 7	C 8	C 9										

Alineación de celdas

Una vez colocadas las celdas, hay que alinear los datos dentro de cada celda. Así, dentro de cada tag de celda podemos encontrar:

El atributo `align=` define horizontalmente los datos al margen izquierdo (`left`), al derecho (`right`) o centrado (`center`).

El atributo `valign=` define verticalmente los datos en la parte superior (`top`), en la parte inferior (`bottom`) o centrado (`middle`).

Alineado horizontal del contenido de las celdas

html											
<pre><table border> <tr> <th>Cabecera alineado de celdas 1</th> <th>Cabecera alineado de celdas 2</th> <th>Cabecera alineado de celdas 3</th> </tr> <tr align=center> <td>Todas centradas</td> <td>Celda 2</td> <td>Otra celda
 Celda 3</td> </tr> <tr> <td align=right>Alineado a la derecha</td> <td align=center>Alineado al centro</td> <td>Por defecto izquierda</td> </tr> </table></pre>											
Salida											
<table><tr><th>Cabecera alineado de celdas 1</th><th>Cabecera alineado de celdas 2</th><th>Cabecera alineado de celdas 3</th></tr><tr><td>Todas centradas</td><td>Celda 2</td><td>Otra celda Celda 3</td></tr><tr><td>Alineado a la derecha</td><td>Alineado al centro</td><td>Por defecto izquierda</td></tr></table>	Cabecera alineado de celdas 1	Cabecera alineado de celdas 2	Cabecera alineado de celdas 3	Todas centradas	Celda 2	Otra celda Celda 3	Alineado a la derecha	Alineado al centro	Por defecto izquierda		
Cabecera alineado de celdas 1	Cabecera alineado de celdas 2	Cabecera alineado de celdas 3									
Todas centradas	Celda 2	Otra celda Celda 3									
Alineado a la derecha	Alineado al centro	Por defecto izquierda									

Alineado vertical del contenido de las celdas

html																
<pre><table border> <tr> <td>Parrafo 1
 Parrafo 2
 Parrafo 3
 Parrafo 4
 <td valign=top>Alineado arriba</td> <td valign=bottom>Alineado abajo</td> <td>Por defecto
alineado al centro</td> </tr> </table></pre>																
Salida																
<table><tr><td>Parrafo 1</td><td>Alineado arriba</td><td></td><td>Por defecto</td></tr><tr><td>Parrafo 2</td><td></td><td></td><td>alineado al centro</td></tr><tr><td>Parrafo 3</td><td></td><td>Alineado abajo</td><td></td></tr><tr><td>Parrafo 4</td><td></td><td></td><td></td></tr></table>	Parrafo 1	Alineado arriba		Por defecto	Parrafo 2			alineado al centro	Parrafo 3		Alineado abajo		Parrafo 4			
Parrafo 1	Alineado arriba		Por defecto													
Parrafo 2			alineado al centro													
Parrafo 3		Alineado abajo														
Parrafo 4																

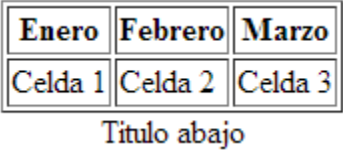
Títulos

Mediante el elemento `caption`, podemos definir el título de una tabla. Este título debería describir de una manera breve y precisa el contenido y la naturaleza de la tabla y es habitualmente representado en algún lugar cercano a la tabla, como arriba o debajo de ella.

Titulo arriba

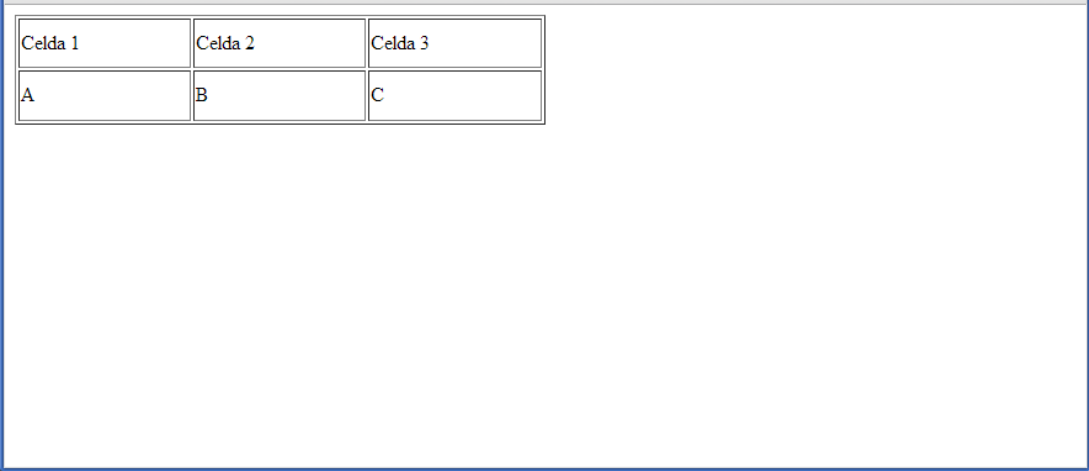
html	Salida						
<pre><table border> <caption align=top>Titulo arriba</caption> <tr> <th>Enero</th> <th>Febrero</th> <th>Marzo</th> </tr> <tr> <td>Celda 1</td> <td>Celda 2</td> <td>Celda 3</td> </tr> </table></pre>	<div>Titulo arriba</div> <table><tr><td>Enero</td><td>Febrero</td><td>Marzo</td></tr><tr><td>Celda 1</td><td>Celda 2</td><td>Celda 3</td></tr></table>	Enero	Febrero	Marzo	Celda 1	Celda 2	Celda 3
Enero	Febrero	Marzo					
Celda 1	Celda 2	Celda 3					

Titulo abajo

html	Salida
<pre><table border> <caption align=bottom>Titulo abajo</caption> <tr> <th>Enero</th> <th>Febrero</th> <th>Marzo</th> </tr> <tr> <td>Celda 1</td> <td>Celda 2</td> <td>Celda 3</td> </tr> </table></pre>	


Longitud de tablas

Tabla que ocupa el 50 % horizontal de la pantalla, y el 25 % vertical.

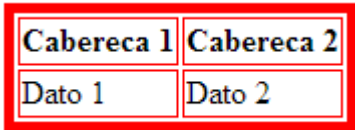
html
<pre><table border width="50%" height="25%"> <tr> <td>Celda 1</td> <td>Celda 2</td> <td>Celda 3</td> </tr> <tr> <td> A </td> <td> B </td> <td> C </td> </tr> </table></pre>
Salida


Colores

Una tabla de cuatro celdas. Cada una de un color.

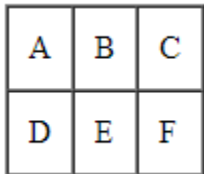
html	Salida
<pre><table border> <caption align=bottom>Titulo abajo</caption> <tr> <td bgcolor="red">Texto ROJO</td> <td bgcolor="green">Texto VERDE</td> </tr> <tr> <td bgcolor="blue">Texto AZUL</td> <td bgcolor="yellow">Texto AMARILLO</td> </tr> </table></pre>	

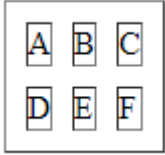
Una tabla con un borde de color

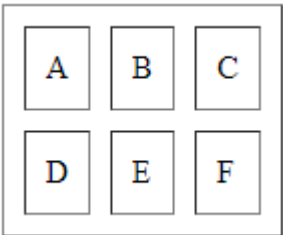
html	Salida
<pre><table border="5" bordercolor="red"> <tr> <th>Cabereca 1</th> <th>Cabereca 2</th> </tr> <tr> <td>Dato 1</td> <td>Dato 2</td> </tr> </table></pre>	

Dimensionado de celdas

El espacio que se encuentra entre los bordes de la celda y su contenido se define con el atributo `cellpadding`. Y el espacio que se encuentra entre celda y celda está definido con el atributo `cellspacing`.

html	Salida
<pre><table border cellpadding="10" cellspacing="0"> <tr> <td>A</td> <td>B</td> <td>C</td> </tr> <tr> <td>D</td> <td>E</td> <td>F</td> </tr> </table></pre>	

html	Salida
<pre> <table border cellpadding="0" cellspacing="10"> <tr> <td>A</td> <td>B</td> <td>C</td> </tr> <tr> <td>D</td> <td>E</td> <td>F</td> </tr> </table> </pre>	

html	Salida
<pre> <table border cellpadding="10" cellspacing="10"> <tr> <td>A</td> <td>B</td> <td>C</td> </tr> <tr> <td>D</td> <td>E</td> <td>F</td> </tr> </table> </pre>	

Referencias:

<http://platea.pntic.mec.es/~abercian/guiahtml/enlaces.htm>

http://librosweb.es/libro/xhtml/capitulo_4/enlaces_basicos.html

<http://www.virtualnauta.com/html-tablas>

<https://www.uv.es/jac/guia/tablaeje.htm>

Los frames

Los frames (marcos en español) en html, permiten a los autores presentar documentos con vistas múltiples, que pueden ser ventanas o subventanas independientes. Las vistas múltiples ofrecen a los autores una manera de mantener cierta información visible mientras otras vistas se desplazan o se sustituyen. Por ejemplo, dentro de una misma ventana, un marco podría mostrar un gráfico estático, un segundo marco un menú de navegación, y un tercero el documento principal que puede ser desplazado, o reemplazado al navegar por el segundo marco.

Los frames se declaran con la etiqueta <frameset>, que debe ser colocada antes de la etiqueta <body>. La etiqueta <frameset> se usa para dividir la ventana del navegador en diferentes marcos y a cada uno asignarle una medida distinta. Si no se cumple este requisito, la etiqueta se ignorará. La estructura es la siguiente:

```
<frameset rows="valor1, valor2, valor_n" cols="valor3, valor4, valor_n">
...el resto de la definición...
</frameset>
```

Cada frameset define un grupo de filas (rows) y columnas (cols). Los posibles valores para definir el tamaño de las filas y de las columnas son en: pixels, %, *.

Los frames permiten una flexibilidad de presentación extraordinaria, y para contenidos muy complejos, de difícil estructura por los medios convencionales del HTML, son casi insustituibles. Los frames son parametrizables en cuanto a tamaño y número de áreas, si éstas se pueden redimensionar por el usuario o son fijas, si tienen o no barras de scrolling, se pueden anidar, relacionar sus contenidos, etc. Antes de comenzar con los ejemplos necesitaremos de tres páginas html como se muestran a continuación.

Texto_1.html	Texto_2.html	Texto_3.html
<pre><html> <head><title>1</title> </head> <body> <p align=center><h1>1</h1> </p> Texto_1 </body> </html></pre>	<pre><html> <head><title>2</title> </head> <body> <p align=center><h1>2</h1> </p> Texto_2 </body> </html></pre>	<pre><html> <head><title>3</title> </head> <p align=center><h1>3</h1> </p> <body> Texto_3 </body> </html></pre>

Un ejemplo de frame sencillo es el siguiente:

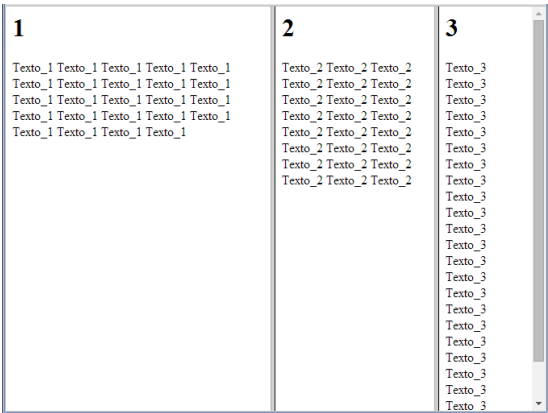
```
<html>
<head><title>Frames 1</title></head>
  <frameset rows="40%,60%">
    <frame src="Texto_1.html" >
    <frame src="Texto_2.html" >
  </frameset>
</html>
```

Para prevenir el caso de que el navegador del visitante no soporte frames. Se utiliza el tag `<noframe>` y dentro de dichos tags se incluye `<body>` texto de aviso `</body>`. Si el navegador no reconoce los frames los ignorara, mostrando el mensaje de aviso. Este atributo está actualmente en desuso, ya que todos los modernos navegadores soportan frames. Un ejemplo de una página simple incluyendo el `noframe` sería:

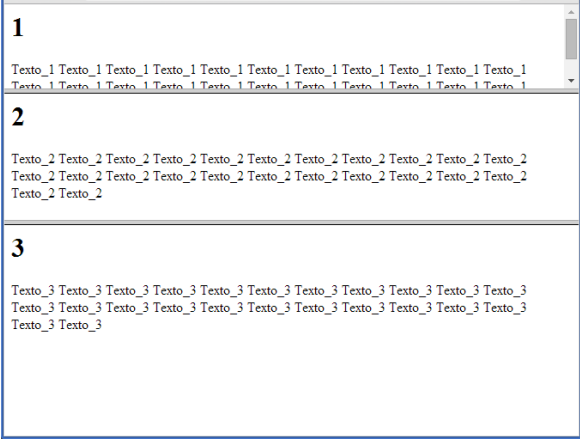
```
<html>
<head><title>Frames 2</title></head>
<frameset cols="25%,75%">
  <noframes>
    <body>
      Su navegador no soporta frames
    </body>
  </noframes>
<frame src="Texto_2.html">
<frame src="Texto_3.html">
</frameset>
</html>
```

El atributo `src` especifica qué página se va a cargar en el frame. La ruta de la página debe especificar su localización en el sistema de archivos local del servidor web o una URL completa. Realizaremos unos ejemplos de frames.

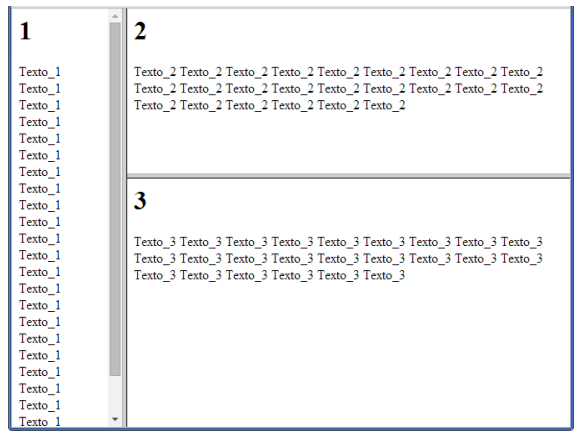
Un frame con varias columnas:

html	Salida
<pre><html> <head><title>Frames 3</title></head> <frameset cols="50%,30%,20%"> <frame src="Texto_1.html"> <frame src="Texto_2.html"> <frame src="Texto_3.html"> </frameset> </html></pre>	

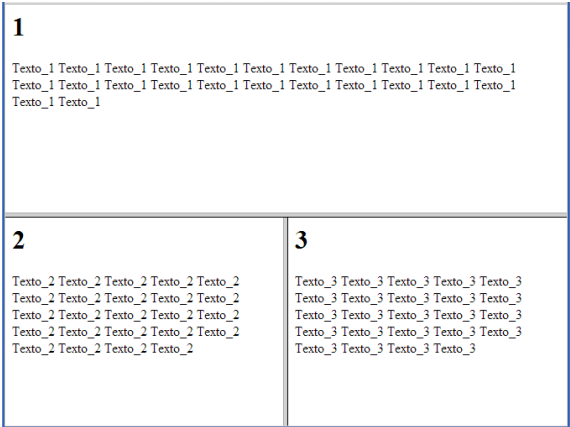
Un frame con varias filas:

html	Salida
<pre><html> <head><title>Frames 4</title></head> <frameset rows="10%,30%,60%"> <frame src="Texto_1.html"> <frame src="Texto_2.html"> <frame src="Texto_3.html"> </frameset> </html></pre>	

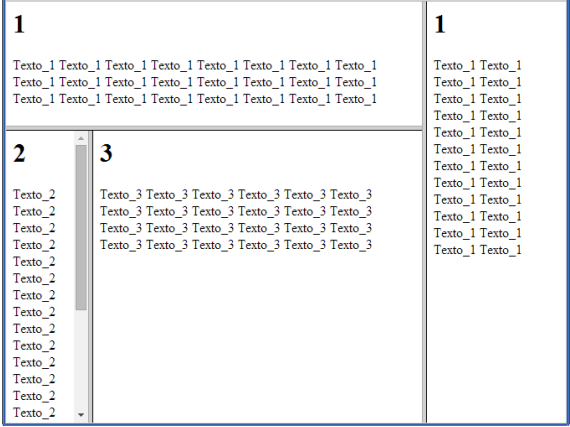
Un frame con una columna y dos filas anidadas

html	Salida
<pre><html> <head><title>Frames 5</title></head> <frameset cols="20%,*"> <frame src="Texto_1.html"> <frameset rows="40%,*"> <frame src ="Texto_2.html"> <frame src ="Texto_3.html"> </frameset> </frameset> </html></pre>	

Un frame con una fila y dos columnas anidadas

html	Salida
<pre><html> <head><title>Frames 6</title></head> <frameset rows="50%,*"> <frame src="Texto_1.html"> <frameset cols="50%,*"> <frame src ="Texto_2.html"> <frame src ="Texto_3.html"> </frameset> </frameset> </html></pre>	

Un frame con filas y columnas

html	Salida
<pre><html> <head><title>Frames 7</title></head> <frameset cols="75%,25%"> <frameset rows="30%,70%"> <frame src="Texto_1.html"> <frameset cols="20%,80%"> <frame src="Texto_2.html"> <frame src="Texto_3.html"> </frameset> </frameset> <frame src="Texto_1.html"> </frameset> </html></pre>	

El atributo `frameborder` establece si serán visibles los bordes del frame o no. Puede tener dos valores: `yes` (por defecto) y `no`. Un ejemplo:

<pre><html> <head><title>Frames 8</title></head> <frameset frameborder="no" cols="30%,30%,30%"> <frame src="Texto_1.html"> <frame src="Texto_2.html"> <frame src="Texto_3.html"> </frameset> </html></pre>
--

El atributo `bordercolor` establece el color de los bordes visibles. Un ejemplo:

```
<html>
<head><title>Frames 9</title></head>
<frameset border="10" bordercolor="yellow" cols="30%,30%,30%">
  <frame src="Texto_1.html">
  <frame src="Texto_2.html">
  <frame src="Texto_3.html">
</frameset>
</html>
```

El atributo `noresize` indica que el frame no se puede redimensionar. Si no se pone este atributo colocando el cursor en el borde del frame, permitiría su deslizamiento.

```
<html>
<head><title>Frames 10</title></head>
<frameset cols="30%,30%,*">
  <frame src="Texto_1.html" noresize>
  <frame src="Texto_2.html">
  <frame src="Texto_3.html">
</frameset>
</html>
```

El atributo `scrolling` nos indica si el frame llevará siempre (yes), nunca (no) o cuando lo necesite (auto), barra de deslizamiento vertical.

```
<html>
<head><title>Frames 10</title></head>
<frameset cols="15%,15%,*">
  <frame src="Texto_1.html" scrolling="yes">
  <frame src="Texto_2.html" scrolling="no">
  <frame src="Texto_3.html">
</frameset>
</html>
```

El atributo `marginwidth` se utiliza para precisar el número de píxels entre los bordes izquierdo y derecho de la zona del documento html a visualizar.

El atributo `marginheight` Se utiliza para precisar el número de píxels entre los bodes superior e inferior de la zona del documento html a visualizar.

```
<html>
<head><title>Frames 11</title></head>
<frameset cols="50%,50%">
  <frame src="Texto_1.html">
  <frame src="Texto_2.html" marginwidth="50" marginheight="50">
</frameset>
</html>
```

Acceso a otros frames

En principio, en una página con frames, un enlace actualiza sólo el frame en el que está. Es decir si ponemos un enlace en una página con frames, la página a la que apunta se verá en el mismo frame en el que está el enlace. Este no es en la mayoría de los casos el comportamiento que queremos, ya que normalmente, al pinchar en un enlace, lo que queremos es actualizar otro frame de la misma página, o cambiar totalmente de página para ir a otra. Debemos pues dirigir los enlaces para que aparezcan en otra ventana, y no en la propia. Para dirigir un enlace hacia otro frame de la misma página principal, debemos en primer lugar darle un nombre al frame donde se va a ver el enlace, esto se hace mediante el atributo `name`; siguiendo el ejemplo de los apartados anteriores, modificamos en la página principal el código del frame de la columna central de contenido, dándole un nombre:

```
<frame src="pagina.html" name="nombre"/>
```

Ahora debemos escribir el enlace dentro de la página correspondiente al frame donde queremos verlo, por ejemplo en la columna izquierda de menú. Este enlace, aparte del atributo `href` "...", debe llevar el atributo `target` "...". El valor de este atributo debe ser el valor que le hayamos dado a la etiqueta `name` en el frame que queremos actualizar. Abrimos el archivo "pagina.html" y escribimos el siguiente enlace:

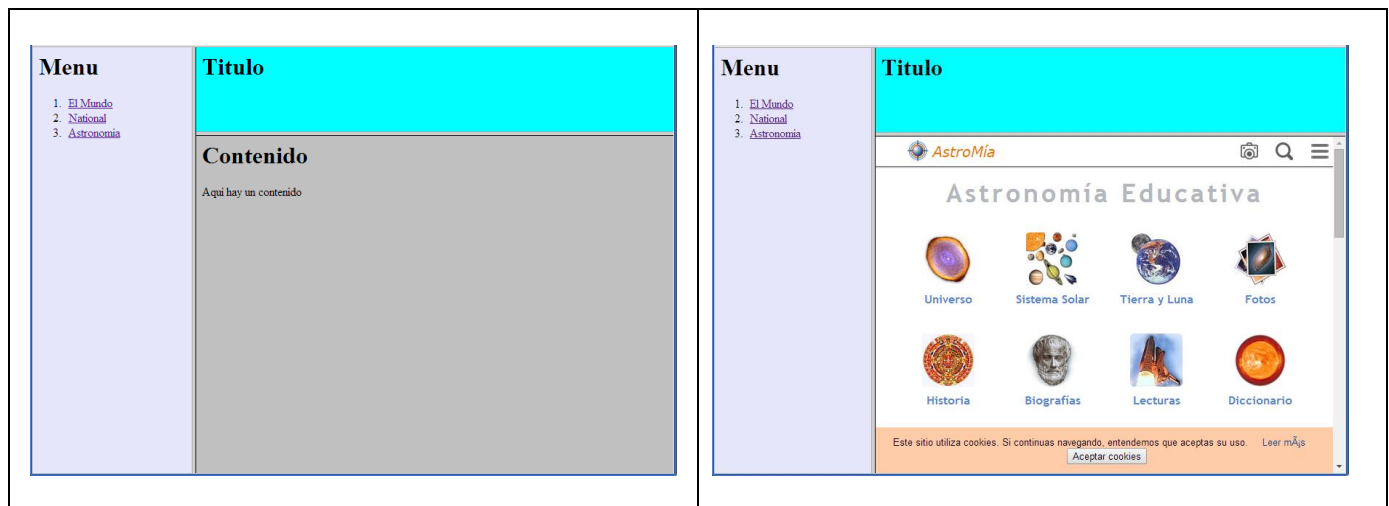
```
<a href="http://www.mipagina.com" target="nombre">Mi pagina</a>
```

Ejemplo:

frame_name.html
<pre><html> <head><title>Frames 12</title></head> <frameset cols="25%,*"> <frame src="menu.html"> <frameset rows="20%,*"> <frame src ="titulo.html"> <frame name="contenido" src ="contenido.html" > </frameset> </frameset> </html></pre>

menu.html
<pre><html> <head> <title>Menu</title> </head> <body bgcolor="#E6E6FA"> <p align=center><h1>Menu</h1></p> El Mundo National Astronomia </body> </html></pre>

titulo.html	contenido.html
<pre> <html> <head> <title>Titulo</title> </head> <body bgcolor="cyan"> <p align=center> <h1>Titulo</h1> </p> </body> </html> </pre>	<pre> <html> <head> <title>Contenido</title> </head> <body bgcolor="blue"> <p align=center> <h1>Contenido</h1> </p> Aquí hay un contenido </body> </html> </pre>



Existen unos valores especiales de `target` que nos permitirán definir destinos distintos a las frames definidas. Estos valores son los siguientes:

`target = "_blank"`: Indica que se muestre en una nueva ventana vacía, para seguir el enlace se lanza otra ventana distinta del navegador.

`target = "_top"`: Indica que se muestre en la ventana completa, eliminando la estructura de frames que tenga la ventana.

`target = "_self"`: Se mostrará en la misma ventana o frame que lo referencia.

`target = "_parent"`: Se muestra en la frame o estructura de frames que llamó al documento actual.

Ejemplo:

frame_target.html

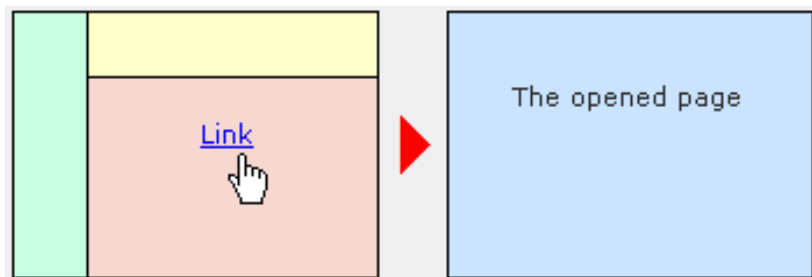
```
<html>
<head><title>Frames 12</title></head>
<frameset cols="25%,*">
  <frame src="Texto_1.html">
    <frameset rows="20%,*">
      <frame src="Texto_2.html">
        <frame src="target_blank.html">
      </frameset>
    </frameset>
  </frameset>
</html>
```

Un ejemplo usando _blank en el target:

target_blank.html

```
<html>
<head>
<title>Link</title>
</head>
<body bgcolor="green">
<p align=center>
<h1>Link</h1>
</p>
Noticias de la Ciencia y Tecnología
<a href="http://noticiasdelaciencia.com/sec/tecnologia/" target="_blank">Lo Nuevo en
Tecnología</a></p>
</body>
</html>
```

target="_top"



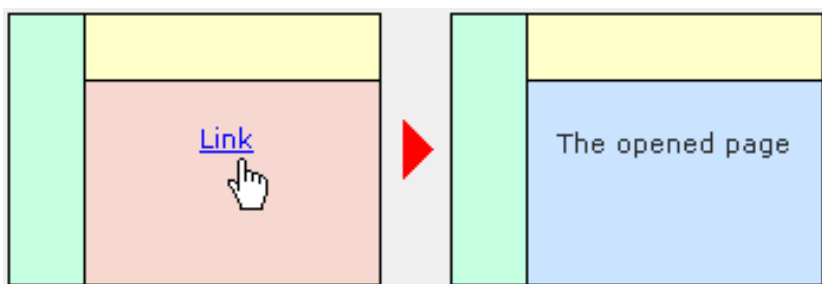
Se sustituye el tag `<frame src="target_blank.html">` por `<frame src="target_top.html">` en el archivo: frame_target.html.

Un ejemplo usando `_top` en el target:

target_top.html

```
<html>
<head>
<title>Link</title>
</head>
<body bgcolor="green">
<p align=center>
<h1>Link</h1>
</p>
Noticias de la Ciencia y Tecnología
<a href="http://noticiasdelaciencia.com/sec/tecnologia/" target="_top">Lo Nuevo en
Tecnología</a></p>
</body>
</html>
```

target="_self"



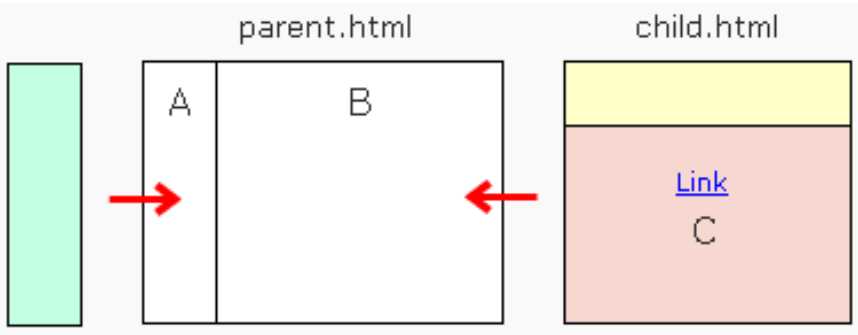
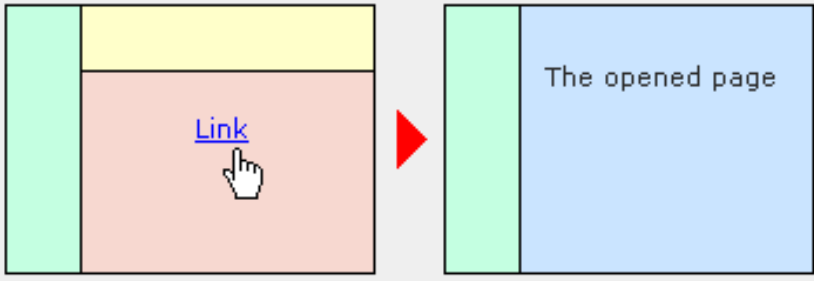
Se sustituye el tag `<frame src="target_blank.html">` por `<frame src="target_self.html">` en el archivo: `frame_target.html`.

Un ejemplo usando `_self` en el target:

target_self.html

```
<html>
<head>
<title>Link</title>
</head>
<body bgcolor="green">
<p align=center>
<h1>Link</h1>
</p>
Noticias de la Ciencia y Tecnología
<a href="http://noticiasdelaciencia.com/sec/tecnologia/" target="_self">Lo Nuevo en
Tecnología</a></p>
</body>
</html>
```


target="_parent"



Un ejemplo usando `_self` en el target :

parent.html
<pre><html> <head><title>Frames 12</title></head> <frameset cols="25%,*"> <frame src="Texto_1.html"> <frame src="child.html"> </frameset> </frameset> </html></pre>

child.html
<pre><html> <head><title>Frames 12</title></head> <frameset rows="25%,*"> <frame src="Texto_2.html"> <frame src="target_parent.html"> </frameset> </frameset> </html></pre>

target_parent.html

```
<html>
<head>
<title>Link</title>
</head>
<body bgcolor="green">
<p align=center>
<h1>Link</h1>
</p>
Noticias de la Ciencia y Tecnología
<a href="http://noticiasdelaciencia.com/sec/tecnologia/" target="_parent">Lo Nuevo en
Tecnología</a></p>
</body>
</html>
```

Formularios

Los formularios son usados para que el usuario ingrese datos. Por lo que es una sección de un documento que contiene etiquetas de tipo especiales llamados controles (casillas de verificación (checkboxes), radiobotones (radio buttons), menús, etc.), y rótulos (labels) en esos controles los usuarios normalmente "completan" un formulario modificándolos (introduciendo texto, seleccionando objetos de un menú, etc.), y éste los datos del formulario se envían a un agente para que lo procese (p.ej., a un servidor web, a un servidor de correo, etc.)

La etiqueta `<form>` se utiliza para definir un formulario. El atributo `action` es obligatorio y nos indica la dirección a la que se enviará la información.

También contamos con el atributo `method` que nos define el método por el cual se enviarán los datos del usuario al servidor. "POST" es el valor que envía los datos al agente de procesamiento almacenándolos en el cuerpo del formulario, en tanto que "GET" envía los datos agregándolos a la dirección URL y separándolos de la dirección con un signo de interrogación.

Sintaxis

```
<form action="URL" method="valor">
...
</form>
```

La etiqueta input

Nos define el registro donde el usuario puede ingresar los datos. Los usuarios interaccionan con los formularios a través de las llamados controles.

Tipos de controles:

- Botones (buttons)
- Casillas de verificación (checkboxes)
- Radiobotones (radio buttons)
- Menús (menus)
- Entrada de texto (text input)
- Selección de ficheros (file select)
- Controles ocultos (hidden controls)
- Controles tipo objeto (object controls)

El atributo `type` especifica el tipo de control a crear.

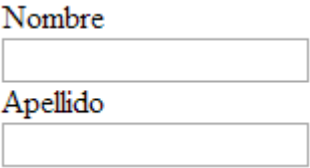
Sintaxis

```
</form>
<input type="control" name="valor">
</form>
```

Estos son los valores posibles del atributo `type`:

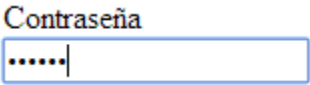
`type = "text | password | checkbox | radio | submit | reset | file | hidden | image | button"` - Indica el tipo de control que se incluye en el formulario

`text`: Representa una casilla de texto para escribir una línea de texto. El tamaño de la casilla puede definirse usando el atributo `size` y la extensión máxima del texto con el atributo `maxlength`.

html	Salida
<pre><html> <head> <title>Link</title> </head> <body> <form> Nombre
 <input type="text" name="nombre" value=""/>
 Apellido
 <input type="text" name="apellid" value=""/> </form> </body> </html></pre>	

El valor que se indica en el atributo `name` es el nombre que utiliza la aplicación del servidor para obtener el valor de este campo de formulario. El atributo `value` se emplea para establecer el valor inicial del cuadro de texto. Si no se especifica un tamaño, el navegador muestra el cuadro de texto con un tamaño predeterminado.

`password`: Una casilla de texto donde los caracteres escritos aparecen como asteriscos para camuflar el texto de entrada, por lo que es ideal para escribir contraseñas y otros datos sensibles.

html	Salida
<pre><html> <head> <title>Link</title> </head> <body> <form> Contraseña
 <input type="password" name="contra" value=""/> </form> </body> </html></pre>	

checkbox: Son controles de formulario que permiten al usuario seleccionar y deselectar opciones individualmente. Aunque en ocasiones se muestran varios checkbox juntos, cada uno de ellos es completamente independiente del resto. Por este motivo, se utilizan cuando el usuario puede activar y desactivar varias opciones relacionadas pero no excluyentes. Las casillas pueden adoptar uno de dos estados: checked (seleccionado) o unchecked (no seleccionado). Cuando la casilla es seleccionada, el par nombre/valor se envía al CGI.

html	Salida
<pre><html> <head> <title>Link</title> </head> <body> <form> Lenguajes de programación
 <input type="checkbox" name="leng_java" value="java"/> Java
 <input type="checkbox" name="leng_cobol" value="cobol"/> Cobol
 <input type="checkbox" name="leng_ruby" value="ruby"/> Ruby
 </form> </body> </html></pre>	<p>Lenguajes de programación</p> <p><input type="checkbox"/> Java</p> <p><input type="checkbox"/> Cobol</p> <p><input type="checkbox"/> Ruby</p>


Si se quiere mostrar un checkbox seleccionado por defecto, se utiliza el atributo checked. Si el valor del atributo es checked, el checkbox se muestra seleccionado.

```
<input type="checkbox" checked="checked" ... /> Checkbox seleccionado por defecto
```

radio: Un botón que permite al usuario elegir entre varias opciones, son similares a los controles de tipo checkbox, pero presentan una diferencia muy importante: son mutuamente excluyentes. Cada uno de estos botones debe tener el mismo atributo name. El par nombre/valor del botón radio seleccionado se enviará al CGI. Al aplicar el atributo checked para uno de estos botones se definirá como seleccionado de forma predeterminada.

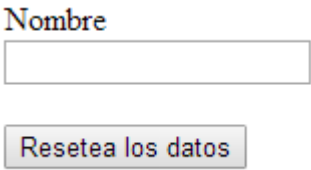
html	Salida
<pre><html> <head> <title>Link</title> </head> <body> <form> Selecciona una opción de color
 <input type="radio" name="color" value="azul" checked="checked"/> Azul
 <input type="radio" name="color" value="rojo"/> Rojo
 <input type="radio" name="color" value="verde"/> Verde
 </form> </body> </html></pre>	<p>Selecciona una opción de color</p> <p><input checked="" type="radio"/> Azul</p> <p><input type="radio"/> Rojo</p> <p><input type="radio"/> Verde</p>

submit: Un botón para enviar al servidor los datos introducidos por el usuario en el formulario. El texto en el botón puede definirse usando el atributo value.

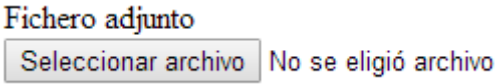
html	Salida
<pre><html> <head> <title>Link</title> </head> <body> <form> <input type="submit" name="datos" value="ok"/> </form> </body> </html></pre>	

reset: Un botón de restauración para quitar todos los elementos en el formulario y restablecer sus valores predeterminados.

html	Salida
<pre><html> <head> <title>Link</title> </head> <body> <form> Nombre
 <input type="text" name="nombre" value=""/>

 <input type="reset" name="limpia" value="Resetea los datos"/> </form> </body> </html></pre>	

file: Es un campo que permite al usuario especificar una ruta de archivo que lleva al archivo que se enviará con el formulario. Los tipos de archivo que pueden ser enviados deben especificarse utilizando el atributo accept de la etiqueta form.

html	Salida
<pre><html> <head> <title>Link</title> </head> <body> <form> Fichero adjunto
 <input type="file" name="adjunto"/> </form> </body> </html></pre>	

Sintaxis para los tipos de archivos

```
<input accept="file_extension|audio/*|video/*|image/*|media_type">
```

Ejemplo:

```
<form action="form_action.asp" accept="image/gif,image/jpeg">  
  First name: <input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  Your image: <input type="file" name="pic" id="pic"><br>  
  <input type="submit" value="Submit">  
</form>
```

hidden: Este atributo establece campos ocultos que no se muestran por pantalla, de forma que el usuario desconoce que el formulario los incluye. Normalmente los campos ocultos se utilizan para incluir información que necesita el servidor pero que no es necesario o no es posible que la establezca el usuario.



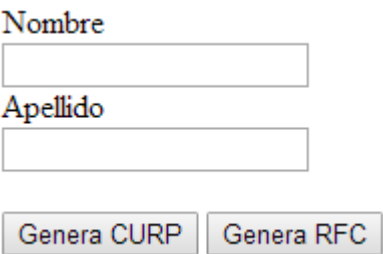
html	Salida
<pre><html> <head> <title>Link</title> </head> <body> <form> Esto se ve
 <input type="text" name="nombre" value=""/>
 <input type="hidden" name="oculto" value="Esto no se ve"/> </form> </body> </html></pre>	<p>Esto se ve</p> 

image: Un botón de envío personalizado que aparece cuando se ubica una imagen en la ubicación definida por el atributo SRC.

html	Salida
<pre><html> <head> <title>Link</title> </head> <body> <form> Imagen en el botón
 <input type="image" name="enviar" src="disk.gif"/> </form> </body> </html></pre>	<p>Imagen en el botón</p> 

button: Este atributo define un botón de tipo genérico.


html	Salida
<pre><html> <head><title>Link</title></head> <body> <form> Nombre
 <input type="text" name="nombre" value=""/>
 Apellido
 <input type="text" name="apellido" value=""/>

 <input type="button" name="curp" value="Genera CURP"/> <input type="button" name="rfc" value="Genera RFC"/> </form> </body> </html></pre>	

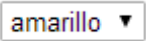
Si pruebas a pulsar un botón de este tipo, verás que el navegador no hace nada: no envía los datos al servidor y no borra los datos introducidos. Este tipo de botones sólo son útiles si se utilizan junto con el lenguaje de programación JavaScript. Por lo que los botones de este tipo se pueden programar para que realicen cualquier tarea compleja cuando se pulsa sobre ellos.

Listas de opciones


Las listas de opciones son de gran ayuda para los formularios en los que queremos saber la opinión de un aspecto concreto en nuestra Web, dentro de diferentes opciones planteadas previamente por nosotros. Las listas de selección u opciones se escriben gracias a la etiqueta `<select>` y a su cierre. Veamos un ejemplo:

html	Salida
<pre><html> <head><title>Select</title></head> <body> <form> <select name="colores"> <option value="blanco">blanco</option> <option value="amarillo">amarillo</option> <option value="verde">verde</option> <option value="rojo">rojo</option> </select> </body> </form> </html></pre>	

Es importante observar la primera opción aparece seleccionada por defecto es la primera de la lista. Podemos elegir la opción que queremos, con la propiedad `selected`.

html	Salida
<pre><html> <head><title>Select</title></head> <body> <form> <select name="colores"> <option value="blanco">blanco</option> <option value="amarillo" selected> amarillo</option> <option value="verde">verde</option> <option value="rojo">rojo</option> </select> </body> </form> </html></pre>	

Podemos personalizar esta lista de opciones mediante una serie de atributos. Por ejemplo, podemos hacer que se vea más de una opción a la vez en la lista. Para ello utilizaremos `size`, indicando el número de opciones que queremos que se muestren. Si quisiéramos que se vieran tres opciones:

html	Salida
<pre><html> <head><title>Select</title></head> <body> <form> <select name="transporte" size="3"> <option>Coche</option> <option>Avión</option> <option>Tren</option> <option>Camión</option> <option>Moto</option> </select> </body> </form> </html></pre>	

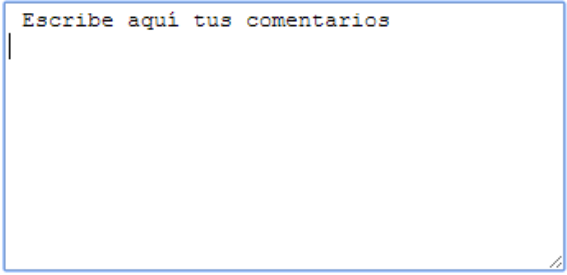
También podemos definir si queremos que se pueda elegir más de una opción a la vez mediante las teclas `ctrl` y `shift`. Para ello utilizaremos el atributo `multiple`. Mira primero cómo quedaría nuestro código.

Campos de texto

Si deseamos poner a la disposición de usuario un campo de texto donde pueda escribir cómodamente sobre un espacio compuesto de varias líneas, hemos de invocar una nueva etiqueta: `<textarea>` y su cierre correspondiente.

Sus atributos básicos son “rows” y “cols” que servirán para definir el tamaño del `textarea`. Además, no debemos olvidar el atributo “name”, que servirá para definir el nombre de este campo.

Un `textarea` llamado “comentarios” y con unas dimensiones de 10 filas y de 40 columnas, tendría el código siguiente:

html	Salida
<pre><html> <head><title>Campo de texto</title></head> <body> <form> <textarea name="comentarios" rows="10" cols="40"> Escribe aquí tus comentarios </textarea> </body> </form> </html></pre>	

Referencias:

<https://www.uv.es/jac/guia/frame.htm>

<http://www.virtualnauta.com/html-frames>

http://www.tutorialspoint.com/html/html_frames.htm

http://www.tagindex.net/html/frame/a_target.html

<http://www.virtualnauta.com/html-formularios>

<http://es.ccm.net/contents/236-formularios-html>

http://www.w3schools.com/html/html_forms.asp

http://librosweb.es/libro/xhtml/capitulo_8/elementos_de_formulario.html

http://www.pcweb.es/manual_html/mostrar.php?opcion=select