

## Indice de contenidos

- Versiones del HTML
  - HTML 2.0
  - HTML 3.0 y 3.2
  - HTML 4.0
  - Estándares en este curso
- Mi primera página
  - El código
  - La explicación
  - El cuerpo del documento
- Formateo básico
  - Formato del párrafo
  - Las 6 cabeceras
  - Cambiando el tipo de letra
  - Formato de frase
  - Otros elementos
- Caracteres especiales
  - Caracteres extendidos en HTML
  - Caracteres de control
- Enlaces
  - La etiqueta <A>
  - Las URLs
  - Anclas
- Listas
  - Listas desordenadas
  - Listas ordenadas
  - Listas de definiciones
- Imágenes
  - Imágenes y enlaces
  - Alineación respecto al texto
- Formateo fino
  - Cambio de color
  - Tamaños del texto
  - Tipo de letra
- Estructura del documento
  - La cabecera
  - El cuerpo
- Formularios
  - Cajas de texto
  - Opciones
  - Botones del formulario
  - Otros elementos
- Tablas
  - Definir las filas
  - Definir las celdas
  - Título de la tabla
- Marcos
  - Etiqueta <FRAMESET>
  - Etiqueta <FRAME>
  - Acceso a otros marcos

## 1. Versiones del HTML

### HTML 2.0

Cuando se produjo la explosión de Internet el estándar de HTML que circulaba era el 2.0 (establecido en noviembre del 95), de modo que cualquier navegador que utilizado actualmente es capaz de interpretarlo.

### HTML 3.0 y 3.2

Aunque la versión 2.0 cumplía bien el objetivo para el que se creó, carecía de herramientas para tener un control mínimamente complejo de los documentos. No se consideró necesario que lo tuviera, ya que por aquel entonces Internet era un fenómeno más bien circunscrito a la actividad académica y el contenido primaba sobre el diseño. Debido a ello, Netscape (líder del mercado de navegadores por aquel entonces) empezó a incluir etiquetas nuevas no incluidas en ningún estándar.

Por estos problemas, el IETF (el comité que suele decidir todos los estándares dentro de Internet) comenzó a elaborar el borrador del HTML 3.0, que resultó ser demasiado grande para la época, lo que dificultó su aceptación en el mercado.

Esto llevó a una serie de compañías (entre ellas Netscape, Microsoft, IBM, Sun, etc...) a crear un nuevo comité llamado W3C, que es el que actualmente elabora las nuevas versiones del HTML. Su primer trabajo consistió en crear el borrador del HTML 3.2, que incluía muchas de las mejoras que los principales navegadores (Netscape y Explorer) habían introducido en Internet, como son las tablas, los *applets*, etc..

Este borrador fue aprobado en enero de 1997 e inmediatamente el W3C se puso a trabajar en la elaboración del siguiente estándar: el 4.0.

## **HTML 4.0**

En julio de 1997 se presenta el borrador de este estándar. Por fin se estandarizan los marcos (*frames*), las hojas de estilo y los *scripts* (entre otras cosas). El 17 de diciembre de 1997 dicho borrador fue finalmente aprobado.

## 2. Estructura del documento

La estructura de un documento HTML se puede resumir así:

```
<HTML>
<HEAD>
  <TITLE>titulo de la página</TITLE>
  cosas que afectan a la página pero no a su contenido
</HEAD>
<BODY parámetros>
  contenido de la página
</BODY>
</HTML>
```

### La cabecera

Suele ser el lugar más indicado para colocar aquellos elementos de la página que no alteren el contenido de la misma, aunque si la forma de presentarlo y su comportamiento. Es por eso que es el lugar más recomendable para colocar los scripts y las hojas de estilo

Además del título de la página, uno de los elementos que se pueden incluir aquí son los META. Entre otras cosas, sirven para indicar propiedades de la página como pueda ser el nombre de su autor. Por ejemplo,

```
<META NAME="GENERATOR" CONTENT="Mozilla/4.03 [es] (Win95; I
[Netscape]">
```

<b>Propiedad</b>	<b>Utilidad</b>
AUTHOR	Autor de la página.
GENERATOR	Herramienta utilizada para hacer la página.
CLASSIFICATION	Palabras que permite clasificar la página dentro de un buscador jerárquico (como Yahoo).
KEYWORDS	Palabras clave por las que encontrarán la página en los buscadores.
DESCRIPTION	Descripción del contenido de la página.

Hay también otro elemento interesante que podremos incluir en nuestras cabeceras. Cuando especificamos una URL relativa en un enlace, en principio es para acceder a una página situada en nuestro mismo servidor. Sin embargo, si especificamos:

```
<BASE HREF="http://www.hornet.org/music">
```

Ahora todas nuestras URLs relativas se referirán al directorio /music dentro del servidor <http://www.hornet.org>.

## El cuerpo

Obviamente no voy a explicar lo que entra dentro del cuerpo (prácticamente todos los capítulos del curso intentan hacerlo) sino los parámetros que admite la etiqueta <BODY>:

BACKGROUND	Permite definir un gráfico de fondo para la página.
BGCOLOR	Permite definir el color de fondo de la página.
TEXT	Cambia el color del texto.
LINK	Cambia el color de un enlace no visitado (por defecto azul).
VLINK	Cambia el color de un enlace ya visitado (por defecto púrpura).
ALINK	Cambia el color que toma un enlace mientras lo estamos pulsando (por defecto rojo).
LEFTMARGIN y TOPMARGIN	Especifican el número de pixels que dejará de margen entre el borde de la ventana y el contenido de la página. Se suelen utilizar para dejarlos a cero.
MARGINWIDTH y MARGINHEIGHT	Más o menos equivalentes a los anteriores, pero éstos funcionan en Netscape.

No resulta recomendable cambiar los colores del texto y enlaces a no ser que exista alguna dificultad al leerlos por haber cambiado el fondo, ya que en muchas ocasiones el usuario ha podido cambiarlos en las opciones de su navegador y estarán ya a su gusto.

### 3. El primer script de HTML

Como se dice que la mejor forma de aprender a caminar es caminando, aplicaremos la misma regla y diremos que la mejor forma de programar es programando. Es por ello que para empezar se mostrará un pequeño script de HTML donde se puede ver la base de formación de las páginas WEB.

El script 1.1 está formado por un conjunto de etiquetas y texto plano. El lenguaje HTML es una división del SGML (Standard Generalized Markup Language) que define una forma de lenguaje Standard de etiquetado. El significado de esto en HTML que las acciones que se quieran realizar deben de estar encerradas entre dos etiquetas, a excepción de algunos casos que se irán mostrando.

```
<HTML>
  <HEAD>
    <TITLE>Mi primera pagina</TITLE>
  </HEAD>

  <BODY>
    <CENTER><H1>Mi Primera pagina</H1></CENTER>
    <HR>
    <P>Este es el primer script de HTML. BIENVENIDO AL CURSO.
  </BODY>
</HTML>
```

Script 1.1: Primero.html

Si se observa la sintaxis de HTML se puede ver como están formadas las etiquetas. Una etiqueta está formada por el símbolo “<” seguida de una palabra clave que define el comportamiento de ésta y el símbolo “>” para cerrar la etiqueta.

En la formación de una frase HTML se utilizan dos tipos de etiquetas, la etiqueta de inicio (<texto>) y la etiqueta de fin (</texto>). Todo lo que queda dentro de estas dos etiquetas recibe el tratamiento de tipo de etiqueta, en este caso de texto.

Existen, además de las frases formadas por las etiquetas de inicio y fin, otras en las que sólo es necesario definir la etiqueta de inicio. Este tipo frase de se

aplican como operadores sencillo, como puede ser un salto de línea, un nuevo párrafo, etc.

En la tabla se puede observar las diferentes etiquetas que aparecen en el script 1.1. Observar como algunas sólo tienen inicio y otras tienen ambas, inicio y fin.

<b>Etiqueta</b>	<b>Inicio</b>	<b>Fin</b>	<b>Significado</b>
HTML	<HTML>	</HTML>	Script HTML
HEAD	<HEAD>	</HEAD>	Cabecera de HTML
TITLE	<TITLE>	</TITLE>	Título de ventana
BODY	<BODY>	</BODY>	Cuerpo de HTML
CENTER	<CENTER>	</CENTER>	Texto Centrado
HR	<HR>		Línea horizontal
P	<P>		Nuevo Párrafo

**Tabla 1.1:** Etiquetas de script primero.html.

## 4. Estructura de un documento HTML

Un documento HTML tiene una estructura formada por dos entidades: la cabecera y el cuerpo. La cabecera contiene el texto que aparece en la parte superior de la ventana y otra serie de elementos que no tienen representación en las páginas WEB pero que sirve para proporcionar información a los buscadores, servidores, etc. La cabecera comienza con la etiqueta <HEAD> y termina con la etiqueta </HEAD>. La parte más importante de este bloque y que no debería de faltar nunca en la frase formada por la etiqueta <TITLE> que determina el nombre de nuestra página, mostrándose este en la parte superior de la ventana.

```
<HEAD>  
  <TITLE>Mi primera pagina</TITLE>  
</HEAD>
```

El cuerpo del documento es el lugar donde los navegadores (Browsers) extraerán la información que presentan por pantalla. Es por tanto donde el programador de páginas WEB define la forma visual de mostrar el contenido. Todo elemento que se encuentre entre las etiquetas <BODY> y </BODY> corresponde a la parte del cuerpo. En esta parte no existe ninguna etiqueta obligatorio, sino todo lo contrario, todo depende del programador y del resultado que se desee.

Para empezar a ver como se utiliza el cuerpo de una página WEB, mostraremos las etiquetas utilizadas en el script del ejemplo.

La primera línea que se puede observar después de la etiqueta de inicio del cuerpo es la siguiente:

```
<CENTER><H1>Mi Primera pagina</H1></CENTER>
```

Esta línea esta formada por varias etiquetas anidadas de forma que contenido de dentro aplica ambas etiquetas.

En primer lugar la etiqueta <H1> define el tamaño del texto que se va a utilizar para la frase “Mi Primera pagina”. A continuación se aplica la etiqueta

<CENTER> que define que todo lo que se encuentra dentro de su radio de acción aparece centrado en la pantalla.

A continuación aplicamos un separador de línea usando la etiqueta <HR>. Este separador muestra una línea horizontal gráfica en sobre toda la pantalla.

Después se utiliza la etiqueta de nuevo párrafo <P>, finalizando con el texto “Este es el primer script de HTML. BIENVENIDO AL CURSO”.

## 5. Formato de elementos HTML

HTML permite definir la forma en que el texto se va a presentar en nuestra página. Al igual que un editor de texto, como puede ser el Word, HTML permite definir el tamaño, el color, el párrafo, la justificación entre otros parámetros de definición de estilo.

### 5.1. Formato de párrafo

Al introducir un texto en un procesador de textos, es importante presentar la información de forma ordenada. Para ello, se necesita algún tipo de herramienta que nos proporcione formas de formatear el texto. HTML proporciona un conjunto de etiquetas que permiten modificar el aspecto en que se va a presentar la información introducida. En la tabla 1.2 se muestran las principales etiquetas que permiten definir el formato del texto que engloban.

Etiqueta	Uso
<P> <CENTER> ... </CENTER> Yo estoy centrado <PRE WIDTH=x> ... </PRE>	Introduce un salto de párrafo. Centra el texto que encierra la etiqueta.  Representa el texto encerrado en ella con un tipo de letra de paso fijo. Muy útil a la hora de representar código fuente. El parámetro WIDTH especifica el número máximo de caracteres en una línea.
<DIV ALIGN=x> ... </DIV>	Permite justificar el texto del párrafo a la izquierda (ALIGN=LEFT), derecha (RIGHT), al centro (CENTER) o a ambos márgenes (JUSTIFY)
<ADDRESS> ... </ADDRESS>	<i>Para escribir direcciones (de esas donde vive la gente, no electrónicas).</i>
<BLOCKQUOTE> ... </BLOCKQUOTE>	Para citar un texto ajeno. Se suele implementar dejando márgenes tanto a izquierda como a derecha, razón por la que se usa habitualmente.

Tabla 1.2: Etiquetas de formato de texto.

### 5.2. Formato lógico

HTML define seis tipos de etiquetas que definen el tamaño fijo de texto. Estas etiquetas llamadas H siguen la estructura de la tabla 1.3.

Este tipo de etiqueta se utiliza cuando queremos resaltar texto frente al resto, pero no queremos definir el tamaño usando la etiqueta <FONT>.

Etiqueta	Resultado
<code>&lt;H1&gt; ... &lt;/H1&gt;</code>	<b>Cabecera de nivel 1</b>
<code>&lt;H2&gt; ... &lt;/H2&gt;</code>	<b>Cabecera de nivel 2</b>
<code>&lt;H3&gt; ... &lt;/H3&gt;</code>	<b>Cabecera de nivel 3</b>
<code>&lt;H4&gt; ... &lt;/H4&gt;</code>	<b>Cabecera de nivel 4</b>
<code>&lt;H5&gt; ... &lt;/H5&gt;</code>	<b>Cabecera de nivel 5</b>
<code>&lt;H6&gt; ... &lt;/H6&gt;</code>	<b>Cabecera de nivel 6</b>

Tabla 1.3: Etiquetas de formato lógico.

Estas etiquetas se pueden definir como de formato lógico . No resulta recomendable utilizarlas para aumentar o disminuir el tamaño del tipo de letra, ya que cada navegador los muestra de manera diferente. Se usan para dividir correctamente en secciones de la página, tal y como se hace en un documento de texto normal.

### 5.3. Estilo de fuente

HTML permite especificar el tipo de estilo fuente que deseamos utilizar en el texto que se va a presentar. Para ello utiliza las etiquetas que se muestran en la tabla 1.4.

Etiqueta	Utilidad	Resultado
<code>&lt;B&gt; ... &lt;/B&gt;</code>	Pone el texto en negrita.	<b>Soy un texto negro como el tizón</b>
<code>&lt;I&gt; ... &lt;/I&gt;</code>	Representa el texto en cursiva.	<i>Estoy ladeado</i>
<code>&lt;U&gt; ... &lt;/U&gt;</code>	Para subrayar algo.	<u>Como soy muy importante estoy subrayado</u>
<code>&lt;S&gt; ... &lt;/S&gt;</code>	Para tachar.	<del>A mí, en cambio, nadie me quiere</del>
<code>&lt;img src="graficos/32.gif" alt="HTML 3.2" /&gt;</code>	Permite representar el texto en un tipo de letra de paso fijo.	No soy variable
<code>&lt;SUP&gt; ... &lt;/SUP&gt;</code>	Letra superíndice.	$E=mc^2$
<code>&lt;SUB&gt; ... &lt;/SUB&gt;</code>	Letra subíndice.	$a_{i,j}=b_{i,j}+1$
<code>&lt;BIG&gt; ... &lt;/BIG&gt;</code>	Incrementa el tamaño del tipo de letra.	<b>Soy GRANDE</b>
<code>&lt;SMALL&gt; ... &lt;/SMALL&gt;</code>	Disminuye el tamaño del tipo de letra.	Creí ver un lindo gatito
<code>&lt;BLINK&gt; ... &lt;/BLINK&gt;</code>	Hace parpadear el texto. Resulta algo irritante.	¿Molesto?

Tabla 1.4: Estilo de fuente

## 5.4. Formatos prefijados

HTML define un conjunto de etiquetas que definen estilos de texto con un formato específico. Estos formatos tienen definido el tipo de letra, el estilo de la fuente, el tamaño, etc. El problema de este tipo de etiquetas es que no permite controlar el resultado que va a aparecer en la pantalla, más bien deja el control completo al navegador que es quien interpreta la etiqueta y decide la forma de presentación.

Etiqueta	Utilidad	Resultado
<CITE> ... </CITE> <CODE> ... </CODE>	Para citar un texto ajeno. Para escribir código fuente.	<i>Esta frase no es mía</i> <code>int x=0;</code>
<STRONG> ... </STRONG> <EM> ... </EM>	La cosa es importante.  Para dar énfasis.	Hay cosas <b>importantes</b> .  Hay que poner <i>énfasis</i> en algunas cosas.
<KBD> ... </KBD>	Texto tecleado por el usuario.	El usuario debe teclear <code>Multivac</code> es el mejor.
<VAR> ... </VAR>	Representar variables de un código.	La variable <i>x</i> , definida anteriormente...
<SAMP> ... </SAMP>	Para representar una serie de caracteres literalmente.	Estoy en un <code>literal</code>
<ABBR> ... </ABBR>	Abreviaturas.	La WWW usa el protocolo <code>http</code>

Tabla 1.5: Etiquetas de formato prefijado.

## 5.5. Etiquetas de inicio

Como en todo lenguaje, siempre hay excepciones que confirman la regla, y este es el caso de las etiquetas que se muestran en la tabla 1.5. Estas etiquetas no se introducen en ninguna clase en concreto porque, además de ser etiquetas de inicio no se aplican a ningún bloque, sino que ejecutan operaciones directas.

Etiqueta	Utilidad	Resultado
<HR>	Inserta una barra horizontal.	<hr/>
 	Salto de línea.	Hay un antes y un

después de saltar a otra  
línea

<!-- ... -->

Comentarios.

Esto se escribe y

**Tabla 1.5:** Etiquetas de formato prefijado.

## 5.6. Caracteres especiales

El lenguaje de Scripts HTML no permite la inclusión directa de caracteres especiales, como pueden ser los acentos, las eñes, los símbolos de interrogación o de exclamación entre otros. Esto es debido al Standard ASCII no es el mismo para todos los idiomas, por lo que se ha elegido únicamente del Standard ASCII los 127 caracteres de la tabla que son comunes en todos los idiomas. El problema surge con los caracteres especiales se implementan sobre los 127 caracteres restantes de la tabla.

HTML soluciona este problema utilizando caracteres extendidos que permiten especificarle al Browser la inclusión de éstos.

El HTML 2.0 eligió como tabla estándar la ISO-Latin-1, que comparte con la ASCII los 127 caracteres e incluye unos cuantos más hasta el número 255.

## 5.7. Caracteres extendidos en HTML

Para poder introducir los caracteres extendidos en una página WEB se deben de encerrar entre los símbolos “&#” y “;”. De esta forma cualquier número que introduzcamos entre estos símbolos muestra su equivalente ASCII.

Por ejemplo si introducimos &#189;, el resultado interpretado por el browser, será ½.

Además de poder utilizar el sistema de impresión de caracteres extendidos de la tabla ASCII, HTML proporciona un conjunto reducido de caracteres extendidos predefinidos. Este tipo de caracteres permite recordar con más facilidad el símbolo que representan.

Por ejemplo para mostrar el carácter ASCII acento en la a (á), se puede realizar tanto con &#225; como con &aacute; donde la primera a representa el

símbolo al que se la va a aplicar el acento. En la tabla 1.6 se muestran los caracteres extendidos predefinidos más comunes en el lenguaje castellano.

<b>Código</b>	<b>Resultado</b>
&acute;, &Aacute;, &eacute;, &Eacute;,...	á, Á, é, É, í, Í, ó, Ó, ú y Ú
ñ y ñ	ñ y ñ
&iquest;	¿
&iexcl;	¡
&ordm;	o
&ordf;	a
&trade; o &#153;	™ o ™
&copy;	©
&reg;	®
&nbsp;	(espacio en blanco que no puede ser usado para saltar de línea)

Tabla 1.6: Caracteres extendidos predefinidos.

## 5.8. Caracteres de control

En el HTML existen cuatro caracteres de control, que se usan para formar etiquetas, establecer parámetros, etc.. Estos caracteres no deben de utilizarse directamente en las páginas WEB, sino que se deben de utilizar equivalentes que el Browser pueda interpretar. En la tabla 1.7 se pueden ver los diferentes tipos de caracteres de control que se utilizan en HTML. Con estos caracteres podemos evitar la ambigüedad que existe entre por ejemplo un mayor que de principio de etiqueta y un mayor que de un texto que representa un inecuación.

<b>Código</b>	<b>Resultado</b>
&lt;	<
&gt;	>
&amp;	&
&quot;	"

Tabla 1.7: Caracteres de control

## 5.9. Formateo fino

Como hemos visto, HTML proporciona un conjunto que etiquetas que permiten establecer el tamaño del texto. El problema radica en que estos tamaños y el tipo de letra utilizados se deja a elección del navegador que es quien lo decide. Para evitar esta falta de control, HTML proporciona una

etiqueta con mayor control en el formato de texto. La etiqueta <FONT> con todos sus parámetros permite definir con total exactitud el resultado que se desee para la presentación del texto. Cambiar el tipo de letra, el tamaño o incluso el color se realiza fácilmente con la nueva etiqueta. La sintaxis que se tiene que seguir para establecer estilos en un texto es el siguiente:

```
<FONT SIZE=2 COLOR="red FACE="Arial">Texto</FONT>
```

### 5.9.1. Tamaño del texto

El parámetro utilizado para poder definir el tamaño de un texto es SIZE. Este parámetro permite definir tanto tamaños absolutos como relativos al tamaño actual.

Para especificar tamaños absolutos, sólo necesitamos especificar el tamaño en pixels que se quiere establecer. Por ejemplo si especificamos el texto “hola” con el tamaño 7, introducimos en HTML la siguiente línea:

```
<FONT SIZE=7>Hola</FONT>
```

Partiendo de la línea anterior, podemos introducir un texto de tamaño 2 pixel más pequeño, para ello se introduce la siguiente línea:

```
<FONT SIZE=-2>Hola</FONT>
```

El parámetro utilizado para indicar el tamaño es SIZE. Puede utilizarse para indicar tamaños absolutos:

SIZE=1 SIZE=2 SIZE=3 SIZE=4 SIZE=5 SIZE=6 SIZE=7

### 5.9.2. Tipo de letra

Para poder especificar el tipo de fuente que se va a utilizar en el texto que encierra las etiquetas FONT tenemos un parámetro FACE. Este parámetro permite indicar a HTML el tipo de fuente que vamos a utilizar de forma que no sea el propio navegador el que seleccione un tipo de fuente por defecto.

Al no conocer los tipos de fuente que tiene instalados el ordenador donde se va a ejecutar el navegador, FACE permite que se le introduzcan más de un nombre de fuente, de forma que si el primero no existe, seleccione el segundo y así sucesivamente. En el caso de no encontrar ninguno de tipos de letra, el navegador selecciona el formato por defecto. El siguiente ejemplo tratará de presentar el texto en el tipo de letra Helvetica. Si no lo consigue lo interta con Arial y luego con Timer New Roman. Sólo en el caso de no encontrar ninguno de los formatos, utiliza el tipo por defecto.

```
<FONT FACE="Helvetica,Arial,Times">Texto a mostrar</FONT>
```

### 5.9.3. Color del texto

Además de poder seleccionar el tipo de letra y su tamaño, HTML permite seleccionar el color con el que va a presentar el texto de forma que no utilice el color por defecto.

Para poder presentar el texto en un color diferente del que el navegador utiliza por defecto, HTML proporciona la etiqueta COLOR. Esta etiqueta permite indicar el color tanto en formato texto ( a través de su nombre) como en formato RGB. En la tabla se pueden ver los nombres de los colores que son reconocidos por la etiqueta.

Black	Silver	Gray	White	Maroon	Red	Purple	Fuchsia
Green	Lime	Olive	Yellow	Navy	Blue	Teal	Aqua

La sintaxis que se debe de seguir para especificar un color en un texto es la siguiente:

```
<FONT COLOR="red">Texto en color rojo</FONT>
```

## 6. Enlaces

Las siglas HTML significan HyperText Markup Language, lo que significa que estamos ante un lenguaje de hipertexto. Este tipo de lenguaje se caracteriza por un elemento básico que es el hiperenlace que permite conectar zonas del texto con otros documentos.

Un enlace es una zona de texto o gráficos de una página que permite acceder a otras páginas o zonas dentro de nuestra página con sólo pulsar el raton sobre ésta. Siendo HTML el lenguaje de Internet, la diferencia que posee con respecto a otros tipos de hipertexto es que el documento al que se va a acceder puede estar en cualquier lugar de la red de Internet.

### 6.1. La etiqueta <A>

HTML proporciona una etiqueta que nos permite transformar un objeto, ya sea texto o gráficos en un enlace. Esta etiqueta esta formada por la etiqueta de inicio <A> y la de cierre </A>. Todo lo que se halle encerrado entre ambas será considerado un enlace. Al incorporar un enlace a un elemento, éste sufre dos modificaciones básicas en el resultado ofrecido por el navegador.

1. El texto encerrado por el enlace, aparece en un color distinto, normalmente azul, mientras que las imágenes aparecen rodeadas por un borde.
2. Al pulsar sobre el enlace, se carga la página a la que apunta.

Para que el enlace sirva para algo debemos especificarle una dirección. Lo haremos de la siguiente manera:

```
<A HREF="hiperenlace">Haz clic</A>
```

La dirección de hiperenlace está en formato URL (Uniform Resource Locator), lo cual nos indica tanto la dirección de Intenet como el servicio que se demanda desde el enlace. El formato de una dirección URL es el siguiente:

`servicio://máquina:puerto/ruta/fichero@usuario`

donde el servicio puede ser cualquiera de los mostrados en la tabla 1.8.

Servicio	Uso
<b>http</b>	Es el servicio invocado para transmitir páginas web.
<b>https</b>	Es una innovación sobre el anterior, que nos permite acceder a servidores (generalmente comerciales) que nos ofrecen el uso de técnicas de encriptación para proteger los datos que se intercambian con terceras personas
<b>ftp</b>	Permite transmitir ficheros desde servidores de <i>ftp</i> anónimo.
<b>mailto</b>	Para poder mandar un mensaje de correo electrónico
<b>news</b>	Para poder acceder a foros de discusión.

Tabla 1.7: Caracteres de control

La dirección de la máquina puede ser tanto una dirección IP formada por una serie de cuatro números entre 0 y 255 (*123.3.5.65*) o bien algo más fácil de recordar como es una dirección DNS formada por una serie de palabras separadas por puntos (*www.microsoft.com*). El puerto generalmente no se indica, ya que cada servicio tiene definido uno.

Si lo que deseamos es acceder a un enlace que se encuentra situado en la misma dirección desde la que se va a ejecutar el enlace, se puede utilizar el formato corto de URL que sigue el siguiente formato:

`RutaRelativa/elemento_de_enlace`

En la ruta relativa se pueden utilizar los dos puntos y barra (*../*) para acceder al directorio donde se encuentra la página desde la que vamos a hacer el enlace.

## 6.2. Anclas

Además de poder acceder mediante los enlaces a otras páginas, también se puede acceder a cualquier posición del documento. Para poder acceder a una posición en la página primero hay que definir los puntos en las que se puede posicionar. Para ello HTML proporciona el parámetro NAME que permite definir anclas en el texto la página. La sintaxis para definir un ancla es la siguiente:

```
<A NAME="ancla">
```

Una vez definida la posición del ancla, hay que colocar el enlace de forma que al pulsar sobre el éste, se posicione el cursor sobre el anclaje definido. La sintaxis para acceder a un ancla es la siguiente:

```
<A HREF="#ancla">Vamos a donde antes</A>
```

De esta manera podremos acceder a la posición ancla definida anteriormente.

Además de podernos colocar en posiciones definidas en nuestra página, podemos colocarnos en posiciones de otras páginas. Para ello basta con indicar además del ancla al que queremos acceder, la dirección URL de la página a que queremos acceder. La sintaxis quedaría de la siguiente forma:

```
<A HREF="enlaces.html#ancla">Otra vez</A>
```

## 7. Listas

Una lista no es más una secuencia de texto ordenada por el mismo nivel de sangrado. Además de tener el mismo sangrado, también tiene un símbolo, por ejemplo un punto a la izquierda de cada uno de los párrafos que forman la lista.

HTML proporciona varios tipos de listas que permiten agrupar elementos con el mismo criterio en la misma lista. El formato que se tiene que seguir para crear una lista de texto es el siguiente:

```
<tipo_lista>
  <LI>Primer elemento
  <LI>Segundo elemento
</tipo_lista>
```

donde:

tipo\_lista hace referencia la forma en que se van a ordenar los elementos, pudiendo ser de los tipos: DIR, DL, MENU, OL y UL.

LI: define un elemento de la lista. Esta etiqueta únicamente tiene elemento de inicio de forma que todo lo que este entre la etiqueta <LI> y la siguiente etiqueta <LI> o final del lista se considera elemento de la lista. A continuación se muestran los diferentes tipos de lista que se pueden utilizar en HTML.

### 7.1. Listas desordenadas

Las listas desordenadas son aquellas que no tienen ningún elemento de orden, es decir no siguen ninguna secuencia ni de números ni de letras. Únicamente se utiliza un punto como común denominador de los elementos de la lista. Para definir una lista como desordenada esta tiene que utilizar la siguiente sintaxis:

```
<UL>
  <LI>Primer elemento
  <LI>Segundo elemento
  .....
  <LI>último elemento
```

</UL>

La etiqueta <UL> admite parámetros que permiten definir diferentes tipos de salida de texto. En la figura

<b>Parámetro</b>	<b>Utilidad</b>
COMPACT	Indica al navegador que debe representar la lista de la manera más compacta posible.
TYPE="disc", "circle", "square"	Indica al navegador el dibujo que precederá a cada elemento de la lista. Para mayor flexibilidad se admite también como parámetro de <LI>.

**Tabla 1.7:** Caracteres de control

Los tipos de lista <DIR> y <MENU> son sinónimos de <UL> por lo que se pueden utilizar de la misma forma.

## 7.2. Listas ordenadas

Las lista ordenadas permiten tener como elemento de orden un número o una letra que se vaya incrementando a medida que colocamos más elementos en la lista.

Para definir una lista como ordenada se tiene que utilizar la etiqueta <OL> que permite presentar los elementos de menor a mayor. La sintaxis que se debe de seguir para escribir una lista ordenada es el siguiente:

```
<OL>
  <LI>Primer elemento
  <LI>Segundo elemento
</OL>
```

Al igual que las listas desordenadas, las listas ordenadas tienen parámetros que permiten personalizarla. En la figura 1.8 se muestran los parámetros de las listas ordenadas.

<b>Parámetro</b>	<b>Utilidad</b>
COMPACT	Indica al navegador que debe representar la lista de la manera más compacta posible.
TYPE="1", "a", "A", "i", "I"	Indica al navegador el tipo de numeración que precederá a cada elemento de la lista. Para mayor flexibilidad se admite también como parámetro de <LI>.
3. START="num"	4. Indica al navegador el número

por el que se empezará a contar los elementos de la lista.

VALUE="num"

Atributo de <LI>, actúa como START pero a partir de un elemento predeterminado.

**Tabla 1.8:** Listas ordenadas.

### 7.3. Listas de definiciones

Una lista de definiciones permite tener dos niveles de sangrado en nuestra lista de forma que se puede hacer niveles y subniveles. A diferencia de las listas ordenadas y desordenadas, las listas de definiciones utilizan otro tipo de etiqueta y de estructura. Para definir una lista de definiciones se utilizan las etiquetas <DT> y <DD> donde la primera define el primer nivel y la segunda define el segundo nivel. La sintaxis para crear una lista de definiciones es la siguiente:

```
<DL>
  <DT>Primer elemento<DD>Es un elemento muy bonito.
  <DT>Segundo elemento<DD>Este, en cambio, es peor.
</DL>
```

La etiqueta <DL> sólo admite como parámetro COMPACT, que tiene el mismo comportamiento que con los otros dos tipos de lista anteriores.

## 8. Imágenes

Además de poder incluir texto en las páginas WEB , el lenguaje de script HTML permite introducir imágenes y hacer que éstas sean enlaces a otras páginas.

### 8.1. La etiqueta <IMG>

Para poder incluir imágenes del tipo JPEG, GIF, el lenguaje HTML proporciona la etiqueta <IMG> que indica al navegador que se va a pegar una imagen en nuestra página.

La sintaxis que sigue la etiqueta <IMG> es la siguiente:

```
<IMG SRC="fichero_grafico" ALT="descripcion">
```

El parámetro SRC especifica el nombre del fichero que contiene el gráfico. Los formatos estándar en la red son el GIF y el JPG. Las últimas versiones de Netscape y Explorer aceptan también el formato PNG.

El parámetro ALT especifica el texto que se muestra en lugar del gráfico en aquellos navegadores que no sean capaces de mostrarlos o en el supuesto de que el usuario los haya desactivado. Algunos navegadores lo muestran cuando se pasa el ratón por encima de la imagen. Es por eso que, aunque algunos usuarios no lo lleguen a ver nunca, conviene siempre ponerlo. De hecho, el estándar HTML 4.0 obliga a hacerlo.

Existen dos atributos que, aunque opcionales, conviene indicar siempre: la altura y la anchura del gráfico en pixels. Estos dos parámetros determinan el tamaño de la imagen que va a aparecer en la pantalla. En el caso de omitirlos, el tamaño de la imagen viene definido por el propio fichero. Si utilizamos los parámetros, HTML los utiliza para determinar el tamaño de la imagen, haciendo que ésta se adapte a los mismos.

### 8.2. Enlaces gráficos

Este apartado únicamente muestra lo que ya es obvio. Si queremos hacer que un gráfico tenga la característica de poder enlazar con otras páginas o ir a otro

anclaje, únicamente debemos de definir una imagen y posteriormente crearle un enlace.

La sintaxis para crear una imagen con enlace en una página Web es el siguiente:

```
<A HREF="http://www.google.com">  
<IMG SRC="imagen.jpg" ALT="Enlace a google" WIDTH=88 HEIGHT=31 BORDER  
=0>  
</A>
```

El parámetro BORDER define si queremos tener un borde alrededor de la imagen y en caso de quererlo su tamaño, de forma que si lo ponemos a cero no aparece.

### 8.3. Alineación respecto al texto

Para poder colocar conjuntamente texto y gráficos, el HTML proporciona, por medio del parámetro ALIGN, las siguientes maneras de alinear una imagen respecto del texto que la acompaña. En la tabla 1.9 se muestran los diferentes valores que puede tomar el parámetro ALIGN.

<b>Valor de ALIGN</b>	<b>Utilidad</b>
TOP	Coloca el punto más alto de la imagen coincidiendo con más alto de la línea de texto actual.
MIDDLE	Alinea el punto medio (en altura) de la imagen con la base del texto.
BOTTOM(Por defecto)	Alinea el punto más bajo de la imagen con la base del texto.
LEFT	Coloca la imagen a la izquierda del texto.
RIGHT	Coloca la imagen a la derecha del texto.

**Tabla 1.9:** Listas ordenadas

## 9. Los formularios

Los formularios son la herramienta que proporciona HTML para hacer que las páginas sean interactivas. Además de poder presentar información en una página, HTML proporciona un conjunto de etiquetas que definen varios tipos de elementos que se utilizan para crear un formulario. Las cajas de diálogo, las cajas de verificación, los botones de selección, etc son elementos que forman parte de un formulario. A continuación se verán los diferentes elementos y la sintaxis que lleva cada uno de ellos.

### 9.1. La etiqueta <FORM>

Todos los formularios tienen una parte común que es la definición del área del formulario. El área del formulario es la zona que comprende todos los elementos de éste. Para definir un área de formulario se utiliza la etiqueta <FORM> la cual tiene tanto etiqueta de inicio como de fin.

Esta etiqueta define dos parámetros obligatorios y un conjunto de opcionales como se puede ver en la tabla.

<b>Parámetro</b>	<b>Significado</b>
ACTION	Acción a realizar por el formulario
METHOD	Método a utilizar en el envío
NAME	Nombre del formulario
ENCTYPE	Forma de cifrar la información enviada

**Tabla 1.9:** Listas ordenadas

El parámetro ACTION define el script del servidor que va a tratar los datos enviados por el formulario.

El parámetro METHOD Indica el método según el que se van a transferir las variables del formulario. POST envía los datos, normalmente al programa CGI definido en action o por correo si en action hemos utilizado mailto. El método GET añade los argumentos del formulario al URL recogido en ACTION.

La sintaxis que se debe de seguir para presentar un formulario es la siguiente:

```
<form action="script.php" method="post" enctype="text/plain"
name="Form1">
...
...elementos del formulario
...
</form>
```

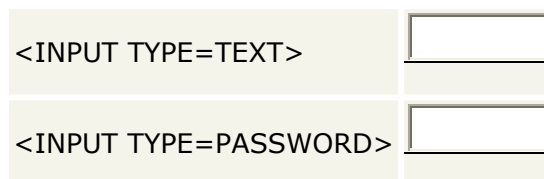
A continuación se muestran cada uno de los elementos que se pueden introducir en un formulario.

## 9.2. Cajas de texto

Las cajas de texto son objetos gráficos que permiten introducir texto que luego es enviado por el formulario al script que lo va a procesar.

Existen tres tipos de cajas de texto dos de la cuales tienen el mismo aspecto y es su funcionalidad lo que las diferencia.

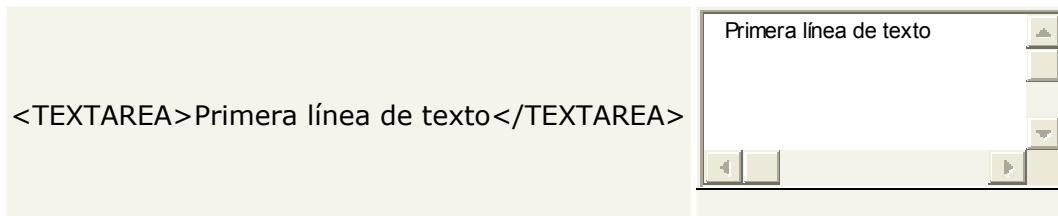
Las cajas de texto sencillas o de una sola línea se definen con la etiqueta INPUT. Esta etiqueta que no tiene cierre proporciona varios parámetros entre los cuales destaca el parámetro TYPE que indica el tipo de caja de texto que se va a utilizar. Si TYPE es TEXT la caja permite visualizar lo que se escribe, por el contrario si TYPE es PASSWORD la caja no visualiza lo que se escribe sino que cada carácter es sustituido por un asterisco de forma que sea imposible visualizar el texto tecleado. En la tabla se puede observar el resultado de crear una caja de texto con el tipo texto o password.



La etiqueta INPUT proporciona tres parámetros que permiten definir ciertos aspectos del comportamiento como se puede ver en la tabla.

Parámetro	Utilidad
SIZE	Tamaño de la caja de texto.
MAXLENGTH	Número máximo de caracteres que puede introducir el usuario.
VALUE	Texto por defecto que contendrá la caja.

En el caso de necesitar una caja en la que tengamos que introducir más de una línea, HTML proporciona la etiqueta <TEXTAREA> asociada al un objeto gráfico que permite introducir varias líneas, conteniendo barras de desplazamiento tanto horizontales como verticales. Como se puede ver en la figura.



Los parámetros de TEXTAREA son los siguientes:

<b>Parámetro</b>	<b>Utilidad</b>
ROWS	Filas que ocupará la caja de texto.
COLS	Columnas que ocupará la caja de texto.

### 9.3. Etiqueta <INPUT>

Las cajas de texto permiten introducir información que el script introducido en el campo ACTION procesa. No siempre se desea que sea el usuario quien introduzca el texto, sino que a veces es más interesante que el usuario sólo tenga que seleccionar entre un conjunto de opciones.

Para poder crear un grupo de selección, de forma que el usuario tenga que elegir un elemento entre un conjunto de opciones, HTML proporciona la etiqueta INPUT. Esta etiqueta depende de la propiedad TYPE , que le indica al navegador el tipo de elemento a crear.

<b>TYPE</b>	<b>Funcionalidad</b>
RADIO	Presenta un elemento de selección única
CHECKBOX	Presenta un elemento de selección múltiple
HIDDEN	Elemento oculto para personalización de páginas
SUBMIT	Presenta un botón de lanzamiento de formulario
RESET	Presenta un botón de reseteo de formulario

### 9.3.1. RadioButtons


Cuando se desea que un usuario elija entre un conjunto de opciones una solamente, es necesario utilizar el radiobutton. Este objeto presenta un círculo y es el programador el que debe de crear tantos elementos como opciones desee.

La sintaxis del elemento de opción radiobutton es la siguiente:

```
<INPUT NAME="Nombre" TYPE=RADIO VALUE="valor">
```

Para que se pueda decidir entre varias opciones, la única condición que se tiene que cumplir es que todos los radiobutton tengan el mismo nombre. En el ejemplo siguiente se puede observar como se crea una selección de tres elementos.

```
3<INPUT NAME="Respuesta" TYPE=RADIO VALUE="mal"><BR>  
4<INPUT NAME="Respuesta" TYPE=RADIO VALUE="bien"><BR>  
5<INPUT NAME="Respuesta" TYPE=RADIO VALUE="mal"><BR>
```



Los parámetros que acepta este elementos son los siguientes.

<b>Parámetro</b>	<b>Utilidad</b>
NAME	Nombre del RadioButton
VALUE	Este es el valor que asignará a la variable.
CHECKED	Si lo indicamos en una de las opciones esta será la que esté activada por defecto.

### 9.3.2. CheckBox

Cuando lo que se desea es que el usuario pueda marcar sobre un conjunto de opciones tantas como opciones existen, es necesario utilizar el tipo CHECKBOX.

La sintaxis que se debe de utilizar para poder utilizar este elemento es la siguiente:

```
<INPUT NAME="Elemento" TYPE=CHECKBOX>
```

Los parámetros que soporta este elemento son los siguientes:

<b>Parámetro</b>	<b>Utilidad</b>
NAME	Nombre del CheckBox
VALUE	Este es el valor que asignará a la variable.
CHECKED	Si lo indicamos en una de las opciones esta será la que esté activada por defecto.

Si queremos que el control esté activado por defecto le añadiremos el parámetro CHECKED. El formulario asignará a la variable NAME el valor on u off.

### 9.3.3. Los botones

Los botones permiten realizar acciones sobre todo el formulario. Dependiendo del parámetro TYPE el botón se comportara como un botón de envío o un botón de reseteo. La sintaxis de creación es la misma para ambos botones diferenciándose en el parámetro TYPE.

SI TYPE es de tipo SUMBIT se crea un botón de ejecución que se encarga de enviar todos los parámetros del formulario al script que contiene el parámetro ACTION del formulario.

Si el tipo es RESET se crea un botón de reseteo que al ser pulsado carga la página con los parámetros por defecto.

La sintaxis para crear un botón del tipo SUBMIT se debe introducir la siguiente línea:

```
<INPUT TYPE=SUBMIT>
```

Para el caso de un botón RESET se debe introducir la siguiente línea:

```
<INPUT TYPE=SUBMIT>
```

Además de estos elementos de botón, también existe un elemento que no presenta ningún tipo de objeto en pantalla. Únicamente sirve para ocultar información en una página, que es enviada al script de ACTION. Aunque no presenta ningún objeto se ha introducido en este apartado porque la sintaxis

mantiene la misma estructura que el resto de los objetos del tipo INPUT, siendo esta la siguiente:

```
<INPUT TYPE=HIDDEN NAME="Nombre" VALUE="valor">
```

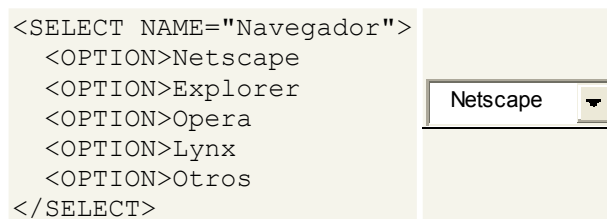
### 9.3.4. Listas desplegables

Una lista desplegable es una caja de texto simple con una flecha apuntando hacia abajo a su derecha. Este elemento permite seleccionar un elemento o varios dependiendo de la configuración que tenga. Este elemento a diferencia del resto visto hasta el momento utiliza una sintaxis diferente.

La nueva sintaxis para crear una lista desplegable esta formada por dos etiquetas.

La etiqueta SELECT define el comienzo de una lista desplegable. Esta etiqueta tiene comienzo y fin.

La etiqueta OPTION define cada elemento de la lista. Esta etiqueta debe de estar dentro del radio de acción de la etiqueta SELECT.



Los parámetros que admite SELECT son las siguientes:

Parámetro	Utilidad
NAME	Nombre del objeto lista desplegable
SIZE	El número de opciones que podremos ver. Si es mayor que 1 veremos una lista de selección y, si no, veremos una lista desplegable.
MULTIPLE	Si lo indicamos podremos elegir más de una opción.

Por otro lado los parámetros que acepta la etiqueta OPTION son los siguiente:

Parámetro	Utilidad
VALUE	Este es el valor que asignará a la variable.
SELECTED	Si lo indicamos en una de las opciones esta será la seleccionada por defecto.

Podemos cambiar el texto que el navegador pone por defecto en esos botones utilizando el parámetro VALUE.

## 10. Tablas

Las tablas son el elemento que proporciona HTML para organizar la información. Una tabla puede contener texto, imágenes, formularios, etc.

### 10.1. La etiqueta <TABLE>

Las tablas actúan como elementos de construcción para las páginas Web, permitiendo alinear toda la información.

Una tabla está constituida por filas y columnas de forma que crean una matriz de trabajo.

Para definir una tabla HTML proporciona la etiqueta TABLE con un conjunto de parámetros como se puede ver en la tabla. Una vez definida la tabla se necesita definir la fila para lo cual se proporciona la etiqueta TR y finalmente se necesita definir la columna para lo cual se proporciona la etiqueta TD. De esta forma la sintaxis de creación de una tabla es la siguiente:

```
<TABLE>
  <CAPTION>
    Título de la tabla
  </CAPTION>
  <TR>
    <TD></TD>
    .
    .
    <TD></TD>
  </TR>
  .
  .
  .
</TABLE>
```

En el siguiente ejemplo se puede observar como se crea una tabla de dos filas por tres columnas.

```
<TABLE>
  <TR>
    <TD>1, 1</TD>
    <TD>1, 2</TD>
    <TD>1, 3</TD>
  </TR>
  <TR>
    <TD>2, 1</TD>
```

1, 1	1, 2	1, 3
2, 1	2, 2	2, 3

```

        <TD>2, 2</TD>
        <TD>2, 3</TD>
    </TR>
</TABLE>

```

La etiqueta <TABLE> permite un conjunto de parámetros que permiten modelar el aspecto resultante de la tabla.

BORDER	Especifica el grosor del borde que se muestra alrededor de las celdas. Por defecto es cero, lo que significa que no muestra ningún borde.
CELLSPACING	Define el número de pixels que separarán las celdas.
CELLPADDING	Especifica el número de pixels que habrá entre el borde de una celda y su contenido.
WIDTH	Especifica la anchura de la tabla. Puede estar tanto en pixels como en porcentaje de la anchura total disponible para él (pondremos "100%" si queremos que ocupe todo el ancho de la ventana del navegador).
ALIGN	Alinea la tabla a izquierda (LEFT), derecha (RIGHT) o centro (CENTER).

## 10.2. Definir título

Para definir el nombre que va a aparecer en la parte superior de la tabla, HTML define la etiqueta CAPTION que permite especificarlo. Esta etiqueta sólo contiene un único parámetro ALING que permite especificar el lugar donde va a aparecer el título pudiendo ser TOP o BOTTON si queremos que aparezca en la parte superior o inferior respectivamente. Por defecto está a TOP.

## 10.3. Definir las filas

Además de poder modelar la tabla completa utilizando los parámetros que HTML proporciona, también se puede modificar el aspecto a nivel de fila de forma que cada una de éstas pueda ser modelada de forma individual.

Los parámetros que se pueden introducir en la etiqueta <TR> son los siguientes:

- ALIGN Alinea el contenido de las celdas de la fila horizontalmente a izquierda (LEFT), derecha (RIGHT) o centro (CENTER).
- VALIGN Alinea el contenido de las celdas de la fila verticalmente arriba (TOP), abajo (BOTTOM) o centro (MIDDLE).

### 10.4. Definición de celdas

Para terminar, se pueden definir parámetros a nivel de celda, para ello basta con utilizar los parámetros proporcionados para la etiqueta <TD> que se muestran en la tabla.

- ALIGN Alinea el contenido de la celda horizontalmente a izquierda (LEFT), derecha (RIGHT) o centro (CENTER).
- VALIGN Alinea el contenido de la celda verticalmente arriba (TOP), abajo (BOTTOM) o centro (MIDDLE).
- WIDTH Especifica la anchura de la celda. También se puede especificar tanto en pixels como en porcentaje, teniendo en cuenta que, en este último caso, será un porcentaje respecto al ancho total de la tabla (no de la ventana del navegador).
- NOWRAP Impide que, en el interior de la celda, se rompa la línea en un espacio.
- COLSPAN Especifica el número de celdas de la fila situadas a la derecha de la actual que se unen a ésta (incluyendo la celda en que se declara este parámetro). Es por defecto uno. Si se pone igual a cero, se unirán todas las celdas que queden a la derecha.
- ROWSPAN Especifica el número de celdas de la columna situadas debajo de la actual que se unen a ésta.

Los dos últimos parámetros son un tanto especiales ya que permiten agrupar celdas y formar una sola tanto de filas como de columnas. La mejor forma de entenderlo es mediante un ejemplo.

```
<TABLE BORDER=1>
  <TR>
    <TD COLSPAN=2>1,1 y 1,2</TD>
    <TD>1,3</TD>
  </TR>
  <TR>
    <TD ROWSPAN=2>2,1 y 3,1</TD>
    <TD>2,2</TD>
    <TD>2,3</TD>
  </TR>
  <TR>
    <TD>3,2</TD>
    <TD>3,3</TD>
  </TR>
</TABLE>
```

1,1 y 1,2		1,3
2,1 y 3,1	2,2	2,3
	3,2	3,3

El ejemplo esta formado por una matriz de 3x3 a la que se le ha realizado un COLSPAN en la primera fila uniando las dos primeras columnas y un ROWSPAN en la segunda fila en la que se han unido las dos últimas filas.

## 11. Marcos

Un marco o recuadro es una zona independiente de la ventana principal en la que se puede cargar una página Web.

Cuando se carga una página Web, ésta se muestra sobre toda la ventana del browser. HTML permite crear zonas de presentación independientes sobre la ventana principal del navegador.

Los frames o marcos no se crean dentro de ninguna zona de la página HTML sino que se crean en un documento especial donde se describe la forma que va a tener la página, es decir los marcos que va a tener y el tamaño de cada uno de estos.

Para definir una página de creación de marcos se debe de seguir la siguiente sintaxis:

```
<HTML>
<HEAD>
  <TITLE>Página de marcos</TITLE>
</HEAD>
<FRAMESET COLS="20%,80%">
  <FRAME NAME="primero" SRC="primero.html">
  <FRAME NAME="segundo" SRC="segundo.html">
  <NOFRAMES>
    <P>No se pueden crear marcos. Posiblemente el navegador no lo
soporte.</P>
  </NOFRAMES>
</FRAMESET>
</HTML>
```

La principal diferencia que existe en la creación de marcos con respecto a la creación de una página HTML sin éstos es que en vez de utilizar la etiqueta BODY, que sirve normalmente para delimitar lo que se va a ver en la pantalla, se hace uso de la etiqueta FRAMESET.

La etiqueta FRAMESET define el principio de la creación de un marco la cual proporciona un

La cabecera de la página es similar a un documento normal de HTML, donde la etiqueta <BODY> es sustituida por la etiqueta <FRAMESET>. Cada

FRAMESET se divide la ventana actual, ya sea la general o el marco, en varias ventanas definidas por los parametros COLS o por ROWS.

### 11.1. La etiqueta <FRAMESET>

La etiqueta <FRAMESET> permite definir un marco sobre la ventana actual o sobre un marco de forma que se subdivide en varias ventanas.

Como el resto de las etiquetas que proporciona HTML, los marcos tienen un conjunto de parámetros que permiten modificar el comportamiento y el aspecto del marco. En la tabla se pueden observar los principales parámetros que soporta la etiqueta <FRAMESET>

<b>Parámetro</b>	<b>Significado</b>
ROWS	Filas en las que se divide la ventana o frame actual
COLS	Columna en las que se divide la ventana o marco actual.
FRAMEBORDER	Grosor del borde que rodea al marco
FRAMESPACING	Hueco que queda entre dos marcos adyacentes
BORDER	

En general, los navegadores dibujan un borde de separación entre los marcos el cual se puede eliminar de dos maneras:

1. En las etiquetas <FRAME> de cada una de los marcos contiguos al borde a eliminar
2. Incluyendo el parámetro FRAMEBORDER=0 en el <FRAMESET>.

Cuando se elimina ese borde, se puede observar cómo el navegador deja aún un hueco entre marcos el cual se puede eliminar añadiendo los parámetros FRAMESPACING=0 BORDER=0.

Los parámetros COLS y ROWS definen el tamaño y cantidad de ventanas que en que se va a dividir la actual. Para ello se tiene que asignar una lista de tamaños separada por comas. Este lista permite introducir los valores de varias formas:

- **Con porcentajes:** Al igual que con las tablas, podemos definir el tamaño de un marco como un porcentaje del espacio total disponible.
- **Absolutos:** Si ponemos un número entero, el marco correspondiente tendrá el tamaño especificado en pixels.
- **Sobre el espacio sobrante:** El símbolo (\*) indica que queremos todo el espacio sobrante para ese marco. Se poner este símbolo en varios marcos, repartiendo el espacio equitativamente. También se puede repartir el espacio sobrante haciendo un una parte tenga el doble que otra, indicándolo con 2\*

En el siguiente ejemplo se muestra como dividir la ventana principal en cuatro partes de tamaño diferente.

```
<FRAMESET COLS="10%, *, 200, 2*">
```

Supongamos que el ancho total de la ventana son 640 pixels. El primer marco ocupará el 10%, es decir, 64 pixels. El tercero necesita 200, luego nos quedan 476 para los otros dos. Como el cuarto debe tener el doble de espacio que el segundo, tenemos aproximadamente 158 pixels para este último y 316 para el cuarto marco.

Hay que tener cuidado cuando se usen valores absolutos en la definición de marcos; se debe de asegurar tener al menos un marco con un tamaño relativo si se quiere estar seguro del aspecto final de la página.

Una característica importante de la definición de marcos, es que estos se pueden anidar.

```
<FRAMESET COLS="20%,80%">
  <FRAME NAME="indice" SRC="indice.html">
  <FRAMESET ROWS="*,80">
    <FRAME NAME="primero" SRC="primero.html">
    <FRAME NAME="segundo" SRC="segundo.html">
  </FRAMESET>
</FRAMESET>
```

## 11.2. Etiqueta <FRAME>

Esta etiqueta define tan sólo las características de un marco determinado, no de un conjunto de ellos. Los parámetros que se pueden aplicar a un marco son los siguientes:

NAME	Asigna un nombre a un marco para que después podamos referirnos a él.
SRC	Indica la dirección del documento HTML que ocupará el marco.
SCROLLING	Decide si se colocan o no barras de desplazamiento al marco para que podamos movernos por su contenido. Su valor es por defecto AUTO, que deja al navegador la decisión. Las otras opciones que tenemos son YES y NO.
NORESIZE	Si lo especificamos el usuario no podrá cambiar de tamaño el marco.
FRAMEBORDER	Al igual que su homónimo en la etiqueta <FRAMESET>, si lo igualamos a cero se eliminará el borde con todos los marcos contiguos que tengan también este valor a cero.
MARGINWIDTH	Permite cambiar los márgenes horizontales dentro de un marco. Se representa en pixels.
MARGINHEIGHT	Igual al anterior pero con márgenes verticales.

## 11.3. Acceso a marcos externos

Por defecto, cuando pulsamos sobre un enlace situado dentro de un marco, la nueva página a la que queremos acceder la veremos encerrada en ese mismo marco. Es posible que deseemos que esto no ocurra. Por ejemplo, si tenemos un marco que no sirve de índice y otro donde mostramos los contenidos sería deseable que los enlaces del marco índice se abrieran en el otro marco. Esto es posible hacerlo gracias al parámetro TARGET.

Este parámetro se puede colocar en tres etiquetas: <A>, <AREA> y <BASE>. En las dos primeras sirve para indicar el marco en el que abriremos ese enlace en particular y el último modificaremos el marco en el que por defecto se nos muestran todos los enlaces.

Pero para que un parámetro funcione, es habitual que le asignemos un valor, y TARGET no es una excepción. Para indicarle el marco que deseamos le asignaremos el nombre del mismo. Así, en los ejemplos anteriores, si en el

marco llamado indice tenemos un enlace que queremos se abra en el marco principal pondremos:

```
<A HREF="pagina.html" TARGET="principal">
```

También existen cuatro nombres reservados que podremos utilizar en el parámetro TARGET:

<b>_top</b>	Elimina todos los marcos existente y muestra la nueva página en la ventana original sin marcos.
<b>_blank</b>	Muestra la nueva página en una ventana nueva y sin nombre del navegador.
<b>_self</b>	Muestra la nueva página en el marco donde está declarado el enlace.
<b>_parent</b>	Muestra la nueva página en el <FRAMESET> que contiene al marco donde se declara el enlace. En el ejemplo que pusimos de <FRAMESET> anidados, una enlace situado en el marco ejemplo cuyo parámetro TARGET fuese igual a _parent eliminaría la separación entre los marcos ejemplo y principal y mostraría en ese nuevo marco la nueva página.



