Curso de HTML

INTRODUCCIÓN

WWW

El servicio Web o WWW es una nueva forma de representar la información en Internet basada en páginas. Una página WWW puede incluir tres tipos de información: texto, gráficos e hipertexto. Un hipertexto es texto resaltado que el usuario puede activar para cargar otra página WWW. La diferencia entre un documento hipertexto y un documento normal consiste en que el hipertexto contiene, además de la información, una serie de enlaces o conexiones con otros documentos relacionados, de manera que el lector puede pasar de un tema a otro y volver al documento original en el momento en que le interese.
Las principales ventajas del servicio WWW son tres:

1. Se puede combinar texto y gráficos.
2. Los hiperenlaces permiten cargar páginas de cualquier otro servidor conectado a Internet, da igual que esté localizado en Argentina o en Australia.
3. La creación de páginas WWW es bastante sencilla mediante el lenguaje HTML.

HTML

El HTML **(Hyper Text Markup Language)** es un sistema para estructurar documentos. Estos documentos pueden ser mostrados por los visores de paginas Web en Internet, como Netscape, o Microsoft Explorer. Por el momento no existe un estándar de HTML, ya que hay directivas que funcionan en Explorer pero no en Netscape, y viceversa. De cualquier manera existen diferentes revisiones o niveles de estandarización, el 1.0, el 2.0 y el 3.0, lo que produce que algunos visores no "comprendan" en su totalidad el contenido de un documento.
Básicamente, el HTML consta de una serie de órdenes o directivas, que indican al visor que estemos utilizando, la forma de representar los elementos (texto, gráficos, etc...) que contenga el documento. Las directivas de HTML pueden ser de dos tipos, cerradas o abiertas. Las directivas cerradas son aquellas que tienen una palabra clave que indica el principio de la directiva y otra que indica el final. Entre la directiva inicial y la final se pueden encontrar otras directivas. Las directivas abiertas constan de una sola palabra clave. Para diferenciar las directivas del resto del texto del documento se encierran entre los símbolos “<” y “>”. Las directivas cerradas incluyen el carácter / antes de la palabra clave para indicar el final de la misma. Una directiva puede contener "parámetros". Estos parámetros se indican a continuación de la palabra clave de la directiva.
Por tanto, como hemos visto, HTML es un lenguaje muy sencillo que nos permite preparar documentos Web insertando en el texto de los mismos una serie de marcas (tags) que controlan los diferentes aspectos de la presentación y comportamiento de sus elementos. Para escribir HTML lo único que se necesita es un editor de texto ASCII, como EDIT del MS-DOS o el Bloc de notas de Windows.

Clientes y servidores WWW

Para poder utilizar el servicio Web se necesitan dos partes. Por un lado, la empresa o institución que quiere facilitar su información tiene que crear páginas WWW, siguiendo el estándar definido por el lenguaje HTML, y ponerlas a disposición del público en Internet, en lo que se llama un servidor WWW. Por otro lado, el usuario que quiere acceder a dichas páginas tiene que utilizar un programa (cliente WWW) que lea las páginas WWW e interprete su significado (por ejemplo, un link). Como dijimos anteriormente, estos programas navegadores o clientes WWW (Netscape, Explorer), son los que permiten al ordenador del usuario interpretar el lenguaje HTML.

Protocolo de direccionamiento de documentos. El URL

Interconectar documentos por todo el planeta sobreentiende un medio único de identificación en la red Internet. La dirección única de un documento en WWW es llamada URL -Uniform Resource Locator- y se compone de los siguientes elementos:

4. el protocolo de intercambio de datos entre el cliente y el servidor. (HTTP)
5. la dirección Internet del servidor que difunde los documentos. Esta dirección es única en toda la red, es la dirección TCP/IP de la máquina. Tiene la forma de una serie de números como 134.158.69.113; al ser estos números difíciles de memorizar, un anuario (DNS) resuelve generalmente la relación entre dirección numérica y nombre simbólico de la máquina/nombre del ámbito (ejemplo: 134.158.48.1 es la dirección de la máquina sioux.in2p3.fr en la que sioux representa el nombre de la máquina y .in2p3.fr el nombre del ámbito);
6. el árbol de directorios (el camino) que conduce al documento;
7. el nombre del documento que tendrá siempre la extensión .html o .htm.
8. Menos frecuentemente esta dirección podrá completarse con otros elementos:
9. el puerto;
10. información de autentificación (username y password);
11. argumentos que se pasarán a un programa en la llamada de un enlace ejecutable.

La sintaxis mínima utilizada para representar el URL de un documento es la siguiente:
protocolo://nombre_del_servidor/
cuando no se especifica un nombre de fichero se acudirá al fichero predeterminado del servidor, habitualmente la *home page.*.

La sintaxis que se encuentra habitualmente es:
protocolo://nombre_del_servidor/directorio/subdirectorio/nombre_del_documento

La sintaxis completa es:

protocolo://username;password@nombre_del_servidor:puerto/directorio/subdirectorio//nombre_del _documento?argumentos
Se observará también en ciertas direcciones la presencia del signo tilde (~) delante del nombre de un directorio. Se trata de home pages personales, posibilidad ofrecida a los usuarios que tienen una cuenta en la máquina servidor.

Ejemplos de URL:
http://www.hotmail.com
http://www.yahoo.com.ar

JAVA SCRIPT

El lenguaje Java Script surge como respuesta a la necesidad de aumentar el dinamismo de las paginas desarrolladas con HTML, permitiendo a los usuarios una mayor interactividad con las mismas. El uso del Java Script (ejecutado en el cliente) descarga la realización de muchas de las funciones relacionadas con la gestión de interfases de usuario, aumentando el rendimiento global del sistema Cliente-Servidor.
Java Script no es un lenguaje de propósito general; sino que se incrusta dentro del documento HTML contenido entre las etiquetas <SCRIPT> y </SCRIPT>. Existen en el mercado otros lenguajes que ofrecen las mismas posibilidades (como VBScript) pero Java Script es en la actualidad el más extendido pasando a ser estándar.

Algunas características de Java Script son las siguientes:

1. Es interpretado por un interprete incorporado en el navegador .
2. La sintaxis es muy similar a la de Java o C++.
3. Es un lenguaje basado en objetos que no implementa el concepto de clases, ni el mecanismo de herencia.
4. Establece una jerarquía de objetos encabezada por el navegador, permitiendo controlar el acceso a todos los elementos u objetos contenidos en él.
5. Facilita el tratamiento de eventos provocados tanto por el navegador o por el mismo usuario.
6. Sirve de base para incorporar otros elementos tecnológicos como Actives, XML, HTMLD, y controles de multimedia.

HTML

Importante

Antes de comenzar, hay que decir es que las directivas utilizadas en este curso no explica ningún estándar específico, ni examina exhaustivamente todos los parámetros de las etiquetas HTML. Los ejemplos están probados con Explorer 4 y Netscape 4; esto implica si usas una versión anterior a estos, puede que algunos de ellos no te funcionen.

Un poco de Historia

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 encontremos será capaz de interpretarlo. Prácticamente todo lo que veamos en los próximos seis capítulos está contemplado por este estándar.

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.

Estructura básica de una página

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

<BODY>
        <CENTER>Mi pagina</CENTER>
        Esta es mi primera pagina.
        <HR>
        Solo Contiene texto plano.
</BODY>
</HTML>
```

Lo primero que conviene explicar es en qué consisten todos esos símbolos de mayor y menor que están distribuidos por ahí. El lenguaje HTML se basa en la sintaxis SGML (toma siglas). Esto quiere decir que cualquier cosa que hagamos en HTML estará encerrada entre dos etiquetas de esta manera:

<ETIQUETA parámetros> ... </ETIQUETA>

Hay ocasiones en que no es necesario *cerrar* la etiqueta. Mirando el código habréis visto un par de ejemplo que ya explicaré más adelante. Pero como lo primero que debemos indicar es que el texto que estamos componiendo es un documento HTML pues lo indicamos así:

<HTML> ... </HTML>

Un documento HTML tiene una estructura que lo separa en dos partes: cuerpo y cabecera. En la primera estará la página en sí, mientras que en la segunda incluiremos algunas cosas que no se ven al principio pero que pueden llegar a ser muy importantes. Lo primero que hay que incluir en el código es la cabecera. La escribimos:

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

Dentro de la cabecera sólo hay otra etiqueta. Es la única imprescindible: el título de la página. Es lo que veremos como título de la ventana en los navegadores que lo permitan. Es como se conocerá nuestra página en algunos buscadores y

en la agenda de direcciones (*bookmarks*) de los usuarios. Por tanto, parece importante pensarnos bien como llamarla.

El cuerpo del documento

Ahora vamos a indicar el contenido. Lo primero será indicar que estamos en el cuerpo del documento:

<BODY> ... </BODY>

Luego pondremos el título algo recalcado:

<CENTER> ... </CENTER>

Con esto colocaremos el texto centrado (<CENTER>). Luego separamos ese título que le hemos puesto a la página del texto por medio de una línea horizontal:

<HR>

La línea horizontal carece de etiqueta de cierre. Esto es normal en etiquetas que no varían los atributos de un texto, sino que insertan un elemento.

Formateo básico

Se pueden establecer varias categorías dentro de las etiquetas usadas para formatear el texto. Nosotros las dividiremos entre aquellas que sirven para cambiar párrafos enteros y las que son capaces de formatear tiras de caracteres dentro del párrafo.

Formato del párrafo

Estas son las etiquetas más importantes (excluyendo algunas que veremos más adelante):

Etiqueta	Utilidad	Resultado
<P>	Sirve para delimitar un párrafo. Inserta una línea en blanco antes del texto.	Soy un párrafo
<CENTER> ... </CENTER>	Permite centrar todo el texto del párrafo.	Yo soy normal Yo estoy centrado
<PRE WIDTH=x> ... </PRE>	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.	`Estoy en paso fijo`
<DIV ALIGN=x> ...	Permite justificar el texto del párrafo a la izquierda (ALIGN=LEFT), derecha (RIGHT),	Yo estoy a la izquierda Yo al centro

</DIV>	al centro (CENTER) o a ambos márgenes (JUSTIFY)	Y yo a la derecha (recuerda a la política esto, oye) Yo soy el más chulo, porque estoy en todos los lados.
<ADDRESS> ... </ADDRESS>	Para escribir direcciones (de esas donde vive la gente, no electrónicas).	*Daniel Rodríguez Herrera* *C/Ecuador 9, 1ºB 28220* *Majadahonda*
<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.	Me gustaría reencarnarme en las yemas de los dedos de Warren Beatty (Woody Allen)

Las cabeceras

El HTML nos ofrece seis etiquetas distintas para mostrar cabeceras. Son éstas:

Etiqueta	Resultado
<H1> ... </H1>	**Cabecera de nivel 1**
<H2> ... </H2>	**Cabecera de nivel 2**
<H3> ... </H3>	**Cabecera de nivel 3**
<H4> ... </H4>	**Cabecera de nivel 4**
<H5> ... </H5>	**Cabecera de nivel 5**
<H6> ... </H6>	**Cabecera de nivel 6**

Estas etiquetas se pueden definir como de formato de párrafo pero por su importancia he prefirido tratarlas aparte. 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 nuestra página, tal y como se hace en un documento de texto normal.

Cambiando el tipo de letra

Todas estas etiquetas nos permiten cambiar de una manera u otra el aspecto del tipo de letra que estemos utilizando y se pueden utilizar con tiras de caracteres dentro de un párrafo.

Etiqueta	Utilidad	Resultado
<B> … </B>	Pone el texto en negrita.	**Soy un texto negro como el tizón**
<I> ... </I>	Representa el texto en cursiva.	*Estoy ladeado*
<U> ... </U>	Para subrayar algo.	Como soy muy importante estoy subrayado
<S> … </S>	Para tachar.	~~A mí, en cambio, nadie me quiere~~
<TT> ... </TT>	Permite representar el texto en un tipo de letra de paso fijo.	`No soy variable`
^{...}	Letra superíndice.	$E=mc^2$
_{...}	Letra subíndice.	$a_{i,j}=b_{i,j}+1$
<BIG> … </BIG>	Incrementa el tamaño del tipo de letra.	Soy GRANDE
<SMALL> ... </SMALL>	Disminuye el tamaño del tipo de letra.	Creí ver un lindo gatito
<BLINK> ... </BLINK>	Hace parpadear el texto. Resulta algo irritante.	¿Molesto?

Formato de frase

En estos elementos indicas el tipo de información que encierran las etiquetas, pero no como se representan:

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

No son muy utilizados, ya que no permiten tener un control exacto de la manera en que la página se representará finalmente.

Otros elementos

Por último, debemos estudiar algunas cosas que no son texto y que podemos incorporar a nuestra página.

Etiqueta	Utilidad	Resultado
<HR>	Inserta una barra horizontal.	
 	Salto de línea.	Hay un antes y un después de saltar a otra línea
<!-- ... -->	Comentarios.	Esto se escribe y <!-- esto no -->

Caracteres especiales

Si os habéis fijado en los ejemplos habréis visto que en los textos de los mismos no hay acentos, ni eñes, ni símbolos de abrir interrogación o exclamación. Esto es debido a los distintos juegos de caracteres que manejan los ordenadores.
Las máquinas manejan la información en formato binario (es decir, en unos y ceros). Estos, a su vez, forman números, los cuales se traducen en letras. ¿Cómo? Mediante tablas. Podemos asignar el valor 64 a la letra a, el 65 a la b, etc..
El problema está en que cada ordenador es de un fabricante distinto y puede adoptar una tabla diferente al resto. Para evitarlo existen diversos estándares y el más extendido es el ASCII. De hecho, actualmente todos los ordenadores tienen la misma tabla ASCII para los primeros 127 caracteres. Pero esa tabla no contiene vocales con acento, ni eñes, ni símbolos de abrir interrogación o exclamación... Esto nos pasa por dejar que los norteamericanos sean quienes construyan las computadoras.
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.

Caracteres extendidos en HTML

La manera de incluir los caracteres extendidos (cuyo número está más allá del 127) consiste en encerrar el código entre los caracteres &# y ;. Así pues, lo siguiente:

½

nos debería dar un medio (½. También existe una serie de sinónimos para poder recordar con más facilidad estos caracteres. Así, por ejemplo, ½ también se puede escribir como ½. Vamos a ver algunos de estos códigos, los más útiles a la hora de escribir en español:

á, Á, é, É,...
á, Á, é, É, í, Í, ó, Ó, ú y Ú

ñ y Ñ
ñ y Ñ

¿
¿

¡
¡

º

ª
ª

™ o ™
™o ™

©
©

®
®

(espacio en blanco que no puede ser usado para saltar de línea)

Caracteres de control

En el HTML existen cuatro caracteres de control, que se usan para formar etiquetas, establecer parámetros, etc.. Para poder emplearlos sin riesgo deberemos escribir los siguiente códigos:

<
<

>
>

&
&

"

Enlaces

Las siglas HTML significan HyperText Markup Language, lo que para nosotros quiere decir que es un lenguaje para hipertexto. Existen múltiples formatos de

hipertexto (por ejemplo, los ficheros de ayuda de Windows) y lo que tienen en común es que todos poseen enlaces.

Un enlace es una zona de texto o gráficos que si son seleccionados nos trasladan a otro documento de hipertexto o a otra posición dentro del documento actual. Siendo HTML el lenguaje de Internet, la diferencia que posee con respecto a otros tipos de hipertexto es que ese otro documento puede estar físicamente en la otra punta del planeta. Son los enlaces lo que hacen de la telaraña o World Wide Web lo que es.

La etiqueta <A>

Para incorporar un enlace hay que utilizar esta etiqueta. Todo lo que encerremos entre <A> y </A>, ya sea texto o imágenes, será considerado como enlace y sufrirá dos modificaciones:

Se visualizará de manera distinta en el navegador. El texto aparecerá subrayado y de un color distinto al habitual, y las imágenes estarán rodeadas por un borde del mismo color que el del texto del enlace.
Al pulsar sobre el enlace, seremos enviados al documento que apuntaba el enlace.

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

<A HREF="direccion">Pulsame</A>

La dirección estará en formato URL (Uniform Resource Locator).

Las URLs

Una URL nos indica tanto una dirección de Internet como el servicio que esperamos nos ofrezca el servidor al que corresponde la dirección. Tiene el siguiente formato:

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

donde el servicio podrá ser uno de los siguientes:

http: Es el servicio invocado para transmitir páginas web y el que usaremos normalmente en los enlaces.

https: 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 intercambiemos con él de terceras personas.

ftp: Permite trasmitir ficheros desde servidores de ftp anónimo. Si no le pedimos un fichero sino un directorio, en general el navegador se encargará de mostrarnos el contenido del mismo para que podamos escogerlo cómodamente. Utilizando la @ podremos acceder a servidores privados.

malito: No es implementado generalmente por los navegadores, que suelen invocar un programa externo. Nos permite conectarnos con otros ordenadores y entrar en ellos como si nuestro ordenador fuese una terminal del mismo.
La dirección de la máquina puede ser, o bien una serie de cuatro números entre 0 y 255 (123.3.5.65) o bien algo más fácil de recordar como es una serie de palabras separadas por puntos (www.programacion.net). El puerto generalmente no se indica, ya que el servicio predetermina uno.

La ruta es una serie de directorios separados por el símbolo /, que es el utilizado en UNIX (el sistema operativo más extendido en los servidores de Internet).

Existe otro formato de URL. Cuando queremos acceder a un fichero situado en la misma máquina que la página web que estamos creando podemos utilizar este formato:

ruta_relativa/fichero

En la ruta relativa podremos utilizar los dos puntos (..) para acceder al directorio padre o comenzar con la barra diagonal (/) para acceder a una ruta absoluta dentro de nuestro ordenador.

Anclas

Como dijimos, es posible acceder a una posición del documento HTML. Para hacerlo, primero debemos especificar el lugar del documento al que queremos acceder:

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

Para poder ver bien como funciona, he colocado un ancla de ejemplo en el título de la sección 6.2. Para poder acceder a ese lugar incluimos el enlace de esta manera:

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

También podemos acceder a anclas situadas en documentos remotos. Para ello añadiremos el nombre del ancla al URL así:

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

Listas

Existen varios tipos de listas en HTML. Todas ellas se pueden meter unas dentro de otras formando árboles muy bonitos y preciosos. Todos los tipos siguen el siguiente formato:

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

tipo_lista puede ser una de las siguientes: DIR, DL, MENU, OL y UL.

Listas desordenadas

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

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

se verá como

1. Primer elemento
2. Segundo elemento

La etiqueta <UL> admite estos parámetros:

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>

También son listas desordenadas aquellas que utilizan las etiquetas <DIR> y <MENU>. En principio tenían como propósito representar una lista estilo directorio (multicolumna) o tipo menú (una sola columna). En la práctica los navegadores lo han implementado como sinónimos de <UL>, por lo que no los estudiaremos aquí.

Listas ordenadas

La etiqueta <OL> nos permite presentar listas de elementos ordenados de menor a mayor. Normalmente cada elemento de la lista irá precedido por su número en el orden. Por ejemplo,

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

se verá como

1. Primer elemento
2. Segundo elemento

La etiqueta <OL> admite estos parámetros:

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>.

START="num" 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.

Listas de definiciones

Este es el único tipo de lista que no utiliza la etiqueta <LI>. Al tener como objetivo presentar una lista de definiciones, de modo que tiene que representar de manera distinta el objeto definido y la definición. Esto se hace por medio de las etiquetas <DT> y <DD>:

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

se verá como:

Primer elemento
    Es un elemento muy bonito.
Segundo elemento
    Este, en cambio, es peor.

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

Imágenes

Para incluir gráficos e imágenes en nuestras páginas utilizaremos la etiqueta <IMG> de esta manera:

```
<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. La últimas versiones de Netscape y Explorer aceptan también el formato PNG.
El parámetro ALT especifica el texto que se mostrará en lugar del gráfico en aquellos navegadores que no sean capaces de mostrarlos y en el supuesto de que el usuario los haya desactivado. Algunos navegadores lo muestran cuando pasamos el ratón por encima de la imagen. Es por eso que, aunque algunos usuarios no lo lleguen a ver nunca, conviene ponerlo siempre. 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 píxeles. De este modo, el navegador puede mostrar un recuadro del tamaño de la imagen mientras la va leyendo de la red y así poder mostrar el resto de la página correctamente mientras tanto.

```
<IMG SRC="graficos/dwnldns.gif" ALT="Netscape 4.0" WIDTH=88 HEIGHT=31>
```

Para los menos avezados en inglés, decir que WIDTH es la anchura y HEIGHT la altura.

Imágenes y enlaces

Suele ser común incluir enlaces dentro de un gráfico. En ese caso, por defecto, los navegadores le pondrán un borde al gráfico para indicar que efectivamente es un enlace. Práctico, pero la mayoría de las veces bastante poco estético. Por medio del parámetro BORDER podremos alterar el grosor de ese borde o incluso eliminarlo poniéndolo a cero.

```
<A HREF="http://www.netscape.com">
  <IMG SRC="graficos/dwnldns.gif" ALT="Netscape 4.0" WIDTH=88 HEIGHT=31>
</A>
```

Formateo fino

Lo ideal cuando trabajas con texto sería poder cambiarlo al tamaño que te viniese bien, ponerlo de colorines y cambiar el tipo de letra. Todo esto puedes hacerlo gracias a la etiqueta <FONT>.

Cambio de color

Para hacerlo vamos a utilizar el parámetro COLOR. La manera de especificarle el color es común a todas las etiquetas HTML: o bien indicando el nombre, si es un color normal, o bien especificando el porcentaje de rojo, verde y azul (código RGB) del mismo. Los colores reconocidos son los siguientes:

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

<FONT color="red">Estoy en rojo</FONT>

El modo de indicar el color RGB es el siguiente:
La primera componente en hexadecimal es el rojo, la segunda el verde y la tercera el azul (Red Green Blue, RGB).

Tamaños del texto

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

El tamaño por defecto es 3. También se puede utilizar los modificadores + y - para indicar un incremento (o decremento) relativo del tamaño del tipo de letra. Así, por ejemplo, si indicamos que queremos un tamaño de -2 estaremos pidiendo al navegador que nos muestre el tipo de letra dos veces más pequeño.

<FONT SIZE=2>Tamaño 2<FONT SIZE="+3">Tamaño 6</FONT></FONT>

Tipo de letra

Por último, podemos especificar el nombre del tipo de letra que queremos utilizar gracias al parámetro FACE. Como en principio no tenemos manera de saber que tipo de letra tiene instalado el ordenador del usuario que está viendo nuestras

páginas, podemos indicar más de uno separado por comas. Si el navegador no encuentra ninguno seguirá utilizando el que tiene por defecto:

```
<FONT FACE="Helvetica,Arial,Times">No sé como voy a salir
exactamente</FONT>
```

De todos modos es recomendable no utilizar con fe ciega este atributo en Internet, ya que impide que todos puedan ver nuestras páginas como nosotros. E Internet, siempre que nos lo permitan Microsoft y Netscape, debe ser lo más estándar posible.

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, como veremos en los capítulos correspondientes.
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]">
```

nos indicaría la herramienta con que hemos creado la página (en este caso la versión 4.03 en español para Windows 95 del Composer de Netscape). Estas son las propiedades más comunes:

Propiedad	Utilidad
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.dc.uba.ar/materias">
```

Ahora todas nuestras URLs relativas se referirán al directorio /materias dentro del servidor http://www.dc.uba.ar

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>:

Parámetro	Utilidad
BACKGROUND	Permite definir un gráfico de fondo para la página.
BGCOLOR	Permite definir el color de fondo de la página.
BGPROPERTIES	Cuando es igual a FIXED el gráfico definido como fondo de la página permanecerá inmóvil aunque utilicemos las barras de desplazamiento.
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 píxeles 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.

Formularios

Una de las mayores ventajas de la web es que resulta tremendamente interactiva. Los usuarios de una página no tienen más que escribir al autor de la misma para comentarle cualquier cosa de la misma. Sin embargo, si deseamos que nos digan sólo unas cosas concretas (responder a alguna pregunta, seleccionar entre varias opciones, etc..) deberemos utilizar formularios. Por ejemplo,

```
<FORM ACTION="" METHOD=POST>
Nombre:<BR><INPUT NAME="nombre" TYPE=TEXT SIZE=32>
<BR>¿Cuantos son dos y dos?<BR>
<INPUT NAME="Respuesta" TYPE=RADIO VALUE="mal">3<BR>
<INPUT NAME="Respuesta" TYPE=RADIO VALUE="bien">4<BR>
<INPUT NAME="Respuesta" TYPE=RADIO VALUE="mal">5<BR>
<INPUT TYPE="Submit" VALUE="Comprobar">
</FORM>
```

se verá así:

Nombre:

¿Cuantos son dos y dos?

- 3
- 4
- 5

Enviar

El botón no hace nada porque no hemos definido qué debe hacer, así que sed buenos y no lo pulséis.
Todos los elementos de un formulario deben estar encerrados entre <FORM> y </FORM>. Como parámetros cabe destacar tres. ACTION define el URL que deberá gestionar el formulario. Puede ser una dirección de correo (precedida del inevitable mailto:, en cuyo caso deberemos añadir el parámetro ENCTYPE="text/plain" para que lo que recibamos resulte legible.

Por otro lado, tenemos el parámetro METHOD define la manera en que se mandará el formulario. Es recomendable utilizar POST. En el caso de que estemos mandando el formulario a nuestra dirección de correo electrónico es obligado usarlo.

Ahora vamos a ver uno a uno todos los elementos que podemos incluir en un formulario. Veremos que todos ellos tienen algo en común. Como el resultado de cualquier formulario es una lista de variables y valores asignados a las mismas, todos ellos tendrán un atributo en común: el nombre de su variable. El parámetro también será común a todos: NAME.

Cajas de texto

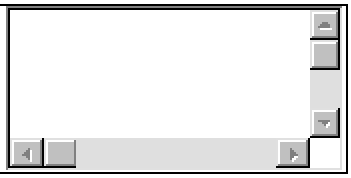
Existen tres maneras de conseguir que el usuario introduzca texto en nuestro formulario. Las dos primeras se obtienen por medio de la etiqueta <INPUT>:

<INPUT TYPE=TEXT>	
<INPUT TYPE=PASSWORD>	

El primero nos dibujará una caja donde escribir un texto (de una sola línea). El segundo es equivalente, pero no veremos lo que tecleemos en él. Estos son los atributos para modificarlos:

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.

Por otro lado, puede que necesitemos que el usuario pueda introducir más de una línea. En ese caso se utilizará la siguiente etiqueta:

<TEXTAREA> Por defecto </TEXTAREA>		

Lo que incluyamos entre las dos etiquetas será lo que se muestre por defecto dentro de la caja. Admite estos parámetros:

Parámetro	Utilidad
ROWS	Filas que ocupará la caja de texto.
COLS	Columnas que ocupará la caja de texto.

Opciones

Si lo que deseamos es que el usuario decida entre varias opciones podremos hacerlo de dos modos. El primero es el que vimos en el ejemplo inicial:

```
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>
```

3 ○
4 ○
5 ○

Para asociar varios botones de radio a una misma variable les pondremos a todos ellos el mismo NAME. Aparte de esto acepta los siguientes parámetros:

Parámetro	Utilidad
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.

Pero también tenemos una posibilidad que ocupa bastante menos: las listas desplegables. Para emplearlas deberemos utilizar dos etiquetas, SELECT y OPTION:

```
<SELECT NAME="Navegador">
  <OPTION>Netscape
  <OPTION>Explorer
  <OPTION>Opera
  <OPTION>Lynx
  <OPTION>Otros
  </SELECT>
```

Los parámetros que admite SELECT son las siguientes:

Parámetro	Utilidad
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.

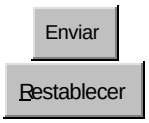
Y OPTION estos:

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.

Botones del formulario

Existen dos: uno que se utiliza para mandar el formulario y otro que sirve para limpiar todo lo que haya rellenado el usuario:

```
<INPUT TYPE=SUBMIT ><BR>
<INPUT TYPE=RESET>
```



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

Otros elementos

Puede que necesito que el usuario sencillamente nos confirme o niegue algo. Lo podremos conseguir por medio de controles de confirmación:

```
<INPUT NAME="Belleza" TYPE=CHECKBOX>Me considero guapo/a
```

 Me considero guapo/a

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.

Por último, existe la posibilidad de que necesitemos que, en el formulario, tengamos alguna variable con un valor previamente asignado. Por ejemplo, en todos los cursos que tengo el formulario es el mismo. Y de alguna manera tendré que distinguirlos cuando me lleguen, digo yo. Así que incluyo algo como esto:

```
<INPUT TYPE=HIDDEN NAME="Curso" VALUE="HTML 4.0">
```

De este modo ya sé de que curso me están hablando.

Controles avanzados para formularios

El estándar HTML 4.0 ha traído varias mejoras a los formularios, que acercan los mismos a las características que tienen en lenguajes como Java o Visual Basic. Desafortunadamente, el Netscape 4, lanzado al mercado antes de la aprobación

del HTML 4.0, no implementa ninguna de estas mejoras, por lo que los ejemplos de este capítulo sólo serán contemplados correctamente por los usuarios de Explorer 4 y 5 y los arriesgados usuarios de las versiones beta del futuro Netscape 5.

Botones

Una de las cosas que más han mejorado son los botones. Ahora disponen de una etiqueta propia, de modo que se pueda encerrar con ella todo tipo de elementos HTML, como gráficos o, incluso, tablas enteras.

Como no podía ser de otra manera, la etiqueta en cuestión se llama BUTTON:

```
<BUTTON TYPE="button">
  <table border="1">
  <tr>
    <th>Soy una</th>
    <th>tabla</th>
  </tr>
  <tr>
    <td>que está</td>
    <td>en un botón</td>
  </tr>
  </table>
</BUTTON>
```

Los parámetros de dicha etiqueta son los normales, TYPE, que podrá tomar los valores SUBMIT (por defecto), RESET y BUTTON, NAME y VALUE.

Por otro lado, ahora podemos declarar un gráfico como un elemento más de entrada. como un nuevo tipo dentro del elemento INPUT:

```
<INPUT TYPE="image" SRC="graficos/download.gif" alt="Now'>
```

Este elemento se comportará de mismo modo que un botón de tipo SUBMIT, pero añadirá como información en el formulario las coordenadas x e y donde el usuario lo pulsó.

Etiquetas

Hasta hora, el texto que acompañaba a los campos de entrada no estaba asociado a los mismos de ninguna manera. Así, por ejemplo, si pulsábamos en el texto que acompañaba a un control de confirmación, no sucedía nada. Ahora, en cambio, si utilizamos la etiqueta LABEL, el control cambiará de estado (sólo disponible en Netscape 5):

```
<LABEL>  <INPUT NAME="Belleza" TYPE=CHECKBOX>
  Me considero guapo/a </LABEL>
```

 Me considero guapo/a

Lo bueno que tiene es que se puede usar sin peligro, ya que no afectará a los usuarios de navegadores antiguos.

Agrupación de elementos

Hasta ahora, no disponíamos de ninguna manera de agrupar visualmente varios controles, si no echábamos mano de elementos que no son del formulario, como tablas o imágenes.

Ahora, si encerramos una parte de un formulario dentro de la etiqueta FIELDSET se mostrará un rectángulo alrededor de los mismos. Si además, le indicamos un título por medio de la etiqueta LEGEND nuestros formularios quedarán hechos un verdadero primor:

```
<FIELDSET>
  <LEGEND>Mi hermoso formulario</LEGEND>
  <LABEL>
    Mi texto:
    <INPUT TYPE="text">
  </LABEL>
</FIELDSET>
```

Mi hermoso formularioMi texto:

LEGEND admite como parámetro ALIGN, que indicará en qué lugar se coloca el título. Por defecto es TOP (arriba), pudiendo estar también abajo (BOTTOM), a la izquierda (LEFT) o a la derecha (RIGHT).

Desactivación de elementos

Todos los controles de un formulario se pueden desactivar, impidiendo así al usuario que los utilice. Se seguirán mostrando en pantalla, aunque con un aspecto distinto para indicar su triste estado. Para ello sólo tenemos que indicarle el parámetro DISABLED:

```
<LABEL DISABLED>Texto:
  <INPUT TYPE=TEXT DISABLED>
</LABEL>
```

Texto:

Esto, en principio, no parece de demasiada utilidad. ¿Para qué queremos tener controles desactivados? Para eso no los ponemos, ¿no? Lo bueno que tiene es que el estado de activación de un control es accesible desde JavaScript. Eso nos permitirá activar o desactivar una parte de nuestro formulario dependiendo de lo que el usuario haya introducido previamente en otros controles del mismo.

Mapas

Hemos visto que podemos hacer enlaces de texto y de gráficos. Pero también existe otra posibilidad: que dentro de una sola imagen podamos acceder a varios enlaces. Se hace declarando zonas dentro de la imagen (rectángulos, círculos, etc..), siendo cada una de ellas un enlace distinto. Tradicionalmente, siempre han existido dos maneras de hacerlo:

1. Mapas gestionados por el cliente (el navegador).
2. Mapas gestionados por el servidor.

Los segundos fueron los primeros en desarrollarse y estaban incluidos dentro del estándar HTML 2.0. Sin embargo, nunca hubo una manera común de gestionar esos mapas. Debido a ello, Netscape elaboró un sistema propio que fue incluido en el estándar 3.2: los mapas gestionados por el navegador.

Mapas gestionados por el cliente

Para utilizarlos necesitaremos dos cosas: declarar el mapa y asignarlo a una imagen. Vamos primero a declarar nuestro mapa:

```
<MAP NAME="mi_mapa">
  <AREA SHAPE=... COORDS=... ALT="Enlace a..">
  ...
</MAP>
```

Dentro de la etiqueta MAP sólo podremos definir el nombre del mapa (algo imprescindible, por otra parte). Es dentro de cada elemento AREA donde podremos definir cosas más interesantes:

Parámetro	Utilidad
SHAPE	Define la forma de la zona, que podrá ser rectangular, circular o poligonal.
COORDS	Coordenadas (separadas por comas) que definen la zona. El número y significado de esas coordenadas dependerá de la zona.
HREF	URL a donde irá el usuario si pulsa sobre esa zona.
NOHREF	Especifica que esa zona no tiene asignado enlace alguno.
ALT	Texto que se presentará en lugar de la imagen en navegadores no gráficos para acceder al enlace.

Como podemos ver, para declarar correctamente una zona necesitamos conocer cómo se definen exactamente las formas y coordenadas:

	SHAPE	COORDS
Rectangular	RECT	"x1,y1,x2,y2" siendo (x1,y1) la esquina superior izquierda y (x2,y2) la inferior derecha.

Circular	CIRC	"x,y,radio" siendo (x,y) el centro del círculo y radio su... eehh.. ¿cómo lo diría yo? ¿Su radio?.
Polígono irregular	POLY	"x1,y1,x2,y2,x3,y3,..." definiendo cada pareja (x,y) una esquina del polígono. La última pareja de coordenadas se unirá a la primera para cerrar el polígono.
Toda la imagen	DEFAULT	No se indican.

Cómo usar un mapa

Ahora que hemos definido un mapa, sólo queda asignarlo a una imagen. Esto se hace del siguiente modo:

<IMG SRC=... USEMAP="#mi_mapa">

Siempre tenemos que añadir al comienzo del nombre de nuestro mapa una almohadilla (#). Vamos a ver un ejemplo:

```
<MAP NAME="navegadores">
  <AREA SHAPE=RECT COORDS="0,0,24,31"
      HREF="http://www.netscape.com" ALT="Netscape">
  <AREA SHAPE=RECT COORDS="26,0,53,31"
      HREF="http://www.microsoft.com" ALT="Microsoft">
  <AREA SHAPE=DEFAULT NOHREF ALT="Nada">
</MAP>
<IMG SRC="nav.gif" WIDTH=53 HEIGHT=31 BORDER=0
USEMAP="#navegadores">
```

Hay que tener en cuenta que, cuando dos zonas se solapan, la que esté declarada primero es la que tiene preferencia. Por eso declaramos al final una zona que no conduce a ningún URL por si el usuario pulsa con el ratón donde no debe.

Tablas

Las tablas son posiblemente la manera más clara de organizar la información. También es el modo más adecuado de maquetar texto y gráficos de una manera algo más controlada que con los parámetros ALIGN.

Las tablas se definen de la siguiente manera. Primero, las características de la tabla, luego las de cada fila y dentro de ésta, cada celda. Así pues, una tabla con 2 filas y 3 columnas se declarará así:

Código	Resultado
<TABLE> <TR> <TD>1,1</TD> <TD>1,2</TD> <TD>1,3</TD> </TR> <TR> <TD>2,1</TD> <TD>2,2</TD> <TD>2,3</TD> </TR> </TABLE>	1,1 1,2 1,3 2,1 2,2 2,3

Como podéis ver (o mejor no ver) la tabla no tiene mucho aspecto de tabla. Quedaría mejor con unos bordes, ¿no? Puede que tampoco le viniese mal mayor espacio entre celdas o mayor anchura. Estas son las cosas que podremos cambiar con los atributos de TABLE:

Parámetro	Utilidad
BORDER	Especifica el grosor del borde que se dibujará alrededor de las celdas. Por defecto es cero, lo que significa que no dibujará borde alguno.
CELLSPACING	Define el número de píxeles que separarán las celdas.
CELLPADDING	Especifica el número de píxeles que habrá entre el borde de una celda y su contenido.
WIDTH	Especifica la anchura de la tabla. Puede estar tanto en píxeles 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).

Definir las filas

Ahora que hemos definido la tabla nos toca hacer lo mismo con las filas. Cada fila se define con una etiqueta TR, que tiene los siguientes atributos:

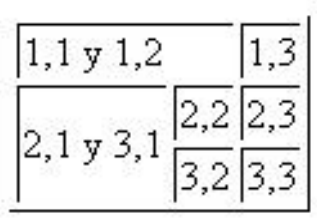
Parámetro	Utilidad
ALIGN	Alinea el contenido de las celdas de la fila horizontalmente a izquierda (LEFT), derecha (RIGHT) o centro (CENTER).
VALIGN	Alinea el contenido de las celdas de la fila verticalmente arriba (TOP), abajo (BOTTOM) o centro (MIDDLE).

Definir las celdas

Por último, nos queda definir cada celda gracias a la etiquetas TD y TH. Estas etiquetas son equivalentes, pero la última se utiliza para encabezados, de modo que su interior se escribirá por defecto en negrita y centrado. Estos son los atributos de ambas:

Parámetro	Utilidad
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 píxeles 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.

Posiblemente los dos últimos parámetros no puedan quedar claros sin ejemplos. De hecho, aún entendiendo perfectamente su función es habitual que confundamos a uno con otro. Pero bueno, vamos a ver una tabla de 3x3 con una celda que se une a una de la derecha y otra que se une a otra de debajo:

Código	Resultado
<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>	

Como podemos ver, cuando declaramos un celda con ROWSPAN o COLSPAN, no deberemos declarar las celdas "absorbidas" o la creación de la tabla se nos complicará de horrible manera.

Título de la tabla

Por último, vamos a ver como definir un título a la tabla. Esto se hace por medio de la etiqueta CAPTION. Para ver cómo funciona, vamos a incluirlo en la declaración de la tabla anterior:

Código	Resultado
<TABLE BORDER=1> <CAPTION> Ejemplo de tablas </CAPTION> … </TABLE>	Ejemplo de tablas

Esta etiqueta admite sólo un parámetro: ALIGN, que es por defecto TOP. Si lo definimos como BOTTOM el título se colocará al final de la tabla en lugar del comienzo.

Marcos

Un marco (o *frame*) es una ventana independiente dentro de la ventana general del navegador. Cada marco tendrá sus bordes y sus propias barras de desplazamiento. Así cada página se dividirá en la práctica en varias páginas independientes.

Para crearlos necesitaremos un documento HTML específico, que lamaremos documento de definición de marcos. En él especificaremos el tamaño y posición de cada marco y el documento HTML que contendrá. Vamos a ver un ejemplo de este tipo de documento:

```
<HTML>
<HEAD>
        <TITLE>Mi primera página con marcos</TITLE>
</HEAD>
        <FRAMESET COLS="20%,80%">
        <FRAME NAME="indice" SRC="indice.html">
        <FRAME NAME="principal" SRC="introduccion.html">
        <NOFRAMES>
                <P>Lo siento, pero sólo podrás ver esta página
                si tu navegador tiene la capacidad de visualizar
                marcos.</P>
        </NOFRAMES>
        </FRAMESET>
</HTML>
```

Vamos a explicar detalladamente este ejemplo antes de investigar algo más a fondo cada una de las etiquetas. Vemos que la cabecera de la página es similar a un documento normal, pero el habitual BODY es sustituido por un FRAMESET. En cada FRAMESET se divide la ventana actual (sea la general o un marco) en varias ventanas definidas o por el parámetro COLS o por ROWS. En éste, separado por comas, se define el número de marcos y el tamaño de cada uno.

Dentro del <FRAMESET> se hacen dos cosas. Primero, definir cada uno de los marcos poniéndoles un nombre y especificando qué fichero HTML le corresponde

mediante la etiqueta <FRAME>. Por último, especificamos lo que verá el usuario en el supuesto (cada vez más raro) de que su navegador no soporte *frames* dentro de la etiqueta <NOFRAMES>. Ahora veremos todos estos elementos en mayor detalle.

Etiqueta <FRAMESET>

Según el estándar, esta etiqueta sólo debería contener el número y tamaño de cada marco, pero las extensiones de Netscape y Explorer al estándar obligan a estudiar un par de parámetros más.

En general, los navegadores dibujan un borde de separación entre los marcos. Si deseas eliminarlo puedes hacerlo de dos maneras: en las etiquetas <FRAME> de cada una de los marcos contiguos al borde a eliminar o incluyendo el parámetro FRAMEBORDER=0 en el <FRAMESET>.
Cuando eliminas ese borde, podrás ver cómo el navegador deja aún un hueco entre marcos. Este se elimina añadiendo los parámetros FRAMESPACING=0 BORDER=0.

Vamos a examinar por último bs parámetros COLS y ROWS. Deberemos asignarles una lista de tamaños separada por comas. Se admiten los siguientes formatos de tamaño:

1. Con porcentajes: Al igual que con las tablas, podemos definir el tamaño de un marco como un porcentaje del espacio total disponible.
2. Absolutos: Si ponemos un número a secas, el marco correspondiente tendrá el tamaño especificado en píxeles.
3. Sobre el espacio sobrante: Si colocamos un asterisco (*) estaremos indicando que queremos todo el espacio sobrante para ese marco. Podemos poner este símbolo en varios marcos, que se repartirán el espacio equitativamente como buenos hermanos. Si queremos que uno tenga más deberemos ponerle al asterisco un número delante. Así, un marco con un espacio de 3* será tres veces más grande que su compañero, que tiene un asterisco sólo, el pobre.

Por ejemplo, el siguiente código es una muestra de cómo combinar los tres métodos:

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

Supongamos que el ancho total de la ventana son 640 píxeles. El primer marco ocupará el 10%, es decir, 64 píxeles. 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 píxeles para este último y 316 para el cuarto marco.

Hay que tener cuidado cuando usamos valores absolutos en la definición de marcos; debemos asegurarnos de tener al menos un marco con un tamaño relativo si queremos estar seguros del aspecto final de la página.

Por último, indicar que las etiquetas <FRAMESET> se pueden anidar. Esto se hace poniendo otro <FRAMESET> donde normalmente colocamos las etiquetas

<FRAME> tal que así:

```
<FRAMESET COLS="20%,80%">
<FRAME NAME="indice" SRC="indice.html">
<FRAMESET ROWS="*,80">
<FRAME NAME="principal" SRC="introduccion.html">
<FRAME NAME="ejemplos" SRC="ejemplo.html">
</FRAMESET>
</FRAMESET>
```

Etiqueta <FRAME>

Esta etiqueta define tan sólo las características de un marco determinado, no de un conjunto de ellos. Estos son los parámetros que admite:

Parámetro	Utilidad
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 píxeles.
MARGINHEIGHT	Igual al anterior pero con márgenes verticales.

Acceso a otros marcos

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:

_top
: Elimina todos los marcos existente y muestra la nueva página en la ventana original sin marcos.

_blank
: Muestra la nueva página en una ventana nueva y sin nombre del navegador.

_self
: Muestra la nueva página en el marco donde está declarado el enlace.

_parent
: 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.

Hojas de estilo

Las hojas de estilo intentan separar el contenido de un página de la forma de presentarlo en pantalla. Esto lo hacen personificando los cambios que las etiquetas de formato HTML realizan en nuestro texto. Por ejemplo, sabemos que usando <P> tendremos una párrafo nuevo con una separación del anterior determinada, etc.. Con las hojas de estilo también podremos decirle a un párrafo que todo su texto sea verde y que además va a tener un margen izquierdo de 30 píxeles. Por ejemplo.
El primer navegador en soportar hojas de estilo fue el Explorer 3.0. Utiliza la llamada sintaxis en cascada. El Communicator acepta esa sintaxis e introduce una nueva basada en JavaScript. Sin embargo, como el estándar más comúnmente aceptado de sintaxis de hojas de estilo es el de cascada, será este el único que veamos. Vamos a empezar con un ejemplo:

```
<STYLE TYPE="text/css">
        P {color: green; margin-left: 30;}
</STYLE>
```

Vamos a detenernos en todos los elementos. Para empezar, la etiqueta HTML con la que deberemos englobar las hojas de estilo será <STYLE>, que sólo podrá estar situada en la cabecera de la página. Su parámetro TYPE indica la sintaxis que utilizaremos para definirlas. En el caso de las hojas en cascada deberá ser text/css.

Ahora examinemos la hoja de estilo propiamente dicha. Indicaremos primero la etiqueta que deseamos personalizar. Será <P>. Después, entre llaves, pondremos una lista de las cosas que queremos modificar. En nuestro caso son dos, el color (en el formato habitual) y el margen, que se define en píxeles.

Hay que destacar que, aunque muchas veces los navegadores tengan una cierta manga ancha, la sintaxis CSS (Cascading Style Sheet) es sensible a la diferencia entre mayúsculas y minúsculas. Conviene tener cuidado.

El HTML 4.0 permite declarar fuera de la página las hojas de estilo, llevando de este modo al extremos su filosofía de separar forma y contenido. Si colocamos nuestras hojas de estilo en un fichero llamado estilos.css (solo las hojas, no la etiqueta <STYLE>) no tendremos más que incluir la siguiente línea en la cabecera de nuestro documento HTML para incluirlas en nuestras páginas:

```
<LINK REL="stylesheet" HREF="estilos.css" TYPE="text/css">
```

Clases

Hasta ahora habíamos definido estilos para una etiqueta determinada. Pero también podemos hacerlo para una clase determinada. ¿Esto que significa? Pues que si, por ejemplo, definimos la hoja de estilo de la clase verde de la siguiente manera:

```
P.verde {color: green; margin-left: 30px;}
```

sólo estarán verdes y con un margen izquierdo de 30 píxeles aquellos párrafos definidos con <P CLASS="verde">. Así de sencillo.

Ampliando un poco más las posibilidades de las clases, se pueden realizar excepciones. Supongamos que tenemos unos párrafos que queremos que tengan unos márgenes determinados y color verde. Y deseamos que uno solo de esos párrafos, con los mismo márgenes, sea azul. Podríamos definir dos clases distintas, pero hay un método mejor: usar el parámetro ID. Por ejemplo:

```
all.verde {color: green; margin-left: 10px;}
#ej1 { color: blue;}
```

Ahora todos los párrafos de clase verde serían, obviamente, verdes y con un margen de 10 píxeles. Sin embargo el párrafo definido por <P CLASS="verde" ID="ej1"> será azul:

Etiquetas <SPAN> y <DIV>

Puede que, a veces, no queramos modificar el comportamiento de un elemento sino que creemos un estilo que queremos actúe sólo, un estilo completo creado de la nada. Una etiqueta nueva y propia. Entonces, ¿qué hacemos? Utilizar las etiqueta <SPAN> y <DIV>.

El método es simple. Definimos una clase rojo que simplemente modifique el color (que será rojo). Ahora, si queremos que una sección de texto esté en rojo lo encerraremos entre las etiquetas <SPAN CLASS="rojo"> y </SPAN> o entre <DIV CLASS="rojo"> y </DIV>.

La diferencia entre ambas es que, mientras SPAN realmente no hace nada por sí misma, DIV convierte a todo lo que encierra en un bloque aparte (poniendo un salto de línea tanto al comienzo como al final). Veremos en el siguiente capítulo que a las etiquetas que se comportan como bloques (<P>, <H1>, las que dijimos modifican un párrafo entero) se les pueden definir atributos propios desde las hojas de estilo. Por ejemplo, si definimos las siguientes hojas:

```
all.titulo {
  margin-top: -24px;
  color: blue;
  font-size: 20px;
}

all.sombra {
  margin-top: 2px;
  margin-left: 2px;
  color: black;
  font-size: 20px;
}
```

cuyos atributos explicaremos en la referencia de las hojas de estilo, y ponemos el siguiente código HTML:

```
<DIV ALIGN="CENTER" CLASS="sombra">El maravilloso curso de HTML</DIV>
<DIV ALIGN="CENTER" CLASS="titulo">El maravilloso curso de HTML</DIV>
```

obtendremos este bello efecto:

El maravilloso curso de HTML

El maravilloso curso de HTML

En el siguiente capítulo tenéis una guía de referencia con todos (o casi todos) los atributos que se pueden modificar con CSS.

Hojas de estilo: referencia

En este capítulo vamos a ver todas (o casi todas) las propiedades que se pueden alterar por medio de las hojas de estilo. Hay que indicar que algunos de ellos no los soporta el Explorer y en cambio otros no los entiende el Communicator. Así que es recomendable probarlos en ambos exploradores antes de incorporarlos a nuestras páginas.

Propiedades de bloque

Vamos a empezar con las propiedades de bloque, que definen cosas como los márgenes o la colocación de bloques de contenido HTML:

Propiedad	Descripción	Posibles valores
margin-top, margin-right, margin-bottom, margin-left, margin: top right bottom left	Distancia mínima entre un bloque y los demás elementos. Tanto margin como margins() se utilizan para cambiar todos estos atributos a la vez.	tamaño, porcentaje o auto. Por defecto es cero.
padding-top, padding-right, padding-bottom, padding-left, padding: top right bottom left	Distancia entre el borde y el contenido de un bloque.	tamaño, porcentaje o auto. Por defecto es cero.
border-top-width, border-right-width, border-bottom-width, border-left-width, border-width: top right bottom left	Anchura del borde de un bloque.	numérico
border-style	Estilo del borde de un bloque.	none, solid o 3D, por defecto ninguno (none).
border-color	Color del borde de un bloque.	cualquier color
width, height	Tamaño de un bloque. Su mayor utilidad está en su aplicación a un elemento gráfico.	tamaño, porcentaje o auto, automático por defecto.
float	Justificación del contenido de un bloque.	left, right o none, por defecto ninguna.
clear	Permiso para que otro elemento se pueda colocar a su izquierda o derecha.	left, right, both o none, por defecto ninguno.

Propiedades de tipo de letra

Ahora vamos a examinar las propiedades del tipo de letra que el usuario va a ver. Son las siguientes:

Propiedad	Descripción	Posibles valores
font-family	Tipo de letra (que puede ser genérico) que vamos a usar.	lista de tipos, ya sean genéricos o no, separados por comas.
font-size	Tamaño del tipo de letra.	xx-small, x-small, small, medium, large, x-large, xx-large, tamaño relativo o tamaño absoluto. Por defecto medium.
font-weight	Grosor del tipo de letra (negrita).	normal, bold, bolder, lighter o 100-900 (donde 900 es la negrita más gruesa). Por defecto normal.
font-style	Estilo del tipo de letra (cursiva).	normal, italic, italic small-caps, oblique, oblique small-caps o small-caps. Por defecto normal.

Cabe recordar que los tipos genéricos son serif, sans-serif, cursive, fantasy y monospace. Cada uno de estos tipos serán equivalentes a alguno que pueda tener instalado el ordenador del usuario. Así, por ejemplo, en un PC con Windows instalado serif puede equivaler a Times New Roman y monospace a Courier.

Propiedades de formato del texto

Nuesto siguiente objetivo van a ser las propiedades de formato del texto que cualquier procesador de textos nos permite cambiar.

Propiedad	Descripción	Posibles valores
line-height	Interlineado.	número o porcentaje.
text-decoration	Efectos variados sobre el texto.	none, underline (subrayado), overline (como subrayado, pero por encima), line-through (tachado) o blink (parpadeante); por defecto ninguno.
vertical-align	Posición vertical del texto.	baseline (normal), sub (subíndice), super (superíndice), top, text-top, middle, bottom, text-bottom o un porcentaje. Por defecto baseline
text-transform	Transforma el texto a mayúsculas o minúsculas.	capitalize (pone la primera letra en mayúsculas), uppercase (convierte todo a mayúsculas), lowecase (a minúsculas) o none, por defecto no hace nada.
text-align	Justificación del texto.	left, right, center o justify
text-indent	Tabulación con que aparece la primera línea del texto.	tamaño o porcentaje, por defecto cero.

Propiedades de color y fondo

También es posible cambiar los colores y el gráfico de fondo de un elemento.

Propiedad	Descripción	Posibles valores
color	Color del texto.	un color (en el formato habitual).
background	Modifica tanto el gráfico el color de fondo.	dirección del fichero que contiene la imagen o un color.

Hay que decir que, en la sintaxis en cascada, las direcciones se expresan del siguiente modo:

background: url(fondobonito.gif);

Propiedades de clasificación

Hasta ahora habíamos distinguido a la hora de ver las propiedades de un elemento en si estos eran tratados como bloques o no. Pero el ser bloques o no... ¿no es acaso otra propiedad? Estas y otras formas de clasificar los elementos se pueden cambiar usando las siguientes propiedades:

Propiedad	Descripción	Posibles valores
display	Decide si un elemento es interior (como <I>), un bloque (<P>) o un elemento de una lista (<LI>).	inline, block, list y none (que 'apaga' el elemento)
list-style	Estilo de un elemento de una lista, pudiendo incluir un gráfico al comienzo del mismo.	disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none o la dirección de un gráfico
white-space	Decide como se manejan los espacios, si de manera normal o como sucede dentro de la etiqueta <PRE>.	normal y pre

INDICE GENERAL