



Lenguaxe Html

Marcas registradas

Todos los nombres de programas, sistemas operativos, equipos hardware, etc. que aparecen en este documento son marcas registradas de sus respectivas compañías u organizaciones.

Aviso

No está permitida la reproducción total o parcial de este documento, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio, ya sea electrónico, mecánico, por fotocopia u otros métodos, sin permiso previo y por escrito del Instituto Galego de Formación.

Instituto Galego de Formación (Vigo)
C/ Zamora 13A (esquina Ramón y Cajal)
36.203 – Vigo
Teléfono: 986 48 52 52

Instituto Galego de Formación (O Burgo)
Rúa Os Templarios 16 Bajo (Frente al Puente del Burgo)
15.670 – O Burgo
Teléfono: 981 65 21 18

www.igf.es

Instituto Galego de Formación 1996 - 2001.

Contenido

1. Una página básica.....	5
2. Dando forma al texto.....	7
	7
<P>.....	7
	7
 <I> <U>.....	7
<PRE>.....	7
<TT>.....	7
<BLOCKQUOTE>.....	8
<SUP> Y <SUB>.....	8
.....	8
<!--.....	9
3. Enlaces con otras páginas.....	10
Enlaces dentro de la misma página.....	10
Enlaces con otra página nuestra.....	10
3. Enlaces con una página fuera de nuestro sistema.....	11
4. Enlaces con una dirección de email.....	11
4. Imágenes.....	12
.....	12
ALT.....	12
ALIGN.....	12
BORDER.....	12
5. Caracteres especiales.....	14
6. Los estándares del HTML.....	15
HTML 3.0.....	15
HTML 3.2.....	15
HTML 4.0.....	15
Los estándares y este manual.....	15
7. Fondos.....	16
Fondos con un color uniforme.....	16
Colores del texto y de los enlaces.....	17
Fondos con una imagen.....	17
8. Alineación y dimensionado de imágenes.....	18
Alineación de las imágenes.....	18
9. Formato de las imágenes.....	19
Tamaño de las imágenes.....	19
El formato JPEG.....	19
GIFs transparentes.....	20
GIFs entrelazados.....	20
10. Tablas.....	21
Estructura de una tabla.....	21
<TABLE>.....	21
<TR>.....	21

<TD>	21
Filas con desigual número de celdas	22
Titular de la tabla.....	22
Variando el espesor de los bordes	22
Celdas de cabecera.....	22
Contenido de las celdas	22
Posicionamiento del contenido dentro de la celda.....	22
Variando las dimensiones de la tabla	23
Celdas que abarcan a otras varias	23
Color de fondo en las tablas	23
Imágenes de fondo en las tablas	23
Separación entre las celdas de una tabla.....	24
Separación entre el borde y el contenido dentro de las celdas	24
11. Formularios.....	25
Estructura de un formulario	25
Etiqueta de inicio	25
Elementos para introducir los datos.....	26
Introducción por medio de texto (una línea)	26
Introducción por medio de texto (múltiples líneas).....	27
Introducción por medio de menús	27
Formulario de confirmación (checkbox).....	28
Botones de radio	28
Botones de envío y de borrado	28
12. GIFs animados	30
Programas para crear animaciones	30
13. Mapas.....	31
Mapas gestionados por el servidor	31
Mapas gestionados por el cliente.....	31
Confección de la imagen del mapa.....	32
Código HTML para definir un mapa gestionado por el cliente.....	32
14. Frames	34
Documento de definición de los frames	34
Documentos HTML de cada frame	35
Alternativa para los navegadores que no soportan frames	36
<FRAMESET>.....	37
Frames sin bordes	38
<FRAME>.....	38
TARGET	39
Frames anidados dentro de otros frames	40
15. Sonidos.....	41
Capacidades sonoras de los navegadores	41
Explorer de Microsoft.....	41
Netscape.....	41
Fondo sonoro para el Microsoft Internet Explorer	41
Fondo sonoro para el Netscape.....	42
Fondo sonoro combinado para el Explorer y el Netscape	42
Activación del sonido por el propio usuario.....	43

16. Texto en movimiento	44
Marquesinas (Marquees)	44
Texto en movimiento con Javascript	46
Cómo incluir Javascript en un documento HTML	46
Scroll en la barra de estado	47
17. Publicación y promoción de una página.....	48
Publicación de una página	48
¿Cómo se envían las páginas al servidor?	48
Promoción de una página	48
Consejos para facilitar la labor a los motores de búsqueda	49
Sistemas de búsqueda en español	50
Sistemas de búsqueda en inglés	50
Registro en buscadores que están organizados como directorios	51
18. Contadores de visitas.....	52
¿Cómo funciona un contador?	52
Servidores públicos de contadores	52
19. Componer con estilo	53
Estructura de una página.....	53
Imágenes	55
Frames	56
Nombres de los ficheros	57
Sonidos	57
20. Applets de Java	59
Java y Javascript	59
Nivel de dificultad del lenguaje Java.....	59
¿Qué son los applets?	59
¿Para qué sirven los applets?.....	60
¿Cómo se coloca un applet en una página?	60
21. Trucos diversos	63
Distribución de programas desde una página del Web.....	63
Texto fijo de un enlace en la barra de estado	63
Salto automático de pantalla.....	63
Botón de envío de un formulario con una imagen.....	64
Menú de direcciones.....	65
Acceso a una página por medio de un password	65
22. Canales de chat en páginas Web	67
¿Cómo se instala un canal de chat en una página del Web?.....	67
23. Hojas de estilo en cascada.....	68
Ejemplos de estas técnicas.....	68
¿Qué navegadores las soportan?.....	69
¿Cómo funcionan?.....	69
Estilo para etiquetas concretas o grupos de etiquetas.....	70
Inclusión global del estilo en un documento HTML.....	71
Enlazando distintos documentos a una hoja de estilo.....	72
Mezclando los métodos	72
24. Hojas de estilo en cascada.....	74
Atributos	74

25. Hojas de estilo en cascada.....	78
Consejos de utilización.....	78
23. Enlaces interesantes.....	82

1. Una página básica

El principio esencial del lenguaje HTML (HyperText Markup Language) es el uso de las etiquetas (tags). Funcionan de la siguiente manera:

```
<XXX> Este es el inicio de una etiqueta.  
</XXX> Este es el cierre de una etiqueta.
```

Las letras de la etiqueta pueden estar en mayúsculas o minúsculas, indiferentemente. Por claridad, se usarán en este manual sólo las mayúsculas

Lo que haya entre ambas etiquetas estará influenciada por ellas. Por ejemplo, todo el documento HTML debe estar entre las etiquetas <HTML> y </HTML>:

```
<HTML> [Todo el documento] </HTML>
```

El documento en sí está dividido en dos zonas principales:

El encabezamiento, comprendido entre las etiquetas <HEAD> y </HEAD>
El cuerpo, comprendido entre las etiquetas <BODY> y </BODY>

Dentro del encabezamiento hay información del documento, que no se ve en la pantalla principal, principalmente el título del documento, comprendido entre las etiquetas <TITLE> y </TITLE>. El título debe ser breve y descriptivo de su contenido, pues será lo que vean los demás cuando añadan nuestra página a su bookmark (o agenda de direcciones).

Dentro del cuerpo está todo lo que queremos que aparezca en la pantalla principal (texto, imágenes, etc.)

Por tanto, la estructura queda de esta manera:

```
<HTML>  
<HEAD>  
<TITLE> Título de la página </TITLE>  
</HEAD>  
<BODY>  
[Aquí van las etiquetas que visualizan la página]  
</BODY>  
</HTML>
```

Antes de crear nuestra primera página, unas consideraciones sobre el texto: Cuando escribimos en el documento el texto que queremos que aparezca en la pantalla, veremos que éste se acomoda a ella, sin que tengamos que pulsar el retorno del carro. Si queremos separar el texto en distintos párrafos debemos usar la etiqueta <P>, (que no tiene su correspondiente etiqueta de cierre </P>)

El texto puede tener unas cabeceras, comprendidas entre las etiquetas <H1> y </H1>, <H2> y </H2>, etc. (hasta el número 6), siendo el número indicativo del tamaño. El tamaño mayor es el correspondiente al número 1. Puedes experimentar en el ejemplo que sigue, cambiando el número para comprobar el efecto que se logra.

Una etiqueta muy interesante es la de centrado <CENTER> y </CENTER> (no la soportan todos los navegadores, aunque sí la mayoría de ellos). Nos centra todo lo que esté dentro de ella, ya sea texto, imágenes, etc.

También tenemos los separadores (horizontal rules), que se consiguen con la etiqueta <HR> (no existe la correspondiente de cierre). Con ella se obtiene una raya horizontal tan ancha como la pantalla, y con la apariencia de estar embutida sobre el fondo, como se puede observar a continuación:

Ejemplo práctico

En el procesador de texto copiamos lo siguiente:

```
<HTML>
<HEAD>
  <TITLE> Mi pagina del Web - 1 </TITLE>
</HEAD>
<BODY>
  <H1> <CENTER> Primera pagina </CENTER> </H1>
  <HR>
  Esta es mi primera pagina, aunque todavia es muy sencilla. Como el
  lenguaje HTML no es dificil, pronto estare en condiciones de hacer
  cosas mas interesantes.
  <P> Aqui va un segundo parrafo.
</BODY>
</HTML>
```


2. Dando forma al texto

Como hemos visto en el ejemplo del capítulo anterior, cuando queremos poner un texto sin ninguna característica especial, lo ponemos directamente. Únicamente, la separación entre párrafos (dejando una línea en blanco) la conseguimos con la etiqueta `<P>`.

`
`

Si queremos separar los párrafos, o cualquier otra cosa, pero sin dejar una línea en blanco, usamos una etiqueta parecida `
` (break, o romper). Tampoco tiene etiqueta de cierre.

`<P>`

Si queremos obtener múltiples líneas en blanco no basta con repetir la etiqueta `<P>`, sino que hay que combinarla con la etiqueta `
`. Así por ejemplo, si queremos obtener cuatro líneas en blanco, pondríamos:

```
<BR><P>
<BR><P>
<BR><P>
<BR><P>
```

` `

Al escribir el texto, si ponemos más de un espacio en blanco entre dos palabras observamos que el navegador sólo reconoce uno de ellos. Si queremos forzarle a que lo haga, debemos poner el código "` `" (non-breaking space).

`` `` `<I>` `</I>` `<U>`

Para destacar alguna parte del texto se pueden usar:

```
<B> y </B> para poner algo en negrita (bold).
<I> y </I> para poner algo en cursiva (italic).
<U> y </U> para subrayar algo (underline).
```

`<PRE>`

Otra etiqueta interesante es `<PRE>` y `</PRE>`. El texto que se encuentre entre ella estará preformateado, es decir que aparecerá como si hubiera sido escrito con una máquina de escribir, con una fuente de espaciado fijo (tipo Courier). Además se respetarán los espacios en blanco y retornos del carro, tal como estaban en nuestro documento HTML (lo cual no ocurre normalmente, como hemos visto anteriormente). Es muy apropiada para confeccionar tablas y otros documentos similares.

`<TT>`

Con la etiqueta `<TT>` y `</TT>` conseguimos también que el texto tenga un tamaño menor y la apariencia de los caracteres de una máquina de escribir (typewriter). La diferencia con la anterior es que no preformatea el texto, sino que únicamente cambia su apariencia.

<BLOCKQUOTE>

La etiqueta <BLOCKQUOTE> y </BLOCKQUOTE> se utiliza para destacar una cita textual dentro del texto general.

<SUP> Y <SUB>

En las fórmulas matemáticas puede interesar poder escribir índices y subíndices, que se consiguen con las etiquetas y respectivamente.

A menudo nos interesará presentar las cosas en forma de listas. Podemos escoger entre tres tipos distintos:

- 1.Listas desordenadas (no numeradas)
- 2.Listas ordenadas (numeradas)
- 3.Listas de definición.

```
<UL>
<LI> Una cosa
<LI> Otra cosa
<LI> Otra más
<LI> Etc.
</UL>
```

```
<UL>
<LI> Mamíferos
<LI> Peces
  <UL>
  <LI> Sardina
  <LI> Bacalao
  </UL>
<LI> Aves
</UL>
```

Las listas ordenadas (ordered lists) sirven para presentar cosas en un orden determinado. Su estructura es muy similar a la anterior. La diferencia estriba en que en el resultado aparecerá automáticamente un número correlativo para cada cosa.

```
<OL>
<LI> Primera cosa
<LI> Segunda cosa
<LI> Tercera cosa
<LI> Etc.
</OL>
```

Al igual que las listas desordenadas, también se pueden anidar las listas ordenadas.

<!--

Comentarios no visibles en la pantalla

A veces es muy útil escribir comentarios en el documento HTML sobre el código que escribimos, que nos pueden servir para recordar posteriormente sobre lo que hicimos, y que no queremos que se vean en pantalla.

Esto se consigue encerrando dichos comentarios entre estos dos símbolos: <!-- y -->

Ejemplo:

<!-- Esto es un comentario al código que no se verá en pantalla -->

Ejemplo

```
<HTML>
<HEAD>
<TITLE> Mi pagina del Web - 2 </TITLE>
</HEAD>
<BODY>
<CENTER>
<H1> Mis aficiones </H1>
</CENTER>
<HR>
Sin un orden particular, mis <B> aficiones </B> son las siguientes:
<UL>
<LI> El cine
<LI> El deporte
</UL>
<LI> Natacion
<LI> Baloncesto
</UL>
<LI> La musica
</UL>
La musica que mas me gusta es <I> (en orden de preferencia): </I>
<OL>
<LI> El rock
<LI> El jazz
<LI> La musica clasica
</OL>
</BODY>
</HTML>
```

3. Enlaces con otras páginas

La característica que más ha influido en el espectacular éxito del Web (o tela de araña) ha sido, aparte la de su carácter multimedia, la posibilidad de unir los distintos documentos repartidos por todo el mundo por medio de enlaces hipertexto.

En general, los enlaces tienen la siguiente estructura:

```
<A HREF="enlace"> texto indicativo </A>
```

Tipos de enlaces

Vamos a distinguir cuatro tipos de enlaces:

1. Enlaces dentro de la misma página
2. Enlaces con otra página nuestra
3. Enlaces con una página fuera de nuestro sistema
4. Enlaces con una dirección de email

Enlaces dentro de la misma página

A veces, en el caso de documentos (o páginas) muy extensos, nos puede interesar dar un salto desde una posición a otra determinada.

En este caso, lo que antes hemos llamado XXX, es decir, el destino del enlace, en este caso el sitio dentro de la página a donde queremos saltar, se sustituye por #marca (la palabra marca puede ser cualquier palabra que queramos). Lo que hemos llamado antes YYY es la palabra (o palabras) que aparecerán en la pantalla en color (en forma de hipertexto). Su estructura es, entonces:

```
<A HREF="#marca"> YYY </A>
```

Y en el sitio exacto a donde queremos saltar, debemos poner la siguiente etiqueta:

```
<A NAME="marca"> </A>
```

Enlaces con otra página nuestra

Puede ser que tengamos una sola página. Pero lo más frecuente es que tengamos varias páginas, una inicial (o principal) y otras conectadas a ella, e incluso entre ellas mismas.

Supongamos que queremos enlazar con la página creada en el ejemplo del capítulo anterior, que la hemos llamado mipag2.html. En este caso, simplemente sustituimos lo que hemos llamado XXX (el destino del enlace) por el nombre del fichero:

```
<A HREF="pag.html"> capítulo 12 </A>
```

Si queremos que vaya a un sitio concreto de otra página nuestra en vez de ir al principio de la página, adonde va por defecto, en ese sitio tenemos que colocar una marca y completar el enlace con la referencia a esa marca.

Un sitio concreto, donde he puesto la marca ``. Entonces la etiqueta tiene que ser: ` Capitulo 4 ` Que resulta: Capitulo 4

3. **Enlaces con una página fuera de nuestro sistema**

Si queremos enlazar con una página que esté fuera de nuestro sistema (es decir, que esté en un servidor distinto al que soporta nuestra página), es necesario conocer su dirección completa, o URL (Uniform Resource Locator). El URL podría ser, además de la dirección de una página del Web, una dirección de ftp, gopher, etc.

Una vez conocida la dirección (o URL), lo colocamos en vez de lo que hemos llamado anteriormente xxx (el destino del enlace). Si queremos enlazar con la página del Instituto Galego de Formación (cuyo URL es: `http://www.igf.es/`), la etiqueta sería:

` Página inicial del Instituto Galego de Formación `

4. **Enlaces con una dirección de email**

En este caso, sustituimos el destino del enlace por `mailto:` seguido de la dirección de email. La estructura de la etiqueta es:

` Texto del enlace `

Un ejemplo de esto está al final de la página principal de este manual. Podría haber puesto:

Comentarios a ` Instituto Galego de Formación`

Ejemplo

```
<HTML>
<HEAD>
<TITLE> Mi pagina del Web - 3 </TITLE>
</HEAD>
<BODY>
<CENTER>
<H1> Mis paginas favoritas </H1>
</CENTER>
<HR>
Estas son algunas paginas favoritas:
<P><A HREF="http://home.netscape.com"> Netscape </A>
<BR> <A HREF="http://www.microsoft.com"> Microsoft </A>
<BR> <A HREF="http://www.yahoo.com"> Yahoo! </A>
<BR> <A HREF="http://www.igf.es"> Instituto Galego de Formación</A>
</BODY>
</HTML>
```

4. Imágenes

La etiqueta que nos sirve para incluir imágenes en nuestras páginas del Web es muy similar a la de enlaces a otras páginas, que hemos visto en el capítulo anterior. La única diferencia es que, en lugar de indicar al programa navegador el nombre y la localización de un documento de texto HTML para que lo cargue, se le indica el nombre y la localización de un fichero que contiene una imagen.

Con el comando IMG SRC (image source, fuente de la imagen) se indica que se quiere cargar una imagen llamada imagen.gif (o el nombre que tenga). La estructura de la etiqueta es:

```
<IMG SRC="imagen.gif">
```

ALT

Dentro de la etiqueta se pueden añadir otros comandos, tal como ALT

```
<IMG SRC="imagen.gif" ALT="descripción">
```

Con el comando ALT se introduce una descripción (una palabra o una frase breve) indicativa de la imagen.

ALIGN

Nos permite escoger la posición del titular con respecto a la imagen (si es que queremos ponerle un titular, claro está). Se puede poner arriba, en medio o abajo del lado de la imagen. Para ello se añade el comando ALIGN a la etiqueta, de la siguiente manera:

```
<IMG SRC="isla.gif" ALIGN=TOP> Titular alineado arriba  
<IMG SRC="isla.gif" ALIGN=MIDDLE> Titular alineado en medio  
<IMG SRC="isla.gif" ALIGN=BOTTOM> Titular alineado abajo
```

Ejemplo de cómo utilizar la imagen (logotipo.gif) para acceder a la página del Instituto Galego de Formación.

```
<A HREF="http://www.igf.es"><IMG SRC="logotipo.gif"></A>
```

BORDER

Si no se desea que aparezca un rectángulo alrededor de la imagen, hay que incluir dentro de la etiqueta de la imagen el atributo BORDER=0, es decir:

```
<A HREF="mipag2.html"><IMG SRC="hombre.gif" BORDER=0></A>
```

Ejemplo

Primeramente es necesario capturar las tres imágenes que se han utilizado en este capítulo: isla.gif, hombre.gif y casa.gif y guardarlas en el mismo directorio en el que se guardará el fichero de texto que se va a crear a continuación.

En el procesador de textos copiamos:

```
<HTML>
<HEAD>
<TITLE> Mi pagina del Web - 4 </TITLE>
</HEAD>
<BODY>
<CENTER>
<H1> Mi pagina del Web </H1>
</CENTER>
<HR>
Esta es mi pagina del Web. No es muy extensa, pero tiene todos los
elementos basicos. Espero que os guste. Poco a poco le ire anadiendo
mas cosas interesantes.
<P> <A HREF="mipag2.html"> <IMG SRC="hombre.gif"> </A> Mis aficiones
<P> <A HREF="mipag3.html"> <IMG SRC="casa.gif"> </A> Mis paginas favoritas
<CENTER>
<H3> Un lugar ideal para mis vacaciones </H3>
<IMG SRC="isla.gif" ALT="isla">
</CENTER>
</BODY>
</HTML>
```

5. Caracteres especiales

Existen algunas limitaciones para escribir el texto. Una de ellas es debido a que las etiquetas se forman como un comando escrito entre los símbolos "<" y ">". Por tanto, si se quisieran escribir estos caracteres como parte normal del texto, daría esto lugar a una ambigüedad, ya que el programa navegador podría interpretarlos como el comienzo o final de una etiqueta, en vez de un carácter más del texto.

Para resolver este problema, existen unos códigos para poder escribir estos caracteres y otros relacionados con las etiquetas.

< para < (less than, menor que)
> para > (greater than, mayor que)
& para & (ampersand)
" para " (double quotation)
á para la á
é para la é
í para la í
ó para la ó
ú para la ú
Á para la Á
É para la É
Í para la Í
Ó para la Ó
Ú para la Ú
ñ para la ñ
Ñ para la Ñ
ü para la ü
Ü para la Ü
¿ para ¿
¡ para ¡

Como se ve, estos códigos empiezan siempre con el signo & y acaban siempre con ;

6. Los estándares del HTML

Hasta hace poco, la última versión oficial del HTML era la HTML 2.0. Cumplía su función perfectamente, pero muchos usuarios del HTML querían tener un control mayor sobre sus documentos, tanto en el formateo del texto como en la apariencia de la página.

HTML 3.0

Mientras tanto, Netscape, que era en esa época el líder indiscutible de los navegadores, introducía con cada nueva versión etiquetas y atributos no contemplados en el estándar oficial, con gran escándalo de algunos.

Debido a su amplia difusión y aceptación, otros navegadores intentaban copiar esas innovaciones, pero debido a que Netscape no especificaba completamente sus nuevas etiquetas, los resultados no eran siempre los deseados. Todo ello provocaba una gran confusión, y numerosos problemas, cuando los autores utilizaban estos elementos y comprobaban que no funcionaba como querían en otros navegadores.

HTML 3.2

Cada vez era más evidente la necesidad de un nuevo estándar que fuera aceptado por todos. Se formó otro comité, el W3C, apoyado por los principales vendedores de software (entre ellos, IBM, Microsoft, Novell, Netscape, Sun, etc.). El nuevo estándar ha sido desarrollado durante todo el año 1996 por el W3C con el sobrenombre de Wilbur, y finalmente, en Enero de 1997 ha sido aprobado como el HTML 3.2

El HTML 3.2 es totalmente compatible con el estándar anterior, pero incorpora muchas de las innovaciones de los navegadores comerciales (Netscape y Explorer principalmente), tales como tablas, applets, texto que bordea las imágenes, etc., aunque no todas, como por ejemplo los frames.

Se puede consultar la documentación oficial del HTML 3.2 en *Introducing HTML 3.2*, pero es más manejable y práctica la presentada en *Wilbur - HTML 3.2*, donde se pueden ver con detalle las distintas etiquetas, e incluso obtener una versión off-line del nuevo estándar.

HTML 4.0

En Julio de 1997 se ha presentado el primer borrador público del próximo estándar oficial HTML 4.0 (que tiene el nombre clave de Cougar). En él se incorporan las especificaciones sobre tablas, frames, scripts, hojas de estilo, accesibilidad por distintos medios, e internacionalización (incluyendo el uso de Unicode, en lugar del Latin-1).

Los estándares y este manual

En los capítulos anteriores a éste (HTML esencial) se han explicado, en líneas generales, las etiquetas que cumplían con la anterior norma HTML 2.0. En los capítulos que siguen se hablará de las etiquetas incorporadas al nuevo estándar, más algunas otras que no lo están, pero que son implementadas por los navegadores más importantes (Netscape y Explorer).

7. Fondos

Se puede cambiar el fondo de dos maneras distintas:

1. Con un color uniforme
2. Con una imagen

Fondos con un color uniforme

Se consigue añadiendo el comando BGCOLOR a la etiqueta <BODY> (situada al principio del documento), de la siguiente manera:

```
<BODY BGCOLOR="#XXYYZZ">
```

donde:

XX es un número indicativo de la cantidad de color rojo
YY es un número indicativo de la cantidad de color verde
ZZ es un número indicativo de la cantidad de color azul

Estos números están en numeración hexadecimal. Esta numeración se caracteriza por tener 16 dígitos (en lugar de los diez de la numeración decimal habitual). Estos dígitos son:

0 1 2 3 4 5 6 7 8 9 A B C D E F

Es decir, que en nuestro caso, el número menor es el 00 y el mayor el FF. Así, por ejemplo, el color rojo es el #FF0000, porque tiene el máximo de rojo y cero de los otros dos colores .

Los colores primarios son:

#FF0000 - Rojo
#00FF00 - Verde
#0000FF - Azul

Otros colores son:

#FFFFFF - Blanco
#000000 - Negro
#FFFF00 - Amarillo

Para hacer un color más oscuro, hay que reducir el número de su componente, dejando los otros dos invariables. Así, el rojo #FF0000 se puede hacer más oscuro con #AA0000, o aún más oscuro con #550000.

Para hacer que un color tenga un tono más suave (más pastel), se deben variar los otros dos colores haciéndolos más claros (número más alto), en una cantidad igual. Así, podemos convertir el rojo en rosa con #FF7070.

Colores del texto y de los enlaces

Si no se variasen los colores habituales del texto y de los enlaces (negro y azul, respectivamente), podría ocurrir que su lectura contra un fondo oscuro fuese muy dificultosa, o incluso imposible, si el fondo fuese precisamente negro o azul.

Para evitar esto, se pueden escoger los colores del texto y de los enlaces, añadiendo a la etiqueta (si se desea) los siguientes comandos:

TEXT - color del texto

LINK - color de los enlaces

VLINK - color de los enlaces visitados

ALINK - color de los enlaces activos (el que adquieren en el momento de ser pulsados)

Los códigos de los colores son los mismos que los que se han visto anteriormente.

La etiqueta, con todas sus posibilidades, sería:

```
< BODY BGCOLOR="#XXYYZZ" TEXT="#XXYYZZ" LINK="#XXYYZZ"
VLINK="#XXYYZZ" ALINK="#XXYYZZ">
```

Fondos con una imagen

El fondo de una página puede ser también una imagen, ya sea en formato GIF o JPEG. Esta imagen se repite por toda la página, de una manera análoga al tapiz de Windows. La estructura de la etiqueta es:

```
<BODY BACKGROUND="imagen.gif">
```

Se pueden añadir también a esta etiqueta todos los comandos para cambiar los colores del texto y de los enlaces, vistos anteriormente. Esto es imprescindible a veces para conseguir que el texto sea legible, en contraste con el fondo.

8. Alineación y dimensionado de imágenes

Alineación de las imágenes

La alineación de las imágenes fue, en su día, el primer golpe de efecto del programa Navigator de Netscape. Permitted alinear una imagen a la izquierda o a la derecha de la página y hacer que el texto la rodee completamente, consiguiéndose así una apariencia similar a la de una revista. Esto se consigue con las extensiones de la etiqueta ALIGN, que ya se vió en el Capítulo 4 con los comandos TOP, MIDDLE y BOTTOM.

9. Formato de las imágenes

Tamaño de las imágenes

Cuando vamos navegando por las páginas del Web parece a veces que todo se detiene completamente. Esto es algo que por desgracia ocurre muy a menudo. Pueden ser muchas las causas: conexiones de la red defectuosas, servidores lentos, etc. Son cosas que en general no nos queda más remedio que soportar.

Pero hay una causa particularmente irritante, y es que hayamos ido a parar a una página que padece del Síndrome de la Imagen Gigantesca, que es la mala costumbre de poner en la página imágenes enormes, lo que se traduce en ficheros de imagen a cargar con un tamaño excesivo.

Hay unas cuantas maneras de minimizar el tamaño de los ficheros de imagen:

Reducir el tamaño de la imagen con un programa gráfico. Esto hará que haya menos pixels que almacenar, y por tanto menos kilobytes que cargar.

Reducir el número de colores utilizados en la imagen. Una imagen GIF puede tener un máximo de 256 colores, pero podemos a veces reducir este número. Por ejemplo, se puede conseguir un logo atractivo para encabezar las páginas con sólo dos colores, uno de primer plano y el otro de fondo. Con un programa gráfico adecuado se pueden hacer estas comprobaciones y reducciones.

Simplificar las imágenes. El formato GIF comprime la imagen buscando las secuencias repetidas en los datos y abreviándolas. Esto quiere decir que las zonas grandes de un solo color se comprimen muy bien, al contrario de las que tienen muchos tonos y graduaciones.

Como referencia, el fichero de una imagen no debe sobrepasar los 75k, mejor aún más pequeño. Tampoco conviene que el tamaño de la imagen en la pantalla sea mayor de 500x400 pixels, para no obligar al usuario a tener que desplazarla con los cursores.

Si a pesar de todo, las imágenes que queremos poner siguen teniendo un tamaño de fichero demasiado grande, se puede elegir la solución, vista en el capítulo anterior, de utilizar thumbnails (reproducciones en pequeño que enlazan con la imagen grande). De esta manera es el usuario quien elige qué imágenes cargar.

El formato JPEG

El método de compresión utilizado por el formato GIF es muy conveniente para comprimir áreas monocolor, como las que puede haber por ejemplo en un diagrama sencillo. Pero no es tan conveniente para cosas más complicadas, como por ejemplo fotografías de objetos de la vida real. Además estos objetos no presentan un aspecto nada favorable si están reducidos a los 256 colores de este formato.

Para realizar este tipo de tareas se creó el formato JPEG. Almacena las imágenes con 16.7 millones de colores, que es mucho más de lo que el ojo humano puede distinguir, y hace uso de complicados algoritmos matemáticos para comprimir el tamaño de los ficheros hasta la décima parte de su tamaño original. Esto quiere decir que una

imagen complicada que se haya comprimido con este formato tendrá, comparando con el formato GIF, una calidad mayor y un tamaño de fichero menor.

Su único inconveniente es que hay navegadores que no son capaces de manejar directamente estos ficheros y que tienen que traspasarlos a un programa auxiliar para ejecutarlos.

GIFs transparentes

Una característica muy útil del formato GIF es la opción de hacer transparente un color determinado, es decir, que en la página del Web ese color concreto no se vea, siendo reemplazado por el fondo de la página. Por tanto, si en una imagen escogemos el color de su fondo como transparente, parecerá flotar sobre el fondo de la página (sea éste el que sea, incluso un fondo de imágenes)

Pero no todas las imágenes son apropiadas para conseguir este efecto. Es condición indispensable que la imagen tenga un fondo de color uniforme. No vale una imagen con un fondo multicolor.

Hay otro aspecto que hay que tener en cuenta y es que el formato GIF tiene dos subformatos diferentes. El más común es el GIF 87a, con el que no se pueden conseguir colores transparentes. El formato adecuado para ello es el GIF 89a. Por tanto, el programa gráfico debe tener la posibilidad de convertir las imágenes a este subformato.

El último aspecto a tener en cuenta es el siguiente: los programas gráficos tienen normalmente un color de primer plano (foreground) y otro de fondo (background). Para conseguir nuestro propósito tenemos que forzar al programa gráfico para que su color de fondo sea precisamente el color que queremos hacer transparente (en este ejemplo, el color gris claro).

GIFs entrelazados

Normalmente, un fichero GIF contiene los datos de cada línea de la imagen de una manera ordenada, de tal manera que al ser cargada por el navegador aparecerá dibujada línea a línea desde arriba hasta abajo.

Se puede cambiar este comportamiento si se ha guardado la imagen como un GIF entrelazado (interlaced GIF). En este caso, las líneas quedan guardadas no de una manera consecutiva, sino en saltos de de cuatro en cuatro, y al llegar al final recomienza desde el principio con otra secuencia diferente, también de cuatro en cuatro, así hasta completar la imagen.

10. Tablas

Hasta que no se empezaron a usar las tablas, la única manera de tabular las cosas era utilizar la etiqueta de preformateado, con la que es necesario poner manualmente los espacios en blanco para que quede todo alineado formando filas y columnas, con un resultado muy poco estético.

Estructura de una tabla

Vamos a ver ordenadamente (de fuera hacia dentro) las etiquetas necesarias para confeccionar las tablas.

<TABLE>

La etiqueta general, que engloba a todas las demás es <TABLE> y </TABLE>.

```
<TABLE>
[resto de las etiquetas]
</TABLE>
```

Con esto se presentarían los datos tabulados, pero faltaría la característica que hace más atractivas a las tablas, y es que estos datos vayan dentro de unos cajetines formados por un borde. Para esto tenemos que añadir el atributo BORDER a la etiqueta, es decir:

```
<TABLE BORDER>
[resto de las etiquetas]
</TABLE>
```

<TR>

En el siguiente nivel, dentro de la anterior, están las etiquetas para formar cada fila (row) de la tabla, que son <TR> y </TR>. Hay que repetirlas tantas veces como filas queremos que tenga la tabla. Es decir, para una tabla con dos filas, sería:

```
<TR>
[etiquetas de las distintas celdas de la primera fila]
</TR>
<TR>
[etiquetas de las distintas celdas de la segunda fila]
</TR>
```

<TD>

En el último nivel (dentro de las anteriores) están las etiquetas de cada celda, que son <TD> y </TD>, que engloban el contenido de cada celda concreta (texto, imágenes, etc.). Hay que repetirla tantas veces como celdas queremos que haya en esa fila.

```
<TABLE BORDER>
  <TR>
    <TD>fila1-celda1</TD> <TD>fila1-celda2</TD> <TD>fila1-celda3</TD>
  </TR>
  <TR>
    <TD>fila2-celda1</TD> <TD>fila2-celda2</TD> <TD>fila2-celda3</TD>
```

```
</TR>
</TABLE>
```

Filas con desigual número de celdas

En este ejemplo hemos puesto dos filas con igual número de celdas. ¿Qué pasa si ese número es distinto? Pues el navegador forma el número de filas y columnas que haga falta, dejando espacios en blanco en las filas que tengan menos celdas.

Titular de la tabla

Se puede añadir un titular (caption) a la tabla, es decir un texto situado encima de la tabla que indica cuál es su contenido. Se consigue con la etiqueta <CAPTION> y </CAPTION>.

Variando el espesor de los bordes

El atributo BORDER (visto más arriba) pone por defecto un borde de espesor igual a la unidad. Pero se puede hacer que este borde sea tan grueso como queramos, poniendo:

```
<TABLE BORDER=número deseado>
```

Celdas de cabecera

Además de las celdas que contienen datos normales, podemos poner, si nos conviene, celdas de cabecera (header), que se distinguen por estar el texto de dichas celdas en negrita y centrado. Esto se consigue con la etiqueta <TH> y </TH> (en vez de la normal <TD> y </TD>)

Contenido de las celdas

Hasta ahora, en todos los ejemplos se ha puesto un texto normal dentro de las distintas celdas. Pero se puede poner en ellas cualquier otro elemento de los que van en un documento HTML, como imágenes, enlaces, etc. No hay más que poner dentro de la etiqueta de la celda la etiqueta correspondiente a una imagen, un enlace, etc.

Ejemplo con una imagen:

```
<TABLE BORDER=2>
<TR><TD><IMG SRC="babe.gif"></TD></TR>
</TABLE>
```

Ejemplo con un enlace:

```
<TABLE BORDER=2>
<TR><TD><A HREF="index.html">Página principal </A></TD></TR>
</TABLE>
```

Posicionamiento del contenido dentro de la celda

Normalmente, el contenido de una celda está alineado a la izquierda. Pero se puede cambiar esto añadiendo dentro de la etiqueta de la celda los siguientes atributos:

```
<TD ALIGN=CENTER> Al centro </TD>
```


`<TD ALIGN=RIGHT> A la derecha </TD>`

`<TH ALIGN=LEFT> Cabecera a la izquierda </TH>` (Recuérdese que por defecto están centradas)

El alineamiento por defecto en el sentido vertical es en el medio. También se puede cambiar, añadiendo dentro de la etiqueta de la celda los siguientes atributos:

`<TD VALIGN=TOP> Arriba </TD>`

`<TD VALIGN=BOTTOM> Abajo </TD>`

Variando las dimensiones de la tabla

El navegador se encarga normalmente de dimensionar el tamaño total de la tabla de acuerdo con el número de filas, de columnas, por el contenido de las celdas, espesor de los bordes, etc.

A veces nos puede convenir forzarle para que la tabla tenga unas dimensiones totales mayores que las que le corresponden, tanto en anchura como en longitud. Esto se consigue añadiendo dentro de la etiqueta de la tabla los atributos WIDTH y HEIGHT igual a un porcentaje de la dimensión de la pantalla, o a una cifra que equivale al número de pixels.

Celdas que abarcan a otras varias

A veces puede interesarnos que una celda se extienda sobre otras varias. Esto se consigue añadiendo dentro de la etiqueta de la celda los atributos COLSPAN=número para extenderse sobre un número determinado de columnas, o ROWSPAN=número para extenderse verticalmente sobre un número determinado de filas.

Por ejemplo, en la primera tabla de este capítulo vamos a añadir una fila con una sola celda, que abarca a dos columnas:

`<TR> <TD COLSPAN=2> Celda sobre 2 columnas </TD> </TR>`

O, en la misma tabla, vamos a añadir una celda en la primera fila. pero que abarque también a la siguiente:

`<TD ROWSPAN=2> Celda junto a 2 filas </TD>`

Color de fondo en las tablas

Podemos conseguir que las tablas tengan un color de fondo, siguiendo un procedimiento totalmente análogo al empleado para que una página tenga un color de fondo uniforme (según vimos en el capítulo 7). Para ello debemos utilizar el atributo BGCOLOR="#XXYYZZ", visto en dicho capítulo.

Imágenes de fondo en las tablas

El Explorer de Microsoft soporta la colocación de imágenes de fondo en el interior de las tablas, de una manera análoga a como se hace en una página (según vimos en el capítulo 7). Para ello debemos utilizar el atributo BACKGROUND="imagen.gif" o BACKGROUND="imagen.jpg".

Si se utiliza dentro de la etiqueta <TABLE> la imagen en cuestión se multiplicará detrás de todas las celdas. Si, por el contrario, sólo se pone este atributo dentro de la etiqueta de una celda concreta (<TD> o <TH>), entonces la imagen de fondo se verá sólo en esa celda, como por ejemplo:

Separación entre las celdas de una tabla

Por defecto, la separación entre las distintas celdas de una tabla es de dos pixels. Pero se puede variar esto con el atributo CELLSPACING, que se pone dentro de la etiqueta TABLE.

Por ejemplo, para obtener una separación de 20 pixels entre celdas ponemos:

```
<TABLE BORDER CELLSPACING=20>
```

A primera vista parece como si esto fuera lo mismo que si hubiéramos aumentado el espesor de los bordes. Pero para comprobar que no es así, hagamos que en el caso anterior, tenga además unos bordes de 5 de espesor:

```
<TABLE BORDER=5 CELLSPACING=20>
```

Separación entre el borde y el contenido dentro de las celdas

Por defecto, la separación entre el borde y el contenido dentro de las celdas es de un pixel. Se puede cambiar esto con el atributo CELLPADDING, que se pone dentro de la etiqueta TABLE.

Por ejemplo, para obtener una separación de 20 pixels entre el contenido y los bordes, dentro de cada celda:

```
<TABLE BORDER CELLPADDING=20>  
(Se omite el resto de las etiquetas de la tabla)
```

11. Formularios

La manera general para que los lectores de nuestra página se puedan comunicar con nosotros es por medio de un enlace a nuestra dirección de email, con lo que recibiríamos un email convencional.

Pero puede ser que lo que necesitemos sea sólomente una respuesta concreta a unas opciones que presentaremos nosotros mismos, o un comentario del usuario, para lo que le suministraremos un espacio en donde introducirlo.

Se puede hacer todo esto, además de otras cosas, utilizando los formularios, con los que se pueden confeccionar páginas que contengan los elementos necesarios para ello, tal como botones de radio, listas de selección, cajetines de introducción de texto y de control, etc., como vamos a ver.

Los formularios permiten que los demás nos envíen la información directamente a nosotros o bien a nuestro servidor, en donde hemos instalado un programa que procese esta información.

Por ejemplo, vamos a suponer que queremos crear una lista de correo. Los usuarios pueden introducir sus nombres y direcciones de email y pulsar un botón de envío. Esos datos los podemos recibir "en bruto" en nuestro correo, con los que confeccionaríamos manualmente dicha lista de correo, sin necesitar ningún programa para ello. Este proceso es el que vamos a comentar en este capítulo.

La otra posibilidad, de la que únicamente se va a hacer esta mención, es que hubiéramos instalado en nuestro servidor un programa especial para procesar esos datos y añadirlos a la lista de correo, y que incluso pudiera devolver automáticamente al usuario algún tipo de información. Para conseguir esto, los formularios necesitan ejecutar programas o scripts por medio del CGI (Common Gateway Interface). El CGI permite a los formularios ser procesados por programas escritos en cualquier lenguaje, aunque los más usados en Internet son el Perl y el C.

Estructura de un formulario

La estructura general de un formulario es:

1. Etiqueta de inicio:
2. Cuerpo del formulario, con los distintos elementos para poder introducir los datos.
3. Botones de envío y de borrado.
4. Etiqueta de cierre </FORM>

Etiqueta de inicio

El atributo ACTION indica la acción que se debe efectuar y que es que los datos sean enviados por email a la dirección indicada. (Si hiciéramos uso del CGI, sería precisamente aquí donde indicaríamos su localización en el servidor, que habitualmente es el directorio cgi-bin, para que procese los datos).

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
```

El atributo METHOD=POST indica que los datos sean inmediatamente enviados por correo a la dirección de email, nada más pulsar el usuario el botón de envío.

Con el atributo ENCTYPE="TEXT/PLAIN" se consigue que las respuestas las recibamos como un fichero de texto, perfectamente legible y sin codificar.

Elementos para introducir los datos

Los vamos a dividir en tres clases:

1. Introducción por medio de texto
2. Introducción por medio de menús
3. Introducción por medio de botones

La introducción de los datos se consigue por medio de la etiqueta:

```
<INPUT TYPE="xxx" NAME="yyy" VALUE="zzz">
```

En donde:

xxx es la palabra que indica el tipo de introducción.

yyy es el nombre que le asignamos nosotros a la variable de introducción del dato.

zzz es la palabra asociada a un elemento.

Todo esto, que de momento parece muy confuso, se aclarará al ir viendo los distintos casos.

Introducción por medio de texto (una línea)

En este caso es xxx=text, es decir, INPUT TYPE="text". El atributo VALUE no procede en este caso.

Vamos a poner un ejemplo: solicitamos el apellido del usuario.

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
Escribe tu apellido:
<BR><INPUT TYPE="text" NAME="Apellido">
</FORM>
```

Si el usuario introduce su apellido, p. ej. Ruiz, y pulsa el botón de envío (que veremos más adelante), recibiremos un email suyo con el siguiente texto:

Apellido=Ruiz

La longitud de este formulario es por defecto de 20 caracteres. Se puede variar incluyendo en la etiqueta el atributo SIZE="número". Por otra parte, sea cual sea la longitud del formulario, si no se indica nada, el usuario puede introducir el número de caracteres que quiera. Se puede limitar esto, incluyendo en la etiqueta el atributo MAXLENGTH="número".

En el caso que hemos visto, si hubiéramos cambiado la etiqueta correspondiente por:

```
<INPUT TYPE="text" NAME="Apellido" SIZE="10" MAXLENGTH="12">
```

Se puede comprobar cómo no se pueden introducir más de 12 caracteres. También se puede hacer que el texto introducido no sea reconocible, es decir que todos los caracteres se representen por asteriscos. Basta con cambiar en la etiqueta INPUT TYPE="text" por INPUT TYPE="password". En el último ejemplo, si cambiamos la etiqueta correspondiente por:

```
<INPUT TYPE="password" NAME="Apellido" SIZE="10" MAXLENGTH="12">
```

Se puede comprobar cómo los caracteres introducidos se representan por asteriscos.

Introducción por medio de texto (múltiples líneas)

Cuando el texto a introducir puede alcanzar una gran longitud, por ejemplo un comentario, es conveniente utilizar un formulario de texto de múltiples líneas.

Esto se consigue con la etiqueta de inicio:

```
<TEXTAREA NAME="yyy" ROWS="número" COLS="número">
```

(en donde no se utiliza INPUT TYPE y donde ROWS representa el número de filas, y COLS el de columnas).

y la de cierre: </TEXTAREA>

Ejemplo: un formulario solicitando los comentarios del usuario:

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
```

Introduce tus comentarios:

```
<BR><TEXTAREA NAME="Comentarios" ROWS="6" COLS="40">
```

```
</TEXTAREA>
```

```
</FORM>
```

Introducción por medio de menús

Si queremos que el usuario, en vez de introducir un texto, como hemos visto en los casos anteriores, escoja entre varias opciones que le presentamos nosotros, haremos uso de un formulario en forma de menú.

Se consigue con la etiqueta de inicio <SELECT NAME="yyy"> y la de cierre </SELECT>. Las distintas opciones a escoger se consiguen con la etiqueta <OPTION>.

Ejemplo: Pedimos al usuario que elija su color preferido:

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
```

```
¿Cuál es tu color preferido?
```

```
<BR><SELECT NAME="ColorPreferido">
```

```
<OPTION>Rojo
```

```
<OPTION>Verde
```

```
<OPTION>Azul
```

```
<OPTION>Amarillo
```

```
</SELECT >
```

```
</FORM>
```

Si queremos que sean visibles múltiples opciones a la vez, añadimos en la etiqueta los atributos MULTIPLE SIZE="número", donde especificamos el número de opciones visibles.

Formulario de confirmación (checkbox)

Si queremos que el usuario confirme una opción determinada, podemos hacer uso de un formulario de confirmación, o checkbox, que se consigue con la etiqueta:

```
<INPUT TYPE="checkbox" NAME="yyy">
```

Ejemplo: Solicitamos al usuario que confirme su inclusión en una lista de correo:

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
```

```
<INPUT TYPE="checkbox" NAME="Lista">
Sí, deseo ser incluido en la lista de correo.
</FORM>
```

Botones de radio

Cuando queremos que el usuario elija una única opción entre varias, podemos hacer uso de los botones de radio, que se consiguen con la etiqueta:

```
<INPUT TYPE="radio" NAME="yyy" VALUE="zzz">
```

Donde yyy es el nombre que le ponemos a la variable que se trata de elegir, y zzz es el nombre de cada una de las opciones en concreto.

Botones de envío y de borrado

Hasta ahora, en todos los ejemplos que hemos visto, faltaba un elemento esencial en cualquier formulario, y es el botón de envío de los datos, que se consigue con la etiqueta:

```
<INPUT TYPE="submit" VALUE="zzz">
```

En donde zzz es el texto que queremos que aparezca en el botón.

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
```

Otro botón interesante es el de borrado de los datos introducidos, muy conveniente en un formulario con muchos elementos. Es muy similar al de envío, pues se consigue con la etiqueta:

```
<INPUT TYPE="reset" VALUE="zzz">
```

En donde zzz es el texto que queremos que aparezca en el botón.

Hasta ahora hemos visto uno a uno los diferentes elementos que se pueden utilizar. Pero no hay ningún inconveniente en usar, dentro del mismo formulario, distintos tipos de introducción de datos. Al pulsar el usuario el botón de envío recibiríamos en email suyo con las distintas parejas NAME=VALUE de cada elemento, encadenadas con el símbolo &.

Ejemplo práctico

```
<P><CENTER>
```

```
<H2>Libro de visitas</H2>
<P><FORM ACTION="mailto:info@igf.es" METHOD="POST"
ENCTYPE="TEXT/PLAIN">
Tu nombre:
<BR><INPUT TYPE="text" NAME="Nombre">
<P>Escribe tus comentarios:
<BR><TEXTAREA NAME="Comentarios" ROWS="6" COLS="40">
</TEXTAREA>
<P><INPUT TYPE="submit" VALUE="Enviar datos">
<INPUT TYPE="reset" VALUE="Borrar datos">
</FORM>
<P>
<HR>
```

12. GIFs animados

Programas para crear animaciones

Para crear una animación, en primer lugar, hay que crear una a una las distintas imágenes que van a constituir la animación. Cualquier programa gráfico que sea capaz de guardar las imágenes en formato GIF es válido para este propósito.

En segundo lugar, es necesario un programa que ensamble estas imágenes individuales de una manera apropiada para crear la animación. Los programas más utilizados para ello son:

GIF Construction Set (Win3.1x y Win95)
Gif Animator de Microsoft (Win95)

Partiendo del ejemplo práctico de la lección anterior mipag11.html, añadimos entre las etiquetas <HR> y <BODY>, situadas al final, lo siguiente:

```
<P><HR>  
<H2>¡Sitio del Web en obras!</H2>  
<IMG SRC="workanim.gif">  
<P><HR>
```


13. Mapas

Hemos visto anteriormente dos maneras distintas de enlazar una página con otra: por medio de un enlace de texto (capítulo 3) o por medio de una imagen (capítulo 4). Pero en este último caso se hacía uso de una imagen para enlazar a una única página.

Se puede utilizar una única imagen para enlazar con varias páginas, yendo a una u otra según la zona en donde se pulse el ratón. Este tipo de imágenes se llaman mapas.

Veremos cómo se crea un mapa partiendo de una imagen, y cómo se implementa dicho mapa con el lenguaje HTML en nuestra página.

Hay dos tipos de mapas:

1. Mapas gestionados por el servidor
2. Mapas gestionados por el cliente

Mapas gestionados por el servidor

En este tipo de mapas al desplazar el cursor sobre la imagen que forma el mapa se van obteniendo las distintas coordenadas. Cuando se pulsa el ratón en un punto determinado, el navegador envía esas coordenadas al servidor, y éste comprueba en un fichero MAP (situado en en su directorio cgi-bin) cuál es la página que se corresponde con estas coordenadas, envía esta información al navegador, y éste solicita a su vez al servidor que le enlace con dicha página.

Estos mapas, que fueron los que aparecieron inicialmente, tienen una ventaja: que pueden ser utilizados por todos los navegadores, incluso por las versiones más antiguas.

Pero tienen varios inconvenientes: No todos los servidores tienen implementado el programa CGI necesario para que pueda funcionar un mapa de este tipo. Además hay que conocer el tipo de servidor (NCSA, el más común, o CERN), pues el código a aplicar es distinto en un caso u otro.. Por otra parte, al requerir un trasiego de información entre el navegador y el servidor, el tiempo de respuesta es mayor que en el otro tipo, como veremos.

Mapas gestionados por el cliente

Estos mapas son gestionados por el cliente (es decir, nuestro navegador) y no por el servidor, como en el caso anterior.

En este tipo de mapas, al ir desplazando el cursor se observa que en ciertas zonas se convierte en el símbolo normal para enlazar con otras páginas (una mano, habitualmente). Si se pulsa en estas zonas activas (hotspots), el navegador consulta con el documento HTML que ha recibido, y decide a qué página o dirección corresponde esa zona activa.

Es decir, no necesita enviar la información al servidor y esperar la respuesta de éste, con lo que el proceso es más rápido que en el caso anterior, reduciéndose además el tráfico de datos y la sobrecarga al servidor. Otras ventajas son que al pasar por una zona activa se muestra la dirección a la que se corresponde (al contrario que en el caso

anterior, en la que sólo se ven unas coordenadas). Además, estos mapas se pueden utilizar off-line, al contrario también que en el otro tipo, ya que no requieren la ayuda del servidor.

Pero tiene el inconveniente de que estos mapas no son implementados por todos los navegadores, como en el caso anterior, aunque sí por la mayoría de los utilizados actualmente, desde las versiones siguientes: Netscape 2.0, Microsoft Explorer 2.0, Mosaic 2.1, etc. Pero se puede obviar este inconveniente suministrando enlaces alternativos para los navegadores que no implementen este tipo de mapas, como veremos.

En lo que sigue, vamos a ver exclusivamente cómo implementar este tipo de mapas, es decir, mapas gestionados por el cliente (client-side image maps).

Confección de la imagen del mapa

Antes que nada, es necesario confeccionar la imagen que va a servir como mapa. Se hará con un programa gráfico, y pueden utilizarse los formatos GIF o JPG (ver el capítulo 9).

Vamos a crear una mapa para utilizarlo en el ejemplo práctico de este capítulo, con dos zonas activas rectangulares que enlancen con los ejemplos prácticos de los capítulos 2 y 3.

Código HTML para definir un mapa gestionado por el cliente

En este caso, el código que debemos escribir es el siguiente:

```
<MAP NAME="mi_mapa">
<AREA SHAPE="RECT" COORDS="46,40,250,75" HREF="mipag2.html">
<AREA SHAPE="RECT" COORDS="46,100,250,135" HREF="mipag3.html">
<AREA SHAPE="DEFAULT" NOHREF>
</MAP>
<IMG SRC="pulsadr1.gif" USEMAP="#mi_mapa">
```

```
<MAP NAME="mi_mapa">
```

Esta es la etiqueta de apertura del mapa, y en donde se define su nombre.

```
<AREA SHAPE="RECT" COORDS="46,40,250,75" HREF="mipag2.html">
```

Esta etiqueta define la primera zona activa. Se indica que su forma (shape, en inglés) es rectangular, y que las coordenadas de su ángulo superior izquierdo son (46,40), y que las del ángulo inferior derecho son (250,75). A continuación se indica cuál es enlace que se desea obtener, en este caso con la página mipag2.html.

```
<AREA SHAPE="DEFAULT" NOHREF>
```

Con esta etiqueta se define el área completa del mapa, indicando que su forma es la que tiene por defecto (default), o sea rectangular. Si se quisiera que el mapa no abarcara la totalidad de la imagen, o incluso que tuviera una forma distinta, habría que indicarlo aquí, en lugar de la instrucción DEFAULT, haciendo uso del mismo tipo de instrucciones que las de las zonas activas.

A continuación se indica, con NOHREF, que el área completa de la imagen es una zona no activa, excepto en las áreas definidas anteriormente como activas.

```
</MAP>
```

Etiqueta de cierre del mapa.

```
<IMG SRC="pulsadr1.gif" USEMAP="#mi_mapa">
```

Con esta etiqueta se solicita primero al servidor que envíe una imagen llamada pulsadr1.gif. Con USEMAP="#mi_mapa" se indica que esta imagen es el mapa definido anteriormente con ese nombre, y que debe actuar en consecuencia a las pulsaciones del ratón, solicitando un enlace determinado en las zonas activas, o no haciendo nada en el resto.

Para definir una zona activa circular es necesario conocer las coordenadas de su centro (x,y) y la longitud de su radio (r). Una vez sabidos estos datos, la etiqueta es como sigue:

```
<AREA SHAPE="CIRCLE" COORDS="x,y,r" HREF="dirección_de_la_página">
```

Para definir una zona activa poligonal hay que conocer las coordenadas de los distintos puntos (x1,y1), (x2,y2), (x3,y3), etc. Con estos datos, la etiqueta se escribe como sigue:

```
<AREA SHAPE="POLYGON" COORDS="x1,y1,x2,y2,x3,y3,..."  
HREF="dirección_de_la_página">
```

14. Frames

Frames (en inglés, marcos o cuadros) es un procedimiento del lenguaje HTML para dividir la pantalla en diferentes zonas, o ventanas, que pueden actuar independientemente unas de otras, como si se trataran de páginas diferentes, pues incluso cada una de ellas pueden tener sus propias barras deslizadoras. Los navegadores que lo implementan son el Netscape 2.0, y el Explorer 2.0 en adelante. Aunque no está incluido en el actual estándar HTML 3.2, está previsto que lo esté en el próximo (Cougar).

Una de sus características más importantes es que pulsando un enlace situado en un frame, se puede cargar en otro frame una página determinada. Esto se utiliza frecuentemente para tener un frame estrecho en la parte lateral (o superior) con un índice del contenido en forma de diferentes enlaces, que, al ser pulsados cargan en la ventana principal las distintas páginas. De esta manera se facilita la navegación entre las páginas, pues aunque se vaya pasando de unas a otras, siempre estará a la vista el índice del conjunto.

Para comprender los distintos conceptos vamos a desarrollar un ejemplo, creando una página con dos frames. El de la izquierda va a servir de índice de lo que veamos en el de la derecha, y en éste veremos inicialmente una página de presentación. Se podrá acceder también aquí a la página personal creada en los capítulos anteriores (mipag13.html), si se pulsa un enlace en el frame de la izquierda.

Documento de definición de los frames

Lo primero que tenemos que hacer es crear un documento HTML en el que definiremos cuántas zonas va a haber, qué distribución y tamaño van a tener, y cuál va ser el contenido de cada una de ellas.

En el ejemplo que vamos a desarrollar, la página va a tener dos frames distribuidos en columnas (es decir, uno al lado del otro, en vez de uno encima del otro, lo que sería una distribución en filas).

Con respecto al tamaño, haremos que el primero (el del izquierda) ocupe el 20% del ancho de la pantalla, y el otro, el 80% restante.

Y con respecto al contenido, el frame de la izquierda va a contener un documento HTML que va a servir de índice de lo que veamos en el otro (y que vamos a llamar mipagind.html), y el de la derecha otro documento HTML que va a servir de página de presentación (al que llamaremos mipagpre.html).

Todo lo anterior se refleja en el siguiente documento HTML:

```
<HTML>
<HEAD>
<TITLE>Mi pagina con frames</TITLE>
</HEAD>
<FRAMESET COLS="20%, 80%">
<FRAME SRC="mipagind.html">
<FRAME SRC="mipagpre.html" NAME="principal">
</FRAMESET>
</HTML>
```

Obsérvese lo siguiente:

Es un documento parecido a los que conocíamos hasta ahora. La diferencia está en 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 (definir los frames).

En este caso, con la etiqueta <FRAMESET COLS="20%, 80%"> se define que va a haber dos frames y que van a ir en columnas. Si hubiéramos querido que fueran en filas, habríamos puesto ROWS (filas, en inglés). También se define el espacio en anchura que van a ocupar cada uno de ellos en la pantalla. Se ha puesto como porcentajes del total, pero se podría también haber puesto una cifra absoluta, que representaría el número de pixels a ocupar.

Ya se ha definido el número de frames, su distribución y su tamaño, pero falta por definir el contenido de cada frame. Esto se hace con las etiquetas:

```
<FRAME SRC="mipagind.html">
<FRAME SRC="mipagpre.html" NAME="principal">
```

Con esto se define que el contenido del primer frame (el de la izquierda) sea el documento HTML mipagind.html y el del segundo (el de la derecha) sea el documento HTML mipagpre.html.

Obsérvese que en la etiqueta del segundo se ha incluido el atributo NAME="principal", pero no así en el primero. El motivo es que se necesita dar un nombre al segundo frame, pues, como veremos a continuación, en el documento del primer frame va a haber unos enlaces que van a ir dirigidos hacia él. En este caso sólo tenemos dos frames, pero podría haber más, y es necesario distinguirlos unos de otros. Y el primero no necesita nombre, pues no va a haber enlaces en el segundo dirigidos hacia él.

A este documento le vamos a llamar mipagina.html, pero todavía no lo vamos a guardar, pues falta por añadir algo que veremos más adelante.

Documentos HTML de cada frame

Necesitamos ahora confeccionar el documento HTML de cada uno de los frames. Recuérdese que son como páginas independientes, que pueden tener cada una su propio fondo, etc., y todo lo que queramos añadir en ellos y que hemos aprendido hasta ahora.

Documento del frame de la izquierda

Va a tener un fondo amarillo, y va a contener dos enlaces dirigidos al frame de la derecha. Además, como muestra de que se puede añadir cualquier cosa en un frame, vamos a incluir este logo animado (glogoan2.gif):

```
<HTML>
<HEAD>
<TITLE> Indice </TITLE>
</HEAD>
<BODY BGCOLOR="#FFBB00">
<P><A HREF="mipagpre.html" TARGET="principal"> Presentación </A>
<P><A HREF="mipag13.html" TARGET="principal"> Mi página </A>
<P><IMG SRC="glogoan2.gif">
</BODY>
</HTML>
```

Dentro de las etiquetas de los enlaces podemos observar algo nuevo, y es el atributo TARGET (en inglés: objetivo, blanco), que sirve para hacer que al ser activado el enlace no se cargue en el propio frame, sino en otro, precisamente en el que hayamos llamado con ese nombre en el documento de definición de los frames.

En nuestro caso, le hemos dado el nombre de "principal" al frame de la derecha, y es por tanto ahí donde se van a cargar los documentos HTML.

Guardamos este documento con el nombre de mipagind.html. Además, capturamos la imagen glogoan2.gif, y la guardamos junto con el documento.

Documento del frame de la derecha

Va a tener un fondo negro, y va a contener sólo un texto. En capítulos posteriores se irán añadiendo aquí más cosas.

```
<HTML>
<HEAD>
<TITLE> Presentacion </TITLE>
</HEAD>
<BODY BGCOLOR="#000000" TEXT="#0000FF">
<CENTER>
<FONT SIZE=+3><STRONG>
<P>ESTA ES LA VERSION
<BR><FONT COLOR="#FF0000">CON FRAMES</FONT>
<BR>DE MI PAGINA
</STRONG>
</FONT>
</CENTER>
</BODY>
</HTML>
```

Alternativa para los navegadores que no soportan frames

Con lo visto anteriormente, ya tenemos las tres piezas que necesitamos:

- Documento de definición de los frames
- Documento del contenido del primer frame
- Documento del contenido del segundo frame

En teoría no necesitamos nada más. Sólo queda un detalle: ¿Qué ocurriría con los navegadores que no soportan los frames? Pues que al desconocer las etiquetas FRAMESET y FRAME del documento de definición, no podría ejecutarlo y no podría cargar la página. Para estos casos está prevista la etiqueta <NOFRAMES> y </NOFRAMES>. Se añaden al final del documento de definición de los frames, y a se pone entre ambas lo que queremos que vean los que acceden con un navegador que no soporta frames. Puede incluso ser el código HTML de una página completa (lo que normalmente va entre las etiquetas <BODY> y </BODY>)

En nuestro caso, vamos a poner un mensaje advirtiendo de esta circunstancia, y dirigiendo al usuario, con un enlace normal, hacia la página mipag13.html. (También

podríamos no decir nada sobre frames, y colocar aquí el código de la página de presentación, pero añadiendo un enlace a mipag13.html, para que se pueda pasar de página)

<NOFRAMES>

Estas utilizando un navegador que no soporta frames.

```
<P>Pulsa para visitar mi <A HREF="mipag13.html"> página</A>
</NOFRAMES>
```

Lo incluimos dentro del documento de definición de los frames, que hemos visto antes, y que queda de esta manera:

```
<HTML>
<HEAD>
<TITLE>Mi pagina con frames</TITLE>
</HEAD>
<FRAMESET COLS="20%, 80%">
<FRAME SRC="mipagind.html">
<FRAME SRC="mipagpre.html" NAME="principal">
</FRAMESET>
<NOFRAMES>
Estas utilizando un navegador que no soporta frames.
<P>Pulsa para visitar mi <A HREF="mipag13.html"> página</A>
</NOFRAMES>
</HTML>
```

<FRAMESET>

La etiqueta FRAMESET, como hemos visto en el ejemplo, es la que define la distribución, el número y tamaño de los frames. Tiene dos atributos: COLS (columnas) y ROWS (filas):

```
<FRAMESET COLS="xx, yy, zz, ..">
<FRAMESET ROWS="xx, yy, zz, ..">
```

Define la distribución de los frames en columnas o en filas, según se use uno u otro atributo.

Define el número de frames que habrá, pues, por ejemplo:

```
<FRAMESET COLS="xx, yy"> (habrá dos frames en columnas)
<FRAMESET COLS="xx, yy, zz"> (habrá tres frames en columnas)
```

Define el tamaño de los frames, según el valor que demos a xx, yy, zz.... Este valor se puede expresar en:

Un porcentaje del ancho del pantalla (para las columnas), o del alto de la pantalla (para las filas). Así, por ejemplo:

```
<FRAMESET COLS="%20, %80"> (la columna de la izquierda ocupará el 20% del ancho de la pantalla, y la de la derecha el 80% restante)
<FRAMESET ROWS="%10, %70, %20"> (la fila superior ocupará el 10% del alto de la pantalla, la del medio el 70%, y la inferior el 20%)
<FRAMESET COLS="40, 600"> (la columna de la izquierda tendrá 40 pixels de ancho y la de la derecha 600). Pero es peligroso utilizar sólo valores absolutos, pues el tamaño de la
```

pantalla varía de un usuario a otro. Si se va a usar algún valor absoluto para un frame, es mejor mezclarlo con alguno relativo, como los que vamos a ver a continuación, para que se ajuste el total a la pantalla del usuario

Un valor relativo que se consigue poniendo un asterisco (*), en vez de un número. Esto se interpreta como que ese frame debe tener el espacio restante. Por ejemplo: `<FRAMESET ROWS="100,*,100">` (Habrá tres filas, la superior y la inferior de una altura fija de 100 pixels, y la del medio obtendrá el espacio restante). Si hay más de un frame con asterisco, ese espacio restante se dividirá por igual entre ellos.

Si hay un número antes del asterisco, ese frame obtiene esa cantidad más de espacio relativo. Así `"2*,*"` daría dos tercios para el primer frame y un tercio para el otro.

Frames sin bordes

Si se desea que no haya un borde de separación entre los frames, se deben incluir el atributo `FRAMEBORDER=0` dentro de la etiqueta `FRAMESET`. No todas las versiones de los navegadores lo implementan.

Para que también desaparezcan los huecos de separación entre frames hay que añadir otros dos atributos (el primero es para el Explorer y el segundo para el Netscape): `FRAMESPACING=0` y `BORDER=0`. con lo que la etiqueta completa quedaría:

```
<FRAMESET FRAMEBORDER=0 FRAMESPACING=0 BORDER=0 COLS="xx, yy">
```

<FRAME>

Esta etiqueta define las características de un frame concreto, no del conjunto de los frames, como era el caso con la etiqueta `<FRAMESET>`. Puede tener los siguientes posibles atributos, que van dentro de la etiqueta `<FRAME>`:

SRC="dirección". Esta dirección puede ser la de un documento HTML (tal como hemos utilizado en el ejemplo), o cualquier otro recurso del Web (o URL). Con este atributo se indica lo que se cargará inicialmente en el frame. Si no se le pone este atributo a la etiqueta `<FRAME>`, entonces dicho frame aparecerá inicialmente vacío, aunque tendrá las dimensiones asignadas.

NAME="nombre_de_la_ventana". Este atributo se usa para asignar un nombre a un frame. De esta manera se podrá "dar en el blanco" (en inglés, target) en esta página, desde un enlace situado en otra página. Es decir, que pulsando en otra página un enlace, se cargará precisamente en ésta, tal como hemos visto en el ejemplo. El atributo `NAME` es opcional. Por defecto, todas las ventanas carecen de nombre. Los nombres que se escojan deben comenzar por un carácter alfanumérico (una letra o un número, pero no otro tipo de símbolo)

MARGINWIDTH="número". Se utiliza este atributo cuando se quiere controlar la el ancho de los márgenes dentro de un frame. El número que se ponga representa los pixels de los márgenes. Este atributo es opcional.

MARGINHEIGHT="número". Igual que el anterior, pero referido a los márgenes en altura.

SCROLLING="yes|no|auto". Este atributo se utiliza para decidir si el frame tendrá o no una barra deslizadora. Si se escoge "yes" tendrá siempre una barra deslizadora. Si se escoge "no" no la tendrá nunca, y si se escoge "auto", será el navegador quien decida si la tendrá o no. Este atributo es opcional. Su valor por defecto es "auto".

NORESIZE. A este atributo no se le asigna un valor numérico, como a los demás. Es un indicador para que la ventana no se pueda re-dimensionar (en inglés, *resize*) por parte del usuario. Se puede comprobar en el ejemplo con frames que si se coloca el cursor del ratón entre los dos frames, al pulsarlo y arrastrarlo en un sentido u otro, varían las dimensiones prefijadas de los frames. Este atributo impide que ocurra esto en un frame. Es un atributo opcional. Por defecto, todos los frames son re-dimensionables.

FRAMEBORDER="no". Este atributo elimina el borde en un frame, pero si se quiere que se elimine completamente, también hay que ponérselo al frame contiguo. Si se quiere eliminar los bordes de todos los frames, se debe colocar en la etiqueta **FRAMESET**, como hemos visto anteriormente.

TARGET

En el ejemplo hemos visto que, como queríamos que los enlaces situados en el frame de la izquierda surtieran efecto no en él mismo, sino en otro frame, teníamos que poner dentro de cada enlace el atributo **TARGET="principal"**, siendo "principal" el nombre que habíamos dado al segundo frame, en el documento de definición de frames. Es decir, hemos utilizado este atributo de esta manera:

TARGET="nombre_dado_a_otro_frame".

Estos nombres, que los escogemos nosotros, pueden ser cualquiera, pero con la condición que el primer carácter sea alfanumérico (letra o número).

Pero hay unos nombres reservados (es decir, que no se pueden usar para denominar a un frame), que hacen que este atributo efectúe unas funciones especiales. Para que cumplan su cometido, es imprescindible escribir estas palabras reservadas (**blank**, **self** y **top**) en minúsculas.

TARGET="_blank". Hace que se abra una nueva copia del navegador, y el enlace activado se carga en ella, a pantalla completa. Es decir, tendríamos dos copias del navegador (Netscape, Explorer, etc.) funcionando a la vez.

TARGET="_self". Hace que el enlace se cargue en el propio frame.

TARGET="_top". Hace que el enlace se cargue a pantalla completa, suprimiendo todos los frames, pero sin que se cargue una nueva copia del navegador. Este es particularmente útil. Un error muy común es olvidarse de poner este atributo en los enlaces que están en un frame, con lo que resulta que al ser activados, la página llamada se carga dentro del propio frame, lo cual es muy molesto si esa página pertenece a otro sitio del Web, y aún más grave si esa página tiene a su vez frames. Este inconveniente se evita poniendo este atributo dentro de las etiquetas de los enlaces.

Frames anidados dentro de otros frames

Hasta ahora hemos contemplado sólo la posibilidad de tener una distribución de los frames bien en filas o bien en columnas, (dependiendo que se utilice el atributo ROWS o COLS en la etiqueta FRAMESET), pero no ambos a la vez.

Se pueden obtener distribuciones más complejas anidando filas dentro de una columna, o a la inversa, columnas dentro de una fila.

Supongamos que queremos la siguiente distribución:

Un frame estrecho en horizontal en la parte superior, de lado a lado de la pantalla (altura, el 15%)

Otro frame estrecho en vertical en la parte izquierda, debajo del anterior (anchura, el 20%)

Un tercero ocupando el resto de la pantalla.

```
<HTML>
<HEAD>
<TITLE>Pagina con dos filas</TITLE>
</HEAD>
<FRAMESET ROWS="15%, *">
<FRAME SRC="documento_fila_superior">
<FRAME SRC="documento_fila_inferior">
</FRAMESET>
</HTML>
```

Como la fila inferior, en realidad, son dos columnas (con una distribución del 20% y resto), sustituimos (anidando) la etiqueta <FRAME SRC="documento_fila_inferior"> por:

```
<FRAMESET COLS="20%, *">
<FRAME SRC="documento_columna_izqda">
<FRAME SRC="documento_columna_dcha">
</FRAMESET>
```

Con lo que queda el documento definitivo así:

```
<HTML>
<HEAD>
<TITLE>Pagina con fila superior y dos columnas inferiores</TITLE>
</HEAD>
<FRAMESET ROWS="15%, *">
<FRAME SRC="documento_fila_superior">
  <FRAMESET COLS="20%, *">
    <FRAME SRC="documento_columna_izqda">
    <FRAME SRC="documento_columna_dcha">
  </FRAMESET>
</FRAMESET>
</HTML>
```

15. Sonidos

Una página del Web puede tener sonidos incorporados, bien sea como un fondo sonoro que se ejecuta automáticamente al cargar la página, o como una opción para que la active el propio usuario.

Capacidades sonoras de los navegadores

Para poder escuchar los sonidos es necesario disponer, como es lógico, de una tarjeta de sonido con sus correspondientes altavoces. Pero esto no es suficiente, pues no todos los programas navegadores están capacitados en la misma medida.

Explorer de Microsoft

Es el que está mejor adaptado para el sonido, pues a partir de la versión 2.0 es capaz de reproducir fondos sonoros sin necesidad de añadir nada, y no hay ninguna complicación con los servidores, como ocurre con el Netscape. Además, a partir de la versión 3.0 del Explorer, es incluso compatible con los plug-ins del Netscape.

Netscape

Las versiones anteriores a la 2.0 no son capaces de reproducir fondos sonoros que se ejecuten automáticamente, sino que requerirá que se activen los programas auxiliares asociados a los formatos .wav o .mid.

La versión 2.0 sí es capaz de reproducir un fondo sonoro, pero es necesario que tenga instalado un plug-in llamado Crescendo que se obtiene en LiveUpdate (Una vez obtenido hay que instalarlo manualmente en el directorio 'plugins' del Netscape 2.0).

La versión 3.0 lleva implícito el plug-in Live Audio (pero sólo la versión completa, no la reducida). En caso afirmativo, es capaz de reproducir un fondo sonoro.

Pero a todas estas complicaciones de las distintas versiones de los navegadores de los usuarios, hay otra más y es que el servidor donde esté alojada la página del Web debe tener configurados como MIME los formatos .mid y .wav. Si esto no es así, aunque depositemos en el servidor nuestro documento HTML acompañado por el correspondiente fichero de sonido, éste no se ejecutará. En dicho caso, es necesario ponerse en contacto con los administradores del servidor para que configuren como MIME los formatos .mid y .wav.

Este problema no existe, sin embargo para el Explorer, con el que no nos tenemos que preocupar por este tema.

Hemos visto hasta aquí las distintas capacidades de los navegadores. Ahora vamos a ver las etiquetas que debemos poner en nuestro documento HTML para que se puedan oír los sonidos en uno y otro navegador, y que lamentablemente no son iguales, pero que se pueden combinar, como veremos más adelante.

Fondo sonoro para el Microsoft Internet Explorer

Para las versiones 2.0 en adelante, se utiliza la etiqueta:

```
<BGSOUND SRC="fichero_de_sonido" LOOP=n>
```

El fichero de sonido puede estar en formato .mid o .wav. El atributo LOOP (en inglés, lazo) sirve para especificar el número (n) de veces que se debe ejecutar el fichero de sonido. Si se elige el número n=-1 o se pone LOOP=infinite, el sonido se ejecutará indefinidamente. Se puede omitir este atributo, y entonces el fichero se ejecutará una sola vez.

Fondo sonoro para el Netscape

La etiqueta básica para el Netscape es:

```
<EMBED SRC="fichero_de_sonido" WIDTH=xxx HEIGHT=yy>
```

donde WIDTH es la anchura y HEIGHT la altura de una consola que aparece, y que tiene diferentes teclas (play, stop, pausa, etc.). Más adelante veremos los valores xxx e yy que debemos atribuirles.

El fichero de sonido puede estar en formato .mid o .wav, pero recuérdese la advertencia hecha anteriormente, de que estos formatos deben estar configurados como MIME por el servidor donde esté alojada la página.

Dentro de la etiqueta se pueden añadir los siguientes atributos opcionales:

```
AUTOSTART="true" (arranca automáticamente)
LOOP="true" (se ejecuta ininterrumpidamente)
```

Fondo sonoro combinado para el Explorer y el Netscape

Se pueden combinar los dos tipos distintos de etiquetas para conseguir que un fondo sonoro sea escuchado por usuarios que utilicen tanto el Explorer como el Netscape (siempre que éste último esté preparado para ello). En este caso, el Explorer ignorará la etiqueta del Netscape, y a la inversa.

Vamos a ver las dos etiquetas necesarias para que se ejecute el fichero prv89.mid como sonido de fondo, tanto por el Explorer como por el Netscape (lo hará una sola vez):

```
<BGSOUND SRC="prv89.mid">
<EMBED SRC="prv89.mid" HIDDEN="true">
```

Ahora vamos a las etiquetas para que el fichero prv89.mid se ejecute como sonido de fondo, tanto por el Explorer como por el Netscape y que se ejecute indefinidamente.

A la etiqueta del Explorer basta con añadirle el atributo LOOP=infinite, como se ha visto anteriormente.

Y con respecto a la etiqueta del Netscape, teóricamente debería también bastar añadirle el atributo LOOP="true", pero curiosamente esto no es así, porque hay que poner las dimensiones de la consola (que no se va a ver).

Las dos etiquetas quedan de esta manera:

```
<BGSOUND SRC="prv89.mid" LOOP=infinite>
<EMBED SRC="prv89.mid" WIDTH=200 HEIGHT=55 AUTOSTART="true" LOOP="true"
HIDDEN="true">
```

Activación del sonido por el propio usuario

Hasta aquí hemos visto cómo poner un sonido de fondo en una página. Hay otra opción, mucho más sencilla, y es la de poner un enlace a un fichero de sonido, de tal manera, que al pulsarlo se ejecute el fichero. (Ver el Capítulo 3, en el que se explica cómo crear enlaces).

Escucha esta `musica`

Al pulsar el enlace se activa, en una ventana aparte, el programa que ejecuta el sonido. Esto es válido para todos los navegadores, incluso las versiones más antiguas, con la única condición de que se haya configurado un programa auxiliar capaz de ejecutar ficheros .mid o .wav.

Vamos a hacer un enlace con el icono sound1.gif al fichero de sonido problema.wav:

```
<A HREF="problemo.wav"><IMG SRC="sound1.gif"></A>
```

16. Texto en movimiento

Vamos a ver distintas soluciones para conseguir texto en movimiento: haciendo uso del lenguaje HTML (las marquesinas), con un GIF animado, con Javascript, etc.

Marquesinas (Marquees)

Una marquesina (en inglés, marquee) es una ventana en la que se desplaza un texto. Sólo son válidas para el Explorer de Microsoft (en el Netscape se verá como texto fijo).

La etiqueta básica es:

```
<MARQUEE> Texto que se desplaza </MARQUEE>
```

Como no hemos añadido ningún atributo dentro de la etiqueta, el comportamiento de la marquesina es el que tiene por defecto: ocupa todo el ancho de la pantalla, tiene la altura de una línea y el texto se desplaza lentamente de derecha a izquierda. Vamos a ver los distintos atributos que modifican su apariencia y comportamiento:

WIDTH, HEIGHT

Ajustan la anchura y altura, respectivamente, de la marquesina. Pueden ser igual a un número de pixels, o a un porcentaje de la pantalla.

Ejemplo:

```
<MARQUEE WIDTH=50% HEIGHT=60>
```

Esta marquesina ocupa el 50% del ancho de la pantalla y tiene una altura de 60 pixels

```
</MARQUEE>
```

ALIGN

Modifica el alineamiento del texto que rodea a la marquesina, que puede ser TOP (arriba), MIDDLE (en medio) o BOTTOM (abajo).

Ejemplo:

```
<MARQUEE WIDTH=50% HEIGHT=60 ALIGN=BOTTOM>La palabra "¡Hola!" estará
```

alineada con la parte inferior de la marquesina

```
</MARQUEE>
```

¡Hola!

BEHAVIOR

Este atributo (que quiere decir en inglés comportamiento) sirve para definir de qué manera se va a efectuar el desplazamiento del texto. Si es igual a SCROLL (el valor por defecto), el texto aparece por un lado, se desplaza hasta el otro hasta desaparecer por él, y vuelve a empezar (como los casos que hemos visto anteriormente). Si es igual a SLIDE, aparece por un lado y se desplaza hasta llegar al otro extremo, y se para ahí. Si es igual a ALTERNATE se desplaza alternativamente hacia un lado y otro, siempre dentro de los límites de la marquesina.

Ejemplo:

```
<MARQUEE BEHAVIOR=ALTERNATE>
```

Este texto se mueve a un lado y otro, sin desaparecer

```
</MARQUEE>
```

BGCOLOR

Con este atributo se modifica el color de fondo de la marquesina, de acuerdo con el método visto en el Capítulo 7.

Ejemplo:

```
<MARQUEE BGCOLOR="#FF7070">
```

Esta marquesina tiene un fondo de color rosa

```
</MARQUEE>
```

DIRECTION

Este atributo sirve para modificar la dirección hacia la que se dirige el texto. Por defecto es LEFT (izquierda). Se puede hacer que el texto se dirija hacia la derecha igualando este atributo a RIGHT.

Ejemplo:

```
<MARQUEE DIRECTION=RIGHT>
```

Este texto se dirige hacia la derecha

```
</MARQUEE>
```

SCROLLAMOUNT

Define la cantidad de desplazamiento del texto en cada movimiento de avance, expresado en pixels. Cuanto mayor es el número, más rápido avanza.

Ejemplo:

```
<MARQUEE SCROLLAMOUNT=50>
```

Doy saltos grandes

```
</MARQUEE>
```

SCROLLDELAY

Define el tiempo entre cada movimiento de avance, expresado en milisegundos. Cuanto mayor es el número más lento avanza.

Ejemplo:

```
<MARQUEE SCROLLDELAY =200> Espero mucho entre cada salto </MARQUEE>
```

LOOP

Especifica el número de veces que aparecerá el texto. Es indefinido por defecto.

HSPACE, VSPACE

Definen, respectivamente, la separación en sentido horizontal o vertical del texto que está fuera de la marquesina.

Como se ha dicho antes, esta etiqueta no es implementada por el Netscape. El texto que está dentro de la etiqueta de la marquesina se verá en este navegador como texto fijo normal.

Normalmente, si se hace uso de este recurso es porque se quiere destacar de una manera muy especial un texto, lográndose plenamente este objetivo en el Explorer, pero no así en el Netscape, en donde dicho texto pasaría totalmente desapercibido. Una manera de paliar este inconveniente, al menos en parte, es introducir la etiqueta de la marquesina dentro de una tabla (ver el Capítulo 10). De esta manera, el texto estará en

cierta medida destacado en el Netscape, y en el Explorer la marquesina se verá de una manera aún más estética.

Ejemplo:

```
<TABLE BORDER=1>
<TR><TD>
<MARQUEE> Marquesina dentro de una tabla </MARQUEE>
</TD></TR>
</TABLE>
```

Texto en movimiento con Javascript

Otra manera de conseguir texto en movimiento es haciendo uso de Javascript. Vamos a ver dos ejemplos: un scroll (deslizamiento) de un texto en la barra de estado del navegador y un scroll en una ventana dentro de la página.

Esto está implementado en los siguientes navegadores:

Netscape, versión 2.0 o superior.
Explorer de Microsoft 3.0 o superior.

El Javascript es un lenguaje distinto del HTML, que es el tema de este manual. En los dos ejemplos que vamos a ver a continuación, únicamente se mostrará la manera de incluirlos en nuestro documento HTML, pero sin pretender explicar su estructura.

Cómo incluir Javascript en un documento HTML

Recordemos que la estructura de un documento HTML es:

```
<HTML>
<HEAD>
<TITLE>
</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

El script (documento) en Javascript lo tenemos que colocar dentro de la cabecera, después del título. Es decir, entre las etiquetas </TITLE> y </HEAD>, contenido dentro de la etiqueta:

```
<SCRIPT LANGUAGE="JavaScript"> </SCRIPT >
```

Es decir, que queda de esta manera:

```
<HTML>
<HEAD>
<TITLE>
</TITLE>
<SCRIPT LANGUAGE="JavaScript">
[Aquí debe ir colocado el script]
```



```

</SCRIPT >
</HEAD>
<BODY>
</BODY>
</HTML>

```

Además de esto, se debe añadir algo dentro de la etiqueta <BODY>, como se indicará en cada caso.

Scroll en la barra de estado

El script es el siguiente:

```

<SCRIPT LANGUAGE="JavaScript">
<!--
/// Scroll en la barra de estado
var txt="Esta es la primera línea de texto que se desplaza "
+ " y esta es la segunda, puedes poner todas las"
+ " que quieras !           ";

function scroll()
{
  window.status = txt;
  txt = txt.substring(1, txt.length) + txt.charAt(0);
  window.setTimeout("scroll()",150);
}
//-->

</SCRIPT>

```

Observación sobre el texto: Se puede poner el texto en una sola línea, tan larga como se quiera, siempre que esté entre " y "; Pero como no debe de haber ningún salto de línea, si el texto es demasiado largo, es mejor dividirlo en distintas líneas, tal como está en el ejemplo.

Al final de la última palabra del texto conviene dejar una serie de espacios en blanco, para que no esté encadenado el comienzo con el final.

Dentro de la etiqueta <BODY> se debe añadir el atributo:

```

onLoad="scroll();"
quedando así:
<BODY onLoad="scroll();">

```

17. Publicación y promoción de una página

Este capítulo trata de responder a estas dos preguntas básicas:

¿Cómo coloco mi página en la red?

¿Cómo doy a conocer a los demás la existencia de mi página?

Publicación de una página

El propósito que tenemos al confeccionar una página del Web es, por supuesto, el de publicarla y ponerla a disposición de todo el mundo.

No podemos colocarla directamente en la red, pues necesitaríamos disponer para nosotros solos de un servidor, es decir, un ordenador conectado permanentemente con Internet y dotado de un programa específico, lo que supone una gran inversión económica y unos elevados conocimientos técnicos.

La solución es instalarla en un servidor conectado a la WWW (World Wide Web: la telaraña mundial), la red de servidores interconectados entre sí que nos permite acceder a cualquier página en cualquier parte del mundo, y navegar a través de ellas.

Los distintos canales para colocar una página en la red son:

Los servidores de las instituciones oficiales y académicas a sus miembros, para la difusión de información relacionada con esos organismos.

Los proveedores comerciales de acceso a Internet, o ISPs (Internet Service Providers) que ceden frecuentemente a sus usuarios un espacio determinado en sus servidores, para que puedan colocar sus páginas personales, bien sea gratuitamente o por una tarifa determinada.

¿Cómo se envían las páginas al servidor?

Una vez de haber confeccionado en nuestro disco duro la página Web, y estando ya lista para ser colocada en el servidor, surge la cuestión de cómo proceder para enviarla.

No se pueden dar aquí unas instrucciones precisas, pues depende de la manera establecida para ello por cada servidor. Generalmente se hace utilizando un programa de FTP (programa de envío y recepción de ficheros). Pero antes habrá que consultar en las páginas del servidor si existen las instrucciones pertinentes para el envío, o en su defecto realizar la consulta por email o por teléfono.

Promoción de una página

No basta con colocar nuestra página en la red. Por muy interesante que sea su contenido, y por bien diseñada que esté, para que los demás puedan acceder a ella tienen que tener conocimiento de su existencia. Es una labor nuestra la de promocionar la página, es decir, darla a conocer por todos los medios posibles.

Para comenzar, es conveniente incluir la dirección (o URL) de la página en la firma de nuestro programa de email, así como también en el de lectura de newsgroups, y mejor aún si además se incluye su título, o una frase que indique el contenido de la página. De esta manera se incita a visitarla al que esté interesado en ese tema.

Pero el paso más eficaz es dar de alta a nuestra página en sitios del Web especializados en almacenar y organizar direcciones. Estos sitios sirven como bases de datos a donde acude la gente en búsqueda de información sobre dónde encontrar las páginas del Web deseadas. Además, dar de alta una página en los buscadores es totalmente gratuito.

En líneas generales, se pueden distinguir dos tipos de estos sitios:

Los que están organizados como directorios, es decir, que catalogan las páginas por su contenido en categorías y sub-categorías. Para darse de alta en ellos, es necesario situarse primero en la categoría apropiada al contenido de nuestra página. Estos sitios sólo contienen las páginas de quienes se hayan dado de alta en ellos de manera voluntaria.

Otros sitios, los llamados motores de búsqueda (search engines) actúan de una forma completamente distinta. Utilizan unos programas (llamados comúnmente robots o arañas) que tienen la misión de rastrear continuamente el Web en búsqueda de páginas nuevas o renovadas. Para ello, van navegando de URL en URL a través de los enlaces que encuentran en las páginas, con la intención de catalogar el número máximo de ellas.

Si uno de estos robots visita nuestra página del Web, grabará el texto completo de cada una de las páginas (la principal y las sub-páginas). De esta manera, todas las palabras de todas las páginas de nuestro sitio son incorporadas a su base de datos. Cuando luego alguien haga una consulta en estos motores de búsqueda introduciendo una palabra que coincida con alguna de ellas, presentará nuestra página del Web como un resultado de la búsqueda.

Consejos para facilitar la labor a los motores de búsqueda

Hay ciertas cosas que podemos hacer para conseguir que nuestra página sea catalogada de la manera más adecuada en estos motores de búsqueda:

Utilizar palabras clave dentro de la etiqueta <TITLE>, haciendo que sea lo más descriptivo posible, porque cuando un motor de búsqueda presenta una página concreta como resultado de una búsqueda, lo hará reproduciendo las palabras que ha encontrado dentro de la etiqueta

Utilizar etiquetas <META> Son unas etiquetas que se colocan en la cabecera de la página (en la zona entre <HEAD> y </HEAD>), que sirven para suministrar una información detallada del contenido de una página, con lo que se obtiene un control mayor de cómo será catalogada la página. No todos los motores de búsqueda hacen uso de estas etiquetas, pero si las ponemos, las haremos mucho más accesibles a los motores de búsqueda que sí las utilizan.

Hay diferentes tipos de esta etiqueta, pero las que nos interesan ahora son la que hace referencia a la descripción (description) de la página y la que presenta las palabras clave (keywords) con las que la gente buscará una página como la nuestra en los motores de búsqueda.

Veamos por ejemplo las utilizadas para este manual:

```
<META NAME = "description" CONTENT="HTML Language Tutorial in spanish. Manual para la creacion de paginas del Web">
```

En este caso, lo que está incluido en el atributo CONTENT (contenido) es lo que presentará el motor de búsqueda, además del título de la página.

```
<META NAME="keywords" CONTENT="HTML tutorial WWW Web spanish Spain html manual guia">
```

En este otro caso, se incluyen en el atributo CONTENT las palabras claves para la búsqueda de nuestra página. Se pueden poner tantas como se crea oportuno, incluso sus plurales.

La utilización de estas etiquetas META es especialmente conveniente para las páginas que hacen uso de frames (ya que la página inicial es la de definición de los frames, que no tiene ninguna indicación del contenido de las otras páginas). También es muy conveniente para los que utilizan Javascript en el comienzo de sus páginas, ya que el código empleado puede tener cientos de palabras, y los robots están programados para dar mayor énfasis a las palabras que encuentran al principio que las situadas al final.

Poner un resumen del contenido de la página en el comienzo del texto

Hay motores de búsqueda que utilizan las primeras 25 palabras del texto de una página a modo de presentación de su contenido. Conviene, por tanto, hacer un breve resumen al inicio de la página, lo que por otra parte es siempre una práctica muy aconsejable.

Utilizar el atributo ALT en las imágenes iniciales

Como vimos en el Capítulo 4, dentro de la etiqueta de las imágenes se puede añadir el atributo ALT que sirve para poner un texto a la imagen, que será visto únicamente por quienes utilizan los navegadores en forma de sólo texto.

Hay motores de búsqueda que toman en cuenta el texto que encuentran de esta manera en las imágenes (sobre todo las iniciales) para hacer una descripción del sitio o para suministrar las palabras clave.

Sistemas de búsqueda en español

Generalmente están organizados como directorios, es decir, que para darse de alta en ellos, hay que situarse primero en la categoría que corresponde a nuestra página, o seguir las instrucciones oportunas. Algunos de los sitios catalogan únicamente páginas del propio país.

En principio, hay que darse de alta en cada uno de ellos individualmente. Pero como primera medida, es muy conveniente registrarse en Atajos Auto-Regístralo, en donde introduciendo los datos de nuestra página una sola vez, hará que se registre simultáneamente en numerosos sistemas de búsqueda, tanto en español como en inglés.

Sistemas de búsqueda en inglés

Están clasificados según el método necesario para registrarse en ellos.

Registro por medio de formularios

- A. Registro múltiple
- B. Buscadores de tipo general
- C. Buscadores de empresas

Registro en buscadores que están organizados como directorios

Hay que navegar entre los distintos directorios, hasta situarse en la categoría que conviene a la página que queremos registrar.

18. Contadores de visitas

En este capítulo se verá la manera práctica de poner un contador de visitas en una página del Web.

En primer lugar, tengamos presente una noción básica: un contador de visitas no se crea simplemente con una etiqueta del lenguaje HTML, sino que se genera en un programa CGI especial residente en el servidor.

Puede tratarse del servidor donde esté alojada nuestra página si es que tiene instalado un programa de este tipo, o en caso contrario se puede recurrir a otros servidores distintos que den este servicio, como veremos más adelante.

En teoría, también podríamos instalar nosotros mismos un programa de este tipo en el servidor para poder hacer uso de él. Pero es muy poco frecuente que los administradores de los servidores permitan a sus usuarios instalar programas ejecutables de este tipo, debido a los riesgos para la seguridad del servidor que ello supone (como la introducción de virus, por ejemplo).

No vamos a ver aquí cómo crear, conseguir o instalar este tipo de programas, sino cómo hacer uso de ellos, una vez que hayamos localizado alguno al que tengamos acceso.

¿Cómo funciona un contador?

Si queremos instalar en nuestra página un contador de accesos, debemos colocar en el documento HTML una etiqueta que haga una llamada a un programa instalado en un servidor que nos suministre este servicio. Estas etiquetas no son un standard del lenguaje HTML, sino que nos la tiene que suministrar quien nos dé el servicio, ya que depende del programa que tengan instalado en el servidor.

Servidores públicos de contadores

¿Qué se puede hacer si el servidor donde va a estar alojada nuestra página no tiene un programa de contadores para sus usuarios? En este caso se puede recurrir a otros servidores que dan este servicio de manera generalmente gratuita.

El único inconveniente que puede tener esta solución es que al cargar el usuario nuestra página, al llegar a la etiqueta del contador, la solicitará a este otro servidor, y puede ocurrir que esté inaccesible en ese momento, o sea lento en dar la respuesta.

<http://www.melodysoft.com>

19. Componer con estilo

Una vez de conocidas las técnicas para componer una página del Web, vamos a considerar los distintos aspectos necesarios para que esa página sea atractiva y efectiva, fácil de utilizar y de mantener, así como también los errores más frecuentes y lo que hay que evitar al componer un documento HTML.

Bastantes de los conceptos indicados a continuación ya han sido expuestos en sus capítulos correspondientes. Otros son nuevos, y otros son de puro sentido común. Pero todos ellos agrupados pueden dar una idea de conjunto sobre las directrices a seguir para confeccionar adecuadamente una página del Web.

Estructura de una página

Una página consiste básicamente en información que se ofrece al usuario. Pero se debe organizar esta información de una manera clara y efectiva.

Contenido: Un sitio del Web debe ofrecer contenido, es decir, información del tipo que sea, no una simple colección de enlaces a otras páginas. Evitar poner enlaces a páginas todavía vacías de contenido, en donde sólo haya un icono indicando "Página en obras".

Longitud: Si una página resulta excesivamente larga, es conveniente descomponerla en otras varias, para facilitar el tiempo de carga, y dar la oportunidad al usuario de elegir lo que le interese.

Cabeceras: Si la información que se suministra se trata principalmente de texto, es muy conveniente utilizar adecuadamente las cabeceras y las listas.

Las cabeceras del nivel 1 (H1) sirven para indicar los temas importantes, mientras que las del nivel 2 (H2) se utilizarán para señalar los sub-temas relacionados con los anteriores, etc.

Es importante recordar que el propósito de las cabeceras no es el de conseguir unos determinados tamaños de fuentes (para ello hay otro tipo de etiquetas), sino que se deben utilizar exclusivamente para organizar un documento en diversas secciones. Con este propósito, he aquí algunas recomendaciones acerca del correcto uso de las cabeceras:

Una cabecera no debe estar a más de un nivel del que la que le precede. Es decir, un elemento H3 no debería seguir directamente a uno H1.

Es incorrecto querer destacar de una manera extra lo que está dentro de la cabecera, utilizando etiquetas que hagan el texto aún más grueso, tales como o .

No se debe marcar el texto con las etiquetas H2 o H3, simplemente porque en nuestro navegador obtenemos así el tamaño de fuente y el grosor deseados, ya que en otros navegadores puede resultar de un tamaño grotescamente grande.

Composición

Es muy conveniente tratar de conseguir que las distintas páginas que componen un sitio del Web tengan un estilo consistente. Una composición uniforme de las páginas ayudará a los usuarios a navegar entre ellas, y dará personalidad al conjunto. Esta consistencia se puede obtener escogiendo un tema gráfico común al conjunto de las

páginas: puede ser un color de fondo, o un logo, o una manera particular de disponer los gráficos o el texto.

Aunque el contenido es lo principal, una presentación acertada y atrayente contribuirá en gran medida al éxito de una página del Web.

Escoger un título significativo: A veces se omite el título de un documento HTML, o se le pone uno inadecuado, porque no se ve en pantalla. Pero ese documento será conocido en el Web precisamente por su título.

Por tanto, conviene hacer que el título sea muy descriptivo, de tal manera que cuando alguien lo guarde en su bookmark pueda saber posteriormente de qué se trata. A ser posible, se debe dar una indicación en el título de la organización (por ejemplo, en el título de esta página se indica el nombre del manual, el número del capítulo, y el contenido del capítulo), con lo que su identificación sea completa fuera de su contexto.

Poner un párrafo introductorio: A menudo, la única indicación del contenido de una página viene suministrada por una imagen. Pero no hay que olvidarse que existen navegadores que funcionan sólo en modo texto (tal como Lynx, por ejemplo), así como que también hay una proporción de usuarios de navegadores gráficos que inhabilitan la carga de imágenes para ganar en velocidad.

Es conveniente, por tanto, poner un párrafo introductorio que exprese de manera clara y concisa el contenido de la página. Otra razón muy importante para hacerlo es que existen motores de búsqueda que catalogan las páginas precisamente por el contenido de las palabras iniciales (como se vió en el Capítulo 17).

Ayudas visuales: Guiar los ojos del usuario por medio de la estructura de la página, barras separadoras, tipos de fuentes, iconos, imágenes, etc. No utilizar el parpadeo más que para las emergencias.

Para hacer particiones de una página se pueden usar los separadores horizontales (<HR>) o tablas con bordes. Pero no hay que abusar de esto. Demasiados separadores dan un aspecto trozeado a la página.

Seleccionar un fondo de color o con imágenes: No todos los navegadores lo soportan, pero si se utiliza adecuadamente puede tener un impacto espectacular.

Es muy importante que el fondo no dificulte la lectura del texto. Si el color del texto es muy llamativo, es preferible usar un color de fondo pastel (ver el Cap. 7). Si se utiliza un color de fondo negro o muy oscuro hay que tener un cuidado muy especial en escoger un color de texto claro, que contraste suficientemente.

Comunicarse con los usuarios: Hay que ofrecer al usuario de la página la posibilidad de que pueda hacer comentarios, indicar posibles problemas, etc. Para ello es muy conveniente suministrar una dirección de email, o incluso un libro de visitas.

Asímismo, es muy útil poner la fecha de la última modificación. Si los cambios son frecuentes, también se puede proporcionar una lista cronológica de las modificaciones (como por ejemplo la lista de este manual)

Ofrecer referencias o otras páginas similares: Si un usuario está interesado por el contenido de nuestra página nos agradecerá mucho si le ofrecemos referencias a otras páginas con el mismo tema u otros relacionados con él. Hay que valorar el trabajo de los demás, y no ver tales páginas como rivales a la nuestra, sino al contrario, como una medida para enriquecerla.

Facilitar el mantenimiento de las páginas

Una cosa que facilita mucho el mantenimiento de una página es escribir el código HTML de una manera clara y ordenada, procurando poner cada etiqueta en su propia línea, dejando líneas en blanco entre los distintos bloques de código, y sobre todo, poner los oportunos comentarios en el documento HTML no visibles en pantalla, (ver Capítulo 2), cuando se hagan cosas un poco elaboradas.

Navegación

Hay que procurar por todos los medios el facilitar al usuario la navegación entre las distintas páginas. Si éstas son numerosas, el usuario no debe sentirse perdido como en un laberinto. Los medio que se pueden emplear para conseguirlo son:

Iconos de navegación : Como por ejemplo: . No olvidarse de incluir dentro de cada etiqueta el atributo ALT, que indica con un texto su función (ver el Cap. 4). Y sobre todo, es muy importante que por lo menos haya siempre un icono que conduzca a la página principal.

Índice con frames : Una de las funciones más útiles de los frames (ver Cap. 14) es la posibilidad de tener un índice dentro de un frame que se mantiene siempre a la vista, con lo que se facilita enormemente la navegación.

Barra de menú: Se puede utilizar como ayuda a la navegación una barra de menú, consistente en un mapa (ver Cap. 13), es decir una imagen que contiene zonas que enlazan con otras páginas. No olvidarse de suministrar los enlaces del mapa también en forma de texto.

Claridad en el texto de los enlaces: Hay que procurar que las palabras que forman parte de un enlace indiquen con claridad a dónde conduce ese enlace, y que estén integradas con el resto de la frase.

Evitar frases como ésta: "Puedes encontrar muchas direcciones útiles de HTML pulsando aquí"

Imágenes

Se consiguen presentaciones muy vistosas con las imágenes, pero pueden representar un problema para los usuarios con conexiones lentas. Incluir las sólo cuando añadan algo al contenido.

Si se juzga conveniente incluir imágenes grandes, hacerlo de tal manera que sea el usuario quien decida si las va a acargar o no. Para ello, suministrar enlaces a las imágenes con indicación de los kbytes, bien sea con un texto, bien sea con un thumbnail (versión muy reducida de la imagen, ver Cap. 8)

Se deben incluir dentro de las etiquetas de las imágenes estos atributos, que aunque son opcionales son muy convenientes:

Atributo ALT: Es un texto asociado a la imagen (ver Cap. 4). Muy conveniente para los que utilizan navegadores en forma texto (Lynx, etc.), o para los que usando navegadores gráficos, han inhabilitado la carga de imágenes para ganar en velocidad.

Atributos WIDTH y HEIGHT: Es una indicación de las dimensiones de la imagen (ancho y alto, ver Cap. 8). Es muy importante ponerlo en todas las etiquetas de las imágenes (incluso de las más pequeñas, iconos, etc.), pues facilita mucho la carga de la página, ya que primero se carga todo el texto y posteriormente se van rellenando los espacios reservados a las imágenes.

Con respecto a los GIFs animados, conviene no abusar de ellos poniendo demasiados en una misma página, no sólo por motivos estéticos, sino también porque esto puede provocar problemas en los ordenadores de algunos usuarios.

Frames

Los frames permiten dividir la pantalla en diferentes ventanas, con un documento HTML distinto en cada una de ellas. Esto permite a su autor ejercer un gran control sobre la disposición general y apariencia de la página, pero como todas las cosas, deben ser utilizados con cuidado.

Cosas a evitar en el uso de los frames:

Excesiva fragmentación de la pantalla: Si se van a utilizar más de dos frames, hay que evitar la impresión de que se ha "parcelado" la pantalla en múltiples ventanitas. Por lo menos uno de ellos debe ser mucho mayor que los demás, para que actúe como página principal, siendo los otros meros auxiliares para la navegación de lo que ocurre en el frame principal.

Disposición demasiado rígida de los frames: A veces, el autor planea cuidadosamente en su propio monitor el tamaño de cada frame, y para evitar que los usuarios "estropeen" ese efecto, elimina las distintas posibilidades que existen para poder manipularlos (redimensionarlos, deslizar su contenido, etc.).

Pero esto puede ser muy perjudicial para usuarios con resolución de pantalla distinta (sobre todo si es menor), ya que puede ocurrir que parte del contenido de los frames quede oculta, y debido a la rigidez con que han sido planeados, no haya manera de resolver este inconveniente. Conviene, por tanto, evitar poner los atributos NORESIZE o SCROLLING="no" dentro de la etiqueta <FRAME>.

Por otra parte, al dimensionar los frames con la etiqueta <FRAMESET>, conviene evitar hacerlo empleando valores absolutos. Es mejor hacerlo con porcentajes de pantalla o valores relativos.

Enlaces al exterior prisioneros dentro de un frame: A veces encontramos páginas con frames que contienen enlaces a otros sitios del Web que al ser pulsados se cargan dentro del mismo frame.

Nombres de los ficheros

Un sitio del Web es en realidad un conjunto de ficheros, bien sea de texto (los documentos HTML), bien sea binarios (imágenes, sonidos, etc.) que vamos a colocar en un servidor, para que sean accesibles a quien quiera visitarlos.

Pero debido a que la mayoría de los servidores están basados en sistemas Unix (aunque crecientemente se van utilizando otros sistemas), es muy conveniente, para evitarnos problemas al instalar nuestros ficheros en el servidor, considerar una particularidad referente a los nombres de los ficheros.

Resulta que en los sistemas más utilizados para confeccionar las páginas del Web (Windows, Mac) es indiferente la utilización de las mayúsculas y minúsculas en los nombres de los ficheros. Es decir, es lo mismo `Indice.html` que `INDICE.HTML` o que `indice.html` Pero esto no es así en los sistemas Unix, en donde los ejemplos anteriores se corresponderían a tres nombres distintos.

Si no se tiene esto en cuenta, puede ocurrir, por ejemplo, que una referencia que se haga en un documento HTML al fichero `Indice.html` resulte en un error de que no existe tal fichero debido a que en realidad se llama `indice.html`

Para evitar estos posibles problemas es muy conveniente tomar estas medidas desde que se comienza la confección de una página del Web:

Utilizar exclusivamente las minúsculas para nombrar los documentos HTML (como se ha hecho con los ejemplos prácticos de este manual). Igualmente, nombrar a todos los ficheros binarios (de imagen, sonido, etc.) también exclusivamente con minúsculas, cambiándoles el nombre si es preciso. Todas las referencias que se hagan dentro de un documento HTML a otros documentos o ficheros deben hacerse también siempre en minúsculas.

Algunos servidores exigen además de lo anteriormente indicado que los documentos HTML tengan la extensión `.html` en lugar de la de `.htm` Si es así, además de nombrar a los ficheros con dicha extensión, no olvidarse de hacer las referencias dentro de los documentos también con esta extensión.

Sonidos

Si se va a incorporar una música de fondo a una página conviene poner la etiqueta del sonido al final del documento HTML, con objeto de que se carguen primero el texto y las imágenes. No abusar con el tamaño de los ficheros de sonido.

Una alternativa muy aconsejable a los sonidos de fondo es la de dar al usuario la oportunidad de que sea él quien active el sonido, si así lo desea. (Ver esto en el Cap. 15)

No confeccionar una página para un navegador concreto

Al contrario de lo que cree mucha gente, una página del Web no se ve de manera idéntica en todos los navegadores.

Existen muchas variantes, aparte de los diferentes tipos de navegadores, tales como la antigüedad de las versiones, los distintos sistemas operativos o equipos.

Además, los usuarios pueden haber escogido una resolución de pantalla distinta a la nuestra, o haber variado la configuración de su navegador para ajustarla a su gusto.

Todo esto no quiere decir que no se puede hacer nada efectivo para resolver este problema. La mejor medida para que una página se vea correctamente en la mayoría de los navegadores es confeccionarla con naturalidad, sin forzar las cosas queriendo controlarlo todo hasta el último detalle y empleando para ello todo tipo de artificios, ya que la probabilidad de que se vea incorrectamente en otros navegadores aumentará de manera proporcional.

También es una buena medida tener más de un tipo de navegador, para comprobar si existen diferencias apreciables de uno a otro.

20. Applets de Java

Java™ es un lenguaje de programación orientado a objetos que ha sido desarrollado por la compañía Sun Microsystems, basándose en el lenguaje C++. Aunque es también un lenguaje de carácter general, su principal característica es la de ser independiente de cualquier plataforma, lo que le hace ser muy adecuado para ser utilizado en Internet, ya que puede ejecutarse en cualquier ordenador que tenga un navegador compatible con Java.

Esto quiere decir que el mismo programa se puede utilizar, sin ningún cambio, en un PC, Mac o máquina Unix. Esto es así porque un programa Java no corre directamente en el ordenador, sino que lo hace en una máquina virtual (la Máquina Virtual Java), quedando además confinado dentro de ella, lo que hace que sea un lenguaje seguro, ya que no puede acceder a otros recursos de nuestro ordenador.

Java y Javascript

Existe bastante confusión entre los términos Java y Javascript. Son dos lenguajes distintos.

El Javascript fue creado por Netscape inicialmente con otro nombre, adoptando finalmente (por evidentes razones de oportunismo) el de Javascript. Se trata de un lenguaje interpretado, cuyo código se incluye directamente en el documento HTML (En el Capítulo 16 se utilizaron varios ejemplos).

Nivel de dificultad del lenguaje Java

Aunque el lenguaje Java es más fácil que el C++, en el que está basado, se trata de un lenguaje de programación de alto nivel, y por tanto es muchísimo más difícil (para quien no tiene conocimientos previos) que el HTML, que en realidad no es más que una sencilla manera de formatear el texto de las páginas del Web.

Pero una cosa es programar en Java y otra mucho más sencilla es utilizar programas ya existentes, como veremos. Estos programas se llaman applets, y se pueden incrustar dentro de una página del Web para realizar todo tipo de tareas.

El propósito de este capítulo es mostrar cómo hacer uso de estos programas y dónde encontrarlos, y no el de aprender a programarlos.

¿Qué son los applets?

Los applets son pequeños programas ejecutables escritos en lenguaje Java, que podemos colocar en nuestro servidor, junto con el resto de ficheros que componen un sitio del Web (documentos HTML, ficheros de imagen, sonido, etc.)

Si en una de nuestras páginas hemos incrustado un applet (de la manera que se verá más adelante), cuando un usuario cargue la página, recibirá del servidor dicho applet (al igual que ocurre con una imagen o sonido), y se ejecutará en su navegador.

Pero para que esto ocurra así, es condición indispensable que el usuario esté utilizando un navegador compatible con Java: Netscape 2.x o superior, Explorer 3.0 o superior, o HotJava (navegador desarrollado por Sun, y escrito totalmente en Java). Quien

utilice un navegador no compatible, no verá la ejecución del programa (ni siquiera lo cargará), aunque podemos informarle de este hecho con un mensaje.

¿Para qué sirven los applets?

Con los applets se pueden conseguir efectos visuales y sonoros (incluso ambos a la vez), textos en movimiento, utilidades (por ej., relojes), pequeños programas educativos, juegos interactivos, presentaciones multimedia, etc. En definitiva, cualquier cosa que se pueda conseguir con un programa pequeño.

No es que haya ninguna limitación teórica en el tamaño de estos programas. La razón de que sean pequeños es la misma por la que no es razonable colocar en una página gigantescos ficheros de imagen o sonido, es decir, no hay que alargar en exceso su tiempo de carga.

¿Cómo se coloca un applet en una página?

Lo primero de todo, es necesario tener el programa del applet. El ejemplo de applet visto anteriormente, y que vamos a examinar a continuación con detalle, se llama SineText.class

Se trata de un applet que hace que un texto (elegido por el usuario) se desplace formando una onda sinusoidal, en dirección reversible con un click del ratón. Se pueden escoger los colores del fondo y de las letras, elegir entre algunos tipos de fuentes, su tamaño, velocidad de desplazamiento, etc.

Alguna observaciones:

Estos programas tienen la extensión .class. Hay que respetar escrupulosamente la disposición de las mayúsculas y minúsculas en su nombre. Es un verdadero programa binario, y por tanto no se puede leer en un procesador de textos, como podemos hacer con un documento HTML.

Para colocar el applet en la página, debemos utilizar las siguientes etiquetas (que se colocan donde queramos dentro del cuerpo del documento HTML, es decir entre las etiquetas <BODY> y </BODY>):

```
<APPLET CODE=SineText.class WIDTH=500 HEIGHT=100>
</APPLET>
```

Con esto se le indica al navegador que cargue un applet, cuyo código compilado es SineText.class, que se encuentra en el mismo directorio que el propio documento HTML, y que le reserve un área de 500x100 pixels.

Pero en este caso concreto, nos falta añadir los parámetros que personalizan el applet a nuestro gusto (sobre todo el texto que queremos que aparezca). Para ello se añaden etiquetas de este tipo:

```
<PARAM NAME=nombre_del_parámetro VALUE="contenido_del_parámetro">
```

Para escoger el texto, será, por ejemplo:

```
<PARAM NAME=Text VALUE="¡Gracias por tu visita!">
```

De momento no vamos a definir más parámetros. El resto de ellos serán los que ha escogido su autor por defecto. Las etiquetas del applet quedan de momento así:

```
<APPLET CODE=SineText.class WIDTH=500 HEIGHT=100>
<PARAM NAME=Text VALUE="¡Gracias por tu visita!">
</APPLET>
```

Este es el resultado.

(Para no recargar demasiado esta página, se van a ver los ejemplos en páginas separadas)

¿Cómo sabemos qué parámetros hay, cuáles son sus nombres y cuáles son los que hay por defecto? Pues sencillamente, leyendo la documentación del applet, ya que son variables que se le han ocurrido al autor al confeccionar el programa. Esto quiere decir que para otro applet distinto los parámetros serán también distintos.

En este applet, su autor ha definido los siguientes parámetros:

Text - Texto que se va a ver.

Rate - Número de imágenes por segundo.

Font - Fuente: Helvetica (defecto), TimesRoman o Courier.

FontSize - Tamaño de la fuente.

TextStyle - Estilo de la fuente: Plain, Bold (defecto) o Italic.

Speed - Velocidad de desplazamiento 1-10 (5 es por defecto)

Direction - Dirección inicial del desplazamiento. Left (defecto) o Right.

Background - Color del fondo. 000000 (negro) es por defecto.

Foreground - Color del texto. FF0000 (rojo) es por defecto.

Vamos a añadir algunos parámetros al ejemplo anterior (el orden en el que se colocan es indiferente). Variaremos también el área del applet a 600x75, y el texto del mensaje.

```
<APPLET CODE=SineText.class WIDTH=600 HEIGHT=75>
<PARAM NAME=Text VALUE="¡Gracias por tu visita! Vuelve pronto">
<PARAM NAME=Rate VALUE="4">
<PARAM NAME=Font VALUE="TimesRoman">
<PARAM NAME=Background VALUE="FFFFFF">
<PARAM NAME=Foreground VALUE="0000FF">
</APPLET>
```

En los dos últimos ejemplos, el usuario con un navegador no compatible no verá absolutamente nada, ya que el applet no se carga ni ejecuta.

Se puede poner un mensaje para que lo lean los que usan un navegador no compatible, y que sin embargo, es invisible para los que sí están viendo ejecutarse el applet. No hay más que añadirlo, sin ningún tipo de etiqueta, a continuación de los parámetros. Por ejemplo:

```
<APPLET CODE=SineText.class WIDTH=600 HEIGHT=75>
<PARAM NAME=Text VALUE="¡Gracias por tu visita! Vuelve pronto">
```

```
<PARAM NAME=Rate VALUE="4">  
<PARAM NAME=Font VALUE="TimesRoman">  
<PARAM NAME=Background VALUE="FFFFFF">  
<PARAM NAME=ForeGround VALUE="0000FF">  
Lo siento, pero no puedes ver este applet funcionando.  
</APPLET>
```


21. Trucos diversos

Aquí se pueden encontrar varios trucos útiles para las páginas. Algunos de ellos se basan en Javascript.

Distribución de programas desde una página del Web

Si tenemos un fichero ejecutable, es decir un programa, y lo queremos distribuir libremente, se puede conseguir fácilmente poniendo un enlace a dicho fichero. Por supuesto, hay que colocarlo previamente en el servidor, junto con los demás elementos que forman la página (documentos HTML, ficheros de imagen, de sonido, etc.)

Es muy conveniente que el fichero esté comprimido (por ejemplo, en formato .zip) para reducir su tiempo de carga, y por otra razón que se dirá más adelante.

Supongamos entonces que ya tenemos instalado en el servidor el fichero comprimido programa.zip. Para ponerlo accesible a los demás, simplemente hay que poner un enlace directo a este fichero, como por ejemplo de esta manera:

Para conseguir el programa, pulsa este `enlace`

Al pulsar el enlace resultante, el navegador da la opción de ejecutarlo o guardarlo. Se podría haber puesto el fichero sin comprimir, p. ej. como programa.exe, pero aparte de que ocupa más espacio, hay el peligro de que en algunos navegadores (como el Explorer) se ejecute inmediatamente, en vez de guardarlo.

Texto fijo de un enlace en la barra de estado

Como se vió en el Capítulo 3, un ejemplo de etiqueta para un enlace es:

```
<A HREF="http://www.microsoft.com">Microsoft</A>
```

Que da como resultado: Microsoft

Al pasar el cursor del ratón por encima del enlace vemos que aparece en la barra de estado del navegador (en la parte inferior de la pantalla) el URL o dirección del enlace, en este caso http://www.microsoft.com

Se puede hacer que aparezca el texto que queramos, y además que permanezca fijo hasta que se pase el cursor por encima de otro enlace. Para ello, basta con introducir lo siguiente, dentro de la etiqueta del enlace:

```
onMouseOver="window.status='Aquí el texto que se quiera';return true;"
```

En el ejemplo anterior podemos poner, por ejemplo:

```
<A HREF="http://www.microsoft.com" onMouseOver="window.status='Página principal de Microsoft';return true;">Microsoft</A>
```

Salto automático de pantalla

Si ponemos en una página la siguiente etiqueta:

```
<META HTTP-EQUIV="Refresh" CONTENT="x;URL=URL_de_la_siguiente_página">
```

una vez de que se cargue, permanecerá a la vista durante x segundos, y luego saltará automáticamente a otra página cuyo URL hayamos especificado.

Por ejemplo:

```
<META HTTP-EQUIV="Refresh" CONTENT="10;URL=http://www.nasa.com">
```

después de 10 segundos nos transportará automáticamente a la página de la NASA. Obsérvense el punto y coma después de la cifra de los segundos y la disposición de las comillas.

Esta etiqueta no se puede colocar en cualquier sitio del documento HTML. Debe ir situada en la cabecera (entre las etiquetas </TITLE> y </HEAD>).

Aunque lo implementan las últimas versiones del Netscape y del Explorer, hay algunos navegadores que no lo hacen, por lo que es conveniente suministrar, por si acaso, un enlace al URL deseado para que el propio usuario lo pueda activar.

A continuación, como un ejemplo completo, se va a crear una página que hará que salte automáticamente a la portada de este manual:

```
<HTML>
<HEAD>
<TITLE>Ejemplo de salto automatico de pantalla</TITLE>
<META HTTP-EQUIV="Refresh" CONTENT="5;URL=http://www.lander.es/webmaestro">
</HEAD>
<BODY>
Ejemplo de salto automatico de pantalla. Despues de 5 segundos debe saltar
AUTOMATICAMENTE a la portada de WebMaestro.
<P>Si esto no ocurre, pulsa este
<A HREF="http://www.lander.es/webmaestro">enlace</A>.
</BODY>
</HTML>
```

Botón de envío de un formulario con una imagen

En el Capítulo 11 se vió la etiqueta para conseguir el botón de envío de los datos de un formulario:

```
<INPUT TYPE="submit" VALUE="zzz">
```

En donde zzz era el texto que queríamos que apareciera en el botón. Se ilustra con el siguiente

ejemplo:

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
Escribe tu apellido:
<BR><INPUT TYPE="text" NAME="Apellido">
<P><INPUT TYPE="submit" VALUE="Enviar datos">
</FORM>
```

Menú de direcciones

En el Capítulo 11 (Formularios) se vió la manera de elegir entre distintas opciones de texto en un menú desplegable. Ahora vamos a ver un menú similar, pero en el que se puedan elegir entre distintas direcciones del Web, a donde será dirigido el usuario una vez que pulse un botón.

Veámoslo con un ejemplo:

```
<FORM>
<SELECT NAME="list">
<OPTION VALUE="http://www.microsoft">Microsoft
<OPTION VALUE="http://www.lander.es/webmaestro">WebMaestro
<OPTION VALUE="http://www.netscape.com">Netscape
<OPTION VALUE="http://www.infoseek.com">Infoseek
</SELECT><P>
<INPUT TYPE=button VALUE="Pulsa para ir"
onClick="top.location.href=this.form.list.options[this.form.list.selectedIndex].value">

</FORM>
```

Se pueden poner tantas opciones como se quiera. También se puede dirigir a páginas propias (como la opción WebMaestro, que dirige a la portada de este manual), pero en este caso se debe poner el URL completo de la página (<http://www.lander.es/webmaestro>), como si tratara de una página ajena.

Acceso a una página por medio de un password

Puede ocurrir que tengamos una página cuyo contenido no queremos que sea accesible por todo el mundo, sino sólomente por quien nos interese. Para conseguirlo, podemos suministrar un password o clave con el que se pueda acceder a dicha página.

La manera más rigurosa es hacerlo por medio de un CGI, es decir, un programa especial que se coloca en el servidor y que actúa de filtro. Pero, como se ha comentado en varias ocasiones en este manual, el inconveniente de esta solución es que es muy poco frecuente que los administradores de los servidores permitan a sus usuarios la instalación de estos programas, como medida de seguridad.

Una solución alternativa es utilizar el siguiente programa en Javascript (Véase en el Cap. 16 sobre cómo incluir un programa de Javascript en un documento HTML; recuérdese que el script se coloca en la cabecera, entre las etiquetas </TITLE> y </HEAD>).

```
<HTML>
<HEAD>
<TITLE>Ejemplo de acceso con password</TITLE>
<SCRIPT LANGUAGE="JavaScript">
function Saltar(pal) {
window.location=pal+".html"
}
</SCRIPT>
</HEAD>
```

```
<BODY>
<FORM>
Introduce la palabra clave para acceder a la página
<P>
<INPUT TYPE="password" NAME="palclave" SIZE=25 VALUE="">
<INPUT TYPE="button" VALUE="Acceder"
onClick="Saltar(this.form.palclave.value)">
</FORM>
</BODY>
</HTML>
```

Esto funciona de la siguiente manera: una vez escogida la clave (por ejemplo: xyz123), debemos dar a la página de acceso restringido precisamente este nombre: xyz123.html, es decir, se debe dar a la página restringida el nombre compuesto por la clave, más la extensión .html (o .htm)

No hay que cambiar ni añadir nada en el script. Si se utiliza las extensiones .htm en vez de .html la página se llamaría xyz123.htm pero entonces hay que sustituir la línea del script `window.location=pal+".html"` por `window.location=pal+".htm"`

22. Canales de chat en páginas Web

En la jerga de las comunicaciones se llama hacer chat (charla, en inglés) al intercambio de texto con otras personas en tiempo real.

Esto es una de las características principales de los servicios on-line (como Compuserve, AOL, MSN, etc.) para sus suscriptores. Internet tiene su propia versión, abierta a todos, llamada IRC (Internet Relay Chat), pero hasta hace poco, sólo se ha podido acceder al él por medio de programas cliente especiales, aparte de ser necesario saber de antemano a qué redes de servidores conectarse y cuáles canales pueden ser interesantes.

Pero todo esto está cambiando con la llegada del chat basado en páginas del Web, que es una variante del IRC que utiliza applets de Java (ver el capítulo 20). Cuando un usuario carga una página Web que contiene uno de estos chats, recibe el applet y su propio navegador lo ejecuta. El applet se encarga de establecer la conexión con el servidor de chat, y con el canal correspondiente, poniéndole en contacto directo con otros usuarios que estén conectados en ese momento con ese mismo canal.

¿Cómo se instala un canal de chat en una página del Web?

Existen unos servidores comerciales que suministran los applets y que se encargan de la gestión del chat, y que además lo hacen de una manera gratuita, pues todo se costea gracias a la publicidad (mientras se utilizan estos canales, los usuarios reciben periódicamente anuncios dentro de la ventana del chat). También suelen existir versiones de pago, con un mayor control del canal.

23. Hojas de estilo en cascada.

Las hojas de estilo representan un gran avance con respecto a cómo los diseñadores de páginas Web pueden aumentar grandemente su capacidad de mejorar la apariencia de sus páginas.

En su origen el Web fue concebido en un entorno científico, como un medio de intercambio de documentación, en el que lo esencial era el contenido, siendo la presentación algo completamente accesorio. Pero a medida que el Web era descubierto por toda clase de personas como un fantástico medio de difusión, las limitaciones del HTML con respecto al control sobre la presentación se convirtieron en una fuente continua de frustraciones.

Los nuevos diseñadores de páginas Web estaban acostumbrados a los medios tradicionales sobre papel, en los que tenían un control total sobre la apariencia. Han ido aprendiendo a utilizar diversos trucos para suplir las limitaciones estilísticas del HTML. Pero aunque su intención ha sido buena - mejorar la presentación de las páginas - las técnicas empleadas han tenido unos efectos secundarios a menudo desafortunados. Estas técnicas o trucos suelen funcionar para algunas personas, algunas veces, pero nunca para todas las personas y en todos los casos.

Ejemplos de estas técnicas

Utilización de etiquetas no estándar, inventadas por los creadores de algunos navegadores (como Netscape o Explorer). •Convertir el texto en imagen. •Usar imágenes transparentes para crear espacios en blanco. •Uso de tablas para forzar determinadas presentaciones. •Utilizar programas o lenguajes ajenos al HTML para conseguir determinados fines, etc.

Estas técnicas aumentan considerablemente la complejidad de las páginas del Web, tienen una flexibilidad limitada, y sobre todo, provocan problemas al ser utilizados por distintos navegadores o plataformas.

Las hojas de estilo devuelven la facilidad de controlar la presentación y sobrepasan la limitada gama de mecanismos de presentación que se han ido añadiendo al HTML durante estos últimos años. Por ejemplo, con ellas es fácil determinar el espacio entre las líneas de texto, el tipo y tamaño de las fuentes, el grosor de los márgenes, los colores usados para el texto o el fondo, así como el fomateo a voluntad de cualquier etiqueta.

Además hay otro aspecto muy interesante de las hojas de estilo, y es que separan por completo la información para la presentación de una página de su contenido, con lo que se facilita mucho el diseño y revisión de las páginas, pues se puede variar la presentación de una página, o de todo el conjunto de ellas, sin cambiar una sola línea del código del HTML.

Como veremos más adelante, se pueden definir variaciones de diseño por medio de las "clases". Por ejemplo, si se necesitan cinco estilos de párrafos diferentes, se pueden definir cinco clases para la etiqueta <P> : P.normal, P.indentado, P.subrayado, P.rojo y P.dobleespacio.

En este capítulo se hace una introducción a las hojas de estilo, pues se trata de un tema muy extenso. En los siguientes capítulos se verán con detalle los distintos atributos y valores así como distintos consejos para su mejor utilización.

¿Qué navegadores las soportan?

Debido a que se trata de una reciente innovación, hay un soporte todavía limitado para las Hojas de estilo en cascada. Actualmente sólo lo soportan el Explorer 3.0 en adelante, así como el Netscape 4.0 en adelante.

Las Hojas de estilo todavía no han sido incorporadas al vigente estándar del HTML, el HTML 3.2, pero lo van a ser en el próximo, el HTML 4.0, del cual se ha publicado ya un borrador. En concreto, la documentación referida a las Hojas de estilo se puede encontrar en esta página.

¿Cómo funcionan?

Las hojas de estilo en cascada nos permiten redefinir las reglas que utiliza el navegador para presentar una página del Web.

Todos los navegadores tienen sus reglas. Así, por ejemplo -como vimos en el capítulo 1, el HTML define que las cabeceras del nivel uno (el texto que está entre las etiquetas <H1> y </H1>) se muestre como un texto alineado a la izquierda, un par de puntos mayor que el resto, en negrita y en la fuente Times Roman.

Ejemplo: <H1> Cabecera de nivel 1 </H1> se verá como:

Cabecera de nivel 1

Cada etiqueta define sus propias reglas para mostrar el texto que engloba. Con las hojas de estilo podemos cambiar cada una de estas reglas si lo deseamos. Es decir, podemos modificar a nuestro antojo el comportamiento de cada etiqueta.

Hay tres maneras de añadir estilos a nuestras páginas Web.

1. Añadiendo instrucciones de estilo sólo a etiquetas concretas, o a un grupo de ellas.
2. Incluyendo las instrucciones de estilo en el documento HTML de una página concreta - de una manera análoga a cómo se incluye un script de Javascript. Esto permite cambiar la apariencia de una hoja entera, cambiando unas pocas líneas.
3. Enlazando todos documentos HTML de todas las páginas que componen un sitio del Web con un fichero de definición del estilo. De esta manera, se puede cambiar la apariencia de múltiples páginas retocando un solo fichero.

Podemos utilizar en nuestras páginas uno, dos o los tres métodos descritos, como se verá más adelante.

Estilo para etiquetas concretas o grupos de etiquetas

Este método es el apropiado si sólo se desea cambiar el estilo en unas secciones determinadas de una página. Con él podemos manipular las propiedades de alguna etiqueta o de algún grupo de etiquetas.

Se hace añadiendo el atributo STYLE, que contiene una serie de propiedades, dentro de la etiqueta correspondiente. Veamos el siguiente ejemplo:

```
<P>
```

"Este es un párrafo normal, sin márgenes a la izquierda ni a la derecha, con el color de su texto en negro".

```
<P STYLE="margin-left: 0.5in; margin-right: 0.5in; color:green">
```

"Quiero que este otro párrafo tenga un margen izquierdo de media pulgada, al igual que el margen derecho. Además quiero que su texto sea de color verde".

Que se verá (en los navegadores que soportan hojas de estilo) de esta manera:

"Este es un párrafo normal, sin márgenes a la izquierda ni a la derecha, con el color de su texto en negro".

"Quiero que este otro párrafo tenga un margen izquierdo de media pulgada, al igual que el margen derecho. Además quiero que su texto sea de color verde".

Como se ve en este ejemplo, hemos variado la manera de cómo se comporta habitualmente la etiqueta <P>. Podíamos haber conseguido otras muchas más cosas, como se verá más adelante.

Obsérvese la sintaxis del atributo STYLE. Se ponen entre comillas una serie de propiedades (por ejemplo, margin-left), y después de dos puntos se pone el valor de esa propiedad (en este caso, 0.5in o sea, 0.5 pulgadas). Las distintas propiedades deberán estar separadas por punto y coma.

Si queremos cambiar la apariencia de una sección entera (que agrupe un conjunto de etiquetas), se puede utilizar la etiqueta <DIV>, con la que definimos el estilo globalmente para esa sección.

En el siguiente ejemplo cambiamos el color (a rojo) y el tamaño de la fuente (a 16 puntos de altura) de un bloque de etiquetas por medio de la etiqueta <DIV>, lo que tiene el mismo efecto que si hubiéramos asignado este estilo separadamente a cada una de las etiquetas englobadas (<P>, y):

```
<DIV STYLE="font-size: 16pt; color: red">
```

```
<P>La definición del estilo (rojo y 16 puntos), afecta a todo el bloque de etiquetas.
```

```
<P>Este texto también es rojo y tiene 16 puntos de altura.
```

```
<P>Al igual que esta línea.
```

```
</DIV>
```


Inclusión global del estilo en un documento HTML

Se hace poniendo un bloque de instrucciones dentro de las etiquetas <STYLE> </STYLE>, que deberá estar colocado dentro de la cabecera del documento, después del título, entre las etiquetas </TITLE> y </HEAD> (de igual manera que se hace con los scripts de Javascript).

Esta etiqueta <STYLE> tiene un atributo, TYPE, que especifica el tipo de medio en que va a ser publicado en Internet, en nuestro caso será "text/css" (que permitirá a los navegadores que no soporten este tipo el ignorar la hoja de estilo). Es decir, la etiqueta queda de esta manera: <STYLE TYPE="text/css">

Por tanto, la estructura será la siguiente:

```
<HTML>
<HEAD>
<TITLE> </TITLE>

<STYLE TYPE="text/css">
[bloque de instrucciones de estilo]
</STYLE>

</HEAD>
<BODY>
[conjunto de todas las etiquetas que componen la página]
</BODY>
</HTML>
```

Veamos ahora con un ejemplo, cómo se escribe el bloque de instrucciones del estilo, incluido dentro de la etiqueta <STYLE> (en el capítulo siguiente se ve con detalle cada uno de los atributos):

```
<STYLE TYPE="text/css">
BODY {background: yellow ;font-size: 10pt; font-family: Arial; margin-left: 0.5in; margin-
right: 0.5in}
H1 {background: blue; font-size: 14pt;font-weight: bold; color: red}
H2 {font-size: 12pt;font-weight: bold; color: red}
DIV {background: URL(nubes.jpg)}
</STYLE>
```

Como se ve en este ejemplo, se ponen las distintas etiquetas a las que se quiere atribuir un estilo (en este caso BODY, H1, H2 y DIV, pero podrían haber sido otras cualesquiera); a continuación del nombre de cada etiqueta, y englobadas por los signos "{" y "}" van las distintas propiedades con sus respectivos valores, separadas unas de otras por un punto y coma.

Si se coloca este bloque en la cabecera del documento HTML, donde se ha indicado antes, veremos que cada vez que se usa una de las etiquetas que tienen una definición de estilo, no se comportará de la forma habitual, sino tal como la hemos definido. El resto de las etiquetas, que no están incluidas en el bloque de definición del

estilo, se comportarán de la forma habitual. Puede comprobarse lo dicho, viendo una página a la que se le ha aplicado el estilo del ejemplo anterior.

Enlazando distintos documentos a una hoja de estilo

Si nuestro sitio del Web está compuesto por muchas páginas, y queremos darles un estilo uniforme a unas cuantas, en vez de incluir un bloque de definición de estilo repetido en cada una de ellas (como se ha visto en la sección anterior), se puede establecer un enlace a un fichero de texto que contiene el bloque de definición del estilo.

Supongamos que queremos aplicar a unas cuantas páginas el estilo visto en el ejemplo anterior. Tendríamos que crear un fichero de texto como este:

```
BODY {background: yellow ;font-size: 10pt; font-family: Arial; margin-left: 0.5in; margin-right: 0.5in}
H1 {background: blue; font-size: 14pt;font-weight: bold; color: red}
H2 {font-size: 12pt;font-weight: bold; color: red}
DIV {background: URL(nubes.jpg)}
```

Obsérvese que no tiene ninguna etiqueta, pues no es un documento normal HTML, sino que es un fichero de texto que sólo contiene el bloque de definición del estilo.

Se guarda con el nombre que se quiera, pero tiene que tener necesariamente la extensión .css Supongamos que lo guardamos con el nombre de miestilo.css

En todas las páginas que queremos que tengan este estilo concreto, sólomente deberemos de añadir (en el mismo sitio de la cabecera que para el caso anterior, es decir entre </TITLE> y </HEAD>, la siguiente etiqueta:

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

Aquí se ha supuesto que el fichero miestilo.css se ha dejado en el mismo directorio donde están los documentos HTML. Si no es así, es decir, si está en otro subdirectorio en el servidor, habría que indicar la ruta (por, ejemplo: HREF="subdirectorio/miestilo.css").

Un ejemplo de este método lo constituye este manual, en donde todos los capítulos están enlazados con un fichero que contiene la hoja de estilo para todos ellos, llamado wmstyle.css (se puede ver en esta página su texto).

Mezclando los métodos

Como se ha visto anteriormente, hay tres métodos de aplicar las hojas de estilo en cascada.

¿Se pueden aplicar más de uno a la vez en una misma página?

Sí. Por ejemplo, en la confección de este manual se utilizan simultáneamente algunos de ellos. En las páginas del índice y de la portada se utilizan el segundo método (inclusión de la hoja de estilo en el documento HTML) y también el primero (inclusión del estilo en etiquetas concretas). Y en la totalidad de los capítulos se utilizan el tercero (enlace a la hoja de estilo) y el primero.

¿Qué ocurre si hay información contradictoria entre ellos?

Para evitar conflictos entre los distintos métodos usados simultáneamente, existe un orden de precedencia, es decir, cuál prevalecerá sobre el otro si dan órdenes contradictorias sobre un aspecto concreto.

El orden de precedencia es el siguiente (de mayor a menor):

- 1.Estilo dentro de una etiqueta.
- 2.Bloque de estilo en la cabecera del documento.
- 3.Enlace a un fichero que contiene la hoja de estilo.

Esto quiere decir que una orden de estilo, por ejemplo el color del texto, puesta dentro de una etiqueta, prevalecerá sobre la que esté indicada en los otros dos métodos, si es que se usan en esa página.

Lo más práctico es utilizar el tercer método (enlace a una hoja de estilo), para dar una apariencia consistente a todas las páginas, y si es necesario modificar un aspecto concreto en alguna de ellas, utilizar el primero o el segundo.

24. Hojas de estilo en cascada.

Atributos

A continuación se muestra en una tabla el resumen de los atributos que se pueden incluir en las hojas de estilo. Pulsando el enlace con el nombre del atributo nos lleva a la sección donde se comenta con más detalle.

font-size

El atributo font-size establece el tamaño del texto en puntos (pt), pulgadas (in), centímetros (cm), o pixels (px). Ejemplos:

```
{font-size: 12pt}
{font-size: 1in}
{font-size: 5cm}
{font-size: 24px}
```

font-family

el atributo font-family establece la fuente del texto. Se puede especificar una única fuente, como por ejemplo:

```
{font-family: Arial}
```

u otras fuentes alternativas, separadas por una coma, como por ejemplo:

```
{font-family: Arial, Helvetica}
```

En el ejemplo anterior, nos aseguramos que los sistemas que no soporten la fuente Arial, utilicen la fuente Helvetica. Es muy aconsejable especificar, como último recurso, un nombre genérico de familia de fuentes. Ejemplo:

```
{font-family: Arial, Helvetica, sans-serif}
```

Estos nombres genéricos de familia de fuentes (serif, sans-serif, cursive, fantasy, o monospace) tienen la ventaja de que son representados como las fuentes que tenga instaladas el usuario.

Si se hace referencia a una fuente cuyo nombre consiste en varias palabras (separadas por espacios en blanco), hay que englobarla entre comillas. Ejemplo:

```
{font-family: "Courier New"}
```

font-weight

El atributo font-weight establece el espesor de la fuente:

```
{font-weight: medium}
{font-weight: bold}
```

Los valores aceptados (extra-light, light, demi-light, medium, demi-bold, bold, y extra-bold) dependen en las fuentes que tenga instaladas el usuario. (Por ejemplo, el sistema del usuario puede que sólo permita medium y bold para una determinada fuente).

font-style

El atributo font-style establece la fuente como cursiva:

```
{font-style: italic}
```

En el borrador de la W3C se contemplan además otros posibles estilos (minúsculas, oblicuas, etc.)

line-height

Este atributo establece la separación entre líneas, que se puede expresar en puntos (pt), pulgadas (in), centímetros (cm), pixels (px) o porcentaje (%). Ejemplo:

```
{line-height: 20pt}
```

También se puede expresar como un porcentaje del valor por defecto:

```
{line-height: 150%}
```

En el Explorer 3.0, el espaciado se añade antes de las líneas, no después de ellas. Además este atributo se comporta de forma impredecible con texto que usa diferentes tamaños de texto en la misma línea.

color

Este atributo establece el color del texto de acuerdo con su valor hexadecimal (véase el Cap. 7), o usando los nombres de colores:

```
{color: #33CC00}
```

```
{color: red}
```

Los nombres de los colores son los siguientes:

```
<Picture><Picture><Picture><Picture>blacksilvergraywhite<Picture><Picture><Picture><P  
icture>maroonredpurplefuchsia<Picture><Picture><Picture><Picture>greenlimeoliveyellow  
<Picture><Picture><Picture><Picture>navybluetealaqua
```

text-decoration

Este atributo permite remarcar el texto. Los valores soportados son underline (subrayado), line-through (tachado), none (ninguno) e italic (cursiva). Ejemplos:

```
{text-decoration: underline}
```

```
{text-decoration: line-through}
```

margin-left, margin-right, y margin-top

Estos atributos establecen los márgenes (izquierdo, derecho y superior respectivamente) en el ámbito de una etiqueta. Se pueden especificar los márgenes en puntos, pulgadas (inches), centímetros o pixels. Por ejemplo:

```
BODY {margin-left: 0.5in;  
margin-right: 0.5in;  
margin-top: 1in}
```

Se pueden usar valores negativos. Entonces, en vez de contraerse el margen, se extenderá en la dirección opuesta.

text-align

Este atributo permite justificar los elementos HTML a la izquierda, al centro o a la derecha. Ejemplos:

```
{text-align: left}
{text-align: center}
{text-align: right}
```

text-indent

Además de establecer los márgenes, se puede provocar una indentación del texto (es decir, que la primera palabra del párrafo se desplace una cantidad deseada). Se puede expresar en puntos, pulgadas, centímetros o pixels. Por ejemplo:

```
P {text-indent: 0.5cm}
```

hace que los párrafos que se crean con la etiqueta <P>, empiecen con su primera línea indentada 0.5 centímetros.

Se pueden usar valores negativos, que sacan el texto hacia los márgenes.

background

Se utiliza este atributo para destacar secciones de una página, estableciendo un color de fondo o una imagen de fondo.

Para establecer un color de fondo, se especifica su valor hexadecimal (véase el Cap. 7), o un nombre de color (ver el atributo color visto anteriormente). Ejemplos:

```
{background: red}
{background: #6633FF}
```

También se puede colocar una imagen de fondo en el ámbito de la etiqueta. Es decir, se puede poner, por ejemplo, una imagen de fondo en un párrafo determinado.

Para colocar una imagen, se especifica el URL entre paréntesis (no entre comillas, como es lo habitual). Por ejemplo:

```
{background: URL(http://www.lander.es/webmaestro/docs/nubes.jpg)}
```

En este caso se ha puesto el URL absoluto, es decir, la referencia completa en el servidor. Pero es más conveniente hacerlo de forma relativa, es decir con respecto al documento HTML. Si el fichero de imagen y el documento HTML están en el mismo directorio, no hay que poner ningún URL, sino el nombre del fichero de imagen directamente. En el ejemplo anterior sería:

```
{background: URL(nubes.jpg)}
```

Vamos a ver una aplicación práctica: poner esta imagen de fondo a un párrafo, utilizando el método de incluir el estilo en la etiqueta <P>:

<P STYLE="background: URL(nubes.jpg)"> Este párrafo tiene un fondo con imágenes, al igual que con el HTML convencional se consigue para toda una página, pero en este caso está limitado al ámbito de esta etiqueta.

<P> En cambio, este otro párrafo no tiene imagen de fondo, porque no se le ha incluido este atributo.

Este es el resultado:

<Picture>Este párrafo tiene un fondo con imágenes, al igual que con el HTML convencional se consigue para toda una página, pero en este caso está limitado al ámbito de esta etiqueta.

En cambio, este otro párrafo no tiene imagen de fondo, porque no se le ha incluido este atributo.

25. Hojas de estilo en cascada.

Consejos de utilización

En este capítulo se amplían los conceptos vistos en los dos capítulos anteriores, y se introducen algunos nuevos sobre las hojas de estilo en cascada.

Agrupando distintos atributos

Supongamos que se quieren atribuir los mismos atributos a diferentes etiquetas, como por ejemplo:

```
H1 {font-size: 15pt;
    font-weight: bold;
    color: maroon}
H2 {font-size: 15pt;
    font-weight: bold;
    color: maroon}
H3 {font-size: 15pt;
    font-weight: bold;
    color: maroon}
```

Se pueden agrupar de esta manera:

```
H1, H2, H3 {font-size: 15pt;
            font-weight: bold;
            color: maroon}
```

Agrupación de los atributos del texto

En el capítulo anterior se han visto una serie de atributos relacionados con la apariencia del texto. Se pueden simplificar agrupándolos de una manera determinada. Así, por ejemplo, en lugar de:

```
P {font-weight: bold;
   font-style: italic;
   font-size: 12pt;
   line-height: 20pt;
   font-family: Times, serif; }
```

Se pueden agrupar en un único atributo llamado font:

```
P {font: bold italic 12pt/20pt Times, serif}
```

Nota: El orden de los atributos es significativo. Los atributos font-wight y font-style se deben especificar antes que los demás.

Agrupación de los atributos de los márgenes

También se pueden agrupar los tres distintos atributos para los márgenes (superior, derecho e izquierdo) en un único atributo llamado margin. Así, por ejemplo, en lugar de:


```
BODY {margin-top: 20px;
margin-right: -10px;
margin-left: -10px}
```

Se puede poner:

```
BODY {margin: 20px -10px-10px}
```

El orden de colocación es significativo. Debe ser: superior (top), derecho (right) e izquierdo (left). Si se pone un único valor, será aplicado a los tres márgenes.

Variaciones por medio de clases

En el capítulo 23 vimos que uno de los métodos era la inclusión global del estilo, en el que se definían los estilos de un bloque de distintas etiquetas. Vimos allí este ejemplo:

```
<STYLE TYPE="text/css">
BODY {background: yellow ; font-size: 10pt; font-family: Arial; margin-left: 0.5in; margin-right: 0.5in}
H1 {background: blue; font-size: 14pt; font-weight: bold; color: red}
H2 {font-size: 12pt; font-weight: bold; color: red}
DIV {background: URL(nubes.jpg)}
</STYLE>
```

Como se puede ver, se define para la etiqueta H2, por ejemplo, que su texto sea de color rojo (red). Pero esto hace que, obligatoriamente, todas las cabeceras de nivel H2 sean de este color en toda la página.

Pero nos podría interesar, por el motivo que sea, que unas veces tenga el color rojo y otras veces sea de otros colores. Para conseguirlo, se pueden emplear unas clases (variantes de esta etiqueta). Para ello, se define separadamente la etiqueta H2, seguida de un punto y un nombre que queramos, como por ejemplo:

```
H2.rojo {font-size: 12pt; font-weight: bold; color: red}
H2.verde {font-size: 12pt; font-weight: bold; color: green}
H2.azul {font-size: 12pt; font-weight: bold; color: blue}
```

y en la página, podremos utilizar, según nos convenga, una u otra de estas variantes de la siguiente manera:

```
<H2 CLASS=rojo>Esta cabecera será de color rojo</H2>
```

```
<H2 CLASS=verde>Esta cabecera será de color verde</H2>
```

```
<H2 CLASS=azul>Esta cabecera será de color azul</H2>
```

Aplicación de estilo a los enlaces

Las hojas de estilo también permiten modificar a voluntad la apariencia de los enlaces, asignando cualquiera de los atributos vistos (color del texto, tamaño de la fuente, existencia o no del subrayado, etc.).

Hay dos tipos de enlaces que se pueden modificar:

A:link enlaces que todavía no han sido visitados (pulsados)

A:visited enlaces que ya han sido visitados (pulsados)

Por ejemplo:

A:link {color: red}

A:visited {color: green}

hace que los enlaces sin visitar sean de color rojo, y una vez visitados se pongan de color verde.

Si se les aplica el atributo text-decoration visto en el capítulo anterior, ajustado al parámetro none (ninguno), hace que los enlaces no estén subrayados. Ejemplo:

A:visited {color: fucsia; text-decoration: none}

hace que los enlaces visitados sean de color fucsia y no estén subrayados.

Comentarios

Se pueden añadir comentarios propios a las hojas de estilo, que pueden servir de recordatorio posterior. Se pueden colocar en cualquier sitio de la especificación, siempre que vayan englobados entre los caracteres /* y */. Ejemplo:

```
H1 {font: 20pt/22pt bold; color=#00FF00} /*Color verde para las cabeceras de nivel 1 */
```

Diseñando para todos los navegadores

Si se utiliza el método de incluir un bloque de estilo en la cabecera (ver cap. 23), que consiste en poner las distintas instrucciones de estilo entre las etiquetas <STYLE> y </STYLE>, tal como en el ejemplo que se vió allí:

```
<STYLE TYPE="text/css">
BODY {background: yellow ; font-size: 10pt; font-family: Arial; margin-left: 0.5in; margin-right: 0.5in}
H1 {background: blue; font-size: 14pt; font-weight: bold; color: red}
H2 {font-size: 12pt; font-weight: bold; color: red}
DIV {background: URL(nubes.jpg)}
</STYLE>
```

en los navegadores que no implementen las hojas de estilo se ignorarán las etiquetas <STYLE> y </STYLE>, pero podría ocurrir que aparezca como texto el bloque de definición del estilo.

Para evitar esto, es conveniente englobar dicho bloque de información entre los símbolos <!-- y --> (como se vió en el Cap. 2), que son los que nos permiten hacer comentarios no visibles en la pantalla. Por tanto, es conveniente poner el bloque de definición del estilo de esta manera:

```
<STYLE TYPE="text/css">
<!--
```

```
BODY {background: yellow ; font-size: 10pt; font-family: Arial; margin-left: 0.5in; margin-right: 0.5in}
H1 {background: blue; font-size: 14pt; font-weight: bold; color: red}
H2 {font-size: 12pt; font-weight: bold; color: red}
DIV {background: URL(nubes.jpg)}
-->
</STYLE>
```

Aprovechando la herencia entre etiquetas

Como sabemos, las etiquetas de un documento HTML tienen una estructura definida, que de manera muy resumida se puede poner de esta forma:

```
<HTML>
<BODY>
.... (conjunto de etiquetas que conforman la página)
</BODY>
</HTML>
```

Como se puede ver, la etiqueta <BODY> engloba a todas las demás. Si se la asigna un estilo determinado a esta etiqueta, todos los elementos que estén dentro de la página (tablas, listas, párrafos, etc.) heredarán este estilo.

Por tanto, para establecer un estilo global a la página entera, lo más apropiado es atribuírselo a la etiqueta <BODY>. Por ejemplo:

```
BODY {font: 10pt/11pt Arial, Helvetica, sans-serif;
background: url(nubes.jpg);
margin-left: 0.5in;
margin-right: 0.5in}
```

establece para la página entera la fuente, separación entre líneas, imagen de fondo y espesor de los márgenes. Si se precisa que ciertos elementos concretos dentro de la página tengan otras características distinta a la general, entonces hay que definirlos por separado.

23. Enlaces interesantes.

<http://www.esgratis.net>
<http://www.dynamicdrive.com/>
<http://www.losrecursos.com>
<http://www.maestrosdelweb.com/editorial/default.asp>
<http://www.lawebdelprogramador.com/>
<http://personales.mundivia.es/cfsierra>
<http://www.aspfacil.com>
<http://www.asptutor.com>
<http://www.interbel.es>
<http://www.tusmanuales.com>
<http://www.rincondelvago.com>
<http://www.abcdatos.com/tutoriales/>