

Versiones del HTML

HTML 2.0

Cuando se produjo la explosión de Internet el estándar de HTML que circulaba era el 2.0 (establecido en noviembre del 95), de modo que cualquier navegador que encontremos será capaz de interpretarlo. Prácticamente todo lo que veamos en los próximos seis capítulos está contemplado por este estándar.

HTML 3.0 y 3.2

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

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

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

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

HTML 4.0

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

Caracteres Especiales

Si os habeis fijado en los ejemplos habréis visto que en los textos de los mismos no hay acentos, ni eñes, ni símbolos de abrir interrogación o exclamación. Esto es debido a los distintos juegos de caracteres que manejan los ordenadores.

Las máquinas manejan la información en formato binario (es decir, en unos y ceros). Estos, a su vez, forman números, los cuales se traducen en letras. ¿Cómo? Mediante tablas. Podemos asignar el valor 64 a la letra a, el 65 a la b, etc..

El problema está en que cada ordenador es de un fabricante distinto y puede adoptar una tabla diferente al resto. Para evitarlo existen diversos estándares y el más extendido es el ASCII. De hecho, actualmente todos los ordenadores tienen la misma tabla ASCII para los primeros 127 caracteres. Pero esa tabla no contiene vocales con acento, ni eñes, ni símbolos

de abrir interrogación o exclamación... Esto nos pasa por dejar que los norteamericanos sean quienes construyan las computadoras.

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

Caracteres extendidos en HTML

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

½

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

CÓDIGO	RESULTADO
<code>&aacute;</code> , <code>&Aacute;</code> , <code>&eacute;</code> , <code>&Eacute;</code> ,...	á, Á, é, É, í, Í, ó, Ó, ú y Ú
<code>&ntilde;</code> y <code>&Ntilde;</code>	ñ y Ñ
<code>&iquest;</code>	¿
<code>&iexcl;</code>	¡
<code>&ordm;</code>	º
<code>&ordf;</code>	ª
<code>&trade;</code> ◦ TM	™ ◦ ™
<code>&copy;</code>	©
<code>&reg;</code>	®
<code>&nbsp;</code>	(espacio en blanco que no puede ser usado para saltar de línea)

Caracteres de control

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

CÓDIGO	RESULTADO
<	<
>	>
&	&
"	“

Ahora podremos ver el ejemplo anterior (Solo hace referencia a una liga del ejemplo) corregido para incluir acentos y demás. También tenéis a vuestra disposición la tabla completa de caracteres del HTML 2.0.

Las siguientes definiciones de entidades SGML han sido extraídas del RFC original que define el HTML 2.0:

```

<!ENTITY nbsp CDATA " " — no-break space ->
<!ENTITY iexcl CDATA "¡" — exclamacion invertida ->
<!ENTITY cent CDATA "¢" — cent sign ->
<!ENTITY pound CDATA "£" — simbolo de la libra esterlina ->
<!ENTITY curren CDATA "¤" — general currency sign ->
<!ENTITY yen CDATA "¥" — simbolo del sign ->
<!ENTITY brvbar CDATA "¦" — barra vertical (partida) ->
<!ENTITY sect CDATA "§" — section sign ->
<!ENTITY uml CDATA "¨" — umlaut (dieresis) ->
<!ENTITY copy CDATA "©" — simbolo de copyright ->
<!ENTITY ordf CDATA "ª" — ordinal indicator, feminine ->
<!ENTITY laquo CDATA "«" — angle quotation mark, left ->
<!ENTITY not CDATA "¬" — not sign ->
<!ENTITY shy CDATA "¸" — soft hyphen ->
<!ENTITY reg CDATA "®" — registered sign ->
<!ENTITY macr CDATA "¯" — macron ->
<!ENTITY deg CDATA "°" — degree sign ->
<!ENTITY plusmn CDATA "±" — plus-or-minus sign ->
<!ENTITY sup2 CDATA "²" — superscript two ->
<!ENTITY sup3 CDATA "³" — superscript three ->
<!ENTITY acute CDATA "´" — acute accent ->
<!ENTITY micro CDATA "µ" — micro sign ->
<!ENTITY para CDATA "¶" — pilcrow (paragraph sign) ->
<!ENTITY middot CDATA "·" — middle dot ->
<!ENTITY cedil CDATA "¸" — cedilla ->
<!ENTITY sup1 CDATA "¹" — superscript one ->
<!ENTITY ordm CDATA "º" — ordinal indicator, masculine ->

```

<!ENTITY raquo CDATA “»” — angle quotation mark, right →
<!ENTITY frac14 CDATA “¼” — fraction one-quarter →
<!ENTITY frac12 CDATA “½” — fraction one-half →
<!ENTITY frac34 CDATA “¾” — fraction three-quarters →
<!ENTITY iquest CDATA “¿” — inverted question mark →
<!ENTITY Agrave CDATA “À” — capital A, grave accent →
<!ENTITY Aacute CDATA “Á” — capital A, acute accent →
<!ENTITY Acirc CDATA “Â” — capital A, circumflex accent →
<!ENTITY Atilde CDATA “Ã” — capital A, tilde →
<!ENTITY Auml CDATA “Ä” — capital A, dieresis or umlaut mark →
<!ENTITY Aring CDATA “Å” — capital A, ring →
<!ENTITY AElig CDATA “Æ” — capital AE diphthong (ligature) →
<!ENTITY Ccedil CDATA “Ç” — capital C, cedilla →
<!ENTITY Egrave CDATA “È” — capital E, grave accent →
<!ENTITY Eacute CDATA “É” — capital E, acute accent →
<!ENTITY Ecirc CDATA “Ê” — capital E, circumflex accent →
<!ENTITY Euml CDATA “Ë” — capital E, dieresis or umlaut mark →
<!ENTITY Igrave CDATA “Ì” — capital I, grave accent →
<!ENTITY Iacute CDATA “Í” — capital I, acute accent →
<!ENTITY Icirc CDATA “Î” — capital I, circumflex accent →
<!ENTITY Iuml CDATA “Ï” — capital I, dieresis or umlaut mark →
<!ENTITY ETH CDATA “Ð” — capital Eth, Icelandic →
<!ENTITY Ntilde CDATA “Ñ” — capital N, tilde →
<!ENTITY Ograve CDATA “Ò” — capital O, grave accent →
<!ENTITY Oacute CDATA “Ó” — capital O, acute accent →
<!ENTITY Ocirc CDATA “Ô” — capital O, circumflex accent →
<!ENTITY Otilde CDATA “Õ” — capital O, tilde →
<!ENTITY Ouml CDATA “Ö” — capital O, dieresis or umlaut mark →
<!ENTITY times CDATA “×” — multiply sign →
<!ENTITY Oslash CDATA “Ø” — capital O, slash →
<!ENTITY Ugrave CDATA “Ù” — capital U, grave accent →
<!ENTITY Uacute CDATA “Ú” — capital U, acute accent →
<!ENTITY Ucirc CDATA “Û” — capital U, circumflex accent →
<!ENTITY Uuml CDATA “Ü” — capital U, dieresis or umlaut mark →
<!ENTITY Yacute CDATA “Ý” — capital Y, acute accent →
<!ENTITY THORN CDATA “Þ” — capital THORN, Icelandic →
<!ENTITY szlig CDATA “ß” — small sharp s, German (sz ligature) →
<!ENTITY agrave CDATA “à” — small a, grave accent →
<!ENTITY aacute CDATA “á” — small a, acute accent →
<!ENTITY acirc CDATA “â” — small a, circumflex accent →
<!ENTITY atilde CDATA “ã” — small a, tilde →

<!ENTITY auml CDATA “ä” — small a, dieresis or umlaut mark →
<!ENTITY aring CDATA “å” — small a, ring →
<!ENTITY aelig CDATA “æ” — small ae diphthong (ligature) →
<!ENTITY ccedil CDATA “ç” — small c, cedilla →
<!ENTITY egrave CDATA “è” — small e, grave accent →
<!ENTITY eacute CDATA “é” — small e, acute accent →
<!ENTITY ecirc CDATA “ê” — small e, circumflex accent →
<!ENTITY euml CDATA “ë” — small e, dieresis or umlaut mark →
<!ENTITY igrave CDATA “ï” — small i, grave accent →
<!ENTITY iacute CDATA “í” — small i, acute accent →
<!ENTITY icirc CDATA “î” — small i, circumflex accent →
<!ENTITY iuml CDATA “ï” — small i, dieresis or umlaut mark →
<!ENTITY eth CDATA “ð” — small eth, Icelandic →
<!ENTITY ntilde CDATA “ñ” — small n, tilde →
<!ENTITY ograve CDATA “ò” — small o, grave accent →
<!ENTITY oacute CDATA “ó” — small o, acute accent →
<!ENTITY ocirc CDATA “ô” — small o, circumflex accent →
<!ENTITY otilde CDATA “õ” — small o, tilde →
<!ENTITY ouml CDATA “ö” — small o, dieresis or umlaut mark →
<!ENTITY divide CDATA “÷” — divide sign →
<!ENTITY oslash CDATA “ø” — small o, slash →
<!ENTITY ugrave CDATA “ù” — small u, grave accent →
<!ENTITY uacute CDATA “ú” — small u, acute accent →
<!ENTITY ucirc CDATA “û” — small u, circumflex accent →
<!ENTITY uuml CDATA “ü” — small u, dieresis or umlaut mark →
<!ENTITY yacute CDATA “ý” — small y, acute accent →
<!ENTITY thorn CDATA “þ” — small thorn, Icelandic →
<!ENTITY yuml CDATA “ÿ” — small y, dieresis or umlaut mark →

Enlaces

Las siglas HTML significan *HyperText Markup Language*, lo que para nosotros quiere decir que es un lenguaje para hipertexto. Existen múltiples formatos de hipertexto (por ejemplo, los ficheros de ayuda de Windows) y lo que tienen en común es que todos poseen enlaces.

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

La etiqueta <A>

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

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

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

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

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

Las URLs

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

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

donde el servicio podrá ser uno de los siguientes:

http

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

https

Es una innovación sobre el anterior, que nos permite acceder a servidores (generalmente comerciales) que nos ofrecen el uso de técnicas de encriptación para proteger los datos que intercambiamos con él de terceras personas.

ftp

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

mailto

Para poder mandar un mensaje. Por ejemplo, la URL `mailto:multivac@idecnet.com` me mandaría un mensaje a mí.

news

Para poder acceder a foros de discusión (mal traducidos a veces como grupos de noticias). Se indica el servidor y el grupo. Por ejemplo `news://news.iberet.es/es.comp.demos` nos conectaría con el foro `es.comp.demos` en el servidor nacional de Telefónica.

telnet

No es implementado generalmente por los navegadores, que suelen invocar un programa externo. Nos permite conectarnos con otros ordenadores y entrar en ellos como si nuestro ordenador fuese una terminal del mismo.

La dirección de la máquina puede ser, o bien una serie de cuatro números entre 0 y 255 (123.3.5.65) o bien algo más fácil de recordar como es una serie de palabras separadas por puntos (www.programacion.net). El puerto generalmente no se indica, ya que el servicio predetermina uno.

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

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

```
ruta_relativa/fichero
```

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

Anclas

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

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

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

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

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

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

Estructura del Documento

La estructura de un documento HTML se puede resumir así:

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

En el tipo de documento deberemos especificar a que estándar del HTML responde nuestra página entre una de las siguientes opciones:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
```

-

Cumple el estándar HTML 2.0

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

-

Cumple el estándar HTML 3.2

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"  
"http://w3.org/TR/REC-html40/loose.dtd">
```

-

Cumple el estándar HTML 4.0

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"  
"http://w3.org/TR/REC-html40/strict.dtd">
```

_Cumple el estándar HTML 4.0 y no contiene además elementos desaconsejables

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
"http://w3.org/TR/REC-html40/frameset.dtd">
```

Es una definición de marcos que cumple el estándar HTML 4.0

El HTML 4.0 considera desaconsejables aquellos elementos que, aún siendo soportados, han sido sustituidos por otros más potentes y, por ello, es posible que sean eliminados del estándar HTML en el futuro.

La cabecera

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

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

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

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

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

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

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

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

El cuerpo

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

PARÁMETRO	UTILIDAD
BACKGROUND	Permite definir un gráfico de fondo para la página.
BGCOLOR	Permite definir el color de fondo de la página.
BGPROPERTIES	Cuando es igual a <code>FIXED</code> el gráfico definido como fondo de la página permanecerá inmóvil aunque utilicemos las barras de desplazamiento.
TEXT	Cambia el color del texto.
LINK	Cambia el color de un enlace no visitado (por defecto azul).
VLINK	Cambia el color de un enlace ya visitado (por defecto púrpura).
ALINK	Cambia el color que toma un enlace mientras lo estamos pulsando (por defecto rojo).
LEFTMARGIN TOPMARGIN	y Especifican el número de pixels que dejará de margen entre el borde de la ventana y el contenido de la página. Se suelen utilizar para dejarlos a cero.
MARGINWIDTH MARGINHEIGHT	y Más o menos equivalentes a los anteriores, pero éstos funcionan en Netscape.

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

Formateo Básico

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

Formato del párrafo. Estas son las etiquetas más importantes (excluyendo algunas que veremos más adelante):

ETIQUETA	UTILIDAD	RESULTADO
<code><P></code>	Sirve para delimitar un párrafo. Inserta una línea en blanco antes del texto.	Soy un párrafo
<code><CENTER></code> ... <code></CENTER></code>	Permite centrar todo el texto del párrafo.	Yo soy normal Yo estoy centrado
<code><PRE WIDTH=x></code> ... <code></PRE></code>	Representa el texto encerrado en ella con un tipo de letra de paso fijo. Muy útil a la hora de representar código fuente. El parámetro WIDTH especifica el número máximo de caracteres en una línea.	Estoy en paso fijo
<code><DIV ALIGN=x></code> ... <code></DIV></code>	Permite justificar el texto del párrafo a la izquierda (<code>ALIGN=LEFT</code>), derecha (<code>RIGHT</code>), al centro (<code>CENTER</code>) o a ambos márgenes (<code>JUSTIFY</code>)	Yo estoy a la izquierda Yo al centro Y yo a la derecha (recuerda a la política esto, oye) Yo soy el más chulo, porque estoy en todos los lados.
<code><ADDRESS></code> ... <code></ADDRESS></code>	Para escribir direcciones (de esas donde vive la gente, no electrónicas).	<i>Daniel Rodríguez Herrera C/Ecuador 9, 1ºB 28220 Majadahonda</i>
<code><BLOCKQUOTE></code> ... <code></BLOCKQUOTE></code>	Para citar un texto ajeno. Se suele implementar dejando márgenes tanto a izquierda como a derecha, razón por la que se usa habitualmente.	<i>Me gustaría reencarnarme en las yemas de los dedos de Warren Beatty (Woody Allen)</i>

Las 6 cabeceras

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

ETIQUETA	RESULTADO
<code><H1> ... </H1></code>	Cabecera de nivel 1
<code><H2> ... </H2></code>	Cabecera de nivel 2
<code><H3> ... </H3></code>	Cabecera de nivel 3
<code><H4> ... </H4></code>	<i>Cabecera de nivel 4</i>
<code><H5> ... </H5></code>	Cabecera de nivel 5
<code><H6> ... </H6></code>	<i>Cabecera de nivel 6</i>

Estas etiquetas se pueden definir como de formato de párrafo pero por su importancia he preferido tratarlas aparte. No resulta recomendable utilizarlas para aumentar o disminuir el tamaño del tipo de letra, ya que cada navegador los muestra de manera diferente. Se usan para dividir correctamente en secciones nuestra página, tal y como se hace en un documento de texto normal.

[Cambiando el tipo de letra](#)

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

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

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

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

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

Otros elementos

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

ETIQUETA	UTILIDAD	RESULTADO
<code><HR></code>	Inserta una barra horizontal.	<hr/>
<code>
</code>	Salto de línea.	Hay un antes y un después de saltar a otra línea
<code><!-- ... --></code>	Comentarios.	Esto se escribe y <code><!-- esto no --></code>

Aquí tenemos en un ejemplo casi todas las etiquetas estudiadas:

Mi segunda pagina

La primera version de esta pagina

Esta es mi primera pagina (chispas). Por el momento no se que tendra, pero dentro de poco pondre aqui muchas cosas interesantes.

La primera version de esta pagina, ahora actualizada

Esta es mi **primera** pagina (*chispas*). Por el momento no se que tendra, pero dentro de poco pondre; *aqui* muchas cosas **interesantes**.

Y esta es la segunda version

Ya tenemos aquí las cosas interesantes. Tenemos a mi novia Loli diciendo que

Como me pongas en tu pagina te atizo.

También podemos a ver a mi querido perro fufu diciendo *guau, guau* y, sobre todo, el código fuente de este ejemplo:

```
<HTML>

<HEAD>
  <TITLE>Mi segunda pagina</TITLE>
</HEAD>

<BODY>
<DIV ALIGN=CENTER><H1>Mi segunda pagina</H1></DIV>

<H2>La primera version de esta página</H2>
<HR>
<P>Esta es mi primera pagina (chispas). Por el momento no se que
tendra, pero dentro de poco pondre aqui muchas cosas
interesantes.

<BR><BR><H2>La primera version de esta pagina, ahora actualizada</H2>
<HR>
<DIV ALIGN=CENTER>Esta es mi <STRONG>primera</STRONG> pagina (<I>chispas</I>). Por el
momento no se que tendra, pero <S>dentro de poco</S> pondre
<EM>aqui</EM> muchas cosas <B>interesantes</B>.</DIV>

<BR><BR><H2>Y esta es la segunda versión</H2>
<P>Ya tenemos aqui las cosas interesantes. Tenemos a mi novia
<BIG>Loli</BIG> diciendo que <BLOCKQUOTE>Como me pongas en tu pagina te atizo.</BLOCKQUOTE>
Tambien podemos a ver a mi querido perro <SMALL>fufu</SMALL> diciendo <CITE>guau, guau</CITE> y,
sobre todo, el código fuente de este ejemplo:
<PRE>
<!-- Aqui estaria el código fuente que no voy a poner porque
    entraríamos en un círculo vicioso.-->
</PRE>
</BODY>
</HTML>
```

Formateo Fino

Formateo fino

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

Cambio de color

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

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

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

El modo de indicar el color RGB es el siguiente:

```
<FONT COLOR="#FF0000">D</FONT>
<FONT COLOR="#EF0000">E</FONT>
<FONT COLOR="#DF0000">G</FONT>
<FONT COLOR="#CF0000">R</FONT>
<FONT COLOR="#BF0000">A</FONT>
<FONT COLOR="#AF0000">D</FONT>
<FONT COLOR="#9F0000">A</FONT>
<FONT COLOR="#8F0000">D</FONT>
<FONT COLOR="#7F0000">O</FONT>
```

Lo que nos mostraría lo siguiente:

DEGRADADO

La primera componente en hexadecimal es el rojo, la segunda el verde y la tercera el azul (Red Green Blue, RGB).

Tamaños del texto

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

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

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

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

Tipo de letra

Por último, podemos especificar el nombre del tipo de letra que queremos utilizar gracias al parámetro **FACE**. Como en principio no tenemos manera de saber que tipo de letra tiene instalado el ordenador del usuario que está viendo nuestras páginas, podemos indicar más de uno separado por comas. Si el navegador no encuentra ninguno seguirá utilizando el que tiene por defecto:

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

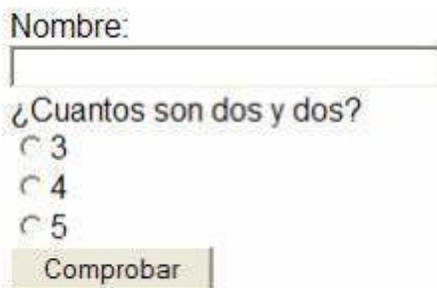
De todos modos es recomendable no utilizar con fe ciega este atributo en Internet, ya que impide que todos puedan ver nuestras páginas como nosotros. E Internet, siempre que nos lo permitan Microsoft y Netscape, debe ser lo más estándar posible.

Formularios

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

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

se verá así:



Nombre:

¿Cuantos son dos y dos?
 3
 4
 5

El botón no hace nada porque no hemos definido qué debe hacer, así que sed buenos y no lo pulséis.

Todos los elementos de un formulario deben estar encerrados entre **<FORM>** y **</FORM>**. Como parámetros cabe destacar tres. **ACTION** define el URL que deberá gestionar el

formulario. Puede ser una dirección de correo (precedida del inevitable `mailto:`, en cuyo caso deberemos añadir el parámetro `ENCTYPE="text/plain"` para que lo que recibamos resulte legible.

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

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

Cajas de texto

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

`<INPUT TYPE=TEXT>`

`<INPUT TYPE=PASSWORD>`

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

PARÁMETRO	UTILIDAD
<code>SIZE</code>	Tamaño de la caja de texto.
<code>MAXLENGTH</code>	Número máximo de caracteres que puede introducir el usuario.
<code>VALUE</code>	Texto por defecto que contendrá la caja.

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

`<TEXTAREA>`
Por
`</TEXTAREA>`

defecto

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

PARÁMETRO	UTILIDAD
ROWS	Filas que ocupará la caja de texto.
COLS	Columnas que ocupará la caja de texto.

Opciones

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

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

3
 4
 5

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

PARÁMETRO	UTILIDAD
VALUE	Este es el valor que asignará a la variable.
CHECKED	Si lo indicamos en una de las opciones esta será la que esté activada por defecto.

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

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

Netscape Explorer Opera Lynx Otros

Los parámetros que admite **SELECT** son las siguientes:

PARÁMETRO	UTILIDAD
SIZE	El número de opciones que podremos ver. Si es mayor que 1 veremos una lista de selección y, si no, veremos una lista desplegable.

MULTIPLE	Si lo indicamos podremos elegir más de una opción.
----------	--

Y OPTION estos:

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

Botones del formulario

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

<pre><INPUT TYPE=SUBMIT>
 <INPUT TYPE=RESET></pre>	
---	--

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

Otros elementos

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

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

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

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

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

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

Imágenes

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

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

El parámetro **SRC** especifica el nombre del fichero que contiene el gráfico. Los formatos estándar en la red son el GIF y el JPG. La últimas versiones de Netscape y Explorer aceptan también el formato PNG. El parámetro **ALT** especifica el texto que se mostrará en lugar del gráfico en aquellos navegadores que no sean capaces de mostrarlos (como el Lynx) y en el supuesto de que el usuario los haya desactivado. Algunos navegadores lo muestran cuando pasamos el ratón por encima de la imagen. Es por eso que, aunque algunos usuarios no lo lleguen a ver nunca, conviene ponerlo siempre. De hecho, el estándar HTML 4.0 obliga a hacerlo. Existen dos atributos que, aunque opcionales, conviene indicar siempre: la altura y la anchura del gráfico en pixels. De este modo, el navegador puede mostrar un recuadro del tamaño de la imagen mientras la va leyendo de la red y así poder mostrar el resto de la página correctamente mientras tanto.

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



Se ve así: Para los menos avezados en inglés, decir que **WIDTH** es la anchura y **HEIGHT** la altura. [Imágenes y enlaces](#)

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

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



Se ve así: Sin embargo,

```
<A HREF="http://www.netscape.com">  
<IMG SRC="graficos/dwnldns.gif" ALT="Netscape 4.0" WIDTH=88 HEIGHT=31  
  BORDER=0>  
</A>
```



Se ve así:

Alineación respecto al texto

Para poder maquetar conjuntamente texto y gráficos, el HTML proporciona, por medio del parámetro **ALIGN**, las siguientes maneras de alinear una imagen respecto del texto que la acompaña:

VALOR DE ALIGN	UTILIDAD	RESULTADO
TOP	Coloca el punto más alto de la imagen coincidiendo con más alto de la línea de texto actual.	 Este es el texto
MIDDLE	Alinea el punto medio (en altura) de la imagen con la base del texto.	 Este es el texto
BOTTOM Por defecto	Alinea el punto más bajo de la imagen con la base del texto.	 Este es el texto
LEFT 	Coloca la imagen a la izquierda del texto.	 Este es el texto
RIGHT 	Coloca la imagen a la derecha del texto.	Este es el texto 

Hay que aclarar que la base del texto es la línea donde descansan casi todas las letras del alfabeto excepto algunas como la p, la g o la j.

Listas

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

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

```
</tipo_lista>
```

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

Listas desordenadas

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

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

se verá como

- Primer elemento
- Segundo elemento

La etiqueta `` admite estos parámetros:

PARÁMETRO	UTILIDAD	RESULTADO
<code>COMPACT</code>	Indica al navegador que debe representar la lista de la manera más compacta posible.	<ul style="list-style-type: none"> • Primer elemento • Segundo elemento
<code>TYPE="disc", "circle", "square"</code>	Indica al navegador el dibujo que precederá a cada elemento de la lista. Para mayor flexibilidad se admite también como parámetro de <code></code> .	<ul style="list-style-type: none"> • Tipo disc • Tipo circle • Tipo square

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

Listas ordenadas

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

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

```
</OL>
```

se verá como

1. Primer elemento
2. Segundo elemento

La etiqueta `` admite estos parámetros:

PARÁMETRO	UTILIDAD	RESULTADO
<code>COMPACT</code>	Indica al navegador que debe representar la lista de la manera más compacta posible.	<ol style="list-style-type: none"> 1. Primer elemento 2. Segundo elemento
<code>TYPE="1", "a", "A", "i", "I"</code>	Indica al navegador el tipo de numeración que precederá a cada elemento de la lista. Para mayor flexibilidad se admite también como parámetro de <code></code> .	<ol style="list-style-type: none"> 1. Tipo 1 2. Tipo a 3. Tipo A 4. Tipo i 5. Tipo I
<code>START="num"</code>	Indica al navegador el número por el que se empezará a contar los elementos de la lista.	<ol style="list-style-type: none"> 1. Primer elemento 2. Segundo elemento
<code>VALUE="num"</code>	Atributo de <code></code> , actúa como <code>START</code> pero a partir de un elemento predeterminado.	<ol style="list-style-type: none"> 1. Primer elemento 2. Segundo elemento 3. Tercer elemento

Listas de definiciones

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

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

se verá como

Primer elemento

Es un elemento muy bonito.

Segundo elemento

Este, en cambio, es peor.

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

Mi primer Página

El código

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

  <BODY>
    <CENTER><H1>Mi Primera pagina</H1></CENTER>
    <HR>
    <P>Esta es mi primera pagina (chispas). Por el
      momento no se que tendra, pero dentro de poco
      pondre aqui muchas cosas interesantes.
    </BODY>
</HTML>
```

La explicación

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

```
<ETIQUETA parámetros> ... </ETIQUETA>
```

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

```
<HTML> ... </HTML>
```

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

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

Dentro de la cabecera sólo hay otra etiqueta. Es la única imprescindible: el título de la página. Es lo que veremos como título de la ventana en los navegadores que lo permitan. Es como se conocerá nuestra página en algunos buscadores y en la agenda de direcciones (*bookmarks*) de los usuarios. Por tanto, parece importante pensarnos bien como llamarla.

El cuerpo del documento

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

```
<BODY> ... </BODY>
```

Luego pondremos el título algo recalcado:

```
<CENTER><H1> ... </H1></CENTER>
```

Con esto colocaremos el texto centrado (`<CENTER>`) y en formato `<H1>` (cabecera 1) que nos asegura que aumentará el tamaño del tipo de letra lo suficiente como para que se vea bastante resaltado. Luego separamos ese título que le hemos puesto a la página del texto por medio de una línea horizontal:

```
<HR>
```

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

queremos separar el texto de la línea horizontal con un espacio vertical correspondiente a un párrafo nuevo le decimos:

```
<P>Esta es mi primera pagina (chispas). Por el  
momento no sé que tendrá, pero dentro de  
poco pondré aquí muchas cosas interesantes.
```

En el siguiente capítulo veremos muchas etiquetas que nos permitirán cambiar el aspecto de nuestros textos.