**Introducción a la programación HTML**

HTML Es el lenguaje de marcado estándar utilizado para los documentos del World Wide Web. HTML utiliza etiquetas (TAGs) para indicar cómo deben mostrar los exploradores Web los elementos de página, como texto y gráficos, y cómo deben responder a las acciones del usuario.

**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.

Ejemplo

<HTML>

<HEAD>

<TITLE>Mi primera pagina</TITLE>

</HEAD>

<BODY>

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

<HR>

<P>Esta es mi primera pagina HTML. Por el momento solo se muestra unas cuantas cosas 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 habrás 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 H1) 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:

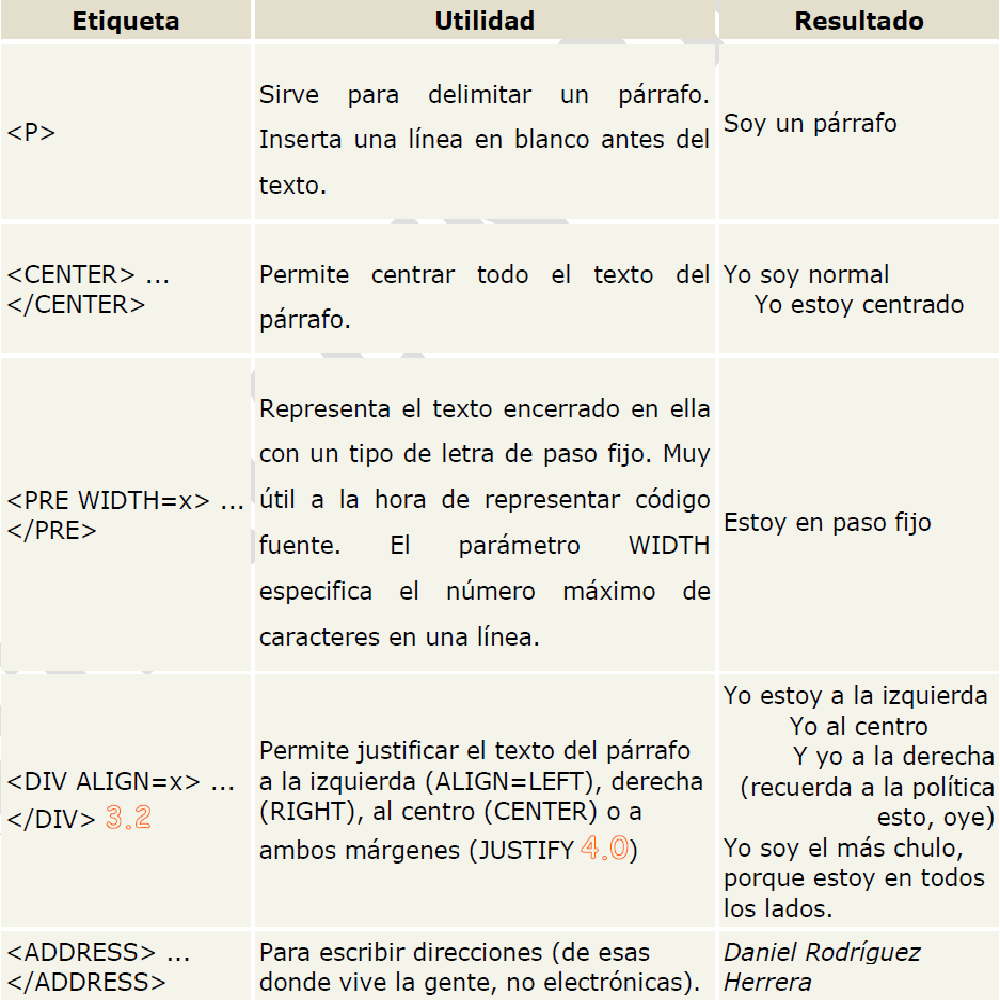
<P> Esta es mi primera pagina HTML. Por el momento solo se muestra unas cuantas cosas dentro de poco pondre aqui muchas cosas interesantes.

**Formateo básico**

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

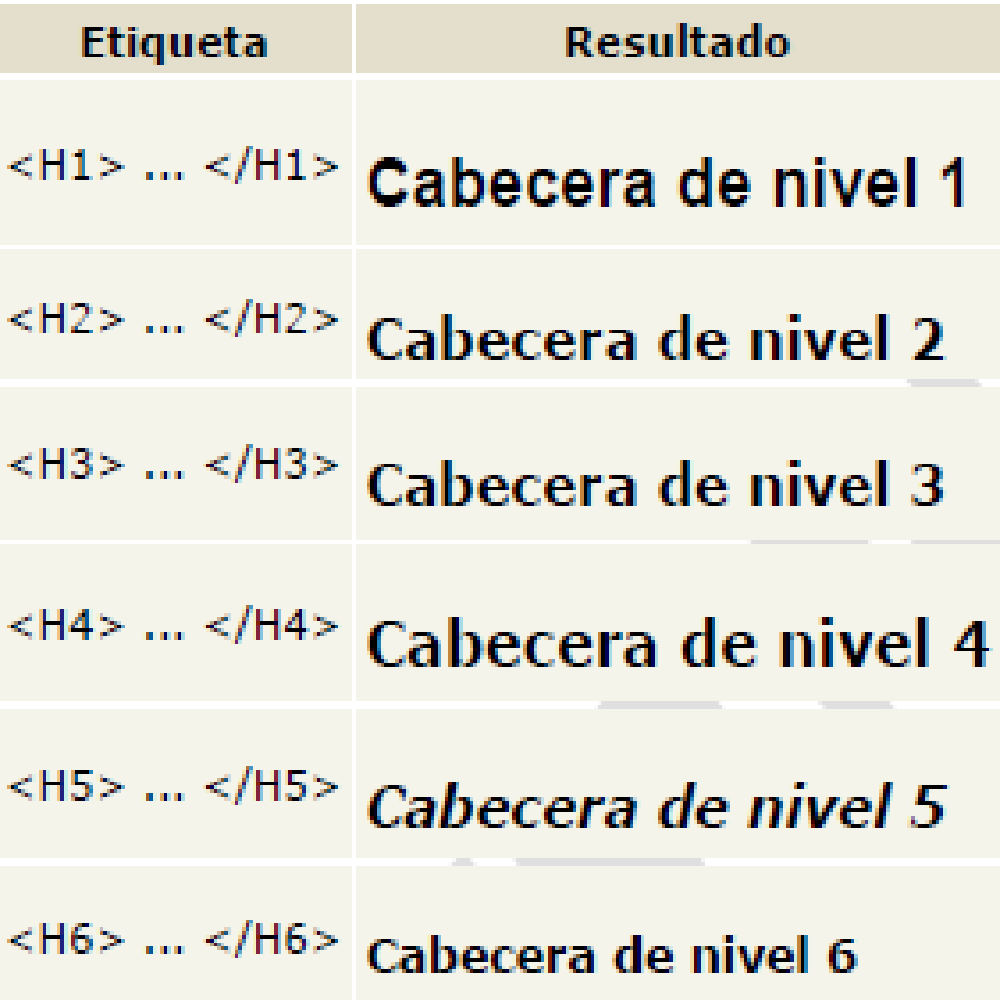
**Formato del párrafo**

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



**Las 6 cabeceras**

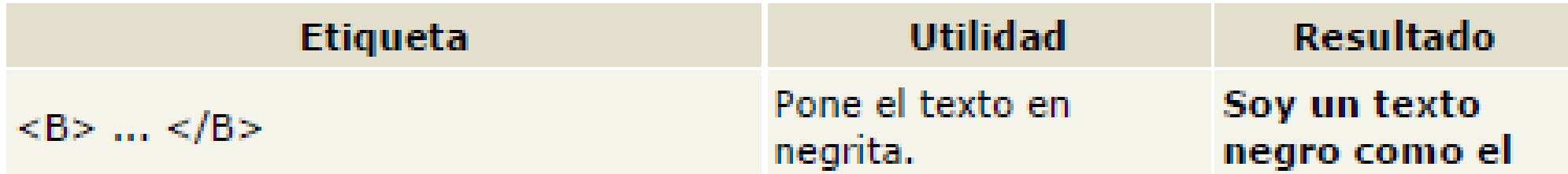
El HTML nos ofrece seis etiquetas distintas para mostrar cabeceras. Son éstas:

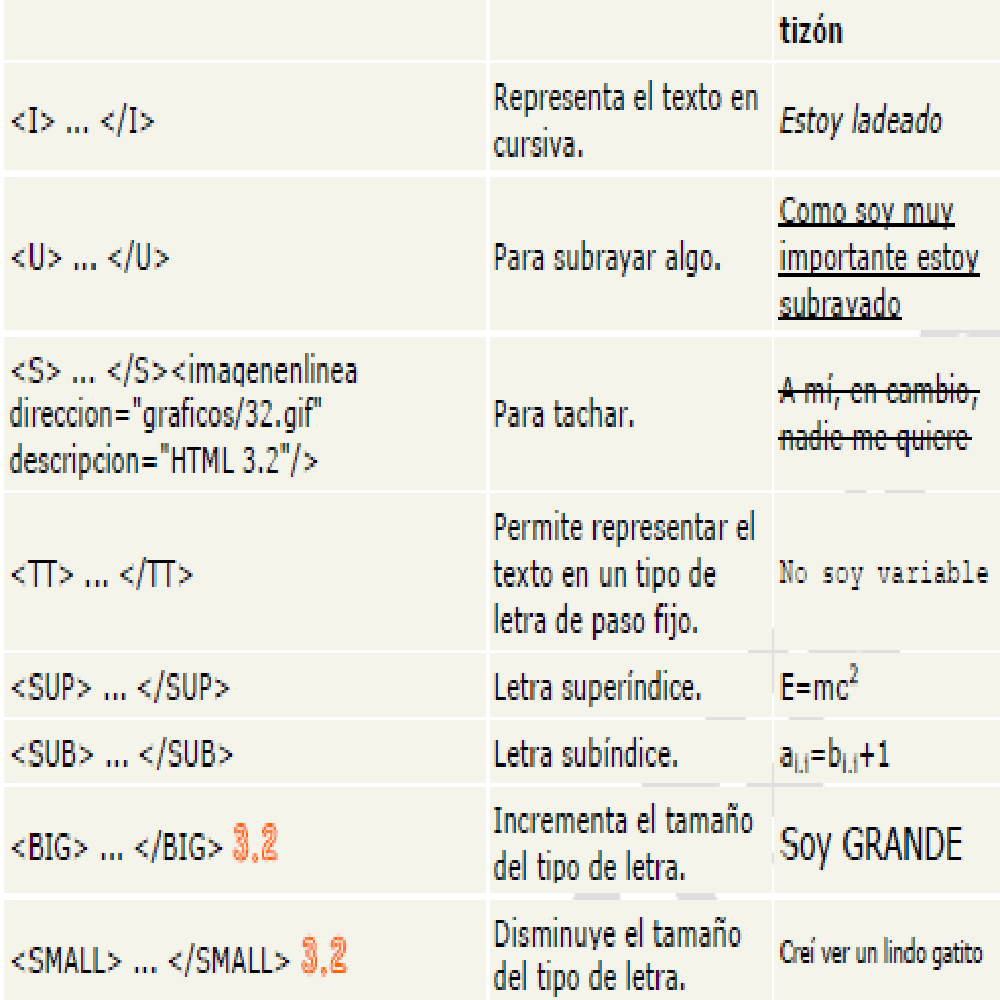


Estas etiquetas se pueden definir como de formato de párrafo pero por su importancia se muestran aparte. No resulta recomendable utilizarlas para aumentar o disminuir el tamaño del tipo de letra, ya que cada navegador los muestra de manera diferente. Se usan para dividir correctamente en secciones nuestra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

**Cambiando el tipo de letra**

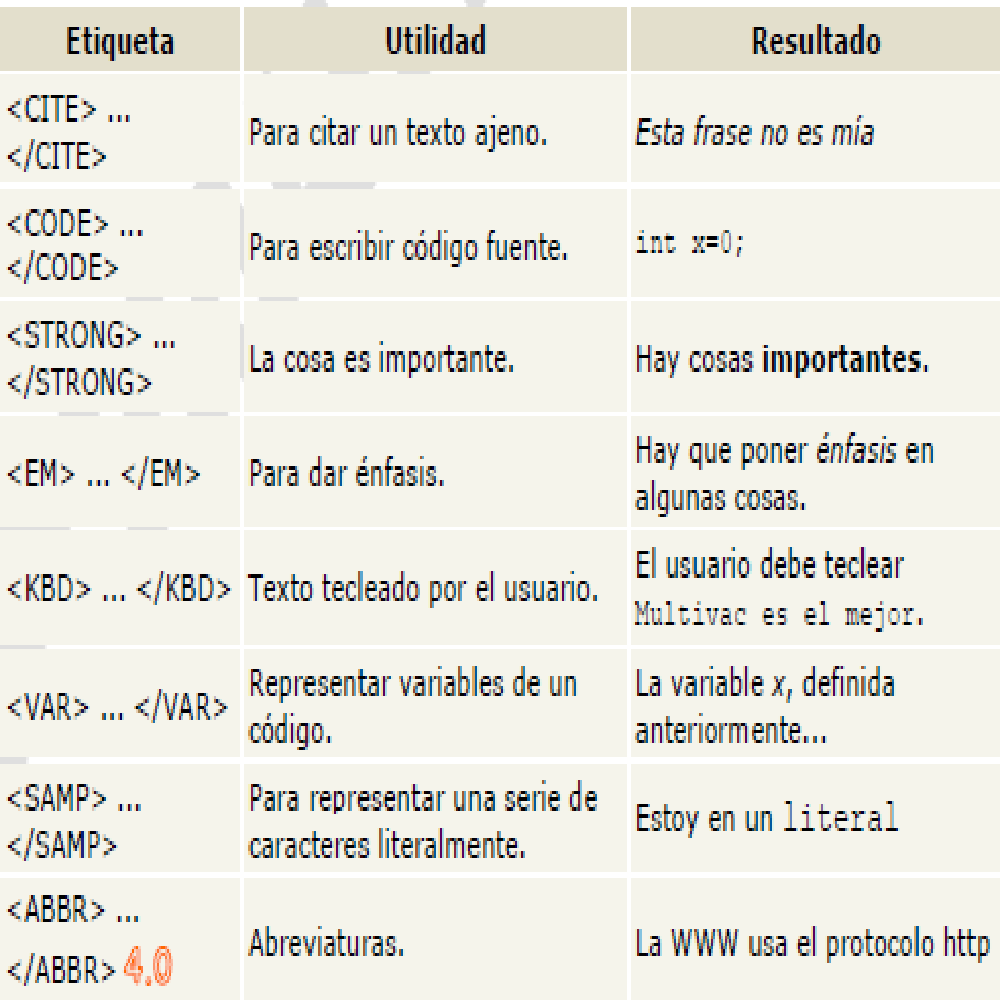
Todas estas etiquetas nos permiten cambiar de una manera u otra el aspecto del tipo de letra que estemos utilizando y se pueden utilizar con tiras de caracteres dentro de un párrafo.





**Formato de frase**

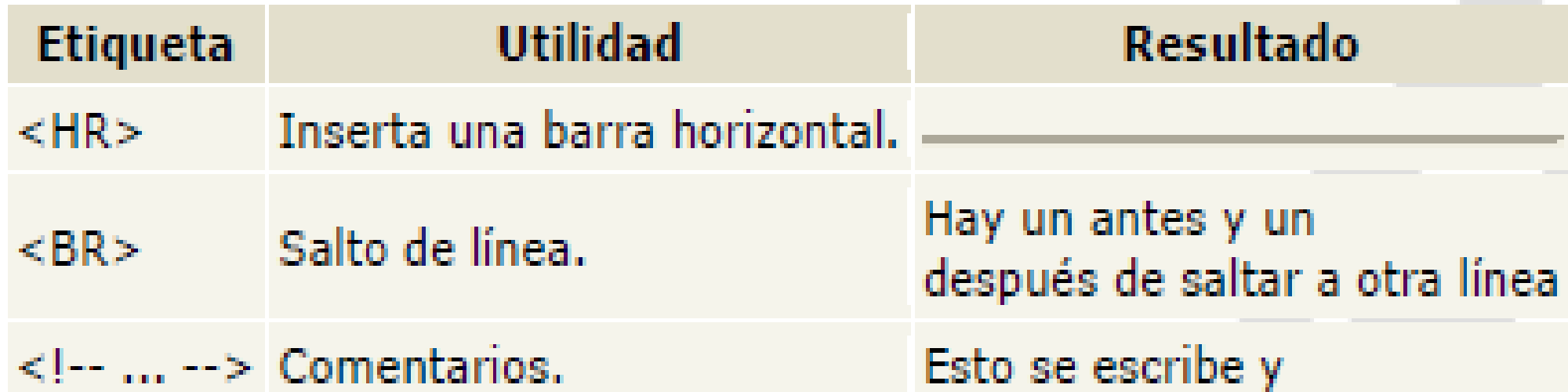
En estos elementos indicas el tipo de información que encierran las etiquetas, pero no como se representan:



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.



**Caracteres especiales**

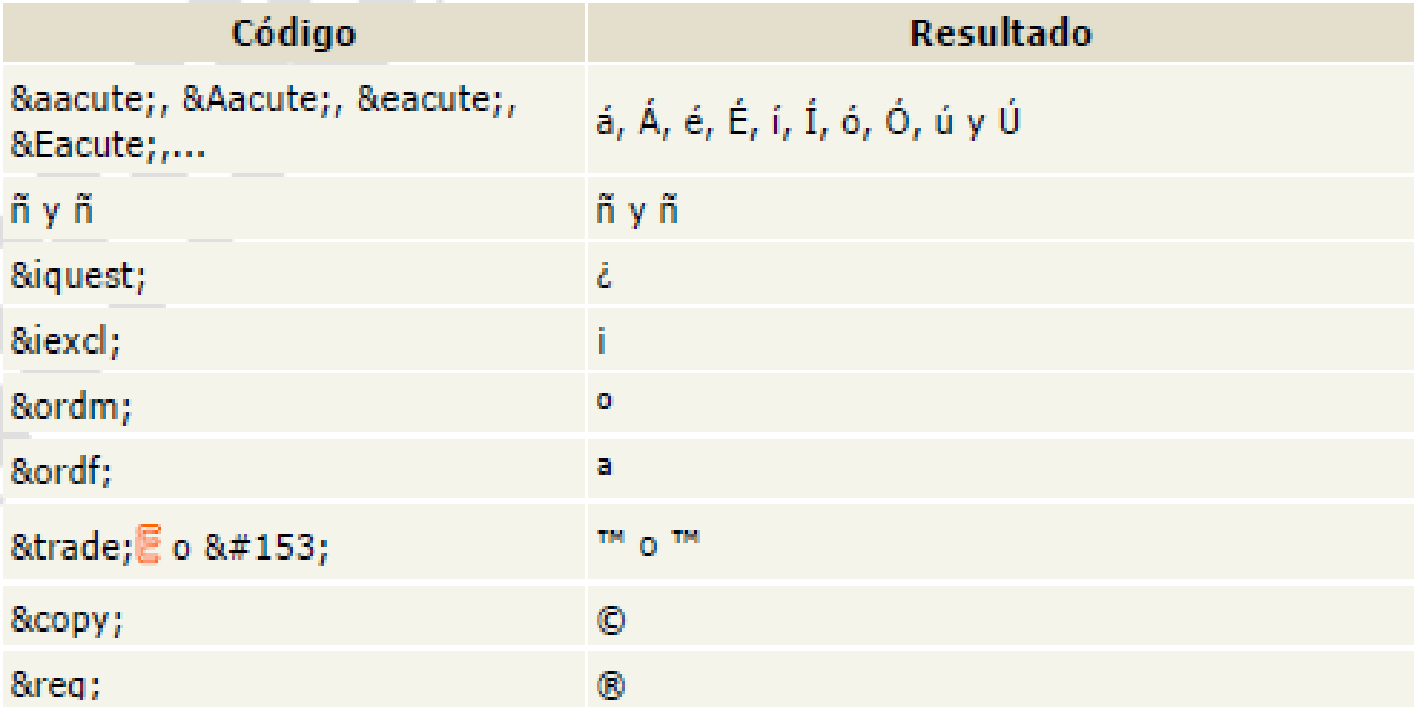
Las máquinas manejan la información en formato binario (es decir, en unos y ceros). Estos, a su vez, forman números, los cuales se traducen en letras. ¿Cómo?

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...

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.

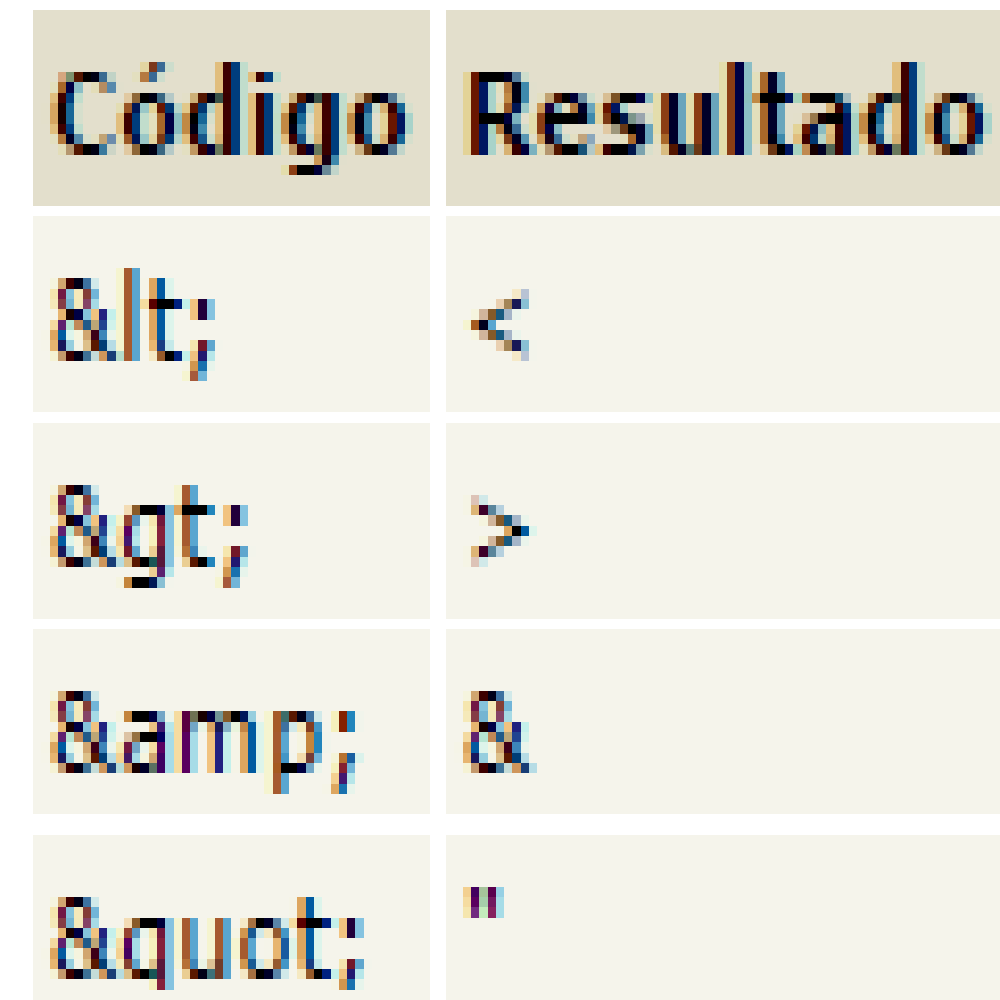
**Caracteres extendidos en HTML**

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: &#189; nos debería dar un medio (½). Tambié existe una serie de sinónimos para poder recordar con más facilidad estos caracteres. Así, por ejemplo, &#189; también se puede escribir como &frac12;. Vamos a ver algunos de estos códigos, los más útiles a la hora de escribir en español:



**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 siguientes códigos:



**Enlaces**

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

Un enlace es una zona de texto o gráficos que si son seleccionados nos trasladan a otro documento de hipertexto o a otra posición dentro del documento actual.

Siendo HTML el lenguaje de Internet, la diferencia que posee con respecto a otros tipos de hipertexto es que ese otro documento puede estar físicamente en la otra punta del planeta. Son los enlaces lo que hacen de la telaraña o World Wide Web lo que es.

**La etiqueta <A>**

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

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

Al pulsar sobre el enlace, seremos enviados al documento que apuntaba el enlace.

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

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

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

**Las URLs**

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

servicio://máquina:puerto/ruta/fichero@usuario donde el servicio podrá ser uno de los siguientes:

**http**

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

**https**

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

**ftp**

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

**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.ibernet.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 facil de recordar como es una serie de palabras separadas por puntos (www.programacion.net). El puerto generalmente no se indica, ya que el servicio lo predetermina uno.

La ruta es una serie de directorios separados por el símbolo /

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**

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">

Ejemplo:

<A HREF="#ancla"><enlaceext direccion="#2">Vamos a donde antes</enlaceext></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"><enlaceext direccion="#2">Otra vez</enlaceext></A>

**Listas**

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

<tipo\_lista>

<LI>Primer elemento

<LI>Segundo elemento

</tipo\_lista>

tipo\_lista puede ser una de las siguientes: DIR, DL, MENU, OL y UL.

**Listas desordenadas**

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

<UL>

<LI>Primer elemento

<LI>Segundo elemento

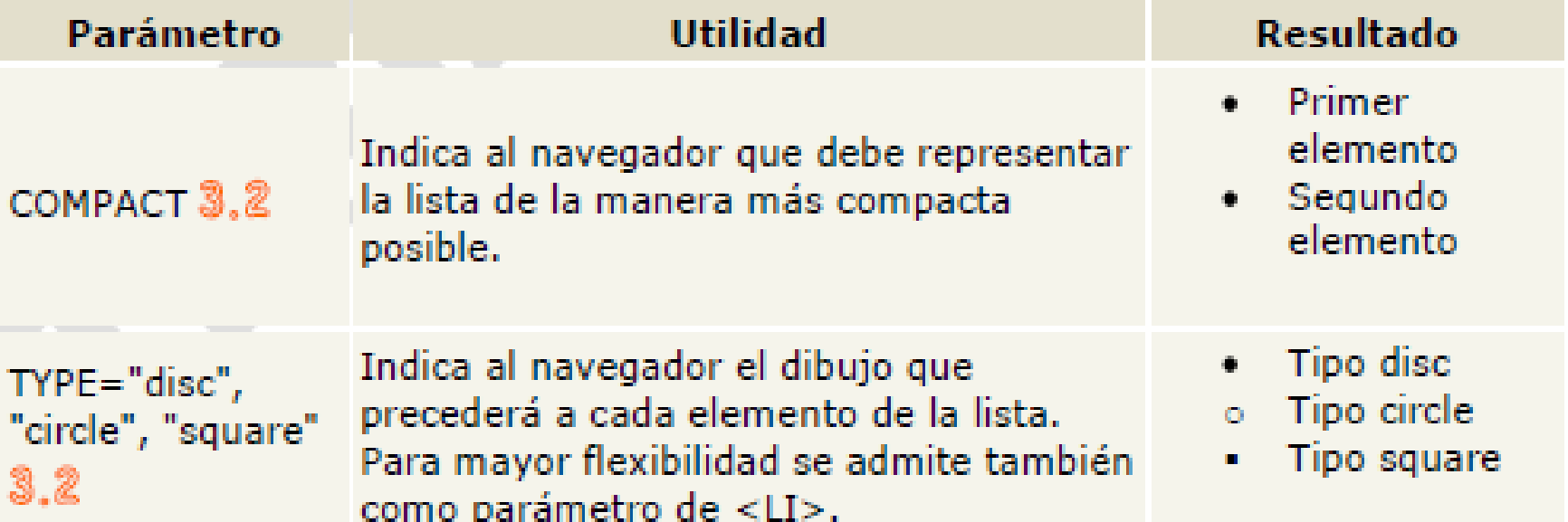
</UL>

se verá como

. Primer elemento

. Segundo elemento

La etiqueta <UL> admite estos parámetros:



**Listas ordenadas**

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

<OL>

<LI>Primer elemento

<LI>Segundo elemento

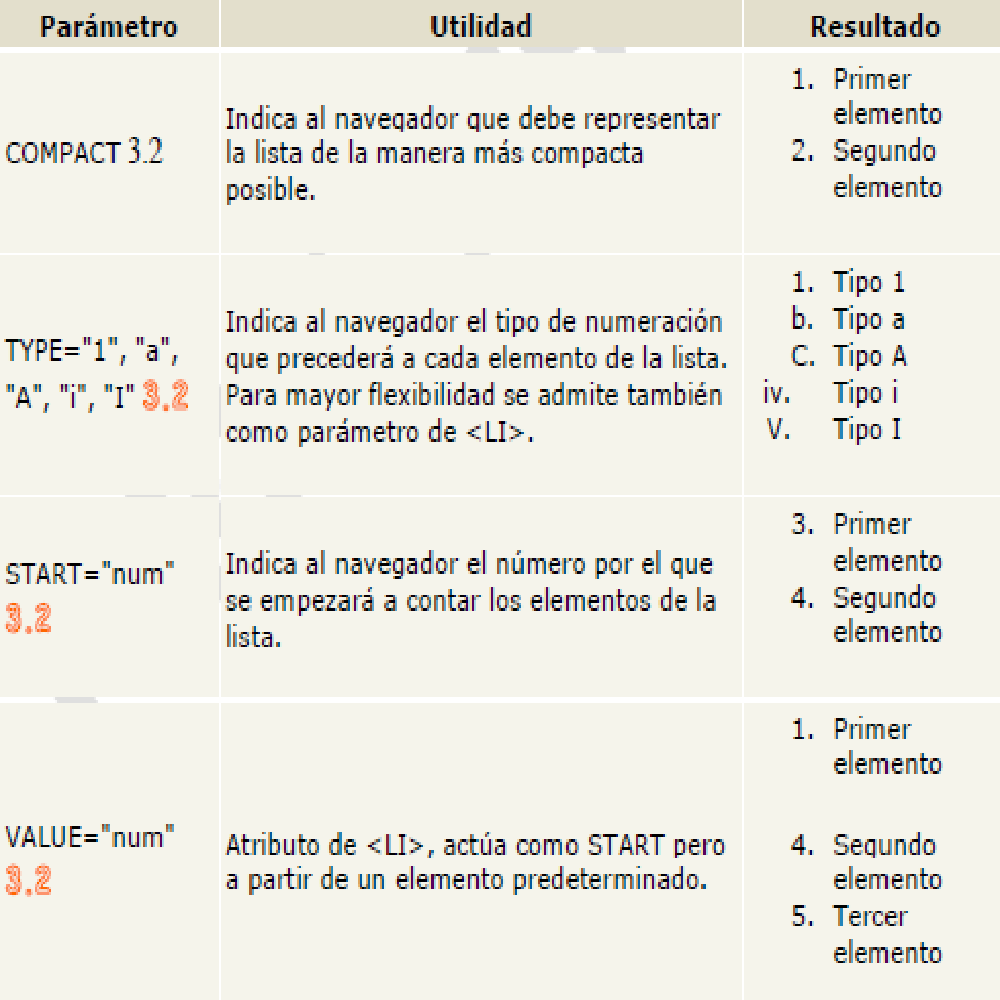
</OL>

se verá como

1. Primer elemento

2. Segundo elemento

La etiqueta <OL> admite estos parámetros:



**Listas de definiciones**

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

<DL>

<DT>Primer elemento<DD>Es un elemento muy bonito.

<DT>Segundo elemento<DD>Este, en cambio, es peor.

</DL>

se verá como

Primer elemento

Es un elemento muy bonito.

Segundo elemento

Este, en cambio, es peor.

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

**Imágenes**

Para incluir gráficos e imágenes en nuestras páginas utilizaremos la etiqueta

<IMG>

de esta manera:

<IMG SRC="fichero\_grafico" ALT="descripcion">

El parámetro SRC especifica el nombre del fichero que contiene el gráfico. Los formatos estándar en la red son el GIF y el JPG. Las 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/WhatsNew.gif" ALT="Wath´s New " WIDTH=88 HEIGHT=31>



**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 poniendolo a cero.

<A HREF="http://www.netscape.com">

<IMG SRC="graficos/dwnldns.gif" ALT="Netscape 4.0" WIDTH=88

HEIGHT=31>

</A>

Sin embargo,

<A HREF="http://www.netscape.com">

<IMG SRC="graficos/dwnldns.gif" ALT="Netscape 4.0" WIDTH=88

HEIGHT=31

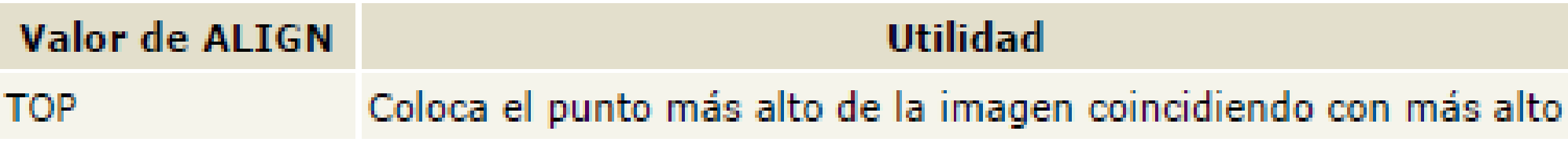
<rojo>BORDER=0</rojo>>

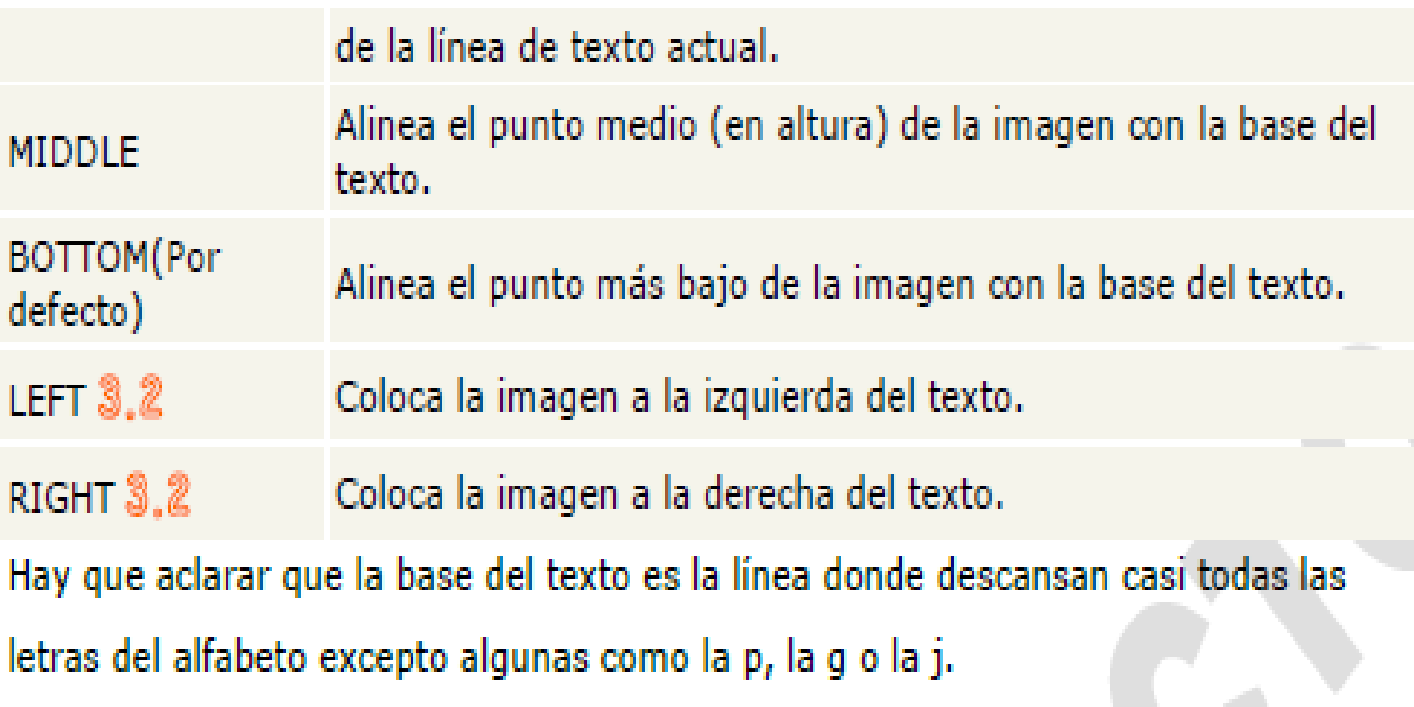
</A>

Se ve sin el borde

**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:





**Formateo fino**

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

**Cambio de color**

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



<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>

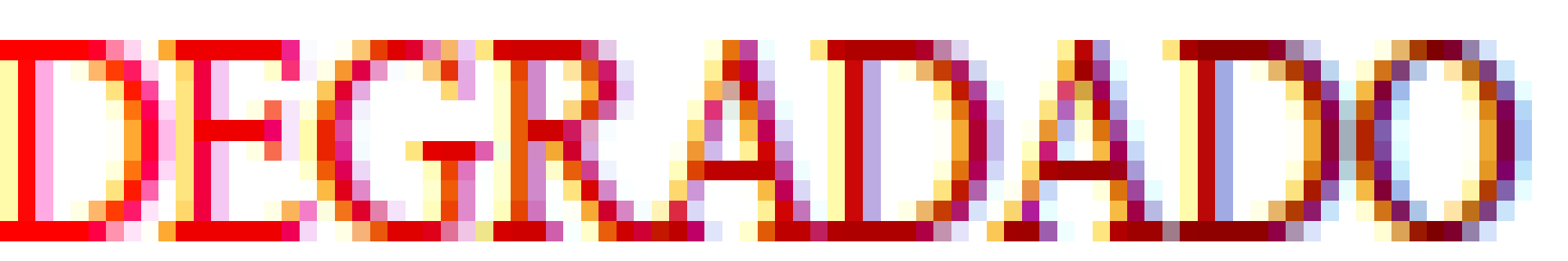
<FONT COLOR="#AF0000">D</FONT>

<FONT COLOR="#9F0000">A</FONT>

<FONT COLOR="#8F0000">D</FONT>

<FONT COLOR="#7F0000">O</FONT>

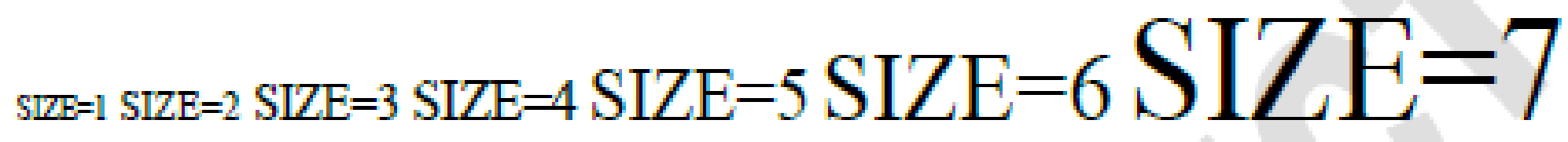
Lo que nos mostraría lo siguiente:



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:



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><font size="2">Tamaño 2</font><FONT SIZE="+3">

<font size="2"><font size="+3">Tamaño 6</font></font></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"><font face="Helvetica,Arial,Times">No sé como voy a salir exactamente</font></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. En Internet, siempre se debe ser lo más estándar posible.

**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**

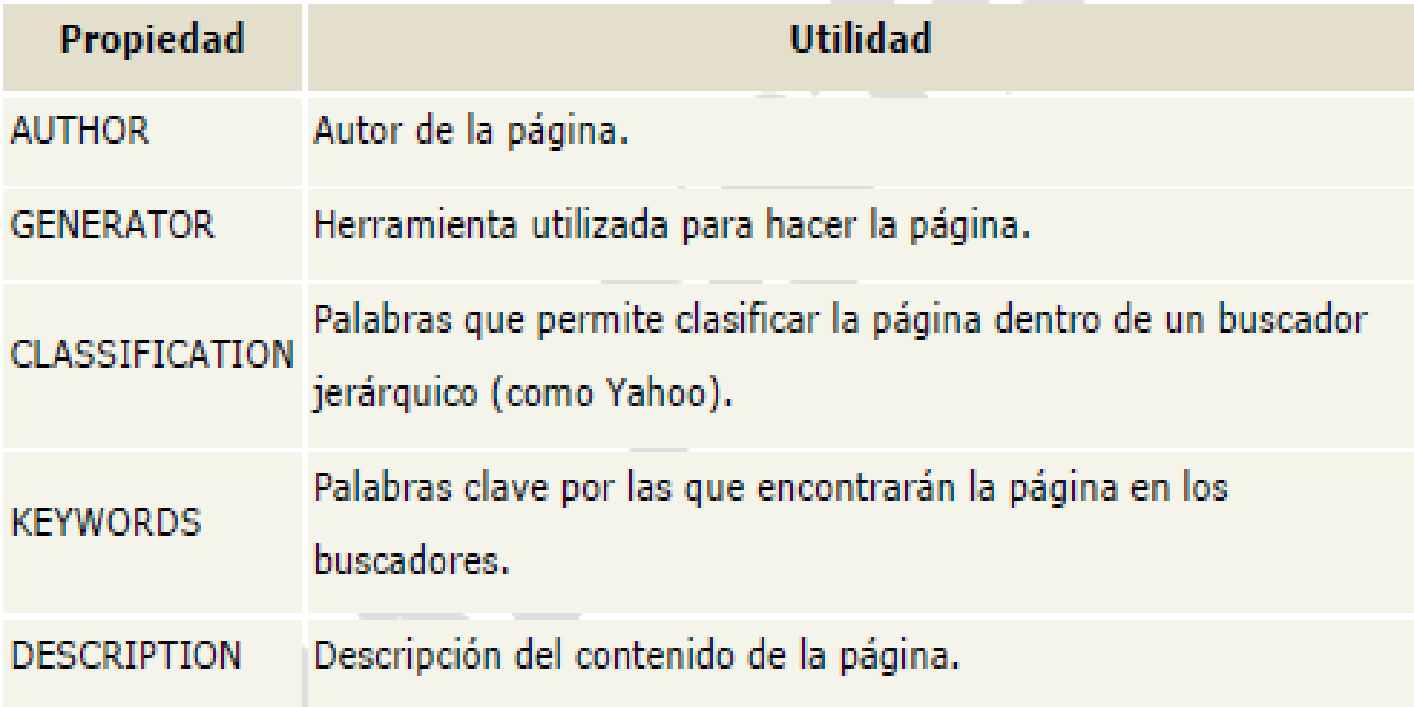
Suele ser el lugar indicado para colocar aquellos elementos de la página que no alteren el contenido de la misma, aunque si la forma de presentarlo y su comportamiento. Es por eso que es el lugar más recomendable para colocar los scripts y las hojas de estilo, como veremos en los capítulos correspondientes.

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

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

[Netscape]">

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



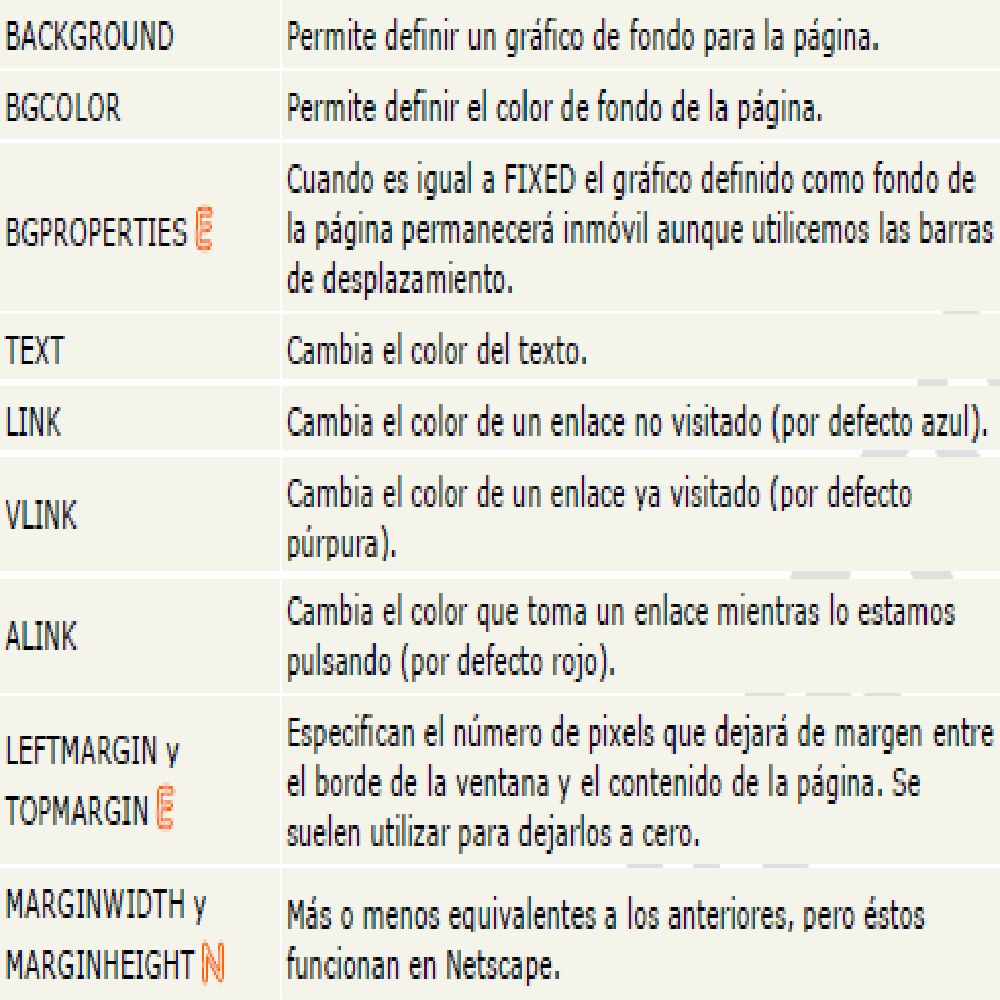
Hay también otro elemento interesante que podremos incluir en nuestras cabeceras. Cuando especificamos una URL relativa en un enlace, en principio es para acceder a una página situada en nuestro mismo servidor. Sin embargo, si especificamos:

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

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

**El cuerpo**

Obviamente no vamos a explicar lo que entra dentro del cuerpo (prácticamente de eso trata este modulo) sino los parámetros que admite la etiqueta <BODY>:



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.

**Mapas**

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

. Mapas gestionados por el cliente (el navegador).

. Mapas gestionados por el servidor.

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

**Mapas gestionados por el cliente**

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

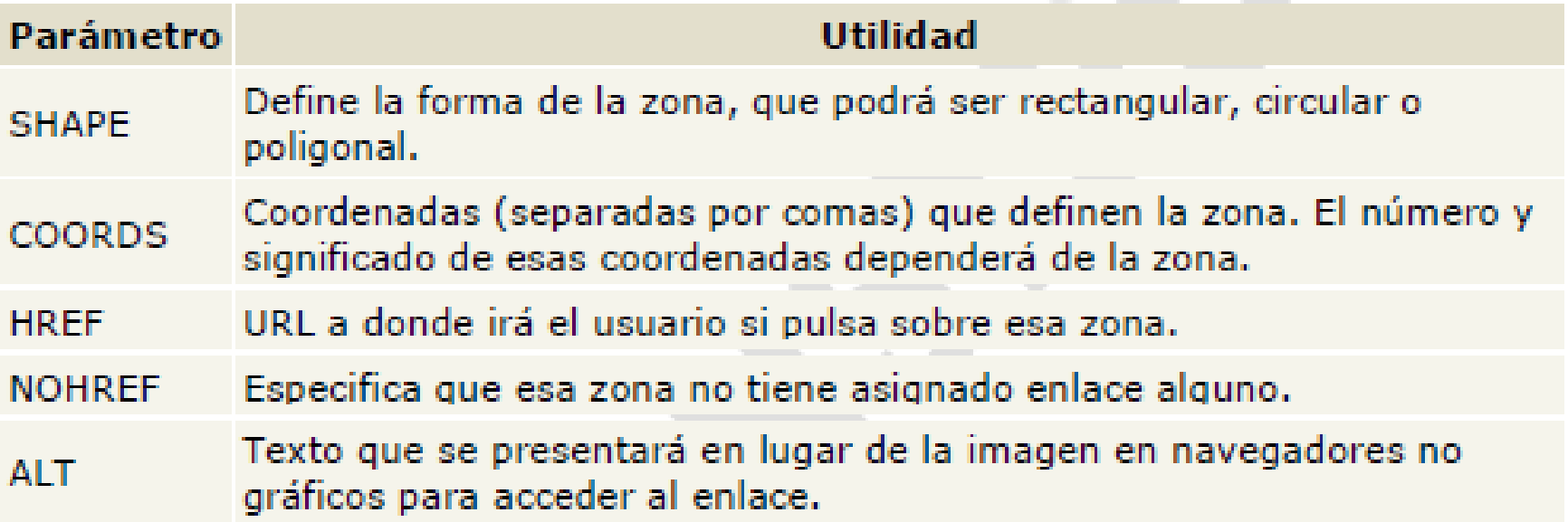
<MAP NAME="mi\_mapa">

<AREA SHAPE=... COORDS=... ALT="Enlace a..">

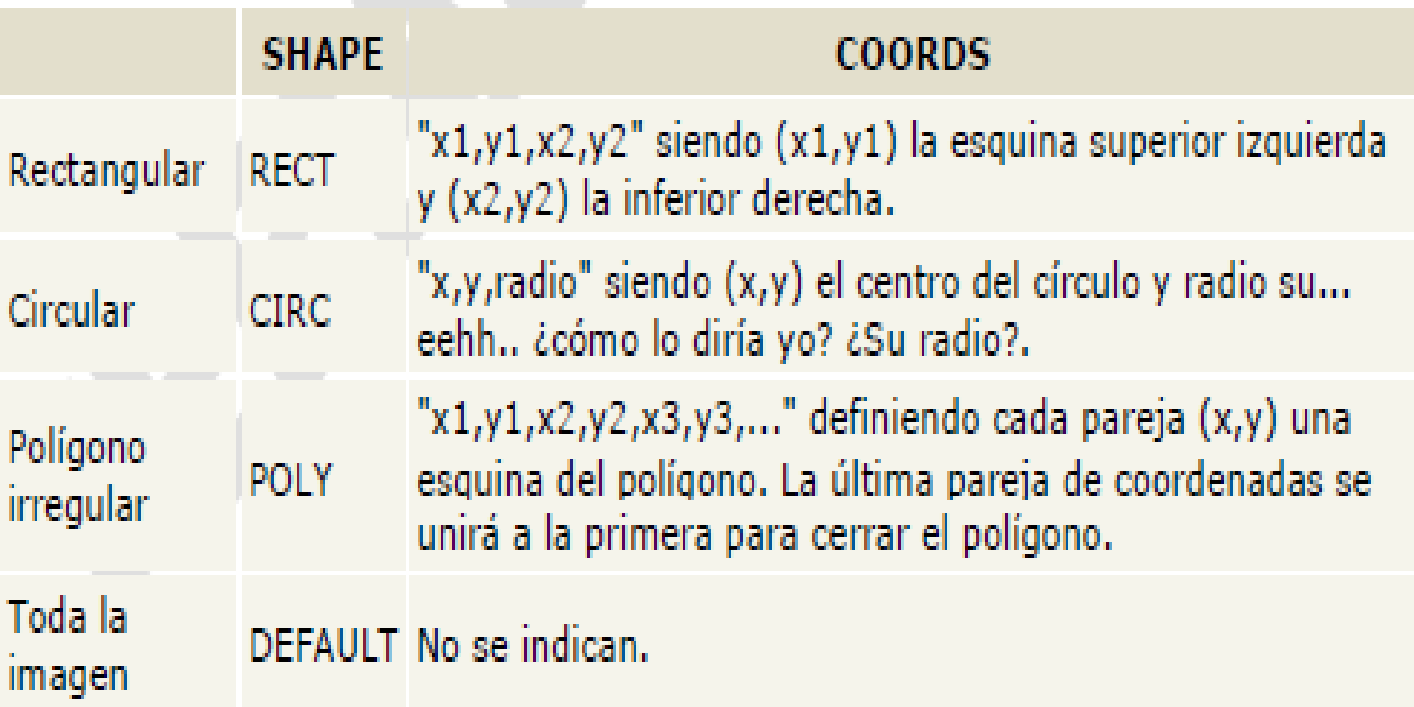
...

</MAP>

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



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



**Cómo usar un mapa**

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

<IMG SRC=... USEMAP="#mi\_mapa">

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

<MAP NAME="navegadores">

<AREA SHAPE=RECT COORDS="0,0,24,31"

HREF="http://www.netscape.com" ALT="Netscape">

<AREA SHAPE=RECT COORDS="26,0,53,31"

HREF="http://www.microsoft.com" ALT="Microsoft">

<AREA SHAPE=DEFAULT NOHREF ALT="Nada">

</MAP>

<IMG SRC="nav.gif" WIDTH=53 HEIGHT=31 BORDER=0 USEMAP="#navegadores">

Se ve tal que así:



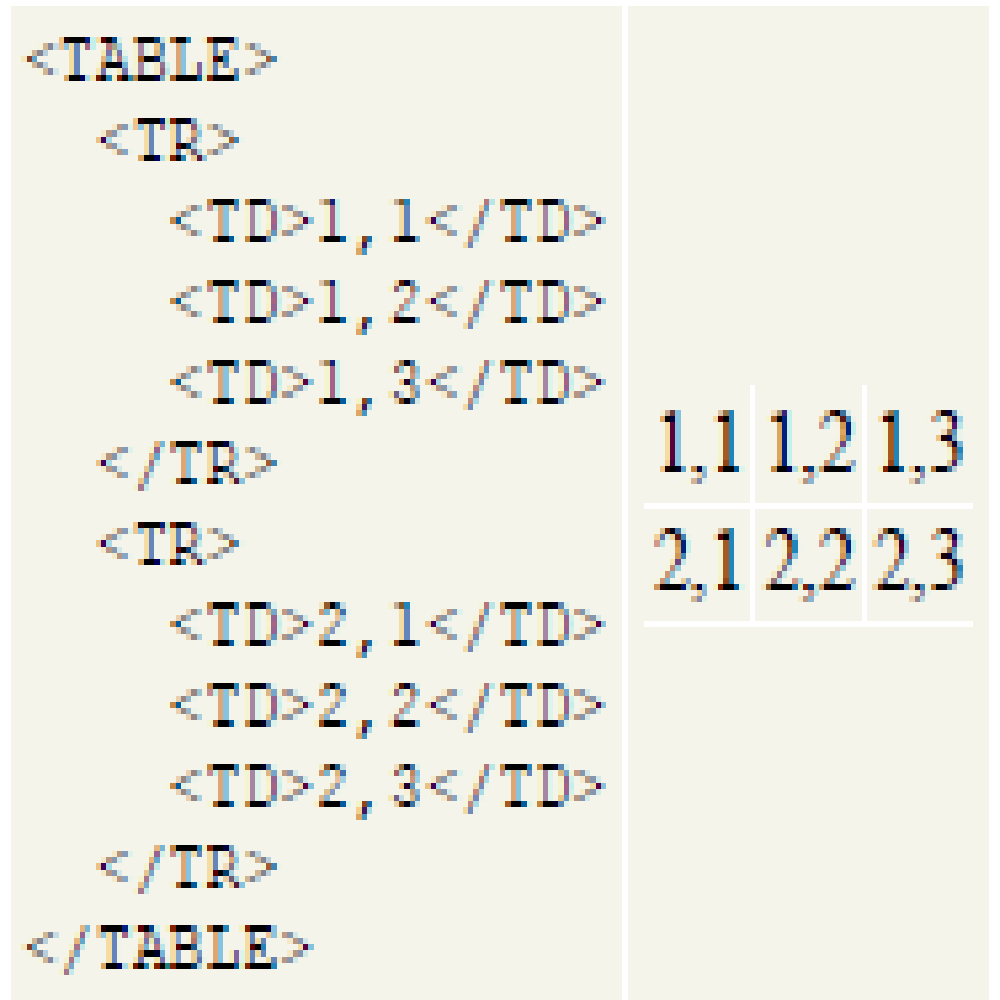
Hay que tener en cuenta que, cuando dos zonas se solapan, la que esté declarada primero es la que tiene preferencia. Por eso declaramos al final una zona que no conduce a ningún URL por si el usuario pulsa con el ratón donde no debe.

**Tablas**

Las tablas son posiblemente la manera más clara de organizar la información.

También es el modo más adecuado de maquetar texto y gráficos de una manera algo más controlada que con los parámetros ALIGN.

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



Como lo puede ver (o mejor no ver) la tabla no tiene mucho aspecto de tabla.

Quedaría mejor con unos bordes, Puede que tampoco le viniese mal mayor espacio entre celdas o mayor anchura. Estas son las cosas que podremos cambiar con los atributos de TABLE:

BORDER

Especifica el grosor del borde que se dibujará alrededor de las celdas. Por defecto es cero, lo que significa que no dibujará borde alguno.

CELLSPACING

Define el número de pixels que separarán las celdas.

CELLPADDING

Especifica el número de pixels que habrá entre el borde de una celda y su contenido.

WIDTH

Especifica la anchura de la tabla. Puede estar tanto en pixels como en porcentaje de la anchura total disponible para él (pondremos "100%" si queremos que ocupe todo el ancho de la ventana del navegador).

ALIGN

Alinea la tabla a izquierda (LEFT), derecha (RIGHT) o centro (CENTER).

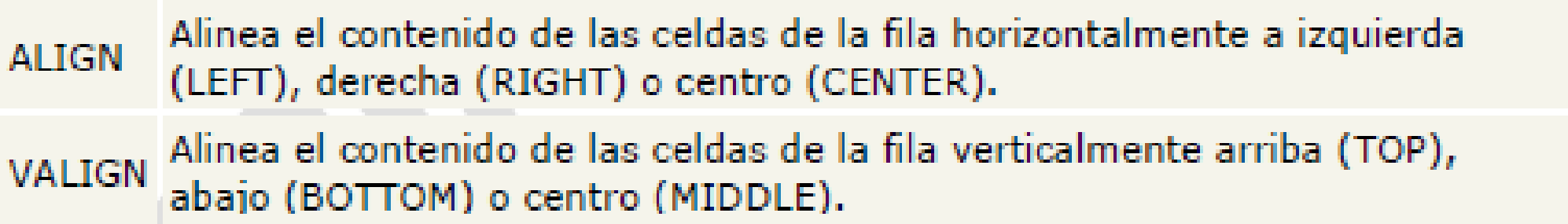
Si ahora, por ejemplo definimos ahora la tabla anterior como <TABLE BORDER=1

WIDTH="50%" ALIGN=CENTER> veremos lo siguiente:



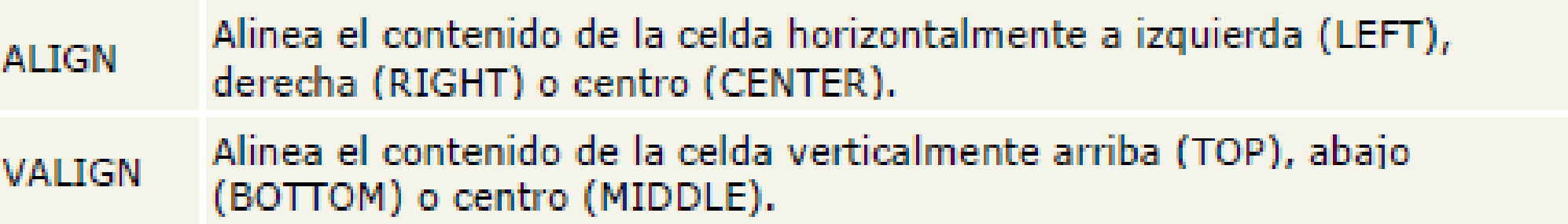
**Definir las filas**

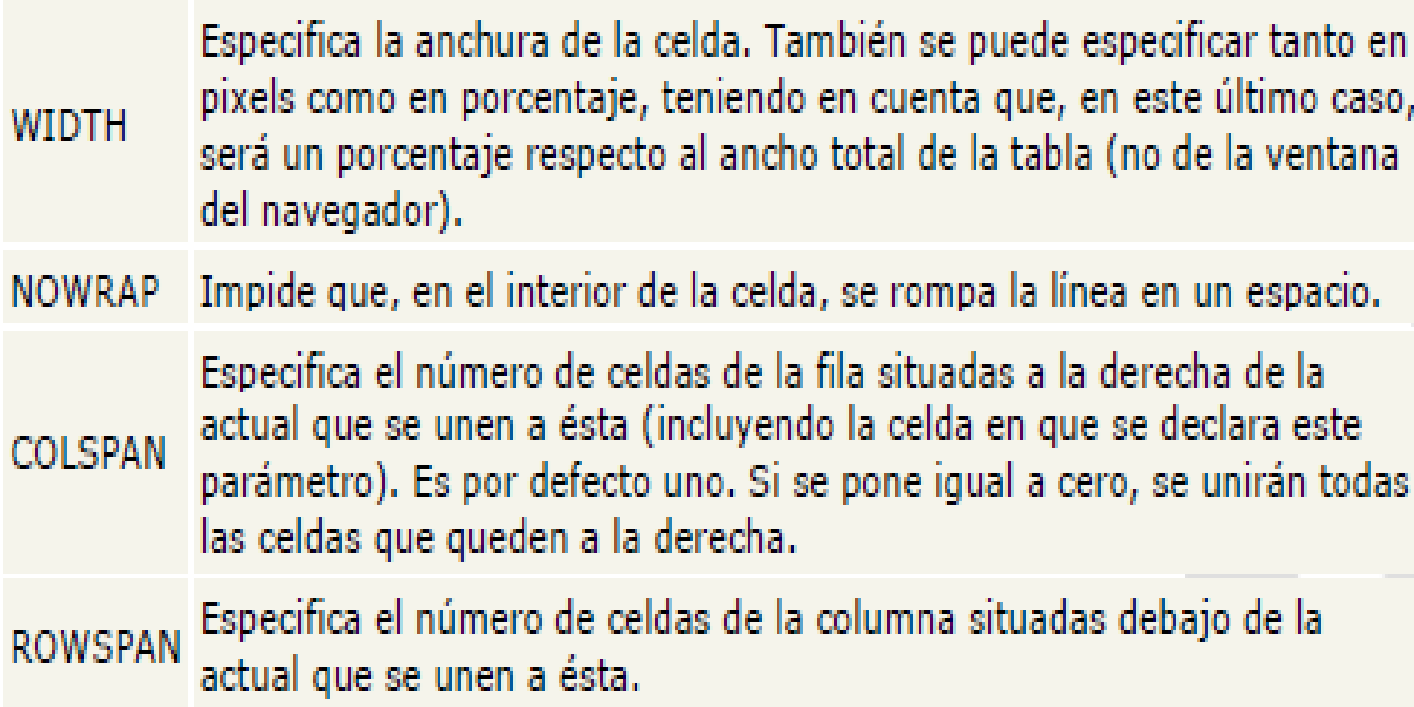
Ahora que hemos definido la tabla nos toca hacer lo mismo con las filas. Cada fila se define con una etiqueta TR, que tiene los siguientes atributos:



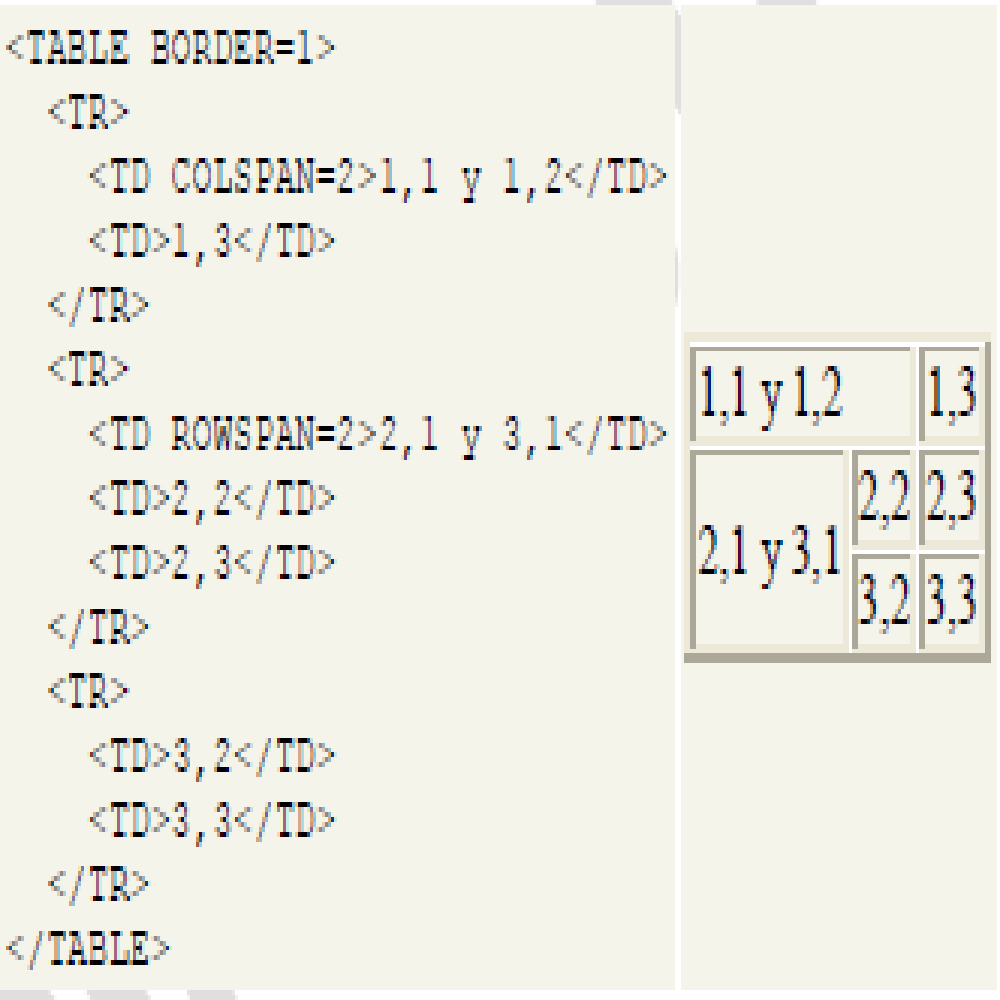
**Definir las celdas**

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





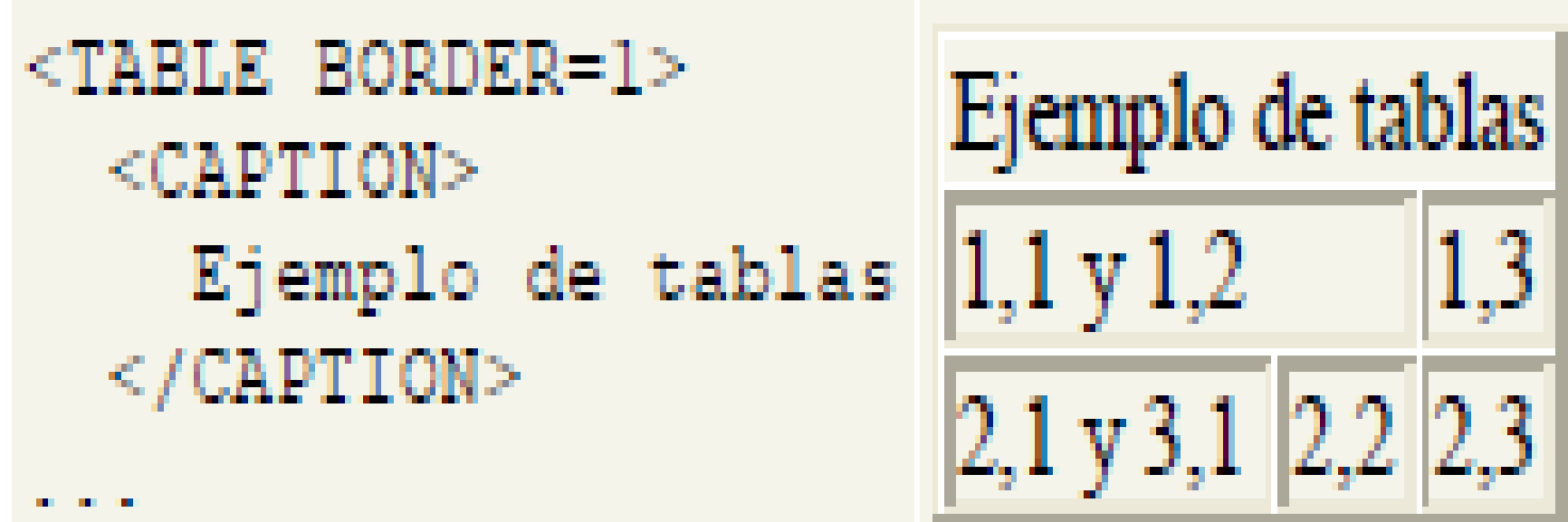
Posiblemente los dos últimos parámetros no puedan quedar claros sin ejemplos. De hecho, aún entendiendo perfectamente su función es habitual que confundamos a uno con otro. Pero bueno, vamos a ver una tabla de 3x3 con una celda que se une a una de la derecha y otra que se une a otra de debajo:



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

**Título de la tabla**

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



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