
Guía para escribir documentos HTML

Versión 1.4 (beta)

© jac. 1996 - 2000. E-mail: jac@uv.es

Si prefiere tenerla en local (recomendado), puede hacer FTP desde aquí:(Versión 1.3)

 [Versión PC](#) (533 Kb.)

1.  [Propósito del documento](#)
2.  [Notas sobre el lenguaje](#)
3.  [¿Cómo funciona esto?](#)
4.  [Introducción al HTML](#)
5.  [Estructura de un documento HTML](#)
6.  [Cabecera de un documento](#)
<HEAD> <TITLE> <META>
7.  [Cuerpo de un documento](#)
<BODY>
 1.  [Tamaños y tipos de letra](#)
<Hx>
 2.  [Texto de colores](#)

 3.  [Párrafos. Saltos de línea. Bloques tabulados. Línea horizontal. Divisiones](#)
<P>
 <BLOCKQUOTE> <HR> <DIV>
 4.  [Texto preformateado](#)
<PRE> <SPACER>
 5.  [Columnas de texto](#)
<MULTICOL>
 6.  [Negrita. Itálica. Centrado. Otros efectos](#)
 <I> <CENTER>
 7.  [Lista desordenada. Lista ordenada. Menú. Lista de definición. Directorio](#)
 <MENU> <DL> <DIR> <DT> <DD>

1.  [Ejemplos de listas](#)
8.  [Tablas](#)
<TABLE> <TR> <TD> <TH>
 1.  [Ejemplos de tablas](#)
9.  [Códigos de colores](#)
 <BODY BGCOLOR>
 1.  [Combinación de efectos de colores](#)
<BODY BGCOLOR FGColor TEXT LINK VLINK ALINK>
10.  [¿Por qué hay que usar códigos?](#)
&#xxx; &alias;
 1.  [Tabla de códigos](#)
11.  [Creando enlaces \(links\)](#)
<A>
12.  [Insertando imágenes](#)
 <MAP>
 1.  [Usar una imagen como fondo de página](#)
<BODY BACKGROUND>
 2.  [Usar una imagen como marca decorativa de una lista](#)
13.  [Introducción a los formularios](#)
14.  [¿Cómo se escriben formularios?](#)
<FORM> <INPUT> <SELECT> <TEXTAREA>
15.  [¿Cómo se escriben formularios?](#) (Página especial para impresión)
 1.  [Ejemplos de formularios](#)
 2.  [Ejemplos de formularios](#) (Página especial para impresión)
 3.  [Tabla de códigos de conversión en formularios](#)
 4.  [Macro de conversión](#)
de ficheros recibidos desde formularios para Word 6.0 de Windows 3.1.
 5.  [Macro de conversión](#)
de ficheros recibidos desde formularios para Word 97 de Windows 95.
16.  [Frames](#)
<FRAMESET> <FRAME>
8.  [Últimas innovaciones del HTML.](#)
 1.  [Páginas con sonido](#)
<EMBED>
 1.  [Ejemplos de páginas con sonido.](#)
 2.  [Ejemplos de páginas con sonido](#) (Página especial para impresión).

2.  [Páginas con vídeo](#)
<EMBED>
3.  [Los lenguajes JavaScript y Java](#)
<SCRIPT> <APPLET>
 1.  [Ejemplos de JavaScript.](#)
4.  [Gráficos animados](#)
5. **NUEVO** [Adiciones a la guía](#)

Propósito de este documento

Algunas reflexiones

Estas páginas han sido creadas como parte del material para impartir un breve cursillo sobre el mundo WWW, en particular, y las herramientas para moverse por la INTERNET, en general, a un grupo de estudiantes con pocos o nulos conocimientos informáticos.

Al escribir esto comencé como suelen comenzar casi todos los manuales que pueden encontrarse a lo largo y ancho del ciberespacio, pero en mitad de una farragosa explicación sobre el ISO 8859-1, pensé: "... bueno, esto en realidad no necesitan saberlo... pero, todo sea por el rigor técnico...". Y entonces hice una prueba... Le di a leer lo que acababa de escribir a un "conejillo de indias" que pasaba por allí y que de esto sólo conocía "eso que dicen por la tele de las autopistas de la información...".

El resultado, como era de esperar, fué algo peor que desastroso. De las cinco primeras líneas, sólo comprendió una docena de palabras, y de las restantes, sus preguntas daban pie a pensar que era algo más que un breve cursillo lo que haría falta para que saliese medianamente enterado.

Y sin embargo, yo tengo claro que el HTML no es tan difícil. Pero, ¿cuál es el objetivo real? Simple: saber lo necesario para poder hacer una página personal en un servidor ya instalado.

Evidentemente, habrá que partir de unos leves conocimientos de "entorno": como haber manejado un procesador de textos en Windows, saber guardar un fichero donde uno quiere y poco más que sea realmente imprescindible. Y esto es poco más o menos lo que saben la mayoría de usuarios.

Había que olvidar el "rigor técnico", y escribir las cosas de forma que no se requiriera ser analista de sistemas para comprenderlas, y sobre todo, pensando qué es lo realmente necesario para poder obtener resultados aceptables, que animen al alumno en un tiempo razonable pero sin dejar de lado los conocimientos de altura, que serán demandados por el propio alumno a medida que avanza.

Esto no es un manual de referencia. Ni siquiera es un manual. No hay referencias técnicas ni siglas, ni lenguaje técnico... ni por supuesto, está "todo" sobre el HTML. Es una serie de apuntes donde no falta nada de lo que hace atractivo el HTML, con ejemplos de los de cortar y pegar, como un recetario. Algunas cosas son, por naturaleza, algo complejas, y puede que tengas que preguntar, es inevitable. Aunque espero que sean las menos, y que después de leer esto, ya puedas trabajar sin ayuda.

Cómo utilizar éste documento

No te aconsejo que te "empolles" toda la guía. Solamente hojéala. Ahora se trata de que sepas lo que se **puede hacer**, no **cómo se puede hacer**. Cuando te hayas dado un ligero baño de sabiduría HTML, piensa lo que quieres hacer. El diseño inicial es importante, comienza a editar tu primera página. No temas cometer burradas, no se puede romper nada. Lo peor que puede pasar es que no funcione.

Cuando ya tengas algo escrito, pruébalo. Si algo no funciona, busca la sintaxis correcta, ves complicando

Proposito

el documento poco a poco, "tirando de guía" cuando te atasques. Copia algunos ejemplos y modifícalos. Compara su funcionamiento antes y después.

[Indice](#)

Notas sobre el lenguaje

El HTML (Hyper Text Markup Language) es un lenguaje que sirve para escribir hipertexto, es decir, documentos de texto presentado de forma estructurada, con enlaces (links) que conducen a otros documentos o a otras fuentes de información (por ejemplo bases de datos) que pueden estar en tu propia máquina o en máquinas remotas de la red. Todo ello se puede presentar acompañado de cuantos gráficos estáticos o animados y sonidos seamos capaces de imaginar.

Por supuesto, la estética de los documentos escritos en HTML no se limita a texto digamos normal; consigue todos los efectos que habitualmente se pueden producir con un moderno procesador de textos: negrita, cursiva, distintos tamaños y fuentes, tablas, párrafos tabulados, sangrías, incluso texto y fondo de página de colores, y muchos más.

Posiblemente te dirás que todo esto está muy bien, pero que tú no vas a tener la necesidad o posibilidad de desarrollar uno de esos famosos WWW de los que todo el mundo habla, a veces sin saber demasiado bien de qué habla.

Mucha gente asocia este lenguaje (que en realidad casi ni siquiera es un lenguaje, sino una forma de definir efectos en el texto de manera similar a como se hacía en los antiguos procesadores de texto para DOS; ¡quién no recuerda Word Star y sus instrucciones con punto!) con complicados y poderosos servidores de información que misteriosamente alguien programa, y que seguramente es un genio en estas cuestiones, y piensa que en el trabajo cotidiano o incluso en casa, esto no sirve para nada...

Pues no es así. Esto es muy sencillo y más práctico de lo que parece. Evidentemente, si lo que quieres es montar un servidor de información con miles de páginas, links a varias bases de datos tanto locales como remotas, posibilidad de búsquedas indexadas, gráficos animados, sonido, páginas interactivas, etc., etc., necesitarás saber algo más que simple HTML. Pero has pensado en:

1. ¿Poner en limpio de una vez por todas esas notas que tienes desperdigadas, llenas de llamadas de unas a otras y que no hay manera de encontrar?
2. ¿Presentar tu colección de sellos con una imagen escaneada de cada uno y un comentario sobre su tema, y organizado por años, valores, o lo que se te ocurra?
3. ¿Idem de la colección de discos?
4. ¿Montar el álbum de fotos de familia con un texto explicativo de cada una?
5. ¿Organizar tu biblioteca, incluyendo el índice escaneado de cada libro y haciendo links entre temas?
6. ¿Las fotos del último viaje de vacaciones, incluyendo llamadas a los planos de las ciudades visitadas, y a todas las historias que te contaron en cada sitio?
7. ¿Si eres estudiante, presentar al "profe" tu próximo trabajo lleno de diagramas, gráficos, tablas, colores... en un disquete (junto con el visualizador por si acaso no lo tiene) ?
8. ¿Organizar para siempre y poner al alcance de todos ese catálogo o listín siempre perdido en la oficina?

9. ¿Preparar los temas de legislación, llenos de referencias cruzadas, de esas oposiciones que te llevan de cabeza?

En fin, la lista puede ser infinita. Pero esto es bastante sencillo de conseguir. ¿Qué necesitas para empezar? pues solamente un editor de texto cualquiera (desde Word al humilde "edit" de MS-DOS), un visualizador WWW como el que estés usando ahora... y haberte leído esto.

[\[Indice\]](#)

¿Cómo funciona esto?

Básicamente, la cosa es simple: la pieza clave es el "browser", "navegador", "visualizador" o "cliente" o como quieras llamarle, HTML, es decir, el programa que ahora mismo estás utilizando para ver esto. Todas las codificaciones de efectos en el texto que forman el lenguaje HTML no son más que instrucciones para el visualizador. Partiendo de esto, se entiende el porqué no se ve lo mismo con todos los visualizadores. Depende de cómo estén diseñados y para qué versión de lenguaje estén diseñados.

Actualmente existen multitud de ellos, los más conocidos son el Explorer de Microsoft, el Mosaic de NCSA y el Netscape de Netscape. Este último es probablemente el mejor de todos... por ahora. Todo lo que se dice en esta guía funciona correctamente con Netscape y algunas cosas no se verán con otros visualizadores o se verán mal, sobre todo si son versiones antiguas. Casi todos han ido evolucionando a medida que Netscape lo hacía.

Netscape es el que soporta más y mejores efectos, incluido programas embebidos en el propio texto (versión 2.0 en adelante), escritos en lenguaje Java (algo muy parecido al lenguaje C), que son interpretados por el visualizador, y que permiten realizar páginas "inteligentes".

Si se está conectado a Internet, con un visualizador Netscape, además de ver documentos HTML se puede recibir y enviar correo electrónico, recibir y enviar NEWS (noticias), visitar los servidores GOPHER (servidores de ficheros), y acceder a servidores FTP (más servidores de ficheros) tanto en entrada como en salida, todo ello con el mismo programa. También, como no, se pueden imprimir los documentos visualizados. Casi todos suelen ser "WYSWYG", es decir, que lo impreso es exactamente como lo ves en pantalla...Siempre que tu impresora esté de acuerdo, claro.

La idea es sencilla: si tienes instalado un visualizador HTML en tu máquina sólo tienes que crear un documento con tu editor preferido, hacerlo todo lo complicado y vistoso que sepas, y ya tienes un hipertexto en marcha. Los documentos creados pueden residir en el disco duro de tu ordenador personal, en un disquete, o estar a varios miles de kilómetros en otro ordenador (llamado servidor http), si es así, entonces te estarás moviendo por la WWW (World Wide Web) que es como se ha dado en llamar a este tipo de servicios de la Internet.

Así pues, en esencia, una página escrita en HTML no es más que texto normal, escrito con cualquier editor, y al que, cuando se le quiere dar algún aspecto especial, como por ejemplo el tamaño de la letra, habrá que acompañar de ciertos códigos para indicar el efecto deseado. A estos códigos se les llama **elementos** del lenguaje.

Pero no hay que olvidar que al fin y al cabo no hablamos más que de texto. Simple y llano texto. Por lo tanto, para montajes sencillos, en los que no se requiera ningún efecto especial, nos podemos olvidar hasta del mismísimo HTML. El visualizador nos presentará perfectamente cualquier página ".txt" generada por cualquier editor, y los links entre documentos sólo requieren un simple y sencillo comando. Y aún así podremos conseguir el tipo y tamaño de letra y colores de texto y fondo que queramos, simplemente configurando el visualizador.

[\[Indice\]](#)

Introducción al HTML

Elementos en un documento HTML

A las instrucciones que forman el lenguaje HTML se les llama **elementos**. La notación de los elementos consiste en los símbolos `<` y `>` que encierran dentro una instrucción.

Los elementos pueden ser llenos o vacíos.

Elementos llenos

Se forman escribiendo la instrucción correspondiente seguida del texto al que se quiere aplicar la instrucción y se termina repitiendo la instrucción pero con una barra inclinada inmediatamente antes de la misma.

Por ejemplo, el elemento **H1** que sirve para dar el máximo tamaño al texto, se escribirá:

```
<H1> Texto de prueba </H1>.
```

y este sería el resultado:

Texto de prueba

Si olvidas poner `</H1>` todo el resto de la página tendrá el mismo tamaño grande.

Elementos vacíos

Los elementos vacíos se escriben como los llenos, pero no es necesario poner la instrucción repetida al final con una barra. Esto se debe a que estos elementos no producen un efecto sobre el texto. Generalmente se utilizan para separar bloques de texto, y por tanto no es necesario indicar su fin. Empiezan y terminan en el mismo punto.

Por ejemplo, el elemento **HR** que sirve para dibujar una línea horizontal en la pantalla, se escribirá:

```
<HR>
```

Y este sería el resultado:

Elementos con argumento

Algunos elementos se escriben con argumento. Es como pasarle parámetros a la instrucción, y se llaman **atributos** del elemento.

Por ejemplo, el elemento `<A>` que sirve para hacer un link (enlace) con otro documento, se escribirá:

```
<A HREF="http://www.miservidor.es/mifichero.htm"> Link de prueba </A>.
```

Este es un elemento lleno donde al atributo `HREF` se le asigna el valor que aparece entre comillas: "http://www.miservidor.es/mifichero.htm". El texto al que afecta este elemento es **Link de prueba** y realiza un link con el fichero **mifichero.htm** que está en el servidor **www.miservidor.es**.

Los elementos pueden escribirse tanto en mayúsculas como en minúsculas. Puede ser preferible la primera opción ya que aporta claridad al documento fuente, y eso se agradece a la hora de hacer correcciones, pero **CUIDADO**, el valor de algunos atributos hay que escribirlos **EXACTAMENTE** como deban ser. En el ejemplo anterior no funcionaría el link si escribiéramos *www.miservidor.es* en mayúsculas, eso es un nombre de máquina y sería interpretado como **OTRA** máquina.

Los elementos pueden anidarse unos con otros, teniendo cuidado de poner los cierres en el lugar adecuado. Por ejemplo, el elemento `<H1>` combinado con `<I>` que sirve para generar texto en itálica, se escribirá:

```
<H1><I>Texto de prueba </I></H1>.
```

y este sería el resultado:

Texto de prueba

[\[Indice\]](#)

Estructura de un documento HTML

Los documentos escritos en HTML estan estructurados en dos partes diferenciadas: la **HEAD** (cabecera) y el **BODY** (cuerpo).

Ejemplo de documento

Si escribes:

```
<HTML>
<HEAD>
<TITLE> Documento de prueba </TITLE>
</HEAD>

<!-- Esto es un comentario-->

<BODY>

<H1> Esto es una "demo" de documento HTML </H1>

Esto es el más sencillo de los documentos HTML.

</BODY>
</HTML>
```

El elemento `<HTML>` `</HTML>` no es obligatorio. Solo sirve como identificación del tipo de contenido del fichero para ciertos programas que hacen cambios masivos en muchas páginas a la vez.

Para escribir comentarios en la página (que sólo se ven en el texto fuente, pero no en el visualizador) se utilizará el elemento `<!-- -->`

Puedes ver su aspecto real pulsando [aquí](#).

Esquema de tipos de un documento HTML

Cuando un visualizador recibe un fichero, antes de mostrarlo necesita saber el tipo de contenido que éste tiene, a fin de procesarlo adecuadamente. No es lo mismo recibir un documento de texto que una fotografía en formato GIF, o un vídeo en formato MPG, o si lo que estamos intentando es transferir un fichero.

En cada caso, el visualizador pondrá en juego distintas habilidades, de las muchas que posee. Si el fichero procede de un servidor **http** remoto, (es decir un servidor WWW; *http* es el nombre del servicio invocado), éste se encargará de decirle al visualizador cuanto necesite saber, pero si lo que hay que procesar es un fichero local, la única forma de saberlo es por la **extensión** del fichero.

La extensión de un fichero son las tres o cuatro letras (depende del sistema operativo con el que se trabaje) que hay después del nombre del fichero y que están separadas del mismo por un punto.

Por ejemplo, esta página se llama estruc.**htm**. Las tres letras que hay después del punto (htm) son la **extensión** del fichero. Las otras, las que están delante del punto, son el **nombre** del fichero.

Hay que tener presente que las extensiones de 4 o más letras son propias de sistemas operativos como Unix, Macintosh o Windows95 y las de 3 o menos del DOS.

Estas son algunas de las extensiones estándar más comunes y sus contenidos:

.html o .htm

Documento HTML. Contiene texto e instrucciones HTML que serán interpretadas.

.text o .txt

Contiene texto plano. El visualizador presentará todo el fichero como si fuera un único bloque de texto y no interpretará ninguna instrucción HTML que pudiera llevar. Esto lo hará con cualquier fichero que lleve una extensión desconocida o simplemente no lleve ninguna.

.gif

Contiene un fichero de imagen en formato GIF.

.xbm

Contiene un fichero de imagen en formato X-Bitmap (blanco y negro).

.xpm

Contiene un fichero de imagen en formato X-Pixmap (color).

.jpeg o .jpg

Contiene un fichero de imagen en formato codificado jpeg.

.mpeg o .mpg

Contiene un fichero de imagen de video o cine (en movimiento).

.au

Contiene un fichero de audio (sonido) codificado en aiff-encoded.

.mid

Contiene un fichero de audio (sonido) generado con secuenciadores midi típicos de Windows.

.avi

Contiene un fichero con video y sonido típico de Windows95.

.exe

Contiene un fichero binario ejecutable en DOS.

.hqx

Contiene un fichero binario ejecutable en Macintosh.

.z o .zip

Contiene un (o varios) fichero/s comprimido/s de cualquier clase. Para poder utilizarlo hay que indicarle al visualizador una herramienta de compresión-descompresión.

En el entorno WWW se utilizan los **MIME** types (**M**ultipurpose **I**nternet **M**ail **E**xtension) para definir el tipo de un fichero transferido. El visualizador determina desde el MIME type cómo hay que tratar cada fichero.

Los visualizadores tienen una configuración de los mapas de tipos aceptables, e instrucciones de cómo proceder en cada caso. Algunos tipos son tratados directamente por el propio visualizador (p.ej.: htm, gif, txt) mientras que para otros necesitan de una herramienta auxiliar adecuada, que hay que indicar al visualizador, para que sea llamada en el momento preciso (p.ej.: zip, mpg, au).

Ahora que ya sabes qué son las extensiones de fichero, es el momento de aclarar una cuestión importante:

Los documentos HTML los puedes escribir con lo que quieras... Siempre que los salves en modo **Solo Texto**. Es decir, que si utilizas, por ejemplo, Word de Windows o de MAC, por defecto estos programas salvan en formato Word. Y aunque luego los renombres como documentos **.htm** o **.html**, el formato interno sigue siendo Word.

Tampoco vale dejarlos con la extensión **.txt** que les dejan la mayoría de editores al salvarlos en modo Solo Texto. No serían interpretados correctamente. Por lo tanto si tienen extensión **.txt** (y por supuesto **son .txt**), entonces puedes renombrarlos a **.htm**, o bien en el momento de guardarlos en formato **Solo Texto** no dejar la extensión por defecto **.txt** y darle ya directamente la extensión **.htm**.

Algunos procesadores de texto, ya incluyen entre sus tipos el **.htm**. Si es así lo que hacen es eliminar la codificación propia, y convertirla a elementos de HTML. Por ejemplo, si tienes un texto en *itálica*, automáticamente es convertido al elemento `<I>`, etc..

[\[Indice\]](#)

[\[Indice\]](#)

Cabecera (HEAD) de un documento HTML

La **HEAD** es la primera de las dos partes en que se estructura un documento HTML.

En la HEAD reside información acerca del documento, y generalmente no se ve cuando se navega por él. En la HEAD se pone el elemento lleno `<TITLE>` que es una breve descripción que identifica el documento. Es lo que el visualizador se guarda en el "**Bookmarks**", con lo que crea la lista que aparece en la orden "**Go**" de la barra de órdenes y lo que aparece en la esquina superior izquierda cuando se imprime el documento.

No hay que confundir el elemento `<TITLE>` con el nombre del fichero.

Por ejemplo, esta página está contenida en un fichero llamado **head.htm** y el texto de su `<TITLE>` es: **Head de un documento**. Se escribirá así:

```
<HEAD>
<TITLE>Head de un documento </TITLE>
</HEAD>
```

Dentro de la HEAD puede utilizarse otro elemento: **META**. Por ejemplo, si se escribe:

```
<HEAD>
<TITLE>Head de un documento </TITLE>
<META HTTP-EQUIV="Refresh" CONTENT="10">
</HEAD>
```

Esto hace que el visualizador vuelva a cargar la página activa al cabo de 10 segundos. También puede hacerse a un servidor. Así:

```
<HEAD>
<TITLE>Head de un documento </TITLE>
<META HTTP-EQUIV="Refresh" CONTENT="10; URL=http://miservidor/mipagina.htm">
</HEAD>
```

Utiliza esto con precaución. Si sabes que el contenido de la página no va a cambiar, es inútil hacer esto, y si lo haces contra un servidor, puedes sobrecargarlo. Este elemento, sólo tendrá utilidad en casos muy especiales.

Hay otros elementos que pueden aparecer en la HEAD, como ISINDEX, NEXTID, LINK o BASE, pero son de uso muy especializado y poco corriente, algunos ya en desuso, y ninguno obligatorio (`<TITLE>` sí lo es).

[\[Indice\]](#)

[\[Indice\]](#)

Cuerpo (BODY) de un documento

El cuerpo (**BODY**) es la segunda y última de las dos partes en que se estructura un documento HTML. Por supuesto es obligatoria, ya que es aquí donde reside el verdadero contenido de la página, y por tanto, al contrario de la HEAD sí se ve cuando navegamos por ella.

Se escribirá así:

```
<BODY>
```

```
texto  
texto texto texto texto texto texto texto texto texto texto texto texto
```

```
</BODY>
```

En las siguientes páginas se analizan los elementos más habituales del lenguaje que deben escribirse en esta parte del documento.

[\[Indice\]](#)

Tamaños y tipos de letra en HTML

Para definir distintos tamaños de letra, en HTML se utiliza el elemento lleno `<Hx> </Hx>` donde x es un número que puede variar entre 1 y 6, siendo 1 el tamaño mayor.

Se escribirán así:

```
<H1> Texto de prueba (H1) </H1> .  
<H2> Texto de prueba (H2) </H2>  
<H3> Texto de prueba (H3) </H3>  
<H4> Texto de prueba (H4) </H4>  
<H5> Texto de prueba (H5) </H5>  
<H6> Texto de prueba (H6) </H6>
```

y este sería el resultado:

Texto de prueba (H1)

Texto de prueba (H2)

Texto de prueba (H3)

Texto de prueba (H4)

Texto de prueba (H5)

Texto de prueba (H6)

No hay que olvidar poner el cierre `</Hx>` a cada elemento utilizado.

Este tipo de elemento se suele utilizar para escribir encabezamientos, ya que después del cierre automáticamente el visualizador inserta un salto de párrafo.

Por ejemplo: si escribes

```
<H1> Texto en H1 </H1> <H3> Texto en H3 </H3>
```

se verá:

Texto en H1

Texto en H3

Y no una cosa al lado de la otra, como cabría esperar.

Otra forma de cambiar los tamaños de letra es utilizar el elemento `` con el atributo **VALOR**, que es un número entre 1 y 7. El valor por defecto del texto es 3, por lo que valor puede ser positivo (+) o negativo (-) respecto a 3. Una gran ventaja de esta notación respecto a la anterior es que no se produce un salto de párrafo después de cada cambio, por lo que pueden hacerse cosas como esta:

Tamanos y tipos de letra

```
<FONT SIZE=3>A</font><FONT SIZE=4>A</font><FONT SIZE=5>A</font>  
<FONT SIZE=6>A</font><FONT SIZE=7>A</font><FONT SIZE=6>A</font>  
<FONT SIZE=5>A</font><FONT SIZE=4>A</font><FONT SIZE=3>A</FONT>
```

Dará como resultado:

AAAAA

Se puede cambiar el tamaño por defecto (3) de toda la página con el elemento **<BASEFONT SIZE=valor>**. El texto tomará el tamaño indicado por **valor** y lo mantendrá hasta que aparezca otro elemento **<BASEFONT SIZE=valor>** y lo restaure o lo cambie por otro diferente.

Con la versión 3.0 de Netscape se ha implementado un nuevo atributo del elemento **** que permite elegir tipos de letra entre los varios de que dispone por defecto Windows. Se trata del atributo **FACE**. Este atributo permite forzar el tipo de letra que el diseñador de la página quiere que vea el cliente, sin importar el que por defecto tenga establecido el visualizador.

Si escribes

```
&LT;FONT FACE="arial"&GT;Texto de prueba 12345 con tipo ARIAL</FONT>  
&LT;FONT FACE="times new roman">Texto de prueba 12345 con tipo TIMES NEW ROMAN</FONT>  
&LT;FONT FACE="courier new">Texto de prueba 12345 con tipo COURIER NEW</FONT>  
&LT;FONT FACE="courier">Texto de prueba 12345 con tipo COURIER</FONT>  
&LT;FONT FACE="roman">Texto de prueba 12345 con tipo ROMAN</FONT>  
&LT;FONT FACE="small fonts">Texto de prueba 12345 con tipo SMALL FONTS</FONT>
```

Se verá:

Texto de prueba 12345 con tipo ARIAL
Texto de prueba 12345 con tipo TIMES NEW ROMAN
Texto de prueba 12345 con tipo COURIER NEW
Texto de prueba 12345 con tipo COURIER
Texto de prueba 12345 con tipo ROMAN
Texto de prueba 12345 con tipo SMALL FONTS

Por supuesto, este atributo es compatible con todos los demás ya conocidos, como color y tamaño. Por ejemplo, si escribes

```
<FONT FACE="impact" SIZE=6 COLOR="red">  
  Texto de prueba 12345 con tipo IMPACT</FONT>
```

Se verá:

Texto de prueba 12345 con tipo IMPACT

Se pueden hacer todas la combinaciones que se quieran, pero hay que tener presente que si en la máquina cliente no está instalada una determinada fuente, ésta no se verá y en su lugar aparecerá la fuente por defecto del visualizador. No es interesante por tanto, definir tipos raros, que probablemente, no llegarán a verse nunca.

[\[Indice\]](#)

Texto en color

Se puede controlar el color del texto utilizando el elemento con el atributo **COLOR=xxx**, donde **xxx** es el nombre en inglés del color que se desea. Hay que tener presente que algunos no se verán o se verán mal si la máquina no soporta 256 colores. Por supuesto, este efecto es anidable con el de tamaño y todos los demás.

Si escribes:

```
<B><FONT COLOR="red">Texto ROJO </FONT>
<br>
<FONT COLOR="blue">Texto AZUL </FONT>
<br>
<FONT COLOR="navy">Texto AZUL MARINO </FONT>
<br>
<FONT COLOR="green">Texto VERDE </FONT>
<br>
<FONT COLOR="olive">Texto OLIVA </FONT>
<br>
<FONT COLOR="yellow">Texto AMARILLO </FONT>
<br>
<FONT COLOR="lime">Texto LIMA </FONT>
<br>
<FONT COLOR="magenta">Texto MAGENTA </FONT>
<br>
<FONT COLOR="purple">Texto PURPURA </FONT>
<br>
<FONT COLOR="cyan">Texto CYAN </FONT>
<br>
<FONT COLOR="brown">Texto MARRON </FONT>
<br>
<FONT COLOR="black">Texto NEGRO </FONT>
<br>
<FONT COLOR="gray">Texto GRIS </FONT>
<br>
<FONT COLOR="teal">Texto TEAL </FONT>
<br>
<FONT COLOR="white">Texto BLANCO </FONT>
<br>
</B>
```

Texto en color

Se verá:

Texto ROJO

Texto AZUL

Texto AZUL MARINO

Texto VERDE

Texto OLIVA

Texto AMARILLO

Texto LIMA

Texto MAGENTA

Texto PURPURA

Texto CYAN

Texto MARRON

Texto NEGRO

Texto GRIS

Texto TEAL

He aquí una combinación de colores y tamaños:

Si escribes:

```
<FONT SIZE=6 COLOR="red">E</FONT><FONT SIZE=4>sto es una </FONT>  
<FONT SIZE=6 COLOR="red">P</FONT><FONT SIZE=4>rueba </FONT>
```

Resulta:

Esto es una **P**rueba

Los colores también se pueden definir en hexadecimal (por ejemplo el rojo es ``). Cuando son colores básicos, como los del ejemplo de arriba, es más cómodo escribir el nombre aunque sea en inglés, pero cuando se trata de definir un color que "no tiene nombre" no hay más remedio que usar la codificación hexadecimal. En el índice encontrarás una tabla con los [códigos de colores](#).

[\[Indice\]](#)

Cambios de párrafo y de línea. Línea horizontal. Divisiones de texto

Para definir los saltos de párrafo se utiliza el elemento lleno `<P>` `</P>` (por **P**aragraph). No suele utilizarse el cierre `</P>`, ya que el texto continuará normalmente hasta que encuentre otro salto `<P>`

Se escribirá así:

```
Texto 1  
Texto 1 Texto 1 <P>
```

```
Texto 2 Texto 2
```

y este sería el resultado:

Texto 1 Texto 1

Texto 2 Texto 2

Como puede verse, hay un línea en blanco entre los dos bloques. El efecto del elemento `
` (por **B**reak) es el mismo, pero sin esa línea vacía.

Se escribirá así:

```
Texto 1  
Texto 1 Texto 1 <BR>
```

```
Texto 2 Texto 2
```

y este sería el resultado:

Texto 1 Texto 1

Texto 2 Texto 2 Texto 2 Texto 2 Texto 2 Texto 2 Texto 2

El elemento `<P>` admite tres atributos: **ALIGN=LEFT** (por defecto) **ALIGN=RIGHT** y **ALIGN=CENTER**

Se escribirán así:

```
<P ALIGN=LEFT>
```

```
Texto 1  
Texto 1 Texto 1 Texto 1 Texto 1 Texto 1 Texto 1 Texto 1 Texto 1
```

```
<P ALIGN=RIGHT>
```

```
Texto 2  
Texto 2 Texto 2 Texto 2 Texto 2 Texto 2 Texto 2 Texto 2 Texto 2
```

```
<P ALIGN=CENTER>
```

Párrafos y líneas

Texto 3
Texto 3 Texto 3 Texto 3 Texto 3 Texto 3 Texto 3 Texto 3 Texto 3

y este sería el resultado:

Texto 1
Texto 1 Texto 1 Texto 1 Texto 1

Texto 2
Texto 2 Texto 2 Texto 2 Texto 2

Texto 3
Texto 3 Texto 3 Texto 3 Texto 3

Un elemento que se comporta de forma parecida a
 es <DIV> pero que admite los mismos atributos que <P>: **ALIGN=LEFT** **ALIGN=RIGHT** y **ALIGN=CENTER**

Se escribe así:

<DIV ALIGN=LEFT>

texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1
texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1
texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1
texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1
</DIV>

<DIV ALIGN=RIGHT>

texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2
texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2
texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2
texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2
texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2
< /DIV>

<DIV ALIGN=CENTER>

texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3
texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3
texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3
texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3
</DIV>

Fíjate en que aquí sí se utiliza el cierre </DIV>. Este sería el resultado:

texto1
texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1
texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1 texto1

texto2
texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2
texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2 texto2

texto3
texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3
texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3 texto3

Otro interesante efecto es el elemento lleno **<BLOCKQUOTE>** que sirve para presentar párrafos adentrados (como si estuviesen tabulados).

Se escribirá así:

<BLOCKQUOTE>

texto
texto texto texto texto texto texto texto texto texto texto texto

<BLOCKQUOTE>

texto
texto texto texto texto texto texto texto texto texto texto texto

</BLOCKQUOTE>

</BLOCKQUOTE>

Y este sería el resultado:

texto
texto texto texto texto

texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto

Fíjate que en este ejemplo hay un anidamiento, y por tanto, debe haber dos cierres **</BLOCKQUOTE>** al final

Otro separador de bloques de texto es el elemento vacío **<HR>** (por **H**orizontal **R**ule). Este sería el resultado:

Se puede cambiar su apariencia añadiéndole el atributo **<NOSHADE>**. Así:

<HR NOSHADE>

El resultado es:

El elemento **<HR>** admite dos parámetros: **WIDTH** y **SIZE**. El primero define la longitud de la línea y el segundo su anchura. No es obligado usar los dos a la vez

Por ejemplo, si escribes

<HR WIDTH=400 SIZE=5>

El resultado será:

El valor del atributo **WIDTH** se puede expresar, como en el ejemplo anterior, en número de puntos (píxels), o en tantos por ciento referidos al ancho total de la ventana. Así:

<HR WIDTH=80% SIZE=5>

El resultado será:

Además se puede indicar su posición respecto a los márgenes de la ventana con los atributos **ALIGN=****CENTER** (por defecto) **ALIGN=LEFT** y **ALIGN=RIGHT**. Por ejemplo:

```
<HR WIDTH=80% SIZE=5 ALIGN=LEFT>
```

El resultado será:



O bien:

```
<HR WIDTH=80% SIZE=5 ALIGN=RIGHT>
```

El resultado será:



[[Indice](#)]

Textos preformateados

Posiblemente ya te habrás dado cuenta de que, cuando escribes un texto con tu editor, a la hora de ver lo hecho con el visualizador, algunas cosas no están como tú las pusiste... Como por ejemplo poner dos o tres espacios en blanco entre palabras, o intentar encolumnar los datos de una pequeña tabla: ¡acaba todo junto y en la misma línea!

Este efecto se puede eliminar con el elemento lleno `<PRE>`. Al texto que va dentro del elemento PRE se le pueden aplicar todos los elementos que se quiera, siempre que sean elementos que no alteren la posición del texto. Por ejemplo si utilizas el elemento `<H>`, se rompería el preformateo pero no ocurrirá lo mismo con ``. Por defecto, los textos preformateados se ven con tipo de letra "curier", es decir de paso fijo y los no preformateados en "Times New Roman". Estos tipos son configurables en el visualizador.

Se escribirá así:

```
<PRE>
```

```

1  2  3  4  5  6  7      <B>Esto es una demostracion de</B>
8  9 10 11 12 13 14      texto preformateado.
15 16 17 18 19 20 21

```

```
</PRE>
```

y este sería el resultado:

```

1  2  3  4  5  6  7      Esto es una demostracion de
8  9 10 11 12 13 14      texto preformateado.
15 16 17 18 19 20 21

```

Dentro de un bloque preformateado también se pueden poner links a otros documentos y codificación para caracteres especiales (se verá más adelante lo que es eso).

Este elemento admite el parámetro `WIDTH=x` donde `x` define la máxima longitud de línea visualizada. No funciona con Netscape.

El navegador, por defecto, elimina varios espacios en blanco y los convierte en uno solo. Si lo que deseas es separar una palabra de otra más de un espacio, se puede recurrir a una instrucción especial que le indica esto al navegador, pero hay que darla codificada; funciona como un avance de carro, y aunque en pantalla cumple su objetivo, cuando se imprime la página, no todas las impresoras son capaces de interpretarlo correctamente. Se escribe: ` `; y como alias se utiliza: ` `;

[Indice](#)

[Indice](#)

Otros efectos en el texto

Una o varias líneas de texto, pueden estar centradas respecto a los márgenes de la ventana. Esto afecta sólo al texto que hay entre `<CENTER>` y `</CENTER>`, pero no a todo el párrafo. Por ejemplo:

```
texto texto
<CENTER>texto centrado texto centrado</CENTER>
texto texto
```

Se verá:

```
texto texto texto texto texto texto texto texto texto txto texto texto texto texto texto texto
                    texto centrado texto centrado
```

```
texto texto texto texto texto texto texto texto texto txto texto texto texto texto texto texto
```

Esto, además de a texto sólo, se puede aplicar a una tabla, una imagen o cualquier otra cosa.

Quedan algunos otros elementos que modifican el aspecto del texto. Algunos, aparentemente, hacen la misma cosa, y otros no funcionan con el visualizador de Netscape, por lo que se omiten aquí.

Texto en negrita:

```
<B>Texto en negrita</B>
```

Texto realzado:

```
<STRONG>Texto realzado</STRONG>
```

Texto en itálica:

```
<I>Texto en itálica</I>
```

Texto con énfasis:

```
<EM>Texto con énfasis</EM>
```

Texto ejemplo de código:

```
<CODE>Texto ejemplo de código</CODE>
```

Texto teletipo:

```
<TT>Texto teletipo</TT>
```

Texto subrayado:

```
<U>Texto subrayado</U>
```

Texto tachado:

```
<STRIKE>Texto tachado</STRIKE>
```

Texto de dirección:

```
<ADDRESS>Texto de dirección</ADDRESS>
```

Otros efectos

Texto intermitente

<BLINK>Texto intermitente</BLINK>

Texto superíndice

<SUP>Texto Superíndice</SUP>

Texto subíndice

<SUB>Texto Subíndice</SUB>

Texto grande

<BIG>Texto grande</BIG>

Texto pequeño

<SMALL>Texto pequeño</SMALL>

[\[Indice\]](#)

Listas y menús

Hay elementos que permiten crear texto con varios formatos de listado: Estos pueden ser **ordenados** (se refiere a numerados, no ordenados por algun criterio), **desordenados** (no numerados), **directorios** <DIR>, **menu** <MENU> y **listados de definición** <DL>. Veamos cómo es la sintaxis básica y apariencia de estos elementos, a partir de los cuales, pueden hacerse combinaciones muy complejas mediante anidamientos de unos con otros, hasta conseguir prácticamente cualquier presentación deseada:

Esto es una **lista ordenada** (numerada):

1. Primera linea
2. Segunda linea

Se escribe:

```
<OL>
<LI>Primera linea
<LI>Segunda linea
</OL>
```

Fíjate en que los elementos no tienen cierre. Terminan cuando aparece otro igual o se cierra la lista.

Esto es una **lista desordenada** (no numerada):

- Primera linea
- segunda linea

Se escribe:

```
<UL>
<LI>Primera linea
<LI>Segunda linea
</UL>
```

En este caso los números han sido sustituidos por unos puntos gruesos. Esa es la apariencia por defecto; se puede cambiar usando el atributo **TYPE** con tres valores (el valor por defecto es **DISC**). Con el valor **CIRCLE** se verá:

- Primera linea
- segunda linea

Se escribe:

```
<UL TYPE=CIRCLE>
<LI>Primera linea
<LI>Segunda linea
```


También puede usarse el valor **SQUARE**. Así:

- Primera línea
- segunda línea

Se escribe:

```
<UL TYPE=SQUARE>
<LI>Primera línea
<LI>Segunda línea
</UL>
```

Las listas ordenadas no sólo se pueden ordenar con números. También se pueden utilizar letras y numeración romana tanto en mayúsculas como minúsculas. Para esto se utiliza el atributo **TYPE** del elemento **** con los siguientes valores: **TYPE=1** (por defecto) para números, **TYPE=A** para letras mayúsculas, **TYPE=a** para letras minúsculas, **TYPE=I** para numeración romana en mayúsculas y **TYPE=i** para numeración romana en minúsculas.

Esto es una **lista ordenada con letras mayúsculas**:

- . Primera línea
- B. Segunda línea
- C. Tercera línea
- D. Cuarta línea

Se escribe:

```
<OL TYPE=A>
<LI>Primera línea
<LI>Segunda línea
<LI>Tercera línea
<LI>Cuarta línea
</OL>
```

Esto es una **lista ordenada con letras minúsculas**:

- . Primera línea
- b. Segunda línea
- c. Tercera línea
- d. Cuarta línea

Se escribe:

```
<OL TYPE=a>
<LI>Primera línea
<LI>Segunda línea
<LI>Tercera línea
```

```
<LI>Cuarta linea  
</OL>
```

Esto es una **lista ordenada con numeración romana en mayúsculas**:

- I. Primera linea
- II. Segunda linea
- III. Tercera linea
- IV. Cuarta linea

Se escribe:

```
<OL TYPE=I>  
<LI>Primera linea  
<LI>Segunda linea  
<LI>Tercera linea  
<LI>Cuarta linea  
</OL>
```

Esto es una **lista ordenada con numeración romana en minúsculas**:

- i. Primera linea
- ii. Segunda linea
- iii. Tercera linea
- iv. Cuarta linea

Se escribe:

```
<OL TYPE=i>  
<LI>Primera linea  
<LI>Segunda linea  
<LI>Tercera linea  
<LI>Cuarta linea  
</OL>
```

En algunos casos puede interesarnos que la lista no comience por el número 1 (por ejemplo si es una lista que continua en otra página). Se puede conseguir con el atributo **START** combinado con **TYPE**.

Esto es una **lista ordenada con letras mayúsculas y que comienza por la E**:

- E. Primera linea
- F. Segunda linea
- G. Tercera linea
- H. Cuarta linea

Se escribe:

```
<OL TYPE=A START=5>  
<LI>Primera linea  
<LI>Segunda linea
```

```
<LI>Tercera linea  
<LI>Cuarta linea  
</OL>
```

El número que pones en el atributo START indica en que número o letra comenzará la lista. la E es la quinta letra.

Esto es un **menú**:

- Primera linea
- Segunda linea

Se escribe:

```
<MENU>  
<LI>Primera linea  
<LI>Segunda linea  
</MENU>
```

La diferencia entre un menú y una lista desordenada, es que las líneas en la lista desordenada comienzan en el margen izquierdo y las del menú unas posiciones más a la derecha (aunque eso depende del visualizador, con Netscape se ven igual).

Esto es un **directorio**:

- Primera linea
- Segunda linea

Se escribe:

```
<DIR>  
<LI>Primera linea  
<LI>Segunda linea  
</DIR>
```

Ocurre lo mismo que con el menú, con Netscape no se aprecia diferencia.

Esto es una **lista de definicion**:

Primera linea
 Segunda linea

Se escribe:

```
<DL>  
<DT>Primera linea  
<DD>Segunda linea  
</DL>
```

Fíjate que ahora en lugar del elemento se utiliza <DT> y <DD>, y como tipo <DL>. Estos tres nuevos elementos también se pueden usar con cualquiera de los anteriores tipos de lista, alterando por completo su apariencia. Para hacerse una idea de la variedad de aspectos que se pueden conseguir, lo mejor es ver la página de [ejemplos de listados](#).

[\[Indice\]](#)

Ejemplos de listas y menús

Desordenadas

Desordenada simple con marcas

- Línea de texto 1
- Línea de texto 2
- Línea de texto 3

Se escribe:

```
<UL>  
  <LI>Línea de texto 1  
  <LI>Línea de texto 2  
  <LI>Línea de texto 3  
</UL>
```

Desordenada simple sin marcas. Comienzo en margen izquierdo

Línea de texto 1
Línea de texto 2
Línea de texto 3

Se escribe:

```
<DL>  
  <DT>Línea de texto 1  
  <DT>Línea de texto 2  
  <DT>Línea de texto 3  
</DL>
```

Desordenada simple sin marcas. Comienzo una posición a la derecha

Línea de texto 1
Línea de texto 2
Línea de texto 3

Se escribe:

```
<UL>  
  <DT>Línea de texto 1  
  <DT>Línea de texto 2
```

Ejemplos de listas

```
<DT>Linea de texto 3
</UL>
```

Desordenada simple con marcas. Comienzo una posición a la derecha

- Linea de texto 1
- Linea de texto 2
- Linea de texto 3

Se escribe:

```
<UL>
  <LI>Linea de texto 1
  <LI>Linea de texto 2
  <LI>Linea de texto 3
</UL>
```

Desordenada simple sin marcas. Comienzo dos posiciones a la derecha

Linea de texto 1
Linea de texto 2
Linea de texto 3

Se escribe:

```
<UL>
  <DD>Linea de texto 1
  <DD>Linea de texto 2
  <DD>Linea de texto 3
</UL>
```

Desordenada simple sin marcas. Comienzo tres posiciones a la derecha

Linea de texto 1
Linea de texto 2
Linea de texto 3

Se escribe:

```
<UL>
  <UL>
    <DD>Linea de texto 1
    <DD>Linea de texto 2
    <DD>Linea de texto 3
  </UL>
</UL>
```

Desordenada simple con marcas. Comienzo dos posiciones a la derecha

- Linea de texto 1

Ejemplos de listas

- Línea de texto 2
- Línea de texto 3

Se escribe:

```
<UL>  
  <UL TYPE=DISC>  
    <LI>Línea de texto 1  
    <LI>Línea de texto 2  
    <LI>Línea de texto 3  
  </UL>  
</UL>
```

Desordenada con 4 niveles sin marcas

Línea de texto 1

Línea de texto 2

Línea de texto 3

Línea de texto 3.1

Línea de texto 3.2

Línea de texto 3.2.1

Línea de texto 3.2.2

Línea de texto 3.2.2.1

Línea de texto 3.2.2.2

Línea 4 de texto

Se escribe:

```
<DL>  
  <DT>Línea de texto 1  
  <DT>Línea de texto 2  
  <DT>Línea de texto 3  
    <DL>  
      <DT>Línea de texto 3.1  
      <DT>Línea de texto 3.2  
        <DL>  
          <DT>Línea de texto 3.2.1  
          <DT>Línea de texto 3.2.2  
            <DL>  
              <DT>Línea de texto 3.2.2.1  
              <DT>Línea de texto 3.2.2.2  
            </DL>  
          </DL>  
        </DL>  
      </DL>  
    </DL>  
  <DT>Línea 4 de texto
```

</DL>

Desordenada con 4 niveles con marcas

- Línea de texto 1
- Línea de texto 2
- Línea de texto 3
 - Línea de texto 3.1
 - Línea de texto 3.2
 - Línea de texto 3.2.1
 - Línea de texto 3.2.2
 - Línea de texto 3.2.2.1
 - Línea de texto 3.2.2.2
- Línea 4 de texto

Se escribe:

```
<UL>
  <LI>Línea de texto 1
  <LI>Línea de texto 2
  <LI>Línea de texto 3
    <UL>
      <LI>Línea de texto 3.1
      <LI>Línea de texto 3.2
        <UL>
          <LI>Línea de texto 3.2.1
          <LI>Línea de texto 3.2.2
            <UL>
              <LI>Línea de texto 3.2.2.1
              <LI>Línea de texto 3.2.2.2
            </UL>
          </UL>
        </UL>
      </UL>
    </UL>
  <LI>Línea 4 de texto
</UL>
```

Ordenadas

Ordenada simple

1. Línea de texto 1
2. Línea de texto 2
3. Línea de texto 3

Se escribe:

```
<OL>
  <LI>Linea de texto 1
  <LI>Linea de texto 2
  <LI>Linea de texto 3
</OL>
```

Ordenada con 4 niveles, todos ordenados

1. Linea de texto 1
2. Linea de texto 2
3. Linea de texto 3
 1. Linea de texto 3.1
 2. Linea de texto 3.2
 1. Linea de texto 3.2.1
 2. Linea de texto 3.2.2
 1. Linea de texto 3.2.2.1
 2. Linea de texto 3.2.2.2
4. Linea de texto 4

Se escribe:

```
<OL>
  <LI>Linea de texto 1
  <LI>Linea de texto 2
  <LI>Linea de texto 3
    <OL>
      <LI>Linea de texto 3.1
      <LI>Linea de texto 3.2
        <OL>
          <LI>Linea de texto 3.2.1
          <LI>Linea de texto 3.2.2
            <OL>
              <LI>Linea de texto 3.2.2.1
              <LI>Linea de texto 3.2.2.2
            </OL>
          </OL>
        </OL>
      </OL>
    </OL>
  <LI>Linea de texto 4
</OL>
```

Ordenada con 4 niveles, todos ordenados. Numeración romana en mayúsculas y minúsculas

- I. Linea de texto 1

Ejemplos de listas

II. Línea de texto 2

III. Línea de texto 3

i. Línea de texto 3.1

ii. Línea de texto 3.2

i. Línea de texto 3.2.1

ii. Línea de texto 3.2.2

i. Línea de texto 3.2.2.1

ii. Línea de texto 3.2.2.2

IV. Línea de texto 4

Se escribe:

```
<OL TYPE=I>
  <LI>Línea de texto 1
  <LI>Línea de texto 2
  <LI>Línea de texto 3
    <OL TYPE=i>
      <LI>Línea de texto 3.1
      <LI>Línea de texto 3.2
        <OL TYPE=i>
          <LI>Línea de texto 3.2.1
          <LI>Línea de texto 3.2.2
            <OL TYPE=i>
              <LI>Línea de texto 3.2.2.1
              <LI>Línea de texto 3.2.2.2
            </OL>
          </OL>
        </OL>
      </OL>
    </OL>
  <LI>Línea de texto 4
</OL>
```

Ordenada con 4 niveles, 2 desordenados con marcas

1. Línea de texto 1

2. Línea de texto 2

3. Línea de texto 3

1. Línea de texto 3.1

2. Línea de texto 3.2

■ Línea de texto 3.2.1

■ Línea de texto 3.2.2

○ Línea de texto 3.2.2.1

○ Línea de texto 3.2.2.2

4. Línea de texto 4

Se escribe:

```
<OL>
  <LI>Línea de texto 1
  <LI>Línea de texto 2
  <LI>Línea de texto 3
    <OL>
      <LI>Línea de texto 3.1
      <LI>Línea de texto 3.2
        <UL TYPE=SQUARE>
          <LI>Línea de texto 3.2.1
          <LI>Línea de texto 3.2.2
            <UL TYPE=CIRCLE>
              <LI>Línea de texto 3.2.2.1
              <LI>Línea de texto 3.2.2.2
            </UL>
          </UL>
        </OL>
      <LI>Línea de texto 4
    </OL>
```

Ordenada con 4 niveles, 2 desordenados sin marcas

1. Línea de texto 1
2. Línea de texto 2
3. Línea de texto 3
 1. Línea de texto 3.1
 2. Línea de texto 3.2
 - Línea de texto 3.2.1
 - Línea de texto 3.2.2
 - Línea de texto 3.2.2.1
 - Línea de texto 3.2.2.2
4. Línea de texto 4

Se escribe:

```
<OL>
  <LI>Línea de texto 1
  <LI>Línea de texto 2
  <LI>Línea de texto 3
    <OL>
      <LI>Línea de texto 3.1
      <LI>Línea de texto 3.2
```

Ejemplos de listas

```
<DL>
  <DD>Linea de texto 3.2.1
  <DD>Linea de texto 3.2.2
    <DL>
      <DD>Linea de texto 3.2.2.1
      <DD>Linea de texto 3.2.2.2
    </DL>
  </DL>
</OL>
<LI>Linea de texto 4
</OL>
```

[\[Indice\]](#)

Tablas

Las tablas son sin duda uno de los elementos más potentes del HTML. Con pocos elementos se pueden conseguir efectos espectaculares. En el interior de las celdas de una tabla, que pueden ser con borde visible o invisible, se puede poner cualquier cosa: texto de cualquier tamaño y color, imágenes, links... Por supuesto, además de permitir cualquier contenido, tienen sus propios atributos de alineación tanto horizontal como vertical, y atributos de dimensionado. Por defecto se autodimensionan, es decir, se adaptan en tamaño a su contenido, pero también es posible definirlo de forma fija.

El elemento básico es `<TABLE>` y en su interior los elementos `<TD>` `<TR>` `<TH>` y `<CAPTION>`.

He aquí un resumen de los elementos utilizados y los atributos que admite cada uno:

	TABLE	TR	TD	TH	CAPTION
BORDER	X	-	-	-	-
ROWSPAN	-	-	X	X	-
COLSPAN	-	-	X	X	-
ALIGN	-	X	X	X	X
VALIGN	-	-	X	-	-
WIDTH	X	-	X	-	-
HEIGHT	X	-	X	-	-
CELLPADDING	X	-	-	-	-
CELLSPACING	X	-	-	-	-

Resumen de tablas

Las posibilidades son tan amplias, que lo mejor es ver la [página de ejemplos de tablas](#) que encontrarás en el índice.

Ejemplos de tablas

Tabla básica de 3x2

A	B	C
D	E	F

```
<TABLE BORDER>
  <TR>
    <TD>A</TD> <TD>B</TD> <TD>C</TD>
  </TR>
  <TR>
    <TD>D</TD> <TD>E</TD> <TD>F</TD>
  </TR>
</TABLE>
```

Dos ejemplos de línea expandida <ROWSPAN>

Item 1	Item 2	Item 3
Item 4		Item 5

```
<TABLE BORDER>
  <TR>
    <TD>Item 1</TD>
    <TD ROWSPAN=2>Item 2</TD>
    <TD>Item 3</TD>
  </TR>
  <TR>
    <TD>Item 4</TD> <TD>Item 5</TD>
  </TR>
</TABLE>
```

Item 1	Item 2	Item 3	Item 4
	Item 5	Item 6	Item 7

```
<TABLE BORDER>
  <TR>
    <TD ROWSPAN=2>Item 1</TD>
    <TD>Item 2</TD> <TD>Item 3</TD> <TD>Item 4</TD>
  </TR>
  <TR>
    <TD>Item 5</TD> <TD>Item 6</TD> <TD>Item 7</TD>
  </TR>
</TABLE>
```

Ejemplo de columna expandida <COLSPAN>

Item 1	Item 2	
Item 3	Item 4	Item 5

```
<TABLE BORDER>
  <TR>
    <TD>Item 1</TD>
    <TD COLSPAN=2>Item 2</TD>
  </TR>
  <TR>
    <TD>Item 3</TD> <TD>Item 4</TD> <TD>Item 5</TD>
  </TR>
</TABLE>
```

Tabla con cabeceras <TH>

Head1	Head2	Head3
A	B	C
D	E	F

```
<TABLE BORDER>
  <TR>
    <TH>Head1</TH> <TH>Head2</TH> <TH>Head3</TH>
  </TR>
  <TR>
    <TD>A</TD> <TD>B</TD> <TD>C</TD>
  </TR>
  <TR>
    <TD>D</TD> <TD>E</TD> <TD>F</TD>
  </TR>
</TABLE>
```

Combinación de columna expandida y cabecera

Head1		Head2	
A	B	C	D
E	F	G	H

```
<TABLE BORDER>
  <TR>
    <TH COLSPAN=2>Head1</TH>
    <TH COLSPAN=2>Head2</TH>
  </TR>
  <TR>
    <TD>A</TD> <TD>B</TD> <TD>C</TD> <TD>D</TD>
  </TR>
```

Ejemplos de tablas

```
<TR>
    <TD>E</TD> <TD>F</TD> <TD>G</TD> <TD>H</TD>
</TR>
</TABLE>
```

Combinación de cabeceras múltiples y columnas expandidas

Head1		Head2	
Head 3	Head 4	Head 5	Head 6
A	B	C	D
E	F	G	H

```
<TABLE BORDER>
  <TR>
    <TH COLSPAN=2>Head1</TH>
    <TH COLSPAN=2>Head2</TH>
  </TR>
  <TR>
    <TH>Head 3</TH> <TH>Head 4</TH>
    <TH>Head 5</TH> <TH>Head 6</TH>
  </TR>
  <TR>
    <TD>A</TD> <TD>B</TD> <TD>C</TD> <TD>D</TD>
  </TR>
  <TR>
    <TD>E</TD> <TD>F</TD> <TD>G</TD> <TD>H</TD>
  </TR>
</TABLE>
```

Cabeceras laterales

Head1	Item 1	Item 2	Item 3
Head2	Item 4	Item 5	Item 6
Head3	Item 7	Item 8	Item 9

```
<TABLE BORDER>
  <TR><TH>Head1</TH>
    <TD>Item 1</TD> <TD>Item 2</TD> <TD>Item 3</TD></TR>
  <TR><TH>Head2</TH>
    <TD>Item 4</TD> <TD>Item 5</TD> <TD>Item 6</TD></TR>
  <TR><TH>Head3</TH>
    <TD>Item 7</TD> <TD>Item 8</TD> <TD>Item 9</TD></TR>
</TABLE>
```

Combinación de cabeceras laterales y líneas expandidas

Head1	Item 1	Item 2	Item 3	Item 4
	Item 5	Item 6	Item 7	Item 8

Ejemplos de tablas

Head2	Item 9	Item 10	Item 3	Item 11
--------------	--------	---------	--------	---------

```
<TABLE BORDER>
  <TR><TH ROWSPAN=2>Head1</TH>
    <TD>Item 1</TD> <TD>Item 2</TD> <TD>Item 3</TD> <TD>Item 4</TD>
  </TR>
  <TR><TD>Item 5</TD> <TD>Item 6</TD> <TD>Item 7</TD> <TD>Item 8</TD>
  </TR>
  <TR><TH>Head2</TH>
    <TD>Item 9</TD> <TD>Item 10</TD> <TD>Item 3</TD> <TD>Item 11</TD>
  </TR>
</TABLE>
```

Combinación de cabeceras superiores y laterales. Tabla centrada. Contenidos centrados. Título al pie.

	TABLE	TR	TD	TH	CAPTION
BORDER	X	-	-	-	-
ROWSPAN	-	-	X	X	-
COLSPAN	-	-	X	X	-
ALIGN	-	X	X	X	X
VALIGN	-	-	X	-	-
WIDTH	X	-	X	-	-
HEIGHT	X	-	X	-	-
CELLPADDING	X	-	-	-	-
CELLSPACING	X	-	-	-	-

Resumen de tablas

```
<CENTER>
<TABLE BORDER>
<CAPTION ALIGN=bottom>Resumen de tablas</CAPTION>
<TR>
<TD><TH>TABLE</TH><TH>TR</TH><TH>TD</TH><TH>TH</TH><TH>CAPTION</TH></TD>
</TR>

<TR ALIGN=CENTER>
<TH>BORDER</TH><TD>X</TD><TD>-</TD><TD>-</TD><TD>-</TD><TD>-</TD>
</TR>

<TR ALIGN=CENTER>
<TH>ROWSPAN</TH><TD>-</TD><TD>-</TD><TD>X</TD><TD>X</TD><TD>-</TD>
</TR>

<TR ALIGN=CENTER>
<TH>COLSPAN</TH><TD>-</TD><TD>-</TD><TD>X</TD><TD>X</TD><TD>-</TD>
</TR>
```

Ejemplos de tablas

```
<TR ALIGN=CENTER>
<TH>ALIGN</TH><TD>-</TD><TD>X</TD><TD>X</TD><TD>X</TD><TD>X</TD>
</TR>
```

```
<TR ALIGN=CENTER>
<TH>VALIGN</TH><TD>-</TD><TD>-</TD><TD>X</TD><TD>-</TD><TD>-</TD>
</TR>
```

```
<TR ALIGN=CENTER>
<TH>WIDTH</TH><TD>X</TD><TD>-</TD><TD>X</TD><TD>-</TD><TD>-</TD>
</TR>
```

```
<TR ALIGN=CENTER>
<TH>HEIGHT</TH><TD>X</TD><TD>-</TD><TD>X</TD><TD>-</TD><TD>-</TD>
</TR>
```

```
<TR ALIGN=CENTER>
<TH>CELLPADDING</TH><TD>X</TD><TD>-</TD><TD>-</TD><TD>-</TD><TD>-</TD>
</TR>
```

```
<TR ALIGN=CENTER>
<TH>CELLSPACING</TH><TD>X</TD><TD>-</TD><TD>-</TD><TD>-</TD><TD>-</TD>
</TR>
</TABLE>
</CENTER>
```

Ejemplo con todos los elementos y parámetros

		Media	
		Altura	Peso
Sexo	Hombres	1.9	85
	Mujeres	1.7	60

```
<TABLE BORDER>
  <TR>
    <TD><TH ROWSPAN=2></TH>
    <TH COLSPAN=2>Media</TH></TD>
  </TR>
  <TR>
    <TD><TH>Altura</TH><TH>Peso</TH></TD>
  </TR>
  <TR>
    <TH ROWSPAN=2>Sexo</TH>
    <TH>Hombres</TH><TD>1.9</TD><TD>85</TD>
  </TR>
  <TR>
    <TH>Mujeres</TH><TD>1.7</TD><TD>60</TD>
  </TR>
</TABLE>
```

Otro ejemplo de línea y columna expandidos

A	1	2
---	---	---

Ejemplos de tablas

	3	4
C	D	

```
<TABLE BORDER>
  <TR>
    <TD ALIGN=center ROWSPAN=2 COLSPAN=2>A</TD>
    <TD>1</TD>
    <TD>2</TD>
  </TR>
  <TR>
    <TD>3</TD>
    <TD>4</TD>
  </TR>
  <TR>
    <TD ALIGN=center ROWSPAN=2 COLSPAN=2>C</TD>
    <TD ALIGN=center ROWSPAN=2 COLSPAN=2>D</TD>
  </TR>
  <TR>
  </TR>
</TABLE>
```

Ajustando márgenes y bordes

Tabla sin bordes

Item 1	Item 2	Item 3
Item 4	Item 5	

```
<TABLE>
  <TR>    <TD>Item 1</TD> <TD ROWSPAN=2>Item 2</TD> <TD>Item 3</TD>
  </TR>
  <TR>    <TD>Item 4</TD> <TD>Item 5</TD>
  </TR>
</TABLE>
```

Aquí no es visible el borde de las celdas debido a que se ha omitido el atributo **BORDER** del elemento `<TABLE>`. La omisión o no es equivalente a parametrizarlo; se consigue lo mismo así: **BORDER=0**, y como ya habrás deducido, el valor de **BORDER** puede ser variable. Mira el siguiente ejemplo:

Tabla con borde de 10 puntos

Item 1	Item 2
Item 3	Item 4

```
<TABLE BORDER=10>
  <TR>    <TD>Item 1</TD> <TD> Item 2</TD>
  </TR>
```

Ejemplos de tablas

```
<TR> <TD>Item 3</TD> <TD>Item 4</TD>
</TR>
</TABLE>
```

Dimensionado de celdas

A	B	C
D	E	F

```
<TABLE BORDER CELLPADDING=10 CELLSPACING=0>
<TR>
<TD>A</TD> <TD>B</TD> <TD>C</TD>
</TR>
<TR>
<TD>D</TD> <TD>E</TD> <TD>F</TD>
</TR>
</TABLE>
```

A	B	C
D	E	F

```
<TABLE BORDER CELLPADDING=0 CELLSPACING=10>
<TR>
<TD>A</TD> <TD>B</TD> <TD>C</TD>
</TR>
<TR>
<TD>D</TD> <TD>E</TD> <TD>F</TD>
</TR>
</TABLE>
```

A	B	C
D	E	F

```
<TABLE BORDER CELLPADDING=10 CELLSPACING=10>
<TR>
<TD>A</TD> <TD>B</TD> <TD>C</TD>
</TR>
<TR>
<TD>D</TD> <TD>E</TD> <TD>F</TD>
</TR>
</TABLE>
```

Ejemplos de tablas

A	B	C
D	E	F

```
<TABLE BORDER=5 CELLPADDING=10 CELLSPACING=10>
  <TR>
    <TD>A</TD> <TD>B</TD> <TD>C</TD>
  </TR>
  <TR>
    <TD>D</TD> <TD>E</TD> <TD>F</TD>
  </TR>
</TABLE>
```

Alineación, títulos y subtablas

Líneas múltiples en un tabla

Enero	Febrero	Marzo
Celda 1	Celda 2	Otra celda Celda 3
Celda 4	y esta es la celda 5	Celda 6

```
<TABLE BORDER>
  <TR>
    <TH>Enero</TH>
    <TH>Febrero</TH>
    <TH>Marzo</TH>
  </TR>
  <TR>
    <TD>Celda 1</TD>
    <TD>Celda 2</TD>
    <TD>Otra celda<br> Celda 3</TD>
  </TR>
  <TR>
    <TD>Celda 4</TD>
    <TD>y esta<br>es la celda 5</TD>
    <TD>Celda 6</TD>
  </TR>
</TABLE>
```

Alineado horizontal del contenido de las celdas. ALIGN=LEFT | RIGHT | CENTER

Se puede aplicar individualmente a una celda o a toda la línea

Enero	Febrero	Marzo
Todas alineadas al centro	Celda 2	Otra celda Celda 3
Alineado a la derecha	Alineado al centro	Por defecto Alineado a la izquierda

```
<TABLE BORDER>
  <TR>
    <TH>Enero</TH>
    <TH>Febrero</TH>
    <TH>Marzo</TH>
  </TR>
  <TR ALIGN=center>
    <TD>Todas alineadas al centro</TD>
    <TD>Celda 2</TD>
    <TD>Otra celda<br> Celda 3</TD>
  </TR>
  <TR>
    <TD ALIGN=right>Alineado a la derecha</TD>
    <TD ALIGN=center>Alineado al centro</TD>
    <TD>Por defecto<br>Alineado a la izquierdat</TD>
  </TR>
</TABLE>
```

Alineado vertical del contenido de las celdas. VALIGN=TOP | BOTTOM | MIDDLE

Se puede aplicar individualmente a una celda o a toda la columna

Enero	Febrero	Marzo
Todas alineadas arriba	Esta es la Celda 2	Celda 3
Alineado arriba	Alineado abajo	Por defecto Alineado al centro

```
<TABLE BORDER>
  <TR>
    <TH>Enero</TH>
    <TH>Febrero</TH>
    <TH>Marzo</TH>
  </TR>
  <TR VALIGN=top>
    <TD>Todas alineadas arriba</TD>
    <TD>Esta es la<br>Celda 2</TD>
    <TD>Celda 3</TD>
```

Ejemplos de tablas

```
</TR>
<TR>
  <TD VALIGN=top>Alineado arriba</TD>
  <TD VALIGN=bottom>Alineado abajo</TD>
  <TD>Por defecto<br>Alineado al centro</TD>
</TR>
</TABLE>
```

Titulando las tablas. CAPTION=TOP | BOTTOM

Título arriba

Enero	Febrero	Marzo
Celda 1	Celda 2	Celda 3

```
<TABLE BORDER>
<CAPTION ALIGN=top>Titulo arriba</CAPTION>
<TR>
  <TH>Enero</TH>
  <TH>Febrero</TH>
  <TH>Marzo</TH>
</TR>
<TR>
  <TD>Celda 1</TD>
  <TD>Celda 2</TD>
  <TD>Celda 3</TD>
</TR>
</TABLE>
```

Enero	Febrero	Marzo
Celda 1	Celda 2	Celda 3

Título abajo

```
<TABLE BORDER>
<CAPTION ALIGN=bottom>Titulo abajo</CAPTION>
<TR>
  <TH>Enero</TH>
  <TH>Febrero</TH>
  <TH>Marzo</TH>
</TR>
<TR>
  <TD>Celda 1</TD>
  <TD>Celda 2</TD>
  <TD>Celda 3</TD>
</TR>
</TABLE>
```

Anidando tablas. La tabla ABCD dentro de la tabla 12345

Ejemplos de tablas

		3
1	2	A B
		C D
4	5	6

```
<TABLE BORDER>
  <TR> <!-- ROW 1, TABLE 1 -->
    <TD>1</TD>
    <TD>2</TD>
    <TD>3
      <TABLE BORDER>
        <TR> <!-- ROW 1, TABLE 2 -->
          <TD>A</TD>
          <TD>B</TD>
        </TR>
        <TR> <!-- ROW 2, TABLE 2 -->
          <TD>C</TD>
          <TD>D</TD>
        </TR>
      </TABLE>
    </TD>
  </TR>
  <TR> <!-- ROW 2, TABLE 1 -->
    <TD>4</TD>
    <TD>5</TD>
    <TD>6</TD>
  </TR>
</TABLE>
```

Longitud horizontal de las tablas

Tabla que ocupa el 50 % de la pantalla, y cuyas celdas están dimensionadas al 50 % cada una respecto a la longitud total de la tabla

Width=50%	Width=50%
Celda 3	Celda 4

```
<TABLE BORDER WIDTH="50%">
  <TR><TD>Width=50%</TD><TD>Width=50%</TD>
</TR>
  <TR><TD>Celda 3</TD><TD>Celda 4</TD>
</TR>
</TABLE>
```

Tabla dimensionada al 50 % de la pantalla, celdas sin dimensionar. Obsévese que se alargan más de lo que lo harían sin forzar la longitud de la tabla

Ejemplos de tablas

Celda 1	2
Celda 3	4

```
<TABLE BORDER WIDTH="50%">
  <TR><TD>Celda 1</TD><TD>2</TD>
</TR>
  <TR><TD>Celda 3</TD><TD>4</TD>
</TR>
</TABLE>
```

El mismo efecto anterior, pero con la tabla dimensionada al 100 %

WIDTH=100%	Celda 2
3	Celda 4

```
<TABLE BORDER WIDTH="100%">
  <TR><TD>WIDTH=100%</TD><TD>Celda 2</TD>
</TR>
  <TR><TD>3</TD><TD>Celda 4</TD>
</TR>
</TABLE>
```

Centrado de una tabla en la página

A	B	C
D	E	F

```
<CENTER>
<TABLE BORDER WIDTH="50%">
  <TR>
    <TD>A</TD> <TD>B</TD> <TD>C</TD>
  </TR>
  <TR>
    <TD>D</TD> <TD>E</TD> <TD>F</TD>
  </TR>
</TABLE>
</CENTER>
```

Tabla forzada al 50 % de la página, conteniendo otra tabla anidada en una de sus celdas. La segunda tabla está forzada a ocupar el 100 % de la celda receptora.

Item 1	Item 2
Item A	Item B
Item 4	

```
<TABLE BORDER WIDTH="50%">
  <TR><TD>Item 1</TD><TD>Item 2</TD>
</TR>
```

Ejemplos de tablas

```
<TR><TD>
  <TABLE BORDER WIDTH=100%>
    <TR><TD>Item A</TD><TD>Item B</TD>
    </TR>
  </TABLE>
</TD>
<TD>Item 4</TD>
</TR>
</TABLE>
```

Longitud vertical de las tablas

Tabla que ocupa el 50 % horizontal de la pantalla, y el 25 % vertical. HEIGHT=25%

HEIGHT=25%	Item 2
3	4

```
<TABLE BORDER WIDTH="50%" HEIGHT="25%">
  <TR><TD>HEIGHT=15%</TD> <TD>Item 2</TD>
  </TR>
  <TR><TD>3</TD><TD>4</TD>
  </TR>
</TABLE>
```

Fondos de colores en las tablas

Una tabla de cuatro celdas. Cada una de un color. (Sólo Netscape 3.0 o superior)

Texto ROJO	Texto VERDE
Texto AZUL	Texto AMARILLO

```
<TABLE BORDER>
<TR><TD BGCOLOR="RED">Texto ROJO</TD>
  <TD BGCOLOR="green">Texto VERDE</TD>
</TR>

<TR><TD BGCOLOR="blue">Texto AZUL</TD>
```

Ejemplos de tablas

```
<TD BGCOLOR="YELLOW">Texto AMARILLO</TD>
</TR>
</TABLE>
```

En este ejemplo se ha dado color individualmente a cada celda. El atributo **BGCOLOR="color"** puede utilizarse en cada elemento de la tabla para efectos globales. Por ejemplo, para hacer toda la tabla en rojo, sólo sería necesario escribir:

```
<TABLE BORDER BGCOLOR="red">
. . . .
. . . .
</TABLE>
```

[\[Indice\]](#)

Códigos hexadecimales de color

Estos códigos de color se pueden utilizar como valor de atributo en los elementos y <BODY BGCOLOR FGColor TEXT LINK VLINK ALINK>. Algunos no funcionarán en ciertos visualizadores, así como si Windows no está configurado con 256 colores al menos.

La codificación de estos colores está basada en mezclas cromáticas RGB (RED GREEN BLUE), es decir, se mezclan ciertas cantidades de los tres colores básicos (rojo verde y azul), para conseguir el color deseado. La cantidad de cada color está definida por el código que utilizamos, que tiene seis dígitos en hexadecimal. Si los cortamos en 3 grupos de dos de izquierda a derecha, obtendremos la cantidad (en hexadecimal) de cada color.

La sintaxis para cambiar el fondo de la página (background) a color rojo es: <BODY BGCOLOR=#FF0000> y para dar color al texto, ver la página [Texto de colores](#). Para combinaciones generales de colores, ver el [ejemplo de control de colores](#).

Lista de códigos RGB

Puedes probar los diez primeros para hacerte una idea.

[White](#) rgb=#FFFFFF

[Red](#) rgb=#FF0000

[Green](#) rgb=#00FF00

[Blue](#) rgb=#0000FF

[Magenta](#) rgb=#FF00FF

[Cyan](#) rgb=#00FFFF

[Yellow](#) rgb=#FFFF00

[Black](#) rgb=#000000

[Aquamarine](#) rgb=#70DB93

[Baker's Chocolate](#) rgb=#5C3317

[Blue Violet](#) rgb=#9F5F9F

[Brass](#) rgb=#B5A642

[Bright Gold](#) rgb=#D9D919

[Brown](#) rgb=#A62A2A

[Bronze](#) rgb=#8C7853

[Bronze II](#) rgb=#A67D3D

[Cadet Blue](#) rgb=#5F9F9F

[Cool Copper](#) rgb=#D98719

[Copper](#) rgb=#B87333

[Coral](#) rgb=#FF7F00

Tabla de colores

Corn Flower Blue rgb=#42426F
Dark Brown rgb=#5C4033
Dark Green rgb=#2F4F2F
Dark Green Copper rgb=#4A766E
Dark Olive Green rgb=#4F4F2F
Dark Orchid rgb=#9932CD
Dark Purple rgb=#871F78
Dark Slate Blue rgb=#6B238E
Dark Slate Grey rgb=#2F4F4F
Dark Tan rgb=#97694F
Dark Turquoise rgb=#7093DB
Dark Wood rgb=#855E42
Dim Grey rgb=#545454
Dusty Rose rgb=#856363
Feldspar rgb=#D19275
Firebrick rgb=#8E2323
Forest Green rgb=#238E23
Gold rgb=#CD7F32
Goldenrod rgb=#DBDB70
Grey rgb=#C0C0C0
Green Copper rgb=#527F76
Green Yellow rgb=#93DB70
Hunter Green rgb=#215E21
Indian Red rgb=#4E2F2F
Khaki rgb=#9F9F5F
Light Blue rgb=#C0D9D9
Light Grey rgb=#A8A8A8
Light Steel Blue rgb=#8F8FBD
Light Wood rgb=#E9C2A6
Lime Green rgb=#32CD32
Mandarian Orange rgb=#E47833
Maroon rgb=#8E236B
Medium Aquamarine rgb=#32CD99
Medium Blue rgb=#3232CD
Medium Forest Green rgb=#6B8E23
Medium Goldenrod rgb=#EAEAAE
Medium Orchid rgb=#9370DB
Medium Sea Green rgb=#426F42
Medium Slate Blue rgb=#7F00FF
Medium Spring Green rgb=#7FFF00
Medium Turquoise rgb=#70DBDB
Medium Violet Red rgb=#DB7093
Medium Wood rgb=#A68064
Midnight Blue rgb=#2F2F4F
Navy Blue rgb=#23238E
Neon Blue rgb=#4D4DFF

Neon Pink rgb=#FF6EC7
New Midnight Blue rgb=#00009C
New Tan rgb=#EBC79E
Old Gold rgb=#CFB53B
Orange rgb=#FF7F00
Orange Red rgb=#FF2400
Orchid rgb=#DB70DB
Pale Green rgb=#8FBC8F
Pink rgb=#BC8F8F
Plum rgb=#EAADEA
Quartz rgb=#D9D9F3
Rich Blue rgb=#5959AB
Salmon rgb=#6F4242
Scarlet rgb=#8C1717
Sea Green rgb=#238E68
Semi-Sweet Chocolate rgb=#6B4226
Sienna rgb=#8E6B23
Silver rgb=#E6E8FA
Sky Blue rgb=#3299CC
Slate Blue rgb=#007FFF
Spicy Pink rgb=#FF1CAE
Spring Green rgb=#00FF7F
Steel Blue rgb=#236B8E
Summer Sky rgb=#38B0DE
Tan rgb=#DB9370
Thistle rgb=#D8BFD8
Turquoise rgb=#ADEAEA
Very Dark Brown rgb=#5C4033
Very Light Grey rgb=#CDCDCD
Violet rgb=#4F2F4F
Violet Red rgb=#CC3299
Wheat rgb=#D8D8BF
Yellow Green rgb=#99CC32

[\[Indice\]](#)

[\[Indice\]](#)

Control de colores

Este es un documento de ejemplo de control de color. El texto es gris claro sobre negro, y los [links](#) son amarillos al principio, azul-verde cuando son activados, y verdes después de haber sido utilizados.

Todo son atributos del elemento <BODY>. Se escribe así:

```
<BODY BGCOLOR="#000000" FGColor="#00FF00" TEXT="#F0F0F0"  
LINK="#FFFF00" VLINK="22AA22" ALINK="#0077FF">
```

[\[Indice\]](#)

¿Por qué hay que usar códigos?

Existen diversos sistemas operativos dentro del mundo de los ordenadores. Estos sistemas no son otra cosa que unos programas especiales que se ejecutan inmediatamente después de encender el ordenador, y son los que se encargan de darle "vida" a nuestra máquina.

Junto con estos programas, los ordenadores cargan en memoria unas tablas especiales que coinciden con los caracteres que tiene tu teclado y algunos más que no están en el mismo, pero que se pueden escribir en pantalla por otros medios. Estas tablas, desafortunadamente, no son siempre las mismas, y varían de un sistema a otro, de una marca de ordenador a otra, y por supuesto, de un idioma a otro. Por ejemplo, no se ven igual las letras con acentos desde un PC que desde un MACintosh, o una terminal UNIX, si las escribimos directamente desde el teclado.

Dado que el HTML pretende ser un lenguaje universal, y que una página debe verse como su creador desea, sin importar si estamos ante un ordenador que "habla" en inglés o en español, o que tiene tal o cual sistema operativo, se ha creado una tabla de caracteres "conflictivos" (en realidad están todos, pero generalmente sólo se usa para caracteres especiales) que se escriben con un código en lugar de pulsar la tecla que lo contiene directamente.

Por ejemplo, nuestra denostada en medio mundo "Ñ" se escribirá:

```
&Ntilde; o bien &#209;
```

Esto habrá que hacerlo con todos los caracteres que no sean las letras del alfabeto, los números y unos pocos signos, como el punto, la coma, el guión y algunos otros. Desde luego hay que codificar todas las letras acentuadas, eñes, cedillas, etc., etc.

En el índice encontrarás [una tabla con todos los caracteres](#), también están las letras, pero éstas sólo te serán útiles en caso de necesitar escribir textos un tanto raros.

Como puedes ver, la tabla tiene una columna con el carácter deseado seguida de su código numérico, una descripción del carácter y después un nombre corto (una especie de alias) que para los más habituales se llegan a memorizar, y para cosas cortas, ayudan. No todos tienen ese nombre corto, y esos hay que escribirlos con el código numérico.

Generalmente, se pueden escribir directamente desde teclado todos los que no tienen alias en la tabla, los que sí tienen normalmente darán problemas en sistemas diferentes al que se ha utilizado para escribir el documento. Si lo que escribes en HTML tienes la seguridad de que sólo va a ser utilizado como ficheros locales en máquinas similares a la tuya, no será necesario complicarse la vida; pero si tus páginas van a residir en un servidor WWW, cualquier tipo de máquina podrá acceder a ellas, y su aspecto no será el adecuado en algunas.

Existen multitud de editores especiales para escribir HTML, y la mayoría de ellos tienen herramientas para convertir de forma automática los caracteres especiales a su correspondiente código. También hay

Porque codigos

paquetes de "macros" para hacer lo mismo con procesadores de texto como Word o Word Perfect.

[Indice](#)

Tabla de códigos

Caracter	Código	Descripción	Alias
!	!	Cerrar exclamación	--
"	"	Dobles comillas	--
#	#	Signo de número	--
\$	$	Dólar	--
%	%	Tanto por ciento	--
&	&	Ampersand	--
'	'	Apóstrofe	--
((Cerrar paréntesis	--
))	Abrir paréntesis	--
*	*	Asterisco	--
+	+	Signo más	--
,	,	Coma	--
-	-	Guión	--
.	.	Punto	--
/	/	Barra de división	--
0 - 9	0 - 9	Dígitos del al 0-9	--
:	:	Dos puntos	--
;	;	Punto y coma	--
<	<	Menor	--
<	<	Menor	lt
=	=	Igual	--
>	>	Mayor	--
>	>	Mayor	gt
?	?	Cerrar interrogación	--
@	@	Arroba	--
A - Z	A - Z	Letras A-Z	--
[[Cerrar corchete (izquierdo)	--
\	\	Barra inversa	--
]]	Abrir corchete	--

^	^	Circunflejo	--
_	_	Subrayado	--
`	`	Acento agudo	--
a - z	a - z	Letras a-z	--
{	{	Abrir llave	--
	|	Barra vertical	--
}	}	Cerrar llave	--
~	~	Tilde	--
	 	Espacio sin separación	nbsp
!	¡	Abrir exclamación	ixcl
¢	¢	Centavo	cent
£	£	Libra Esterlina	pound
¤	¤	Signo de divisa general	curren
¥	¥	Yen	yen
	¦	Barra vertical partida (pipe)	brvbar
§	§	Sección	sect
¨	¨	Diéresis	uml
©	©	Copyright	copy
ª	ª	Doña	ordf
«	«	Abrir comillas francesas	laquo
¬	¬	No (símbolo lógico)	not
-	­	Guión débil	shy
®	®	Registrado	reg
ˆ	¯	Macrón	macr
°	°	Grados	deg
±	±	Más-menos	plusmn
²	²	Dos superíndice	sup2
³	³	Tres superíndice	sup3
´	´	Acento agudo	acute
µ	µ	Micro	micro
¶	¶	Fin de párrafo	para
·	·	Punto medio	middot
¸	¸	Cedilla	cedil
¹	¹	Uno superíndice	sup1
º	º	Género masculino	ordm

»	»	Cerrar comillas francesas	raquo
¼	¼	Un cuarto	frac14
½	½	Un medio	frac12
¾	¾	Tres cuartos	frac34
¿	¿	Abrir interrogación	iquest
À	À	A mayúscula, acento grave	Agrave
Á	Á	A mayúscula, acento agudo	Aacute
Â	Â	A mayúscula, acento circunflejo	Acirc
Ã	Ã	A mayúscula, tilde	Atilde
Ä	Ä	A mayúscula, diéresis	Auml
Å	Å	A mayúscula, anillo	Aring
Æ	Æ	Æ mayúscula	AElig
Ç	Ç	C cedilla mayúscula	Ccedil
È	È	E mayúscula, acento grave	Egrave
É	É	E mayúscula, acento agudo	Eacute
Ê	Ê	E mayúscula, acento circunflejo	Ecirc
Ë	Ë	E mayúscula, diéresis	Euml
Ì	Ì	I mayúscula, acento grave	Igrave
Í	Í	I mayúscula, acento agudo	Iacute
Î	Î	I mayúscula, acento circunflejo	Icirc
Ï	Ï	I mayúscula, diéresis	Iuml
Ð	Ð	Eth mayúscula	ETH
Ñ	Ñ	Eñe mayúscula	Ntilde
Ò	Ò	O mayúscula, acento grave	Ograve
Ó	Ó	O mayúscula, acento agudo	Oacute
Ô	Ô	O mayúscula, acento circunflejo	Ocirc
Õ	Õ	O mayúscula, tilde	Otilde
Ö	Ö	O mayúscula, diéresis	Ouml
×	×	Signo de multiplicación	times
Ø	Ø	O barrada mayúscula	Oslash
Ù	Ù	U mayúscula, acento grave	Ugrave
Ú	Ú	U mayúscula, acento agudo	Uacute
Û	Û	U mayúscula, acento circunflejo	Ucirc
Ü	Ü	U mayúscula, diéresis	Uuml
Ý	Ý	Y mayúscula, acento agudo	Yacute

Þ	Þ	THORN mayúscula	THORN
β	ß	Beta minúscula	szlig
à	à	a minúscula, acento grave	agrave
á	á	a minúscula, acento agudo	aacute
â	â	a minúscula, acento circunflejo	acirc
ã	ã	a minúscula, tilde	atilde
ä	ä	a minúscula, diéresis	auml
å	å	a minúscula, anillo	aring
æ	æ	ae minúscula	aelig
ç	ç	c cedilla minúscula	ccedil
è	è	e minúscula, acento grave	egrave
é	é	e minúscula, acento agudo	eacute
ê	ê	e minúscula, acento circunflejo	ecirc
ë	ë	e minúscula, diéresis	euml
ì	ì	i minúscula, acento grave	igrave
í	í	i minúscula, acento agudo	iacute
î	î	i minúscula, acento circunflejo	icirc
ï	ï	i minúscula, diéresis	iuml
ð	ð	eth minúscula	eth
ñ	ñ	eñe minúscula	ntilde
ò	ò	o minúscula, acento grave	ograve
ó	ó	o minúscula, acento agudo	oacute
ô	ô	o minúscula, acento circunflejo	ocirc
õ	õ	o minúscula, tilde	otilde
ö	ö	o minúscula, diéresis	ouml
÷	÷	Signo de división	divide
ø	ø	o barrada minúscula	oslash
ù	ù	u minúscula, acento grave	ugrave
ú	ú	u minúscula, acento agudo	uacute
û	û	u minúscula, acento circunflejo	ucirc
ü	ü	u minúscula, diéresis	uuml
ý	ý	y minúscula, acento agudo	yacute
þ	þ	thorn minúscula	thorn
ÿ	ÿ	y minúscula, diéresis	yuml

[Indice](#)

Creación de enlaces (links)

El siguiente es sin duda el elemento más importante del HTML, ya que permite realmente "navegar" por uno o varios documentos, que pueden residir en cualquier parte, pareciéndonos que siempre estamos en el mismo; a esto se le llama **hipertexto** ya que con esta posibilidad, en realidad, nuestro documento puede ser infinito... En efecto, para la persona que está leyendo sobre un determinado tema, no hay diferencias ostensibles que le hagan notar dónde está el documento que lee. Para el lector, todo parece un mismo documento, cuando en realidad, probablemente el conjunto de lo que lee está repartido por medio mundo, o en un plano más modesto, el documento leído puede estar compuesto en realidad por varios cientos de páginas que "saltan" de unas a otras sin notarlo.

Todo esto lo consigue el elemento `<A>` (por **A**nchor, en inglés = ancla o punto de anclaje). En lo sucesivo le llamaremos enlace o simplemente **link** (en inglés link=eslabón o enlace).

Pero seguramente te preguntarás qué es eso de un "link". Pues un link es un área de la pantalla, que puede contener una o varias palabras o una imagen, que es "sensible" al puntero del ratón, y al ponerlo sobre ella y pulsar el botón izquierdo el visualizador llamará a la página que tiene asignada el link. Habitualmente por defecto los visualizadores señalan un área linkada subrayándola, si es texto, o poniéndole un borde si es una imagen, ambas cosas en color azul. Por supuesto, eso se puede cambiar en las configuraciones de visualizador. Si no se desea que aparezca el subrayado para mejorar la estética en algunos casos, se puede parametrizar con una instrucción de estilo.

Se escribirá:

```
<A HREF="http://www.miservidor.es/mifichero.htm"> Link de prueba </A>.
```

Y se vería así: [Link de prueba](http://www.miservidor.es/mifichero.htm)

Si lo que se desea es que aparezca sin subrayado o de otro color, o ambas cosas, se puede recurrir a una instrucción de estilo. Así:

```
<A HREF="http://www.miservidor.es/mifichero.htm" style="color:red; text-decoration:none";> Link de prueba </A>.
```

Y se vería así: [Link de prueba](http://www.miservidor.es/mifichero.htm) (Esto puede que no funcione en algunos navegadores antiguos)

El elemento `<A>` tiene dos atributos: **HREF** y **NAME**.

En el ejemplo de arriba al atributo HREF se le asigna el valor que aparece entre comillas: **"http://www.miservidor.es/mifichero.htm"**. El texto sobre el que se monta el enlace es **Link de prueba** y realiza un link con el fichero **mifichero.htm** que está en el servidor **www.miservidor.es**.

Esta forma de escribir la ruta del link de forma completa, se utiliza cuando hay que saltar de una máquina a otra, si el enlace es para otra página de nuestro propio servidor es suficiente escribir la ruta virtual corta: **/mifichero.htm** o bien: **/otro_directorio/mifichero.htm** según proceda.

El valor de los atributos hay que escribirlos EXACTAMENTE como se deba. En el ejemplo anterior no funcionaría el link si escribiéramos **www.miservidor.es** en mayúsculas o con alguna otra diferencia, eso es un

nombre de máquina y sería interpretado como OTRA máquina. Esto es aplicable a todos los parámetros: **servicio** (<http://>) **servidor** (www.miservidor.es) y **fichero** (mifichero.htm).

Estas precauciones son ineludibles en el caso de servidores montados en máquinas con sistema operativo UNIX. En el caso de que el servidor resida en una máquina con Windows NT o Windows 95/98, es indiferente. Si no se sabe el sistema que tiene la máquina, es mejor atenerse a la forma UNIX.

Dentro de los parámetros pasados a HREF, podrás ver que al principio pone **http://**, esto es, el tipo de servicio al que el visualizador va a llamar, y hay varios :

Servicio	Descripción	Ejemplo	Efecto
http://	Servicios WWW	<code> WWW</code>	WWW
ftp://	Servicios FTP	<code>FTP</code>	FTP
gopher://	Servicios GOPHER	<code>GOPHER </code>	GOPHER
news://	Servicios NEWS	<code>NEWS</code>	NEWS
mailto://	Servicios E-mail	<code>E-mail</code>	E-mail
file:///C 	Fichero local	<code>Fichero</code>	Fichero

Todos estos servicios se pueden utilizar desde el visualizador de Netscape.

Además de HREF, el elemento `<A>` puede tener otro atributo: **TARGET**. Se utiliza para ordenar la apertura de otra ventana del visualizador con la página que se desee. Se escribirá:

```
<A HREF="indice.htm" TARGET="Ventana-2"> </A>
```

Esta instrucción mostrará la página **indice.htm** pero con [otra ventana](#) del visualizador, es decir tendremos lanzado el visualizador **2 veces**. Esto sólo funciona si el sistema operativo de tu máquina es multi tarea (p.ej.: Windows-95).

Como se acaba de ver, el atributo **HREF** sirve para enlazar con otro documento, que puede estar en un servidor o ser un fichero local, pero que siempre se nos presentará desde la primera línea del mismo.

El atributo **NAME** se utiliza para definir algo así como "un punto de aterrizaje" en cualquier línea del documento de destino, de forma que nos aparecerá en pantalla desde la línea deseada y no desde el principio. Esto es muy útil cuando se trata de documentos largos divididos en secciones.

Se escribirá así:

En el documento activo:

```
<A HREF="http://www.miservidor.es/mifichero.htm#punto1"> Ir al punto 1</A>
```

En el documento destino:

```
<A NAME="punto1"></A>
```

NAME también puede utilizarse para saltar de una línea a otra dentro de un mismo documento. Se escribirá así:

Dentro del documento activo **En la línea de partida:**

```
<A HREF="#punto1">Ir al punto 1</A>
```

En la línea de destino

```
<A NAME="punto1"></A>
```

Por ejemplo, si haces "klik" [aquí](#) con el ratón, saltarás al principio de esta página.

Debes saber que...

- El texto que pongas después del símbolo # puede ser cualquiera, siempre que no tenga espacios en blanco, caracteres especiales ni caracteres codificados, y por supuesto, que no esté repetido en el mismo documento de destino.
- Si en el documento de destino no existe el punto definido en el documento de origen, el visualizador nos presentará en pantalla la última línea del mismo.
- Cuando hagas un link, es preferible utilizar direcciones relativas. Si utilizas direcciones absolutas y después tienes que mover los ficheros por cualquier razón, tendrás que modificar todas las direcciones.
- Si al hacer un link, después del nombre de la máquina no pones el nombre de un fichero, por defecto el servidor entiende que se le pide la "home page" (página inicial).
- Si un servidor no tiene definida página inicial, simplemente nos mostrará una lista de todos los ficheros y directorios que tenga en el directorio definido como raíz si el servidor tiene activada la opción de listar directorios, en caso contrario, dará un error.
- Si después de la dirección de la máquina a la que le haces el link en lugar de un nombre de fichero pones el de un directorio, generalmente el servidor nos mostrará un simple listado de los ficheros y directorios que éste contenga si el servidor tiene activada la opción de listar directorios, en caso contrario, dará un error.
- No sólo puedes montar el link sobre un texto, también puedes hacerlo sobre una imagen cualquiera. Es decir, después del link y antes de puedes poner lo que quieras.
- El arte de escribir buen HTML reside en dar una buena estructura a la información. Siempre que puedas huye de crear páginas muy largas. Crea cuantas necesites, con buena estructura y enlázalas con cuantos links sean precisos.
- Recuerda: si tu trabajo HTML no va a residir en un servidor en red, y has utilizado direcciones absolutas, al hacer un link, no empieces con aquello de: "**http://....**". No funcionará.
- Si continuas deseando poner direcciones absolutas con ficheros locales, tendrá que ser así **file:///C:/MIDIRECTORIO/mifichero.htm**
- Después de C (que puede ser A, B, D o la unidad de disco que quieras) fíjate que va una barra vertical (pipe), y que las barras que separan los directorios y ficheros son barras **a la derecha**.

[\[Indice\]](#)

Insertar imágenes

Insertar imágenes en un documento permite crear páginas mucho más atractivas. Según el tipo de gráficos utilizado se pueden conseguir efectos realmente sorprendentes.

Para insertar una imagen en un documento HTML se utiliza el elemento ****. Este elemento puede ir acompañado de los atributos **SRC ALT ISMAP ALIGN WIDTH HEIGHT BORDER VSPACE HSPACE**. Con las nuevas implementaciones que Netscape hace del elemento, se consiguen efectos de *imagen flotante* y por tanto se ha hecho necesario dotar al elemento **
** del atributo **CLEAR**.

Las imágenes utilizadas pueden estar en formato GIF o JPG. Estos formatos son interpretados directamente por el visualizador. Se puede usar cualquier otro formato como el MPG (vídeo), pero entonces el visualizador llamará a un programa auxiliar, que previamente le habrás definido, para que sea éste el que visualice el fichero. La desventaja de las imágenes en formatos no tratados por el propio visualizador es que no verás el texto junto con la imagen, y por tanto no podrás imprimir la página compuesta.

Algunos visualizadores no son capaces de tratar imágenes, o aunque el visualizador sí pueda, tal vez tu máquina tenga muy poca memoria o una tarjeta de vídeo floja. Entonces podemos recurrir al atributo **ALT** (por alternativo) para definir un texto que aparecerá en lugar de la imagen. Esto es importante cuando una imagen, además de ser un elemento decorativo, soporta un link.

Veamos ejemplos del uso de ****:

Imagen alineada a la izquierda (por defecto)

```
<IMG SRC="sugeren.gif">texto texto texto texto
texto texto texto texto
```



```
texto texto
texto texto texto
```

Imagen con un texto alternativo

Para verlo tienes de deshabilitar la opción de cargar imágenes de tu visualizador.

```
<IMG SRC="sugeren.gif" ALT="AQUI VA UNA IMAGEN">
```



Imagen alineada a la izquierda. Texto alineado arriba

```
<IMG SRC="sugeren.gif" ALIGN=TOP> texto texto texto texto texto
texto exto texto texto texto texto texto
```

Insertar imagen

texto texto texto texto texto texto texto texto texto textzo texto texto texto texto texto



texto texto texto

Imagen alineada a la izquierda. Texto alineado abajo

```
<IMG SRC="sugeren.gif" ALIGN=BOTTOM> texto texto texto texto texto
texto exto texto texto texto texto texto
```



texto textzo texto texto texto texto texto
texto texto texto

Imagen alineada a la izquierda. Texto alineado al centro

```
<IMG SRC="sugeren.gif" ALIGN=MIDDLE> texto texto texto texto texto
texto exto texto texto texto texto texto
```



texto textzo texto texto texto texto texto
texto texto texto

Imagen alineada a la izquierda. Texto alineado a la derecha de la imagen

```
<IMG SRC="sugeren.gif" ALIGN=LEFT> texto texto texto texto texto
texto texto texto texto texto texto texto
```



texto textzo texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto texto texto texto

Imagen alineada a la izquierda. Texto envolviendo la imagen

```
<IMG SRC="sugeren.gif" ALIGN=LEFT> texto texto texto texto texto <BR CLEAR>
texto texto texto texto texto texto texto
```



texto textzo texto texto texto texto texto
texto texto texto texto texto texto texto
texto texto texto texto texto texto

Imagen alineada a la derecha. Texto alineado a la izquierda de la imagen

```
<IMG SRC="sugeren.gif" ALIGN=RIGHT> texto texto texto texto texto
texto texto texto texto texto texto texto
```

texto textzo texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto texto texto texto



Si tu visualizador no soporta esto, hay un truco para conseguir algo parecido:

```
<TABLE> <TR><TD>texto texto texto</TD><TD
align=center><IMG SRC="sugeren.gif"></TD></TR> </TABLE>
```

texto texto



Imagen redimensionada a más. Texto alineado a la derecha de la imagen

`` texto texto



texto textzo texto texto.

Aquí la imagen es la misma de los ejemplos anteriores, pero ha sido redimensionada por el visualizador. Su tamaño original es de 136x46 puntos. Este es su aspecto al doble: 272x92.

Imagen redimensionada a menos. Texto alineado a la derecha de la imagen

`` texto texto



texto textzo texto texto.

Aquí la imagen es la misma de los ejemplos anteriores, pero ha sido redimensionada por el visualizador. Su tamaño original es de 136x46 puntos. Este es su aspecto a la mitad: 75x23

Imagen alineada a la izquierda con marco. Texto alineado a la derecha de la imagen

`` texto texto



texto textzo texto texto.

Imagen alineada a la izquierda. Texto alineado a la derecha de la imagen.

Espacios verticales y horizontales forzados.

texto texto texto texto texto texto textzo texto
` ALIGN=LEFT HSPACE=100 VSPACE=30>` texto texto

texto texto texto texto texto texto textzo texto texto

texto texto texto texto texto texto texto texto texto texto textzo texto texto.



Insertar imagen

texto texto

Usar una imagen como punto de montaje de un link

texto texto texto texto.



Como puedes ver, la imagen aparece rodeada de un marco de color como es habitual en los puntos de link (en el texto es el subrayado). Si quieres eliminar el marco, escribe:

```
.... IMG SRC="sugeren.gif" BORDER=0 >
```

Ahora que ya sabes cómo manejar las imágenes, veamos algunos efectos especiales algo más complicados.

Mapeado de una imagen para usarla como direccionador

```
<A HREF="/cgi-bin/imagemap/mimapa"><IMG SRC="sugeren.gif" ISMAP></A>
```



Si mueves el ratón por encima de la imagen, verás que en la ventanita de abajo del visualizador (la que hay junto a una llave) aparecen junto a un interrogante unos números que cambian según muevas el ratón. Con cuidado intenta poner el puntero en la esquina superior izquierda y en la inferior derecha. Verás las cifras 1,1 y 136,46, respectivamente, que son las medidas en puntos que tiene el gráfico. Como ya habrás supuesto, esto es un sistema de coordenadas, y para que nuestra imagen funcione como un menú direccionador, lo que necesitamos es que algún elemento convierta esas coordenadas en una dirección del estilo ya conocido: "http://miservidor.es/mifichero.htm". Esto significa que con un solo link podemos obtener un enorme número de direcciones, teóricamente tantas como pares de coordenadas tiene la imagen. En la práctica serán menos, ya que andar intentando atinar en el punto 11,32, por ejemplo, es un poco engorroso.

Existen multitud de programas llamados mapeadores que parten la imagen en imaginarios trozos, y cada uno de ellos indica un link con alguna parte. Estos programas generan un fichero con todas las direcciones. Algo así:

```
rect http://miservidor.es/mifichero1.htm 12,35  
rect http://miservidor.es/mifichero2.htm 90,42  
rect http://miservidor.es/mifichero3.htm 112,46
```

Este podría ser el aspecto del fichero del ejemplo, el llamado **mimapa**. El parámetro **rect** significa rectángulo, y nos indica que la imagen ha sido troceada en rectángulos y las coordenadas indican la esquina superior izquierda y la inferior derecha. También podría ser en círculos (**circle**) o polígonos (**poly**). Ahora veamos el resto de instrucciones del ejemplo:

```
<A HREF="/cgi-bin/imagemap/mimapa"><IMG SRC="sugeren.gif" ISMAP></A>
```

La primera parte **A HREF** ya la conocemos, es un link. Después viene un directorio: **cgi-bin**. Este es un directorio especial que tienen los servidores http, donde se ejecutan los programas auxiliares del servidor, y al que normalmente sólo puede acceder el administrador del sistema. En ese directorio hay un fichero llamado **imagemap** que es un programa que se encarga de leer tu fichero **mimapa**, capturar las coordenadas que señale el

Insertar imagen

puntero del ratón y servir la dirección que tiene asociada, con lo que acabamos obteniendo un link normal. Ya por último, aparece **ISMAP** que es el que convierte una simple imagen en una potente herramienta.

Sólo queda por decir que el fichero **mimapa** tiene que estar en otro directorio especial que tiene el servidor a estos efectos, y al que por supuesto, solo puede acceder el administrador del sistema.

Todo esto es suponiendo que el servidor está en una máquina UNIX. Si reside en tu propio PC con Windows o en un MAC (que los hay), tendrás que ver las instrucciones concretas del programa servidor. La forma de montar el mapa puede variar de un programa servidor a otro, pero la base de funcionamiento es siempre la misma.

Otra forma alternativa de conseguir que una imagen responda con ciertas acciones, es operar con los elementos de [formularios](#). En el índice encontrarás información sobre un tipo de formulario que devuelve un par de coordenadas, que una vez evaluadas permite ejecutar una acción.

Pero, una vez más, Netscape ha propuesto una solución mucho más simple: que sea el propio visualizador quien haga las veces de programa conversor. En efecto, un recurso tan bueno como los mapas, no debe depender de tener tu máquina en red y de que haya un servidor http que te atienda. Además, eso resta portabilidad a los documentos locales y a los trabajos personales en disquete. Para ello ha implementado el elemento **MAP** que acompañará a **IMG**.

Por ejemplo, en la siguiente imagen, la mitad izquierda nos envía a [indice.htm](#) y la mitad derecha a [intro.htm](#). Mueve el puntero del ratón horizontalmente sobre la imagen y observa la ventanita que hay en la parte baja de la pantalla, verás como cambia el nombre del link.



Se escribe así:

```
<MAP NAME="nombre1">
  <AREA SHAPE="rect" COORDS="1,1,75,46" HREF="indice.htm">
  <AREA SHAPE="rect" COORDS="76,1,136,46" HREF="intro.htm">
</MAP>
```

```
<IMG SRC="sugeren.gif" USEMAP="#nombre1">
```

Aunque se comprende a simple vista, analicémoslo:

En primer lugar tenemos el elemento **MAP**, que lo que está haciendo es definir un mapa de coordenadas. Va acompañado del atributo **NAME** que da nombre al mapa. Es necesario nombrarlo porque podría haber más de uno en la misma página, y evidentemente, sus nombres no deberán repetirse.

A continuación tenemos el atributo **AREA** que define las áreas de la imagen. El parámetro **SHAPE="rect"** indica la figura geométrica que estamos utilizando para ello. Al igual que con los otros mapas, puede ser **rect**, **circle** y **poly**.

El parámetro **COORDS** fácilmente se adivina que indica las coordenadas del área, en este caso vértice superior izquierdo e inferior derecho, respectivamente.

HREF ya sabemos lo que hace: indica un link con una página, pero esta vez no va acompañando al elemento

Insertar imagen

<A>, digamos que es un atributo prestado. Hay un área por cada link definido. Si un área no queremos que tenga link se definirá con **NOHREF**.

IMG SRC también son conocidos: hacen que se visualice la imagen.

Y por fin, **USEMAP** nos dice qué mapa de coordenadas hay que aplicar a la imagen. A una misma imagen se le pueden aplicar distintos mapas si se desea. Fíjate en que **nombre1**, (el nombre del mapa) va **precedido** del símbolo #. Esto se debe a que, en este caso, el mapa al que se hace referencia está en la misma página que la instrucción IMG. Podría estar en otra, y en ese caso el símbolo tendría que estar **después** del nombre de la página. Por ejemplo: **otrapagina.htm#nombre1**

[\[Indice\]](#)

[\[Indice\]](#)

Imágenes de fondo

El fondo (background) de este documento es una imagen en formato GIF.

Es un atributo del elemento <BODY>

Se escribe así:

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

Y es compatible con todos los demás atributos de este elemento.

[\[Indice\]](#)

Algunos ejemplos de imágenes en listas

Se escriben todas igual, sólo hay que cambiar el nombre de los gráficos:

```
<DL>  
  <DT><IMG SRC="bolaroja.gif">Linea de texto  
  <DT><IMG SRC="bolaroja.gif">Linea de texto  
  <DT><IMG SRC="bolaroja.gif">Linea de texto  
</DL>
```

-  Linea de texto
 -  Linea de texto
 -  Linea de texto
 -  Linea de texto
 -  Linea de texto
 -  Linea de texto
 -  Linea de texto
 -  Linea de texto
 -  Linea de texto
 -  Linea de texto
 -  Linea de texto
 -  Linea de texto
-

-  **Linea de texto**
-  **Linea de texto**
-  **Linea de texto**
-  **Linea de texto**
-  **Linea de texto**
-  **Linea de texto**

 **Linea de texto**

 **Linea de texto**

 **Linea de texto**

 **Linea de texto**

 Linea de texto

 Linea de texto

 Linea de texto

 Linea de texto

 Linea de texto

 Linea de texto

 Linea de texto

 Linea de texto

[\[Indice\]](#)

Introducción a los formularios

Generalidades

Esta es la parte más técnica y compleja del HTML. Los formularios o *forms* en inglés, son unas páginas especiales que se utilizan para realizar transacciones. En una página que contenga un formulario pueden además existir todos los elementos hasta ahora conocidos, incluso el propio formulario puede estar integrado dentro de otros elementos, como por ejemplo una tabla. Un fallo que de momento tienen los formularios es que no se pueden imprimir. Sólo aparece el texto que los acompaña, pero no sus componentes, como ventanas de texto o botones.

¿Cómo funcionan los formularios?

El formulario no es más que una página escrita en HTML como cualquier otra. Lo que la hace diferente es que permite capturar datos de varias maneras: directamente desde el teclado, escogiendo un valor de una lista desplegable o seleccionando una opción desde botones fijos o de todas ellas combinadas.

Una vez terminada la captura de datos, estos forman un registro que es enviado a un servidor http que lo procesará y devolverá una respuesta. También pueden crearse formularios que simplemente abren un página en local.

Hasta aquí todo se reduce a escribir HTML, pero la parte complicada viene cuando los datos capturados por tu formulario llegan al servidor. Allí tiene que haber un programa especial, creado especialmente para ese formulario, que reciba esos datos y sepa qué hacer con ellos, procesarlos y confeccionar la respuesta que remitirá al visualizador. A esa respuesta se le denomina *documento virtual* ya que esa página no está escrita en ninguna parte; es generada, enviada y destruida.

A los programas que hacen de puente entre el formulario y el servidor, se les llama programas **CGI** (Common Gateway Interface).

Los programas CGI pueden estar escritos en cualquier lenguaje de programación que sea soportado por el sistema operativo del servidor, y pueden estar diseñados para cualquier función: desde una simple captura de datos que serán guardados en un fichero hasta la más sofisticada consulta a una base de datos.

¿Cómo es un CGI?

Aunque el propósito de esta guía no es tratar temas de programación, como breve orientación se muestra un sencillo programa hecho con comandos shell de UNIX. Este CGI está instalado en el Servidor http del Servicio de Estudiantes, y puede utilizarse como prueba, se llama **test2-cgi** y está en **/cgi-bin**. Muestra los valores de las variables más usuales, así como el nombre de los campos definidos en el formulario y contenido de cada uno de ellos.

Ejemplo de programa CGI (shell UNIX)

```
#!/bin/sh

echo Content-type: text/plain
echo

echo Test CGI. Se visualiza el contenido de las variables mas usuales
echo
echo numero de argumentos: $#. valor argumentos: "$*".
echo

echo SERVER_SOFTWARE      = $SERVER_SOFTWARE
echo SERVER_NAME          = $SERVER_NAME
echo GATEWAY_INTERFACE    = $GATEWAY_INTERFACE
echo SERVER_PROTOCOL      = $SERVER_PROTOCOL
echo SERVER_PORT          = $SERVER_PORT
echo REQUEST_METHOD       = $REQUEST_METHOD
echo HTTP_ACCEPT          = $HTTP_ACCEPT
echo PATH_INFO            = $PATH_INFO
echo PATH_TRANSLATED      = $PATH_TRANSLATED
echo SCRIPT_NAME          = $SCRIPT_NAME
echo QUERY_STRING         = $QUERY_STRING
echo REMOTE_HOST          = $REMOTE_HOST
echo REMOTE_ADDR          = $REMOTE_ADDR
echo REMOTE_USER          = $REMOTE_USER
echo REMOTE_IDENT         = $REMOTE_IDENT
echo AUTH_TYPE            = $AUTH_TYPE
echo CONTENT_TYPE         = $CONTENT_TYPE
echo CONTENT_LENGTH      = $CONTENT_LENGTH
echo HTTP_USER_AGENT      = $HTTP_USER_AGENT
echo HOME                 = $HOME
echo HTTP_FROM            = $HTTP_FROM
echo HTTP_REFERER         = $HTTP_REFERER
echo REFERER_URL         = $REFERER_URL

if [ "$CONTENT_TYPE" = "application/x-www-form-urlencoded" ]; then

read lo_que_viene
echo
echo ESTO VIENE POR LA INPUT:
echo
echo $lo_que_viene

exit 0
else
```

```
echo NO VIENE NADA POR LA INPUT
exit 0
fi
```

Esta sería la respuesta del CGI **test2-cgi** a un imaginario formulario llamado **prueba.htm** con dos campos de captura por teclado llamados **CAMPO1** y **CAMPO2** en los que hemos introducido: "**Esto es una**" y "**demonstracion de formularios**" respectivamente.

Ejemplo de respuesta del programa test2-cgi al formulario prueba.htm

Test CGI. Se visualiza el contenido de las variables mas usuales

numero de argumentos: 0. valor argumentos: .

```
SERVER_SOFTWARE = NCSA/1.3
SERVER_NAME = mecaniza.sestud.uv.es
GATEWAY_INTERFACE = CGI/1.1
SERVER_PROTOCOL = HTTP/1.0
SERVER_PORT = 80
REQUEST_METHOD = POST
HTTP_ACCEPT = image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
PATH_INFO =
PATH_TRANSLATED =
SCRIPT_NAME = /cgi-bin/test2-cgi
QUERY_STRING =
REMOTE_HOST = mimaquina.misubdominio.midominio.mipais
REMOTE_ADDR = 255.255.255.255
REMOTE_USER =
REMOTE_IDENT = unknown
AUTH_TYPE =
CONTENT_TYPE = application/x-www-form-urlencoded
CONTENT_LENGTH = 52
HTTP_USER_AGENT = Mozilla/2.02 (Win95; I)
HOME =
HTTP_FROM =
HTTP_REFERER = http://mecaniza/manual.esp/prueba.htm
REFERER_URL =
```

ESTO VIENE POR LA **INPUT**:

CAMPO1=Esto+es+una&**CAMPO2**=demonstracion+de+formularios

Fíjate que las palabras aparecen separadas por el signo + y no por espacios como hemos escrito. Si además hubiéramos escrito algún carácter especial, no se vería como tal, sino su valor en hexadecimal precedido del carácter %. Por ejemplo, el carácter & es transferido como %26. Esta conversión se hace

para no confundirlo con el separador de campos, que como puedes ver es el mismo símbolo. El signo = después del nombre del campo, también lo pone el visualizador.

Todo esto y algunas cosas más (que habrá que ver en la documentación de referencia) han de tenerse presentes a la hora de realizar programas CGI. Para terminarlo de complicar, no todos los visualizadores utilizan la misma forma de enviar los datos recibidos, por lo que se plantea a menudo la necesidad de decidir para qué visualizador queremos programar nuestro WEB.

Los programas de test, como el del ejemplo, son muy útiles cuando se están diseñando formularios, ya que permiten comprobar de forma real su funcionamiento y analizar el registro que envían, cosa imprescindible para poder programar el correspondiente CGI. Intenta hacer uno a tu medida, o simplemente copia éste, instálalo en tu servidor http, (si no tienes acceso, pide a tu administrador de sistema que lo haga), empieza a probar... y buena suerte.

[\[Indice\]](#)

¿Cómo se escriben los formularios ?

La base del formulario es el elemento lleno **<FORM>**, es el que define una zona de la página como formulario. En una página puede haber varias zonas definidas como formulario. Dentro de este elemento a su vez se utilizan otros elementos, que son los que realmente dibujan en pantalla los componentes del formulario, son:

■ **<INPUT>**

- Campos de entrada por teclado.
- Botones de selección.
- Casillas de marca.
- Botones de proceso.
- Botones de inicialización (reset).
- Imágenes sensibles al ratón.

■ **<SELECT>**

- Listas desplegables de valores.
- Listas fijas de valores.

■ **<TEXTAREA>**

- Ventanas de escritura libre.

Este sería el más elemental de los formularios, con sólo un campo y un botón de envío:

Analicemos cómo se escribe:

```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">  
<INPUT NAME="campo1" >  
<INPUT TYPE="submit" VALUE="Procesar"></FORM>
```

En la primera línea vemos el elemento de definición de formulario: **<FORM>**. Va acompañado de dos atributos: **METHOD** y **ACTION**.

METHOD se refiere al método que emplearemos para enviar los datos al servidor, y pueden ser dos: **POST** y **GET**. El optar por uno u otro obedece a complejas cuestiones de programación basadas en la máxima longitud de registro que puede enviarse; cada uno usa un canal de salida distinto. El que soporta más longitud es **POST**, y será el que utilizemos en casi todos los ejemplos.

Echale un vistazo a la respuesta del programa [test2-cgi](#) de la página anterior. Fíjate en las variables **QUERY_STRING** e **INPUT**. Por la primera llegarán los datos cuando se utilice **METHOD=GET** y por la segunda cuando es **POST** el método elegido.

ACTION se refiere a la acción que queremos que ejecute el formulario en un servidor http o en local. Con el valor del ejemplo **ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi"** se le está indicando que ejecute un programa llamado **test2-cgi** que está en el directorio **/cgi-bin** del servidor http **mecaniza.sestud.uv.es** cuya

Como se escriben los formularios

respuesta será similar a la que hemos visto en la página de [introducción a los formularios](#).

Además de enviar datos a un servidor, **ACTION** también puede realizar una acción en local, por ejemplo traer una página:

Se escribe:

```
<FORM METHOD="POST" ACTION="indice.htm">
  .....
</FORM>
```

En este caso funcionaría igual que un link.

Otra cosa que puede hacer **ACTION**, y que además tiene la ventaja de que no sería preciso diseñar un CGI, es enviar un e-mail (correo electrónico) a un usuario o a una lista de usuarios. Los datos enviados por el formulario, tendrán la forma que ya conocemos, pero una vez recibidos por esta vía, pueden ser tratados con programas de edición de texto, y convertidos al formato que se quiera. Por ejemplo con Word y las macros de conversión para [Word 6.0](#) y [Word 97](#) que hay en esta guía.

Se escribe:

```
<FORM METHOD="POST" ACTION="mailto:mi-usuario@miservidor-e-mail">
  .....
</FORM>
```

Si no quieres complicarte la vida con descodificaciones (depende de lo que se vaya a hacer con los datos recibidos), el navegador puede remitirlos con un formato más sencillo y ya descodificado. Para ello sólo hay que añadir el parámetro **ENCTYPE** con el valor **TEXT/PLAIN**. Se escribe así:

```
<FORM METHOD="POST" ACTION="mailto:mi-usuario@miservidor-e-mail"
  ENCTYPE="TEXT/PLAIN">
  .....
</FORM>
```

Resumiendo: sin poner ningún parámetro al elemento **ACTION** los datos te llegarán así:

CAMPO1=Esto+es+una&CAMPO2=demostracion+de+formularios

y poniendo **ENCTYPE="TEXT/PLAIN"** así:

CAMPO1=Esto es una
CAMPO2=demostracion de formularios

Después tenemos

El elemento INPUT

INPUT sin ningún atributo define por defecto una ventana de escritura de 20 caracteres de longitud por una línea de ancho:

Se escribe:

Como se escriben los formularios

```
<FORM> <INPUT> </FORM>
```

<INPUT> admite varios atributos:

SIZE define la longitud de la ventana de texto.

MAXLENGTH define la máxima longitud de la cadena que se puede escribir dentro de la ventana.

NAME define el nombre de la ventana (que en realidad es un campo dentro del registro, que en conjunto, es el formulario). Si escribes:

```
<FORM> <INPUT SIZE=10 MAXLENGTH=10 NAME="campo1"></FORM>
```

Como puedes comprobar, sólo se pueden escribir 10 caracteres dentro de la ventana, que también es de longitud 10. No es obligatorio que concidan ambos valores, puedes definir la ventana al valor que quieras y la longitud de la cadena puede ser mayor o menor.

VALUE sirve para que la ventana aparezca con un valor predeterminado, y no vacía como hace por defecto:

Se escribe:

```
<FORM> <INPUT VALUE="HOLA"> </FORM>
```

TYPE=TEXT Es el valor por defecto de TYPE. Puede omitirse.

TYPE=PASSWORD para que el valor predeterminado de la ventana esté en formato oculto:

Se escribe:

```
<FORM> <INPUT TYPE="PASSWORD" VALUE="HOLA"> </FORM>
```

TYPE=HIDDEN define que el valor predeterminado de la ventana y la ventana misma estén en formato oculto:

Se escribe:

```
<FORM> <INPUT TYPE="HIDDEN" VALUE="HOLA"> </FORM>
```

Esta opción se utiliza cuando es necesario enviar un valor fijo al servidor, pero no se desea que el usuario intervenga o que lo vea. CUIDADO, esto no es un formato encriptado, viendo el documento fuente se puede saber el valor de la ventana.

TYPE=SUBMIT para generar un botón que al ser pulsado cierra la captura de datos del formulario y procede a ejecutar lo definido en el atributo ACTION que ya conocemos.

Se escribe:

```
<FORM METHOD="POST" ACTION="http://mcaniza.sestud.uv.es/cgi-bin/test2-cgi">
```

Como se escriben los formularios

```
<INPUT NAME="campo1" >  
<INPUT TYPE="SUBMIT" VALUE="Procesar"></FORM>
```

VALUE puede acompañar a **SUBMIT**, y en este caso sirve para definir el texto que queremos que tenga el botón en su interior. Si se omite, por defecto el visualizador le da el valor "Submit Query".

TYPE=RESET para generar un botón que al ser pulsado inicializa todos los componentes del formulario.

Se escribe:

```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">  
<INPUT NAME="campo1" >  
<INPUT TYPE="RESET" VALUE="Inicializar"></FORM>
```

VALUE puede acompañar a **RESET**, y en este caso sirve para definir el texto que queremos que tenga el botón en su interior. Si se omite, por defecto el visualizador le da el valor "Reset".

TYPE=FILE Este es un nuevo tipo que sólo funciona con Netscape 3.0 o superior, y sirve para enviar un fichero a una máquina remota (hacer FTP), pero no funciona solo, necesita de ciertos acompañantes que hacen su uso un poco más complicado:

Aquí hay que echar mano de un atributo parametrizado de <FORM>: **ENCTYPE="multipart/form-data"** para generar un botón que permitirá buscar un fichero en nuestra máquina, ponerlo en una ventanita de entrada como las ya conocidas (por tanto también se puede escribir directamente el nombre del fichero en lugar de buscarlo) y junto con un botón tipo submit enviar el fichero.

Enviar el fichero:

Se escribe:

```
<FORM enctype="multipart/form-data"  
ACTION="ftp://miservidor/" METHOD="POST">  
Enviar el fichero:  
<INPUT NAME="mifichero" TYPE="FILE">  
<INPUT TYPE="SUBMIT" VALUE="Enviar fichero"></FORM>
```

el botón generado aparece con el texto "Browse.." y no sirve de nada añadirle el parámetro **VALUE** para darle otro nombre, como se puede hacer con el botón de tipo submit.

En este ejemplo se haría una conexión al servicio FTP general de la máquina remota, pero se puede hacer a un directorio en particular de un usuario concreto. Sólo habría que cambiar la línea

```
ACTION="ftp://miservidor/" METHOD="POST">
```

y escribir:

```
ACTION="ftp://miusuario@miservidor/" METHOD="POST">
```

Otra forma más directa de hacer un FTP en las dos direcciones, es desde la ventana "Location" del visualizador. La sintaxis de los parámetros de **ACTION** son válidos en esa ventana.

Por supuesto, todo esto si no tienes cuenta y permiso de escritura en la máquina remota, no sirve para nada...

Como ya se ha dicho antes, el atributo **ACTION** es capaz de enviar el contenido del formulario por correo

Como se escriben los formularios

electrónico, pero no sólo es capaz de enviar el formulario: también puede adjuntar al e-mail un fichero cualquiera. Lo Único que hay que hacer es añadirle un instrucción como la que acabamos de ver para el FTP. Así:

```
<FORM enctype="multipart/form-data"
```

```
ACTION="mailto:mi-usuario@miservidor-e-mail/" METHOD="POST">
```

Adjuntar el fichero:

```
<INPUT NAME="mifichero" TYPE="FILE">
```

```
<INPUT TYPE="SUBMIT" VALUE="Adjuntar fichero"></FORM>
```

TYPE=IMAGE hace que el visualizador presente una imagen que es sensible al ratón. Lo que el formulario envía al servidor es un registro cuyos campos son el nombre definido para la imagen seguido de los parámetros **.x=n .y=n** donde **n** son los números de las coordenadas **x** y del punto en el que estaba el ratón en el momento del envío. Para presentar la imagen se utiliza el atributo **SRC** del elemento **IMG**. Este sería un ejemplo de respuesta:

```
imagen.x=99&imagen.y=15
```

Prueba a pulsar en cualquier parte de esta imagen:



Se escribe:

```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">
```

```
<INPUT TYPE="IMAGE" NAME="imagen" SRC="sugeren.gif">
```

```
</FORM>
```

Los formularios que utilicen este tipo de recursos pueden prescindir del tipo **SUBMIT**, ya que el formulario se cierra con la imagen.

TYPE=RADIO genera botones circulares que permiten seleccionar un valor predeterminado o activar una opción

Clase A Clase B

Se escribe:

```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">
```

```
<INPUT TYPE="SUBMIT" VALUE="Procesar">
```

```
<INPUT TYPE="RESET" VALUE="Inicializar">
```

Clase A

```
<INPUT TYPE="RADIO" NAME="clase" VALUE="A">
```

Clase B

```
<INPUT TYPE="RADIO" NAME="clase" VALUE="B"></form>
```

En este ejemplo, los botones permiten seleccionar la clase A, la clase B o ninguna de las dos. Si se desea que por defecto un valor esté seleccionado, por ejemplo clase B, hay que añadir el parámetro **CHECKED** después del valor:

```
... VALUE="B" CHECKED></form>
```

TYPE=CHECKBOX genera casillas de marca que permiten seleccionar un valor predeterminado o activar una

Como se escriben los formularios

opción

Clase A Clase B

Se escribe:

```
<FORM METHOD="POST" ACTION="http://mcaniza.sestud.uv.es/cgi-bin/test2-cgi">
  <INPUT TYPE="SUBMIT" VALUE="Procesar">
  <INPUT TYPE="RESET" VALUE="Inicializar">
Clase A
  <INPUT TYPE="CHECKBOX" NAME="clase" VALUE="A">
Clase B
  <INPUT TYPE="CHECKBOX" NAME="clase" VALUE="B"></form>
```

En este ejemplo, las casillas de marca permiten seleccionar la clase A, la clase B, ninguna, o las dos. Si se desea que por defecto un valor esté seleccionado, por ejemplo clase B, hay que añadir el parámetro **CHECKED** después del valor:

```
... VALUE="B" CHECKED></form>
```

El elemento SELECT

SELECT sin ningún atributo define por defecto una lista desplegable de toda la pantalla de alto y una columna de ancho:

Se escribe:

```
<FORM> <SELECT> </SELECT> </FORM>
```

Como evidentemente esto no es muy práctico, hay que parametrizar el elemento. En primer lugar se le da nombre, ya que éste es un campo más del formulario, la diferencia es que no hay que escribir en él sino escoger un valor de los que nos muestre al desplegarlo, se usará el atributo **NAME**, y para entrar valores en la lista se usa el atributo **OPTION**.

Se escribe:

```
<FORM METHOD="POST" ACTION="http://mcaniza.sestud.uv.es/cgi-bin/test2-cgi">
  <INPUT TYPE="SUBMIT" VALUE="Procesar">
  <INPUT TYPE="RESET" VALUE="Inicializar">

  <SELECT NAME="lista1">
    <OPTION>Valor 1
    <OPTION>Valor 2
    <OPTION>Valor 3
  </SELECT>
</FORM>
```

Se deben tener en cuenta varias cosas respecto a **SELECT**:

Como se escriben los formularios

La longitud de la ventana de selección se autoajusta al valor más largo de la lista.

La ventana de selección, por defecto, muestra una línea, si se quieren mostrar más se utilizará el atributo **SIZE**.

Se debe procurar que los contenidos de los campos sean lo más cortos posibles; una excesiva longitud implica mayor tráfico por la red, pero se deben construir las listas de forma clara, y que resulten de fácil lectura al usuario, lo que puede aumentar la longitud de los valores. Para enviar un valor diferente del que aparece en la lista, se utiliza el parámetro **VALUE** asociado al atributo **OPTION**. Prueba con el valor 5 de la siguiente lista:



The screenshot shows a web form with two buttons labeled "Procesar" and "Inicializar". To the right of the buttons is a dropdown menu with a blue header and a white body. The menu is open, showing a list of options: "Valor 1", "Valor 2", "Valor 3", "Valor 4", "Este es el Valor 5", "Valor 6", and "Valor 7".

Se escribe:

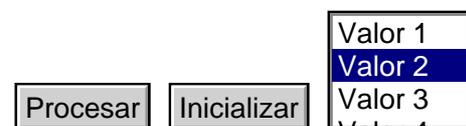
```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">
<INPUT TYPE="SUBMIT" VALUE="Procesar">
<INPUT TYPE="RESET" VALUE="Inicializar">

<SELECT NAME="lista1" SIZE=3>
<OPTION>Valor 1
<OPTION>Valor 2
<OPTION>Valor 3
<OPTION>Valor 4
<OPTION VALUE="Valor 5">Este es el Valor 5
<OPTION>Valor 6
<OPTION>Valor 7
</SELECT>
</FORM>
```

Esto significa que cuando seleccionas "Este es el valor 5" en realidad lo que se envía al servidor es "Valor 5".

Con las listas desplegables es obligatorio seleccionar siempre un valor, y si no hay ninguno seleccionado, por defecto el visualizador enviará el primer valor de la lista.

Se puede hacer que haya un valor preseleccionado añadiendo el parámetro **SELECTED** al atributo **OPTION**, y si se quiere permitir más de una selección, se pondrá el atributo **MULTIPLE** al elemento **SELECT**. Para seleccionar y deseleccionar hay que atenerse a las normas de Windows.



The screenshot shows a web form with two buttons labeled "Procesar" and "Inicializar". To the right of the buttons is a dropdown menu with a white header and a white body. The menu is open, showing a list of options: "Valor 1", "Valor 2", "Valor 3", and "Valor 4". The option "Valor 2" is highlighted with a blue background, indicating it is selected.

Se escribe:

```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">
<INPUT TYPE="SUBMIT" VALUE="Procesar">
<INPUT TYPE="RESET" VALUE="Inicializar">

<SELECT NAME="lista1" MULTIPLE SIZE=3>
<OPTION>Valor 1
<OPTION SELECTED>Valor 2
<OPTION>Valor 3
</SELECT>
```

Como se escriben los formularios

```
<OPTION>Valor 4
<OPTION>Valor 5
<OPTION>Valor 6
<OPTION>Valor 7
</SELECT>
</FORM>
```

El elemento TEXTAREA

Este elemento, como su nombre indica, permite definir un área de texto en la pantalla en la que podemos escribir cualquier cosa. Se debe escribir ya parametrizada en cuanto a dimensiones, con los atributos **ROWS** (líneas) y **COLS** (COLUMNS). No tiene otros atributos o posibles variantes. Por supuesto, para que sea operativo, deberá ir acompañado de los componentes necesarios para enviar e inicializar.



The image shows a web form with a large rectangular text area. To the left of the text area are two buttons: 'Procesar' and 'Inicializar'. The text area is empty.

Se escribe:

```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">
  <INPUT TYPE="SUBMIT" VALUE="Procesar">
  <INPUT TYPE="RESET" VALUE="Inicializar">

  <TEXTAREA NAME="texto1" ROWS=5 COLS=40 ></TEXTAREA>
</FORM>
```

Aunque no es habitual, dentro de un área de texto puede haber un contenido por defecto (que puede ser borrado o modificado por el usuario). Para ello simplemente se escribirá el texto entre **<TEXTAREA>** y **</TEXTAREA>**. Así:

```
<>TEXTAREA NAME="texto1" ROWS=5 COLS=40>
  Contenido del area de texto
</TEXTAREA>
```

El tratamiento de los datos recibidos en el servidor desde los formularios requiere conocer la tabla de conversión hexadecimal que utilizan la mayoría de visualizadores. Recuerda que las palabras en destino aparecen separadas por el signo +, y las letras acentuadas y otros caracteres, por valores hexadecimales precedidos del símbolo %

Se puede proceder a la descodificación en el programa CGI que recibe los datos, o bien definir macros en programas auxiliares que serán ejecutados después.

En el índice encontrarás la [tabla de conversiones](#) utilizada con el visualizador Netscape desde Windows y una [macro para Word 6 de Windows](#) que convierte los caracteres más habituales.

[Indice](#)

Ejemplos de formularios

Juego de botones direccionados a páginas LOCALES.

Esto sólo funciona abriendo la página como **file:///** no como servicio **http://**.

Indice	Notas	Cómo funciona?
--------	-------	----------------

```
<TABLE>
<TR>
  <TD>
    <FORM METHOD="POST" ACTION="indice.htm">
      <INPUT TYPE="SUBMIT" VALUE="Indice">
    </FORM>
  </TD>

  <TD>
    <FORM METHOD="POST" ACTION="notas.htm">
      <INPUT TYPE="SUBMIT" VALUE="Notas">
    </FORM>
  </TD>

  <TD>
    <FORM METHOD="POST" ACTION="como.htm">
      <INPUT TYPE="SUBMIT" VALUE="Cómo funciona?">
    </FORM>
  </TD>
</TR>
</TABLE>
```

Juego de botones direccionados a páginas REMOTAS.

Puede funcionar abriendo la página como servicio **http://** o como **file:///**. pero deberá utilizarse **METHOD=GET** en lugar de **METHOD=POST**.

Indice	Notas	Cómo funciona?
--------	-------	----------------

```
<TABLE>
<TR>
  <TD>
    <FORM METHOD="GET" ACTION="http://mecaniza.sestud.uv.es/manual.esp/indice.htm">
      <INPUT TYPE="SUBMIT" VALUE="Indice">
    </FORM>
  </TD>

  <TD>
    <FORM METHOD="GET" ACTION="http://mecaniza.sestud.uv.es/manual.esp/notas.htm">
      <INPUT TYPE="SUBMIT" VALUE="Notas">
    </FORM>
  </TD>
</TR>
</TABLE>
```

Ejemplos de formularios

</TD>

<TD>

```
<FORM METHOD="GET" ACTION="http://mecaniza.sestud.uv.es/manual.esp/como.htm">  
<INPUT TYPE="SUBMIT" VALUE="Cómo funciona?">  
</FORM>
```

</TD>

</TR>

</TABLE>

1 campo vacío. Longitud no limitada.

1 campo vacío. Longitud limitada a 10


```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">  
<INPUT NAME="campo1">  
<INPUT NAME="campo2" SIZE=10 MAXLENGTH=10>  
<P>  
<INPUT TYPE="submit" VALUE="Procesar">  
<INPUT TYPE="reset" VALUE="Inicializar"></FORM>
```

1 campo vacío. Longitud no limitada.

1 campo con contenido visible.

1 campo con contenido oculto


```
<form method="POST" action="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">  
<input name="campo1">  
<input name="campo2" VALUE="TEXTO VISIBLE">  
<input type="password" name="campo3" VALUE="TEXTO OCULTO">  
<P>  
<input type="submit" value="Procesar">  
<input type="reset" value="Inicializar"></form>
```

1 campo vacío. Longitud sin limitar.

1 campo vacío. Longitud limitada a 10.

2 botones de selección.

Clase A Clase B

```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">
```

Ejemplos de formularios

```
<INPUT NAME="campo1">
<INPUT NAME="campo2" SIZE=10 MAXLENGTH=10>
<P>
Clase A <INPUT TYPE="RADIO" NAME="clase" VALUE="A">
Clase B <INPUT TYPE="RADIO" NAME="clase" VALUE="B">
<P>
<INPUT TYPE="SUBMIT" VALUE="Procesar">
<INPUT TYPE="RESET" VALUE="Inicializar">

</form>
```

1 campo vacío. Longitud sin limitar.
1 campo vacío. Longitud limitada a 10.
2 botones selección
2 casillas selección

Clase A Clase B

Tipo 1 Tipo 2


```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">
<INPUT NAME="campo1">
<INPUT NAME="campo2" SIZE=10 MAXLENGTH=10>
<P>
Clase A <INPUT TYPE="RADIO" NAME="clase" VALUE="A">
Clase B <INPUT TYPE="RADIO" NAME="clase" VALUE="B">
<P>
Tipo 1 <INPUT TYPE="CHECKBOX" NAME="tipo" VALUE="1">
Tipo 2 <INPUT TYPE="CHECKBOX" NAME="tipo" VALUE="2">
<P>
<INPUT TYPE="SUBMIT" VALUE="Procesar">
<INPUT TYPE="RESET" VALUE="Inicializar">

</form>
```

1 lista desplegable de 3 valores. 1 visible


```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">
<INPUT TYPE="SUBMIT" VALUE="Procesar">
<INPUT TYPE="RESET" VALUE="Inicializar">

<SELECT NAME="lista1">
<OPTION>Valor 1
<OPTION>Valor 2
<OPTION>Valor 3
</SELECT>
```

</FORM>

1 lista desplegable de 5 valores. 3 visibles.



```
<FORM METHOD="POST" ACTION="http://mecniza.sestud.uv.es/cgi-bin/test2-cgi">  
  
<SELECT NAME="lista1" SIZE=3>  
  <OPTION>Valor 1  
  <OPTION>Valor 2  
  <OPTION>Valor 3  
  <OPTION>Valor 4  
  <OPTION>Valor 5  
</SELECT>  
<P>  
  <INPUT TYPE="SUBMIT" VALUE="Procesar">  
  <INPUT TYPE="RESET" VALUE="Inicializar">  
</FORM>
```

1 Ventana de texto de 5x40



```
<FORM METHOD="POST" ACTION="http://mecniza.sestud.uv.es/cgi-bin/test2-cgi">  
<TEXTAREA NAME="texto1" ROWS=5 COLS=40 ></TEXTAREA>  
<P>  
<INPUT TYPE="SUBMIT" VALUE="Procesar">  
<INPUT TYPE="RESET" VALUE="Inicializar">  
</FORM>
```

Un típico y complejo formulario

Nombre:	<input type="text"/>	F.Nac.:	<input type="text"/>	DNI:	<input type="text"/>
Apellidos:	<input type="text"/>				
Calle y número:	<input type="text"/>				
Código Postal:	<input type="text"/>	Ciudad:	<input type="text"/>		
Provincia:	<input type="text"/>	Teléfono:	<input type="text"/>		

Ejemplos de formularios

Opción:	<input type="text" value="OPCION 1"/> Escoja una opción
Comentarios personales:	<input type="text"/>
Pulse aquí:	<input type="button" value="Enviar datos"/> <input type="button" value="Borrar los datos"/>

```
<FORM ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi" METHOD=POST>
```

```
<CENTER>
```

```
<TABLE BORDER>
```

```
<TR>
```

```
<TD>Nombre:</TD>
```

```
<TD> <INPUT TYPE="text" NAME="nombre" SIZE=18 MAXLENGTH=18> F.Nac.:
```

```
<INPUT TYPE="text" NAME="edad" SIZE=8 MAXLENGTH=8> DNI:
```

```
<INPUT TYPE="text" NAME="dni" SIZE=8 MAXLENGTH=8></TD>
```

```
<TR>
```

```
<TD>Apellidos:</TD>
```

```
<TD> <INPUT TYPE="text" NAME="apellidos" SIZE=48 MAXLENGTH=48></TD>
```

```
<TR>
```

```
<TD>Calle y número:</TD>
```

```
<TD> <INPUT TYPE="text" NAME="domicilio" SIZE=48 MAXLENGTH=48></TD>
```

```
<TR>
```

```
<TD>Código Postal:</TD>
```

```
<TD> <INPUT TYPE="text" NAME="postal" SIZE=5 MAXLENGTH=5>Ciudad:
```

```
<INPUT TYPE="text" NAME="localidad" SIZE=36 MAXLENGTH=36></TD>
```

```
<TR>
```

```
<TD>Provincia: </TD>
```

```
<TD> <INPUT TYPE="text" NAME="provincia" SIZE=20 MAXLENGTH=20>
```

```
Teléfono:
```

```
<INPUT TYPE="text" NAME="telefono" SIZE=19 MAXLENGTH=19></TD>
```

```
<TR>
```

```
<TD>Opción:</TD>
```

```
<TD> <SELECT NAME="OPCION">
```

```
<OPTION>OPCION 1
```

```
<OPTION>OPCION 2
```

```
<OPTION>OPCION 3
```

```
<OPTION>OPCION 4
```

```
<OPTION>OPCION 5
```

```
</SELECT>Escoja una opción</TD>
```

```
<TR>
```

```
<TD>Comentarios<BR> personales:</TD>
```

Ejemplos de formularios

```
<TD> <TEXTAREA NAME="coment" ROWS=2 COLS=48></TEXTAREA></TD>
```

```
<TR>  
  <TD><B>Pulse aquí:</B></TD>  
  <TD ALIGN=CENTER>  
    <INPUT TYPE="submit" VALUE="Enviar datos ">  
    <INPUT TYPE="reset" VALUE="Borrar los datos"></TD>  
</TABLE>  
</CENTER>  
</FORM>
```

[\[Indice\]](#)

¿Cómo se escriben los formularios ?

La base del formulario es el elemento lleno **<FORM>**, es el que define una zona de la página como formulario. En una página puede haber varias zonas definidas como formulario. Dentro de este elemento a su vez se utilizan otros elementos, que son los que realmente dibujan en pantalla los componentes del formulario, son:

■ **<INPUT>**

- Campos de entrada por teclado.
- Botones de selección.
- Casillas de marca.
- Botones de proceso.
- Botones de inicialización (reset).
- Imágenes sensibles al ratón.

■ **<SELECT>**

- Listas desplegables de valores.
- Listas fijas de valores.

■ **<TEXTAREA>**

- Ventanas de escritura libre.

Este sería el más elemental de los formularios, con sólo un campo y un botón de envío:



A screenshot of a simple web form. It consists of a single text input field on the left and a submit button on the right. The button is rectangular with a light gray background and a thin border, containing the text "Procesar". The entire form is set against a light gray background.

Analicemos cómo se escribe:

```
<FORM METHOD="POST" ACTION="http://mcaniza.sestud.uv.es/cgi-bin/test2-cgi">  
<INPUT NAME="campo1">  
<INPUT TYPE="submit" VALUE="Procesar"></FORM>
```

En la primera línea vemos el elemento de definición de formulario: **<FORM>**. Va acompañado de dos atributos: **METHOD** y **ACTION**.

METHOD se refiere al método que emplearemos para enviar los datos al servidor, y pueden ser dos: **POST** y **GET**. El optar por uno u otro obedece a complejas cuestiones de programación basadas en la máxima longitud de registro que puede enviarse; cada uno usa un canal de salida distinto. El que soporta más longitud es **POST**, y será el que utilizemos en casi todos los ejemplos.

Echale un vistazo a la respuesta del programa [test2-cgi](#) de la página anterior. Fíjate en las variables **QUERY_STRING** e **INPUT**. Por la primera llegarán los datos cuando se utilice **METHOD=GET** y por la segunda cuando es **POST** el método elegido.

ACTION se refiere a la acción que queremos que ejecute el formulario en un servidor http o en local. Con el

Como se escriben los formularios

valor del ejemplo `ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi"` se le está indicando que ejecute un programa llamado `test2-cgi` que está en el directorio `/cgi-bin` del servidor `http://mecaniza.sestud.uv.es` cuya respuesta será similar a la que hemos visto en la página de [introducción a los formularios](#).

Además de enviar datos a un servidor, **ACTION** también puede realizar una acción en local, por ejemplo traer una página:

Se escribe:

```
<FORM METHOD="POST" ACTION="indice.htm">
    . . . . .
</FORM>
```

En este caso funcionaría igual que un link.

Otra cosa que puede hacer **ACTION**, y que además tiene la ventaja de que no sería preciso diseñar un CGI, es enviar un e-mail (correo electrónico) a un usuario o a una lista de usuarios. Los datos enviados por el formulario, tendrán la forma que ya conocemos, pero una vez recibidos por esta vía, pueden ser tratados con programas de edición de texto, y convertidos al formato que se quiera. Por ejemplo con Word y las macros de conversión para [Word 6.0](#) y [Word 97](#) que hay en esta guía.

Se escribe:

```
<FORM METHOD="POST" ACTION="mailto:mi-usuario@miservidor-e-mail">
    . . . . .
</FORM>
```

Si no quieres complicarte la vida con descodificaciones (depende de lo que se vaya a hacer con los datos recibidos), el navegador puede remitirlos con un formato más sencillo y ya descodificado. Para ello sólo hay que añadir el parámetro **ENCTYPE** con el valor **TEXT/PLAIN**. Se escribe así:

```
<FORM METHOD="POST" ACTION="mailto:mi-usuario@miservidor-e-mail"
  ENCTYPE="TEXT/PLAIN">
    . . . . .
</FORM>
```

Resumiendo: sin poner ningún parámetro al elemento **ACTION** los datos te llegarán así:

`CAMPO1=Esto+es+una&CAMPO2=demostracion+de+formularios`

y poniendo **ENCTYPE="TEXT/PLAIN"** así:

`CAMPO1=Esto es una`
`CAMPO2=demostracion de formularios`

Después tenemos

El elemento INPUT

INPUT sin ningún atributo define por defecto una ventana de escritura de 20 caracteres de longitud por una línea de ancho:

Como se escriben los formularios



Se escribe:

```
<FORM> <INPUT> </FORM>
```

<INPUT> admite varios atributos:

SIZE define la longitud de la ventana de texto.

MAXLENGTH define la máxima longitud de la cadena que se puede escribir dentro de la ventana.

NAME define el nombre de la ventana (que en realidad es un campo dentro del registro, que en conjunto, es el formulario). Si escribes:

```
<FORM> <INPUT SIZE=10 MAXLENGTH=10 NAME="campo1"></FORM>
```



Como puedes comprobar, sólo se pueden escribir 10 caracteres dentro de la ventana, que también es de longitud 10. No es obligatorio que concidan ambos valores, puedes definir la ventana al valor que quieras y la longitud de la cadena puede ser mayor o menor.

VALUE sirve para que la ventana aparezca con un valor predeterminado, y no vacía como hace por defecto:



Se escribe:

```
<FORM> <INPUT VALUE="HOLA"> </FORM>
```

TYPE=TEXT Es el valor por defecto de TYPE. Puede omitirse.

TYPE=PASSWORD para que el valor predeterminado de la ventana esté en formato oculto:



Se escribe:

```
<FORM> <INPUT TYPE="PASSWORD" VALUE="HOLA"> </FORM>
```

TYPE=HIDDEN define que el valor predeterminado de la ventana y la ventana misma estén en formato oculto:

Se escribe:

```
<FORM> <INPUT TYPE="HIDDEN" VALUE="HOLA"> </FORM>
```

Como se escriben los formularios

Esta opción se utiliza cuando es necesario enviar un valor fijo al servidor, pero no se desea que el usuario intervenga o que lo vea. CUIDADO, esto no es un formato encriptado, viendo el documento fuente se puede saber el valor de la ventana.

TYPE=SUBMIT para generar un botón que al ser pulsado cierra la captura de datos del formulario y procede a ejecutar lo definido en el atributo ACTION que ya conocemos.



Se escribe:

```
<FORM METHOD="POST" ACTION="http://mecniza.sestud.uv.es/cgi-bin/test2-cgi">
  <INPUT NAME="campo1">
  <INPUT TYPE="SUBMIT" VALUE="Procesar"></FORM>
```

VALUE puede acompañar a **SUBMIT**, y en este caso sirve para definir el texto que queremos que tenga el botón en su interior. Si se omite, por defecto el visualizador le da el valor "Submit Query".

TYPE=RESET para generar un botón que al ser pulsado inicializa todos los componentes del formulario.



Se escribe:

```
<FORM METHOD="POST" ACTION="http://mecniza.sestud.uv.es/cgi-bin/test2-cgi">
  <INPUT NAME="campo1">
  <INPUT TYPE="RESET" VALUE="Inicializar"></FORM>
```

VALUE puede acompañar a **RESET**, y en este caso sirve para definir el texto que queremos que tenga el botón en su interior. Si se omite, por defecto el visualizador le da el valor "Reset".

TYPE=FILE

NUEVO Este es un nuevo tipo que sólo funciona con Netscape 3.0 o superior, y sirve para enviar un fichero a una máquina remota (hacer FTP), pero no funciona solo, necesita de ciertos acompañantes que hacen su uso un poco más complicado:

Aquí hay que echar mano de un atributo parametrizado de <FORM>: **ENCTYPE="multipart/form-data"** para generar un botón que permitirá buscar un fichero en nuestra máquina, ponerlo en una ventanita de entrada como las ya conocidas (por tanto también se puede escribir directamente el nombre del fichero en lugar de buscarlo) y junto con un botón tipo submit enviar el fichero.



Se escribe:

```
<FORM enctype="multipart/form-data"
  ACTION="ftp://miservidor/" METHOD="POST">
Enviar el fichero:
<INPUT NAME="mifichero" TYPE="FILE">
```

Como se escriben los formularios

```
<INPUT TYPE="SUBMIT" VALUE="Enviar fichero"></FORM>
```

el botón generado aparece con el texto "Browse.." y no sirve de nada añadirle el parámetro VALUE para darle otro nombre, como se puede hacer con el botón de tipo submit.

En este ejemplo se haría una conexión al servicio FTP general de la máquina remota, pero se puede hacer a un directorio en particular de un usuario concreto. Sólo habría que cambiar la línea

```
ACTION="ftp://miservidor/" METHOD="POST">
```

y escribir:

```
ACTION="ftp://miusuario@miservidor/" METHOD="POST">
```

Otra forma más directa de hacer un FTP en las dos direcciones, es desde la ventana "Location" del visualizador. La sintaxis de los parámetros de ACTION son válidos en esa ventana.

Por supuesto, todo esto si no tienes cuenta y permiso de escritura en la máquina remota, no sirve para nada...

Como ya se ha dicho antes, el atributo **ACTION** es capaz de enviar el contenido del formulario por correo electrónico, pero no sólo es capaz de enviar el formulario: también puede adjuntar al e-mail un fichero cualquiera. Lo Único que hay que hacer es añadirle un instrucción como la que acabamos de ver para el FTP. Así:

```
<FORM enctype="multipart/form-data"
```

```
ACTION="mailto:mi-usuario@miservidor-e-mail/" METHOD="POST">
```

Adjuntar el fichero:

```
<INPUT NAME="mifichero" TYPE="FILE">
```

```
<INPUT TYPE="SUBMIT" VALUE="Adjuntar fichero"></FORM>
```

TYPE=IMAGE hace que el visualizador presente una imagen que es sensible al ratón. Lo que el formulario envía al servidor es un registro cuyos campos son el nombre definido para la imagen seguido de los parámetros **.x=n .y=n** donde **n** son los números de las coordenadas **x** y del punto en el que estaba el ratón en el momento del envío. Para presentar la imagen se utiliza el atributo **SRC** del elemento **IMG**. Este sería un ejemplo de respuesta:

```
imagen.x=99&imagen.y=15
```

Prueba a pulsar en cualquier parte de esta imagen:



Se escribe:

```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">
```

```
<INPUT TYPE=IMAGE NAME="imagen" SRC="sugeren.gif">
```

```
</FORM>
```

Los formularios que utilicen este tipo de recursos pueden prescindir del tipo **SUBMIT**, ya que el formulario se cierra con la imagen.

TYPE=RADIO genera botones circulares que permiten seleccionar un valor predeterminado o activar una opción

Como se escriben los formularios



Se escribe:

```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">  
<INPUT TYPE="SUBMIT" VALUE="Procesar">  
<INPUT TYPE="RESET" VALUE="Inicializar">  
Clase A  
<INPUT TYPE="RADIO" NAME="clase" VALUE="A">  
Clase B  
<INPUT TYPE="RADIO" NAME="clase" VALUE="B"></form>
```

En este ejemplo, los botones permiten seleccionar la clase A, la clase B o ninguna de las dos. Si se desea que por defecto un valor esté seleccionado, por ejemplo clase B, hay que añadir el parámetro **CHECKED** después del valor:

```
... VALUE="B" CHECKED></form>
```

TYPE=CHECKBOX genera casillas de marca que permiten seleccionar un valor predeterminado o activar una opción



Se escribe:

```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">  
<INPUT TYPE="SUBMIT" VALUE="Procesar">  
<INPUT TYPE="RESET" VALUE="Inicializar">  
Clase A  
<INPUT TYPE="CHECKBOX" NAME="clase" VALUE="A">  
Clase B  
<INPUT TYPE="CHECKBOX" NAME="clase" VALUE="B"></form>
```

En este ejemplo, las casillas de marca permiten seleccionar la clase A, la clase B, ninguna, o las dos. Si se desea que por defecto un valor esté seleccionado, por ejemplo clase B, hay que añadir el parámetro **CHECKED** después del valor:

```
... VALUE="B" CHECKED></form>
```

El elemento SELECT

SELECT sin ningún atributo define por defecto una lista desplegable de toda la pantalla de alto y una columna de ancho:

Como se escriben los formularios



Se escribe:

```
<FORM> <SELECT> </SELECT> </FORM>
```

Como evidentemente esto no es muy práctico, hay que parametrizar el elemento. En primer lugar se le da nombre, ya que éste es un campo más del formulario, la diferencia es que no hay que escribir en él sino escoger un valor de los que nos muestre al desplegarlo, se usará el atributo **NAME**, y para entrar valores en la lista se usa el atributo **OPTION**.



Se escribe:

```
<FORM METHOD="POST" ACTION="http://mecniza.sestud.uv.es/cgi-bin/test2-cgi">
<INPUT TYPE="SUBMIT" VALUE="Procesar">
<INPUT TYPE="RESET" VALUE="Inicializar">

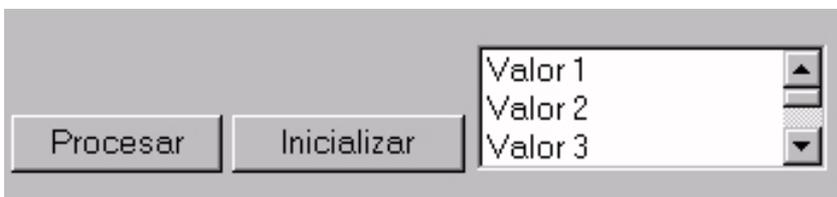
<SELECT NAME="lista1">
  <OPTION>Valor 1
  <OPTION>Valor 2
  <OPTION>Valor 3
</SELECT>
</FORM>
```

Se deben tener en cuenta varias cosas respecto a **SELECT**:

La longitud de la ventana de selección se autoajusta al valor más largo de la lista.

La ventana de selección, por defecto, muestra una línea, si se quieren mostrar más se utilizará el atributo **SIZE**.

Se debe procurar que los contenidos de los campos sean lo más cortos posibles; una excesiva longitud implica mayor tráfico por la red, pero se deben construir las listas de forma clara, y que resulten de fácil lectura al usuario, lo que puede aumentar la longitud de los valores. Para enviar un valor diferente del que aparece en la lista, se utiliza el parámetro **VALUE** asociado al atributo **OPTION**. Prueba con el valor 5 de la siguiente lista:



Se escribe:

```
<FORM METHOD="POST" ACTION="http://mecniza.sestud.uv.es/cgi-bin/test2-cgi">
<INPUT TYPE="SUBMIT" VALUE="Procesar">
<INPUT TYPE="RESET" VALUE="Inicializar">
```

Como se escriben los formularios

```
<SELECT NAME="lista1" SIZE=3>
  <OPTION>Valor 1
  <OPTION>Valor 2
  <OPTION>Valor 3
  <OPTION>Valor 4
  <OPTION VALUE="Valor 5">Este es el Valor 5
  <OPTION>Valor 6
  <OPTION>Valor 7
</SELECT>
</FORM>
```

Esto significa que cuando selecciones "Este es el valor 5" en realidad lo que se envía al servidor es "Valor 5".

Con las listas desplegables, es obligatorio seleccionar siempre un valor, y si no hay ninguno seleccionado, por defecto el visualizador enviará el primer valor de la lista.

Se puede hacer que haya un valor preseleccionado añadiendo el parámetro **SELECTED** al atributo **OPTION**, y si se quiere permitir más de una selección, se pondrá el atributo **MULTIPLE** al elemento **SELECT**. Para seleccionar y deseleccionar hay que atenerse a las normas de Windows.



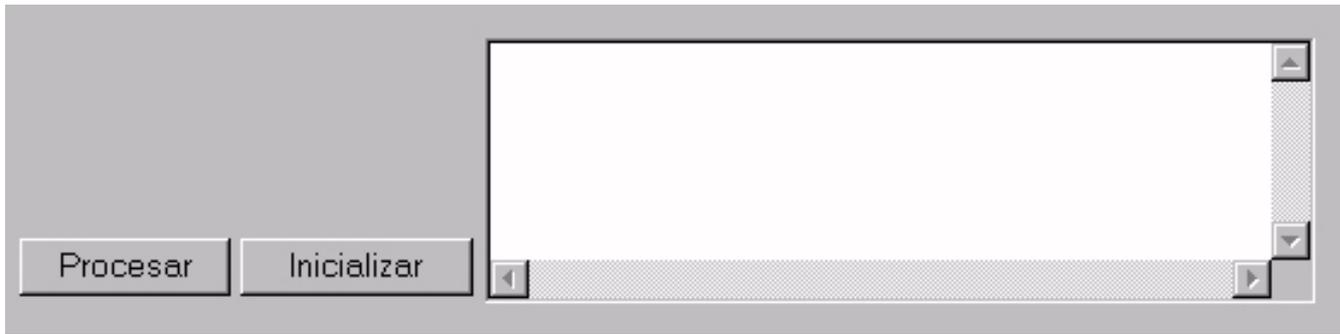
Se escribe:

```
<FORM METHOD="POST" ACTION="http://mecniza.sestud.uv.es/cgi-bin/test2-cgi">
  <INPUT TYPE="SUBMIT" VALUE="Procesar">
  <INPUT TYPE="RESET" VALUE="Inicializar">

  <SELECT NAME="lista1" MULTIPLE SIZE=3>
    <OPTION>Valor 1
    <OPTION SELECTED>Valor 2
    <OPTION>Valor 3
    <OPTION>Valor 4
    <OPTION>Valor 5
    <OPTION>Valor 6
    <OPTION>Valor 7
  </SELECT>
</FORM>
```

El elemento TEXTAREA

Este elemento, como su nombre indica, permite definir un área de texto en la pantalla en la que podemos escribir cualquier cosa. Se debe escribir ya parametrizada en cuanto a dimensiones, con los atributos **ROWS** (líneas) y **COLS** (COLUMNAS). No tiene otros atributos o posibles variantes. Por supuesto, para que sea operativo, deberá ir acompañado de los componentes necesarios para enviar e inicializar.

A screenshot of a web form. On the left, there are two buttons: 'Procesar' and 'Inicializar'. To the right of these buttons is a large, empty text area with a scroll bar on the right side.

Se escribe:

```
<FORM METHOD="POST" ACTION="http://mecniza.sestud.uv.es/cgi-bin/test2-cgi">  
<INPUT TYPE="SUBMIT" VALUE="Procesar">  
<INPUT TYPE="RESET" VALUE="Inicializar">  
  
<TEXTAREA NAME="texto1" ROWS=5 COLS=40 ></TEXTAREA>  
</FORM>
```

Aunque no es habitual, dentro de un área de texto puede haber un contenido por defecto (que puede ser borrado o modificado por el usuario). Para ello simplemente se escribirá el texto entre **<TEXTAREA>** y **</TEXTAREA>**. Así:

```
<>TEXTAREA NAME="texto1" ROWS=5 COLS=40>  
    Contenido del area de texto  
</TEXTAREA>
```

El tratamiento de los datos recibidos en el servidor desde los formularios requiere conocer la tabla de conversión hexadecimal que utilizan la mayoría de visualizadores. Recuerda que las palabras en destino aparecen separadas por el signo +, y las letras acentuadas y otros caracteres, por valores hexadecimales precedidos del símbolo %

Se puede proceder a la descodificación en el programa CGI que recibe los datos, o bien definir macros en programas auxiliares que serán ejecutados después.

En el índice encontrarás la [tabla de conversiones](#) utilizada con el visualizador Netscape desde Windows y una [macro para Word 6 de Windows](#) que convierte los caracteres más habituales.

[\[Indice\]](#)

Ejemplos de formularios

Juego de botones direccionados a páginas LOCALES.

Esto sólo funciona abriendo la página como **file:///** no como servicio **http://**.

Indice	Notas	Cómo funciona?
--------	-------	----------------

```

<TABLE>
<TR>
  <TD>
    <FORM METHOD="POST" ACTION="indice.htm">
      <INPUT TYPE="SUBMIT" VALUE="Indice">
    </FORM>
  </TD>

  <TD>
    <FORM METHOD="POST" ACTION="notas.htm">
      <INPUT TYPE="SUBMIT" VALUE="Notas">
    </FORM>
  </TD>

  <TD>
    <FORM METHOD="POST" ACTION="como.htm">
      <INPUT TYPE="SUBMIT" VALUE="Cómo funciona?">
    </FORM>
  </TD>
</TR>
</TABLE>

```

Juego de botones direccionados a páginas REMOTAS.

Puede funcionar abriendo la página como servicio **http://** o como **file:///**. pero deberá utilizarse **METHOD=GET** en lugar de **METHOD=POST**.

Indice	Notas	Cómo funciona?
--------	-------	----------------

```

<TABLE>
<TR>
  <TD>
    <FORM METHOD="GET" ACTION="http://mcaniza.sestud.uv.es/manual.esp/indice.htm">
      <INPUT TYPE="SUBMIT" VALUE="Indice">
    </FORM>
  </TD>

  <TD>

```

Ejemplos de formularios

```
<FORM METHOD="GET" ACTION="http://mecaniza.sestud.uv.es/manual.esp/notas.htm">
<INPUT TYPE="SUBMIT" VALUE="Notas">
</FORM>
</TD>
```

```
<TD>
<FORM METHOD="GET" ACTION="http://mecaniza.sestud.uv.es/manual.esp/como.htm">
<INPUT TYPE="SUBMIT" VALUE="Cómo funciona?">
</FORM>
</TD>
</TR>
</TABLE>
```

1 campo vacío. Longitud no limitada.

1 campo vacío. Longitud limitada a 10

A screenshot of a web form. It features two input fields at the top, one larger than the other. Below the fields are two buttons: "Procesar" and "Inicializar".

```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">
<INPUT NAME="campo1">
<INPUT NAME="campo2" SIZE=10 MAXLENGTH=10>
<P>
<INPUT TYPE="submit" VALUE="Procesar">
<INPUT TYPE="reset" VALUE="Inicializar"></FORM>
```

1 campo vacío. Longitud no limitada.

1 campo con contenido visible.

1 campo con contenido oculto

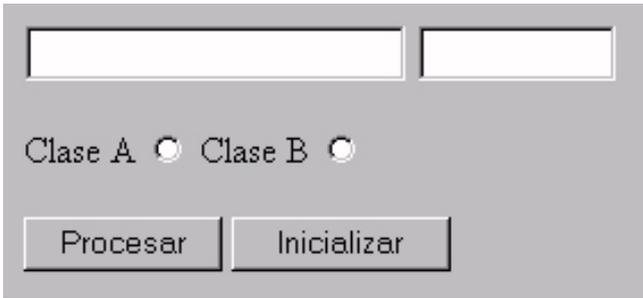
```
<form method="POST" action="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">
<input name="campo1">
<input name="campo2" VALUE="TEXTO VISIBLE">
<input type="password" name="campo3" VALUE="TEXTO OCULTO">
<P>
<input type="submit" value="Procesar">
<input type="reset" value="Inicializar"></form>
```

1 campo vacío. Longitud sin limitar.

1 campo vacío. Longitud limitada a 10.

2 botones de selección.

Ejemplos de formularios



A screenshot of a web form. At the top, there are two empty text input fields. Below them, there are two radio buttons labeled "Clase A" and "Clase B". At the bottom, there are two buttons labeled "Procesar" and "Inicializar".

```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">
  <INPUT NAME="campo1">
  <INPUT NAME="campo2" SIZE=10 MAXLENGTH=10>
<P>
Clase A <INPUT TYPE="RADIO" NAME="clase" VALUE="A">
Clase B <INPUT TYPE="RADIO" NAME="clase" VALUE="B">
<P>
  <INPUT TYPE="SUBMIT" VALUE="Procesar">
  <INPUT TYPE="RESET" VALUE="Inicializar">

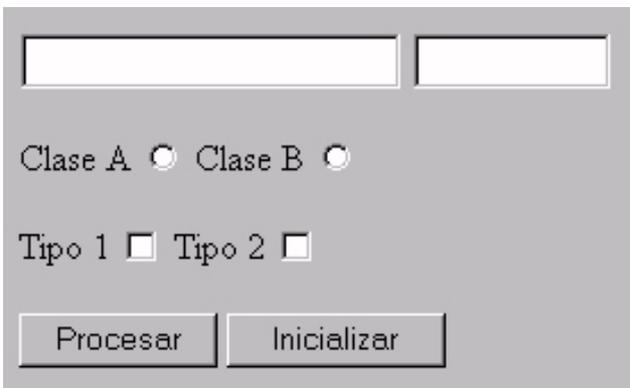
</form>
```

1 campo vacío. Longitud sin limitar.

1 campo vacío. Longitud limitada a 10.

2 botones selección

2 casillas selección



A screenshot of a web form. At the top, there are two empty text input fields. Below them, there are two radio buttons labeled "Clase A" and "Clase B". Below that, there are two checkboxes labeled "Tipo 1" and "Tipo 2". At the bottom, there are two buttons labeled "Procesar" and "Inicializar".

```
<FORM METHOD="POST" ACTION="http://mecaniza.sestud.uv.es/cgi-bin/test2-cgi">
  <INPUT NAME="campo1">
  <INPUT NAME="campo2" SIZE=10 MAXLENGTH=10>
<P>
Clase A <INPUT TYPE="RADIO" NAME="clase" VALUE="A">
Clase B <INPUT TYPE="RADIO" NAME="clase" VALUE="B">
<P>
Tipo 1 <INPUT TYPE="CHECKBOX" NAME="tipo" VALUE="1">
Tipo 2 <INPUT TYPE="CHECKBOX" NAME="tipo" VALUE="2">
<P>
<INPUT TYPE="SUBMIT" VALUE="Procesar">
<INPUT TYPE="RESET" VALUE="Inicializar">

</form>
```

1 lista desplegable de 3 valores. 1 visible



```
<FORM METHOD="POST" ACTION="http://mecniza.sestud.uv.es/cgi-bin/test2-cgi">  
<INPUT TYPE="SUBMIT" VALUE="Procesar">  
<INPUT TYPE="RESET" VALUE="Inicializar">  
  
<SELECT NAME="lista1">  
  <OPTION>Valor 1  
  <OPTION>Valor 2  
  <OPTION>Valor 3  
</SELECT>  
</FORM>
```

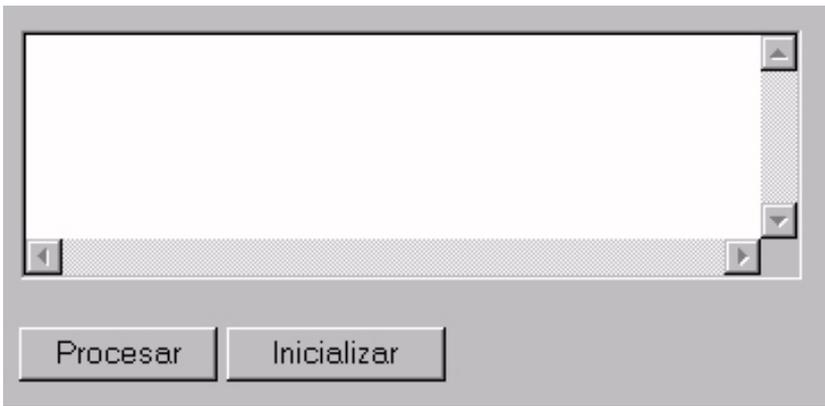
1 lista desplegable de 5 valores. 3 visibles.



```
<FORM METHOD="POST" ACTION="http://mecniza.sestud.uv.es/cgi-bin/test2-cgi">  
  
<SELECT NAME="lista1" SIZE=3>  
  <OPTION>Valor 1  
  <OPTION>Valor 2  
  <OPTION>Valor 3  
  <OPTION>Valor 4  
  <OPTION>Valor 5  
</SELECT>  
<P>  
  <INPUT TYPE="SUBMIT" VALUE="Procesar">  
  <INPUT TYPE="RESET" VALUE="Inicializar">  
</FORM>
```

1 Ventana de texto de 5x40

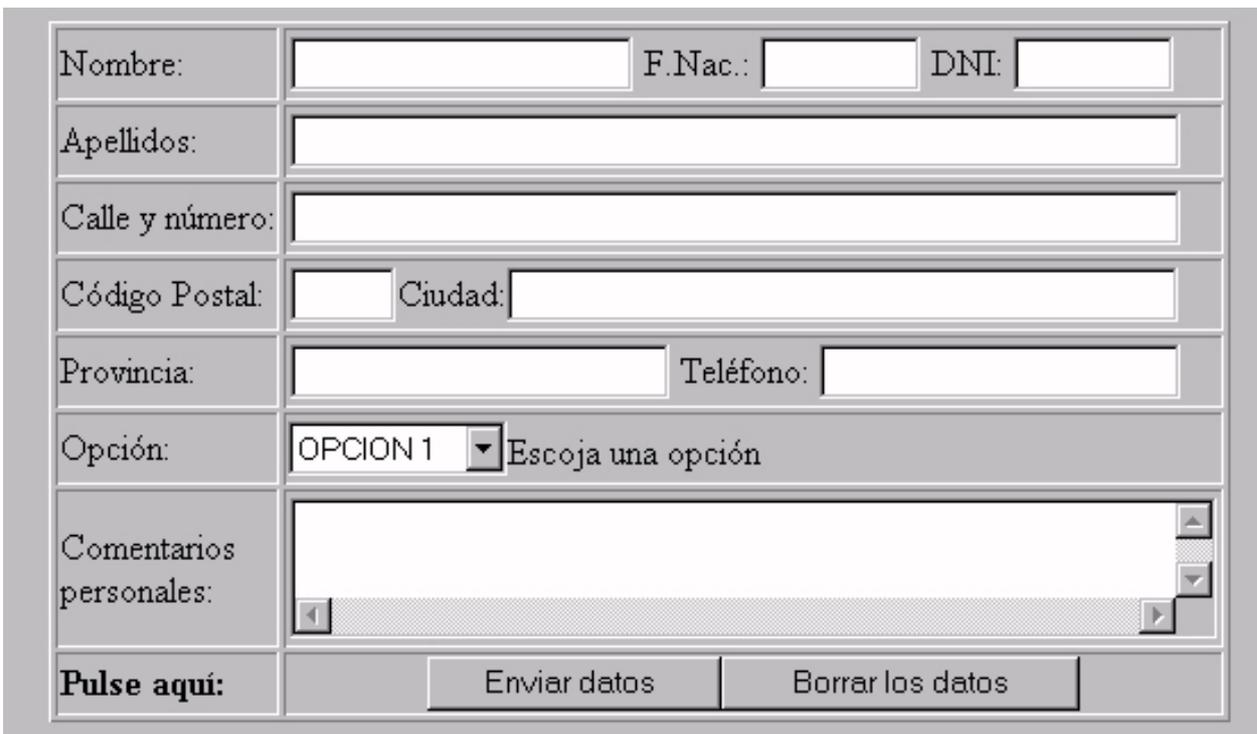
Ejemplos de formularios



A screenshot of a web form. It features a large text area at the top with a vertical scrollbar on the right and horizontal scrollbars at the top and bottom. Below the text area are two buttons: "Procesar" and "Inicializar".

```
<FORM METHOD="POST" ACTION="http://mecniza.sestud.uv.es/cgi-bin/test2-cgi">  
<TEXTAREA NAME="texto1" ROWS=5 COLS=40 ></TEXTAREA>  
<P>  
<INPUT TYPE="SUBMIT" VALUE="Procesar">  
<INPUT TYPE="RESET" VALUE="Inicializar">  
</FORM>
```

Un típico y complejo formulario



A screenshot of a complex web form. It is organized into a table-like structure with several rows. The first row contains three input fields labeled "Nombre:", "F.Nac.:", and "DNI:". The second row is a single wide input field labeled "Apellidos:". The third row is a single wide input field labeled "Calle y número:". The fourth row contains two input fields labeled "Código Postal:" and "Ciudad:". The fifth row contains two input fields labeled "Provincia:" and "Teléfono:". The sixth row contains a dropdown menu labeled "Opción:" with "OPCION 1" selected and the text "Escoja una opción". The seventh row is a large text area labeled "Comentarios personales:". The eighth row contains two buttons labeled "Enviar datos" and "Borrar los datos" under the heading "Pulse aquí:".

```
<FORM ACTION="http://mecniza.sestud.uv.es/cgi-bin/test2-cgi" METHOD=POST>  
<CENTER>  
<TABLE BORDER>  
<TR>  
<TD>Nombre:</TD>  
<TD> <INPUT TYPE="text" NAME="nombre" SIZE=18 MAXLENGTH=18> F.Nac.:  
<INPUT TYPE="text" NAME="edad" SIZE=8 MAXLENGTH=8> DNI:
```

Ejemplos de formularios

```
<INPUT TYPE="text" NAME="dni" SIZE=8 MAXLENGTH=8></TD>
```

```
<TR>
```

```
<TD>Apellidos:</TD>
```

```
<TD> <INPUT TYPE="text" NAME="apellidos" SIZE=48 MAXLENGTH=48></TD>
```

```
<TR>
```

```
<TD>Calle y número:</TD>
```

```
<TD> <INPUT TYPE="text" NAME="domicilio" SIZE=48 MAXLENGTH=48></TD>
```

```
<TR>
```

```
<TD>Código Postal:</TD>
```

```
<TD> <INPUT TYPE="text" NAME="postal" SIZE=5 MAXLENGTH=5>Ciudad:
```

```
<INPUT TYPE="text" NAME="localidad" SIZE=36 MAXLENGTH=36></TD>
```

```
<TR>
```

```
<TD>Provincia: </TD>
```

```
<TD> <INPUT TYPE="text" NAME="provincia" SIZE=20 MAXLENGTH=20>
```

```
Teléfono:
```

```
<INPUT TYPE="text" NAME="telefono" SIZE=19 MAXLENGTH=19></TD>
```

```
<TR>
```

```
<TD>Opción:</TD>
```

```
<TD> <SELECT NAME="OPCION">
```

```
<OPTION>OPCION 1
```

```
<OPTION>OPCION 2
```

```
<OPTION>OPCION 3
```

```
<OPTION>OPCION 4
```

```
<OPTION>OPCION 5
```

```
</SELECT>Escoja una opción</TD>
```

```
<TR>
```

```
<TD>Comentarios<BR> personales:</TD>
```

```
<TD> <TEXTAREA NAME="coment" ROWS=2 COLS=48></TEXTAREA></TD>
```

```
<TR>
```

```
<TD><B>Pulse aquí:</B></TD>
```

```
<TD ALIGN=CENTER>
```

```
<INPUT TYPE="submit" VALUE="Enviar datos ">
```

```
<INPUT TYPE="reset" VALUE="Borrar los datos"></TD>
```

```
</TABLE>
```

```
</CENTER>
```

```
</FORM>
```

[\[Indice\]](#)

[\[Indice\]](#)

Tabla de códigos decimal-hexadecimal

Conversión de caracteres que realiza el visualizador Netscape para Windows en los formularios.

El espacio en blanco es traducido como +

Los visualizadores que trabajan en Windows envían al final del registro los códigos **0D** (decimal 13) y **0A** (decimal 10) que son el retorno de carro y salto de línea respectivamente y por ese orden.

Caracter	Cod. Decimal	Descripción	Cod. hexadecimal
Invisible			Tabulador horizontal (TAB)	%09
Invisible	
	Salto de línea (LF)	%0A
Invisible		Retorno de carro (CR)	%0D
!	!	Cerrar exclamación	%21
"	"	Dobles comillas	%22
#	#	Signo de número	%23
\$	$	Dólar	%24
%	%	Tanto por ciento	%25
&	&	Ampersand	%26
'	'	Apóstrofe	%27
((Cerrar paréntesis	%28
))	Abrir paréntesis	%29
*	*	Asterisco	%2A
+	+	Signo más	%2B
,	,	Coma	%2C
-	-	Guión	%2D
.	.	Punto	%2E
/	/	Barra de división	%2F
:	:	Dos puntos	%3A
;	;	Punto y coma	%3B
<	<	Menor	%3C
=	=	Igual	%3D
>	>	Mayor	%3E
?	?	Cerrar interrogación	%3F

Tabla hexadecimal

@	@	Arroba	%40
[[Cerrar corchete (izquierdo)	%5B
\	\	Barra de división inversa	%5C
]]	Abrir corchete	%5D
^	^	Circunflejo	%5E
_	_	Subrayado	%5F
`	`	Acento agudo	%60
{	{	Abrir llave	%7B
	|	Barra vertical	%7C
}	}	Cerrar llave	%7D
~	~	Tilde	%7E
	 	Espacio sin separación	%A0
!	¡	Abrir exclamación	%A1
¢	¢	Centavo	%A2
£	£	Libra Esterlina	%A3
¤	¤	Signo de divisa general	%A4
¥	¥	Yen	%A5
	¦	Barra vertical partida (pipe)	%A6
§	§	Sección	%A7
¨	¨	Diéresis	%A8
©	©	Copyright	%A9
ª	ª	Doña	%AA
«	«	Abrir comillas francesas	%AB
¬	¬	No (símbolo lógico)	%AC
-	­	Guión débil	%AD
®	®	Registrado	%AE
ˆ	¯	Macrón	%AF
°	°	Grados	%B0
±	±	Más-menos	%B1
²	²	Dos superíndice	%B2
³	³	Tres superíndice	%B3
´	´	Acento agudo	%B4
µ	µ	Micro	%B5
¶	¶	Fin de párrafo	%B6
·	·	Punto medio	%B7

Tabla hexadecimal

¸	¸	Cedilla	%B8
¹	¹	Uno superíndice	%B9
º	º	Género masculino	%BA
»	»	Cerrar comillas francesas	%BB
¼	¼	Un cuarto	%BC
½	½	Un medio	%BD
¾	¾	Tres cuartos	%BE
¿	¿	Abrir interrogación	%BF
À	À	A mayúscula, acento grave	%C0
Á	Á	A mayúscula, acento agudo	%C1
Â	Â	A mayúscula, acento circunflejo	%C2
Ã	Ã	A mayúscula, tilde	%C3
Ä	Ä	A mayúscula, diéresis	%C4
Å	Å	A mayúscula, anillo	%C5
Æ	Æ	Æ mayúscula	%C6
Ç	Ç	C cedilla mayúscula	%C7
È	È	E mayúscula, acento grave	%C8
É	É	E mayúscula, acento agudo	%C9
Ê	Ê	E mayúscula, acento circunflejo	%CA
Ë	Ë	E mayúscula, diéresis	%CB
Ì	Ì	I mayúscula, acento grave	%CC
Í	Í	I mayúscula, acento agudo	%CD
Î	Î	I mayúscula, acento circunflejo	%CE
Ï	Ï	I mayúscula, diéresis	%CF
Ð	Ð	Eth mayúscula	%D0
Ñ	Ñ	Eñe mayúscula	%D1
Ò	Ò	O mayúscula, acento grave	%D2
Ó	Ó	O mayúscula, acento agudo	%D3
Ô	Ô	O mayúscula, acento circunflejo	%D4
Õ	Õ	O mayúscula, tilde	%D5
Ö	Ö	O mayúscula, diéresis	%D6
×	×	Signo de multiplicación	%D7
Ø	Ø	O barrada mayúscula	%D8
Ù	Ù	U mayúscula, acento grave	%D9
Ú	Ú	U mayúscula, acento agudo	%DA

Tabla hexadecimal

Û	Û	U mayúscula, acento circunflejo	%DB
Ü	Ü	U mayúscula, diéresis	%DC
Ý	Ý	Y mayúscula, acento agudo	%DD
Þ	Þ	THORN mayúscula	%DE
ß	ß	Beta minúscula	%EF
à	à	a minúscula, acento grave	%E0
á	á	a minúscula, acento agudo	%E1
â	â	a minúscula, acento circunflejo	%E2
ã	ã	a minúscula, tilde	%E3
ä	ä	a minúscula, diéresis	%E4
å	å	a minúscula, anillo	%E5
æ	æ	ae minúscula	%E6
ç	ç	c cedilla minúscula	%E7
è	è	e minúscula, acento grave	%E8
é	é	e minúscula, acento agudo	%E9
ê	ê	e minúscula, acento circunflejo	%EA
ë	ë	e minúscula, diéresis	%EB
ì	ì	i minúscula, acento grave	%EC
í	í	i minúscula, acento agudo	%ED
î	î	i minúscula, acento circunflejo	%EE
ï	ï	i minúscula, diéresis	%EF
ð	ð	eth minúscula	%F0
ñ	ñ	eñe minúscula	%F1
ò	ò	o minúscula, acento grave	%F2
ó	ó	o minúscula, acento agudo	%F3
ô	ô	o minúscula, acento circunflejo	%F4
õ	õ	o minúscula, tilde	%F5
ö	ö	o minúscula, diéresis	%F6
÷	÷	Signo de división	%F7
ø	ø	o barrada minúscula	%F8
ù	ù	u minúscula, acento grave	%F9
ú	ú	u minúscula, acento agudo	%FA
û	û	u minúscula, acento circunflejo	%FB
ü	ü	u minúscula, diéresis	%FC
ý	ý	y minúscula, acento agudo	%FD

Tabla hexadecimal

þ	þ	thorn minúscula	%FE
ÿ	ÿ	y minúscula, diéresis	%FF

[\[Indice\]](#)


```
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "%F9", .Reemplazar = "ù", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "%C1", .Reemplazar = "Á", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "%C9", .Reemplazar = "É", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "%CD", .Reemplazar = "Í", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "%D3", .Reemplazar = "Ó", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "%DA", .Reemplazar = "Ú", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "%DC", .Reemplazar = "Ü", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "%C0", .Reemplazar = "À", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "%C8", .Reemplazar = "È", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "%CC", .Reemplazar = "Ì", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "%D2", .Reemplazar = "Ò", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "%D9", .Reemplazar = "Û", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "+", .Reemplazar = " ", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "%2B", .Reemplazar = "+", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
EdiciónReemplazar .Buscar = "%0D%0A", .Reemplazar = "^p", .Direcciones = 0,
.CoincidirMayúsMinús = 0, .PalabraCompleta = 0, .CoincidirModelo = 0, .Reservado23 = 0,
.ReemplazarTodo, .Formato = 0, .Ajuste = 2
End Sub
```

```
Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = "%09"
    .Replacement.Text = "^t"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "%0A%0D"
    .Replacement.Text = "^p"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "%21"
    .Replacement.Text = "!"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "%23"
    .Replacement.Text = "#"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "%24"
    .Replacement.Text = "$"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
```

```
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%25"
.Replacement.Text = "%"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%26"
.Replacement.Text = "&"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%27"
.Replacement.Text = ""
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "+"
.Replacement.Text = " "
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%2B"
```

```
.Replacement.Text = "+"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%28"
.Replacement.Text = "("
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%29"
.Replacement.Text = ")"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%2A"
.Replacement.Text = "*"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%2C"
.Replacement.Text = ","
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
```

```
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "%2D"
    .Replacement.Text = "-"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "%2E"
    .Replacement.Text = "."
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "%2F"
    .Replacement.Text = "/"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "%3A"
    .Replacement.Text = ":"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "%3B"
    .Replacement.Text = ";"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
```

```
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%3C"
.Replacement.Text = "<"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%3D"
.Replacement.Text = "="
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%3E"
.Replacement.Text = ">"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%3F"
.Replacement.Text = "?"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%40"
.Replacement.Text = "@"
```

```
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%5B"
.Replacement.Text = "["
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%5C"
.Replacement.Text = "\"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%5D"
.Replacement.Text = "]"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%5F"
.Replacement.Text = "_"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
```

```
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "%7B"
    .Replacement.Text = "{"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "%7C"
    .Replacement.Text = "|"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "%7D"
    .Replacement.Text = "}"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "%A1"
    .Replacement.Text = ";"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
    .Text = "%AA"
    .Replacement.Text = "a"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
```

```
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%B0"
.Replacement.Text = "°"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%BF"
.Replacement.Text = "¿"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%C0"
.Replacement.Text = "À"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%C1"
.Replacement.Text = "Á"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%C7"
.Replacement.Text = "Ç"
.Forward = True
```

```
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%C8"
.Replacement.Text = "È"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%C9"
.Replacement.Text = "É"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%CC"
.Replacement.Text = "Ì"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%CD"
.Replacement.Text = "Í"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
```

```
With Selection.Find
.Text = "%D1"
.Replacement.Text = "Ñ"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%D2"
.Replacement.Text = "Ò"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%D3"
.Replacement.Text = "Ó"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%DA"
.Replacement.Text = "Ú"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%DC"
.Replacement.Text = "Ü"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
```

```
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%E0"
.Replacement.Text = "à"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%E1"
.Replacement.Text = "á"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%E1"
.Replacement.Text = "á"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%E7"
.Replacement.Text = "ç"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%E8"
.Replacement.Text = "è"
.Forward = True
.Wrap = wdFindContinue
```

```
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%E9"
.Replacement.Text = "é"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%ED"
.Replacement.Text = "í"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%F1"
.Replacement.Text = "ñ"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
.Text = "%F2"
.Replacement.Text = "ò"
.Forward = True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
With Selection.Find
```

```
.Text = "%F3"  
.Replacement.Text = "ó"  
.Forward = True  
.Wrap = wdFindContinue  
.Format = False  
.MatchCase = False  
.MatchWholeWord = False  
.MatchWildcards = False  
.MatchSoundsLike = False  
.MatchAllWordForms = False  
End With  
Selection.Find.Execute Replace:=wdReplaceAll  
With Selection.Find  
.Text = "%FA"  
.Replacement.Text = "ú"  
.Forward = True  
.Wrap = wdFindContinue  
.Format = False  
.MatchCase = False  
.MatchWholeWord = False  
.MatchWildcards = False  
.MatchSoundsLike = False  
.MatchAllWordForms = False  
End With  
Selection.Find.Execute Replace:=wdReplaceAll  
With Selection.Find  
.Text = "%FC"  
.Replacement.Text = "ü"  
.Forward = True  
.Wrap = wdFindContinue  
.Format = False  
.MatchCase = False  
.MatchWholeWord = False  
.MatchWildcards = False  
.MatchSoundsLike = False  
.MatchAllWordForms = False  
End With  
Selection.Find.Execute Replace:=wdReplaceAll
```

Qué son los frames

Los frames (en inglés *frame* = cuadro, bastidor o marco) es un elemento implementado por Netscape, que permite dividir la pantalla en varias áreas independientes unas de otras, y por tanto con contenidos distintos, aunque puedan estar relacionados. No hay límites para el contenido de cada una de estas áreas: tienen las mismas propiedades que la pantalla completa normal, tal y como la conocemos. No hay que confundir los frames con las tablas. Su apariencia, a veces, puede ser similar, pero mientras el contenido de la celda de una tabla es fijo, en un área de pantalla creado por el elemento **FRAME** se dispone de todos los recursos del HTML. Es una zona viva.

Las páginas que contienen una definición de **FRAME** **no pueden** contener el elemento **BODY** ni ninguno de los elementos que habitualmente aparecen en el **BODY** **antes** del elemento **FRAMESET** que es el que define la creación del **FRAME**. Si esto no se cumple, el **FRAME** será ignorado.

Algunos visualizadores no soportan los frames. Para que nuestra página con frames no resulte opaca a ellos, se utilizará el elemento **NOFRAMES** que permite ofrecer un texto alternativo en entorno normal.

Los frames permiten una flexibilidad de presentación extraordinaria, y para contenidos muy complejos, de difícil estructura por los medios convencionales del HTML, son casi insustituibles. De nuevo se presenta la duda de para qué visualizador es preferible programar nuestro WWW.

Por supuesto, los frames son parametrizables en cuanto a tamaño y número de áreas, si éstas se pueden redimensionar por el usuario o son fijas, si tienen o no barras de scrolling, se pueden anidar, relacionar sus contenidos, etc. Veamos algunos ejemplos prácticos y su sintaxis:

Un frame básico. Sintaxis general

En general, todas las páginas que contengan definiciones de frames, se comportan como si fuesen llamadores o "lanzaderas", y deberán ser más o menos así:

```
<HTML>
<HEAD><TITLE> Mi titulo ></TITLE></HEAD>

<FRAMESET>
  <NOFRAMES>
    <BODY>
      Su visualizador no soporta frames. Pulse
      <A HREF="indice.htm">aquí </A> para volver.
    </BODY>
  </NOFRAMES>
```

Frames

```
<FRAME SRC="pagina1.htm" >  
<FRAME SRC="pagina2.htm" >
```

```
</FRAMESET>
```

```
</HTML>
```

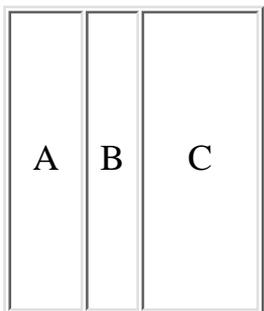
Fíjate en que no aparece el elemento `<BODY>` en su posición habitual, sino dentro de un elemento especial que se activa sólo cuando el visualizador no soporta los frames: **NOFRAMES**. Si no declaras el área **NOFRAMES** y el visualizador no soporta este efecto, no se verá nada. Obviamente, los visualizadores que sí soportan frames ignorarán el contenido del área **NOFRAMES**.

Hasta aquí ya podemos hacernos una idea de cómo funcionan los frames: Lo primero es crear una minipágina con la definición del frame, viene a ser algo así como una "lanzadera" y sólo contiene la definición del frame. Y por último, crear las páginas que constituirán el contenido de cada una de las áreas definidas en la "lanzadera". En el ejemplo anterior son **pagina1.htm** y **pagina2.htm**, y aquí es donde escribirás tus cosas; es decir que son páginas completamente normales, y que también pueden ser utilizadas de la forma habitual. A su vez, la "lanzadera" puede ser invocada con un link desde cualquier página normal. Como puedes ver, no es obligado crear todo tu documento con frames: puedes utilizarlos solamente allí donde sea necesario, si quieres.

Es posible definir cualquier combinación de áreas verticales y horizontales. La clave está en combinar adecuadamente los anidamientos del elemento `<FRAMESET>` con sus atributos **COLS** o **ROWS** según interese. Como puedes ver, el más importante es el primer `<FRAMESET>`, ya que es el que define cómo va a ser "troceada" la página inicialmente, si en porciones verticales u horizontales, y sobre esta base se deberán definir todos los demás anidamientos.

Un frame de 3 áreas verticales (COLS)

Se vería así:



Y se escribiría; así:

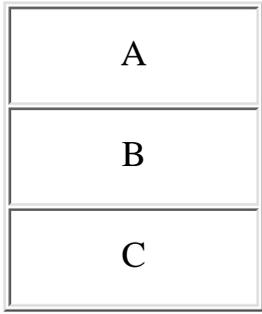
```
<FRAMESET COLS=30%,20%,50%>  
  <FRAME SRC="a.htm">  
  <FRAME SRC="b.htm">  
  <FRAME SRC="c.htm">  
</FRAMESET>
```

[Ejemplo real](#)

Un frame de 3 áreas horizontales (ROWS)

Se vería así:

Frames



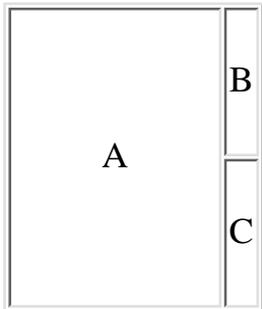
Y se escribiría así:

```
<FRAMESET ROWS=25%, 25%, 50%>  
  <FRAME SRC="a.htm">  
  <FRAME SRC="b.htm">  
  <FRAME SRC="c.htm">  
</FRAMESET>
```

[Ejemplo real](#)

Un frame combinado de un área vertical y dos horizontales

Se vería así:



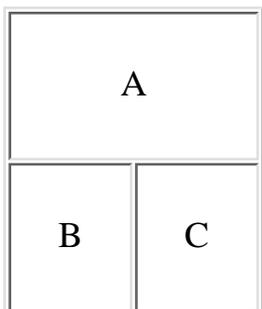
Y se escribiría así:

```
<FRAMESET COLS=20%, *>  
  <FRAME SRC="a.htm">  
  <FRAMESET ROWS=40%, *>  
    <FRAME SRC="b.htm">  
    <FRAME SRC="c.htm">  
  </FRAMESET>  
</FRAMESET>
```

[Ejemplo real](#)

Un frame combinado de dos áreas horizontales, la de abajo dividida en dos verticales

Se vería así:



Y se escribiría así:

```
<FRAMESET ROWS=50%, *>  
  <FRAME SRC="a.htm">  
  <FRAMESET COLS=50%, *>  
    <FRAME SRC="b.htm">  
    <FRAME SRC="c.htm">  
  </FRAMESET>  
</FRAMESET>
```

[Ejemplo real](#)

Habrás podido comprobar que los parámetros que dimensionan los frames actúan al presentar la página inicialmente. Después, si el usuario lo desea, puede redimensionarlos como quiera. Esta es la opción por defecto. Si no se quiere permitir el redimensionado, se aplica el atributo **NORESIZE** al elemento **FRAME**. Así:

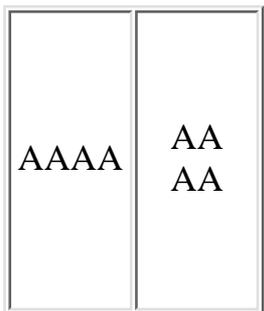
```
<FRAME NORESIZE SRC.....>
```

Algo parecido ocurre con las barras de scrolling. Están regidas por el atributo **SCROLLING**, que puede valer **YES NO** o **AUTO**. Por defecto es **AUTO**. Con esta opción el visualizador decide, en función del contenido, si son necesarias las barras o no. Con **YES** las pondrá siempre, aunque no sean necesarias, y con **NO** no las pondrá nunca, aunque sean necesarias.

```
<FRAME SCROLLING=YES .....>
```

Un frame con dos áreas verticales. Una normal, la otra con márgenes forzados para el texto (**MARGINWIDTH MARGINHEIGHT**)

Se vería así:



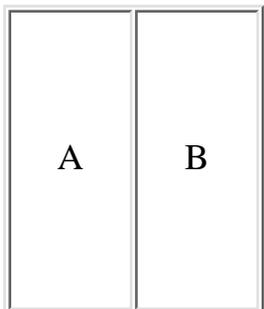
Y se escribiría así:

```
<FRAMESET COLS=50%,50%>
  <FRAME SRC="a.htm">
  <FRAME SRC="a.htm"
    MARGINWIDTH=50
    MARGINHEIGHT=50>
</FRAMESET>
```

[Ejemplo real](#)

Un frame con referencias cruzadas (**NAME TARGET**)

Se vería así:



Y se escribiría así:

```
<FRAMESET COLS=50%,50%>
  <FRAME SRC="aa.htm">
  <FRAME SRC="bb.htm" NAME="VENTANA">
</FRAMESET>
```

[Ejemplo real](#)

Esto de los cruces es un poco complicado... fíjate en el contenido de cada página que entra en juego.

Cuando actives **Ejemplo real** el fichero llamado es **cruzado.htm**. Pruébalo.

Fichero	Contenido	Acción
aa.htm		Pone CC en el frame derecho
		Trae CC
bb.htm	Esto es "VENTANA"	No hace nada
cc.htm		Pone BB en el frame derecho porque se llama "VENTANA"
		Trae AA
cruzado.htm	<FRAMESET>	-
	<FRAME SRC="aa.htm">	Pone AA en el frame izqdo.
	<FRAME SRC="bb.htm" NAME="VENTANA">	Pone BB en el frame derecho y le da como nombre "VENTANA"
	</FRAMESET>	-

Hasta aquí se ha utilizado el nombre de "VENTANA" como destino del atributo TARGET. Este nombre, como ya sabes, es el que hemos definido en el código de este ejemplo. Si en lugar de poner el nombre de destino que se ha definido en el FRAMESET, pones otro cualquiera, el visualizador lo que hace es abrir otra ventana y colocar allí el contenido de la página. Evidentemente, cuantas más ejecuciones del navegador haya en marcha, más memoria del ordenador se necesita, y abrir una nueva ventana, implica ejecutar otra vez el navegador. recuerda cuando hagas esto, que algunos no andan muy sobrados de memoria....

La versión 3.0 de Netscape, implementa dos nuevas posibilidades de los frames: la de darle color a la barra de separación de los distintos frames o la de que no se vea dicha barra. Para ello se utilizan los nuevos atributos de FRAMESET y FRAME:

FRAMEBORDER se utiliza como atributo de FRAMESET y establece si serán visibles los bordes del frame o no. Puede tener dos valores: **YES** (por defecto) y **NO**.

Si escribes:

```
<FRAMESET FRAMEBORDER=NO COLS=30%, 30%, 30%>
  <FRAME SRC="a.htm">
  <FRAME SRC="b.htm">
  <FRAME SRC="c.htm">
</FRAMESET>
```

Se obtiene un frame de tres columnas sin barras de separación entre ellas.

[Ejemplo real](#)

BORDERCOLOR se utiliza como atributo de FRAME y establece el color de los bordes visibles.

Evidentemente, para que funcione, se tendrá que haber establecido **FRAMEBORDER=yes**

Si escribes:

```
<FRAMESET COLS=30%,30%,30%>  
  <FRAME BORDERCOLOR="red" SRC="a.htm">  
  <FRAME BORDERCOLOR="blue" SRC="b.htm">  
  <FRAME SRC="c.htm">  
</FRAMESET>
```

Se obtiene un frame de tres columnas con la primera barra de separación roja y la segunda azul. [Ejemplo real](#)

[\[Indice\]](#)

Ultimas innovaciones

Nuevo Como es normal el HTML, como cualquier otro lenguaje, no está terminado ni lo estará nunca. Día a día se le van implementando nuevas utilidades que le hacen, si cabe, más atractivo y práctico. Hay que distinguir entre lo que son elementos "oficiales" del lenguaje, y las implementaciones que los creadores de cada visualizador le van añadiendo a sus respectivos programas. Buena muestra de ello son los CGI y los applets, plug-ins y JavaScripts de Netscape.

No hay que confundir los **applets** con los plug-ins o los CGI. En la versión 2.0, Netscape introdujo extensiones al lenguaje HTML para permitir incorporar un programa escrito en Java a un documento HTML, del mismo modo que se puede insertar, por ejemplo, una imagen. A estos programas, se les denomina "applets". Son programas diseñados para un fin concreto dentro de la página, como producir una animación o filtrar la entrada de información en un formulario. Los **JavaScripts** no son applets, no tienen la entidad de auténticos programas, y sus acciones son más limitadas. En realidad, y aunque aún no forma parte del HTML oficial, podría considerarse el java interpretado, como una extensión del HTML, igual que las tablas, o cualquier otro elemento, aunque un poco más complejo, al tratarse de un lenguaje basado en objetos y muy parecido al lenguaje C.

Los **plug-ins**, en cambio, son módulos de software (programas) que se ensamblan con el visualizador, añadiéndole nuevas prestaciones o habilidades, que le permitirán descodificar nuevos formatos para los que no estaba preparado originalmente, como el VRML, para la navegación en mundos 3D, el LiveAudio, para escuchar sonidos a través de Internet, o el LiveVideo para ver ficheros de vídeo digital. De esto se deduce, que si no se le han instalado los plug-ins necesarios a nuestro visualizador, al intentar leer una página con, por ejemplo, un fichero de vídeo, lo que éste hará es pedirnos el nombre de un programa auxiliar para que sea él quien gestione ese fichero en lugar de procesarlo por sí mismo.

Todo esto, es respecto a Netscape. Microsoft llegó tarde, pero por supuesto, tiene algo parecido: el **Active X**. Esto no es más que un pequeño módulo capaz de hacer de enlace entre el Internet Explorer (el visualizador de Microsoft) y paquetes de desarrollo como MS Visual Basic, MS Visual J++ o aplicaciones de uso común como MS Excel. El problema es que sólo puede utilizarse en entornos Microsoft, lo que le resta la portabilidad intersistemas que tiene Java.

Te habrás dado cuenta, de que todo lo dicho es referido siempre al propio visualizador, es decir que se trata de acciones realizadas en **local**, en tu propia máquina, y por tanto no es necesario estar conectado a una red. Los programas **CGI** se ejecutan en el servidor, es decir en **remoto**, y sí es necesario estar conectado a la red. Constituyen una solución más fiable que las anteriores, ya que las máquinas donde residen los servidores, suelen ser mucho más potentes que los PC's clientes, y con más recursos siempre disponibles, lo que permite realizar cualquier diseño por complicado que sea, sin obligar a quien se conecta a tener un ordenador caro, pero su eficacia depende absolutamente de la velocidad de la red, que a veces es desesperante... Aunque se mencionan aquí a efectos de comparación, los CGI no son nuevos, ya que aparecieron en las primeras versiones oficiales del HTML.

A medida que vayan apareciendo, se recogerán en esta guía las innovaciones más notables, aunque sólo la parte que sea estrictamente HTML, es decir, lo que hay que saber para incorporar los nuevos inventos a tus páginas, pero no cómo se hacen esos inventos. Por ejemplo, veremos cómo embeber un script interpretado de JavaScript en una página HTML, pero no cómo se programa en JavaScript.

[\[Indice\]](#)

Páginas con sonido

Netscape implementa en la versión 3.0 los sonidos, que junto con el vídeo, son sin duda dos de los efectos más atractivos del HTML. Pueden utilizarse indistintamente ficheros de sonido con los formatos WAV, AIFF, AU o MIDI. En Windows los más utilizados normalmente son los **.WAV** o los **.MID**.

Se utiliza el elemento **<EMBED>** junto a un buen número de atributos: **SRC**, **AUTOSTART**, **LOOP**, **STARTTIME**, **ENDTIME**, **VOLUME**, **WIDTH**, **HEIGHT**, **ALIGN**, **CONTROLS** y **HIDDEN**.

Veamos para qué sirve cada uno:

SRC Indica dónde está el fichero de sonido, pero no puede utilizarse solo, necesita la compañía de otros atributos del elemento, concretamente los que definen el tamaño de la consola (que es un icono parecido al teclado que tiene un cassette pequeño): **<WIDTH>** (largo) y **<HEIGHT>** (alto) Por ejemplo:

```
<EMBED SRC="mozart.mid" WIDTH=160 HEIGHT=70>
```

Hará aparecer una consola como esta:



En caso de omitir los atributos **WIDTH** y **HEIGHT**, los tamaños por defecto no permiten ver la consola entera, y no es posible activar el sonido. Los tamaños máximos son **WIDTH=160** y **HEIGHT=70**; no sirve de nada aumentarlos y si se disminuyen, la imagen se recorta de derecha a izquierda y de abajo a arriba, pero no proporcionalmente. Como puedes ver, la consola incluye un control de sonido, que por defecto se sitúa en el 50%. Este valor puede modificarse con el atributo **VOLUME** y puede oscilar entre 0 y 100. La consola también permite cambiar el volumen manualmente.

Es posible cambiar el aspecto de la consola con el atributo **CONTROLS** que puede tener varios valores: **CONSOLE** (por defecto) o **SMALLCONSOLE**. Si escribes

```
<EMBED SRC="mozart.mid" WIDTH=144 HEIGHT=15 CONTROLS=SMALLCONSOLE>
```

Aparece una consola como esta:



Con **SMALLCONSOLE** los tamaños por defecto sólo dejan ver los botones de **START** Y **STOP**, desapareciendo el control de volumen. Los tamaños máximos son **WIDTH=144** y **HEIGHT=15**, y al igual que antes, no sirve de nada aumentarlos.

Los otros valores de **CONTROLS** son: **PLAYBUTTON**, **PAUSEBUTTON**, **STOPBUTTON** y **VOLUMELEVER**. Aunque en inglés, es fácil deducir lo que hace cada uno, y sus tamaños máximos son **WIDTH=37** x **HEIGHT=22**. Sólo es posible utilizar un valor por cada elemento **EMBED**, si pones más de uno sólo hace efecto el primero. Si se quiere usar más de un botón, por ejemplo el de start y el stop en distintos lugares de la página, pero actuando sobre el mismo fichero de sonido, usaremos el atributo **MASTERSOUND** en combinación

Paginas con sonido

con **NAME**. Si escribes:

```
<EMBED SRC="mozart.mid" WIDTH=37 HEIGHT=22 CONTROLS=PLAYBUTTON  
  NAME="misonido" MASTERSOUND>
```

```
<EMBED SRC="mozart.mid" WIDTH=37 HEIGHT=22 CONTROLS=STOPBUTTON  
  NAME="misonido">
```

Se obtiene:



Lo que se ha hecho en el primer elemento EMBED, mediante el atributo **NAME**, es darle nombre a un grupo de botones (**misonido**) y designarlo como "maestro", con el atributo **MASTERSOUND**, de todos los siguientes elementos EMBED que se definan con el mismo nombre (misonido). De esta manera, es posible controlar un sonido repartiendo los botones de la consola de la forma más conveniente. Se puede definir en la misma página más de un grupo de sonidos, sólo hay que procurar no liarse con los nombres.

Otro atributo de EMBED es **AUTOSTART** con dos posibles valores: **FALSE** (por defecto) y **TRUE**, y como ya habrás deducido, **TRUE** hace que suene inmediatamente después de cargar la página, y sin necesidad de pulsar el botón "start" de la consola. Por ejemplo:

```
<EMBED SRC="mozart.mid" WIDTH=160 HEIGHT=70 AUTOSTART=TRUE>
```

El siguiente atributo a analizar es **HIDDEN** que sólo tiene un valor: **TRUE**, y hace invisible la consola, pero evidentemente, si es invisible no puedes activar el sonido, y por tanto lo hace él mismo, siempre que no hayas puesto **AUTOSTART=FALSE**, claro. Por ejemplo:

```
<EMBED SRC="mozart.mid" WIDTH=160 HEIGHT=70 HIDDEN=TRUE>
```

Un atributo de repetición no podía faltar, y lo tenemos en **LOOP** y sus tres posibles valores: **FALSE** (por defecto), **TRUE** que fuerza a la repetición sin fin de la audición, e **INTEGER** donde **INTEGER** es el número de veces que deseas que se repita la audición. Por ejemplo:

```
<EMBED SRC="mozart.mid" WIDTH=160 HEIGHT=70 LOOP=2>
```

No es obligado comenzar una audición por el principio del fichero. Imagínate que tienes un fichero con instrucciones habladas, y quieres dar la oportuna en cada momento, no todas de golpe. Para esto tenemos los atributos **STARTTIME=mm:ss** y **ENDTIME=mm:ss** donde **mm** son los minutos y **ss** los segundos del punto de comienzo o fin de la audición. **Cuidado**: los valores hay que darlos con dos dígitos: Por ejemplo:

```
<EMBED SRC="mozart.mid" WIDTH=160 HEIGHT=70 STARTTIME=01:10 ENDTIME=02:00>
```

Este atributo funciona sólo con Windows 95 o Macintosh.

Por último tenemos el atributo **ALIGN** y sus valores **TOP**, **BOTTOM**, **CENTER**, **LEFT**, **RIGHT**, **MIDDLE** y **TEXTTOP** que funcionan igual que sus respectivos del elemento [IMG](#), es decir, que no tiene efectos sobre el sonido, sino sobre la imagen de la consola y su alineación con el texto de la página.

Un truco..

Si en una página pones dos o más ficheros **.mid**, sólo se oye el último que sea activado. La secuencia es: se carga el primero, arranca y se oirá durante el tiempo que tarde en cargarse el segundo, y en cuanto arranca el segundo, el primero pasa a la posición de "stop". Si los ficheros son **.wav** ocurre lo mismo, pero ¿que pasa si pones un **.mid** y un **.wav**? Pues que cada uno de ellos utiliza controladores de recursos de sistema diferentes, y por tanto se pueden oír los dos a la vez. Imagínate tu voz explicando algo, y una suave música de fondo... Como en los más sofisticados documentales. Eso sí, necesitas Windows 95.

Y un aviso:

Las páginas que incluyen sonidos pueden dar problemas a la hora de imprimirlos, llegando incluso a colgar la máquina.

En el índice tienes algunos [ejemplos](#) de sonidos.

[\[Índice\]](#)

Ejemplos de páginas con sonido

Consola grande. Repetición sin fin. Volumen al 30%. Sin autoarranque.

Se escribe:

```
<EMBED SRC="mozart.mid" CONTROLS=CONSOLE WIDTH=160 HEIGHT=70 LOOP=TRUE  
VOLUME=30 AUTOSTART=FALSE>
```

Consola grande. Sin repetición. Volumen al 50% (por defecto). Sin autoarranque. Fragmento entre los 10 y los 20 segundos del fichero (Sólo con windows 95)

Se escribe:

```
<EMBED SRC="mozart.mid" CONTROLS=CONSOLE WIDTH=160 HEIGHT=70 LOOP=FALSE  
AUTOSTART=FALSE STARTTIME=00:10 ENDTIME=00:20>
```

Consola pequeña. Repetición sin fin. Volumen al 40%. Sin autoarranque.

Se escribe:

```
<EMBED SRC="minueto.mid" CONTROLS=SMALLCONSOLE WIDTH=144 HEIGHT=15 LOOP=TRUE  
VOLUME=40 AUTOSTART=FALSE>
```

Consola pequeña sin control de volumen (tamaños por defecto). Repetición sin fin. Volumen al 50% (por defecto). Sin autoarranque.

Se escribe:

```
<EMBED SRC="adagio.mid" CONTROLS=SMALLCONSOLE LOOP=TRUE AUTOSTART=FALSE>
```

Mezcla de sonidos. Tres ficheros de tipos distintos. Sólo botón de start. 1 sola audición por fichero (por defecto). Volumen al 40%. Sin autoarranque.

Mientras suena el primer fichero puede oírse cualquiera de los otros dos, mezclando los sonidos.

Se escribe:

```
<EMBED SRC="pinkpant.mid" CONTROLS=PLAYBUTTON VOLUME=40 AUTOSTART=FALSE>  
<EMBED SRC="caballo.wav" CONTROLS=PLAYBUTTON VOLUME=40 AUTOSTART=FALSE>  
<EMBED SRC="fanfar.wav" CONTROLS=PLAYBUTTON VOLUME=40 AUTOSTART=FALSE>
```

[\[Indice\]](#)

Ejemplos de páginas con sonido

Consola grande. Repetición sin fin. Volumen al 30%. Sin autoarranque.



Se escribe:

```
<EMBED SRC="mozart.mid" CONTROLS=CONSOLE WIDTH=160 HEIGHT=70 LOOP=TRUE  
VOLUME=30 AUTOSTART=FALSE>
```

Consola grande. Sin repetición. Volumen al 50% (por defecto). Sin autoarranque. Fragmento entre los 10 y los 20 segundos del fichero (Sólo con windows 95)



Se escribe:

```
<EMBED SRC="mozart.mid" CONTROLS=CONSOLE WIDTH=160 HEIGHT=70 LOOP=FALSE  
AUTOSTART=FALSE STARTTIME=00:10 ENDTIME=00:20>
```

Consola pequeña. Repetición sin fin. Volumen al 40%. Sin autoarranque.



Se escribe:

```
<EMBED SRC="minueto.mid" CONTROLS=SMALLCONSOLE WIDTH=144 HEIGHT=15 LOOP=TRUE  
VOLUME=40 AUTOSTART=FALSE>
```

Consola pequeña sin control de volumen (tamaños por defecto). Repetición sin fin. Volumen al 50% (por defecto). Sin autoarranque.



Se escribe:

```
<EMBED SRC="adagio.mid" CONTROLS=SMALLCONSOLE LOOP=TRUE AUTOSTART=FALSE>
```

Mezcla de sonidos. Tre ficheros de tipos distintos. Sólo botón de arranque. 1 sola audición por fichero. Volumen al 40%. Sin autoarranque.

Mientras suena el primer fichero puede oirse cualquiera de los otros dos.

Ejemplos de sonido (impresion)



Se escribe:

```
<EMBED SRC="pinkpant.mid" CONTROLS=PLAYBUTTON LOOP=INTEGER1 VOLUME=40  
AUTOSTART=FALSE>  
<EMBED SRC="caballo.wav" CONTROLS=PLAYBUTTON LOOP=INTEGER1 VOLUME=40  
AUTOSTART=FALSE>  
<EMBED SRC="fanfar.wav" CONTROLS=PLAYBUTTON LOOP=INTEGER1 VOLUME=40  
AUTOSTART=FALSE>
```

[\[Indice\]](#)

Páginas con vídeo

Netscape implementa en la versión 3.0 el vídeo. Este elemento sólo funciona en los navegadores para Windows 95 (con el correspondiente plug-in). La razón estriba en que éste es un sistema operativo multitarea, es decir, que puede hacer más de una cosa simultáneamente, mientras que Windows 3.1 es mono tarea; y aunque no lo parezca, presentar el texto de la página y que al mismo tiempo pueda verse una película, para el sistema son dos trabajos diferentes, aunque estén integrados en un mismo programa, que es el visualizador.

El formato de vídeo utilizado es el AVI de Windows, que como es sabido, incluye imágenes y sonido digitales.

El mayor problema de los ficheros digitales de vídeo es el gran tamaño de los mismos, ya que en la práctica unos pocos segundos de vídeo con sonido puede representar un fichero de varios Mb. (mega bytes), en realidad es como si se trataran dos ficheros separados, uno con el sonido y otro con las imágenes, pero sincronizados. No hay que confundir este formato AVI con el MPEG, que también puede integrar imágenes y sonido, aunque con un nivel de compresión diferente, y peor sincronización con el sonido. Habitualmente los ficheros .MPG se suelen encontrar sin sonido.

Al igual que con el sonido, se utiliza el elemento **EMBED**, aunque con menos atributos: **SRC**, **HEIGHT**, **WIDTH**, **AUTOSTART**, **LOOP** y **ALIGN**; todos ellos funcionan de forma parecida a los del sonido.

En este ejemplo se utilizan casi todos:

```
<EMBED SRC="colon.avi" HEIGHT=300 WIDTH=400 AUTOSTART=TRUE LOOP=FALSE>
```

Analicémoslos brevemente:

SRC Indica dónde está el fichero de vídeo. **HEIGHT** (alto) y **WIDTH** (ancho) no pueden omitirse, ya que sus valores por defecto no permiten ver ninguna imagen real.

Los tamaños standard de los ficheros .AVI son 90x120, 120x160, 180x240, 240x320, 300x400, etc., las medidas deben conservar las proporciones de la televisión, es decir 4:3. Hay que tener cuidado a la hora de poner el valor de estos atributos, ya que si no coinciden con los que realmente tiene la imagen, ésta será recortada (no redimensionada) de derecha a izquierda y de abajo a arriba.

El siguiente atributo es **AUTOSTART**, con dos valores posibles: **FALSE** (por defecto) y **TRUE** que como puedes suponer, sirve para que el vídeo arranque nada más cargarse el fichero. Si estableces **FALSE**, entonces el vídeo se activa al hacer click con el ratón sobre él, lo mismo que para repetir el pase una vez finalizado.

Para hacer que el vídeo se repita continuamente o se pare después de un pase, tenemos el atributo **LOOP** con dos valores: **FALSE** (por defecto) y **TRUE**.

Por último tenemos el atributo **ALIGN** y sus valores **TOP**, **BOTTOM**, **CENTER**, **LEFT**, **RIGHT**, **MIDDLE** y **TEXTTOP** que funcionan igual que sus respectivos del elemento [IMG](#), es decir, que no tiene efectos sobre el vídeo, sino sobre su alineación con el texto de la página.

[\[Indice\]](#)

Los lenguajes JavaScript y Java

Como ya es sabido, para que ciertos [formularios](#) sean eficaces necesitan que en un servidor http haya un programa específico que realice las transacciones que deseamos mediante un programa [CGI](#); también necesitaremos uno de estos programas si lo que queremos es, por ejemplo, realizar un sencillo cálculo. Esto además de complicado precisa la colaboración del administrador del sistema, carga el servidor con multitud de pequeñas tareas que pueden llegar a colapsarlo, depende de la velocidad de la red, a veces muy lenta, y limita la portabilidad de muchos documentos a máquinas conectadas a la red. Netscape junto con SUN ha implementado el lenguaje Java en el visualizador con el fin de trasladar al mismo la solución de parte de estas tareas.

El lenguaje Java es muy parecido al lenguaje C. No vamos a tratar aquí de programación, sólo veremos qué es esto y algunos ejemplos para hacernos una idea de lo que se puede conseguir.

El visualizador de Netscape (versión 2.0 o superior) tiene dos implementaciones de Java: el **JavaScript** que es un java interpretado, es decir que se ejecuta interpretando paso a paso un programa que está escrito en la propia página HTML, y los **applets** en Java, que son verdaderos programas compilados y que son ejecutados a través del visualizador. En este caso, lo que hay en la página HTML únicamente son referencias del programa, donde está, parámetros de funcionamiento, etc.

Los JavaScript necesitan "menos máquina" para ejecutarse, aunque también son menos potentes en cuanto a lo que son capaces de hacer. Por ejemplo, no tienen funciones de fichero. Pero solucionan con brillantez bastantes problemas sin necesidad de programa de apoyo en el servidor.

Los programas java compilados, o applets, son mucho más exigentes en cuanto a necesidades de máquina, pero se puede hacer casi cualquier cosa con ellos, siempre que seas un "enterado" en la programación en C++, ya que son prácticamente iguales.

Para embeber un JavaScript en una página HTML, se utiliza el elemento **<SCRIPT>**. Este elemento, habitualmente, va dentro del HEAD de la página, y **no** en el BODY.

Por ejemplo si escribes:

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

<SCRIPT LANGUAGE="JavaScript">
<!--
    document.write("<CENTER><H3>Mi primer JavaScript</H3></CENTER>")
```

```
// -->
</SCRIPT>

</HEAD>

<BODY>

<HR>
<CENTER>
<H3>
[<A HREF="java.htm">Volver</A>]
</H3>
</CENTER>

</BODY>
</HTML>
```

Tendríamos una página con el más elemental de los scripts. Fíjate en que después del elemento SCRIPT se abre un comentario (<!-- -->) que contiene las instrucciones. Esto es para ocultar el script a los visualizadores que no sean capaces de interpretarlo. Fíjate también en que antes del cierre del comentario (-->) hay dos barras (//): Esto es para que java no interprete el cierre como instrucción y dé un error. Puedes probar el script pulsando [aquí](#).

En el índice encontrarás algunos [ejemplos](#) de páginas con programas JavaScript. Para ver cómo están hechos utiliza la orden **View** (Ver) de la barra de comandos del visualizador, y escoge la opción **Document Source** (Documento fuente).

Los APPLETs escritos en Java son algo más complicados, sobre todo en cuanto a su escritura, ya que es prácticamente igual que el lenguaje C++. El elemento para embeberlos en una página HTML es **APPLET** que sí puede escribirse en el BODY. Así:

```
<BODY>

<APPLET CODE="AnimText.class" WIDTH=600 HEIGHT=40>
  <PARAM NAME=text VALUE="Esto es una prueba de Applet Java">
  <PARAM NAME=font VALUE="TimesRoman">
  <PARAM NAME=type VALUE=1>
  <PARAM NAME=fgcolor VALUE="yellow">
  <PARAM NAME=min VALUE=12>
  <PARAM NAME=max VALUE=36>
  <PARAM NAME=naptime VALUE=100>
  <PARAM NAME=align VALUE=left>
</APPLET>
```

Este es un applet que forma "olas" con un texto, y puedes probarlo desde [aquí](#). (Si estas utilizando la guía empaquetada, primero debes renombrar el fichero **anim.cla** a su nombre real: **AnimText.class**, de lo

contrario no funcionará)

Analicémoslo brevemente:

El valor del atributo **CODE** es el nombre del programa: "**AnimText.class**". Los programas Java cuando se compilan, no son ejecutables directamente desde el sistema operativo de tu máquina, son compilados a nivel objeto con la extensión **.class**, y ejecutados a través del visualizador.

Los siguientes elementos son **PARAM NAME**, y puede deducirse fácilmente para que sirve cada uno: definición del texto que va a aparecer, fuente de la letra, color del texto, etc..

Si pulsas [aquí](#) podrás ver el programa fuente de este applet. Aunque aquí se llama applet.txt, su nombre real es **AnimText.java**. Esta extensión (**.java**) es obligatoria para que el compilador acepte el fichero.

Como ya habrás notado, estamos usando extensiones de fichero de **cuatro** letras, es decir, con una máquina que tenga Windows 3.1 no podrás hacer nada de esto. En efecto, los applets de Java sólo funcionan (y se compilan) en máquinas con sistemas operativos de 32 bits (por ejemplo Windows 95 o Macintosh).

Hay que tener cuidado, a la hora de crear scripts o programas de java, con la versión de lenguaje que se utiliza, ya que los visualizadores no soportan todos el mismo nivel de lenguaje, y puedes ver con sorpresa que un script o un applet que te funcionaba perfectamente con Netscape 2.0, no trabaja bien, no trabaja en absoluto o incluso te "cuelga" la máquina con Netscape 3.0, y no digamos si es al revés. Los resultados de un programa o script hecho para un determinado navegador, son imprevisibles con otro si no estan hechos con mucho cuidado, e indefectiblemente funcionarán mal o no lo harán con Windows 3.1 (de nuevo la duda de siempre: ¿lo hago bonito y los demás que se fastidien, o lo hago para todo el mundo...?)

[\[Indice\]](#)

[\[Indice\]](#)

Ejemplos de páginas con JavaScript

He aquí algunos ejemplos de páginas con "scripts" escritos en JavaScript (java interpretado).

● Ejemplos JavaScript

- [Un reloj en pantalla](#)
 - [Dar el nombre del mes](#)
 - [Una calculadora](#)
 - [Una agenda-calendario](#)
 - [Control de contenido de campos](#) Sin envío.
 - [Control de contenido de campos](#) Envío final.
 - [Texto que se desplaza en la ventana de estado](#)
 - [Texto que se desplaza en la ventana de estado opcionalmente](#)
 - [Formulario con lista de selección sin botón de submit](#)
 - [Imprimir desde una imagen](#)
 - [Texto que sigue al puntero en forma de cola de cometa](#)
-

[\[Indice\]](#)

Esto es un reloj hecho con JavaScript

Hoy es: <input type="text"/>	La hora es: <input type="text"/>
------------------------------	----------------------------------

[\[Volver\]](#)

Ejemplo de una página con calendario que da el mes en letra

[\[Volver\]](#)

Calculadora. Ejemplo de Javascript

Esta calculadora trabaja de una forma un tanto diferente a como es habitual. Supongamos que quieres sumar dos números. para ello entra el primer número, pulsa ENTER, después el segundo número, y por fin el signo de la operación que quieres hacer, en este caso +.

Stack:	<input type="text" value="0"/>
Acc:	<input type="text" value="0"/>

[\[Volver\]](#)

Calendario-agenda. Ejemplo de JavaScript

Este es un sencillo ejemplo de programación en JavaScript. La base es el formulario. Los textos correspondientes a cada día son valores ocultos de campo. Lo único que hace el script de Java es mostrarlos.

<p>ENERO, 1996 Dom Lun Mar Mie Jue Vie Sab</p>	<div data-bbox="517 568 1241 824" style="border: 1px solid black; height: 114px;"></div> <div data-bbox="804 833 954 878" style="border: 1px solid black; padding: 2px; text-align: center;">Inicializar</div>
--	--

[\[Volver\]](#)

Verificación de entrada de datos en un formulario con JavaScript

Entrar tu E-mail (p.ej.: pepe@uv.es):	<input type="text"/>
Entrar 5 caracteres sólo:	<input type="text"/>
Entrar 3 o más caracteres:	<input type="text"/>
Entrar cualquier cosa:	<input type="text"/>
<input type="button" value="Inicializar"/>	

[\[Volver\]](#)

Verificación de campos en un formulario con JavaScript. Envío a un servidor.

Entra tu E-mail (p.ej.: pepe@uv.es):	<input type="text"/>
Entrar cinco caracteres:	<input type="text"/>
Entrar tres o más caracteres:	<input type="text"/>
Entrar cualquier cosa:	<input type="text"/>
	<input type="button" value="Inicializar"/>

[\[Volver\]](#)

Ejemplo de JavaScript

Texto que se desplaza en la ventana de estado del visualizador

[\[Volver\]](#)

Ver mensaje desplazandose

[\[Volver\]](#)

Ejemplo de una lista de selección sin botón de submit

Sólo hay que seleccionar un valor y se ejecuta la instrucción al hacer clic

Indice
Propósito
Notas sobre el lenguaje
Como funciona esto
Introducción al HTML
Estructura de un documento
Cabecera de un documento
Cuerpo de un documento
Tamaño y posición de los

[\[Volver\]](#)

Imprimir desde una imagen con JavaScript

(Sólo Netscape 4.0 o superior)



[\[Volver\]](#)

Prueba de Java Script que genera una cola de cometa con un texto

Netscape 4.2 o superior

[\[Volver\]](#)

Gráficos animados

Una nueva propiedad de algunos visualizadores es la posibilidad de presentar gráficos con animación. No hay que confundir los gráficos animados en formato GIF con los "movies" o vídeos en formato MPG que no son tratados directamente por el visualizador, sino por una herramienta auxiliar que hay que definirle.

Los visualizadores suelen presentar correctamente los formatos GIF y JPG. Netscape ha implementado la posibilidad de mostrar un tipo de gráfico tipo GIF que incluye en su interior una secuencia, que junto con instrucciones de repetición (loop), espera (wait) y controles de cabecera permiten reproducir una animación. La diferencia de tamaño entre un GIF animado y un MPG es considerable. Evidentemente, las animaciones GIF tienen peor resolución que las MPG, y sólo están pensadas para fines sencillos.

La forma de incluir una imagen animada en el documento es la misma que una imagen fija. No se requiere ningún tipo de programa ni script, todo está definido en la propia imagen. Existen en la red en forma de shareware programas para hacer ficheros de imágenes animadas. Su construcción es muy sencilla: sólo necesitas tener la serie de imágenes que quieres animar e ir ensamblándolas con algún programa al efecto, definir el tiempo entre imagen, el ciclo de repetición... y listo.

Los visualizadores que no soportan este efecto, lo que hacen es presentar solamente la primera o la última de las imágenes de la secuencia. Hay que tener esto presente, y no comenzar o terminar las secuencias con, por ejemplo, una imagen vacía.

Aquí tienes dos ejemplos sencillos de GIF animado:

Una bola que corre.



Una carta que se mete en un sobre.



[\[Indice\]](#)

Adición de nuevos temas a la guía

NUEVO Nueva macro de conversión para ficheros recibidos desde formularios para [Word97 de Windows95](#)

[\[Indice\]](#)

[\[Volver\]](#)

Este es un documento de ejemplo de color de fondo "#FFFFFF" blanco.

[\[Volver\]](#)

[\[Volver\]](#)

Este es un documento de ejemplo de color de fondo "#FF0000" rojo.

[\[Volver\]](#)

[\[Volver\]](#)

Este es un documento de ejemplo de color de fondo "#00FF00" verde.

[\[Volver\]](#)

[]

Este es un documento de ejemplo de color de fondo "#0000FF" azul.

[]

[\[Volver\]](#)

**Este es un documento de ejemplo de color de fondo "#00FFFF"
cyan.**

[\[Volver\]](#)

[\[Volver\]](#)

Este es un documento de ejemplo de color de fondo "#FFFF00" amarillo.

[\[Volver\]](#)

[\[Volver\]](#)

Este es un documento de ejemplo de color de fondo "#70DB93" aguamarina.

[\[Volver\]](#)

[\[Volver\]](#)

Este es un documento de ejemplo de color de fondo "#5C3317" marrón chocolate.

[\[Volver\]](#)

[\[Volver\]](#)

Este es un documento de ejemplo de color de fondo "#FF00FF" magenta.

[\[Volver\]](#)

[Volver](#)

[Volver](#)

Esto es una "demo" de documento HTML

Esto es el más sencillo de los documentos.