

Manual del Curso Básico de **HTML**

(Introducción a la programación de páginas Web)

Juan Luis Campos Salcedo

Colima, Colima (Nov. De 2005)

QUÉ ES HTML

HTML (*HyperText Markup Language*) es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con *enlaces* (*hyperlinks*) que conducen a otros documentos o fuentes de información relacionadas, y con *inserciones* multimedia (gráficos, sonido...) La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Mosaic, o Netscape).

CÓMO ESPECIFICAR EFECTOS DEL TEXTO

La mayoría de los efectos se especifican de la misma forma: rodeando el texto que se quiere marcar entre dos *etiquetas* o *directivas* (*tags*, en inglés), que definen el efecto o unidad lógica que se desea. Las etiquetas están formadas por determinados códigos metidos entre los signos < y >, y con la barra / cuando se trata de la segunda etiqueta de un efecto (la de cierre). Por ejemplo: <efecto> para abrir y </efecto> para cerrar. Ciertas directivas sólo se ponen una vez en el lugar del texto donde queramos que aparezca el efecto concreto. Esto es lo que ocurre, por ejemplo, cuando queremos poner un gráfico, caso en el que se usa algo parecido a <poner_gráfico_aquí> (más adelante ya veremos la directiva concreta que se utiliza).

A veces es necesario ofrecer datos adicionales en una directiva. Por ejemplo, cuando se define un *hiperenlace* hay que especificar su destino. Para ello se incluyen parámetros en la directiva inicial (la de apertura), de la siguiente forma: <efecto parametro1 parametro2 . . .>. La directiva de cierre, caso de ser necesaria, queda como antes: </efecto>.

Más adelante en el presente documento se muestra el efecto de las directivas más usadas en la creación de un documento HTML. Para cada una de ellas, primero se muestra el texto fuente, y bajo éste, el efecto que produce.

ESTRUCTURA BÁSICA DE UN DOCUMENTO HTML

Un documento HTML comienza con la etiqueta <html>, y termina con </html>. Dentro del documento (entre las etiquetas de principio y fin de html), hay dos zonas bien diferenciadas: el *encabezamiento*, delimitado por <head> y </head>, que sirve para definir diversos valores válidos en todo el documento; y el *cuerpo*, delimitado por <body> y </body>, donde reside la información del documento.

La directiva `<title>`, permite especificar el título de un documento HTML. Este título no forma parte del documento en sí: no aparece, por ejemplo, al principio del documento una vez que este se presenta con un programa adecuado, sino que suele servir como título de la ventana del programa que nos la muestra.

El cuerpo de un documento HTML contiene el texto que, con la presentación y los efectos que se decidan, se presentará ante el *hiperlector*. Dentro del cuerpo son aplicables todos los efectos que se van a mencionar en el resto de esta guía. Dichos efectos se especifican exclusivamente a través de directivas. Esto quiere decir que los espacios, tabulaciones y retornos de carro que se introduzcan en el archivo fuente no tienen ningún efecto a la hora de la presentación final del documento. Por ejemplo, escribiendo:

```
Estas
    palabras
forman      una
    frase.
```

producimos exactamente lo mismo que con:

```
Estas palabras forman una frase.
```

A la hora de la verdad lo que se ve es:

```
Estas palabras forman una frase.
```

En resumen, la estructura básica de un documento HTML queda de la forma siguiente:

```
<html>
  <head>
    <title>Título</title>
  </head>
  <body>
    Texto del documento, menciones a gráficos, enlaces, etc.
  </body>
</html>
```

ESTILOS Y EFECTOS BÁSICOS

Como ya hemos dicho, la estructura lógica del texto y los diferentes efectos que se le apliquen se especifican mediante *directivas*. En este punto vamos a repasar algunas de las más importantes. En cada uno de los casos que veremos, primero se presenta el texto original HTML, es decir, lo que nosotros editamos, con las directivas situadas en los lugares adecuados; y después se presenta el efecto que dicho *texto fuente* produce una vez que se interpreta y se representa con el programa adecuado.

TÍTULOS

Mediante los títulos, en sus diferentes niveles de importancia, podemos definir el *esqueleto* del documento, su estructura básica.

<code><h1>Mucha importancia</h1></code>	Mucha importancia
<code><h2>Menos importancia</h2></code>	Menos importancia
<code><h3>Mucha menos importancia</h3></code>	Mucha menos importancia

ATRIBUTOS DEL TEXTO

Mediante estos atributos determinamos el estilo y el tipo de letra que tendrá la presentación del documento final.

El primero en el que nos deberíamos detener es el *texto normal* entendiendo como tal el que no tiene ninguna característica especial. Para definir un párrafo como *normal* no es necesario poner ninguna etiqueta. Lo único que hay que tener en cuenta, como ya se ha dicho antes, es que al presentar el documento se hace caso omiso de los espacios, tabulaciones y retornos de carro que se encuentren en el texto fuente. Por ello cuando se quiera forzar un final de línea es necesario utilizar dos directivas especiales: `<p>` para marcar un fin de párrafo, y `
` para un único retorno de carro. La diferencia entre ambas es que la separación de líneas que provoca `<p>` es algo mayor que la de `
`, para que los párrafos se distingan bien entre sí. Las dos directivas mencionadas se sitúan en el punto en que queremos poner la separación. Por ejemplo:

```
Este será un texto normal (párrafo 1, línea 1).<br>
El primer párrafo estará formado por 2 líneas (párrafo 1, línea 2).<p>
Este ya es el segundo párrafo (párrafo 2, línea 1).<p>
```

Por supuesto, estas dos etiquetas se puede aplicar donde queramos, no sólo en el texto normal.

El *texto preformateado* (etiqueta `<pre>`) se aplica cuando queremos que en la presentación final del documento se respeten los espacios y retornos de carro que hayamos puesto en el texto fuente. Además se utilizará un tipo de letra de espaciado fijo, parecido al de una máquina de escribir, más pequeño que el del texto normal. Este estilo de texto puede ser adecuado, por ejemplo, para una tabla numérica sencilla:

```
<pre>
Texto preformateado
-----
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
-----
</pre>
```

Para hacer una cita textual dentro de nuestro documento, se puede utilizar la directiva **<blockquote>**:

```
<blockquote>Muchos años después, frente al pelotón de fusilamiento,
el coronel Aureliano Buendía había de recordar aquella tarde remota
en que su padre lo llevó a conocer el hielo.<br>
(Gabriel García Márquez, Cien años de soledad)</blockquote>
```

Las *direcciones* de correo electrónico se suelen marcar con esta directiva:

```
<address>Dirección: webmaster@etsit.upm.es</address>
```

Se pueden dar también los atributos más tradicionales: negrita y cursiva:

```
<b>Esto en negrita</b> y <i>esto en cursiva</i>
```

Se puede utilizar un tipo de letra similar al de una máquina de escribir:

```
<tt>Máquina de escribir</tt>
```

Para centrar texto (o, en general, cualquier cosa: un gráfico, por ejemplo) se usa la directiva **<center>**:

```
<center>Este texto va centrado </center>
```

LISTAS

Las listas se definen de forma muy sencilla: se dice dónde empieza la lista, dónde empieza cada punto y dónde acaba la lista. Las etiquetas que se utilicen en cada caso deben aparecer al principio de línea, o al menos sin texto por delante (sólo espacios o tabulaciones).

Podemos recurrir a tres tipos distintos de listas, cada una con una presentación diferente: no numeradas, numeradas y listas de definiciones (glosarios).

Las listas se pueden anidar, es decir, en el lugar donde debería ir uno de los términos de la lista se pone una nueva lista, que por supuesto no tiene porqué ser del mismo tipo.

Esto es una lista no numerada:

```
<ul>
<li>Tomates
<li>Zanahorias
<li>Puerros
</ul>
```

- Tomates
- Zanahorias
- Puerros

Esto una lista numerada:

```
<ol>
<li>Miguel Induráin
<li>Tony Rominger
<li>Eugeni Berzin
</ol>
```

1. Miguel Induráin
2. Tony Rominger
3. Eugeni Berzin

VARIOS

La directiva `<hr>` sitúa en el documento una línea horizontal de separación. En este documento, por ejemplo, se han utilizado líneas horizontales para separar las diferentes secciones:

```
<hr>
```

Para poner un comentario en un documento HTML, es decir, una aclaración que no aparece en la presentación final del documento, se encierra el texto que formará el comentario entre los símbolos `<!--` y `-->`. Por ejemplo, un caso típico podría ser:

```
<!-- Modificado por José el viernes 2 de julio -->
```

ENLACES Y GRÁFICOS

INTRODUCCIÓN

Además de los muchos estilos y capacidades de presentación que nos ofrece HTML para estructurar el documento en sí, disponemos de varias directivas que nos permiten definir relaciones entre diferentes documentos y estructurar todo un conjunto de documentos para crear una unidad lógica. La facilidad para definir este tipo de enlaces es una de las razones de la potencia y versatilidad de HTML. Por la similitud de tratamiento que tienen los enlaces y los gráficos, tocaremos también en esta sección cómo pueden incluirse estos últimos en un documento.

Los enlaces en HTML se expresan rodeando con la directiva `<a>` el objeto (que puede ser un fragmento de texto o un gráfico) que vaya a servir como *anclaje* para el enlace. Por ejemplo, si marcamos con `<a>` un gráfico, cuando en el documento final se pulse con el ratón sobre dicho gráfico *saltaremos* al objeto referenciado en el enlace: otro documento, un vídeo musical, o un servidor de información meteorológica.

QUÉ ES UN URL

Para especificar de manera uniforme el objeto al que apunta nuestro enlace, se utiliza una forma estandarizada que se denomina URL (*Uniform Resource Locator*, es decir, *Localizador Uniforme de Recursos*). Un URL está formado de la siguiente manera: *esquema://maquina/ruta*

La *máquina* y la *ruta* sirven para localizar el objeto al que apunta nuestro enlace. La *máquina* es la identificación del servidor en el cual está situado el objeto al que apunta el enlace. Puede ser simplemente el nombre de un ordenador (como `www.etsit.upm.es`) o también un nombre y un puerto (por ejemplo `www.etsit.upm.es:8000`).

La *ruta* es el nombre del archivo que contiene el documento en concreto, incluyendo el nombre del subdirectorio en el que se encuentra.

ENLACES

Para definir un enlace es necesario marcar con la directiva `<a>` el objeto del cual va a partir dicho enlace. Dicha directiva debe incluir el parámetro `href="URL"` para especificar el destino del enlace. Es decir, que antes del objeto elegido debemos abrir con ``, y después cerrar con ``. Por ejemplo, si queremos que el texto *pulse aquí para visitar la NASA* nos conduzca a la *home page* de la NASA, debemos escribir en nuestro texto HTML:

```
<a href="http://www.nasa.gov/">Pulse aquí para visitar a la NASA</a>
```


Si lo que queremos es solamente enlazar con otro documento que se encuentra en el mismo servidor, incluso en el mismo subdirectorío. En este caso no es necesario escribir el *camino completo* al destino del enlace, sino que basta con dar la mínima información imprescindible.

Es decir, que si desde cierto documento queremos enlazar con otro que se encuentra en el mismo subdirectorío, basta con poner su nombre: `pulse aquí`. O si se encuentra en otro subdirectorío del mismo servidor, es suficiente con poner `pulse aquí`. También pueden utilizarse rutas relativas: `cosa`.

Imágenes

Para incluir un gráfico en un documento HTML se utiliza la directiva ``. En dicha directiva debe incluirse un parámetro `src="URL"`, con el cual indicamos dónde está el archivo con el gráfico concreto que queremos para nuestro documento. Esto pone a nuestra disposición una gran flexibilidad, ya que podemos complementar el contenido de nuestro documento tanto con gráficos que se encuentren disponibles en nuestro servidor de WWW como con una foto situada en un servidor de la NASA o del Ministerio de Cultura, por ejemplo, sin que el lector final tenga por qué apreciar ninguna diferencia.

Existe alguna limitación respecto a los formatos gráficos que los programas lectores de HTML puede interpretar sin problemas. Los formatos fundamentales son el GIF y el JPG, que cualquier programa con capacidades gráficas debería poder mostrar directamente en nuestro texto. Si utilizamos otro formato diferente, lo más probable es que cuando un lector esté accediendo al documento, el programa no comprenda ese formato y se tenga que solicitar la *ayuda* de otro programa, con lo cual al final el gráfico no se insertará en el lugar estratégico de nuestro documento, sino que aparecerá en otra ventana diferente.

Hay un parámetro optativo de la directiva `` que sirve para proponer un texto alternativo a un gráfico. Este texto aparecerá cuando se esté usando para leer el HTML un programa sin capacidades gráficas (por ejemplo una consola de Lynx, que sólo trabaja con texto). Se trata de `alt="texto"`. Conviene utilizarlo cuando los gráficos sirven como origen a *hiperenlaces*, porque si no los programas sin capacidades gráficas no podrían mostrar los enlaces que nosotros queremos.

Como ocurría antes con los enlaces, por lo general no es necesario escribir el URL completo, sino que basta con dar la mínima información. Por ejemplo, para colocar en este punto del documento un monigote que está en el mismo subdirectorío que este manual, en el archivo `monigote.gif`, escribiremos:

```
<p>
```

Lo que se traduce en:



Como se ve, hemos especificado el texto alternativo "MONIGOTE", con lo cual una persona que no dispusiera del programa adecuado hubiera visto algo parecido a [MONIGOTE] en lugar del dibujo.

Podemos también incluir un dibujo que esté en otro lugar especificando un URL completo, por ejemplo:

```
 <p>
```

Y además podemos hacer que un gráfico sea un enlace, utilizando la directiva <a>. En este caso no debemos olvidar utilizar la opción `alt="texto"` para que todos los usuarios puedan seguir el enlace:

```
<a href="http://www.nasa.gov/">  
    
</a> <p>
```

CARACTERES ESPECIALES

Existen ciertas exigencias de HTML respecto al uso de caracteres especiales, denominación que incluye a las vocales acentuadas y a la letra ñe. Existen también ciertas limitaciones relativas al uso de ciertos símbolos que significan algo en HTML, como el de *menor que* (<) o el signo inglés de *and* (llamado *ampersand*: &).

Trataremos primero el caso más sencillo. Existe una razón evidente que impide que podamos escribir ciertos símbolos directamente en un texto HTML, como por ejemplo el <: dichos símbolos tienen un significado en HTML, y es necesario diferenciar claramente cuándo poseen ese significado y cuándo queremos que aparezcan literalmente en el documento final. Por ejemplo, como ya sabemos, < indica el comienzo de una directiva, y, por ello, si queremos que aparezca en el texto como tal tendremos que *dar un rodeo* escribiendo algo que no de lugar a confusión, en este caso <. Los símbolos afectados por esta limitación, y la forma de escribirlos, se detallan a continuación:

- < (*Menor que*): <
- > (*Mayor que*): >
- & (símbolo de *and*, o *ampersand*): &
- " (*comillas dobles*): "

Es decir, que para escribir <"> en nuestro texto HTML original debemos poner <">.

El otro caso especial se da cuando en un texto HTML se quiere escribir una ñe, por ejemplo. Existen dos formas de hacerlo. La primera, que es a la que obliga el estándar de HTML, consiste en utilizar *entidades*, es decir, *palabrejas* como las que antes se presentaron para escribir ciertos símbolos. Las entidades comienzan siempre con el símbolo &, y terminan con un punto y coma (;). Entre medias va un identificador del carácter que queremos que se escriba. Las entidades necesarias en nuestro idioma son:

- á: á
- é: é
- í: í
- ó: ó
- ú: ú
- Á: Á
- É: É
- Í: Í
- Ó: Ó
- Ú: Ú
- ü: ü
- Û: Ü
- ñ: ñ
- Ñ: Ñ

- ÿ: ¿
- ÿ: ¡

Como puede verse, las vocales acentuadas se identifican añadiendo el sufijo *acute* a la vocal sin acentuar (puesto que se trata de un acento agudo). Para la u con diéresis y la eñe se usan *uml* tras una u y *tilde* detrás una ene, respectivamente. La equivalencia de los signos de abrir interrogación y exclamación es algo más oscura: a falta de una denominación más evidente, tenemos que usar el valor numérico de dichos caracteres en el código estándar latin1 (ISO-8859-1). Esto se puede hacer con cualquier otro carácter del código latin1, que es el código de caracteres básico en HTML, escribiendo *&#numero;*.

La segunda manera, que sin duda es más cómoda, consiste en no preocuparse por esta limitación y escribir literalmente los caracteres afectados. A pesar de que este método suele funcionar en las conexiones WWW directas (porque el protocolo HTTP, que transporta el HTML por los vericuetos de Internet, requiere un canal de 8 bits), no tiene por qué funcionar bien cuando los documentos HTML se envían por correo electrónico, por ejemplo. Por tanto, y a pesar de los inconvenientes, es absolutamente recomendable respetar la norma especificada en HTML.

Tablas

Las tablas nos permiten representar cualquier elemento de nuestra página (texto, listas, imágenes, etc...) en diferentes filas y columnas separadas entre sí. Es una herramienta muy útil para "ordenar" contenidos de distintas partes de nuestra página. La tabla se define mediante la directiva `<TABLE></TABLE>`. Los parámetros opcionales de esta directiva son :

border = num.

Indica el ancho del borde de la tabla en puntos.

cellspacing = num

Indica el espacio en puntos que separa las celdas que están dentro de la tabla.

cellpadding = num

Indica el espacio en puntos que separa el borde de cada celda y el contenido de esta.

width = num ó %

Indica la anchura de la tabla en puntos o en porcentaje en función del ancho de la ventana del visor. Si no se indica este parámetro, el ancho se adecuará al tamaño de los contenidos de las celdas.

height = num ó %

Indica la altura de la tabla en puntos o en porcentaje en función del alto de la ventana del visor. Si no se indica este parámetro, la altura se adecuará a la altura de los contenidos de las celdas.

bgcolor = código de color

Especifica el color de fondo de toda la Tabla.

Para definir las celdas que componen la tabla se utilizan las directivas `<TD>` y `<TH>`.

`<TD>` indica una celda normal, y `<TH>` indica una celda de "cabecera", es decir, el contenido será resaltado en negrita y en un tamaño ligeramente superior al normal. Los parámetros opcionales de ambas directivas son :

align = LEFT / CENTER / RIGHT / JUSTIFY

Indica como se debe alinear el contenido de la celda, a la izquierda (LEFT), a la derecha (RIGHT), centrado (CENTER) o justificado (JUSTIFY).

valign = TOP / MIDDLE / BOTTOM

Indica la alineación vertical del contenido de la celda, en la parte superior (TOP), en la inferior (BOTTOM), o en el centro (MIDDLE).

rowspan = num

Indica el número de filas que ocupará la celda. Por defecto ocupa una sola fila.

colspan = num

Indica el número de columnas que ocupará la celda. Por defecto ocupa una sola columna.

width = num ó %

Indica la anchura de la columna en puntos o en porcentaje en función del ancho de la ventana del visor. Si no se indica este parámetro, el ancho se adecuará al tamaño de los contenidos.

bgcolor = código de color

Especifica el color de fondo del elemento de la Tabla.

Para indicar que acaba una fila de celdas se utiliza la directiva <TR>. A continuación mostraremos un ejemplo de una tabla que contiene solo texto. Como se indicó anteriormente el contenido de las celtas puede ser cualquier elemento de HTML, un texto, una imagen, un Hyperenlace, una Lista, etc...

Ejemplo :

```
<TABLE border = 4 cellpadding = 4 cellspacing = 4 width =80%>
  <TH align = center>Buscadores
  <TH align = center colspan = 2>Otros Links
<TR>
  <TD align = LEFT>Yahoo
  <TD align = LEFT>Microsoft
  <TD align = LEFT>IBM
<TR>
  <TD align = LEFT>Infoseek
  <TD align = LEFT>Apple
  <TD align = LEFT>Digital
</TABLE>
```

Se vería como:

Buscadores	Otros Links	
Yahoo	Microsoft	IBM
Infoseek	Apple	Digital

Las directivas <TD> y <TH> son cerradas según el estandar de HTML, es decir que un elemento de tabla <TD> debería cerrarse con un </TD> , sin embargo los visores asumen que un elemento de la tabla, queda automáticamente "cerrado" cuando se "abre" el siguiente.

Formularios

Los formularios nos permiten dentro de una página Web solicitar información al visitante y procesarla. En un formulario podremos solicitar diferentes datos (campos) cada uno de los cuales quedará asociado a una variable. Una vez se hayan introducido los valores en los campos, el contenido de estos será enviado a la dirección (URL) donde tengamos el programa que pueda procesar las variables. Para poder realizar este último paso de procesar las variables necesitaremos realizar un programa externo en algún lenguaje de programación como PERL, C++ o Visual Basic. A este programa externo se le suele llamar **CGI** (Common Gateway Interface). La creación de este tipo de programas sería tema de otro curso diferente, por lo que aquí veremos como se pueden enviar las variables a nuestra dirección de correo electrónico.

La declaración del formulario se pone entre las directivas `<FORM></FORM>`. En el interior de la declaración se indican los elementos (variables) de entrada. La directiva `<FORM>` tiene los parámetros **action** y **method**.

action = "programa"

Indica el programa que va a "tratar" a las variables que se envíen con el formulario. En nuestro caso enviaremos las variables por correo electrónico, con lo que el "programa" será "**mailto: direccion_de_correo**".

method = POST / GET

Indica el método según el que se transferiran las variables. POST produce la modificación del documento de destino (como en el caso de enviar por correo las variables). GET no produce cambios en el documento destino (como en el caso de una consulta a una base de datos, p.ej. una página de búsqueda en Internet).

Campos de Entrada

Para la introducción de las variables se utiliza la directiva `<INPUT>`. Esta directiva tiene el parámetro **type** que indica el tipo de variable a introducir y **name** que indica el nombre que se le dará al campo. Cada tipo de variable tiene sus propios parámetros.

type= text **name** = campo

Indica que el campo a introducir será un texto. Sus parámetros son :

maxlength = numero

Numero máximo de caracteres a introducir en el campo.

size = numero

Tamaño en caracteres que se mostrará en pantalla.

value = "texto"

Valor inicial del campo. Normalmente será " ", o sea, vacío.

type = password **name** = campo

Indica que el campo será una palabra de paso. Mostrará asteriscos (*) en lugar de las letras escritas.

Sus parámetros opcionales son los mismos que para text.

type = checkbox **name** = campo

El campo se elegirá marcando una casilla. Se permite marcar varias casillas. Los valores de las casillas serán indicados por :

value = "valor"

checked

La casilla aparecera marcada por defecto.

type = radio **name** = campo

El campo se elegira marcando una casilla. Solo permite marcar una sola de las casillas. Los valores de las casillas seran indicados por :

value = "valor"

type = image **name** = campo

El campo contendra el valor de las coordenadas del punto de la imagen pinchado.

Debe indicarse la imagen con el parámetro :

src = "archivo de imagen".

type = hidden **name** = campo

El usuario no puede modificar su valor, ya que el campo no es visible se manda siempre con el valor indicado por el parámetro :

value = "valor"

type = submit

Representa un botón. Al pulsar este botón la información de todos los campos se envía al programa indicado en **<FORM>**. Tiene el parámetro **value** = "texto" que indica el texto que aparecerá en el boton.

type = reset

Representa un botón. Al pulsar este botón se borra el contenido de todos los campos. El parámetro **value** = "texto" indica el texto que aparecerá en el botón.

Campos de Seleccion

Este tipo de campos despliegan una lista de opciones, entre las que debemos escoger una o varias. Se utiliza para ellos la directiva **<SELECT>** **</SELECT>** . Sus parámetros son :

name = campo

Nombre del campo

size = num

Numero de opciones visibles. Si se indica 1 se presenta como un menú desplegable, se se indica mas de uno se presenta como una lista con barra de desplazamiento.

multiple

Permite seleccionar mas de un valor para el campo.

Las diferentes opciones de la lista se indican con la directiva **<OPTION>**. Esta directiva puede incluir el parametro **selected** para indicar cual es la opcion por defecto. En caso de que no se especifique, se tomara por defecto la primera opción de la lista.

Areas de texto.

Representa un campo de texto de múltiples líneas. Normalmente se utiliza para que se incluyan en el comentarios. La directiva usada es **<TEXTAREA>** **</TEXTAREA>**, y sus parámetros :

name = campo

Nombre del campo.

cols = num.

Numero de columnas de texto visibles.

rows = num.

Numero de filas de texto visibles.

wrap = VIRTUAL / PHYSICAL

Justifica el texto automáticamente en el interior de la caja. La opción PHYSICAL envia las líneas de texto separadas en líneas físicas. La opción VIRTUAL envia todo el texto seguido.

Veamos a continuación un ejemplo de formulario utilizando todas las formas de introducción de datos.

```
<FORM action = "mailto: cenedic@ucol.mx" method = post >
Tu Nombre:<INPUT type = text name = nombre size = 30 >
Tu Clave: <INPUT type = password name = clave size = 8 >
<P>
Archivos a Enviar:
<INPUT type = checkbox name = archivo value = "Manual" > Manual de Html
<INPUT type = checkbox name = archivo value ="Mapthis" > Programa Mapthis
<INPUT type = checkbox name = archivo value ="Help" > Archivo de Ayuda
<P>
Tu Edad :
<INPUT type = radio name = edad value = "-20" > Menos de 20 años
<INPUT type = radio name = edad value = "20-40" > Entre 20 y 40 años
<INPUT type = radio name = edad value = "+40" > Mas de 40 años
<P>
<INPUT type = hidden name = lugar value = "pagina personal" >
Como encontraste mi página :
<SELECT name = donde >
<OPTION>De casualidad
<OPTION>Por el buscador Ole
<OPTION>Por el buscador Yahoo
<OPTION>Me la comentaron
</SELECT>
<P>
Tus Comentarios:
<BR>
<TEXTAREA name = comentario rows = 5 cols = 40 wrap = virtual
></TEXTAREA>
<P>
<INPUT type = submit value = "Enviar" >
<INPUT type = reset value = "Borrar" >
</FORM>
```

Ahora veamos el efecto producido en la página Web :

Tu Nombre:	<input type="text"/>	Tu Clave:	<input type="text"/>		
Archivos a Enviar:					
<input type="checkbox"/>	Manual de Html	<input type="checkbox"/>	Programa Mapthis	<input type="checkbox"/>	Archivo de Ayuda
Tu Edad :					
<input checked="" type="checkbox"/>	Menos de 20 años				
<input type="checkbox"/>	Entre 20 y 40 años				
<input type="checkbox"/>	Mas de 40 años				
Como encontraste mi página :					
<input type="text" value="Me la comentaron"/>					
Tus Comentarios:					
<div style="border: 1px solid black; height: 100px; width: 100%;"></div>					
<input type="button" value="Submit"/>		<input type="button" value="Borrar"/>			

Si rellenas este FORM y pulsas sobre el botón **Enviar**, (estando conectado a Internet), se generará un mensaje de correo a mi direccion de correo *luisfd@jet.es*. Si pulsas el boton **Borrar** se borrarán los datos que hayas introducido en el Formulario. El texto que se recibiría por correo electrónico sería parecido a este:

```
nombre=Pedro+Perez &clave=12345678 &archivo=Manual
&archivo=Mapthis &edad=20-40 &lugar=pagina+personal
&donde=Por+el+buscador+Ole &comentario%94=
Espero+que+me+mandes+los%0D%0Aarchivos+antes+del+martes%0D%0A%0D%0
AS aludos.%0D%0A
```

Podemos observar que en el correo se separan las variables con el símbolo **&**, los espacios se sustituyen por el signo **+** y se representan los códigos de retorno de carro y avance de línea del campo de texto con los caracteres **%0D** y **%0A** respectivamente.

Si en vez de enviar estas variables por correo electrónico, fuesen enviadas a un programa (CGI), este programa podría tratarlas y dar como respuesta una nueva página Web.

IMPORTANTE: La opción **mailto:** en el parámetro **action** de la directiva **FORM** solo funciona correctamente con aquellos navegadores que tengan incorporado un programa de correo.

Se puede, además, utilizar un Formulario como "lanzador" de links, es decir, por medio de una lista desplegable permitir al usuario escoger un Link y acceder a el. Para ello se utiliza el parámetro "OnClick". Ejemplo :

```
<FORM>
<SELECT name = "list" >
<OPTION SELECTED value="http://web.ucol.mx/usuario2">Mi Pagina Web
<OPTION value="http://www.microsoft.com">Microsoft
<OPTION value="http://www.ibm.com">Ibm
<OPTION value="http://www.novell.com">Novell
<OPTION value="http://www.hp.com">Hewlett Packard
<OPTION value="http://www.fujitsu.com">Fujitsu
<OPTION value="http://www.3com.com">3Com
</SELECT>
<INPUT TYPE=BUTTON value="Ir a..."
onClick="top.location.href=this.form.list.options[this.form.list.selectedIndex].value">
</FORM>
```

Este seria el efecto producido:

A screenshot of a web browser showing a dropdown menu. The menu is open, and the text '3Com' is visible in the input field. The dropdown arrow is pointing downwards.

Atributos del Texto

Para indicar atributos del texto (negrilla, subrayado, etc.) tenemos varias directivas. Algunas de ellas no son reconocidas por determinados visores de Internet, es por ello que según el visor que este usted utilizando, verá el resultado correctamente o no.

Atributo	Directiva	Ejemplo
Negrilla		Texto de prueba
Cursiva	<I></I>	<I>Texto de prueba</I>
Teletype	<TT></TT>	<TT>Texto de prueba</TT>
Subrayado	<U></U>	<U>Texto de prueba</U>
Tachado	<S></S>	<S>Texto de prueba</S>
Parpadeo	<BLINK></BLINK>	<BLINK>Texto de prueba</BLINK>
Superíndice		^{Texto de prueba}
Subíndice		_{Texto de prueba}
Centrado	<CENTER></CENTER>	<CENTER>Texto de prueba</CENTER>

Por otro lado la directiva nos permite variar el tamaño, el color, y el tipo de letra de un texto determinado. Utiliza para ello los parámetros **size**, **bgcolor** y **face**.

size = valor

Da al texto un tamaño en puntos determinado.

size = +/- valor

Da al texto un tamaño tantas veces superior (+) o inferior (-) como indique el valor.

color = "código de color"

Escribe el texto en el color cuyo código se especifica.

face = "nombre de la fuente o tipo de letra"

Escribe el texto en el tipo de letra especificado. Si este tipo de letra no existe en el ordenador que "lee" la página se usará el predeterminado del navegador.

Ejemplo	Se vería como
 Texto de prueba 	Texto de prueba

Existen otras directivas que realizan las mismas operaciones que las antes vistas en los atributos del texto.

Directiva	Hace lo mismo que
	

<code><CITE></CITE></code>	<code><I></I></code>
<code><STRIKE></STRIKE></code>	<code><S></S></code>

Marquee

La directiva `<MARQUEE></MARQUEE>` crea una marquesina con un texto en su interior que se desplaza. Funciona únicamente con MS-Explorer. Sus parámetros son los siguientes:

align = top / middle / bottom

Indica si el texto del interior de la marquesina se alinea en la zona alta (top), en la baja (bottom) o en el centro (middle) de la misma.

bgcolor = "codigo de color"

Indica el color del fondo de la marquesina.

direction = left / right

Indica hacia que lugar se desplaza el texto, hacia la izquierda (left) o hacia la derecha (right)

height = num o %

Indica la altura de la marquesina en puntos o porcentaje en función de la ventana del visor.

width = num o %

Indica la anchura de la marquesina en puntos o porcentaje en función de la ventana del visor.

loop = num / infinite

Indica el numero de veces que se desplazará el texto por la marquesina. Si se indica infinite, se desplazará indefinidamente.

scrolldelay = num.

Indica el número de milisegundos que tarda en reescribirse el texto por la marquesina, a mayor número mas lentamente se desplazará el texto.

Veamos un ejemplo de esta directiva :

```
<MARQUEE bgcolor = "#FFFFFF" width = 50% scrolldelay = 0 >
Bienvenido a mi pagina personal en Internet.
</MARQUEE>
```

Sonido de fondo

Nuestra página Web puede tener un sonido que se active al entrar en la página. Esta característica de MS Explorer utiliza la directiva `<BGSOUND>` y tiene los siguientes parámetros :

src = "archivo"

Indica el nombre del archivo que contiene el sonido (.wav, .mid).

loop = num / infinite

Indica el número de veces que se reproducirá el sonido. Si se indica infinite, el sonido se reproducirá de forma continua hasta que abandonemos la página.

Un ejemplo de esta directiva sería :

```
<BGSOUND src= "yesterday.mid" loop= infinite>
```

Para utilizar esta función en Netscape se utiliza la directiva <EMBED>. Esta directiva se utiliza realmente para "incrustar" un objeto en nuestra pagina Web. Dicho objeto puede ser un archivo de sonido, un video, un grafico BMP, etc... Tiene los siguientes parámetros :

src = "archivo"

Indica el nombre del archivo que contiene el sonido (.wav, .mid) o el video (.avi).

autostart = true

Incluirlo si deseamos que la reproducción se inicie inmediatamente.

loop = true

Incluirlo si deseamos que la reproducción no se detenga. (al terminar, vuelve a comenzar automáticamente).

volume = numero

Volumen al que se reproducen los archivos de sonido.

width = numero

height = numero

Anchura y Altura de la representación del objeto. (Si es un sonido no es necesario este parámetro).

controls = smallconsole

Visualiza una serie de controles que nos permiten iniciar la reproducción del archivo, así como realizar una pausa o detenerlo.

Un ejemplo de esta directiva sería:

```
<EMBED src= "yesterday.mid" loop= true autostart= true  
volume=50 width=50 height=15 controls=smallconsole>
```

Frames

Las frames es una técnica para subdividir la pantalla del visor en diferentes ventanas. Cada una de estas ventanas se podrá manipular por separado, permitiéndonos mostrar en cada una de ellas una página Web diferente. Esto es muy útil para, por ejemplo, mostrar permanentemente en una ventana los diferentes contenidos de nuestra página, y en otra ventana mostrar el contenido seleccionado.

Para definir las diferentes subventanas o frames se utilizan las directivas `<FRAMESET>` `</FRAMESET>` y `<FRAME>`. La directiva `<FRAMESET>` indica como se va a dividir la ventana principal. Pueden incluirse varias directivas `<FRAMESET>` anidadas con el objeto de subdividir una subdivisión. Los parámetros de `<FRAMESET>` son **rows** y **cols** en función de si la división de la pantalla se realiza por filas (rows) o columnas (cols). Los parámetros **rows** y **cols** se acompañan de un grupo de números que indican en puntos o en porcentaje el tamaño de cada una de las subventanas. En caso de utilizar **rows** los tamaños de las subventanas se entienden indicados de arriba a abajo, es decir, el primer valor será el asignado a la ventana superior, el segundo a la ventana inmediatamente inferior, etc. En el caso de **cols** los tamaños se aplican de izquierda a derecha.

Ejemplos	Resultado
<code><FRAMESET rows = "25%,50%,25%"></code>	Crea tres subventanas horizontales, la primera ocupará un 20% de la ventana principal, la segunda un 50% y la tercera un 25%.
<code><FRAMESET cols = "120,*,100"></code>	Crea tres subventanas verticales, la primera y la tercera tendrán un "ancho" fijo de 120 y 100 puntos respectivamente. La segunda ocupará el resto de la ventana principal (*).
<code><FRAMESET cols = "15%,*"></code> <code><FRAMESET rows = 20%,*"></code>	En este caso "anidamos" dos directivas. La primera divide la ventana principal en dos subventanas verticales, la primera ocupa un 15% de la ventana principal y la segunda el resto. La segunda directiva vuelve a subdividir la primera subventana creada anteriormente, pero esta vez en dos subventanas horizontales, la superior ocupará un 20% de la subventana, y la inferior el resto.

La directiva `<FRAME>` indica las propiedades de cada subventana. Es necesario indicar una directiva `<FRAME>` para cada subventana creada. Los parámetros de `<FRAME>` son:

name = "nombre"

Indica el nombre por el que nos referiremos a esa subventana.

src = "URL"

La ventana mostrará en principio el contenido del documento HTML que se indique.

marginwidth = num.

Indica el margen izquierdo y derecho de la subventana en puntos.

marginheight = num

Indica el margen superior e inferior de la subventana en puntos.

scrolling = "yes / no / auto"

Indica si se aplica una barra de desplazamiento a la subventana en el caso de que la página que se cargue en ella no quepa en los límites de la subventana. el valor "yes" muestra siempre la barra de desplazamiento, "no" no la muestra nunca (la zona de la página que no quepa en la subventana no la veremos), y "auto" la muestra solo en caso de que sea necesario para poder ver la página.

noresize

Si se indica este parámetro, el usuario no podrá "redimensionar" las subventanas con el visor. Un usuario que este viendo una pagina con frames puede redimensionarlas seleccionando un borde de la subventana con el cursor y desplazándolo.

border = num.

Indica el "borde" que separara esta frame de la siguiente. Si se indica cero (0) no se mostrara borde entre las frames, consiguiendo un efecto muy elegante, siempre y cuando el "fondo" de todas las frames sea el mismo, o sean colores solidos.

Los visores que no soportan la característica de subventanas, no mostrarán nada de lo indicado con estas directivas. Es por ello que existe una directiva llamada **<NOFRAMES>** **</NOFRAMES>**. Todo lo indicado entre esta directiva será lo que muestren los visores sin capacidad para visualizar Frames. Los visores que soporten Frames obviaran las directivas incluidas entre **<NOFRAMES>** **</NOFRAMES>**.

Si su visor puede visualizar Frames tendrá en estos momentos la pantalla dividida en dos zonas independientes, en la izquierda verá el índice de materias de este manual, y en la derecha estará viendo la sección del manual que haya seleccionado. En otro caso simplemente verá la sección del manual.

La definición de las Frames debe ir antes de la definición del cuerpo de documento (**<BODY>**).

Veamos un ejemplo completo de Frames:

```
<HTML>
<HEAD>
<TITLE>Pagina con Frames</TITLE>
</HEAD>
<FRAMESET cols = "15%,*">
<!-- Dos subventanas verticales, la de la izquierda con un 15% de la pantalla, la de la
derecha el resto. -->
<FRAMESET rows = "35%,*">
<!-- Dos subventanas horizontales dentro de la subventana izquierda. -->
  <FRAME name = "upd" src = "update.htm" scrolling = "auto">
  <!-- La subventana horizontal superior izquierda con el nombre "upd" y muestro el
documento update.htm -->
  <FRAME name = "menu" src = "menu1.htm" scrolling = "auto">
  <!-- La subventana horizontal inferior izquierda con el nombre "menu" y muestro el
documento menu1.htm -->
</FRAMESET>
```



```

<!--Se cierra la definicion de las subventanas horizontales de la ventana de la izquierda -->
  <FRAME name = "home" src = "home.htm" scrolling = "auto">
  <!-- Llamo a la subventana vertical derecha con el nombre "home" y muestro el
  documento home.htm -->
  <NOFRAMES>
  <!-- Indico las órdenes para aquellos visores que no soporten Frames -->
  <BODY>
  SU VISOR NO MUESTRA FRAMES. Pulse <A href= "home.htm"> AQUI </A> para ir a la página
  sin Frames.
  </BODY>
  </NOFRAMES>
</FRAMESET>
<!-- Cierro la definición de las subventanas verticales -->
</HTML>

```

Al usar Frames, nos encontramos con un problema. Cuando queramos mostrar una página Web debemos indicarle al visor en que subventana queremos que se muestre. Por defecto se mostrará en la ventana donde se encuentre el enlace. Para poder escoger la subventana de destino del Hyperenlace se añade un nuevo parámetro a la directiva . Este parámetro se llama **target** y puede tener los siguientes valores:

target = "nombre_ventana"

Muestra el Hyperenlace en la ventana cuyo nombre se indica.

target = "_blank"

Abre una nueva copia del visor y muestra el Hyperenlace en ella. (Si usamos Netscape esto provoca que tengamos funcionando dos copias del programa).

target = "_self"

Se muestra el Hyperenlace en la subventana activa.

target = "_parent"

El Hyperenlace se muestra en el <FRAMESET> definido anteriormente al actual. Si no hay ningún <FRAMESET> anterior se muestra a pantalla completa suprimiendo todas las subventanas de la pantalla.

target = "_top"

Suprime todas las subventanas de la pantalla y muestra el Hyperenlace a pantalla completa.

Ejemplos	Resultado
	Muestra la pagina de Microsoft en una nueva copia del visor
	Muestra la pagina de IBM en la subventanna llamada "home"
	Muestra la pagina del buscador Yahoo a pantalla completa

Elementos de la Cabecera <HEAD>

El único elemento nuevo es el `STYLE`, que hace referencia a la hoja de estilos a emplear para visualizar el documento.

Por ejemplo:

```
<HEAD>
<STYLE HREF="../estilos/estilo1.css">
...
</HEAD>
```

Indica el **URL** de la hoja de estilos que se debe cargar y usar. La extensión CSS se refiere a uno de los métodos en desarrollo para representar las hojas de estilos. Hace poco la **W3 Organization** ha llegado a un acuerdo con muchas de las empresas que desarrollan productos para la **WWW** para definir el estándar de las hojas de estilos, refinando la propuesta del formato *CSS (Cascading Style Sheets)*.

Qué son las hojas de estilo

A menudo queremos que las páginas de nuestra web tengan un estilo común, la misma fuente, el mismo estilo de párrafos, etc. Para hacer esto podemos repetir una y otra vez estos estilos o podemos englobarlos todos en una única instrucción.

Si tomamos como ejemplo este manual veremos que la fuente es del tipo Arial, y la de los títulos y actividades es Comic Sans Serif. Además las actividades van dentro de un recuadro, los vínculos no están subrallados y cambian de color al pasar sobre ellos. Para conseguir este estilo propio y común a todas las páginas he creado una *hoja de estilo en cascada*, *cascade style sheets* en inglés, donde defino el formato de cada una de las etiquetas que me interesan, y en cada página hago una llamada a esta hoja de estilos para que aplique el formato sin tener que repetirlo una y otra vez.

En esta lección se introducen algunas de las propiedades más comunes de las hojas de estilo en cascada según el convenio CSS2, es decir, la versión 2 de las hojas de estilo en cascada cuyas iniciales en inglés son CSS como hemos visto más arriba.

Definir el estilo de una etiqueta

Podemos definir un estilo dentro de una etiqueta de modo que sólo afecte a ésta mediante el atributo `style`.

Este atributo es válido para todas las etiquetas de Html y su formato es el siguiente:

```
<etiqueta style="propiedad: valor;">Texto</etiqueta>
```

Veamos un ejemplo:

Código	Resultado
<pre><p style="color: red;">Texto Rojo</p></pre>	Texto Rojo

Como puedes observar la estructura es muy sencilla: dentro del atributo `style` escribimos la propiedad (`color`), dos puntos, el valor (`red`) y un punto y coma. Si quisiéramos podríamos añadir tantas propiedades como deseemos, todas ellas separadas por un punto y coma.

Nota: Esta es la forma correcta de aplicar colores, fuentes y tamaños al texto de nuestras páginas en Html 4 y no utilizar la etiqueta `` que vimos anteriormente.

A continuación tienes una pequeña tabla con propiedades de la fuente y la alineación del párrafo.

Propiedad	Valor	Resultado
<code>font-style</code>	Normal	Normal
	Italic	<i>Itálica</i>
<code>font-weight</code>	Normal	Normal
	Bold	Negrita
	Número de 100 a 900	Grueso 900
<code>font-family</code>	serif	Familia serif
	sans-serif	Familia sans-serif
	cursive	Familia cursive
	monospace	Familia monospace
	fantasy	FAMILIA FANTASY
	Nombre de la fuente, p.e. wingdings	☛☘☙☚☛☙☛
<code>font-size</code>	Tamaño en píxeles (px), puntos tipográficos (pt), centímetros (cm), milímetros (mm), pulgadas (in), etc., p.e. 30px	30 píxeles
<code>Color</code>	Nombre del color en inglés, p.e. red	Texto Rojo

	Color en hexadecimal RGB, p.e. #AA00AA	Texto violeta
background-color	Nombre del color de fondo en inglés, p.e. red	fondo Rojo
	Color de fondo en hexadecimal RGB, p.e. #AA00AA	Fondo violeta
text-align	left, alinear a la izquierda	Texto a la izquierda
	right, alinear a la derecha	Texto a la derecha
	center, alinear al centro	Texto centrado
	justify, alinear a la derecha	Texto justificado a izquierda y derecha
	Nota: Esta es una propiedad sólo para etiquetas de párrafo como <p>, <h1> o <blockquote>, y por tanto no funcionará con etiquetas de frases como , , etc.	

Estas propiedades puedes aplicarlas a todas las etiquetas que ya conoces que, de una manera u otra, dan formato a un texto de tu página. Así podemos cambiar las propiedades de <p>, <body>, <h1>, <blockquote>, , etc.

A menudo querrás dar formato a un párrafo o a un texto, pero no con una etiqueta en particular sino con una más genérica. Para ello existen dos etiquetas sobre las que podemos aplicar estos estilos. Estas etiquetas son <div> para dar formato a un párrafo y para dar formato a un texto.

Por ejemplo:

```
<span style="color: red;">Texto rojo</span>
```

Definir el estilo de una página

Si queremos que todos los títulos de nuestra página, definidos con una etiqueta del tipo <h1>, sean de color azul, podemos indicarlo en cada etiqueta como acabamos de aprender o podemos definirlo para toda la página en el encabezado de ésta, es decir, dentro de la etiqueta <head>.

Para definir estos estilos comunes a toda la página se utiliza la etiqueta <style> de la siguiente manera:

```
<html>
  <head>
    <title>Mi página web</title>
    <style type="text/css">
```

```
    ...
  </style>
</head>
<body>
  ...
</body>
</html>
```

El atributo `type="text/css"` indica al programa que se trata de una hoja de estilo (*Cascade Style Sheet*).

Dentro de la etiqueta `<style>` escribiremos las distintas etiquetas y entre llaves sus propiedades según vimos en el apartado anterior. Veamos un ejemplo:

```
<style type="text/css">
  body {
    background: yellow;
    font-family: monospace;
  }
  h1,h2,h3,h4,h5 {
    color: navy;
  }
</style>
```

Es posible también definir estilos diferentes para la misma etiqueta. Observa el siguiente ejemplo:

```
<style type="text/css">
  p {font-family: serif;}
  p.azul {color: navy;}
</style>
```

En el ejemplo anterior hemos definido el tipo de fuente de la etiqueta `<p>` como `serif`, pero hemos definido una "subetiqueta" o clase de `<p>` llamada `azul` y que tendrá color azul marino.

Para indicar que un párrafo definido por `<p>` es de la clase `azul` lo indicaremos con el atributo `class="azul"`. Observa como queda el código y el resultado final:

Código	<pre><p>Este es un párrafo normal</p> <p class="azul">...y este azul marino.</p></pre>
Resultado	<pre>Este es un párrafo normal ...y este azul marino.</pre>

También es posible definir una clase genérica que podrá ser utilizada dentro de cualquier etiqueta. Por ejemplo:

```
<style type="text/css">
  .centrado {text-align: center;}
</style>
```

Con esta clase podemos centrar cualquier texto definido por una etiqueta de bloque escribiendo, por ejemplo `<p class="centrado">` o `<h1 class="centrado">`.

Crear una hoja de estilo externa

Por lo general, al diseñar una web queremos que todas las páginas tengan el mismo estilo para dar mayor uniformidad y un aspecto personalizado a nuestro gusto. Podemos entonces repetir en el encabezado de cada una de las páginas el estilo de cada etiqueta, lo cual sería muy tedioso, o podemos escribir estos estilos en un archivo aparte y llamarla desde cada página.

El contenido de este archivo es simplemente lo que escribíamos dentro de la etiqueta `<style>` en el apartado anterior, y lo guardaremos con la extensión `.css`.

Por ejemplo, creamos con el bloc de notas un archivo nuevo llamado `miestilo.css` y cuyo contenido es el siguiente:

```
body {
  background-color: yellow;
  font-family: monospace;
}
h1,h2,h3,h4,h5 {
  color: navy;
}
```

Una vez creada la hoja de estilo y guardado el archivo lo llamaremos escribiendo la siguiente etiqueta dentro del encabezado, donde antes pusimos la etiqueta `<style>`:

```
<link href="miestilo.css" rel="stylesheet" type="text/css">
```

Y en definitiva la estructura final de la página quedará de la siguiente manera:

```
<html>
  <head>
    <title>Mi página web</title>
    <link href="miestilo.css" rel="stylesheet"
type="text/css">
```

```
</head>  
<body>  
  ...  
</body>  
</html>
```

Programas escritos en JavaScript.

Además de leerse las especificaciones de JavaScript para realizar programas en Java, existe una forma más fácil de incluir estos scripts en nuestra página. Si estamos viendo una página que tiene algún efecto especial que nos guste, simplemente hemos de visualizar el código fuente de la misma (en Netscape opción View - Document Source), marcar con el ratón todo el bloque contenido entre las directivas `<SCRIPT>` `</SCRIPT>`, copiarlas al portapapeles con CTRL+C y posteriormente incluir este bloque en nuestra página modificando los literales o frases animadas. Estas frases serán fáciles de localizar porque contendrán la palabra `var xx= "....."` antecediéndolas.

Sirva como ejemplo este código completo de una función JavaScript muy utilizada, que muestra varios mensajes desplazándose por la barra inferior del visor de páginas Web. Debemos modificar los literales indicados en `var m1`, `var m2` y `var m3` para adecuarlos al texto que queramos mostrar. Si se desea incluir más mensajes hay que añadir más líneas del tipo `var mx = "texto "` y "sumar" dichas variables en el apartado `var msg = .`

IMPORTANTE: Nótese que el código JAVASCRIPT se encuentra en la sección de "cabecera" (`<HEAD>`), y que la directiva `<BODY>` incluye el parámetro `onload` para que se comience a ejecutar el código JavaScript en el momento de cargar la página.

```
<HTML>
<HEAD>
<SCRIPT language="JavaScript">
<!-- Ocultar el código para los viejos visores html --
function scrollit(seed)
{
  var m1 = " Texto 1 "
  var m2 = " Texto 2 "
  var m3 = " Texto 3 "
  var msg=m1+m2+m3;
  var out = " ";
  var c = 0;
  if (seed > 100)
  {
    seed--;
    var cmd="scrollit(" + seed + ")";
    timerTwo=window.setTimeout(cmd,7);
  }
  else
  if (seed <= 100 && seed > 0)
  {
    for (c=0 ; c < seed ; c++)
    {
      out+=" ";
    }
    out+=msg;
    seed--;
    var cmd="scrollit(" + seed + ")";
```



```
        window.status=out;
        timerTwo=window.setTimeout(cmd,7);
    }
else if (seed <= 0)
    {
    if (-seed < msg.length)
        {
        out+=msg.substring(-seed,msg.length);
        seed--;
        var cmd="scrollit(" + seed + ")";
        window.status=out;
        timerTwo=window.setTimeout(cmd,7);
        }
    else
        {
        window.status=" ";
        timerTwo=window.setTimeout("scrollit(100)",75);
        }
    }
}
// --Final del codigo oculto -->
</SCRIPT>
<TITLE></TITLE>
</HEAD>
<BODY onload="timerONE=window.setTimeout('scrollit(100)',500)">

    Aquí va el contenido de tu página

</BODY>
</HTML>
```

