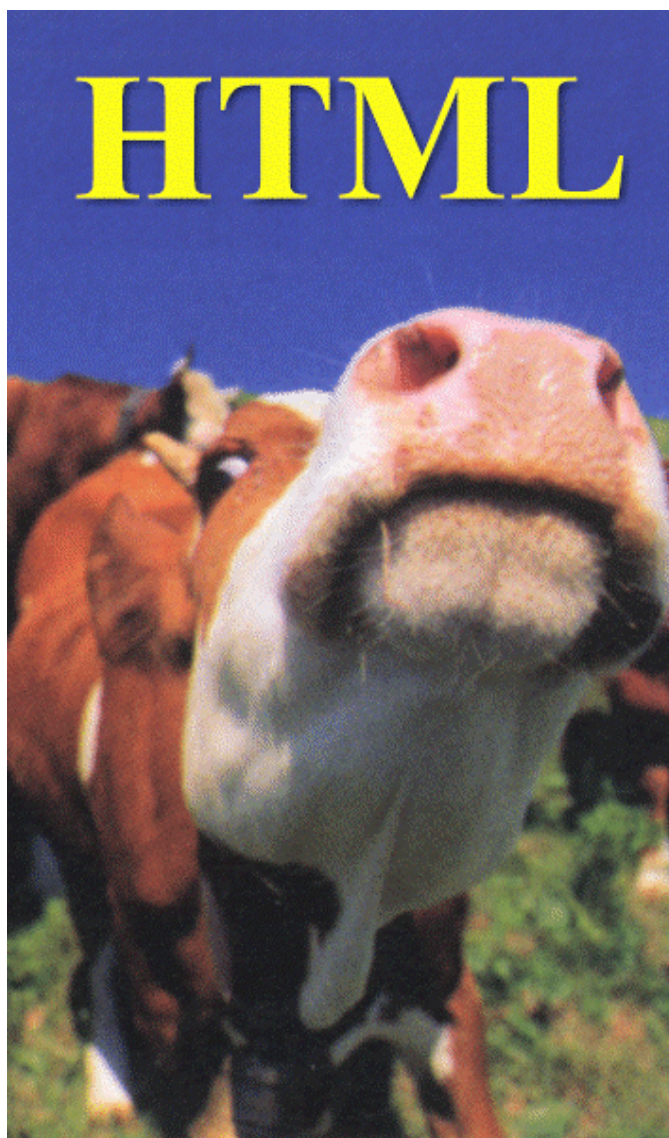


Este texto está diseñado para enseñar a confeccionar páginas web a través del uso directo del lenguaje HTML.

El objetivo es adiestrar al lector en las técnicas para confeccionar páginas web para Internet/Intranet, así como enseñar los fundamentos sintácticos del lenguaje HTML.

Los requisitos previos son: estar familiarizado con los aspectos básicos de Internet/Intranet, estar familiarizado con la navegación por la web, conocer el sistema operativo Windows a nivel de usuario.



LENGUAJE HTML



Índice

ÍNDICE.....	5
INTRODUCCIÓN AL HTML	7
¿QUÉ ES HTML?.....	7
REGLAS DE FORMATO	8
ESTRUCTURA DE UN DOCUMENTO HTML	9
NUESTRA PRIMERA PÁGINA HTML	9
FORMA DEL URL	10
<i>Localizando un documento hipertexto (http):</i>	<i>10</i>
<i>Localización de un fichero por el servicio FTP (ftp:File Transfer Protocol)</i>	<i>11</i>
<i>Accediendo a un grupo de noticias (news):</i>	<i>11</i>
<i>Accediendo a una dirección e-mail (mailto):</i>	<i>11</i>
<i>Localización de un archivo cualquiera (file):</i>	<i>11</i>
CABECERA Y CUERPO DEL DOCUMENTO HTML	13
CABECERA	13
TÍTULO DEL DOCUMENTO	13
INDICADOR DE REFRESCO DEL DOCUMENTO	14
INDICADOR DE LA URL BASE DEL DOCUMENTO	14
CUERPO.....	15
CARACTERES ESPECIALES.....	17
ESPACIADO Y SALTOS DE LÍNEA	19
<i>Inclusión de espacios en blanco: &nbsp;.....</i>	<i>19</i>
<i>Salto de línea:
 {BR: Break}</i>	<i>19</i>
<i>Cambio de párrafo: <P> {P: Paragraph}</i>	<i>19</i>
<i>Línea Horizontal: <HR> {HR: Horizontal Rule}</i>	<i>20</i>

<i>Texto preformateado: <PRE> {PRE: Preformatted}</i>	21
ELEMENTOS DEL LENGUAJE I	23
CABECERAS Y ATRIBUTOS DE TEXTO.....	23
<i>Cabeceras: <H1>...<H6> {H: Heading}</i>	23
<i>Atributos del texto</i>	24
LISTAS.....	26
<i>Listas no numeradas: {UL: Unordered List}</i>	26
<i>Listas numeradas: {OL: Ordered List}</i>	27
<i>Listas de definiciones: <DL> {DL: Definition List}</i>	29
HIPERENLACES.....	29
<i>Enlaces a otras páginas: </i>	29
<i>Enlaces dentro de la misma página: </i>	30
IMÁGENES	31
ELEMENTOS DEL LENGUAJE II.....	37
TABLAS	37
FORMULARIOS	43
<i>Entrada básica de datos</i>	44
<i>Texto corto: type=text</i>	44
<i>Claves: type=password</i>	45
<i>Botones de selección: type=checkbox</i>	45
<i>Selección entre varias opciones: type=radio</i>	46
<i>Campos ocultos: type=hidden</i>	46
<i>Botón de envío de datos: type=submit</i>	47
<i>Botón gráfico de envío de datos: type=image</i>	47
<i>Botón de borrado de datos: type=reset</i>	47
<i>Entrada datos en múltiples líneas:</i>	48
<i>Entrada de datos a través de campos de selección</i>	48
ELEMENTOS DEL LENGUAJE III	51
FRAMES.....	51
<i>Estructura de un documento con frames:</i>	51
<i>Definición de cada una de las frames</i>	53
ELEMENTOS MULTIMEDIA.....	54
<i>Sonido de fondo</i>	54
<i>Reproducción de vídeo</i>	54
<i>Inserción de cualquier tipo de fichero</i>	55

1

Introducción al HTML

¿Qué es HTML?

HTML es la abreviatura de *HyperText Markup Language*, y es el lenguaje que todos los programas navegadores usan para presentar información en la World Wide Web (WWW).

Este es un lenguaje muy sencillo que se basa en el uso de **etiquetas**, consistentes en un texto ASCII encerrado dentro de un par de paréntesis angulares(<.>). El texto incluido dentro de los paréntesis nos dará una explicación de la utilidad de la etiqueta. Así por ejemplo la etiqueta <TABLE> nos permitirá definir una tabla.

Las etiquetas podrán incluir una serie de *atributos* o parámetros, en su mayoría opcionales, que nos permitirán definir diferentes posibilidades o características de la misma. Estos atributos quedarán definidos por su nombre (que será explicativo de su utilidad) y el valor que toman separados por un signo de igual. En el caso de que el valor que tome el atributo tenga más de una palabra deberá expresarse entre comillas, en caso contrario no será necesario. Así por ejemplo la etiqueta <TABLE border=2> nos permitirá definir una tabla con borde de tamaño 2.

Entre otras cosas, el manejo de estas *etiquetas* nos permitirá:

- Definir la estructura lógica del documento HTML.
- Aplicar distintos estilos al texto (**negrita**, *cursiva*, ...).

- La inclusión de *hiperenlaces*, que nos permitirán acceder a otros documentos relacionados con el actual.
- La inclusión de imágenes y ficheros multimedia (gráficos, vídeo, audio).

¿Qué vamos a necesitar para crear un documento HTML?

- Un *procesador de textos* para escribir y editar el código HTML. Este podrá ser cualquiera que no formatee el texto, ya que el lenguaje HTML está basado en el código ASCII. Si usamos un procesador como el Word, tendremos que guardar el documento como "sólo texto".
- Un *navegador Web* como el *Explorer*, *Netscape*, *Mosaic*, etc., el cual se encargará del interpretar el código HTML de nuestro documento y mostrárnoslo en todo su esplendor.

La secuencia de trabajo para crear nuestro documento es:

1. Con el procesador de texto crearemos el documento HTML y lo guardaremos con el nombre que deseemos, a excepción de la extensión, que deberá ser necesariamente .htm ó .html.
2. Para visualizar nuestro documento HTML abriremos el documento con el programa navegador.

Reglas de formato

Todos los navegadores usan unas reglas básicas para poder mostrar una página web con un buen formato:

- **El espacio en blanco es ignorado.** Ya que un documento HTML puede estar en cualquier tipo de fuente y además la ventana del navegador puede ser de cualquier tamaño.
- **Las etiquetas pueden ser escritas en mayúsculas o en minúsculas.** En todo caso se aconseja su escritura en mayúsculas para poder distinguirlas del texto normal.
- **Existe normalmente una etiqueta de inicio y otra de fin.** La etiqueta de fin contendrá el mismo texto que la de inicio añadiéndole al principio una barra inclinada /. La etiqueta afectará por tanto a todo lo que esté incluido entre las etiquetas de inicio y fin. No obstante, existen algunas que no necesitan cierre, ya que en estas etiquetas se presupone su final, como por ejemplo; <P> párrafo,
 salto de línea ó inclusión de una imagen.

Vemos un ejemplo en la Tabla 1.

Instrucción HTML	Resultado
Texto en negrita	Texto en negrita
<I>Texto en cursiva</I>	<i>Texto en cursiva</i>
<I>Texto en negrita y cursiva</I>	<i>Texto en negrita y cursiva</i>

Tabla 1

Estructura de un documento HTML

La estructura básica de un documento HTML es la siguiente:

<HTML>	Indica el inicio del documento
<HEAD>	Indica el inicio de la cabecera
<TITLE>	Inicio del título del documento
</TITLE>	Final del título del documento
</HEAD>	Final de la cabecera
<BODY>	Inicio del cuerpo del documento
Instrucciones HTML	
</BODY>	Final del cuerpo del documento
</HTML>	Final del documento

Ninguno de estos elementos es obligatorio, pudiendo crear documentos HTML sin incluir estas etiquetas de identificación. No obstante es altamente recomendable la construcción de páginas HTML siguiendo esta estructura, para una buena estructuración y legibilidad del código.

Nuestra primera página HTML

Una vez que conocemos la estructura básica de un documento HTML, vamos a crear una página HTML muy sencillita.

1. Lo primero será abrir el procesador de textos que hayamos elegido y copiar el Código fuente 1.

```
<HTML>
<HEAD>
<TITLE> Mi primera página HTML </TITLE>
</HEAD>
<BODY>
Esta es mi primera página <B>HTML</B>
</BODY>
</HTML>
```

Código fuente 1

2. Una vez tecleado el código, guardamos el documento como Pagina1.htm, y procedemos a abrirlo con el navegador correspondiente. El resultado será:

Esta es mi primera página HTML

Forma del URL

El URL (*Uniform Resource Locator*) nos van a permitir localizar y acceder a cualquier recurso de la red desde nuestro programa navegador. A través de los URL podremos referenciar cualquier fichero que deseemos incluir en nuestro documento HTML. Cada elemento de Internet vendrá definido por su URL correspondiente, independientemente del tipo de servidor en el que se encuentre.

La forma general del URL es:

Tipo de servicio: `//maquina.dominio:puerto/camino/archivo`

El Tipo de servicio, como su propio nombre indica, hace referencia a alguno de los servicios de Internet. El servicio correspondiente al WWW es `http` (*HiperText Transport Protocol*), protocolo para la transmisión de hipertexto. Por lo que cualquier referencia a un documento HTML debería comenzar por `http://`. Otros servicios son: `ftp`, `news`, `telnet` o `mailto`.

La `máquina.dominio` indica el nombre del servidor en que se encuentra el documento al que estamos haciendo referencia, junto con el dominio al que pertenece dicho servidor.

El `puerto` es opcional, y lo normal es no ponerlo, indicándose sólo en el caso de que el servidor utilice un puerto distinto al puerto por defecto.

El `camino` indica la ruta de directorios que hay que seguir para encontrar el documento que estamos referenciando.

Finalmente especificaremos el nombre del *archivo* en que está guardado el documento en cuestión. Si no indicamos un archivo, accederemos al archivo por defecto del directorio al que estemos referenciando. (En el caso del servicio `http` este archivo por defecto suele ser `default.htm` ó `default.html`).

Vamos a ver a continuación algunos ejemplos de URL' s, accediendo a distintos servicios de la WWW.

Localizando un documento hipertexto (http):

Sintaxis:

<http://maquina.dominio:puerto/camino/archivo>

Ejemplo1: <http://www.eidos.es>

En este ejemplo estamos indicando el servicio al que estamos accediendo (`http`), la máquina y el dominio, tomándose el resto de parámetros por defecto, de forma que accederemos al directorio raíz del servidor y al documento por defecto de ese directorio, en este caso estamos accediendo a la página de portada de *Villa Eidos*.

Ejemplo2: <http://www.eidos.es/vulcan/informa.htm>

Con esta URL estamos accediendo al fichero `informa.htm` que se encuentra en el directorio `vulcan` del servidor www.eidos.es.

Localización de un fichero por el servicio FTP (ftp:File Transfer Protocol)

Sintaxis:

<ftp://maquina.dominio:puerto/camino/archivo>

Ejemplo: [ftp:// ftp.jet.es/pub/auto ftp/](ftp://ftp.jet.es/pub/auto_ftp/)

En este caso, estamos accediendo al servicio ftp (protocolo de transferencia de archivos), ftp.jet.es. En este caso no estamos indicando ningún archivo, con lo que se referenciará un índice con los contenidos del directorio: /pub/auto_ftp/.

Accediendo a un grupo de noticias (news):

Sintaxis:

<news:GRUPO>

Ejemplo: news:jet.es

En este ejemplo estamos accediendo al grupo de noticias del servidor de news jet.es.

Accediendo a una dirección e-mail (mailto):

Sintaxis:

mailto: Destinatario_1, Destinatario_2,...,Destinatario_n

Ejemplo: mailto:cursos@eidos.es

Enviaré un mail a la dirección de correo electrónico indicada.

Localización de un archivo cualquiera (file):

Sintaxis:

<file://maquina.dominio:puerto/camino/archivo>

Ejemplo: <file://c:/windows/>

En este caso estamos accediendo a la carpeta Windows de la unidad C: de nuestro ordenador local.

Cabecera y cuerpo del documento HTML

Cabecera

Cabecera del documento HTML

```
<HEAD>
.....
</HEAD>
```

En la cabecera de un documento HTML incluiremos las definiciones generales que afecten a todo el documento, como por ejemplo el título de la página que aparecerá en la parte superior de la ventana del navegador.

Las etiquetas `<HEAD>` y `</HEAD>` son las que van a delimitar la cabecera de nuestro documento. Todas las etiquetas de la cabecera son opcionales, aunque se recomienda incluir en todos nuestros documentos la etiqueta `<TITLE>` correspondiente al título del documento.

A continuación vamos a tratar cada uno de los posibles componentes de la cabecera por separado.

Título del documento

```
<TITLE>.....</TITLE>
```

El título de nuestro documento viene especificado por las etiquetas: `<TITLE>` y `</TITLE>` y, como ya hemos mencionado, este título es el que aparecerá en la ventana de nuestro programa

navegador. Lo normal y recomendable es que el título guarde relación con el contenido de nuestro documento, ya que por ejemplo es utilizado por algunos servidores de búsqueda para poder intuir el contenido de nuestro documento. Veamos un ejemplo en el Código fuente 2.

```
<TITLE>Mi primera página Web</TITLE>
```

Código fuente 2

Indicador de refresco del documento

```
<META http-equiv="refresh" content="número_segundos;url=URL de refresco">
```

Esta etiqueta sirve para indicar un documento que deberá sustituir al actual transcurrido un número determinado de segundos. Una posible utilidad de esta etiqueta podría ser para visualizar documentos de forma secuencial, de forma que se empieza por un documento y se pasa al siguiente transcurrido un período de tiempo sin necesidad de que la persona que lee el documento realice ninguna acción.

Veamos un ejemplo en el Código fuente 3.

```
<META http-equiv="refresh" content="5;url=http://www.eidos.es">
```

Código fuente 3

Si insertamos esta instrucción en la cabecera de nuestro página HTML, transcurridos 5 segundos ésta se refrescará con la página Web de portada de: "Villa Eidos".

Indicador de la URL base del documento

```
<BASE href="URL">
```

Con esta etiqueta indicamos la localización de los ficheros a los que se hace referencia en nuestra página Web. Si no incluimos esta etiqueta el navegador entenderá que dichos ficheros se encuentran en el mismo lugar donde reside nuestra página Web.

Vemos un ejemplo en el Código fuente 4.

```
<BASE href="http://www.eidos.es/vulcan/">
```

Código fuente 4

Si especificamos como URL base de nuestro documento HTML la indicada en este ejemplo y posteriormente hacemos referencia en nuestra página a un fichero llamado "aula.htm", se entenderá que la localización del fichero es "http://www.eidos.es/vulcan/aula.htm".

Cuerpo

Cuerpo del documento HTML: <BODY>

 </BODY>

En el cuerpo de un documento HTML es donde incluiremos las distintas instrucciones del lenguaje junto con el contenido en sí de nuestra página Web: *textos, imágenes, enlaces a otras páginas, etc..*

Las etiquetas <BODY> y </BODY> son las que van a delimitar el cuerpo de nuestro documento. Esta posee una serie de argumentos que nos van a permitir variar las características del documento en su conjunto, como por ejemplo el color del texto o del color de fondo de nuestra página web.

Antes de analizar los componentes y atributos de BODY vamos a ver como se especifican los colores en un documento HTML.

Estos se pueden especificar por su nombre (name) o por su código de color, que es un número compuesto de tres pares de cifras hexadecimales que nos indican la proporción de los colores primarios (rojo, verde y azul) que forman el color deseado (#rrggbb):

Nombre	Código de color	Color mostrado
Black	#000000	Negro
Teal	#008080	Teal
Blue	#0000FF	Azul
Navy	#000080	Azul marino
Lime	#00FF00	Lima
White	#FFFFFF	
Purple	#800080	Púrpura
Yellow	#FFFF00	Amarillo
Olive	#808000	Oliva
Red	#FF0000	Rojo
maroon	#800000	Marrón
gray	#808080	Gris
fuchsia	#FF00FF	Fucsia
green	#008000	Verde

silver	#C0C0C0	Plata
aqua	#00FFFF	Agua

Tabla 2

La etiqueta `<BODY>` presenta una serie de atributos que van a afectar a todo el documento en su conjunto. Estos atributos nos van a permitir definir los colores del texto, del fondo, y de los hiperenlaces del documento. Incluso nos permitirán insertar una imagen de fondo en nuestra página.

```
<BODY background="URL" bgcolor="#rrggbb ó name" text="name"
link="name" vlink="name" >
```

- `background="URL"`. Nos va a permitir mostrar una imagen como fondo de nuestro documento HTML. El camino a esta imagen vendrá especificado por la URL que definamos. Si la imagen no rellena todo el fondo del documento, ésta será reproducida tantas veces como sea necesario hasta completar todo el fondo.

Si insertas el Código fuente 5 en el documento HTML que creamos como ejemplo en el capítulo anterior: *Pagina1.htm*, este presentará como fondo la imagen "fondo.gif"

```
<BODY background="fondo.gif">
```

Código fuente 5

- `bgcolor=#rrggbb ó name {bgcolor: Background Color}`. Nos va a permitir definir un color para el fondo de nuestro documento. Este atributo será ignorado si previamente hemos utilizado el atributo *background*.

Inserte el Código fuente 6 en *Pagina1.htm* y compruebe el resultado.

```
<BODY bgcolor="blue">
```

Código fuente 6

- `text=#rrggbb ó name`. Nos permitirá definir un color para el texto de nuestro documento. Por defecto es negro. Inserte el Código fuente 7 en *Pagina1.html* y compruebe el resultado

```
<BODY text="red">
```

Código fuente 7

Y ahora, inserte el Código fuente 8 en el documento y comprueba el resultado.


```
<BODY bgcolor="yellow" text="navy">
```

Código fuente 8

- `link=#rrggbb` ó `name`. Indica el color que tendrán los hiperenlaces que no han sido accedidos. Por defecto es azul. Como todavía no sabemos insertar hiperenlaces en nuestro documento vamos a dejar el ejemplo correspondiente para más adelante.
- `vlink=#rrggbb` ó `name {vlink: Visited Links}`. Indica el color de los hiperenlaces que ya han sido accedidos. Por defecto es púrpura.

Caracteres especiales

Hasta ahora hemos escrito algunos ejemplos de código HTML, pero hemos pasado por alto algunas limitaciones de HTML respecto al uso de caracteres especiales. Una de estas limitaciones es, por ejemplo, el uso de los caracteres `<` y `>`, que como ya sabemos indican el inicio y fin de una etiqueta HTML. Pues bien, si quisiéramos escribir estos caracteres como parte normal de un texto, el navegador no sabría si se trata de texto normal o del comienzo y final de una etiqueta, o sea, que se haría un lío.

Como todo tiene solución, existen unos códigos para poder escribir estos caracteres y otros relacionados con las etiquetas los vemos en la Tabla 3.

Símbolo	Código
<code><</code> (menor que)	<code>&lt;</code>
<code>></code> (mayor que)	<code>&gt;</code>
<code>&</code> (ampersand)	<code>&amp;</code>
<code>"</code> (comillas)	<code>&quot;</code>

Tabla 3

Para las letras específicas del idioma castellano: las vocales acentuadas, la ñ, la ü y los signos ¿ y ¡, existen los códigos que muestra la Tabla 4.

Letra	Código	Letra	Código	Letra	Código
á	<code>&aacute;</code>	Á	<code>&Aacute;</code>	ñ	<code>&ntilde;</code>
é	<code>&eacute;</code>	É	<code>&Eacute;</code>	Ñ	<code>&Ntilde;</code>
í	<code>&iacute;</code>	Í	<code>&Iacute;</code>	ü	<code>&uuml;</code>

ó	ó	Ó	Ó	Ü	Ü
ú	ú	Ú	Ú	¿	¿
				¡	¡

Tabla 4

Para los navegadores actuales, podemos escribir nuestro código HTML de manera directa sin tener en cuenta estos códigos, pero nunca podemos asegurar que las personas que accedan a nuestras páginas con otros navegadores puedan leerlas correctamente. Así que lo más adecuado es el empleo de los mismos. Otros símbolos especiales aparecen en la Tabla 5.

Símbolo	Código
ç	ç
Ç	Ç
©	©
®	®

Tabla 5

Veamos un ejemplo en el Código fuente 9.

```
&quot;La inspiraci&oacute;n existe, pero tiene que encontrarse trabajando&quot;.  
&copy; Pablo Ru&iacute;z Picasso.
```

Código fuente 9

Esta instrucción HTML, debería producir el siguiente resultado:

```
"La inspiración existe, pero tiene que encontrarse trabajando". ©  
Pablo Ruíz Picasso.
```

Para la inclusión de comentarios en nuestra página HTML, podemos usar la etiqueta: `<!-- -->`. Estos comentarios nos podrán servir para efectuar anotaciones en nuestro documento HTML que ayuden a una mayor comprensión del código. En el Código fuente 10 se muestra un ejemplo de comentario.

```
<!-- Esto es un comentario y no será mostrado por el navegador -->
```

Código fuente 10

Ejemplo	Resultado
Esta frase tiene aquí<P> un cambio de párrafo	Esta frase tiene aquí un cambio de párrafo

Tabla 9

La etiqueta de fin párrafo </P> es opcional, no siendo necesario incluirla. Aunque siempre es recomendable delimitar claramente el inicio y final de un párrafo. Además, cuando usemos esta etiqueta como cerrada <P>..... </P>, tenemos la posibilidad de incluirle el atributo align el cual indica al navegador la forma de justificar el texto incluido en el párrafo. El formato sería el siguiente:

`<P align= left / right / center / justify >Texto contenido en el párrafo </P>`

Veamos algunos ejemplos en la Tabla 10.

Ejemplo	Resultado
<code><P align=right>Este es un ejemplo de párrafo alineado a la derecha</P></code>	Este es un ejemplo de párrafo alineado a la derecha
<code><P align=center>Este es un ejemplo de párrafo centrado</P></code>	Este es un ejemplo de párrafo centrado

Tabla 10

Línea Horizontal: <HR> {HR: *Horizontal Rule*}

Nos permite insertar una línea horizontal, cuyo tamaño podremos determinar a través de sus atributos. Si no especificamos ningún atributo dibujará una línea que ocupe el ancho de la pantalla del navegador. Su utilidad es la de permitarnos dividir nuestra página en distintas secciones. No será necesaria la etiqueta de fin </HR>.

El formato de la etiqueta con sus posibles atributos es:

`<HR align= left / right / center noshade size=n width=n >`

- **align=** left / right / center. Permite establecer la alineación de la línea a la izquierda, a la derecha o centrarla.
- **noshade**. No muestra la sombra de la línea, evitando el efecto de tres dimensiones.
- **size=n**. Indica el grosor de la línea en pixels.
- **width=n** ó **n%**. Especificará el ancho de la línea, este se podrá especificar en pixels (n) o en tanto por ciento del ancho de la ventana (n%). Vamos a ver algunos ejemplos en la Tabla 11.





Ejemplo	Resultado
<code><HR></code>	
<code><HR size=3 noshade></code>	
<code><HR size=5 width=50% align=right></code>	
<code><HR size=10 width=50% align=center></code>	

Tabla 11

Texto preformateado: `<PRE>` {*PRE: Preformatted*}

Nos permitirá visualizar el texto tal y como se ha escrito, respetando los saltos de línea, las tabulaciones, y espacios en blanco de todo el texto incluido entre las etiquetas `<PRE>` y `</PRE>`.

El texto se mostrará con una fuente de espaciado fijo más pequeña que el texto normal. No es recomendable su uso ya que presenta muchas limitaciones y no se obtiene una buena presentación.

Ejemplo	Resultado
<code><PRE>Este texto está preformateado.</PRE></code>	Este texto está preformateado.

Tabla 12

Elementos del lenguaje I

Cabeceras y atributos de texto

Cabeceras: <H1>...<H6> {H: Heading}

En un documento HTML podemos incluir seis tipos distintos de cabeceras, que van a constituir normalmente el título y los distintos subapartados que forman el documento, aunque podremos utilizarlas en el punto que queramos del documento para resaltar cualquier texto. Estas cabeceras son definidas por las etiquetas <H1><H2><H3><H4><H5> y <H6>. El texto que componga cada cabecera deberá estar incluido entre las etiquetas de inicio (<Hn>) y fin (</Hn>) correspondiente. La cabecera <H1>será la que muestre el texto de mayor tamaño, este tamaño irá disminuyendo hasta llegar a la cabecera </H6>. Como podemos ver en la Tabla 13. Para alinear las cabeceras utilizamos el atributo align. El formato sería: <Hn align=center>, como muestra la Tabla 14.

Ejemplo	Resultado
<H1>Cabecera 1</H1>	Cabecera 1
<H2>Cabecera 2</H2>	Cabecera 2

<H3>Cabecera 3</H3>	Cabecera 3
<H4>Cabecera 4</H4>	Cabecera 4
<H5>Cabecera 5</H5>	Cabecera 5
<H6>Cabecera 6</H6>	Cabecera 6

Tabla 13

Ejemplo	Resultado
<H5 align=center>Cabecera 5</H5>	Cabecera 5

Tabla 14

Atributos del texto

Al texto de nuestro documento HTML le podemos aplicar distintos atributos (**negrita**, *cursiva*, subrayado, etc.), al igual que hacemos cuando trabajamos con el procesador de textos en la edición de nuestros documentos. Para aplicar estos atributos disponemos de distintas etiquetas, que aplicarán su efecto al texto incluido entre sus indicadores de inicio y fin. Las vemos en la Tabla 15.

Atributo	Etiqueta	Ejemplo	Resultado
Negrita		Texto en negrita	Texto en negrita
Cursiva	<I></I>	<I>Texto en cursiva</I>	<i>Texto en cursiva</i>
Teletype	<TT></TT>	<TT>Texto en modo teletype</TT>	Texto en modo teletype
Subrayado	<U></U>	<U>Texto subrayado</U>	<i>Texto subrayado</i>
Tachado	<S></S>	<S>Texto tachado</S>	Texto tachado
Parpadeo	<BLINK></BLINK>	<BLINK>Texto parpadeando</BLINK>	Texto parpadeando
Superíndice		^{Texto en modo superíndice}	Texto en modo superíndice
Subíndice		_{Texto en modo subíndice}	Texto en modo subíndice
Centrado	<CENTER></CENTER>	<CENTER>Texto centrado</CENTER>	Texto centrado

{B: Bold} {I: Italic} {TT: Teletype} {U: Underlined} {S: Strike} {SUP: Superscript} {SUB: Subscript}

Tabla 15

Alguna de estas etiquetas no es reconocida por determinados navegadores. Así por ejemplo, la etiqueta `<BLINK>` (Parpadeo) no es implementada por el Explorer. Por lo que según el explorador que estemos utilizando, podremos ver el resultado correctamente o no.

Existen otras etiquetas que nos van a servir para especificar, de manera diferenciada, unidades lógicas de nuestro documento HTML tales como citas, direcciones de correo, etc. En algunos casos el formato obtenido con estas etiquetas de estilo lógico va a ser el mismo que con las anteriores, a las que se les denomina también etiquetas de estilo físico. Algunas etiquetas de estilo lógico las vemos en la Tabla 16.

Etiqueta	Ejemplo	Resultado
<code></code>	<code>Especifica texto resaltado (igual) </code>	Especifica texto resaltado (igual)
<code><CITE></CITE></code>	<code><CITE>Indica una cita o título (igual <I>) </CITE></code>	<i>Indica una cita o título (igual <I>)</i>
<code><STRIKE></STRIKE></code>	<code><STRIKE>Texto tachado (igual <S>) </STRIKE></code>	Texto tachado (igual <S>)

Tabla 16

Si queremos aplicar efectos más espectaculares a nuestro documento HTML, debemos variar el tamaño, el color y el tipo de letra del texto. La etiqueta que nos permite todo esto es `...`, por medio de sus atributos `size`, `color` y `face`:

``

- `size="n" ó "+/- n"`. El atributo `size` nos permite especificar un tamaño determinado para la fuente del texto incluido entre las etiquetas de inicio y fin, el cual puede estar entre 1 y 7. El texto de tamaño normal equivale a la fuente de tamaño 3 (fuente base). Por tanto, si especificamos `size=+2`, el tamaño de la fuente será 5. Y si especificamos `size=-1`, el tamaño será 2.

Ejemplo	Resultado
<code>Tamaño 2</code>	Tamaño 2
<code>Tamaño 5 (3+2)</code>	Tamaño 5 (3+2)
<code>Tamaño 2 (3-1)</code>	Tamaño 2 (3-1)

Tabla 17

- `color="#rrggbb ó name"`. Nos va a permitir definir el color que tendrá el texto incluido entre las etiquetas. Tabla 18.

Ejemplo	Resultado
<code>Texto de color rojo</code>	Texto de color rojo
<code>Texto de color azul</code>	Texto de color azul
<code>Texto verde y de tamaño 5</code>	Texto verde y de tamaño 5

Tabla 18

- `face="nombre de font"`. Nos va a permitir escribir texto con el tipo de letra que le especifiquemos. En el caso de que el tipo de letra que le hayamos especificado no esté cargada en el ordenador que lee la página, se usará el font por defecto del navegador.

Ejemplo	Resultado
<code>Tipo de letra Tahoma</code>	Tipo de letra Tahoma
<code>Texto azul, de tamaño 4 y Tahoma</code>	Texto azul, de tamaño 4 y Tahoma

Tabla 19

Listas

Podemos representar elementos en forma de lista dentro de nuestros documentos de una forma muy sencilla y con una gran versatilidad. Estas listas podrán incluir cualquiera de los elementos HTML e incluso podemos definir listas anidadas, es decir, listas dentro de listas. HTML nos permite crear tres tipos distintos de listas:

- **Listas no numeradas**
- **Listas numeradas**
- **Listas de definiciones**

Listas no numeradas: `` {UL: Unordered List}

Con este tipo de listas podemos especificar una serie de elementos sin un orden predeterminado, precedidos de una marca o viñeta que nosotros mismos podemos definir. Para la definición de los límites de la lista utilizaremos la etiqueta `.... `, y para determinar cada uno de los elementos que la componen usaremos la etiqueta ``. El formato es el siguiente:

```
<UL type="disk" ó "circle" ó "square">
    <LI>Título de la lista</LI>
```

```

    <LI>Elemento 1
    <LI>Elemento 2
    ....
    <LI>Elemento n
</UL>
{LH: List Head} {LI: List Item}

```

Con el atributo type vamos a especificar el tipo de marca o viñeta que antecederá a cada uno de los elementos de la lista. Estas marcas podrán ser: un disco (disk), un círculo (circle) o un cuadrado (square). La etiqueta <LH> nos va a servir para especificar un posible título de la lista, siendo esta opcional. Vamos a ver algunos ejemplos en la Tabla 20

Ejemplo	Resultado
<pre> <UL type=circle> <LH>Los cinco continentes</LH> Europa Asia Africa América Oceanía </pre>	<p>Los cinco continentes</p> <ul style="list-style-type: none"> <input type="checkbox"/> Europa <input type="checkbox"/> Asia <input type="checkbox"/> África <input type="checkbox"/> América <input type="checkbox"/> Oceanía
<pre> <UL type=square> Europa Asia Africa América <UL type=circle > América del Norte América del Sur Oceanía </pre>	<ul style="list-style-type: none"> ■ Europa ■ Asia ■ África ■ América <input type="checkbox"/> América del Norte <input type="checkbox"/> América del Sur ■ Oceanía

Tabla 20

Listas numeradas: {OL: Ordered List}

Con este tipo de listas podemos especificar una serie de elementos numerados según el lugar que ocupan en la lista. Para la definición de los límites de la lista utilizaremos la etiqueta , y para determinar cada uno de los elementos que la componen usaremos la etiqueta . El formato es el siguiente:

```

<OL start="n" type="Tipo de lista">
  <LH>Título de la lista</LH>

```

```

    <LI>Elemento 1
    <LI>Elemento 2
        ....
    <LI>Elemento n
</OL>

```

Con el atributo `start` vamos a especificar el número por el que va a empezar la lista. Si no indicamos este argumento la lista empezará a numerarse a partir del 1. Con el atributo `type` vamos a especificar el tipo de lista numerada. Sus posibles valores aparecen a continuación y veremos algunos ejemplos en la Tabla 21.

A: Letras mayúsculas (A, B, C, ...)

a: Letras minúsculas (a, b, c, ...)

I: Números romanos en mayúsculas (I, II, III, IV, ...)

i: Números romanos en minúsculas (i, ii, iii, iv, ...)

1: Numéricamente (1, 2, 3, 4,) (es la numeración por defecto y por tanto no habría que indicarla).

Ejemplo	Resultado
<pre> <OL type=A> <LH>Los cinco continentes</LH> Europa Asia África América Oceanía </pre>	<p>Los cinco continentes</p> <p>A. Europa</p> <p>B. Asia</p> <p>C. África</p> <p>D. América</p> <p>E. Oceanía</p>
<pre> <OL type=I start=2> Europa Asia África América <OL type=i > América del Norte América del Sur Oceanía </pre>	<p>II. Europa</p> <p>III. Asia</p> <p>IV. África</p> <p>V. América</p> <p> i. América del Norte</p> <p> ii. América del Sur</p> <p>VI. Oceanía</p>

Tabla 21

Listas de definiciones: <DL> {DL: Definition List}

Estas listas nos van a servir para especificar una serie de términos y sus definiciones correspondientes. Para la definición de la lista usaremos la etiqueta <DL> . . . </DL>, para especificar los términos usaremos la etiqueta <DT> y para especificar la definición correspondiente a cada término usaremos la etiqueta <DD>. El formato es el siguiente:

```
<DL>
    <LH>Título de la lista</LH>
    <DT>Término 1
    <DD>Definición 1
    <DT>Término 2
    <DD>Definición 2
    . . . .
    <DT>Término n
    <DD>Definición n
</DL>
{DT: Definition Term} {DD: Definition Description}
```

Vamos a ver un ejemplo en la Tabla 22.

Ejemplo	Resultado
<pre><DL> <DT>París <DD>Capital de Francia <DT>Roma <DD>Capital de Italia <DT>Madrid <DD>Capital de España </DL></pre>	<p>París</p> <p>Capital de Francia</p> <p>Roma</p> <p>Capital de Italia</p> <p>Madrid</p> <p>Capital de España</p>

Tabla 22

Hiperenlaces

Los hiperenlaces son *enlaces de hipertexto* que nos van a permitir acceder de manera directa a otros documentos HTML independientemente de su ubicación, o a otras zonas dentro de nuestro propio documento.

Enlaces a otras páginas:

Con este tipo de hiperenlaces vamos a poder acceder tanto a otras páginas que estén ubicadas dentro de nuestro propio sistema como a páginas ubicadas en puntos muy distantes del globo. El formato de este tipo de hiperenlaces es:

```
<A href="URL a la que se accede">Texto del hiperenlace</A>
```

Con el atributo `href` vamos a especificar la URL del documento al que se pretende acceder. El texto contenido entre las etiquetas de comienzo y fin nos va a servir para definir el hipervínculo, por lo que debería ser clarificador del contenido del documento con el que vamos a enlazar. Esta definición aparecerá resaltada normalmente en azul y subrayada. En la mayoría de los navegadores esta definición del hipervínculo es sensible, por lo que cuando el cursor pasa por encima del mismo este cambia de aspecto indicándolo.

Veamos un ejemplo en la Tabla 23.

Ejemplo	Resultado
<code>Página inicial de Villa Eidos</code>	Página inicial de Villa Eidos

Tabla 23

Si situamos el ratón encima del hipervínculo y pulsamos, el navegador accederá a la página Web indicada por `href`, es decir, accederá a la página situada en `http://www.eidos.es`.

Igualmente podríamos indicar el hipervínculo por medio de una imagen. Si esta imagen tiene definido un borde, este aparecerá resaltado en color azul. El formato correspondiente sería:

```
<A href="URL a la que se accede"><IMG src="Imagen"> y también
texto</A>
{IMG: Image} {src: Source}
```

Veamos un ejemplo en la Tabla 24


Ejemplo	Resultado
<code> Villa Eidos</code>	 Villa Eidos

Tabla 24

Si pulsamos sobre la imagen, el navegador accederá a la página Web inicial de Villa Eidos.

Enlaces dentro de la misma página: ``

Este tipo de hipervínculos nos va a permitir marcar distintas zonas del documento activo con el objeto de tener un acceso directo a las mismas. Una buena utilidad de este tipo de enlaces radica en la creación de índices para documentos largos, de forma que si pinchamos en el hipervínculo correspondiente al título de un capítulo determinado, el navegador saltará automáticamente hasta el comienzo de dicho capítulo.

Para la creación de estos hipervínculos, debemos seguir dos pasos:

1. Marcar las distintas zonas o secciones del documento. Esto lo haremos con el parámetro name:

```
<A name="Identificador de sección">Texto de la
sección</A>
```

A cada sección le asignaremos un identificador distinto, para poder referenciarlas posteriormente de manera inequívoca.

2. Especificar un enlace a cada una de las secciones que hayamos definido.

```
<A href="#Identificador de sección">Texto del enlace a la
sección</A>
```

Si pinchamos en este enlace, el navegador saltará automáticamente a la sección correspondiente. Veamos un ejemplo sencillito en la Tabla 25, pero demostrativo, de todo lo que acabamos de explicar.

Ejemplo	Resultado
<pre>Capítulo1 Capítulo 1</pre>	<p>Capítulo 1</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>Capítulo 1</p>

Tabla 25

Si nos creamos una página HTML con distintos capítulos, podríamos crear una sección para cada uno de ellos, de forma que si pinchamos en el hipervínculo correspondiente al [Capítulo 1](#), el navegador saltaría directamente a la sección correspondiente al **Capítulo 1**.

Además de los dos tipos de hipervínculos vistos, podemos crear hipervínculos que nos permitan acceder a cualquiera de los servicios de Internet. Es decir, podríamos indicar enlaces a servidores FTP, GOPHER, NEWS, MAILTO, etc. Para ello tendríamos que indicar la URL correspondiente al servicio al cual queremos acceder, como ya vimos en el primer capítulo de este manual.

Imágenes

Podremos dar un aspecto mucho más vistoso a nuestros documentos HTML con la inclusión de imágenes. Los formatos de imagen que todos los navegadores reconocen son: gif y jpg. Si queremos insertar una imagen en otro formato, el visor debería disponer del programa externo que permita su visualización. De todas formas, una buena solución es utilizar algún programa de tratamiento de imágenes que nos convierta la imagen a uno de los formatos mencionados anteriormente.

El formato gif es más recomendado para la inclusión de iconos, gráficas,... y el formato jpg para el caso de insertar fotografías en nuestras páginas.

La etiqueta que nos va a permitir la inserción de imágenes es . La imagen se insertará justo en el punto del documento donde coloquemos dicha etiqueta, pudiendo insertar imágenes dentro de listas, tablas o formularios.

El formato de la etiqueta, con sus posibles argumentos, es el siguiente:

```
<IMG src="URL de la imagen" alt="Texto alternativo"
align="top/middle/botton/left/rigth" border="Tamaño"
height="Tamaño" width="Tamaño" hspace="margen" vspace="margen">
```

- src="URL de la imagen". El atributo src nos va servir para indicar la URL de la imagen que queremos insertar, es decir, el servidor y camino hasta llegar al fichero de la imagen.

No será necesario que la imagen que queremos insertar en nuestro documento esté en el ordenador local, aunque sí es recomendable ya que el acceso a las mismas puede ser más lento.

Por tanto, es una buena práctica copiar todas las imágenes, que vayamos a incluir en nuestro documento HTML, al ordenador local.


Ejemplo	Resultado
<code></code>	

Tabla 26

- alt="Texto alternativo". Nos va a permitir mostrar un texto alternativo para el caso en que el navegador no sea capaz de mostrar la imagen. Su uso es muy recomendable ya que cuando publiquemos nuestras páginas en la WWW, no sabremos desde qué tipo de navegadores accederán los internautas a nuestras páginas.

En la Tabla 27 vamos a ver un ejemplo.

Si usas un navegador que no tiene ningún problema para la visualización de imágenes, desactiva temporalmente la opción de mostrar imágenes, prueba el ejemplo que muestra la Tabla 27 y compáralo con el anterior.


Ejemplo	Resultado
<code></code>	

Tabla 27

- `align=top/middle/botton left/rigth`. Nos permite indicar cómo se alineará el texto que siga a la imagen.
 - `top`: alinea el texto con la parte superior de la imagen.
 - `middle`: con la parte central.
 - `bottom`: con la parte inferior.
 - `left`: imagen alineada a la izquierda de la página.
 - `rigth`: imagen alineada a la derecha de la página.




Ejemplo	Resultado
<pre>OVNI·modelo· 747·</pre>	 <div>OVNI·modelo</div> <div>747</div>
<pre>OVNI·modelo 747·</pre>	 <div>OVNI·modelo</div> <div>747</div>
<pre>OVNI·modelo 747·</pre>	 <div>OVNI· modelo</div> <div>747</div>

Tabla 28

En el caso de que el texto que acompaña a la imagen tenga más de una línea, tan sólo la primera será la que se alineará según alguna de las anteriores alineaciones, el resto de líneas se incluirán debajo de la imagen quedando un efecto bastante feo.

Para solventar esto, están las alineaciones **LEFT** (imagen alineada a la izquierda de la página) y **RIGHT** (imagen alineada a la derecha de la página). Las cuales permiten que la imagen se inserte dentro del párrafo que la rodea, produciendo un resultado mucho más profesional en la integración de texto e imágenes.

Estas dos alineaciones no se podrán usar en combinación con las anteriores. Veamos algunos ejemplos en la Tabla 29

Vemos como en el primer ejemplo la segunda línea de texto aparece debajo de la imagen produciendo un efecto menos vistoso que en los dos ejemplos siguientes.




Ejemplo	Resultado
<pre>OVNI
 (Objeto Volante No Identificado) . .</pre>	 <div>OVNI (Objeto Volante No Identificado)</div>
<pre>OVNI
 (Objeto Volante No Identificado) . .</pre>	 <div>OVNI (Objeto Volante No Identificado)</div>
<pre>OVNI
 (Objeto Volante No Identificado) . .</pre>	<div>OVNI (Objeto Volante No Identificado)</div> 

Tabla 29

- `border="Tamaño"` Permite añadirle un borde a la imagen del tamaño que le especifiquemos.

Si la imagen forma parte de un hiperenlace esta aparecerá con un borde del color del hiperenlace. Si no deseáramos que aparezca este borde tendríamos que especificar `border=0`.


Ejemplo	Resultado
<pre><i>Ovni.gif</i> . .</pre>	 <div>Ovni.gif</div>

Tabla 30

- `height="Tamaño"`. Permite especificar el alto de la imagen, en puntos de pantalla (pixels) o en % del tamaño del documento
- `width="Tamaño"`. Con este argumento especificaremos el ancho al que mostraremos la imagen: en puntos de pantalla (pixels) o en % del tamaño del documento

En el segundo de los ejemplos de la Tabla 31, el ancho de la imagen se ha extendido al 50% del total del ancho de la celda donde está contenida la imagen. Si pruebas este ejemplo en tu documento HTML, la imagen se extenderá horizontalmente hasta ocupar el 50% del ancho de la página.



Ejemplo	Resultado
<code></code>	
<code></code>	

Tabla 31

- `hspace="Margen"` {hspace: Horizontal Space}. Permite especificar el número de espacios horizontales (en puntos) que separarán a la imagen de cualquier elemento que le siga o le anteceda.
- `vspace="Margen"` {vspace: Vertical Space}. Con este argumento especificaremos el margen vertical (en puntos) que habrá entre la imagen y cualquier otro elemento de nuestro documento.

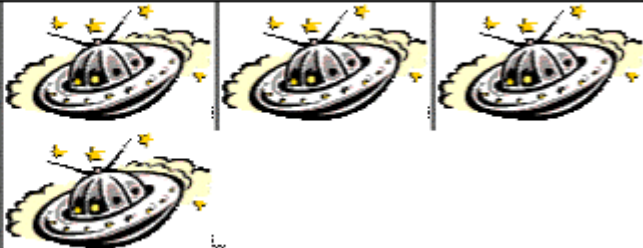
Ejemplo	Resultado
<code></code>	

Tabla 32

Elementos del lenguaje II

Tablas

HTML nos va a permitir la inclusión de cualquiera de los elementos de nuestra página (texto, imágenes, hiperenlaces, listas, etc.), dentro de una tabla. Gracias a lo cual conseguiremos dar una mayor y mejor estructuración a los contenidos de nuestros documentos. Además, la definición de las tablas en HTML es muy abierta, pudiendo en cualquier momento redimensionar la tabla, es decir, cambiar su número de filas o de columnas, cambiar el tamaño de alguna de sus celdas, etc.

La etiqueta que nos va a permitir la definición de tablas es `<TABLE> </TABLE>`.

El formato general de la etiqueta sin ningún argumento, es el siguiente:

```
<TABLE>
  <TR >
    <TH>Contenido de la celda </TH>
    <TD>Contenido de la celda </TD>
  </TR>
</TABLE>
```

{TR: Table Row} {TH: Table Header} {TD: Table Data}

Vamos a analizar cada una de estas etiquetas de forma separada:

1. `<TABLE> </TABLE>`: Definición general de la tabla. Dentro de ella definiremos las filas y columnas que la constituyen, pudiendo incluso definir tablas dentro de tablas, es decir, tablas anidadas.

2. `<TR>` `</TR>`: Definición de las filas de la tabla. Por cada etiqueta `<TR>` que incluyamos se creará una fila en la tabla. No será necesario indicar la etiqueta de cierre.
3. `<TH></TH>` ó `<TD></TD>`: Definición de cada una de las celdas de la tabla. Vemos que estas etiquetas están contenidas dentro de otra etiqueta de definición de fila, de forma que por cada etiqueta `<TH>` o `<TD>` que incluyamos se creará una celda dentro de la fila correspondiente. La etiqueta `<TH>` la usaremos para crear celdas de tipo cabecera, es decir, celdas cuyo contenido está resaltado en negrita, dejando la etiqueta `<TD>` para definir celdas de datos. En este caso tampoco es necesario indicar las etiquetas de cierre.

Todas estas etiquetas tienen sus respectivos argumentos. Pero antes de analizar cada uno de ellos, vamos a ver un ejemplo muy sencillito, en la Tabla 33, que nos creará una tabla de una sola celda (1 fila X 1 columna = 1 celda). Para ello usaremos la etiqueta de creación de tabla `<TABLE>` con el argumento `border=1` (nos dibuja un borde de tamaño 1 alrededor de la tabla).

<i>Ejemplo</i>	Resultado
<pre><TABLE border=1> <TR> <TD>Mi primera tabla </TABLE></pre>	Mi primera tabla

Tabla 33

Vamos ahora a analizar lo distintos argumentos que componen las etiquetas de la tabla, analizándolos desde lo general (la definición de la tabla), hasta lo específico (la definición de las celdas):

1. Definición de la tabla: `<TABLE>`.

```
<TABLE border="n" cellpadding="n" cellspacing="n" width="n ó %"
height="n ó %" bgcolor="#rrggbb ó nombre" >
..... Contenido .....
</TABLE>
```

- `border="n"`. Si especificamos este argumento, se dibujará un borde alrededor de la tabla del ancho que le indiquemos, como muestra la Tabla 34.

Ejemplo	Resultado
<pre><TABLE border=4> <TR> <TD>Mi primera tabla </TABLE></pre>	
	Mi primera tabla

Tabla 34

- `cellpadding="n"`. Indica el espacio en puntos que separa el contenido de la celda con el borde de la misma, siendo 1 por defecto.

Ejemplo	Resultado
<pre><TABLE border=4 cellpadding=8> <TR> <TD>Mi primera tabla </TABLE></pre>	Mi primera tabla

Tabla 35

- `cellspacing="n"`. Indica el espacio en puntos que separa a las celdas contenidas dentro de la tabla, siendo 2 por defecto.
- `width="n ó %"`. Indica la anchura de la tabla en puntos o en % en función del ancho de la ventana del visualizador. Si no indicamos este argumento, el ancho de la tabla se ajustará al tamaño del contenido de las celdas. Esto es lo que ha ocurrido en el último ejemplo que hemos realizado. Veamos un ejemplo en el Código fuente 11. El resultado se muestra en la Tabla 36.

```
<TABLE border=4 width=50%><TR><TD>Mi primera tabla</TABLE>
```

Código fuente 11

Mi primera tabla

Tabla 36

- `height="n ó %"`. Nos permite definir la altura de la tabla en puntos o en % de la altura de la ventana del visualizador. Si indicamos este argumento, lo recomendable es darlo en puntos ya que es más complicado adecuarnos a la altura de la ventana del visualizador. Al igual que en el caso anterior, si no especificamos este argumento la altura se adecuará al contenido de las celdas. El Código fuente 12 muestra un ejemplo y la Tabla 37 nos muestra el resultado.

```
<TABLE border=4 width=50% height=50><TR><TD>Mi primera tabla</TABLE>
```

Código fuente 12

Mi primera tabla

Tabla 37

- `bgcolor= "#rrggbb" o "nombre del color"`. Nos permite definir un color de fondo para todas las celdas de la tabla. Más adelante veremos como podemos definir colores específicos para cada una de las celdas por separado.

Ejemplo	Resultado
<pre><TABLE border=4 bgcolor="yellow"> <TR> <TD>Mi primera tabla </TABLE></pre>	Mi primera tabla

Tabla 38

2. Definición de las filas de la tabla: <TR>

```
<TR align="left/right/center" valign="top/middle/bottom
bgcolor="#rrggbg ó nombre">
```

Antes de empezar a descifrar el significado de cada uno de los argumentos de esta etiqueta vamos a ver un ejemplo, en la Tabla 39 de cómo se construiría una tabla con dos filas (2 filas X 1 columna = 2 celdas).

Ejemplo	Resultado
<pre><TABLE border=4> <TR ><TD>Primera fila <TR><TD>Segunda fila </TABLE></pre>	Primera fila
	Segunda fila

Tabla 39

- `align="left/right/center"`. Con este argumento vamos a indicar la alineación horizontal (*left*:izquierda, *right*:derecha, *center*:centrado) del contenido de todas las celdas de la fila correspondiente. Posteriormente veremos como podremos especificar la alineación del contenido de cada celda de la fila por separado. Veamos un ejemplo en el Código fuente 13 y su correspondiente resultado en la Tabla 40.

```
<TABLE border=4 width=50% > <TR><TD>Primera fila<TR
align=right><TD>Segunda fila</TABLE>
```

Código fuente 13

Primera fila
Segunda fila

Tabla 40

- `valign="top/middle/bottom"`. Con este argumento vamos a indicar la alineación vertical (*top*:arriba, *middle*:centrado, *bottom*:abajo) del contenido de todas las celdas de la fila correspondiente. Posteriormente veremos como podremos especificar la alineación del contenido de cada celda de la fila por separado.

- `bgcolor= "#rrggbb"` o "nombre del color". En este caso, este argumento nos permitirá definir un color de fondo para todas las celdas de una fila, veamos un ejemplo en la

Ejemplo	Resultado
<pre><TABLE border=4> <TR bgcolor="yellow"><TD>Primera fila <TR bgcolor="green"><TD>Segunda fila </TABLE></pre>	Primera fila
	Segunda fila

Tabla 41

3. Definición de las celdas de la tabla: <TH> ó <TD>

```
<TH ó TD align="left/right/center/justify"
valign="top/middle/bottom bgcolor="#rrggbb ó nombre" width="n"
rowspan="n" colspan="n">
```

Como ya explicamos anteriormente, la etiqueta <TH> la usaremos para crear celdas de tipo cabecera y la etiqueta <TD> para la creación de celdas de datos. Vamos a completar nuestro ejemplo inicial añadiéndole tres filas de dos celdas cada una (3 filas X 2 columnas = 6 celdas). Tabla 42

Ejemplo	Resultado	
<pre><TABLE border=4> <TR ><TH>Cabecera 1<TH>Cabecera 2 <TR ><TD>Celda 1.1<TD>Celda 1.2 <TR><TD>Celda 2.1<TD>Celda 2.2 </TABLE></pre>	Cabecera 1	Cabecera 2
	Celda 1.1	Celda 1.2
	Celda 2.1	Celda 2.2

Tabla 42

- `align="left/right/center/justify"`. Con este argumento vamos a indicar la alineación horizontal (*left*: izquierda, *right*: derecha, *center*: centrado, *justify*:justificado) del contenido de cada celda por separado. Su significado es el mismo que en la etiqueta <TR>.
- `valign="top/middle/bottom"`. Con este argumento vamos a indicar la alineación vertical (*top*:arriba, *middle*:centrado, *bottom*:abajo) del contenido de cada celda.
- `bgcolor= "#rrggbb"` o "nombre del color". En este caso, podremos definir un color de fondo para cada una de las celdas de manera independiente.
- `width="n ó %"`. Indica el ancho de la celda en puntos o en % en función del tamaño de la tabla.

Ejemplo	Resultado	
<pre><TABLE border=4> <TR ><TH>Cabecera 1<TH>Cabecera 2 <TR ><TD align=right>Celda 1.1<TD bgcolor=red>Celda 1.2 <TR><TD bgcolor=yellow>Celda 2.1<TD align=center>Celda 2.2 </TABLE></pre>	Cabecera 1	Cabecera 2
	Celda 1.1	Celda 1.2
	Celda 2.1	Celda 2.2

Tabla 43

- `rowspan="n"`. Con este argumento podemos lograr que una celda concreta abarque más de una fila, ya sabemos que por defecto una celda ocupa una sola fila.

Ejemplo	Resultado	
<pre><TABLE border=4> <TR ><TH>Cabecera 1<TH>Cabecera 2 <TR ><TD rowspan=2>Celda 1.1<TD>Celda 1.2 <TR><TD>Celda 2.2 </TABLE></pre>	Cabecera 1	Cabecera 2
	Celda 1.1	Celda 1.2
		Celda 2.2

Tabla 44

- `colspan="n"`. Con este argumento podemos lograr que una celda se expanda a más de una columna.

Ejemplo	Resultado	
<pre><TABLE border=4> <TR ><TH>Cabecera 1<TH>Cabecera 2 <TR ><TD>Celda 1.1<TD>Celda 1.2 <TR><TD colspan=2 align=center>Celda 2.2 </TABLE></pre>	Cabecera 1	Cabecera 2
	Celda 1.1	Celda 1.2
	Celda 2.2	

Tabla 45

Para finalizar este tema sobre la creación de tablas, vamos a ver el ejemplo que aparece en el Código fuente 14 y su resultado en la Tabla 46.

```
<TABLE border=4>
  <TR ><TH colspan=3 bgcolor=red>Título
  <TR bgcolor=yellow><TH>Cabecera 1<TH>Cabecera 2<TH>Cabecera 3
```

```
<TR><TD>Celda 1.1<TD>Celda 1.2<TD>Celda 1.3
<TR><TD rowspan=2>Celda 2.1<TD bgcolor=green>Celda 2.2<TD>Celda 2.3
<TR><TD>Celda 3.2<TD>Celda 3.3
</TABLE>
```

Código fuente 14

Título			
Cabecera 1	Cabecera 2	Cabecera 3	
Celda 1.1	Celda 1.2	Celda 1.3	
Celda 2.1	Celda 2.2	Celda 2.3	
	Celda 3.2	Celda 3.3	

Tabla 46

Formularios

El formulario es el elemento de HTML que nos va a permitir interactuar con los visitantes de nuestras páginas web, de forma que podemos solicitarle información al usuario y procesarla. De esta forma, podremos crear en nuestras páginas: *encuestas* para solicitar la opinión del visitante sobre algún tema e incluso sobre el contenido de nuestra propia página web, *cuestionarios* para evaluar la asimilación de contenidos sobre un tema concreto que se trate en la página, etc.

El contenido de la información introducida por medio del formulario será enviado a la dirección URL donde resida el programa que se encargará de procesar los datos. A este tipo de programas externos se les denomina programas CGI (Common Gateway Interface).

La etiqueta HTML que nos va a permitir la creación de formularios es `<FORM>`. Su formato es el siguiente:

```
<FORM action="URL del programa que trata el formulario"
method="post/get">
..... Elementos del formulario .....
</FORM>
```

- `action="URL del programa que trata el formulario"`. A través de este argumento indicaremos, como ya hemos mencionado, la dirección del fichero de programa que va a tratar la información suministrada por el formulario.

También podríamos incluir como URL que trata el formulario una dirección de correo electrónico. Por tanto, el valor de este argumento podría ser: `action="mailto: dirección de correo"`.

Nota: Esta opción sólo funciona adecuadamente con el navegador *Netscape*. Con el *Explorer* este envío daría como resultado un correo en blanco. Para que esta opción funcione de manera independiente del navegador que se esté utilizando se ha de emplear un programa externo que realice este proceso.

- `method="post/get"`. A través de este argumento indicaremos el método empleado para transferir la información generada por el formulario.

Si indicamos *post*, se producirá la modificación del documento destino, como es el caso de enviar la información a una dirección de correo electrónico. Mediante el método *get* no se producirán cambios en el documento destino.

Dentro de un formulario podremos incluir cualquier texto o instrucción HTML de las vistas a lo largo de los capítulos anteriores además, claro está, de los elementos típicos de un formulario: *cajas de texto*, *botones de selección*, *menús de opciones*, etc. Vamos a ver cómo se incluirán estos elementos en un formulario.

Entrada básica de datos

Para definir los distintos tipos de campos básicos de entrada de datos usaremos la etiqueta `<INPUT>`.

El formato es el siguiente:

```
<INPUT type= text / password / checkbox / radio / hidden / submit /  
image / reset name="Variable" value="Valor inicial">
```

- El argumento `type` determinará el tipo de campo de entrada que estamos insertando. A continuación describiremos cada uno de ellos.
- El argumento `name` especifica el nombre de la variable que tomará el valor introducido en el campo.
- El argumento `value` especifica el valor por defecto que tendrá el campo.

Vamos a continuación a describir los distintos tipos campos de entrada, que vendrán dados por los valores del argumento `type`:

Texto corto: `type=text`

Con este argumento vamos a indicar que el campo a introducir será un texto. El formato sería:

```
<INPUT type= text name="Variable" value="Valor inicialización"  
size="Tamaño" maxlength="Longitud máxima">
```

El significado de los nuevos atributos es:

- `size="Tamaño"`. Indicaremos el tamaño en caracteres de la ventana de introducción de texto.
- `maxlength="Longitud máxima"`. Indicaremos el número máximo de caracteres a introducir en el campo.

Vamos a ver un ejemplo en la Tabla 47.

Ejemplo	Resultado
Nombre:<INPUT type=text name=Var1 Value="Texto por defecto" size=20 maxlength=25>	Nombre: <input type="text" value="Texto por defecto"/>

Tabla 47

Claves: type=password

Con este argumento indicamos que el campo a introducir será una palabra clave, por lo que los caracteres que se introduzcan serán sustituidos por asteriscos en la visualización por pantalla. El formato es:

```
<INPUT type= password name="Variable" value="Valor inicialización"
size="Tamaño" maxlength="Longitud máxima">
```

El significado de los distintos argumentos es el mismo que en el caso anterior. Veamos un ejemplo: en la Tabla 48.

Ejemplo	Resultado
Clave:<INPUT type=password name=Var2 Value="tutankamon" size=11 maxlength=10>	Clave: <input type="password" value="tutankamon"/>

Tabla 48

Botones de selección: type=checkbox

El checkbox es un botón que presenta dos estados: marcado (1) y desmarcado (0). Podremos variar su estado simplemente pinchando con el ratón. El formato es:

```
<INPUT type= checkbox name="Variable" value="Valor" checked>
```

Si especificamos el argumento checked, el botón aparecerá marcado por defecto.

Si incluimos el argumento value, cuando el botón esté marcado su variable asociada adquirirá el valor dado por value.

Veamos un ejemplo en la Tabla 49.

Ejemplo	Resultado
<INPUT type=checkbox name=Var3 value=Op1 checked> Opción1 <INPUT type=checkbox name=Var4 value=Op2> Opción 2	<input checked="" type="checkbox"/> Op1 Opción1 <input type="checkbox"/> Op2 Opción 2

Tabla 49

Selección entre varias opciones: type=radio

Este argumento se usa cuando hay que hacer una selección entre varias alternativas excluyentes, pudiéndose seleccionar únicamente una de las alternativas. Debemos incluir una etiqueta radio por cada una de las posibles alternativas. El formato es:

```
<INPUT type= radio name="Variable" value="Valor" checked>
```

Si especificamos el argumento checked, el botón aparecerá marcado por defecto. En este caso únicamente uno de los botones de radio podrá aparecer marcado, ya que se usa para el caso de opciones excluyentes.

Cuando un botón esté seleccionado la variable asociada a la lista de botones adquirirá el valor dado por value.

Veamos un ejemplo en la Tabla 50.

Ejemplo	Resultado
<code><INPUT type=radio name=Var5 value=Op1></code> Opción1 	<input type="radio"/> Op1 Opción1.
<code><INPUT type=radio name=Var5 value=Op2 checked></code> Opción 2 	<input checked="" type="radio"/> Op2 Opción 2.
<code><INPUT type=radio name=Var5 value=Op3 ></code> Opción 3	<input type="radio"/> Op3 Opción 3.

Tabla 50

En este ejemplo la variable "Var5" tendrá el valor Op2.

Campos ocultos: type=hidden

Este tipo de campos no son visibles para el usuario. Su uso tiene sentido en el caso de enviar algún tipo de información que no deba ser visualizada o variada por el lector de nuestra página Web. El formato es:

```
<INPUT type= hidden name="Variable" value="Valor" >
```

Con esta etiqueta asignaríamos a la "Variable" referenciada por name el "Valor" de value y se mandaría junto con el formulario sin que el usuario de la página se entere de nada. Veamos un ejemplo en el Código fuente 15.

```
<INPUT type=hidden name=Var6 value="tutankamon">
```

Código fuente 15

Botón de envío de datos: type=submit

Con este argumento especificamos un botón en el que al pulsar, los datos serán enviados al programa o dirección de correo encargada de procesar la información recogida por el formulario. El formato es:

```
<INPUT type= submit value="Mensaje a mostrar">
```

En este caso el parámetro value indica el mensaje que se mostrará en el botón. Veamos un ejemplo en la Tabla 51.


Ejemplo	Resultado
<code><INPUT type=submit value="Enviar los Datos"></code>	

Tabla 51

Botón gráfico de envío de datos: type=image

Con este argumento especificamos un botón de tipo imagen en el que, al igual que con el botón anterior, si pulsamos en la imagen los datos serán enviados al programa o dirección de correo encargada de procesar la información recogida por el formulario. Su formato es el siguiente:

```
<INPUT type= image src="URL de la imagen" name="Variable">
```

Con el parámetro src especificamos la URL de la imagen que será insertada en el botón. En este caso, en la variable referenciada por name se almacenarán las coordenadas del punto de la imagen donde se halla pinchado (de forma que estas coordenadas podrían ser procesadas por el programa que recibe los datos del formulario).

Botón de borrado de datos: type=reset

Con este argumento especificamos un botón que al ser pulsado borrará el contenido actual de todos los campos, dejándolos con sus valores por defecto. Su formato es:

```
<INPUT type= reset value="Texto del botón">
```

Con el parámetro value especificamos el texto que etiquetará al botón.

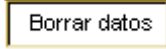
Ejemplo	Resultado
<code><INPUT type=reset value="Borrar datos"></code>	

Tabla 52

Entrada datos en múltiples líneas:

En un formulario también podremos introducir un campo de texto que abarque varias líneas. Para ello usaremos la etiqueta `<TEXTAREA>` `</TEXTAREA>`. Su formato es el siguiente:

```
<TEXTAREA name="Variable" rows=Filas cols=Columnas>
Contenido por defecto.
</TEXTAREA>
```

Los argumentos `rows` y `cols` nos van a permitir especificar, respectivamente, el número de filas de texto visibles y el número de columnas de texto visibles en el campo.

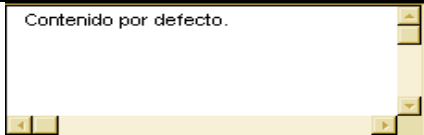
Ejemplo	Resultado
<pre><TEXTAREA name="Var7" rows=5 cols=40> Contenido por defecto. </TEXTAREA></pre>	

Tabla 53

Entrada de datos a través de campos de selección

Con los campos de selección podremos desplegar una lista de opciones, entre las que podremos seleccionar una. Para ello usaremos la etiqueta `<SELECT>` `</SELECT>`. Su formato es el siguiente:

```
<SELECT name="Variable" multiple size=n>
  <OPTION selected value="Valor_1">Primera Opción
  <OPTION value="Valor_2">Segunda Opción
  .....
  <OPTION value="Valor_n">Enésima Opción
</SELECT>
```

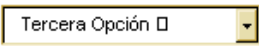
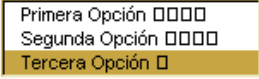
Ejemplo	Resultado
<pre><SELECT name="Var8" > <OPTION value=1>Primera Opción <OPTION value=2>Segunda Opción <OPTION selected value=3>Tercera Opción </SELECT></pre>	
<pre><SELECT name="Var8" multiple size=3> <OPTION value=1>Primera Opción <OPTION value=2>Segunda Opción <OPTION selected value=3>Tercera Opción </SELECT></pre>	

Figura 1

- La variable referenciada por `name` tomará el valor(`value`) de la opción seleccionada.
- Si especificamos el argumento `múltiple`, se mostrarán todas la opciones en forma de tabla. El número de opciones visibles en la tabla vendrá dado por el argumento `size`.

- Con la etiqueta <OPTION> iremos definiendo cada una de las opciones de la lista.
- El argumento `selected` va a determinar cuál es la opción por defecto.

Para finalizar vamos a ver el Código fuente 16, correspondiente a una encuesta realizada por la editorial ficticia *EditoPc*, y en la Figura 2 observamos el resultado del formulario.

```
<P align=center><B>EDITORIAL <I>EditoPc</I></B></P>
<FORM action="mailto:direccion@ejemplo" method = post>
Nombre:<INPUT type="text" size="30" name="nombre"><BR>
E_mail: <INPUT type="text" size="30" name="e_mail">
<P>Deseas recibir información sobre:
<INPUT type="checkbox" name="infor1" value="Internet">Internet
<INPUT type="checkbox" name="infor2" value="Programación">Programación
<INPUT type="checkbox" name="infor3" value="Sist.Operativo">Sist.Operativo
<INPUT type="checkbox" name="infor4" value="Ofimática">Ofimática
<P>¿Te conectas a Internet desde casa?: <br>
<INPUT type="radio" name="intercasa" value="si">Sí
<INPUT type="radio" name="intercasa" value="no">No
<P>¿Qué uso le das a tu ordenador?: <BR>
<SELECT name="uso" multiple size="3">
<OPTION selected>Profesional
<OPTION>Doméstico
<OPTION>Ocio/Juegos
</SELECT>
<P>Características de tu PC:<BR>
<TEXTAREA name="caract" rows="5" cols="40"></TEXTAREA>
<P><INPUT type="submit" value="Enviar datos"> <INPUT type="reset" value="Borrar
datos">
```

Código fuente 16

Nombre:

E_mail:

Deseas recibir información sobre:

☐ Internet ☐ Programación ☐ Sist. Operativo ☐ Ofimática

¿Te conectas a Internet desde casa?:

☐ Sí ☐ No

¿Qué uso le das a tu ordenador?:

☒ Profesional
☐ Doméstico
☐ Ocio/Juegos

Características de tu PC:

Figura 2

Elementos del lenguaje III

Frames

Las frames son una utilidad del HTML que nos permite dividir la ventana del navegador en diferentes subventanas, que podrán ser manipuladas de manera independiente. Esto nos permitirá mostrar una página web diferente en cada una de la subventanas o frames.

Estructura de un documento con frames:

Un documento con frames se estructura de manera diferente a un documento HTML normal. Así, en este tipo de documentos no aparecerá la etiqueta `<BODY>`. En su lugar se incluirá la etiqueta `<FRAMESET>` `</FRAMESET>` dentro de la cual se definirán las distintas frames que se incluyan en el documento.

La estructura general de un documento de este tipo sería, por tanto:

```
<HTML>
  <HEAD>Cabecera</HEAD>
  <FRAMESET>
    Definición de la distintas frames
  </FRAMESET>
  <NOFRAMES>
    Instrucciones HTML
  </NOFRAMES>
</HTML>
```

Puede darse el caso de que programemos nuestro documento con frames siguiendo la estructura anterior y que el navegador que se use para leer el documento no soporte la característica de frames, con lo cual no se mostrará nada de lo indicado entre las etiquetas `<FRAMESET>` y `</FRAMESET>`. Para solucionar este problema, todas las instrucciones HTML incluidas entre las etiquetas `<NOFRAMES>` y `</NOFRAMES>` serán ejecutadas por los navegadores que no soportan las frames.

La sintaxis de la etiqueta `<FRAMESET>` es:

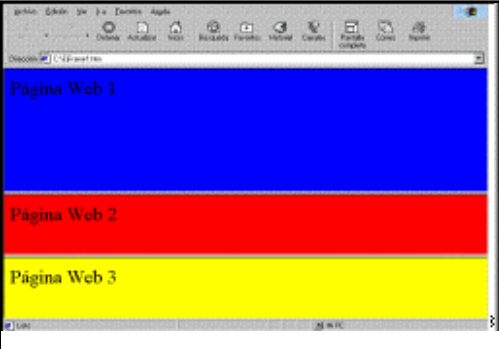
```
<FRAMESET rows="Lista de filas" cols="Lista de columnas">
  <FRAME src="URL de la frame 1" name="Nombre de la frame 1">
  <FRAME src="URL de la frame 2" name="Nombre de la frame 2">
  .....
  <FRAME src="URL de la frame n" name="Nombre de la frame n">
</FRAMESET>
```

Como vemos, por cada una de la frames que incluyamos en nuestro documento debemos incluir una etiqueta `<FRAME>` indicando mediante sus argumentos la dirección URL del documento que se insertará en la subventana (`src`) y el nombre de la frame (`name`). Posteriormente describiremos la etiqueta `<FRAME>` con más detalle.

Los argumentos de `<FRAMESET>` son `rows` y `cols`, definiéndose únicamente uno de los dos atributos en función de si la división de la pantalla se realizará por filas (`rows`) o por columnas (`cols`).

Por cada una de las subventanas que creemos debemos especificar su tamaño en puntos o en porcentaje del tamaño de cada subventana. Así por ejemplo, si especificamos `rows= "50%, 25%, 25%"` crearemos tres subventanas horizontales ocupando la primera el 50% de la ventana principal y las otras dos el resto. Si especificamos `cols="100, 150, * "` crearemos tres subventanas verticales ocupando la primera 100 puntos de la ventana principal, la segunda 150 puntos y tercera el espacio restante.

Vamos a verlo más claramente con un par de ejemplos prácticos, que vemos en la Tabla 54.

Ejemplo	Resultado
<pre><HTML> <FRAMESET rows = "50%,25%,25%"> <FRAME src="Pagweb1.htm" name="Pagweb1"><FRAME src="Pagweb2.htm" name="Pagweb2"><FRAME src="Pagweb3.htm" name="Pagweb3"></FRAMESET> </HTML></pre>	

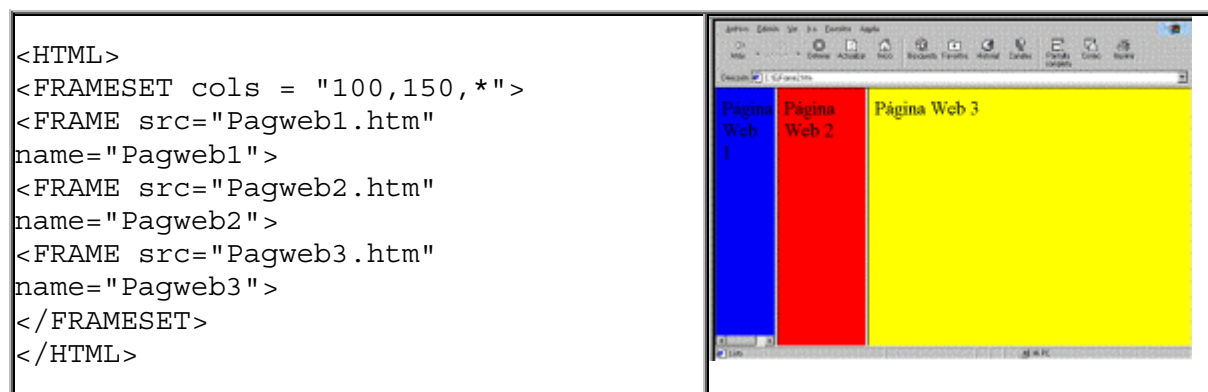


Tabla 54

Definición de cada una de las frames

Como ya sabemos, cada una de las frames que incluyamos en nuestro documento vendrá definida por la etiqueta `<FRAME>` correspondiente. Su formato es el siguiente:

```

<FRAME src="URL" name="Nombre" marginwidth="n" marginheight="n"
scrolling="yes/no/auto noresize>

```

- `Src = "URL"`. La subventana mostrará el contenido del documento HTML que se indique con dicha URL.
- `Name = "Nombre"`. Indica el nombre por el que nos referiremos a esa subventana.
- `marginwidth = "n"`. Establecemos los márgenes izquierdo y derecho del contenido de la frame en puntos por pantalla.
- `marginheight = "n"`. Establecemos los márgenes superior e inferior del contenido de la frame en puntos por pantalla.
- `scrolling = "yes / no / auto"`. Indicaremos si se aplica una barra de desplazamiento a la subventana. El valor **"yes"** muestra siempre la barra de desplazamiento, **"no"** no la muestra nunca y **"auto"** la muestra solo en caso de que sea necesario para poder ver la página.
- `Noresize`. Cuando el usuario visualiza nuestra página con frames en su navegador podrá redimensionar las subventanas seleccionando un borde con el cursor del ratón y desplazándolo. Si se indica este argumento, el usuario no podrá variar el tamaño de las subventanas al visualizarlas con el navegador.

Antes de finalizar con el tema de las frames vamos a hacer referencia a un problema con el que nos vamos a encontrar cuando incluyamos en nuestro documento con frames un hipervínculo:

Supongamos que en nuestro documento tenemos dos subventanas o frames llamadas `frame1` y `frame2`. Hemos incluido un hipervínculo a otra página web en la `frame1`, pero queremos que el resultado del hipervínculo se visualice en la `frame2` (ya que por defecto el resultado se mostraría en la frame donde se incluyó el hipervínculo (`frame1`)). Esto lo podríamos resolver incluyendo en la definición del hipervínculo el atributo `target="frame"`, indicando con esto la frame de destino donde se mostrará el hipervínculo: ``

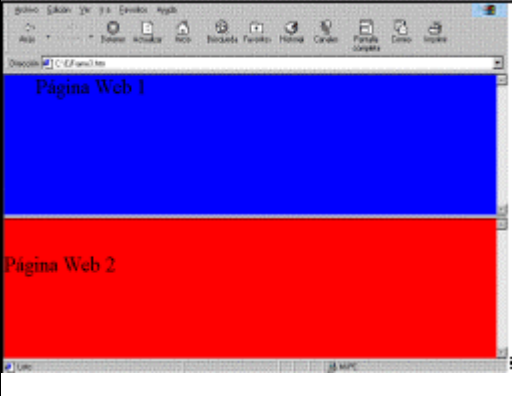
Ejemplo	Resultado
<pre><HTML> <FRAMESET rows = "50%,*"><FRAME src="Pagweb1.htm" name="Pagweb1" marginwidth = 50 scrolling="yes"> <FRAME src="Pagweb2.htm" name="Pagweb2" marginheight = 50 scrolling="yes"> </FRAMESET> </HTML></pre>	

Tabla 55

Elementos multimedia

La inclusión de elementos multimedia en nuestros documentos HTML dependerá de la capacidad del navegador para soportar estas características, como son la posibilidad de mostrar vídeo o sonidos de fondo en nuestras páginas web.

El Explorer de Microsoft incluye etiquetas que permitirán explotar su posibilidades multimedia, pero hay que tener en cuenta que éstas no son estándar.

Sonido de fondo

```
<BGSOUND src="fichero de sonido" loop= n / infinite>
{BGSOUND: Background Sound}
```

Si insertamos esta etiqueta en nuestro documento, se reproducirá el sonido especificado como sonido de fondo de nuestra página web.

Con el argumento `src` especificaremos el fichero de sonido que se va a reproducir (.wav, .mid).

Con el argumento `loop` indicaremos el número de veces (n) que se reproducirá el sonido. Si indicamos `loop=infinite` el sonido se reproducirá de manera indefinida hasta abandonar la página. Vemos un ejemplo en el Código fuente 17.

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

Código fuente 17

Reproducción de vídeo

```
<IMG dynsrc="fichero de video" loop= n / infinite start=fileopen /
mouseover controls>
```

En este caso podemos reproducir un vídeo en formato .avi dentro de nuestro documento HTML.

Con el argumento `dynsrc` especificaremos el fichero .avi a reproducir. El argumento `loop` indica el número de veces que se reproducirá el video. Con argumento `start` indicaremos si la secuencia de video empezará al abrir el fichero (`fileopen`) o cuando el cursor del ratón esté encima (`mouseover`). Si especificamos el argumento `controls` aparecerán los botones de control de vídeo. Veamos el Código fuente 18.

```
<IMG dynsrc="macarena.avi" loop=infinite start=fileopen controls>
```

Código fuente 18

Inserción de cualquier tipo de fichero

```
<EMBED src="URL del fichero" width= n ó n% height= n ó n%>
```

Con esta etiqueta podremos insertar cualquier tipo de fichero en nuestro documento: fichero de sonido, de video, gráfico, etc. Como es lógico, el navegador deberá tener la capacidad de leer el tipo de fichero especificado o ayudarse de algún otro programa externo que le permita visualizar el fichero. En caso contrario dará un mensaje de error informándonos de que no puede leer el fichero especificado.

Con el argumento `src` indicamos la URL del fichero a mostrar. Con los argumentos `width` y `height` indicaremos el tamaño del objeto insertado, estos argumentos son opcionales.

Veamos un ejemplo en el Código fuente 19.

```
<EMBED src="macarena.avi" width=100 height=100>
```

Código fuente 19

Si quiere ver más textos en este formato, visítenos en: <http://www.lalibreriadigital.com>.

Este libro tiene soporte de formación virtual a través de Internet, con un profesor a su disposición, tutorías, exámenes y un completo plan formativo con otros textos. Si desea inscribirse en alguno de nuestros cursos o más información visite nuestro campus virtual en: <http://www.almagesto.com>.

Si quiere información más precisa de las nuevas técnicas de programación puede suscribirse gratuitamente a nuestra revista *Algoritmo* en: <http://www.algoritmodigital.com>. No deje de visitar nuestra revista *Alquimia* en <http://www.eidos.es/alquimia> donde podrá encontrar artículos sobre tecnologías de la sociedad del conocimiento.

Si quiere hacer algún comentario, sugerencia, o tiene cualquier tipo de problema, envíelo a la dirección de correo electrónico lalibreriadigital@eidos.es.