

Manual de HTML



César García

Introducción al HTML

HTML es el lenguaje con el que se escriben las páginas web. Las páginas web pueden ser vistas por el usuario mediante un tipo de aplicación llamada navegador. Podemos decir por lo tanto que el HTML es el lenguaje usado por los navegadores para mostrar las páginas webs al usuario, siendo hoy en día la interface más extendida en la red.

Este lenguaje nos permite aglutinar textos, sonidos e imágenes y combinarlos a nuestro gusto. Además, y es aquí donde reside su ventaja con respecto a libros o revistas, el HTML nos permite la introducción de referencias a otras páginas por medio de los enlaces hipertexto.

El HTML se creó en un principio con objetivos divulgativos. No se pensó que la web llegara a ser un área de ocio con carácter multimedia, de modo que, el HTML se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro. Sin embargo, pese a esta deficiente planificación, si que se han ido incorporando modificaciones con el tiempo, estos son los estándares del HTML.

Esta evolución tan anárquica del HTML ha supuesto toda una serie de inconvenientes y deficiencias que han debido ser superados con la introducción de otras tecnologías accesorias capaces de organizar, optimizar y automatizar el funcionamiento de las webs. Ejemplos que pueden sonaros son las [CSS](#)¹, [JavaScript](#)² u otros. Veremos más adelante en qué consisten algunas de ellas.

Otros de los problemas que han acompañado al HTML es la diversidad de navegadores presentes en el mercado los cuales no son capaces de interpretar un mismo código de una manera unificada. Esto obliga al webmáster a, una vez creada su página, comprobar que esta puede ser leída satisfactoriamente por todos los navegadores, o al menos, los más utilizados.

Además del navegador necesario para ver los resultados de nuestro trabajo, necesitamos evidentemente otra herramienta capaz de crear la página en si. Un archivo HTML (una página) no es más que un texto. Es por ello que para programar en HTML necesitamos un **editor de textos**.

Es recomendable usar el [Bloc de notas](#) que viene con windows, u otro editor de textos sencillo. Hay que tener cuidado con algunos editores más complejos como Wordpad o Microsoft Word, pues colocan su propio código especial al guardar las páginas y HTML es **únicamente texto plano**, con lo que podremos tener problemas.

Es importante tener claro todo ello puesto que en función de vuestros objetivos puede que, más que aprender HTML, resulte más interesante aprender el uso de una aplicación para la creación de páginas.

Así pues, una página es un archivo donde está contenido el código HTML en forma de texto. Estos archivos tienen extensión **.html** o **.htm** (es indiferente cuál utilizar). De modo que cuando programemos en HTML lo haremos con un editor de textos y guardaremos nuestros trabajos con extensión **.html**, por ejemplo mipágina.html.

¹ CSS, es una tecnología que nos permite crear páginas web de una manera más exacta.

² *Javascript* es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web.

Sintaxis del HTML

El HTML es un lenguaje que basa su sintaxis en un elemento de base al que llamamos etiqueta. La etiqueta presenta frecuentemente dos partes:

Una apertura de forma general <etiqueta>

Un cierre de tipo </etiqueta>

Todo lo incluido en el interior de esa etiqueta sufrirá las modificaciones que caracterizan a esta etiqueta. Así por ejemplo:

Las etiquetas y definen un texto en negrita. Si en nuestro documento HTML escribimos una frase con el siguiente código:

```
<b>Esto esta en negrita</b>
```

El resultado Será:

Esto esta en negrita

Las etiquetas <p> y </p> definen un párrafo. Si en nuestro documento HTML escribiéramos:

```
<p>Hola, estamos en el párrafo 1</p>
<p>Ahora hemos cambiado de párrafo</p>
```

El resultado sería:

Hola, estamos en el párrafo 1
Ahora hemos cambiado de párrafo

Partes de un documento HTML

Además de todo esto, **un documento HTML ha de estar delimitado por la etiqueta <html> y </html>**. Dentro de este documento, podemos asimismo distinguir dos partes principales:

El encabezado, delimitado por <head> y </head> donde colocaremos etiquetas de índole informativo como por ejemplo el título de nuestra página.

El cuerpo, flanqueado por las etiquetas <body> y </body>, que será donde colocaremos nuestro texto e imágenes delimitados a su vez por otras etiquetas como las que hemos visto.

El resultado es un documento con la siguiente estructura:

```
<html>
```

```
<head>
```

Etiquetas y contenidos del encabezado

Datos que no aparecen en nuestra página pero que son importantes para catalogarla: Título, palabras clave,...

```
</head>
```

<body>

Etiquetas y contenidos del cuerpo

Parte del documento que será mostrada por el navegador: Texto e imágenes
</body>

</html>

Las mayúsculas o minúsculas son indiferentes al escribir etiquetas

A notar que las etiquetas pueden ser escritas con cualquier tipo de combinación de mayúsculas y minúsculas. <html>, <HTML> o <HtMl> son la misma etiqueta. Resulta sin embargo aconsejable acostumbrarse a escribirlas en minúscula ya que otras tecnologías que pueden convivir con nuestro HTML (XML por ejemplo) no son tan permisivas y nunca viene mal coger buenas costumbres desde el principio para evitar fallos triviales en un futuro.

Tu primera página

Podemos ya con estos conocimientos, y alguno que otro más, crear nuestra primera página. Para ello, abre tu editor de textos y copia y pega el siguiente texto en un nuevo documento.

<html>

<head>

<title>Cocina Para Todos</title>

</head>

<body>

<p>Bienvenido,</p>

<p>Estás en la página Comida para Todos.</p>

<p>Aquí aprenderás recetas fáciles y deliciosas.</p>

</body>

</html>

Ahora guarda ese archivo con extensión .html o .htm en tu disco duro. Para ello accedemos al menú Archivo y seleccionamos la opción Guardar como. En la ventana elegimos el directorio donde deseamos guardarlo y colocaremos su nombre, por ejemplo mi_pagina.html

Utiliza nombres en tus archivos que tengan algunas normas básicas para ahorrarte disgustos y líos.

Lo ideal es que no utilices acentos ni espacios ni otros caracteres raros. También te ayudará escribir siempre las letras en minúsculas.

Esto no quiere decir que debes hacer nombres de archivos cortos, es mejor hacerlos descriptivos para que te aclaren lo que hay dentro. Algún carácter como el guión "-" o el guión bajo "_" te puede ayudar a separar las palabras. Por ejemplo quienes_somos.html

formato de párrafos en HTML

En este capítulo veremos con más detalle las etiquetas más ampliamente utilizadas y ejemplificaremos algunas de ellas posteriormente.

Formatear un texto pasa por tareas tan evidentes como definir los párrafos, justificarlos, introducir viñetas, numeraciones o bien poner en negrita, itálica...

Hemos visto que para definir los párrafos nos servimos de la etiqueta `<p>` que introduce un salto y deja una línea en blanco antes de continuar con el resto del documento.

Podemos también usar la etiqueta `
`, de la cual no existe su cierre correspondiente (`</br>`), para realizar un simple retorno de carro con lo que no dejamos una línea en blanco sino que solo cambiamos de línea.

Podéis comprobar que cambiar de línea en nuestro documento HTML sin introducir alguna de estas u otras etiquetas no implica en absoluto un cambio de línea en la página visualizada. En realidad el navegador introducirá el texto y no cambiara de línea a no ser que esta llegue a su fin o bien lo especifiquemos con la etiqueta correspondiente.

Los párrafos delimitados por etiquetas `<p>` pueden ser fácilmente justificados a la izquierda, centro o derecha especificando dicha justificación en el interior de la etiqueta por medio de un atributo **align**. Un atributo no es más que un parámetro incluido en el interior de la etiqueta que ayuda a definir el funcionamiento de la etiqueta de una forma más personal. Veremos a lo largo de este manual cantidad de atributos muy útiles para todo tipo de etiquetas.

Así, si deseásemos introducir un **texto alineado a la izquierda** escribiríamos:

```
<p align="left">Texto alineado a la izquierda</p>
```

El resultado sería:

Texto alineado a la izquierda

Para una **justificación al centro**:

```
<p align="center">Texto alineado al centro</p>
```

que daría:

Texto alineado al centro

Para **justificar a la derecha**:

```
<p align="right">Texto alineado a la derecha</p>
```

cuyo efecto sería:

Texto alineado a la derecha

Como veis, en cada caso el atributo align toma determinados valores que son escritos entre comillas. En algunas ocasiones necesitamos especificar algunos

atributos para el correcto funcionamiento de la etiqueta. En otros casos, el propio navegador toma un valor definido por defecto. Para el caso de align, el valor por defecto es left.

Los atributos tienen sus valores indicados entre comillas ("), pero si no los indicamos entre comillas también funcionará en la mayoría de los casos. Sin embargo, es aconsejable que pongamos siempre las comillas para acostumbrarnos a utilizarlas, por dar homogeneidad a nuestros códigos y para evitar errores futuros en sistemas más quisquillosos.

Imaginemos un texto relativamente largo donde todos los párrafos están alineados a la izquierda (por ejemplo). Una forma de simplificar nuestro código y de evitar introducir continuamente el atributo align sobre cada una de nuestras etiquetas es utilizando la etiqueta **<div>**.

Esta etiqueta por si sola no sirve para nada. Tiene que estar acompañada del atributo align y lo que nos permite es alinear cualquier elemento (párrafo o imagen) de la manera que nosotros deseemos.

Así, el código:

```
<p align="left">Parrafo1</p>
<p align="left"> Parrafo3</p>
<p align="left"> Parrafo2</p>
```

es equivalente a:

```
<div align="left">
<p>Parrafo1</p>
<p>Parrafo2</p>
<p>Parrafo3</p>
</div>
```

Como hemos visto, la etiqueta <div> marca divisiones en las que definimos un mismo tipo de alineado.

Ejemplo práctico:

Para practicar un poco lo que acabamos de ver vamos a proponer un ejercicio que podéis resolver en vuestros ordenadores. Simplemente queremos construir una página que tenga, por este orden:

- 2 Párrafos centrados
- 3 Párrafos alineados a la derecha
- Un salto de línea triple
- 1 párrafo alineado a la izquierda

Encabezados

Existen otras etiquetas para definir párrafos especiales, formateados como títulos. Son los encabezados o Header en inglés. Como decimos, son etiquetas que formatean el texto como un titular, para lo cual asignan un tamaño mayor de letra y colocan el texto en negrita.

Hay varios tipos de encabezados, que se diferencian en el tamaño de la letra que utilizan. La etiqueta en concreto es la <h1>, para los encabezados más grandes,

<h2> para los de segundo nivel y así hasta <h6> que es el encabezado más pequeño.

Los encabezados implican también una separación en párrafos, así que todo lo que escribamos dentro de <h1> y </h1> (o cualquier otro encabezado) se colocará en un párrafo independiente.

Podemos ver cómo se presentan algunos encabezados a continuación.

<h1>Encabezado de nivel 1</h1>

Se verá de esta manera en la página:

Encabezado de nivel 1

Los encabezados, como otras etiquetas de HTML, soportan el atributo align. Vemos un ejemplo de encabezado de nivel 2 alineado al centro.

<h2 align="center">Encabezado de nivel 2</h2>

Se verá de esta manera en la página:

Encabezado de nivel 2

Ejemplo práctico:

Otro ejercicio interesante es construir una página web que contenga todos los encabezados posibles. Se puede ver a continuación.

```
<html>
```

```
<head>
```

```
<title>Todos los encabezados</title>
```

```
</head>
```

```
<body>
```

```
<h1>Encabezado de nivel 1</h1>
```

```
<h2>Encabezado de nivel 1</h2>
```

```
<h3>Encabezado de nivel 1</h3>
```

```
<h4>Encabezado de nivel 1</h4>
```

```
<h5>Encabezado de nivel 1</h5>
```

```
<h6>Encabezado de nivel 1</h6>
```

```
</body>
```

```
</html>
```

No debemos utilizar las etiquetas de encabezado para formatear el texto, es decir, si queremos colocar un tipo de letra más grande y en negrita debemos utilizar las etiquetas que existen para ello (que veremos en seguida). Los encabezados son para colocar titulares en páginas web y es el navegador el responsable de formatear el texto de manera que parezca un titular. Cada navegador, pues, puede formatear el texto a su gusto con tal de que parezca un titular.

formateando el texto

Además de todo lo relativo a la organización de los párrafos, uno de los aspectos primordiales del formateo de un texto es el de la propia letra. Resulta muy común y práctico presentar texto resaltado en negrita, itálica y otros. Paralelamente el uso de índices, subíndices resulta vital para la publicación de textos científicos. Todo esto y mucho más es posible por medio del HTML a partir de multitud de etiquetas entre las cuales vamos a destacar algunas.

Negrita

Podemos escribir texto en negrita incluyéndolo dentro de las etiquetas `` y `` (bold). Esta misma tarea es desempeñada por `` y `` siendo ambas equivalentes. Nosotros nos inclinamos por la primeras por simple razón de esfuerzo.

La diferencia entre `` y `` se podrá entender ahora. Mientras que `` significa simplemente negrita y todos los navegadores la interpretarán como negrita, `` es una etiqueta que significa que se tiene que resaltar fuertemente el texto y cada navegador es el responsable de resaltarlo como desee.

Escribiendo un código de este tipo:

```
<b>Texto en negrita</b>
```

Obtenemos este resultado:

Texto en negrita

Itálica

También en este caso existen dos posibilidades, una corta: `<i>` e `</i>` (italic) y otra un poco más larga: `` y ``. En este manual, y en la mayoría de las páginas que veréis por ahí, os encontraréis con la primera forma sin duda más sencilla a escribir y a acordarse.

He aquí un ejemplo de texto en itálica:

```
<i>Texto en itálica</i>
```

Que da el siguiente efecto:

Texto en itálica

Subrayado

El HTML nos propone también para el subrayado el par de etiquetas: `<u>` y `</u>` (underlined). Sin embargo, el uso de subrayados ha de ser aplicado con mucha precaución dado que los enlaces hipertexto van, a no ser que se indique lo contrario, subrayados con lo que podemos confundir al lector y apartarlo del verdadero interés de nuestro texto.

Subíndices y supraíndices

Este tipo de formato resulta de extremada utilidad para textos científicos. Las etiquetas empleadas son:

`^{` y `}` para los supraíndices
`_{` y `}` para los subíndices

Aquí tenéis un ejemplo:

La `¹³CC₃H₄CINOS es un heterociclo
alergeno enriquecido`

El resultado:

La ¹³CC₃H₄CINOS es un heterociclo alergeno enriquecido

Anidar etiquetas

Todas estas etiquetas y por supuesto el resto de las vistas y que veremos más adelante pueden ser anidadas unas dentro de otras de manera a conseguir resultados diferentes. Así, podemos sin ningún problema crear texto en negrita e itálica embebiendo una etiqueta dentro de la otra:

``Esto sólo está en negrita `<i>`y esto en negrita e itálica`</i>`

Esto nos daría:

Esto sólo está en negrita y esto en negrita e itálica

Cuando anides etiquetas HTML hazlo correctamente. Nos referimos a que si abres etiquetas dentro de otra más principal, antes de cerrar la etiqueta principal cierras las etiquetas que hayas abierto dentro de ella.

Debemos evitar códigos como el siguiente:

``Esto está en negrita e `<i>`itálica`</i>`

En favor de códigos con etiquetas correctamente anidadas:

``Esto está en negrita e `<i>`itálica`</i>`

Esto es muy aconsejable, aunque los navegadores entiendan bien las etiquetas mal anidadas.

Color, tamaño y tipo de letra

A pesar de que por razones de homogeneidad y sencillez de código este tipo de formatos son controlados actualmente por hojas de estilo en cascada, existe una forma clásica y directa de definir color tamaño y tipo de letra de un texto determinado.

Esto se hace a partir de la etiqueta `` y su cierre correspondiente. Dentro de esta etiqueta deberemos especificar los atributos correspondientes a cada uno de estos parámetros que deseamos definir. A continuación os comentamos los atributos principales de esta etiqueta:

Atributo face

Define el tipo de letra. Este atributo es interpretado por versiones de Netscape a partir de la 3 y de MSIE 3 o superiores. Otros navegadores las ignoran completamente y muestran el texto con la fuente que utilizan.

Hay que tener cuidado con este atributo ya que cada usuario, dependiendo de la plataforma que utilice, puede no disponer de los mismos tipos de letra que nosotros con lo que, si nosotros elegimos un tipo del que no dispone, el navegador se verá forzado a mostrar el texto con la fuente que utiliza por defecto (suele ser Times New Roman). Para evitar esto, dentro del atributo suelen seleccionarse varios tipos de letra separados por comas. En este caso el navegador comprobará que dispone del primer tipo enumerado y si no es así, pasará al segundo y así sucesivamente hasta encontrar un tipo que posea o bien acabar la lista y poner la fuente por defecto. Veamos un ejemplo.

```
<font face="Comic Sans MS,arial,verdana">Este texto tiene otra tipografía</font>
```

Que se visualizaría así en una página web.

Este texto tiene otra tipografía

Aquí tenemos un ejemplo de atributo cuyo valor debe estar limitado por comillas ("). Habíamos dicho que las comillas eran opcionales en los atributos, sin embargo esto no es así siempre. Si el valor del atributo contiene espacios, como es el caso de:

```
face="Comic Sans MS,arial,verdana"
```

debemos colocar las comillas para limitarlo. En caso de no tener comillas

```
face=Comic Sans MS,arial,verdana
```

se entendería que face=Comic, pero no se tendría en cuenta todo lo que sigue, porque HTML no lo asociaría al valor del atributo. En este caso HTML pensaría que las siguientes palabras (después del espacio) son otros atributos, pero como no los conoce como atributos simplemente los desestimaré.

Atributo size

Define el tamaño de la letra. Este tamaño puede ser absoluto o relativo.

Si hablamos en términos absolutos, existen 7 niveles de tamaño distintos numerados de 1 a 7 por orden creciente. Elegiremos por tanto un valor size="1" para la letra más pequeña o size="7" para la más grande.

```
<font size=4>Este texto es más grande</font>
```

Que se visualizaría así en una página web.

Este texto es más grande

Podemos asimismo modificar el tamaño de nuestra letra con respecto al del texto mostrado precedentemente definiendo el número de niveles que queremos subir o bajar en esta escala de tamaños por medio de un signo + o -. De este modo, si definimos nuestro atributo como size="+1" lo que queremos decir es que aumentamos de un nivel el tamaño de la letra. Si estábamos escribiendo previamente en 3, pasaremos automáticamente a 4.

Los tamaños reales que veremos en pantalla dependerán de la definición y del tamaño de fuente elegido por el usuario en el navegador. Este tamaño de fuente puede ser definido en el Explorer yendo al menú superior, Ver/Tamaño de la fuente. En Netscape elegiremos View/Text Size. Esta flexibilidad puede en más de una ocasión resultarnos embarazosa ya que en muchos casos desearemos que el tamaño del texto permanezca constante para que éste quepa en un determinado espacio. Veremos en su momento que esta prefijación del tamaño puede ser llevada a cabo por las hojas de estilo en cascada.

Atributo color

El color del texto puede ser definido mediante el atributo color. Cada color es a su vez definido por un número hexadecimal que esta compuesto a su vez de tres partes. Cada una de estas partes representa la contribución del rojo, verde y azul al color en cuestión.

Podéis entender cómo funciona esta numeración y cuáles son los colores que resultan más compatibles en el **Anexo: Los colores y HTML**.

Por otra parte, es posible definir de una manera inmediata algunos de los colores más frecuentemente usados para los que se ha creado un nombre más mnemotécnico:

```
<font color="red">Este texto está en rojo</font>
```

Que se visualizaría así en una página web.

Este texto está en rojo

Con todo esto estamos ya en disposición de crear un texto formateado de una forma realmente elaborada.

Ejercicio práctico:

Pongamos pues en practica todo lo que hemos aprendido en estos capítulos haciendo un ejercicio consistente en una página que tenga las siguientes características:

- Un titular con encabezado de nivel 1, en itálica y color verde oliva.
- Un segundo titular con encabezado de nivel 2, también de color verde oliva.
- Todo el texto de la página deberá presentarse con una fuente distinta de la fuente por defecto. Por ejemplo "Comic Sans MS" y en caso de que ésta no esté en el sistema que se coloque la fuente "Arial".

Atributos para páginas

Las páginas HTML pueden construirse con variedad de atributos que le pueden dar un aspecto a la página muy personalizado. Podemos definir atributos como el color de fondo, el color del texto o de los enlaces. Estos atributos se definen en la etiqueta <body> y, como decíamos son generales a toda la página.

Lo mejor para explicar su funcionamiento es verlos uno por uno.

Atributos para fondos

bgcolor: especificamos un color de fondo para la página. En el **Anexo: Los colores y html**, aprendemos a construir cualquier color, con su nombre o su valor RGB. El color de fondo que podemos asignar con bgcolor es un color plano, es decir el mismo para toda la superficie del navegador.

background: sirve para indicar la colocación de una imagen como fondo de la página. La imagen se coloca haciendo un mosaico, es decir, se repite muchas veces hasta ocupar todo el espacio del fondo de la página. En capítulos más adelante veremos como se insertan imágenes con HTML y los tipos de imágenes que se pueden utilizar.

Siempre que coloquemos una imagen de fondo, debemos poner también un color de fondo cercano al color de la imagen.

Esto se debe a que, al colocar una imagen de fondo, el texto de la página debemos colocarlo en un color que contraste suficientemente con dicho fondo. Si el visitante no puede ver el fondo por cualquier cuestión (Por ejemplo tener deshabilitada la carga de imágenes) puede que el texto no contraste lo suficiente con el color de fondo por defecto de la web.

Si la imagen de fondo es oscura, tendremos que poner un texto claro para que se pueda leer. Si el visitante que accede a la página no ve la imagen de fondo, le saldrá el fondo por defecto, que generalmente es blanco, de modo que al tener un texto con color claro sobre un fondo blanco, nos pasará que no podremos leer el texto convenientemente.

Ocurre parecido cuando se está cargando la página. Si todavía no ha llegado a nuestro sistema la imagen de fondo, se verá el fondo que hayamos seleccionado con bgcolor y es interesante que sea parecido al color de la imagen para que se pueda leer el texto mientras se carga la imagen de fondo. El resultado es el siguiente:

Ejemplo de fondo

Vamos a colocar esta imagen como fondo en la página.



La imagen se llama fondo.jpg y suponemos que se encuentra en el mismo directorio que la página. En este caso se colocaría la siguiente etiqueta:

```
<body background="fondo.jpg">
```

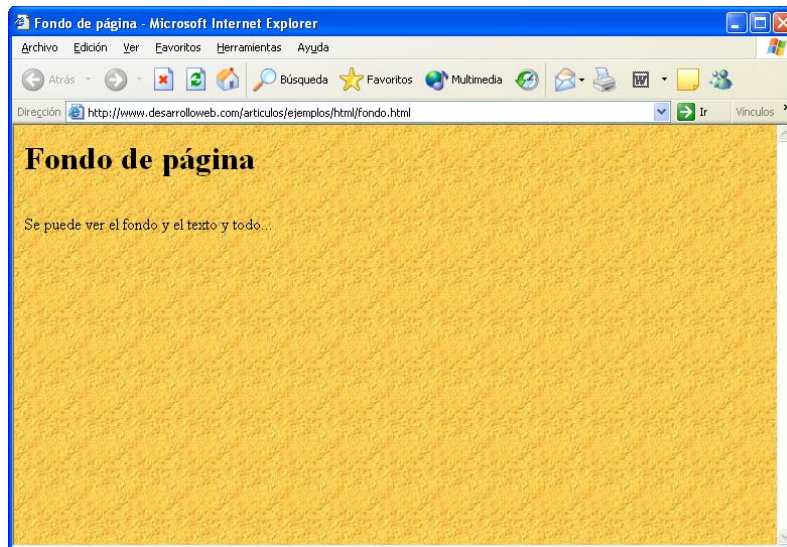
En la práctica:

```
<html>
<head>
  <title>Fondo de página</title>
</head>

<body background="../../images/html/fondo.jpg" bgcolor=gold>

<h1>Fondo de página</h1>
<br>
Se puede ver el fondo y el texto y todo...
</body>
</html>
```

El resultado seria:



Color del texto

text: este atributo sirve para asignar el color del texto de la página. Por defecto es el negro.

Además del color del texto, tenemos tres atributos para asignar el color de los enlaces de la página. Ya debemos saber que los enlaces deben diferenciarse del resto del texto de la página para que los usuarios puedan identificarlos fácilmente. Para ello suelen aparecer subrayados y con un color más vivo que el texto. Los tres atributos son los siguientes:

link: el color de los enlaces que no han sido visitados. (por defecto es azul clarito)

vlink: el color de los enlaces visitados. La "v" viene justamente de la palabra visitado. Es el color que tendrán los enlaces que ya hemos visitado. Por defecto su color es morado. Este color debería ser un poco menos vivo que el color de los enlaces normales.

alink: es el color de los enlaces activos. Un enlace está activo en el preciso instante que se pulsa. A veces es difícil darse cuenta cuando un enlace está activo porque en el momento en el que se activa es porque lo estamos pulsando y en ese caso el navegador abandonará la página rápidamente y no podremos ver el enlace activo más que por unos instantes mínimos.

Ejemplo de color del texto:

Vamos a ver una página donde el color de fondo sea negro, y los colores del texto y los enlaces sean claros. Pondremos el color de texto blanco y los enlaces amarillos, más resaltados los que no estén visitados y menos resaltados los que ya están visitados. Para ello escribiríamos la etiqueta <body> así:

```
<body bgcolor="#000000" text="#ffffff" link="#ffff33" alink="#ffffcc"
alink="#ffff00">
```

Márgenes

Con otros atributos de la etiqueta <body> se pueden asignar espacios de margen en las páginas, lo que es muy útil para eliminar los márgenes en blanco que aparecen a los lados, arriba y debajo de la página. Estos atributos son distintos para Internet Explorer y para Netscape Navigator, por lo que debemos utilizarlos todos si queremos que todos los navegadores los interpreten perfectamente.

leftmargin: para indicar el margen a los lados de la página. Válido para iexplorer.

topmargin: para indicar el margen arriba y debajo de la página. Para iexplorer.

marginwidth: la contrapartida de leftmargin para Netscape. (Margen a los lados)

marginheight: igual que topmargin, pero para Netscape. (Margen arriba y abajo)

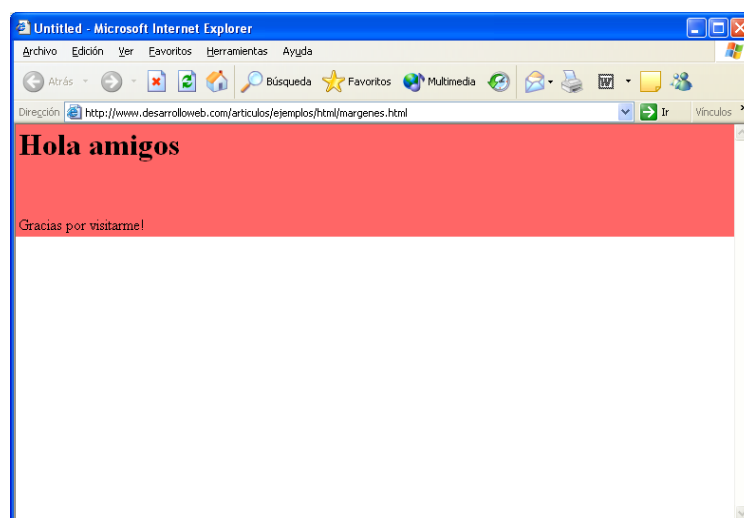
Ejemplo práctico:

Vamos a ver una página sin márgenes, por si alguien necesita ver el ejemplo en estas líneas.

```
<html>
<head>
    <title>Untitled</title>
</head>

<body topmargin=0 leftmargin=0 marginheight=0 marginwidth=0 bgcolor="ffffff">
<table width=100% bgcolor=ff6666><tr><td>
<h1>Hola amigos</h1>
<br>
<br>
Gracias por visitarme!
</td></tr></table>
</body>
</html>
```

Esta página tiene el fondo blanco y dentro una tabla con el fondo rojo. En la página podremos ver que la tabla ocupa el espacio en la página sin dejar sitio para ningún tipo de margen.



Listas

Las posibilidades que nos ofrece el HTML en cuestión de tratamiento de texto son realmente notables. No se limitan a lo visto hasta ahora, sino que van más lejos todavía. Varios ejemplos de ello son las listas, que sirven para enumerar y definir elementos, los textos preformateados y las cabeceras o títulos.

Las listas son utilizadas para citar, numerar y definir objetos. También son utilizadas corrientemente para desplazar el comienzo de línea hacia la derecha.

Podemos distinguir tres tipos de listas:

- Listas desordenadas
- Listas ordenadas
- Listas de definición

Las veremos detenidamente una a una.

Listas desordenadas

Son delimitadas por las etiquetas `` y `` (unordered list). Cada uno de los elementos de la lista es citado por medio de una etiqueta `` (sin cierre, aunque no hay inconveniente en colocarlo). La cosa queda así:

```
<p>Países del mundo</p>
<ul>
  <li>Argentina
  <li>Perú
  <li>Chile
</ul>
```

El resultado:

Países del mundo

- Argentina
- Perú
- Chile

Podemos definir el tipo de viñeta empleada para cada elemento. Para ello debemos especificarlo por medio del atributo `type` incluido dentro de la etiqueta de apertura ``, si queremos que el estilo sea válido para toda la lista, o dentro de la etiqueta `` si queremos hacerlo específico de un solo elemento. La sintaxis es del siguiente tipo:

```
<ul type="tipo de viñeta">
```

donde tipo de viñeta puede ser uno de los siguientes:

circle
disc
square

Vamos a ver un ejemplo de lista con un cuadrado en lugar de un redondel, y en el último elemento colocaremos un círculo. Para ello vamos a colocar el atributo type en la etiqueta , con lo que afectará a todos los elementos de la lista.

```
<ul type="square">
<li>Elemento 1
<li>Elemento 2
<li>Elemento 3
<li type="circle">Elemento 4
</ul>
```

Que tiene como resultado

- Elemento 1
- Elemento 2
- Elemento 3
- Elemento 4

Continuamos estudiando las listas de HTML, con las que crear estructuras atractivas para presentar la información.

Listas ordenadas

En este caso usaremos las etiquetas (ordered list) y su cierre. Cada elemento sera igualmente precedido de su etiqueta .

Pongamos un ejemplo:

```
<p>Reglas de comportamiento en el trabajo</p>
<ol>
<li>El jefe siempre tiene la razón
<li>En caso de duda aplicar regla 1
</ol>
```

El resultado es:

Reglas de comportamiento en el trabajo

1. El jefe siempre tiene la razón
2. En caso de duda aplicar regla 1

Del mismo modo que para las listas desordenadas, las listas ordenadas ofrecen la posibilidad de modificar el estilo. En concreto nos es posible especificar el tipo de numeración empleado eligiendo entre números (1, 2, 3...), letras (a, b, c...) y sus mayúsculas (A, B, C,...) y números romanos en sus versiones mayúsculas (I, II, III,...) y minúsculas (i, ii, iii,...).

Para realizar dicha selección hemos de utilizar, como para el caso precedente, el atributo type, el cual será situado dentro de la etiqueta . Los valores que puede tomar el atributo en este caso son:

- 1 Para ordenar por números
- a Por letras del alfabeto
- A Por letras mayúsculas del alfabeto
- i Ordenación por números romanos en minúsculas
- I Ordenación por números romanos en mayúsculas

Puede que en algún caso deseemos comenzar nuestra enumeración por un número o letra que no tiene por qué ser necesariamente el primero de todos. Para solventar esta situación, podemos utilizar un segundo atributo, `start`, que tendrá como valor un número. Este número, que por defecto es 1, corresponde al valor a partir del cual comenzamos a definir nuestra lista. Para el caso de las letras o los números romanos, el navegador se encarga de hacer la traducción del número a la letra correspondiente.

He aquí un ejemplo usando este tipo de atributos:

```
<p>Ordenamos por numeros</p>
<ol type="1">
<li>Elemento 1
<li> Elemento 2
</ol>
```

```
<p>Ordenamos por letras</p>
<ol type="a">
<li>Elemento a
<li> Elemento b
</ol>
```

```
<p>Ordenamos por números romanos empezando por el 10</p>
```

```
<ol type="i" start="10">
<li>Elemento x
<li> Elemento xi
</ol>
```

El resultado:

Ordenamos por números

1. Elemento 1
2. Elemento 2

Ordenamos por letras

- a. Elemento a
- b. Elemento b

Ordenamos por números romanos empezando por el 10

- x. Elemento x
- xi. Elemento xi

Listas de definición

Cada elemento es presentado junto con su definición. La etiqueta principal es `<dl>` y `</dl>` (definition list). Las etiquetas del elemento y su definición son `<dt>` (definition term) y `<dd>` (definition definition) respectivamente.

He aquí un código que podrá aclarar este sistema:

```
<p>Diccionario de la Real Academia</p>
<dl>
  <dt>Brujula
  <dd>Señórula montada en una escóbula
  <dt>Oreja
  <dd>Sesenta minutejos
</dl>
```

El efecto producido:

Diccionario de la Real Academia

```
Brujula
  Señórula montada en una escóbula
Oreja
  Sesenta minutejos
```

Fijaos en que cada línea <dd> esta desplazada hacia la izquierda. Este tipo de etiquetas son usadas a menudo con el propósito de crear textos más o menos desplazados hacia la izquierda.

El código:

```
<dl>
<dd>Primer nivel de desplazamiento
  <dl>
    <dd>Segundo nivel de desplazamiento
    <dl>
      <dd>Tercer nivel de desplazamiento
    </dl>
  </dl>
</dl>
```

El resultado:

```
Primer nivel de desplazamiento
Segundo nivel de desplazamiento
Tercer nivel de desplazamiento
```

Anidando listas

Nada nos impide utilizar todas estas etiquetas de forma anidada como hemos visto en otros casos. De esta forma, podemos conseguir listas mixtas como por ejemplo:

```
<p>Ciudades del mundo</p>
<ul>
  <li>Argentina
  <ol>
    <li>Buenos Aires
    <li>Bariloche
  </ol>
  <li>Uruguay
</ol>
```

```
<li>Montevideo
<li>Punta del Este
</ol>
</ul>
```

De esta forma creamos una lista como esta:

Ciudades del mundo

- Argentina
 1. Buenos Aires
 2. Bariloche
- Uruguay
 1. Montevideo
 2. Punta del Este

Caracteres especiales

Una página web se ha de ver en países distintos, que usan conjuntos de caracteres distintos. El lenguaje HTML nos ofrece un mecanismo por el que podemos estar seguros que una serie de caracteres raros se van a ver bien en todos los ordenadores del mundo, independientemente de su juego de caracteres.

Este conjunto son los caracteres especiales. Cuando queremos poner uno de estos caracteres en una página, debemos sustituirlo por su código.

Por ejemplo, la "á" (a minúscula acentuada) se escribe "á" de modo que la palabra página se escribiría en una página HTML de este modo:
páaacute;gina

En el Anexo: Caracteres especiales se muestra una extensa lista de los más utilizados.

Enlaces en HTML

Hasta aquí, hemos podido ver que una página web es un archivo HTML en el que podemos incluir, entre otras cosas, textos formateados a nuestro gusto e imágenes (las veremos enseguida). Del mismo modo, un sitio web podrá ser considerado como el conjunto de archivos, principalmente páginas HTML e imágenes, que constituyen el contenido al que el navegante tiene acceso.

Sin embargo, no podríamos hablar de navegante o de navegación si estos archivos HTML no estuviesen debidamente conectados entre ellos y con el exterior de nuestro sitio por medio de enlaces hipertexto. En efecto, el atractivo original del HTML radica en la posible puesta en relación de los contenidos de los archivos introduciendo referencias bajo forma de enlaces que permitan un acceso rápido a la información deseada. De poco serviría en la red tener páginas aisladas a las que la gente no puede acceder y desde las que la gente no puede saltar a otras.

Un enlace puede ser fácilmente detectado en una página. Basta con deslizar el puntero del ratón sobre las imágenes o el texto y ver como cambia de su forma original transformándose por regla general en una mano con un dedo señalador. Adicionalmente, estos enlaces suelen ir, en el caso de los textos, coloreados y subrayados para que el usuario no tenga dificultad en reconocerlos. Si no especificamos lo contrario (ya tendremos ocasión de explicar como), estos enlaces

texto estarán subrayados y coloreados en azul. En el caso de las imágenes que sirvan de enlace, veremos que están delimitadas por un marco azul por defecto.

Para colocar un enlace, nos serviremos de las etiquetas `<a>` y ``. Dentro de la etiqueta de apertura deberemos especificar asimismo el destino del enlace. Este destino será introducido bajo forma de atributo, el cual lleva por nombre href.

La sintaxis general de un enlace es por tanto de la forma:

```
<a href="destino">contenido</a>
```

Siendo el *contenido* un texto o una imagen. Es la parte de la página que se colocará activa y donde deberemos pulsar para acceder al enlace.

Por su parte, *destino* será una página, un correo electrónico o un archivo. En función del destino los enlaces son clásicamente agrupados del siguiente modo:

- **Enlaces internos:** los que se dirigen a otras partes dentro de la misma página.
- **Enlaces locales:** los que se dirigen a otras páginas del mismo sitio web.
- **Enlaces remotos:** los dirigidos hacia páginas de otros sitios web.
- **Enlaces con direcciones de correo:** para crear un mensaje de correo dirigido a una dirección.
- **Enlaces con archivos:** para que los usuarios puedan hacer download de ficheros.

Enlaces internos

Son los enlaces que apuntan a un lugar diferente dentro de la misma página. Este tipo de enlaces son esencialmente utilizados en páginas donde el acceso a los contenidos puede verse dificultado debido al gran tamaño de la misma. Mediante estos enlaces podemos ofrecer al visitante la posibilidad de acceder rápidamente al principio o final de la página o bien a diferentes párrafos o secciones.

Para crear un enlace de este tipo es necesario, aparte del enlace de origen propiamente dicho, un segundo enlace que será colocado en el destino. Veamos más claramente como funcionan estos enlaces con un ejemplo sencillo:

Supongamos que queremos crear un enlace que apunte al final de la página. Lo primero será colocar nuestro enlace origen. Lo pondremos aquí mismo y lo escribiremos del siguiente modo:

```
<a href="#abajo">Ir abajo</a>
```

Como podéis ver, el contenido del enlace es el texto "Ir abajo" y el destino, abajo, es un punto de la misma página que todavía no hemos definido. Ojo al símbolo #; es él quien especifica al navegador que el enlace apunta a una sección en particular.

En segundo lugar, hay que generar un enlace en el destino. Este enlace llevara por nombre abajo para poder distinguirlo de los otros posibles enlaces realizados dentro de la misma página. En este caso, la etiqueta que escribiremos será ésta:

```
<a name="abajo"></a>
```

A decir verdad, estos enlaces, aunque útiles, no son los más extendidos de cuantos hay. La tendencia general es la de crear páginas (archivos) independientes con tamaños más reducidos enlazados entre ellos por enlaces locales (los veremos enseguida). De esta forma evitamos el exceso de tiempo de carga de un archivo y la introducción de exceso de información que pueda desviar la atención del usuario.

Una aplicación corriente de estos enlaces consiste en poner un pequeño índice al principio de nuestro documento donde introducimos enlaces origen a las diferentes secciones. Paralelamente, al final de cada sección introducimos un enlace que apunta al índice de manera que podamos guiar al navegante en la búsqueda de la información útil para él.

Enlaces locales

Como hemos dicho, un sitio web esta constituido de páginas interconexas. En el capítulo anterior hemos visto como enlazar distintas secciones dentro de una misma página. Nos queda pues estudiar la manera de relacionar los distintos documentos HTML que componen nuestro sitio web.

Para crear este tipo de enlaces, hemos de crear una etiqueta de la siguiente forma:

```
<a href="archivo.html">contenido</a>
```

Por regla general, para una mejor organización, los sitios suelen estar ordenados por directorios. Estos directorios suelen contener diferentes secciones de la página, imágenes, sonidos...Es por ello que en muchos casos no nos valdrá con especificar el nombre del archivo, sino que tendremos que especificar además el directorio en el que nuestro archivo.html esta alojado.

Si habéis trabajado con MS-DOS no tendréis ningún problema para comprender el modo de funcionamiento. Tan solo hay que tener cuidado en usar la barra "/" en lugar de la contrabarra "\".

Para aquellos que no saben como mostrar un camino de un archivo, aquí van una serie de indicaciones que os ayudaran a comprender la forma de expresarlos. No resulta difícil en absoluto y con un poco de practica lo haréis prácticamente sin pensar.

1. Hay que situarse mentalmente en el directorio en el que se encuentra la página con el enlace.
2. Si la página destino esta en un directorio incluido dentro del directorio en el que nos encontramos, hemos de marcar el camino enumerando cada uno de los directorios por los que pasamos hasta llegar al archivo y separándolos por el símbolo barra "/". Al final obviamente, escribimos el archivo.
3. Si la página destino se encuentra en un directorio que incluye el de la página con el enlace, hemos de escribir dos puntos y una barra "../" tantas veces como niveles subamos en la arborescencia hasta dar con el directorio donde esta emplazado el archivo destino.
4. Si la página se encuentra en otro directorio no incluido ni incluyente del archivo origen, tendremos que subir como en la regla 3 por medio de "../" hasta encontrar un directorio que englobe el directorio que contiene a la página destino. A continuación haremos como en la regla 2. Escribiremos todos los directorios por los que pasamos hasta llegar al archivo.

Ejemplo:

Para clarificar este punto podemos hacer un ejemplo a partir de la estructura de directorios de la imagen.



Para hacer un enlace desde index.html hacia yyy.html:

```
<a href="seccion1/paginas/yyy.html">contenido</a>
```

Para hacer un enlace desde xxx.html hacia yyy.html:

```
<a href="../../seccion1/paginas/yyy.html">contenido</a>
```

Para hacer un enlace desde yyy.html hacia xxx.html:

```
<a href="../../seccion2/xxx.html">contenido</a>
```

Los enlaces locales pueden a su vez apuntar ya no a la página en general sino más precisamente a una sección concreta. Este tipo de enlaces resultan ser un híbrido de interno y local. La sintaxis es de este tipo:

```
<a href="archivo.html#seccion">contenido</a>
```

Como para los enlaces internos, en este caso hemos de marcar la sección con otro enlace del tipo:

```
<a name="seccion"></a>
```

Enlaces externos, de correo y hacia archivos.

Para acabar con los enlaces vamos a ver los últimos 3 tipos de enlaces que habíamos señalado.

Enlaces remotos

Son los enlaces que se dirigen hacia páginas que se encuentran fuera de nuestro sitio web, es decir, cualquier otro documento que no forma parte de nuestro sitio.

Este tipo de enlaces es muy común y no representa ninguna dificultad. Simplemente colocamos en el atributo HREF de nuestra etiqueta <A> la URL o dirección de la página con la que queremos enlazar. Será algo parecido a esto.

```
<a href="http://www.guiarte.com">ir a guiarte.com</a>
```

Sólo cabe destacar que todas las direcciones web (URLs) empiezan por **http://**. Esto indica que el protocolo por el que se accede es HTTP, el utilizado en la web. No debemos olvidarnos de colocarlas, porque si no los enlaces serán tratados como enlaces locales a nuestro sitio.

Otra cosa interesante es que no tenemos que enlazar con una página web con el protocolo HTTP necesariamente. También podemos acceder a recursos a través de otros protocolos como el FTP. En tal caso, las direcciones de los recursos no comenzarán por http:// sino por ftp://.

Enlaces a direcciones de correo

Los enlaces a direcciones de correo son aquellos que al pincharlos nos abre un nuevo mensaje de correo electrónico dirigido a una dirección de mail determinada.

Estos enlaces son muy habituales en las páginas web y resultan la manera más rápida de ofrecer al visitante una vía para el contacto con el propietario de la página.

Para colocar un enlace dirigido hacia una dirección de correo colocamos **mailto:** en el atributo href del enlace, seguido de la dirección de correo a la que se debe dirigir el enlace.

```
<a href="mailto:pamela@hotmail.com">pamela@hotmail.com</a>
```

Cuando coloques enlaces a direcciones de correo procura indicar en el contenido del enlace (lo que hay entre <A> y) la dirección de correo a la que se debe escribir. Esto es porque si un usuario no tiene configurado un programa de correo en su ordenador no podrá enviar mensajes, pero por lo menos podrá copiar la dirección de mail y escribir el correo a través de otro ordenador o un sistema web-mail.

Además de la dirección de correo del destinatario, también podemos colocar en el enlace el asunto del mensaje. Esto se consigue colocando después de la dirección de correo un interrogante, la palabra subject, un signo igual (=) y el asunto en concreto.

```
<a href="mailto:pamela@hotmail.com?subject=contacto a través de la pagina">pamela@hotmail.com</a>
```

Podemos colocar otros atributos del mensaje con una sintaxis parecida. En este caso indicamos también que el correo debe ir con copia a samanta@hotmail.com.

```
<a href="mailto:pamela@hotmail.com?subject=contacto a través de la pagina&cc=samanta@hotmail.com">pamela@hotmail.com</a>
```

El visitante de la página necesitará tener configurada una cuenta de correo electrónico en su sistema para enviar los mensajes. Lógicamente, si no tiene servicio de correo en el ordenador no se podrán enviar los mensajes y este sistema de contacto con el visitante no funcionará.

Enlaces con archivos

Este no es un tipo de enlace propiamente dicho, pero lo señalamos aquí porque son un tipo de enlaces muy habitual y que presenta alguna complicación para el usuario novato.

El mecanismo es el mismo que hemos conocido en los enlaces locales y los enlaces remotos, con la única particularidad de que en vez de estar dirigidos hacia una página web está dirigido hacia un archivo de otro tipo.

Si queremos enlazar con un archivo `mi_fichero.zip` que se encuentra en el mismo directorio que la página se escribiría un enlace así.

```
<a href="mi_fichero.zip">Descarga mi_fichero.zip</a>
```

Si pinchamos un enlace de este tipo nuestro navegador descargará el fichero, haciendo la pregunta típica de "Qué queremos hacer con el archivo. Abrirlo o guardarlo en disco".

No colocar en Internet archivos ejecutables directamente sino archivos comprimidos. Por dos razones:

1. El archivo ocupará menos, con lo que será más rápida su transferencia.
2. Al preguntar al usuario lo que desea hacer con el fichero le ofrece la opción de abrirlo y guardarlo en disco. Nosotros generalmente desearemos que el usuario lo guarde en disco y no lo ejecute hasta que lo tenga en su disco duro. Si se decide a abrirlo en vez de guardarlo simplemente lo pondrá en marcha y cuando lo pare no se quedará guardado en su sistema. Si los archivos están comprimidos obligaremos al usuario a descomprimirlos en su disco duro antes de ponerlos en marcha, con lo que nos aseguramos que el usuario lo guarde en su ordenador antes de ejecutarlo.

Si queremos enlazar hacia otro tipo de archivo como un PDF o un mundo VRML (Realidad virtual para Internet) lo seguimos haciendo de la misma manera. El navegador, si reconoce el tipo de archivo, es el responsable de abrirlo utilizando el conector adecuado para ello. Así, si por ejemplo enlazamos con un PDF pondrá el programa Acrobat Reader en funcionamiento para mostrar los contenidos. Si enlazamos con un mundo VRML pondrá en marcha el plug-in que el usuario tenga instalado para ver los mundos virtuales (Cosmo Player por ejemplo).

Este sería un ejemplo de enlace a un documento PDF.

```
<a href="mi_documento_pdf">Descarga el PDF</a>
```

Imágenes en HTML

En este capítulo no explicaremos como crear ni tratar las imágenes, únicamente diremos que para ello se utilizan aplicaciones como [Paint Shop Pro](#), [Photoshop](#) o [Corel Draw](#). Tampoco explicaremos las particularidades de cada tipo de archivo GIF o JPG y la forma de optimizar nuestras imágenes.

Las imágenes son almacenadas en forma de archivos, principalmente GIF (para dibujos) o JPG (para fotos). Estos archivos pueden ser creados por nosotros mismos o pueden ser [descargados gratuitamente en sitios web especializados](#). En desarrolloweb contamos con la mayor [base de datos de gifs animados e imágenes](#) de todo tipo en castellano, que nos provee el sitio internacional [GOgraph](#).

Así pues, en estos primeros capítulos nos limitaremos a explicar como insertar y alinear debidamente en nuestra página una imagen ya creada.

La etiqueta que utilizaremos para insertar una imagen es `` (image). Esta etiqueta no posee su cierre correspondiente y en ella hemos de especificar obligatoriamente el paradero de nuestro archivo grafico mediante el atributo `src` (source).

La sintaxis queda entonces de la siguiente forma:

```

```

Para expresar el camino, lo haremos de la misma [forma que vimos para los enlaces](#). Las reglas siguen siendo las mismas, lo único que cambia es que, en lugar de una página destino, el destino es un archivo grafico.

Aparte de este atributo, indispensable obviamente para la visualización de la imagen, la etiqueta `` nos propone otra serie de atributos de mayor o menor

utilidad:

Atributo alt

Dentro de las comillas de este atributo colocaremos una brevísima descripción de la imagen. Esta etiqueta no es indispensable pero presenta varias utilidades.

Primeramente, durante el proceso de carga de la página, cuando la imagen no ha sido todavía cargada, el navegador mostrara esta descripción, con lo que el navegante se puede hacer una idea de lo que va en ese lugar.

Esto no es tan trivial si tenemos en cuenta que algunos usuarios navegan por la red con una opción del navegador que desactiva el muestreo de imágenes, con lo que tales personas podrán siempre saber de qué se trata el grafico y eventualmente cambiar a modo con imágenes para visualizarla.

Además, determinadas aplicaciones para discapacitados o teléfonos vocales que no muestran imágenes ofrecen la posibilidad de leerlas por lo que nunca esta de más pensar en estos colectivos.

En general podemos considerar como aconsejable el uso de este atributo salvo para imágenes de poca importancia y absolutamente indispensable si la imagen en cuestión sirve de enlace.

Atributos height y width

Definen la altura y anchura respectivamente de la imagen en pixels.

Todos los archivos gráficos poseen unas dimensiones de ancho y alto. Estas dimensiones pueden obtenerse a partir del propio diseñador grafico o bien haciendo clic con el botón derecho sobre la imagen vista por el navegador para luego elegir propiedades sobre el menú que se despliega.

El hecho de explicitar en nuestro código las dimensiones de nuestras imágenes ayuda al navegador a confeccionar la página de la forma que nosotros deseamos antes incluso de que las imágenes hayan sido descargadas.

Así, si las dimensiones de las imágenes han sido proporcionadas, durante el proceso de carga, el navegador reservara el espacio correspondiente a cada imagen creando una maquetación correcta. El usuario podrá comenzar a leer tranquilamente el texto sin que este se mueva de un lado a otro cada vez que una imagen se cargue.

Además de esta utilidad, el alterar los valores de estos dos atributos, es una forma inmediata de redimensionar nuestra imagen. Este tipo de utilidad no es aconsejable dado que, si lo que pretendemos es aumentar el tamaño, la perdida de calidad de la imagen será muy sensible. Inversamente, si deseamos disminuir su tamaño, estaremos usando un archivo más grande de lo necesario para la imagen que estamos mostrando con lo que aumentamos el tiempo de descarga de nuestro documento innecesariamente.

Es importante hacer hincapié en este punto ya que muchos debutantes tienen esa mala costumbre de crear gráficos pequeños redimensionando la imagen por medio de estos atributos a partir de archivos de tamaño descomunal. Hay que pensar que

el tamaño de una imagen con unas dimensiones de la mitad no se reduce a la mitad, sino que resulta ser aproximadamente 4 veces inferior.

Atributo border

Definen el tamaño en pixels del cuadro que rodea la imagen.

De esta forma podemos recuadrar nuestra imagen si lo deseamos. Es particularmente útil cuando deseamos eliminar el borde que aparece cuando la imagen sirve de enlace. En dicho caso tendremos que especificar `border="0"`.

Atributos vspace y hspace

Sirven para indicar el espacio libre, en píxeles, que tiene que colocarse entre la imagen y los otros elementos que la rodean, como texto, otras imágenes, etc.

Atributo lowsrc

Con este atributo podemos indicar un archivo de la imagen de baja resolución. Cuando el navegador detecta que la imagen tiene este atributo primero descarga y muestra la imagen de baja resolución (que ocupa muy poco y que se transfiere muy rápido). Posteriormente descarga y muestra la imagen de resolución adecuada (señalada con el atributo `src`, que se supone que ocupará más y será más lenta de transferir).

Este atributo está en desuso, aunque supone una ventaja considerable para que la descarga inicial de la web se realice más rápido y que un visitante pueda ver una muestra de la imagen mientras se descarga la imagen real.

Utilizar imágenes como enlaces

Ni que decir tiene que una imagen, lo mismo que un texto, puede servir de enlace. Vista la estructura de los enlaces podemos muy fácilmente adivinar el tipo de código necesario:

```
<a href="archivo.html"></a>
```

Ejemplo práctico

Resultará obvio para los lectores hacer ahora una página que contenga una imagen varias veces repetida pero con distintos atributos.

- Una de las veces que salga debe mostrarse con su tamaño original y con un borde de 3 píxeles.
- En otra ocasión la imagen aparecerá sin borde, con su misma altura y con una anchura superior a la original
- También mostraremos la imagen sin borde, con su misma anchura y con una altura superior a la original
- Por último, mostraremos la imagen con una altura y anchura mayores que las originales, pero proporcionalmente igual que antes.

Vamos a utilizar esta imagen para hacer el ejercicio:



Las dimensiones originales de la imagen son 28x21, así que este sería el código fuente:

```

<br>
<br>

<br>
<br>

<br>
<br>

```

Alineación de imágenes con HTML

Vimos en su momento el atributo align que nos permitía alinear el texto a derecha, izquierda o centro de nuestra página. Dijimos que este atributo no era exclusivo de la etiqueta <p> sino que podía ser encontrado en otro tipo de etiquetas.

Pues bien, resulta ser una de esas etiquetas que aceptan este atributo aunque en este caso el funcionamiento resulta ser diferente.

Para alinear una imagen horizontalmente podemos hacerlo de la misma forma que el texto, es decir, utilizando el atributo align dentro de una etiqueta <p> o <div>. En este caso, lo que incluiremos dentro de esa etiqueta será la imagen en lugar del texto:

Este código mostrará la imagen en el centro:

```
<div align="center"></div>
```

Quedaría así:



Sin embargo, ya hemos dicho que la etiqueta puede aceptar el atributo align. En este caso, la utilidad que le damos difiere de la anterior.

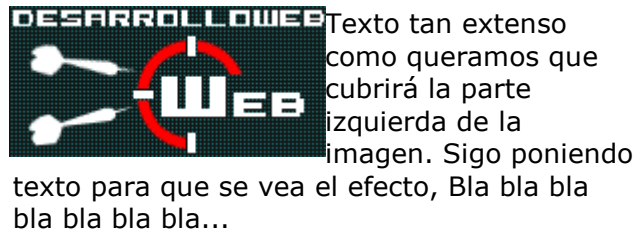
El hecho de utilizar el atributo align dentro de la etiqueta nos permite, en el caso de darle los valores left o right, justificar la imagen del lado que deseamos a la vez que rellenamos con texto el lado opuesto. De esta forma embebemos nuestras imágenes dentro del texto de una manera sencilla.

Aquí podéis ver el tipo de código a crear para obtener dicho efecto:

```
<p>
Texto tan extenso como queramos que
cubrirá la parte izquierda de la imagen. Sigo poniendo texto para que se vea el
efecto, Bla bla bla bla bla bla bla...
```

</p>

Quedaría así:

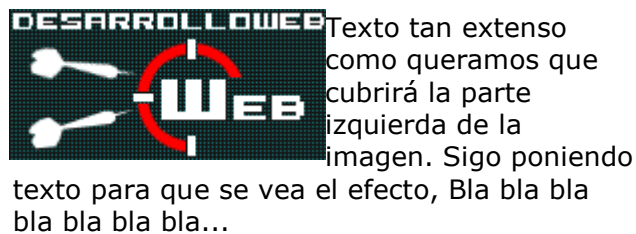


<p>

Texto tan extenso como queramos que cubrirá la parte derecha de la imagen. Sigo poniendo texto para que se vea el efecto, Bla bla bla bla bla bla bla...

</p>

Quedaría así:



Si en algún momento deseásemos dejar de rellenar ese espacio lateral, podemos pasar a una zona libre introduciendo un salto de línea
 dentro del cual añadiremos un atributo: clear

Así, etiquetas del tipo:

<br clear="left">

Saltará verticalmente hasta encontrar el lateral izquierdo libre.

<br clear="right">

Saltará verticalmente hasta encontrar el lateral derecho libre.

<br clear="all">

Saltará verticalmente hasta encontrar ambos laterales libres.

Ejemplo de clear:



Existen otro tipo de valores que puede adoptar el atributo align dentro de la etiqueta . Estos son relativos a la alineación vertical de la imagen. Supongamos que escribimos una línea al lado de nuestra imagen. Esta línea puede quedar por ejemplo arriba, abajo o al medio de la imagen. Asimismo, puede que en una misma línea tengamos varias imágenes de alturas diferentes que pueden ser alineadas de distintas formas.

Estos valores adicionales del atributo align son:

top

Ajusta la imagen a la parte más alta de la línea. Esto quiere decir que, si hay una imagen más alta, ambas imágenes presentaran el borde superior a la misma altura.

bottom

Ajusta el bajo de la imagen al texto.

Absbottom

Colocara el borde inferior de la imagen a nivel del elemento más bajo de la línea.

middle

Hace coincidir la base de la línea de texto con el medio vertical de la imagen.

absmiddle

Ajusta la imagen al medio absoluto de la línea.

Estas explicaciones, que pueden resultar un poco complicadas, pueden ser más fácilmente asimiladas a partir con un poco de practica.

Tablas en HTML

Una tabla en un conjunto de celdas organizadas dentro de las cuales podemos alojar distintos contenidos.

Para empezar, nada más sencillo que por el principio: las tablas son definidas por las etiquetas `<table>` y `</table>`.

Dentro de estas dos etiquetas colocaremos todas las otras etiquetas, textos e imágenes que darán forma y contenido a la tabla.

Las tablas son descritas por líneas de izquierda a derecha. Cada una de estas líneas es definida por otra etiqueta y su cierre: `<tr>` y `</tr>`

Asimismo, dentro de cada línea, habrá diferentes celdas. Cada una de estas celdas será definida por otro par de etiquetas: `<td>` y `</td>`. Dentro de estas etiquetas será donde coloquemos nuestro contenido.

Aquí tenéis un ejemplo de estructura de tabla:

```
<table>
<tr>
<td>Celda 1, linea 1</td>
<td> Celda 2, linea 1</td>
</tr>
<tr>
<td> Celda 1, linea 2</td>
<td> Celda 2, linea 2</td>
</tr>
</table>
```

El resultado:
Celda 1, linea 1 Celda 2, linea 1
Celda 1, linea 2 Celda 2, linea 2

Tablas en HTML. Atributos para filas y celdas.

Hemos visto en el capítulo anterior que las tablas están compuestas de líneas que, a su vez, contienen celdas. Las celdas son delimitadas por las etiquetas `<td>` o por las etiquetas `<th>` (si queremos texto en negrita y centrado) y constituyen un entorno independiente del resto del documento. Esto quiere decir que:

- Podemos usar prácticamente cualquier tipo de etiqueta dentro de la etiqueta `<td>` para, de esta forma, dar forma a su contenido.
- Las etiquetas situadas en el interior de la celda no modifican el resto del documento.
- Las etiquetas de fuera de la celda no son tenidas en cuenta por ésta.

Así pues, podemos especificar el formato de nuestras celdas a partir de etiquetas introducidas en su interior o mediante atributos colocados dentro de la etiqueta de celda `<td>` o bien, en algunos casos, dentro de la etiqueta `<tr>`, si deseamos que el atributo sea válido para toda la línea.

Veamos a continuación algunos atributos útiles para la construcción de nuestras tablas. Empecemos viendo atributos que nos permiten modificar una celda en concreto o toda una línea:

align	Justifica el texto de la celda del mismo modo que si fuese el de un párrafo.
valign	Podemos elegir si queremos que el texto aparezca arriba (top), en el centro (middle) o abajo (bottom) de la celda.
bgcolor	Da color a la celda o línea elegida.
bordercolor	Define el color del borde.

Otros atributos que pueden ser únicamente asignados a una celda y no al conjunto de celdas de una línea son:

background	Nos permite colocar un fondo para la celda a partir de un enlace a una imagen.
height	Define la altura de la celda en pixels o porcentaje.
width	Define la anchura de la celda en pixels o porcentaje.
colspan	Expande una celda horizontalmente.
rowspan	Expande una celda verticalmente.

Probemos todos estos comandos:

```
<td width="80">
```

Dará una anchura de 80 pixels a la celda. Sin embargo,

```
<td width="80%">
```

Dará una anchura a la celda del 80% de la anchura de la tabla.

Los atributos **rowspan** y **colspan** son también utilizados frecuentemente. Gracias a ellos es posible expandir celdas fusionando éstas con sus vecinas. El valor que pueden tomar estas etiquetas es numérico. El número representa la cantidad de

celdas fusionadas.

Así,

```
<td colspan="2">
```

Fusionara la celda en cuestión con su vecina derecha.

Esta celda tiene un colspan="2"	
Celda normal	Otra celda

Del mismo modo,

```
<td rowspan="2">
```

Esta celda tiene rowspan="2", por eso tiene fusionada la celda de abajo.	Celda Normal
	Otra celda normal

Expandirá la celda hacia abajo fusionándose con la celda inferior.

El resto de los atributos presentados presentan una utilidad y uso bastante obvios.

Además de los atributos específicos de cada celda o línea, las tablas pueden ser adicionalmente formateadas a partir de los atributos que nos ofrece la propia etiqueta `<table>`. He aquí aquellos que pueden parecernos en un principio importantes:

- align** Alinea horizontalmente la tabla con respecto a su entorno.
- background** Nos permite colocar un fondo para la tabla a partir de un enlace a una imagen.
- bgcolor** Da color de fondo a la tabla.
- border** Define el número de pixels del borde principal.
- bordercolor** Define el color del borde.
- cellpadding** Define, en pixels, el espacio entre los bordes de la celda y el contenido de la misma.
- cellspacing** Define el espacio entre los bordes (en pixels).
- height** Define la altura de la tabla en pixels o porcentaje.
- width** Define la anchura de la tabla en pixels o porcentaje.

Podéis comprobar vosotros mismos que los atributos definidos para una celda tienen prioridad con respecto a los definidos para una tabla. Podemos definir, por ejemplo, una tabla con color de fondo rojo y una de las celdas de color de fondo verde y se verá toda la tabla de color rojo menos la celda verde. Del mismo modo, podemos definir un color azul para los bordes de la tabla y hacer que una celda particular sea mostrada con un borde rojo. (Aunque esto no funcionará en todos los navegadores debido a que algunos no reconocen el atributo `bordercolor`).

Tabla de color	El atributo bgcolor de la
----------------	---------------------------

rojo de fondo	tabla está en rojo.
Celda normal	Esta celda está en verde. tiene el atributo bgcolor en color verde

Tablas anidadas

Muy útil también es el uso de tablas anidadas. De la misma forma que podíamos incluir listas dentro de otras listas, las tablas pueden ser incluidas dentro de otras. Así, podemos incluir una tabla dentro de la celda de otra. El modo de funcionamiento sigue siendo el mismo aunque la situación puede complicarse si el número de tablas embebidas dentro de otras es elevado.

Vamos a ver un código de anidación de tablas. Veamos primero el resultado y luego el código, así conseguiremos entenderlo mejor.

Celda de la tabla principal	<table> <tr> <td>Tabla anidada, celda 1</td><td>Tabla anidada, celda 2</td></tr> <tr> <td>Tabla anidada, celda 3</td><td>Tabla anidada, celda 4</td></tr> </table>	Tabla anidada, celda 1	Tabla anidada, celda 2	Tabla anidada, celda 3	Tabla anidada, celda 4
Tabla anidada, celda 1	Tabla anidada, celda 2				
Tabla anidada, celda 3	Tabla anidada, celda 4				

Este sería el código:

```
<table cellspacing="10" cellpadding="10" border="3">
<tr>
  <td align="center">
    Celda de la tabla principal
  </td>
  <td align="center">
    <table cellspacing="2" cellpadding="2" border="1">
      <tr>
        <td>Tabla anidada, celda 1</td>
        <td>Tabla anidada, celda 2</td>
      </tr>
      <tr>
        <td>Tabla anidada, celda 3</td>
        <td>Tabla anidada, celda 4</td>
      </tr>
    </table>
  </td>
</tr>
</table>
```

Realizar ejercicios de tablas incluidos en el Glosario de ejercicios.

Formularios HTML

Hasta ahora hemos visto la forma en la que el HTML gestiona y muestra la información, esencialmente mediante texto, imágenes y enlaces. Nos queda por ver de qué forma podemos intercambiar información con nuestro visitante. Desde luego, este nuevo aspecto resulta primordial para gran cantidad de acciones que se pueden llevar a cabo mediante el Web: Comprar un artículo, rellenar una encuesta, enviar un comentario al autor...

Puede ocurrir que lo que deseamos es que el navegante nos envíe una información bien precisa. Es por ello que el HTML propone otra solución mucho más amplia: Los formularios.

Los formularios son esas famosas cajas de texto y botones que podemos encontrar en muchas páginas web. Son muy utilizados para realizar búsquedas o bien para introducir datos personales por ejemplo en sitios de comercio electrónico. Los datos que el usuario introduce en estos campos son enviados al correo electrónico del administrador del formulario o bien a un programa que se encarga de procesarlo automáticamente.

Los formularios son definidos por medio de las etiquetas **<form>** y **</form>**. Entre estas dos etiquetas colocaremos todos los campos y botones que componen el formulario. Dentro de esta etiqueta **<form>** debemos especificar algunos atributos:

action

Define el tipo de acción a llevar a cabo con el formulario. Como ya hemos dicho, existen dos posibilidades:

- El formulario es enviado a una dirección de correo electrónico
- El formulario es enviado a un programa o script que procesa su contenido

En el primer caso, el contenido del formulario es enviado a la dirección de correo electrónico especificada por medio de una sintaxis de este tipo:

```
<form action="mailto:direccion@correo.com" ...>
```

Si lo que queremos es que el formulario sea procesado por un programa, hemos de especificar la dirección del archivo que contiene dicho programa. La etiqueta quedaría en este caso de la siguiente forma:

```
<form action="dirección del archivo" ...>
```

La forma en la que se expresa la localización del archivo que contiene el programa es la misma que la vista para los enlaces.

method

Este atributo se encarga de especificar la forma en la que el formulario es enviado. Los dos valores posibles que puede tomar esta atributo son post y get. A efectos prácticos y, salvo que se os diga lo contrario, daremos siempre el valor post.

enctype

Se utiliza para indicar la forma en la que viajará la información que se mande por el formulario. En el caso más corriente, enviar el formulario por correo electrónico, el valor de este atributo debe de ser "text/plain". Así conseguimos que se envíe el contenido del formulario como texto plano dentro del email.

Si queremos que el formulario se procese automáticamente por un programa, generalmente no utilizaremos este atributo, de modo que tome su valor por defecto, es decir, no incluiremos enctype dentro de la etiqueta <form>

Ejemplo de etiqueta <form> completa

Así, para el caso más habitual -el envío del formulario por correo- la etiqueta de creación del formulario tendrá el siguiente aspecto:

```
<form action="mailto:direccion@correo.com (o nombre del archivo de proceso)"
method="post" enctype="text/plain">
```

Entre esta etiqueta y su cierre colocaremos el resto de etiquetas que darán forma a nuestro formulario, las cuales serán vistas en capítulos siguientes.

Elementos de Formularios. Campos de texto

El HTML nos propone una gran diversidad de alternativas a la hora de crear nuestros formularios. Estas van desde la clásica caja de texto hasta la lista de opciones pasando por las cajas de validación.

Veamos en qué consiste cada una de estas modalidades y como podemos implementarlas en nuestro formulario.

Texto corto

Las cajas de texto son colocadas por medio de la etiqueta <input>. Dentro de esta etiqueta hemos de especificar el valor de dos atributos: **type** y **name**.

La etiqueta es de la siguiente forma:

```
<input type="text" name="nombre">
```

El nombre del elemento del formulario es de gran importancia para poder identificarlo en nuestro programa de procesamiento o en el mail recibido. Por otra parte, es importante indicar el atributo type, ya que, como veremos, existen otras modalidades de formulario que usan esta misma etiqueta.

El empleo de estas cajas está fundamentalmente destinado a la toma de datos breves: palabras o conjuntos de palabras de longitud relativamente corta. Veremos más adelante que existe otra forma de tomar textos más largos a partir de otra etiqueta.

Además de estos dos atributos, esenciales para el correcto funcionamiento de nuestra etiqueta, existen otra serie de atributos que pueden resultarnos de utilidad pero que no son imprescindibles:

size

Define el tamaño de la caja en número de caracteres. Si al escribir el usuario llega al final de la caja, el texto ira desfilando a medida que se escribe haciendo desaparecer la parte de texto que queda a la izquierda.

maxlength

Indica el tamaño máximo del texto que puede ser tomado por el formulario. Es importante no confundirlo con el atributo size. Mientras el primero define el tamaño aparente de la caja de texto, maxlength indica el tamaño máximo real del texto que se puede escribir. Podemos tener una caja de texto con un tamaño aparente (size) que es menor que el tamaño máximo (maxlength). Lo que ocurrirá en este caso es que, al escribir, el texto ira desfilando dentro de la caja hasta que lleguemos a su tamaño máximo definido por maxlength, momento en el cual nos será imposible continuar escribiendo.

value

En algunos casos puede resultarnos interesante asignar un valor definido al campo en cuestión. Esto puede ayudar al usuario a rellenar más rápidamente el formulario o darle alguna idea sobre la naturaleza de datos que se requieren. Este valor inicial del campo puede ser expresado mediante el atributo value. Veamos su efecto con un ejemplo sencillo:

```
<input type="text" name="nombre" value="Perico Palotes">
```

Genera un campo de este tipo:

Veremos posteriormente que este atributo puede resultar relevante en determinadas situaciones.

Texto oculto

Podemos esconder el texto escrito por medio asteriscos de manera a aportar una cierta confidencialidad. Este tipo de campos son análogos a los de texto con una sola diferencia: remplazamos el atributo type="text" por type="password":

```
<input type="password" name="nombre">
```

En este caso, podéis comprobar que al escribir dentro del campo en lugar de texto veréis asteriscos.

Estos campos son ideales para la introducción de datos confidenciales, principalmente códigos de acceso. Se ve en funcionamiento a continuación.

Texto largo

Si deseamos poner a la disposición de usuario un campo de texto donde pueda escribir cómodamente sobre un espacio compuesto de varias líneas, hemos de invocar una nueva etiqueta: `<textarea>` y su cierre correspondiente.

Este tipo de campos son prácticos cuando el contenido a enviar no es un nombre teléfono o cualquier otro dato breve, sino más bien, un comentario, opinión, etc.

Dentro de la etiqueta `textarea` deberemos indicar, como para el caso visto anteriormente, el atributo `name` para asociar el contenido a un nombre que será asemejado a una variable en los programas de proceso. Además, podemos definir las dimensiones del campo a partir de los atributos siguientes:

rows Define el número de líneas del campo de texto.

cols Define el número de columnas del campo de texto.

La etiqueta queda por tanto de esta forma:

```
<textarea name="comentario" rows="10" cols="40"></textarea>
```

El resultado es el siguiente:

Asimismo, es posible predefinir el contenido del campo. Para ello, no usaremos el atributo `value` sino que escribiremos dentro de la etiqueta el contenido que deseamos atribuirle. Veámoslo:

```
<textarea name="comentario" rows="10" cols="40">Escribe tu comentario....</textarea>
```

Dará como resultado:

Otros elementos de formulario

Efectivamente, los textos son una manera muy práctica de hacernos llegar la información del navegante. No obstante, en muchos casos, los textos son difícilmente adaptables a programas que puedan procesarlos debidamente o bien, puede que su contenido no se ajuste al tipo de información que requerimos. Es por ello que, en determinados casos, puede resultar más efectivo proponer una elección al navegante a partir del planteamiento de una serie de opciones.

Este es el caso de, por ejemplo, ofrecer una lista de países, el tipo de tarjeta de crédito para un pago,...

Este tipo de opciones pueden ser expresadas de diferentes formas. Veamos a continuación cuáles son:

Listas de opciones

Las listas de opciones son ese tipo de menús desplegables que nos permiten elegir una (o varias) de las múltiples opciones que nos proponen. Para construirlas emplearemos una etiqueta con su respectivo cierre: `<select>`

Como para los casos ya vistos, dentro de esta etiqueta definiremos su nombre por medio del atributo `name`. Cada opción será incluida en una línea precedida de la etiqueta `<option>`.

Podemos ver, a partir de estas directivas, la forma más típica y sencilla de esta etiqueta:

```
<select name="estacion">
<option>Primavera</option>
<option>Verano</option>
<option>Otoño</option>
<option>Invierno</option>
</select>
```

El resultado es:



Esta estructura puede verse modificada principalmente a partir de otros dos atributos:

size

Indica el número de valores mostrados de la lista. El resto pueden ser vistos por medio de la barra lateral de desplazamiento.

multiple

Permite la selección de más varios elementos de la lista. La elección de más de un elemento se hace como con el explorador de Windows, a partir de las teclas `ctrl` o `shift`. Este atributo se expresa sin valor alguno, es decir, no se utiliza con el igual: simplemente se pone para conseguir el efecto, o no se pone si queremos una lista desplegable común.

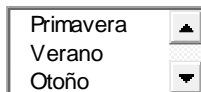
Veamos cuál es el efecto producido por estos dos atributos cambiando la línea:

```
<select name="estacion">
```

por:

```
<select name="estacion" size="3" multiple>
```

La lista quedara de esta forma:



La etiqueta **<option>** puede asimismo ser matizada por medio de **otros atributos**

selected

Del mismo modo que multiple, este atributo no toma ningún valor sino que simplemente indica que la opción que lo presenta esta elegida por defecto.

Así, si cambiamos la línea del código anterior:

```
<option>Otoño</option>
```

por:

```
<option selected>Otoño</option>
```

El resultado será:



value

Define el valor de la opción que será enviado al programa o correo electrónico si el usuario elige esa opción. Este atributo puede resultar muy útil si el formulario es enviado a un programa puesto que a cada opción se le puede asociar un número o letra, lo cual es más fácilmente manipulable que una palabra o texto. podríamos así escribir líneas del tipo:

```
<option value="1">Primavera</option>
```

De este modo, si el usuario elige primavera, lo que le llegara al programa (o correo) es una variable llamada estación que tendrá **com** valor 1. En el correo electrónico recibiríamos:

```
estacion=1
```

Botones de radio

Existe otra alternativa para plantear una elección, en este caso, obligamos al internauta a elegir únicamente una de las opciones que se le proponen.

La etiqueta empleada en este caso es **<input>** en la cual tendremos el atributo type ha de tomar el valor radio. Veamos un ejemplo:

```
<input type="radio" name="estacion" value="1">Primavera  
<br>  
<input type="radio" name="estacion" value="2">Verano  
<br>  
<input type="radio" name="estacion" value="3">Otoño
```

```
<br>
<input type="radio" name="estacion" value="4">Invierno
```

El resultado es el siguiente:

- ☐ Primavera
- ☐ Verano
- ☐ Otoño
- ☐ Invierno

Como puede verse, a cada una de las opciones se le atribuye una etiqueta input dentro de la cual asignamos el mismo nombre (name) para todas las opciones y un valor (value) distinto. Si el usuario elige supuestamente Otoño, recibiremos en nuestro correo una línea tal que esta:

estacion=3

Cabe señalar que es posible preseleccionar por defecto una de las opciones. Esto puede ser conseguido por medio del atributo **checked**:

```
<input type="radio" name="estacion" value="2" checked>Verano
```

Veamos el efecto:

- ☐ Primavera
- ☒ Verano
- ☐ Otoño
- ☐ Invierno

Cajas de validación

Este tipo de elementos pueden ser activados o desactivados por el visitante por un simple clic sobre la caja en cuestión. La sintaxis utilizada es muy similar a las vistas anteriormente:

```
<input type="checkbox" name="paella">Me gusta la paella
```

El efecto:

☐ Me gusta la paella

La única diferencia fundamental es el valor adoptado por el atributo type.

Del mismo modo que para los botones de radio, podemos activar la caja por medio del atributo **checked**.

El tipo de información que llegara a nuestro correo (o al programa) será del tipo:

paella=on (u off dependiendo si ha sido activada o no)

Envío, borrado y demás en formularios HTML

Los formularios han de dar plaza no solamente a la información a tomar del usuario sino también a otra serie de funciones. Concretamente, han de permitirnos su envío mediante un botón. También puede resultar práctico poder proponer un botón de borrado o bien acompañarlo de datos ocultos que puedan ayudarnos en su procesamiento.

En este capítulo, para terminar la saga de formularios, daremos a conocer los medios de instalar todas estas funciones.

botón de envío

Para dar por finalizado el proceso de relleno del formulario y hacerlo llegar a su gestor, el navegante ha de validarlo por medio de un botón previsto a tal efecto. La construcción de dicho botón no reviste ninguna dificultad una vez familiarizados con las etiquetas input ya vistas:

```
<input type="submit" value="Enviar">
```

Con este código generamos un botón como este:



Como puede verse, tan solo hemos de especificar que se trata de un botón de envío (type="submit") y hemos de definir el mensaje del botón por medio del atributo value.

botón de borrado

Este botón nos permitirá borrar el formulario por completo en el caso de que el usuario desee rehacerlo desde el principio. Su estructura sintáctica es análoga a la anterior:

```
<input type="reset" value="Borrar">
```

A diferencia del botón de envío, indispensable en cualquier formulario, el botón de borrado resulta meramente optativo y no es utilizado frecuentemente. Hay que tener cuidado de no ponerlo muy cerca del botón de envío y de distinguir claramente el uno del otro.

Datos ocultos

En algunos casos, aparte de los propios datos enviados por el usuario, puede resultar práctico enviar datos definidos por nosotros mismos que ayuden al programa en su procesamiento del formulario. Este tipo de datos, que no se muestran en la página pero si pueden ser detectados solicitando el código fuente, no son frecuentemente utilizados por páginas construidas en HTML, son más bien usados por páginas que emplean tecnologías de servidor. No os asustéis, veremos más adelante qué quiere decir esto. Tan solo queremos dar constancia de su existencia y de su modo creación. He aquí un ejemplo:

```
<input type="hidden" name="sitio" value="www.desarrolloweb.com">
```


Esta etiqueta, incluida dentro de nuestro formulario, enviara un dato adicional al correo o programa encargado de la gestión del formulario. podríamos, a partir de este dato, dar a conocer al programa el origen del formulario o algún tipo de acción a llevar a cabo (una redirección por ejemplo).

Botones normales

Dentro de los formularios también podemos colocar botones normales, pulsables como cualquier otro botón. Igual que ocurre con los campos hidden, estos botones por si solos no tienen mucha utilidad pero podremos necesitarlos para realizar acciones en el futuro. Su sintaxis es la siguiente.

```
<input type=button value="Texto escrito en el botón">
```

El uso más frecuente de un botón es en la programación en el cliente. Utilizando lenguajes como [Javascript](#) podemos definir acciones a tomar cuando un visitante pulse el botón de una página web.

Ejemplo de formulario

Vemos el formulario en esta página. Vosotros tratar de construirlo para ver si habéis entendido bien los temas sobre formularios.

Nombre

Email

Población

Sexo

☒ Hombre

☐ Mujer

Frecuencia de los viajes

▼

Comentarios sobre su satisfacción personal

☒ Deseo recibir notificación de las novedades en las líneas de autobuses.

Final del formulario

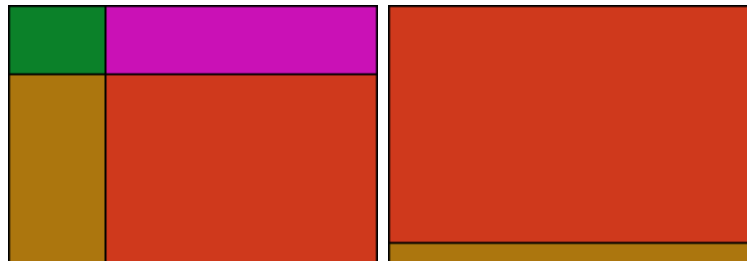
Frames en HTML

Los frames -que significan en castellano marcos- son una manera de partir la página en distintos espacios independientes los unos de los otros, de modo que en cada espacio se coloca una página distinta que se codifica en un fichero HTML distinto.

Los frames nos permiten partir la ventana del navegador en diferentes áreas. Cada una de estas áreas son independientes y han de ser codificadas con archivos HTML también independientes. Como resultado, cada frame o marco contiene las propiedades específicas que le indiquemos en el código HTML a presentar en ese espacio. Así mismo, y dado que cada marco es independiente, tendrán sus propias barras de desplazamiento, horizontales y verticales, por separado.

Existen en la web muchas páginas que contienen frames y seguro que todos hemos tenido la ocasión de conocer algunas. Se suelen utilizar para colocar en una parte de la ventana una barra de navegación, que generalmente se encuentra fija y permite el acceso a cualquier zona de la página web. Una de las principales ventajas de la programación con frames viene derivada de la independencia de los distintos frames, pues podemos navegar por los contenidos de nuestro web con la barra de navegación siempre visible, y sin que se tenga que recargar en cada una de las páginas que vamos visitando.

Un ejemplo de las áreas que se pueden construir en una construcción de frames se puede ver en las imágenes siguientes.



Frames - Explicación básica

Las páginas web que están hechas con frames se componen de una declaración de los marcos y tantas páginas en formato HTML corriente como distintas divisiones hemos definido. La declaración o definición de frames es la única página que realmente debemos aprender, puesto que las páginas que se van a visualizar en cada uno de los cuadros son ficheros HTML de los que venimos aprendiendo anteriormente en este manual.

Dicha definición está compuesta por etiquetas `<FRAMESET>` y `<FRAME>`, con las que se indicamos la disposición de todos los cuadros. La etiqueta `<FRAMESET>` indica las particiones de la ventana del navegador y la etiqueta `<FRAME>` indica cada uno de los cuadros donde colocaremos una página independiente.

Las particiones que se pueden hacer con un `<FRAMESET>` son en filas o columnas. Por ejemplo, podríamos indicar que deseamos hacer una división de la página en dos filas, o dos columnas, tres filas, etc. Para indicar tanto la forma de partir la ventana -en filas o columnas- como el número de particiones que pretendemos hacer, se ha de utilizar el atributo `COLS` o `ROWS`. El primero sirve para indicar una partición en columnas y el segundo para una partición en filas.

En el atributo COLS o ROWS -sólo podemos elegir uno de los dos- colocamos entre comillas el número de particiones que deseamos realizar, indicando de paso el tamaño que va a asignarse a cada una. Un valor típico de estos atributos sería el siguiente:

cols="20%,80%"

Indica que se deben colocar dos columnas, la de la izquierda tendría un 20% del espacio total de la ventana y la de la derecha un 80%.

rows="15%,60%,25%"

Así indicamos que deseamos tres filas, la de arriba con un 15% del espacio total, la del medio con un espacio correspondiente al 60% del total y la de abajo con un 25%. En total suman el 100% del espacio de la ventana.

Además del porcentaje para indicar el espacio de cada una de las casillas, también podemos indicarlo en píxeles. De esta manera.

cols="200,600"

Para indicar que la columna de la izquierda debe tener 200 píxeles de ancho y la de la derecha 600. Esto está bien si nuestra ventana tiene 800 píxeles de ancho, pero esto no tiene porque ser así en todos los monitores de los usuarios, por lo que este modo de expresar los marcos es importante que se indique de la siguiente manera.

cols="200,*"

Así indicamos que la primera columna ha de medir 200 píxeles y que el resto del espacio disponible -que será mayor o menor dependiendo de la definición de la pantalla del usuario- se le asignará a segunda columna.

En la práctica podemos mezclar todos estos métodos para definir los marcos de la manera que deseemos, con porcentaje, con píxeles o con el comodín (*). No importa cómo se definan, la única recomendación es que uno de los valores que indiquemos sea un asterisco, para que el área correspondiente a dicho asterisco o comodín sea más o menos grande dependiendo del espacio que tenga la ventana de nuestro navegador. Otros métodos de definir filas y columnas, atendiendo a este consejo, serían los siguientes:

rows="100*,12%"

Definimos tres filas, la primera con 100 píxeles de ancho, la segunda con el espacio que sobre de las otras dos, y la tercera con un 12% del espacio total.

cols="10%,50%,120,*"

Estamos indicando cuatro columnas. La primera del 10% del espacio de la ventana, la segunda con la mitad justa de la ventana, la tercera con un espacio de 120 píxeles y la última con la cantidad de espacio que sobre al asignar espacio a las demás particiones.

Una vez hemos indicado el número de filas o columnas y el espacio reservado a cada una con la etiqueta <FRAMESET>, debemos especificar con la etiqueta <FRAME> la procedencia de cada uno de los frames en los que hemos partido la ventana.

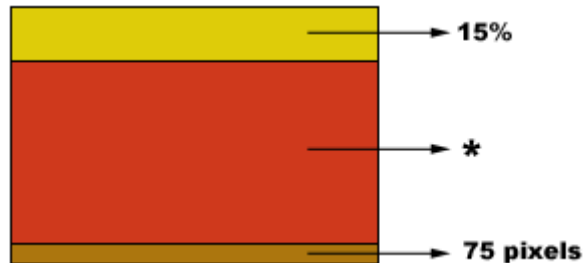
Para ello, disponemos del atributo SRC, que se ha de definir para cada una de las filas o columnas. De esta manera.

<FRAME src="marco1.html">

Así queda indicado que el frame que estamos definiendo debe mostrar la página marco1.html en su interior.

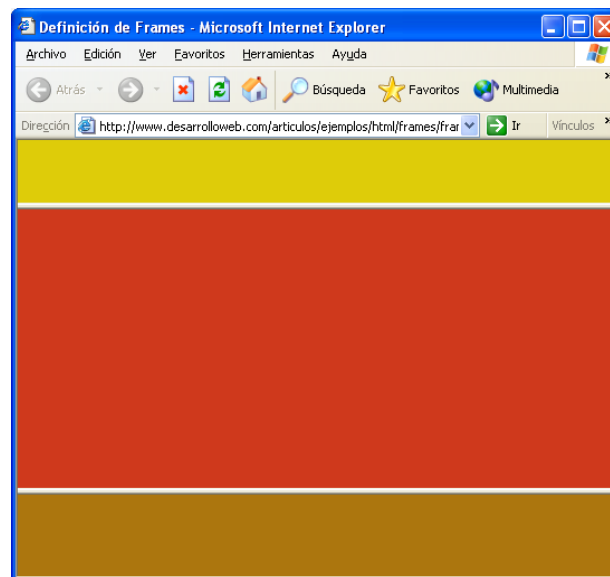
Frames - Creación de una estructura simple

Para ilustrar todo lo que venimos explicando podemos ver el ejemplo sobre cómo se crearía la definición de frames de la imagen que podemos ver a continuación.



```
<html>
<head>
  <title>Definición de Frames</title>
</head>
<frameset rows="15%,*,75">
  <frame src="pagina1.html">
  <frame src="pagina2.html">
  <frame src="pagina3.html">
</frameset>
</html>
```

El resultado sería el siguiente:



Además tenemos algunas consideraciones que hacer para terminar de comprender este ejemplo:

- El título de la definición de frames es el que hereda toda la página web, por ello, no es buena idea titular como "definición de frames" por ejemplo, ya que entonces toda nuestra página se titularía así y seguramente no sea muy descriptivo. Si estuviésemos haciendo una página para la carnicería pepe sería mejor titular a la definición de frames algo como "Carnicería Pepe, las mejores carnes en Madrid".
- La página que define los frames no tiene body. HTML puede arrojarnos un error si lo incluimos.
- Las páginas "pagina1.html", "pagina2.html"y "pagina3.html" han de escribirse en archivos independientes con el nombre indicado. En este ejemplo, dichas páginas deberían encontrarse en el mismo directorio que la declaración de frames. Si especificamos una ruta para acceder al archivo podemos colocarlo en el directorio que deseemos.
- Los colores de cada uno de los frames los hemos colocado con el atributo bgcolor colocado en la etiqueta <BODY> de cada una de las páginas que se muestran en los marcos.

Frames - Una página en cada marco

Las páginas que mostraremos en cada marco son documentos HTML iguales a los que venimos creando anteriormente. Podemos colocar cualquier elemento HTML de los estudiados en este manual, como etiquetas de párrafo, imágenes, colores de fondo, etc.

Cada documento, como ya hemos indicado, se escribe por separado en su propio archivo HTML. Para el ejemplo del capítulo anterior podemos definir los archivos HTML de la siguiente manera.

pagina1.html

Es la página que contiene el titular de la web. Simplemente se trata de una etiqueta <H1> de titular. La página tiene su propio título, con la etiqueta <TITLE>, que no se podrá visualizar por ningún sitio a no ser que se muestre esta página sin los frames, ya que las páginas dentro de los marcos heredan el título de la definición de los frames.

```
<html>
<head>
  <title>Titulo Carnicería Pepe</title>
</head>

<body bgcolor="#DECC09">
<h1 align=center>Carnicería PEPE</h1>
</body>
</html>
```

pagina2.html

Es la página que se presentará en el área principal de la definición de frames, es decir, la página que tiene más espacio para visualizarse y donde pondremos los contenidos de la web. En este caso muestra un mensaje de bienvenida a la web, que hará las veces de portada.

```

<html>
<head>
  <title>Portada de Carnicería PEPE</title>
</head>

<body bgcolor="#CF391C" text="#ffffff">
<h1 align="center">Bienvenidos a nuestra web</h1>
<br>
<br>
La carnicería PEPE, con más de 100 años de experiencia, es la mejor fuente de
carnes de vacuno y cerdo de la comunidad.
<br>
<br>
Tanto en invierno como en verano puede encontrar nuestras ofertas de temporada
de primera calidad.
</body>
</html>

```

pagina3.html

En esta página se mostrará la barra de navegación por los contenidos del sitio. Contiene enlaces que deberían actualizar el contenido del área principal de la declaración de frames, para mostrar los distintos contenidos del sitio, por ejemplo, la portada, los productos, la página de contacto, etc.

```

<html>
<head>
  <title>Barra de navegación de carnicería PEPE</title>
</head>

<body bgcolor="#AC760E" link="ffffcc" vlink="ffffcc">
<div align="center">
<b>
<a href="pagina2.html">Portada</a> |
<a href="productos.html">Productos</a> |
<a href="contacto.html">Contacto</a>
</b>
</div>
</body>
</html>

```

Frames - Dirigir los enlaces

La única particularidad destacable en el ejemplo del capítulo anterior, y en el manejo de frames en general, se trata de que **cada uno de los enlaces que colocamos en las páginas actualizan el frame donde está colocado este enlace**. Por ejemplo, si tenemos enlaces en la parte inferior de la ventana, en el espacio correspondiente al tercer marco, actualizarán los contenidos del tercer frame, que es donde están situados los enlaces.

Lo lógico es que al pulsar sobre un enlace de la barra de navegación actualicemos el frame principal, que es donde habíamos planeado colocar los contenidos, en lugar del frame donde colocamos la barra de navegación, que debería mantenerse fija. Para conseguir este efecto debemos hacer un par de cosas:

1. Darle un nombre al frame que deseamos actualizar

Dicho nombre se indica en la etiqueta <FRAME> de la definición de frames. Para ello utilizamos el atributo name, igualado al nombre que le queremos dar a dicho marco.

2. Dirigir los enlaces hacia ese frame

Para ello debemos colocar en el atributo target de los enlaces -etiqueta <A>- el nombre del frame que deseamos actualizar al pulsar el enlace.

Después de darle un nombre al frame principal, nuestra declaración de frames quedaría de la siguiente manera.

```
<frameset rows="15%,*,75">  
  <frame src="pagina1.html">  
  <frame src="pagina2.html" name="principal">  
  <frame src="pagina3.html">  
</frameset>
```

Además, deberíamos colocar el atributo target a los enlaces, tal como sigue.

```
<a href="pagina2.html" target="principal">Portada</a> |  
<a href="productos.html" target="principal">Productos</a> |  
<a href="contacto.html" target="principal">Contacto</a>
```

Valores para el atributo target

Como hemos visto, con el atributo target de la etiqueta <A> podemos indicar el nombre del frame que deseamos que actualice ese enlace. Sin embargo, no es este el único valor que podemos aplicarle al atributo. Tenemos algunos valores adicionales que podemos asignar a cualquier enlace en general.

_blank

Para hacer que ese enlace se abra en una ventana a parte. Nuestros ejemplos en este manual se suelen abrir en una ventana a parte, colocando este valor en el target de los enlaces que llevan a los ejemplos.

_self

Se actualiza el frame donde está situado el enlace. Es el valor por defecto.

_parent

El enlace se actualiza sobre su padre o sobre la ventana que estamos trabajando, si es que no hay un padre.

_top

La página se carga a pantalla completa, es decir, eliminando todos los frames que pudiera haber. Este atributo es muy importante porque si colocamos en nuestra página con frames un enlace a una página externa, se abriría en uno de los frames y se mantendrían visibles otros frames de la página, haciendo un efecto que suele ser poco agradable, porque parece que están evitando que nos escapemos.

La sintaxis de uno de estos valores de atributos colocados en un enlace sería la siguiente.

`Acceder a guiarte.com`

Frames - Anidar frames

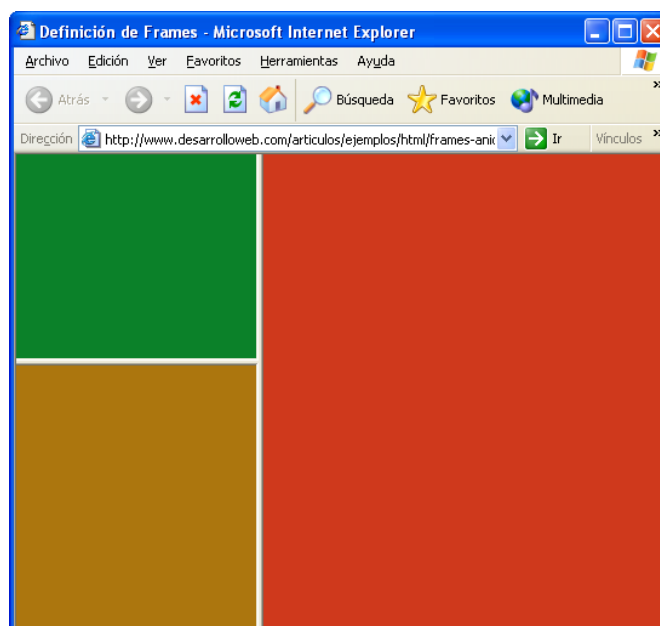
Para crear estructuras de marcos en las que se mezclen las filas y las columnas debemos anidar etiquetas `<FRAMESET>`. Empezando por la partición de frames más general, debemos colocar dentro las particiones de frames más pequeñas. La manera de indicar esto se puede ver fácilmente con un ejemplo.



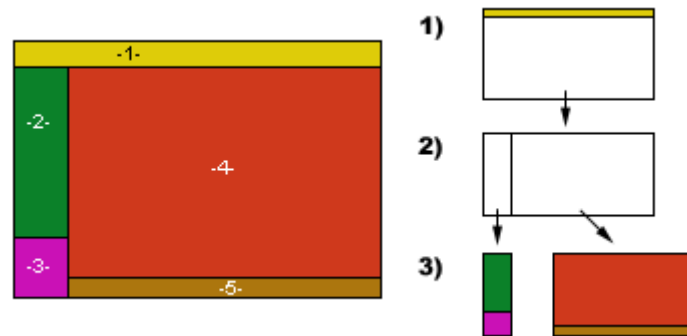
Los pasos para definir la anidación se pueden encontrar a la derecha.
Los distintos frames vienen numerados con el orden
en el que se escriben en el código.

En la imagen se puede ver el resultado final acompañada de la representación sobre la manera de definirlos. En primer lugar definimos una estructura de frames en dos columnas y dentro de la primera columna colocamos otra partición de frames en dos filas. El código necesario es el siguiente.

```
<frameset cols="200,*">  
  <frameset rows="170,*">  
    <frame src="pagina1.html">  
    <frame src="pagina2.html">  
  </frameset>  
  <frame src="pagina3.html">  
</frameset>
```



El ejemplo anterior se puede complicar un poco más si incluimos más particiones. Vamos a ver algo un poco más complicado para practicar más con las anidaciones de frames.

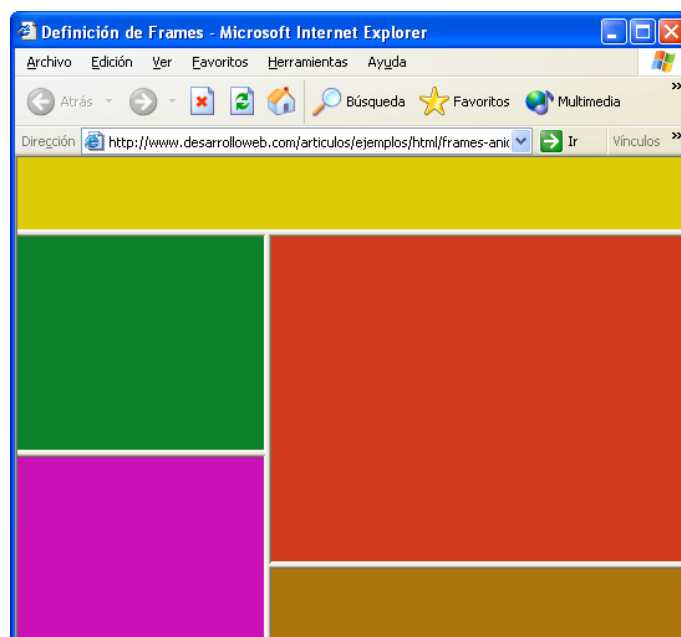


Los pasos para definir la anidación se pueden encontrar a la derecha.
Los distintos frames vienen numerados con el orden en el que se escriben en el código.

En la imagen se observa que el primer frameset a definir se compone de dos filas. Posteriormente, dentro de la segunda fila del primer frameset, tenemos otra partición en dos columnas, dentro de las que colocamos un tercer nivel de frameset con una definición en filas en los dos casos. El código se puede ver a continuación.

```
<frameset rows="60,*">
  <frame src="pagina1.html">
  <frameset cols="200,*">
    <frameset rows="*,150">
      <frame src="pagina2.html">
      <frame src="pagina3.html">
    </frameset>
    <frameset rows="*,60">
      <frame src="pagina4.html">
      <frame src="pagina5.html">
    </frameset>
  </frameset>
</frameset>
```

El resultado es el siguiente:



Anexo 1: Resumen de las etiquetas usadas para párrafos

ETIQUETA	FUNCION
<code><p>texto</p></code>	Marcan el comienzo y el final de un párrafo
<code>
texto</code>	Realiza un retorno de carro que inserta una línea en blanco
<code>texto</code>	Resaltan un texto en negrita
<code><p align="left"></code>	Alinea un texto a la izquierda
<code><p align="center"></code>	Alinea un texto al centro
<code><p align="right"></code>	Alinea un texto a la derecha
<code><div></code> párrafos <code></div></code>	Marca divisiones en las que definimos un mismo tipo de alineado para varios textos delimitados por <code><p>...</p></code>
<code><h1>texto</h1></code>	Introduce un encabezado de nivel 1 (existen 6 niveles diferentes)
<code><i>texto</i></code>	Resalta el texto en Itálica
<code><u>texto</u></code>	Subraya un texto (¡Ojo! Puede confundir con enlaces)
<code><sup>texto</sup></code>	Introduce un superíndice
<code><sub>texto</sub></code>	Introduce un subíndice

Anexo 2: Los colores y HTML

En la composición de webs juegan un papel muy importante los colores. Se indican en valores RGB, es decir, que para conseguir un color cualquiera mezclaremos cantidades de Rojo, Verde y Azul. Los valores RGB se indican en numeración hexadecimal, en base 16. (Los dígitos pueden crecer hasta 16. Como no hay tantos dígitos numéricos se utilizan las letras de la A a la F.



Tabla de color

0=0	4=4	8=8	C=12
1=1	5=5	9=9	D=13
2=2	6=6	A=10	E=14
3=3	7=7	B=11	F=15

Para conseguir un color, mezclaremos valores de esta manera:

RRGGBB

Donde cada valor puede crecer desde 00 hasta FF.

Ejemplo: Cómo se cambiaría la fuente para escribir en rojo:

`Rojo`

Al Atributo **color** le damos un valor RGB en formato hexadecimal. El carácter # se coloca al principio de la cadena.

Otros colores:

Naranja #FF8000

Verde turquesa #339966

Azul oscuro #000080

Como las páginas web las tienen que ver todos los usuarios, y los sistemas que utilizan para entrar son distintos, hay que utilizar colores compatibles con la paleta de todos ellos.

La forma de conseguir esto es limitando nuestros colores a los que se pueden conseguir utilizando la siguiente norma:

Utilizaremos siempre estos valores:
00
33
66
99
CC
FF

Nombre Color

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

Anexo 3: Colores compatibles en todos los sistemas

Ejemplos: **#3366FF** **#FF9900** **#666666**

Se consiguen los colores siguientes:

#000000	#000033	#000066	#000099	#0000CC	#0000FF
#003300	#003333	#003366	#003399	#0033CC	#0033FF
#006600	#006633	#006666	#006699	#0066CC	#0066FF
#009900	#009933	#009966	#009999	#0099CC	#0099FF
#00CC00	#00CC33	#00CC66	#00CC99	#00CCCC	#00CCFF
#00FF00	#00FF33	#00FF66	#00FF99	#00FFCC	#00FFFF
#330000	#330033	#330066	#330099	#3300CC	#3300FF
#333300	#333333	#333366	#333399	#3333CC	#3333FF
#336600	#336633	#336666	#336699	#3366CC	#3366FF
#339900	#339933	#339966	#339999	#3399CC	#3399FF
#33CC00	#33CC33	#33CC66	#33CC99	#33CCCC	#33CCFF
#33FF00	#33FF33	#33FF66	#33FF99	#33FFCC	#33FFFF
#660000	#660033	#660066	#660099	#6600CC	#6600FF
#663300	#663333	#663366	#663399	#6633CC	#6633FF
#666600	#666633	#666666	#666699	#6666CC	#6666FF
#669900	#669933	#669966	#669999	#6699CC	#6699FF
#66CC00	#66CC33	#66CC66	#66CC99	#66CCCC	#66CCFF
#66FF00	#66FF33	#66FF66	#66FF99	#66FFCC	#66FFFF
#990000	#990033	#990066	#990099	#9900CC	#9900FF
#993300	#993333	#993366	#993399	#9933CC	#9933FF
#996600	#996633	#996666	#996699	#9966CC	#9966FF
#999900	#999933	#999966		#9999CC	#9999FF
#99CC00	#99CC33	#99CC66	#99CC99	#99CCCC	#99CCFF
#99FF00	#99FF33	#99FF66	#99FF99	#99FFCC	#99FFFF
#CC0000	#CC0033	#CC0066	#CC0099	#CC00CC	#CC00FF
#CC3300	#CC3333	#CC3366	#CC3399	#CC33CC	#CC33FF
#CC6600	#CC6633	#CC6666	#CC6699	#CC66CC	#CC66FF
#CC9900	#CC9933	#CC9966	#CC9999	#CC99CC	#CC99FF
#CCCC00	#CCCC33	#CCCC66	#CCCC99	#CCCCCC	#CCCCFF
#CCFF00	#CCFF33	#CCFF66	#CCFF99	#CCFFCC	#CCFFFF
#FF0000	#FF0033	#FF0066	#FF0099	#FF00CC	#FF00FF
#FF3300	#FF3333	#FF3366	#FF3399	#FF33CC	#FF33FF
#FF6600	#FF6633	#FF6666	#FF6699	#FF66CC	#FF66FF
#FF9900	#FF9933	#FF9966	#FF9999	#FF99CC	#FF99FF
#FFCC00	#FFCC33	#FFCC66	#FFCC99	#FFCCCC	#FFCCFF
#FFFF00	#FFFF33	#FFFF66	#FFFF99	#FFFFCC	#FFFFFF

Anexo 4: Caracteres especiales en HTML

Caracteres especiales básicos

<	<	>	>
&	&	"	"

Caracteres especiales del HTML 2.0

Á	Á	À	À
É	É	È	È
Í	Í	Ì	Ì
Ó	Ó	Ò	Ò
Ú	Ú	Ù	Ù
á	á	à	à
é	é	è	è
í	í	ì	ì
ó	ó	ò	ò
ú	ú	ù	ù
Ä	Ä	Â	Â
Ë	Ë	Ê	Ê
Ï	Ï	Î	Î
Ö	Ö	Ô	Ô
Ü	Ü	Û	Û
ä	ä	â	â
ë	ë	ê	ê
ï	ï	î	î
ö	ö	ô	ô
ü	ü	û	û
Ã	Ã	å	å
Ñ	Ñ	Å	Å
Õ	Õ	Ç	Ç
ã	ã	ç	ç
ñ	ñ	Ý	Ý
õ	õ	ý	ý
Ø	Ø	ÿ	ÿ
ø	ø	Þ	Þ
Ð	Ð	þ	þ
ð	ð	Æ	Æ
ß	ß	æ	æ

Caracteres especiales del HTML 3.2

¼	¼	 	
½	½	¡	!

¾	¾	£	£
©	©	¥	¥
®	®	§	§
ª	ª	¤	¤
²	²	¦	¦
³	³	«	«
¹	¹	¬	¬
¯	—	­	­
µ	µ	º	º
¶	¶	´	´
·	·	¨	¨
°	°	±	±
¸	¸	»	»
¿	¿		

Otros caracteres especiales

×	×	¢	¢
÷	÷	€	€
“	“	™	™
”	”	‰	‰
Œ	Œ	ƒ	ƒ
‡	±	†	†

INDICE

Introducción al HTML	2
Sintaxis del HTML.....	3
Partes de un documento HTML.....	3
Tu primera página	4
formato de párrafos en HTML	5
Encabezados.....	6
formateando el texto.....	8
Color, tamaño y tipo de letra.....	9
Atributos para páginas.....	11
Listas	15
Listas desordenadas	15
Listas ordenadas.....	16
Listas de definición.....	17
Anidando listas	18
Caracteres especiales.....	19
Enlaces en HTML	19
Enlaces internos	20
Enlaces locales	21
Enlaces externos, de correo y hacia archivos.....	22
Enlaces remotos	22
Enlaces a direcciones de correo	22
Enlaces con archivos.....	23
Imágenes en HTML	24
Utilizar imágenes como enlaces.....	26
Alineación de imágenes con HTML	27
Tablas en HTML	29
Tablas en HTML. Atributos para filas y celdas.	30
Tablas anidadas.....	32
Formularios HTML.....	33
Elementos de Formularios. Campos de texto.....	34
Otros elementos de formulario	37
Envío, borrado y demás en formularios HTML.....	40
Frames en HTML	42
Frames - Explicación básica.....	42
Frames - Creación de una estructura simple	44
Frames - Una página en cada marco	45
Frames - Dirigir los enlaces	46
Frames - Anidar frames.....	48
Anexo 1: Resumen de las etiquetas usadas para párrafos	50
Anexo 2: Los colores y HTML	51
Anexo 3: Colores compatibles en todos los sistemas.....	52
Anexo 3: Colores compatibles en todos los sistemas.....	52
Anexo 4: Caracteres especiales en HTML.....	53
Ejercicio 1: Comida para todos (Tu primera página).....	¡Error! Marcador no definido.
Ejercicio 3: Colores y fuentes en html (Color, tamaño y tipo de letra) ..	¡Error! Marcador no definido.
Ejercicio 4: Colores en html (Atributos para páginas).....	¡Error! Marcador no definido.
Ejercicio 5: Operar sobre imágenes (Imágenes en HTML)	¡Error! Marcador no definido.
Ejercicio 6: Tablas I (Tablas en HTML).....	¡Error! Marcador no definido.

Ejercicio 7: Tablas II (Tablas en HTML)	¡Error! Marcador no definido.
Ejercicio 8: Formulario de datos (Formularios en HTML).....	¡Error! Marcador no definido.
Ejercicio 9: Frames (Frames).....	¡Error! Marcador no definido.