

HTML

HTML

1.	Introducción al HTML	4
2.	Sintaxis del HTML	6
3.	Tu primera página	8
4.	Formato de párrafos en HTML	11
5.	Formateando el texto	15
6.	Color, tamaño y tipo de letra	18
7.	Atributos para páginas	21
8.	Listas I	24
9.	Listas II	26
10.	Listas III	28
11.	Caracteres especiales	30
12.	Enlaces en HTML	32
13.	Enlaces internos	33
14.	Enlaces locales	34
15.	Enlaces externos, de correo y hacia archivos	36
16.	Imágenes en HTML	39
17.	Alineación de imágenes con HTML	42
18.	Tablas en HTML	45
19.	Tablas en HTML. Atributos para filas y celdas	47
20.	Tablas en HTML. Atributos de la tabla y conclusión	50
21.	Formularios HTML	
22.	Elementos de Formularios. Campos de texto	
23.	Otros elementos de formulario	
24.	Envío, borrado y demás en formularios HTML	
25.	Mapas de imágenes con HTML	
26.	Frames en HTML	
27.	Frames - Explicación básica	
28.	Frames - Creación de una estructura simple	
29.	Frames - Una página en cada marco	
30.	Frames - Dirigir los enlaces	
31.	Frames - Anidar frames	
32.	Frames - Atributos avanzados	
33.	Ventajas e inconvenientes del uso de frames	
34.	Las nuevas etiquetas de HTML 4.0	
35.	Las nuevas etiquetas de HTML 4.0 (1)	
36.	Las nuevas etiquetas de HTML 4.0 (2)	
37.	Sonido en HTML I, introducción	
38.	Sonido en HTML II, características del sonido digital	
39.	Sonido en HTML (III)	
40.	Sonido en HTML (IV)	
41.	Sonido en HTML (V)	
42.	Declaración DOCTYPE en documentos HTML	
	APENDICE A. Los colores y HTML	54

1. 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. Numerosos estándares se han presentado ya. El HTML 4.01 es el último estándar a septiembre de 2001.

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, JavaScript 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 webmaster 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 sí. 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.

Existen otro tipo de editores específicos para la creación de páginas web los cuales ofrecen muchas facilidades que nos permiten aumentar nuestra productividad. No obstante, es aconsejable en un principio utilizar una herramienta lo más sencilla posible para poder prestar la máxima atención a nuestro código y familiarizarnos lo antes posible con él. Siempre tendremos tiempo más delante de pasarnos a editores más versátiles con la consiguiente ganancia de tiempo.

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"

Consejo: Utiliza siempre la misma extensión en tus archivos HTML. Eso evitará que te confundas al escribir los nombres de tus archivos unas veces con .htm y otras con .html. Si trabajas con un equipo en un proyecto todavía es más importante que os pongáis todos de acuerdo en la extensión.

2. 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:

1. Una apertura de forma general <etiqueta>
2. 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.

3. Tu primera página

Vamos a ver cómo se hace una página muy sencilla en HTML, que sirva de práctica a los debutantes.

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><b>Bienvenido,</b></p>
<p>Estás en la página <b>Comida para Todos</b>. </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 guardarla y colocaremos su nombre, por ejemplo mi_pagina.html

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

Nuestro consejo 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. Algun carácter como el guion "-" o el guion bajo "_" te puede ayudar a separar las palabras. Por ejemplo quienes_somos.html

Con el documento HTML creado, podemos ver el resultado obtenido a partir de un navegador. Es conveniente, llegado a este punto, hacer hincapié en el hecho de que no todos los navegadores son idénticos. Desgraciadamente, los resultados de nuestro código pueden cambiar de uno a otro por lo que resulta aconsejable visualizar la página en varios. Generalmente se usan Internet Explorer y Netscape como referencias ya que son los más extendidos.

A decir verdad, en el momento que estas líneas son escritas, Internet Explorer acapara la inmensa mayoría de usuarios (90% más o menos) y Netscape está relegado a un segundo plano. Esto no quiere decir que lo debemos dejar totalmente

de lado ya que el 10% de visitas que puede proporcionarnos puede resultar muy importante para nosotros. Por otra parte, parece que se ha hecho pública la intención de Netscape de desviar un poco su temática de negocios hacia otros derroteros y abandonar esta llamada "lucha de navegadores" en la cual estaba recibiendo la peor parte.

Pues bien, volviendo al tema, una vez creado el archivo .html o .htm, podemos visualizar el resultado de nuestra labor abriendo dicha página con un navegador. Para hacerlo, la forma resulta diferente dependiendo del navegador:

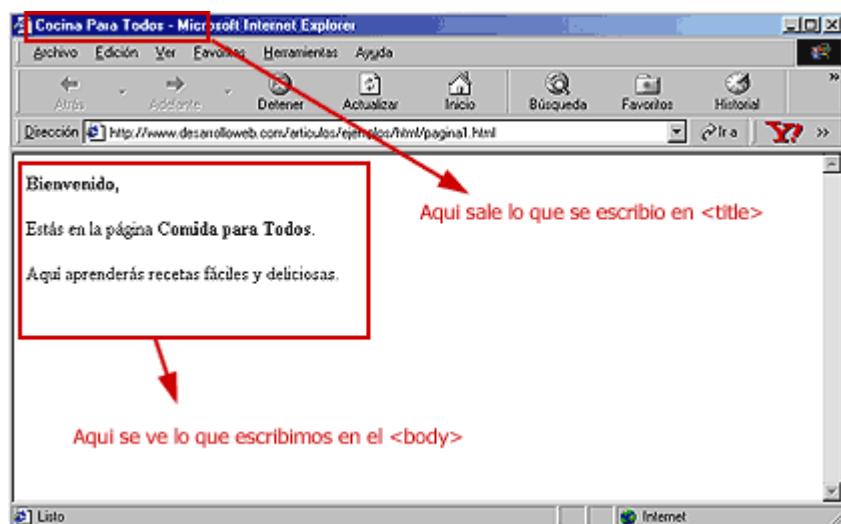
Si estamos empleando el Explorer, hemos de ir a la barra de menú, elegir Archivo y seleccionar Abrir. Una ventana se abrirá. Pulsamos sobre el botón Examinar y accederemos a una ventana a partir de la cual podremos movernos por el interior de nuestro disco duro hasta dar con el archivo que deseamos abrir.

La cosa no resulta más difícil para Netscape. En este caso, nos dirigimos también a la barra de menú principal y elegimos File y a continuación Open File. La misma ventana de búsqueda nos permitirá escudriñar el contenido de nuestro PC hasta dar con el archivo buscado.

Nota: También puedes abrir el archivo si accedes al directorio donde lo guardaste. En él podrás encontrar tu archivo HTML y verás que tiene como ícono el logotipo de Netscape o el de Internet Explorer. Para abrirlo simplemente hacemos un doble click sobre él.

Una vez abierto el archivo podréis ver vuestra primera página web. Algo sencillita pero por algo se empieza. Ya veréis como en poco tiempo seremos capaces de mejorar sensiblemente.

Fijaos en la parte superior izquierda de la ventana del navegador. Podréis comprobar la presencia del texto delimitado por la etiqueta <title>. Esta es una de las funciones de esta etiqueta, cuyo principal cometido es el de servir de referencia en los motores de búsqueda como Altavista, Yahoo, etc.



Por otro lado, los elementos que colocamos entre la etiqueta <body> y </body> se pueden ver en el espacio reservado para el cuerpo de la página.

Si ahora hacéis click con el botón derecho sobre la página y elegís Ver código fuente (o View page source) veréis como en una ventana accesoria aparece el código de nuestra página. Este recurso es de extremada importancia ya que nos permite ver el tipo de técnicas empleadas por otros para la confección de sus páginas.

Con todo esto asimilado ya estamos en condiciones de adentrarnos un poco más en la descripción de algunas de las etiquetas más empleadas del HTML.

Possible problema: Al utilizar el Block de Notas en Windows en ocasiones, aunque le digamos que es un archivo .html, el documento se guarda como si fuera un texto y no una página web. Lo que está pasando es que el Block de Notas tiene predeterminado guardar sus archivos con extensión .txt y en realidad lo que está guardando en el disco duro es mi_pagina.html.txt

Para conseguir tener el control de las extensiones en el block de notas y en Windows en general podemos acceder a MI-PC y en el menú de Ver seleccionáis "Opciones de carpeta". En la ventana que sale pulsamos en la solapa "Ver" y nos permite deseleccionar una caja de selección que pone algo como "Ocultar extensiones para los tipos de archivos conocidos". (Así se hace en Win98, puede variar un poco en otras versiones de Windows.)

Con ello conseguiremos que se vea siempre la extensión del archivo con el que estamos trabajando y que el Block de Notas nos haga caso cuando le indicamos que grabe el archivo con otra extensión que no sea .txt

4. Formato de párrafos en HTML

En los capítulos anteriores hemos presentado a título de ejemplo algunas etiquetas que permiten dar formato a nuestro texto. En este capítulo veremos con más detalle las 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.

Nota: Existen otras etiquetas que no tienen su correspondiente de cierre, como `` para las imágenes, las veremos más adelante. Esto ocurre porque un salto de línea o una imagen no empiezan y acaban más adelante sino que sólo tienen presencia en un lugar puntual.

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 seria:

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.

Nota: 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.

El atributo align no es exclusivo de la etiqueta `<p>`. Otras etiquetas muy comunes, que veremos más adelante, entre las cuales se introducen texto o imágenes, suelen hacer uso de este atributo de una forma habitual.

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

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 2</h2>
<h3>Encabezado de nivel 3</h3>
<h4>Encabezado de nivel 4</h4>
<h5>Encabezado de nivel 5</h5>
<h6>Encabezado de nivel 6</h6>
</body>
</html>
```

Consejo: 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.

5. 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 primera por simple razón de esfuerzo.

Escribiendo un código de este tipo:

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

Obtenemos este resultado:

Texto en negrita

Nota: ¿Qué diferencia hay entre **** y ****?

Aunque las dos etiquetas hacen el mismo efecto, tienen una peculiaridad que las hace distintas. La etiqueta **** indica negrita, mientras que la etiqueta **** indica que se debe escribir resaltado. El HTML lo interpretan los navegadores según su criterio, es por eso que las páginas se pueden ver de distinta manera en unos browsers y en otros. La etiqueta **<H1>** quiere decir "encabezado de nivel 1", es el navegador el responsable de formatear el texto de manera que parezca un encabezado de primer nivel. En la práctica los encabezados de Internet Explorer y Netscape son muy parecidos (tamaño de letra grande y en negrita), pero otro navegador podría colocar los encabezados con subrayado si le pareciese oportuno.

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 deseé. En la práctica **** coloca el texto en negrita, pero podría ser que un navegador decidiese resaltar colocando negrilla, subrayado y color rojo en el texto.

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 $^{13}\text{CC}_3\text{H}_4\text{CINOS}$ es un heterociclo alergeno enriquecido

El resultado:

La $^{13}\text{CC}_3\text{H}_4\text{CINOS}$ 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

Consejo: Cuando anides etiquetas HTML hazlo correctamente. Nos referimos a que si abres etiquetas dentro de otra más principal, antes de cerrar la etiqueta principal cierres 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, por dos razones:

1. Sistemas como XML no son tan permisivos con estos errores y puede que en el futuro nuestras páginas no funcionen correctamente.
2. A los navegadores les cuesta mucho tiempo de procesamiento resolver este tipo de errores, incluso más que construir la propia página y debemos evitarles que sufran por una mala codificación.

6. 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 (CSS), 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

Nota: 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 desestimarán.

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. 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 Apendice A: "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 nemotécnico:

Nombre	Color
Aqua	
Black	
Blue	
Fuchsia	
Gray	
Green	

Lime	
Maroon	
Navy	
Olive	
Purple	
Red	
Silver	
Teal	
White	
Yellow	

Este texto está en rojo

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.

Pongamos pues en práctica 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".

(Ver página de ejemplo)

7. 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 capítulo anterior y en el Apéndice A hemos aprendido 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 posteriores veremos como se insertan imágenes con HTML y los tipos de imágenes que se pueden utilizar.

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>

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

Consejo: 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 seleccionarlo de 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.

Creo que lo mejor será poner un ejemplo. 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.

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 claro)

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)

Además, vamos a ver otra página sin márgenes, por si alguien necesita ver el ejemplo en estas líneas.

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

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.

8. Listas I

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

Nota: En algunos navegadores no funciona la opción de cambiar el tipo de viñeta a mostrar y por mucho que nos empeñemos, siempre saldrá el redondel negro.

En caso de que no funcione siempre podemos construir la lista a mano con la viñeta que queramos utilizando las tablas de HTML. Veremos más adelante cómo trabajar con tablas.

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

9. Listas II

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 será 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

Nota: Recordamos que en algunos navegadores no funciona la opción de cambiar el tipo de viñeta a mostrar

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.

Os proponemos 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 numeros romanos empezando por el 10

- x. Elemento x
- xi. Elemento xi

10. Listas III

Terminamos el tema de listas estudiando las listas de definición. Veremos también la anidación de listas.

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.

Aquí os proponemos un código que podrá aclarar este sistema:

```
<p>Diccionario de la Real Academia</p>
```

```
<dl>
  <dt>Brújula
  <dd>Señórula montada en una escóbula
  <dt>Oreja
  <dd>Sesenta minutejos
</dl>
```

El efecto producido:

Diccionario de la Real Academia

Brújula
 Señórula montada en una escóbula
Oreja
 Sesenta minutejos

Fíjate en que cada línea `<dd>` está 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>
  </dd>
</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

11. 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: página

Caracteres especiales básicos

En realidad estos caracteres se usan en HTML para no confundir un principio o final de etiqueta, unas comillas o un & con su correspondiente carácter.

<	<	>	>
&	&	"	"

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	Ã	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	&lquo;	«
¹	¹	¬	¬
¯	‐	­	‐
µ	µ	º	º
¶	¶	´	‘
·	·	¨	…
°	°	±	±
¸	¸	»	»
¿	¿		

Otros caracteres especiales

×	×	¢	¢
÷	÷	€	€
“	“	™	™
”	”	‰	%o
Œ	Œ	ƒ	f
‡	‡	†	†

12. 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 HTLM 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 apuntador. 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.

13. 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 llevará 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.

14. Enlaces locales

Como hemos dicho, un sitio web esta constituido de páginas ínter conexas. En el capitulo 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 práctica 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>
```

15. 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:pepe@empresa.com">pepe@empresa.com</a>
```

Consejo: 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.

[pepe@empresa.com](mailto:pepe@empresa.com?subject=contacto)

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

[eugim@desarrolloweb.com](mailto:eugim@desarrolloweb.com?subject=contacto&cc=colabora@desarrolloweb.com)

Nota: 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 están dirigidos 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.

[Descarga mi_fichero.zip](mi_fichero.zip)

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 guardarla en disco".

Consejo: 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>
```

16. Imágenes en HTML

Sin duda uno de los aspectos más vistosos y atractivos de las páginas web es el grafismo. La introducción en nuestro texto de imágenes puede ayudarnos a explicar más fácilmente nuestra información y darle un aire mucho más estético. El abuso no obstante puede conducirnos a una sobrecarga que se traduce en una distracción para el navegante, quien tendrá más dificultad en encontrar la información necesaria, y un mayor tiempo de carga de la página lo que puede ser de un efecto nefasto si nuestro visitante no tiene una buena conexión o si es un poco impaciente.

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.

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 gráfico 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 gráfico.

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 brevíssima 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 mostrará 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 gráfico 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 reservará 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 pixels, 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.

Truco: 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 pixels.
- 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>

```

17. 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...
</p>
```

Quedaría así:

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...

A logo for 'DESARROLLOWEB'. It features a red stylized letter 'W' inside a circle. Two white arrows point towards the center of the circle from the left side. The background is black with a grid pattern.

```
<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...  
</p>
```

Quedaría así:



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...

Si en algún momento deseásemos dejar de llenar 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">
```

Saltara verticalmente hasta encontrar el lateral izquierdo libre.

```
<br clear="right">
```

Saltara verticalmente hasta encontrar el lateral derecho libre.

```
<br clear="all">
```

Saltará verticalmente hasta encontrar ambos laterales libres.

Ejemplo de clear:



Texto tan extenso como queramos que cubrirá la parte izquierda.

Esto está debajo de la imagen.

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 práctica.

Nos queda explicar como introducir debajo de la imagen un pie de foto o explicación. Para ello tendremos que ver antes de nada las tablas, en los próximos capítulos...

18. Tablas en HTML

En un principio nos podría parecer que las tablas son raramente útiles y que pueden ser utilizadas principalmente para listar datos como agendas, resultados y otros datos de una forma organizada. Nada más lejos de la realidad.

Hoy, gran parte de los diseñadores de páginas basan su maquetación en este tipo de artilugios. En efecto, una tabla nos permite organizar y distribuir los espacios de la manera más optima. Nos puede ayudar a generar texto en columnas como los periódicos, prefijar los tamaños ocupados por distintas secciones de la página o poner de una manera sencilla un pie de foto a una imagen.

Puede que en un principio nos resulte un poco complicado trabajar con estas estructuras pero, si deseamos crear una página de calidad, tarde o temprano tendremos que vernos las con ellas y nos daremos cuenta de las posibilidades nos ofrecen.

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

Nota: Hasta aquí hemos visto todas las etiquetas que necesitamos conocer para crear tablas. Existen otras etiquetas, pero lo que podemos conseguir con ellas se puede conseguir también usando las que hemos visto.

Por poner un ejemplo, señalamos la etiqueta `<th>`, que sirve para crear una celda cuyo contenido esté formateado como un título o cabecera de la tabla. En la práctica, lo que hace es poner en negrita y centrado el contenido de esa celda, lo que se puede conseguir aplicando las correspondientes etiquetas dentro de la celda. Así: `<td align="center">contenido de la celda</td>`.

A partir de esta idea simple y sencilla, las tablas adquieren otra magnitud cuando les incorporamos toda una batería de atributos aplicados sobre cada tipo de etiquetas que las componen. A lo largo de los siguientes capítulos nos adentraremos en el estudio de estos atributos de manera a proporcionaros los útiles indispensables para una buena puesta en página.

19. 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. La forma más útil y actual de dar forma a las celdas es a partir de las hojas de estilo en cascada.

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.

Nota: El atributo `height` no funciona en todos los navegadores, además, su uso no está muy extendido. Las celdas por lo general tienen el alto que necesitan para que quepa todo el contenido que se le haya insertado, es decir, crecen lo suficiente para que quepa lo que hemos colocado dentro.

El atributo `width` si que funciona en todos los navegadores y lo tendrás que utilizar constantemente. Si le asignamos un ancho a la celda, el ancho será respetado y si

dicha celda tiene mucho texto o cualquier otro contenido, la celda crecerá hacia abajo todo lo necesario para que quepa lo que hemos colocado.

Un matiz al último párrafo. Se trata de que si definimos una celda de un ancho 100 por ejemplo, y colocamos en la celda un contenido como una imagen que mida más de 100 pixeles, la celda crecerá en horizontal todo lo necesario para que la imagen quepa. Si el elemento, aunque más ancho, fuera divisible (como un texto) el ancho sería respetado y el texto crecería hacia abajo o lo que es lo mismo, en altura, como señalábamos en el anterior párrafo.

Estos últimos cuatro atributos descritos son de gran utilidad. Concretamente, height y width nos ayudan a definir las dimensiones de nuestras celdas de una forma absoluta (en pixels o puntos de pantalla) o de una forma relativa, es decir por porcentajes referidos al tamaño total de la tabla.

A título de ejemplo:

```
<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.

Hay que tener en cuenta que, definidas las dimensiones de las celdas, el navegador va a hacer lo que buenamente pueda para satisfacer al programador. Esto quiere decir que puede que en algunas ocasiones el resultado que obtengamos no sea el esperado. Concretamente, si el texto presenta una palabra excesivamente larga, puede que la anchura de la celda se vea aumentada para mantener la palabra en la misma línea. Por otra parte, si el texto resulta muy largo, la celda aumentara su altura para poder mostrar todo su contenido.

Análogamente, si por ejemplo definimos dos anchuras distintas a celdas de una misma columna, el navegador no sabrá a cual hacer caso. Es por ello que resulta conveniente tener bien claro desde un principio como es la tabla que queremos diseñar. No esta de más si la prediseñamos en papel si la complejidad es importante. El HTML resulta en general fácil pero las tablas pueden convertirse en un verdadero quebradero de cabeza si no llegamos a comprenderlas debidamente.

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. Los dejamos a vuestra propia investigación.

20. Tablas en HTML. Atributos de la tabla y conclusión.

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.

Los atributos que definen las dimensiones, height y width, funcionan de una manera análoga a la de las celdas tal y como hemos visto en el capítulo anterior. Contrariamente, el atributo align no nos permite justificar el texto de cada una de las celdas que componen la tabla, sino más bien, justificar la propia tabla con respecto a su entorno.

Vamos a poner tres ejemplos de alineado de tablas, centradas, alineadas a la derecha y a la izquierda.

Ejemplo de tabla centrada

Esta tabla está centrada
(aling="center"). Solo tiene una celda.

Este sería un texto cualquiera colocado al lado de una tabla centrada

Ejemplo de tabla alineada a la derecha

Para que se vea el efecto de alineado a la tabla debemos colocar un texto al lado y el texto rodeará la tabla, igual que ocurría con las imágenes alineadas a un lado.

Esta tabla está alineada a la derecha
(aling="right"). Solo tiene una celda.

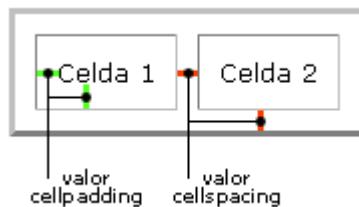
Ejemplo de tabla alineada a la izquierda

Esta tabla está alineada a la izquierda
(aling="left"). Solo tiene una celda.

Para que se vea el efecto de alineado a la tabla debemos colocar un texto al lado y el texto rodeará la tabla, igual que ocurría con las imágenes alineadas a un lado.

Los atributos cellpadding y cellspacing nos ayudaran a dar a nuestra tabla un aspecto más estético. En un principio puede parecernos un poco confuso su uso pero un poco de práctica será suficiente para hacerse con ellos.

En la siguiente imagen podemos ver gráficamente el significado de estos atributos.



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 rojo de fondo	El atributo bgcolor de la 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.

Consejo: Páginas como DesarrolloWeb.com y muchas otras (La mayoría de las páginas avanzadas) que basan su diseño en tablas, realizan anidaciones de tablas constantemente para meter unos elementos de la página dentro de otros. Se pueden anidar tablas sin límite, sin embargo, en el caso de Netscape 4 hay que tener cuidado con el número de tablas que anidamos, porque a medida que metemos una tabla dentro de otra y otra dentro de esta y otra más, aumentando el grado de anidación sucesivamente... podemos encontrar problemas en su visualización y puede que la página tarde un poco de tiempo más en mostrarse en pantalla.

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	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>
```

Ejemplos prácticos

Hasta aquí la información que pretendíamos transmitiros sobre las tablas en HTML. Sería importante ahora realizar algún ejemplo de realización de una tabla un poco compleja. Por ejemplo la siguiente:

Animales en peligro de extinción			
Nombre	Cabezas	Previsión 2010	Previsión 2020
Ballena	6000	4000	1500
Oso Pardo	50	0	
Lince	10		
Tigre	300	210	

Otro ejemplo de tabla con el que podemos practicar:

Climas de América del Sur			
Parte de arriba de América del Sur. Países como:	Venezuela Colombia Ecuador Perú	Parte de abajo de América del Sur. Países como:	Argentina Chile Uruguay Paraguay
Bosque tropical, clima de sabana, clima marítimo con inviernos secos.		Climas marítimos con veranos secos, con inviernos secos, climas frios, clima de estepa, clima desértico.	

Apéndice A. Los colores y HTML

El 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 RBG 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.

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:

```
<font color="#FF0000">Rojo</font>
```

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

Colores compatibles en todos los sistemas

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

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