



Manual de HTML

Alonso Cornejo

Publicado: 2012

Categoría(s): No Ficción, Informática, Programación, HTML

Etiqueta(s): computacion html lenguaje pagina web

Introducción a HTML

Introducción al manual de HTML y al lenguaje de modelado de páginas web. Veremos qué es HTML y las primeras nociones que nos ayudarán a realizar las primeras pruebas de creación de una página web sencilla.

Parte 1

Prólogo al manual de HTML

A quién va dirigido este manual y las lecturas aconsejadas antes de abordarlo.

Bienvenidos al manual de HTML de DesarrolloWeb. A través de todos estos capítulos vamos a descubrir el lenguaje utilizado para la creación de páginas web: el **Hyper Text Markup Language**, más conocido como **HTML**.

Puede que en un principio, el hecho de hablar de un lenguaje informático pare los pies a más de uno. No os asustéis, el HTML no deja de ser más que una forma un tanto peculiar de dar formato a los textos e imágenes que pretendemos ver por medio de un navegador.

Antes de entrar en materia, lo cual haremos de una forma directa y practica, os **recomendamos fervorosamente la lectura previa de nuestro manual Publicar en Internet**. A partir de esta guía, aprenderéis los conceptos más básicos necesarios para creación de un sitio web. También os permitirá acceder a este manual con unos conocimientos de base sobre HTML imprescindibles y os dejara bien claro lo que su conocimiento aporta con respecto al simple uso de editores de HTML. Además, en dicho manual sobre la publicación en Internet, explicamos también algo tan importante como qué hace falta para subir una página web realizada que tenemos en nuestro ordenador a un servidor de Internet. Si lo deseamos, lo podemos ver en vídeo en el Videotutorial sobre subir una web a Internet.

El público al que va enfocado este manual es a todos aquellos que, con conocimientos mínimos de informática, desean hacer mundialmente público un mensaje, una idea o una información usando para ello el medio más práctico, económico y actual: Internet.

Lo que necesitáis como base para llevar a buen término el aprendizaje (aparte de leer el manual Publicar en Internet) es:

- Saber escribir con un teclado
- Saber manejar un ratón
- Tener ganas de aprender

Como podréis ver, cualquier persona que sepa manejar un ordenador tiene los conocimientos básicos para aprender HTML. Si le pones un poco de ganas y sigues este manual hasta el final, tendrás las siguientes habilidades o conocimientos:

- Capacidad para crear y publicar vuestro propio sitio web con un mínimo de calidad.
- Conocimientos de todo tipo sobre las tecnologías y herramientas empleadas en el ámbito de la Red.
- Posiblemente una afición que puede convertirse en pasión y terminar, en algunos casos, siendo un vicio o un oficio.

Podéis formular vuestras cuestiones y, esperamos que en un futuro ayudar a otros compañeros, enviandodiscusiones sobre HTML o bien en la lista de correo de DesarrolloWeb.com.

Para quien no sepa nada sobre crear una página web, y le gusta que le expliquen las cosas desde cero y de manera visual, recomendamos ver el vídeo donde mostramos el proceso de creación de la primera página básica. Además, para complementar las explicaciones de este manual, también recomendamos el videotutorial de HTML.

Finalmente, antes de comenzar con el temario, queremos daros una referencia importante a la sección HTML a fondo, donde publicamos todos los contenidos que tienen que ver con HTML y donde encontrarás este y otros manuales relacionados con el lenguaje.

Pasemos pues sin más preámbulos a ver de qué se trata el HTML...

Parte 2

Introducción al HTML

Las primeras cosas que debes saber sobre HTML: historia, objetivos y demás conocimientos donde sentar las bases del manual.

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 sonar son las CSS, JavaScript u otros. Veremos más adelante en qué consisten algunas de ellas.

Otros de los problemás 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.

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 adelante de pasarnos a editores más versátiles con la consiguiente ganancia de tiempo.

Para tener más claro todo el tema de editores y los tipos que existen, visita los artículos:

- Editores de HTML.
- Bloc de notas.
- También puedes acceder a descripciones editores más complejos que el Block de Notas, pero más potentes como Homesite o UltraEdit.

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 más importante que os pongáis todos de acuerdo en la extensión.

Parte 3

Sintaxis del HTML

Descripción de la sintaxis con la que se trabaja en el lenguaje HTML, así como la estructura que tendrá el documento básico HTML.

El HTML es un lenguaje de marcas que basa su sintaxis en un elemento de base al que llamamos etiqueta. A través de las etiquetas vamos definiendo los elementos del documento, como enlaces, párrafos, imágenes, etc. Así pues, un documento HTML estará constituido por texto y un conjunto de etiquetas para definir la forma con la que se tendrá que presentar el texto y otros elementos en la página.

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.

Parte 4

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 guardarlo 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. Algún caracter como el guión "-" o el guión 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 Firefox 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 Firefox esta 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.

Nota: En la disputa por ser el navegador más usado, en la actualidad otros productos vienen tomando fuerza como Chrome, el navegador de Google. Para el que le interese este tema y conocer datos reales sobre los porcentajes de uso de cada navegador, se puede leer uno de los informes que publicamos periódicamente, como el ranking de uso de los navegadores en noviembre de 2010.

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 al 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 icono 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 o Yahoo.

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.

Se puede ver la página del ejemplo en funcionamiento aquí.

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 extrema 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.

Posible 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 seleccionar 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

Formatos básicos con HTML

Cómo realizar el formato de textos que se colocan en una página web. Aprende a utilizar tus primeras etiquetas HTML y atributos para definir los contenidos de la página y aplicar un formato básico.

Parte 5

Formato de párrafos en HTML

Cómo colocar párrafos y saltos de línea en páginas web. También vemos los encabezados como párrafos que sirven de título.

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 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.

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

No vamos a escribir en esta ocasión el código fuente del ejercicio. Podemos verlo en funcionamiento en nuestro navegador y en la ventana podemos obtener el código fuente seleccionando en el menú Ver la opción Código fuente.

Ver el ejercicio en marcha.

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>
```

Se puede ver el ejercicio en una página aparte.

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.

Parte 6

New Part

Vemos como colocar negritas, itálicas, subrayados, subíndices y supréndices.

Por **Rubén Alvarez**

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.

Escribiendo un código de este tipo:

`Texto en negrita`

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

desea. 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

`¹³CC₃H₄ClNOS` es un heterociclo alergeno enriquecido

El resultado:

La ¹³CC₃H₄ClNOS 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 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, 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.

Todo lo que se ha visto hasta el momento en el Manual de HTML se puede ver en un vídeo para aprender visualmente. Si te interesa puedes acceder al Videotutorial de HTML - Parte 1, documento básico y formato de texto.

Parte 7

Color, tamaño y tipo de letra

Seguimos aprendiendo etiquetas que nos sirven para formatear el texto.

Por **Rubén Alvarez**

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 (de las cuales ya tendremos tiempo de hablar), 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.

``Este texto tiene otra tipografía``

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 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.

`Este texto es más grande`

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 a partir de este artículo: 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 memotécnico:

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 disposicion de crear un texto formateado de una forma realmente elaborada.

Pongamos pues en practica todo lo que hemos aprendido en estos capitulos 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".

Se puede ver una posible solución del ejercicio en este enlace. (Ver el código fuente de la página para ver cómo lo hemos resuelto)

Parte 8

Los colores y HTML

Los colores y HTML

01 de enero de 2001

Compartir en redes sociales

Valoración del artículo:

3 votos

Enviar un comentario

Ver comentarios del artículo (14)

En este reportaje tratamos de acercarte todos los detalles relativos al buen uso de colores en HTML. Colores compatibles con todos los sistemas.

Por **Miguel Angel Alvarez**

Tabla de color

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 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.

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 caracter # 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:

Por último, te ofrecemos unos archivos con programillas en JavaScript que te pueden ayudar a encontrar los colores adecuados. Entre otros, podrás encontrar el script que genera la tabla de arriba.

Parte 9

Atributos para páginas

Explicamos una serie de atributos que se aplican de manera global a toda la página, como el color de fondo el del texto, de los enlaces, márgenes, etc.

Por **Miguel Angel Alvarez**

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 taller de los colores y HTML 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 más adelante 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">
```

Se puede ver el efecto de colocar ese fondo en una página a parte.

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 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.

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 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" vlink="ffff00">
```

El efecto se puede ver en una página a parte.

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)

Tenemos un artículo sobre la utilización de estos atributos para hacer diseños avanzados con tablas en distintas definiciones de pantalla, que puede ser interesante de leer.

Un ejemplo de página sin margen es la propia página de DesarrolloWeb.com, que estás visitando actualmente. (Por lo menos a la hora de escribir este artículo) 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 margin-
width=0 bgcolor="ffffff">
```

```
<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. Puede verse el ejemplo en una página a parte.

Todo lo que hemos visto hasta ahora en el Manual de HTML lo podemos encontrar en vídeo y en concreto las explicaciones de los últimos artículos se han recogido en el Videotutorial de HTML - Fuentes, colores y estilos de BODY.

Parte 10

Caracteres especiales

Una referencia útil, y una lista completa de los caracteres especiales del HTML.

Por **Miguel Angel Alvarez**

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 Û

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 Ì 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 Ⓒ ƒ f
‡ ‡ † †

Para descargar la lista de caracteres especiales: **caracteres-especiales.zip 2Kb**

Referencia: Hay un videotutorial en DesarrolloWeb.com que trata sobre enlaces y caracteres especiales. No son temas muy relacionados entre si, pero seguro que nos sirve como práctica para aprender todo lo que se ha tratado en este artículo.

Listas en HTML

A continuación comenzaremos a explicar las listas de HTML que implican varias etiquetas para crear su estructura. Veremos varios tipos de listas que se pueden utilizar para diversos objetivos.

Parte 11

Listas I - Listas desordenadas

Vemos lo que son las listas y señalamos los tres tipos que hay. Estudiamos las listas desordenadas.

Por **Rubén Alvarez**

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 originalmente están pensadas para citar, numerar y definir cosas a través de características, o al menos así lo hacemos en la escritura de textos. Sin embargo, las listas finalmente se utilizan para mucho más que enumerar una serie de puntos, en realidad son un recurso muy interesante para poder maquetar elementos diversos, como barras de navegación, pestañas etc. Pero esto lo veremos más adelante, cuando apliquemos estilos CSS a las listas.

De momento, en este Manual de HTML, trataremos las listas desde el punto de vista de su construcción y veremos los diferentes tipos que existen, y que podemos utilizar para resolver nuestras distintas necesidades a la hora de escribir textos en HTML.

Referencia: Todos los tipos de listas los hemos explicado en vídeo y se pueden revisar en el Videotutorial de HTML, más concretamente en la entrega siguiente: Vídeo sobre las Listas HTML.

Podemos distinguir tres tipos de listas HTML:

- 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

Parte 12

New Part

Estudiamos otro tipo de listas: las listas ordenadas.

Por **Rubén Álvarez**

Estamos en el Manual de HTML y continuamos estudiando las listas de HTML, con las que crear estructuras atractivas para presentar la información. En el capítulo anterior vimos las listas desordenadas y ahora estudiaremos las listas ordenadas.

Listas ordenadas

Las listas ordenadas sirven también para presentar información, en diversos elementos o items, con la particularidad que éstos estarán precedidos de un número o una letra para enumerarlos, siempre por un orden.

Para realizar las listas ordenadas usaremos las etiquetas `` (ordered list) y su cierre. Cada elemento será igualmente indicado por la etiqueta ``, que ya vimos en las listas desordenadas.

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

1. Elemento a

2. Elemento b

Ordenamos por numeros romanos empezando por el 10

1. Elemento x
2. Elemento xi

Parte 13

Listas III - Listas de definición

Vemos las listas de definición y aprendemos a anidar listas para crear estructuras lista más complejas.

Por **Rubén Álvarez**

Terminamos el tema de listas en el manual de HTML de DesarrolloWeb.com estudiando las listas de definición. Veremos también la anidación de listas, que resultará un recurso interesante para estructurar datos un poco más complejos o enumerar elementos con una jerarquía.

Listas de definición

Las listas de definición sirven para hacer un conjunto de elementos con pares concepto-descripción. Es decir, se especificarán varios términos por su nombre y se escribirá una definición para cada uno. Cada elemento es presentado junto con su definición, uno detrás de otro.

Para realizar una lista de 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>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>
    </dd>
  </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>
  </li>
</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

```


Referencia: Con esto hemos acabado el tema sobre Listas HTML, sin embargo, podemos completar las explicaciones en el Videotutorial de HTML, en el Vídeo sobre las Listas HTML.

Todo sobre los enlaces en HTML

Los enlaces son los elementos que nos permiten navegar por las páginas HTML y son tan importantes que la web no tendría sentido sin ellos. Dedicaremos varios capítulos a explorar los distintos tipos de enlaces, sus usos y diversos consejos para hacer páginas navegables.

Parte 14

Enlaces en HTML

Vemos qué son los enlaces en HTML y los distintos tipos.

Por **Rubén Álvarez**

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.

Referencia: Para complementar las explicaciones sobre los enlaces HTML recomendamos ver el videotutorial sobre enlaces.

Parte 15

Enlaces internos

Los enlaces HTML que se hacen con otras partes de la misma página.

Por **Rubén Alvarez**

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:

Enlace con final de este documento, para que probéis su funcionamiento:

Ir abajo

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.

Parte 16

Enlaces locales

Enlaces HTML con otras páginas del mismo sitio web.

Por **Rubén Álvarez**

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>
```

Como ejemplo, he aquí un enlace que apunta al capítulo anterior al final de la página.

Parte 17

Enlaces externos, de correo y hacia archivos

Vemos tres tipos de enlaces. Los dirigidos a otras páginas de otros webs, a direcciones de correo y a ficheros externos.

Por **Miguel Angel Alvarez**

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) empienzan 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:eugim@desarrolloweb.com">eugim@desarrolloweb.com</a>
```

Este enlace se puede ver en funcionamiento aquí: eugim@desarrolloweb.com

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.

eugim@desarrolloweb.com

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

eugim@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.

Tenemos un artículo en desarrolloweb que habla sobre el contacto con el navegante.

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".

Podemos ver un ejemplo de enlace a archivo con su consiguiente ventana de descarga de un archivo.

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>
```

Parte 18

Atributo nofollow en los enlaces

El atributo `rel=nofollow`, de los enlaces, sirve para que los buscadores no continúen reastreando a partir de esos enlaces.

Por **Miguel Angel Alvarez**

Dejemos aquí una pequeña reseña sobre un atributo que se está convirtiendo en esencial en los enlaces y que no habíamos comentado en nuestro manual de HTML, aunque sí habíamos visto algo en nuestro manual de posicionamiento en buscadores.

Nofollow es un atributo de los links HTML (etiqueta `<A>`), que sirve para definir o marcar el comportamiento de los buscadores cuando rastrean nuestra página web en busca de contenidos para indexarlos. Por defecto, cuando un buscador visita una página web va recorriendo todos sus enlaces y va indexando su contenido en su base de datos de páginas. Esto es algo que hacen automática y periódicamente. Ocurre tanto con los enlaces internos a otras páginas de nuestro sitio como con los enlaces externos a otras páginas web. Los motores que recorren las webs de enlace a enlace, se llaman arañas.

Ahora bien, nosotros podemos, a través del atributo `rel="nofollow"`, decirle a un buscador que no continúe indexando los contenidos de la página a la que lleva un enlace en particular. Lo haríamos así:

```
<a href="URL_DEL_ENLACE" rel="nofollow">TEXTO_DEL_ENLACE</a>
```

Un poco de historia sobre nofollow

Para completar estas notas habría que comentar que el atributo fue creado por iniciativa de Google y Blogger en 2005, como una idea para detener el spam de enlaces en sitios web que permiten a los usuarios la participación.

Una práctica común de las personas que pretenden promocionar una página web es realizar comentarios o participaciones en páginas web, en las que insertan enlaces a sus propios sitios. Esto se conoce como spam de enlaces, o con el término de spamdexing.

Con el atributo `rel="nofollow"` se pretenden mitigar los efectos de ese tipo de spam en los sitios como foros, blogs, o cualquier página que permita la interacción con la comunidad de visitantes. Algunas páginas como la Wikipedia lo utilizan en todos los enlaces externos.

Actualmente el atributo `rel=nofollow` está bajo patente, aunque esta está liberada de royalties, lo que quiere decir que se puede utilizar sin limitación y sin tener que pagar nada.

Cómo interpretan los buscadores el Nofollow

El atributo es tenido en cuenta no sólo por el motor de indexación (araña) del buscador Google, sino también por otras arañas de buscadores tan importantes como Yahoo! o MSN. Aunque hay que decir que cada buscador hace un uso particular del atributo.

Habría que aclarar que los buscadores a menudo siguen el enlace que se ha marcado como `nofollow`, visitando aquella web que se había enlazado. Lo que ocurre realmente es que no tienen en cuenta el link para asignarle ranking o directamente no lo indexan.

Promoción de webs y `nofollow`

Sobra decir que este atributo resulta bastante potente a la hora de trabajar con la promoción de una página web, puesto que altera directamente el comportamiento de los buscadores al rastrear las páginas, enlace a enlace.

Así pues, los SEO, promocionadores de páginas o personas que trabajan en el posicionamiento en buscadores, lo tienen muy en cuenta en sus técnicas para su trabajo del día a día. Existen diversas técnicas que utilizan este atributo para realizar promoción de webs, como la que explicamos en el artículo *Arriesgada pero excelente técnica SEO usando `rel=nofollow`*, pero el uso más importante es el que se ha comentado ya en este artículo: ponerlo a los enlaces externos para no transferir ranking o posicionamiento desde nuestra web a otras webs externas.

Otra técnica relacionada con `nofollow`, pero más segura, para que no se indexe parte del contenido de nuestra página, es el uso del archivo `robots.txt`, que ya hemos comentado anteriormente en DesarrolloWeb.com.

Imágenes, formatos gráficos e optimización

Veremos todo lo que los creadores de webs deben conocer sobre las imágenes, no sólo cómo incluir imágenes en las páginas, sino también qué formatos gráficos son adecuados en cada caso y cómo podemos optimizar las imágenes para reducir el tiempo de carga de las webs.

Parte 19

Imágenes en HTML

Vemos cómo colocar una imagen en una página web y algunos atributos básicos para asignarle estilos a las imágenes en HTML.

Por **Rubén Alvarez**

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.

El uso de imágenes también tiene que ser realizado con cuidado porque aumentan el 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. Por ello es recomendable siempre optimizar las imágenes para Internet, haciendo que su tamaño en bytes sea lo mínimo posible, para facilitar la descarga, pero sin que ello comprometa mucho su calidad.

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 Gimp. Tampoco explicaremos las particularidades de cada tipo de archivo: GIF, JPG o PNG y la forma de optimizar nuestras imágenes. Un capítulo posterior al respecto será dedicado a este menester: Formatos gráficos para páginas web.

Las imágenes son almacenadas en forma de archivos, principalmente GIF (para dibujos) o JPG (para fotos). Estos archivos los podemos obtener desde diversas vías, como por ejemplo nuestra cámara digital, aunque también pueden ser creados por nosotros mismos con algún editor gráfico 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, que listamos a continuación:

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 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 píxeles.

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 pérdida 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 píxeles 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.

Truco: Utilizar imagenes 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 pixeles.
- 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>
```

```

```



```
border="0">
```

```
<br>
```

```
<br>
```

```

```

Se puede ver el ejemplo en una página aparte.

Nota: A lo largo de los próximos artículos veremos muchas otras cosas sobre imágenes, pero si lo deseas, también puedes acceder a un vídeo donde se trata lo visto anteriormente y muchas otras cosas adicionales sobre las imágenes y la creación de webs: Videotutorial HTML: imágenes.

Parte 20

Alineación de imágenes con HTML

Explica la manera de alinear la imagen dentro de la página: centrarla, colocarla a la derecha, a la izquierda, etc.

Por **Rubén Alvarez**

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 po-  
niendo texto para que se vea el efecto, Bla 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 bla bla bla...

```
<p>  
Texto tan extenso como  
queramos que cubrirá la parte derecha de la imagen. Sigo pon-  
iendo texto para que se vea el efecto, Bla bla bla bla bla bla  
bla...  
</p>
```

Quedaría así:

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

Si en algún momento deseásemos dejar de rellenar ese espa-
cio 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`:

Texto tan extenso como queramos que cubrirá la parte izquier-
da.

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 ali-
neación vertical de la imagen.

Supongamos que escribimos una línea al lado de nuestra ima-
gen. Esta línea puede quedar por ejemplo arriba, abajo o al me-
dio de la imagen. Asimismo, puede que en una misma línea ten-
gamos 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 de-
cir que, si hay una imagen más alta, ambas imágenes presenta-
ran 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.

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 el próximos capítulos...

Parte 21

Formatos gráficos para páginas web

Presenta los formatos gráficos utilizados en las páginas web, el GIF, el JPG y PNG. Hace hincapié en los dos primeros, resumiendo sus características y enseñando a optimizar los ficheros.

Por **Miguel Angel Alvarez**

El componente gráfico de las páginas web tiene mucha importancia, es el que hace que estas sean vistosas y el que nos permite aplicar nuestra creatividad para hacer del diseño de sitios una tarea agradable. Es también una herramienta para acercar los sitios al mundo donde vivimos, si embargo, es también el causante de errores graves en las páginas y hacer de estas, en algunos casos, un martirio para el visitante.

Las nociones básicas para el uso de archivos gráficos son sencillas, conocerlas, aunque sea ligeramente, nos ayudará a crear sitios agradables y rápidos. No cometer errores en el uso de las imágenes es fundamental, aunque no seas un diseñador y las imágenes que utilices sean feas, utilízalas bien y así estarás haciendo más agradable la visita a tus páginas.

Tipos de archivos

En Internet se utilizan principalmente dos tipos de archivos gráficos GIF y JPG, pensados especialmente para optimizar el tamaño que ocupan en disco, ya que los archivos pequeños se transmiten más rápidamente por la Red.

El formato de archivo GIF se usa para las imágenes que tengan dibujos, mientras que el formato JPG se usa para las fotografías. Los dos comprimen las imágenes para guardarlas. La forma de comprimir la imagen que utiliza cada formato es lo que los hace ideales para unos u otros propósitos.

Adicionalmente, se puede usar un tercer formato gráfico en las páginas web, el PNG. Este formato no tiene tanta aceptación como el GIF o JPG por varias razones, entre las que destacan el desconocimiento del formato por parte de los desarrolladores, que las herramientas habituales para tratar gráficos (como por ejemplo Photoshop) generalmente no lo soportaban y que los navegadores antiguos también tienen problemas para visualizarlas. Sin embargo, el formato se comporta muy bien en cuanto a compresión y calidad del gráfico conseguido, por lo que resulta muy útil en infinidad de casos. Todos estos problemas han pasado y ya sólo Internet Explorer 6 tiene algunos fallos cuando trata con PNG, pero la aceptación actual es más

que suficiente para incorporarlo a nuestras posibilidades reales de trabajo con formatos y optimización de archivos.

A continuación se puede ver una tabla comparativa de las principales características de los formatos gráficos para crear páginas web:

Formato GIF

Un logotipo es un ejemplo claro de imagen GIF

A parte de ser un archivo ideal para las imágenes que estén dibujadas tiene muchas otras características que son importantes y útiles.

Compresión: Es muy buena para dibujos, como ya hemos dicho. Incluso puede ser interesante si la imagen es muy pequeña, aunque sea una foto.

Parte de esta imagen es transparente

Transparencia: es una utilidad para definir ciertas partes del dibujo como transparentes. De este modo podemos colocar las imágenes sobre distintos fondos sin que se vea el cuadrado donde está inscrito el dibujo, viendose en cambio la silueta del dibujo en cuestión.

Para crear un gif transparente debemos utilizar un programa de diseño gráfico, con el podemos indicar qué colores del dibujo queremos que sean transparentes. Generalmente, definimos la transparencia cuando vamos a guardar el gráfico.

Colores: Con este formato gráfico podemos utilizar paletas, conjuntos, de 256 colores o menos. Este es un detalle muy importante, puesto que cuantos menos colores utilicemos en la imagen, por lo general, menos ocupará el archivo. En ocasiones, aunque utilicemos menos colores en un gráfico, este no pierde mucho en calidad, llegando a ser inapreciable a la vista. En algunos programas podemos modificar la cantidad de colores al guardar el archivo, en otros lo hacemos mientras creamos el gráfico.

32 colores

16 8
colores colores

Imagen tomada con distintas paletas de colores.
Se puede apreciar como con pocos colores se ve

bien el gráfico y como pierde un poco a medida que le restamos colores.

Formato JPG

Una fotografía con formato JPG

Veamos ahora cuales son las características fundamentales del formato JPG:

Compresión: Tal como hemos dicho anteriormente, su algoritmo de compresión hace ideal este formato para guardar fotografías. Además, con JPG podemos definir la calidad de la imagen, con calidad baja el fichero ocupará menos, y viceversa.

Intento de transparencia en JPG. Pulsar para ampliar

Transparencia: Este formato no tiene posibilidad de crear áreas transparentes. Si deseamos colocar una imagen con un área que parezca transparente procederemos así: con nuestro programa de diseño gráfico haremos que el fondo de la imagen sea el mismo que el de la página donde queremos colocarla. En muchos casos los fondos de la imagen y la página parecerán el mismo.

Colores: JPG trabaja siempre con 16 millones de colores, ideal para fotografías.

Optimizar ficheros

Para que las imágenes ocupen lo menos posible y se transfieran rápidamente por la Red debemos aprender a optimizar los ficheros gráficos. Para ello debemos hacer lo siguiente:

Para los archivos GIF: Reduciremos el número de colores de nuestra paleta. Esto se hace con nuestro editor gráfico, en muchos casos podremos hacerlo al guardar el archivo.

GIF 256 colores - 10,8 KB

GIF 32 colores - 5,5 KB

GIF 4 colores - 2 KB

Para los archivos JPG: Ajustaremos la calidad del archivo cuando lo estemos guardando. Este formato nos permite bajar mucho la calidad de la imagen sin que esta pierda mucho en su aspecto visual.

JPG
calidad 0
3 KB
JPG
calidad 20
5,9 KB
JPG
calidad 50
10 KB

Photoshop es una herramienta excelente para optimizar ficheros. Viendo varias copias podemos elegir la más adecuada. Es imprescindible disponer para optimizar la imagen de una herramienta buena que nos permita configurar estas características de la imagen con libertad y fácilmente. Photoshop 5.5 o 6 es un programa bastante recomendable, pues incorpora una opción que se llama "Guardar para el Web" con la que podemos definir los colores del gif, calidad del JPG y otras opciones en varias muestras a la vez. Así con todas las opciones configurables, viendo los resultados a la vez que el tamaño del archivo podemos optimizar la imagen de una manera precisa con los resultados que deseamos.

También existen en el mercado otros programas que nos permiten optimizar estas imágenes de manera sorprendente. Una vez hemos creado la imagen la pasamos por estos programas y nos comprimen aun más el archivo, haciéndolo rápido de transferir y, por tanto, más optimo para Internet. Al ser estas utilidades tan especializadas los resultados suelen ser mejores que con los programas de edición gráfica.

Ejemplos de optimizadores gráficos:

- WebGraphics Optimizer
- ProJPG, GIF Imantion

Y con versiones Online:

- GIF Wizard

Nota: Si te interesa reforzar todos los conceptos tratados en este artículo y ver cómo optimizamos nosotros imágenes para la web, te recomendamos ver el Vídeo sobre Formatos gráficos para páginas web.

Parte 22

Mapas de imágenes con HTML

Explicamos detalladamente el proceso para crear mapas de imágenes, osea, imágenes que tienes varios enlaces asociados en distintas áreas.

Por **Rubén Alvarez**

En capítulos anteriores hemos podido adentrarnos en el elemento básico de navegación del web: El enlace hipertexto. Hemos visto que estos enlaces son palabras, textos o imágenes que, al pinchar sobre ellos, nos envían a otras páginas o zonas. Los mapas de imágenes es un nuevo planteamiento de navegación que incorpora una serie de enlaces dentro de una misma imagen. Estos enlaces son definidos por figuras geométricas y funcionan exactamente del mismo modo que los otros enlaces. Podéis ver el funcionamiento de uno en este enlace.

En un principio, estos mapas no eran directamente reconocidos por los navegadores y recurrían a tecnologías de lado del servidor para ser visualizados. Hoy en día pueden ser implementados por medio de código HTML tal y como veremos en este capítulo.

Podemos utilizar estos mapas, por ejemplo, en portadas donde damos a conocer cada una de las secciones del sitio por medio de una imagen. También puede ser muy práctico en mapas geográficos donde cada ciudad, provincia o punto cualquiera representa un enlace a una página.

En cualquier caso, el uso de estos mapas ha de estar sistemáticamente acompañado de un texto explicativo que dé a conocer al usuario la posibilidad de hacer clic sobre los distintos puntos de la imagen. Frases como "Haz clic sobre tal icono para acceder a tal información" resultan muy indicativas a la hora de hacer intuitiva la navegación por los mapas de imágenes. Por otro lado, no esta de más introducir esa misma explicación en el atributo alt de la imagen.

Así pues, un mapa de imagen esta compuesto de dos partes:

- La imagen propiamente dicha que estará situada como de costumbre dentro de la etiqueta `<body>` de nuestro documento HTML.
- Un código, situado en el interior de la etiqueta `<map>`, que delimitara por medio de líneas geométricas imaginarias cada una de las áreas de los enlaces presentados en la imagen.

Las líneas geométricas que delimitan los enlaces, es decir, las áreas de los enlaces, han de ser definidas por medio de coordenadas. Cada imagen es definida por unas dimensiones de ancho (X) y alto (Y) y cada punto de la imagen puede ser definido por tanto diciendo a que altura (x) y anchura (y) nos encontramos. De este modo, la esquina superior izquierda corresponde a la posición 0,0 y la esquina inferior derecha corresponde a las coordenadas X,Y. Si deseamos saber qué coordenadas corresponden a un punto concreto de nuestra imagen, lo mejor es utilizar un programa de diseño grafico como Photoshop o Paint Shop Pro.

La mejor forma de explicar el funcionamiento de este tipo de mapas es a partir de un ejemplo práctico. Supongamos que tenemos una imagen con un mapa como esta:

Pulsa en los círculos para acceder a las secciones!

Dentro de ella queremos introducir un enlace a cada uno de los elementos que la componen. Para ello, definiremos nuestros enlaces como zonas circulares de pequeño tamaño que serán distribuidas a lo largo y ancho de la imagen.

Veamos a continuación el código que utilizaremos:

```
<table border=0 width=450><tr><td align="center">
<map name="mapa1">
<area alt="Pulsa para ver la página de mis amigos" shape="CIRCLE" coords="44,36,29" href="#">
<area alt="Pulsa para ver mi novia" shape="CIRCLE" coords="140,35,31" href="#">
<area alt="Pulsa para conocer a mi Familia" shape="circle" coords="239,37,30" href="#">
<area alt="Pulsa para conocer mi trabajo" shape="CIRCLE" coords="336,36,31" href="#">
</map>

<br>
Pulsa en los círculos para acceder a las secciones!
</td></tr></table>
```

Nota: Los href de las áreas van a #

Este es un ejemplo parcial de utilización de los mapas, faltaría colocar los href con valores reales y no con la #. Cada uno de los enlaces de las áreas -atributo href de la etiqueta <area>- deberían llevar a una página web. El ejemplo quedaría completo si creasemos todas las páginas donde enlazar las áreas y colocasemos los href dirigidos hacia dichas páginas. Como no hemos hecho las páginas "destino" hemos colocado enlaces que no llevan a ningún sitio, que, como puedes ver, se indica con el caracter "#".

Podéis observar, tal y como hemos explicado antes, que nuestro mapa consta de dos partes principales: la imagen y la etiqueta <map> que define las áreas de cada enlace.

Cada área se indica con una etiqueta <area>, que tiene los siguientes atributos:

alt

Para indicar un texto que se mostrará cuando situemos el ratón en el área.

shape

Indica el tipo de área.

coords

Las coordenadas que definen el área. Serán un grupo de valores numéricos distintos dependiendo del tipo de área (shape) que estemos definiendo.

href

Para indicar el destino del enlace correspondiente al área.

En este caso hemos utilizado unas áreas circulares (shape="CIRCLE"), que se definen indicando el centro del círculo -una coordenada (X,Y) y el radio, que es un número entero que se corresponde con el número de pixels desde el centro hasta el borde del círculo.

Tipos de áreas: shape distintas

Existen tres tipos de áreas distintas, suficientes para hacer casi cualquier tipo de figura. En el dibujo que acompaña estas líneas se puede ver una representación de las áreas, que detallamos a continuación.

shape="RECT"

Crea un área rectangular. Para definirla se utilizan las coordenadas de los puntos de la esquina superior izquierda y la

esquina inferior derecha. Tal como están nombradas dichas coordenadas en nuestro dibujo, el área tendría la siguiente etiqueta:

```
<area shape="RECT" coords="X1,Y1,X2,Y2" href="#">  
shape="CIRCLE"
```

Crea un área circular, que se indica con la coordenada del centro del círculo y el radio. A la vista de nuestro dibujo, la etiqueta de un área circular tendría esta forma:

```
<area shape="CIRCLE" coords="X1,Y1,R" href="#">  
shape="POLY"
```

Este tipo de área, poligonal, es la más compleja de todas. Un polígono queda definido indicando todos sus puntos, pero atención, los tenemos que indicar en orden, siguiendo el camino marcado por el perímetro del polígono. A la vista del dibujo y los nombres que hemos dado a los puntos del polígono, la etiqueta `<area>` quedaría de esta forma.

```
<area shape="POLY" coords=" X1,Y1, X2,Y2, X3,Y3, X4,Y4"  
href="#">
```

Tablas en HTML

Las tablas fueron muy importantes en una época para maquetar páginas web. Hoy lo adecuado es utilizarlas sólo para presentar información tabulada, es decir, colocada en una rejilla de filas y columnas. En los siguientes artículos aprenderemos todo sobre las tablas en HTML.



www.feedbooks.com

Food for the mind