Formateando el texto en HTML

En esta sección repasaremos y profundizaremos en algunos conceptos introducidos en la unidad anterior sin dejar de aclarar cómo manejaremos los estilos en nuestra web.

A continuación, trabajaremos en colocar negritas, itálicas, subrayados, subíndices y superíndices.

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.

Pero antes de comenzar cabe hacer una reflexión sobre por qué son interesantes estas etiquetas y se siguen usando, a pesar que están entrando prácticamente en el terreno de CSS, ya que en la práctica están directamente formateando el aspecto de las fuentes. Son importantes porque las etiquetas en si no están para definir un estilo en concreto, sino una función de ciertas palabras dentro de un contenido. Por ejemplo, las negritas quieren decir que algo tiene más fuerza o importancia dentro de un texto y una itálica se puede usar para un texto que citado o algún énfasis particular. En cuanto a subíndices y superíndices todavía es más claro, ya que éstos especifican cosas que tiene que ver con el contenido y no con la presentación.

Negrita

Podemos escribir texto en negrita incluyéndose dentro de las etiquetas B y su cierre (bold). Esta misma tarea es desempeñada por STRONG y su cierre, siendo ambas equivalentes. Nosotros nos inclinamos por la primera por simple razón de esfuerzo.

Escribiendo un código de este tipo:

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

Obtenemos este resultado:

Texto en negrita

Nota: ¿Qué diferencia hay entre B y STRONG? Aunque las dos etiquetas hacen el mismo efecto, tienen una peculiaridad que las hace distintas. La etiqueta B indica negrita, mientras que la etiqueta STRONG indica que se debe escribir con fuerza.

HTML lo interpretan los navegadores según su criterio, es por eso que las páginas se pueden ver de distinta manera en unos navegadores 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 los navegadores habituales 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 b y STRONG se podrá entender ahora. Mientras que B significa simplemente negrita y todos los navegadores la interpretarán como negrita, STRONG es una etiqueta que significa que se tiene que resaltar fuertemente el texto y cada navegador es el responsable de resaltarlo como desee. En la práctica STRONG 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 y su cierre (italic) y otra un poco más larga: EM y su cierre. En la mayoría de las páginas que podrás ver por ahí, te encontrarás con la primera forma sin duda más sencilla para escribir y acordarse.

He aquí un ejemplo de texto en itálica:

Texto en itálica

Que da el siguiente

efecto:

Texto en itálica

Subrayado

El HTML nos propone también para el subrayado el par de etiquetas: 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.

Además, cabe decir que la etiqueta U se ha quedado obsoleta, debido a que es algo que realmente se debe hacer del lado del CSS, al ser básicamente un estilo.

Subíndices y superíndices

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

^y para los superíndices

_y para los subíndices

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 y esto en negrita e itálica

Esto nos arroja un texto como este:

Esto sólo está en negrita *y esto en negrita e itálica*

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

Debemos evitar códigos como el siguiente:

Esto está en negrita e itálica

En favor de códigos con etiquetas correctamente anidadas:

Esto está en negrita e itálica

Esto es muy aconsejable, ya que aunque los navegadores entiendan bien las etiquetas mal anidadas, por dos razones:

Sistemas como XML no son tan permisivos con estos errores y puede que en el futuro nuestras páginas no funcionen correctamente.

A algunos navegadores les cuesta tiempo de procesamiento resolver este tipo de errores, incluso más que mostrar una página, por ende, debemos evitar que una mala codificación interfiera en la fluidez de transmisión de datos del servidor a nuestro dispositivo.

Los colores y HTML

Ahora aprenderás a crear colores en notación RGB con valores en hexadecimal, la manera más habitual de expresar un color en HTML. Explicamos la correcta utilización de los colores en el HTML.

En la composición de webs juegan un papel muy importante los colores. Usar una paleta de colores definida suele ayudar a la consistencia de un diseño y a transmitir ciertas sensaciones al usuario. Como parte de nuestro aprendizaje de HTML tenemos que detenernos a comprender cómo se expresan los colores en el lenguaje.

En HTML se usa una notación específica de especificar un color, compuesta por tres valores "RGB": Red, Green, Blue. Rojo, Verde y Azul. Es decir, que para conseguir un color cualquiera mezclaremos cantidades de cada uno de esos colores. RGB es el modelo usado para la creación de colores de los monitores y televisores, así que es un excelente modo de expresar color en un medio digital como una web.

Los valores RGB en HTML 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.

Para conseguir un color, mezclaremos valores asignando dos dígitos a cada valor RGB. De esta manera: "#RRGGBB"

Como has observado, colocamos también una almohadilla "#" al principio, para indicar que esa cadena es un valor de color en hexadecimal.

Más adelante en el artículo veremos ejemplos en una paleta, con valores en RGB. No obstante ejemplos podrían ser #000000 para el negro, #FFFFFF para el blanco, #660000 sería un rojo oscuro o #FF0000 un rojo brillante.

Actualmente hay otras formas de escribir colores en las que profundizaremos en CSS. Muchas veces se utilizan diferentes estilos de escritura para definir colores, porque alguna notación es más versátil y podemos conseguir más fácilmente cualquier tonalidad deseada o el manejo de la opacidad (transparencia) en un sólo código.

Atributos de color en etiquetas HTML

En HTML existen numerosas etiquetas que soportan atributos de color. Para que tengas una primera referencia, así se cambiaría la fuente para escribir en rojo:

Rojo

Como ves, al Atributo color le damos un valor RGB en formato hexadecimal. El caracter # se coloca al principio de la cadena.

Nota: de nuevo tenemos que advertir sobre la necesidad de expresar todo lo que son estilos mediante CSS. En HTML nos debemos centrar en lo que es escribir el contenido y en CSS en aplicar el estilo. Es de destacar, que el color es más estilo que contenido, así que debería ir en el CSS.

El motivo por el cual toda la etiqueta <font> ha quedado en desuso, es justamente porque solo se utiliza para aplicar un estilo.

En CSS los colores se pueden expresar de la misma manera que en HTML y también abordaremos otras opciones de codificación que mencioné en esta sección.

Combinar otros colores

Al principio puede parecer difícil crear combinaciones de color con valores hexadecimales, pero con la práctica nos iremos acostumbrando y hasta seremos capaces de pensar un color y conseguir de cabeza un valor RGB aproximado. Nos vendrá bien tener en mente la rueda de colores.

Algunos editores como el Geany propuesto, o Visual Studio Code, vienen con “color pickers” integrados para facilitar esta tarea, sin tener que cambiar de programa. La mayoría de los editores puede instalar de manera adicional plugins (complementos) para implementar selectores de color, ya que es una demanda muy habitual de los desarrolladores.

Para obtener una rueda y los códigos de colores te recomendamos que ingreses en el sitio: https://htmlcolorcodes.com/es/ donde, en forma totalmente gratuita y sin registración previa, podrás consultar códigos, tablas y nombres de colores.

Colores seguros

En lo que respecta a los colores, no podemos saber que tipo de pantalla va a tener la persona que nos visita y la resolución de color. Por eso una buena idea es usar aquellos colores considerados seguros, que son los "Safe colors", y corresponden a colores compatibles con todos los sistemas.

Nota: hoy la necesidad de usar colores seguros (aquellos que se verán bien en todos los monitores, independientemente de su paleta de color), no es tan grande como hace años, porque la tecnología ha evolucionado mucho y es raro encontrar un monitor que solo soporte 256 colores. No obstante es un conocimiento que resulta interesante por el hecho de remarcar la naturaleza universal de la web y la necesidad de construir páginas que sean capaces de adaptarse a cada medio donde va a ser consultada. En este curso no vamos a entrar en detalles que tienen que ver con el diseño gráfico, pero si es de tu interés te recomiendo leer sobre el Responsive Web Design y el uso de colores. Igualmente, en nuestro proyecto de página web, abordaremos casos Responsive en webs para pc y celulares.

La forma de conseguir colores seguros es limitando nuestros colores a los que se pueden conseguir utilizando los siguientes valores:

00

33

66

99

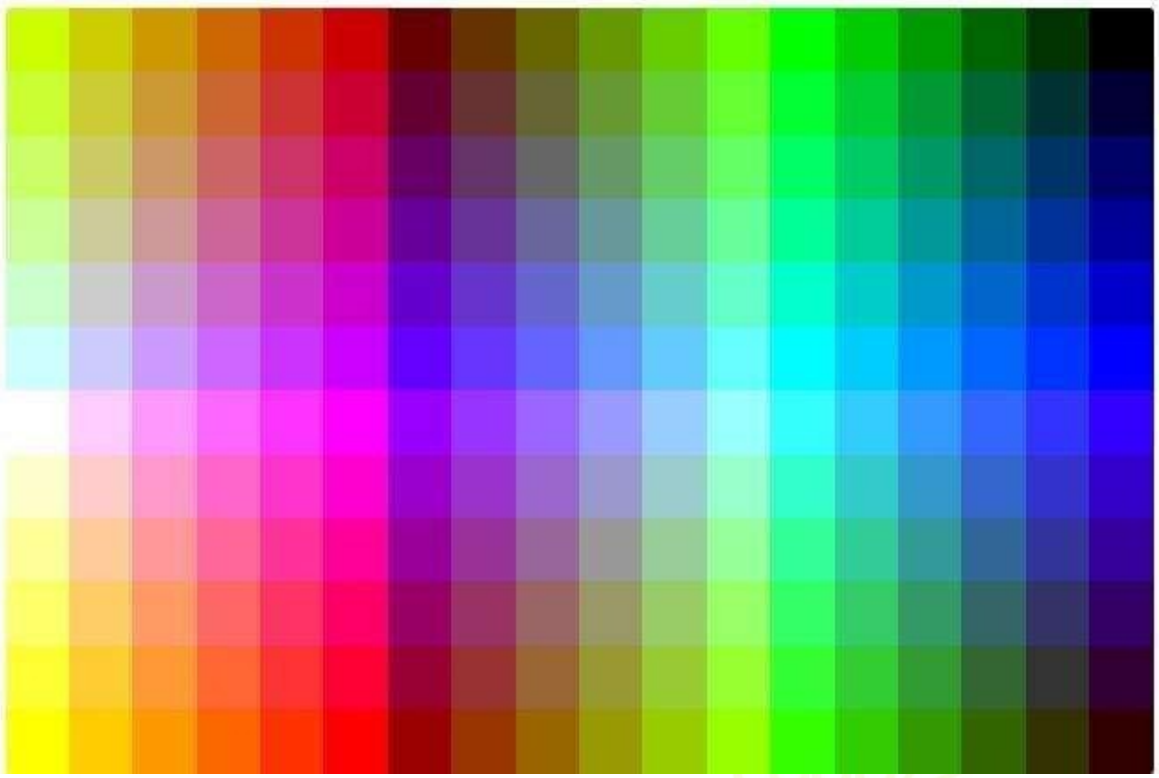
AA

CC

FF

Es interesante comentar que, cuando usamos colores seguros, podemos resumir la notación RGB usando tres caracteres en vez de 6. Por ejemplo, #000 equivale a #000000. O #ABC equivale a #AABBCC.

Usando todas las combinaciones de “safe colors”, conseguimos una paleta de colores variados. La siguiente paleta muestra un cuadro reducido de los safe colors más utilizados:



Atributos para páginas

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

En este artículo nos metemos de nuevo en el terreno del CSS. Veremos todo tipo de estilos que se pueden aplicar a una página, colores o imágenes de fondo, colores para los enlaces, etc. Todo eso tiene que ir en el CSS. Si estás decidido a aprender CSS, puedes saltarte esta parte tranquilamente. Ahora bien, si quieres seguir aprendiendo cosas de HTML y te interesa empezar a poner un poco de color a la página, te recomendamos leer el contenido que figura a continuación.

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. Anteriormente 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. Más adelante veremos cómo se insertan imágenes con HTML y los tipos de imágenes que se pueden utilizar.

Ejemplo: en el video de esta sección vamos a colocar una imagen como fondo en una página.

Te invitamos a practicar a vos también. Para trabajar con una imagen vamos a colocarla en la misma carpeta donde está el HTML que vamos a trabajar. Más adelante también hablaremos sobre cómo acceder a otros archivos que están en otras carpetas, mediante la composición de rutas un poco más complejas que harán más prolijo el ambiente de trabajo, pero por el momento suponemos que la imagen se encuentra en el mismo directorio que la página.

Para colocar esta imagen como fondo de mosaico, se escribiría la siguiente etiqueta BODY.

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

Recomendación: siempre que coloques 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, puede que el texto no contraste lo suficiente con el color de fondo por defecto de la web.

Entonces, supongamos que: la imagen de fondo es oscura, tendremos que poner un texto claro para que se pueda leer.

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

Color del texto

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

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

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

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

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

Ejemplo de color del texto

Si a una página le queremos poner color de fondo negro, y los colores del texto y los enlaces queremos que sean claros. Podemos poner el color de texto blanco y los enlaces amarillos, más resaltados los que no estén visitados y menos resaltados lo que ya están visitados. Para ello podríamos escribir la etiqueta BODY así:

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

Márgenes

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

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 Firefox. (Margen a los lados)

marginheight: igual que topmargin, pero para Firefox. (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 sería:

```
<body topmargin=0 leftmargin=0 marginheight=0 marginwidth=0 bgcolor="ffffff">

<h1>Hola amigos</h1>

<br>

<br>

Gracias por visitarme!

</body>
```

Esta página tiene el fondo blanco.

¿Por qué decimos que todos estos estilos deberían definirse en CSS ?

Como hemos dicho, todos estos estilos deberían indicarse en CSS. Existen muchos motivos para ello pero uno de ellos seguro que ahora se podrá comprender. Imagina un sitio web con 30 páginas distintas (no tiene que ser muy grande para llegar a ese número). Imagina que llegado un día te cansas del color negro de fondo y lo quieres azul, y el color de los enlaces amarillo y lo quieres verde. Si tienes los estilos en el HTML tendrías que ir, página a página, cambiando los estilos 30 veces.

CSS, es un tipo de codificación que entre otras cosas, permite tener los estilos definidos en un único lugar, un archivo con código en texto plano, y todas las páginas de tu sitio web usarán ese mismo archivo para definir su presentación. Así, si un día te cansas del color de fondo, el color del texto, el tipo de letra o su tamaño, entonces solo tienes que ir a un único lugar (el archivo CSS) y cambiarlo una única vez.

Todo eso lo veremos en detalle en nuestras próximas clases de CSS.

Enlaces en HTML y sus distintos tipos

Un sitio web podría ser considerado como un conjunto de archivos, basado en páginas HTML e imágenes, que constituyen el contenido al que el usuario tiene acceso.

Sin embargo, no podríamos hablar de navegación si estos archivos HTML no estuvieran debidamente conectados entre ellos y el exterior de nuestro sitio por medio de enlaces urls.

La particularidad de HTML de relacionar contenidos de los archivos introduciendo referencias bajo forma de enlaces que permitan un acceso rápido a otra información deseada, fue una de las características que lo impulsó vertiginosamente. De poco serviría tener páginas aisladas a las que el usuario no puede acceder o no vincular una información con otra.

Un enlace puede ser fácilmente detectado por el usuario 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.

Sintaxis de un enlace

Para colocar un enlace, nos serviremos de las etiquetas A y su cierre. 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">hacé click aquí</a>
```

Siendo el "hacé click aquí" 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, y también (con algún cambio) puede ser un correo electrónico o un archivo.

Por ejemplo, un enlace a Google, tendría esta manera

```
<a href="https://www.google.com/">Ir a Google</a>
```

Ahora, si queremos que el contenido del enlace sea una imagen y no un texto, podremos colocar la correspondiente etiqueta IMG dentro de la etiqueta A.

```
<a href="https://www.google.com/"><img src="logo.jpg"></a>
```

Nota: no hemos visto aún nada sobre imágenes, aunque lo veremos en breve.

El aspecto de los enlaces

Es posible, mediante HTML, y las hojas de estilo CSS, definir el aspecto que tendrán los enlaces en una página. De manera predeterminada el navegador los destaca para que podamos distinguirlos entre el texto de la página. Generalmente encontraremos los enlaces subrayados y coloreados en azul, aunque esta regla depende del navegador del usuario y de sus estilos definidos como predeterminados.

Ese estilo lo correcto es colocarlo en el código de CSS, pero también se puede definir en la etiqueta BODY. Eso es algo que se explicó anteriormente.

Tipos de enlaces

Para estudiar en profundidad los enlaces tenemos que clasificarlos por su tipo, porque dependiendo ese tipo algunas cosas cambiarán a la hora de construirlos.

En función del destino los enlaces pueden ser agrupados en:

- 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 emails: para crear un mensaje de correo dirigido a una dirección.
- Enlaces con archivos: para que los usuarios puedan hacer download de ficheros.

Sintaxis de los enlaces en la misma página

Crear un enlace que apunte al final de la página. Lo primero será colocar nuestro enlace origen. Este enlace de origen es el que el usuario podrá hacer clic.

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

Como podés 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, a un punto interno dentro de la misma página.

En segundo lugar, hay que generar un enlace en el destino, al que hemos llamado “ancla”. Este enlace no llevará contenido, puesto que no queremos que nadie lo pulse, sino que nos sirva de ancla. Tampoco llevará el atributo "href", porque no tiene que apuntar a ningún lugar, sino que le apuntarán a él. Para poder distinguirlo de otros posibles enlaces realizados dentro de la misma página a cada ancla se le asigna un nombre por medio del atributo "name". En este caso, la etiqueta que escribiremos será ésta:

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

Para entender cómo crear los enlaces internos nos tenemos que fijar en el atributo name="abajo" del ancla. Pues bien, si queremos crear un enlace interno a esta ancla, colocaremos en en enlace de origen el href="#abajo", o sea, el nombre del enlace más un "#" para que el navegador sepa que es un enlace interno.

Cómo construir enlaces en HTML cuyo destino sean otras páginas dentro del mismo sitio web

Como hemos dicho, un sitio web está constituido de páginas interconexas, que se relacionan mediante enlaces de hipertexto. Para abordar el estudio dividimos la materia por los distintos tipos de enlaces que nos podemos encontrar, atendiendo al tipo de destino. Anteriormente vimos cómo enlazar distintas secciones dentro de una misma página.

Ahora nos pondremos con otros tipos de enlaces, a los que hemos llamado "Enlaces locales". Se trata de un tipo de enlace mucho más común en el día a día del desarrollo. De hecho, es el tipo de enlace que más se produce en lo general. Estos enlaces locales nos permiten relacionar distintos documentos HTML que componen un sitio web. Gracias a los enlaces locales podremos convertir varias páginas sueltas en un sitio web completo, compuesto de varios documentos.

Para crear este tipo de enlaces, hemos de usar la misma etiqueta A que ya conocemos, de la siguiente forma:

contenido

Rutas de los enlaces

Hacer un enlace en si no es para nada complejo. No requiere muchas explicaciones con lo que ya conocemos de HTML, sin embargo hay que abordar con detalle un tema importante: las rutas de los enlaces. Como rutas nos referimos al destino del enlace, o sea, lo que ponemos en el atributo "href" y es importante que nos paremos aquí porque nos puede dar algunos problemas al desarrollar, sobre todo para las personas que puedan tener menos experiencia en el trabajo con el ordenador.

Por regla general, para una mejor organización, los sitios suelen estar ordenados por directorios. Estos directorios suelen contener diferentes secciones de la página, imágenes, scripts, estilos y demás. 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 está alojado.

Si la página de destino está en una carpeta o subdirectorio interior al directorio donde está el archivo de origen, hemos de marcar la ruta enumerando cada uno de los directorios por los que pasamos hasta llegar al archivo de destino, separándolos por el símbolo barra "/". Al final, obviamente, escribimos el nombre del archivo destino.

Si la página destino se encuentra en un directorio padre (superior al de la página del enlace), hemos de escribir dos puntos y una barra "../" tantas veces como niveles subamos de carpetas hasta dar con el directorio donde está emplazado el archivo destino.

Enlazar con una página diferente, pero en una sección interna

Los enlaces locales pueden, a su vez, apuntar a la página y a una sección concreta. Este tipo de enlaces resultan ser un híbrido entre interno y local. La sintaxis es de este tipo:

contenido

Como para los enlaces internos, en este caso hemos de marcar la sección con un ancla:

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. Generalmente nuestro sitio web estará en un dominio determinado, tipo example.com. Los enlaces remotos son los que van a páginas que están en otro dominio diferente al nuestro.

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.

Ir a google.com

Sólo cabe destacar que todos las direcciones web (URLs) empiezan por http://, o https:// en el caso que la página de destino se sirva mediante un servidor seguro. Este tipo de rutas que comienzan por "http" también se conocen como "rutas absolutas". Cuando enlazas con páginas que están en otros dominios necesitas usar necesariamente rutas absolutas.

Nota: la parte por la que inician las direcciones de sitios web (http://) nos indica que el protocolo por el que se accede es HTTP, el utilizado en la web. No debemos olvidarnos de colocarlas, porque si no lo hacemos, los enlaces serán tratados como enlaces locales a nuestro sitio.

Otra cuestión 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.Más adelante en el curso hablaremos más en detalle de protocolos y cual es su uso y funcionamiento.

Enlaces a direcciones de correo

Los enlaces a direcciones de correo son aquellos que al seleccionarlos 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 correocolocamos “mailto:” en el atributo href del enlace, seguido de la dirección de correo a la que se debe dirigir el enlace.

```
<a  href="mailto:juan@gmail.com">juan@gmail.com</a>
```

Importante: si un usuario no tiene configurado un programa de correo por defecto en su ordenador no podrá enviar mensajes con esta metodología.

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

```
<a  href="malito:juan@gmail.com?subject=contacto  por  email">juan@gmail.com</a>
```

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

```
<a href="mailto:juan@gmail.com?subject=contacto por
mail&cc=roberto@gmail.com">juan@gmail.com</a>
```

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.

Más adelante veremos el tema Formularios, mediante los cuales seremos capaces de utilizar una serie de campos donde podremos solicitar información de todo tipo, que luego se podría enviar por email.

Enlaces con archivos

Este no es un tipo de enlace propiamente dicho, aunque es de uso muy habitual.

El mecanismo es el mismo que hemos conocido en los enlaces locales y los enlaces remotos, con la particularidad de estar dirigidos hacia un archivo que puede abrirse o descargarse.

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

```
<a href="fichero.zip">Descargá fichero.zip</a>
```

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

Si queremos enlazar hacia otro tipo de archivo como un PDF, podemos hacerlo de la misma manera ya que si el navegador reconoce el tipo de archivo, será el responsable de abrirlo utilizando el conector o programa adecuado para ello. Así, si enlazamos con un PDF, por ejemplo, pondrá el programa Adobe Acrobat Reader en funcionamiento para mostrar los contenidos.

Este sería un ejemplo de enlace a un documento PDF.

```
<a href="documento.pdf">Descarga el PDF aquí</a>
```

Rutas absolutas y relativas

Los link están formados por una serie de rutas (path), donde se indica la dirección a la que tiene que ir el navegador cuando pulsamos sobre ese link o cuando busca una imagen, o un archivo.

Hay dos tipos de direcciones:

- Absolutas: incluye el nombre del do minio. Ejemplo: 'http://i.imgur.com/SpZyc.png'
- Relativa: indica el orden de directorios a partir de donde estoy. Ejemplo: '/imgenes/flor.gif'

Formatos gráficos para páginas web

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, su buen uso también redundará en una carga del sitio más ágil y dinámica.

Tipos de archivos

En Internet se utilizan principalmente los tipos de archivos gráficos PNG, 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. Actualmente, también el uso de formatos SVG, se está incrementando debido a que se caracterizan por su pequeño tamaño, sencillez y nitidez.

Hay varios formatos de imágenes o gráficos vectoriales, el más usado de ellos es el SVG.

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 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.

Formato GIF

Además 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.

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, mostrando 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.

En esta imagen, tomada con distintas paletas de colores, se puede apreciar cómo con pocos colores se ve bien el gráfico y como pierde un poco a medida que le restamos colores.

Formato JPG

Compresión: cuenta con un algoritmo de compresión casi ideal. Este formato, es ideal para fotografías. Además, con JPG podemos definir la calidad de la imagen, con calidad baja el fichero ocupará menos, y viceversa.

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 y PNG: reduciremos el número de colores de nuestra paleta. Esto se hace con nuestro editor gráfico, en muchos casos podremos hacerlo al guardar el archivo.

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.

Es imprescindible disponer para optimizar la imagen de una herramienta buena que nos permita configurar estas características de la imagen con libertad y fácilmente. Hay muchas herramientas de uso libre que incorporan una opción de "Exportar para Web" con la que podemos definir los colores del gif, calidad del JPG y PNG u otras opciones en varias muestras a la vez. Así con todas las opciones disponibles podemos optimizar la imagen de una manera precisa con los resultados que deseamos.

Qué es el formato PNG

El formato PNG (Portable Network Graphics) es un formato de archivos de gráficos de mapa de bits (una trama). Fue desarrollado en 1995 como una alternativa gratuita al formato GIF, cuyos derechos pertenecen a Unisys (propietario del algoritmo de compresión LZW) y a quien los editores de software deben pagar regalías por usar este formato. Por lo tanto, PGN es un acrónimo recursivo de PNG no es GIF.

Características del formato PNG

El formato PNG permite almacenar imágenes en blanco y negro (una profundidad de color de 16 bits por píxel) y en color real (una profundidad de color de 48 bits por píxel), así

como también imágenes indexadas, utilizando una paleta de 256 colores. Además, soporta la transparencia de canal alfa, es decir, la posibilidad de definir 256 niveles de transparencia, mientras que el formato GIF permite que se defina como transparente solo un color de la paleta. También posee una función de entrelazado que permite mostrar la imagen de forma gradual.

La compresión que ofrece este formato es (compresión sin pérdida) de 5 a 25 % mejor que la compresión GIF.

Por último, PNG almacena información gama de la imagen, que posibilita una corrección de gama y permite que sea independiente del dispositivo de visualización. Los mecanismos de corrección de errores también están almacenados en el archivo para garantizar la integridad.

Características del formato SVG

SVG es la abreviatura en inglés de Scalable Vector Graphics (Gráficos Vectoriales Escalables).

Es un nuevo estándar usado para la creación y representación de gráficos e imágenes vectoriales en las páginas web y aplicaciones de Internet, aunque también se pueden emplear en la computadora offline.

SVG es un lenguaje gráfico que emplea el formato XML.

Las imágenes en este formato a diferencia de las tradicionales se pueden editar usando editores de texto plano como NotePad++ o hasta con el Bloc de Notas de Windows.

El formato SVG es recomendado por el W3C y es compatible con la mayoría de los navegadores web modernos.

Las imágenes SVG se caracterizan por su pequeño tamaño y por ser imágenes vectoriales.

Son usadas en las páginas cuando se necesita mostrar o representar elementos sencillos que no necesitan de alta resolución como gráficos, diagramas, mapas, logotipos, iconos, texto, etc.

Atributos básicos para imágenes en HTML.

La etiqueta que utilizaremos para insertar una imagen es IMG (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:

```
<img src="logo.png">
```

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. En el código anterior estamos enlazando con un archivo con extensión .png, pero podrá ser otro tipo de archivo como .gif o .jpg.

Aparte de este atributo, indispensable obviamente para la visualización de la imagen, la etiqueta IMG 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. La sintaxis te quedaría de esta manera:

```
<img src="logo.png" alt="Descripción del logo o imagen">
```

Primeramente, sirve para el posicionamiento de la página en buscadores. De los atributos alt el buscador puede extraer palabras clave y le ayuda a entender qué función o contenido tiene la imagen, y por lo tanto la página.

Otra utilidad importante la encontramos en determinadas aplicaciones, usadas por personas con problemas visuales. Navegadores especiales, por ejemplo, no muestran las imágenes y por ende alt ofrece la posibilidad de leerlas. Nunca está de más pensar en crear páginas contemplando accesibilidad para cualquier navegante.

Atributos height y width

Estos atributos definen la altura y anchura respectivamente de la imagen en píxeles. Aunque estas dimensiones forman parte del estilo de la imagen, y por tanto podrían ir en el CSS, todavía puede ser interesante definirlas dentro del HTML.

Todos los archivos gráficos poseen unas dimensiones de ancho y alto. Estas dimensiones pueden obtenerse a partir del propio diseñador gráfico o bien haciendo clic con el botón derecho sobre la imagen, vista desde el explorador de archivos de tu ordenador, para luego elegir “propiedades” o “información de la imagen” sobre el menú que se despliega.

Un ejemplo de etiqueta IMG con sus valores de anchura y altura declarados te quedaría así:

<codoa codo/>

<img src="logo.gif" width="200" height="300">

Cuando las dimensiones de las imágenes han sido proporcionadas, durante el proceso de carga, el navegador reservará el espacio correspondiente a cada imagen creando una maquetación correcta.

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 siempre aconsejable dado que, si lo que pretendemos es aumentar el tamaño, la pérdida de calidad de la imagen será sensible. Inversamente, si deseamos disminuir su tamaño, estaremos usando un archivo más pesado en KB de lo necesario para la imagen que estamos mostrando con lo que aumentamos el tiempo de descarga de nuestro documento innecesariamente.

Imágenes que son enlaces y el atributo border

Ya hablamos sobre que una imagen puede servir de enlace. Vista la estructura de los enlaces en HTML, podemos muy fácilmente adivinar el tipo de código necesario:

<img src="imagen.jpg">

El problema de hacer esto en ciertos navegadores es que se crea un borde en la imagen, del mismo color que el color configurado para los enlaces, lo que suele ser un efecto poco deseado.

Sin embargo, en HTML podemos indicar que una imagen tenga o no borde. Mediante el atributo "border" se define el tamaño en píxeles del cuadro que rodea la imagen. De esta forma podemos encuadrar nuestra imagen si lo deseamos. No es algo que se use mucho, pero resulta particularmente útil cuando deseamos eliminar el borde que aparece cuando la imagen sirve de enlace. En dicho caso tendremos que especificar border="0".

Aunque de cualquier modo, ese borde se puede eliminar igualmente con CSS y será la manera correcta de hacerlo, porque no olvidemos que el borde es un estilo.

Atributos vspace y hspace

Sirven para indicar el espacio libre, en pixeles, que tiene que colocarse entre la imagen y los otros elementos que la rodean, como texto, otras imágenes, etc. Estos atributos forman parte también de la responsabilidad de CSS, así que no sería recomendable usarlos.

Qué es un favicon

Un favicon es la pequeña imagen que se muestra en la pestaña del navegador o en la lista de marcadores (favoritos). En la barra de dirección, el tamaño del favicon es bastante reducido, 16x16 píxeles, pero en otros lugares como los marcadores puede tener un tamaño mayor (24x24, 32x32, 48x48 o 64x64).

El primer navegador en mostrar favicons fue el Internet Explorer 5 (publicado en 1999). Internet Explorer utilizaba el formato ICO de Windows, que permite guardar varias imágenes de distintos tamaños en un único archivo .ico.

Con el paso del tiempo, otros navegadores permitieron que se pudieran utilizar otros formatos de imagen (GIF, PNG, SVG, etc.), pero este es un aspecto que todavía no está completamente resuelto. Por ejemplo, desde hace años los navegadores admiten favicons en formato PNG, pero el formato SVG sólo lo admite actualmente (octubre de 2019) Firefox.

La IANA definió en 2003 el tipo MIME image/vnd.microsoft.icon que define el formato de imagen ico.

En 2005, el W3C publicó unos consejos sobre favicons, en las que se aconsejaba utilizar el valor icon para el atributo rel (en vez del valor shortcut icon que utilizó Internet Explorer hasta su versión 8) e indicar los distintos tipos de imagen mediante el atributo type, como en los ejemplos siguientes:

```
<link rel="icon" type="image/png" href="imagen.png">

<link rel="icon" type="image/gif" href="imagen.gif">

<link  rel="icon"  type="image/vnd.microsoft.icon"  href="imagen.ico">
```

La recomendación HTML5 fue la primera en definir los favicons, añadiendo a los consejos del 2005 el atributo sizes, para indicar los tamaños de las imágenes (para imágenes escalables, como SVG, se puede utilizar el valor any), como en los ejemplos siguientes:

```
<link  rel="icon"  type="image/png"  href="imagen.png"  sizes="64x64">

<link rel="icon" type="image/svg+xml" href="imagen.svg" sizes="any">

<link rel="icon" type="image/vnd.microsoft.icon" href="imagen.ico" sizes="16x16
24x24 36x36 48x48">
```

En vez de utilizar un fichero .ico con varias imágenes, se puede hacer referencia a varias imágenes para que el navegador elija la más conveniente en cada situación:

```
<link rel="icon" type="image/png" href="imagen_16.png" sizes="16x16">
<link rel="icon" type="image/png" href="imagen_32.png" sizes="32x32">
<link rel="icon" type="image/png" href="imagen_64.png" sizes="64x64">
```

Si se incluye un archivo favicon.ico en la carpeta raíz del sitio web, los navegadores utilizan automáticamente esa imagen como favicon de todas las páginas del sitio, sin que sea necesario incluir la etiqueta <link>. La etiqueta <link> es necesaria si se quieren utilizar otros formatos de imagen, guardarlo en otras carpetas o hacer que diferentes páginas hagan referencia a favicons diferentes (dando diferentes valores al atributo href).

Desgraciadamente, en los últimos años, los navegadores de los móviles han complicado esta situación, ya que utilizan imágenes en muchos tamaños diferentes y con etiquetas no normalizadas para distintos fines.

Si se quieren incluir en una web todos los tamaños de imágenes y las etiquetas requeridas por los diferentes navegadores, lo más fácil es utilizar aplicaciones web especializadas en esta tarea (como https://favicomatic.com/ o https://realfavicongenerator.net/) y hacerlo periódicamente para comprobar las novedades en este campo.

Para crear una imagen en formato .ico, se pueden utilizar editores gráficos como IcoFX (sólo para Windows), pero resulta más cómodo utilizar aplicaciones web como https://favicomatic.com/ o https://realfavicongenerator.net/, que crean todos los archivos necesarios a partir de una imagen. Como mínimo se aconseja incluir dos tamaños imágenes, 16x16 y 32x32, pero se pueden incluir también los tamaños 24x24, 48x48 y 64x64 (aunque eso aumenta bastante el tamaño del archivo .ico).

Listas en HTML

Las posibilidades que nos ofrece 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, enumerar 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, al aplicar estilos CSS a las listas.

Podemos distinguir tres tipos de listas HTML:

- Listas desordenadas
- Listas ordenadas
- Listas de definición

Listas desordenadas

Son delimitadas por las etiquetas UL y su cierre (Unordered List). Cada uno de los elementos de la lista es citado por medio de una etiqueta LI (LI tiene su cierre, aunque si no lo colocas el navegador al ver el siguiente LI interpretará que estás cerrando el anterior). Un ejemplo es:

```
<p>Lenguajes de programación</p>
<ul>
      <li>PHP</li>
      <li>JAVA</li>
      <li>Python</li>
</ul>
```

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 UL, si queremos que el estilo sea válido para toda la lista; o dentro de la etiqueta LI 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 círculo negro.

En caso de que no funcione siempre podemos construir la lista a mano con la viñeta que queramos utilizando CSS.

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

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

Listas ordenadas

Las listas ordenadas sirven también para presentar información, en diversos elementos o ítems, 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 OL (Ordered List) y su cierre. Cada elemento será igualmente indicado por la etiqueta LI, que ya vimos en las listas desordenadas.

Pongamos un ejemplo:

```
<p>Reglas de convivencia</p>
<ol>
    <li>No hacer ruidos molestos</li>
    <li>Ser amable con los vecinos</li>
</ol>
```

Saldría en pantalla:

Reglas de convivencia

1. No hacer ruidos molestos
2. Ser amable con los vecinos

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 minúsculas (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 estará situado dentro de la etiqueta OL. Los valores que puede tomar el atributo en este caso son:

	números (1, 2, 3...)
<ol type="A">	mayúsculas (A, B, C,...)
<ol type="a">	minúsculas (a, b, c...)
<ol type="I">	mayúsculas (I, II, III,...)
<ol type="i">	minúsculas (i, ii, iii,...)

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 ello podemos utilizar un segundo atributo, llamado 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.

```
<p>Reglas de convivencia</p>
<ol start="2" type="A">
    <li>No hacer ruidos molestos</li>
    <li>Ser amable con los vecinos</li>
</ol>
```

Nos arrojaría en pantalla:

Reglas de convivencia

B. No hacer ruidos molestos

C. Ser amable con los vecinos

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 su cierre (Definition List). La etiquetas del elemento y su definición son DT (Definition Term) y DD (Definition Definition) respectivamente.

Aquí un código de ejemplo y su resultado:

```
<p>Alimentos Lácteos</p>
<dl>
    <dt>Leche</dt>
        <dd>descremada</dd>
        <dd>entera</dd>
    <dt>Queso</dt>
        <dd>Fontina</dd>
        <dd>Muzzarella</dd>
</dl>
```

El resultado obtenido:

Alimentos Lácteos

Leche

descremada

entera

Queso

Fontina

Muzzarella

Veremos que cada línea DD se ubica más hacia la izquierda. Este tipo de etiquetas son usadas a menudo con el propósito de crear textos más o menos desplazados hacia la izquierda.

Anidando listas

Podemos anidar estas etiquetas obteniendo listas de la manera deseada, como por ejemplo:

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

obteniendo:

Ciudades del mundo

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

Tablas en HTML

Una tabla es un conjunto de celdas dentro de las cuales podemos incorporar contenidos.

HTML dispone de una gran variedad de etiquetas para crear tablas, con sus atributos, de las cuales veremos una introducción en este artículo.

En general, se utilizan para representar información tabulada, en filas y columnas. Esto es una realidad en los últimos años, desde que las tablas se han descartado para fines relacionados con la maquetación.

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

Hablar hoy de tablas como solución para la maquetación ha pasado a la historia. Las webs actuales han acabado con técnicas que incrementan el tamaño del código fuente de las páginas web, mezclando presentación y contenido. Actualmente toda la maquetación de una página se organiza con CSS, lo que nos da un mayor control de todos los elementos de la página y la posibilidad de separar todos los estilos para definir el aspecto de una web en un fichero aparte del HTML.

Por ello, las tablas actualmente se utilizan mucho menos que en el pasado y realmente la recomendación es usarlas solo en los casos en los que necesitemos incluir en una página información tabulada, es decir, dispuesta en filas y columnas. Todo uso basado en tablas para procurar colocar elementos en determinadas posiciones de la página sería incorrecto en las técnicas actuales de diseño de páginas web.

Como veremos, existen diversas etiquetas que se deben utilizar en una forma determinada para la creación de tablas. Por ello, puede que en un principio nos resulte un poco complicado trabajar con estas estructuras. Si deseamos mostrar datos de una manera sencilla de leer, dispuestos en filas y columnas, tarde o temprano observaremos que las tablas son una rápida solución y vamos a apreciar las posibilidades que nos ofrecen. Por otro lado, las tablas han dejado de utilizarse debido a su falta de correlación con el desarrollo Responsive, entre otras características.

Etiquetas básicas para tablas en HTML

Las tablas son definidas por las etiquetas TABLE y su cierre.

Dentro de estas dos etiquetas colocaremos todas las otras etiquetas de las tablas, hasta llegar a las celdas. Dentro de las celdas ya es permitido colocar textos e imágenes que darán el contenido a la tabla.

Las tablas son descritas por líneas de arriba a abajo (y luego por columnas de izquierda a derecha). Cada una de estas líneas, llamada fila, es definida por otra etiqueta y su cierre: TR (Table Register).

Asimismo, dentro de cada línea, habrá diferentes celdas. Cada una de estas celdas será definida por otra etiqueta: TD (Table Data). Dentro de ésta y su cierre será donde coloquemos nuestro contenido, el contenido de cada celda.

Ejemplo de estructura de tabla:

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

El resultado:

Celda 1, linea 1	Celda 2, linea 1
Celda 1, linea 2	Celda 2, linea 2

También es parte de una tabla la etiqueta TH (Table Header), que sirve para crear una celda cuyo contenido posea un título o cabecera de la tabla.

Ejemplo:

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

Atributos para tablas, filas y celdas

A partir de esta idea simple y sencilla, las tablas adquieren otra magnitud cuando les incorporamos toda una lista de atributos aplicados sobre cada tipo de etiquetas que las componen.

- cellspacing: es el espacio entre celdas de la tabla.
- cellpadding: es el espacio entre el borde de la celda y su contenido.
- border: es el número de píxeles que tendrá el borde de la tabla.
- bordercolor: es el color a asignar al borde de la tabla.

Podemos usar prácticamente cualquier tipo de etiqueta dentro de la etiqueta TD para, de esta forma, escribir su contenido.

Las etiquetas situadas en el interior de la celda no modifican el resto del documento.

Las etiquetas de fuera de la celda no son tenidas en cuenta por ésta.

Podemos especificar el formato de nuestras celdas a partir de etiquetas introducidas en su interior o mediante atributos colocados dentro de la etiqueta de celda TD o bien, dentro de la etiqueta TR, si deseamos que el atributo sea válido para toda la línea.

La forma más útil y actual de dar forma a las celdas es a partir de CSS que veremos más adelante.

A continuación algunos atributos útiles para la construcción de nuestras tablas:

- align: Justifica el texto de la celda del mismo modo que si fuese el de un párrafo.
- valign: Podemos elegir si queremos que el texto aparezca arriba (top), en el centro (middle) o abajo (bottom) de la celda.
- bgcolor: Da color a la celda o línea elegida.
- bordercolor: Define el color del borde.
- background: Nos permite colocar un fondo para la celda a partir de un enlace a una imagen.
- height: Define la altura de la celda en pixels o porcentaje.
- width: Define la anchura de la celda en pixels o porcentaje.
- colspan: Expande una celda horizontalmente.
- rowspan: Expande una celda verticalmente.

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

Ejemplo:

<td width="70">

Define un ancho de 70 pixels a la celda. Sin embargo,

<td width="70%">

define un ancho a la celda del 80% de la anchura del total de la tabla.

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

Así:

<td colspan="2">

Fusiona la celda en cuestión con su vecina

derecha. Del mismo modo,

<td rowspan="2">

Expandirá la celda hacia abajo fusionándose con la celda inferior.

No debes maquetar con tablas

En HTML (antes utilizar CSS) se utilizaban las tablas para maquetar páginas.

Con maquetar nos referimos al proceso por el cual se posicionan contenidos en la pantalla atendiendo a una estructura. Este proceso se conoce como maquetación y a la estructura muchas veces se la conoce como layout.

Con las tablas podemos generar una serie de columnas, espacios como cabecera o pie donde podemos mostrar contenidos estructurados que den la sensación de un diseño bien realizado, dividido en columnas y filas, como la maquetación de una revista o un portal. Sin embargo, usar las tablas NO es una práctica recomendada.

Esta sección sirve para estudiar cómo se hacían las cosas antes y para practicar con HTML, pero hoy ya no se utiliza este tipo de técnicas excepto para los casos ya mencionados.

Actualmente se usa el lenguaje CSS y sus múltiples herramientas para producir un contenido correctamente maquetado.

Etiqueta Iframe

Los frames (en inglés, marcos) son herramientas que han pasado a ser soportadas por todos los navegadores y forman parte de las especificaciones de HTML, para luego retirarse de nuevo del estándar en HTML5.

No obstante, los frames, han permanecido en uso y dentro del estándar con la etiqueta IFRAME que vamos a ver a continuación, que aún hoy tiene mucha utilidad.

En concreto iframe sirve para crear un espacio dentro de la página donde se puede incrustar otra web. Es un cuadrado cuyas dimensiones debe especificar el desarrollador en la propia página, incluidas por los atributos width y height en la propia etiqueta IFRAME.

El iframe tiene asociada una página web, que se carga en el espacio y operará de manera totalmente independiente. Esa página web tendrá sus propios contenidos y estilos. Además será perfectamente funcional: si tiene enlaces se mostrarán en ese mismo espacio y si tiene scripts o aplicaciones dentro se ejecutarán también de manera autónoma en el espacio reservado al iframe.

Iframe se utiliza en muchos contextos. Dentro de un iframe podemos mostrar contenidos de otras páginas, como si estuvieran en la nuestra, por lo que sirven para ejemplos como:

- Visualizar contenidos de terceros, como bloques de noticias o novedades que ofrecen en otras webs.
- Interfaces de usuario, en el que ciertas actividades se realizan de forma autónoma y el procesamiento está en otra página web.
- Incrustar videos de YouTube.
- Incrustar mapas de Google Maps.
- Banners de publicidad desde otro sitio.

Construcción de la etiqueta iframe

Como decimos, el iframe se coloca directamente en el código HTML, en el lugar donde queremos que aparezca.

Se coloca con un código como este:

```
<iframe src="pagina_fuente.html" width=290 height=250>Texto para cuando el  navegador no conoce la etiqueta iframe</iframe>
```

Los atributos principales de iframe son la página web que se va a mostrar en el espacio y el ancho y alto del recuadro que reservemos para el frame flotante.

La etiqueta iframe tiene su correspondiente etiqueta de cierre. Todo el texto que coloquemos entre la etiqueta de inicio y la de cierre es texto alternativo, que sólo se mostrará en caso que el navegador del visitante no acepte la etiqueta iframe.

Atributos de iframe

Atributos disponibles para la etiqueta iframe. No obstante, cabe ya señalar que algunos de los atributos que vamos a ver se engloban más en el terreno de los estilos y por tanto se podrían, y sería más correcto, especificar dentro de las CSS.

src: Para indicar la página web que se mostrará en el espacio del frame flotante.

width: Para definir la anchura del recuadro del iframe.

height: Para definir la altura del iframe.

name: Para especificar el nombre del frame, que podemos utilizar luego para referirnos a él con el target de los links, o mediante javascript.

id: Para indicar el identificador del iframe, y poder referirnos a él desde javascript.

frameborder: para definir si queremos o no que haya un borde en el frame. Los valores posibles son 0 | 1. frameborder=0 indicaría que no queremos borde y frameborder=1 que sí.

scrolling: indica si se quiere que aparezcan barras de desplazamiento para ver los contenidos del iframe completo, en el caso que no aparezcan en el espacio reservado para el iframe. Los valores posibles son: yes | no | auto. El valor "yes" es para que aparezcan siempre las barras de desplazamiento o barras de scroll, "no" sirve para que no aparezcan nunca y "auto" es para que aparezcan sólo cuando son necesarias (es el valor por defecto).

marginwidth: Para definir el margen a izquierda y derecha que debe tener la página que va dentro del iframe, con respecto al borde. Este margen va en pixels, pero prevalecerá el margen que pueda tener asignada la página web que mostremos en el frame flotante.

marginheight: lo mismo que marginwidth, pero en este caso para el tamaño del margen por la parte de arriba y abajo.

margin: para especificar alineación del frame, igual que se especifica para las imágenes.

style y class: los atributos para definir el aspecto del iframe por medio de hojas de estilo css.

Para acabar, aquí vemos otro ejemplo de iframe con unos cuantos atributos más:

```
<iframe name=miframeflotante src="colabora.htm" width=400 height=550 frameborder="0" scrolling=yes marginwidth=2 marginheight=4 align=left>Tu navegador no soporta frames!
</iframe>
```

Es de destacar, que pueden existir páginas web que tengan bloqueados los accesos desde páginas web externas para mostrar un iframe con información de ese sitio.

Elementos en bloque y en línea

HTML clasifica a todos los elementos en dos grupos: inline y block.

De forma predeterminada, los elementos en bloque comienzan en una nueva línea y los elementos en línea pueden comenzar en cualquier parte de una línea.

- BLOCK: , <p>, <h1>..<h6>, <ul>, <ol>, <li>, <table>, <form>, etc.
- INLINE: <a>, <img>, , <b>, , <mark>, <sub>, etc.

Formularios en HTML

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

Hemos visto anteriormente que podíamos, mediante los enlaces a direcciones de email, contactar directamente con un correo electrónico. Sin embargo, esta opción puede resultar en algunos casos poco versátil, si lo que deseamos es que el navegante nos envíe una información bien precisa y además requiere que el visitante tenga instalado en su ordenador algún correo electrónico en un programa como Outlook Express. Es por ello que el HTML propone otra solución mucho más amplia: los formularios.

Los formularios son cajas de texto y botones que podemos encontrar en muchas páginas web y se utilizan para realizar búsquedas o bien para introducir datos personales o claves de acceso. Los datos que el usuario introduce en estos campos son enviados al correo electrónico del administrador del formulario o bien a un programa que se encarga de procesarlo automáticamente.

Qué se puede hacer con un formulario

Usando HTML podemos enviar el contenido del formulario a un correo electrónico, es decir, construir un formulario con diversos campos y, a la hora pulsar el botón de enviar, generar una ventana de redacción de un email con los datos que el usuario haya escrito en cada uno de esos campos.

A menudo desearemos hacer cosas más complejas con los formularios y para ello tendremos que procesar el formulario mediante un programa.

Entonces, tendremos que emplear otros lenguajes más sofisticados que el propio HTML. En este caso, la solución más sencilla es utilizar programas prediseñados que nos ofrecen un gran número de servidores de alojamiento y que nos permiten almacenar y procesar los datos en forma de archivos u otros formatos. Si tenemos una web alojada en un servidor que no nos propone este tipo de ventajas, siempre es posible recurrir a servidores de terceros que ofrecen este u otro tipo de servicios gratuitos para webs. Por supuesto, existe otra alternativa que es la de aprender lenguajes como JavaScript o PHP que nos permitirán, entre otras cosas, el tratamiento de formularios.

Así pues, en resumen, con HTML podremos construir los formularios, con diversos tipos de campos, como cajas de texto, botones de radio, cajas de selección, menús desplegables, etc. Sin embargo, debe quedar claro que desde HTML no se puede procesar la información, sino que deberemos contar con las herramientas de programación Back-End que también veremos en este curso. Entonces, si deseamos que el formulario se envíe automáticamente o se procese en el servidor para generar otro tipo de respuesta, necesitaremos lenguajes de programación (Python, Java, PHP, etc.). Por el momento, estamos viendo Front-End por lo que nos limitaremos a explicar la creación de formularios y estudiaremos próximamente cómo procesar esa información.

Cómo hacer un formulario en HTML

Los formularios son definidos por medio de las etiquetas FORM y su cierre. Entre estas dos etiquetas colocaremos todos los campos y botones que componen el formulario. Dentro de esta etiqueta FORM debemos especificar algunos atributos:

action: define el tipo de acción a llevar a cabo con el formulario. Existen dos formas de enviar la información a que sea procesada:

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

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

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

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

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

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

method: este atributo se encarga de especificar la forma en la que el formulario es enviado. Los dos valores posibles que puede tomar este atributo son post y get. A efectos prácticos, y hasta tanto no veamos Back-End, utilizaremos el valor post.

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

Si queremos que el formulario se procese automáticamente por un programa, generalmente no utilizaremos este atributo, de modo que tome su valor por defecto, es decir, no debemos incluir enctype dentro de la etiqueta FORM.

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

Elementos de Formularios. Campos de texto

El lenguaje HTML nos propone una gran diversidad de alternativas a la hora de crear nuestros formularios, es decir, una gran variedad de elementos para diferentes propósitos. Estas van desde la clásica caja de texto, hasta la lista de opciones en un menú desplegable, pasando por las cajas de validación, etc.

Etiqueta INPUT para texto corto

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

La etiqueta tendrá la siguiente forma:

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

De este modo expresamos nuestro deseo de crear una caja de texto cuyo contenido será llamado "nombre" (por ejemplo, en el caso de la etiqueta anterior, pero podemos poner distintos nombres a cada uno de los campos de texto que habrán en los formularios).

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

El empleo de estas cajas está fundamentalmente destinado a la toma de datos breves: palabras o conjuntos de palabras de longitud relativamente corta.

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

size: define el tamaño de la caja de texto, en número de caracteres visibles. Si al escribir el usuario llega al final de la caja, el texto que escriba a continuación también podrá caber dentro del campo pero irá desfilando, a medida que se escribe, haciendo desaparecer la parte de texto que queda a la izquierda.

maxlength: indica el tamaño máximo del texto, en número de caracteres, que puede ser escrito en el campo. En caso que el campo de texto tenga definido el atributo maxlength, el navegador no permitirá escribir más caracteres en ese campo que los que hayamos indicado.

Nota: es importante no confundir el atributo maxlength con el atributo size. Mientras size define el tamaño visible de la caja de texto, maxlength indica el tamaño máximo real del texto que se puede escribir. Podemos tener una caja de texto con un tamaño aparente (size) que es menor que el tamaño máximo (maxlength). Lo que ocurrirá en este caso es que, al escribir, si sobrepasamos el espacio marcado por size, el texto irá desfilando dentro de la caja hasta que lleguemos a su tamaño máximo definido por maxlength, momento en el cual nos será imposible continuar escribiendo.

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

```
<form>

<input type="text" name="nombre" value="Juan Perez">

</form>
```

Etiqueta INPUT, modalidad de texto oculto

Hay determinados casos en los que podemos desear esconder el texto escrito en el campo INPUT, por medio asteriscos, de manera que aporte una cierta confidencialidad.

Este tipo de campos son análogos a los de texto, con una sola diferencia: remplazamos el atributo type="text" por type="password":

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

En este caso, al escribir dentro del campo, en lugar de texto se desplegarán asteriscos.

Etiqueta TEXTAREA

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

Este tipo de campos son prácticos cuando el contenido a enviar no es un nombre, teléfono, edad o cualquier otro dato breve, sino más bien, un comentario, opinión, etc. en los que existe la posibilidad que el visitante desee rellenar varias líneas.

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

rows: define el número de líneas del campo de texto.

cols: define el número de columnas del campo de

texto. La etiqueta queda por tanto de esta forma:

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

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

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

Otros elementos de

formulario Listas de opciones

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

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

<select name="estacion">

<option>Primavera</option>

<option>Verano</option>

<option>Otoño</option>

<option>Invierno</option>

</select>

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

size: indica el número de valores mostrados a la vez en la lista. Lo típico es que no se incluya ningún valor en el atributo size, en ese caso tendremos un campo de opciones desplegable, pero si indicamos size aparecerá un campo donde veremos las opciones definidas por size y el resto podrán ser vistos por medio de la barra lateral de desplazamiento.

multiple: permite la selección de varios elementos de la lista. La elección de más de un elemento se hace como con el explorador de Windows, a partir de las teclas Ctrl o Mayúsculas (la flecha hacia arriba, también llamada Shift). Este atributo se expresa sin valor alguno, es decir, no se utiliza con el igual: simplemente se pone para conseguir el efecto, o no se pone si queremos una lista desplegable común.

La **etiqueta OPTION** puede asimismo ser matizada por medio de otros atributos:

selected: del mismo modo que multiple, este atributo no toma ningún valor sino que simplemente indica que la opción que lo presenta está elegida por defecto.

Así, si cambiamos la línea del código anterior:

<option>Otoño</option>

por:

<option selected>Otoño</option>

El resultado será que Otoño se despliegue como la opción inicial.

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

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

De este modo, si el usuario elige primavera, lo que le llegara al programa es una variable llamada estación que tendrá con valor 1:

estacion=1

Botones de radio

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

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

<input type="radio" name="estacion" value="1">Primavera

```
<br>
<input type="radio" name="estacion" value="2">Verano
<br>
<input type="radio" name="estacion" value="3">Otoño
<br>
<input type="radio" name="estacion" value="4">Invierno
```

Nota: hay que fijarse que la etiqueta INPUT type="radio" sólo coloca la casilla pinchable en la página. Los textos que aparecen al lado, así como los saltos de línea los colocamos con el correspondiente texto en el código de la página y las etiquetas HTML que necesitemos.

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

estacion=3

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

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

Cajas de validación

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

```
<input type="checkbox" name="queso">Me gusta el queso.
```

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

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

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

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

Envío, borrado y demás en formularios HTML

Siguiendo con la explicación de todo lo relativo a formularios que estamos ofreciendo, ha llegado el momento de explicar cómo podemos hacer un botón para provocar el envío del formulario, entre otras cosas.

Como podremos imaginarnos, en formularios no solamente habrá elementos o campos donde solicitar información del usuario, sino también habrá que implementar otra serie de funciones. Concretamente, han de permitirnos su envío mediante un botón. También puede resultar práctico poder proponer un botón de borrado o bien acompañar el formulario de datos ocultos que puedan ayudarnos en su procesamiento.

En este capítulo, para terminar la saga de formularios, daremos a conocer los medios de instalar todas estas funciones y acabaremos mostrando un ejemplo de formulario completo.

Botón de envío de formulario (submit)

Para dar por finalizado el proceso de relleno del formulario y hacerlo llegar a su gestor, el navegante ha de enviarlo por medio de un botón previsto a tal efecto. La construcción de un botón se realiza con las etiquetas INPUT ya vistas:

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

Como puede verse, tan solo hemos de especificar que se trata de un botón de envío (type="submit") y hemos de definir el mensaje que queremos que aparezca escrito en el botón por medio del atributo value. Este tipo de campos INPUT, para envío de formularios, a menudo se conocen simplemente como "botones de submit".

Una vez enviada la información, es necesario realizar algo de programación, aparte del propio formulario en HTML, en un lenguaje avanzado, que sea del lado del servidor, como PHP, para su procesamiento.

Botón de borrado (reset)

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

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

A diferencia del botón de envío, indispensable en cualquier formulario, el botón de borrado resulta meramente optativo y no es utilizado frecuentemente. Hay que tener cuidado de no ponerlo muy cerca del botón de envío y de distinguir claramente el uno del otro, para que ningún usuario borre el contenido del formulario que acaba de rellenar por error.

Datos ocultos (campos hidden)

En algunos casos, aparte de los propios datos rellenados por el usuario, puede resultar práctico enviar datos definidos por nosotros mismos que ayuden al programa en su procesamiento del formulario. Este tipo de datos, que no se muestran en la página pero sí pueden ser detectados solicitando el código fuente, no son frecuentemente utilizados por páginas construidas en HTML, son más bien usados por páginas que emplean tecnologías de servidor. He aquí un ejemplo:

```
<input type=hidden name="clave" value="1234">
```

Esta etiqueta, incluida dentro de nuestro formulario, enviará un dato adicional al programa encargado de la gestión del formulario. Podríamos, a partir de este dato, dar a conocer al programa el origen del formulario o algún tipo de acción a llevar a cabo (el usuario actual, por citar un ejemplo).

Botones normales

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

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

El uso más frecuente de un botón es en la programación en el cliente. Utilizando lenguajes como Javascript podemos definir acciones a tomar cuando un visitante pulse botón de una página web, esta acción genera lo que llamamos un "evento click" sobre el el
botón presionado.

Ejemplo completo de formulario

El formulario está construido para que envíe los datos por correo electrónico a un buzón determinado.

El código del formulario se puede ver a continuación. Pero antes de analizarlo te recomendamos construir el formulario por tu propia cuenta para practicar.

```
<form action="mailto:mail@gmail.com"  method="post"  enctype="text/plain">
  Nombre <input type="text" name="nombre" size="30" maxlength="100">
  <br>
  Email <input type="text" name="email" size="25" maxlength="100" value="@">
  <br>
  Población <input type="text" name="poblacion" size="20" maxlength="60">
  <br>
  Sexo
  <br>
  <input type="radio" name="sexo" value="Varon" checked> Hombre
  <br>
  <input type="radio" name="sexo" value="Mujer"> Mujer
  <br>
```

```
    <br>
    Frecuencia de los viajes
    <br>
    <select name="utilizacion">
        <option value="1">Varias veces al dia
        <option value="2">Una vez al dia
        <option value="3">Varias veces a la semana
        <option value="4">varias veces al mes
    </select>
    <br>
    <br>
    Comentarios sobre su satisfacción personal
    <br>
    <textarea  cols="30"  rows="7"  name="comentarios"></textarea>
    <br>
    <br>
    <input type="checkbox" name="recibir_info" checked> Deseo recibir notificación de las novedades en las líneas de autobuses.
    <br>
    <br>
    <input type="submit" value="Enviar formulario">
    <br>
    <br>
    <input type="Reset" value="Borrar todo">
</form>
```