

Capítulo 3

HTML

1. Creación de una página web

Cuando desee crear una página web o un sitio web, sería lógico pensar que es suficiente con encender el ordenador y abrir un editor de texto donde escribir el código.

Desgraciadamente, este método nunca ha resultado ser útil y, además, hace que el tiempo de creación de una página o de un sitio sea dos veces más lento.

Es preferible preparar sobre el papel, a modo de borrador, el diseño de lo que se debería obtener al final, un poco como el realizador que va a utilizar un storyboard sobre el que se reflejan y diseñan algunos de los planos que va a tener que filmar: saber exactamente qué se desea filmar, con qué ángulo, con qué claridad...

A continuación, una página web se construye con palabras propias del lenguaje HTML. Algunas de ellas son obligatorias y las vamos a estudiar en la siguiente sección. Otras son más específicas para lo que sería una visualización u otra dependiendo del borrador que hayamos realizado.

80 _____ HTML5, CSS3 y JavaScript

Para crear su primer sitio web

La estructura de una página web escrita en lenguaje HTML se parece a los cimientos de una casa. Sobre esta estructura reposan los estilos CSS y el código JavaScript. Resulta vital construirla bien. Al mismo tiempo, es evidente que nos olvidaremos de ciertos elementos y obtendremos avisos, y podremos revisar la estructura, cambiarla, siempre en detrimento de una cierta pérdida de tiempo y, en ocasiones, bastante esfuerzo.

■ Observación

Como se ha descrito, si no se realiza una preparación previa y una reflexión sobre el resultado que se desea obtener antes, incluso, de encender el ordenador, el tiempo de trabajo será el doble.

2. Código HTML obligatorio

Una página HTML debe contener, como mínimo, el siguiente código, que podrá dividirse en dos grandes secciones:

- el encabezado de la página: `<head></head>`;
- el cuerpo de la página: `<body></body>`.

```
<!DOCTYPE html>
<html>

  <head>
    <title>El título de mi página</title>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF">
  </head>

  <body>
    Estructura, texto e imágenes...
  </body>

</html>
```

Examinemos con detalle este código, pues nos lo encontraremos **sistemáticamente**.

3. El doctype

La primera etiqueta que se utiliza en una página HTML es la etiqueta `<!DOCTYPE html>`.

Hasta HTML5, por tanto en HTML4 o XHTML, por ejemplo, el doctype era una etiqueta que contenía unos cien caracteres, más compleja, que se respetaba y se comprendía más o menos por parte de los desarrolladores. Actualmente, con HTML5, esta etiqueta se escribe de manera mucho más sencilla. Su contenido permite indicar al navegador que el código que sigue es HTML5.

Por tanto, `<!DOCTYPE html>` será, siempre, la primera cosa que escribiremos en el código de una página web en HTML5.

A continuación, tenemos las etiquetas `<html></html>`, dos etiquetas que contienen toda la página y que indican que a partir de esta línea, `<html>`, se utiliza lenguaje HTML.

Las etiquetas `<head></head>` o encabezado de página contienen la información relativa a la configuración que se envía al navegador **antes** de que muestre cualquier cosa, para prepararse.

Las etiquetas `<body></body>`, para el cuerpo de la página, contienen los elementos que debe mostrar el navegador en la pantalla.

La parte esencial del trabajo de un webmaster se encuentra en la etiqueta BODY, aunque si el HEAD está mal escrito o si se omiten elementos importantes, la página podrá funcionar mal o incluso no funcionar en absoluto.

4. La etiqueta `<head>`

Entre las etiquetas indispensables para el correcto funcionamiento de un sitio web, tenemos las etiquetas `<title></title>` que se encuentran en el encabezado `<head>`.

Ejemplo

```
■ <title>Lista de nuevos productos</title>
```

Esta instrucción, un título, mostrará el texto contenido entre las etiquetas `<title>` y `</title>` en algún lugar del navegador. Puede ser en la parte superior de la ventana del navegador o incluso en una pestaña.

Además de ofrecer a la persona que visita la página una idea de lo que se va a mostrar, la información de la etiqueta `<title>` se utiliza en los motores de búsqueda.

Dicho de otro modo, el contenido de la etiqueta `<title>` del ejemplo anterior informa al internauta y a los motores de búsqueda acerca del contenido de la página.

Si bien un internauta puede contentarse con la palabra "productos" y comprender que se trata de zapatos porque está navegando sobre un sitio web especializado en la venta de zapatos, los motores de búsqueda no son capaces de adivinar de qué tipo de producto se trata. Dicho de otro modo, si un internauta busca la palabra clave "zapatos" en un motor de búsqueda, el resultado de la búsqueda no mostrará, probablemente, la página que tiene como título "Lista de nuevos productos".

Afortunadamente, la palabra "zapatos" estará escrita, probablemente, en el cuerpo de la página, y el motor de búsqueda mostrará este sitio entre sus resultados, a pesar de todo. Pero si el título fuera:

```
■ <title>Lista de nuevos zapatos de temporada.</title>
```

o algo similar, es probable que el motor de búsqueda proporcione este sitio de manera mucho más sencilla.

■ Observación

Dicho de otro modo, utilizar la mayor cantidad de palabras clave en el título de nuestras páginas nos va a permitir mejorar su posicionamiento.

La siguiente etiqueta es la etiqueta `<meta></meta>`.

Esta etiqueta contiene información relativa a la página que va a ayudar al navegador, por ejemplo, a mostrar correctamente la página. Hablamos de metadatos.

Ejemplo

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

En este caso, la etiqueta `<meta>` indica al navegador que el charset (el juego de caracteres utilizados para la visualización) es UTF-8. Dicho de otro modo, esto permite mostrar todo tipo de caracteres. Sin entrar en los detalles de UTF-8, diremos que es el formato más utilizado actualmente en los sitios web, por su simplicidad y su eficacia.

Existen muchas posibilidades a nuestra disposición para la etiqueta `<meta>`, por ejemplo:

```
<meta name='description' content='Descripción de la página actual... '>
```

Este texto `description` se encuentra en ciertos motores de búsqueda. Esta `description` puede utilizarse o no en una página HTML. Esto no cambia en nada el funcionamiento del sitio, pero si se implementa la `description` las herramientas de análisis podrán recuperar información y esto ayudará al sitio a ser más visible sobre Internet.

```
<meta name='keywords' content='lista de palabras clave separadas por una coma '>
```

Aquí, la etiqueta `<meta>` contiene las `keywords` (las palabras clave) e indica al motor de búsqueda los asuntos abordados en la página, permitiendo así realizar una visualización más precisa de los sitios correspondientes a los resultados de la búsqueda del internauta.

Observación

*Existe cierta polémica en lo relativo a la utilidad de `<meta name='keywords' . . .`, pues los motores de búsqueda no siempre las utilizan necesariamente. Por el contrario, todo el texto contenido en el sitio se tiene en cuenta siempre, por lo que **optaremos preferentemente por escribir palabras clave en los enlaces y en el texto.***

4.1 Enlazar una hoja de estilo

Es posible indicar otros elementos, opcionales, en la etiqueta `<head>`.

Por ejemplo, es posible enlazar una hoja de estilo.

```
<link href="../../sitio.css" type="text/css" rel="stylesheet"
media="screen"/>
```

De este modo, se indica al navegador que la página web que está intentando mostrar utiliza estilos que se describen en el archivo `sitio.css` y que estos estilos se definen para la visualización de la página en una pantalla (`screen`: una pantalla, como por ejemplo la del ordenador) en contraposición a estilos creados para una salida por TV o incluso para la impresión (`print`).

4.2 Enlazar un archivo JavaScript

La etiqueta `<head>` contiene, también, el acceso a los scripts, que estarán ubicados en un archivo dedicado a JavaScript.

Por ejemplo:

```
<script src="../../js/utiles.js"></script>
```

Aquí, `src` (la fuente, en inglés *source*) nos indica que el archivo `utiles.js`, almacenado en la carpeta `js`, debe cargarse en el navegador.

Cabe destacar que cuando se crea un sitio web es preciso definir, a menudo, diversas hojas de estilo, así como varios archivos JavaScript.

Observación

*Es preferible (por no decir vital) declarar en la etiqueta `<head>` de su página **primero** los accesos a los estilos CSS y **a continuación** los scripts JavaScript. Esto permite al navegador mostrar mucho más rápidamente la página respecto a otra página en la que se describan los estilos y los scripts de manera desordenada, declarando un script, después un estilo, luego otra vez un script... El orden resulta muy importante para la eficacia del navegador, incluso aunque el funcionamiento sea correcto en ambos casos.*

Un último aspecto a tener en cuenta relativo a la inserción de CSS y JavaScript.

Es posible escribir directamente estilos en la página HTML:

```
<style type='text/css'>
body {
    font-size : 12px;
}
</style>
```

En este caso, utilizaremos la etiqueta `<style>`, que nos va a permitir fijar el tamaño del texto de nuestra página a 12 píxeles.

En el siguiente ejemplo, se incluye código JavaScript directamente en la página utilizando la etiqueta `<script>` e indicando el tipo: `"text/javascript"`.

```
<script type='text/javascript'>
    var nombre = 'Bob';
    var edad  = 72;
</script>
```

Aquí, se crean dos variables, nombre y edad, y se inicializan con un valor inicial.

El hecho de escribir código CSS o JavaScript directamente en una página funciona muy bien. Por el contrario, cuando el código se escribe directamente en la página web, cada vez que se carga la página es preciso enviar este código del servidor hasta el cliente.

El interés de almacenar los estilos en un archivo a parte y los scripts también separados es que el navegador solicita al servidor que le envíe los archivos la primera vez que los necesita, pero no los vuelve a solicitar para otra página en la que ya estén cargados, y los guarda cierto tiempo en lo que denominamos una caché (una carpeta del equipo que contiene todo lo que ha cargado el navegador). De esta manera, el navegador ya no vuelve a solicitar al servidor estos archivos lo cual implica que las páginas se muestren mucho más rápido, pues muchos de los elementos no tienen que volver a cargarse.

5. La etiqueta `<body>`

En el interior de `<body></body>` se utilizan todas las etiquetas que sirven para mostrar algo en la pantalla.

En líneas generales, vamos a encontrar todas las etiquetas que permiten **estructurar la página**, es decir, organizarla creando celdas, en las que será posible incluir un menú del sitio, una zona con imágenes y una leyenda debajo, o bien un bloque para un formulario.

De hecho, la estructura de los sitios se ha diseñado desde hace tiempo con tablas. Se trataba de tablas muy sencillas: bastaba con incluir dentro de cada celda el menú, la lista de artículos propuestos en el sitio, y un anuncio arriba del todo, por ejemplo, etc.

Este método que utiliza las tablas `<table></table>` (consulte el capítulo Representación HTML y CSS - sección Las tablas para obtener más detalle acerca de las tablas) es relativamente simple y puede resultar interesante para mostrar datos, como haríamos con una hoja de cálculo. El problema que encontramos actualmente es que debería mostrarse correctamente, también, en smartphones cuyas pantallas son más pequeñas que las de los ordenadores. No será posible, por tanto, mostrar el sitio de la misma manera. Dicho de otro modo, va a ser necesario organizar la representación para que funcione en cualquier pantalla. Habrá que tener en cuenta, también, el paso del modo apaisado (horizontal) al modo vertical.

Existen etiquetas que son bloques (como un rectángulo) y que pueden tener cualquier tamaño. Pueden mostrarse fácilmente en los lugares deseados dentro de la página, teniendo delante o detrás otro bloque. Estos bloques son muy fáciles de manipular y permiten realizar acciones que las tablas `<table>` no permiten.

Para indicar algunas etiquetas correspondientes a estos bloques, tenemos por ejemplo las etiquetas `<div>`, `<section>`, `<article>`, `<nav>`, `<aside>`, `<header>`, `<footer>`...

Una vez definidas todas las zonas en las que se muestra texto, imágenes, vídeos, información... es posible escribir el contenido.

Antes de construir una página web, debemos hacernos una buena idea del resultado final deseado. La etapa de creación de la estructura, de la organización de los bloques, de las dimensiones y de las posiciones es probablemente la más importante y la que va a requerir una mayor atención, pues todo el resto del sitio se va a basar en esta estructura. Si más adelante es necesario cambiar la representación o la disposición, por uno u otro motivo, en ciertos casos se podrá hacer rápidamente, pero en otros puede que suponga un gran esfuerzo, más o menos como si alguien decidiese cambiar de sitio la bodega: puede que sea posible, pero si se ha pensado desde el momento de la construcción, será mucho más sencillo y rápido de realizar.

Una vez implementada la estructura, "bastará" con completarla con las imágenes y los textos deseados. Para ello, el webmaster tiene acceso a diversas etiquetas que le permiten crear listas no numeradas `` (*Unordered List*: lista no ordenada) u ordenadas `` (*Ordered List*), o incluso otras etiquetas que permiten escribir un título en grande y en negrita `<h1></h1>`.

Existe, también, una etiqueta que permite crear un párrafo `<p>`, y agregaremos a todas ellas estilos para finalizar la representación. Todos estos puntos se verán con más detalle conforme avancemos en el libro.

Si ha sobrevivido a toda esta parte teórica, a continuación llega la parte práctica que le aliviará un poco con algunos ejemplos concretos de representación.

5.1 Métodos y etiquetas para estructurar una página

Una de las etiquetas que resulta interesante saber manipular es la etiqueta `<div>`. Ha sido la primera de su género para, a continuación, ser imitada por las demás etiquetas `<section>`, `<article>`, `<nav>`, `<aside>`, `<header>` y `<footer>` que funcionan como `<div>`, pero que tienen una identidad más marcada que permite saber un poco mejor la naturaleza de su contenido.

Por ejemplo, la etiqueta `<footer>`, como su propio nombre indica, contiene el pie de alguna cosa. Por pie se entiende la zona inferior. Si se trata de una página completa, el `<footer>` será la parte inferior de la página. Si se trata de un artículo, el `<footer>` será el pie del artículo, que podrá contener, por ejemplo, información relativa al autor del artículo.