

**IMPORTANTE:** Para acceder a información adicional y a los listados de ejemplo, visite nuestro sitio web [www.minkbooks.com](http://www.minkbooks.com).

## 1.2 Estructura global

Los documentos HTML se encuentran estrictamente organizados. Cada parte del documento está diferenciada, declarada y determinada por etiquetas específicas. En esta parte del capítulo vamos a ver cómo construir la estructura global de un documento HTML y los nuevos elementos semánticos incorporados en HTML5.

### <!DOCTYPE>

En primer lugar necesitamos indicar el tipo de documento que estamos creando. Esto en HTML5 es extremadamente sencillo:

---

```
<!DOCTYPE html>
```

---

*Listado 1-1. Usando el elemento <doctype>.*

**IMPORTANTE:** Esta línea debe ser la primera línea del archivo, sin espacios o líneas que la precedan. De esta forma, el modo estándar del navegador es activado y las incorporaciones de HTML5 son interpretadas siempre que sea posible, o ignoradas en caso contrario.

**Hágalo usted mismo:** Puede comenzar a copiar el código en su archivo de texto y agregar los próximos a medida que los vamos estudiando.

### <html>

Luego de declarar el tipo de documento, debemos comenzar a construir la estructura HTML. Como siempre, la estructura tipo árbol de este lenguaje tiene su raíz en el elemento <html>. Este elemento envolverá al resto del código:

---

```
<!DOCTYPE html>
<html lang="es">

</html>
```

---

*Listado 1-2. Usando el elemento <html>.*

El atributo `lang` en la etiqueta de apertura `<html>` es el único atributo que necesitamos especificar en HTML5. Este atributo define el idioma humano del contenido del documento que estamos creando, en este caso `es` por español.

**Conceptos básicos:** HTML usa un lenguaje de etiquetas para construir páginas web. Estas etiquetas HTML son palabras clave y atributos rodeados de los signos mayor y menor (por ejemplo, `<html lang="es">`). En este caso, `html` es la palabra clave y `lang` es el atributo con el valor `es`. La mayoría de las etiquetas HTML se utilizan en pares, una etiqueta de apertura y una de cierre, y el contenido se declara entre ellas. En nuestro ejemplo, `<html lang="es">` indica el comienzo del código HTML y `</html>` indica el final. Compare las etiquetas de apertura y cierre y verá que la de cierre se distingue por una barra invertida antes de la palabra clave (por ejemplo, `</html>`). El resto de nuestro código será insertado entre estas dos etiquetas: `<html> ... </html>`.

**IMPORTANTE:** HTML5 es extremadamente flexible en cuanto a la estructura y a los elementos utilizados para construirla. El elemento `<html>` puede ser incluido sin ningún atributo o incluso ignorado completamente. Con el propósito de preservar compatibilidad (y por algunas razones extras que no vale la pena mencionar aquí) le recomendamos que siga algunas reglas básicas. En este libro vamos a enseñarle cómo construir documentos HTML de acuerdo a lo que nosotros consideramos prácticas recomendadas.

Para encontrar otros lenguajes para el atributo `lang` puede visitar el siguiente enlace: [www.w3schools.com/tags/ref\\_language\\_codes.asp](http://www.w3schools.com/tags/ref_language_codes.asp).

### `<head>`

Continuemos construyendo nuestra plantilla. El código HTML insertado entre las etiquetas `<html>` tiene que ser dividido entre dos secciones principales. Al igual que en versiones previas de HTML, la primera sección es la cabecera y la segunda el cuerpo. El siguiente paso, por lo tanto, será crear estas dos secciones en el código usando los elementos `<head>` y `<body>` ya conocidos.

El elemento `<head>` va primero, por supuesto, y al igual que el resto de los elementos estructurales tiene una etiqueta de apertura y una de cierre:

---

```
<!DOCTYPE html>
<html lang="es">
<head>

</head>

</html>
```

---

**Listado 1-3.** Usando el elemento `<head>`.

## El gran libro de HTML5, CSS3 y Javascript

La etiqueta no cambió desde versiones anteriores y su propósito sigue siendo exactamente el mismo. Dentro de las etiquetas `<head>` definiremos el título de nuestra página web, declararemos el set de caracteres correspondiente, proveeremos información general acerca del documento e incorporaremos los archivos externos con estilos, códigos Javascript o incluso imágenes necesarias para generar la página en la pantalla.

Excepto por el título y algunos íconos, el resto de la información incorporada en el documento entre estas etiquetas es invisible para el usuario.

### **`<body>`**

La siguiente gran sección que es parte principal de la organización de un documento HTML es el cuerpo. El cuerpo representa la parte visible de todo documento y es especificado entre etiquetas `<body>`. Estas etiquetas tampoco han cambiado en relación con versiones previas de HTML:

---

```
<!DOCTYPE html>
<html lang="es">
<head>

</head>
<body>

</body>
</html>
```

---

**Listado 1-4.** Usando el elemento `<body>`.

**Conceptos básicos:** Hasta el momento tenemos un código simple pero con una estructura compleja. Esto es porque el código HTML no está formado por un conjunto de instrucciones secuenciales. HTML es un lenguaje de etiquetas, un listado de elementos que usualmente se utilizan en pares y que pueden ser anidados (totalmente contenidos uno dentro del otro). En la primera línea del código del Listado 1-4 tenemos una etiqueta simple con la definición del tipo de documento e inmediatamente después la etiqueta de apertura `<html lang="es">`. Esta etiqueta y la de cierre `</html>` al final del listado están indicando el comienzo del código HTML y su final. Entre las etiquetas `<html>` insertamos otras etiquetas especificando dos importantes partes de la estructura básica: `<head>` para la cabecera y `<body>` para el cuerpo del documento. Estas dos etiquetas también se utilizan en pares. Más adelante en este capítulo veremos que más etiquetas son insertadas entre estas últimas conformando una estructura de árbol con `<html>` como su raíz.

## <meta>

Es momento de construir la cabecera del documento. Algunos cambios e innovaciones fueron incorporados dentro de la cabecera, y uno de ellos es la etiqueta que define el juego de caracteres a utilizar para mostrar el documento. Ésta es una etiqueta **<meta>** que especifica cómo el texto será presentado en pantalla:

---

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">

</head>
<body>

</body>
</html>
```

---

**Listado 1-5.** Usando el elemento **<meta>**.

La innovación de este elemento en HTML5, como en la mayoría de los casos, es solo simplificación. La nueva etiqueta **<meta>** para la definición del tipo de caracteres es más corta y simple. Por supuesto, podemos cambiar el tipo **iso-8859-1** por el necesario para nuestros documentos y agregar otras etiquetas **<meta>** como **description** o **keywords** para definir otros aspectos de la página web, como es mostrado en el siguiente ejemplo:

---

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5, CSS3, Javascript">

</head>
<body>

</body>
</html>
```

---

**Listado 1-6.** Agregando más elementos **<meta>**.

**Conceptos básicos:** Hay varios tipos de etiqueta **<meta>** que pueden ser incluidas para declarar información general sobre el documento, pero esta información no es mostrada en la ventana del navegador, es solo importante para motores de búsqueda y dispositivos que necesitan hacer una vista previa del documento u obtener un sumario de la información que contiene. Como comentamos

## El gran libro de HTML5, CSS3 y Javascript

anteriormente, aparte del título y algunos íconos, la mayoría de la información insertada entre las etiquetas `<head>` no es visible para los usuarios. En el código del Listado 1-6, el atributo `name` dentro de la etiqueta `<meta>` especifica su tipo y `content` declara su valor, pero ninguno de estos valores es mostrado en pantalla. Para aprender más sobre la etiqueta `<meta>`, visite nuestro sitio web y siga los enlaces proporcionados para este capítulo.

En HTML5 no es necesario cerrar etiquetas simples con una barra al final, pero recomendamos utilizarlas por razones de compatibilidad. El anterior código se podría escribir de la siguiente manera:

---

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1" />
  <meta name="description" content="Ejemplo de HTML5" />
  <meta name="keywords" content="HTML5, CSS3, JavaScript" />

</head>
<body>

</body>
</html>
```

---

*Listado 1-7. Cierre de etiquetas simples.*

### `<title>`

La etiqueta `<title>`, como siempre, simplemente especifica el título del documento, y no hay nada nuevo para comentar:

---

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Este texto es el título del documento</title>

</head>
<body>

</body>
</html>
```

---

*Listado 1-8. Usando la etiqueta `<title>`.*

**Conceptos básicos:** El texto entre las etiquetas `<title>` es el título del documento que estamos creando. Normalmente este texto es mostrado en la barra superior de la ventana del navegador.

### `<link>`

Otro importante elemento que va dentro de la cabecera del documento es `<link>`. Este elemento es usado para incorporar estilos, códigos Javascript, imágenes o iconos desde archivos externos. Uno de los usos más comunes para `<link>` es la incorporación de archivos con estilos CSS:

---

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Este texto es el título del documento</title>
  <link rel="stylesheet" href="misestilos.css">
</head>
<body>

</body>
</html>
```

---

*Listado 1-9. Usando el elemento `<link>`.*

En HTML5 ya no se necesita especificar qué tipo de estilos estamos insertando, por lo que el atributo `type` fue eliminado. Solo necesitamos dos atributos para incorporar nuestro archivo de estilos: `rel` y `href`. El atributo `rel` significa “relación” y es acerca de la relación entre el documento y el archivo que estamos incorporando por medio de `href`. En este caso, el atributo `rel` tiene el valor `stylesheet` que le dice al navegador que el archivo `misestilos.css` es un archivo CSS con estilos requeridos para presentar la página en pantalla (en el próximo capítulo estudiaremos cómo utilizar estilos CSS).

**Conceptos básicos:** Un archivo de estilos es un grupo de reglas de formato que ayudarán a cambiar la apariencia de nuestra página web (por ejemplo, el tamaño y color del texto). Sin estas reglas, el texto y cualquier otro elemento HTML sería mostrado en pantalla utilizando los estilos estándar provistos por el navegador. Los estilos son reglas simples que normalmente requieren solo unas pocas líneas de código y pueden ser declarados en el mismo documento. Como veremos más adelante, no es estrictamente necesario obtener esta información de archivos externos pero es una práctica recomendada. Cargar las reglas CSS desde un documento externo (otro archivo) nos permitirá organizar el documento