



```
15 <li><a href="ruta">Página 1- Inicio</a></li>
16 <li><a href="ruta">Página 2- Quiénes somos</a></li>
17 <li><a href="ruta">Página 3- Productos</a></li>
18 </ul>
19 </nav>
20 <section>
21 <article>
22 <h1group>
23 <h1> Etiqueta h1, Título del artículo 1</h1>
24 <h2> Etiqueta h2, Subtítulo del artículo 1</h2>
25 </h1group>
26 <p> Este es el elemento 'p' que representa a un párrafo de
página web. Lorem ipsum dolor sit amet, consectetur adipiscing
incididunt ut labore et dolore magna aliqua. Ut enim ad minim ven
ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aut
in voluptate velit esse cillum dolore eu fugiat nulla pariatur. E
cupidatat non proident, sunt in culpa qui officia deserunt mollit
27 </article>
```

Aprender

HTML5, CSS3 y Javascript con 100 ejercicios prácticos

MEDIAactive



Aprender

HTML5, CSS3

y Javascript

con 100 ejercicios prácticos



Aprender

HTML5, CSS3

y Javascript

con 100 ejercicios prácticos

Aprender HTML5, CSS3 y Javascript con 100 ejercicios prácticos

© 2016 MEDIAactive
Primera edición, 2016

© 2016 MARCOMBO, S.A.

Gran Via de les Corts Catalanes, 594
08007 Barcelona
www.marcombo.com

Diseño de la cubierta: NDENU DISSENY GRÀFIC

«Cualquier forma de reproducción, distribución, comunicación pública o transformación de esta obra sólo puede ser realizada con la autorización de sus titulares, salvo excepción prevista por la ley. Diríjase a CEDRO (Centro Español de Derechos Reprográficos, www.cedro.org) si necesita fotocopiar o escanear algún fragmento de esta obra».

ISBN: 978-84-267-2086-3

D.L.: B-21373-2013
Printed in Spain

Presentación

APRENDER HTML5, CSS3 Y JAVASCRIPT CON 100 EJERCICIOS PRÁCTICOS

100 ejercicios prácticos resueltos que conforman un recorrido por las principales funciones de cada uno de los lenguajes. Si bien es imposible recoger en las páginas de este libro todas las prestaciones de una tríada tan compleja, hemos escogido las más interesantes y utilizadas. Una vez realizados a conciencia los 100 ejercicios que componen este manual, el lector será capaz de manejar crear su propia página web con las herramientas más actualizadas, aplicar estilos complejos y realizar programación básica.

LA FORMA DE APRENDER

Nuestra experiencia en el ámbito de la enseñanza nos ha llevado a diseñar este tipo de manual, en el que cada una de las funciones se ejercita mediante la realización de un ejercicio práctico. Dicho ejercicio se halla explicado paso a paso y pulsación a pulsación, con el fin de no dejar ninguna duda en su proceso de ejecución. Además, lo hemos ilustrado con imágenes descriptivas de los pasos más importantes o de los resultados que deberían obtenerse y con recuadros **IMPORTANTE** que ofrecen información complementaria sobre cada uno de los temas tratados en los ejercicios.

Gracias a este sistema se garantiza que una vez realizados los 100 ejercicios que componen el libro, el usuario será capaz de desenvolverse cómodamente con las funcionalidades básicas de HTML5, CSS3 y Javascript y sacar el máximo partido de sus múltiples prestaciones.

LOS ARCHIVOS NECESARIOS

En el caso de que desee utilizar los archivos de ejemplo de este libro puede descargarlos desde la zona de descargas de la página de Marcombo (www.marcombo.com) y desde la página específica de este manual.

A QUIÉN VA DIRIGIDO EL MANUAL

Si se inicia usted en la práctica y el trabajo con HTML5, CSS3 y

Javascript, encontrará en estas páginas un completo recorrido por sus principales prestaciones. Pero si es usted un experto en estos lenguajes, le resultará también muy útil para consultar determinados aspectos más avanzados o repasar códigos específicos que podrá localizar en el índice.

En general, cada ejercicio está tratado de forma independiente, por lo que no es necesario que los realice por orden (aunque así se lo recomendamos, puesto que hemos intentado agrupar aquellos ejercicios con temática común). De este modo, si necesita realizar una consulta puntual, podrá dirigirse al ejercicio en el que se trata el tema y llevarlo a cabo sobre su propio proyecto web.

HTML5, CSS3 Y JAVASCRIPT

Podemos asegurar con toda seguridad que la triada formada por HTML5, Javascript y CSS3 se consolidará en los próximos años como el estándar para el desarrollo de proyectos web. En este equipo, el HTML5 aporta la estructura y los contenidos de la página web, CSS3 define los estilos o el aspecto que tendrán cada uno de estos elementos y Javascript programa las acciones que realizarán determinados elementos bajo determinadas condiciones.

A lo largo de las lecciones de este manual utilizará las nuevas etiquetas semánticas de HTML5 para crear un completo sitio web con barras de navegación, vídeos, archivos de audio, etc; definirá colores, tipografías, sombras, bordes redondeados, filtros, sencillas animaciones y mucho más utilizando estilos CSS3. Finalmente, mediante los métodos y propiedades Javascript, creará efectos y otros procesos que se ejecutarán gracias a eventos, como el clic del usuario o la carga de la página. Entre otras cosas, aprenderá a programar su propio reproductor de vídeo.

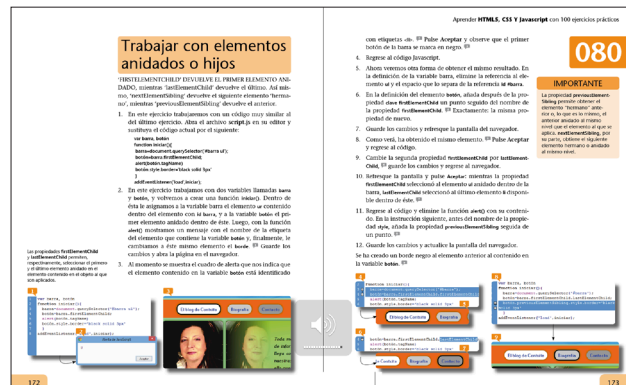
Cómo funcionan los libros "Aprender..."

El título de cada ejercicio expresa sin lugar a dudas en qué consiste éste. De esta forma, si le interesa, puede acceder directamente a la acción que desea aprender o refrescar.

Los ejercicios se han escrito sistemáticamente paso a paso, para que nunca se pierda durante su realización.

El número a la derecha de la página le indica claramente en qué ejercicio se encuentra en todo momento.

Los recuadros Importante incluyen acciones que deben hacerse para asegurarse de que realiza el ejercicio correctamente y también contienen información que es interesante que aprenda porque le facilitarán su trabajo con el programa.



En la parte inferior de todas las páginas puede seguir el ejercicio de forma gráfica y paso a paso. Los números de los pies de foto le remiten a entradas en el cuerpo de texto.

Índice

001	Crear y visualizar un documento HTML.....	14
002	Trabajar con etiquetas y elementos.....	16
003	Declarar idioma y codificación.....	18
004	Declarar el título y otros metadatos	20
005	Declarar el elemento header de la página	22
006	Declarar los elementos aside y 'footer'	24
007	Declarar los elementos nav y ul	26
008	Declarar los elementos section y article	28
009	Cabeceras de artículos con hgroup y header.....	30
010	Elementos image, figure y figcaption.....	32
011	Crear elementos expandibles con details.....	34
012	Marcar fecha con <time> y definición con <dfn>.....	36
013	Etiquetas <cite>, <small>, y <mark>.....	38
014	Usar reversed, data-, ping y download.....	40
015	Contenteditable, spellcheck y draggable.....	42
016	Incrustar un archivo de audio	44
017	Incrustar un archivo de vídeo	46
018	Usar el elemento canvas.....	48
019	Crear un formulario (I).....	50
020	Crear un formulario (II).....	52
021	Definir el campo con atributos	54
022	Conocer el nuevo documento HTML.....	56
023	Declarar estilo CSS dentro del elemento	58
024	Declarar estilos CSS en la cabecera o head	60
025	Crear una hoja de estilos externa	62

026	Crear referencias de clase.....	64
027	Crear referencias de id	66
028	Crear referencias con pseudo-clases (I).....	68
029	Crear referencias con pseudo-clases (II)	70
030	Insertar una imagen de fondo	72
031	Establecer ancho, alto y márgenes	74
032	Ubicar elementos con display y float.....	76
033	Definir y dar estilos a la fuente	78
034	Utilizar fuentes personalizadas.....	80
035	Definir el tamaño de la fuente	82
036	Definir alineación, altura y versalitas.....	84
037	Establecer columnas	86
038	Aplicar sombra, relleno y trazado a una fuente	88
039	Aplicar sombras a una caja	90
040	Crear borde con image-border.....	92
041	Redondear esquinas con border-radius	94
042	Aplicar degradados lineales	96
043	Aplicar degradados radiales	98
044	Usar reflexiones e imágenes degradadas	100
045	Ajustar textos y títulos a sus contenedores	102
046	Transformar la escala de un elemento	104
047	Desplazar un elemento	106
048	Inclinar un elemento.....	108
049	Rotar un elemento.....	110
050	Transformar el origen de un elemento.....	112

Índice

051	3D con la función perspective.....	114
052	Crear efectos con transformaciones	116
053	Crear transiciones entre transformaciones	118
054	Crear animaciones en CSS3.....	120
055	Crear animación con capas	122
056	Crear transparencias	124
057	Crear filtros con CSS3.....	126
058	Modelo de caja flexible	128
059	Elementos flexibles que son cajas flexibles	130
060	Cajas flexibles con medidas de referencia.....	132
061	Enlazar a un punto de la misma página.....	134
062	Conocer Javascript y asignar variables	136
063	Calcular con operadores matemáticos	138
064	Asignar múltiples valores (en una matriz)	140
065	Condicionales con if y else, y switch y case.....	142
066	Crear bucles con for y while.....	144
067	Declarar y llamar a una función.....	146
068	Crear y modificar objetos y propiedades.....	148
069	Crear y modificar métodos.....	150
070	Crear objetos a partir de otros (I)	152
071	Crear objetos a partir de otros (II)	154
072	Conocer los eventos de Javascript.....	156
073	Crear código Javascript en un archivo externo.....	158
074	Seleccionar un elemento del HTML (I)	160
075	Seleccionar un elemento del HTML (II)	162


076	Modificar un contenido con innerHTML	164
077	Intercalar imágenes cambiando valor de src.....	166
078	Modificar estilos CSS3 desde Javascript.....	168
079	Identificar la etiqueta y el elemento padre	170
080	Trabajar con elementos anidados o hijos.....	172
081	Utilizar className para aplicar estilos (I).....	174
082	Utilizar className para aplicar estilos (II)	176
083	Crear el canvas y un contexto de dibujo.....	178
084	Dibujar líneas y rectángulos en canvas	180
085	Definir esquinas y extremos de trazados.....	182
086	Trazar arcos y círculos	184
087	Crear curvas cuadráticas y bezier.....	186
088	Crear degradados lineales y radiales.....	188
089	Escribir en el canvas	190
090	Insertar una imagen en el canvas.....	192
091	Crear patrones con el método 'createPattern'	194
092	Transformar el lienzo	196
093	Crear un reproductor personalizado	198
094	Controlar botones de 'silencio' y 'pausa'	200
095	Controlar la barra de progreso	202
096	Controlar volumen y pantalla completa.....	204
097	Depuración del código o buscando errores.....	206
098	Crear un servidor local con WampServer	208
099	Instalar servidor local y ponerlo a funcionar	210
100	Utilizar su servidor local.....	212

Crear y visualizar un documento HTML

IMPORTANTE

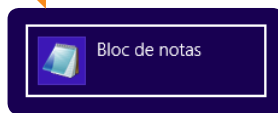
Aunque ya es asumido por casi todos como el estándar para la creación de páginas web, aún no ha sido aceptado oficialmente como tal, entre otras cosas, porque ningún navegador ha implementado todas sus características aún. Sin embargo, todos los navegadores son capaces de entender sus características fundamentales y actualmente no existe una opción capaz de superarlo. Así que manos a la obra.

PUEDA CREAR UN DOCUMENTO HTML en cualquier procesador de texto. En caso de que el procesador trabaje con diversos formatos de texto, deberá asegurarse de crearlo en un archivo de texto sin formato. Es necesario, eso sí, que cambie la extensión del archivo a .html ya sea desde el cuadro Guardar como o desde el Explorador de archivos, una vez haya creado el documento.

1. En este ejercicio aprenderá a crear una estructura básica de HTML, conocerá algunos de sus elementos estructurales fundamentales y, además, aprenderá a crear un documento HTML y visualizarlo en su navegador. Para comenzar, abra la aplicación Bloc de notas  u otro procesador de textos de su preferencia y cree un documento en blanco.
2. En el nuevo documento, introduzca el siguiente código HTML:

```
<!DOCTYPE HTML>
<html lang = "es">
<head>
  <meta charset=utf-8"/>
  <title>Título</title>
</head>
<body>
  <h1>Etiqueta h1, Título principal</h1>
  <h2>Etiqueta h2, Título de segundo nivel</h2>
  <p> Este es el elemento 'p' que representa a un párrafo dentro de la
estructura de la página web. </p>
</body>
</html>
```

1

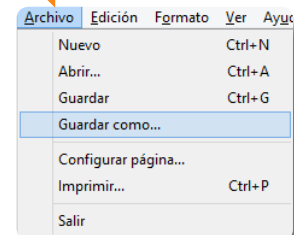


En Windows, Bloc de notas es probablemente el editor de textos más adecuado para la creación de documentos HTML, por lo simple que es.

2



3



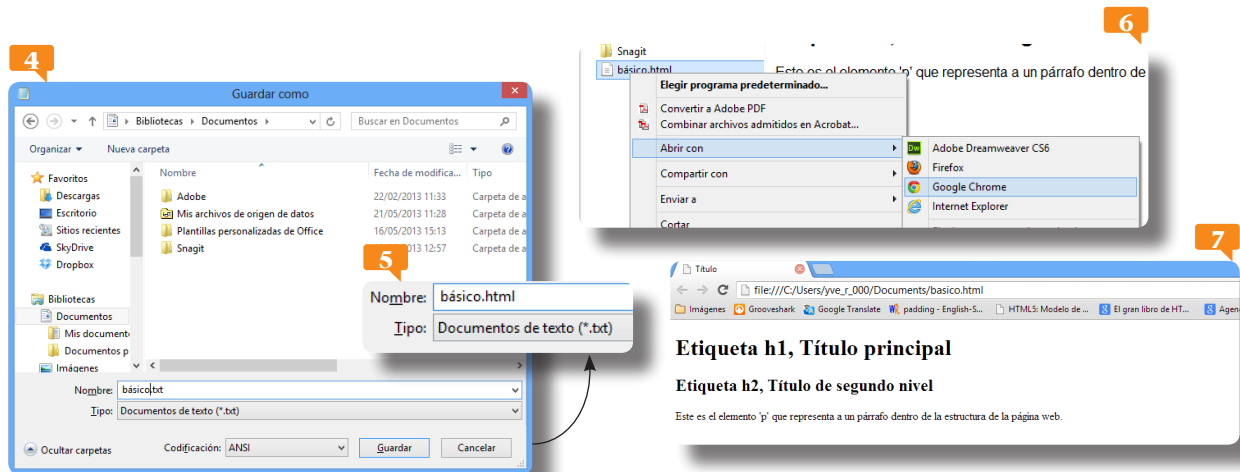
Si pulsa directamente el comando **Guardar** en un archivo que no ha sido guardado antes, normalmente se abre el cuadro **Guardar como**.

3. No se preocupe si no entiende nada del código anterior. En los próximos ejercicios comentaremos cada una de las etiquetas y elementos utilizados en el mismo. De momento, una vez haya introducido el código, despliegue el menú **Archivo** de la aplicación y seleccione la opción **Guardar como**. ³
4. En el cuadro **Guardar como** que se abre, introduzca un nombre para el archivo, por ejemplo **básico**. ⁴
5. Seleccione la extensión propuesta para el archivo, que por defecto es **.txt** y corresponde a un documento de texto, y sustitúyala por **.html**. ⁵
6. Despliegue el menú **Codificación** y, de las opciones que incluye, seleccione **UTF-8**.
7. Finalmente, escoja la ubicación deseada para el archivo y pulse el botón **Guardar**.
8. Una vez que ha guardado el archivo, sólo queda abrirlo en el explorador. Le recomendamos trabajar con la última versión de Google Chrome o de Mozilla Firefox, que son compatibles con la mayor parte de las características de HTML5. Para hacerlo, puede pulsar sobre el nombre del archivo (en el Explorador de archivos) con el botón secundario del ratón y usar la opción **Abrir con** para escoger la aplicación que desea utilizar. Ábralo ahora. ⁶
9. Como por arte de magia, nuestro código se traduce en una sencilla “página web”. ⁷ Tenga en cuenta que también puede realizar todas las pruebas desde un editor de páginas web como Dreamweaver, lo que será más cómodo y práctico en general.

001

IMPORTANTE

Si prefiere trabajar en un editor de páginas web como Dreamweaver para realizar las prácticas de este manual, sólo tiene que crear un nuevo documento con una página en blanco del tipo HTML, establecer HTML5 como tipo de documento. Para visualizar los resultados obtenidos, active la vista **Diseño** en el modo **En vivo**.





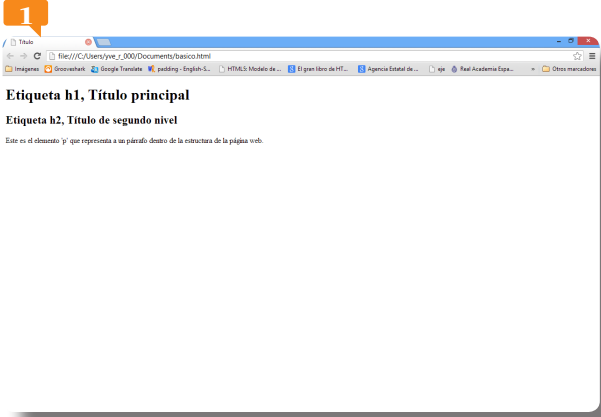
Trabajar con etiquetas y elementos

IMPORTANTE

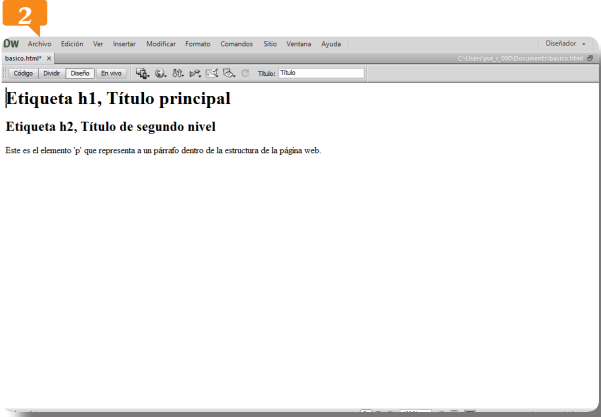
HTML5 es más que la quinta versión del lenguaje HTML. Es la base de una estructura, en la cual CSS aporta la estética y los detalles, y Javascript aporta el funcionamiento de cada elemento.

HTML ES UN LENGUAJE DE MARCADO en el que un conjunto de etiquetas organizan e identifican cada elemento de un código de texto que proporciona la estructura de una página web. Su nombre viene de 'HyperText Markup Language' que significa lenguaje de marcado de hipertexto. Toda página en HTML está dividida en dos elementos básicos que son <head> (la cabecera) y <body> (el cuerpo).

1. En este ejercicio analizaremos el código HTML que presentamos en el ejercicio anterior. Para comenzar, observemos el resultado obtenido en el navegador  o en el editor de páginas web,  según la aplicación que haya usado.
2. Si compara el aspecto de la sencilla página creada con el código que lo ha generado, podrá identificar fácilmente a cada uno de los elementos que contiene. La primera línea del código, <!DOCTYPE HTML>, indica que trabajaremos con el estándar HTML5 y debe usarse siempre en esta ubicación. Debe saber que aunque no es imprescindible, sí que es extremadamente recomendable, pues facilitará al explorador la tarea de identificar el tipo de documento.
3. La segunda línea, <html lang = "es">, es lo que llamamos una etiqueta de apertura. Marca, además del idioma (sobre el que



Este es el aspecto de nuestra página web base en el explorador Google Chrome. Compruebe que dentro de la página en sí sólo podemos ver el contenido del cuerpo.



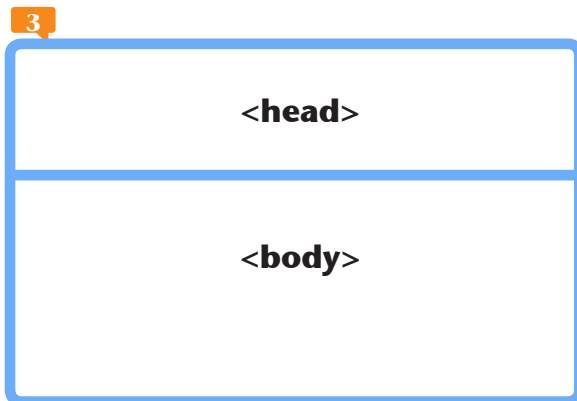
Si crea sus documentos HTML en Dreamweaver, puede ir comprobando el resultado en cualquier momento con sólo un clic en el botón correcto.

002

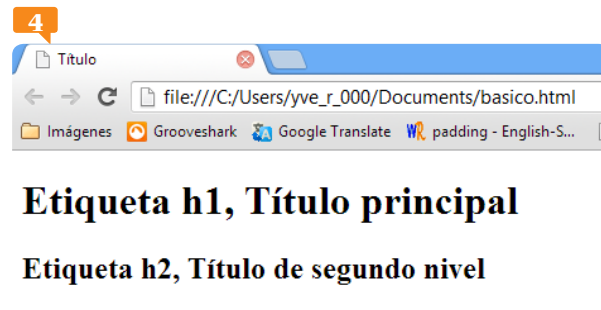
IMPORTANTE

Le recomendamos que cuando cree una página web propia, comience por los cuatro puntos básicos que analizamos en este ejercicio: la declaración `<!DOCTYPE HTML>`, y las etiquetas `<html>`, `<head>` y `<body>`, sin olvidar insertar las etiquetas de cierre de cada una de ellas en el lugar adecuado.

- hablaremos en el próximo capítulo) el inicio del elemento estructural **html**, dentro del cual anidaremos el resto de los elementos de la página web. Así mismo, la última línea del código, `</html>`, indica el cierre del elemento html. De este modo, cada par de etiquetas delimita un elemento; la de cierre se distingue de la de apertura porque incluye una barra inclinada antes de su nombre y, evidentemente, porque se encuentra en segundo lugar.
4. Toda página web (y todo elemento **html**) tiene dos partes fundamentales, que son una cabecera y un cuerpo. La cabecera está identificada por las etiquetas `<head>` mientras que el cuerpo de la página está marcado por las etiquetas `<body>`. ³
 5. Dentro de cada uno de estos elementos, a su vez, anidaremos otros elementos. Tenga en cuenta que una página admite un solo elemento **head** y un solo elemento **body**, y que hay ciertos elementos que pueden estar exclusivamente dentro de alguno de estos dos elementos. Mientras la cabecera o elemento **head** contiene elementos generales de la página como la codificación de caracteres, el título, los iconos que la identifican o los referencias a los archivos externos (como pueden ser los estilos CSS o los códigos Javascript), en el cuerpo o elemento **body** ubicamos los contenidos visibles en la página web.
 6. En nuestro ejemplo hemos colocado dentro de las etiquetas `<head>` la codificación y el título, y dentro de las etiquetas `<body>` un título `<h1>`, un subtítulo o título de segundo nivel `<h2>` y un pequeño párrafo `<p>`. ⁴



El elemento HTML está dividido en dos partes básicas que son la cabecera `<head>` y el cuerpo `<body>`. Las etiquetas están rodeadas por corchetes angulares de apertura y cierre.



El elemento `<title>`, que hemos declarado dentro del elemento `<head>`, identifica el título de la página que se mostrará en la pestaña del navegador y no el que será mostrado como título de los contenidos de la misma. Para los títulos de los contenidos tenemos las etiquetas `<h>`, que analizaremos más adelante.

Declarar idioma y codificación

IMPORTANTE

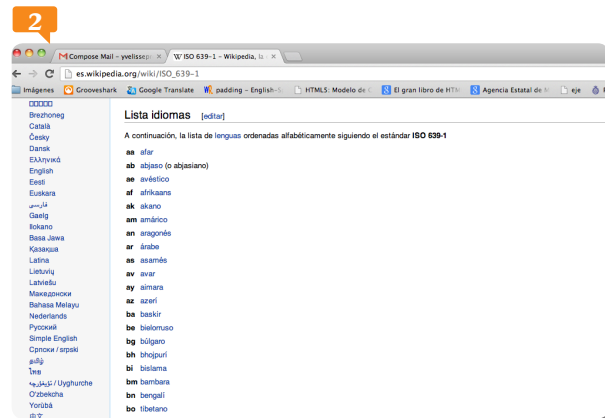
En HTML5 hay gran flexibilidad en cuanto al uso de las etiquetas. Por ejemplo, no es necesario cerrarlas y no hay distinción entre mayúsculas y minúsculas. Sin embargo, cerrar siempre las etiquetas evita errores de lectura e interpretación (al navegador y a cualquier persona que tenga que, por ejemplo, modificar el código) y hacer un uso consistente de mayúsculas y minúsculas ayuda a mantener organizado el código.

DECLARAR EL IDIOMA DE UNA PÁGINA web y el sistema de codificación de caracteres que se utiliza en ella resulta de gran importancia para facilitar que la página sea indexada correctamente por los buscadores y que sea leída correctamente en los navegadores, aunque no es una práctica obligada.

1. En este ejercicio analizaremos dos líneas que declaran el idioma y la codificación de la página web de ejemplo en la que estamos trabajando. En primer lugar, la segunda línea de nuestro primer documento. En ella, y dentro de la misma etiqueta de apertura del elemento `html`, hemos indicado el idioma utilizado en la página: `<html lang = "es">`.
2. Aunque realmente no es obligatorio hacerlo, es una práctica recomendable que ayuda a los motores de búsqueda a identificar los contenidos. En esta línea indicamos que el valor del atributo `lang` (cuyo nombre viene evidentemente de *language*, idioma) del elemento `html` es el español, identificado por el código `es`. Tome nota del uso del signo `=` para asignar el valor.
3. Los códigos que pueden ser utilizados como valores del atributo `lang` o, lo que es lo mismo, que pueden ser utilizados para identificar el idioma de un documento o parte de éste, siempre constan de dos letras y están definidos por la **ISO 639-1**. Puede consultar el que corresponde a distintos idiomas en el artículo sobre esta ISO en la Wikipedia.

```
1 <!DOCTYPE HTML>
2 <html lang = "es">
```

El idioma general del documento debe ser declarado dentro del elemento `html`, en la propia etiqueta de apertura, para que de este modo sea heredado por todos los elementos que se encuentran anidados en su interior.



003

IMPORTANTE

Los atributos nos amplían la información sobre un elemento. Se declaran dentro de la etiqueta de apertura de éste y siguen la siguiente sintaxis: atributo = valor.

4. El atributo **lang** no tiene que ser declarado necesariamente en el elemento **html**, pero lo hacemos así para que sea heredado por todos los elementos anidados en él. Por ejemplo, si su página tiene varios idiomas en diversas secciones, puede declarar el idioma de cada una de ellas por separado.
5. Muy ligada a la declaración del idioma está la declaración de la codificación de caracteres utilizada. En este primer ejemplo la hemos hecho en la línea 4 de nuestro código, `<meta charset=utf-8"/>`; ³ debe tener en cuenta que ésta debe incluirse siempre después de la etiqueta `<head>` de apertura. En este caso, se trata de una etiqueta única, es decir, no tiene una etiqueta de cierre, sino que se utiliza una sola etiqueta, sobre todo porque la etiqueta `<meta>` no tiene más elementos anidados.
6. La barra inclinada que hemos utilizado antes del corchete de cierre tampoco es indispensable en HTML5 pero es preferible utilizarla por razones de compatibilidad y para facilitar la lectura del código, pues deja claro que no habrá una segunda etiqueta.
7. Esta declaración `<meta>` de la que estamos hablando le indica al navegador el tipo de codificación utilizada en la página y asegura, entre otras cosas, la correcta lectura de caracteres propios del castellano, como los acentos y la letra ñ. ⁴
8. Igual que sucedía con la declaración del idioma, no es obligatorio declarar la codificación de caracteres utilizada pero es una práctica definitivamente recomendable para evitar problemas.

```
1
3 <head>
4 <meta charset=utf-8" />
```

La codificación de caracteres utilizada en el documento debe ser declarada dentro del elemento `<head>`, dentro de una etiqueta `<meta charset>`.

4

Etiqueta h1, Título principal

Etiqueta h2, Título de segundo nivel

Este es el elemento 'p' que representa a un párrafo dentro de la estructura de l

Caracteres especiales, como los acentos y la letra ñ, podrían mostrarse alterados si no se utiliza la codificación de caracteres adecuada.

Declarar el título y otros metadatos

IMPORTANTE

En estos primeros ejemplos utilizamos textos meramente enunciativos de las partes de la página que representan. Siéntase libre de ir sustituyéndolos por textos reales de su proyecto web.

EN LÍNEAS GENERALES, EL ENCABEZADO de un documento HTML contiene metadatos sobre el contenido en sí de la página, establece normas para el comportamiento y la presentación del mismo, y define la relación con otros documentos vinculados a aquel que encabeza. Las etiquetas `<title>` y `<meta>` son otros ejemplos.

1. En este ejercicio analizaremos el uso de las etiquetas `<title>` y haremos un acercamiento a los otros elementos que pueden ser incluidos en el elemento **head** de la página. En primer lugar, por lo tanto, nos ocuparemos de la línea 5 de nuestro primer documento de ejemplo: `<title>Título</title>`.
2. En una página web conformada por diversas páginas independientes (a las que se accede mediante un menú o una barra de navegación, por ejemplo), cada una de estas páginas independientes tendrá su propio elemento **title**. Por lo tanto, éste no define el nombre del sitio sino de cada una de las páginas que contiene. Es importante diferenciarlo también del título de los contenidos de la página en sí, pues aunque evidentemente puede coincidir con éste, no será mostrado como parte de los contenidos sino en la cabecera del navegador, o en la pestaña activa en caso de que sea un navegador que trabaje con ventanas y pestañas como es habitual hoy en día.
3. Este título será, normalmente, el resultado más visible que devolverán los motores de búsqueda. Por lo tanto, debe ser oportuno, conciso y claro. En el caso del sitio de una empre-

1
`<title>Título</title>`
`</head>`

El título marcado por las etiquetas `<title>` no requiere el uso de comillas. Si las incluye, serán visibles en el título en cuestión.



El texto marcado con las etiquetas `<title>` se muestra habitualmente en la pestaña activa de la ventana del navegador.



El contenido más visible devuelto por los motores de búsqueda es el marcado por etiquetas **title**.

004

IMPORTANTE

La etiqueta `<meta>` admite como atributo (para declarar el tipo de datos que contiene) a cualquier palabra. Sin embargo, lo recomendable es que ésta esté en inglés pues es el idioma que “hablan” los motores de búsqueda de uso habitual. Otros atributos usualmente manejados para esta etiqueta son: **keywords** (para palabras clave asociadas a la web), **author** (para indicar el autor del sitio), **generator** (para declarar la aplicación utilizada para la creación de la página).

- sa, por ejemplo, el título debe ser el nombre de la empresa y, tal vez, alguna frase que la identifique y resuma su misión, ⁴ aunque ésta no logre mostrarse en la cabecera.
- Además de la etiqueta **meta charset**, hay otras etiquetas `<meta>` que puede utilizar para declarar otros metadatos, es decir, informaciones que no serán mostradas en la pantalla del navegador, sino que serán utilizada por estas aplicaciones, los motores de búsqueda o los dispositivos. Ha de saber que los metadatos de una página web siempre deben ir en el encabezado del documento.
 - Por ejemplo, podemos insertar una etiqueta meta para insertar una descripción de la página, que no será mostrada en pantalla sino consultada por los motores de búsqueda. Una fórmula sería la siguiente: `<meta name="description" content="Este es un ejemplo de un documento html básico"/>`. En este ejemplo damos al atributo **name** de la etiqueta **meta** el valor **description** (que indica en realidad el tipo de metadatos), y al atributo **content** le asignamos como valor el contenido de la descripción en sí, es decir, el texto **Este es un ejemplo de un documento html básico**. Tome nota de las necesarias comillas. Pulse al final de la línea 4 del código de ejemplo, es decir, al final de la declaración de codificación `<meta charset=utf-8" />`, pulse la tecla **Retorno** e introduzca en este punto la etiqueta meta de descripción que hemos citado como ejemplo. ⁵
 - También son utilizadas dentro del elemento **head** las etiquetas `<link>`, que son utilizadas para especificar las hojas de estilo utilizadas en el documento y las etiquetas `<script>`, que indican el documento externo que contiene los códigos Javascript de la página, aunque más adelante trabajaremos con estos dos elementos a fondo. ⁶

4

```
<title>Intermón Oxfam | ONG que trabaja para erradicar la pobreza y la injusticia | </title>
```

El menú del navegador incluye una opción que le permite acceder al código fuente de una página web. Así hemos consultado el código de la web de Intermón Oxfam. En Google Chrome deberá seguir la ruta **Herramientas / Ver código fuente**.

5

```
<meta charset=utf-8" />
<meta name="descripción" content="Este es un ejemplo de un documento html básico" />
```

Los nombres de los atributos de la etiqueta meta deben estar preferiblemente en inglés, que es el idioma usado por los navegadores.

6


```
<link href="styles.css" rel="stylesheet">
<script src="scripts.js"></script>
```

Declarar el elemento header de la página

IMPORTANTE

HTML5 incorpora una serie de nuevos elementos destinados a identificar y organizar las diferentes partes del contenido de la página y declarar sus respectivos objetivos. Son llamados elementos semánticos porque identifican el significado de cada sección y no añaden ninguna funcionalidad ni característica particular. Sin embargo, la correcta organización de los contenidos con el uso adecuado de las etiquetas semánticas asegura la accesibilidad del sitio web y facilita su posterior edición y mantenimiento. Además favorece la posición de la página en los buscadores.

EL CUERPO DEL DOCUMENTO TAMBIÉN SE DIVIDE en diferentes secciones que deben ser diferenciadas claramente con las etiquetas semánticas correctas. La primera de éstas es la etiqueta `<header>` que se utiliza para identificar el documento del web site, de la página o incluso de alguno de sus apartados, según la forma en la que sea utilizado.

1. En este ejercicio comenzaremos a trabajar con los elementos semánticos del cuerpo de un documento HTML, que identifican y otorgan significado a cada una de las partes del contenido. El primer elemento con el que trabajaremos y al que dedicaremos este ejercicio es `<header>`, que identifica la cabecera del sitio web en sí (y no del documento, como sucede con las etiquetas `<head>`). En nuestro ejemplo, las etiquetas `<h1>` podrían estar dentro de un elemento `<header>`, aunque no es algo imprescindible. Podrían estar, incluso, las etiquetas `<h2>` si su contenido fuera un título de segundo nivel que afectara a toda la página.
2. De hecho el elemento `header` puede tener diferentes usos: puede identificar al título del sitio web, o puede identificar los títulos de diferentes secciones. Para este ejercicio, haga clic en su documento de ejemplo al final de la etiqueta `<body>`, pulse la tecla **Retorno** para insertar una nueva línea, e introduzca una etiqueta `<header>` de apertura. 

1

```
<body>
  <header>
    <h1> Etiqueta h1, Título principal</h1>
    <h2> Etiqueta h2, Título de segundo nivel</h2>
```

En el cuerpo del documento se encuentran los contenidos visibles de la página web. Dentro del elemento `<header>` insertaremos los elementos de la cabecera de la página, por ejemplo su título, que debe estar identificado con una etiqueta `<h1>`.

2

```
<body>
  <header>
    <h1> Etiqueta h1, Título principal</h1>
  </header>
```

Las etiquetas de cierre son muy importantes pues indican que ya han sido declarados todos los elementos de la sección que delimitan. Aunque en este ejemplo las etiquetas `<header>` no producen ningún cambio visible, es importante respetarlas y usarlas correctamente.