Manual Imprescindible

CURSO DE DESARROLLO WEB:

HTML, CSS Y JAVASCRIPT

Edición 2021

Mario Rubiales Gómez

ANAYA MULTIMEDIA

ndice de contenidos

- Multiplataforma, disponible de versiones para Windows, Mac y Linux.
- Es totalmente gratuito y se distribuye bajo licencia MIT, su código se encuentra disponible en Github.
- Traducido a múltiples idiomas, incluido el castellano.
- Coloreado y resaltado de la sintaxis.
- Gran cantidad de *plugins* disponibles para ampliar sus funcionalidades.
- URL de descarga: `https://code.visualstudio.com/`.

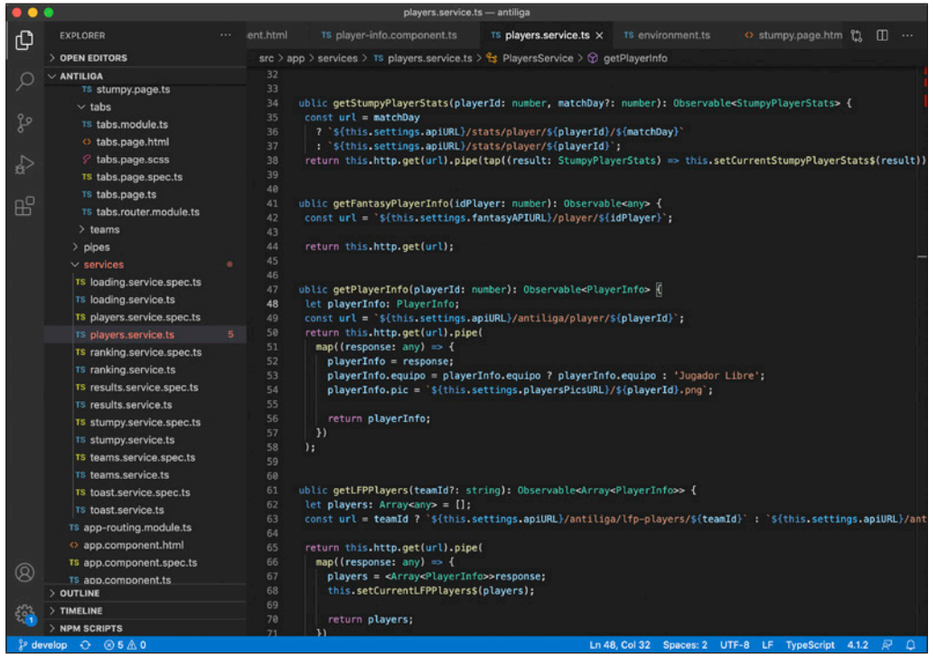

Figura 1.1. Captura de pantalla de Visual Studio Code.

- **UltraEdit:** Todo un clásico en el mundo de la programación; más que un editor web, se trata de un completo entorno de programación con soporte para infinidad de lenguajes de programación. Se caracteriza por:
  - Disponible para plataformas Windows, Mac y Linux.
  - Es un software de pago, existen versiones gratuitas, pero únicamente por un periodo de pruebas de 30 días.

- Disponible en varios idiomas, entre ellos el castellano.
- Coloreado, resaltado y autocompletado del código.
- Cliente FTP integrado.
- Indentación automática de bloques de código.
- Soporte e integración de multitud de plantillas predefinidas.
- Comparador de ficheros.
- URL de descarga: `https://www.ultraedit.com/downloads/ultraedit-download/`.

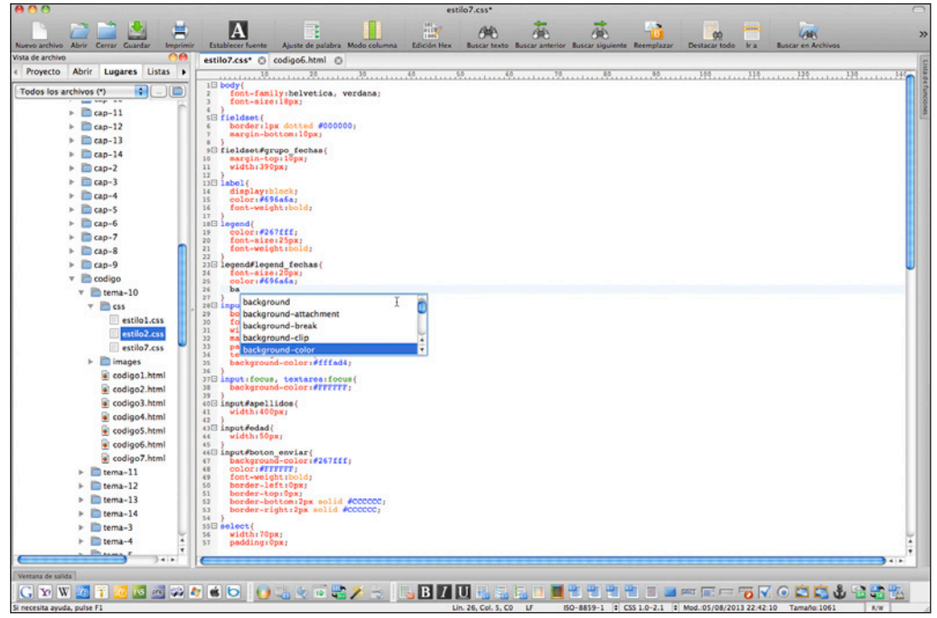

Figura 1.2. Captura de pantalla de UltraEdit.

- **WebStorm:** Entorno de programación creado por la prestigiosa JetBrains y concebido principalmente para JavaScript; sin embargo, tiene soporte para multitud de lenguajes y tecnologías. Es sin duda alguna uno de los entornos de desarrollo con mejor reputación entre la comunidad de programadores. Se caracteriza por:
  - Es una aplicación de pago y accesible previamente mediante una versión de prueba de 30 días.

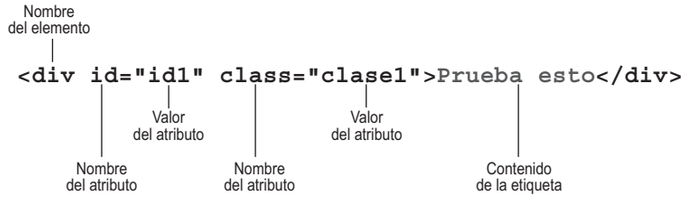

Figura 3.1. Esquema básico de un elemento HTML.

No todos los elementos disponen de etiquetas de inicio y etiquetas de cierre. Hay ciertos tipos de elementos que, al no albergar contenido, no llevan etiqueta de cierre, un ejemplo puede ser el elemento que inserta una nueva línea en un documento: `<br>`.

NOTA:

En las versiones anteriores a HTML5, las etiquetas del tipo `<br>`*, es decir, aquellas que no tienen su etiqueta de cierre, sí que deben incluir una barra invertida o slash antes del símbolo* `>`*, quedando de la siguiente forma:* `<br/>`.

Principales elementos HTML

Actualmente, existen aproximadamente unas 100 etiquetas o elementos definidos en HTML. Es posible que algunas de estas etiquetas no se encuentren disponibles en algunas versiones HTML. A continuación, presentamos una lista detallada con las etiquetas más comunes:

- `<html>`: Esta etiqueta engloba o representa el inicio y el fin de un documento HTML. Salvo alguna excepción que veremos más adelante, cualquier otra etiqueta del documento debe estar definida dentro de la misma.
- `<body>`: Etiqueta que engloba el cuerpo del documento HTML, cualquier contenido o información que vayamos a mostrar en nuestra página web debe estar codificado dentro de esta etiqueta.
- `<head>`: Es la etiqueta que define el inicio y el fin de la cabecera del documento HTML.
- `<meta>`: Emplearemos esta etiqueta para definir las propiedades de nuestro documento HTML. Por norma general, esta etiqueta recoge datos tales como autor, descripción de la página, palabras claves, software o editor con el que se ha generado el documento y tipo de contenido. Los atributos y uso de esta etiqueta son:
  - `name`: Dependiendo del valor de este atributo estaremos describiendo una información u otra: `author` para definir el autor de la página web, `description` para describir el contenido del sitio web, `generator` para establecer el nombre del editor HTML que estamos utilizando y `keywords` para definir las palabras claves que identifican el contenido de la web.
  - `http-equiv`: Maneja información que se envía al servidor en la cabecera `http`.
  - `content`: Recoge el valor de los atributos `name` y `http-equiv`.

Listado 3.2. Ejemplos de uso de la etiqueta `<meta>`.

```
<meta name="author" content="Mario Rubiales">
<meta name="description" content="Curso de HTML">
<meta name="keywords" content="HTML,XHTML,JAVASCRIPT,CSS,WEB">
<meta http-equiv="content-type" content="text/html;charset=utf-8">
```

- `<title>`: Etiqueta que define el título o nombre de nuestra página web. Este nombre es el que suele aparecer en la barra de título de las ventanas. Es obligatorio definir esta etiqueta dentro de `<head>`.

  ```
  <title>Este sería el título de nuestra web</title>
  ```

- `<link>`: Utilizaremos esta etiqueta para definir las rutas donde el documento HTML debe buscar las hojas de estilo externas (CSS) o iconos. Si declaramos esta etiqueta, debe ser siempre dentro de la cabecera. Los principales atributos de esta etiqueta son:
  - `href`: Ruta donde se encuentra el recurso en cuestión.
  - `rel`: Relación entre el documento y el objeto o recurso que estamos vinculando.
  - `type`: Tipo de MIME del objeto enlazado.

  Un ejemplo de uso del elemento `<link>`:

  ```
  <link href="css/estilo.css" rel="stylesheet" type="text/css" >
  ```

- `<script>`: Etiqueta pensada para incluir código JavaScript en nuestra página web. A diferencia de las anteriores, esta etiqueta y su contenido puede ser declarada tanto en la cabecera como en el cuerpo del documento. Sus posibles atributos son:
  - `src`: Especifica la ruta del fichero externo.
  - `type`: Tipo MIME del fichero al que estamos haciendo referencia o bien del código incluido en la etiqueta.

Además de estas características, las etiquetas `<param>` definen las siguientes propiedades para este vídeo:

- **Reproducción automática:** Desactivada (`name="autoplay" value="false"`).
- **Reproducir indefinidamente:** Desactivado (`name="autoplay" value="false"`).

El resultado de este código HTML lo podemos ver en la figura 5.12.

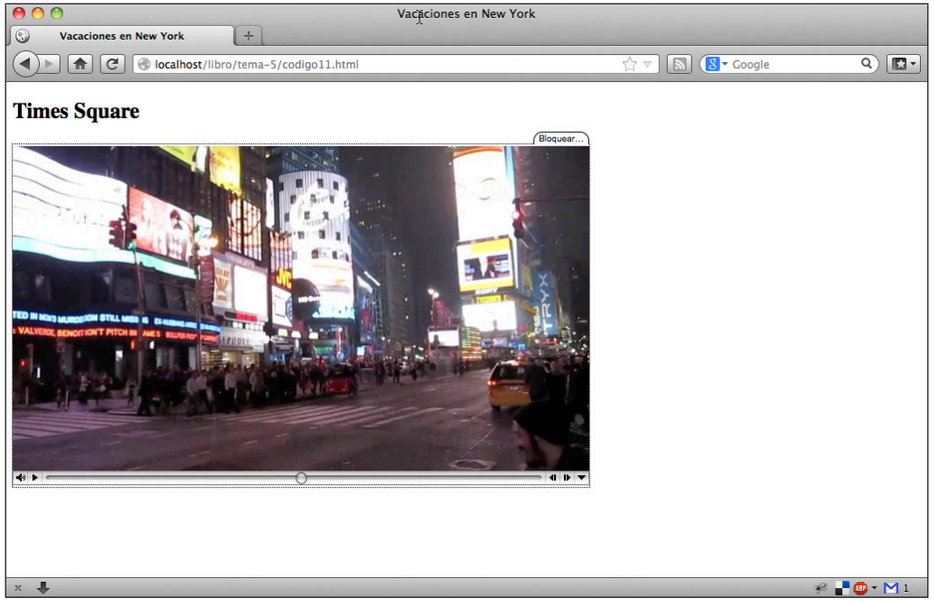

Figura 5.12. Vídeo insertado en una página web.

Teniendo claro este ejemplo, la forma de incluir otro tipo de fichero de vídeo o de audio será exactamente igual. Únicamente tendríamos que prestar atención al nuevo tipo MIME del objeto multimedia para definirlo de forma correcta en el atributo `type` y a los posibles nuevos parámetros que hubiera que definir en la etiqueta `<param>`.

NOTA:

Cuando el contenido multimedia que pretendemos insertar en nuestra página proviene de una web externa, es recomendable y casi imprescindible agregar directamente el trozo de código HTML que normalmente nos proporciona el sitio que comparte el objeto multimedia. Normalmente, se trata de un código más extenso y complejo, en el que sí que aparecen atributos tales como `classid`, `codebase` *y* `codetype` *y cuyos valores influyen directamente en el valor de otros atributos.*

Al hilo de lo que comentábamos anteriormente acerca de la inserción de contenido externo en nuestra web, en la figura 5.13 podemos ver una captura de pantalla de la mayor plataforma de vídeos *online*, YouTube. En la parte inferior de la pantalla, tenemos el trozo de código HTML que se corresponde con el vídeo que estamos visualizando.

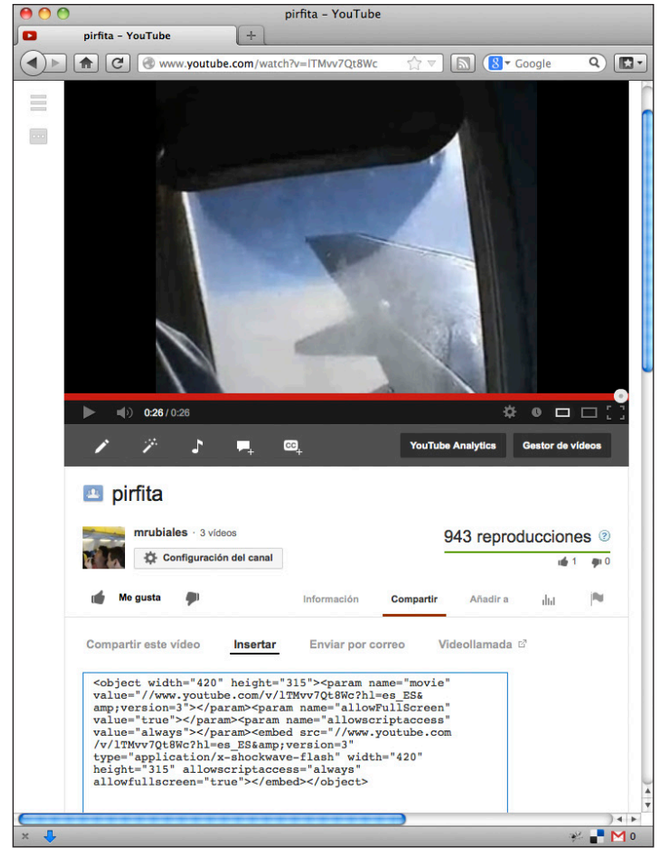

Figura 5.13. Cualquier plataforma de vídeos *online* proporciona el código de inserción de elementos multimedia, en este caso, YouTube.

7 Conceptos básicos de CSS

En este capítulo aprenderás:

- Finalidad de las hojas de estilo y conceptos básicos.
- Cómo integrar HTML y CSS.
- Estructura básica de una regla CSS.
- Cuáles son y cómo funcionan las diferentes unidades y medidas en CSS.

El objetivo de este capítulo es explicar y detallar los conceptos principales de las hojas de estilo, así como su funcionamiento e integración en un documento HTML. Como ya hemos hecho en los anteriores capítulos del libro, toda explicación irá acompañada de ejemplos con los que podrá afianzar los conocimientos adquiridos.

Introducción

Las hojas de estilo en cascada (CSS, *Cascading Style Sheets*) es un lenguaje de estilos que define el aspecto, la presentación y la posición que tendrán los diferentes elementos que componen una página web. El protagonismo que ha ido adquiriendo el uso de las hojas de estilo en el mundo del diseño web ha ido creciendo año tras año hasta convertirse en un mecanismo totalmente imprescindible. A continuación, podemos ver una lista detallada con las principales ventajas que nos aportan las hojas de estilo:

- El uso de CSS nos permite obtener una completa separación entre el contenido y la presentación de los datos, algo muy importante a tener en cuenta cuando el desarrollo web se encuentra dividido en grupos de personas.
- Posibilidad de modificar el diseño de una página web por interacción del usuario.
- Con una hoja de estilos bien diseñada, conseguiremos una mejora en la velocidad de carga de nuestra página web al economizar líneas de código.
- Al usar hojas de estilos, indirectamente estamos evitando el uso de ciertos elementos que no cumplen con las normas de accesibilidad, como, por ejemplo, el uso de tablas como base del diseño de una página web.

Historia del lenguaje CSS

El padre de la criatura fue el científico Håkon Wium Liel, quien, en 1994, promovió la creación de las hojas de estilo en cascada con el fin de lograr una web mejor estructurada y con una separación más eficiente de las distintas partes que conforman el desarrollo web. En la actualidad, son cuatro las especificaciones o versiones disponibles de CSS; por norma general, cada versión ha ido aumentando las prestaciones y el alcance de las versiones anteriores.

- **CSS1:** Primera especificación oficial de CSS que vio la luz allá por diciembre de 1996. Pese a ser una primera versión, ya contaba con el visto bueno del W3C. Con esta primera versión ya se podían configurar aspectos tales como:

- `.texto { font-size:10px; }`: Define un tamaño de letra de 10 px para cualquier etiqueta del documento que tenga un atributo `class="texto"`. Resultado: el texto `Párrafo-2` tendrá un tamaño de 10 px.
- `div h3 a.link{ color:green; }`: Especifica un color de texto verde en cualquier elemento `<a>` con atributo `class="link"` que esté encerrado dentro de un elemento `<h3>` y que este a su vez se encuentre dentro de un `<div>`. Resultado: el texto `Enlace-1` irá en color verde.

Selector de identificador de elemento

Este tipo de selector nos permite aplicar una regla CSS al elemento HTML que tenga definido el mismo atributo `id` que el declarado en selector de la regla. Debido a que en un documento HTML no pueden existir dos elementos con el mismo atributo `id`, este selector es el más específico que existe en CSS, su naturaleza solamente le permite seleccionar un único elemento. Este selector se establece mediante el carácter # (almohadilla) seguido del valor del atributo `id`, por ejemplo, `#titulo{color:orange;}`, esta regla define un color de texto naranja para el elemento del documento HTML que tiene como atributo `id="titulo"`.

Del mismo modo que ocurre con otro selectores, también en este caso es posible combinar el nombre de un elemento con el selector de `id`: `h1#titulo{color: orange;}`; sin embargo, esta definición solo tendría sentido cuando la regla CSS se aplica a varios documentos HTML, ya que, como hemos mencionado con anterioridad, no puede existir más de un elemento con el mismo `id` en un mismo documento HTML. En ocasiones, el atributo `id="texto"` puede aparecer en varios documentos HTML y en tipos de elementos diferentes; en ese caso, sí que sería útil la definición combinada `h1#titulo{color:orange;}` para concretar a qué tipos de elementos con `id="texto"` se le aplicaría la regla CSS.

NOTA:

De nuevo es importante prestar atención a los espacios a la hora de crear el selector, no es lo mismo `h1#titulo` *que* `h1 #titulo`.

Selector universal

El selector universal está representado por un asterisco (`*`) y tiene como función seleccionar todos los elementos del documento HTML. El ejemplo `* { font-size:12px; color:red }` es un selector poco utilizado, ya que no es una práctica muy común crear una regla de estilos para todos los elementos del documento.

Combinar selectores

Es una práctica muy común y casi imprescindible combinar los diferentes selectores que hemos visto para poder diseñar hojas de estilo eficientes y bien codificadas. La mejor forma de aprender y coger soltura en este aspecto es practicando un poco, así pues, tomando como base el documento HTML que viene a continuación, el listado 8.7, explicaremos cuál sería el resultado de aplicar varias reglas de estilo con selectores combinados.

Listado 8.7. Código HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Selectores descendentes en CSS</title>
    <meta http-equiv="content-type" content="text/html;charset=utf-8">
  </head>
  <body>
    <div>
      <p class="texto">Párrafo-1</p>
      <div class="texto">Párrafo-2</div>
      <h3><a href="" class="link">Enlace-1</a></h3>
      <a href="">Enlace-2</a>
      <div class="capa1">Párrafo-3</div>
      <div>Párrafo-4</div>
    </div>
  </body>
</html>
```

- `div p{ color: green }`: Color de texto verde para los elementos `<p>` definidos dentro de un elemento `<div>`. Resultado: `Párrafo-1` de color verde.
- `div, p { color:green; }`: Aplicar texto de color verde a todos los elemento `<div>` y `<p>`. Resultado: `Párrafo-1`, `Párrafo-2`, `Párrafo-3` y `Párrafo-4` de color verde.
- `a { font-size:20px }`: Tamaño de fuente de 20 px para todos los elementos `<a>` del documento. Resultado: Texto `Enlace-1` y `Enlace-2` con un tamaño de 20 px.
- `a.link { font-size:30px }`: Tamaño de fuente de 30 px únicamente para los elementos `<a>` con atributo `class="link"`. Resultado: Texto `Enlace-1` con un tamaño de 30 px.
- `div div.capa1, div p.texto { color:red; }`: Color rojo a los elementos `<div>` con atributo `class="capa1"` que estén definidos dentro de un `<div>` y a los elementos `<p>` con atributo `class="texto"` que se encuentren encerrados en un `<div>`. Resultado: `Párrafo-1` y `Párrafo-3` de color rojo.

seguidamente la Capa-2 y así sucesivamente. Si no fuera posible ubicar más capas en la misma línea, el resto de elementos flotantes bajarían una línea y se posicionarían, en este caso, lo más a la izquierda posible.

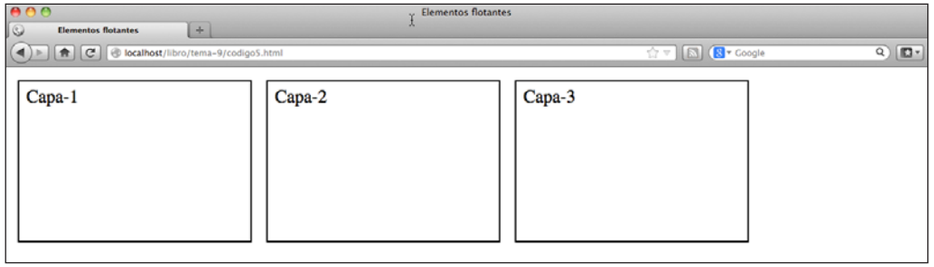

Figura 9.11. Las tres cajas flotan a la izquierda respetando el orden y sus posiciones.

Listado 9.5. Código HTML y CSS correspondiente a la figura 9.11, todas las cajas flotan a la izquierda.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Elementos flotantes</title>
    <meta http-equiv="content-type" content="text/html;charset=utf-8">
    <style type="text/css">
      div{
        float:left;
        font-size:25px;
        border:2px solid black;
        margin:10px;
        padding:10px;
        height:200px;
        background-color:white;
        width:300px;
      }
    </style>
  </head>
  <body>
    <div id="capa1">Capa-1</div>
    <div id="capa2">Capa-2</div>
    <div id="capa3">Capa-3</div>
  </body>
</html>
```

Una práctica muy común dentro del diseño web con hojas de estilos es utilizar elementos flotantes para representar una imagen o fotografía rodeada de un párrafo, algo que normalmente podemos ver en artículos de periódicos *online* o entradas de blogs. En la figura 9.12 podemos observar un ejemplo de esto que acabamos de mencionar.

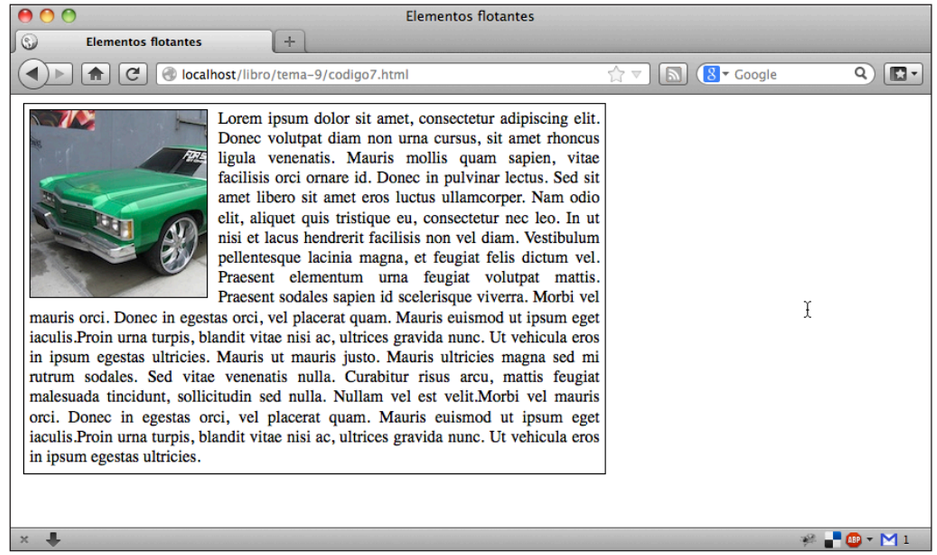

Figura 9.12. Texto flotando alrededor de una fotografía.

Listado 9.6. Uso de elementos flotantes para representar una fotografía incrustada en un párrafo.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Elementos flotantes</title>
    <meta http-equiv="content-type" content="text/html;charset=utf-8">
    <style type="text/css">
      div#contenedor{
        border:1px solid black;
        margin:5px;
        padding:5px;
        width:550px;
      }
      img#foto{
        float:left;
        border:1px solid black;
        height:180px;
        width:170px;
        background-color:green;
        margin-right:10px;
        margin-bottom:10px;
      }
      div#parrafo{
        font-size:16px;
        text-align:justify;
```

Esquinas redondeadas

Crear esquinas redondeadas de las capas o diferentes partes de nuestra página web es una de las características más demandadas por la comunidad de diseñadores web. Con CSS3 ya no es necesario echar mano de trucos raros o superponer imágenes para conseguir este efecto. La propiedad encargada para esta funcionalidad es `border-radius`. Por ejemplo, con `border-radius: 20px` estaríamos aplicándo unas esquinas con un radio de redondeo de 20 píxeles. El listado 11.1 muestra un ejemplo de cómo aplicar esta propiedad a una capa.

Listado 11.1. Código HTML y CSS correspondiente con la figura 11.1.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Esquinas redondeadas</title>
    <meta http-equiv="content-type" content="text/html;charset=utf-8">
    <style type="text/css">
      .capa_redondeada{
        width: 300px;
        height: 300px;
        background-color: green;
        border-radius: 20px;
      }
    </style>
  </head>
  <body>
  <div class="capa_redondeada"></div>
  </body>
</html>
```

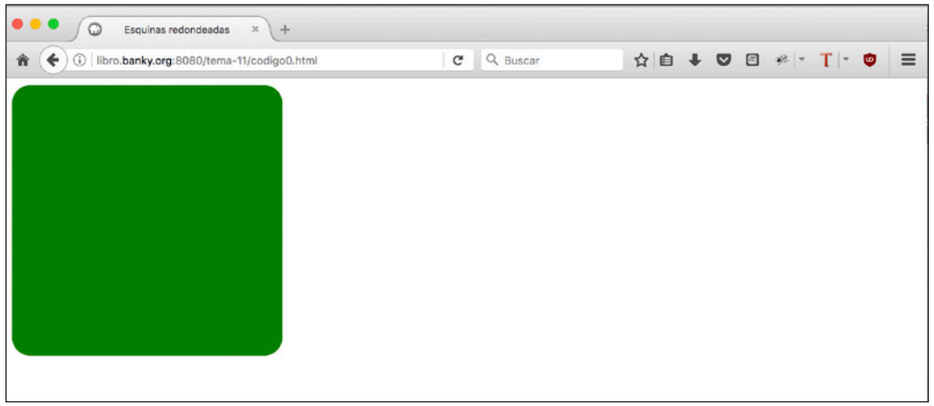

Figura 11.1. Capa con las esquinas redondeadas.

Como hemos podido observar en la figura 11.1, la propiedad `border-radius` establece el mismo redondeado para las cuatro esquinas de la capa en cuestión. CSS3 nos ofrece la posibilidad de especificar valores independientes para cada una de las cuatro esquinas existentes, estas propiedades son las siguientes:

- `border-top-left-radius`: Usar esta propiedad para especificar el radio de redondeo de la esquina superior izquierda de la capa.
- `border-top-right-radius`: Propiedad para especificar el radio de redondeo de la esquina superior derecha de la capa.
- `border-bottom-right-radius`: Usar esta propiedad para especificar el radio de redondeo de la esquina inferior derecha de la capa.
- `border-bottom-left-radius`: Propiedad para especificar el radio de redondeo de la esquina inferior izquierda de la capa.

Existe una forma más cómoda de especificar los diferentes valores de las cuatro esquinas utilizando la propiedad `border-radius`. Para esto, simplemente tendremos que establecer los valores de cada esquina en el mismo orden que han sido definidos anteriormente, por ejemplo: `border-radius: 5px 6px 7px 8px` generaría unas esquinas redondeadas de 5 px para la esquina superior izquierda, 6 px para esquina superior derecha, 7 px para la esquina inferior derecha y 8 px para la esquina inferior izquierda.

Sombras

Crear sombras sobre los diferentes elementos que componen una página web es otra de las características interesantes que nos permite hacer con facilidad la nueva versión de CSS. Para generar este tipo de efecto, tenemos disponibles dos propiedades: `text-shahow` y `box-shadow`.

- `text-shadow`: Propiedad específica para generar un efecto de sombreado sobre cualquier texto o párrafo. El uso de esta propiedad en su forma más básica es muy sencillo, consiste en especificar el valor de la sombra horizontal y sombra vertical respectivamente. Por ejemplo, con `text-shadow: 2px 3px` estaríamos definiendo una sombra horizontal de 2 px y otra vertical de 3 px. A partir de ahora y de forma opcional tendríamos la posibilidad de agregar un nivel de difuminado o el color de la sombra. Ejemplo: `text-shadow: 2px 3px 2px #CCCCCC`, en esta definición hemos agregado un nivel de difuminado de 2 px y un color gris claro (en hexadecimal) para la sombra.

sabría distinguir que comillas forman parte del texto y cuáles delimitan el inicio y el fin de la cadena. Para solucionar este tipo de problema, existe lo que se conoce como carácter de escape o barra invertida (\), el mecanismo consiste en insertar este carácter antes de las comillas; de esta forma, el navegador sabrá que debe interpretar las comillas tal y como son y no como un inicio o fin de declaración, es un método conocido comúnmente como escapar caracteres.

Una segunda opción para solucionar este problema consiste en algo tan sencillo como utilizar un tipo de comillas para encerrar el texto y otro tipo para usarlo dentro del propio texto. A continuación, podemos ver un ejemplo de ambas opciones con el mismo resultado:

```
  //Escapar comillas con la barra invertida
  var texto1="Este texto no lleva comillas \"pero este otro texto, sí que las
lleva\", hasta pronto";
  //Combinar el uso de comillas simples y comillas dobles
  var texto2='Este texto no lleva comillas "pero este otro texto, sí que las
lleva", hasta pronto';
```

El método de escapar caracteres también se usa cuando necesitamos incluir ciertos caracteres especiales dentro de nuestra cadena de texto. Estos caracteres especiales son el retorno de carro o nueva línea y la tabulación. La tabla 13.1 recoge de forma esquemática el funcionamiento del carácter de escape para cada una de estas situaciones especiales.

Tabla 13.1. Caracteres especiales en JavaScript.

Descripción	Carácter especial
Comillas dobles	\"
Comillas simples	\'
Nueva línea	\n
Golpe de tabulador	\t

El listado 13.5 muestra un ejemplo de varias cadenas de texto en las que se pone en práctica todo lo que acabamos de aprender.

Listado 13.5. Caracteres especiales en las cadenas de texto.

```
<script type="text/javascript">
    //Escapar comillas con la barra invertida y retornos de carro al final del texto
    var texto1="Este texto no lleva comillas \"pero este otro texto, sí que las
lleva\", hasta pronto\n\n";

    //Combinar el uso de comillas simples y comillas dobles y retornos de carro
al final del texto
    var texto2='Este texto no lleva comillas "pero este otro texto, sí que las
lleva", hasta pronto\n\n';

    //Uso del retorno de carro y tabulador
    var texto3="Esta frase pertenece a una línea\n y esta frase a otra
diferente. Ahora\t insertamos un golpe de tabulador."
    // Mostramos un mensaje con el valor de todas las variables.
    alert(texto1+texto2+texto3);
</script>
```

En un principio, quizás este método de mezclar texto con caracteres especiales pueda parecernos algo confuso, nada que no se solucione con un poco de práctica. Mientras tanto, únicamente debemos tener claro qué carácter especial necesitamos insertar y en qué posición exactamente queremos que aparezca. La figura 13.5 muestra el resultado final de este código en el navegador.

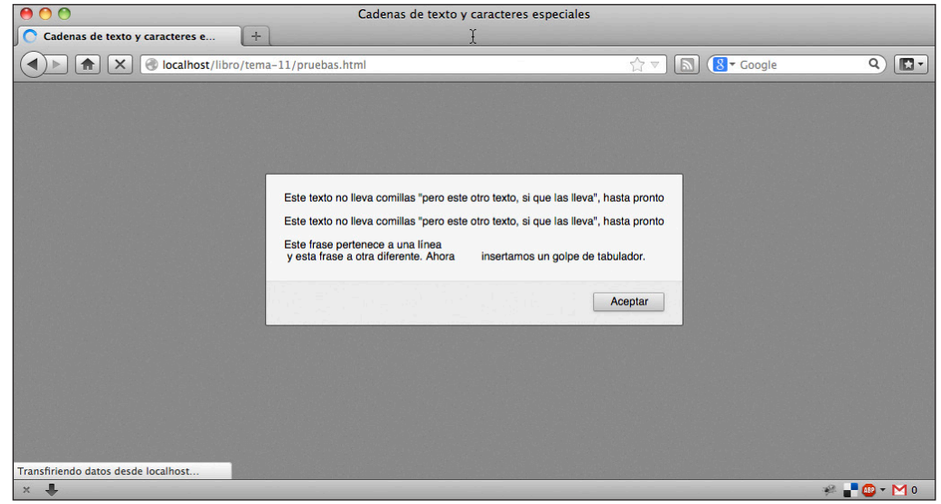

Figura 13.5. Ejemplo del uso de cadenas de texto y caracteres especiales.

NOTA:

La barra invertida o carácter de escape no es tratado por el intérprete de JavaScript como un carácter más. Este carácter le indica al navegador que lo que sigue debe tratarlo de forma especial. De hecho, la barra invertida por sí sola es transparente al usuario, no se muestra en pantalla una vez se ejecuta en el navegador. De tal forma que, para agregar un carácter de barra invertida como tal en una cadena de texto, deberemos definirla dos veces consecutivas, //.

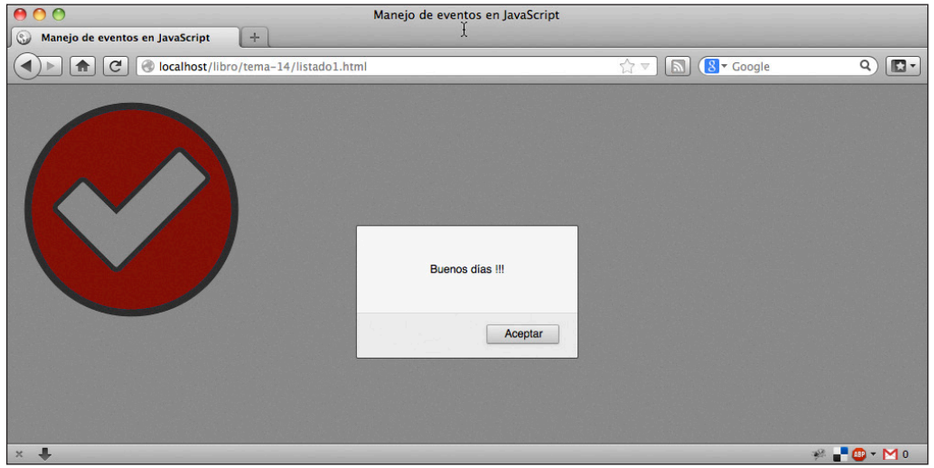

Figura 16.1. Funcionamiento del evento `onclick` sobre una imagen.

Las siguientes tablas recogen información resumida de los principales eventos existentes dependiendo del tipo acción o elemento implicado.

Eventos para elementos de formulario

Tabla 16.1. Tipos de eventos para formulario.

Evento	Descripción	Elementos
`onblur`	En la pérdida del foco	`<button>`, `<input>`, `<select>`, `<textarea>`
`onchange`	Al cambiar el valor de un elemento	`<input>`, `<select>`, `<textarea>`
`onfocus`	Al perder el foco	`<button>`, `<input>`, `<select>`, `<textarea>`
`onselect`	Al seleccionar el texto	`<button>`, `<input>`, `<select>`, `<textarea>`
`onsubmit`	Al enviar el formulario	`<form>`

Eventos para el documento web

Tabla 16.2. Tipos de eventos propios del documento HTML.

Evento	Descripción	Elementos
`onload`	Al cargarse la página web por completo	`<body>`
`onunload`	Al abandonar la página web actual	`<body>`

Eventos de ratón

Tabla 16.3. Tipos de eventos de ratón.

Evento	Descripción	Elementos
`onclick`	Al hacer clic	Todos los elementos
`ondblclick`	Al hacer doble clic	Todos los elementos
`onmousedown`	Al pulsar sin soltar el botón del ratón	Todos los elementos
`onmouseup`	Al soltar el botón que estaba pulsado	Todos los elementos
`onmousemove`	Al mover el ratón sobre la página web	Todos los elementos
`onmouseover`	Al situarse el puntero del ratón sobre un elemento	Todos los elementos
`onmouseout`	Al abandonar el puntero del ratón un elemento	Todos los elementos

Eventos de teclado

Tabla 16.4. Eventos de teclado.

Evento	Descripción	Elementos
`onkeydown`	Al presionar una tecla sin soltar	Todos los elementos
`onkeyup`	Al soltar la tecla que estaba pulsada	Todos los elementos
`onkeypress`	Al pulsar una tecla	Todos los elementos

Manejar eventos con JavaScript

Un evento por sí mismo no tiene funcionalidad alguna, es necesario asociar ese evento o acción del usuario a una instrucción JavaScript y de esta forma obtener la interactividad deseada. Como ya se mencionó con anterioridad, para enlazar un evento con JavaScript es necesario que el valor del atributo que representa el evento sea la instrucción o función JavaScript a ejecutar. Se trata simplemente de analizar qué evento del usuario deseamos capturar y qué acción queremos ejecutar tras la ejecución del evento. El siguiente código muestra un ejemplo de cómo utilizar el evento `onclick` sobre un campo de formulario para recordarle al usuario que los datos los debe introducir en mayúsculas.

Listado 16.2. Uso del evento `onclick` sobre un campo de formulario.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Manejo de eventos en JavaScript</title>
```

Curso de desarrollo web: HTML, CSS y JavaScript. Edición 2021

Manual Imprescindible

HTML, CSS y JavaScript son las tres tecnologías básicas en las que se sustenta el desarrollo de páginas web. Este libro le servirá de ayuda a la hora de adentrarse en el uso y aprendizaje de estas tecnologías.

Con este libro aprenderá a usar HTML para generar contenido en su sitio web, agregando bloques de texto, párrafos, tablas, listas, formularios, imágenes o cualquier elemento multimedia de forma sencilla. Del mismo modo, comprenderá cómo el uso de CSS le permitirá gestionar el aspecto de todo el contenido adaptándolo a sus necesidades o preferencias personales. Y, finalmente, aprenderá a programar funcionalidades básicas con JavaScript para proveer a su sitio web del nivel de interactividad adecuado.

En lo que al proceso de aprendizaje se refiere, partiremos de cero para estudiar los conceptos básicos de cada tecnología y seguidamente, de forma progresiva, iremos profundizando en las diferentes funcionalidades y características que cada tecnología nos ofrece. Todas las explicaciones estarán apoyadas en códigos de ejemplo bien documentados y en ejercicios resueltos.

2311364
ISBN 978-84-415-4414-7
9 788441 544147

ANAYA MULTIMEDIA
www.anayamultimedia.es