



Introducción a HTML5 y CSS3

Introducción

«Ojos que no ven, corazón que no siente», dice el antiguo dicho, y en desarrollo web hay mucho de eso. Las aplicaciones webs, además de ser sólidas, confiables y estables, deben ser atractivas. En este caso, las apariencias son importantes, ¡y muy! Pero no todo es meramente estético, sino que la experiencia de usuario en nuestras aplicaciones webs también es esencial. Es por esto que dedicaremos la lectura a conocer dos herramientas fundamentales para definir el aspecto visual de nuestras aplicaciones webs: HTML5 y CSS3. Ahondaremos en la sintaxis de cada una de estas, conoceremos sus bases y presentaremos algunas posibilidades para escalar con frameworks el diseño de la web.

1. HTMLS

Pongamos la lupa en el acrónimo HTML: HyperText Markup Language. Vamos a ver qué significa cada palabra:

- **HyperText.** Significa hipertexto, que no es más que un texto que enlaza con otros contenidos, que pueden ser otro texto u otro archivo. Esto es la base del funcionamiento de la web, tal y como la conocemos.
- **Markup.** Significa marca o etiqueta, ya que todas las páginas webs están construidas con base en etiquetas, desde las primeras versiones hasta las últimas etiquetas de HTML5. Un ejemplo de una etiqueta HTML es la que identifica a un párrafo, que se compone de la etiqueta, el contenido de la etiqueta y el cierre del párrafo, como puede ser `<p>HOLA</p>`.
- **Language.** Significa lenguaje, porque HTML es un lenguaje, es decir, tiene sus normas, su estructura y una serie de convenciones que nos sirven para definir tanto la estructura como el contenido de una web. Pero —¡atención aquí!— es un lenguaje, no un lenguaje de programación, ya que carece de estructuras de control.

Podemos definir al HTML, entonces, **como un estándar que sirve para definir la estructura y el contenido de una página web**. El 5 que lo acompaña refiere a su más reciente versión.

W3C

El world wide web consortium (W3C) es una comunidad internacional que colaborativamente reúne los estándares de la web. Está liderada por quien conocemos como el «inventor de la web», Tim Berners-Lee, y por Jeffrey Jaffe. Algunas de las decisiones que tomamos en el maquetado de las páginas webs responden al seguimiento de dichos estándares, ya que queremos que las páginas de las aplicaciones webs que desarrollemos puedan ser utilizadas por el mayor número de navegadores y dispositivos del mundo.

Sintaxis de HTML5

Te recomiendo, para ir probando cómo funciona esto, **que crees un archivo nuevo de texto y lo guardes con extensión «html»**. Luego, **ejecútalo automáticamente**, se va a abrir en tu navegador por defecto.

Todo el contenido está dentro de etiquetas de marcado, que llevan la *siguiente estructura*:

- Símbolo menor (<)
- Nombre de la etiqueta (por ejemplo, p)
- Símbolo mayor (>)
- El contenido
- Símbolo menor (<)
- Barra (/)
- Nombre de la etiqueta (por ejemplo, p)
- Símbolo mayor (>)

Por ejemplo, un título con la etiqueta h1 que dice «**este es el contenido**», se escribe así:

<h1>Este es el contenido </h1>.

La salvedad es para las etiquetas que no llevan contenido, como ser el salto de línea (br). En esos casos, la estructura es la siguiente:

- Símbolo menor (<)
- Nombre de la etiqueta (por ejemplo, p)

- Barra (/)
- Símbolo mayor (>)

Por ejemplo,
.

Algunas etiquetas, además de su contenido, pueden tener atributos para especificar detalles. La sintaxis general es la siguiente:

- Símbolo menor (<)
- Nombre de la etiqueta (por ejemplo, p)
- Espacio
- Esto se repite según los atributos que tenga la etiqueta:

- Nombre de atributo
- Símbolo igual (=)
- Valor de la etiqueta

- Símbolo mayor (>)
- El contenido
- Símbolo menor (<)
- Barra (/)
- Nombre de la etiqueta (por ejemplo, p)
- Símbolo mayor (>)

Por ejemplo, la etiqueta «a», que se usa para enlaces (links), la usamos con el atributo «href», para indicarle al enlace a dónde debe apuntar. Veamos el código y cómo se visualiza en el navegador:

Tabla 1. Ejemplo de sintaxis

`Ir a Google`

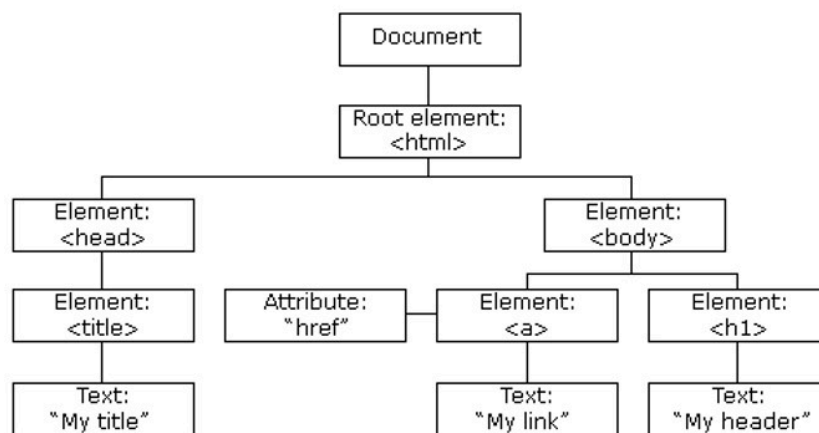
[Ir a Google](https://www.google.com)

Fuente: elaboración propia

Estructura de un archivo HTML

Cuando carga una página web en nuestro navegador, este crea un modelo de objetos del documento (DOM, por las siglas en inglés de document object model) de la página. El modelo HTML DOM es construido con una estructura de árbol de objetos, como la que se observa en la siguiente figura.

Figura 1. Estructura de árbol de objetos de una página web



Fuente: W3 Schools, s.f., <https://lc.cx/tE6AW->

Con la estructura de árbol, **nos referimos a que cada objeto está dentro de otro, excepto el elemento raíz «HTML»**.

El siguiente es un archivo prácticamente vacío, solo tenemos:

- **doctype**, que es la etiqueta que indica el tipo de notación que usaremos;
- **html**, que delimita el documento;
- **head**, que es la etiqueta que usaremos para delimitar las indicaciones de configuración (elementos que no queremos visibles en el navegador);
- **body**, que delimita el cuerpo de la página, lo visible.

```
<!doctype html>
<html>
  <head>
```

```
</head>
<body>
</body>
</html>
```

Dentro del head, vamos a comenzar agregando el charset. El charset permite definir la codificación de caracteres a utilizar. Aunque no es obligatorio, se debería incluir en cualquier documento HTML para que se vea correctamente en todos los navegadores y para respetar estándares. Lo incluimos del siguiente modo:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
  </body>
</html>
```

Dentro del **body**, colocamos todos los elementos que necesitamos. Algunas etiquetas disponibles son las siguientes:

- <div> para división dentro del contenido.
- <a>: para enlaces.
- para poner el texto en negrita.
-
 para saltos de línea.
- <h1>...<h6> para títulos dentro del contenido.
- para añadir imágenes al documento.
- para listas ordenadas, para listas desordenadas, para elementos dentro de la lista.
- <p> para párrafos.
- <form> para formularios.
- <input> para campos de un formulario.
- <label> para etiquetas de un formulario.

No tiene sentido que aprendas todas estas etiquetas antes de necesitarlas, pero sí es importante conocerlas.

A continuación, hagamos pruebas, para observar algunas de las etiquetas (en las imágenes del código HTML mostraremos solo el body, dado que el resto permanecerá idéntico).

Tabla 2. Prueba de un encabezado y dos párrafos

<pre><body> <h1> Hola </h1> <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus rutrum in lectus ac tristique. Ut a efficitur metus. Vivamus venenatis molestie felis a dictum. Praesent ut eleifend leo, et pulvinar felis. Sed ut condimentum elit. Donec bibendum feugiat lacus, cursus pretium turpis sagittis ut. </p> <p> Phasellus sodales sem eu urna finibus, eu maximus arcu porttitor. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque ullamcorper diam nec elit condimentum lacinia.</p> </body></pre>	<h2>Hola</h2> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus rutrum in lectus ac tristique. Ut a efficitur metus. Vivamus venenatis molestie felis a dictum. Praesent ut eleifend leo, et pulvinar felis. Sed ut condimentum elit. Donec bibendum feugiat lacus, cursus pretium turpis sagittis ut.</p> <p>Phasellus sodales sem eu urna finibus, eu maximus arcu porttitor. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque ullamcorper diam nec elit condimentum lacinia.</p>
--	---

Fuente: elaboración propia

Nota: el texto utilizado es texto de relleno, de la herramienta Lorem Ipsum (<https://es.lipsum.com/>). Esta herramienta te resultará muy útil cuando estés probando algún diseño y aún no tengas definidos los contenidos reales.

Tabla 3. Prueba de lista desordenada

<pre><body> <h1> Hola </h1> <h2> Esta es una lista desordenada: </h2> Primer elemento Segundo elemento Tercer elemento </body></pre>	<h2>Hola</h2> <p>Esta es una lista desordenada:</p> <ul style="list-style-type: none">• Primer elemento• Segundo elemento• Tercer elemento
--	---


Fuente: elaboración propia

Tabla 4. Prueba de saltos de línea

<pre><body> Sin saltos de línea.
 Con
 saltos
 de
 línea </body></pre>	<p>Sin saltos de línea.</p> <p>Con saltos de línea.</p>
--	---

Fuente: elaboración propia

Tabla 5. Prueba de imagen

<p>Ruta relativa (misma ubicación que el HTML):</p> <pre><body> <h1> Persona </h1> </body></pre> <p>Ruta relativa (otra ubicación):</p> <pre><body> <h1> Persona </h1> </body></pre> <p>Ruta absoluta en el equipo:</p> <pre><body> <h1> Persona </h1> </body></pre> <p>Ruta absoluta en la web (URL):</p> <pre><body> <h1> Persona </h1> </body></pre>	<p>Persona</p> 
---	--

Fuente: elaboración propia

Tabla 6. Prueba de formulario con campos

<pre><h1> Por favor, complete sus datos:</h1> <div> <div> <label for="nombre">Nombre:</label><input type="text" name="nombre"/> </div> <div> <label for="apellido">Apellido:</label><input type="text" name="apellido"/> </div> <div> <label for="genero">Género:</label> <input type="radio" name="genero" value="varon"> <label for="varon">Varón</label> <input type="radio" name="genero" value="mujer"> <label for="mujer">Mujer</label> <input type="radio" name="genero" value="otro"> <label for="otro">Otro</label> </div> <div> <label for="comoConociste">¿Cómo conociste el curso?</label> <div> <input type="checkbox" name="redesSociales" value="redesSociales"> <label for="redesSociales">A través de redes sociales</label>
 <input type="checkbox" name="email" value="email"> <label for="email">Por un email (newsletter)</label>
 <input type="checkbox" name="recomendado" value="recomendado"> <label for="recomendado">Me lo recomendó alguien</label> </div> </div> <div> <input type="submit" value="Enviar" /> </div> </div></pre>	<p>Por favor, complete sus datos:</p> <p>Nombre: <input type="text"/></p> <p>Apellido: <input type="text"/></p> <p>Género: <input type="radio"/> Varón <input type="radio"/> Mujer <input type="radio"/> Otro</p> <p>¿Cómo conociste el curso?</p> <p><input type="checkbox"/> A través de redes sociales</p> <p><input type="checkbox"/> Por un email (newsletter)</p> <p><input type="checkbox"/> Me lo recomendó alguien</p> <p><input type="button" value="Enviar"/></p>
--	---

Fuente: elaboración propia

Estilo en línea de una etiqueta

El atributo «style» para las etiquetas visibles en el navegador, nos permiten definir detalles de la visualización. El valor es un conjunto de pares clave: valor, separados por punto y coma.

Vamos con algunos ejemplos.

Tabla 7. Ejemplos de uso de atributo «style»

<pre><body> <ol style="background: #3399ff;"> Uno Dos Tres </body></pre>	<div>1. Uno</div> <div>2. Dos</div> <div>3. Tres</div>
<pre><body> <h1 style="background-color: #00ff00; font-family: sans-serif; text-align: center; color: #444444;"> Hola </h1> </body></pre>	<div>Hola</div>

Fuente: elaboración propia

2. CSS

Supongamos que tenemos muchos párrafos en la página y queremos que todos se vean similares, con la misma tipografía y colores. Para ello, **debemos repetir la etiqueta «style» por cada una de ellas y repetir todos los estilos que tienen**. Pero *¿qué pasa si tenemos la página de inicio y luego un listado que tiene la misma estética, pues es la misma página?* Bueno, tendremos que **copiar todas las indicaciones de estilos para cada párrafo de cada página**.

¿Cuál es el problema con tanto copy & paste? Imagínate que nuestro cliente, La Siglo, la editorial que comercializa sus libros online, cuando ya tiene su aplicación funcionando, opta por un cambio de imagen de la marca: se modernizaron y quieren una estética que comunique ese cambio. Pensemos algo simple: solo quieren cambiar los colores que usan. Si tenemos repetido el mismo color en cada página, en cada elemento que lo toma, tenemos que ir uno por uno a reemplazarlo, y es muy probable que olvidemos algún elemento o página. Es poco mantenible, es muy trabajoso hacer cambios. Encima, tenemos mezclado el código HTML con los estilos. En resumen, no nos sirve.

¿Cómo lo resolvemos? **Con las llamadas hojas de estilos en cascadas —en inglés, cascading style sheets (CSS)—**. Vamos a utilizar un archivo CSS, que va a agrupar todos los estilos necesarios para nuestra página. En muchas ocasiones, podremos aprovechar el mismo archivo CSS para diferentes HTML.

Cabe mencionar, que también es posible utilizar un **tag style** en el *head* del **HTML** y volcar allí todo el contenido del CSS. Sin embargo, **no consideramos esto conveniente**, ya que acopla por demás el HTML al CSS y **no permite la reutilización**.

Para vincular un CSS al HTML, vamos a incluir la etiqueta «*link*» en el head con la ruta relativa o absoluta al archivo CSS:

Figura 2. Etiqueta «link» en el head


```
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="styles.css">
</head>
```

Fuente: elaboración propia

Ahora, vamos a conocer un poco de la sintaxis de CSS. **Cada regla CSS consiste en un selector y un bloque de declaraciones.** El selector determina a qué elementos HTML se les va a aplicar el estilo. El bloque, rodeado por llaves, **contiene una o más declaraciones, finalizada cada una por punto y coma.** Cada declaración incluye **un atributo CSS y un valor, separado por dos puntos.**

Vamos con un ejemplo sencillo:

Figura 3. Ejemplo de declaración

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1> Hola</h1>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit.
      tristique sed vitae orci. Proin vel erat leo. Nullam
      quam, sit amet congue tortor mattis vel. Fusce mattis
    </p>
    <p>
      In felis nulla, suscipit sit amet lacus vel, commodo
      tortor. Proin rutrum viverra orci ac porttitor. Aenean
      vitae, feugiat erat.
    </p>
  </body>
</html>
```

Hola

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vulputate non lacus nec eleifend. Donec et ante sed justo placerat tristique sed vitae orci. Proin vel erat leo. Nullam tincidunt erat arcu, et luctus quam dignissim vel. Vestibulum lacinia felis quam, sit amet congue tortor mattis vel. Fusce mattis id sapien et pretium.

In felis nulla, suscipit sit amet lacus vel, commodo volutpat lorem. Maecenas nulla felis, fermentum nec porta in, pellentesque in tortor. Proin rutrum viverra orci ac porttitor. Aenean at varius nisi. Sed ut iaculis sapien. Cras ac nunc mollis, volutpat sem vitae, feugiat erat.

```
body {  
  background-color: black;  
}  
h1 {  
  color: white;  
}  
p {  
  color: #f2f597;  
  font-size: 0.8em;  
}
```

Fuente: elaboración propia

Ahondemos en el código CSS. La primera regla utiliza el selector «body», lo cual significa que todo lo que esté dentro del body, tendrá esas declaraciones aplicadas. En este caso, es una sola declaración que indica el color para el fondo. Los colores pueden ir con nombre, en RGB o en código hexadecimal, como en el color para p.

La segunda regla **afecta a los elementos h1 del HTML**. En este caso, el texto «Hola» aparece en color *blanco*.

La tercera regla **afecta a los párrafos, cambiándoles el color y disminuyendo su tamaño (por defecto, 1 em)**. Cabe destacar que el tamaño de texto (*font-size*) lo colocamos en em y no en

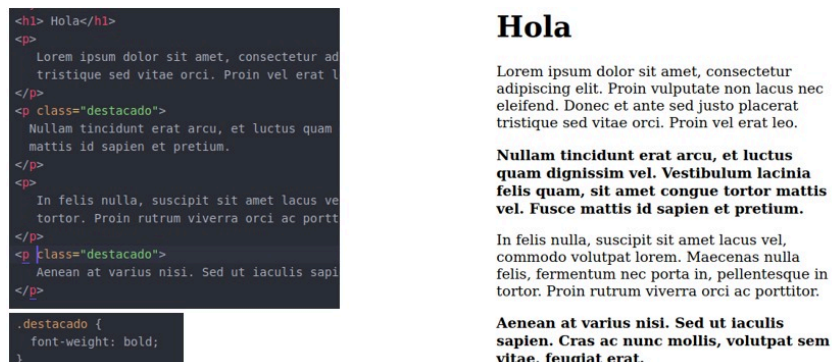
píxeles, con motivo de hacerlo más adaptable. Profundizaremos en este tema más adelante.

Los mencionados son los selectores más sencillos y que abordan todos los elementos de ese tipo del HTML.

Selectores

Tenemos también los selectores por clase, que solo afectan a los elementos HTML que tengan el atributo «class» con ese valor. En el CSS, el selector lleva un punto, seguido del nombre de la clase. Veamos un ejemplo:

Figura 4. Ejemplo de atributo «class»



Fuente: elaboración propia.

También, podemos usar el selector por ID. A diferencia de la clase, que puede ser compartida entre varios elementos del DOM, el ID debería ser único. En el CSS, el selector lleva el símbolo numeral (#), seguido del ID. Veamos en un ejemplo.

Figura 5. Ejemplo de selector por ID



Fuente: elaboración propia.

Los selectores pueden combinarse entre sí, indicando «p.destacado» o «h1.destacado», por ejemplo. Dos selectores separados por coma indican que se aplican a ambas esas declaraciones. A continuación, se presenta un ejemplo.

Figura 6. Ejemplo de selectores separados

```
.destacado, #primerParrafo, .importante {  
    font-weight: bold;  
}
```

Fuente: elaboración propia.

Dos selectores separados por un espacio, en cambio, indican seleccionar lo del término de la derecha que esté dentro de elementos del término de la izquierda. Por ejemplo, «div .importante» indica los elementos con clase importante que estén dentro de algún div; mientras que «div p» indica los párrafos que estén dentro de algún div.

Existen muchísimos símbolos y formas de combinar los selectores, pero con unos pocos nos bastará para manejar las bases del diseño web. A medida que necesites hacer trabajos más elaborados (por ejemplo, usar determinado estilo solo en las filas pares), irás descubriendo los secretos.

Box model

Los elementos visuales en CSS se organizan como si cada uno estuviera dentro de una caja (box). **Cada caja posee la siguiente estructura:**

- **Contenido (*content*):** texto o imágenes. Se delimita con atributos «width» y «height». «Width» se puede manejar en porcentaje o píxeles (50 % o 30 px); aunque es preferente el porcentaje, porque resulta más adaptable a diferentes resoluciones y tamaños de pantalla. «Height» es recomendable indicarlo en píxeles.
- **Relleno (*padding*):** es el área alrededor del contenido; es transparente.
- **Borde (*border*):** va alrededor del contenido y el padding, puede o no ser transparente.
- **Margen (*margin*):** es el área alrededor del borde; es transparente.

Figura 7. Box model



Fuente: W3 Schools, s.f.a., <https://lc.cx/Wm2rAg>

El *margin*, *border* y *padding* pueden ser configurables para cada elemento del DOM, al igual que el resto de los atributos. **Se puede seleccionar la misma medida para las cuatro direcciones (top, bottom, left, right) o seleccionar una diferente para cada una.**

Cuando tenemos una imagen, la recomendación para ajustar su tamaño es colocarla dentro de un contenedor; que la imagen tenga su ancho al 100 % y su alto en modo automático (para evitar deformarla modificando su relación de aspecto). Luego, ajustar el ancho o alto del contenedor.

Tabla 8. Ajustes del tamaño de la imagen

<pre><div class="cajaImagen"> </div></pre>	<pre>.cajaImagen { width:40%; } .cajaImagen img { width: 100%; height: auto; }</pre>
--	--

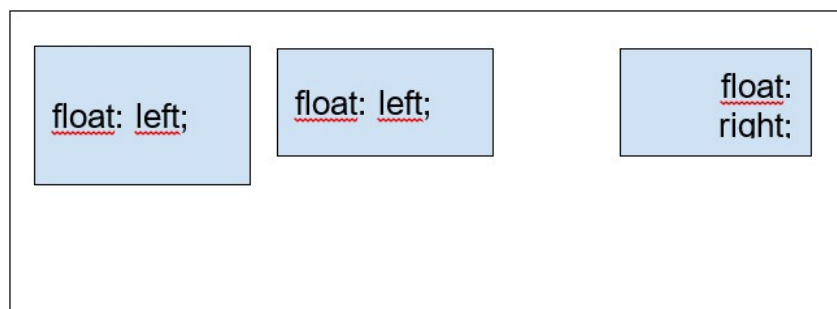
Fuente: elaboración propia.

Maquetado

Al maquetar, puede servirnos ubicar a los elementos HTML, como los contenedores, como *elementos flotantes*. Para esto, es utilizando el atributo «float» con valor «left» o «right», en el CSS.

El «float: left;» y «float: right;» se comportan permitiendo que diferentes bloques se ubiquen en el mismo renglón, con otros elementos que también floten.

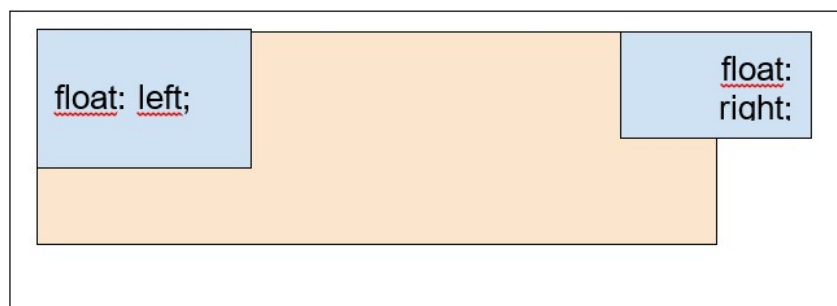
Figura 8. «float: left;» y «float: right;»



Fuente: elaboración propia.

Los elementos que están por debajo del elemento flotado asumen que este último no existe y tienden a ocupar el lugar vacío que el elemento flotante deja libre.

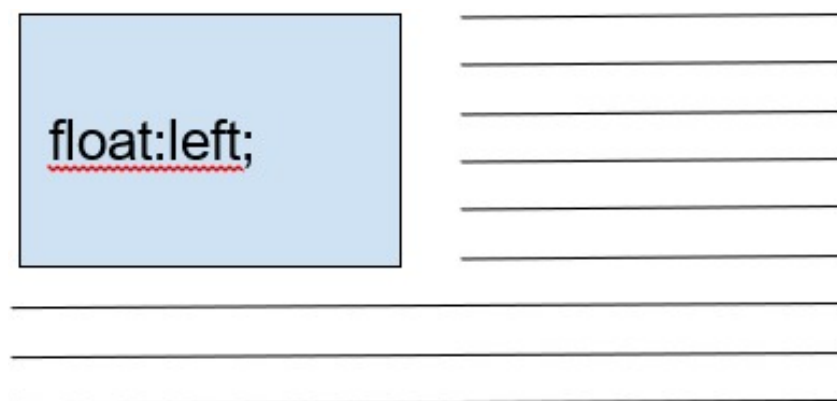
Figura 9. «float: left;» y «float: right;»



Fuente: elaboración propia.

Si los elementos «por debajo» son de texto en lugar de bloques, se acomodan de otra manera, como se observa en la figura 10.

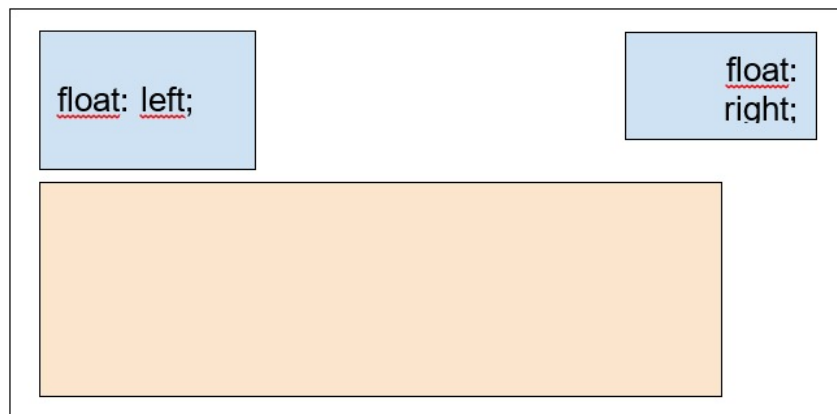
Figura 10. Elementos de texto «por debajo»



Fuente: elaboración propia.

Si se desea limpiar la flotación en el siguiente elemento y que no se comporte también flotando ni quedando por detrás, debemos usar «clear: both;».

Figura 11. Limpiar la flotación



Fuente: elaboración propia.

Pruebas en el navegador

Es recomendable ir actualizando en el navegador a medida que hacemos cada cambio para probar cómo se visualiza. También, podemos hacer modificaciones «en vivo» en el navegador, aprovechando la solapa de inspeccionar código. Debemos tener en cuenta que estas modificaciones, a pesar de impactar instantáneamente en la visualización de la página, no se guardan en los archivos HTML ni CSS; solo sirven para hacer pruebas.

Frameworks CSS

Muchas veces, para ordenar y ahorrar trabajo en diseño web, es recomendable utilizar frameworks CSS. Los frameworks nos proveen de formas de organizar el código, así como código CSS prearmado, que se incluye a modo de librerías en nuestro proyecto, para no empezar desde cero el estilo.

Así, por ejemplo, nos brindan clases para utilizar en nuestro DOM que, al colocarlas, directamente transforman nuestros elementos visualmente.

Vamos con un sencillo ejemplo con el **framework CSS** más utilizado, **Bootstrap**. Primero, debemos incluir el CSS que nos indican en la documentación más reciente en el head. Luego, atender al nombre de clase que nos solicita usar para ver un cartel que indica algo exitoso (color verde, etcétera), en este caso «**alert-success**».

Figura 12. Ejemplo de framework CSS - Bootstrap

```

<html>
<head>
<meta charset="utf-8">
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-Jckb8q31q61gN9Kgb8thS5NjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9U30Z" crossorigin="anonymous">
<link rel="stylesheet" href="styles.css">
</head>
<body>
<p class="alert-success">
  Este es un mensaje de éxito
</p>
<h1> Hola</h1>
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vulputate non lacus nec eleifend. Donec et ante sed justo tristique sed vitae orci. Proin vel erat leo.
</p>
<p>
  In felis nulla, suscipit sit amet lacus vel, commodo volutpat lorem. Maecenas nulla felis, fermentum nec porta in, pellentesque in tortor. Proin rutrum viverra orci ac porttitor.
</p>
<p>
  Aenean at varius nisi. Sed ut iaculis sapien. Cras ac nunc mollis, volutpat sem vitae, feugiat erat.
</p>
</body>
</html>

```

Este es un mensaje de éxito

Hola

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vulputate non lacus nec eleifend. Donec et ante sed justo placerat tristique sed vitae orci. Proin vel erat leo.

In felis nulla, suscipit sit amet lacus vel, commodo volutpat lorem. Maecenas nulla felis, fermentum nec porta in, pellentesque in tortor. Proin rutrum viverra orci ac porttitor.

Aenean at varius nisi. Sed ut iaculis sapien. Cras ac nunc mollis, volutpat sem vitae, feugiat erat.

Fuente: elaboración propia.

Esto es solo un ejemplo, pero Bootstrap **realmente es una herramienta muy potente** para estilar nuestra página rápidamente con una estética moderna y actualizada.

Preprocesadores CSS

Cuando trabajamos mucho en CSS, en proyectos medianos y grandes, escribir el código puede volverse una tarea un poco repetitiva y tediosa. Debemos buscar códigos hexadecimales de colores que tenemos anotados por algún lado, no olvidar cerrar cada regla, etcétera. Un preprocesador de CSS es un lenguaje de scripting que extiende CSS y luego lo compila en CSS regular.

Existen actualmente *tres famosos preprocesadores de CSS*: **Less**, **Sass** y **Stylus**. Los dos primeros son los que se llevan gran parte del mercado, por lo que nos enfocaremos en ellos.

Less está basado en **JavaScript** y los archivos llevan la extensión «.less» antes de ser compilados en CSS. **Sass**, en cambio, está basado en **Ruby**, y sus archivos llevan la extensión «.scss». **Ambos permiten la utilización de variables que en CSS regular no tenemos.** El siguiente es un pequeño ejemplo en Sass con variables utilizadas para colores. Como puedes observar, además de las variables, se pueden realizar cálculos.

Figura 13. Ejemplo de uso de Sass


```
@azul-principal: #5B83AD;  
@celeste-secundario: @azul-principal + #111;  
  
#header {  
    color: @celeste-secundario;  
}
```

Fuente: elaboración propia

Al compilarlo, con Sass, se producirá un archivo así:

Figura 14. Ejemplo de uso de Sass

```
#header {  
    color: #6c94be;  
}
```

Fuente: elaboración propia

El proceso de compilación, en el caso de nuestro **stack tecnológico**, será realizado dentro del **framework Laravel**, luego de configurar el preprocesador queremos utilizar; pero esto lo veremos más adelante.

Referencias

W3 Schools, (s.f.). *What is the HTML DOM?*
https://www.w3schools.com/whatis/whatis_htmlDOM.asp

W3 Schools, (s.f.a.). *CSS Box Model*. https://www.w3schools.com/css/css_boxmodel.asp