

Añadiendo CSS al proyecto

Añadiendo CSS al proyecto	1
¿Qué aprenderás?	2
Introducción	2
Color	3
Palabras clave	3
RGB hexadecimal	4
RGB numérico	4
RGB porcentual	5
Texto	5
Color	5
Tipografía	6
Tamaño de fuente	6
em	7
rem	7
Alineación	8
Fondo	9
background-color	9
background-image	9
background-position	9
background-repeat	10
background-size	10
Ejercicio guiado: Mi primer Hero-Section	11
Resumen	14



¡Comencemos!

¿Qué aprenderás?

- Implementar estilos CSS texto, color, fondo, alineación, imágenes y el modelo de cajas, para la definición de aspectos visuales de una interfaz web.

Introducción

Los estilos CSS son de vital importancia para el desarrollo web. Piensa en una página que contenga solo texto, ¿No es muy atractiva verdad?

Hasta ahora, hemos conocido la sintaxis de CSS y hemos probado algunas de sus propiedades. En esta lectura tendremos un primer acercamiento a los estilos relacionados con el **color, texto y fondo**, para que poco a poco nuestro proyecto vaya teniendo un estilo más profesional.

Una vez que hemos creado los contenidos y estructurado nuestro archivo HTML, utilizaremos lenguaje CSS, para definir el aspecto de cada uno de los elementos: color, tipografía, márgenes, posiciones, etc.

¡Vamos con todo!



Color

Los colores en CSS se pueden indicar de cinco formas diferentes:

Palabras clave

En CSS existen 17 palabras clave para referirse a los colores básicos. Las palabras se corresponden con el nombre en inglés de cada color:

maroon #800000	red #ff0000	orange #ffa500	yellow #ffff00	olive #808000
purple #800080	fuchsia #ff00ff	white #ffffff	lime #00ff00	green #008000
navy #000080	blue #0000ff	aqua #00ffff	teal #008080	
black #000000	silver #c0c0c0	gray #808080		

Imagen 1. Colores CSS.

Fuente: w3.org.

Un ejemplo de esto es:

```
p { color: red; }
```

Colores del sistema

Estos son similares a los mencionados anteriormente pero, en este caso, se refieren a los colores de los elementos que muestra el sistema operativo del usuario. Un ejemplo de esto es:

```
p { color: WindowText; background-color: Window }
```

Tanto esta forma como la anterior, son poco utilizadas al escribir código CSS, ya que son muy limitadas en relación a las otras opciones.

Acá te dejamos una lista completa de estos colores: w3.org.

RGB hexadecimal

Este es el método más habitual. El modelo RGB consiste en definir un color indicando la cantidad de color rojo, verde y azul que se debe mezclar para obtener ese color. El código de color hexadecimal está compuesto por un # más 6 números y/o letras:

Se especifica con: **#RRGGBB**, donde los enteros hexadecimales **RR** (rojo), **GG** (verde) y **BB** (azul).

En cada uno de sus 6 caracteres se permiten valores que van desde el 0 hasta la f (a partir del número 10 se empiezan a mostrar como letras, desde la a hasta la f): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e y f.

Eso especifica los componentes del color (0 = menor valor | f = mayor valor).

Un ejemplo de esto es:

```
p { color: #FF0000; } /*rojo*/
```

Gracias a esto, se pueden obtener hasta 16.777.216 combinaciones diferentes.

RGB numérico

Al igual que el RGB hexadecimal, el color se define indicando sus tres componentes R (rojo), G (verde) y B (azul). Cada uno puede tomar un valor entre 0 y 255.

Por ejemplo, para crear el color rojo, el primer componente tiene un valor máximo y las otras dos, toman un valor de 0:

Un ejemplo de esto es:

```
p { color: rgb(255, 0, 0); } /*rojo*/
```

RGB porcentual

Al igual que la lógica anterior, se toman los tres componentes R, G y B. La diferencia es que el valor mínimo (0) y máximo (255) están expresados en porcentajes.

Para el ejemplo del color rojo, sería expresado de la siguiente manera:

```
p { color: rgb(100%, 0%, 0%); } /*rojo*/
```

Texto

Color

La propiedad básica que define CSS relacionada con la tipografía se denomina color y se utiliza para establecer el color de la letra.

El color por defecto, si no definimos uno, en la mayoría de los navegadores es negro. Para establecer un estilo de color, lo haremos con el atributo **color** utilizando cualquiera de las cinco formas vistas anteriormente, de la siguiente manera:

```
p { color: #ffffff; } /*blanco*/
```

Si observamos con el inspector de elementos la página principal de Desafío Latam, veremos las propiedades de color en la fuente del banner:



Imagen 2. Color Texto.
Fuente: Desafío Latam.

Normalmente, la propiedad de color se define en el elemento `body`, para establecer un color de fuente base para todos los elementos de la página, junto con la tipografía y otras propiedades, como la alineación.

Tipografía

El tipo de letra se define con la propiedad **font-family** y se puede especificar de dos formas diferentes:

- Mediante el nombre de una familia tipográfica: Es decir, el nombre de la fuente. Por ejemplo: "Arial", "Verdana", "Garamond", etc.
- Mediante el nombre genérico de una familia tipográfica: Es decir, el estilo de la fuente. Por ejemplo: serif, sans-serif, cursive, fantasy y monospace.

Si quieres conocer más sobre las familias tipográficas, puedes revisarlo aquí: [MDN Web Docs](#).

La forma de utilizar esta propiedad en css, es la siguiente:

```
p { font-family: Montserrat,Times New Roman,fantasy; }
```

¿Qué pasa si el navegador del cliente no tiene la fuente que hemos definido? CSS nos permite definir más de una fuente, para prevenir estos casos.

En el ejemplo anterior, el navegador intentará mostrar la página con la fuente "Montserrat", si el usuario no la tuviera, probará con "Times New Roman", finalmente con la familia tipográfica "Fantasy". En el caso que no encontrara ninguna de ellas, mostrará el sitio con la fuente por defecto del navegador.

Tamaño de fuente

El tamaño de fuente se define con la propiedad **font-size** y se puede especificar de las siguientes maneras:

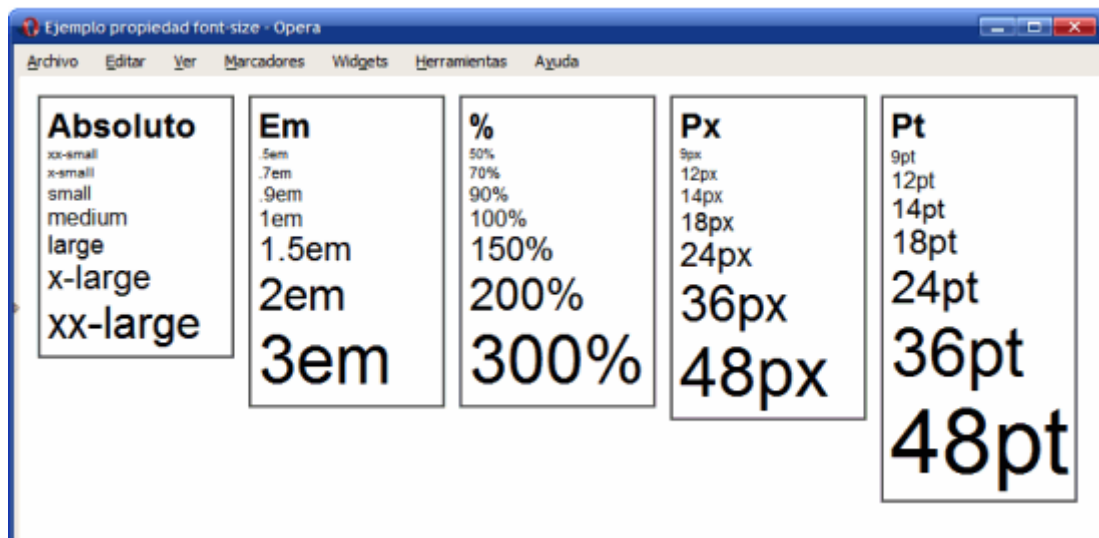


Imagen 3. Propiedad font-weight.

Fuente: [Uniwebsidad](http://uniwebsidad.com).

Para asignar tamaños a las fuentes no se recomienda utilizar los píxeles (px). Como estudiamos anteriormente, mencionamos las unidades de medida, los píxeles son de tipo absoluta, por ende rígidos. Si usamos píxeles también estamos ignorando las configuraciones que cada usuario pueda tener en su navegador.



Los navegadores de manera predeterminada definen un font-size de 16px al elemento HTML.

- *em*

La unidad em es escalable y siempre depende de su elemento padre. Por ejemplo, si el elemento body tiene un tamaño de fuente de 16px y un elemento hijo tiene una fuente con tamaño 1.3em, este texto se mostrará de un tamaño un 30% más grande que el del body (20.8px), mientras que si dentro de ese elemento tenemos otro hijo con un font-size de 1.3 em, el tamaño de fuente de este objeto sería un 30% más grande que el tamaño de su padre (27.04px). Es recomendable usarla para:

- Tamaños de fuente.
- Saltos de línea.
- Margen entre párrafos.

- *rem*

La unidad rem no es escalable porque no depende del elemento padre, sino del elemento raíz del documento, el elemento HTML. Si el elemento HTML tiene un tamaño de fuente de 16px (como es por defecto), entonces 1rem, sería igual a 16px, y si queremos aplicar un tamaño basado en rem a cualquier elemento de la página, no importará cual sea el tamaño

de fuente que tenga asociado ese elemento, ya que 1 rem siempre será igual a 16 píxeles a no ser que se modifique el elemento raíz. Es recomendable usarla para:

- Elementos del layout que tengan medida fija.

Por otra parte, el grosor de la letra se puede controlar con la propiedad **font-weight**, mientras que el estilo utiliza la propiedad **font-style**. Pensemos en el grosor como qué tan “negrita” queremos la fuente y el estilo, si es cursiva o no.

Su sintaxis es la siguiente:

```
p {  
  font-size: 1em;  
  font-weight: 300;  
  font-style: italic;  
}
```

Más adelante abordaremos en profundidad el uso de los estilos tipográficos, pero si quieres saber más sobre estas propiedades, puedes consultar el material disponible en lenguajecss.com

Alineación

La propiedad que define la alineación del texto se denomina `text-align` y los valores permitidos son: izquierda (`left`), derecha (`right`), centrado (`center`) y justificado (`justify`):

Su sintaxis es la siguiente:

```
p { text-align: center; }
```

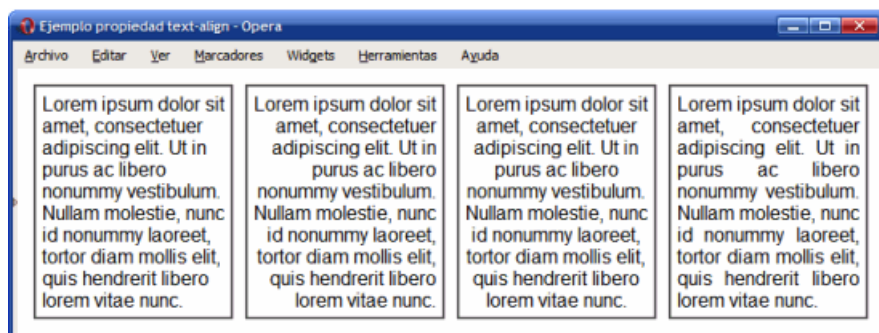


Imagen 4. Propiedad `text-align`.

Fuente: Uniwebssidad.

Existen además otras propiedades del texto como el interlineado (**line-height**), la decoración (**text-decoration**), transformación (**text-transform**), alineación vertical (**vertical-align**), indentación (**text-indent**), entre muchos otros. Puedes leer más sobre esto, en [este enlace](#).

Fondo

Existen distintas propiedades relacionadas al estilo del fondo (background) de nuestra página, algunos de ellos son:

- background-color.
- background-image.
- background-position.
- background-repeat.

background-color

Como su nombre lo indica nos permite fijar un color de fondo para el elemento, de la siguiente manera:

```
body {  
  background-color: #fff000;  
}
```

background-image

Permite establecer una imagen de fondo. El origen puede ser una carpeta del proyecto o un enlace a una fuente externa. Lo recomendable es almacenar todos los recursos de manera local.

```
body {  
  background-image: url("imagen.jpg");  
}
```

background-position

Indica la posición del elemento en relación al fondo.

```
body {  
  background-position: center; /*se posicionará al centro*/  
}
```

background-repeat

Indica si la imagen, de no cubrir completamente el fondo, se repetirá o no y en qué posición.

```
body {  
  background-repeat: no-repeat; /*no se repetirá*/  
}
```

background-size

Define el tamaño de la imagen que se utilizará como fondo en el elemento aplicado.

```
body {  
  background-size: cover; /*La imagen se auto ajustará al espacio  
disponible*/  
}
```

Si quieres conocer más valores o propiedades sobre fondos en CSS, puedes visitar la sección [CSS Backgrounds](#) del W3Schools.



Antes de continuar:

- ¿Reconoces la diferencia entre los 3 códigos de color RGB?
- ¿Conocías las propiedades del texto que te presentamos? ¿Cuál de ellas te costó más comprender?
- ¿Conoces alguna otra propiedad de estilo de fondo? ¿Cuál?

Si pudiste responder con facilidad todas las preguntas, puedes seguir adelante, si alguna te presenta dificultad, asegúrate de aclarar los puntos que más te hayan costado.

Ejercicio guiado: Mi primer Hero-Section

En este ejercicio aprenderemos a crear un Hero-Section aplicando los estilos que aprendimos en cuanto a textos y fondos con las propiedades de fuentes y background respectivamente. Ocuparemos como modelo el Hero Section del [Meet & Coffee](#).

Para este ejercicio ocuparemos el siguiente bloque de código HTML.

```
<header>
  <h1>Descubre lo último en tecnología bebiendo café</h1>
  <h3>Charlas, eventos y simposios sobre tecnología</h3>
</header>
```

El fondo de nuestra sección lo podrás conseguir en la carpeta [assets](#), el nombre del archivo es **bg-hero.png**. Crea la estructura de carpetas de un sitio web básico que aprendiste en lecturas pasadas e incluye en la carpeta **img** el archivo descargado.

¿Todo listo? ¡Comencemos!

- **Paso 1:** Asignemos el fondo a la etiqueta **header** utilizando la dirección de nuestra imagen, un posicionamiento central, un tamaño autoajutable y evitamos que se repita la imagen.

```
header {
  background-image: url(../img/bg-hero.png);
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}
```



Imagen 5. Imagen de fondo en Hero Section.
Fuente: Desafío Latam

Como puedes observar, la imagen se visualiza según la dimensión del elemento en el que es aplicada la propiedad de **background-image**.

- **Paso 2:** Como puedes notar en la [maqueta](#), el texto del Hero Section es blanco, tiene sombra, una tipografía específica y está alineado a la derecha, para esto ocupemos las propiedades **color**, **text-align**, **font-family** y una cuarta propiedad llamada **text-shadow** que nos ayudará a aplicar la sombra negra en el texto.

```
header {  
  background-image: url(../img/bg-hero.png);  
  background-position: center;  
  background-repeat: no-repeat;  
  background-size: cover;  
  
  /* Propiedades Agregadas */  
  font-family: Open sans;  
  text-shadow: 0px 3px 6px black;  
  text-align: right;  
  color: white;  
}
```

Estamos aplicando los estilos de texto al header ya que estos serán común para ambos encabezados. Para este punto nuestro Hero Section debe verse como en la siguiente imagen.



Imagen 6. Textos modificados en el Hero Section.
Fuente: Desafío Latam

Ahora tiene mucho más parecido a la [maqueta](#) de nuestro Meet & Coffee ¿Cierto? Pasemos ahora a modificar cada encabezado para que cumpla aún más lealmente a nuestro diseño

- **Paso 3:** Para apuntar al encabezado **h1** de nuestro header, utilizaremos la especificidad de CSS, en donde declaramos de forma horizontal la jerarquía de los elementos, siendo este encabezado parte del **header**, declaramos entonces los estilos de la siguiente manera:

```
header h1{  
  font-size: 60px;  
}
```

- **Paso 4:** Ahora solo falta definir el tamaño del **h3** de nuestro Hero Section, el cual tiene un tamaño de **60px**. Quedando de la siguiente manera:

```
header h3{  
  font-size: 40px;  
  font-weight: lighter;  
}
```

Con esto aplicado, nuestro Hero Section deberá verse de la siguiente manera:

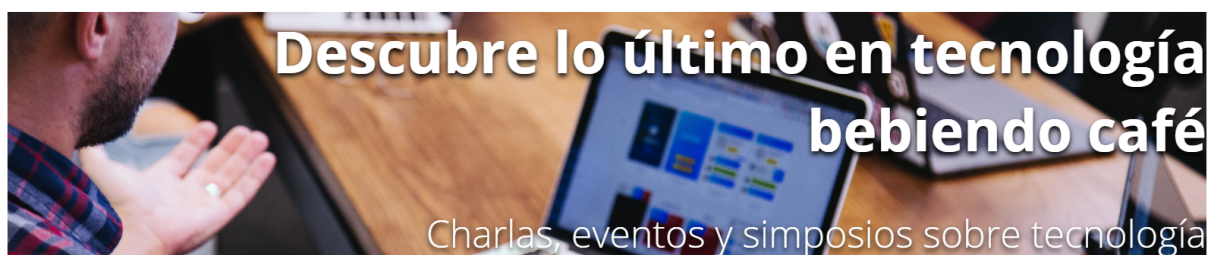


Imagen 7. Cabeceras modificadas en el Hero Section.

Fuente: Desafío Latam

Ten en cuenta que la presentación de los textos en relación con el fondo dependerá del zoom que tengas o de la resolución de tu pantalla, más adelante aprenderemos sobre el responsive design y los diferentes instrumentos para tratar de homologar lo más posible un diseño en función a su propuesta gráfica en diferentes dispositivos.

Como puedes apreciar, hay varias formas de aplicar estilos CSS, entre identificadores, clases, especificidad y demás, tenemos infinitudes de caminos para llegar a lo mismo, lo cual nos abre una amplia brecha de maneras y estilos de codificación, alejándonos de una “única y perfecta” manera de hacer algo.

Resumen

- En CSS existen 17 palabras clave para referirse a los colores básicos. Las palabras se corresponden con el nombre en inglés de cada color.
- El código de color RGB hexadecimal está compuesto por un # más 6 números y/o letras.
- El código de color RGB numérico se define indicando sus tres componentes R (rojo), G (verde) y B (azul). Cada uno puede tomar un valor entre 0 y 255.
- En el código de color RGB percentual se toman los tres componentes R, G y B. El valor mínimo (0) y máximo (255) están expresados en porcentajes.
- Para establecer un estilo de color, lo haremos con el atributo **color**.
- El tipo de letra se define con la propiedad **font-family**.
- El tamaño de fuente se define con la propiedad **font-size**.
- La unidad em es escalable y siempre depende de su elemento padre.
- La unidad rem no es escalable porque no depende del elemento padre, sino del elemento raíz del documento, el elemento HTML.
- La propiedad que define la alineación del texto se denomina text-align.
- background-color permite fijar un color de fondo para el elemento.
- background-image permite establecer una imagen de fondo.
- background-position indica la posición del elemento en relación al fondo.
- background-repeat indica si la imagen, de no cubrir completamente el fondo, se repetirá o no y en qué posición.