

Guía de estudio - CSS



¡Hola! Te damos la bienvenida a esta nueva guía de ejercicios.

¿En qué consiste esta guía?

En esta guía repasaremos cómo integrar hojas de estilo (CSS) en un proyecto web, y luego crearemos la estructura de un proyecto guardando nuestros archivos CSS en una carpeta diseñada para eso.

Además, revisaremos nuevamente la anatomía de una instrucción CSS y profundizaremos en las distintas formas de aplicar color, tamaños, unidades de medida y alineación de texto, con el fin de tener más herramientas para implementar diseños web efectivos. Por último, aplicaremos lo aprendido en una actividad práctica en la que trabajaremos sobre una tabla.

¿Cómo sacarle el máximo provecho a esta guía?

- En esta guía encontrarás muchos ejercicios similares a la guía anterior.
- Estos están diseñados para ayudarte a practicar y aplicar los conceptos y habilidades que se presentan en la guía.
- La práctica es fundamental para el aprendizaje efectivo, y los ejercicios te permitirán poner a prueba lo que has aprendido y mejorar tus habilidades en el tema abordado.

¡Vamos con todo!



Tabla de contenidos

¿Qué es CSS?	4
Formas de añadir CSS	4
Aregar CSS inline	5
Actividad 1: Agregando tus primeras líneas de CSS	5
Agrega CSS en el head	6
Actividad 2: Una página web con CSS en el head	6
Aregar CSS en un archivo externo	7
Actividad 3: Una página web con CSS en el head	8
Estructura de un proyecto con CSS	9
Actividad 4: Estructurando un proyecto con CSS	9
Anatomía de una instrucción CSS	10
Tipos de selectores	11
Profundizando en colores	12
Colores con palabra clave	12
RGB numérico	12
RGB porcentual	13
RGB hexadecimal	13
Tipografías	13
Tipografías soportadas en todos los navegadores	13
Estrategia de fallback	14
Unidades de medida y tamaños	14
El problema con PX	15
em y rem	15
Porcentajes	17
Grosor y cursiva	19
Fondo	19
background-color	19
background-image	20
background-position	20
background-repeat	20
background-size	20
Alineación del texto	21
Actividad 5: Aplicando lo aprendido en una tabla	21
Modelo de cajas	24
Propiedades del modelo de cajas	25
Reseteando las propiedades	25
Implementando lo aprendido	27
Actividad 6: Revisando los aprendizajes	38



¡Comencemos!

¿Qué es CSS?

Observa la siguiente imagen, ¿qué te llama la atención?

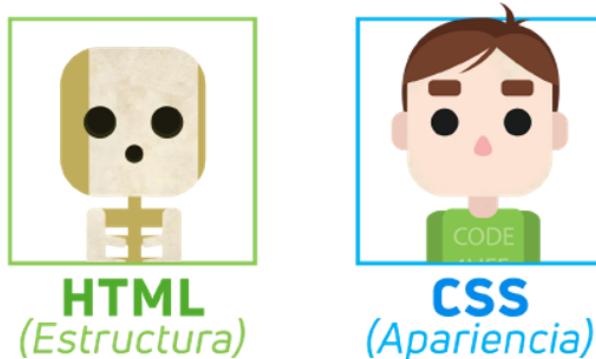


Imagen 1. HTML y CSS.

Fuente: Desafío Latam.

En la imagen se muestra que HTML es el esqueleto del personaje, es decir, su estructura, mientras que CSS da el estilo a esta estructura, por lo que nos sirve para agregar aspectos visuales a la página como colores, fuentes, tipografías, etc.



CSS es la sigla de “**Cascading Style Sheets**”, que significa “**Hojas de estilo en cascada**”. Su objetivo es definir la apariencia de los elementos HTML en la pantalla, en papel o en otros medios especificados.

Formas de añadir CSS

Existen tres formas de añadir CSS a las páginas:

1. En la misma etiqueta, utilizando el atributo style, lo que se conoce como estilo “inline”.
2. Añadiendo la etiqueta `<style>` en el head.
3. A través de un archivo de tipo CSS, llamado desde el head (formato que recomendamos).

Agregar CSS inline

Para agregar CSS a una etiqueta HTML, podemos utilizar el atributo style de la etiqueta. Este se emplea para especificar estilos de CSS directamente en la etiqueta HTML.

Por ejemplo, para cambiar el color del texto de un párrafo a rojo, puedes agregar el atributo style con la propiedad color y el valor red, de la siguiente manera:

```
<p style="color: red">Este es un párrafo en rojo</p>
```

También es posible incluir múltiples estilos en una sola etiqueta, separándolos con punto y coma (;). Por ejemplo, para cambiar el color de fondo y el tamaño de letra de un encabezado, puedes hacer lo siguiente:

```
<h1 style="background-color: blue; font-size: 24px;">Este es un encabezado azul y grande</h1>
```

Sin embargo, es importante mencionar que utilizar el atributo style puede resultar poco práctico en proyectos más grandes, ya que no permite reutilizar estilos entre múltiples etiquetas. En estos casos, se recomienda utilizar CSS externo.



Actividad 1: Agregando tus primeras líneas de CSS

- Crea un proyecto nuevo llamado mi-primer-css
- Dentro del proyecto, crea el archivo index.html
- Da una estructura base al archivo agregando el doctype y las etiquetas, html, head y body,
- Dentro del body agrega un párrafo,
- Agrega el atributo style a la etiqueta body con "background-color: black; color: white".

Agrega CSS en el head

Para agregar CSS en el head de una página web se puede utilizar la etiqueta <style>. Esta etiqueta se coloca dentro de la sección <head> del documento HTML y permite definir las **reglas de estilo** para los elementos de la página.

La sintaxis para utilizar la etiqueta <style> es la siguiente:

```
<head>
  <style>
    p {
      color: red;
    }
  </style>
</head>
```

Agregar CSS en el head nos permite reutilizar el código para múltiples etiquetas. Por ejemplo, podríamos lograr que todos los párrafos tuvieran algún color en específico, sin embargo, hay que especificar a qué elemento se le deben aplicar las reglas de estilo. Para esto, indicaremos la etiqueta, clase o identificador del elemento y luego entre llaves agregaremos las propiedades y sus valores.

Las clases e identificadores de un elemento las veremos más adelante en esta guía.



Actividad 2: Una página web con CSS en el head

- Crea un proyecto nuevo llamado mi-segundo-css
- Dentro del proyecto crea el archivo index.html
- Da una estructura base al archivo agregando el doctype y las etiquetas, html, head y body.
- Dentro de <body> agrega un encabezado h1 y tres párrafos.
- Dentro de <style> agrega una regla de estilo para body que diga {width: 80%; margin:auto}
- Dentro de <style> agrega una regla de estilo para párrafo que diga {color: #333}

Agregar CSS en un archivo externo

La última forma de agregar CSS, y la más recomendada, es a través de un archivo externo al documento HTML, el cual puede estar en nuestro mismo servidor o en otro. Para hacerlo, crearemos en nuestro proyecto un archivo llamado style.css, y luego utilizaremos la etiqueta <link> dentro de head para vincular este documento a nuestro HTML.

```
<link rel="stylesheet" href="style.css">
```

En esta línea de código, el atributo “rel” especifica la relación entre el documento actual y el documento vinculado, como es un archivo CSS, especificaremos que es de tipo **stylesheet**. Por otra parte, el atributo href especifica la ubicación del documento que queremos vincular, lo que significa que debemos proporcionar la ruta del archivo .css correspondiente.



Añadiendo múltiples archivos css

Es posible añadir múltiples archivos CSS. Cuando agregamos más de uno, el orden es muy importante, debido a que el segundo archivo podría reescribir las reglas de estilo definidas en el primero.

```
<link rel="stylesheet" href="estilo1.css">
<link rel="stylesheet" href="estilo2.css">
```

A modo de ejemplo, escribiremos el siguiente código HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1> Hola </h1>
  <p> Probando CSS </p>
</body>
</html>
```

A continuación, en la raíz del mismo proyecto, agregaremos un nuevo archivo llamado style.css. En este archivo escribiremos código CSS:

```
body {  
    width: 80%;  
    margin: auto;  
}  
  
p {  
    font-size: 1.5em;  
    color: #333;  
}
```

Al abrir la página con el navegador deberíamos poder observar el cambio de tamaño y color en los párrafos, así como los márgenes de la misma forma que observamos en la actividad anterior.



Actividad 3: Una página web con CSS en el head

- Crea un proyecto nuevo llamado mi-tercer-css
- Dentro del proyecto crea el archivo index.html
- Da una estructura base al archivo agregando el doctype y las etiquetas, html, head y body.
- Agrega al menos un párrafo al index.html
- Dentro de la carpeta del proyecto crea un archivo llamado style.css
- Dentro del archivo style.css, agrega una regla de estilo para los párrafos para que sean de color rojo (color:red).
- Utilizando link, carga el CSS creado en el archivo index.html

Estructura de un proyecto con CSS

En lugar de guardar los archivos CSS en la raíz del proyecto, los guardaremos en una carpeta “CSS” que crearemos dentro de la carpeta assets.

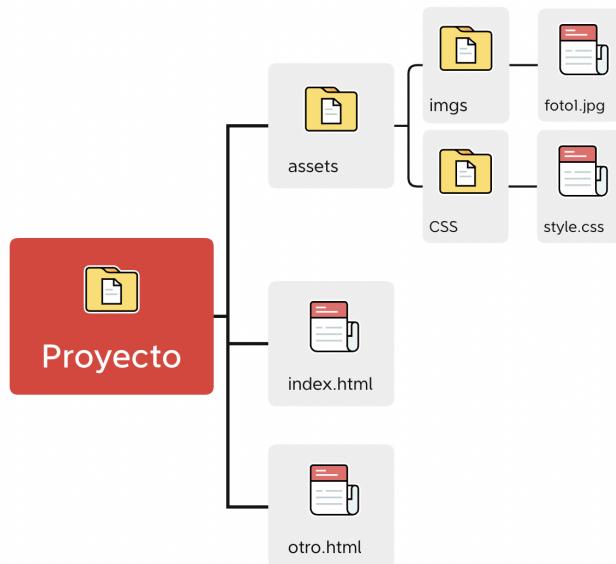


Imagen 2. Estructura de un proyecto web.

Fuente: Desafío Latam

Para cargar el archivo CSS creado en nuestro proyecto, tenemos que vincularlo utilizando la nueva dirección del archivo:

```
<link rel="stylesheet" href="assets/css/style.css">
```



Actividad 4: Estructurando un proyecto con CSS

1. Crea un proyecto nuevo llamado `landing-1`.
2. Carga el proyecto en el editor de código.
3. En él, crea la carpeta `assets` y dentro de esta, agrega la carpeta `imgs` y `css`.
4. Crea el archivo `index.html` con la estructura base de un documento html en la raíz del proyecto, y agrega un párrafo con `lorem ipsum` dentro del `body`.
5. Crea el archivo `style.css` dentro de la carpeta `css`.
6. Vincula el archivo CSS utilizando la etiqueta `link` en el archivo `index.html`.
7. Dentro del archivo `style.css` agrega la siguiente regla: `body { color: red; }`
8. Verifica que el color del texto haya cambiado.

Anatomía de una instrucción CSS

Cuando agregamos instrucciones CSS a través de una hoja estilos externo u ocupando el atributo style en el header, debemos incluir un **selector de etiqueta**.

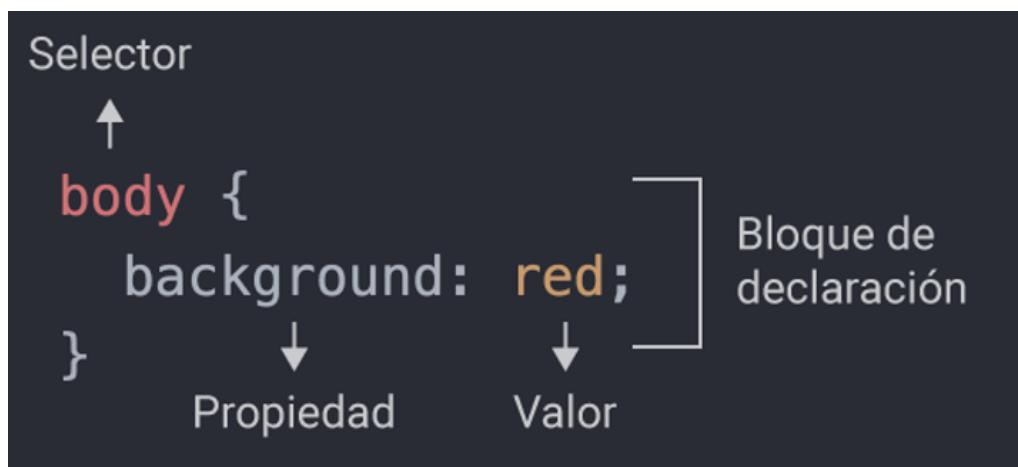


Imagen 3. Sintaxis CSS.

Fuente: Desafío Latam.

Observando la imagen, podemos identificar los siguientes componentes de un selector:

- Selector: indica qué elemento del HTML queremos modificar, en este caso seleccionamos body.
- Llaves curvas({ ... }): determinan el inicio y cierre del bloque de declaración.
- Declaración: compuesta por una propiedad (background) y su valor (red)



Atención:

Las instrucciones de CSS terminan con un punto y coma; si lo omites, las instrucciones siguientes no funcionarán. De todas formas, si lo omites, el editor de código te indicará que hay un error en la siguiente instrucción.

Tipos de selectores

En CSS existen diversos tipos de selectores. De momento vamos a ver los siguientes:

1. Selector por etiqueta.
2. Selector por clase.
3. Selector por id (estos últimos no los utilizaremos en el curso).

El **selector por etiqueta** selecciona el elemento según su nombre de etiqueta, mientras que el **selector por clase** selecciona todos los elementos con el mismo valor del atributo de clase.

Las clases sirven para que varias etiquetas comparten características. Por ejemplo, si queremos que algunos textos sean de color verde, entonces a esos párrafos les agregaremos la clase “verde”.

```
<h1 class="verde">Descubre lo último en tecnología bebiendo café</h1>
<h4 class="verde">Rafaela Valdés</h4>
<h2 class="verde">Eventos anteriores</h2>
```

Para seleccionar esa clase en específico en el CSS, debemos escribir un **punto (.)** seguido por el valor de la clase:

```
.verde {
    /* Soy un comentario de CSS, no confundirse con comentarios de HTML. */
    color: green;
}
```



Las etiquetas pueden tener más de una clase, las cuales se escriben separadas por un espacio:

```
<h1 class="verde centrado">Descubre lo último en tecnología bebiendo café</h1>
```

Además del selector por etiqueta y del selector por clase, existe el **selector por id**, que utiliza el valor del atributo id de una etiqueta HTML para seleccionar un elemento específico.



El valor id de un elemento debe ser único, no puede repetirse en ninguna otra etiqueta dentro de la página. Además, cada etiqueta puede tener como máximo un solo id:

```
<h2 id="subtitulo">Charlas, eventos y simposios sobre tecnología</h2>
```

A lo largo de este módulo, trataremos de utilizar clases por sobre los id, porque la etiqueta es menos específica que la clase, mientras que la clase es menos específica que el id.

Profundizando en colores

En esta sección profundizaremos en las distintas formas de agregar un color.

Aquí es muy importante entender que no es necesario memorizarlas, sino que tener conocimiento de su existencia para poder identificar los formatos cuando copiemos un color para nuestra página.

Colores con palabra clave

En CSS existen 17 colores básicos definidos como palabras clave y 147 colores extendidos también definidos con sus palabras claves para referirse a colores básicos.

Las palabras se corresponden con el nombre en inglés de cada color, y el que sean básicos o extendidos no tiene relevancia para su uso.



Imagen 4. 17 Colores básicos.

Fuente: <https://www.w3.org/TR/css-color-3/>

Un ejemplo de esto es:

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

RGB numérico

RGB significa red, green and blue, o sea rojo, verde y azul, y permite especificar colores de esta forma flexible, dándonos gran cantidad de combinaciones.

Aquí el color se define indicando sus tres componentes R (rojo), G (verde) y B (azul), donde cada uno puede tomar un valor entre 0 y 255, siendo cero nada de ese color, y 255 todo ese color.

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

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

RGB porcentual

Es similar a RGB numérico, donde 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*/
```

RGB hexadecimal

Similar a los anteriores, en hexadecimal los números van de 0 a 255, en una combinación que se expresa en valores que van de 0 a 15, pero donde el 10 se escribe con la letra a, el 11 con la letra b y así sucesivamente hasta la letra f.

De esta forma, con un par de estas combinaciones podemos escribir cualquier valor de 0 a 255 y cada uno de estos pares representa un color.

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

Utilizar códigos hexadecimales no es algo que debamos saber de memoria y fácilmente podemos convertir un valor numérico a hexadecimal ocupando Google u alguna de muchas herramientas online.



Cuando ambos valores son iguales, por ejemplo 00, se puede escribir como un único valor; de esta forma, en lugar de escribir #FF0000, podemos escribir F00, o en lugar de escribir #000000, podemos escribirlo como #000.

Tipografías

Podemos cambiar la tipografía utilizando la propiedad **font-family**, por ejemplo:

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

Tipografías soportadas en todos los navegadores

Es importante tener en cuenta que no todos los navegadores son compatibles con todas las tipografías disponibles. Sin embargo, existe un conjunto de fuentes seguras conocido como "safe fonts" que pueden ser utilizadas en diferentes navegadores.

Estas fuentes son:

- Arial (sans-serif)
- Verdana (sans-serif)
- Helvetica (sans-serif)
- Tahoma (sans-serif)
- Trebuchet MS (sans-serif)
- Times New Roman (serif)
- Georgia (serif)
- Garamond (serif)
- Courier New (monospace)
- Brush Script MT (cursive)

También es posible importar las fuentes utilizando CSS para disponer de una mayor variedad,

Estrategia de fallback

Es posible definir varias fuentes para un mismo selector, con el objetivo de asegurar que el estilo deseado para el texto se muestre correctamente en el navegador. Si una de las fuentes no es compatible, el navegador puede utilizar la siguiente en la lista como alternativa.

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

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

En este ejemplo, el navegador intentará mostrar la página con la fuente “Montserrat”. Si el usuario no tuviera cargada esta fuente o si el navegador no fuera compatible, probará con “Times New Roman”, y finalmente, con la familia tipográfica “Fantasy”. En el caso de que no encontrara ninguna de ellas, mostrará el sitio con la fuente por defecto del navegador.

Unidades de medida y tamaños

El tamaño de fuente se define con la propiedad **font-size**, por ejemplo:

```
p { font-size: 20px; }
```



Los navegadores de manera predeterminada definen un font-size de 16px al elemento p. Entre mayor el número más grande se verá la letra del elemento.

El problema con PX

Para asignar tamaños a las fuentes no se recomienda utilizar los píxeles (px) pues estos son rígidos.

¿Por qué son rígidos?

Imaginemos que tenemos una hoja de estilo en la que se han definido tamaños distintos para cada elemento. De repente, cambiamos de opinión sobre el tamaño base del elemento padre y queremos que sea un poco más grande. En este caso, tendríamos que cambiar manualmente cada uno de los tamaños de los demás elementos. Probablemente, también tendríamos que ajustar muchos márgenes y otros elementos, lo que podría resultar en un gran trabajo. Desde esta perspectiva, se dice que las unidades absolutas son difíciles de mantener, ya que su uso puede dificultar la realización de cambios futuros

¿La solución?

Utilizar otro tipo de unidades llamadas “relativas”, cuyo tamaño depende de otro elemento.

em y rem

- em: La unidad “em” no siempre mide lo mismo , ya que su tamaño depende del tamaño del elemento padre. Por lo tanto, se trata de una unidad relativa.

Por ejemplo, si el elemento “body” mide 1em y un párrafo dentro de “body” mide 2em, el tamaño final será de 2em. Sin embargo, si definimos el tamaño base de “body” como 2em y un párrafo dentro de body como 2em, el tamaño final será equivalente a 4em.

Es importante tener en cuenta que, por defecto, 1em mide 16px. Para ilustrar esto, veamos el siguiente ejemplo:

```
<body style="font-size: 2em;">
  <p style="font-size: 2em"> Párrafo 1. </p>
  <p style="font-size: 64px;"> Párrafo 2</p>
</body>
```

Si visualizamos esto en el navegador, veremos que ambos párrafos tienen el mismo tamaño. Esto se debe a que 1em base mide 16px, y dentro de “body” hemos definido que el tamaño es de 2em, lo que equivale a $2 \times 16 = 32$ px. Luego, el primer párrafo mide 2em, lo que equivale al doble de su padre, es decir, $2 \times 32 = 64$ px.

Sin embargo, es importante tener en cuenta que no es recomendable combinar px con em para definir tamaños de fuentes. Por lo general, utilizaremos px para definir, por ejemplo, el ancho de un borde, y unidades relativas para el tamaño de tipografías.

- rem: La unidad rem (root em) es relativa solo al tamaño del elemento raíz. Si el elemento HTML tiene un tamaño de fuente de 16px (como es por defecto), entonces 1rem sería igual a 16px sin importar el tamaño del elemento padre.

Veamos un ejemplo:

```
<ul style="font-size: 24px">
  <li> Elemento en px</li>
</ul>
<ul style="font-size: 1.5rem">
  <li> Elemento en px
    <ul style="font-size: 1.5rem">
      <li> Subelemento en rem </li>
      <li> Subelemento en rem </li>
    </ul>
  </li>
</ul>
```

Al abrir la página en el navegador deberíamos ver lo siguiente:

- Elemento en px
- Elemento en px
 - Subelemento en rem
 - Subelemento en rem

Imagen 5. Screenshot de experimento con rem

Fuente: [w3.org](https://www.w3.org).

¿Qué sucedió? A diferencia de las unidades “em”, las unidades “rem” de un elemento hijo no se multiplican con las del elemento padre, ya que solo dependen del elemento raíz. Por lo tanto, en este caso 1.5 rem será equivalente a 24px ($1.5 \times 16 = 24$).

Para seguir explorando prueba los siguientes casos:



- Realiza el mismo ejercicio de lista y sublista de la imagen 5, cambiando las unidades de “rem” por “em” y observa los resultados.
- Vuelve a poner las unidades en “rem”, pero además agrega al body un tamaño de tipografía (font-size) de 32px y observa los resultados.

Porcentajes

Los porcentajes son una excelente unidad de medida para definir el tamaño de las cajas (div) en relación con la página. De esta manera, podemos crear una columna que ocupe la mitad del ancho de la página (50%), o definir que nuestro sitio web tendrá una única columna central que abarque el 80% del ancho total del sitio.

En un proyecto nuevo probemos el siguiente código:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
<style>
body{
  width: 80%;
  margin: auto;
}
</style>
</head>
<body>
<h1>Título ridículamente largo para demostrar el margen máximo de la página incluso en un computador con pantalla gigante</h1>
<p> Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolores voluptatum exercitationem consectetur ducimus distinctio itaque ipsum? Veniam cumque non alias eligendi doloribus ipsam. Obcaecati non asperiores repellat, consequatur nulla repellendus!</p>
</body>
</html>
```

Al abrir la página con el navegador y seleccionar body con el inspector de elementos veremos lo siguiente:

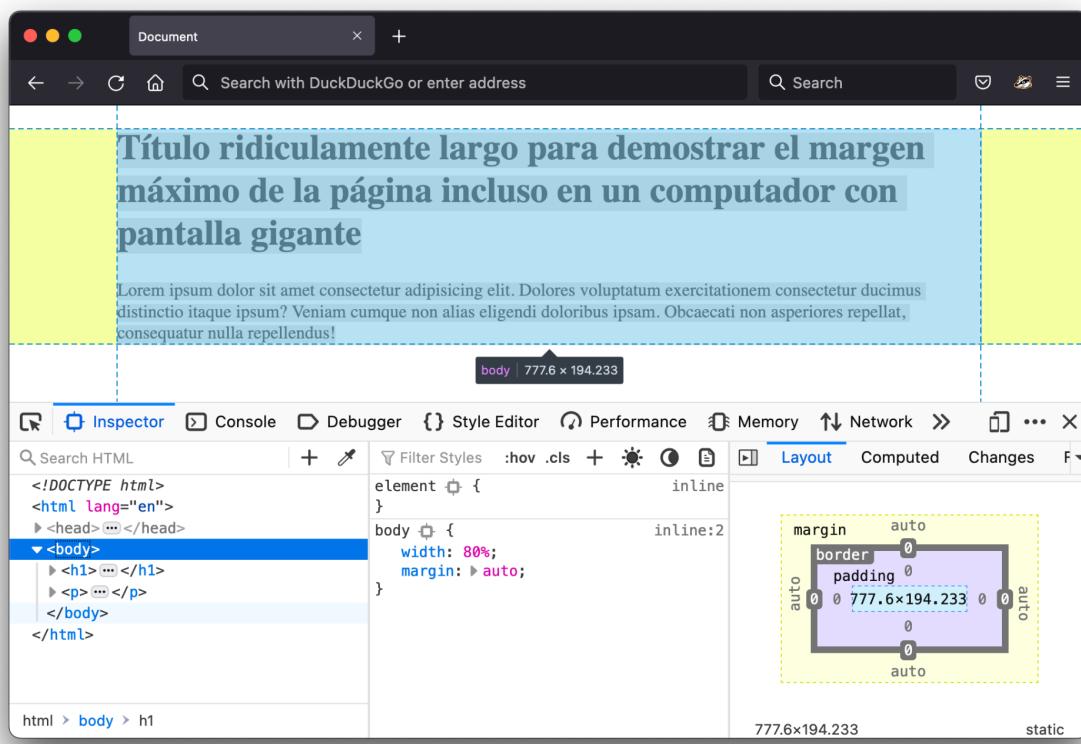


Imagen 6. Screenshot de experimento de margin sobre el body

Fuente: w3.org.

En la imagen podemos ver el margen en amarillo. Como experimento, reduciremos el ancho a 60%, y luego probaremos quitando el `margin: auto` y asignando manualmente alguna medida con "px", "%" u otro tipo de valor a gusto.

Ahora, observa qué sucede si aplicamos al body lo siguiente:

```
body{  
    width: 80%;  
    margin: 0 0% 0 20%;  
}
```

Grosor y cursiva

El grosor de la letra (bold) se puede controlar con la propiedad **font-weight**, mientras que el estilo de la letra (italic o cursiva) se controla con la propiedad **font-style**.

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 quieras saber más sobre estas propiedades, puedes consultar el material disponible en lenguajecss.com



No hay problema si cuando crees tus primeros sitios utilices como unidad de medida px, pero a medida que vayas creando más sitios y estos sean más complejos y con mayor cantidad de elementos, te darás cuenta de que te será más cómodo trabajar con em y rem.

Fondo

Existen diferentes propiedades relacionadas al estilo del fondo (background) de nuestra página. Algunas de ellas son:

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

background-color

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

```
body {  
    background-color: #ffff00;  
}
```

background-image

Permite establecer una imagen de fondo, cuyo 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 de fondo.

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

background-repeat

Cuando una imagen de fondo no cubre completamente todo el fondo una página, el parámetro “background-repeat” indica si la imagen se repetirá o no, y en qué posición se colocará la imagen repetida.

```
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 sitio W3Schools.

Alineación del texto

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; }
```



Actividad 5: Aplicando lo aprendido en una tabla

En el siguiente ejercicio aplicaremos estilos a una tabla, ya que esto nos permitirá utilizar lo aprendido hasta ahora.

	Id	Nombre	Apellido	Email	Monto Compra
1	1	Maisie	Harner	mharner0@posterous.com	3171
2	2	Steffie	Arnoll	sarnoll1@webeden.co.uk	2097
3	3	Twyla	Kobke	tkobke2@mayoclinic.com	4139
4	4	Tadd	Prendeville	tprendeville3@dyndns.org	3518
5	5	Evanne	Baulch	ebaulch4@salon.com	5963
6	6	Amby	Guild	aguild5@cpanel.net	7835
7	7	Gar	Chalkly	gchalkly6@canalblog.com	7025
8	8	Dani	Costy	dcosty7@spiegel.de	8790
9	9	Clovis	Kail	ckail8@ning.com	6961
10	10	Thalia	Slessar	tslessar9@discovery.com	2100
11	11	Ansel	Picard	apicarda@joomla.org	7863
12	12	Briny	Colleck	bcolleckb@example.com	1031
13	13	Gretal	Asbrey	gasbreye@bandcamp.com	7179
14	14	Mareah	Turone	mturoned@time.com	1808
15	15	Erek	Callander	ecallandere@xrea.com	6909
16	16	Edwin	Cash	ecashf@fc2.com	3906
17	17	Clyve	Jancso	cjancsog@ucla.edu	4716
18	18	Harmony	Southworth	hsouthworth@mapquest.com	6361
19	19	Lacie	Delap	ldelapi@bing.com	8251
20	20	Lurline	Dunabie	ldunabiej@npr.org	7023
21	21	Brantley	Kitchen	bkitchenk@umich.edu	6207
22	22	Xena	Cockren	xcockrel@salon.com	1891
23	23	Giralda	McCarty	gmccartym@pbs.org	1822
24	24	Emlynn	Danielovitch	edanielovitchn@auda.org.au	8871

	Id	Nombre	Apellido	Email	Monto Compra
1	1	Maisie	Harner	mharner0@posterous.com	3171
2	2	Steffie	Arnoll	sarnoll1@webeden.co.uk	2097
3	3	Twyla	Kobke	tkobke2@mayoclinic.com	4139
4	4	Tadd	Prendeville	tprendeville3@dyndns.org	3518
5	5	Evanne	Baulch	ebaulch4@salon.com	5963
6	6	Amby	Guild	aguild5@cpanel.net	7835
7	7	Gar	Chalkly	gchalkly6@canalblog.com	7025
8	8	Dani	Costy	dcosty7@spiegel.de	8790
9	9	Clovis	Kail	ckail8@ning.com	6961
10	10	Thalia	Slessar	tslessar9@discovery.com	2100
11	11	Ansel	Picard	apicarda@joomla.org	7863
12	12	Briny	Colleck	bcolleckb@example.com	1031
13	13	Gretal	Asbrey	gasbreye@bandcamp.com	7179
14	14	Mareah	Turone	mturoned@time.com	1808
15	15	Erek	Callander	ecallandere@xrea.com	6909
16	16	Edwin	Cash	ecashf@fc2.com	3906
17	17	Clyve	Jancso	cjancsog@ucla.edu	4716
18	18	Harmony	Southworth	hsouthworth@mapquest.com	6361
19	19	Lacie	Delap	ldelapi@bing.com	8251
20	20	Lurline	Dunabie	ldunabiej@npr.org	7023
21	21	Brantley	Kitchen	bkitchenk@umich.edu	6207

Imagen 7. Aplicando estilo a una tabla

Fuente: Desafío Latam

Para obtener la tabla, debes ir a <https://codepen.io/gsanchezd/pen/rNzKyyB>

En este sitio podrás aplicar el CSS directamente, o también puedes copiar el html en tu editor de texto.

Paso 1: Centrar la tabla

```
table{  
    width: 50%;  
    margin: auto;  
}
```

Paso 2: Aplicar bordes

1. Intento 1: Aplicar bordes solo a la tabla.

```
table {  
    width: 50%;  
    margin: auto;  
    border: solid 1px #ddd;  
}
```

2. Intento 2: Aplicar bordes a la tabla y a cada celda.

```
table {  
    width: 50%;  
    margin: auto;  
}  
  
table, th, td {  
    border: 1px solid #ddd;  
}
```

3. Intento 3: Aplicar bordes a la tabla y a cada celda, y además colapsar los bordes.

```
table {  
    width: 50%;  
    margin: auto;  
}  
  
table, th, td {  
    border: 1px solid #ddd;  
    border-collapse: collapse;  
}
```

4. Intento 4: Aplicar bordes a la tabla y a cada celda colapsando los bordes, pero aplicando solo borde en la caja inferior.

```
table {  
    width: 50%;  
    margin: auto;  
}  
  
table, th, td {  
    border-bottom: 1px solid #ddd;  
    border-collapse: collapse;  
}
```

Lo logramos 

Paso 3: Aplicando color de fondo

Aplicaremos color al encabezado de la tabla (recordemos que la etiqueta “thead” es la etiqueta padre de las celdas correspondientes a la cabecera de una tabla).

```
table {  
    width: 50%;  
    margin: auto;  
}  
  
table, th, td { /* Se pueden asignar valores a más de una propiedad simultáneamente */  
    border-bottom: 1px solid #ddd;  
    border-collapse: collapse;  
}  
  
thead {  
    background-color: cornflowerblue;  
    color: white;  
    font-weight: 800;  
}
```

Paso 4: Alineando el texto

```
td {  
    text-align: right;  
}
```

Paso 5: Aplicando relleno

```
th {  
    padding: 0.5em;  
}  
  
td {  
    text-align: right;  
    padding: 0.2em;  
}
```

Modelo de cajas

El modelo de cajas es uno de los conceptos más importantes detrás de CSS, y se refiere a cómo se representa visualmente un elemento HTML en la página.

En CSS cada elemento se considera como una “caja” rectangular que contiene contenido, bordes, rellenos y márgenes.



Imagen 8. Modelo de cajas.
Fuente: Desafío Latam.

Propiedades del modelo de cajas

Primero, vamos a analizar el **área de contenido**, que es lo que está comprendido dentro del elemento, como por ejemplo, textos e imágenes.

Para editar esta área podemos agregar o quitar contenido, así como también editarlo, asignándole un **ancho (width)** y/o un **alto (height)**. Esto será muy útil para ajustar imágenes a nuestro sitio web.

Además del área de contenido existe el **padding**, que se utiliza para generar espacio alrededor del contenido, dentro del borde del elemento.



El padding es siempre transparente.

A continuación, podemos observar el **borde (border)**, el cual contiene al contenido y el padding del elemento. Las propiedades de borde de CSS permiten especificar el estilo, ancho y color del borde de cada elemento. Por defecto, en la mayoría de las etiquetas el valor de la propiedad de borde es **none (border: none;)**.

Por último, el **margen (margin)** es la propiedad que define los márgenes del elemento. En otras palabras, es el espacio que rodea al elemento y lo separa de otros elementos cercanos. Esta propiedad puede tener de uno a cuatro valores, los cuales permitirán establecer márgenes en los 4 lados del elemento

Reseteando las propiedades

Es importante tener en consideración que existen propiedades por defecto asociadas a cada navegador, las cuales pueden afectar la forma en que se muestran los estilos que hemos aplicado a nuestra página.

Para evitar que estos estilos interfieran, se suele dar a las propiedades generales un valor por defecto, antes de comenzar a añadir nuestros propios estilos. Esta técnica se llama “reset CSS”.

Un ejemplo de esto, sería lo siguiente:

```
* {  
    margin: 0;  
    border: 0;  
    padding: 0;  
}
```

El * es un selector CSS universal que aplica a todas las etiquetas. Es decir, en este ejemplo le estamos dando margen, borde y relleno 0 a todas las etiquetas de nuestra página. Si al selector * le agregamos la palabra “body”, significa que estaremos alterando todas las etiquetas que estén dentro de body.

Podemos lograr una versión mejorada del script anterior utilizando un archivo de reseteo o un archivo normalizador. Ambos son distintas estrategias para que el sitio web se vea igual en cada uno de los distintos navegadores. No profundizaremos en las diferencias de estas estrategias, pero adjuntamos links a las hojas de estilo para que las pruebes.



Reset: <https://meyerweb.com/eric/tools/css/reset/>

Normalizador.:

<https://github.com/necolas/normalize.css/blob/master/normalize.css>

Descárgalos (o copia en un archivo nuevo), y agrega un link desde el index.html. Cuando incorporas muchos archivos CSS, el orden importante: el reset o normalizador debería ser el primero para que luego puedas reescribir las reglas (recuerda que la última regla manda).

Antes de continuar:

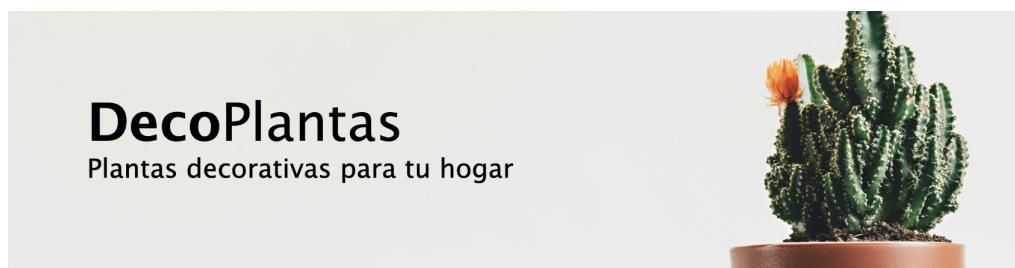
- ¿Reconoces la diferencia entre los 3 códigos de color RGB?
- ¿Conocías las propiedades del texto que te presentamos? ¿Cuál de ellas te costó más comprender?
- ¿Conoces alguna otra propiedad de estilo de fondo? ¿Cuál?

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.

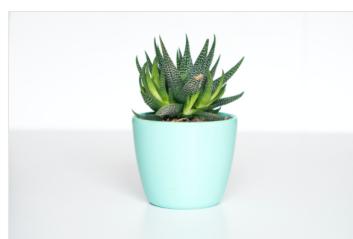


Implementando lo aprendido

La siguiente implementación la dividiremos en pasos. Estos no necesariamente se tienen que hacer en el orden establecido, pero la división busca que puedas identificar fácilmente si te saltaste alguno durante la implementación.



Planta 1
Precio: \$9.900



Planta 2
Precio: \$12.700



Planta 3
Precio: \$11.950

@decoplantas

Imagen 9. Sitio a construir.
Fuente: www.type-scale.com

Paso 1: Identificar las secciones

Vamos a construir el sitio web presentando en la imagen. Lo primero es distinguir las secciones que en este caso son 3.

Una forma fácil de distinguirlas es con base en la diferencia de fondos, ya sea por color o imagen, sin embargo, lo que realmente las hace distintas es el tipo de contenido que muestran.



Imagen 10. Sitio a construir: secciones.

Fuente: www.type-scale.com

1. En la primera sección vemos la propuesta principal, de que trata el sitio.
2. La segunda sección corresponde a los productos, en donde mostramos imágenes y precios
3. Finalmente, tenemos un pie de página o footer.

Paso 2: Descargar las fotos

Para crear el sitio necesitaremos las imágenes, las cuales podemos descargar desde la plataforma o desde algún banco de imágenes. Recomendamos realizar este ejercicio descargando las fotos de la plataforma para que puedas seguir los pasos sin problemas.

Paso 3: Crear la carpeta del proyecto y el archivo index.html

Con las fotos descargadas, lo primero que haremos es crear una carpeta para el proyecto y dentro de la carpeta el archivo `index.html`

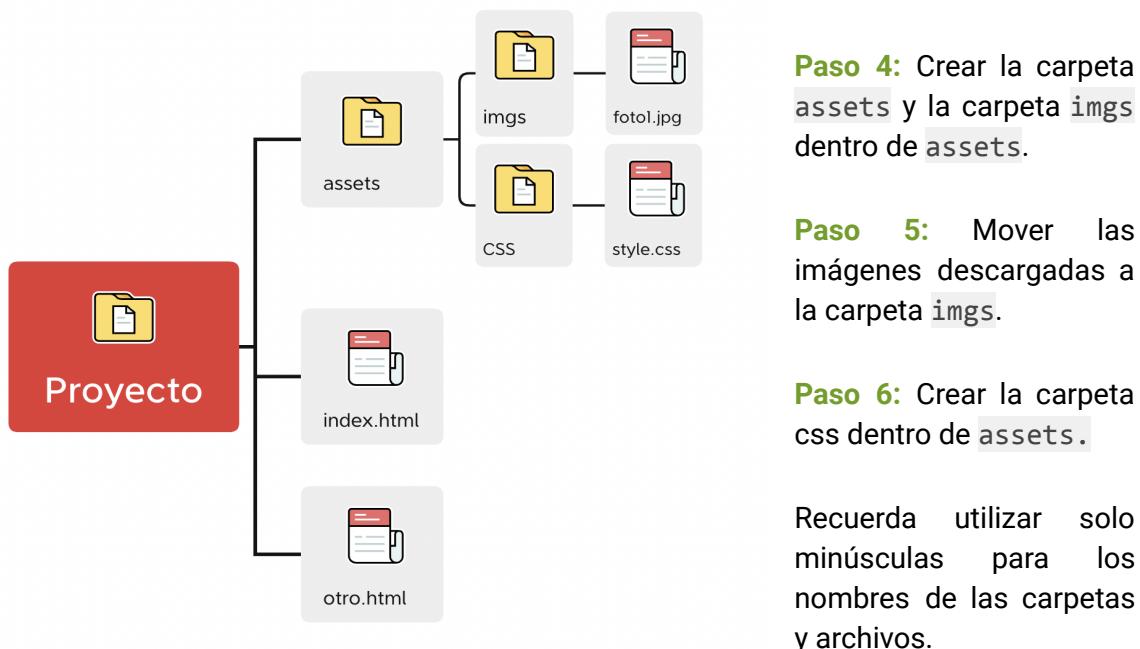


Imagen 11. Sitio a construir: crear carpeta.

Fuente: Desafío Latam.

Paso 7: Dentro de la carpeta `css` crearemos el archivo `normalize.css` y agregaremos el css copiado desde <https://necolas.github.io/normalize.css/>

Paso 8: Agregaremos la estructura del html y cambiaremos el título del sitio.

```
<!--index.html-->
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>DecoPlantas</title>
</head>
<body>
</body>
</html>
```

Paso 9: Agregamos el CSS para resetear las propiedades

```
<!--index.html-->
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link rel="stylesheet" href="assets/css/normalize.css">
  <title>DecoPlantas</title>
</head>
<body>
</body>
</html>
```

Revisemos con el inspector de elementos que el CSS se haya cargado. En firefox, el tab de style editor te ayudará a detectar eso.

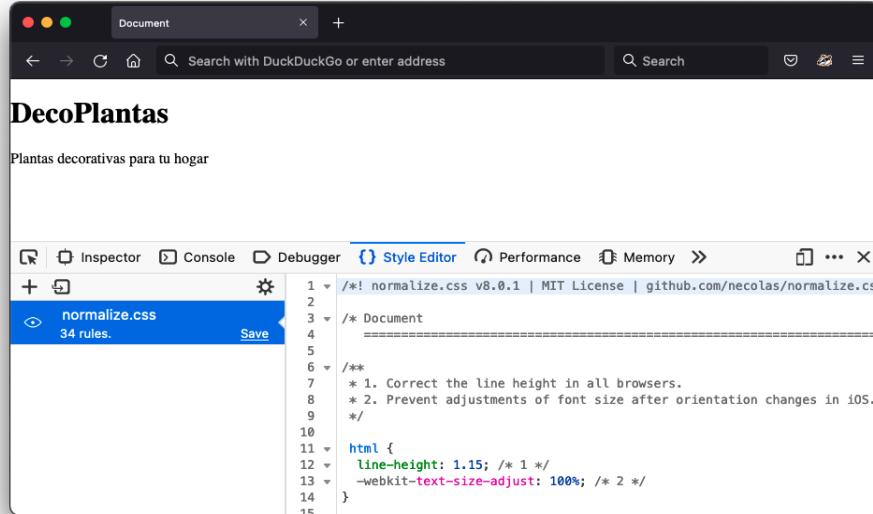


Imagen 12. Inspector de elementos.

Fuente: Desafío Latam.

También es posible detectar las propiedades cambiadas utilizando el inspector de elementos, seleccionando sobre decoplantas, ahí veremos algunas propiedades agregadas.

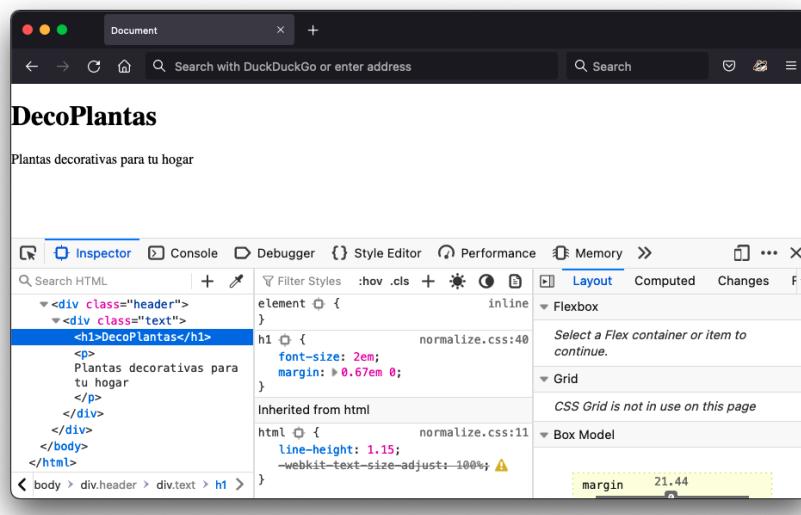


Imagen 13. Revisión de pasos.

Fuente: Desafío Latam.

En caso de no poder observarlas, revisa que hayas ingresado el texto del paso 9 y que el archivo exista dentro de la carpeta assets/css

Paso 10: Agregando el contenido base al body

Dentro del archivo `index.html` utilizaremos el autocompletado de VSCode para dar la estructura base. Luego, en “body” agregaremos el siguiente código:

```
<!-- index.html -->
<div class="header">
  <h1>DecoPlantas</h1>
  <p>Plantas decorativas para tu hogar</p>
</div>
<div class="products"></div>
<div class="footer">@decoplantas</div>
```

Cada una de las clases en el código html corresponde a una de las secciones: la clase `header` será la sección principal, la clase `products` la sección de productos y la clase `footer` serán el pie de página.

Paso 11: Crear el archivo style.css

A continuación, tenemos que crear nuestro archivo `style.css` en `assets/css`. Dentro de este archivo escribiremos el CSS para la primera sección. Específicamente, tenemos que agregar la imagen de fondo y el alto.

```
/* style.css */
.header {
  height: 20em;
  background-image: url("../imgs/fondo.jpg");
  background-size: cover;
  background-position: center;
}
```

Paso 12: Agregar el estilo creado al archivo index.html

```
<!-- index.html -->
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Document</title>
  <link rel="stylesheet" href="assets/css/normalize.css">
  <link rel="stylesheet" href="assets/css/style.css">
</head>
<body>
  <div class="header">
    <h1>DecoPlantas</h1>
    <p>Plantas decorativas para tu hogar</p>
  </div>
</body>
</html>
```

Recuerda que el orden de carga de los estilos importa

 Es importante que el archivo de estilo sea agregado después del normalizador, porque si no, el normalizador podría deshacer algunas propiedades específicas que queramos agregar al documento.

Al abrir la página en el navegador deberíamos ver lo siguiente:

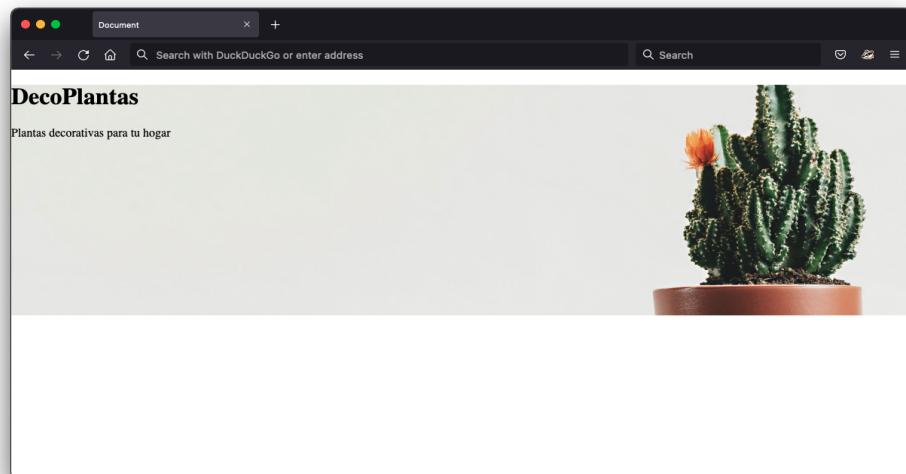


Imagen 15. Agregar estilos.

Fuente: Desafío Latam.

Paso 13: Remover el margen superior

Con el inspector de elementos podemos observar que el elemento h1 es el responsable del margen superior, por lo que lo quitaremos agregando lo siguiente:

```
.header h1 {  
    margin: 0px;  
}
```

¿Qué significa .header h1?

Esto se lee como: "todos los h1 dentro de un elemento con clase header". Es decir, la propiedad CSS y su valor solo aplicará para los elementos dentro de header y no afectará a los de afuera. Esta forma de escribir CSS será utilizada de manera frecuente, ya que con esto evitamos crear reglas para elementos que no las necesitan.

Paso 14: Posicionando el texto

Hay varias formas de posicionar el texto, pero por ahora lo haremos solo utilizando las reglas que ya conocemos. Para eso, crearemos una caja interna y le daremos margin o padding, de manera que nuestro CSS quedaría así:

```
body {  
    font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode',  
    Geneva, Verdana, sans-serif;  
}  
  
.header{  
    height: 10em;  
    padding: 5em;  
    font-size: 2em;  
    background-image: url("../imgs/fondo.jpg");  
    background-size: cover;  
    background-position: center;  
}  
  
.header h1{  
    margin: 0px;  
}
```

Como puedes observar, redujimos el alto de la caja, de forma que ahora la suma del padding superior, el padding inferior y el alto sigue siendo equivalente a 20em. Por ahora no es necesario hacer cálculos, lo importante es que te guste visualmente el resultado. Puedes explorar con el inspector de elementos cambiando los valores una vez terminado el ejercicio.

Paso 15: Destacando el Deco

Para destacar la palabra deco cambiaremos el font-weight del h1 a 400 y utilizaremos la etiqueta strong sobre ella.

```
<div class="header">
  <h1><strong>Deco</strong>Plantas</h1>
  <p>Plantas decorativas para tu hogar</p>
</div>
```

```
body {
  font-family: "Lucida Sans", "Lucida Sans Regular", "Lucida Grande",
  "Lucida Sans Unicode", Geneva, Verdana, sans-serif;
}

.header {
  height: 10em;
  padding: 5em;
  font-size: 2em;
  background-image: url("../imgs/fondo.jpg");
  background-size: cover;
  background-position: center;
}

.header h1 {
  margin: 0px;
  font-weight: 400;
}
```

Paso 16: Agregando el html de los productos

Para agregar los productos utilizaremos una tabla. Si bien en la próxima unidad aprenderemos otras formas de agregar imágenes al lado de texto, por ahora, usando una tabla podremos cumplir con el requerimiento.

```
<div class="products">
  <table>
    <tr>
      <td>
        
      </td>
      <td>
        <p>Planta 1</p>
        <p>Precio: $9.900</p>
      </td>
    </tr>

    <tr>
      <td>
        
      </td>
      <td>
        <p>Planta 2</p>
        <p>Precio: $12.700</p>
      </td>
    </tr>

    <tr>
      <td>
        
      </td>
      <td>
        <p>Planta 3</p>
        <p>Precio: $11.950</p>
      </td>
    </tr>
  </table>
</div>
```

Paso 17: Agregando el CSS de los productos

```
/* Agregar a style.css */  
.products {  
width: 80%;  
margin: auto;  
}  
  
.products td {  
width: 50%;  
padding: 2em;  
}  
  
.products img {  
width: 100%;  
border: solid 1px #ddd;  
}
```

Revisemos el código:

1. Centramos la sección de productos dejando un margen de 10% a cada lado.
2. Cada celda de la tabla va a tener un máximo de 50%, lo que se calcula respecto al contenedor padre. Como la caja de productos tiene un ancho de 80%, en la práctica cada celda tendrá un 50% de ese 80%.
3. Especificamos que las imágenes tendrán un ancho del 100%, pero nuevamente esto es respecto al tamaño del contenedor padre. Si la imagen es de distinto tamaño al contenedor donde fue puesta, su tamaño se ajusta.

Paso 18: Agregamos el html del footer al index.html (solo si no lo hemos hecho todavía)

```
<div class="footer">@decoplantas</div>
```

Paso 19: Agregamos el CSS del footer

```
.footer {  
height: 3em;  
line-height: 3em;  
background-color: rgb(185, 112, 41);  
color: white;  
text-align: center;  
font-weight: 900;  
}
```

Cuando utilizamos height y line-height en conjunto y con el mismo valor, podemos centrar verticalmente un texto de una sola línea.



Actividad 6: Revisando los aprendizajes

- ¿Qué línea de código deberíamos utilizar para integrar un CSS que esté en la subcarpeta estilos?
- ¿Cuál es el selector universal o comodín que permite seleccionar todos los elementos? Si no lo recuerdas, revisa dentro de la sección modelo de caja.
- ¿Qué propiedad y valor deberíamos ocupar para asignar margen solo a la derecha y arriba?
- En el siguiente CSS, ¿cuál es el error?

```
.header{  
width: 100%  
height: 20em  
}
```

- ¿Para tipografías es mejor ocupar px o em?

Revisa que puedes hacer las siguientes acciones

Recuerda que no es necesario poder hacerlas de memoria, pero deberías practicarlas hasta que puedas lograr implementarlas de forma rápida:

- Utilizar el inspector de elementos para detectar si un archivo CSS fue cargado.
- Utilizar el inspector de elementos para cambiar propiedades y valores en un archivo CSS.
- Cambiar el color de un texto utilizando palabras claves o código rgb en cualquiera de sus formas.
- Cambiar el color de fondo de una celda en una tabla.
- Cambiar el margin, padding y border de un elemento.
- Crear un div y agregar una imagen de fondo o un color de fondo.
- Centrar un texto.
- Agregar una imagen de fondo a una sección y utilizar las propiedades y valores correspondientes para ajustar la imagen.
- Ajustar el texto dentro de una sección utilizando padding.

¡Continúa aprendiendo y practicando!

