

# Introducción a CSS

## Introducción

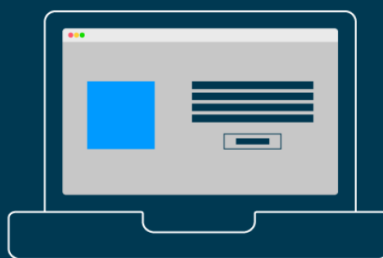
¡Hola, de nuevo! 🤝 En el encuentro anterior, aprendimos sobre los conceptos básicos de HTML y creamos una página web simple con encabezados y párrafos. Hoy aprenderemos sobre CSS, que significa "Cascading Style Sheets" ("Hojas de estilo en cascada").

**CSS es un lenguaje que se utiliza para dar estilo a los HTML con elementos como colores, fuentes y espaciado** para generar páginas web visualmente atractivas. Separar el estilo de la estructura de una página web hace que sea más fácil mantener y actualizar el contenido.

Sin CSS



Con CSS



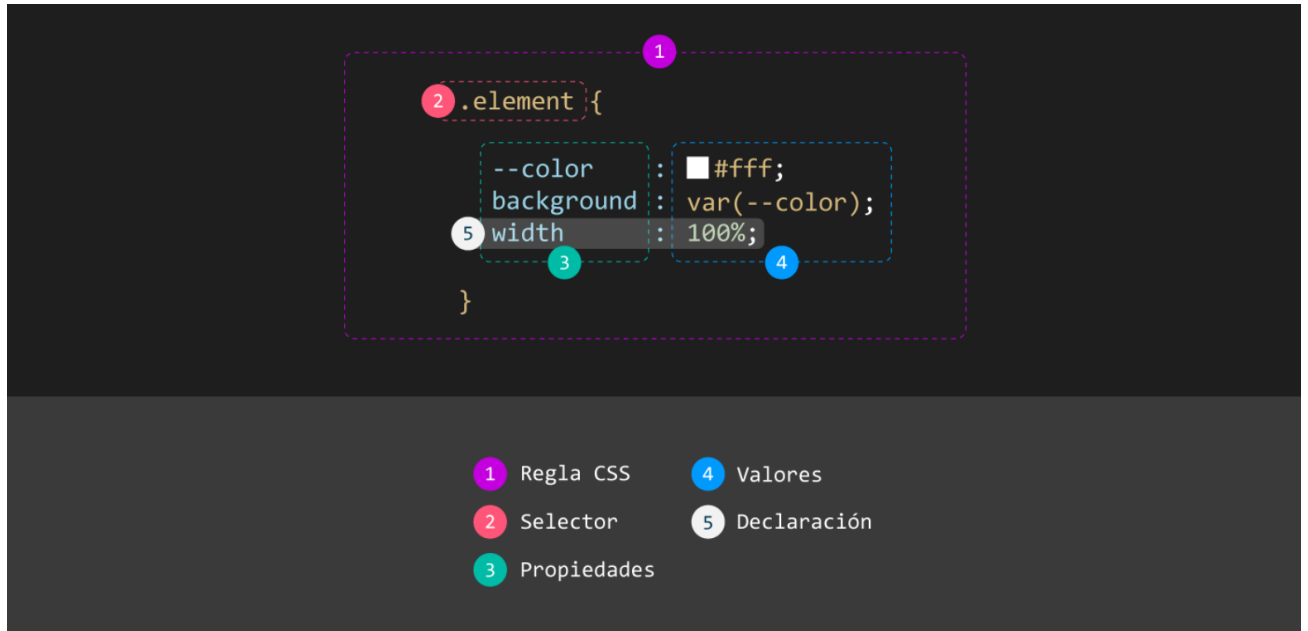
A continuación, cubriremos los conceptos básicos de CSS incluyendo los tipos, selectores y propiedades, y luego aplicaremos CSS a la página HTML que creamos en la clase anterior.

¡Allá vamos!

# Sintaxis de CSS

La **sintaxis de CSS** es fundamental para darle el estilo deseado a una página web porque permite definir las propiedades que determinarán la apariencia visual de los elementos del documento HTML.

Veamos un ejemplo de cómo se estructura la sintaxis:



A tener en cuenta sobre las **declaraciones CSS**:

- **Tienen un formato específico de "propiedad: valor;"** (diferente al formato de igualdad usado en los atributos HTML).

Si queremos, por ejemplo, aplicar el estilo de color rojo y un tamaño de fuente de 18px a un título que se encuentra dentro de una etiqueta h1 en HTML, lo declararíamos en CSS de la siguiente manera:



Título de un sitio web

Con lo anterior, estamos seleccionando el elemento "h1" del HTML y aplicando las propiedades "color" y "font-size" con sus respectivos valores para cambiar su estilo en la página web.

- **No se ven afectadas por el espacio en blanco.** Las propiedades se pueden escribir de corrido o una debajo de la otra. Otra forma en la que podemos declarar el CSS anterior, por ejemplo, sería la siguiente:

```
h1{color: red; font-size: 18px;}
```

- **Al definir una propiedad numérica se debe especificar la unidad de medida que se está usando.** No se deben agregar espacios entre el número y la unidad de medida.  
Por ejemplo, si queremos definir el tamaño de letra en 18 píxeles, debemos escribir *"font-size: 18px;"* y no solo *"font-size: 18;"*.
- **No se ven afectadas por los comentarios.** En CSS, los comentarios se realizan con la sintaxis */\* comentario \*/*. Todo lo que se encuentra dentro de los

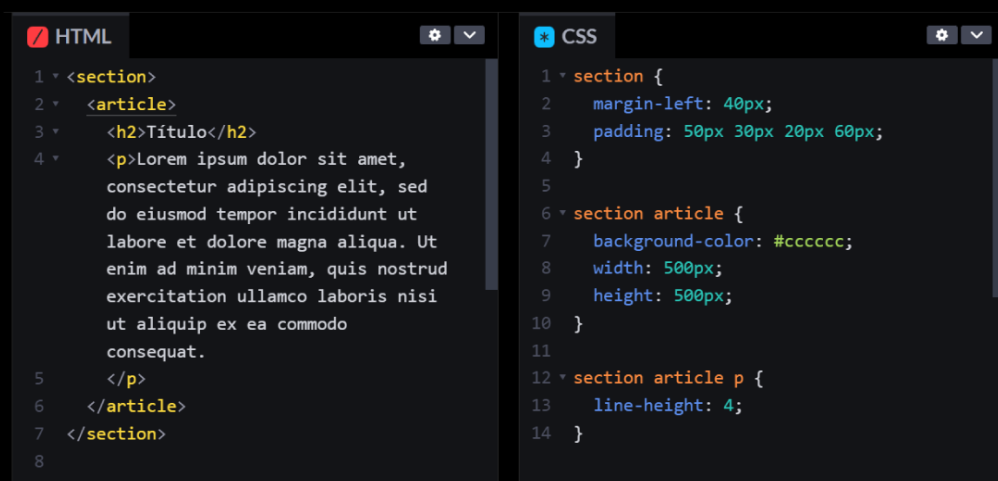
símbolos `/*` y `*/` será ignorado por el navegador, lo que permite agregar notas o aclaraciones en el código sin afectar su funcionamiento.

## Etiquetas “padres e hijos”

Dentro de la sintaxis del CSS, nos podemos encontrar con el concepto de “padres” e “hijos”. **Cuando una etiqueta está contenida dentro de otra, se utiliza el término “padre” para referirse a la primera e “hijo” para hacer referencia a la segunda.**

Esta relación permite agregar atributos específicos a los elementos “hijos”, sin afectar a los del “padre”. **Cada “padre” puede tener varios “hijos”, y todos ellos heredan las características del “padre”, pero también pueden tener características únicas.** Esta relación permite una mayor flexibilidad en la estructuración de los elementos en un documento o página web.

Ten presente que al seleccionar una etiqueta, debes incluir las etiquetas “padre/s” para que sean más específicas a la hora de aplicar estilos. Veamos un ejemplo:



```
HTML
1 <section>
2   <article>
3     <h2>Titulo</h2>
4     <p>Lorem ipsum dolor sit amet,
      consectetur adipiscing elit, sed
      do eiusmod tempor incididunt ut
      labore et dolore magna aliqua. Ut
      enim ad minim veniam, quis nostrud
      exercitation ullamco laboris nisi
      ut aliquip ex ea commodo
      consequat.
5   </p>
6 </article>
7 </section>
8

CSS
1 section {
2   margin-left: 40px;
3   padding: 50px 30px 20px 60px;
4 }
5
6 section article {
7   background-color: #cccccc;
8   width: 500px;
9   height: 500px;
10 }
11
12 section article p {
13   line-height: 4;
14 }
```


# Propiedades básicas de CSS

Las **propiedades de CSS** se utilizan para **definir estilos para los elementos HTML**. Estas son como las instrucciones de diseño que le damos a una página web para que se vea visualmente bonita y atractiva. Podemos cambiar el color del texto, el tamaño de la fuente, la imagen de fondo y muchos otros aspectos visuales.


En otras palabras, las propiedades CSS son una forma de hacer que nuestra página web se vea única y personalizada.

A continuación, te compartimos algunas de las propiedades de CSS más comunes:

- **color:** Se utiliza para establecer el color del texto dentro de un elemento HTML. El valor de la propiedad puede ser un nombre de color, un código hexadecimal o un valor RGB. Por ejemplo:

```
h1 {  
  color:  red; /*Establece el color del texto en rojo*/  
}
```

- **background-color:** Se utiliza para establecer el color de fondo de un elemento HTML. Como sucede con la propiedad color, el valor de la propiedad puede ser un nombre de color, un código hexadecimal o un valor RGB. Por ejemplo:

```
body {  
  background-color:  #f1f1f1; /*Establece el color de fondo en gris claro*/  
}
```

- **font-family:** Se utiliza para establecer la fuente para el texto dentro de un elemento HTML. El valor de la propiedad es una lista de nombres de fuentes separados por comas, y si la primera fuente no está disponible, se utilizará la siguiente en la lista. Por ejemplo:

```
p {  
  font-family: "Helvetica Neue", Arial, Helvetica, sans-serif;  
}
```

- **font-size:** Se utiliza para establecer el tamaño del texto dentro de un elemento HTML. El valor de la propiedad puede ser un número seguido de una unidad de medida como píxeles o em, o puede ser una palabra clave como small, medium o large. Por ejemplo:

```
h2 {  
  font-size: 24px; /*Establece el tamaño del texto en 24 píxeles*/  
}
```

- **margin:** Se utiliza para establecer el espacio entre el borde de un elemento y cualquier elemento adyacente. El valor de la propiedad puede ser un número seguido de una unidad de medida como píxeles o em, o puede ser una serie de valores separados por espacios para establecer diferentes márgenes para diferentes lados del elemento. Por ejemplo:

```
div {  
  margin: 10px; /*Establece un margen de 10 píxeles en todos los lados  
  del elemento*/  
}  
  
p {  
  margin: 5px 10px 15px 20px; /*Establece un margen de 5 píxeles arriba,  
  10 píxeles a la derecha, 15 píxeles abajo y 20 píxeles a la izquierda*/  
}
```

- **padding:** Se utiliza para establecer el espacio entre el contenido de un elemento y su borde. El valor de la propiedad puede ser un número seguido de una unidad de medida como píxeles o em, o puede ser una serie de valores

separados por espacios para establecer diferentes rellenos para diferentes lados del elemento. Por ejemplo:

```
div {  
  padding: 10px; /*Establece un relleno de 10 píxeles en todos los lados  
  del elemento*/  
}  
  
p {  
  padding: 5px 10px 15px 20px; /*Establece un relleno de 5 píxeles arriba,  
  10 píxeles a la derecha, 15 píxeles abajo y 20 píxeles a la izquierda*/  
}
```

Ahora, te proponemos realizar la siguiente actividad que te ayudará a comprender mejor las propiedades de CSS y cómo aplicarlas:

### ¡Manos a la obra!

1. Ingresar a la siguiente página → [www.codepen.io](https://www.codepen.io), y hacer clic en el botón de **“Start coding”** que se encuentra en la columna izquierda de la pantalla.
2. En la *pestaña de HTML*, colocar el siguiente código:

```
<html>  
<head>  
  <title>Ejemplo de CSS</title>  
</head>  
<body>  
  <h1>Introducción a CSS</h1>  
  <p>CSS es un lenguaje que se utiliza para dar estilo a los HTML con elementos como  
  colores, fuentes y espaciado para generar páginas web visualmente atractivas.</p>  
</body>  
</html>
```

3. En la pestaña de CSS, utilizar las siguientes propiedades para modificar el estilo de las etiquetas “h1” y “p”:
  - **color** (blue, red, green, pink, etc.)
  - **background-color** (black, orange, yellow, etc.)
  - **font-family** (arial, helvetica, etc.)

- **font-size** (medidas tales como 20px, 40px, etc.)
  - **margin** (medidas tales como 20px, 50px, 100px, etc.)
4. Experimentar con diferentes valores para cada propiedad para lograr un diseño que te guste.

Tanto el título como el párrafo te deberían quedar con: un **color de fuente**, un **color de fondo**, un **tipo y tamaño de letra** y un **margen** diferente al original.

Te mostramos un ejemplo de un posible diseño a modo de inspiración:

## Introducción a CSS

CSS es un lenguaje que se utiliza para dar estilo a los HTML con elementos como colores, fuentes y espaciado para generar páginas web visualmente atractivas.

¿Lo has logrado? ¡Muy bien! Sigamos...

## Selectores de CSS

Los **selectores de CSS** son fundamentales para aplicar estilos y diseños específicos a los elementos de una página web. **Estos selectores permiten una selección precisa y detallada de los elementos HTML que se desea estilizar.**

Hay varios tipos de selectores. Algunos de ellos son:

- **Selectores de ID:** Seleccionan elementos con un atributo de ID específico. *Cada elemento HTML puede tener un solo ID y este debe ser único en todo el documento HTML. Los selectores de ID son precedidos por un numeral (#), por ejemplo, **#header** o **#footer**.*
- **Selectores de clase:** Seleccionan elementos con un atributo de clase específico. *Los elementos HTML pueden tener múltiples clases, y estas pueden*



ser compartidas por diferentes elementos HTML. Los selectores de clase son precedidos por un punto (.), por ejemplo: **.container** o **.highlight**.

- **Selectores de elemento:** Seleccionan elementos basados en sus nombres de etiquetas. Por ejemplo: **h1**, **p**, o **body**. Supongamos que deseamos seleccionar todos los títulos h1 en un documento HTML, podemos utilizar el selector de elemento "h1".

## Sumar IDs y clases al código HTML

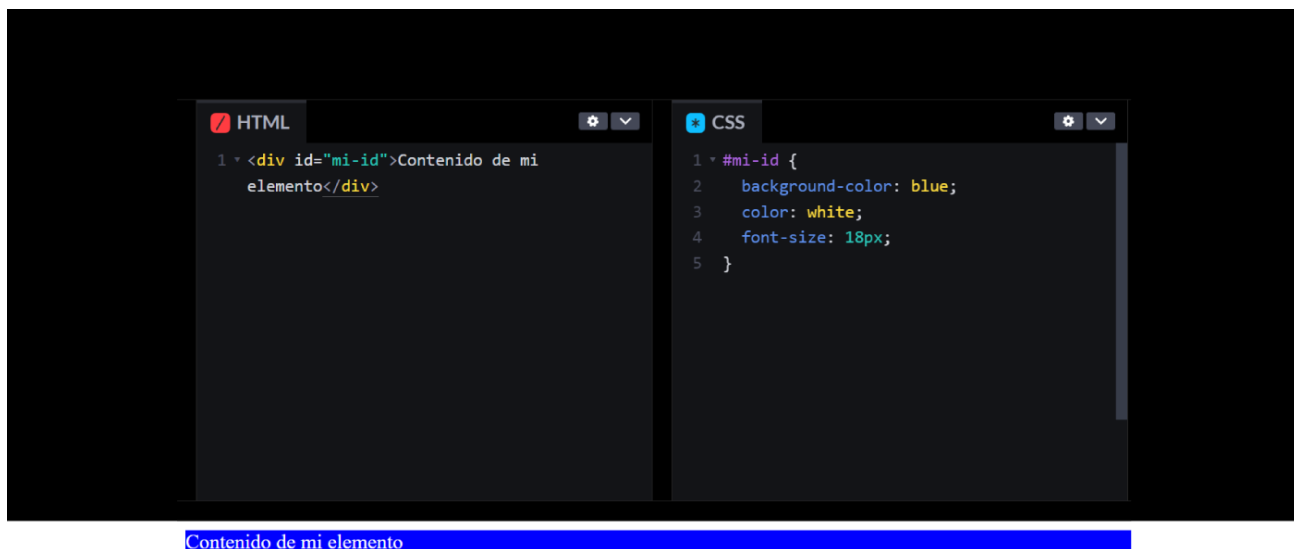
Para agregar IDs y clases a nuestro código HTML, debemos incluir el atributo **"id"** o **"class"** en la etiqueta HTML correspondiente. Por ejemplo, si queremos agregar un ID a un elemento "div", podríamos escribir:

```
<div id="mi-id">Contenido de mi elemento</div>
```

En este caso, agregamos el ID *"mi-id"* al elemento "div". Ahora, podemos utilizar este ID en nuestro código CSS para aplicar estilos específicos a este elemento. Por ejemplo:

```
#mi-id {  
  background-color: blue;  
  color: white;  
  font-size: 18px;  
}
```

En este código CSS, estamos aplicando un color de fondo azul, un color de texto blanco y un tamaño de fuente de 18 píxeles al elemento con el ID *"mi-id"*. Así se vería el resultado final:



Del mismo modo, para agregar una clase a un elemento, podemos incluir el atributo "class" en la etiqueta HTML correspondiente. Por ejemplo:

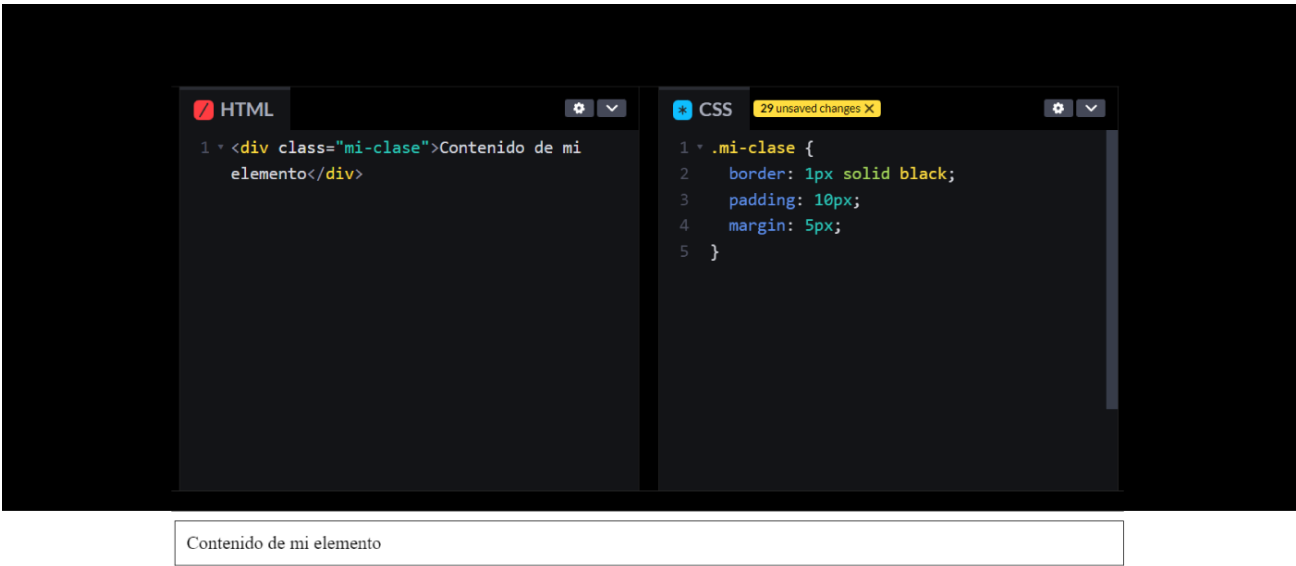
```
<div class="mi-clase">Contenido de mi elemento</div>
```

En este caso, agregamos la clase "mi-clase" al elemento "div". Ahora, podemos utilizar esta clase en nuestro código CSS para aplicar estilos a todos los elementos que tengan la clase "mi-clase". Por ejemplo:

```
.mi-clase {
  border: 1px solid black;
  padding: 10px;
  margin: 5px;
}
```

En este código CSS, estamos aplicando un borde de 1 píxel con un estilo sólido y color negro, un relleno de 10 píxeles y un margen de 5 píxeles a todos los

elementos que tengan la clase "mi-clase". Así se vería el resultado final:



## Reglas de especificidad

Las **reglas de especificidad de los selectores de CSS** determinan qué estilo se aplicará a un elemento HTML cuando existen múltiples reglas que compiten por dar estilo al mismo elemento.

Cada selector tiene un valor de especificidad que se basa en su tipo (ID, clase, elementos o universales) y en su posición en la hoja de estilos. El selector con la mayor especificidad ganará y se aplicará a ese elemento en particular:

- **Selectores universales:** Tienen una especificidad de 0, es decir, no se les asigna ningún valor de especificidad.
- **Selectores de elementos y pseudo-elementos:** Tienen una especificidad de 1.
- **Selectores de clase, atributo y pseudo clase:** Tienen una especificidad de 10.
- **Selectores de ID:** Tienen una especificidad de 100.
- **Estilos en líneas:** Tienen una especificidad de 1000.

Menor  
especificidad



Mayor  
especificidad

0	0	0	0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0

\*Selector universal

Elementos y pseudo-elementos

Clases, atributos y pseudo-clases

ID

Estilos en líneas

---

**!important** gana a todos

El valor de especificidad se calcula sumando los valores de los diferentes tipos de selectores utilizados en una regla de estilo. Un *estilo en línea* (que luego veremos de qué trata) tiene la máxima prioridad en CSS con una especificidad mayor que la de cualquier selector de ID, clase o elemento. Un *selector de ID* tiene un valor de especificidad mayor que un *selector de clase*, y un *selector de clase* tiene un valor mayor que un *selector de elementos*.

Por ejemplo, si hay un *selector de elemento* y un *selector de clase* que están aplicando diferentes estilos a un mismo elemento HTML, el selector de clase tendrá una especificidad mayor y, por lo tanto, su estilo será el que se aplicará al elemento.

En otras palabras, **cuando se aplican múltiples reglas a un elemento, se determina cuál es la más específica y se aplica esa regla. Si dos reglas tienen la misma especificidad, se aplica la que aparece más tarde en el código CSS.**

💡 También existe el “**!important**” que no provee especificidad pero sobrescribe cualquier otra regla sin importar cuanta especificidad tenga, no se recomienda su uso porque es difícil de depurar y no respeta la cascada de estilos.

Para que quede un poco más claro, veamos un ejemplo de diferentes selectores con su correspondiente valor de especificidad:

Selector	Especificidad			
<code>style="background-color: red;"</code>	1	0	0	0
<code>#wrapper #content {}</code>	0	2	0	0
<code>#content {}</code>	0	1	0	0
<code>p #name {}</code>	0	1	0	1
<code>div p #name {}</code>	0	1	0	2
<code>.details #name {}</code>	0	1	1	0

Es importante tener en cuenta las reglas de especificidad al diseñar hojas de estilo para evitar conflictos y asegurarse de que los estilos se apliquen correctamente.

💡 En el caso de los anidamientos, los puntos de especificidad se suman. Por ejemplo:

- Con el siguiente HTML:

```
<div class="container">  
  <h1 class="title">Título</h1>  
</div>
```

- Y el siguiente código CSS:

```
.container .title {  
  color: red;  
}
```

*Este selector tiene una especificidad de 20 (10 por la clase `.container` y 10 por la clase `.title`). Si se aplicara un estilo diferente al mismo elemento utilizando un selector con una especificidad de 10, el selector más específico tendría prioridad y se aplicaría el estilo correspondiente.*

## Tipos de CSS

Existen tres **tipos de CSS** que se utilizan para aplicar estilos a un documento HTML:


- CSS en línea (o “estilos en línea”)
- CSS interno
- CSS externo

Veamos de qué trata cada uno:

### CSS en línea

Los estilos se agregan directamente a los elementos HTML usando el atributo “`style`”. Por ejemplo:

```
<html>
<head>
</head>
<body>
  <p style="color: red;"> Un texto de color rojo </p>
</body>
</html>
```

 Aunque es fácil de implementar, no es recomendado ya que puede hacer que el código HTML sea difícil de leer y mantener.

### CSS interno

Los estilos se colocan dentro de una etiqueta `<style>` dentro de la sección `<head>` del documento HTML. Por ejemplo:

```
<html>
<head>
  <style>
    p {color: red;}
  </style>
</head>
<body>
  <p> Un texto de color rojo </p>
</body>
</html>
```

 Este método sólo es adecuado para proyectos pequeños o sitios web de una sola página.

## **CSS externo**

Los estilos se colocan en un archivo separado con la extensión ".css", el cual luego se enlaza al documento HTML usando la etiqueta <link>.

Archivo "index.html":

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
  <p> Un texto de color rojo </p>
</body>
</html>
```

Archivo "style.css":

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

💡 **Este método es el más utilizado y, a su vez, es la mejor opción de los 3 tipos de CSS**, ya que permite mantener un estilo consistente en todo el sitio y hace que la actualización de estilos sea más manejable.

## Desafío

A continuación, te presentamos el desafío de hoy. Deberás agregar algunos estilos CSS a la página HTML que creaste en el encuentro anterior cumpliendo con los siguientes requisitos:

- Cambiar el color del fondo de la página.
- Cambiar el color de la fuente de la página.
- Dar estilo al encabezado h1 y a los párrafos.

A resolver:

**1.** Crear un archivo nuevo en Visual Code (o en tu editor de código) y guárdalo con la extensión ".css", como por ejemplo "estilos.css". Este archivo contendrá los estilos CSS.

**2.** Enlazar el archivo CSS externo que creaste con tu documento HTML. Abrir tu archivo "mi-primer-website.html" del encuentro anterior y agregar el siguiente código dentro de la sección <head>, justo debajo de la etiqueta <title>:

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

**El atributo "href" se utiliza para especificar la ubicación de un archivo.**

Puede ser una dirección de internet o una dirección de la propia computadora. En este caso, nos centraremos en la segunda opción. La ruta especificada es relativa al archivo HTML actual:

- Si el archivo "styles.css" está en una carpeta llamada "CSS" al mismo nivel que el archivo HTML, la ruta relativa sería: *href = "css/styles.css"*.
- Si los archivos HTML y CSS están en carpetas separadas al mismo nivel, la ruta relativa sería: *href = "../css/styles.css"*.



En este caso, el código **"href"** le indica al navegador que busque un archivo llamado "styles.css" en el mismo directorio que el archivo HTML y aplique los estilos definidos en él.

**3.** Abrir el archivo "styles.css" y agregar algunos estilos básicos. Empecemos cambiando el color de fondo y el color de fuente de toda la página. Por ejemplo, puedes agregar el siguiente código:

```
body {  
    background-color: lightblue;  
    color: darkblue;  
}
```

Este código apunta al elemento <body>, y establece su color de fondo en azul claro y su color de fuente en azul oscuro.

**4.** Dar estilo al encabezado <h1>. Por ejemplo, puedes agregar el siguiente código a tu archivo "styles.css":

```
h1 {  
    font-family: 'Arial', sans-serif;  
    font-size: 36px;  
    text-align: center;  
}
```

Este código apunta al elemento <h1> y establece su familia de fuente en Arial, tamaño de fuente en 36 píxeles y alineación de texto centrada.

**5.** Finalmente, dar estilo a los párrafos. Por ejemplo, puedes agregar algunos márgenes y rellenos aplicando el siguiente código:

```
p {  
    font-family: 'Georgia', serif;  
    font-size: 18px;  
    padding: 10px;  
    margin: 20px;  
}
```

Este código apunta a los elementos <p> y establece su familia de fuentes en Georgia, el tamaño de la fuente en 18 píxeles, el padding en 10 píxeles y el margen en 20 píxeles.

**6.** Guardar los archivos: "mi-primer-website.html" y "styles.css", y luego actualizar la página web en el navegador. Ahora deberías ver los nuevos

estilos aplicados a tu página HTML, con un color de fondo, un color para los textos, y un encabezado y párrafos con otro aspecto.

**¡Desafío terminado!**

## Resolución del desafío

En el siguiente video te compartimos una de la formas en la que puedes resolver el desafío anterior:

 [Como agregar una hoja de estilo \(CSS\) | Introducción al Desarrollo Web & HTML | Egg](#)

**¡Felicidades!** 🎉

Añadiste exitosamente CSS a tu página HTML para darle estilo. A medida que continúes en el curso, aprenderás más sobre propiedades de CSS, selectores avanzados y técnicas para crear páginas web responsivas y visualmente atractivas.

En el próximo encuentro, profundizaremos en el diseño y la disposición de CSS, cubriendo conceptos como el “Box Model”, “display” y posicionamiento, “Flexbox” y “Grid”, y diseño responsivo. Estos conceptos te ayudarán a crear diseños más complejos y pulidos para tus páginas web.

## Valida tus conocimientos

Te compartimos el siguiente enlace para que puedas acceder al [material complementario](#) que contiene información que puede resultar útil y enriquecedora:

- [Guía rápida de selectores](#)

Además, te proponemos realizar una autoevaluación para poner a prueba los principales aprendizajes del día de hoy. ¿Te animas?:

 [Realizar test](#)

Ten en cuenta que si has llegado hasta aquí, ya has cumplido con el objetivo del encuentro.

¡Hasta la próxima! 

## Mapa de conceptos vistos

