

# Programación Web Desde Cero

## Selectores en CSS

¡Hola! 🖐️

Hasta ahora vimos cómo utilizar CSS teniendo en cuenta selectores de etiqueta generales, como **div**, **section**, **header**, **nav**.

Pero, a menos que quieras que cada sección de tu sitio web se vea exactamente igual vamos a tener que ver **otro tipo de selectores**. Esta es una parte crucial de la funcionalidad para nosotros. Es la forma en que decimos cosas como "**Quiero que este párrafo sea azul y ese otro párrafo sea amarillo**". Hasta ahora, solo hemos podido hacer que todos nuestros párrafos sean azules (o amarillos).

Por ello vamos a ver cómo utilizar otro tipo de selectores.

¡Comencemos! 🚀

---

## Selectores CSS

Teniendo en cuenta que ya podemos asignarle estilos a todo un sitio web, mediante un archivo `css` que usa selectores para elegir las etiquetas a las que asignarles los estilos, también tenemos que entender que existen varios tipos de selectores.

### Selector universal

Se utiliza para seleccionar todos los elementos de la página. El siguiente ejemplo elimina el margen y el relleno de todos los elementos HTML (por ahora no es importante fijarse en la parte de la declaración de la regla CSS):

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

## Selector de etiqueta

Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector. El siguiente ejemplo selecciona todos los párrafos de la página:

```
p {  
  text-align: justify;  
  font-family: Verdana;  
}
```

*/\* El siguiente ejemplo selecciona todas las tablas y div de la página:\*/*

```
table, div {  
  border: 1px solid red;  
}
```

## Selector de clase

¿Qué hacer para aplicar estilos solo al primer párrafo?

Una de las soluciones más sencillas para aplicar estilos a un solo elemento de la página consiste en utilizar el atributo class de HTML sobre ese elemento para indicar directamente la regla CSS que se le debe aplicar.

Ejemplo:

### HTML:

```
<body>
  <p class="destacado"> Párrafo 1</p>
  <p class="error"> Párrafo 2</p>
  <p> Párrafo 3</p>
</body>
```

### CSS:

```
.destacado {
  font-size: 15px;
}
.error {
  color: red;
}
```

En nuestro archivo CSS para especificar una clase, vamos a poner punto (':') y el nombre de la clase que queremos que coincida con valor que pongamos en nuestro atributo class en el html.

Entonces, en el ejemplo podemos ver como el primer párrafo tiene el valor destacado y el segundo párrafo el valor error para el atributo class y en nuestro archivo CSS, hemos definido un estilo para esas clases.

El beneficio del atributo class, además de dejarnos asignar estilos a un solo elemento, es que después podemos reutilizar esa class para asignarle ese estilo a otros párrafos concretos o a otras etiquetas, solo deberemos ponerle el valor de un estilo que ya existe en el atributo class.

### Selector de ID

En un documento HTML, los selectores de ID de CSS buscan un elemento basado en el contenido del atributo id. El atributo ID del elemento seleccionado debe coincidir exactamente con el valor dado en el selector. Este tipo de selectores sólo seleccionan un elemento de la página porque el valor del atributo id no se puede repetir en dos elementos diferentes de una misma página.

## HTML:

```
<div id="identificador"> ¡Este div tiene un ID especial! </div>
<div> Este solo es un div regular. </div>
```

## CSS:

```
#identificador{
background-color: blue;
}
```

En nuestro archivo CSS para especificar un ID, vamos a poner el numeral ('#') y el nombre del ID que queremos que coincida con el valor que pongamos en nuestro atributo ID en el html.

Como podemos ver el ID, es muy parecido al atributo class, pero la diferencia es que el ID se puede usar para identificar un solo elemento, mientras que una clase se puede usar para agrupar más de uno.

---

## Usando más de una clase

Ahora bien, como vimos anteriormente usamos los selectores de clase e id si queremos ser más específicos en los estilos que aplicamos.

Pero, **¿pueden los elementos tener más de una clase?**, sí, claro que sí!

Si por ejemplo, tenemos botones que comparten la mayoría de sus estilos, pero de repente quiero que el color de fondo de uno sea diferente al del otro, puedo aplicar estilos diferentes a cada uno de ellos, usando una clase diferente para cada uno.

### Por ejemplo

En HTML tendremos algo como lo siguiente:

```
<div class="button">Saber más</div>
<div class="button contacto">Contactarme</div>
```

En CSS, tendremos algo así:

```
.button {
  color: #fff;
  background-color: #5995da; /* Blue */
  font-weight: bold;
  padding: 20px;
  text-align: center;
  border: 2px solid #5d6063; /* Dark gray */
  border-radius: 5px;
  cursor: pointer;
}

.contacto {
  font-style: italic;
  background-color: #eeb75a; /* Yellow */
}
```

---

## Pseudo Clases

Las pseudoclases se utilizan para aplicar estilos específicos a elementos HTML en ciertos estados o situaciones. Estas pseudoclases permiten seleccionar y estilizar elementos basados en su estado o posición en la estructura del documento, lo que brinda un mayor control sobre la apariencia y el comportamiento de una página web.

Las pseudoclases se utilizan en conjunto con selectores para identificar elementos específicos. Un selector se combina con una pseudoclase para definir el estado o situación en la que se desea aplicar un estilo. Algunas de las pseudoclases más comunes incluyen:

- **:hover:** Se aplica cuando el cursor del mouse se coloca sobre un elemento, generalmente utilizado para proporcionar retroalimentación visual al usuario.
- **:active:** Se aplica cuando un elemento es activado, como cuando se hace clic en un enlace o un botón.
- **:focus:** Se aplica cuando un elemento obtiene el foco, como cuando se selecciona mediante la navegación del teclado.
- **:first-child:** Se aplica al primer elemento hijo de su elemento padre.
- **:last-child:** Se aplica al último elemento hijo de su elemento padre.
- **:nth-child():** Se aplica a elementos específicos basados en su posición en la jerarquía de su elemento padre, utilizando una fórmula matemática.
- **:nth-of-type():** Similar a :nth-child(), pero selecciona elementos específicos basados en su tipo en lugar de su posición.
- **:not():** Se utiliza para seleccionar elementos que no coinciden con un selector dado.

Por ejemplo:

```
a:hover {
  background-color: lightgray;
}
```

Cambia el color de fondo de un enlace cuando el cursor del mouse está sobre él.

```
button:active {
  background-color: darkblue;
  color: white;
}
```

Cambia el color de fondo de un botón cuando está siendo presionado.

```
ul li:nth-child(odd) {
  background-color: #f2f2f2;
}
```

Cambia el estilo de los elementos de lista impares en una lista desordenada.

Estas son solo algunas de las muchas pseudoclases disponibles en CSS. Permiten a los diseñadores y desarrolladores web aplicar estilos específicos a elementos en función de su relación con otros elementos o de su estado en la página. Esto puede ayudar a mejorar la usabilidad y la estética de una página web, al tiempo que permite un mayor grado de personalización y control en el diseño.

Pueden revisar más contenido sobre pseudo clases en el siguiente enlace

👉 [Pseudo-Clases](#)

---

## Selectores descendentes en CSS

En CSS, los selectores descendentes son una forma de seleccionar elementos HTML específicos que están anidados dentro de otros elementos. Los selectores descendentes permiten aplicar estilos a elementos hijos que están contenidos dentro de un elemento padre particular. Esto es útil cuando deseas aplicar estilos a elementos específicos en una estructura HTML más compleja y jerárquica.

La sintaxis básica de un selector descendente es la siguiente:

```
elemento-padre elemento-hijo {  
    /* Estilos que se aplicarán al elemento hijo */  
}
```

Aquí tienes una explicación más detallada:

- **elemento-padre:** Este es el elemento HTML que contiene al elemento hijo que deseas seleccionar y estilizar. Puede ser cualquier selector válido, como un nombre de etiqueta, una clase o un ID.
- **elemento-hijo:** Este es el elemento HTML que está anidado dentro del elemento padre y al que deseas aplicar estilos.

Por ejemplo, si tienes una estructura HTML con una lista desordenada (<ul>) que contiene elementos de lista (<li>), y quieres aplicar estilos solo a los elementos de lista que están dentro de esa lista desordenada, podrías usar un selector descendente de esta manera:

```
ul li {  
    /* Estilos para los elementos de lista dentro de una lista desordenada  
    */  
}
```

En este caso, los estilos se aplicarán solo a los elementos <li> que están contenidos dentro de un elemento <ul>.

Los selectores descendentes son especialmente útiles cuando deseas estilizar elementos de manera específica en función de su contexto o jerarquía en el árbol de elementos HTML. **Puedes combinar selectores descendentes con otras técnicas de selección, como clases, IDs y pseudoclases, para lograr un mayor nivel de precisión en la aplicación de estilos a elementos específicos en tu página web.**

---

## Especificidad en CSS

La especificidad en CSS es un concepto que determina qué reglas de estilo se aplicarán a un elemento cuando existen múltiples reglas que compiten por aplicar estilos al mismo elemento. En otras palabras, la especificidad **define la jerarquía de importancia de las reglas CSS** y cómo se resuelven los conflictos cuando se aplican estilos a elementos en una página web.

La especificidad se calcula mediante la combinación de diferentes componentes en una puntuación específica para cada regla. Estos componentes son:



1. **Número de IDs en el selector:** Cada ID en un selector suma 100 puntos a la especificidad.
2. **Número de clases, atributos y pseudoclases en el selector:** Cada clase, atributo o pseudoclase suma 10 puntos.
3. **Número de elementos y pseudoelementos en el selector:** Cada elemento HTML o pseudoelemento suma 1 punto.
4. **Importancia del estilo:** Si se utiliza !important en una regla, esto otorga un alto nivel de especificidad, anulando la mayoría de las reglas.

**Cuando hay conflictos entre reglas, la regla con la especificidad más alta se aplicará.** Si dos reglas tienen la misma especificidad, se aplicará la última que aparezca en la hoja de estilos.

Por ejemplo, considera estas reglas:

```
p {  
    color: red;  
}  
  
#mi-id {  
    color: blue;  
}  
  
p#mi-id {  
    color: green;  
}
```

Si tienes un párrafo `<p id="mi-id">`, se aplicará el color verde, porque la regla **p#mi-id** tiene una especificidad más alta que las otras reglas.

Es importante entender la especificidad para evitar conflictos inesperados y entender por qué ciertas reglas no se aplican como se espera. Una buena práctica es tratar de **mantener la especificidad lo más baja posible y evitar el uso excesivo de !important**, ya que esto puede dificultar el mantenimiento de tus hojas de estilo a medida que el proyecto crece.

---

## Resumen

### ¡Felicidades!

Esperamos que esta información te haya sido de utilidad y que te lleves para repasar:

- Diferencias entre los tipos de selectores en CSS
  - Universales
  - De etiqueta
  - De clase
  - De ID
- Cómo usar más de una clase en CSS
- Qué es y cómo se usa una pseudo-clase
- Concepto de especificidad y priorización en CSS