



# CSS

# Pseudo-classes ⚡

## CSS Pseudo-clases

Una pseudo-clase CSS es una palabra clave que se añade a los selectores y que especifica un estado especial del elemento seleccionado.

Por ejemplo:

- `:hover` aplicará un estilo cuando el usuario haga hover sobre el elemento especificado por el selector.
- `:checked` (específico a inputs) aplicará un estilo cuando
- `:nth-child()` aplicará un estilo a uno o varios n-elementos (probar 1, 2, odd, even, 3n...)

```
nav a:hover {  
  background-color: #ffffff;  
  color: #dddddd;  
}
```

Son casi 40 y se van a agregar más

# Selectores



## Selectores - Intro

Al principio del curso vimos tres formas de “seleccionar” un elemento vía CSS:



```
/* Por nombre de elemento, ej div, p, body */  
div { ... }  
p { ... }  
body { ... }
```



```
/* Por identificador */  
#principal { ... }
```



```
/* Por nombre de clase */  
.contenedor { ... }
```

## Selectores - Hijos directos

Imaginemos que tenemos una lista y necesitamos que cada elemento tenga un padding de 10px.

**¿Cómo podemos lograr esto?**

```
<ul>
  <li>Primer elemento</li>
  <li>Segundo elemento</li>
  <li>Tercer elemento</li>
  <li>Cuarto elemento</li>
  <li>Quinto elemento</li>
  <li>Sexto elemento</li>
  <li>Séptimo elemento</li>
  <li>Octavo elemento</li>
  <li>Noveno elemento</li>
  <li>Décimo elemento</li>
</ul>
```

## Selectores - Hijos directos

Una solución posible podría ser agregarle a cada etiqueta li la clase “padding-10”.

Esto funciona, pero se vuelve difícil de mantener y escalar. Imaginen si tuviésemos que agregar 10 clases.

```
<ul>
  <li class="padding-10">Primer elemento</li>
  <li class="padding-10">Segundo elemento</li>
  <li class="padding-10">Tercer elemento</li>
  <li class="padding-10">Cuarto elemento</li>
  <li class="padding-10">Quinto elemento</li>
  <li class="padding-10">Sexto elemento</li>
  <li class="padding-10">Séptimo elemento</li>
  <li class="padding-10">Octavo elemento</li>
  <li class="padding-10">Noveno elemento</li>
  <li class="padding-10">Décimo elemento</li>
</ul>
```

```
.padding-10 {
  padding: 10px;
}
```

## Selectores - Hijos directos

Otra solución un poco más elegante podría ser la de utilizar el selector de **“hijos directos”**, el cual se representa con el símbolo de mayor **“>”**.

De esta manera, cualquier elemento **“li”** que sea **“hijo directo”** de la clase **“lista-elementos”** tendrá un padding de 10px.

```
<ul class="lista-elementos">
  <li>Primer elemento</li>
  <li>Segundo elemento</li>
  <li>Tercer elemento</li>
  <li>Cuarto elemento</li>
  <li>Quinto elemento</li>
  <li>Sexto elemento</li>
  <li>Séptimo elemento</li>
  <li>Octavo elemento</li>
  <li>Noveno elemento</li>
  <li>Décimo elemento</li>
</ul>
```

```
.lista-elementos > li {
  padding: 10px;
}
```



## Selectores - Hijos directos

Algunos ejemplos del selector o combinador de selectores:

```
/* Cualquier elemento "li" que sea hijo directo de un elemento "ul" */
ul > li { ... }

/* Solo los elementos "li" dentro de otro elemento con clase "lista-elementos */
.lista-elementos > li { ... }

/* Solo los elementos "li" que sean hijos directos de un "ul" con clase "lista-elementos */
ul.lista-elementos > li { ... }

/*
    Cualquier elemento "span" con la clase "subrayado"
    ...que sea hijo directo de un elemento "p"
    ...que sea hijo directo de un elemento "li"
    ...que sea hijo directo de un elemento "ul"
*/
ul > li > p > span.subrayado > { ... }
```

## Selectores - Hijos

Además de poder seleccionar “**hijos directos**” de otro selector, podemos seleccionar cualquier descendiente sin importar la generación, utilizando un espacio entre selectores “ ”.

**(Parece la misma imagen pero no!)**

```
/* Cualquier elemento "li" que sea hijo de un elemento "ul" */
ul li { ... }

/* Solo los elementos "li" dentro de otro elemento con clase "lista-elementos */
.lista-elementos li { ... }

/* Solo los elementos "li" que sean hijos de un "ul" con clase "lista-elementos */
ul.lista-elementos li { ... }

/*
    Cualquier elemento "span" con la clase "subrayado"
    ...que sea hijo directo de un elemento "p"
    ...que sea hijo directo de un elemento "li"
    ...que sea hijo de un elemento "ul"
*/
ul li > p > span.subrayado { ... }
```

## Selectores - Hermanos

También podemos seleccionar al “**siguiente hermano**” de un elemento, utilizando el símbolo “+”.

**(Esto puede ser útil cuando hablemos de pseudo-clases)**



```
/* Cualquier elemento "b" que le siga a un elemento "span" */  
span + b { ... }
```

```
/* Cualquier elemento con clase "contenido" que le siga a  
un elemento con clase "titulo" */  
.titulo + .contenido { ... }
```

## Selectores - Hermanos

También podemos seleccionar todos los “**siguientes hermanos**” de un elemento, utilizando el símbolo “~”.

```
/* Cualquier elemento "b" que esté al mismo nivel
   y venga después que un elemento "span */
span ~ b { ... }

/* Cualquier elemento con clase "contenido" que esté al mismo nivel
   y venga después que un elemento con clase "titulo" */
.titulo ~ .contenido { ... }
```

# LINKS

## LINKS

- **Pseudo-clases**
  - <https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes>
- **Selectores**
  - [https://developer.mozilla.org/es/docs/Web/CSS/Child\\_combinator](https://developer.mozilla.org/es/docs/Web/CSS/Child_combinator)
  - [https://developer.mozilla.org/es/docs/Web/CSS/General\\_sibling\\_combinator](https://developer.mozilla.org/es/docs/Web/CSS/General_sibling_combinator)



[gustavgueez](#)



[gustavgueez](#)



[gustavgueez](#)

**GUSTAVO RODRIGUEZ**

FULL STACK DEVELOPER  
SOLCRE