

10 CSS: Pseudoclases por Posición y Tipo

Además de las pseudoclases de estado, CSS ofrece un conjunto de pseudoclases estructurales que nos permiten seleccionar elementos basándonos en su posición dentro del árbol de elementos hermanos (elementos que comparten el mismo padre) o por su tipo en relación a sus hermanos. Estas son increíblemente útiles para aplicar estilos de forma alterna, seleccionar elementos específicos en una secuencia, o tratar casos especiales como el primer o último elemento de una lista.

Selección por Posición (independiente del tipo):

Estos selectores consideran la posición del elemento entre *todos* sus hermanos, sin importar la etiqueta (tipo) de cada uno.

1. `:first-child`

- **Selecciona:** Un elemento que es el **primer hijo** de su elemento padre.
- **Ejemplo:**

```
<div>
  <h2>Título</h2> <!-- Es el primer hijo del div →>
  <p>Primer párrafo.</p>
  <p>Segundo párrafo.</p>
</div>
<ul>
  <li>Item 1</li> <!-- Es el primer hijo del ul →>
  <li>Item 2</li>
</ul>
```

```
p:first-child { /* Selecciona un PÁRRAFO que sea el primer hijo */
  font-weight: bold; /* No aplicará en el ejemplo, el p no es el primer hijo */
}
h2:first-child { /* Selecciona un H2 que sea el primer hijo */
  color: green; /* Aplicará al h2 del div */
}
li:first-child { /* Selecciona un LI que sea el primer hijo */
  color: blue; /* Aplicará al "Item 1" */
}
```

- **Nota:** `elemento:first-child` selecciona un `elemento` *solo si* ese elemento es, además, el primer hijo de su padre.

2. `:last-child`

- **Selecciona:** Un elemento que es el **último hijo** de su elemento padre.
- **Ejemplo:**

```

<div>
  <p>Primer párrafo.</p>
  <p>Segundo párrafo.</p> <!-- Es el último hijo del div →
</div>
<ul>
  <li>Item 1</li>
  <li>Item 2</li> <!-- Es el último hijo del ul →
</ul>

```

```

p:last-child { /* Selecciona un PÁRRAFO que sea el último hijo */
  border-bottom: 1px solid grey; /* Aplicará al segundo párrafo del div */
}
li:last-child { /* Selecciona un LI que sea el último hijo */
  font-style: italic; /* Aplicará al "Item 2" */
}

```

3. :nth-child(n)

- **Selecciona:** Elementos basados en su posición numérica entre todos sus hermanos. `n` puede ser un número, una palabra clave (`even` o `odd`), o una fórmula (`an+b`).
- **Sintaxis y Ejemplos:**
 - `li:nth-child(3)` : Selecciona el tercer elemento `` (si es el tercer hijo en general).
 - `p:nth-child(even)` : Selecciona los párrafos `<p>` que son hijos pares (2º, 4º, 6º, etc.).
 - `p:nth-child(odd)` : Selecciona los párrafos `<p>` que son hijos impares (1º, 3º, 5º, etc.).
 - `tr:nth-child(2n)` : Equivalente a `even` . ($20=0$, $21=2$, $2*2=4$, ...)
 - `tr:nth-child(2n+1)` : Equivalente a `odd` . ($20+1=1$, $21+1=3$, $2*2+1=5$, ...)
 - `div:nth-child(3n)` : Selecciona los `<div>` que son el 3er, 6º, 9º... hijo.
 - `span:nth-child(n+4)` : Selecciona los `` que son el 4º hijo o posterior (4º, 5º, 6º...). ($0+4=4$, $1+4=5$, ...)
 - `span:nth-child(-n+3)` : Selecciona los `` que están entre el 1º y el 3er hijo. ($-0+3=3$, $-1+3=2$, $-2+3=1$, $-3+3=0$, ...)
- **Ejemplo:** Estilizar filas pares de una tabla.

```

<table>
  <tbody>
    <tr><td>Fila 1</td></tr> <!-- Impar →
    <tr><td>Fila 2</td></tr> <!-- Par →
    <tr><td>Fila 3</td></tr> <!-- Impar →
    <tr><td>Fila 4</td></tr> <!-- Par →
  </tbody>
</table>

```

```
tr:nth-child(even) { /* Selecciona las filas (tr) que son hijos pares */
  background-color: #f2f2f2; /* Estilo cebra */
}
```

4. `:nth-last-child(n)`

- **Selecciona:** Igual que `:nth-child(n)`, pero contando desde el **último** hijo hacia atrás.
- **Ejemplo:** Seleccionar el penúltimo elemento de una lista, si es un ``.

```
li:nth-last-child(2) {
  border-bottom: 2px solid red;
}
```

5. `:only-child`

- **Selecciona:** Un elemento que es el **único hijo** de su padre (no tiene hermanos).
- **Ejemplo:**

```
<div><p>Soy hijo único.</p></div> <!-- El p será seleccionado →
<div><p>No soy único.</p><p>Tengo un hermano.</p></div>
```

```
p:only-child {
  color: purple;
  text-align: center;
}
```

Selección por Tipo (dentro de los hermanos):

Estos selectores consideran la posición del elemento solo entre los hermanos que son del **mismo tipo** (misma etiqueta). Esto es crucial cuando tienes elementos de diferentes tipos mezclados.

La Diferencia Clave: `-child` vs `-of-type`

Imagina esta estructura:

```
<section>
  <h1>Título</h1>    <!-- 1er hijo, 1er h1 →
  <p>Párrafo 1</p>    <!-- 2º hijo, 1er p →
  <p>Párrafo 2</p>    <!-- 3er hijo, 2º p →
  <h2>Subtítulo</h2> <!-- 4º hijo, 1er h2 →
  <p>Párrafo 3</p>    <!-- 5º hijo, 3er p →
</section>
```

- `p:nth-child(2)`: Selecciona "Párrafo 1" (porque es un `<p>` Y es el 2º hijo en general).
- `p:nth-child(3)`: Selecciona "Párrafo 2" (porque es un `<p>` Y es el 3er hijo en general).
- `p:nth-child(4)`: **No selecciona nada**, porque el 4º hijo es un `<h2>`, no un `<p>`.
- `p:nth-of-type(2)`: Selecciona "Párrafo 2" (porque es el **segundo elemento de tipo** `<p>` dentro del `<section>`).

- `p:nth-of-type(3)` : Selecciona "Párrafo 3" (porque es el **tercer elemento de tipo** `<p>` dentro del `<section>`).

Ahora, los selectores `-of-type` :

1. `:first-of-type`

- **Selecciona:** El **primer** elemento de su tipo entre un grupo de hermanos.
- **Ejemplo (usando el HTML anterior):**

```
p:first-of-type { /* Selecciona el primer <p> ("Párrafo 1") */
  font-weight: bold;
}
h1:first-of-type { /* Selecciona el primer <h1> ("Título") */
  color: teal;
}
h2:first-of-type { /* Selecciona el primer <h2> ("Subtítulo") */
  color: orange;
}
```

2. `:last-of-type`

- **Selecciona:** El **último** elemento de su tipo entre un grupo de hermanos.
- **Ejemplo (usando el HTML anterior):**

```
p:last-of-type { /* Selecciona el último <p> ("Párrafo 3") */
  font-style: italic;
}
```

3. `:nth-of-type(n)`

- **Selecciona:** Elementos basados en su posición numérica entre los hermanos del **mismo tipo**. `n` funciona igual que en `:nth-child(n)` (número, `even`, `odd`, fórmula `an+b`).
- **Ejemplo:** Aplicar estilo cebra solo a los párrafos, ignorando otros elementos intercalados.

```
/* Usando el HTML anterior */
p:nth-of-type(even) { /* Selecciona el 2º <p> ("Párrafo 2") */
  background-color: lightyellow;
}
```

4. `:nth-last-of-type(n)`

- **Selecciona:** Igual que `:nth-of-type(n)`, pero contando desde el último hermano de ese tipo hacia atrás.
- **Ejemplo:** Seleccionar el penúltimo párrafo.

```
/* Usando el HTML anterior */
p:nth-last-of-type(2) { /* Selecciona el penúltimo <p> ("Párrafo 2") */
```

```
text-decoration: underline;
}
```

5. `:only-of-type`

- **Selecciona:** Un elemento que es el **único de su tipo** entre sus hermanos (puede haber otros hermanos de diferentes tipos).
- **Ejemplo (usando el HTML anterior):**

```
h1:only-of-type { /* Selecciona "Título" porque es el único <h1> */
border-bottom: 1px solid black;
}
h2:only-of-type { /* Selecciona "Subtítulo" porque es el único <h2> */
border-bottom: 1px solid grey;
}
p:only-of-type { /* No selecciona nada porque hay varios <p> */
color: red;
}
```

Otros Selectores Estructurales Útiles:

1. `:empty`

- **Selecciona:** Elementos que **no tienen hijos** en absoluto. Esto incluye no tener otros elementos anidados ni siquiera nodos de texto (ni espacios en blanco significativos entre las etiquetas de apertura y cierre).
- **Ejemplo:**

```
<p>Tengo contenido.</p>
<p></p> <!-- Seleccionado (completamente vacío) →
<p> </p> <!-- NO seleccionado (contiene espacios) →
<p><!-- comentario →</p> <!-- Seleccionado (comentarios no cuentan como hijos) --
>
<div><span></span></div> <!-- El div NO está vacío (tiene un span), el span SÍ está vacío →
```

```
div:empty {
border: 1px dashed red;
padding: 10px; /* Para hacerlo visible */
background-color: #ffeeee;
}
```

- **Uso:** Útil para ocultar contenedores vacíos o darles un estilo distintivo (p.ej., en interfaces generadas dinámicamente).

2. `:root`

- **Selecciona:** El elemento raíz del documento. En documentos HTML, siempre es el elemento `<html>`.

- **Uso:** Es el lugar recomendado para definir **Variables CSS (Custom Properties)** globales y a veces para establecer estilos base para toda la página como `font-size` (afectando la base para unidades `rem`) o `background-color` para el documento completo.
- **Ejemplo:**

```
:root {  
  --color-principal: #336699; /* Define una variable CSS */  
  font-size: 16px; /* Base para unidades rem */  
  background-color: #fdfdfd;  
}  
  
body {  
  background-color: white; /* Puede sobrescribir el fondo solo para el body */  
  color: var(--color-principal); /* Usa la variable definida en :root */  
}
```