

# Unidad 6: Selectores

## *Selectores de hijo*

### **:first-child**

Permite seleccionar el primer elemento hijo dentro de un contexto (padre).

### **:last-child**

Permite seleccionar el último elemento hijo.

### **:only-child**

Permite seleccionar un elemento que es el único hijo dentro de un contexto.

### **:nth-child()**

Permite elegir un elemento hijo con las condiciones que le pasemos entre paréntesis: los elementos pares (even) o los impares (odd), un número en concreto (por ejemplo, 2), cada cierto número, indicado con la partícula n (por ejemplo 2n).

Si queremos indicar una repetición (cada n), pero que empiece en un número en concreto, lo podemos poner con el símbolo + . Por ejemplo, `:nth-child(2n+3)` sería "cada dos, empezando a contar por el tercero".

## *Selectores de tipo*

### **:first-of-type**

Permite seleccionar el primer elemento de su tipo (tipo de etiqueta).

### **:last-of-type**

Permite seleccionar el último elemento de su tipo.

### **:nth-of-type()**

Permite seleccionar un elemento de su tipo, con las mismas condiciones que hemos visto en `:nth-child`.

## *Selectores de atributo*

### **[atributo]**

El selector de atributo se escribe entre `[]` y nos permite hacer referencia a los elementos que tengan ese atributo. Por ejemplo, `a[href]` indicaría las etiquetas de enlace que tengan el atributo `href`.

### **`[atributo="valor"]`**

Podemos emplear el selector de atributo para ver si coincide exactamente con un valor, empleando el símbolo `=`. Por ejemplo, `a[href="http://google.com"]` solo afectaría a los enlaces cuyo `href` coincida **exactamente** con este valor.

### **`[atributo^="valor"]`**

Mediante la partícula `^` indicamos que el valor del atributo **empieza** por un trozo concreto de texto.

### **`[atributo$="valor"]`**

Mediante la partícula `$` indicamos que el valor del atributo **acaba** por un trozo concreto de texto.

### **`[atributo*="valor"]`**

Mediante la partícula `*` indicamos que el valor del atributo **contiene** un trozo concreto de texto.

## ***Otros selectores útiles***

### **`:empty`**

Selecciona una etiqueta vacía de contenido.

### **`:not()`**

Selecciona un elemento que **no** cumpla la condición dada entre los paréntesis. Ejemplo: `p:not(.important)` seleccionaría todos los párrafos que no tengan la clase `.important`.

### **`:lang()`**

Selecciona un elemento cuyo atributo `lang` coincida con el valor declarado.

### **`:root`**

Selecciona el elemento padre del documento. Normalmente, selecciona la etiqueta `<html>`.

### **`:target`**

Un elemento tiene estado de `:target` cuando la url de la página apunta directamente al id del mismo. Por ejemplo, en `http://ejemplo.com/index.html#menu`, el bloque con el `id="menu"` tendría `:target`. Con este selector podemos estilar al elemento que tenga `:target`.

## ***Pseudoelementos***

### **::before y ::after**

Los pseudoelementos `::before` y `::after` nos permiten crear un elemento virtual al principio o al final de un elemento HTML, pudiendo introducir trozos de texto que tengan un sentido estilístico o una imagen, a través de la propiedad `content`. Una vez declarado `content` (aunque sea vacío), estos elementos virtuales pueden recibir estilos como cualquier otro elemento.

Ejemplo:

```
div::after{  
  content: " ";  
  display: inline-block;  
  width: 10px;  
  height: 10px;  
}
```

En ocasiones podemos encontrar `::before` y `::after` escritos con un solo ":" (es decir, `:before` y `:after`). Es una sintaxis anticuada, pero los navegadores la aceptan igual.

### **::first-line**

Este selector permite indicar la primera línea de un elemento (por ejemplo, un párrafo).

### **::first-letter**

Este selector permite indicar la primera letra de un elemento.

### **::selection**

Este selector permite dar estilos a la selección de texto.