

# PSEUDO-CLASES CSS3 PARA SELECCIONAR ELEMENTOS

Una [pseudo-clase](#) es un selector [CSS](#) que nos ayuda a seleccionar elementos del [DOM](#) guiándonos por su estado en lugar de su información. Algunas de las *pseudo-clases* más populares que normalmente se utilizan a la hora de crear una hoja de estilo son [:hover](#), [:active](#) o [:focus](#). En [CSS3](#) contamos con algunas *pseudo-clases* que nos ayudan a seleccionar elementos guiándonos por el lugar que ocupan en el árbol del *DOM*. En este tutorial abordaremos algunas de las *pseudo-clases* de *CSS3* que más nos ayudan con este tipo de tarea.

## Pseudo-clase :first-child

Esta *pseudo-clase* nos permite seleccionar cada uno de los elementos que ocupe el primer lugar dentro del elemento que los contiene. Para entenderla mejor observemos los siguientes códigos y el resultado final:

### Código HTML

```
<div id="container">

  <div class="row">
    <div class="box">fila 1 | columna 1</div>
    <div class="box">fila 1 | columna 2</div>
  </div>

  <div class="row">
    <div class="box">fila 2 | columna 1</div>
    <div class="box">fila 2 | columna 2</div>
    <div class="box">fila 2 | columna 3</div>
  </div>

  <div class="row">
    <div class="box">fila 3 | columna 1</div>
    <div class="box">fila 3 | columna 2</div>
    <div class="box">fila 3 | columna 3</div>
    <div class="box">fila 3 | columna 4</div>
  </div>

</div>
```

### Código CSS

```
.box:first-child {
  background: #FFF;
  color: #232b2b;
}
```

### Resultado

fila 1   columna 1	fila 1   columna 2		
fila 2   columna 1	fila 2   columna 2	fila 2   columna 3	
fila 3   columna 1	fila 3   columna 2	fila 3   columna 3	fila 3   columna 4

Como hemos podido notar, hemos aplicado el estilo deseado a los primeros elementos con la clase *box* dentro de cada fila.

## Pseudo-clase :last-child

Esta *pseudo-clase* funciona muy similar a la anterior con la diferencia de que nos permite seleccionar cada uno de los elementos que ocupen el último lugar dentro del elemento que los contiene. Utilizando el mismo código *HTML* anterior analicemos el siguiente código *CSS* y el resultado final:

### Código CSS

```
.box:last-child {  
    background: #FFF;  
    color: #232b2b;  
}
```

### Resultado

fila 1   columna 1	fila 1   columna 2		
fila 2   columna 1	fila 2   columna 2	fila 2   columna 3	
fila 3   columna 1	fila 3   columna 2	fila 3   columna 3	fila 3   columna 4

En este caso hemos seleccionado los últimos elementos con la clase *box* dentro de cada fila.

## Pseudo-clase :nth-child

Esta *pseudo-clase* nos permite seleccionar uno o más elementos basado en la posición de estos dentro del elemento que los contiene. Esta *pseudo-clase* necesita un parámetro que podrá ser la palabra clave *odd* o *even*, un entero o una fórmula. Explicaremos cada uno de estos parámetros a continuación. Para todos los ejemplos usaremos el siguiente código *HTML*:

### Código HTML

```
<div id="container">  
    <div class="box">elemento 1</div>  
    <div class="box">elemento 2</div>  
    <div class="box">elemento 3</div>  
    <div class="box">elemento 4</div>  
    <div class="box">elemento 5</div>  
    <div class="box">elemento 6</div>  
    <div class="box">elemento 7</div>  
    <div class="box">elemento 8</div>  
    <div class="box">elemento 9</div>  
    <div class="box">elemento 10</div>  
    <div class="box">elemento 11</div>  
    <div class="box">elemento 12</div>  
</div>
```

### Uso de la palabra clave “odd”

Con esta palabra clave podremos seleccionar todos los elementos que ocupen un lugar impar dentro del elemento que los contiene, al usarla seleccionaremos al elemento 1, 3, 5, 7, y así sucesivamente. Observemos el código *CSS* y el resultado:

## Código CSS

```
.box:nth-child(odd) {  
    background: #FFF;  
    color: #232b2b;  
}
```

## Resultado

elemento 1	elemento 2	elemento 3	elemento 4
elemento 5	elemento 6	elemento 7	elemento 8
elemento 9	elemento 10	elemento 11	elemento 12

## Uso de la palabra clave “even”

Con esta palabra clave podremos seleccionar todos los elementos que ocupen un lugar par dentro del elemento que los contiene, al usarla seleccionaremos al elemento 2, 4, 6, 8, y así sucesivamente. Observemos el código CSS y el resultado:

## Código CSS

```
.box:nth-child(even) {  
    background: #FFF;  
    color: #232b2b;  
}
```

## Resultado

elemento 1	elemento 2	elemento 3	elemento 4
elemento 5	elemento 6	elemento 7	elemento 8
elemento 9	elemento 10	elemento 11	elemento 12

## Uso de un entero

Si usamos un entero como parámetro de esta *pseudo-clase* seleccionaremos al elemento que se encuentra en esa posición. Observemos el código CSS y el resultado:

## Código CSS

```
.box:nth-child(6) {  
    background: #FFF;  
    color: #232b2b;  
}
```

### Resultado

elemento 1	elemento 2	elemento 3	elemento 4
elemento 5	elemento 6	elemento 7	elemento 8
elemento 9	elemento 10	elemento 11	elemento 12

### Uso de una fórmula

Si usamos una fórmula podremos seleccionar varios elementos que cumplan los requisitos pautados por la misma. La fórmula a utilizar seguirá el patrón **an +/- b** Donde **a** correspondería a un número entero, **n** es un carácter variable que le indicará a la *pseudo-clase* que se trata de una fórmula y **b** correspondería a un entero y sólo se utiliza si hemos situado previamente el operador + o -.

El carácter **n** representa números enteros partiendo de 0. Por ejemplo, si situamos la fórmula **4n** analicemos las selecciones que se logran a partir de los siguientes cálculos:

Fórmula aplicada	Elemento a seleccionar
$4 \times 0 = 0$	ninguno
$4 \times 1 = 4$	4
$4 \times 2 = 8$	8
$4 \times 3 = 12$	12
...	...

La primera selección no contaría porque los índices de selección comienzan en 1 y el resultado ha dado 0. Como el *HTML* cuenta con 12 elementos se seleccionarían los elementos 4, 8 y 12. Observemos el código *CSS* y el resultado de esta fórmula:

### Código CSS

```
.box:nth-child(4n) {  
  
    background: #FFF;  
    color: #232b2b;  
  
}
```

Resultado

elemento 1	elemento 2	elemento 3	elemento 4
elemento 5	elemento 6	elemento 7	elemento 8
elemento 9	elemento 10	elemento 11	elemento 12

Ahora pasaremos a utilizar un operador en la fórmula. Si usamos la siguiente fórmula  $4n + 2$  estos serían los cálculos a realizar y los elementos seleccionados:

Fórmula aplicada	Elemento a seleccionar
$4 \times 0 + 2 = 0 + 2 = 2$	2
$4 \times 1 + 2 = 4 + 2 = 6$	6
$4 \times 2 + 2 = 8 + 2 = 10$	10
...	...

Como el *HTML* con el que estamos trabajando cuenta sólo con 12 elementos, se seleccionarían los elementos 2, 6 y 10. Observemos el código *CSS* y el resultado de esta fórmula:

Código CSS

```
.box:nth-child(4n + 2) {  
    background: #FFF;  
    color: #232b2b;  
}
```

Resultado

elemento 1	elemento 2	elemento 3	elemento 4
elemento 5	elemento 6	elemento 7	elemento 8
elemento 9	elemento 10	elemento 11	elemento 12

Otro caso es que no usemos el entero multiplicador en la fórmula (a). Por ejemplo la fórmula  $n + 7$  generaría los siguientes cálculos y seleccionaría a los siguientes elementos:

Fórmula aplicada	Elemento a seleccionar
$0 + 7 = 7$	7
$1 + 7 = 8$	8
$2 + 7 = 9$	9
$3 + 7 = 10$	10
...	...
$5 + 7 = 12$	12
...	...

Por lo tanto, esta fórmula seleccionaría del *HTML* con el que estamos trabajando los elementos que ocupen una posición mayor o igual que la 7, es decir, los elementos 7, 8, 9, 10, 11 y 12. Observemos el código *CSS* y el resultado de esta fórmula:

### Código CSS

```
.box:nth-child(n + 7) {  
    background: #FFF;  
    color: #232b2b;  
}
```

### Resultado

elemento 1	elemento 2	elemento 3	elemento 4
elemento 5	elemento 6	elemento 7	elemento 8
elemento 9	elemento 10	elemento 11	elemento 12

Veremos que sucede si usamos la variable **n** con signo negativo (**-n**) en la fórmula. Si usamos la fórmula **-n + 6** estos serían los cálculos y los elementos a seleccionar:

Fórmula aplicada	Elemento a seleccionar
$-0 + 6 = 6 - 0 = 6$	6
$-1 + 6 = 6 - 1 = 5$	5
$-2 + 6 = 6 - 2 = 4$	4

Fórmula aplicada	Elemento a seleccionar
$-3 + 6 = 6 - 3 = 3$	3
...	...
$-5 + 6 = 6 - 5 = 1$	1
...	...

Por lo tanto, esta fórmula seleccionaría del *HTML* con el que estamos trabajando los elementos que ocupen una posición menor o igual que la 6, es decir, los elementos 6, 5, 4, 3, 2 y 1. Observemos el código *CSS* y el resultado de esta fórmula:

### Código CSS

```
.box:nth-child(-n + 6) {  
    background: #FFF;  
    color: #232b2b;  
}
```

### Resultado

elemento 1	elemento 2	elemento 3	elemento 4
elemento 5	elemento 6	elemento 7	elemento 8
elemento 9	elemento 10	elemento 11	elemento 12

Por supuesto, podemos unir dos *pseudo-clases* para seleccionar un rango de elementos más personalizado. Por ejemplo, la fórmula  $n + 3$  unida a la fórmula  $-n + 10$  seleccionaría todos los elementos mayores o iguales a 3 y los menores o iguales a 10. Observemos el código *CSS* y el resultado de estas dos fórmulas unidas:

### Código CSS

```
.box:nth-child(n + 3):nth-child(-n + 10) {  
    background: #FFF;  
    color: #232b2b;  
}
```

### Resultado

elemento 1	elemento 2	elemento 3	elemento 4
elemento 5	elemento 6	elemento 7	elemento 8
elemento 9	elemento 10	elemento 11	elemento 12

## Pseudo-clase :nth-last-child

Esta *pseudo-clase* actúa igual que *nth-child*, por lo que las dos utilizan el mismo sistema para seleccionar elementos (palabra clave, entero o fórmula). La única diferencia es que empieza por el último de los elementos, es decir, si el resultado del cálculo es 1 se seleccionaría el último elemento, si es 2 el penúltimo, si es 3 el antepenúltimo y así sucesivamente. Veamos un ejemplo usando una de las fórmulas utilizadas con *nth-child*. Si usamos la fórmula **4n** estos serían los cálculos y los elementos a seleccionar:

Fórmula aplicada	Elemento a seleccionar
$4 \times 0 = 0$	ninguno
$4 \times 1 = 4$	9
$4 \times 2 = 8$	5
$4 \times 3 = 12$	1
...	...

Por lo tanto, esta fórmula seleccionaría del *HTML* con el que estamos trabajando los elementos 9, 5 y 1. Observemos el código *CSS* y el resultado de esta fórmula:

### Código CSS

```
.box:nth-last-child(4n) {  
    background: #FFF;  
    color: #232b2b;  
}
```

### Resultado

elemento 1	elemento 2	elemento 3	elemento 4
elemento 5	elemento 6	elemento 7	elemento 8
elemento 9	elemento 10	elemento 11	elemento 12



# Pseudo-clase :nth-of-type

Esta *pseudo-clase* actúa muy parecido a *:nth-child* pero con una diferencia. Mientras *:nth-child* selecciona elementos sin importar su tipo, es decir, cualquier hijo del contenedor, *:nth-of-type* tiene en cuenta el tipo de elemento a seleccionar. Tomando en cuenta la estructura *HTML* siguiente, analicemos el código *CSS* usando la fórmula **2n** y el resultado final:

## Código HTML

```
<div id="container">

  <div class="type1">div 1</div>
  <section class="type2">section 1</section>
  <div class="type1">div 2</div>
  <section class="type2">section 2</section>
  <div class="type1">div 3</div>
  <section class="type2">section 3</section>
  <div class="type1">div 4</div>
  <section class="type2">section 4</section>
  <div class="type1">div 5</div>
  <section class="type2">section 5</section>
  <div class="type1">div 6</div>
  <section class="type2">section 6</section>

</div>
```

## Código CSS

```
section:nth-of-type(2n) {

  background: #FFF;
  color: #232b2b;

}
```

## Resultado

div 1	section 1	div 2	section 2
div 3	section 3	div 4	section 4
div 5	section 5	div 6	section 6

En el resultado podemos ver que al situar la fórmula **2n** se han seleccionado los elementos de tipo *section* 2, 4 y 6 aunque en realidad en el árbol del *DOM* estos elementos corresponden al 4, 8 y 12 respectivamente, ya que la pseudo-clase *:nth-of-type* sólo toma en cuenta el tipo de elemento al que se le ha aplicado.

# Pseudo-clase :nth-last-of-type

Esta *pseudo-clase* es igual que la anterior, lo que al igual que *:nth-last-child* empieza la selección por el último elemento. Por ejemplo, tomando el mismo *HTML* anterior y usando la misma fórmula, este sería el resultado usando esta *pseudo-clase*:

## Código CSS

```
#container div:nth-last-of-type(2n) {

  background: #FFF;
  color: #232b2b;

}
```

## Resultado

div 1	section 1	div 2	section 2
div 3	section 3	div 4	section 4
div 5	section 5	div 6	section 6

Como hemos podido ver, el uso de estas *pseudo-clases* para seleccionar elementos en *CSS3* tiene infinitas posibilidades.

<http://xprimiendo.com/pseudo-clases-css3-seleccionar-elementos/>