

Pseudo Clases

Dentro de CSS existen clases especiales a las cuales podemos denominar pseudoclasas, estas nos ayudarán a modificar el comportamiento de los elementos HTML.

Por ejemplo, se pueden usar para:

1. Seleccionar N cantidad de elementos para que se agreguen nuevas propiedades CSS.
2. Dependiendo de la posición de un elemento (Inicio/Final) en una lista deberá agregar X propiedad CSS.
3. Cambiar una propiedad de un elemento cuando un usuario pase el mouse sobre él.

Pseudoclases

La estructura para utilizar una pseudoclase es la siguiente:

```
selector : pseudoClase {  
    prop1: valor;  
    prop2:valor;  
    prop3:valor;  
}
```

Observar que una pseudoClase inicia siempre con un “:”

Seleccionar un elemento específico

Pseudoclase: nth-child(**número**)

Observación 1: El primer Hijo se encuentra en la posición 1, no 0.

Observación 2: nth-child Busca siempre etiquetas que contengan a otras etiquetas y respecto a eso selecciona.



Escriba el siguiente código

```
<body>  
  <p>1.- Párrafo</p>  
  <p>2.- Párrafo</p>  
  <span>3.- Span</span>  
  <div>4.- Div</div>  
  <span>5.- Span</span>  
  <p>6.- Párrafo</p>  
</body>
```

Ejercicio:

Pintar el hijo número 5 de Body.

Observación: Fijarse que el símbolo “:” de la pseudoclase va separado de body.

```
body :nth-child(5) {  
    background-color:  coral;  
    color:  white;  
}
```

1.- Párrafo

2.- Párrafo

3.- Span

4.- Div

5.- Span

6.- Párrafo



Ejercicio

Pintar el hijo número 1 y 4 de Body.





```
<body>  
  <p>1.- Párrafo</p>  
  <p>2.- Párrafo</p>  
  <span>3.- Span</span>  
  <div>4.- Div</div>  
  <span>5.- Span</span>  
  <p>6.- Párrafo</p>  
</body>
```

Solución

Solución 1

```
body :nth-child(1),  
body :nth-child(4) {  
    background-color:  coral;  
    color:  white;  
}
```

Solución 2

```
body :nth-child(1) {  
    background-color:  coral;  
    color:  white;  
}  
  
body :nth-child(4) {  
    background-color:  coral;  
    color:  white;  
}
```




Ejercicio

Escriba el siguiente código y observe su comportamiento.

¿Qué sucede si cambio el valor de 3?

Fijarse que la palabra span va pegado a “:”.

```
<body>
  <p>1.- Párrafo</p>
  <p>2.- Párrafo</p>
  <span>3.- Span</span>
  <div>4.- Div</div>
  <span>5.- Span</span>
  <p>6.- Párrafo</p>
</body>
```



```
body span:nth-child(3) {
  background-color:  coral;
  color:  white;
}
```

Respuesta

Lo que quiere decir es:

Span debe estar en el interior de Body y ser sí o sí el **Hijo número tres**. En caso contrario, no se debe aplicar las propiedades.



Por lo tanto, ¿Qué sucedería si se agrega un div como el primer Hijo de Body?

```
body span:nth-child(3) {  
  background-color:  coral;  
  color:  white;  
}
```

Respuesta



La etiqueta span ya no se pintaría, porque no cumple la segunda condición (Estar exactamente en esa posición).

Bajo el nuevo cambio, ¿Cómo se podrá pintar nuevamente la etiqueta?

```
body span:nth-child(3) {  
    background-color:  coral;  
    color:  white;  
}
```

Respuesta

Actualizando su nueva posición.

```
body span:nth-child(4) {  
  background-color:  coral;  
  color:  white;  
}
```

Notación funcional

Representa los elementos cuya posición numérica en una serie de hermanos coincide con el patrón **$An+B$** , para cada entero positivo o valor cero de n . El índice del primer elemento es 1. Los valores A y B deben ser ambos enteros.



Observación: **N** inicia en cero.

Ejercicio

Pintar todos los elementos.

Solución

```
<body>
  <p>1.- Párrafo</p>
  <p>2.- Párrafo</p>
  <span>3.- Span</span>
  <div>4.- Div</div>
  <span>5.- Span</span>
  <p>6.- Párrafo</p>
</body>
```



```
body :nth-child(n) {
  background-color:  coral;
  color:  white;
}
```

Ejercicio

Pinte solamente las etiquetas <p>.

```
<body>  
  <p>1.- Párrafo</p>  
  <p>2.- Párrafo</p>  
  <span>3.- Span</span>  
  <div>4.- Div</div>  
  <span>5.- Span</span>  
  <p>6.- Párrafo</p>  
</body>
```

Solución

```
body p:nth-child(n) {  
  background-color:  coral;  
  color:  white;  
}
```


Ejercicio



Utilizando el ejercicio anterior.

Utilice el siguiente Html.

Pinte solamente las `<p>` que son hijos directo de `body`.

```
<body>
  <p>1.- Párrafo</p>
  <p>2.- Párrafo</p>
  <span>3.- Span
    |   <p>Especial</p>
  </span>
  <div>4.- Div</div>
  <span>5.- Span
    |   <p>Especial</p>
  </span>
  <p>6.- Párrafo</p>
</body>
```

Solución



```
body > p:nth-child(n) {  
  background-color:  coral;  
  color:  white;  
}
```

Ejercicio

Escriba el siguiente código CSS e indique su comportamiento.

Observación: Ahora body está apegado al símbolo “.”.

Prueba diferentes valores.

```
body:nth-child(2) {  
    background-color:  teal;  
    color:  white;  
}
```

Solución

Lo que indica el selector es:

Para toda etiqueta `<body>` que se encuentre como Hijo número dos, se le deberá aplicar el color de fondo teal y letras blancas.

Recordar que `body` está anidado en el interior de la etiqueta `html` y su primer hermano es `head`.

```
<html lang="en">
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

```
body:nth-child(2) {
  background-color: teal;
  color: white;
}
```

Ejercicio – Pintar Hijos Pares

```
<body>
  <p>1.- Párrafo</p>
  <p>2.- Párrafo</p>
  <span>3.- Span </span>
  <div>4.- Div</div>
  <span>5.- Span </span>
  <p>6.- Párrafo</p>
</body>
```

Recordar que bajo notación funcional está permitido usar **An+B**.



En matemáticas se dice que “Todo número entero multiplicado por 2 es par.”

Por lo tanto.

$2n$ cumple la condición.

¿Cómo quedará el selector?



Solución

```
body > :nth-child(2n) {  
  background-color:  coral;  
  color:  white;  
}
```

Ejercicio

Seleccionar todas las etiquetas impares.



Solución



```
body > :nth-child(2n + 1) {  
  background-color:  coral;  
  color:  white;  
}
```


Bonus

Para obtener el primer o último elemento, existe las pseudo clases first-child y last-child.

```
<body>
  <section>
    <p>1.- Párrafo</p>
    <p>2.- Párrafo</p>
    <span>3.- Span </span>
    <div>4.- Div</div>
    <span>5.- Span </span>
    <p>6.- Párrafo</p>
  </section>
</body>
```

```
section > :first-child {
  background-color:  coral;
  color:  white;
}
```

```
section > :last-child {
  background-color:  coral;
  color:  white;
}
```

Observación: Se cambió a section por motivo a que last-child no pintaría el sexto párrafo, eso se debe a que si estás trabajando con Live Server incluirá un etiqueta al final de body, dejando al sexto párrafo como penúltimo hijo y no último.

Si deseas continuar con la versión anterior y probar, se deberá copiar la ruta del archivo html y pegarla en el navegador de forma directa.

Ejercicio

Pintar solamente las etiquetas span.

Para este caso nth-child es de uso general, por lo que se requiere cambiar a nth-of-type(**número**).

nth-child(número)	nth-of-type(número)
Número: Indica la posición de la etiqueta .	Número: Indicará el número de vez encontrado.

Ejemplo

Seleccionar la segunda span de Body

```
<body>
  <section>
    <p>1.- Párrafo</p>
    <p>2.- Párrafo</p>
    <span>3.- Span </span>
    <div>4.- Div</div>
    <span>5.- Span </span>
    <p>6.- Párrafo</p>
  </section>
</body>
```



```
section > span:nth-of-type(2) {
  background-color: coral;
  color: white;
}
```



Ejercicio

1. Pintar todas las etiquetas span pares de un color.
2. Pintar todas las etiquetas p desde la posición 3 en adelante de un color.

```
<body>
  <span>1 Span</span>
  <p>1 P</p>
  <span>2 Span</span>
  <p>2 P</p>
  <span>3 Span</span>
  <p>3 P</p>
  <span>4 Span</span>
  <p>4 P</p>
  <span>5 Span</span>
  <p>5 P</p>
  <span>6 Span</span>
  <p>6 P</p>
  <span>7 Span</span>
  <p>7 P</p>
  <span>8 Span</span>
  <p>8 P</p>
</body>
```

Solución

```
body > span:nth-of-type(2n) {  
  background-color:  coral;  
  color:  white;  
}
```

```
body > p:nth-of-type(n + 3) {  
  background-color:  teal;  
  color:  white;  
}
```

Ejercicio



Pintar de un color las dos últimas etiquetas span.

Utilice la pseudoclase:

`nth-last-of-type(número)`

```
<body>
  <span>1 Span</span>
  <p>1 P</p>
  <span>2 Span</span>
  <p>2 P</p>
  <span>3 Span</span>
  <p>3 P</p>
  <span>4 Span</span>
  <p>4 P</p>
  <span>5 Span</span>
  <p>5 P</p>
  <span>6 Span</span>
  <p>6 P</p>
  <span>7 Span</span>
  <p>7 P</p>
  <span>8 Span</span>
  <p>8 P</p>
</body>
```



Solución

```
body > span:nth-last-of-type(1),  
body > span:nth-last-of-type(2) {  
    background-color:  coral;  
    color:  white;  
}
```

Negación

A un conjunto de elementos es posible filtrar un resultado con la pseudoclase **:not()**

Para todos los hijos de body menos los <p>
y

```
body > :nth-child(n):not(p):not(span) {  
  background-color:  coral;  
  color:  white;  
}
```

Observación: Después de un :not() se puede agregar más not() para aumentar la filtración.

.not()

Valores aceptados (Algunos ejemplos):

:not(**#mild**) : Si una etiqueta incluye el Id “mild”, deberá ser filtrado.

:not(**.miClase**) : Si una etiqueta incluye la clase “.miClase”, deberá ser filtrado.

:not(**span**) : Si es una etiqueta de tipo , deberá ser filtrado.

:not(**:first-child**) : Filtra el primer elemento.

:not(**:last-child**) : Filtra el último elemento.

Observación: Solamente acepta valores simples.
Algo como #mild.miClase.miClase2, no funcionará.
Solamente un selector.

Ejercicio

1. Todas las etiquetas que contengan la clase m1 o m2 deberán ser ignoradas.
2. Todas las etiquetas que sean de tipo i deberán ser ignoradas.
3. El resto de etiquetas deberán pintarse de color “teal”

```
<body>
  <i>Modo i</i>
  <p class="m1 m2">1 P</p>
  <span class="m1">1 Span</span>
  <p class="m1">2 P</p>
  <span>2 Span</span>
  <p class="m1">3 P</p>
  <i>Modo i</i>
  <p class="m1 m2">4 P</p>
  <span>3 Span</span>
  <p>5 P</p>
  <span class="m1">4 Span</span>
  <p class="m2">6 P</p>
  <i>Modo i</i>
  <p class="m1 m2">7 P</p>
  <span class="m2">5 Span</span>
  <p>8 P</p>
</body>
```

Solución

```
body > :nth-child(n):not(i):not(.m1):not(.m2) {  
  background-color: ■ teal;  
  color: ■ white;  
}
```

Ejercicio

Pinte con el mismo color el primer y último elemento de cada tipo de etiqueta.

Cada grupo debe ser pintado de un color diferente al resto.

Utilice las siguientes pseudoclase:

:first-of-type

:last-of-type

```
<body>
  <i>Modo i</i>
  <p>1 P</p>
  <span>1 Span</span>
  <p>2 P</p>
  <span>2 Span</span>
  <p>3 P</p>
  <i>Modo i</i>
  <p>4 P</p>
  <span>3 Span</span>
  <p>5 P</p>
  <span>4 Span</span>
  <p>6 P</p>
  <i>Modo i</i>
  <p>7 P</p>
  <span>5 Span</span>
  <p>8 P</p>
</body>
```

Modo i

1 P

1 Span

2 P

2 Span

3 P

Modo i

4 P

3 Span

5 P

4 Span

6 P







Modo i

7 P

5 Span

8 P

Solución

```
body > span:first-of-type,  
body > span:last-of-type {  
    background-color:  teal;  
    color:  white;  
}  
  
body > i:first-of-type,  
body > i:last-of-type {  
    background-color:  coral;  
    color:  white;  
}  
  
body > p:first-of-type,  
body > p:last-of-type {  
    background-color:  indigo;  
    color:  white;  
}
```

Ejercicio

Utilizando pseudo clases, diseñe la siguiente tabla:

1. Utilice `<table>`, `<tr>`, `<th>`, `<td>`.
2. No es posible asignar a las etiquetas **clases** y **ids**.

Material	Temperatura	Velocidad (m/s)
Aire	0°C	331,5°C
Agua	25°C	1.593
Acero	-	6.100
Aluminio	-	6.400
Cadmio	-	12.400
Estudio desarrollado por: ...		

Observación: Cuidado para los que usen el símbolo “>”, ya que por defecto Chrome añade `<tbody>` a la estructura de table.

Solución

```
table,
tr,
th,
td {
    border: 1px solid black;
    color: white;
    text-align: center;
}

table {
    background-color: #585858;
    border-collapse: collapse;
    width: 300px;
}

table tr:first-of-type {
    background-color: #008790;
}

table tr:last-of-type {
    background-color: #00393d;
}

table tr:nth-child(2n):not(:first-of-type):not(:last-of-type) {
    background-color: #0f4c75;
}

table tr:nth-child(2n + 1):not(:first-of-type):not(:last-of-type) {
    background-color: #3282b8;
}
```

Pseudoclase hover

La pseudoclase hover de CSS, se activa generalmente cuando el usuario se desplaza sobre un elemento con el cursor (puntero del mouse).

Ejercicio

Escriba el siguiente ejercicio.

Por último, posicione el puntero sobre el div para notar el efecto.

```
.miBoton {  
  padding: 10px 20px 10px 20px;  
  background-color: teal;  
  color: white;  
  width: fit-content;  
  font-weight: bolder;  
}  
  
.miBoton:hover {  
  background-color: rgb(0, 61, 61);  
}
```

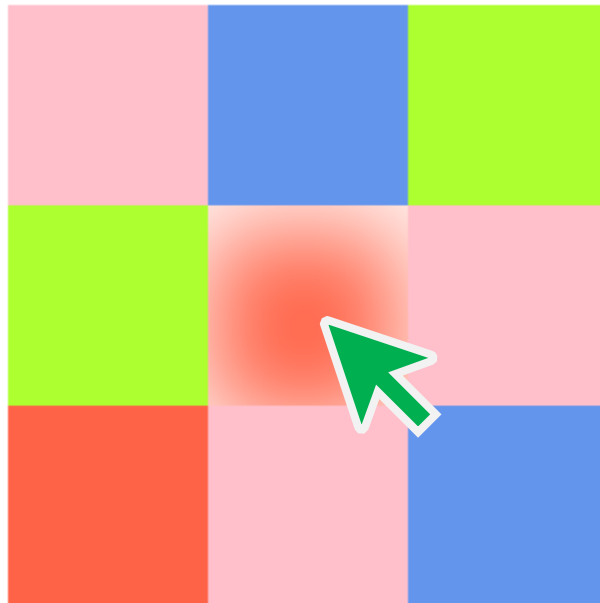
```
<div class="miBoton">Mi botón</div>
```

Ejercicio – Modo Disco

Desarrolle un Grid 3x3.

Cada vez que se posicione el mouse en el interior de un div este deberá brillar más respecto a su color base. (Usar un color más claro o utilizar sombreado).

Sugerencia: <https://www.cssmatic.com/es/box-shadow>



Solución

```
.grid {
  display: grid;
  width: 300px;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: repeat(3, 100px);
}

.red:hover,
.blue:hover,
.green:hover,
.pink:hover {
  -webkit-box-shadow:
    inset -2px 10px 50px 1px
    ■ rgba(255, 255, 255, 1);
  -moz-box-shadow:
    inset -2px 10px 50px 1px
    ■ rgba(255, 255, 255, 1);
  box-shadow:
    inset -2px 10px 50px 1px
    ■ rgba(255, 255, 255, 1);
}
```

```
.red {
  background-color: ■ tomato;
}

.blue {
  background-color: ■ cornflowerblue;
}

.green {
  background-color: ■ greenyellow;
}

.pink {
  background-color: ■ pink;
}
```

```
<div class="grid">
  <div class="pink"></div>
  <div class="blue"></div>
  <div class="green"></div>
  <div class="green"></div>
  <div class="red"></div>
  <div class="pink"></div>
  <div class="red"></div>
  <div class="pink"></div>
  <div class="blue"></div>
</div>
```