

Selectores de CSS 3

La versión CSS 3 incluye todos los selectores de CSS 2.1 y añade otras decenas de selectores, pseudo-clases y pseudo-elementos.

En primer lugar, CSS 3 añade tres nuevos selectores de atributos:

⌚ `elemento[atributo^="valor"]`, selecciona todos los elementos que disponen de ese atributo y cuyo valor comienza exactamente por la cadena de texto indicada.

⌚ `elemento[atributo$="valor"]`, selecciona todos los elementos que disponen de ese atributo y cuyo valor termina exactamente por la cadena de texto indicada.

⌚ `elemento[atributo*="valor"]`, selecciona todos los elementos que disponen de ese atributo y cuyo valor contiene la cadena de texto indicada.

De esta forma, se pueden crear reglas CSS tan avanzadas como las siguientes:

```
/* Selecciona todos los enlaces que apuntan a una dirección de correo electrónico */
a[href^="mailto:"] { ... }

/* Selecciona todos los enlaces que apuntan a una página HTML */
a[href$=".html"] { ... }

/* Selecciona todos los títulos h1 cuyo atributo title contenga la palabra "capítulo" */
h1[title*="capítulo"] { ... }
```

Otro de los nuevos selectores de CSS 3 es el "*selector general de elementos hermanos*", que generaliza el selector adyacente de CSS 2.1. Su sintaxis es `elemento1 ~ elemento2` y selecciona el `elemento2` que es hermano de `elemento1` y se encuentra detrás en el código HTML. En el selector adyacente la condición adicional era que los dos elementos debían estar uno detrás de otro en el código HTML, mientras que ahora la única condición es que uno esté detrás de otro.

Si se considera el siguiente ejemplo:

```
h1 + h2 { ... } /* selector adyacente */
h1 ~ h2 { ... } /* selector general de hermanos */

<h1>...</h1>
<h2>...</h2>
<p>...</p>
<div>
  <h2>...</h2>
</div>
<h2>...</h2>
```

El primer selector (`h1 + h2`) sólo selecciona el primer elemento `<h2>` de la página, ya que es el único que cumple que es hermano de `<h1>` y se encuentra justo detrás en el código HTML. Por su parte, el segundo selector (`h1 ~ h2`) selecciona todos los elementos `<h2>` de la página salvo el segundo. Aunque el segundo `<h2>` se encuentra detrás de `<h1>` en el código HTML, no son elementos hermanos porque no tienen el mismo elemento padre.

Los pseudo-elementos de CSS 2.1 se mantienen en CSS 3, pero cambia su sintaxis y ahora se utilizan `::` en vez de `:` delante del nombre de cada pseudo-elemento:

🕒 `::first-line`, selecciona la primera línea del texto de un elemento.

🕒 `::first-letter`, selecciona la primera letra del texto de un elemento.

🕒 `::before`, selecciona la parte anterior al contenido de un elemento para insertar nuevo contenido generado.

🕒 `::after`, selecciona la parte posterior al contenido de un elemento para insertar nuevo contenido generado.

CSS 3 añade además un nuevo pseudo-elemento:

🕒 `::selection`, selecciona el texto que ha seleccionado un usuario con su ratón o teclado.

Las mayores novedades de CSS 3 se producen en las pseudo-clases, ya que se añaden 12 nuevas, entre las cuales se encuentran:

🕒 `elemento:nth-child(numero)`, selecciona el elemento indicado pero con la condición de que sea el hijo *enésimo* de su padre. Este selector es útil para seleccionar el segundo párrafo de un elemento, el quinto elemento de una lista, etc.

🕒 `elemento:nth-last-child(numero)`, idéntico al anterior pero el número indicado se empieza a contar desde el último hijo.

🕒 `elemento:empty`, selecciona el elemento indicado pero con la condición de que no tenga ningún hijo. La condición implica que tampoco puede tener ningún contenido de texto.

🕒 `elemento:first-child` y `elemento:last-child`, seleccionan los elementos indicados pero con la condición de que sean respectivamente los primeros o últimos hijos de su elemento padre.

🕒 `elemento:nth-of-type(numero)`, selecciona el elemento indicado pero con la condición de que sea el *enésimo* elemento hermano de ese tipo.

🕒 `elemento:nth-last-of-type(numero)`, idéntico al anterior pero el número indicado se empieza a contar desde el último hijo.

Algunas pseudo-clases como `:nth-child(numero)` permiten el uso de expresiones complejas para realizar selecciones avanzadas:

```
li:nth-child(2n+1) { ... } /* selecciona todos los elementos impares de una
lista */
li:nth-child(2n) { ... } /* selecciona todos los elementos pares de una lista */

/* Las siguientes reglas alternan cuatro estilos diferentes para los párrafos
*/

p:nth-child(4n+1) { ... }
p:nth-child(4n+2) { ... }
p:nth-child(4n+3) { ... }
p:nth-child(4n+4) { ... }
```

Empleando la pseudo-clase `:nth-of-type(numero)` se pueden crear reglas CSS que alternen la posición de las imágenes en función de la posición de la imagen anterior:

```
img:nth-of-type(2n+1) { float: right; }
img:nth-of-type(2n) { float: left; }
```

Otros de los nuevos selectores que incluirá CSS 3 son :

not(), que se puede utilizar para seleccionar todos los elementos que no cumplen con la condición de un selector:

```
:not(p) { ... } /* selecciona todos los elementos de la página que no sean
párrafos */
:not(#especial) { ... } /* selecciona cualquier elemento cuyo atributo id no sea
"especial" */
```

checked

La pseudo-clase `:checked` de CSS representa cualquier radio (`<input type="radio">`), checkbox (`<input type="checkbox">`) u option (`<option>` en un elemento `<select>`) que está marcado o conmutado a un estado on.

```
/* Coincide con cualquier checked/selected radio, checkbox, u option */
:checked {
    margin-left: 25px;
    border: 1px solid blue;
}
```

Ejemplos

HTML

```
<div>
  <input type="radio" name="input" id="yes" />
  <label for="yes">Si</label>

  <input type="radio" name="input" id="no" />
  <label for="no">No</label>
</div>

<div>
  <input type="checkbox" name="checkbox" id="in" />
  <label for="in">Check me!</label>
</div>

<select name="select" id="fruta">
  <option value="Manzana">Manzana</option>
  <option value="Uva">Uva</option>
  <option value="Pera">Pera</option>
</select>
```

CSS

```
div,select {
  margin: 8px;
}
/* Etiquetas para entradas marcadas */
input:checked + label {
  color: red;
}
/* Elemento Radio, cuando está marcado */
input[type="radio"]:checked {
  box-shadow: 0px 0 5px 1px orange;
}
/* Elemento Checkbox, cuando está marcado */
input[type="checkbox"]:checked {
  box-shadow: 0 0 0px 2px pink;
}
/* Elementos Option, cuando se seleccionan */
option:checked {
  color: blue;
}
```

Resultado



Si No

☒ Check me!

Manzana ▼

Manzana

Uva

Pera

Alternar elementos con un checkbox oculto

Este ejemplo utiliza la pseudoclase `:checked` para permitir al usuario alternar contenido según el estado de un checkbox, todo sin usar JavaScript.

HTML

```
<input type="checkbox" id="muestra" />

<table>
  <caption>LUNES</caption>
  <tr>
    <th>Hora</th>
    <th>Módulo</th>
    <th>Aula</th>
  </tr>
  <tr class="oculta">
    <td>8-9</td>
    <td> Matemáticas</td>
    <td> 111</td>
  </tr>
  <tr>
    <td>9-10</td>
    <td> física </td>
    <td> 211</td>
  </tr>
  <tr>
    <td>10-11</td>
    <td> Lengua</td>
    <td> 425</td>
  </tr>
  <tr class="oculta">
    <td> 11-12</td>
    <td> Inglés</td>
    <td> 311</td>
  </tr>
  <tr class="oculta">
    <td> 12-13</td>
    <td> Historia</td>
    <td> 225</td>
  </tr>
</table>

<label for="muestra" >Toggle hidden rows</label>
```

CSS

```
/* Oculta el checkbox */
#muestra {
  display: none;
}
/* Estilo del label */
label {
  display: inline-block;
  margin-top: 12px;
  padding: 5px 11px;
  box-shadow: 2px 3px 3px #ccc;
  background-color: lightcoral;
  border: 1px solid;
  border-radius: 6px;
}
/* Oculta contenido cuando el checkbox está checked */
#muestra:checked ~ table.oculta {
  visibility: collapse;
}
/* Estilo del label cuando el checkbox está checked */
#muestra:checked ~ label {
  background-color: #ccc;
  box-shadow: 2px 3px 3px lightcoral;
  color: white;
}
```

Resultado

LUNES			LUNES		
Hora	Módulo	Aula	Hora	Módulo	Aula
8-9	Máticas	111	9-10	Física	211
9-10	Física	211	10-11	Lengua	425
10-11	Lengua	425			
11-12	Inglés	311			
12-13	Historia	225			

Toggle hidden rows

Toggle hidden rows

focus

La pseudo-clase `:focus` CSS representa un elemento (como una entrada de formulario) que ha recibido el foco. Generalmente se activa cuando el usuario hace clic, toca un elemento o lo selecciona con la tecla "Tab" del teclado.

HTML

```
<input
  class="red"
  value="Voy a ser de color rojo cuando enfoque." /><br />
<input class="blue" value="Voy a ser de color azul cuando enfoque." />
```

CSS

```
.red:focus {  
  background: yellow;  
  color: red;  
}  
  
.blue:focus {  
  background: yellow;  
  color: blue;  
}
```

Resultado

Voy a ser de color rojo cuar
Voy a ser de color azul cuai

Voy a ser de color rojo cuar
Voy a ser de color azul cuai

Voy a ser de color rojo cuar
Voy a ser de color azul cuai