

4. Pseudo-clases y pseudo-elementos en CSS

Gracias a las pseudo-clases y los pseudo-elementos de CSS podemos **realizar una selección más específica de los elementos a los que queremos aplicar un cierto estilo sin necesidad de crear una clase concreta.**

4.1. Pseudo-clases para la selección de hijos o hermanos

Las pseudo-clases más importantes para la selección de hijos o hermanos son las que se muestran en la siguiente tabla.

Pseudo-clase	Descripción
:first-child	Primer hijo
:last-child	Último hijo
:first-of-type	Primer hermano de su tipo
:last-of-type	Último hermano de su tipo
:only-child	Hijos únicos
:only-of-type	Únicos hermanos de su tipo
:empty	Elementos que no tienen hijos
:nth-child(n)	Enésimo elemento hijo
:nth-last-child(n)	Enésimo elemento hijo contando desde el último
:nth-of-type(n)	Enésimo hermano de su tipo
:nth-last-of-type(n)	Enésimo hermano de su tipo comenzando desde el último

Ejemplo pseudoclasas first-child, last-child y only-child:

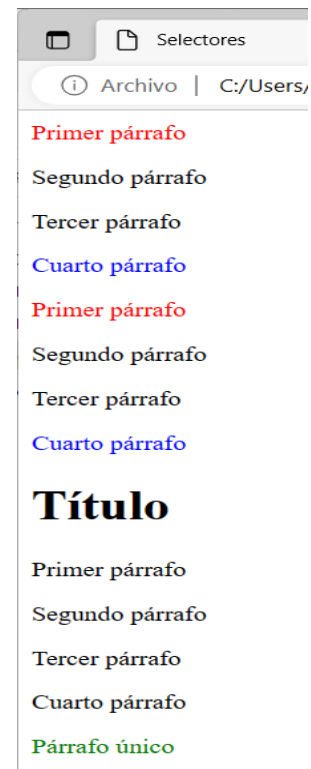
Vamos a utilizar las propiedades first-child, last-child y only-child para seleccionar los párrafos que sean el primer y el último hijo de un contenedor o que sea hijo único.

HTML	CSS
<pre><div> <p>Primer párrafo</p> <p>Segundo párrafo</p> <p>Tercer párrafo</p> <p>Cuarto párrafo</p> </div> <p>Primer párrafo</p> <p>Segundo párrafo</p> <p>Tercer párrafo</p> <p>Cuarto párrafo</p> <div> <h1>Título</h1> <p>Primer párrafo</p> <p>Segundo párrafo</p></pre>	<pre>p:first-child { color: red; } p:last-child { color: blue; } p:only-child { color: green; }</pre>

```

    <p>Tercer párrafo</p>
    <p>Cuarto párrafo</p>
  <div>
    <p>Párrafo único</p>
  </div>
</div>

```



Ejemplo pseudo-clase nth-child:

Veamos cómo funciona la pseudo-clase `nth-child` mediante un ejemplo en el que aprovechamos la jerarquía de los elementos para referenciar una etiqueta. En el siguiente código tenemos cuatro elementos `<p>` que son hermanos entre sí e hijos del elemento `<div>`.

```

<div>
  <p>Primer párrafo</p>
  <p>Segundo párrafo</p>
  <p>Tercer párrafo</p>
  <p>Cuarto párrafo</p>
</div>

```

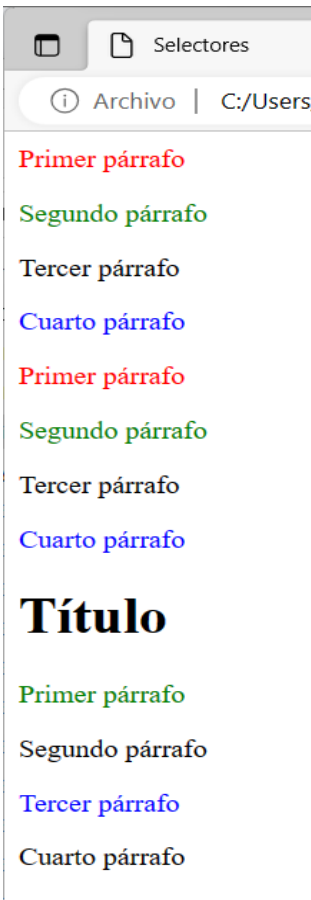
Utilizando la pseudo-clase **`nth-child()`** podemos seleccionar un hijo específico. El número que indiquemos entre paréntesis será la posición del hijo.

En el siguiente código, seleccionaremos todos los párrafos que sean el primer, segundo y cuarto hijo de un contenedor (`div`, `span`, etc.).

```


p:nth-child(1){ color: red; }
p:nth-child(2){ color: green; }
p:nth-child(4){ color: blue; }

```

HTML	CSS
<pre> <div> <p>Primer párrafo</p> <p>Segundo párrafo</p> <p>Tercer párrafo</p> <p>Cuarto párrafo</p> </div> <p>Primer párrafo</p> <p>Segundo párrafo</p> <p>Tercer párrafo</p> <p>Cuarto párrafo</p> <div> <h1>Título</h1> <p>Primer párrafo</p> <p>Segundo párrafo</p> <p>Tercer párrafo</p> <p>Cuarto párrafo</p> </div> </pre>	<pre> p:nth-child(1){ color: red; } p:nth-child(2){ color: green; } p:nth-child(4){ color: blue; } </pre> 

Ejemplo odd (impar) y even (par):

Los valores *odd* y *even* son muy útiles para seleccionar elementos pares e impares. El valor *even* representa elementos en posición par y el valor *odd* representa los elementos en posición impar. En el siguiente ejemplo en el que se ha creado una tabla se muestran en verde las filas impares y en rojo las filas pares.

HTML	CSS
<pre> <table> <tr> <th>Fruta</th> <th>Unidades</th> </tr> <tr> <td>Manzanas</td> <td>40</td> </tr> <tr> <td>Peras</td> <td>45</td> </tr> <tr> <td>Plátanos</td> </pre>	<pre> tr:nth-child(odd){ color: green;} tr:nth-child(even){ color: red;} table{border-collapse: collapse;} tr td, tr th{border: 1px solid black;} </pre> 

```

        <td>30</td>
    </tr>
    <tr>
        <td>Fresas</td>
        <td>100</td>
    </tr>
    <tr>
        <td>Naranjas</td>
        <td>200</td>
    </tr>
</table>


```

4.2. Pseudo-clases para los estados de un elemento

Podemos utilizar diferentes pseudo-clases **para definir las propiedades de ciertos elementos con diferentes estados**. Uno de los **usos más comunes es en los enlaces <a>** aunque se pueden aplicar en otros elementos.

Pseudo-clase	Descripción
:link	No visitado por el usuario
:visited	Visitado por el usuario
:hover	Modifica el estilo cuando un elemento apuntador pasa por encima
:active	Se activa cuando el usuario pulsa el elemento
:focus	Se activa cuando tiene el foco sobre el elemento

Ejemplo:

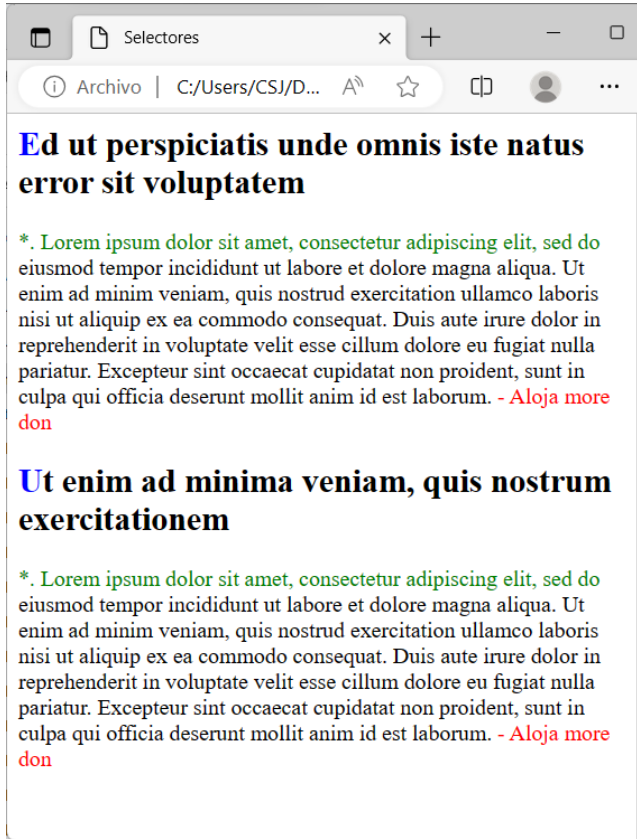
HTML	CSS
<pre> w3schools .com wikipedia .org IES Ágora </pre>	<pre> a:hover{ /*Elemento apuntador pasa por encima*/ text-decoration: overline; } a:active{ /*Se activa cuando el usuario pulsa el elemento*/ background-color: yellow; } a:link{ /*No visitado por el usuario*/ color: green; } a:visited{ /*Visitado por el usuario*/ color: brown; } </pre> 

4.3. Pseudo-elementos

A diferencia de las pseudo-clases, los pseudo-elementos **no describen un estado** sino que **permiten añadir estilos a una parte específica del documento**. Por ejemplo, el pseudo-elemento `::first-line` selecciona la primera línea del elemento indicado por el selector.

Pseudo-elemento	Descripción
<code>::first-line</code>	Primera línea de texto de un elemento
<code>::first-letter</code>	Primera letra de la primera línea de texto de un elemento
<code>::before</code>	Añade contenido al principio del documento
<code>::after</code>	Añade contenido al final del documento
<code>::selection</code>	Coge la porción del texto que se está seleccionando por el usuario

Ejemplo:

HTML	CSS
<pre><h2>Ed ut perspiciatis unde omnis iste natus error sit voluptatem</h2> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p> <h2>Ut enim ad minima veniam, quis nostrum exercitationem</h2> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p></pre>	<pre>p::first-line{ color: green;} h2::first-letter{ color: blue;} p::after { content: " - Aloja more don"; color: red;} p::before { content: "*. ";} p::selection { color: brown;}</pre> 

4.4. Prefijos CSS de los navegadores

Actualmente los navegadores tienen implementadas muchas de las nuevas características de CSS3 utilizando sus propias versiones de cada propiedad mediante prefijos. Esto se hace así para **evitar los posibles errores ocasionados por las primeras implementaciones que aún no son estables**. Por ello, los navegadores proporcionan valores utilizando sus prefijos propios y una declaración sin prefijo.

Después de un tiempo, **cuando las especificaciones son estables, se eliminarán las propiedades con prefijo**. Los prefijos para los navegadores más comunes son los siguientes:

Prefijo	Navegador
-moz-	Firefox
-webkit-	Safari y Chrome
-o-	Opera
-khtml-	Konqueror
-ms-	Internet Explorer y Microsoft Edge

Así por ejemplo para transformar un elemento en Firefox, es necesario utilizar la propiedad `-moz-transform`; en los navegadores basados en WebKit, como Safari y Google Chrome, se utiliza la propiedad `-webkit-transform`. Por este motivo, no es de extrañar que en algunos casos, tengamos que añadir hasta cuatro líneas de código para una única propiedad CSS.

```
transform: rotate(20deg);
```

```
-moz-transform: rotate(20deg); /* Firefox */
```

```
-webkit-transform: rotate(20deg); /* Safari y Chrome */
```

```
-o-transform: rotate(20deg); /* Opera */
```

```
-ms-transform: rotate(20deg); /* Internet Explorer y Microsoft Edge */
```

Es importante destacar que con el tiempo, muchos navegadores han adoptado estándares y ya no requieren prefijos para las propiedades CSS más comunes. Sin embargo, en algunos casos, especialmente cuando se trabaja con características experimentales, los prefijos todavía pueden ser útiles para garantizar la compatibilidad entre navegadores. Se pueden ver los navegadores que soportan una determinada propiedad CSS o un elemento HTML5 en la página caniuse.com.