

Agrupación de selectores

Muchas veces queremos aplicar el mismo estilo a diferentes elementos, CSS nos permite crear reglas con selectores múltiples para poder así aplicar las mismas declaraciones a distintos grupos de elementos, de esta manera se facilita el mantenimiento y ahorramos líneas en las hojas de estilo:

Sintaxis

```
Elemento1, elemento2,...,elementoN{
propiedad:valor;
}
```

Ejemplo

```
<html>
<head><title> selectores múltiples</title></head>
<style>
  h1,h2,h3{
    color:cadetblue;
  }
</style>
<body>
  <h1>CSS</h1>

  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas odio tellus, tincidunt
sit amet dictum
  scelerisque, faucibus in ligula. Quisque commodo erat ex, vel viverra nisi consequat
efficitur</p>

  <h2>Introducción</h2>

  <p>Maecenas tincidunt erat vitae malesuada molestie. Duis ultrices dapibus bibendum. Etiam mi
metus, mollis
  vulputate consequat at, mattis vitae lectus. Etiam posuere eleifend metus at hendrerit.
Vestibulum ante ipsum
  primis in faucibus orci luctus et ultrices posuere cubilia curae; Nullam ullamcorper purus
sed lacinia semper.
  Maecenas accumsan auctor posuere. Etiam cursus libero vitae justo vulputate, dictum finibus
elit gravida</p>

  <h3>Historia</h3>

  <p>Curabitur vitae scelerisque ante. Suspendisse pretium sed sem a vulputate. Vestibulum
ullamcorper dolor vitae dui
  placerat varius. Aliquam pharetra est ante, sed accumsan lectus egestas non.</p>
</body>
</html>
```

CSS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas odio tellus, tincidunt sit amet dictum scelerisque, faucibus in ligula. Quisque commodo erat ex, vel viverra nisi consequat efficitur

Introducción

Maecenas tincidunt erat vitae malesuada molestie. Duis ultrices dapibus bibendum. Etiam mi metus, mollis vulputate consequat at, mattis vitae lectus. Etiam posuere eleifend metus at hendrerit. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Nullam ullamcorper purus sed lacinia semper. Maecenas accumsan auctor posuere. Etiam cursus libero vitae justo vulputate, dictum finibus elit gravida

Historia

Curabitur vitae scelerisque ante. Suspendisse pretium sed sem a vulputate. Vestibulum ullamcorper dolor vitae dui placerat varius. Aliquam pharetra est ante, sed accumsan lectus egestas non.

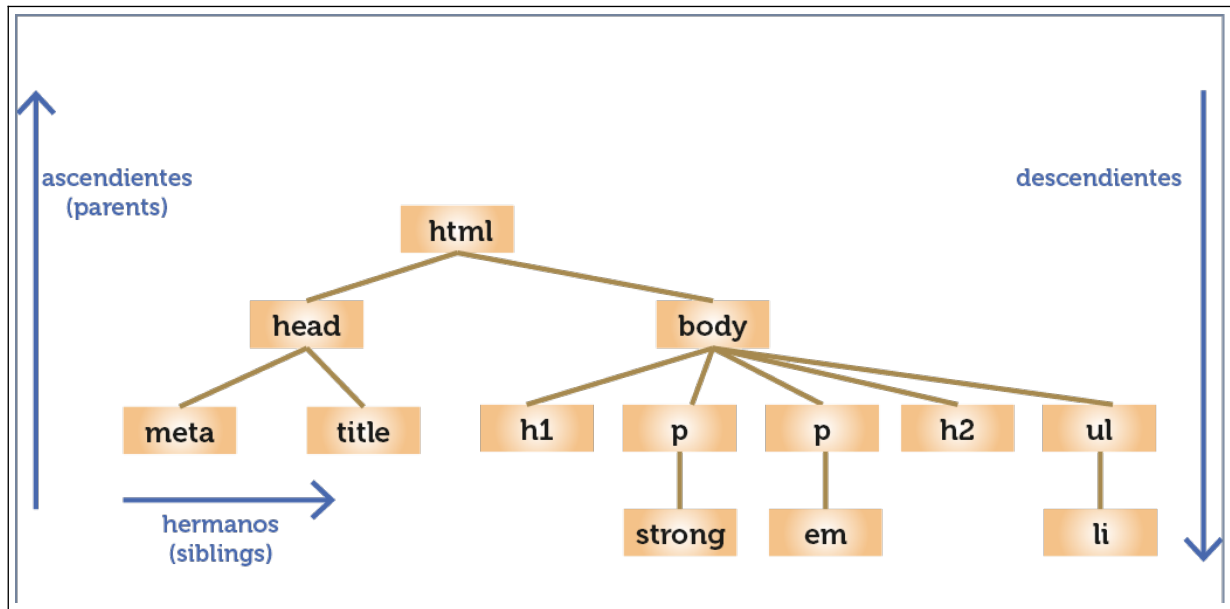
Selectores jerárquicos

Podríamos decir que un documento HTML esta formado de manera jerárquica, donde hay elementos que contienen otros elementos formando una estructura de árbol.

Esta es una parte importante del funcionamiento tanto de una página web HTML como de un documento XML.

De esa manera, los elementos del siguiente documento HTML podría representarse con la siguiente estructura de árbol:

```
<!doctype html>
<html lang="es-ES">
<head>
  <meta charset="UTF-8">
  <title>Monumentos de Palencia</title>
</head>
<body>
  <h1>Monumentos de Palencia</h1>
  <p>Palencia dispone de numerosos monumentos interesantes. Debido a su importante actividad medieval posee una gran cantidad de edificios religiosos entre los que destaca la <strong>Catedral de Palencia</strong> así como la iglesia de San Miguel con su peculiar torre de defensa y el monumento más conocido de la ciudad: El Cristo del Otero</p>
  <p>A finales del siglo <em>XIX</em> y principios del <em>XX</em> aparecieron edificios suntuosos y civiles que han embellecido una buena parte de la ciudad, en especial la transitada Calle Mayor.</p>
  <h2>Edificios religiosos</h2>
  <ul>
    <li>Cristo del Otero</li>
    <li>Catedral Mayor</li>
    <li>Iglesia de San Miguel</li>
    <li>Iglesia de San Lázaro</li>
    <li>Convento de San Pablo</li>
    <li>Iglesia de la Compañía</li>
  </ul>
</body>
</html>
```



Selector descendiente

Se usa cuando un elemento es descendiente de otro, es decir, se encuentra dentro de otro (entre las etiquetas de apertura y cierre). No hace falta que sea descendiente directo.

Sintaxis

```
Elemento descendiente{  
  propiedad:valor  
}
```

Ejemplo

```
<html>
  <head>
    <title>Selectores combinadores</title>
    <style>
      h3 i {
        color: coral;
        font-family: Verdana, Geneva, Tahoma, sans-serif;
        text-decoration: underline;
      }
    </style>
  </head>
  <body>
    <h1>CSS</h1>
    <h2>Cultivos</h2>
    <h3>Cultivo del <b>Litchi</b> en la zona de <i>Axarquía</i></h3>
    <h3>Cultivo de la <b>naranja</b> en la zona de <i>Valencia</i></h3>
    <h3>Cultivo del <b>la carambola</b> en la zona de <i>Málaga</i></h3>
    <h3>Cultivo del <b>plátano</b> en la zona de <i>Canarias</i></h3>
  </body>
</html>
```

Los elementos i que se encuentran dentro de un h3

Selector hijo

Similar a descendiente, pero sólo afecta a los hijos directos del elemento

Sintaxis

```
Elemento > Hijo
{
  propiedad: valor
}
```

Ejemplo

```
<html>
  <head>
    <title>Selectores combinadores</title>
    <style>
      h3 i{
        font-family: Verdana, Geneva, Tahoma, sans-serif;
        text-decoration: underline;
      }
      body>i
      {
        color:coral;
      }
    </style>
  </head>
  <body>
    <h1>CSS</h1>
    <h2>Cultivos</h2>
    <h3>Cultivo del <b>Litchi</b> en la zona de <i>Axarquia</i></h3>
    <h3>Cultivo de la <b>naranja</b> en la zona de <i>Valencia</i></h3>
    <h3>Cultivo del <b>la carambola</b> en la zona de <i>Málaga</i></h3>
    <h3>Cultivo del <b>plátano</b> en la zona de <i>Canarias</i></h3>
    <i>Muchos cultivos de todo</i>
  </body>
</html>
```

Afecta a los i hijos directos de body

No le afectará la propiedad porque i es hijo de <h3>, no de body

CSS

Cultivos

Cultivo del Litchi en la zona de **Axarquia**

Cultivo de la naranja en la zona de **Valencia**

Cultivo del la carambola en la zona de **Málaga**

Cultivo del plátano en la zona de **Canarias**

Muchos cultivos de todo

Selector de hermanos

Se utiliza para elementos que son hermanos, es decir, tienen el mismo padre

Sintaxis

```
Elemento1 ~ Elemento2{
propiedad:valor;
}
```

Ejemplo

```
<!DOCTYPE html>
<html>
  <head>
    <title>Selectores combinadores</title>
    <style>
      h2 ~ h3{
        color:coral;
      }
    </style>
  </head>
  <body>
    <h1>CSS</h1>
    <h2>Cultivos</h2>
    <h3>Cultivo del <b>Litchi</b> en la zona de <i>Axarquia</i></h3>
    <h3>Cultivo de la <b>naranja</b> en la zona de <i>Valencia</i></h3>
    <h3>Cultivo del <b>la carambola</b> en la zona de <i>Málaga</i></h3>
    <h3>Cultivo del <b>plátano</b> en la zona de <i>Canarias</i></h3>
    <i>Muchos cultivos de todo</i>
  </body>
</html>
```

El estilo se aplica a los h3 que sean hermanos de h2, es decir hijos de su padre, body

CSS

Cultivos

Cultivo del Litchi en la zona de *Axarquia*

Cultivo de la naranja en la zona de *Valencia*

Cultivo del la carambola en la zona de *Málaga*

Cultivo del plátano en la zona de *Canarias*

Muchos cultivos de todo

Selector adyacente

Se utiliza para elementos que son hermanos y uno está al lado del otro (tienen el mismo padre)

Sintaxis

Elemento1 + Elemento2

```
{
  propiedad:valor;
}
```

Ejemplo

```
<html>
  <head>
    <title>Selectores combinadores</title>
    <style>
      h2+h3{
        color:coral;
      }
    </style>
  </head>
  <body>
    <h1>CSS</h1>
    <h2>Cultivos</h2>
    <h3>Cultivo del <b>Litchi</b> en la zona de <i>Axarquía</i></h3>
    <h3>Cultivo de la <b>naranja</b> en la zona de <i>Valencia</i></h3>
    <h3>Cultivo del <b>la carambola</b> en la zona de <i>Málaga</i></h3>
    <h3>Cultivo del <b>plátano</b> en la zona de <i>Canarias</i></h3>
    <i>Muchos cultivos de todo</i>
  </body>
</html>
```

Afectará a los h3 que sean hermanos de h2 y estén a su lado.
H2 y H3 son hermanos porque ambos son hijos de body

CSS

Cultivos

Cultivo del Litchi en la zona de Axarquía

Cultivo de la naranja en la zona de Valencia

Cultivo del la carambola en la zona de Málaga

Cultivo del plátano en la zona de Canarias

Muchos cultivos de todo

No se aplica el estilo porque aunque es hermano de H2 no está a su lado

Pseudoclases

Una **pseudoclase** es una palabra clave que se añade a los selectores y que especifica un estado especial del elemento seleccionado.

Por ejemplo, `:hover` aplicará un estilo cuando el usuario haga *hover* sobre el elemento especificado por el selector.

Ejemplo

```
<html>
  <head>
    <title>Pseudoclases</title>
    <style>
      h1:hover{
        background-color: burlywood;
      }
    </style>
  </head>
  <body>
    <h1>CSS</h1>
    <h2>Cultivos</h2>
    <h3>Cultivo del <b>Litchi</b> en la zona de <i>Axarquia</i></h3>
    <h3>Cultivo de la <b>naranja</b> en la zona de <i>Valencia</i></h3>
    <h3>Cultivo del <b>la carambola</b> en la zona de <i>Málaga</i></h3>
    <h3>Cultivo del <b>plátano</b> en la zona de <i>Canarias</i></h3>
    <i>Muchos cultivos de todo</i>
  </body>
</html>
```

CSS

Cultivos

Cultivo del Litchi en la zona de *Axarquia*

Cultivo de la naranja en la zona de *Valencia*

Cultivo del la carambola en la zona de *Málaga*

Cultivo del plátano en la zona de *Canarias*

Muchos cultivos de todo

elemento: empty	Se aplica cuando el elemento está vacío
elemento: nth-child (número)	<p>Se aplica al elemento indicado cuando sea el hijo con el número indicado (por ejemplo número sería 3, para el tercer hijo).</p> <p>Se pueden utilizar expresiones más complejas mediante el uso de la variable <i>n</i> para conseguir fórmulas más complejas</p> <p>Se permite también usa las palabras clave odd (<i>impar</i>) y even(<i>par</i>)</p> <p>Ejemplos:</p> <ul style="list-style-type: none"> • tr:nth-child(3). Selecciona la tercera fila de una tabla • tr:nth-child(odd). Selecciona las filas impares • td:nth-child(even). Selecciona las celdas en las columnas pares • tr:nth-child(2n+1). Selecciona las filas 1,3,5,... (las impares) • tr:nth-child(3n+1). Selecciona las filas 1, 4, 7, 10, 13... (de tres en tres)
elemento: nth-last-child (n)	Funciona igual que el anterior pero cuenta los elementos hijos a partir del último (es decir, el orden de atrás hacia delante).
elemento: first-child	Se aplica al elemento cuando es el primer hijo
elemento: last-child	Se aplica al elemento cuando es el último hijo
elemento: only-child	Se aplica cuando el elemento es el único hijo
elemento: first-of-type	Primer descendiente de su tipo
elemento: last-of-type	Último descendiente de su hijo
elemento: nth-of-type (n)	<p>Funciona como nth-child, pero se refiere al número de hijo de ese tipo. No cuenta hijos, sino hijos del tipo indicado.</p> <p>Ejemplo:</p> <pre><style> article p:nth-of-type(2) { color:red; } </style></pre>

	<pre> <body> <article> <h1> Primer hijo, no se colorea </h1> <p> Segundo hijo, primero de tipo p. No se colorea </p> <p> Tercer hijo, segundo de tipo p. SE COLOREA </p> <p> Cuarto hijo, tercero de tipo p. No se colorea </p> </article> </body> </pre>
elemento: nth-last-of-type (n)	Como el anterior pero cuenta n desde el final.
elemento: only-of-type	Se aplica cuando el elemento es el único hijo de ese tipo

Otras pseudoclases:

:focus	Cuando el elemento obtiene el foco, es decir cuando el elemento ha capturado la entrada del teclado. Muy útil en formularios.
:enabled	Cuando el elemento está habilitado (útil en formularios)
:disabled	Cuando el elemento está deshabilitado (útil en formularios)
:checked	En controles de formulario de tipo radio o checkbox , cuando el elemento pasa a estar activado.
:target	<p>Se aplica cuando el elemento al que se refiere ha sido destino de un enlace.</p> <p>Ejemplo:</p> <pre> <style> #intro:target{ color:blue; } ... <h1 id="intro">Introducción</h1> ... Ir a la introducción </pre> <p>Cuando hagamos clic sobre el enlace ir a la introducción, el título Introducción se pondrá de color azul.</p>

:focus	Cuando el elemento obtiene el foco, es decir cuando el elemento ha capturado la entrada del teclado. Muy útil en formularios.
:valid	Controles de formulario que contienen valores válidos (lo serán o no dependiendo del tipo de elemento). Por ejemplo no será válido un texto en un cuadro input de tipo numérico (<i>number</i>).
:invalid	Inverso al anterior. Es decir, selecciona controles cuyos valores no sean válidos.
:required	Controles de formulario que están marcados para ser rellenados obligatoriamente (atributo required)
:optional	Contrario al anterior. Controles que no están obligados a ser rellenados.
:first-line	La primera línea del elemento.
:first-letter	La primera letra del elemento.
:before	<p>Contenido anterior al elemento. Siempre se suele usar con la propiedad content para añadir contenido al elemento. Ejemplo:</p> <pre>p:before{ content:url('flecha.gif'); }</pre> <p>Coloca la imagen de una flecha delante del párrafo</p>
:after	Para indicar contenido después del elemento.
:not(selector)	<p>Se aplica cuando el elemento no cumple ser del tipo indicado por el selector dentro de la palabra not. Ejemplo:</p> <pre>div >:not(p){ color:red; }</pre> <p>El color rojo se aplicará al contenido de los elementos div que no estén dentro de elementos p.</p>

Puedes encontrar más pseudoclases y su uso en

<https://lenguajecss.com/css/selectores/pseudoclases/>

Ejercicio resuelto

Asignar los estilos que se indican a la tabla del siguiente documento HTML:

- Encabezado y pie de la tabla con color de fondo marrón, color de texto blanco y convertir el texto a mayúsculas.
- Última columna con el texto en negrita
- A las filas pares del cuerpo de la tabla se les asigna color de fondo *cadetblue* y color de texto blanco
- A las filas impares del cuerpo de la tabla se les asigna color de fondo blanco y color de texto *cadetblue*

```
<html>
<head>
  <title>Ejercicio</title>
  <style>
    thead>tr,
    tfoot>tr {
      background-color: brown;
      color: white;
      text-transform: uppercase;
    }
    td:last-child
    {
      font-weight: bold;
    }
    tbody>tr:nth-child(odd)
    {
      background-color: cadetblue;
      color: white;
    }
    tbody>tr:nth-child(even)
    {
      background-color: white;
      color: cadetblue;
    }
  </style>
</head>

<body>
  <table>
    <thead>
      <tr>
```

```

        <td>Concepto</td>
        <td>Primer semestre</td>
        <td>Segundo semestre</td>
        <td>Total</td>
    </tr>
</thead>
<tbody>
    <tr>
        <td>Alimentación</td>
        <td>1500</td>
        <td>1750</td>
        <td>3200</td>
    </tr>
    <tr>
        <td>Tecnología</td>
        <td>1500</td>
        <td>1750</td>
        <td>3200</td>
    </tr>
    <tr>
        <td>Calzado</td>
        <td>1500</td>
        <td>1750</td>
        <td>3200</td>
    </tr>
    <tr>
        <td>Papelería</td>
        <td>1500</td>
        <td>1750</td>
        <td>3200</td>
    </tr>
</tbody>
<tfoot>
    <tr>
        <td>Totales</td>
        <td>6000</td>
        <td>7000</td>
        <td>12800</td>
    </tr>
</tfoot>
</table>
</body>
</html>

```

Pseudoelementos

Los pseudo-elementos se añaden a los selectores pero no para filtrar, sino que permiten añadir estilos a una parte concreta del documento.

Sintaxis

```
elemento::selector{  
propiedad:valor;  
}
```

::after	Se aplica para colocar código tras el elemento
::before	Se aplica para colocar código antes del elemento
::first-letter	Primera letra del elemento
::first-line	Primera línea del elemento
::selection	Se aplica al texto seleccionado del elemento. En el caso de los navegadores de Mozilla, requiere usar el prefijo <i>-moz-</i>

Ejemplo

```

<html>
<head>
  <title></title>
  <style>
    #quijote::after{
      display:block;
      content: "Seguir leyendo ...";
    }
    #quijote::first-letter{
      font-size: 3em;
      font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
    }
  </style>
</head>
<body>
  <p id="quijote">
    En un lugar de la Mancha de cuyo nombre no puedo acordarme no ha mucho tiempo que vivía un
    hidalgo
  </p>
</body>
</html>

```

Salta a la línea siguiente

Se añade el texto "Seguir leyendo" después del elemento con id="quijote"

Se aplican las siguientes propiedades a la primera letra del elemento con id="quijote"

En un lugar de la Mancha de cuyo nombre no puedo acordarme no ha mucho tiempo que vivía un hidalgo
Seguir leyendo ...