

Selectores en CSS



Combinador de hermanos adyacentes

El combinador `+` selecciona hermanos adyacentes. Esto quiere decir que el segundo elemento sigue directamente al primero y ambos comparten el mismo elemento padre (Ambos están al mismo nivel).

Sintaxis: `A + B`

Ejemplo: La regla `h2 + p` se aplicará a todos los elementos `<p>` que siguen directamente a un elemento `<h2>`.

Ejemplo:

```
h2 + p {  
  text-decoration: underline;  
  color:  white;  
  background-color:  coral;  
}
```

```
<div>  
  <h2>Título h2</h2>  
  <p>Texto P</p>  
  <p>Texto P</p>  
  <h2>Título h2</h2>  
</div>
```

¿Cuál es el
resultado?

Observación:

El último elemento de la suma es quien se le aplica las propiedades CSS.

Por lo tanto si tengo $A + B$, es B quien recibe las propiedades.

¿Por qué al segundo y último “p” no se aplican las propiedades CSS?

```
h2 + p {  
  text-decoration: underline;  
  color: white;  
  background-color: coral;  
}
```

```
<h2>Título h2</h2>  
<div>  
  <h2>Título h2</h2>  
  <p>Texto P</p>  
  <p>Texto P</p>  
  <h2>Título h2</h2>  
</div>  
<p>Texto P</p>
```



Respuesta:

Segundo “p”: No se aplican las propiedades porque antes de él debe venir una etiqueta “h2” y **viene una etiqueta “p”**.

Tercer “p”: No se aplican las propiedades porque el “h2” que está antes de él, **está en un nivel diferente**.

Ejercicio

Escriba una estructura HTML válida para aplicar estas propiedades

```
#miId + .color.texto > div + span {  
  color:  white;  
  width: 200px;  
  font-weight: bolder;  
  background-color:  teal;  
}
```

Solución

```
<div id="miId"></div>
<div class="color texto">
  <div></div>
  <span>Propiedades Aplicadas!</span>
</div>
```


Ejercicio

Escriba una estructura HTML válida para aplicar estas propiedades

```
div > span + span + #miId > span + h1.estilo.color
{
  color: ■ white;
  background-color: ■ rgb(21, 165, 134);
  border-radius: 20px;
  width: 200px;
  text-align: center;
}
```

Solución

```
<div>
  <span></span>
  <span></span>
  <div id="miId">
    <span></span>
    <h1 class="estilo color">Propiedades Aplicadas!</h1>
  </div>
</div>
```

**Propiedades
Aplicadas!**

Combinador general de hermanos


El combinador \sim selecciona hermanos. Esto quiere decir que el segundo elemento sigue al primero (no necesariamente de forma inmediata) y ambos comparten el mismo elemento padre (Deben estar al mismo nivel).

Sintaxis: $A \sim B$

Ejemplo: La regla $p \sim \text{span}$ se aplicará a todos los elementos `` que siguen un elemento `<p>`.

Observación: Si viene un `p` seguido de 3 `span`, todos los `span` se les aplicará las propiedades.

Ejemplo

```
p ~ span {  
  font-size: 32px;  
  color:  coral;  
}
```

¿Qué resultado
se obtendrá?

```
<div>  
  <p>Etiqueta P</p>  
  <div></div>  
  <span>Tengo un P hermano declarado antes de mi.</span>  
  <div></div>  
  <span>Yo también tengo un P hermano antes de mi!</span>  
</div>
```

Resultado:

Etiqueta P

Tengo un P hermano declarado antes de mi.
Yo también tengo un P hermano antes de mi!

Ejercicio

¿Qué resultado se obtendrá y por qué?

```
.textoEspecial ~ p {  
  font-style: italic;  
}
```

```
<div>  
  <div class="textoEspecial"></div>  
  <p>Esto es un texto en cursiva</p>  
  <div>  
    <p>Tendré cursiva?</p>  
    <div class="textoEspecial">  
      <p>Cursiva para mi?</p>  
    </div>  
  </div>  
  <p>Estoy muy lejos ... pero tendré cursiva?</p>  
</div>
```

Resultado:

Esto es un texto en cursiva

Tendré cursiva?

Cursiva para mi?

Estoy muy lejos ... pero tendré cursiva?

Consejo de día

Un gran poder conlleva una gran responsabilidad.

Bajo los nuevos combinadores es posible crear reglas muy complejas, pero a la vez comportamientos difíciles de entender.

Es mejor intentar mantener el equilibrio.

Antes de continuar, desarrolle la guía de ejercicios:


Ejercicios - CSS - Selectores –
Combinadores.

Password: **css-selectores**

Problema de día - Conflictos

¿Hay algún problema con los siguientes códigos?

¿Cuál es el problema en específico?

```
#miId.estilo {  
    background-color: yellowgreen;  
    width: 100px;  
    height: 100px;  
}  
  
.color.fondo {  
    background-color: violet;  
    width: 150px;  
    height: 150px;  
}
```

```
<div id="miId" class="estilo color fondo"></div>
```

Respuesta

Ambos bloques de códigos CSS desean interactuar con la etiqueta.

El problema se encuentra en que ambos quieren modificar las mismas propiedades.

Hay un conflicto de quien tiene prioridad sobre el otro.

Para este ejemplo el bloque `#mild.estilo` tendrá prioridad sobre `.color.fondo`



Reglas y Pesos

Cuando se cuenta con varias reglas CSS en cascada que afectan a un mismo elemento, el orden de prioridad que determina como se mostrará el elemento se indica bajo los siguiente tres criterios.

- Criterio 1 - **!important**: Un estilo marcado como importante prevalecerá sobre el resto de estilos. En caso de tener varios estilos marcados con **!important**, prevalecerá el de mayor peso según las reglas.
- Criterio 2 - Origen de las reglas: Las reglas del autor de la web prevalecerán sobre las reglas del lector de similar peso. Y tanto las reglas de autor como de lector prevalecerán sobre las del navegador.
- Criterio 3 - Peso de la regla: Una regla con mayor peso prevalecerá sobre otra de menor peso.



Peso = **A****B****C** (número de 3 cifras, cada una de las cuales se calcula contando los selectores de cada tipo según se indica a continuación)

A = Números de selectores de Id (selectores que acceden al atributo «id» del elemento mediante «#»)


B = Números de selectores de Clase (selectores que acceden al atributo “class” del elemento mediante “.”)

C = Números de selectores de elemento. (Nombre de etiquetas de HTML)

Ejemplo

```
#miId.estilo {  
    background-color:  yellowgreen;  
    width: 100px;  
    height: 100px;  
}  
  
.color.fondo {  
    background-color:  violet;  
    width: 150px;  
    height: 150px;  
}
```

Para calcular quien tiene prioridad sobre el otro, se deberán contar para cada bloque cuantas etiquetas tiene y su tipo, de acuerdo a eso se obtendrá el resultado final.

```
#miId.estilo {  
  background-color: yellowgreen;  
  width: 100px;  
  height: 100px;  
}
```

Tiene un selector de Id.


Tiene un selector de clase.

Tiene cero selectores de elemento.

Formula

$$\text{Peso} = (\text{Selectores de Id}) * 100 + (\text{Selectores de Clase}) * 10 + (\text{Selectores de elementos}) * 1$$

Resultado = 100 + 10 + 0 = **110**.


```
.color.fondo {  
  background-color: violet;  
  width: 150px;  
  height: 150px;  
}
```

Tiene cero selectores de Id.


Tiene dos clases.

Tiene cero selectores de elemento.


Formula

$$\text{Peso} = (\text{Selectores de Id}) * 100 + (\text{Selectores de Clase}) * 10 + (\text{Selectores de elementos}) * 1$$

Resultado = 0 + 20 + 0 = **20**.

```
#miId.estilo {  
  background-color:  yellowgreen;  
  width: 100px;  
  height: 100px;  
}
```

Peso = 110

```
.color.fondo {  
  background-color:  violet;  
  width: 150px;  
  height: 150px;  
}
```

Peso = 20

Por lo tanto el primer bloque tiene prioridad sobre el segundo

Observación

**Cuando dos reglas
tienen el mismo peso
prevalecerá la última
regla especificada.**

Ejercicios

Calcular el peso de las siguiente regla:

#Mild .MiClase p

Resultado

#Mild .MiClase p

- Tiene un selector de Id.
- Tiene un selector de clase.
- Tiene un selector de elemento.

$$\text{Peso} = 1*100 + 1*10 + 1*1 = 111.$$

Ejercicios

Calcular el peso de las siguiente regla:

`span #Mild h1`

Resultado

span #Mild h1

- Tiene un selector de Id.
- Tiene cero selectores de clase.
- Tiene dos selectores de elemento.

$$\text{Peso} = 1 * 100 + 0 * 10 + 2 * 1 = 102.$$

Ejercicios

Calcular el peso de las siguiente regla:

.MiClase_1 li.MiClase_2 a

Resultado

.MiClase_1 li.MiClase_2 a

- Tiene cero selectores de Id.
- Tiene dos selectores de clase.
- Tiene dos selectores de elemento.

$$\text{Peso} = 0 * 100 + 2 * 10 + 2 * 1 = 22.$$

Ejercicios

Calcular el peso de las siguiente regla:

$\text{div } h1 \ h2$

Resultado

div h1 h2

- Tiene cero selectores de Id.
- Tiene cero selectores de clase.
- Tiene tres selectores de elemento.

$$\text{Peso} = 0 * 100 + 0 * 10 + 3 * 1 = 3.$$

Propiedad marcada con !important

El primer bloque tiene un peso de 100 y el segundo bloque un peso de 10. Por lo tanto, el bloque uno tiene prioridad sobre el segundo bloque.

```
#miId {  
    background-color: yellowgreen;  
    width: 100px;  
    height: 100px;  
}  
  
.fondo {  
    background-color: violet !important;  
    width: 150px;  
    height: 150px;  
}
```

Exceptuando la propiedad de background-color de la clase “fondo” ya que está marcada como importante.


!important

Cuando una propiedad está con la palabra !important, superará a cualquier otra regla de peso.

Por lo tanto

```
#miId {  
    background-color: yellowgreen;  
    width: 100px;  
    height: 100px;  
}  
  
.fondo {  
    background-color: violet !important;  
    width: 150px;  
    height: 150px;  
}
```



Las propiedades que se terminarían aplicando a la etiqueta serian:

```
background-color: violet !important;  
width: 100px;  
height: 100px;
```

Observación:

En el caso que ambos bloques tuvieran !important en el mismo atributo, se tomaría como segundo criterio el peso.

En el caso que ambos tuvieran el mismo peso, el último declarado en el CSS es aplicado.

```
#miId {  
    background-color:  yellowgreen !important;  
}  
  
.fondo {  
    background-color:  violet !important;  
}
```

Ejercicios

Antes de continuar, desarrolle la guía de ejercicios:

Ejercicios - CSS - Selectores – Pesos

Password: **weight**

Selectores de Atributo

El selector de atributos CSS coincide con los elementos en función de la presencia o el valor de un atributo determinado.

Los atributos son valores adicionales que configuran los elementos (Etiquetas) o ajustan su comportamiento de diversas formas.

Ejemplos

```
<a href="https://www.google.com/">Esto es un Enlace!</a>
```

```
<div style="width: 20px;"></div>
```

```

```

```
<td colspan="2" rowspan="2"></td>
```

```
<section id="Noticias"></section>
```

```
<audio src="musica.mp3" autoplay loop></audio>
```

Un atributo puede o no contener un valor

Selector de Atributos

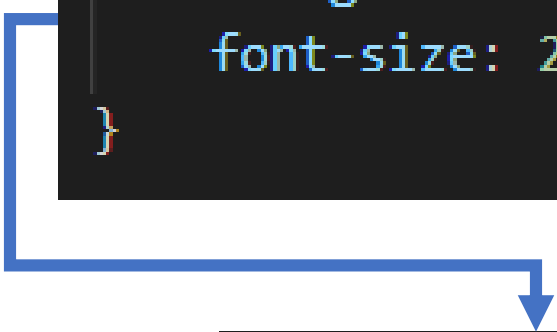
Primera forma de uso.

Estructura CSS –
Ejemplo:

```
[atributo]{  
    prop1: valor;  
    prop2:valor;  
    ...  
}
```

- a. Naranja
- b. Manzana
- c. Piña
- d. Pera
- e. Uva

```
[type] {  
    background-color: blue;  
    font-size: 20px;  
}
```



```
<ol type="a">  
    <li>Naranja</li>  
    <li>Manzana</li>  
    <li>Piña</li>  
    <li>Pera</li>  
    <li>Uva</li>  
</ol>
```

Ejemplo


Todas las etiquetas `<a>` que contengan el atributo `target`, deberán ser de color rojo.

```
<a href="#" target="_blank">Ejemplos</a>
<ul>
  <li>
    <a href="#" target="_blank">Google</a>
  </li>
  <li>
    <a href="#">Facebook</a>
  </li>
  <li>
    <a href="#" target="_blank">Instagram</a>
  </li>
  <li>
    <a href="#">Gmail</a>
  </li>
  <li>
    <a href="#" target="_blank">Dropbox</a>
  </li>
</ul>
```

Ejemplos

- Google
- Facebook
- Instagram
- Gmail
- Dropbox

Solución

```
[target] {  
  color:  red;  
}
```

Selector de atributos

Los atributos que se agregan a una etiqueta no necesariamente deben ser los establecidos por HTML. Es posible inventar y asignar cualquier nombre.

```
<div cajitaAtributo></div>  
<aside miHeader="Hola!" otroAtributo></aside>
```

Ejercicio

Establezca las siguientes propiedades CSS dependiendo de su atributo.

1. m1

1. Color: coral
2. Alto: 50px
3. Ancho: 50px

2. m2

1. Color: verde
2. Alto 50px
3. Ancho: 100px




3. m3

1. Color: violeta
2. Alto: 50px
3. Ancho: 150px

```
<section m1></section>
<div m2="Un Valor"></div>
<article m1></article>
<div m3></div>
<article m3="??"></article>
<footer m2></footer>
```



Solución

```
[m1] {  
  width: 50px;  
  height: 50px;  
  background-color:  coral;  
}  
  
[m2] {  
  width: 100px;  
  height: 50px;  
  background-color:  rgb(80, 255, 95);  
}  
  
[m3] {  
  width: 150px;  
  height: 50px;  
  background-color:  rgb(255, 80, 246);  
}
```



Selector de Atributo - Filtrado por elemento.

Hasta el momento se ha indicado como modificar cualquier etiqueta que contenga un atributo en específico, pero también es posible filtrar solamente por un tipo de etiqueta y su atributo.

Estructura CSS – Ejemplo:

```
nombreEtiqueta[atributo]{  
    prop1: valor;  
    prop2:valor;  
    ...  
}
```

¿A qué elementos se le aplicará las siguientes propiedades?

```
section[noticia] {  
  width: 100%;  
  height: auto;  
  background-color:  rgb(95, 95, 255);  
}
```

```
<section noticia>1</section>  
<header noticia>2</header>  
<section noticia>3</section>  
<footer noticia>4</footer>
```

Solución

1
2
3
4



Selector de Atributo - Contenido



Aparte de poder seleccionar todos los elementos por un atributo, también es posible filtrar para un mismo atributo por su contenido.

Estructura CSS – Ejemplo:

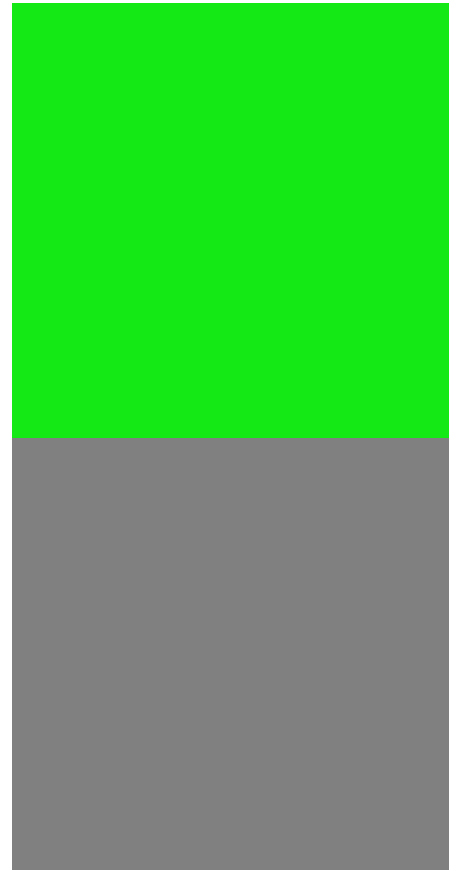
El valor va entre comillas dobles.

```
[atributo="valor"]{  
    prop1: valor;  
    prop2:valor;  
    ...  
}
```

Ejemplo

```
[estado="on"]{  
  height: 100px;  
  width: 100px;  
  background-color:  rgb(21, 233, 21);  
}  
  
[estado="off"]  
{  
  height: 100px;  
  width: 100px;  
  background-color:  gray;  
}
```

```
<div estado="on"></div>  
<div estado="off"></div>
```






Ejercicio

Agregue propiedades a las celdas bajo las siguientes condiciones:

1. Si el colspan contiene el valor 1, la celda es de color verde.
2. Si el colspan contiene el valor 2, la celda es de color amarillo.
3. Si el colspan contiene el valor 3, la celda es de color rojo.
4. El selector de atributo solamente debe funcionar para etiquetas <td>
5. Si un td no contiene el atributo colspan no se debe aplicar ninguna propiedad.

```
<table border="1">
  <tr>
    <td colspan="2">1</td>
    <td colspan="1">2</td>
    <td>3</td>
  </tr>
  <tr>
    <td>4</td>
    <td colspan="3">5</td>
  </tr>
  <tr>
    <td colspan="1">6</td>
    <td>7</td>
    <td colspan="2">8</td>
  </tr>
</table>
```

Solución

```
td[colspan="1"] {  
    background-color:  rgb(22, 214, 32);  
}  
  
td[colspan="2"] {  
    background-color:  rgb(255, 252, 47);  
}  
  
td[colspan="3"] {  
    background-color:  rgb(230, 62, 32);  
}
```

Observación

A diferencia de los otros selectores, el selector de atributo siempre debe ser declarado **al principio del bloque css.**

Ejemplo:

div[miAtributo="Hola!"] > p

pero no

p > **div[miAtributo="Hola!"]**

Selector de Atributos - Contenido

Al indicar por ejemplo

```
a[href="https://www.google.cl"] {...}
```

Se está diciendo que si una etiqueta <a> contiene el valor en un href de <https://www.google.cl> se aplicarán las propiedades CSS.

¿Y si necesito que dependiendo de como termine la URL (.com/.cl/.org/etc.) se deba aplicar un color diferente?

Selector de Atributos - Contenido

Formas posibles	
[attr=valor]	Selecciona los elementos cuyo atributo attr tenga exactamente el valor.
[attr~= valor]	Selecciona los elementos cuyo atributo attr tenga por valor una palabra de la lista separada por espacios.
[attr = valor]	Selecciona los elementos cuyo atributo attr tenga exactamente el valor o empiece por el valor seguido de un guión - (U+002D). Se puede usar para coincidencias de subcódigos en otros idiomas. Ej: es-cl, es-mx.

Selector de Atributos - Contenido

Formas posibles	
[attr^= valor]	Selecciona los elementos cuyo atributo attr inicie con el valor.
[attr\$= valor]	Selecciona los elementos cuyo atributo attr finalice con el valor.
[attr*= valor]	Selecciona los elementos cuyo atributo attr contenta en alguna parte el valor.

Selector de Atributos - Contenido

Bonus

[attr operator=valor i]

Agregar una i (o I) antes del corchete de cierre hace que el valor sea comparado sin distinguir entre mayúsculas y minúsculas (para caracteres dentro del rango ASCII).

Ejercicios

Descargar guía en la página del curso.

Nombre: **Ejercicios - CSS - Selector de Atributos**

Pass: **attr**