

# Selectors y unidades

## Hojas de Estilo en Cascada

# Selector

Un patrón que se usa al principio de un conjunto de reglas (ruleset) para elegir qué elementos serán afectados por las declaraciones. Existen varios tipos de selectores:

# Selector

- Selector de tipo: Selecciona todos los elementos de una etiqueta HTML específica. Por ejemplo, h1 selecciona todos los elementos <h1>.
- Selector de clase: Selecciona todos los elementos con un atributo de clase (class) específico. Los selectores de clase llevan un prefijo .; por ejemplo, .foo selecciona todos los elementos HTML con el atributo class="foo".
- Selector de ID: Selecciona el elemento con un atributo id específico. Los selectores de ID llevan un prefijo #; por ejemplo, #bar selecciona el elemento HTML con el atributo id="bar".

# Selector

- Selector de atributo: Selecciona todos los elementos con un atributo HTML específico establecido a un valor concreto. Se escriben entre [] y usan = como delimitador entre el nombre del atributo y su valor.

Por ejemplo, `[type="submit"]` selecciona todos los elementos con el atributo `type` igual a `"submit"`.

Si se omite el valor, se seleccionan todos los elementos que tengan ese atributo, sin importar su valor.

Por ejemplo, `[type]` selecciona todos los elementos que tengan el atributo `type`, sea cual sea su valor.

# Selector

Los selectores de atributo también pueden usar una sintaxis especial para hacer coincidencias (pattern matching) dentro del valor. Algunas de las opciones más comunes son:

`[href*="algoexpert.io"]` selecciona todos los elementos con un atributo href que contenga el texto "algoexpert.io" en cualquier parte.

`[href$="algoexpert.io"]` selecciona todos los elementos con un atributo href que termine con el texto "algoexpert.io".

`[href^="https://algoexpert.io"]` selecciona todos los elementos con un atributo href que empiece con el texto "https://algoexpert.io".

# Selector

- Combinador:

Combina múltiples selectores para seleccionar elementos basados en su ubicación dentro del DOM. Existen algunos tipos de combinadores:

- Combinador de descendiente: Representado por un espacio, con el formato

`selector1 selector2.`

Selecciona todos los elementos que coinciden con `selector2` y que son descendientes de un elemento que coincide con `selector1`.

(Un descendiente no necesita ser un hijo directo; por ejemplo, el abuelo de un elemento seleccionado podría coincidir con `selector1`).

# Selector

- Combinador de hijo: Representado por  $>$ , con el formato `selector1 > selector2`.

Selecciona todos los elementos que coinciden con `selector2` y que son hijos directos de un elemento que coincide con `selector1`.

- Combinador de hermano (sibling): Representado por  $\sim$ , con el formato `selector1 ~ selector2`.

Selecciona todos los elementos que coinciden con `selector2` y que son hermanos de un elemento que coincide con `selector1`.

El elemento que coincide con `selector2` debe aparecer después del que coincide con `selector1`.

# Selector

- Combinador de hermano adyacente: Representado por +, con el formato

`selector1 + selector2.`

Selecciona todos los elementos que coinciden con `selector2` y que tienen un elemento que coincide con `selector1` inmediatamente antes de ellos en el DOM.

Aprende más:

[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Selectors](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors)



# Selector

```
head
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Selectors</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>Top Level Heading</h1>

    <section>
      <h2>Section Heading</h2>
      <p>Section paragraph</p>
    </section>

    <footer>
      <h2>Footer Heading</h2>
      <p>Footer paragraph</p>
    </footer>
  </body>
</html>
```

## Top Level Heading

### Section Heading

Section paragraph

### Footer Heading

Footer paragraph

# Selector

```
h1 {  
  color: red;  
}
```

**Top Level Heading**

**Section Heading**

Section paragraph

**Footer Heading**

Footer paragraph

# Selector

```
h1, h2 {  
  color: red;  
}
```

**Top Level Heading**

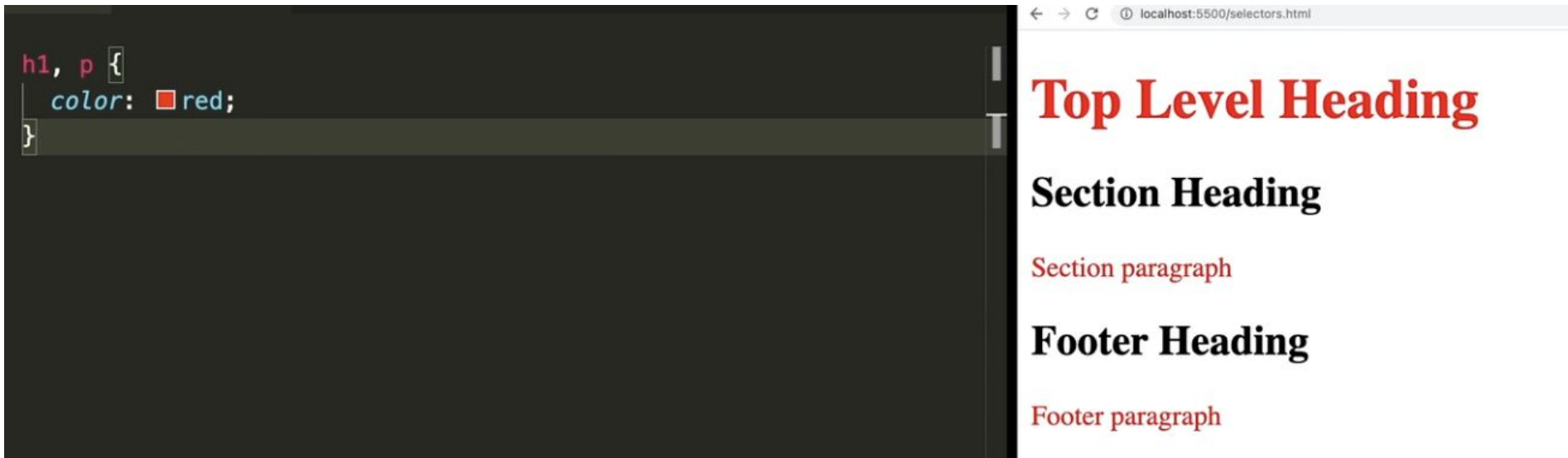
**Section Heading**

Section paragraph

**Footer Heading**

Footer paragraph

# Selector



# Selector

```
✓ * {  
  color: red;  
}
```

**Top Level Heading**

**Section Heading**

Section paragraph

**Footer Heading**

Footer paragraph

# Selector

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Selectors</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1 class="red">Top Level Heading</h1>

    <section>
      <h2>Section Heading</h2>
      <p class="red">Section paragraph</p>
    </section>
```

# Selector

```
.red {  
  color: red;  
}
```

**Top Level Heading**

**Section Heading**

Section paragraph

**Footer Heading**

Footer paragraph

# Selector

```
.red, h2 {  
  color: red;  
}
```

**Top Level Heading**

**Section Heading**

Section paragraph

**Footer Heading**

Footer paragraph



# Selector

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Selectors</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>Top Level Heading</h1>

    <section>
      <h2>Section Heading</h2>
      <p id="red">Section paragraph</p>
    </section>
```

# Selector

```
#red {  
  color: red;  
}
```

**Top Level Heading**

**Section Heading**

Section paragraph

**Footer Heading**

Footer paragraph

# Selector

```
h1 {  
  color: blue;  
}  
  
#red {  
  color: red;  
}
```

**Top Level Heading**

**Section Heading**

Section paragraph

**Footer Heading**

Footer paragraph

# Selector

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Selectors</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>Top Level Heading</h1>

    <section>
      <h2>Section Heading</h2>
      <p id="red">Section paragraph</p>
      <a href="https://systemsexpert.io">SystemsExpert</a>
      <a href="https://algoexpert.io">AlgoExpert</a>
    </section>
```

# Selector

```
h1 {  
  color: blue;  
}  
  
#red {  
  color: red;  
}  
  
a {  
  color: green;  
}
```

**Top Level Heading**

**Section Heading**

Section paragraph

SystemsExpert AlgoExpert

**Footer Heading**

Footer paragraph

# Selector

```
h1 {  
  color: blue;  
}
```

```
#red {  
  color: red;  
}
```

```
a[href="https://algoexpert.io"] {  
  color: green;  
}
```

**Top Level Heading**

**Section Heading**

Section paragraph

[SystemsExpert](#) [AlgoExpert](#)

**Footer Heading**

Footer paragraph

# Selector

```
<section>
  <h2>Section Heading</h2>
  <p id="red">Section paragraph</p>
  <a href="https://systemsexpert.io">SystemsExpert</a>
  <a href="https://algoexpert.io">AlgoExpert</a>
  <a href="https://algoexpert.io/product">AlgoExpert</a>
</section>
```

# Selector

```
h1 {  
  color: blue;  
}
```

```
#red {  
  color: red;  
}
```

```
a[href^="https://algoexpert.io"] {  
  color: green;  
}
```

## Top Level Heading

### Section Heading

Section paragraph

[SystemsExpert](#) [AlgoExpert](#) [AlgoExpert](#)

### Footer Heading

Footer paragraph



# Selector

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Selectors</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>Top Level Heading</h1>

    <section>
      <h2>Section Heading</h2>
      <p id="red">Section paragraph</p>
      <a href="https://systemsexpert.io">SystemsExpert</a>
      <a href="https://algoexpert.io">AlgoExpert</a>
      <a href="https://algoexpert.io/product">AlgoExpert</a>
    </section>

    <footer>
      <h2>Footer Heading</h2>
      <p>Footer paragraph</p>
      <div>
        <p>Div paragraph</p>
      </div>
    </footer>
  </body>
</html>
```

# Selector

```
footer p {  
  color: red;  
}  
  
a[href^="https://algoexpert.io"] {  
  color: green;  
}
```

## Top Level Heading

### Section Heading

Section paragraph

[SystemsExpert](#) [AlgoExpert](#) [AlgoExpert](#)

### Footer Heading

Footer paragraph

Div paragraph

# Selector

```
footer > p {  
  color: red;  
}  
  
a[href^="https://algoexpert.io"] {  
  color: green;  
}
```

**Top Level Heading**

**Section Heading**

Section paragraph

[SystemsExpert](#) [AlgoExpert](#) [AlgoExpert](#)

**Footer Heading**

Footer paragraph

Div paragraph

# Selector

```
h2 ~ a {  
  color: red;  
}
```

**Top Level Heading**

**Section Heading**

Section paragraph

SystemsExpert AlgoExpert AlgoExpert

**Footer Heading**

Footer paragraph

Div paragraph

# Selector

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Selectors</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>Top Level Heading</h1>

    <section>
      <h2>Section Heading</h2>
      <!-- <p id="red">Section paragraph</p> -->
      <a href="https://systemsexpert.io">SystemsExpert</a>
      <a href="https://algoexpert.io">AlgoExpert</a>
      <a href="https://algoexpert.io/product">AlgoExpert</a>
    </section>
```

# Selector

```
h2 + a {  
  color: red;  
}
```

## Top Level Heading

### Section Heading

SystemsExpert AlgoExpert AlgoExpert

### Footer Heading

Footer paragraph

Div paragraph

# Pseudo Clase

Una adición a un selector de CSS para seleccionar en función del estado actual del elemento.

Estas comienzan con :, por ejemplo:

`button:hover` selecciona los botones sobre los que el cursor está actualmente posado.

Aprende más:

<https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes>

# Pseudo Elemento

Una adición a un selector de CSS para seleccionar una porción específica de un elemento.

Estas comienzan con ::, por ejemplo:

`p::first-letter` selecciona la primera letra de los párrafos.

`::before` y `::after` son pseudo-elementos especiales que insertan contenido antes o después del contenido del elemento, permitiendo aplicar estilos antes o después de dicho contenido.

Esto suele usarse con la propiedad de CSS `content`, aunque no siempre.

Aprende más:

<https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements>



# Pseudo Elemento

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Pseudo Classes And Elements</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <h1>Hello World</h1>
    <p>
      Go to
      <a href="https://algoexpert.io">AlgoExpert</a>
    </p>

    <p>AlgoExpert is the best!</p>
    <p>FrontendExpert is the best!</p>

    <label for="input">Input: </label>
    <input id="input" type="text" />
  </body>
</html>
```

## Hello World

Go to [AlgoExpert](https://algoexpert.io)

AlgoExpert is the best!

FrontendExpert is the best!

Input:

# Links a los que no se han dado click

```
a:link {  
  color:  red;  
}
```

## Hello World

Go to [AlgoExpert](#)

AlgoExpert is the best!

FrontendExpert is the best!

Input:

# Links a los que se han dado click

```
a:link {  
  color: ■ red;  
}  
  
a:visited {  
  color: ■ green;  
}
```

## Hello World

Go to [AlgoExpert](#)

AlgoExpert is the best!

FrontendExpert is the best!

Input:

# Otros ejemplos

```
input:focus {  
  outline: 2px solid blue;  
}  
  
a:link {  
  color: red;  
}  
  
a:visited {  
  color: green;  
}
```

## Hello World

Go to [AlgoExpert](#)

AlgoExpert is the best!

FrontendExpert is the best!

Input:

# Otros ejemplos

```
<label for="input">Input: </label>  
<input id="input" type="text" minlength="6" />
```

```
input:invalid {  
  outline: 2px solid red;  
}
```

## Hello World


Go to [AlgoExpert](#)

AlgoExpert is the best!

FrontendExpert is the best!

Input:

# Otros ejemplos

```
input:invalid:not(:focus) {  
  outline: 2px solid  red;  
}
```

Input:

# Otros ejemplos

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Pseudo Classes And Elements</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <h1>Hello World</h1>
    <p>
      Go to
      <a href="https://algoexpert.io">AlgoExpert</a>
    </p>

    <div>
      <p>AlgoExpert is the best!</p>
      <p>FrontendExpert is the best!</p>
    </div>

    <label for="input">Input: </label>
    <input id="input" type="text" minlength="6" />
  </body>
</html>
```

# Otros ejemplos

```
p:first-of-type {  
  color:  orange;  
}
```

## Hello World

Go to [AlgoExpert](#)

AlgoExpert is the best!

FrontendExpert is the best!

Input:

```
p:last-of-type {  
  color:  orange;  
}
```

## Hello World

Go to [AlgoExpert](#)

AlgoExpert is the best!

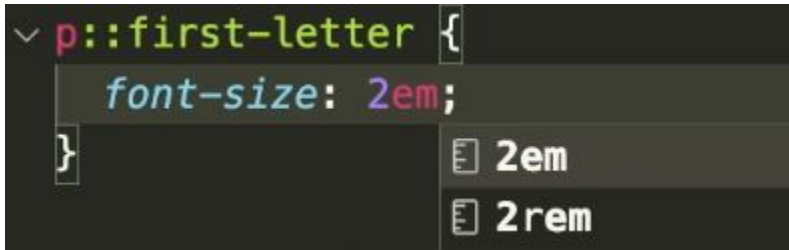
FrontendExpert is the best!

Input:



# Otros ejemplos

```
▼ p::first-letter {  
  font-size: 2em;  
}
```

A screenshot of a code editor showing a CSS rule for the first letter of a paragraph. The rule is 'p::first-letter' with a font-size of '2em'. To the right of the code, there are two small icons, each followed by '2em' and '2rem' respectively, indicating the font size in different units.

**H**ello World

**G**o to [AlgoExpert](#)

**A**lgoExpert is the best!

**F**rontendExpert is the best!

Input:

# Otros ejemplos

```
p::before {  
  content: ">";  
}
```

```
p::after {  
  content: "!";  
}
```

## Hello World

> Go to [AlgoExpert](#) !

> AlgoExpert is the best!!

> FrontendExpert is the best!!

Input:

# Selector Specificity

Aunque los humanos desafortunadamente suelen recurrir a la violencia como medio de resolver conflictos, CSS prefiere tomar un enfoque más matemático para resolver disputas.

En CSS, cuando dos o más reglas “compiten” por aplicarse al mismo elemento, no se resuelve con fuerza, sino con un proceso lógico llamado cascada y especificidad.

# Selector Specificity

```
a[href="https://algoexpert.io"] {  
  color: green;  
}
```

```
section.links a {  
  color: red;  
}
```

# Cómo CSS resuelve “conflictos”

## 1. Cascada:

- El navegador sigue un orden jerárquico para decidir qué estilo aplicar:
- Estilos del navegador (por defecto).
- Estilos definidos por el usuario.
- Estilos definidos por el autor (los que escribimos en CSS).
- Reglas con !important, que tienen la máxima prioridad.

# Cómo CSS resuelve “conflictos”

Evitar important cuando sea posible

```
1  a {  
2    color: orange !important;  
3  }
```

# Cómo CSS resuelve “conflictos”

## 2. Especificidad:

- Si varias reglas afectan al mismo elemento, se aplica la más “específica”:
- Selectores de tipo (ejemplo: `div`) tienen menor peso.
- Selectores de clase (`.miClase`) pesan más.
- Selectores de ID (`#miID`) pesan aún más.
- Estilos en línea (`style="..."`) tienen el mayor peso (excepto `!important`).

# Cómo CSS resuelve “conflictos”

## 3. Orden de aparición:

Si dos reglas tienen la misma especificidad, gana la que aparece más tarde en el archivo CSS.



# Cómo CSS resuelve “conflictos”

## Calculating Specificity

Inline styles: 1000pts

IDs: 100pts

Classes: 10pts

Pseudo classes: 10pts

Attributes: 10pts

Elements: 1pt

Pseudo elements: 1pt

$$1 + 10 = 11$$

```
1 a[href="https://algoexpert.io"] {  
2   color: green;  
3 }  
4  
5 1 + 10 + 1 = 12  
6 section.links a {  
7   color: red;  
}
```

# Actividad - Explorando el DOM y su relación con CSS

Explica con tus propias palabras:

¿Qué es el DOM?

¿En qué se diferencia del archivo HTML que escribimos?

¿Por qué decimos que el DOM es un “árbol de nodos”?

¿Cómo se relaciona el DOM con CSS y JavaScript?

Incluye un diagrama sencillo del árbol DOM de una página básica.

# Actividad - Explorando el DOM y su relación con CSS

Exploración práctica en el navegador

Abre una página web (ejemplo: wikipedia.org).

Da clic derecho → Inspeccionar → pestaña "Elements".

Localiza:

- La etiqueta `<html>`
- La etiqueta `<head>`
- La etiqueta `<body>`

Busca un elemento sencillo, como un `<h1>` o un `<p>`.

Modifica el estilo desde el inspector:

Ejemplo: cambia su color (`color: red;`) o su tamaño (`font-size: 30px;`).

# Actividad - Explorando el DOM y su relación con CSS

Crea un archivo HTML sencillo con un `<div>` y varios `<p>`.

Ábrelo en el navegador e inspecciónalo.

Describe cómo aparece en el DOM y si hay diferencias con el código escrito (ejemplo: nodos que el navegador agrega automáticamente).

# Actividad - Explorando el DOM y su relación con CSS

Elabora un pequeño reporte con:

Resumen de tu investigación.

Capturas de pantalla del DOM en el inspector.

Ejemplo de un cambio de estilo antes y después.

Conclusión: ¿Por qué es importante entender el DOM al usar CSS?