

CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

Use our [CSS Selector Tester](#) to demonstrate the different selectors.

CSS Simple Selectors

The simple selectors select elements based on element-name, id, and class. In addition, there is the universal selector (*).

Selector	Example	Example description
element	p	Selects all <p> elements
#id	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
.class	.intro p.intro	Selects all elements with class="intro" Selects all <p> elements with class="intro"

CSS Attribute Selectors

The attribute selector selects HTML elements with a given attribute set.

Selector	Example	Example description
[attribute]	[lang]	Selects all elements with a lang attribute
[attribute=value]	[lang="it"]	Selects all elements with lang="it"
[attribute~=value]	[title~="flower"]	Selects all elements with a title attribute containing the word "flower"
[attribute =value]	[lang "en"]	Selects all elements with a lang attribute value equal to "en" or starting with "en-"
[attribute^=value]	[href^="https"]	Selects all elements with a href attribute value that begins with "https"
[attribute\$=value]	[href\$=".pdf"]	Selects all elements with a href attribute value ends with ".pdf"
[attribute*=value]	[href*="w3schools"]	Selects all elements with a href attribute value containing the substring "w3schools"

CSS Nesting Selector

Selector	Example	Example description
&	&	Applies styles for an element within the context of another element

Comparison Table

Feature	Flexbox	CSS Grid
Dimension	One-dimensional (row or column)	Two-dimensional (rows and columns)
Use Case	Simple, linear layouts	Complex grid-based layouts
Alignment	Aligns items along main and cross axes	Aligns items in both dimensions
Content vs. Layout	Focuses on content flow	Focuses on overall layout structure
Item Placement	Depends on content order	Can position items explicitly
Best For	Smaller components or dynamic items	Full-page or grid-like layouts

When to Use Each

- Use **Flexbox** when you need to align and distribute space among items in a single axis (e.g., a row of buttons or navigation).
- Use **CSS Grid** for more complex layouts where both rows and columns need control, like building an entire webpage layout.