

# CSS Selectors

## Basic Selectors

Basic selectors are the most frequently used and provide direct ways to target elements.

- **Universal Selector (\*)**  
Targets all elements on the page.
- `* { margin: 0; padding: 0; }`

*Use case:* Resetting default browser styles.

- **Element Selector (p, h1, etc.)**  
Targets all elements of a specific type.
- `p { font-size: 16px; }`

*Use case:* Applying consistent formatting to all paragraphs.

- **Class Selector (.classname)**  
Targets elements with a specific class attribute.
- `.button { background-color: blue; color: white; }`

*Use case:* Reusable styling across multiple elements.

- **ID Selector (#idname)**  
Targets a single, unique element.
- `#header { background-color: gray; }`

*Use case:* Styling one specific section, such as a navigation bar.

---

## 2. Combinator Selectors

Combinators define relationships between elements, allowing more precise targeting.

- **Descendant Selector (A B)**  
Targets elements nested within another.
- `div p { color: red; }`

# CSS Selectors

*Meaning:* All `<p>` elements inside a `<div>`.

- **Child Selector (`A > B`)**  
Targets only direct children.
- `div > p { margin-left: 20px; }`

*Meaning:* Paragraphs that are immediate children of a `<div>`.

- **Adjacent Sibling Selector (`A + B`)**  
Targets the element immediately following another.
- `h1 + p { font-weight: bold; }`

*Meaning:* The first `<p>` after an `<h1>`.

- **General Sibling Selector (`A ~ B`)**  
Targets all siblings that follow a specified element.
- `h1 ~ p { font-style: italic; }`

*Meaning:* All `<p>` elements that come after an `<h1>`.

---

## 3. Group Selectors

Group selectors allow multiple elements to share the same style rule.

```
h1, h2, p {  
  font-family: Arial, sans-serif;  
}
```

*Use case:* Ensuring consistent typography across headings and paragraphs.

---

## 4. Attribute Selectors

Attribute selectors target elements based on the presence or value of attributes.

- **Presence Selector (`[attr]`)**
- `input[type] { border: 2px solid black; }`

# CSS Selectors

*Meaning:* All inputs with a type attribute.

- **Exact Value Selector ([attr="value"])**
- `input[type="text"] { background-color: yellow; }`

*Meaning:* Only text input fields.

- **Starts With Selector (^=)**
- `a[href^="https"] { color: green; }`

*Meaning:* Links beginning with “https”.

- **Contains Selector (\*=)**
- `a[href*="example"] { text-decoration: underline; }`

*Meaning:* Links containing the word “example”.

---

## 5. Pseudo-Classes

Pseudo-classes define styles for elements in specific states or positions.

- **:hover** – Applies when the user hovers over an element.
- `a:hover { color: red; }`
- **:focus** – Applies when an element is active (e.g., input field).
- `input:focus { outline: 3px solid blue; }`
- **:first-child** – Targets the first child of a parent.
- `p:first-child { color: brown; }`
- **:last-child** – Targets the last child of a parent.
- `p:last-child { color: green; }`
- **:not()** – Excludes elements from selection.
- `p:not(.highlight) { color: gray; }`

---

## 6. Pseudo-Elements

Pseudo-elements allow styling of specific parts of an element’s content.

- **::before** – Inserts content before an element.

## CSS Selectors

- `h1::before { content: "★ "; }`
  - **::after** – Inserts content after an element.
  - `h1::after { content: " ✓"; }`
  - **::first-line** – Styles the first line of text.
  - `p::first-line { font-weight: bold; }`
  - **::first-letter** – Styles the first letter.
  - `p::first-letter { font-size: 24px; color: red; }`
  - **::placeholder** – Styles placeholder text in inputs.
  - `input::placeholder { font-style: italic; }`
- 

### 7. CSS Comments

Comments are used to annotate code for clarity. They are ignored by browsers.

```
/* This is a single-line comment */
```

```
/*  
This is a  
multi-line comment  
*/
```

*Best practice:* Use comments to explain complex rules or sections of your stylesheet.

---

### Summary

CSS selectors are the foundation of web styling. By mastering basic selectors, combinators, group selectors, attribute selectors, pseudo-classes, and pseudo-elements, developers gain precise control over the presentation of web content. Proper use of selectors not only enhances design but also improves maintainability and readability of code.