# 1.1 element

**DESCRIPTION:**

The element selector targets all instances of a specific HTML tag. For example, writing p { color: blue; } applies the style to every tag on the page.

It’s the most basic and widely used selector, perfect for setting default styles for headings, paragraphs, or sections without needing extra classes or IDs

**PROGRAM:**

<html>

<head>

<style>

/\* Element selector: targets all <h2> elements \*/

h2 { color: teal; font-family: Arial, sans-serif;

}

</style>

</head>

<body>

<h2>This is styled using an element selector.</h2>

<h2>Another heading with the same style.</h2>

</body>

</html>

# 1.2 id

**DESCRIPTION:**

An ID selector targets a single, unique element using the # symbol followed by the element’s ID. For instance, #header { background-color: black; } applies styles only to the element with id="header". Since IDs must be unique within a page, this selector is ideal for styling one-off components like navigation bars or footers.

**PROGRAM:**

<!DOCTYPE html>

<html>

<head>

<style>

/\* ID selector: targets only the element with id="special" \*/

#special { color: purple; font-size: 20px; text-decoration: underline;

}

</style>

</head>

<body>

<p>This is a normal paragraph.</p>

<p id="special">This paragraph is styled using an ID selector.</p>

</body>

</html>

**1.3 class**

**DESCRIPTION:**

Class selectors use a . followed by the class name to style multiple elements that share the same class. For example, .highlight { color: orange; } will affect all elements with class="highlight". This is especially useful for applying consistent styles across different tags, like buttons, cards, or alerts.

**PROGRAM:**

<!DOCTYPE html>

<html>

<head>

<style> .center { text-align: center; color: red;

}

</style>

</head>

<body>

<h1 class="center">Red and center-aligned heading</h1>

<p class="center">Red and center-aligned paragraph.</p>

</body>

</html>

**1.4 group**

**DESCRIPTION:**

Group selectors allow you to apply the same style to multiple elements by separating them with commas. For example, h1, h2, p { margin-bottom: 10px; } ensures that all these elements have the same spacing. It’s a clean way to reduce repetition and keep your CSS concise when styling similar elements.

PROGRAM:

<html>

<head> <style> h1,h2,p{ text\_align:center; colour:red;

}

</style>

</head>

<body>

<h1>hello world!</h1>

<h2>smaller heading!</h2>

<p>this is a paragraph</p>

</body>

</html>

**1.5 universal**

**DESCRIPTION:**

The universal selector uses an asterisk \* to target every element on the page. For example, \* { box-sizing: border-box; } ensures consistent box sizing across all elements. While powerful, it should be used cautiously to avoid unintended overrides, especially in large or complex layouts.

**PROGRAM:**

<html>

<head>

<style> \*{ text-align:center; color:blue;

}

</style>

</head>

<body>

<h1>Hello World</h1>

<h2>Smaller heading</h2>

<p>Every element on the page will be affected by the style</p>

<p>And me!</p>

</body>

</html>

**2 Combinator selector**

**DESCRIPTION:**

Combinator selectors define relationships between elements in the DOM tree, allowing you to style elements based on their structural context.

* Descendant Selector (A B)

This targets all elements B that are nested anywhere inside element A, regardless of depth. For example, div p styles every inside a even if it’s several levels deep. It’s useful for applying styles broadly within a container, like styling all text inside a card or section.

* Child Selector (A > B)

This targets only the direct children of element A. For instance, ul > li applies styles only to elements that are immediate children of a , ignoring nested lists. This is ideal when you want to maintain strict control over layout hierarchy.

* Adjacent Sibling Selector (A + B)

This selects element B that comes immediately after element A, sharing the same parent. For example, h2 + p targets the first that follows an It’s great for styling introductory paragraphs or spacing elements that follow headers.

* General Sibling Selector (A ~ B)

This selects all siblings B that follow element A, not just the first one. For example, h2 ~ p styles every that comes after an within the same parent. It’s useful for applying consistent styles to grouped elements that follow a trigger or label.

**PROGRAM**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width,initial-scale=1.0">

<title>CSS combinator selectors</title>

<style>

/\* Descendant selector(div p) \*/ div p{ colour:blue;

}

/\* child selector (div > p) \*/ div > p{ background-color:lightyellow;

}

/\* Adjacent sibling selector (h2+p) \*/ h2+p{ color:red; font-weight:bold;

}

/\* General sibling selector (h2-p) \*/ h2~p{

font-style:italic;

}

/\* styling for visibility \*/ div, selection { border:1px solid #ccc; padding:10px; margin:10px;

}

</style>

</head>

<body bgcolor=”lavender”>

<h1>css combinator selectors example</h1>

<div>

<p>child of div(affected by both descendant & child selectors)</p>

<section>

<p>Descendant of div (only affected by descendant selector)</p>

</section>

</div>

<h2>Heading(h2)</h2>

<p>Adjacent sibling (immediately after h2,styled red & bold)</p>

<p>general sibling (come after h2,styled italic)</p>

<p>Another general sibling (also italic)</p>

</body>

</html>

**3 Pseudo-class selector**

**DESCRIPTION:**

Pseudo-classes target elements based on their state, position, or user interaction—without needing extra classes or IDs.

* Structural Pseudo-classes like :first-child, :last-child, and :nth-child(n) allow you to style elements based on their position in the DOM. For example, li:first-child styles only the first list item, while li:nth-child(odd) targets every odd-numbered item.
* User Interaction Pseudo-classes like :hover, :focus, :active, and :visited respond to how users interact with elements. For instance, a:hover changes the link color when hovered, and input:focus highlights a form field when selected.
* Form State Pseudo-classes like :checked, :disabled, and :required let you style form elements based on their state. For example, input:checked can be used to highlight selected checkboxes or radio buttons.

**PROGAM:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Pseudo class selectors example</title>

<style>

/\* links \*/ a:link{ color:blue;

}

a:visited{ color:purple;

}

a.hover{ color:red;

}

a:active{ color:green;

}

/\* Input focus \*/ input:focus{ border:2px solid orange; background-color:#f9f2d0;

}

/\* list styling \*/ ul li:first-child {

color:crimson;

} ul li:last-child { color:teal;

}

ul li:nth-child(2) {

color:darkorange;

}

/\* checkbox \*/ input:checed+label color:dargreen; ront-weigt:bold;

}

</style>

</head>

<body>

<h2>pseudo-class selector demo</h2>

<p><a href="#">this is a link</a></p>

<p><input type="text" placeholder="click here to see:focus effect"></p>

<ul>

<li>first item</li>

<li>second item</li>

<li>third item</li>

</ul>

<p>

<input type="checkbox" id="check1">

<label for="check1">i agree</label>

</p>

</body>

</html>

**4 Pseudo-element selector**

**DESCRIPTION:**

Pseudo-elements allow you to style specific parts of an element or insert virtual content before or after it.

* ::before and ::after are used to inject content into the DOM visually, without altering the HTML. For example, button::before { content: "✓ "; } adds a checkmark before every button label.
* ::first-letter and ::first-line target the first letter or line of a block of text, useful for stylized typography like drop caps or editorial layouts.
* ::selection styles the portion of text a user highlights, allowing for custom highlight colors and effects.

These selectors are powerful for enhancing user experience and adding creative flair to your designs.

**PROGRAM:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>pseudo-element selectors example</title>

<style>

/\* ::first-letter - style the first letter of the paragraph \*/ p::first-letter{

font-size :200% font-weight: bold; colour: darkred;

}

/\* ::first-line -style the first line of a paragraph \*/ p:: first -line{ font style::italic;

colour:green;

}

/\* ::before - insert content before an element \*/ h2::before { content :" star";

colour:orange;

}

/\* :: after-insert content after an element\*/ h2::after{ content:"star"; colour orange;

}

/\* ::selection-style selected text\*/

::selection{ background-colour:yellow;

color:black;

}

</style>

</head>

<body>

<h1>pseudo-element selectors demo</h1>

<h2>css magic</h2>

<p>this paragraph demonstrates the use of pseudo-elements.the first letter is styled big and red, the first is italic , and selecting text will highlight it in yellow.</p>

</body>

</html>

**5 Attribute selector**

**DESCRIPTION:**

Attribute selectors target elements based on the presence or value of their attributes, offering precision without needing extra classes.

* [type="text"] targets input fields of type text.
* [href^="https"] matches links that start with "https".
* [src$=".jpg"] matches images ending in ".jpg".
* [data-role="admin"] targets custom data attributes used in JavaScript frameworks or semantic markup.

**PROGRAM:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Attribute Selectors Demo</title>

<style>

/\* Targets all input elements with type="text" \*/ input[type="text"] { border: 2px solid #4CAF50; padding: 5px;

}

/\* Targets links that open in a new tab \*/ a[target="\_blank"] { color: crimson; text-decoration: underline;

}

/\* Targets images with alt attribute \*/

img[alt] { border: 2px dashed #2196F3;

}

/\* Targets elements with a data-role attribute \*/ div[data-role="admin"] {

background-color: #f9f9f9; border-left: 5px solid orange; padding: 10px;

}

/\* Partial match: href contains "example" \*/ a[href\*="example"] { font-weight: bold;

}

/\* Starts with: class begins with "btn-" \*/ button[class^="btn-"] { padding: 10px 20px; border-radius: 5px;

}

/\* Ends with: src ends with ".jpg" \*/ img[src$=".jpg"] { box-shadow: 0 0 5px #888;

}

</style>

</head>

<body>

<input type="text" placeholder="Enter your name">

<a href="https://example.com" target="\_blank">Visit Example</a>

<img src="photo.jpg" alt="Sample Image">

<div data-role="admin">Admin Panel</div>

<button class="btn-primary">Click Me</button>

</body>

</html>