

# CSS Selectors

**Unit-III**

**SCS181 Web Technologies-I**

# CSS Selectors

- CSS selectors are patterns used to select and style specific HTML elements.
- They enable applying styles to elements based on their name, id, class, attributes, or relationships with other elements.
- In CSS, selectors are used to target the HTML elements on our web pages that we want to style.
- There are a wide variety of CSS selectors available, allowing for fine-grained precision when selecting elements to style.

# What is a selector?

- A CSS selector is the first part of a CSS Rule.
- It is a pattern of elements and other terms that tell the browser which HTML elements should be selected to have the CSS property values inside the rule applied to them.
- The element or elements which are selected by the selector are referred to as the *subject of the selector*.

```
body{  
width:100%;  
height: auto;  
}
```

# Type selectors

- A **type selector** is sometimes referred to as a *tag name selector* or *element selector* because it selects an HTML tag/element in your document.
  - Type selectors are not case-sensitive.
  - In the example below, we have used the span, em and strong selectors.
- `<h1>Type selectors</h1>`
  - `<p> Veggies es bonus vobis, proinde vos postulo  
essum magis <span>kohlrabi welsh onion</span>  
daikon amaranth tatsoi tomatillo melon azuki bean  
garlic. </p> <p> Gumbo beet greens corn soko  
<strong>endive</strong> gumbo gourd. Parsley  
shallot courgette tatsoi pea sprouts fava bean collard  
greens dandelion okra wakame tomato. Dandelion  
cucumber earthnut pea peanut soko zucchini. </p>  
<p> Turnip greens yarrow ricebean rutabaga  
<em>endive cauliflower</em> sea lettuce kohlrabi  
amaranth water spinach avocado daikon napa  
cabbage asparagus winter purslane kale. Celery  
potato scallion desert raisin horseradish spinach </p>`

# Example

```
body { font-family: sans-serif; } span { background-color: aqua; }
```

```
strong { color: rebeccapurple; } em { color: rebeccapurple; }
```

```
<h1>Type selectors</h1>
```

```
<p> CSS is an essential tool for <span> web developers</span>, offering powerful capabilities to style and layout web content.</p>
```

```
<p>By separating the presentation layer from the content, it ensures <em>clean, maintainable, and scalable websites</em>. With CSS, web designers can create visually appealing, responsive, and accessible websites that offer a consistent user experience across various devices and platforms.</p>
```

```
<p>As the web evolves, CSS continues to <strong>advance with new features and methodologies</strong>, making it a critical skill for anyone involved in web development.</p>
```

# Universal Selector (\*)

The universal selector is indicated by an asterisk (\*). It selects everything in the document. If \* is chained using a descendant combinator, it selects everything inside that ancestor element.

## Example:

```
body { font-family: sans-serif; }
*{
margin: 0;
padding: 0;
box-sizing:
border-box;
}
```

This applies the specified styles to all HTML elements.

# CSS ID Selector

The **ID** selector is used to target a single, unique element on a webpage. It is denoted by the # symbol followed by the ID name.

## a. How to Define an ID

IDs are defined in HTML using the id attribute, and the corresponding CSS rule is defined with the # symbol.

Each ID should be unique within an HTML document.

# Example

```
<html><head><style>#header {  
    background-color: lightblue;  
    color: white;  
    text-align: center;  
} </style></head><body>  
<div id="header">  
    This is a header.  
</div></body></html>
```



## b. Characteristics of the ID Selector

- **Uniqueness:** IDs should be unique within a page. No two elements should share the same ID.
- **Specificity:** IDs have high specificity, meaning that styles applied using an ID will override those applied using classes or element selectors.

## 2. CSS Class Selector

The **class** selector is used to target multiple elements that share a common class. It is denoted by the . symbol followed by the class name.

### a. How to Define a Class

Classes are defined in HTML using the class attribute, and the corresponding CSS rule is written with the . symbol.

Multiple elements can share the same class, making it useful for applying styles to groups of elements.

# Example

```
<html><head><style>
```

```
.box { border: 2px solid black; padding: 10px; background-color:  
lightgray; }
```

```
</style></head><body>
```

```
<div class="box"> This is a box. </div> <div class="box"> This is  
another box. </div>
```

```
</body></html>
```

# Differences Between ID and Class Selectors

Feature	ID Selector	Class Selector
<b>Symbol</b>	# (e.g., #header)	. (e.g., .box)
<b>Uniqueness</b>	Must be unique for each element on a page	Can be applied to multiple elements
<b>Specificity</b>	Higher specificity (overrides class styles)	Lower specificity
<b>Use case</b>	Best for styling a single, unique element	Best for styling groups of similar elements

# Using IDs and Classes Together

It's common to combine IDs and classes when you need to apply both specific and shared styles to elements.

## Example:

```
<div id="main-content" class="content-box"> This is the main  
content area. </div>
```

## CSS:

```
#main-content { background-color: yellow; /* Unique styles for this  
ID */ } .content-box { border: 1px solid black; padding: 20px; /*  
Common styles for all elements with the content-box class */ }
```

# Chaining Selectors

You can chain ID and class selectors together to apply styles to more specific elements.

## Example:

```
<div id="footer" class="content-box"> This is the footer. </div>
```

## CSS:

```
#footer.content-box { background-color: blue; /* Specific to  
elements with both #footer ID and content-box class */ }
```

# Selecting Child Elements within IDs and Classes

You can also target child elements inside elements with specific IDs or classes.

## Example:

```
<div id="header">  
<h1>Title</h1> <p>Subtitle</p>  
</div>
```

## CSS:

```
#header h1 { color: red; /* Only the h1 inside the #header element will be  
red */ }  
#header p { color: gray; /* Only the p inside the #header element will be  
gray */ }
```

# CSS Specificity Hierarchy

CSS follows a hierarchy of specificity, which determines which styles are applied when multiple rules target the same element. Here's a general order of specificity from lowest to highest:

**1.Type selectors (element selectors):** e.g., p, h1

**2.Class selectors:** e.g., .box

**3.ID selectors:** e.g., #header

**4.Inline styles:** e.g., style="color: red;"



# Overriding Styles with IDs and Classes

Due to the higher specificity of ID selectors, they will override class selectors if both are applied to the same element.

## Example:

```
<div id="content" class="box"> Content area </div>
```

## CSS:

```
.box { color: green; } #content { color: blue; }
```

# Group Selector (,)

Combines multiple selectors to apply the same styles.

## Example:

```
h1, h2, h3 { color: navy; font-family: Arial, sans-serif; }
```

This applies the styles to all <h1>, <h2>, and <h3> elements.

# Descendant Selector (Space)

Selects elements nested within another element.

## Example:

```
div p { color: green; }
```

This applies the style to `<p>` elements that are inside `<div>` elements.

# Child Selector (>)

Selects elements that are direct children of a parent.

## Example:

```
ul > li { list-style-type: square; }
```

This applies the style only to <li> elements that are direct children of <ul>.

# Adjacent Sibling Selector (+)

Selects the next sibling immediately following an element.

## Example:

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

This styles a `<p>` that immediately follows an `<h1>`.

# General Sibling Selector (~)

Selects all siblings after a specified element.

## Example:

```
h1 ~ p { font-style: italic; }
```

This styles all <p> elements that follow an <h1>.

# Attribute Selector

Targets elements based on attributes and their values.

## Examples:

**[attribute]:** Matches elements with the attribute.

```
input[type] { border: 1px solid black; }
```

**[attribute=value]:** Matches elements with a specific attribute value.

```
input[type="text"] { background-color: lightyellow; }
```

**[attribute^=value]:** Matches elements with attribute values starting with a value.

```
a[href^="https"] { color: green; }
```

# Pseudo-classes

Define the special state of an element.

## Examples:

**:hover:** Style an element when hovered over.

```
button:hover { background-color: lightblue; }
```

**:nth-child(n):** Style the nth child of a parent element.

```
tr:nth-child(even) { background-color: #f2f2f2; }
```



# Pseudo-elements

Target specific parts of an element.

## Examples:

**::before:** Insert content before an element.

```
h1::before { content: "★ "; color: gold; }
```

**::after:** Insert content after an element.

```
h1::after { content: " ★"; color: gold; }
```