**Assignment - 1 (CSS Selector and Styling**)

# **Question 1:** What is a CSS selector? Provide examples of element, class, and ID selectors.

* A **CSS selector** is a pattern used to select and style specific HTML elements on a webpage. It tells the browser which HTML elements the associated CSS rules should apply to.

**Types of CSS Selectors with Examples:**

1. **Element Selector**

* Definition: Targets all HTML elements of a specific type.
* Example:

p {

color: blue;

}

* Explanation: This rule sets the text color of all <p> (paragraph) elements to blue.

1. **Class Selector**

* Definition: Targets all elements with a specific class attribute.
* Syntax: Use a period . followed by the class name.
* Example:
  + .highlight {

background-color: yellow;

}

* Explanation: This rule applies a yellow background to any element with class="highlight". (<div class="highlight">Important info</div>)

1. **ID Selector**

* Definition: Targets a single element with a specific id attribute.
* Syntax**:** Use a hash # followed by the ID name.
* Example:
  + #main-title {

font-size: 24px;

}

* Explanation: This rule sets the font size of the element with id="main-title".( <h1 id="main-title">Welcome</h1>)

1. **Universal Selector**

* Definition: Target all element on the page.
* Syntax: \*
* Example:
  + \* {

margin: 0;

padding: 0;

}

* Explanation: Removes margin and padding from every element on the page.

1. **Group Selector**

* Definition: Target multiple elements, classes, ids at once.
* Syntax: separate selectors with commas ,.
* Example:
  + h1, h2, p {

font-family: Arial, sans-serif;

}

* Explanation: Applies the same font to all <h1>, <h2>, and <p> elements.

# **Question 2: Explain the concept of CSS specificity. How do conflicts between multiple styles get resolved?**

* **What is CSS Specificity?**

CSS specificity **is a set of rules that the browser uses to determine** which CSS rule **should be applied to an element** when multiple rules target the same element**.**

**It's like a** priority system **— more specific rules override less specific ones.**

**Specificity Hierarchy (Highest to Lowest)**

|  |  |
| --- | --- |
| **Selector Type** | **Specificity Value** |
| Inline styles | 1000 |
| ID selectors | 0100 |
| Class, pseudo-class, attribute selector | 0010 |
| Element & pseudo elements | 0001 |
| Universal Selector | 0000 |

**How Specificity is Calculated**

Specificity is usually represented as a 4-part number:  
a, b, c, d where:

* a = Inline styles
* b = Number of ID selectors
* c = Number of class selectors, attributes, and pseudo-classes
* d = Number of element selectors and pseudo-elements

Example:

**HTML**

<h1 id="title" class="heading">Hello</h1>

**CSS**

h1 {

color: blue; /\* specificity: (0, 0, 0, 1) \*/

}

.heading {

color: green; /\* specificity: (0, 0, 1, 0) \*/

}

#title {

color: red; /\* specificity: (0, 1, 0, 0) \*/

}

<style>

<h1 style="color: orange;">Hello</h1> <!-- specificity: (1, 0, 0, 0) -->

</style>

**Result:** The heading will be **orange** because **inline styles** have the highest specificity.

* **If Specificity is the Same:**

If two rules have **equal specificity**, the **last one declared in the CSS** will be applied.

* **Specificity Comparison from Weakest to Strongest**

 \* (universal selector) – Lowest

 element (e.g., h1)

 .class, :hover, [type="text"]

 #id

 Inline styles – Highest (e.g., style="")

 !important (overrides **everything**, but should be avoided unless necessary)

* **How Conflicts Are Resolved**
* **Compare specificity values** → higher wins.
* If **same specificity**, use the **last declared** rule.
* If one rule uses **!important**, it **overrides all others**, unless another also uses !important with higher specificity.

# **Question 3: What is the difference between internal, external, and inline CSS? Discuss the advantages and disadvantages of each approach.**

* **Difference Between Internal, External, and Inline CSS**

CSS can be applied to HTML in **three main ways**, each with its own use case, advantages, and disadvantages.

1. **Inline CSS**

* **Definition:** Styles are added directly to an HTML element using the style attribute.
* **Example:**
* <p style="color: red; font-size: 18px;">Hello</p>

**Advantages:**

* Very specific and immediate.
* Useful for quick fixes or testing.

**Disadvantages:**

* Poor maintainability.
* Repetitive for multiple elements.
* Doesn't separate content from style.
* Has the highest specificity, making overrides difficult.

1. **Internal CSS**

* **Definition:** CSS is written inside a <style> tag within the <head> section of an HTML file.
* **Example:**

<head>

<style>

p {

color: blue;

}

</style>

</head>

**Advantages:**

* Good for single-page styling.
* Easier to manage than inline CSS.
* Keeps CSS and HTML in one file (useful for small projects).

**Disadvantages:**

* Not reusable across multiple pages.
* Increases page size.
* Harder to maintain in large websites.

1. **External CSS**

* **Definition:** CSS is stored in a separate .css file and linked to the HTML with the <link> tag.
* **Example:**

<head>

<link rel="stylesheet" href="style.css">

</head>

**Advantages:**

* Best for large or multi-page websites.
* Keeps content and style completely separate.
* Reusable and easy to maintain.
* Enables faster page loading after the first request (cached by browser).

**Disadvantages:**

* Requires an extra HTTP request (can slow down initial load if not optimized).
* Won’t work if the external file fails to load.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Where Defined** | **Reusability** | **Best For** | **Specificity Level** |
| Inline | Inside HTML Element | No | Quick, One-time fix | High |
| Internal | <style> in <head> | No | Single-page site | Medium |
| External | Separate .css file | Yes | Multi-page Website | Medium |