**SONIYA KAMBLE – Assignment 3**

1. **Explain the difference between id selector and class selector.**

In CSS, both id selectors and class selectors are used to apply styles to HTML elements, but they have different purposes and characteristics:

**ID Selector**

* **Syntax:** An ID selector is prefixed with a hash (#) symbol.
* **Uniqueness:** Each ID should be unique within a page. This means that only one element should have a particular ID.
* **Usage:** Typically used for elements that require a unique style or script interaction.

#### ***Example:***

<!DOCTYPE html>

<html>

<head>

<style>

#header {

color: blue;

font-size: 24px;

}

</style>

</head>

<body>

<h1 id="header">Welcome to My Website</h1>

</body>

</html>

**Class Selector**

* **Syntax:** A class selector is prefixed with a dot (.) symbol.
* **Reusability:** Multiple elements can share the same class. Classes are used to apply the same style to multiple elements.
* **Usage:** Typically used for styling groups of elements that share common characteristics or for applying styles that are reused across multiple elements.

#### ***Example:***

<!DOCTYPE html>

<html>

<head>

<style>

.highlight {

color: red;

font-weight: bold;

}

</style>

</head>

<body>

<p class="highlight">This text will be highlighted.</p>

<span class="highlight">This span will also be highlighted.</span>

</body>

</html>

* **Key Differences**
* ID Selector: Should be unique per page. (#header)
* Class Selector: Can be reused for multiple elements. (.highlight)

1. **When you would select id selector,class selector and tag selector. Explain with example.**

Selecting the appropriate CSS selector—id, class, or tag—depends on the level of specificity and the nature of the styling you need.

**ID Selector (#id)**

#### ***When to Use:***

* When you need to apply unique styling to a single element.
* When you need to uniquely identify an element for JavaScript interaction.
* When the style should only be applied to one specific element on the page.

#### ***Example:***

<!DOCTYPE html>

<html>

<head>

<style>

#unique-header {

background-color: lightblue;

text-align: center;

padding: 20px;

}

</style>

</head>

<body>

<h1 id="unique-header">Unique Header</h1>

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

</body>

</html>

**Class Selector (.class)**

#### ***When to Use:***

* When you need to apply the same styles to multiple elements.
* When you want to group elements with common styling.
* When you want reusable styles that can be applied to different parts of the page.

#### ***Example:***

<!DOCTYPE html>

<html>

<head>

<style>

.highlight {

background-color: yellow;

font-weight: bold;

}

</style>

</head>

<body>

<p class="highlight">Highlighted text here.</p>

<div class="highlight">This div is also highlighted.</div>

<p>This text is not highlighted.</p>

</body>

</html>

**Tag Selector (tag)**

#### ***When to Use:***

* When you need to apply styling to all elements of a specific type.
* When you want to apply default styling across the entire page or site.
* When the styling should be consistent for all elements of that type.

#### ***Example:***

<!DOCTYPE html>

<html>

<head>

<style>

p {

font-size: 16px;

color: darkgray;

}

</style>

</head>

<body>

<p>This is a paragraph with default styling.</p>

<p>All paragraphs on the page will have this styling.</p>

</body>

</html>

#### ***Summary***

* **ID Selector (#id):** Use for unique elements requiring specific styling. High specificity.
* **Class Selector (.class):** Use for groups of elements with shared styling. Moderate specificity.
* **Tag Selector (tag):** Use for all elements of a specific type. Low specificity.

1. **What is difference between inline css and external css?**

**Inline CSS**

#### ***Definition:***

Inline CSS involves applying styles directly within HTML elements using the style attribute.

#### ***Example:***

<p style="color: blue; font-size: 16px;">This is a styled paragraph.</p>

**External CSS**

#### ***Definition:***

External CSS involves placing CSS rules in a separate .css file and linking that file to the HTML document using the <link> tag.

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="styles.css">

</head>

<body>

<p class="styled-paragraph">This is a styled paragraph.</p>

</body>

</html>

**CSS (styles.css):**

.styled-paragraph {

color: blue;

font-size: 16px;

}

1. **Explain the types of Combinators with example.**

There are four main types of combinators: descendant, child, adjacent sibling, and general sibling combinators. They are used to define relationships between selectors and apply styles based on the relationships between HTML elements.

**1. Descendant Combinator (Space)**

#### ***Definition:***

The descendant combinator is represented by a space ( ) between two selectors. It selects elements that are descendants of a specified element.

#### ***Syntax:***

#### ***css***

ancestor descendant {  
 /\* styles \*/  
}

#### ***Example:***

<!DOCTYPE html>

<html>

<head>

<style>

/\* Selects all <p> elements that are inside a <div> \*/

div p {

color: green;

}

</style>

</head>

<body>

<div>

<p>This paragraph is styled green.</p>

<span>

<p>This paragraph is also styled green.</p>

</span>

</div>

<p>This paragraph is not styled green because it's not inside a <div>.</p>

</body>

</html>

**2. Child Combinator (>)**

#### ***Definition:***

The child combinator selects elements that are direct children of a specified element. It is represented by a > symbol.

#### ***Syntax:***

#### ***css***

parent > child {  
 /\* styles \*/  
}

#### ***Example:***

<!DOCTYPE html>

<html>

<head>

<style>

/\* Selects only direct <p> children of <div> \*/

div > p {

color: red;

}

</style>

</head>

<body>

<div>

<p>This paragraph is styled red.</p>

<span>

<p>This paragraph is not styled red because it is not a direct child of <div>.</p>

</span>

</div>

<p>This paragraph is not styled red because it's not inside a <div>.</p>

</body>

</html>

**3. Adjacent Sibling Combinator (+)**

#### ***Definition:***

The adjacent sibling combinator selects an element that is immediately preceded by a specified element. It is represented by a + symbol.

#### ***Syntax:***

#### ***css***

preceding + sibling {  
 /\* styles \*/  
}

#### ***Example:***

<!DOCTYPE html>

<html>

<head>

<style>

/\* Selects the first <p> element immediately following a <h2> \*/

h2 + p {

font-style: italic;

}

</style>

</head>

<body>

<h2>Heading 2</h2>

<p>This paragraph is italicized because it is immediately after the <h2>.</p>

<p>This paragraph is not italicized because it is not immediately following the <h2>.</p>

</body>

</html>

**4. General Sibling Combinator (~)**

***Definition:***

The general sibling combinator selects all elements that are siblings of a specified element. It is represented by a ~ symbol and selects elements that come after the specified element.

***Syntax:***

#### ***css***

preceding ~ siblings {  
 /\* styles \*/  
}

<!DOCTYPE html>

<html>

<head>

<style>

/\* Selects all <p> elements that follow an <h2> \*/

h2 ~ p {

color: purple;

}

</style>

</head>

<body>

<h2>Heading 2</h2>

<p>This paragraph is purple because it follows the <h2>.</p>

<p>This paragraph is also purple because it follows the <h2>.</p>

<span>

<p>This paragraph is not purple because it is not a sibling of <h2>.</p>

</span>

</body>

</html>

**2. Look at the code given, identify the combinator used in it.**

**Q1 -** General Sibling Combinator

Q2 - Adjacent Sibling Combinator

Q3 - Child Combinator