CSS Selectors & Styling :-

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

Ans :- A **CSS selector** is a pattern used to select and style specific elements in an HTML document. It defines which elements the styles will apply to, allowing developers to control the appearance of content on a web page.

Types of selector :-

1. Simple CSS selector.
    I. Id selector
    II. Element selector
    III. Class selector.
2. Combinator Selector.
3. Pseudo element Selector.
4. Universal Selector.
5. Pusedo Class Selector.
6. Attributes Selector.
7. Grouping Selector.

1.Element Selector

Targets all elements of a specific type in the document.

```
p {
   color: blue;
}
```

2.Class Selector

Targets elements with a specific class attribute. Classes are reusable and can be applied to multiple elements.

```
.highlight {
   background-color: yellow;
}
```

3. **ID Selector**

   Targets a single element with a unique id attribute. IDs should be unique within a page.

```
#header {
   font-size: 24px;
   font-weight: bold;
}
```

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

Ans:- CSS specificity is a set of rules that determine which CSS rule applies to an element when multiple rules match the same element. The browser calculates specificity to resolve conflicts between styles and applies the most specific rule.

Understanding Specificity

1. **Specificity Weight** CSS assigns a weight to each type of selector based on its specificity. Specificity is calculated as a combination of numbers in the format (a, b, c, d):

   - **a**: Inline styles (e.g., style="color: red;") have the highest specificity.
   - **b**: ID selectors (e.g., #header) are next in priority.

- **c**: Class selectors, attributes, and pseudo-classes (e.g., .highlight, [type="text"], :hover).
- **d**: Element selectors and pseudo-elements (e.g., h1, p, ::before).

2. **Conflict Resolution**

- The rule with the highest specificity wins.
- If two rules have the same specificity, the one that appears **later** in the stylesheet (or in the source order) takes precedence.

Eg:-

Specificity: (0, 0, 0, 1) :-

```
p {
   color: blue;
}
```

Specificity: (0, 0, 1, 0) :-

```
.highlight {
   color: green;
}
```

Specificity: (0, 1, 0, 0) :-

```
#header {
   color: red;
}
```

Specificity: (1, 0, 0, 0) :-

<style>

color: black;

</style>

For the element:

Sample Text

- **Inline style** (style="color: black;") has the highest specificity (1, 0, 0, 0) and will apply.
- If the inline style were removed, the **ID selector** #header (0, 1, 0, 0) would apply.
- If both inline styles and ID selectors were removed, the **class selector** .highlight (0, 0, 1, 0) would apply.
- The **element selector** p (0, 0, 0, 1) has the lowest specificity.

Tips to Resolve Conflicts

1. **Use specificity wisely**: Avoid overusing !important, which forces a rule to apply regardless of specificity.
2. **Organize stylesheets**: Write rules in a logical order, from least specific (general styles) to most specific (exceptions or overrides).
3. **Refactor redundant rules**: Ensure clarity by avoiding conflicting rules in the first place.

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

Ans: CSS can be applied to HTML documents using **internal**, **external**, or **inline** styles. Each method has unique features, advantages, and disadvantages.

1. Inline CSS

CSS rules are written directly within the style attribute of an HTML element.

Eg :- This is a paragraph.

Advantages:

- Easy and quick to apply styles to specific elements.
- Useful for testing or applying one-off styles.

Disadvantages:

- Difficult to maintain, especially in large projects, as styles are scattered throughout the HTML.
- Increases file size and reduces readability.
- Inline styles have the highest specificity, which can make overriding them tricky.

2. Internal CSS

CSS rules are written within a <style> tag inside the <head> of an HTML document.

Eg :- <!DOCTYPE html>

<html>

<head>

<style>

p {

color: green;

}

```
    </style>
</head>
<body>
    <p>This is a paragraph.</p>
</body>
</html>
```

Advantages:

- Styles are consolidated in one place for the specific page, making them easier to manage compared to inline CSS.
- Does not require an additional file, which can be convenient for small projects or single-page applications.

Disadvantages:

- Styles are not reusable across multiple pages, leading to redundancy in multi-page projects.
- Larger <style> sections can clutter the HTML file.

3. External CSS

CSS rules are stored in a separate .css file and linked to the HTML document using a <link> tag.

Eg :- <link rel="stylesheet" href="styles.css">

Style.css

```
p {
    color: red;
```

}

Advantages:

- Promotes reusability, as one CSS file can style multiple HTML pages.
- Keeps the HTML code clean and easier to read.
- Makes it easier to update styles across multiple pages by editing a single file.

Disadvantages:

- Requires an additional HTTP request to load the CSS file, which may slightly impact initial page load time (though modern techniques like caching mitigate this).
- External stylesheets may fail to load if the file path is incorrect or the file is missing.