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 target and apply styles to specific HTML elements. It tells the browser *which elements* the CSS rules should be applied to.

The main types with examples:

1. Element Selector :-

   Targets all elements of a specific type for example p tag.

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

2. Class Selector :-

   Targets elements with a specific class attribute.Uses a dot (.) before the class name.

   EX. :

```
<button class="button">Click Me</button>

.button {
  background-color: green;
  color: white;
}
```

3. Id Selector :-

   Targets a single unique element with a specific id attribute.Uses a hash # before the id name.

   EX. :

```
<h1 id="header">Welcome</h1>
```

```
#header {
  font-size: 24px;
  text-align: center;
}
```

Question 2: Explain the concept of CSS specificity. How do conflicts between multiple stylesheet resolve?

ANS. :

Specificity is a set of rules that browsers use to decide which CSS style is applied when multiple selectors target the same element.In other words, if two or more CSS rules could apply to the same element, the browser uses specificity to decide which rule wins.

Specificity :

1. Type selectors (element names) → Example: <p> , <h1> ,
2. Class selectors, attributes, and pseudo-classes :- Example: .box , [type=”text”] , :hover
3. ID selectors :- Example: #main
4. Inline styles :- Example: <p style=”color:red”>

Conflict Between Multiple Stylesheets :-

1. Origin of Stylesheets (Priority Order) :

   The browser checks where the styles come from :

   - User agent styles browser defaults → lowest priority
   - External / internal styles your CSS files or <style> → normal priority
   - Inline styles style= “ ” inside the HTML → higher priority

2. Specificity Check :

   Specificity check among the competing rules, the one with higher specificity is applied as explained earlier.

3. Cascade Order / Last Rule Wins :

   If two rules have the same specificity and come from the same level of origin, the browser applies the one that appears last in the code.

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

Ans :

Difference Between

Inline css	Internal css	External css
It is used within an HTML tag using the style attribute.	It is used within the <head> section of an HTML document.	It is used in a separate .css file

Affects a single element or a group of elements.	Affects multiple elements within the same HTML element.	Affects multiple HTML documents or an entire website.
Not reusable. Styles need to be repeated for each element.	Can be reused on multiple elements within the same HTML document.	Can be reused on multiple HTML documents or an entire website.
Highest priority. Overrides internal and external styles.	Medium priority. Overrides external styles but can be overridden by inline styles.	Lowest priority. Can be overridden by both inline and internal styles.
Inline styles increase the HTML file size, which can affect the page load time.	Internal styles are part of the HTML file, which increases the file size.	External styles are in a separate file, which reduces the HTML file size and can be cached for faster page loads.
Not easy to maintain. Changes need to be made manually to each element.	Relatively easy to maintain. Changes need to be made in one place in the <head> section.	Easiest to maintain. Changes need to be made in one place in the external .css file.

Advantages of internal, external, and inline

Internal	External	Inline
Keeps styles with the document (convenient for single-page documents).	Reusable across many pages; promotes DRY.	Applies immediately and has highest specificity for that element.
Good for page-specific overrides.	Keeps HTML clean (separation of concerns).	Handy for quick one-off tweaks or dynamically injected styles via JS.
No extra file to manage for small projects.	Browser caching improves performance on repeat visits.	
	Easier team collaboration & maintainability.	

Disadvantages of internal, external, and inline

Internal	External	Inline
Not reusable across pages (duplicates if used on many pages).	Requires correct linking/path.	Hard to maintain and scale; mixes content & presentation.
Larger HTML file; weaker separation of concerns.	Extra HTTP request (minimal with HTTP/2 & caching).	Cannot be reused; leads to repetition and bloated HTML.
Less efficient caching than external files.	- If the file fails to load the site can be unstyled.	Can make overriding and debugging harder.