**Module 16 : CSS In Full Stack Course**

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

**Answer:** A **CSS selector** is a pattern used to select HTML elements you want to style. It tells the browser **which elements in the HTML should get the specified CSS rules**.

Type of Selector:

**1) Element Selector:**

p {

color: blue;

font-size: 16px;

}

**2) Class Selector:**

.highlight {

background-color: yellow;

font-weight: bold;

}

**3) ID Selector:**

#txt {

Background-color: green;

font-weight: italic;

}

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

**Answer:** CSS **specificity** is the set of rules that determine **which CSS style is applied** when multiple selectors target the same element.

Specificity is calculated based on the type of selectors: Element, Class and ID

p { color: blue; }

.highlight { color: green; }

#para { color: red; }

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

**Answer:**

**Inline CSS:**  
Written inside the HTML element using the style attribute.  
 **Advantage:** Quick for small changes.  
 **Disadvantage:** Hard to maintain, no reusability.

**Internal CSS:**  
Written in <style> inside <head>.  
 **Advantage:** Good for single page.  
 **Disadvantage:** Not reusable, increases page size.

**External CSS:**  
Stored in a separate .css file and linked with <link>.  
 **Advantage:** Best for large/multi-page sites, reusable, easy to maintain.  
 **Disadvantage:** Requires extra HTTP request.

**Question 4) Explain the CSS box model and its components (content, padding, border, and margin). How does each affect the size of an element?**

**Answer:** The **CSS Box Model** is the fundamental layout principle in CSS that defines **how every element on a web page is represented as a rectangular box** and how its total size is calculated.

* **Content** – Actual text or image inside. Controlled by width and height.
* **Padding** – Space between content and border.
* **Border** – Surrounds padding and content.
* **Margin** – Space outside the border.

**Question 5) What is the difference between border-box and content-box box-sizing in CSS? Which is the default?**

**Answer:**

* **Content-box:**  
  Width & height apply **only to content**. Padding & border **increase total size**.
* **Border-box:**  
  Width & height include **content + padding + border**. Total size stays fixed.

**Default:** Content-box.

**Question 6) What is CSS Flexbox, and how is it useful for layout design? Explain the terms flex-container and flex-item.**

**Answer: CSS Flexbox** is a layout model that helps create responsive and flexible designs easily.

* **Flex-Container** → The parent element with display: flex;, which controls the layout and alignment.
* **Flex-Item** → The child elements inside the container that adjust and align according to Flexbox rules.

**Question 7) Describe the properties justify-content, align-items, and flex-direction used in Flexbox.**

**Answer:**

* **Justify-content** → Aligns items on the **main axis** (e.g., left, right, center, space-between).
* **Align-items** → Aligns items on the **cross axis** (e.g., top, bottom, center, stretch).
* **Flex-direction** → Sets the **axis direction** (row, column, or reverse).

**Question 8) Explain CSS Grid and how it differs from Flexbox. When would you use Grid over Flexbox?**

**Answer: CSS Grid** is a 2D layout system that lets you arrange elements in **rows and columns**.

**Difference:**

* **Flexbox** → 1D layout (row **or** column).
* **Grid** → 2D layout (row **and** column).

**Question 9) Describe the grid-template-columns, grid-template-rows, and grid-gap properties. Provide examples of how to use them.**

**Answer:**

* **grid-template-columns** → Defines the number & width of columns.

grid-template-columns: 200px 1fr 2fr;

* **grid-template-rows** → Defines the number & height of rows.

grid-template-rows: 100px auto;

* **grid-gap** (or gap) → Sets spacing between rows & columns.

grid-gap: 20px;

**Question 10) What are media queries in CSS, and why are they important for responsive design?**

**Answer: Media queries** in CSS allow you to apply styles based on device features like **screen size, resolution, or orientation**.

@media (max-width: 600px) {

body {

background: lightblue;

}

}

**Question 11) Write a basic media query that adjusts the font size of a webpage for screens smaller than 600px.**

**Answer:**

@media (max-width: 600px) {

body {

font-size: 14px;

}

}

**Question 12) Explain the difference between web-safe fonts and custom web fonts. Why might you use a web-safe font over a custom font?**

**Answer:**

* **Web-safe fonts** → Pre-installed on most devices (e.g., Arial, Times New Roman, Verdana). They don’t need downloading, so they load fast and look consistent across systems.
* **Custom web fonts** → Imported using @font-face or services like Google Fonts. They provide unique styles but require downloading, which may affect performance.

**Question 13) What is the font-family property in CSS? How do you apply a custom Google Font to a webpage?**

**Answer:**

* **font-family** → The CSS property used to define the typeface of text. You can list multiple fonts as fallbacks.

.p {

font-family: Arial, Helvetica, sans-serif;

}

* **Applying a Google Font:**

1. Import the font in HTML <head>:

<link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">

2. Use it in CSS with font-family:

body {

font-family: 'Roboto', sans-serif;

}

**Submitted By Dhairya Thakkar**