Module 16: CSS in Full Stack Course

❖ CSS Selectors & Styling

Q1]: What is a CSS selector? Provide examples of element, class, and ID selectors.

A **CSS selector** is a pattern used to select and style HTML elements. It tells the browser which HTML elements should be affected by a CSS rule.

- **Element Selector**: Targets HTML elements by tag name.

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

(Styles all <p> elements.)

- **Class Selector**: Targets elements with a specific class.

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

(Styles elements with class="highlight")

- **ID Selector**: Targets a unique element with a specific ID.

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

Q2]: Explain the concept of CSS specificity. How do conflicts between multiple styles get resolved?

CSS specificity determines which rule is applied when multiple rules target the same element. The more specific a selector, the higher its priority.

Specificity hierarchy (from lowest to highest):

1. Element selectors (e.g., div, p)
2. Class selectors, attributes, and pseudo-classes (e.g., .class, [type="text"], :hover)
3. ID selectors (e.g., #id)
4. Inline styles (e.g., style="color:red;")
5. !important overrides all (but should be avoided when possible)

Conflict resolution: The browser applies the style with the highest specificity. If specificity is equal, the last rule defined in the CSS wins.

Q3]: What is the difference between internal, external, and inline CSS? Discuss the advantages and disadvantages of each approach.

Type	Description	Example	Advantages	Disadvantages
Inline CSS	CSS applied directly to an HTML element using style attribute	Hello	Quick for small changes	Hard to maintain; poor reusability
Internal CSS	CSS written inside <style> tags in the HTML <head>	<style> p { color: blue; } </style>	Useful for single pages	Doesn't scale well for multiple pages
External CSS	CSS written in a separate .css file linked to HTML	<link rel="stylesheet" href="styles.css">	Best for large sites; reusable	One extra HTTP request; not visible without the file

❖ CSS Box Model

Q4]: Explain the CSS box model and its components (content, padding, border, margin). How does each affect the size of an element?

The CSS box model describes how every HTML element is structured in terms of boxes.

1. Content – The actual content (text, image, etc.)
2. Padding – Space between content and border
3. Border – The edge surrounding the padding
4. Margin – Space outside the border (between elements)

Total width = content + padding + border + margin
Each layer adds to the overall space the element takes up.

Q5]: What is the difference between border-box and content-box box-sizing in CSS? Which is the default?

- **content-box** (default):
  width and height apply only to the content. Padding and border are added outside it.
- **border-box**:
  width and height include content + padding + border. Easier for layout control.

Example:

box-sizing: border-box;

Default: content-box
Preferred in modern layouts: border-box for easier sizing control.

❖ CSS Flexbox

Q6]: What is CSS Flexbox, and how is it useful for layout design? Explain the terms

flex-container and flex-item.

Flexbox is a CSS layout module that allows flexible and efficient arrangement of elements in one direction (row or column).

- **flex-container**: The parent element with display: flex;
- **flex-items**: The child elements inside the container

Advantages:

- Automatic spacing and alignment
- Responsive by default
- Easily handles dynamic content sizes

Q7]: Describe the properties justify-content, align-items, and flex-direction used in Flexbox.

1. **flex-direction**: Defines the main axis direction.

Syntax: flex-direction: row; /* default */

flex-direction: column;

2. **justify-content**: Aligns items along the **main axis**.

Syntax: justify-content: center;

justify-content: space-between;

3. **align-items**: Aligns items along the **cross-axis**.

Syntax: align-items: center;

align-items: stretch; /* default */

❖ CSS Grid

Q8]: Explain CSS Grid and how it differs from Flexbox. When would you use Grid over Flexbox?

CSS Grid: It is a layout system for creating two-dimensional layouts (rows and columns).

Flexbox : It is one-dimensional (row *or* column).

Flexbox	**Grid**
1D (either row or column)	2D (rows and columns)

Flexbox	Grid
Great for content flow	Great for page layout
Items flow naturally	Grid defined upfront

Use Grid when you need a structured layout (e.g., website layout, dashboards).
Use Flexbox for aligning elements inside containers (e.g., buttons, navbars).

Q9]: Describe the grid-template-columns, grid-template-rows, and grid-gap properties. Provide examples of how to use them.

- **grid-template-columns**: Defines the number and width of columns.
- grid-template-columns: 1fr 2fr 1fr;
- **grid-template-rows**: Defines the number and height of rows.
- grid-template-rows: 100px auto;
- **grid-gap** (or gap): Sets spacing between rows and columns.
- gap: 20px;

Example:

```
.container {
  display: grid;
  grid-template-columns: 200px 1fr;
  grid-template-rows: 100px auto;
  gap: 10px;
}
```

❖ **Responsive Web Design with Media Queries**

Q 10]: What are media queries in CSS, and why are they important for responsive design?

Media queries are CSS techniques used to apply styles depending on the device's screen size, resolution, or orientation.

Syntax:

```
@media (max-width: 600px) {
  body {
    background-color: lightblue;
  }
```

}

Importance:

- Ensures content looks good on all screen sizes
- Helps create responsive designs.
- Improves usability on mobile, tablet, and desktop

Q 11]: Write a basic media query that adjusts the font size of a webpage for screens smaller than 600px.

Example: @media (max-width: 400px){

body {

font-size: 14px;

background-color: pink; } }

❖ **Typography and Web Fonts**

Q 12]: Explain the difference between web-safe fonts and custom web fonts. Why might you use a web-safe font over a custom font?

- **Web-safe fonts** are standard fonts pre-installed on most operating systems.
  Examples: Arial, Times New Roman, Courier New, Verdana.
- Web-safe fonts are used when speed and compatibility are more important than design uniqueness.
- **Custom web fonts** are fonts that are downloaded from the web and not necessarily installed on the user's device. These are often loaded using services like Google Fonts.

Why choose web-safe fonts?

- No extra loading time (already on the device)
- More reliable performance
- Better for emails and older browsers

Why choose custom fonts?

- Branding and design consistency
- Unique and modern typography

Q 13]: What is the font-family property in CSS? How do you apply a custom Google Font to a webpage?

➢ **font-family:** The font-family property defines the typeface used for text.

Example:

```
p {
  font-family: 'Arial', sans-serif;
}
```

You can specify a fallback chain (e.g., if the first font fails, use the next one).

How to use a Google Font:

Step 1 : Go to Google Fonts and select a font (e.g., *Roboto*).

Step 2: Copy the <link> tag into your HTML <head> :

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

Step3 : Use the font in your CSS:

```
body {
  font-family: 'Roboto', sans-serif;
}
```
