CSS Assignments - Theory

# CSS Selectors & Styling

* Question 1: 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. Examples:  
- Element selector: p { color: blue; } (applies to all <p> elements)  
- Class selector: .btn { background: green; } (applies to elements with class='btn')  
- ID selector: #header { font-size: 20px; } (applies to the element with id='header')

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

CSS specificity determines which CSS rule is applied when multiple rules target the same element. Specificity is based on a hierarchy:  
1. Inline styles (highest priority)  
2. ID selectors  
3. Class, attribute, and pseudo-class selectors  
4. Element and pseudo-element selectors  
When conflicts occur, the rule with higher specificity wins. If specificity is the same, the last rule in the stylesheet is applied.

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

- Inline CSS: Written inside the HTML element using the style attribute. Quick but not reusable.  
- Internal CSS: Written inside a <style> tag within the <head>. Good for small projects but not reusable across multiple pages.  
- External CSS: Written in a separate .css file and linked via <link>. Best practice for large projects as it separates style from content and allows reusability.

# CSS Box Model

* Question 1: 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 element on a webpage is represented as a rectangular box. Components:  
- Content: The actual text or image inside the box.  
- Padding: The space between the content and the border (increases element size).  
- Border: The edge around padding and content.  
- Margin: The space outside the border (creates space between elements).  
Padding, border, and margin all affect the total space the element occupies.

* Question 2: What is the difference between border-box and content-box box-sizing in CSS? Which is the default?

- content-box: Default value. The width and height only apply to the content. Padding and border are added outside.  
- border-box: The width and height include content, padding, and border, making layout easier to manage.  
Default: content-box.

# CSS Flexbox

* Question 1: 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 model designed for arranging elements in a flexible and efficient way. It helps align items horizontally and vertically, and makes responsive design easier.  
- flex-container: The parent element with display: flex; that holds flex items.  
- flex-item: The child elements inside the flex container.

* Question 2: Describe the properties justify-content, align-items, and flex-direction used in Flexbox.

- justify-content: Aligns items horizontally (start, end, center, space-between, space-around).  
- align-items: Aligns items vertically (stretch, start, center, end).  
- flex-direction: Defines the direction of items (row, column, row-reverse, column-reverse).

# CSS Grid

* Question 1: Explain CSS Grid and how it differs from Flexbox. When would you use Grid over Flexbox?

CSS Grid is a two-dimensional layout system (rows and columns), while Flexbox is one-dimensional (row or column). Use Grid for complex layouts with rows and columns, and Flexbox for simpler, one-directional layouts.

* Question 2: Describe the grid-template-columns, grid-template-rows, and grid-gap properties. Provide examples of how to use them.

- grid-template-columns: Defines column structure (e.g., grid-template-columns: 1fr 2fr 1fr;).  
- grid-template-rows: Defines row structure (e.g., grid-template-rows: 100px 200px;).  
- grid-gap: Defines spacing between grid items (e.g., grid-gap: 20px;).

# Responsive Web Design with Media Queries

* Question 1: What are media queries in CSS, and why are they important for responsive design?

Media queries are CSS rules that apply styles based on device characteristics like screen size. They are important for making websites responsive, ensuring usability across desktops, tablets, and mobile devices.

* Question 2: Write a basic media query that adjusts the font size of a webpage for screens smaller than 600px

Example:  
@media (max-width: 600px) {  
 body { font-size: 14px; }  
}

# Typography and Web Fonts

* Question 1: 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: Pre-installed on most devices (e.g., Arial, Times New Roman). They load quickly but have limited variety.  
- Custom web fonts: Loaded from external sources (e.g., Google Fonts). They provide more design flexibility but may slow down page load.  
Web-safe fonts are used when performance and compatibility are a priority.

* Question 2: What is the font-family property in CSS? How do you apply a custom Google Font to a webpage?

The font-family property specifies the typeface used for text. To apply a Google Font:  
1. Import the font in the <head>:  
<link href='https://fonts.googleapis.com/css2?family=Roboto&display=swap' rel='stylesheet'>  
2. Apply it in CSS:  
body { font-family: 'Roboto', sans-serif; }