CSS Theory Assignment

# CSS Selector & Styling

* Question 1

A CSS selector is a pattern used to select the HTML elements you want to style.  
• Element selector: p { color: blue; }  
• Class selector: .btn { background-color: green; }  
• ID selector: #header { font-size: 24px; }

* Question 2

CSS specificity decides which style is applied when multiple rules target the same element.  
• ID selectors have the highest specificity.  
• Class selectors are in the middle.  
• Element selectors have the lowest.  
Higher specificity wins. If there is a tie, the last rule written in the CSS wins.

* Question 3

Type Advantages Disadvantages  
Inline Quick and easy for small changes Not reusable, hard to maintain  
Internal Good for single-page styling Not reusable across pages  
External Reusable, cleaner, and efficient Needs extra HTTP request

# CSS Box Model

* Question 1

The CSS box model describes how elements are structured and spaced.  
• Content: The actual text or image.  
• Padding: Space inside the element, around the content.  
• Border: The line surrounding the padding and content.  
• Margin: Space outside the border, between elements.  
Total size = Content + Padding + Border + Margin

* Question 2

• content-box (default): Width and height apply only to content.  
Padding and border are added outside the content.  
• border-box: Width and height include content + padding + border.  
Default is content-box.

# CSS Flexbox

* Question 1

CSS Flexbox is a layout model that helps arrange items in a flexible and responsive way, especially in one direction (row or column).  
• Flex-container: The parent element that holds flex items. Set using display: flex.  
• Flex-item: The child elements inside the flex container that are arranged using Flexbox rules.  
Flexbox makes alignment, spacing, and resizing easier across screen sizes.

* Question 2

• justify-content: Aligns items horizontally (main axis). Example: center, flex-start, space-between  
• align-items: Aligns items vertically (cross axis). Example: center, stretch, flex-end  
• flex-direction: Sets the direction of the main axis. Example: row, column, row-reverse

# CSS Grid

* Question 1

CSS Grid is a layout system for creating 2D layouts — it works in rows and columns.  
Flexbox is mainly for 1D layout (row or column).  
Grid is best for complex layouts with both rows and columns.  
Use Grid when you need full control over both directions.  
Use Flexbox for simpler, one-direction layouts.

* Question 2

• grid-template-columns: Defines the number and size of columns. Example: grid-template-columns: 200px 1fr;  
• grid-template-rows: Defines the number and size of rows. Example: grid-template-rows: 100px 2fr;  
• grid-gap (or gap): Adds space between rows and columns. Example: grid-gap: 20px;  
Example CSS:  
.container {  
 display: grid;  
 grid-template-columns: 1fr 1fr;  
 grid-template-rows: 100px 100px;  
 gap: 10px;  
}

# Responsive Web Design with Media Queries

* Question 1

Media queries are CSS rules that apply styles based on the device’s screen size, resolution, or type.  
They help make websites responsive — meaning they look good on all screen sizes, from mobiles to desktops.

* Question 2

@media (max-width: 600px) {  
 body {  
 font-size: 14px;  
 }  
}  
This makes the text smaller on small screens for better readability.

# Typography and Web Fonts

* Question 1

• Web-safe fonts are common fonts already installed on most devices (e.g., Arial, Times New Roman).  
• Custom web fonts (like Google Fonts) are downloaded from the web when the page loads.  
Use web-safe fonts for faster loading and compatibility. Use custom fonts for branding.

* Question 2

The font-family property sets the font of text in CSS.  
Example:  
body {  
 font-family: 'Arial', sans-serif;  
}  
  
To apply a custom Google Font:  
1. Add link in <head>:  
<link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">  
2. Use in CSS:  
body {  
 font-family: 'Roboto', sans-serif;  
}