CSS Assignment

CSS Selectors & Styling

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

Answer: A CSS selector is used to target HTML elements to apply styling. It helps determine which elements on a webpage will be affected by a particular rule.

Examples:

- **Element Selector: Targets all elements of a specific type.**
  Example: `p { color: blue; }`
- **Class Selector: Targets elements with a particular class attribute.**
  Example: `.box { font-size: 16px; }`
- **ID Selector: Targets a specific element with a unique ID.**
  Example: `#header { background-color: gray; }`
- **Universal Selector: Applies the rule to every element on the page.**
  Example: `* { margin: 0; padding: 0; }`
- **Group Selector: Applies the same styles to multiple elements.**
  Example: `h1, h2, h3 { font-family: Arial, sans-serif; }`

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

Answer: CSS specificity determines the priority of rules when multiple styles are applied to an element. The more specific a selector, the higher its priority.

Specificity Calculation:

1. **Inline styles: 1000**
2. **ID selectors: 100**
3. **Class selectors, attributes, and pseudo-classes: 10**
4. **Element selectors: 1**
5. **Universal selector and inherited styles: 0**

Conflict Resolution:

- **A more specific rule wins.**
- **If two rules have equal specificity, the last rule in the CSS file will be applied.**
- `!important` **overrides all unless another** `!important` **rule with higher specificity exists.**

Example with `!important`**:**

css

Copy

```
p { color: red !important; }
```

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

Answer: There are three primary methods for applying CSS: Inline, Internal, and External.

1. **Inline CSS**
    - **Directly applied within the HTML element using the `style` attribute.**
    - **Advantages: Quick styling, overrides other styles.**
    - **Disadvantages: Hard to maintain, reduces reusability, increases HTML file size.**
      Example: `<p style="color: red;">Text</p>`
2. **Internal CSS**
    - **Defined in the `<style>` tag in the `<head>` section of an HTML document.**
    - **Advantages: Good for single-page websites.**
    - **Disadvantages: Not reusable across multiple pages, can increase page load time if the CSS is large.**
      Example:

html
Copy

```
<style>

  p { color: blue; }

</style>
```

3.
4. **External CSS**
    - **Linked from an external `.css` file using the `<link>` tag.**
    - **Advantages: Best for larger projects, improves maintainability and reusability, reduces HTML file size, allows caching.**
    - **Disadvantages: Requires an extra HTTP request, and styles won't apply if the CSS file is missing.**
      Example:

html
Copy

```
<link rel="stylesheet" href="styles.css">
```

5.

CSS Box Model

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

Answer: The CSS Box Model is a fundamental concept in web design, defining the structure of HTML elements and how their size is calculated. It consists of four parts:

1. **Content: The actual content, like text or an image.**
2. **Padding: Space around the content, inside the element's border.**
3. **Border: The area around the padding, separating the element from others.**
4. **Margin: Space outside the border, providing distance between elements.**

Size Calculation:
The total width and height of an element are calculated as follows:

```
Total width = Content width + Padding + Border + Margin
```

Example:

- **Content width: 200px**
- **Padding: 10px on both sides**
- **Border: 5px on both sides**
- **Margin: 5px on both sides**
  Total width = 200px + 10px + 10px + 5px + 5px = 230px

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

Answer:

1. **content-box (default):**
   - **The width and height only apply to the content area. Padding and border are added separately, expanding the total size.**
2. **border-box:**
   - **The width and height include padding and border, meaning the content area shrinks to fit within the specified dimensions.**

Example for content-box:

- **Final width = 200px (content) + 10px (padding) + 10px (border) = 230px.**

Example for border-box:

- **Final width = 200px (no added size for padding or border).**

CSS Flexbox

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 makes it easier to design flexible, one-dimensional layouts. It enables responsive designs that adapt to the container's size.

- **Flex-container: The parent element that holds all flex items. It is defined with `display: flex;`.**
- **Flex-item: The child elements inside the flex container. They adjust according to the container's layout rules.**

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

Answer:

1. **justify-content: Aligns items along the main axis (horizontal by default).**
    - **Values: `flex-start`, `flex-end`, `center`, `space-between`, `space-around`, `space-evenly`.**
2. **align-items: Aligns items along the cross axis (vertical by default).**
    - **Values: `stretch` (default), `flex-start`, `flex-end`, `center`, `baseline`.**
3. **flex-direction: Defines the direction of the main axis.**
    - **Values: `row` (default), `row-reverse`, `column`, `column-reverse`.**

CSS Grid

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

Answer: CSS Grid is a two-dimensional layout system, designed for arranging items into rows and columns. In contrast, Flexbox is one-dimensional, either aligning items in a row or a column.

Differences:

Feature	CSS Grid	Flexbox
Layout Type	2D (rows & columns)	1D (row OR column)
Item Control	Precise positioning	Content-driven
Best for	Complex layouts	Simple alignments

When to use Grid over Flexbox?

- **Grid is ideal for complex layouts involving both rows and columns.**
- **Flexbox is better for simpler, linear layouts like navigation bars.**

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

Answer:

1. **grid-template-columns: Defines the number and size of columns.**
    - **Example: `grid-template-columns: 100px 200px auto;`**
2. **grid-template-rows: Defines the number and size of rows.**
    - **Example: `grid-template-rows: 100px 150px auto;`**
3. **grid-gap (or `gap`): Specifies the space between grid items.**
    - **Example: `gap: 20px;`**

Responsive Web Design with Media Queries

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

Answer: Media queries allow different CSS styles to be applied based on conditions such as screen size, resolution, or device type, making websites more responsive.

Why Are They Important?

- **They enable layouts to adapt to different screen sizes, enhancing user experience.**
- **They allow for mobile-first or desktop-first design strategies.**

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

Answer:

css

Copy

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

Typography and Web Fonts

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). They load quickly but offer fewer style choices.**
- **Custom web fonts: Loaded from external sources (e.g., Google Fonts), offering more design options but potentially increasing load time.**

Why Use Web-Safe Fonts?

- **Faster load times and universal compatibility.**
  Why Use Custom Fonts?
- **For unique branding and more design flexibility.**

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

Answer: The `font-family` property specifies the font of the text. Multiple fonts can be listed as fallbacks.

Example of applying a custom Google Font:

Add the following `<link>` tag to the `<head>` section:
html
Copy

```
<link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&dis
play=swap" rel="stylesheet">
```

1.

In the CSS, use the `font-family` property:
css
Copy

```
body {

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

}
```