**Frontend – CSS and CSS**

**1. What is a CSS selector?**

**ANS** - A **CSS selector** targets HTML elements to apply styles.

* **Element selector:** p { color: red; }
* **Class selector:** .btn { background: blue; }
* **ID selector:** #header { font-size: 20px; }

**2. What is CSS specificity?**

**ANS- Specificity** determines which CSS rule is applied when multiple rules target the same element. It’s calculated based on:

* Inline styles (highest)
* IDs > Classes > Elements

**Conflict resolution:** More specific rules override less specific ones.

**3. Internal, External, Inline CSS**

| **ANS - Type** | **Definition** | **Advantages** | **Disadvantages** |
| --- | --- | --- | --- |
| **Inline** | In style="" attribute in HTML tag | Quick fixes, overrides other styles | Hard to maintain |
| **Internal** | In <style> tag in HTML <head> | Useful for single-page styles | Not reusable |
| **External** | Linked .css file via <link> | Reusable, clean, easier to maintain | Extra HTTP request (unless cached) |

**4. CSS Box Model**

ANS - Components:

1. **Content** – Actual text or image.
2. **Padding** – Space around content.
3. **Border** – Edge around padding.
4. **Margin** – Space outside the border.

**Total size = content + padding + border + margin**

1. **border-box vs content-box**

**ANS-**

* **content-box (default):** Width = content only. Padding & border add to total size.
* **border-box:** Width includes content + padding + border. Easier for layout control.

**6. What is CSS Flexbox?**

ANS- Flexbox is a layout model for aligning items in a **1D row or column**.

* **Flex-container:** Parent with display: flex
* **Flex-item:** Direct children of the container

1. **Flexbox Properties**

**ANS-**

* \*\*justify-content\*\*: Aligns items **horizontally** (flex-start, center, space-between)
* \*\*align-items\*\*: Aligns items **vertically** (flex-start, center, stretch)
* \*\*flex-direction\*\*: Row or column layout (row, column, etc.)

1. **CSS Grid vs Flexbox**

**ANS-**

* **Grid**: 2D layout (rows & columns)
* **Flexbox**: 1D layout (row **or** column)

**Use Grid** when you need complex layouts with both rows & columns.

1. **Grid Properties**

**ANS-**

* \*\*grid-template-columns\*\*: Defines column sizes  
  grid-template-columns: 1fr 2fr;
* \*\*grid-template-rows\*\*: Defines row sizes  
  grid-template-rows: 100px auto;
* \*\*grid-gap\*\*: Space between rows/columns  
  grid-gap: 20px;

**10. What are media queries?**

ANS - Media queries apply styles based on device features (like screen size).  
**Important for responsive design** (adjusts layout on different devices).

**11. Basic media query example**

@media (max-width: 600px) {

body {

font-size: 14px;

}

}

**12. Web-safe fonts vs Custom fonts**

* **Web-safe fonts**: Pre-installed on most devices (e.g., Arial, Times)
* **Custom fonts**: Downloaded from web (e.g., Google Fonts)

**Use web-safe** for performance and compatibility.  
**Use custom** for unique branding and design.

**13. What is the font-family property?**

Specifies the font to use:

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

**Using Google Fonts:**

1. Add link in <head>:

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

1. Use in CSS:

body {

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

}