Module 3 – Mernstack – CSS and CSS3

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

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

* Simple selectors (select elements based on name, id, class)
* [Pseudo-class selectors](https://www.w3schools.com/css/css_pseudo_classes.asp) (select elements based on a certain state)
* [Pseudo-elements selectors](https://www.w3schools.com/css/css_pseudo_elements.asp) (select and style a part of an element)
* [Attribute selectors](https://www.w3schools.com/css/css_attribute_selectors.asp) (select elements based on an attribute or attribute value)

EX:

<!DOCTYPE html>

<html>

<head>

  <title>CSS Selectors Example</title>

  <style>

    /\* Element selector \*/

    p {

      color: black;

    }

    /\* Class selector \*/

    .highlight {

      color: blue;

    }

    /\* ID selector \*/

    #unique {

      color: red;

    }

  </style>

</head>

<body>

  <p>This is a paragraph (element selector).</p>

  <p class="highlight">This is a highlighted paragraph (class selector).</p>

  <p id="unique">This is a unique paragraph (ID selector).</p>

</body>

</html>

**OUTPUT:**

This is a paragraph (element selector).

This is a highlighted paragraph (class selector).

This is a unique paragraph (ID selector).

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

***Specificity:*** If there are two or more CSS rules that point to the same element, the selector with the highest specificity will "win", and its style declaration will be applied to that HTML element.

| **Level** | **Description** | **Example** |
| --- | --- | --- |
| a | Inline styles | <p style="color:red"> |
| b | Number of ID selectors | #header |
| c | Number of class, attribute, pseudo-classes | .box, [type="text"], :hover |
| d | Number of element and pseudo-elements | div, p, ::before |

| **Rule Type** | **Specificity** |
| --- | --- |
| Element | Low |
| Class | Medium |
| ID | High |
| Inline Style | Very High |
| !important | Overrides All |

### **How Conflicts Are Resolved:**

When multiple rules apply to the same element:

1. **More specific selector wins**  
   (e.g., #id beats .class)
2. **If specificity is equal**, the **later rule in the CSS file** wins.
3. **!important overrides everything**, even inline styles (unless another !important is more specific).

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

Inline CSS:

<p style="color: red; font-size: 16px;">This is inline styled text.</p>

Internal CSS:

<!DOCTYPE html>

<html>

  <head>

    <title>CSS Selectors Example</title>

    <style>

      p {

        color: black;

      }

    </style>

  </head>

  <body>

    <p>This is a paragraph (element selector).</p>

  </body>

</html>

External CSS:

<!DOCTYPE html>

<html>

  <head>

    <title>CSS</title>

    <link rel="stylesheet" href="style.css" />

  </head>

  <body>

    <p>This is a paragraph (element selector).</p>

  </body>

</html>

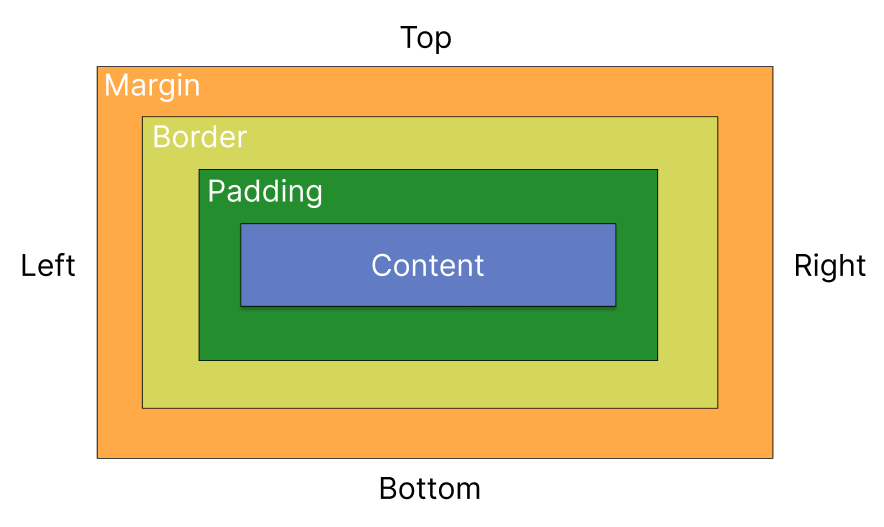
**Style.css**

p{

    color: red;

}

| **Feature** | **Inline CSS** | **Internal CSS** | **External CSS** |
| --- | --- | --- | --- |
| Location | Inside HTML element | <style> in <head> | Separate .css file |
| Reusability | ❌ No | ❌ No | ✅ Yes |
| Maintenance | ❌ Hard | ⚠️ Moderate | ✅ Easy |
| Performance | ⚠️ Poor (cluttered) | ⚠️ Slower (big files) | ✅ Better (cached) |
| Best for | Quick fixes/testing | Single page styling | Multi-page websites |

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

* **Content** - The content of the box, where text and images appear
* **Padding** - Clears an area around the content. The padding is transparent
* **Border** - A border that goes around the padding and content
* **Margin** - Clears an area outside the border. The margin is transparent

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

By default, elements have box-sizing: content-box

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>GeeksForGeeks</title>

    <style>

      h1 {

        color: green;

        text-align: center;

      }

      div {

        width: 200px;

        height: 200px;

        padding: 15px;

        border: 10px solid black;

        background: green;

        display: inline-block;

      }

      .content-box {

        box-sizing: content-box;

      }

      .border-box {

        box-sizing: border-box;

      }

    </style>

  </head>

  <body>

    <div class="content-box">

      <h3>Content Box</h3>

    </div>

    <div class="border-box">

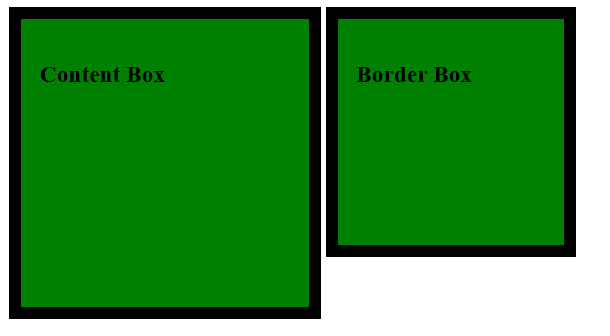
      <h3>Border Box</h3>

    </div>

  </body>

</html>

**OUTPUT :**

****

**Q6. What is CSS Flexbox, and how is it useful for layout design? Explain the terms flex-container and flex-item?**

* Flexbox is short for the Flexible Box Layout module.
* Flexbox is a layout method for arranging items in rows or columns.
* Flexbox makes it easier to design a flexible responsive layout structure, without using float or positioning.
* **Flex Container** - the parent (container) <div> element
* **Flex Items** - the items inside the container <div>

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

**justify-content** property aligns the flexible container's items when the items do not use all available space on the main-axis (horizontally).

[**align-items**](https://www.w3schools.com/cssref/css3_pr_align-items.php) property to align the items vertically.

**flex-direction** property specifies the direction of the flexible items.

**Q.8 Explain the difference between web-safe fonts and custom web fonts. Why might you use a web-safe font over a custom font?**

| **Aspect** | **Web-Safe Fonts** | **Custom Web Fonts** |
| --- | --- | --- |
| **Definition** | Pre-installed fonts available on most devices and browsers. | Fonts that are downloaded from the web when a page loads. |
| **Examples** | Arial, Times New Roman, Verdana, Courier New, Georgia | Google Fonts (e.g., Roboto, Lato), Adobe Fonts, custom @font-face fonts |
| **Browser Support** | Guaranteed to work across all browsers and operating systems | Depends on browser and internet connection |
| **Performance** | No additional load time since fonts are already on the device | Can increase page load time due to external font downloads |
| **Customization** | Limited to standard styles and weights | Highly customizable in style, weight, and appearance |
| **Control over Look** | Less design flexibility | Greater design and branding flexibility |

### **Why Might You Use a Web-Safe Font Over a Custom Font?**

1. **Performance & Speed**:  
   Web-safe fonts load instantly because they are already on the user’s system. No extra HTTP requests or downloads are needed, which helps improve **page load speed** — especially important for mobile or slow connections.
2. **Compatibility**:  
   Web-safe fonts are reliably rendered across all browsers, operating systems, and devices. This ensures **consistent appearance** and avoids fallback issues.
3. **Simplicity**:  
   For simple or utility-focused websites (e.g. admin panels, documentation), using web-safe fonts can reduce complexity without compromising functionality.
4. **Reduced Bandwidth Use**:  
   Since custom fonts are often downloaded from a server, they consume bandwidth. Web-safe fonts help **minimize bandwidth usage**, especially useful on low-data or metered connections.
5. **Fallback Readiness**:  
   They act as reliable fallbacks if custom fonts fail to load due to slow internet or restrictions.

**Q.9 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 font for an element.
* The font-family property can hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font.

There are two types of font family names:

* **family-name** - The name of a font-family, like "times", "courier", "arial", etc.
* **generic-family** - The name of a generic-family, like "serif", "sans-serif", "cursive", "fantasy", "monospace".

CUSTOM GOOGLE FONT APPLY:

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">

<style>

body {

  font-family: "Sofia", sans-serif;

}

</style>

</head>

<body>

<h1>Sofia Font</h1>

<p>Lorem ipsum dolor sit amet.</p>

<p>123456790</p>

</body>

</html>

OUTPUT:



**Q.10 Explain CSS Grid and how it differs from Flexbox. When would you use Grid over Flexbox?**

CSS [Grid Layout](https://www.geeksforgeeks.org/css/css-grid-property/), is a two-dimensional grid-based layout system with rows and columns. It makes easier to design web pages without having to use floats and positioning. Like tables, grid layout allow us to align elements into columns and rows.

CSS Grid arranges items in rows and columns (2-Dimension), while Flexbox aligns items in a single row or column (1-Dimension).

**Q.11 Describe the grid-template-columns, grid-template-rows, and grid-gap properties. Provide examples of how to use them.**

### **grid-template-columns**

**Purpose:**  
Defines the number and size of columns in a grid container.

EX: grid-template-columns: 200px 1fr 2fr;

### **grid-template-rows**

**Purpose:**  
Defines the number and size of rows in a grid container.

EX: grid-template-rows: 100px auto 1fr;

* **grid-gap (or better: *gap*)**

**Purpose:**  
Sets spacing (gutters) between **rows** and **columns** in the grid.

EX:

gap: 20px; /\* 20px gap for both rows and columns \*/

gap: 10px 30px; /\* 10px row gap, 30px column gap \*/