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.

I) Element Selector

```
P
{
        Color:blue;
}
```

II) Class Selector

```
.my class
{
Color:red;
}
```

III) ID Selector

```
#My id
{
        Color:green;
}
```

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

CSS specificity is a set of rules that determines which CSS rule is applied when multiple rules target the same element

Specificity (higher to lower)

1. **Inline style**
2. **Id selector**
3. **Class selector**
4. **Element selector**
5. **Universal selector**

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

    CSS styles are written directly within an HTML element using the `style` attribute

    - **Advantage**
        - Quick to apply for small, one-off changes.
        - Overrides internal and external styles

- Disadvantage

  - Poor code maintainability
  - Makes HTML messy and hard to read.
  - Difficult to apply consistent styling across pages.

2. Internal CSS

Definition:
CSS rules are written inside a `style` tag within the `head` section of the HTML file.

Advantages

- Keeps CSS and HTML in one file.
- Easier to style multiple elements without repetition.

Disadvantages:

- CSS is not reusable across multiple pages.
- Increases page size.
- Still mixes structure of HTMLand css style which is not ideal for large projects.

3. External CSS

Definition:
CSS rules are written in a separate `.css` file and linked to the HTML using the `link` tag.

Advantages:

- Clean separation of content and style.
- CSS can be reused across multiple HTML files.
- Better for performance.
- Easier to maintain and scale.

Disadvantages:

- Requires additional HTTP request.
- May take a bit more setup for beginners

4. **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 HTML element is a rectangular box with these 4 layers:

- Content – The actual text or image inside the box.
- Padding – Space inside the box, between content and border.
- Border – The line that wraps around padding and content.
- Margin – Space outside the box, separating it from other elements.

How Each Affects Size

Total Width =
`Content + Padding + Border`
(Margin is **outside** the box and adds space around it)

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

1. content-box

- Width and height apply only to the content.
- Padding and border are added outside the content size.

2. border-box

- **Width and height include content + padding + border.**
- Makes layout easier and more predictable.

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

CSS Flexbox is a one-dimensional layout model in CSS used to arrange items in a row or column efficiently, even when their sizes are unknown or dynamic.

flex-container: The **parent** element where Flexbox is applied.

flex-item: The **child** elements inside a flex container.

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

In Flexbox.

`1.flex-direction: the direction in which flex items are placed`

- row: left to right
- row-reverse: right to left
- column: top to bottom
- column-reverse: bottom to top

2.justify-content: Aligns flex items horizontal in row, vertical in column

- flex-start: items align to start
- flex-end: items align to end
- center: items align to center
- space-between: equal space **between** items
- space-around: equal space **around** items
- space-evenly: equal space **between and outside** items

3. align-items: Aligns items **along the cross axis**

- stretch (default): items stretch to fill container
- flex-start: items align to top (if row) or left (if column)
- flex-end: items align to bottom or right
- center: items align to center
- baseline: align based on text baseline

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

CSS Grid is 2D layout system in CSS that allows you to design grid-based layouts using rows and columns.

Flexbox, which is 1D, Grid handles both **horizontal and vertical** placement simultaneously.

Use Grid When:	Use Flexbox When:
You need full control over rows & columns	You are aligning items in a single line (1D)
You want to place items in specific areas	Items need to wrap or reorder easily
Layout is visual-first (e.g., webpage sections)	Layout is content-first (e.g., nav bar)

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

1.grid-template-columns

```
   . grid
 {
  display: grid;
  grid-template-columns: 100px 200px;
}
```

2. grid-template-rows

```
. grid
{
  display: grid;
  grid-template-rows: 50px 100px;
}
```

3. grid-gap

```
grid-row-gap: vertical space
grid-column-gap: horizontal space
Gap: Modern CSS uses gap for both
```

```
. grid
{
display: grid;
grid-template-columns: 1fr 1fr 1fr;
grid-template-rows: 100px 100px;
 gap: 20px;
}
```

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

Media queries are a feature in CSS that allow you to apply different styles based on the device's characteristics.

why are they important for responsive Design?

- Make websites responsive across devices.
- Improve user experience on all screen sizes.
- Enable mobile-first and adaptive design.
- Avoid the need for separate mobile websites.

- Optimize layout, fonts, and spacing for readability.
- Help websites look good in both portrait and landscape.

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

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8" />
 <meta name="viewport" content="width=device-width, initial-scale=1" />
 <title>Responsive Color Box</title>
 <style>
  .color-box {
   width: 200px;
   height: 200px;
   background-color: red;
   margin: 50px;
  }

  @media (max-width: 600px) {
   .color-box {
    background-color: blue;
   }
  }
 </style>
</head>
<body>

 <div class="color-box"></div>

</body>
</html>
```

12. 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

- These are **pre-installed fonts** available on most operating systems.
- Examples: Arial, Times New Roman, Courier New, Verdana, Georgia, Tahoma.
- No need to load files from the web — they are instantly available.

Custom Web Fonts

- Fonts **not typically installed** on user's devices.
- Delivered via the web using services like **Google Fonts**, Adobe Fonts, or self-hosted font files.
- Examples: Roboto, Open Sans, Lora, custom branding fonts.

Why Use Web-Safe Fonts Over Custom Fonts?

- **Faster page load times:** No extra font files to download.
- **Better performance on slow connections or older devices.**
- **Guaranteed consistent display** across all browsers and devices.
- **Simpler to implement** with no external dependencies.
- Useful when font style is **not critical** to brand or design.

13. **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 to be used for text content on a webpage.

How do you apply a custom Google Font to a webpage

1.Choose a font from Google Fonts.

2. Get the embed `link` tag

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