CSS Selectors & Styling

Que 1:

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

1. A CSS selector is a pattern used to select HTML elements that you want to style. It tells the browser which elements to apply CSS rules to.

EX. – Element Selector:

```
p {
    color: blue;
    font-size: 10px;
 }
```

Class Selector:

```
 .form-field {
    padding: 10px;
    border: 1px solid gray;
}
```

ID Selector:

```
 # {
    background-color: green;
    color: white;
  }
```

Que 2:

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

- CSS Specificity determines which CSS rule is applied when multiple rules target the same element.

Specificity hierarchy:

1. Element selectors (e.g., p) → low priority
2. Class, attribute, and pseudo-class selectors
   (e.g., .class, [type="text"], :hover) → medium priority
3. ID selectors (e.g., #id) → high priority
4. Inline styles (e.g., style="color:-red") → highest priority

Que 3:

What is the difference between internal, external, and inline CSS? Discuss the advantages and disadvantages.

Inline CSS:

Advantages:- Quick for small changes.

Disadvantages:- Hard to maintain, mixes content with style.

Internal CSS:

Advantages:- Easy for single-page styling, overrides external styles.

Disadvantages:- Cannot be reused across multiple pages.

External CSS:

Advantages:- Reusable, separates content and design, keeps HTML clean.

Disadvantages:- Requires an additional HTTP request, might take time to load.

CSS Box Model

Que 1:

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 elements are displayed and how their size is calculated. Every HTML element is considered a box consisting of four parts:

1. **Content** – The actual content of the element (text, image, etc.).
    - Size is defined by width and height.
2. **Padding** – Space inside the box, between content and border.
    - Adds extra space inside the element, increasing the total size.
3. **Border** – Surrounds the padding and content.
    - Thickness adds to the total width and height of the element.
4. **Margin** – Space outside the border, separating the element from others.
    - Does not affect the element’s own size but affects spacing around it.

Que 2:

Difference between border-box and content-box box-sizing in CSS.

Border-box:- Width & height include content + padding + border; total size stays the same.

- No default.

Content-box:- Width & height include only content; padding and border added outside.

- Default.

/*Content box*/

```
div {

  box-sizing: content-box;

  width: 200px;

  padding: 20px;

  border: 10px solid black;

}
```

/*border box*/

```
div {
  box-sizing: border-box;
  width: 200px;
  padding: 20px;
  border: 10px solid black;
}
```

CSS Flexbox

Que 1:

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

- CSS Flexbox (Flexible Box Layout) is a layout model that makes it easier to design responsive and flexible web layouts.
- It allows you to align, distribute, and reorder elements within a container efficiently, even when the size of elements is unknown or dynamic.

Key terms:

1.flex-container:-

- The parent element where display: flex is applied.
- It establishes a flex context for its children**.**

2.Flex-item:-

- The child elements of a flex-container.
- These items can grow, shrink, and be aligned based on flex properties.

Que 2:

Describe justify- content , align-items, and flex-direction properties.

Justify-content: Aligns flex items horizontally along the main axis.

- flex-start, flex-end, center, space-between, space-around.

Align-items: Aligns flex items vertically along the cross axis.

- flex-start, flex-end, center, stretch, baseline.

flex-direction: Defines the direction of the main axis.

- row (default), row-reverse, column, column-reverse.

CSS grid

Que 1:

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

- CSS Grid is a two-dimensional layout system in CSS that allows you to create layouts using rows and columns.
- It is highly suitable for complex layouts like galleries, dashboards, or product grids.

When to use Grid:

- You want both rows and columns aligned simultaneously
- Creating structured layouts like image galleries, dashboards, or card layouts

When to use Flexbox:

- Layout along one direction
- Aligning items in a row or column, flexible sizing

Que 2:

Describe the grid-template-columns, grid-template-rows, and grid-gap properties. Provide examples.

1.**grid-template-columns**:– Defines the number and width of columns in a grid.

```
- .container {

    Display: grid:

    Grid-template-columns: 100px 200px 100px:

  }
```

2.**grid-template-rows**:- Defines the height of rows.

- grid-template-rows: 150px 150px 150px;

3.**grid-gap**:- Adds spacing between rows and columns.

- grid-gap: 20px:

Responsive web design with media queries

Que 1:

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

- Media queries are a CSS feature that allows you to apply different styles depending on the characteristics of the device or screen, such as width, height, resolution, or orientation.

Importance for responsive design:

- Ensures a website looks good on all devices (desktop, tablet, mobile)
- Allows dynamic adjustments like font sizes, layout changes, or hiding elements.
- Improves user experience and accessibility on smaller or larger screens.

Que 2:

Write a basic media query that adjusts the font size for screens smaller than 600px

```
@media (max-width: 600px) {

  body {

    font-size: 14px;

   }

}
```

Typography and Web Fonts

Que 1:

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

1.web-safe fonts:

- Fonts that are commonly available across all devices and operating systems.
- Examples: arial , times new roman , courier new , verdana
- Advantages**:** Reliable, fast-loading, no extra files needed
- Disadvantages: Limited design options

2.custom web fonts:

- Fonts that are imported from external sources like Google Fonts or Adobe Fonts.
- Examples: roboto ,lobster ,open sans
- Advantages: Unique design, better branding
- Disadvantages**:** Can increase page load time; may not be available offline

Que 2:

What is the font-family property in CSS? How to apply a custom Google Font?

- font-family specifies the typeface for text content in HTML.
  ```
  Body {
     Font – family: Arial, sans-serif:
  }
  ```

Applying a Google Font:

- Go to Google Fonts and select a font (e.g. roboto)
- Copy the <link> into the <head> of your HTML
- <link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">

Apply the font in CSS:

-
  ```
  body {
     font-family: ‘Roboto’ , sans-serif:
   }
  ```