**Q1: What is axis in CSS Flexbox? Explain with an example.**

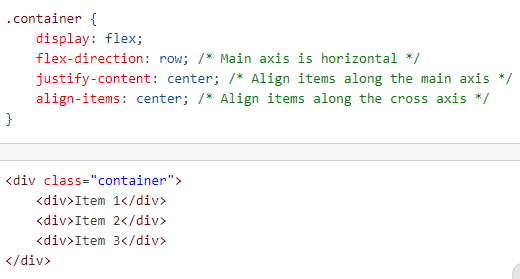
In CSS Flexbox, an axis refers to the direction in which flex items are placed. There are two axes:

* **Main Axis**:

It is the primary axis along which flex items are laid out. It is determined by the flex-direction property (e.g., row or column).

* **Cross Axis:**

It is the perpendicular axis to the main axis. It is used for alignment (e.g., align-items).



**Q2: Can I use multiple CSS Background images in a div? Explain with an example.**

Yes, we can use multiple background images in a single div using the background-image property. we can specify multiple images by separating them with commas.

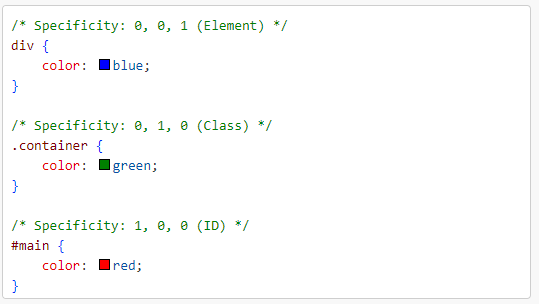


**Q4: Explain CSS Specificity**

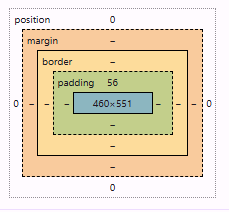
CSS specificity determines which CSS rule is applied when multiple rules target the same element. It is a way to calculate the importance of a selector.

**Specificity Calculation:**

* Inline Styles: Highest specificity (e.g., style="color: red;").
* IDs: More specific than classes (e.g., #id).
* Classes, Attributes, and Pseudo-classes: Less specific than IDs (e.g., .class, [attr], :hover).
* Elements and Pseudo-elements: Least specific (e.g., div, h1, ::before)



**Q5: Explain CSS Box Model**

The CSS Box Model describes how every HTML element is represented as a rectangular box. It consists of the following parts:

* Content: The actual content of the element (e.g., text, images).
* Padding: Space between the content and the border. It adds extra space inside the element.
* Border: The edge around the padding. It can have a width, style, and color.
* Margin: Space outside the border. It creates distance between elements.

**Q8: What are pseudo selectors? According to you explain 5 important pseudo selectors.**

Pseudo-selectors in CSS are special keywords used to style specific parts or states of elements without adding extra classes or IDs. They are prefixed with a colon (:) or double colon (::).

**5 Important Pseudo-selectors:**

1. **hover**

Styles an element when the user hovers over it with a mouse.

Example:

button:hover {

background-color: #B88E2F;

color: white;

}

1. **nth-child(n)**

Targets elements based on their position within a parent.

li:nth-child(2) {

    color: red; /\* Styles the second list item \*/

}

1. **3focus**

Styles an element when it is focused (e.g., clicked or tabbed into).

ex:

input:focus {

    border: 2px solid blue;

}

1. **first-child**

Targets the first child of a parent element.

Example:

p:first-child {

    font-weight: bold;

}

1. **not(selector)**

Selects elements that do not match the given selector.

Example:

div:not(.active) {

    opacity: 0.5;

}

**Why Use Pseudo-selectors?**

They allow you to style elements dynamically based on their state or position without modifying the HTML structure.

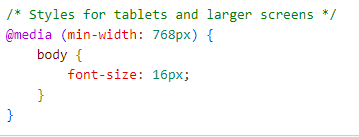
**Q9: What are media queries? Min-width or max-width which will be more helpful on Mobile First Approach? Explain.**

**Media Queries** in CSS are used to apply styles based on the characteristics of the user's device, such as screen size, resolution, or orientation. They help create responsive designs that adapt to different devices.

**Min-width vs Max-width:**

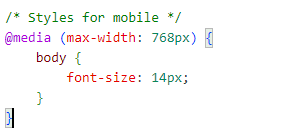
* *min-width:*

Targets devices larger than or equal to a specific width. It is ideal for a Mobile-First Approach because you start with mobile styles and add styles for larger screens.



* *max-width:*

Targets devices smaller than or equal to a specific width. It is more suitable for a Desktop-First Approach, where you start with desktop styles and adjust for smaller screens.



**Which is More Helpful for Mobile-First?**

min-width is more helpful for a Mobile-First Approach because:

* You design for smaller screens by default.
* You progressively enhance styles for larger screens.
* It aligns with modern responsive design practices.

**Q10: What is float property and why do we use clear property with float?**

The float property in CSS is used to position elements to the left or right of their container, allowing other content (like text) to flow around them.

**Why Use the float Property?**

* To create layouts where elements are aligned side by side.
* To wrap text around images or other elements.

Values of float:

* left: Floats the element to the left of its container.
* right: Floats the element to the right of its container.
* none: Default value; the element does not float.
* inherit: Inherits the float value from its parent.

