1. What Box Model in CSS?

   **ANS:** The Box model is fundamental concept in CSS that describes how elements are displayed and sized on the web page. It defines the properties that affect an element's dimensions such as its width, height, padding, border, and margin.

   1. Content: it represents the actual content of an element, such as text, image, video, and other HTML elements.

   2. Padding: It is space between the content and the element’s border. Padding can be set individually for each side (top, bottom, left, right) using padding property.

   3. Border: It is a line that surrounds the content and padding of an element. The border’s width style, and color can be defined using the border property.

   4. Margin: It is the space outside the border, creating a gap between the element and other elements on the page. Margin can be set individually for each side using the margin property.

2. What are the Different Types of selector in CSS & what are the advantages of them?

**ANS:** Selector in CSS is a way to specify which HTML elements on the web page you want to style. It helps you identify the elements you want to target by their names, classes, and IDs. By using selectors you can apply CSS rules and properties to those selected elements, controlling their appearance and behaviour.

1. **Universal Selector:** the universal selector (*) in CSS is used to select and apply styles to all elements on a web page.
   **Ex-** * {
   font-family: Arial, sans-serif;
   background-color: #f2f2f2;
   }
2. **Individual Selector:** individual selector in CSS is used to target and style a specific HTML element or group of elements.
   **Ex-** <h1 class="heading">Hello, World!</h1>
   <p id="intro">This is a paragraph.</p>

3. **Class Selector:** class selector in CSS is used to target and style a group of HTML elements that share the same class attribute. It allows you to apply

styles to multiple elements without having to repeat the styles for each individual element.

To use a class selector, you assign a class name to the desired elements in your HTML markup using the class attribute. Then, in your CSS, you reference that class name with a dot (.) to target all elements with that class.

**Ex-** .highlight {
  background-color: yellow;
  font-weight: bold;
}

4. **Id Selector:** ID selector in CSS is used to target and style a specific HTML element that has a unique identifier (ID) assigned to it.
To use an ID selector, you assign a unique ID to the desired element in your HTML markup using the id attribute. Then, in your CSS, you reference that ID name with a hash (#) to target that specific element.
**Ex-** #pageTitle {
  color: blue;
  font-size: 24px;
}

3. What is VW/VH?

**ANS: VW (View Width) and VH (View Height)** are units of measurement in CSS that represent a percentage of the width or height of the visible area of web page. They are relative units that adjust automatically based on the size of the viewport. **Pixel (px)**, on the other hand, are absolute units that represent a fixed number of pixels on the screen. Pixels provide precise control over element sizes but don't adjust based on the viewport size.
1vh = 10px
1vw = 10px

4. What difference between inline, inline-block and block?

**ANS:**
**Inline Element:-**

- Do not start on a new line; they flow within the text content.
- Cannot have a specified width or height.
- Margin and padding only affect the left and right sides, not top and bottom.
  **Example: <span>, <a>, <em>, <strong>**

**Inline-Block Element:-**

- Flow within the text content like inline element, its have a specific width, height, padding, and margin.

**Block:-** Start on a new line and take up the full available width of their parent element. its have specified width, height, padding and margin.

5. How is Border Box different from Content Box?

   **ANS: "Boz-sizing"** is CSS Property that determines how the width and height of an element are calculated, including content padding, and border. There are two values for the box-sizing property: "content-box", and "Border-box."

   **Content-box:** When the box-sizing property is set to "content-box"(The default value), the width and height of an element only including its content area. The padding and border are added to the specified width and height values. This means if you set 300px for an element, the total width of the element will be 300px plus the padding and border.

   **Border-box:** When the box-sizing property is set to "border-box" the width and height of an element include its content area, padding and border. The specified width and height values represent the entire space occupied by the element, including the padding and border. This means that if set a width 300px for an element, the total width of the element will be 300px and any padding and border will be included within that width.

6. What's Z-index and How Does it Function?

   **ANS:** The z-index property in CSS controls the stacking order of elements on a web page. It determines which element appears on top of others.
   The z-index property takes a numeric value as its argument. The higher the value, the closer the element is to the viewer's screen, and the more it will appear on top of other elements. Negative values are also allowed, pushing the element further back in the stacking order.

   ```
   .element1 {
     position: relative;
     z-index: 2;
   }

   .element2 {
     position: relative;
     z-index: 1;
   }
   ```
   In this example, element 1 will be displayed on top of element 2 because it has a higher z-index value.

7. What Grid and Flex and difference between them?

    - Grid is made for two-dimensional layout while Flexbox is for one. This means Flexbox can work on either row or columns at a time, but Grids can work on both.
    - Flexbox gives you more flexibility while working on either element (row or column). HTML markup and CSS will be easy to manage in this type of scenario.
    - GRID gives you more flexibility to move around the blocks irrespective of your HTML markup.

8. Difference between absolute and relative and sticky and fixed positions explain with example?

    **ANS:** CSS Position is a property that allows you to control the positioning of elements on a web page. It provides different positioning options to change the element's position.

    1. **Static position (Default);**
       Elements are positioned according to the normal flow of the document. Top, Bottom, left, right and z-index properties have no effect.

       ```
       Ex .element {
         position: static;
       }
       ```

    2. **Relative Position:**
       This position value is similar to static position, it follows HTML document flow. Allows to adjust the element's position using top, bottom, left, and right properties.

       ```
       Ex
       .element {
         position: relative;
         top: 20px;
         left: 10px;
       }
       ```

    3. **Absolute Position:** This position value is a powerful type of positioning it allows you to literally place anywhere on the page, exactly where you want, it depends on the parent element, not Html document flow. If there is no such parent element available it'll follow root element. You can set attributes like top, bottom, right, and left.

       ```
       Ex
       .element {
       ```

```
  position: absolute;
  top: 50px;
  right: 20px;
}
```

4. **Fixed Position:** This position does not depend on the parent element, HTML document flow and root element. It depends on the browser window itself. The view doesn't change when the window is scrolled.
   **Ex**
   ```
   .element {
     position: fixed;
     top: 0;
     right: 0;
   }
   ```

5. **Sticky Position:** Elements are positioned based on the user's scroll position.
   **Ex**
   ```
   .element {
     position: sticky;
     top: 20px;
   }
   ```

10. Build Responsive Layout Both desktop and mobile and Tablet, see below image for reference.

**ANS:** Live Preview | Github