INTERVIEW QUESTION & ANSWER

CSS (CASCADING STYLE SHEETS)

1)What is the Box model in CSS? Which CSS properties are a part of it?

Ans: Box Model in CSS: The box model in CSS describes how elements are rendered on a web page. It consists of the content area, padding, border, and margin of an element. CSS properties that are part of the box model include width, height, padding, border, and margin.

2) What is specificity?

Ans: Specificity: Specificity in CSS determines which styles are applied to an element when multiple conflicting styles are present. It is calculated based on the type of selector used (e.g., class, ID, element), where ID selectors have higher specificity than class selectors, which have higher specificity than element selectors.

3)How to align a block element inside another element? Give code example.

Ans: Aligning a Block Element Inside Another Element: To align a block element inside another element, you can use CSS flexbox or grid layout. Here's an example using flexbox:

index.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <div class="block"></div>
    </div>
</body>
</html>
```

style.css

```css
.container{
    border: 1px solid black;
    padding: 5px;
    margin: 5px;
    width: 200px;
    height: 200px;
    display: flex;
    justify-content: center;
    align-items: center;
}
.block{
    border: 2px solid black;
    background-color: red;
    width: 50px;
    height: 50px;
}
```

4)Difference between Static, Relative, Absolute and Fixed position?

Ans: Static: Elements are positioned according to the normal flow of the document. This is the default value.

Relative: Elements are positioned relative to their normal position. Using top, bottom, left, and right properties, you can adjust the position.

Absolute: Elements are removed from the normal document flow and positioned relative to the nearest positioned ancestor. If there's no positioned ancestor, it's positioned relative to the initial containing block.

Fixed: Elements are removed from the normal document flow and positioned relative to the viewport.

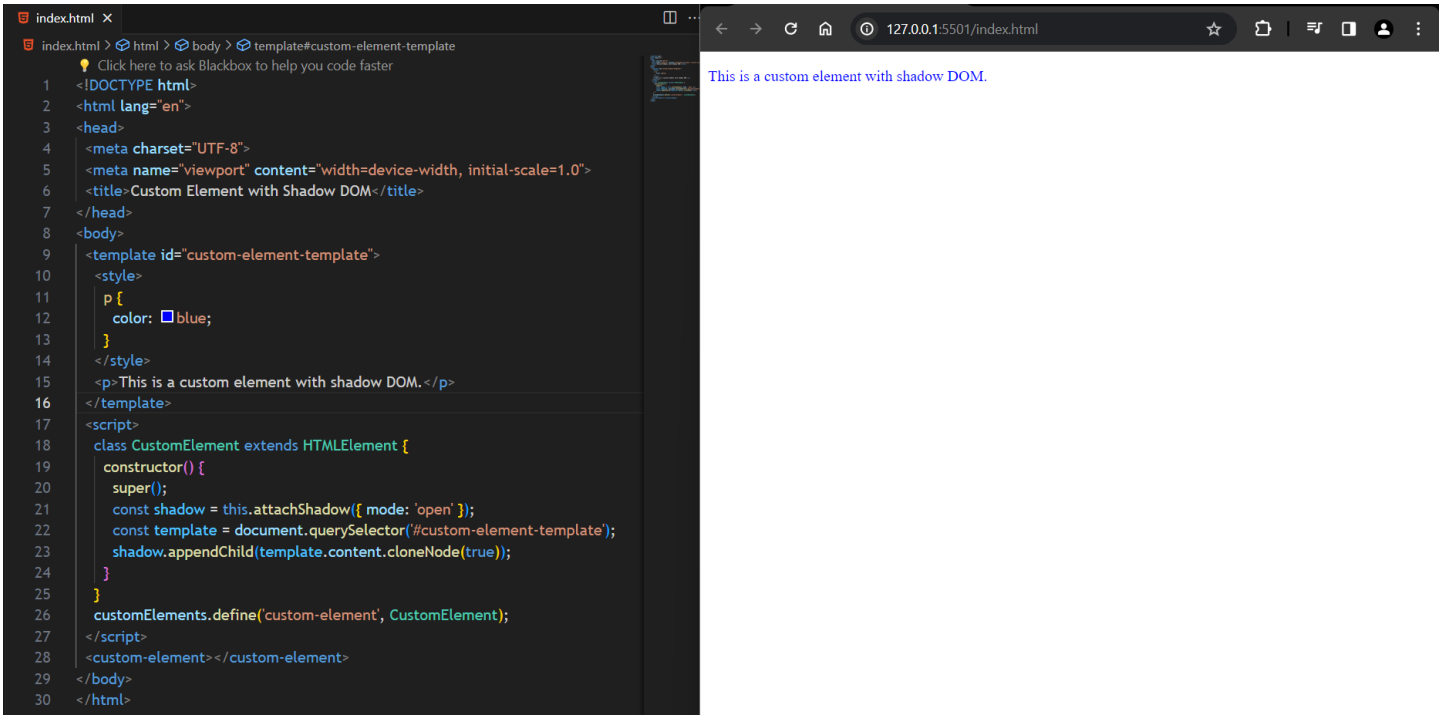
5)Difference between visibility: hidden; and display: none;

Ans: visibility: hidden hides an element but it still occupies space in the layout. The element is not visible, but space is reserved for it.

display: none hides an element and removes it from the layout entirely. The element is not visible and does not take up any space.

1) What is shadow DOM? Give an example.

Ans: Shadow DOM: The Shadow DOM is a way to encapsulate the styling and functionality of a web component, keeping it separate from the rest of the document. It allows for creating self-contained components that can be reused without worrying about style conflicts. Here's a basic example:



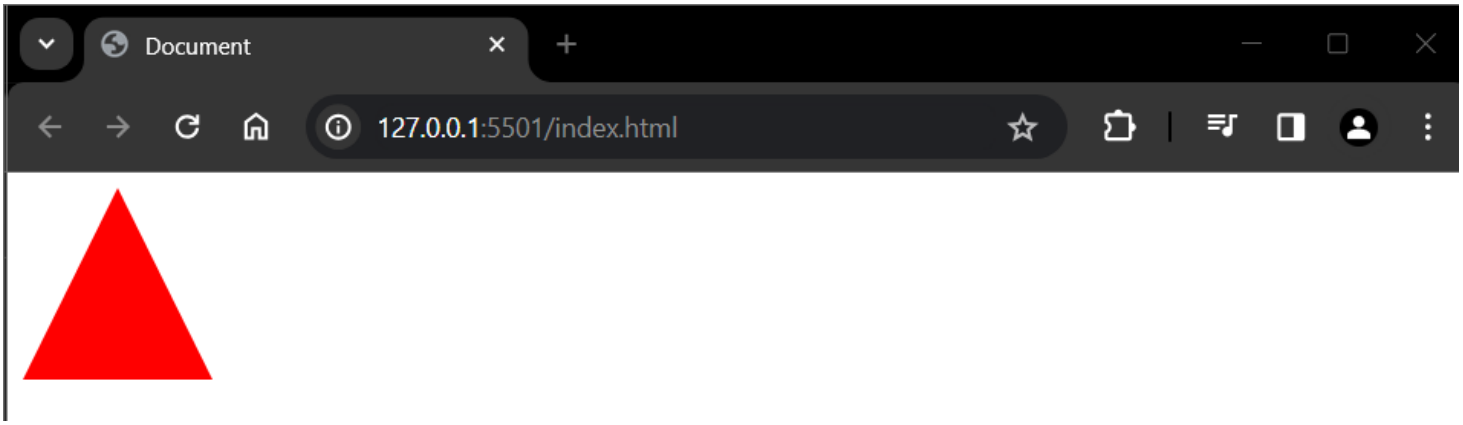
2) How to build a triangle in CSS? Give code example .

index.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="triangle"></div>
</body>
</html>
```

style.css

```css
.triangle {
    width: 0;
    height: 0;
    border-left: 50px solid transparent;
    border-right: 50px solid transparent;
    border-bottom: 100px solid red;
}
```



3) What are pseudo elements? Give Examples.

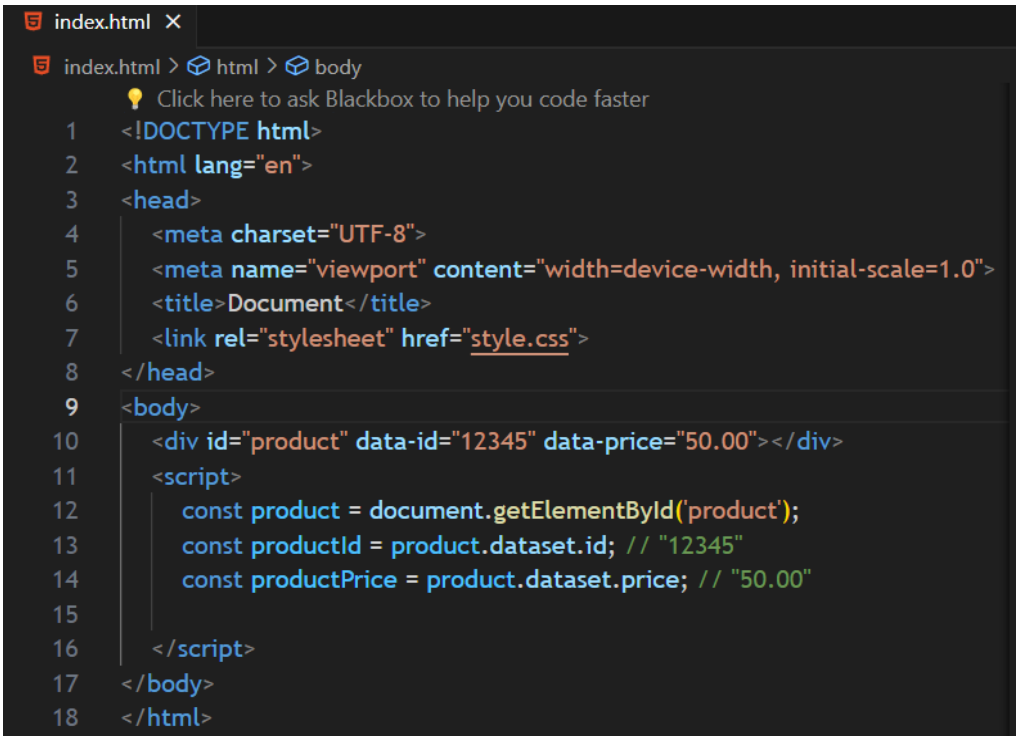
Ans: Pseudo-elements: Pseudo-elements are used to style certain parts of an element. They allow you to style elements based on their state or position in the document. Examples include ::before and ::after:

style.css

```css
p::before {
    content: 'Before';
    color: red;
}
p::after {
    content: 'After';
    color: blue;
}
```

4) What are Data attributes?

Ans: Data Attributes: Data attributes allow you to store extra information on an element that can be used by scripts or CSS. They are prefixed with data-. For example:



1. What Is zindex?

Ans: z-index is a CSS property that controls the stacking order of elements on a web page. It specifies the stack level of an element along the z-axis, which determines the order in which elements are layered on top of each other.

2. Why do we even need it? Where will you use it?

Ans: We need z-index to control the visual stacking order of elements, especially when dealing with elements that overlap. For example, if you have a dropdown menu that should appear above other content on the page, you can use z-index to ensure that it is displayed on top.

3. What is the best practice of using it?

Ans: Avoid using large values for z-index, as this can lead to unexpected behavior and make it difficult to manage the stacking order.

Use z-index sparingly and only when necessary, as excessive use can lead to a cluttered and confusing layout.

Use relative values for z-index where possible, such as z-index: 1; or z-index: -1;, to maintain a consistent stacking order across different elements.

Use z-index in conjunction with other CSS properties, such as position, to ensure that elements are positioned correctly within the layout.

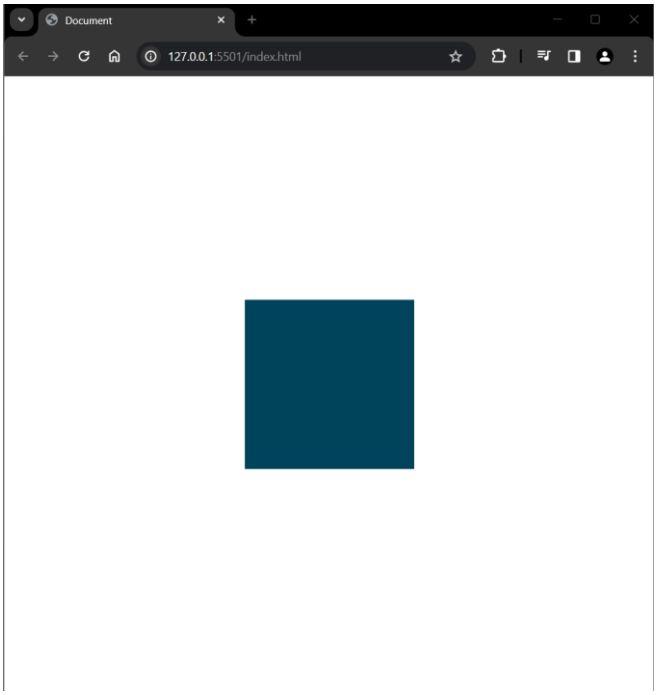
3) How can you align a div element in the center of the screen using flexbox?

index.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="centered-div"></div>
</body>
</html>
```

style.css

```css
body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
}
.centered-div {
    width: 200px;
    height: 200px;
    background-color: rgb(0, 68, 91);
}
```



4)What are the different properties of flex box. Explain each one of them?

display: flex;

Defines a flex container. Inline or block-level elements within this container become flex items.

flex-direction: row | row-reverse | column | column-reverse;

Specifies the direction of the main axis. Default is row.

justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly;

Aligns flex items along the main axis. Default is flex-start.

align-items: flex-start | flex-end | center | baseline | stretch;

Aligns flex items along the cross axis. Default is stretch.

flex-wrap: nowrap | wrap | wrap-reverse;

Controls wrapping of flex items when there is not enough space along the main axis. Default is nowrap.

align-content: flex-start | flex-end | center | space-between | space-around | stretch;

Aligns flex lines along the cross axis when there is extra space. Default is stretch.

order: <integer>;

Specifies the order of a flex item. Default is 0.

flex-grow: <number>;

Specifies how much a flex item will grow relative to the rest of the flex items. Default is 0.

flex-shrink: <number>;

Specifies how much a flex item will shrink relative to the rest of the flex items. Default is 1.

flex-basis: <length> | auto;

Specifies the initial size of a flex item. Default is auto.

1)What is VH/VW (viewport height/ viewport width) in CSS?

Ans: vh and vw are units in CSS that represent a percentage of the viewport height and viewport width, respectively. They are useful for creating designs that are responsive to the size of the viewport.

2)What property is used for changing the font face?

Ans: The font-family property is used to change the font face of text. You can specify multiple font names as a fallback in case the browser does not support the first choice.

3)When should we use float property in CSS?

Ans: The float property in CSS is used to align elements horizontally, allowing them to float to the left or right within their containing element. It is commonly used for creating layouts where elements flow around each other, such as in a multi-column design.

4) What do the following CSS selectors mean? “div, p”, “div p”, “div ~ p”, “div + p”, “div > p”

div, p: This selector targets both and <p> elements.

div p: This selector targets all <p> elements that are descendants of an element.

div ~ p: This selector targets all <p> elements that are siblings of a element.

div + p: This selector targets the <p> element that is immediately preceded by a element.

div > p: This selector targets all <p> elements that are direct children of a element.

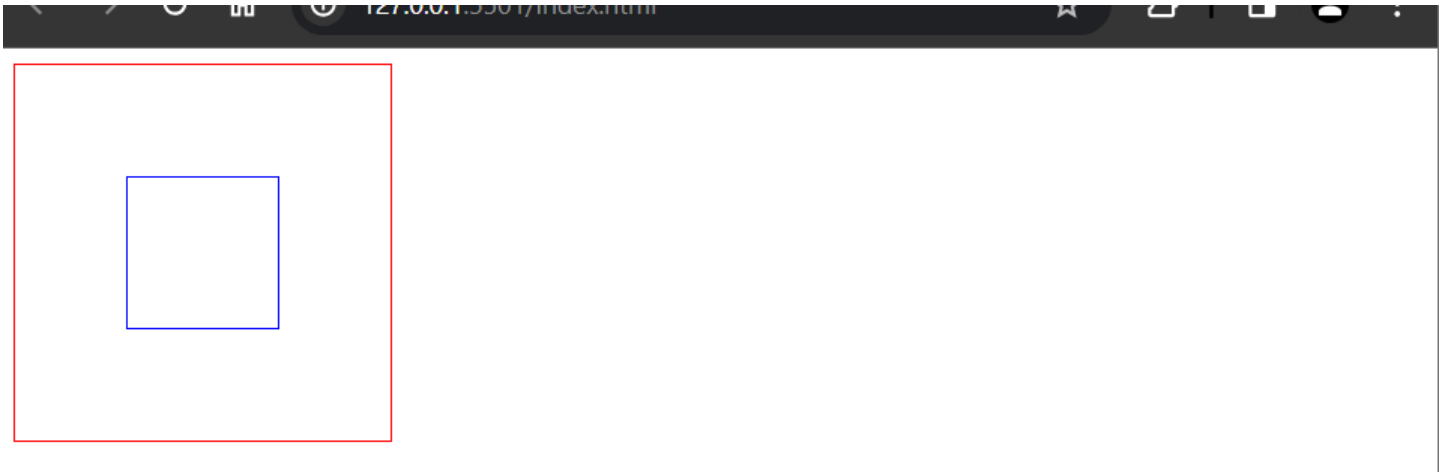
5)How to center align a div inside another div? Write the code by yourself

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="outer">
        <div class="inner">
        </div>
    </div>
</body>
</html>
```

style.css

```
.outer {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 200px;
    width: 200px;
    border: 1px solid red;
}
.inner {
    height: 80px;
    width: 80px;
    border: 1px solid blue;
}
```



6)What does "!important" mean in CSS?

Ans: !important is a CSS declaration that gives a rule higher priority than normal. It overrides any other declarations of the same property in the same rule set or in inherited rules. It should be used sparingly, as it can make CSS harder to maintain and debug.

7)What is the difference between CSS2 and CSS3?

Ans: CSS2 and CSS3 are different versions of the Cascading Style Sheets language. CSS3 is the latest version and includes many new features and enhancements over CSS2, such as new selectors, properties, and modules for advanced styling and layout options.