z-index is a CSS property that specifies the stack order of an HTML element relative to other elements on the page. Elements with a higher z-index value will appear in front of elements with a lower value.

Here is an example of how to use z-index in CSS:

<div class="box1"></div>

<div class="box2"></div>

<style>

  .box1 {

    width: 100px;

    height: 100px;

    background-color: red;

    position: absolute;

    z-index: 2;

  }

  .box2 {

    width: 150px;

    height: 150px;

    background-color: blue;

    position: absolute;

    z-index: 1;

  }

  </style>

In this example, the red box will appear in front of the blue box because it has a higher z-index value.

## There are several type of selectors in csss

**Class selectors:** Selects all elements with a particular class attribute.

Remember in class selectors we target the elemebt by use of(.)

Eg <div class=”box1”></div>

.box3{

        background-color: brown;

        background-size: cover;

        background-position: center;

        width: 100%;

        height: 100px;

}

In Id Class selectors its serves as the same as class selectors ,the difference is that’s in id we use (#) when targeting an element

<div id=”box1”></div>

#box3{

        background-color: brown;

        background-size: cover;

        background-position: center;

        width: 100%;

        height: 100px;

}

We also have Pseudo-class Selector: Selects elements based on a state or action, such as :hover or :focus.

Eg #header :hover{

background-color: aqua;

color: blueviolet;

    }

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

**vh** (viewport height) and **vw** (viewport width) are CSS units that allow you to size elements relative to the size of the browser viewport.

**1vh** is equal to 1% of the height of the viewport, while **1vw** is equal to 1% of the width of the viewport.

Here is an example

<div class=”box5></di>

<style>

   .box5{

    width: 50vw;

    height: 50vh;

    background-color: blue;

   }

</style>

In this example, the **.box5** element will be sized to 50% of the width and 50% of the height of the viewport, regardless of the size of the browser window.

What is the difference between inline, inline-block, and block?

**inline** elements are best used for small elements within text

An **inline** element takes up only as much width as necessary to display its content and flows within the surrounding text. **inline** elements cannot have a width or height set.eg

<span>This is an inline element.</span>

**inline-block**: An **inline-block** element is similar to an **inline** element, but it can have a width and height set, and it does not flow within the surrounding text.

 .box6{

            display: inline-block;

            width: 100px;

            height: 50px;

            background-color: greenyellow;

        }

**block**: A **block** element takes up the full width of its container and starts on a new line. **block** elements can have a width and height set.

  .box7{

            display: block;

            width: 100px;

            height: 50px;

            background-color: green;

        }

In summary, the main differences between **inline**, **inline-block**, and **block** are their width and height behavior, and how they flow within the surrounding content. **inline** elements are best used for small elements within text, while **inline-block** and **block** elements are better for larger elements or elements that need to be positioned or sized more precisely.

How do you specify units in the CSS? What are the different ways to do it?

In CSS, you can specify units using a variety of measurement values. Here are some of the most commonly used units and how to use them:Pixels (**px**): A fixed unit of measurement. 1px is equivalent to one screen pixel.

h1 {

font-size: 24px;

}

Percentages (**%**): A relative unit of measurement. Percentages are calculated based on the size of the element's containing block.

.container {

width: 80%;

margin: 0 auto;

}

How is the border-box different from the content box? The **border-box** box model sets the width and height of an element to include the content, padding, and border, but not the margin. In other words, the size of an element is calculated based on the size of the entire box, including its content area, padding, and border.

 .box8{

            width: 200px;

  height: 100px;

  padding: 10px;

  border: 2px solid black;

  margin: 20px;

  box-sizing: border-box;

        }

**content-box** (default): The **content-box** box model sets the width and height of an element to only include the content, but not the padding, border, or margin. In other words, the size of an element is calculated based on its content area only.

 .box9{

        width: 200px;

  height: 100px;

  padding: 10px;

  border: 2px solid black;

  margin: 20px;

  box-sizing: content-box;

    }

Using **border-box** can make it easier to create layouts and size elements, because the size of an element is based on the entire box, rather than just its content area.

How is opacity specified in CSS3?

In CSS, opacity is a property that allows you to adjust the transparency of an element. It can be specified using the **opacity** property, which takes a value between 0 and 1. A value of 0 makes the element completely transparent, while a value of 1 makes the element completely opaque.

#box4{

        background-color: black;

        width: 100%;

        height: 100px;

        background-size: cover;

        background-position: center;

        opacity: 0.5;

    }

Differentiate between absolute and relative in CSS.

Relative positioning moves an element relative to its current position, without affecting the position of other elements on the page

Absolute positioning, on the other hand, removes an element from the normal document flow and positions it relative to its nearest positioned ancestor (if any) or to the viewpor