Q.1 What's Box Model in CSS ?

ANS: The **CSS box model** is a container that contains multiple properties including borders, margins, padding, and the content itself. It is used to create the design and layout of web pages. It can be used as a toolkit for customizing the layout of different elements. The web browser renders every element as a rectangular box according to the CSS box model

-PADDING
-MARGIN
-BORDER
-CONTENT

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

ANS: **CSS selectors** are used *to select the content you want to style*. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc.

There are several different types of selectors in CSS.

1. CSS Element Selector
2. CSS Id Selector
3. CSS Class Selector
4. CSS Universal Selector
5. CSS Group Selector

Advantages of Using CSS Selector

It's faster than XPath.
It's much easier to learn and implement.
You have a high chance of finding your elements.
It's compatible with most browsers to date.

Q.3 What is VW/VH ?

ANS: Viewport Height (vh). This unit is based on the height of the viewport. A value of `1vh` is equal to 1% of the viewport height.

- **Viewport Width (vw). This unit is based on the width of the viewport. A value of `1vw` is equal to 1% of the viewport width.**

Q-4 Q.4 Whats difference between Inline, Inline Block and block ?

ANS: display:inline

When an element is styled with `display:inline`, it will not start on a new line, will only take up as much width as the content it contains, and will not cause a line break after it.

Display:inline-block

The difference between an inline element and an inline-block element is that an inline-block element can take up specified width and height. But, it will also not start on a new line within its parent or cause a line break after it.

display:block

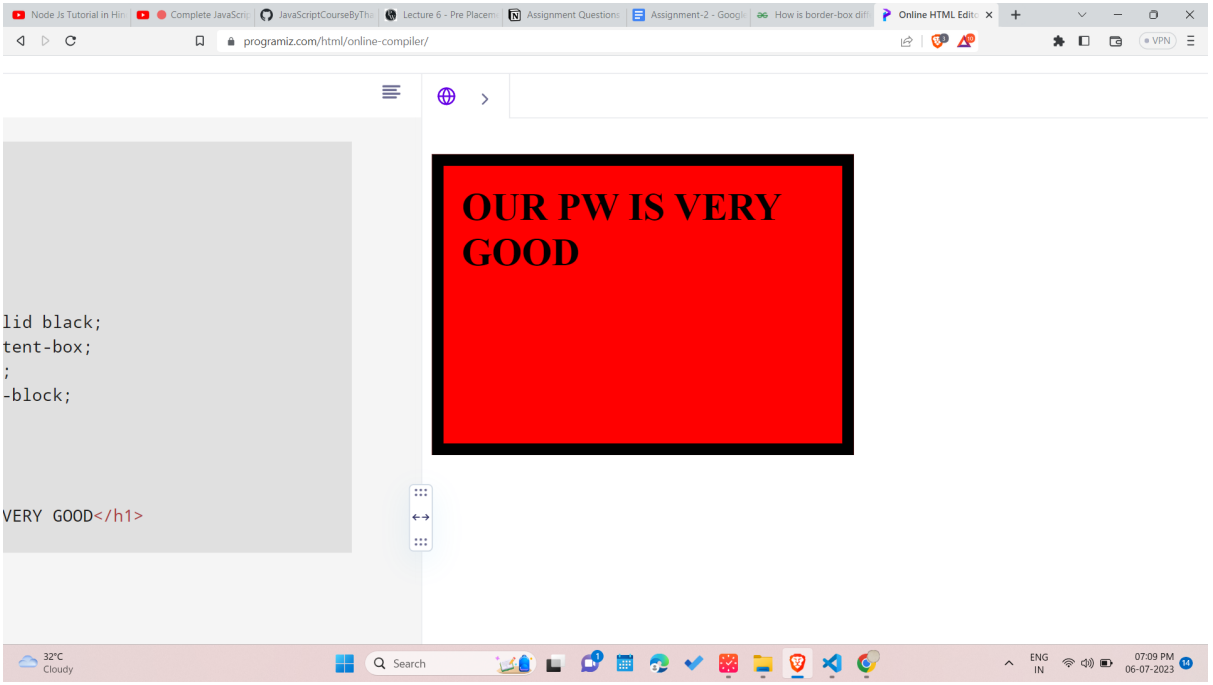
Any element styled with `display: block` is the polar opposite of `display:inline`. A block element starts on a new line and occupies the available width of its parent element or its specified width.

Q.5 How is Border-box different from Content Box?

ANS: content-box: This is the default value of box-sizing. The dimension of element only includes ‘height’ and ‘width’ and does not include ‘border’ and ‘padding’ given to element. Padding and Border take space outside the element.

CODE:<html>

```
    <head>
        <style>
            .box {
                width: 300px;
                height: 200px;
                padding: 15px;
                border: 10px solid black;
                box-sizing: content-box;
                background: red;
                display: inline-block;
            }
        </style>
    </head>
    <body>
        <h1 class="box">OUR PW IS  VERY GOOD</h1>
    </body>
</html>
```



border-box: In this value, not only width and height properties are included but you will find padding and border inside of the box for example .box {width: 200px; border: 10px solid black;} renders a box that is 200px wide.

CODE:

```
<html>

<head>

 <style>

  .box {

    width: 200px;

         height: 200px;
```

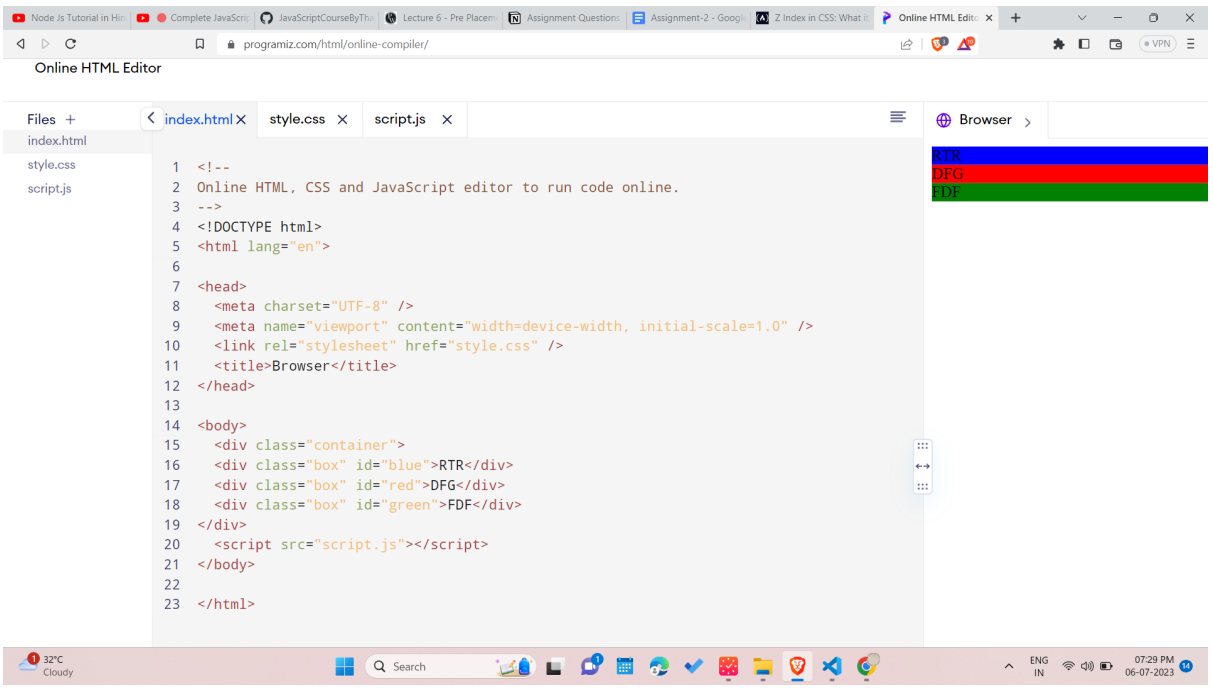
```
        padding: 15px;
        border: 10px solid green;
        box-sizing: border-box;
        background: red;
        display: inline-block;
       }
 </style>
</head>
<body>
 <h1 class="box">PW IS GREAT</h1>
</body>
</html>
```

Q.6 What's z-index and How does it Function ?

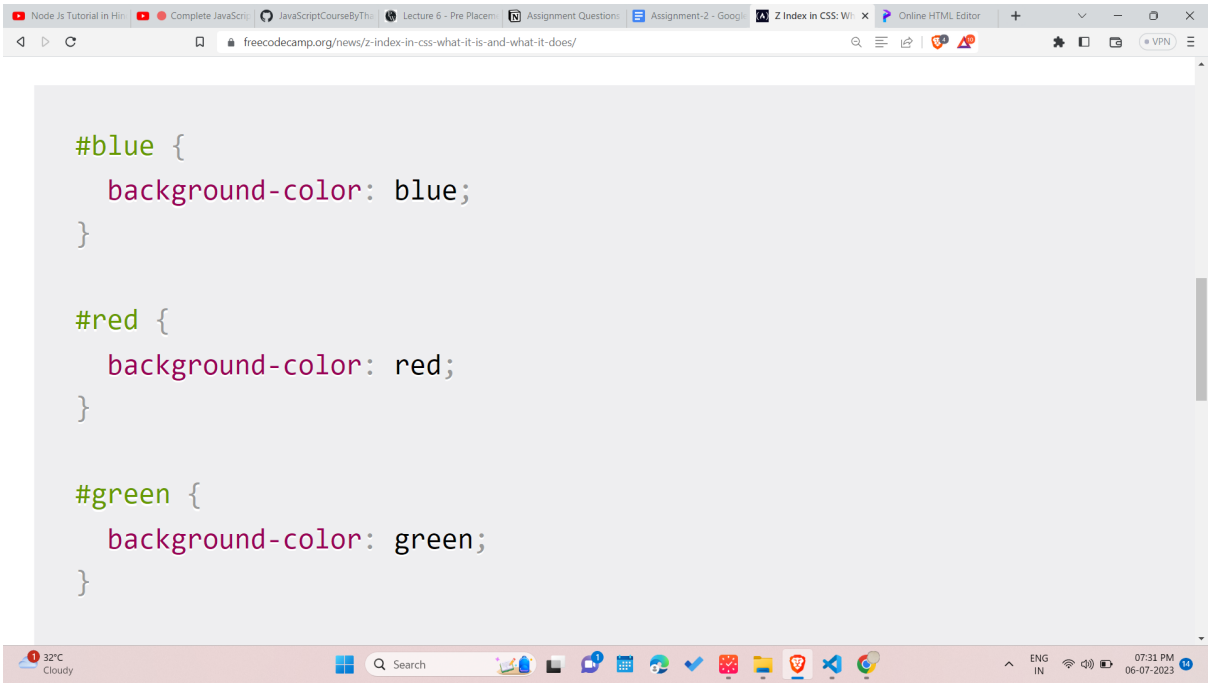
ANS: Z- Index is a CSS property that defines the order of overlapping HTML elements. Elements with a higher index will be placed on top of elements with a lower index.

How to use the Z Index

HTML



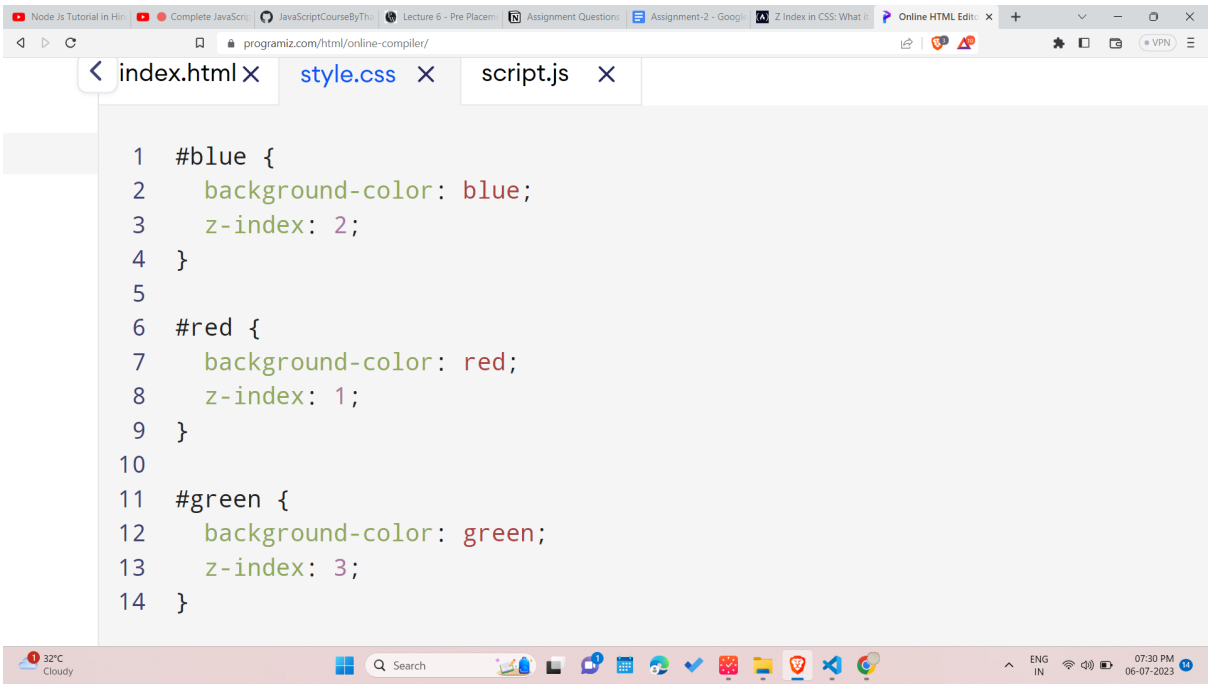
CSS



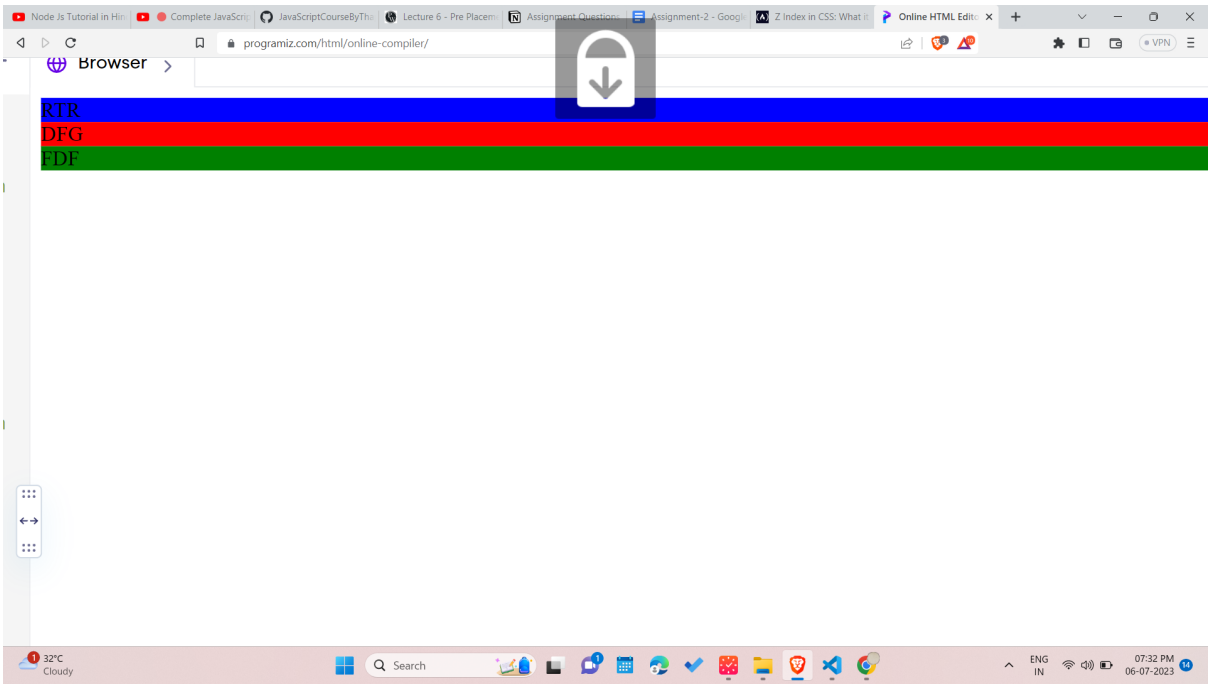
`z-index` wasn't defined, it will have a default value of `auto`.

This is a result:

Try to change the order to Green, Blue, Red in CSS using `z-index`.

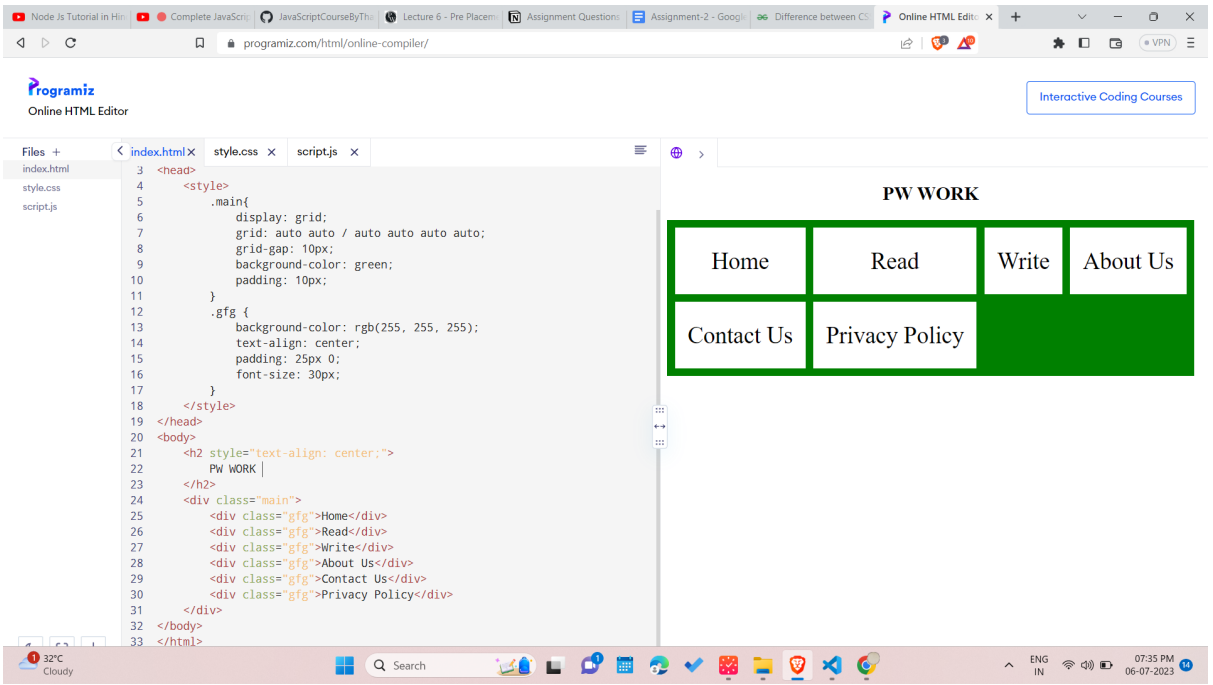


OUTPUT:

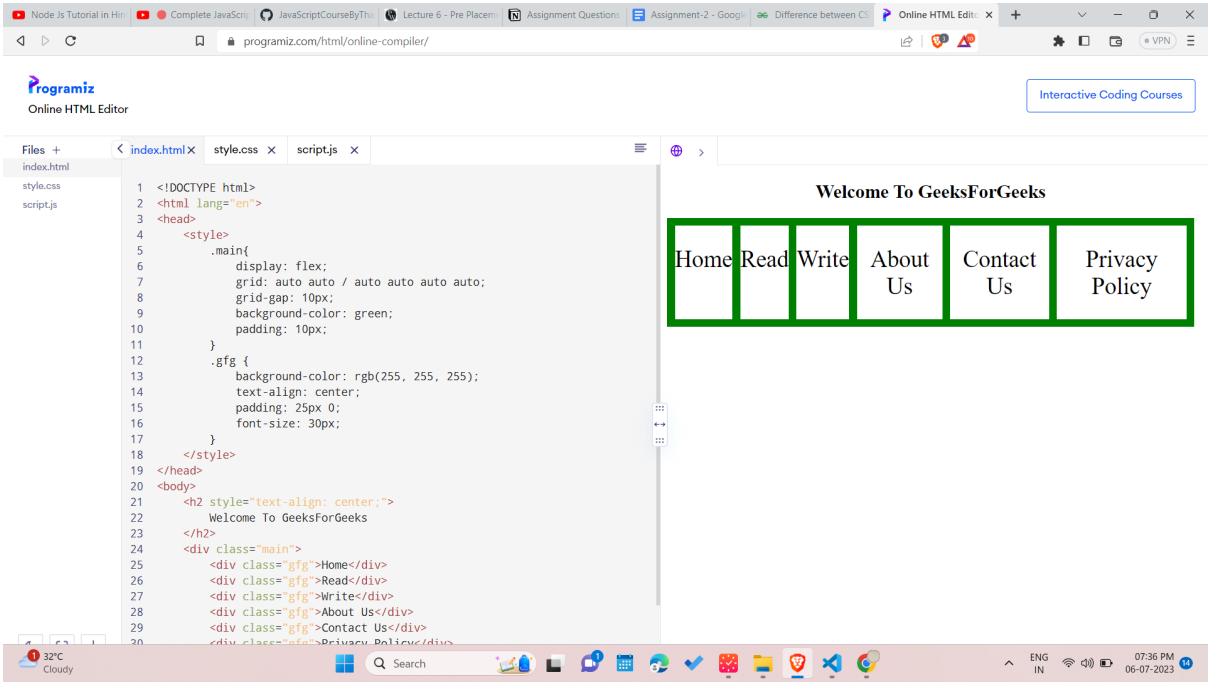


Q-7: What's Grid & Flex and difference between them?

ANS: Grid:CSS Grid Layout, is a two-dimensional grid-based layout system with rows and columns, making it easier to design web pages without having to use floats and positioning. Like tables, grid layout allow us to align elements into columns and rows.



Flexbox: The CSS Flexbox offers a one-dimensional layout. It is helpful in allocating and aligning the space among items in a container (made of grids). It works with all kinds of display devices and screen sizes.

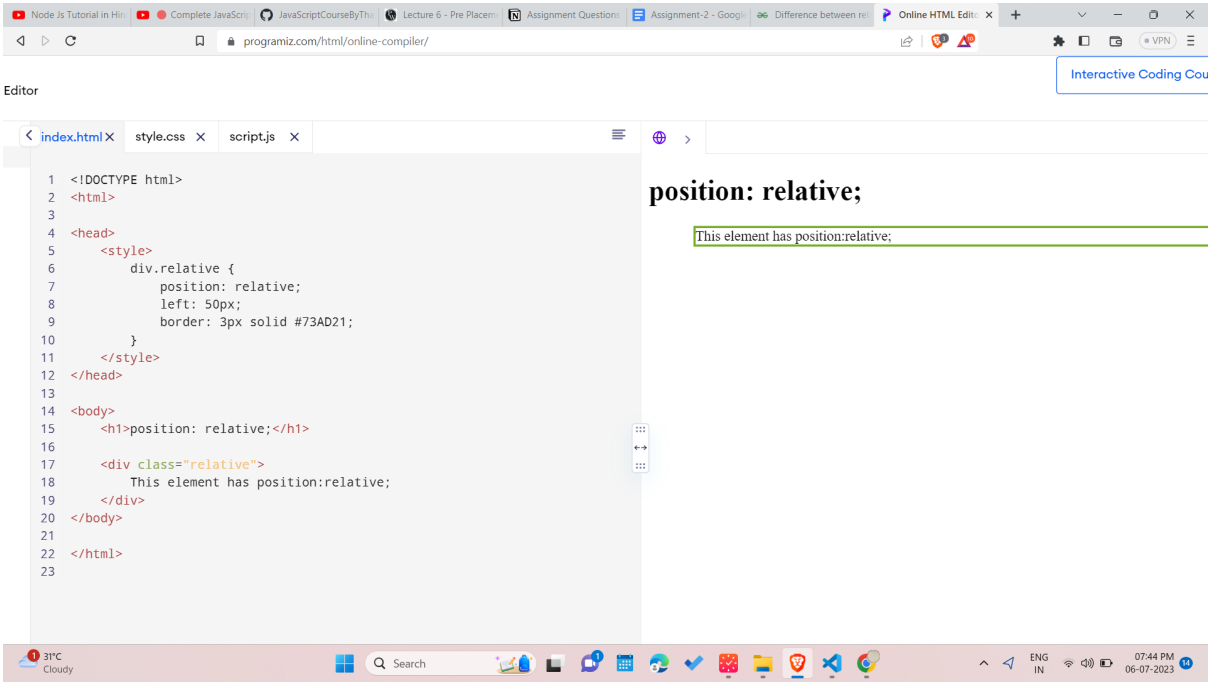


Q.8 Difference between absolute and relative and sticky and fixed position explain with example.

ANS:

1[Relative Position:](#) **Setting the top, right, bottom, and left properties of an element with *position: relative;* property will cause it to adjust from its normal position. The other objects or elements will not fill the gap.**

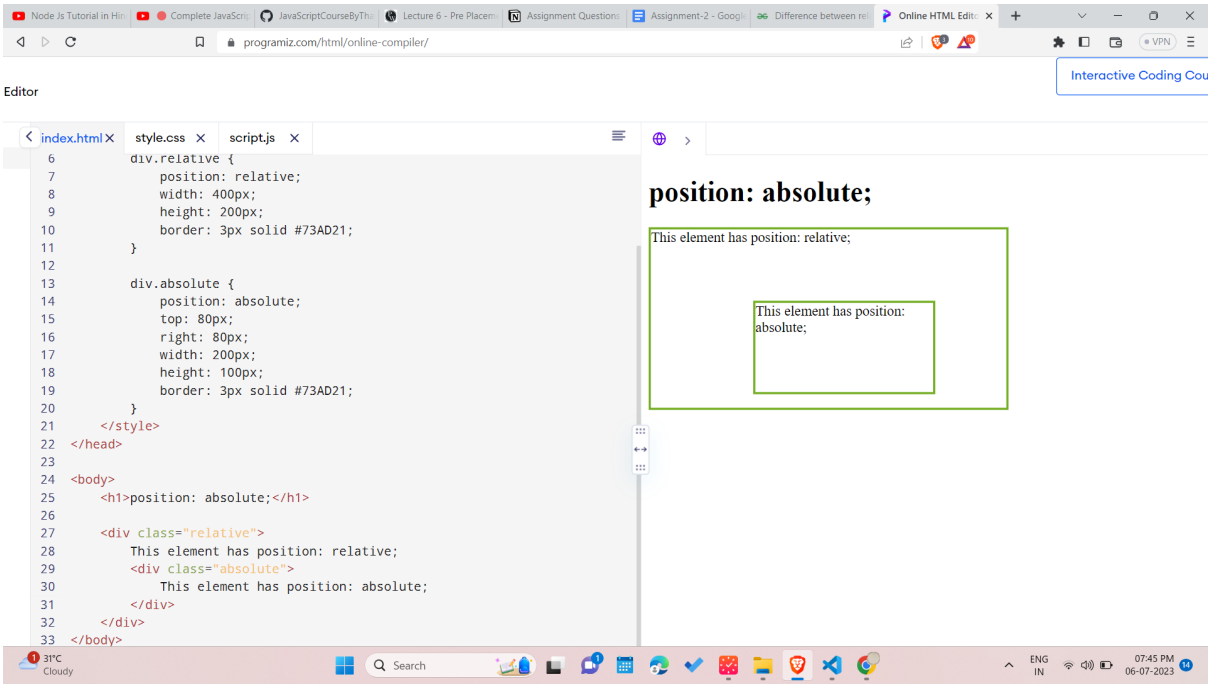
```
position: relative;
```




```
<!DOCTYPE html>
<html>

<head>
    <style>
        div.relative {
            position: relative;
            left: 50px;
            border: 3px solid #73AD21;
        }
    </style>
</head>

<body>
    <h1>position: relative;</h1>

    <div class="relative">
        This element has position:relative;
    </div>
</body>

</html>
```

Absolute Position: An element with *position: absolute;* will cause it to adjust its position with respect to its parent. If no parent is present, then it uses the document body as parent.

```
position: absolute;
```




```
        div.relative {
            position: relative;
            width: 400px;
            height: 200px;
            border: 3px solid #73AD21;
        }

        div.absolute {
            position: absolute;
            top: 80px;
            right: 80px;
            width: 200px;
            height: 100px;
            border: 3px solid #73AD21;
        }
    </style>
</head>

<body>
    <h1>position: absolute;</h1>

    <div class="relative">
        This element has position: relative;
        <div class="absolute">
            This element has position: absolute;
        </div>
    </div>
</body>
```

Fixed Position: Position: fixed; property applied to an element will cause it to always stay in the same place even if the page is scrolled. To position the element we use top, right, bottom, left properties.

```
position: fixed;
```