Name: Sownthari R P

Date: 06.08.2024

Part 1: CSS Positioning

Objective: Create a web page demonstrating different CSS positioning techniques.

Instructions:

1. **Create an HTML file named index.html.**
2. **Add a div element with the class container and three child div elements with classes absolute, relative, and fixed.**
3. **Style the container to have a width of 500px and height of 300px.**
4. **Apply different positioning styles to each child div.**

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSS Positioning</title>
    <style>
        .container {
            width: 500px;
            height: 300px;
            border: 1px solid black;
            position: relative;
        }

        .absolute {
            position: absolute;
            top: 20px;
```

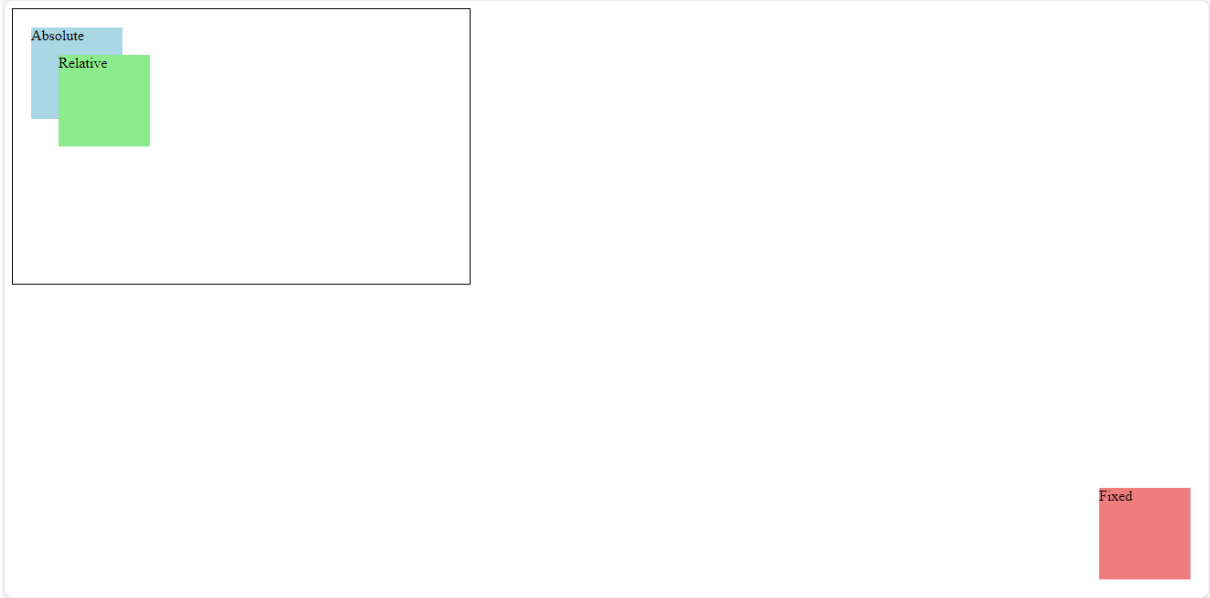
```
            left: 20px;
            width: 100px;
            height: 100px;
            background-color: lightblue;
        }

        .relative {
            position: relative;
            top: 50px;
            left: 50px;
            width: 100px;
            height: 100px;
            background-color: lightgreen;
        }

        .fixed {
            position: fixed;
            bottom: 20px;
            right: 20px;
            width: 100px;
            height: 100px;
            background-color: lightcoral;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="absolute">Absolute</div>
        <div class="relative">Relative</div>
        <div class="fixed">Fixed</div>
    </div>
```

```
</body>
</html>
```

Result:



2. Try changing the width and give only 10px to border property. Mention what changes you have noticed with the content.

Index.html

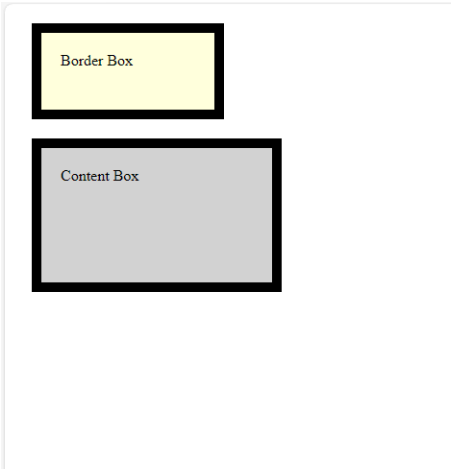
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Box Sizing</title>
    <style>
        .border-box, .content-box {
            width: 200px;
            height: 100px;
            margin: 20px;
```

```
            padding: 20px;
            border: 10px solid black;
        }

        .border-box {
            box-sizing: border-box;
            background-color: lightyellow;
        }

        .content-box {
            box-sizing: content-box;
            background-color: lightgray;
        }
    </style>
</head>
<body>
    <div class="border-box">Border Box</div>
    <div class="content-box">Content Box</div>
</body>
</html>
```

Result:



Observation:

- The .border-box element includes the padding and border in the specified width and height, so the content area is smaller.
- The .content-box element does not include the padding and border in the specified width and height, so the content area is larger.

3.Javascript – show difference between substr and substring with negative index and positive index for the string "The world is wonderful".

```
const str = "The world is wonderful";

        console.log("Using substr:");
        console.log(str.substr(4, 5));
        console.log(str.substr(-9, 5));

        console.log("Using substring:");
        console.log(str.substring(4, 9));
        console.log(str.substring(-9, 5));
```

Result:

```
Using substr:
world
wonde
Using substring:
world
The w
```

4.Show what's inline, internal and external scripts.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Inline, Internal, and External Scripts</title>
    <script>
        // Internal script
        console.log("This is an internal script.");
    </script>
</head>
<body>
    <script>
        // Inline script
        console.log("This is an inline script.");
    </script>
    <script src="external.js"></script>
</body>
</html>
```

External script - script.js

```
console.log("This is an external script.");
```

5.As per naming convention, which variable is advisable to use for functions or arrays: const or let or var?

```
const myArray = [1, 2, 3];
```

```
const myFunction = () => {
    console.log("Hello, world!");
};
let count = 0;
count++;
var oldVar = "This is an old variable.";
oldVar = "Changed value.";
```

It is advisable to use const for declaring arrays and functions.