

**Name:Siddarth S**

**Date: 06.08.2024**

## **Part 1: CSS Positioning**

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

**Instructions:**

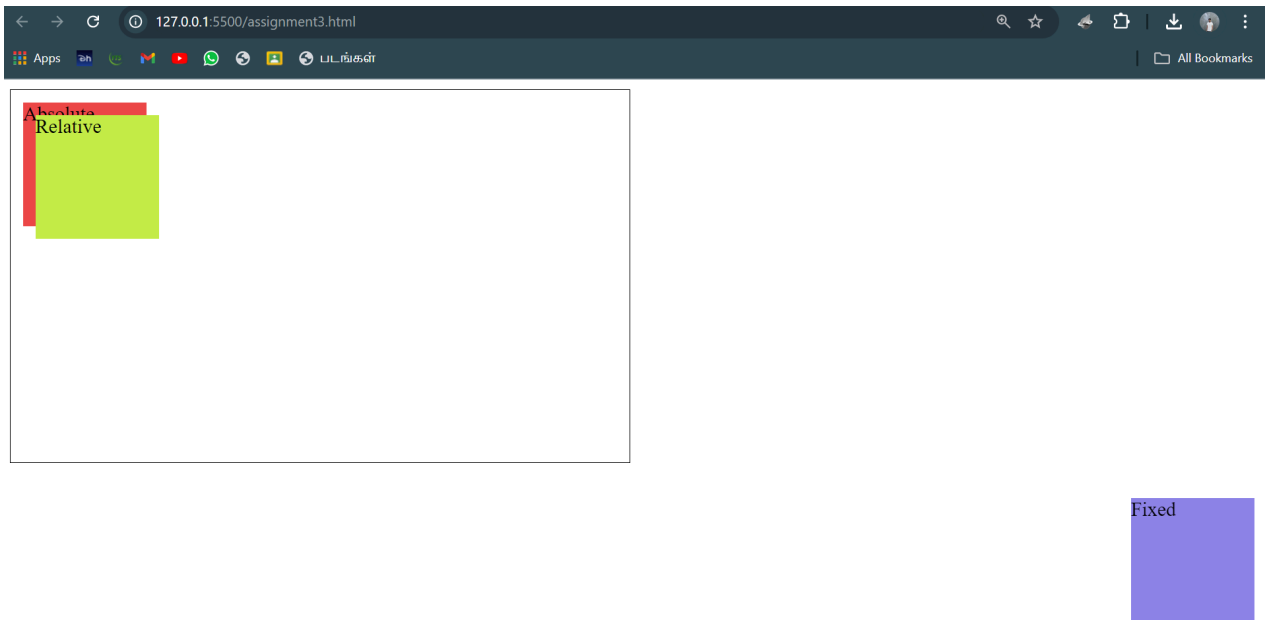
- 1. Create an HTML file named index.html.**
- 2. Add a div element with the class container and three child div elements with classes absolute, relative, and fixed.**
- 3. Style the container to have a width of 500px and height of 300px.**
- 4. Apply different positioning styles to each child div.**

### **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;
      margin-bottom: 20px;
    }
    .absolute {
      position: absolute;
      top: 10px;
      left: 10px;
      background-color: rgb(236, 71, 71);
      width: 100px;
```

```
        height: 100px;
    }
    .relative {
        position: relative;
        top: 20px;
        left: 20px;
        background-color: rgb(195, 236, 74);
        width: 100px;
        height: 100px;
    }
    .fixed {
        position: fixed;
        bottom: 10px;
        right: 10px;
        background-color: rgb(144, 132, 234);
        width: 100px;
        height: 100px;
    }
</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;
```

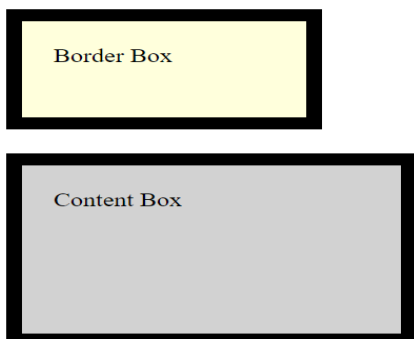
```
padding: 20px;
border: 10px solid black;
}

.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:

- **Border-box:** The total width and height include the padding and border. The element maintains a fixed size.
- **Content-box:** The width and height apply only to the content. The padding and border increase the element's total size.

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

Code and Output:

```
const sentence = "The world is wonderful";  
  
console.log(sentence.substr(4, 6));  
console.log(sentence.substr(-9, 9));  
console.log(sentence.substring(4, 10));  
console.log(sentence.substring(-10, 6));  
world  
wonderful  
world  
The wo  
undefined
```

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

#### Inline Script

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width,  
initial-scale=1.0">  
  <title>Inline Script</title>
```

```
</head>
<body>
    <button onclick="alert('Hello Universe')">Click me</button>
</body>
</html>
```

### Internal Script

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Internal Script</title>
    <script>
        function showAlert() {
            alert('Hello Universe');
        }
    </script>
</head>
<body>
    <button onclick="showAlert()">Click me</button>
</body>
</html>
```

### External Script

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>External Script</title>
    <script src="external.js"></script>
</head>
<body>
    <button onclick="showAlert()">Click me</button>
</body>
</html>
```

**external.js:**

```
function showAlert() {  
    alert('Hello Universe');  
}
```

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

For functions or arrays, it is advisable to use const.

### **Example**

```
const numbers = [1, 2, 3, 4];  
const calculateSum = function(arr) {  
    return arr.reduce((sum, num) => sum + num, 0);  
};
```

```
let result = calculateSum(numbers);  
console.log(result); // 10
```

Using const ensures that the reference to the array or function cannot be reassigned, providing more predictable and maintainable code.