

**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`.

**CODE:****HTML:**

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Positioning Example</title>

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <div class="container">

    <div class="absolute">Absolute</div>

    <div class="relative">Relative</div>

    <div class="fixed">Fixed</div>

  </div>
```

```
</body>
```

```
</html>
```

## **CSS:**

```
.container {
```

```
    width: 500px;
```

```
    height: 300px;
```

```
    border: 1px solid #000;
```

```
    position: relative;
```

```
}
```

```
.absolute {
```

```
    position: absolute;
```

```
    top: 20px;
```

```
    left: 20px;
```

```
    width: 100px;
```

```
    height: 100px;
```

```
    background-color: lightblue;
```

```
}
```

```
.relative {
```

```
    position: relative;
```

```
    top: 40px;
```

```
    left: 40px;
```

```
    width: 100px;
```

```
    height: 100px;
```

```
    background-color: lightgreen;
```

```
}
```

```
.fixed {
```

```
    position: fixed;
```

```
    top: 60px;
```

```
    left: 60px;
```

```
    width: 100px;
```

```
    height: 100px;
```

```
    background-color: lightcoral;
```

```
}
```

## OUTPUT:



**Part 2: Try changing the width and give only 10px to border property. Mention what changes you have noticed with the content. Hint: Create a html with div containers and classes accordingly.**

```
.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;
```

```
}
```

## QUERY:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Box Sizing Example</title>
```

```
<link rel="stylesheet" href="style2.css">
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<div class="border-box">Border Box</div>
```

```
<div class="content-box">Content Box</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

## CSS:

```
.container {
```

```
    display: flex;
```

```
    flex-direction: column;
```

```
    align-items: center;
```

```
}
```

```
.border-box, .content-box {
```

```
    width: 200px;
```

```
    height: 100px;
```

```
    margin: 20px;
```

```
    padding: 20px;
```

```
    border: 10px solid black;
```

```
}
```

```
.border-box {
```

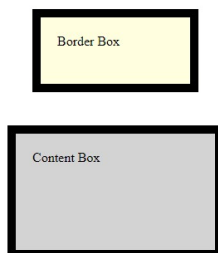
```
    box-sizing: border-box;
```

```
    background-color: lightyellow;
```

```
}
```

```
.content-box {  
  
    box-sizing: content-box;  
  
    background-color: lightgray;  
  
}
```

### OUTPUT:



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

### QUERY:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

<title>substr vs substring</title>

</head>

<body>

<h1>JavaScript substr vs substring</h1>

<p id="demo"></p>

<script>

let str = "The world is wonderful";

let substrPos = str.substr(4, 5);

let substrNeg = str.substr(-9, 5);

let substringPos = str.substring(4, 9);

let substringNeg = str.substring(-9, 5);

document.getElementById("demo").innerHTML = `

<strong>Original string:</strong> "\${str}"<br><br>

<strong>substr with positive index (4, 5):</strong> "\${substrPos}"<br>

<strong>substr with negative index (-9, 5):</strong> "\${substrNeg}"<br><br>

<strong>substring with positive index (4, 9):</strong> "\${substringPos}"<br>

<strong>substring with negative index (-9, 5):</strong> "\${substringNeg}"

`;



</script>

</body>

</html>

## OUTPUT:

### JavaScript substr vs substring

Original string: "The world is wonderful"

substr with positive index (4, 5): "world"

substr with negative index (-9, 5): "wonde"

substring with positive index (4, 9): "world"

substring with negative index (-9, 5): "The w"

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

## QUERY:

<!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>

<!-- Internal Script -->

<script>

```
function showInternalMessage() {
```

```
    alert("This is an internal script!");
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<h1>JavaScript Examples</h1>
```

```
<!-- Inline Script -->
```

```
<button onclick="alert('This is an inline script!')">Inline Script</button>
```

```
<br><br>
```

```
<!-- Internal Script Button -->
```

```
<button onclick="showInternalMessage()">Internal Script</button>
```

```
<br><br>
```

```
<!-- External Script Button -->
```

```
<button onclick="showExternalMessage()">External Script</button>
```

```
<!-- External Script -->
```

```
<script src="script.js"></script>
```

```
</body>
```

```
</html>
```

EXTERNAL:

```
<script>
```

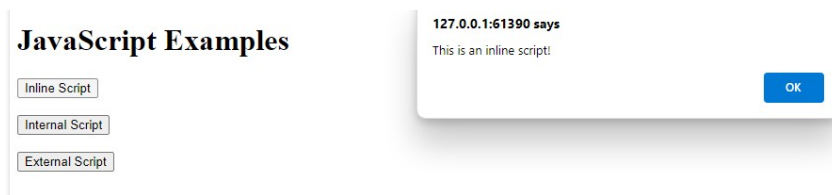
```
function showInternalMessage() {
```

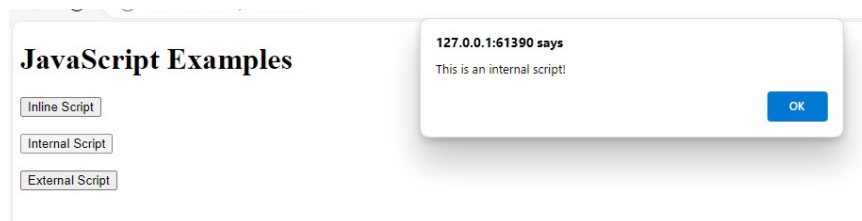
```
    alert("This is an internal script!");
```

```
}
```

```
</script>
```

**OUTPUT:**





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

**ANSWER:**

Const