TASK2:

**Debugging exercise 1:** JavaScript Debugging

**Problem Statement:**

You've been given a simple JavaScript code snippet that's intended to toggle the visibility of an element when a button is clicked. However, it's not working as expected.

Code:  
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Toggle Element</title>

</head>

<body>

    <button onclick="toggleElement()">Toggle Element</button>

    <div id="target" style="display: none;">This is the target element.</div>

    <script>

        function toggleElement() {

            var element = document.getElementById("target");

            element.style.display = (element.style.display === "none") ? "block" : "none";

        }

    </script>

</body>

</html>

**Solution:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Toggle Element</title>

</head>

<body>

<button onclick="toggleElement()">Toggle Element</button>

<div id="target" style="display: none;">This is the target element.</div>

<script>

function toggleElement() {

var element = document.getElementById("target");

if (element.style.display === "none" || element.style.display === "") {

element.style.display = "block";

} else {

element.style.display = "none";

}

}

</script>

</body>

</html>

**Explanation:**

The problem with the provided JavaScript code is that it tries to toggle the visibility of an element based on its display property without taking into consideration the element's initial state. The #target div is initially set to display: none; when the page loads, but the JavaScript code doesn't consider this initial state. As a result, the code may not work as expected if the element is hidden by default.

We check whether the display property is set to none or is empty or hidden regardless of its initial state or any changes made to it, So when you click the button it will work correctly to show or hide the element, regardless of what was its initial value and hence it will toggle the visibility of an element.

**Debugging exercise 2:** CSS Troubleshooting

**Problem Statement:**

You've been given an HTML and CSS code snippet that's supposed to create a centered, responsive container. However, it's not displaying as expected.

Code:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Centered Container</title>

<style>

.container {

margin: auto;

width: 50%;

background-color: #f0f0f0;

padding: 20px;

}

</style>

</head>

<body>

<div class="container">

<h1>Centered Container</h1>

<p>This container should be centered on the page.</p>

</div>

</body>

</html>

Tasks:

Identify the issue in the provided CSS code.

Debug and fix the code so that the container is centered on the page.

**Solution:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Centered Container</title>

<style>

.container {

width: 50%;

margin: 0 auto;

background-color: #f0f0f0;

padding: 20px;

}

</style>

</head>

<body>

<div class="container">

<h1>Centered Container</h1>

<p>This container should be centered on the page.</p>

</div>

</body>

</html>

**Explanation:**

The issue with the above is that , it centred the container only horizontally at the page ,to centre it vertically, we need to set the top and bottom of the margin to **auto** i.e. **margin:0 auto;** This way we can centre the container horizontally as well as vertically .Hence the container is centred on the page.

**Debugging exercise 3:** Debugging JavaScript Functions

**Problem Statement:**

Objective: Identify and fix issues in JavaScript functions.

This code snippet with a JavaScript function that performs a specific task, but contains bugs or inefficiencies.

Debug the function and ensure it works correctly and efficiently.

Code:

function calculateSum(arr) {

let sum = 0;

for (let i = 0; i < arr.length; i++) {

sum += arr[i];

}

return sum;

}

const numbers = [1, 2, 3, 4, 5];

const result = calculateSum(numbers);

console.log(result); // Should output 15

**Debugging exercise 4:** Debugging CSS Styling Issues

**Problem Statement:**

Objective: Identify and fix CSS styling issues to achieve the desired layout.

This code snippet with HTML and CSS code that creates a specific layout, but contains CSS issues like misalignment, overlapping elements, or incorrect colors.

Debug the CSS to achieve the desired layout.

Code:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Styling Debugging Exercise</title>

<style>

.container {

width: 50%;

margin: 0 auto;

background-color: #f0f0f0;

padding: 20px;

}

.box {

width: 100px;

height: 100px;

background-color: #007bff;

color: #ffffff;

text-align: center;

line-height: 100px;

}

</style>

</head>

<body>

<div class="container">

<div class="box">Box 1</div>

<div class="box">Box 2</div>

<div class="box">Box 3</div>

</div>

</body>

</html>

**Solution:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Styling Debugging Exercise</title>

<style>

.container {

width: 50%;

margin: 0 auto;

background-color: #f0f0f0;

padding: 20px;

text-align: center; /\* Center the text horizontally \*/

}

.box {

width: 100px;

height: 100px;

background-color: #007bff;

color: #ffffff;

text-align: center;

line-height: 100px; /\* Vertically center the text \*/

margin: 10px; /\* Add margin to separate the boxes \*/

display: inline-block; /\* Make the boxes appear side by side \*/

}

</style>

</head>

<body>

<div class="container">

<div class="box">Box 1</div>

<div class="box">Box 2</div>

<div class="box">Box 3</div>

</div>

</body>

</html>

**Explanation:**

* To centre the text inside the boxes vertically, you should set the line-height property to match the height of the boxes (100px). This ensures that the text is vertically centred within each box.
* To separate the boxes and make them appear side by side, you can add some margin or padding to the. box elements.