**Exercise 1: JavaScript Debugging**

Corrected 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" >This is the target element.</div>

    <script>

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

element.style.display = “none”;

        function toggleElement(){

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

        }

    </script>

</body>

</html>

Explanations:

When an element has **display: none;**, the **element.style.display** property won't work as expected because it refers to inline styles, and the element is initially hidden using a CSS rule. To handle this scenario, you can use the **window.getComputedStyle** method to get the computed styles of the element

**Exercise 2: CSS Troubleshooting**

Corrected 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>

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.container {

width: 50%;

background-color: #f0f0f0;

padding: 20px;

text-align: center;

}

</style>

</head>

<body>

<div class="container">

<h1>Centered Container</h1>

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

</div>

</body>

</html>

Explanations:

in the provided CSS code is that the container is set to have a margin of **auto**, which is intended to horizontally center the container. However, the **margin: auto;** rule works when the element has a specified width and the parent element has a specified width as well.

In this case, the container has a width of **50%**, but the parent element (the **<body>** element) doesn't have a specified width. To make the container centered on the page, you need to ensure that both the container and its parent have specified widths

<body> element is set to display: flex; with justify-content: center; and align-items: center;. This centers the .container within the body element. Additionally, the height: 100vh;

**Exercise 3: Debugging JavaScript Functions**

Corrected 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);

Explanation:

The provided JavaScript code appears to be correct, and it should output **15** as expected. The **calculateSum** function iterates over the **numbers** array and calculates the sum of its elements, which are **[1, 2, 3, 4, 5]**. When you run this code, it correctly calculates the sum of these numbers and logs the result to the console.

**Exercise 2: Debugging CSS Styling Issues**

Corrected 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>

Explanation:

The provided code aims to create a layout with three blue boxes inside a container. Based on the code you provided, there shouldn't be any misalignment, overlapping elements, or incorrect colors. The CSS rules you've written for the **.container** and **.box** classes seem correct.