Exercise 1: JavaScript

DebuggingProblem

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>
```

Tasks:

Identify the issue in the provided JavaScript code.
Debug and fix the code so that clicking the button toggles the visibility of the element.

Solution:

The JavaScript code you provided seems correct. There is a small issue, with how the display property is being toggled. Currently the code only checks if the display value is "none". Sets it to "block" in that case. However it doesn't account for values like "inline" or "flex".
To fix this issue you can modify the code to toggle the display property between "none" and an empty string. This will cause it to revert back to the elements default display style. Here's the updated code.

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" style="display: none;">This is the target element.</div>
 <script>
 function toggleElement() {
 var element = document.getElementById("target");
 element.style.display = (element.style.display === "none") ? "" : "none";
 }
 </script>
</body>
</html>
```

Exercise 2: CSS

TroubleshootingProblem

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>
.contain
er {
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>
</

ht

ml

>

T

as

ks

:
```

Identify the issue in the provided CSS code.
Debug and fix the code so that the container is centered on the page.

Solution:

The issue in the provided CSS code is that the margin: auto; property is not working as expected for centering the container.
To center the container, you need to use a combination of margin and text-align. Here's the corrected code:

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>
```

Exercise 3: Debugging JavaScript Functions

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
```

Solution:

The provided JavaScript function is straightforward, and the initial code appears to be free of

syntax errors.
no wrong in code

Exercise 2: Debugging CSS Styling Issues
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>
.contai
ner {
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:

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: 100%;
   text-align: center;
  }

  .box {
   width: 100px;
   height: 100px;
   background-color: #007bff;
   color: #ffffff;
   text-align: center;
   line-height: 100px;
   display: inline-block;
   margin: 10px;
  }
 </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>
```

In this corrected code:

I set the .container width to 100% so that it occupies the full width of its parent.

I added text-align: center; to the .container to center-align the boxes horizontally within the container.

I used display: inline-block; for the .box elements to ensure that they are displayed inline and align horizontally.

I added some margin to the .box elements to create space between them.

With these changes, the boxes should be properly centered and spaced in the container, and the colors remain as specified in your code.