Exercise 10 - Temperature Converter Tool

Introduction

In this exercise, you'll be implementing the JavaScript functionality for a Temperature Converter Tool. The HTML and CSS have already been provided, and your task is to write the JavaScript code to make the tool functional.

Objective

- Understand and implement closures in JavaScript.
- Utilize event listeners to respond to user interactions.
- Manipulate the DOM to display the conversion results.

Requirements

You have been provided with the HTML and CSS code, and the interface consists of:

- An input field to enter the temperature.
- Two buttons to convert the temperature to Celsius or Fahrenheit.
- A display area to show the conversion result.

Your JavaScript code must:

- Retrieve the input temperature value.
- Determine which button was clicked (Celsius or Fahrenheit) and perform the corresponding conversion.
- Update the display area with the converted temperature.

Instructions

1. **Utilizing a Self-Invoking Function (IIFE) to Create Closure**

In this exercise, we'll use a self-invoking function (IIFE) to encapsulate the JavaScript code for the Temperature Converter Tool. Here's how this approach is beneficial in this particular context:

1. **Encapsulation**: By wrapping the temperature conversion logic inside an IIFE, we make sure that all variables, such as temperature input and buttons, are confined within the scope

of the function. This protects them from potential naming conflicts with other parts of the code.

2. **Closure Creation**: The use of closure within the IIFE enables us to maintain access to the temperature input and conversion buttons, even after the function has executed. This is essential as we'll be adding event listeners to these elements that must remember their state and associated behavior.

Why Closure for Temperature Converter Tool?

- **Maintaining State**: Closure ensures that the temperature value and conversion methods are always accessible to the event listeners, maintaining the consistency of the user's input and conversion selection.
- **Clean Code**: By grouping the related temperature conversion logic inside a closure, the code becomes more modular and easier to follow. This also aids in making adjustments or extensions to the functionality in the future.

This pattern not only aligns with professional coding standards but also provides a robust and maintainable structure specifically suited to the temperature conversion functionality.

By following this pattern, we'll create a robust foundation of our JavaScript knowledge, aligning with best practices in real-world development.

---

2. **Access DOM Elements**: Inside the closure, access the input field, buttons, and display area using the `getElementById` method.

3. **Add Event Listeners**: Attach event listeners to the Celsius and Fahrenheit buttons. These will detect click events and call corresponding functions.

4. **Implement Conversion Functions**: In this part of the exercise, you'll implement the event listeners and conversion logic for both Celsius and Fahrenheit buttons. Follow the steps below to achieve this:

Celsius Button Event Listener

1. **Attach an Event Listener**: Add an event listener to the "Convert to Celsius" button. Listen for the 'click' event.

2. **Retrieve Input Value**: Inside the callback function, retrieve the temperature value from the input field. Make sure to convert this value into a floating-point number.

3. **Perform Conversion to Celsius**: Search Google to find the way to this. Please don't use ChatGPT.

4. **Display Result**: Format the result and update the display area with the converted temperature in Celsius. You may include both the original Fahrenheit value and the converted Celsius value in the result.

Fahrenheit Button Event Listener

1. **Attach an Event Listener**: Add an event listener to the "Convert to Fahrenheit" button, also listening for the 'click' event.

2. **Retrieve Input Value**: Similar to the previous step, get the temperature value from the input field, converting it into a floating-point number.

3. **Perform Conversion to Fahrenheit**: Search Google to find the way to this. Please don't use ChatGPT.

4. **Display Result**: Format and update the display area with the converted temperature in Fahrenheit. Include both the original Celsius value and the converted Fahrenheit value in the result.

Tips and Hints

- Remember to use the correct method to convert the input value to a floating-point number.
- For formatting the result, you may use string interpolation or concatenation.
- Consider using the `toFixed` method to format the temperature to a specific number of decimal places.
- Remember, closures allow you to create private scopes in JavaScript. Consider how you might use this feature to encapsulate the code.
- Use `parseFloat` to ensure that the input temperature is a number.
- You may use the `toFixed` method to format the converted temperature to a specific number of decimal places.

Evaluation Criteria

- Correct implementation of closure.
- Proper usage of event listeners.
- Accurate temperature conversion.
- Effective DOM manipulation to display the results.

Conclusion

This exercise is aimed at enhancing your understanding of closures and event listeners in JavaScript. Carefully follow the step-by-step guidance and apply your existing knowledge to solve the problem. Don't hesitate to refer to the additional resources if needed. Good luck, and enjoy coding!