

## Hw11 Prompts

I need to set up event listeners for two input elements with IDs "cInput" and "fInput". And the "input" event fires whenever the user types in the fields. Answer me without introductions.

For the first input field (cInput), whenever its value changes, the code checks if the field now contains any text. If it does, it clears the value of the second input field (fInput).

Similarly, for the second input field (fInput), whenever its value changes and contains text, the code clears the first input field (cInput).

```
cInput.addEventListener("input", function() {  
  if (cInput.value !== "") {  
    fInput.value = "";  
  }  
});
```

```
fInput.addEventListener("input", function() {  
  if (fInput.value !== "") {  
    cInput.value = "";  
  }  
});
```

This creates a mutual exclusivity between the two input fields - only one can contain data at any given time. This pattern is often used when you want the user to provide input in only one of two related fields.

I'm trying to understand this JavaScript code that checks input values. What does this section do?

```
} else if (fInput.value !== "") {  
  const fahrenheit = parseFloat(fInput.value);
```

This is part of a conditional statement that checks if an input field with ID "fInput" has any content. If the field is not empty (value is not an empty string), it proceeds to execute the code inside this block.

Inside the block, it's converting the value from the input field (which would be a string) into a floating-point number using the `parseFloat()` function, and storing it in a variable called `fahrenheit`.

This pattern is common when you need to perform numerical operations on user input that comes in as text. The `parseFloat()` function attempts to convert the string to a number, which will allow mathematical operations to be performed on it later in the code.

The click button has not been working, check my code

```
convertButton.addEventListener("click", function()
```

This code is setting up an event listener for a button with ID "convertButton". The event being listened for is the "click" event, which occurs when a user clicks on the button.

The `addEventListener()` method takes two parameters:

1. The event type to listen for ("click" in this case)
2. A callback function that will be executed when the event occurs

The second parameter is an anonymous function (a function without a name) defined using the `function()` syntax. All the code inside this function will run whenever the button is clicked.

This is a fundamental pattern in JavaScript for making interactive web pages. It allows the page to respond to user actions like clicking buttons without having to reload the entire page.

Your code should be correct. Check any syntax errors that might have led to malfunctioning.