```
what is wrong with this code? const htmlBody =
document.body
const submitButtonClick =
document.querySelector('button')
const colorInput =
document.querySelector('input').value
submitButtonClick.addEventListener('click',btnClick(col
orInput))
function btnClick (colorInput) {
  if (colorInput = 'red' || colorInput=='green' ||
colorInput=='blue' || colorInput=='white' ||
colorInput=='black'){
     htmlBody.style.backgroundColor =
${colorInput}
  }else{
     alert ("You can only enter one of the following:
'red', 'green', 'blue', 'black', or 'white'. ")
  }
}
                                                  ብ ⁄ዖ
```

There are a few issues with your JavaScript code:

Issues:

Incorrect addEventListener Syntax:

```
js Copy Dedit

submitButtonClick.addEventListener('click', btnClick(colorInput))
```

- btnClick(colorInput) is immediately invoked when the script runs instead
 of being assigned as an event listener.
- Fix: Pass a function reference instead of calling it immediately. You can use an arrow function to pass the latest colorInput value.
- 2. Incorrect Assignment in if Condition:

```
if (colorInput = 'red' || colorInput == 'green' || colorInput ==
'blue' || colorInput == 'white' || colorInput == 'black')
```

- = is an assignment operator, not a comparison operator. You should use === for strict comparison.
- The condition is also incorrect because colorInput = 'red' always evaluates as true, making the whole condition always true.
- Fix: Use === and wrap the condition properly with parentheses.

3. Value of colorInput is Captured Too Early:

- This captures the input value **only once when the script loads**, meaning it won't update when the user types a new value.
- Fix: Move colorInput inside the event listener function to get the latest value.

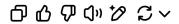
Fixed Code:

```
const htmlBody = document.body; const submitButtonClick =
document.querySelector('button');
submitButtonClick.addEventListener('click', () => { const colorInput =
document.querySelector('input').value.trim().toLowerCase();
btnClick(colorInput); }); function btnClick(colorInput) { const
validColors = ['red', 'green', 'blue', 'white', 'black']; if
(validColors.includes(colorInput)) { htmlBody.style.backgroundColor =
colorInput; } else { alert("You can only enter one of the following:
'red', 'green', 'blue', 'black', or 'white'."); } }
```

Improvements:

- ✓ Fixes the addEventListener issue by using an arrow function to pass the latest colorInput.
- Ensures proper comparison by using === and .includes().
- Retrieves the input value on each button click instead of only once at page load.
- ✓ Handles case sensitivity (.toLowerCase()) and removes extra spaces (.trim()).

This should now work correctly! 🚀



there is still an issue with this code. The background color entered by the user only appears for a fraction of a second, after which it reverts back to white



You're right! The issue is likely caused by the default behavior of the button inside a <form> .

ssue:

- If the button is inside a <form> , clicking it submits the form by default, causing the page to reload immediately.
- This makes the background color change briefly before resetting when the page refreshes.

Fix:

Prevent the default form submission using event.preventDefault().

Updated Code:

```
const htmlBody = document.body; const submitButtonClick =
document.querySelector('button');
submitButtonClick.addEventListener('click', (event) => {
  event.preventDefault(); // Prevent form from refreshing the page const
  colorInput = document.querySelector('input').value.trim().toLowerCase();
  btnClick(colorInput); }); function btnClick(colorInput) { const
  validColors = ['red', 'green', 'blue', 'white', 'black']; if
  (validColors.includes(colorInput)) { htmlBody.style.backgroundColor =
  colorInput; } else { alert("You can only enter one of the following:
  'red', 'green', 'blue', 'black', or 'white'."); } }
```

Alternative Fix:

If the button is inside a <form>, another solution is to change:



instead of:

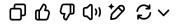


This ensures that clicking the button doesn't trigger form submission.

Why Does This Work?

- event.preventDefault(); stops the form from submitting, preventing the page reload.
- .trim().toLowerCase(); ensures the input is clean and case-insensitive.
- Background color now correctly stays after clicking the button.

This should now work perfectly! 🚀 Let me know if you need more help. 😊



Indeed, it works perfectly now! Thanks a lot!



You're very welcome! I'm glad it's working perfectly now. Happy coding, and feel free to ask if you ever need more help!

