



LEARNING PATHS

Improve your solution

Take some time to find where you can improve your solution. You can use the design comparison slider and reports to help discover areas to work on. After updating your code and pushing the updates to your live site, generate a new screenshot or report to see the changes.

Improved your solution?

Generate a new screenshot to display your changes. You have unlimited screenshots.

REGENERATE SCREENSHOT

SOLUTION

DESIGN

Shift the overall look and feel by adding these wonderful touches to furniture in your home

Ever been in a room and felt like something was missing? Perhaps it felt slightly bare and uninspiring. I've got some simple tips to help you make any room feel complete.

Michelle Appleton
28 Jun 2020

[Accessibility report](#) [HTML report](#) [CSS report](#) [JavaScript report](#)

Reports are only visible to you

Improved your solution?

Generate a new report to resolve errors and warnings.

[REGENERATE REPORTS](#)[Learn more about our JavaScript report](#)

AI Analysis Summary

You've demonstrated good use of modern JavaScript features like `const` and provided helpful comments explaining the toggle logic. To take it further, consider removing unnecessary event parameter checks and clarifying UI state changes on `hideIcon` click for better user experience.

Priority recommendations:

- 1 Remove the `if (share)` checks inside event listeners since the event object is always truthy and the condition is redundant.
- 2 Update the `hideIcon` click handler to clearly reflect UI state changes, such as hiding the icon or adjusting styles to indicate the social links are hidden.
- 3 Ensure the `.social` element has the `hidden` class initially in your HTML to make the toggle behavior intuitive.

Issues 2

[All \(2\)](#) [Error \(0\)](#) [Warning \(2\)](#) [Info \(0\)](#)

WARNING [Logic & Correctness](#) AI Detected 1 occurrence

The event listener callbacks check the event object with `if (share)` which is unnecessary and misleading. This causes confusion and may lead to incorrect assumptions about event handling.

```
shareButton.addEventListener("click", function (share) {
  if (share) {
    socialLinks.classList.toggle("hidden");
    /* ... */
    hideIcon.style.top = "0";
  }
});
```

script.js:6

HOW TO FIX THIS ISSUE

What's the issue?

The event listener callback uses `if (share)` to check the event object, which is unnecessary and misleading. In JavaScript event handlers, the event object is always truthy, so this condition doesn't add any meaningful check. It can confuse readers into thinking the event might sometimes be falsy or missing, which is not the case.

Why does this matter?

Writing clear, intentional code helps maintainability and reduces confusion for anyone reading or updating your code—including future you. Misleading conditions can cause developers to waste time debugging or misunderstanding the event flow. Clean, straightforward event handlers demonstrate professionalism and improve code quality, which is crucial when building projects for portfolios or real-world applications.

How to fix it

Remove the unnecessary `if` check and directly execute the intended logic inside the event listener. Here's the before and after:

```
// Before: unnecessary and misleading check on event object
shareButton.addEventListener("click", function (share) {
  if (share) { // event object is always truthy, so this check is redundant
    socialLinks.classList.toggle("hidden");
    /* ... */
    hideIcon.style.top = "0";
  }
});
```

```
// After: clean and clear event handler without redundant checks
shareButton.addEventListener("click", function (event) {
  // event is always passed and truthy, so no need to check it
  socialLinks.classList.toggle("hidden");
  /* ... */
  hideIcon.style.top = "0";
});
```

Best practice tip

Trust the event system: event listeners always receive a valid event object. Avoid redundant checks on the event parameter to keep your code clean and focused on the actual logic you want to implement.

TLDR

Remove unnecessary checks on the event object in event listeners to write clearer, more maintainable code.

Next steps

Update your event listener by removing the `if` condition, test the toggle behavior locally to confirm it works as expected, then push your changes to GitHub. Generate a new solution report to verify the issue is resolved.

Remember, refining your code iteratively is a key part of learning—don't hesitate to ask the community if you want feedback or clarification!

WARNING **Logic & Correctness** **AI Detected** 1 occurrence

The `hideIcon` click handler toggles the 'hidden' class and sets `hideIcon.style.top = '0'` but does not hide the icon or update UI state clearly. This may confuse users as the icon remains visible and positioned the same.

```
hideIcon.addEventListener("click", function (share) {
  if (share) {
    socialLinks.classList.toggle("hidden");
    hideIcon.style.top = "0";
  }
});
```

script.js:15

LEARN HOW TO FIX THIS ISSUE

AI-detected issues notice

Some issues shown above were detected by AI analysis and may require human review for accuracy. If you're unsure about any findings or need clarification, please reach out in the help channel on Discord for assistance.

Strengths 2

Readability & Formatting

Clear and descriptive comments explain the purpose and behavior of the `classList.toggle` usage, helping learners understand the toggle mechanism.

Modern JavaScript

Uses `const` for DOM element references, ensuring variables are not reassigned and improving code safety.

CONTINUE

Your path progress

Welcome to the JS fundamentals path!

Setting up your project to use JavaScript

The Document Object Model

Handling user events

Article preview component

Start solution

Submit solution

Improve solution

Peer review

Challenge completed

Managing forms with JavaScript

Newsletter sign-up form with success message

Working with data

Time tracking dashboard

Refactoring your code

Tip calculator app

Password generator app

Frontend Quiz app



Stay up to date

with new challenges, featured solutions, selected articles, and our latest news

SUBSCRIBE

FRONTEND MENTOR

Unlock Pro

Contact us

FAQs

Become a partner

EXPLORE

Learning paths

Challenges

Solutions

Articles

Use cases

COMMUNITY

Discord
Guidelines

FOR COMPANIES

Hire developers
Train developers

© Frontend Mentor 2019 - 2026

[Terms](#) [Cookie Policy](#) [Privacy Policy](#) [License](#)