HOW SOFTWARE WORKS · BUILDER · 14 MIN READ

JavaScript in the browser: events and the DOM

How JavaScript finds things on a page, reacts to clicks and typing, handles forms, and how frameworks like React do the same job for you.

Pass the check (4 of 5) online to earn the “Event Handler” badge:
https://digitalbuilders757.com/university/study/javascript-events-and-the-dom

The DOM: the page as a tree

When a browser loads HTML it builds the DOM (Document Object Model): a live tree of every element on the page. JavaScript can find elements in it, read them and change them, and the page updates straight away.

Events: run code when something happens

Clicks, key presses, typing and form submissions are events. You attach a function to an element with addEventListener, and it runs each time the event happens. For example: const button = document.querySelector('#like'); let likes = 0; button.addEventListener('click', () => { likes++; button.textContent = 'Likes: ' + likes }). Find the element, listen for the event, update the page.

Forms the right way

- Listen for the form's submit event (it covers the Enter key too), not just a button click.
- Call event.preventDefault() when you handle it yourself, so the page doesn't reload.
- Input values are always text: convert numbers with Number() and trim spaces.
- Check the input, show a clear message next to the field, and check again on the server, because browser checks can be bypassed.

Frameworks and debugging

React, Vue and similar frameworks handle the DOM for you: you change state, and they update the page. The ideas are the same: events trigger functions that change data, and the screen reflects the data. When something doesn't react, open the browser's developer tools, use console.log inside your handler to see whether it runs, and read the Console for errors.

Key terms

DOM (Document Object Model)
The browser's live tree of a page's elements, which JavaScript can read and change.

Event listener
A function that runs when something happens on the page, like a click or a form submit.

JavaScript
The programming language that makes web pages interactive.

Function
A named, reusable recipe in code that takes inputs and can return an output.

State
Data a component or app remembers and updates, like what's in a cart.

Browser console
The developer-tools panel (Inspect, then Console) that shows a page's errors and logs.

Developer tools (Inspect)
Browser tools that show a page's HTML and CSS and let you experiment live; changes aren't saved.

HTML
The language for a web page's structure: headings, paragraphs, buttons, images.

Try it: Make a button do something

1. Ask an AI for a single HTML file with a button and a counter, using addEventListener (no frameworks).
2. Open it in your browser, click the button, and find the listener in the code.
3. Add console.log inside the listener and watch the Console in developer tools.
4. Add a small form with a number field that shows double the number on submit, using preventDefault and Number().

KNOWLEDGE CHECK

JavaScript in the browser: events and the DOM

5 questions. Circle one answer for each; 4 or more correct passes. Answers are at the back.

1. What is the DOM?

A. A database on the web server that stores each page's text and images
B. The browser's live tree of the page that JavaScript can change
C. A type of domain name
D. A CSS framework

2. What does addEventListener('click', handler) do?

A. Clicks the element automatically as soon as the page has loaded
B. Adds a click counter to the element and shows it on the page
C. Runs the handler function every time the element is clicked
D. Adds a new button

3. Your form reloads the page and loses what you typed. What's missing?

A. A required attribute on every input
B. More CSS
C. A second form
D. event.preventDefault() in the submit handler

4. Why check form input on the server as well as in the browser?

A. Browser checks can be bypassed
B. Servers are faster at checking than browsers
C. It's required by the HTML standard
D. Browsers can't check input at all

5. Your click handler seems to do nothing. Good first debugging step?

A. Rewrite the whole page with a different framework, then test again
B. Add console.log in the handler and check the Console for errors
C. Change the button colour
D. Restart your computer

Official sites

MDN: Introduction to the DOM: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction
MDN: addEventListener: https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener

Last reviewed 2026-11-25. AI tools change fast; check each tool's own site for current features, plans and prices.

ANSWER KEY

JavaScript in the browser: events and the DOM

1. B: The browser's live tree of the page that JavaScript can change
Change the DOM and the page on screen changes.

2. C: Runs the handler function every time the element is clicked
The listener waits for the event and calls your function each time.

3. D: event.preventDefault() in the submit handler
Browsers submit and reload by default; preventDefault lets your code handle it.

4. A: Browser checks can be bypassed
Anyone can send requests without your page. The server is the real gatekeeper.

5. B: Add console.log in the handler and check the Console for errors
Logging shows whether the handler runs at all, and errors explain why not.