

Introduction to JavaScript

40 minutes

Introduction to Web Frontend

Outcome 1.22

Outcome 2.7

Outcome 2.16

Outcome 2.20

Outcome 3.3

WHAT THIS LESSON DOES

Get started with the programming language that brings web pages to life. In this lesson, you'll explore basic JavaScript syntax, create variables, and learn to integrate JavaScript into HTML. Build a simple page to display messages dynamically.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
1.22	read, write, test, and modify computer programs
2.7	implement algorithms using a programming language to solve a range of problems
2.16	use data types that are common to procedural high-level languages
2.20	identify and fix/debug warnings and errors in computer code and modify as required
3.3	use appropriate programming languages to develop an interactive website that can display information from a database that meets a set of users' needs

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the role of JavaScript in adding interactivity to web pages alongside HTML and CSS.
- Learn to set up a basic project structure using VS Code with HTML and JavaScript files.
- Master the inclusion of JavaScript in HTML through inline scripting and external files.
- Grasp the concept of variables using 'let' and 'const' to store and manage data.
- Develop skills in manipulating web page content dynamically using the Document Object Model (DOM).

BEFORE THE LESSON

- Confirm VS Code is installed on every machine and that students can create a folder and files in the Explorer pane.
- Test the full walkthrough yourself once: an alert, a console.log visible in browser dev tools, and an innerHTML update. Know how to open the console (F12) so you can show it.
- Have the starter HTML ready to paste or dictate, since the code blocks in the steps are truncated on screen.

Introduction to JavaScript

HOW THE LESSON RUNS

Welcome to JavaScript
Setting Up Your Project
Adding JavaScript to HTML
Using an External JavaScript File
Introducing Variables
Using Variables in HTML
Basic Syntax and Data Types
Challenge: Create Your Own Variable
Lesson Wrap-Up

TEACHING MOVES

- 2. Setting Up Your Project. Walk the folder-then-file order explicitly. Many will make files on the desktop or forget to open the folder. Have everyone show you an index.html sitting inside IntroToJS before moving on.
- 3. Adding JavaScript to HTML. Point out the script tag goes at the end of the body so the HTML above it exists first. Demonstrate the alert, then note it blocks the page until dismissed, which is why we soon stop using it.
- 4. Using an External JavaScript File. Open the browser console (F12) so console.log output is visible. Stress that src must match the filename exactly and that the external script replaces the inline one, not sits beside it.
- 5. Introducing Variables. Demonstrate the const error live: try to reassign number and show the red console message. That failure is more memorable than the rule. Frame let as changeable, const as fixed.
- 6. Using Variables in HTML. Make the link explicit: the id in the HTML and the string in getElementById must match character for character, including case. This is the single most common failure point in the lesson.
- 7. Basic Syntax and Data Types. Highlight that strings need quotes but numbers and booleans do not. Show what happens if quotes are added to age: it still works but is now text, which matters once you do arithmetic.
- 8. Challenge: Create Your Own Variable. Let them choose their own variable to build ownership. Circulate and check the new variable is both declared and added to the innerHTML string, since forgetting the second half is common.

WHAT TO WATCH FOR

- Students expect a change to show on the page without saving the file or refreshing the browser. Make save-then-refresh a reflex. Say it aloud after every edit for the first few steps until it sticks.
- `getElementById` returns nothing because the id in the HTML and the string in the JavaScript do not match exactly, or the case differs. When a page does not update, have them read the id and the string aloud together. Mismatched capitals cause silent failure with no error.
- Confusing where `console.log` output appears, expecting it on the page rather than in the browser console. Show the console once at the start and be clear: `console.log` is for us the programmer, `innerHTML` is for the user.

DIFFERENTIATION

- Emerging: Pair them with a partner who has a working folder so environment problems do not stall the whole period. Give them the full `index.html` to paste rather than type, so the focus stays on the JavaScript concept.
- Developing: Ask them to predict what appears on the page before they refresh, then check. Have them explain aloud the difference between `let` and `const` before moving to the next step.
- Proficient: Ask them to add two numbers and display the result, then investigate what happens when one is typed as a string, previewing type coercion at Higher Level. Challenge them to move all their variables into a single object or to write output using a template literal instead of concatenation.