LESSON 49

Dynamic Form Validation with JavaScript

40 minutes | Introduction to Web Frontend | Outcome 1.22 | Outcome 2.6 | Outcome 2.7 | Outcome 2.19 | Outcome 2.20 | Outcome 1.15 | Outcome 3.3

WHAT THIS LESSON DOES

In this lesson, you'll learn to validate forms using JavaScript, ensuring users input correct data. Set up a workspace in VS Code, build a registration form, style it with CSS, and add validation for name, email, and password fields.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
1.22	read, write, test, and modify computer programs
2.6	construct algorithms using appropriate sequences, selections/conditionals, loops and operators to solve a range of problems, to fulfil a specific requirement
2.7	implement algorithms using a programming language to solve a range of problems
2.19	test solutions and decisions to determine their short-term and long-term outcomes
2.20	identify and fix/debug warnings and errors in computer code and modify as required
1.15	consider the quality of the user experience when interacting with computers and list the principles of universal design, including the role of a user interface and the factors that contribute to its usability
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 importance of form validation in enhancing web application reliability and user experience.
- Develop skills to create and structure a registration form using HTML and CSS.
- Implement dynamic form validation using JavaScript for various input fields.
- Master event handling and DOM manipulation to manage form submissions and display error messages.
- Apply logical checks to validate user input data against specific criteria.

BEFORE THE LESSON

- Confirm VS Code opens and can create folders and files on the student machines, and that students know how to open an HTML file in a browser (Live Server or double-click).
- Work through the full solution yourself once. The step descriptions are truncated, so have the complete script.js and index.html to hand for students who fall behind.
- Decide where students save the FormValidation folder so work survives to the next class.

Dynamic Form Validation with JavaScript

HOW THE LESSON RUNS

- Introduction to Form Validation
- Setting Up Your Project Folder
- Creating the HTML Form
- Adding CSS for Styling
- Adding the JavaScript File
- Validating the Name Field
- Validating the Email Field
- Validating the Password Field
- Testing the Form
- Challenge: Add Age Field and Validation
- Wrap Up

TEACHING MOVES

- 1. Introduction to Form Validation. Ask why a shop website checks your email format before it accepts an order. Land the point that validation protects data before it reaches storage, so the code they write today has a real job.
- 3. Creating the HTML Form. Stress that each input needs a matching id and a nearby empty span with a matching error id, because the JavaScript targets those ids exactly. A mistyped id here is the commonest reason nothing shows up later.
- 4. Adding CSS for Styling. Keep this brief. Styling is not the lesson's point, so let weaker students copy it and move on rather than debugging box-shadow syntax.
- 6. Validating the Name Field. Walk the class through preventDefault first: without it the form submits and the page reloads before any check runs. Then show the isValid flag pattern, since every later field reuses it.
- 7. Validating the Email Field. Be honest that checking for '@' and '.' is a crude check, not a real email test. Ask what addresses it would wrongly accept. Good bridge to Higher Level discussion of validation limits.
- 9. Testing the Form. Insist they test empty, invalid and valid input in that order. A form that looks right is not proven right. Have them predict the outcome before each submit.
- 10. Challenge: Add Age Field and Validation. Hold the solution back. Prompt with parseInt and the 13 to 99 range only when a student is genuinely stuck, and ask what happens if someone types letters into a number field.

WHAT TO WATCH FOR

- Students omit preventDefault, so the page reloads on submit and their validation never appears to run. Have them submit once with preventDefault commented out to see the reload, then restore it. The visible refresh makes the fix stick.
- An id in the HTML does not match the id the JavaScript looks up, so getElementById returns null and the script errors silently. Open the browser console together and read the null error aloud. Teach console checking as the first move when nothing happens.
- Students treat a form that displays correctly as one that works correctly, and skip testing invalid input. Require the three test cases in Step 9 and ask them to break their own form on purpose before declaring it done.

DIFFERENTIATION

- Emerging: Give the complete, correct HTML and CSS so they only write the JavaScript validation blocks. Pair them with a partner and have them say the rule aloud, name at least 3 characters, before writing the if condition.
- Developing: Ask them to add the second and third fields by copying the name pattern themselves rather than pasting the provided block, so they internalise the structure.
- Proficient: Set the age challenge, then push further: reject non-numeric age input and give the exact reason in the error span. Ask them to research why real sites validate on the server too, not just in the browser. Useful evidence for a coursework write-up.