LESSON 46

Setting up your Development Environment

40 minutes | Introduction to Web Frontend | Outcome 1.19 | Outcome 1.22 | Outcome 1.23 | Outcome 2.15 | Outcome 2.19 | Outcome 2.20 | Outcome 3.3

WHAT THIS LESSON DOES

In this lesson, you'll learn how to set up a local web development environment on your computer. Follow step-by-step guidance to install Visual Studio Code, create a project folder, build a simple web page, and understand file structures.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
1.19	identify features of both staged and iterative design and development processes
1.22	read, write, test, and modify computer programs
1.23	reflect and communicate on the design and development process
2.15	explain what is meant by the World Wide Web (WWW) and the Internet, including the client server model, hardware components and communication protocols
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
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 a local development environment for web development.
- Master the installation and setup of Visual Studio Code as a primary coding tool.
- Learn to organise project files and folders effectively for web projects.
- Develop skills in creating and linking basic HTML and CSS files for web pages.
- Gain confidence in previewing and testing web content in a browser.

BEFORE THE LESSON

- Confirm VS Code either installs or is already present on the room machines. The download and install may be blocked by the school filter or need admin rights, so test one machine before class and know your workaround.
- Check students can save files into Documents and open them in a browser. On locked-down accounts the file path may differ from Documents\MyFirstWebPage.
- Have a sample image ready to distribute, since the linked pic-1.jpg and pic-2.jpg may not be reachable behind the filter. Any small jpg saved as sample.jpg works.

Setting up your Development Environment

HOW THE LESSON RUNS

- Introduction to Web Development Environments
- Downloading and Installing VS Code
- Creating a Project Folder
- Creating Your HTML File
- Viewing the Web Page
- Adding an Image to Your HTML
- Creating a CSS File
- Linking to Another Page
- Activity
- Wrap Up

TEACHING MOVES

- 1. Introduction to Web Development Environments. Sell the why: a local environment works offline, keeps files together and matches how real developers work. Contrast it briefly with the online editors they have been using.
- 2. Downloading and Installing VS Code. If installs are blocked, have this open before class rather than watching thirty download bars stall. Mention version control and Git in passing only, it is context here, not the day's skill.
- 3. Creating a Project Folder. Insist everyone opens the folder as the workspace, not a loose file. Warn about the 'trust the authors' prompt so nobody freezes on it. The Explorer pane on the left is where they live from now on.
- 4. Creating Your HTML File. Stress the exact name index.html, all lowercase, no spaces. Have them type the boilerplate rather than paste it so the structure sinks in, then save with Ctrl+S before moving on.
- 5. Viewing the Web Page. Show them double-clicking the file in Explorer opens it in the browser. Make the point that a saved file must be reopened or refreshed to show changes: editing alone changes nothing on screen.
- 6. Adding an Image to Your HTML. This is where paths break. The src must match the images subfolder and the exact filename including .jpg. A broken image icon almost always means a wrong path or a typo, not broken code.
- 7. Creating a CSS File. The css lives in a separate file now, so the link tag in the head is what connects them. If styling does not appear, check that link line and the href path before touching the CSS itself.
- 8. Linking to Another Page. Have them click back and forth between index.html and about.html in the browser. The href points to a filename in the same folder, so a relative path with no leading slash.
- 9. Activity. Set one required change and let the rest be open: everyone adds a working gallery.html linked from the home page. Circulate and fix broken paths, which is where most will get stuck.

WHAT TO WATCH FOR

- Students edit a file, do not save, then wonder why the browser looks unchanged. Make Ctrl+S then refresh the browser a fixed ritual. Point out the unsaved dot on the VS Code tab.
- A broken image icon is read as broken code or a corrupt file. Show that it is almost always a path or filename mismatch. Have them compare the src string letter by letter against the actual file in the images folder, including capitalisation and .jpg.
- Believing a page that displays without errors is therefore correct, even when CSS never applied. Ask what they expected to see versus what appeared. If the background is not lightblue, the link tag or href is wrong, even though the page loads fine.

DIFFERENTIATION

- Emerging: If the install is fighting them, pair them onto a working machine so they still complete the HTML and CSS rather than losing the period to setup. Give the boilerplate as a starter they modify rather than type from scratch, so a single typo does not stop everything.
- Developing: Have them explain aloud why the image and CSS live in subfolders before you help with a path error, so they debug the relationship rather than guess.
- Proficient: Push them to add a third page and a shared navigation block repeated across all pages, and to notice the maintenance cost of copying it. Ask them to keep this project folder tidy as evidence of file organisation they could reuse in the Coursework Project.