

CSS Website Layout

20 minutes

Introduction to Web Frontend

Outcome 1.15

Outcome 1.16

Outcome 1.19

Outcome 1.22

Outcome 1.23

Outcome 2.20

Outcome 3.3

WHAT THIS LESSON DOES

Delve into the world of CSS website layouts with this step-by-step lesson. Learn about the common areas of a website, how to code a header, content and footer, and how to structure your layout with columns. Experiment with HTML and CSS code to create your own website layout.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
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
1.16	compare two different user interfaces and identify different design decisions that shape the user experience
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
2.20	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 fundamental components of a website layout, including header, content, and footer areas.
- Identify common website layout structures such as one, two, and three-column designs.
- Apply CSS properties to style and position website elements effectively.
- Create responsive column layouts using CSS width and float properties.
- Integrate header, content, and footer sections into a cohesive website design.

BEFORE THE LESSON

- Run each of the built-in Run Code examples yourself first so you know what a correct header, column and footer preview looks like when a student's does not match.
- Confirm the in-browser code editor loads for the class and that students can edit both the HTML and CSS panes.

CSS Website Layout

HOW THE LESSON RUNS

Introduction

Header

Content

Columns

Footer

Putting it all together

TEACHING MOVES

- 1. Introduction. Sketch the three stacked regions on the board and ask which stay the same across pages. Draw out that header and footer repeat while content changes, so branding and navigation belong in the header.
- 2. Header. Point out the brand link wraps to the homepage and the menu holds the site's main links. Have students trace which CSS class styles which element before they edit anything.
- 3. Content. Show the one, two and three column diagrams and ask which suits a product list versus an About Us page. Keep this a discussion, no code yet.
- 4. Columns. Stress that without `float:left` the divs stack instead of sitting side by side. Have students total the widths and padding out loud: 31 plus 65 plus the padding must stay under 100 percent.
- 5. Footer. Compare the footer structure to the header they already built: same div-with-classes pattern, different content. Ask what secondary links a car dealer would put here.
- 6. Putting it all together. Walk the combined file top to bottom so students see the regions appear in source order. Have them find each earlier example inside the full page before running it.

WHAT TO WATCH FOR

- Students set two column widths to 50 percent each plus padding and cannot see why the columns break to separate lines. Have them add the widths and all padding together. Show that the total must stay under 100 percent, which is why the example uses 31 and 65.
- Students omit `float:left` and expect columns side by side, then think the width is broken. Delete `float` from a working example live so they watch the columns stack, then add it back. The `float`, not the width, creates the row.
- A preview that looks roughly right is assumed correct even when the HTML is malformed. Have them check the browser renders each region in the right place and that every `div` they opened is closed, not just that something appeared.

DIFFERENTIATION

- Emerging: Give them one region only, the footer, and have them change just the address text and link labels before touching any CSS. Pair them with a student who has a working preview so they can copy the structure and then explain each line back.
- Developing: Ask them to switch the two-column example to a three-column layout by choosing three widths that sum correctly under 100 percent.
- Proficient: Challenge them to make the layout respond to a narrow window, then explain why fixed percentage floats behave the way they do compared with a Higher Level responsive approach. Have them produce a screenshot and short note on their layout choices as evidence they could reuse in an Applied Learning Task.