

CSS Fonts

20 minutes

Introduction to Web Frontend

Outcome 1.15

Outcome 1.16

Outcome 1.22

Outcome 3.3

WHAT THIS LESSON DOES

In this step-by-step lesson, you'll learn how to use CSS to set fonts for different text elements on a webpage. You'll understand how to set font family, use fallback fonts, adjust font weight, size, and style. Finally, you'll apply these skills in an exercise to code a paragraph of text with specific font properties.

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
1.22	read, write, test, and modify computer programs
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 font selection in enhancing website user experience.
- Master the use of the font-family property to specify and apply different fonts.
- Learn to implement fallback fonts to ensure text visibility across browsers.
- Apply font-weight and font-size properties to adjust text appearance.
- Utilise font-style property to modify text presentation effectively.

BEFORE THE LESSON

- Open the in-lesson code runner yourself and confirm the Run Code examples render, so students see fonts change live rather than reading about them.
- Note that Lato in the fallback example is deliberately not web-safe, so the browser shows Courier New. That is the point of the step, not a bug.

CSS Fonts

HOW THE LESSON RUNS

Introduction

Font Family

Fallback

Font Weight

Font Size

Font Style

Exercise

TEACHING MOVES

- 2. Font Family. Have students run the five examples and say aloud which fonts look serif and which look sans-serif. Point out these work because they are pre-installed web-safe fonts, not because CSS supplies them.
- 3. Fallback. Ask why anyone would list a font the browser cannot find. Then run the Lato example: the text appears in Courier New. Fallbacks are insurance for when a font is missing.
- 4. Font Weight. Contrast the keyword bold with the numeric 700. Show 100 and 400 side by side so students see many fonts only render a few of the numeric weights.
- 5. Font Size. Run the pixel ladder and have students change one value and re-run. Point out px is one unit among several; keep it simple here.
- 7. Exercise. Let students attempt all four properties in one style attribute before revealing the answer. Note the sample answer misspells Georgia as Geogia, so ask them to catch the typo.

WHAT TO WATCH FOR

- Students think font-family downloads or installs the font, so any font name they type will just work. Show a made-up font name rendering as the browser default. Explain the font must already be on the machine or supplied by a web font, which is why fallbacks matter.
- Students read the comma list in font-family as applying several fonts at once rather than as a priority order. Rewrite it as a sentence: try the first, if missing try the next, and so on. Re-run the Lato example to prove only the first available one is used.

DIFFERENTIATION

- Emerging: Give them one property at a time on a paragraph they already have, rather than the combined exercise, and check the result runs before adding the next.
- Developing: Ask them to predict what each example will look like before pressing Run, then confirm, so they reason rather than just observe.