

HTML Tables

20 minutes

Introduction to Web Frontend

Outcome 3.3

Outcome 1.22

Outcome 2.20

Outcome 2.18

Outcome 1.15

Outcome 2.15

WHAT THIS LESSON DOES

In this step-by-step lesson, you'll learn how to create HTML tables. Starting with an introduction to tables, you'll understand their structure and how to code them. You'll then move on to creating rows, columns, and cells, and adding multiple rows. Finally, you'll get hands-on experience by coding your own table, adding a header, body, and multiple rows.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
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
1.22	read, write, test, and modify computer programs
2.20	identify and fix/debug warnings and errors in computer code and modify as required
2.18	collect, store and sort both continuous and discrete data
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
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

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the purpose and structure of tables in organising data into rows and columns.
- Master the basic HTML tags used to create tables, including table, thead, tbody, and tfoot elements.
- Learn to construct table rows and cells using tr and td tags effectively.
- Develop the ability to build a complete HTML table with headers, body, and multiple rows.
- Apply practical skills by coding a personalised table with structured data.

BEFORE THE LESSON

- Open the in-browser HTML editor yourself and confirm the Run Code and 'see the correct code' functions work, and that the pre-filled CSS loads for the build steps.
- Have a blank editor tab ready to model live so you can type tags in front of the class rather than only reading them.

HTML Tables

HOW THE LESSON RUNS

Introduction

Structure of a HTML table

Table rows

Table columns and cells

Multiple rows

Code a table

Add the first body row

Add more body rows

TEACHING MOVES

- 1. Introduction. Project the mountains table and have students name the three columns and count the rows aloud. Point out which row is header, which is body, which is footer, before any tags appear. Anchor the vocabulary to the visible layout.
- 2. Structure of a HTML table. Stress that thead, tbody and tfoot are the three sections that live inside the table tags. Type the empty skeleton live and run it so they see nothing displays yet: structure without content shows nothing.
- 3. Table rows. Add three empty tr tags, run it, and let them see it stays blank. Use this deliberately: rows need cells before anything appears. This prevents the 'my code is broken' panic later.
- 4. Table columns and cells. Now add td cells and run it so content finally appears. Point out that number of td tags per row equals number of columns. Mention the CSS borders are only there to make the invisible grid visible.
- 5. Multiple rows. Show that every row must have the same number of td cells. Have them count cells per row in the example. Emphasise the repeating tr block as a copy-and-edit pattern.
- 6. Code a table. Set the build clearly: table, then thead, then one tr, then three td cells for Name, Age, Eye Colour. Circulate and check closing tags before anyone reaches for 'see the correct code'.
- 7. Add the first body row. Watch for students who put tbody inside thead. Say plainly: tbody goes after the closing thead tag, not inside it. Have them use their own real details so the data means something.
- 8. Add more body rows. Reinforce that each new person is a new tr with three td cells inside tbody. If a table looks misaligned, count the td tags in the short row: a missing cell is almost always the cause.

WHAT TO WATCH FOR

- Students think their table is broken when empty tr tags or an empty skeleton displays nothing. Explain that a table only renders once cells contain content. Add one td with a letter in it and re-run to prove the structure was fine all along.
- Rows with different numbers of td cells, causing a ragged or misaligned table. Teach the habit of counting cells per row. Every tr in this table needs exactly three td tags to match the three headers.
- Placing tbody inside thead, or forgetting closing tags so nesting collapses. Model the indentation on screen so open and close tags line up vertically. Have students point to each closing tag and name what it closes before they run.

DIFFERENTIATION

- Emerging: Give the header row as a template and ask them to change only the words inside the td tags, so they succeed before adding structure. Pair them with a partner and have them read the tag names aloud while the partner types.
- Developing: Ask them to add a tfoot section with a total or a note, using the same pattern they just used for tbody. Have them explain to you why nothing showed when rows had no cells, before moving on.
- Proficient: Challenge them to research and add a caption element or the th tag for proper header cells, comparing it to using td in thead. Ask them to justify when a table is the right structure for data and when it is being misused for page layout, a Higher Level distinction.