

Crafting Complex Tables

20 minutes

Introduction to Web Frontend

Outcome 3.3

Outcome 1.15

Outcome 2.21

WHAT THIS LESSON DOES

This step-by-step lesson guides you through the process of crafting complex HTML tables. You'll learn how to create a basic table, add a header and footer row, and use the 'rowspan' and 'colspan' attributes to merge cells. By the end, you'll be able to create advanced tables with ease.

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
1.15	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
2.21	critically reflect on and identify limitations in completed code and suggest possible improvements

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the fundamental structure and components of HTML tables.
- Implement header and footer rows to enhance table organisation.
- Apply rowspan to merge cells vertically across multiple rows.
- Use colspan to merge cells horizontally across multiple columns.
- Create complex table layouts by combining various HTML attributes.

BEFORE THE LESSON

- Open the embedded playground yourself and run the basic table code first, so you can confirm it renders and know where the Run Code button sits.
- Have the finished table code from the final step to hand, so you can help a student whose cell counts have drifted.

Crafting Complex Tables

HOW THE LESSON RUNS

- Create a basic table
- Add a header row
- Add Rowspan
- Add colspan
- Add a footer row
- Review your table

TEACHING MOVES

- 1. Create a basic table. Have students type the table rather than paste it, so the tr and td nesting registers. Point out the border="1" is only there to make the grid visible while they work.
- 2. Add a header row. Draw out that th differs from td by rendering bold and centred, and that it carries meaning, not just style. Stress the header row goes above the existing rows, inside the table.
- 3. Add Rowspan. The key move is the deletion: rowspan="2" only works if the cell it now covers is removed. Have them count cells per row before and after so they see one td disappear.
- 4. Add colspan. Same pairing as rowspan: add colspan="2" and delete the neighbouring cell. Ask them to predict what happens if they forget to delete, then let them try it and see the row overflow.
- 5. Add a footer row. Point out colspan="3" makes one cell fill the whole width, which is why a summary or total row uses it. Confirm it sits after the last data row, still inside the table tag.
- 6. Review your table. Have students diff their code against the model line by line rather than just checking the picture looks right. A table can render close to correct while the cell counts are wrong.

WHAT TO WATCH FOR

- Students add rowspan or colspan but leave the cell it should cover in place, so the row gains an extra cell and the grid overflows. Teach the rule as a pair: every span added is a cell deleted. Have them count cells in the affected rows before and after each edit.
- Students judge the table correct because it looks roughly right in the preview, missing a misplaced or extra cell. Ask them to trace the code, not the picture: count td and th tags per row and match against the intended layout.
- Students put the header or footer row outside the table element, or nest td directly without a tr. Reinforce the nesting: table holds tr, tr holds th or td. Get them to indent their code so the levels are visible.

DIFFERENTIATION

- Emerging: If the playground is fighting them, give the working basic table from step one and have them make only the header change first. Pair with a student who has a clean grid rendering so they start from something correct.
- Developing: Ask them to explain aloud why a cell was deleted when a span was added before they move to the next step.
- Proficient: Challenge them to build a small real table, such as a class timetable, that uses both rowspan and colspan meaningfully. Introduce the semantic thead, tbody and tfoot elements as the Higher Level way to structure the same table, and have them refactor theirs.