

HTML Basic Elements

20 minutes

Introduction to Web Frontend

Outcome 1.22

Outcome 1.23

Outcome 2.15

Outcome 2.20

Outcome 3.3

WHAT THIS LESSON DOES

In this step-by-step lesson, you'll learn the basics of HTML. You'll explore how to create headings, paragraphs, breaks, images, and links. Finally, you'll apply your new skills to write your own HTML code, incorporating all these elements. Get ready to dive into the world of web development!

CURRICULUM OUTCOMES

Outcome	What the specification asks for
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
2.20	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 purpose and hierarchy of heading tags in structuring web content.
- Master the use of paragraph tags to organise text on a webpage.
- Apply line break tags to control text formatting and layout.
- Incorporate images into webpages using appropriate attributes for size and source.
- Create functional hyperlinks to connect to external content with specified attributes.

BEFORE THE LESSON

- Run each code sample yourself in the in-browser editor and confirm the codingireland.ie image URLs still load through the school filter; have a fallback image link ready if they are blocked.
- Check that outbound links open (target="_blank") without the filter blocking new tabs.

HTML Basic Elements

HOW THE LESSON RUNS

Headings

Paragraphs

Break

Images

Links

Write some HTML

TEACHING MOVES

- 1. Headings. Stress that h1 to h6 signal importance and structure, not just size. Ask why you would not just make text bigger with a paragraph. One h1 per page is the convention worth planting now.
- 2. Paragraphs. Point out the browser adds space above and below each p automatically, so students should not add blank lines in the code to space text out.
- 3. Break. Contrast br with p: br is a single line break inside a block, p is a new paragraph. Show the stacked-br example so they see line breaks are not the tool for spacing sections.
- 4. Images. Name src, width and height as attributes living inside the tag. Demonstrate a broken src so they see the missing-image icon and learn the URL is the usual culprit.
- 5. Links. Show href as the destination and the text between the tags as what the user clicks. Compare target="_blank" against the default to prove one opens a new tab and one does not.
- 6. Write some HTML. Circulate as they build. Have each student say aloud which of the five required elements they still need. Show the right-click Copy image link trick on a real page before they start.

WHAT TO WATCH FOR

- Students pick a heading level by how big they want the text to look rather than by its place in the document structure. Reframe headings as an outline: h1 is the title, h2 are sections under it. Tell them styling changes size later, tags carry meaning.
- Using repeated br tags to create vertical space between blocks of content instead of using paragraphs. Show both approaches side by side and explain that p groups related text as a unit while br only nudges a line down.
- A broken image is assumed to be a code error when the src URL is simply wrong or blocked. Have them check the URL loads in a plain browser tab first, then check the quotes and spelling in the src attribute.

DIFFERENTIATION

- Emerging: Give a two-line skeleton with a heading and an empty paragraph already typed so they only fill in content. Let them copy one sample tag at a time and run it before adding the next.
- Developing: Ask them to explain what each attribute does before they run their image or link code.
- Proficient: Challenge them to nest a link inside a paragraph, or make an image itself clickable by wrapping it in an anchor tag, then predict the result before running.