Your first web page

A practical beginner guide to headings, paragraphs, links, and saving your own HTML page. No coding experience or paid enrolment is required.

Use this with the interactive lesson:
academy.aurumdigitalconsulting.com/learn/fsp-101

What you will build

A page with a main heading, a short introduction, and a useful link. You will change the example text, check your understanding, and download a working HTML file. This is one introductory lesson; the wider FSP 101 programme is still being prepared.

1. Understand what the browser reads

A browser displays web documents. HTML describes their content and structure. A heading introduces a page, a paragraph holds a sentence, and a link takes the reader somewhere useful. HTML is markup; it does not make decisions or calculate values like JavaScript.

An element usually has an opening tag, its content, and a closing tag. The words between the tags are what the reader sees. A slash identifies the closing tag.

```
<h1>Hello, world!</h1>
```

2. Write a main heading

Open the lesson and replace "Hello, world!" with "My learning journal". Keep both h1 tags. A descriptive main heading helps the reader understand the purpose of the page.

```
<h1>My learning journal</h1>
```

Explain it: Which part is the opening tag? Which part is the visible text? What tells you that the last tag closes the element?

3. Add a paragraph

Move to the next chapter. A paragraph uses the p element. Add a sentence about why you want to learn. Changing the words does not require changing the tags.

```
<h1>My learning journal</h1>
<p>I want to build useful tools for my community.</p>
```

The live preview updates while you type. If the paragraph does not appear as expected, check the closing tag and look for a missing bracket. A spelling mistake in the visible sentence changes the text; a mistake in a tag can change the structure.

4. Link to a useful resource

The a element creates a link. Its href attribute contains the destination. The text between the tags tells the reader what they will find. Use a complete address beginning with https:// in this exercise.

```
<a href="https://developer.mozilla.org/">
  Read web documentation
</a>
```

The quotation marks surround the address. "Read web documentation" is more informative than "Click here". The lesson preview blocks scripts and external images, so use it for this simple markup exercise.

Try it: Change the link text while keeping the destination. Then choose a different useful public resource and change both the address and text. Can another person predict where the link leads?

5. Check your understanding

Keep a heading, paragraph, and link in your code. Select the attribute that tells the browser where the link leads, then choose "Check my work". The checkpoint looks for actual elements with meaningful text and a complete http or https destination.

A passing checkpoint is practice feedback. It is not a certificate or a completed module qualification. Before moving on, explain the purpose of all three elements in your own words.

6. Save and open your page

Choose "Download my HTML page" in the lesson. Find the downloaded file and open it in your browser. An HTML file can work on your computer without being published on the internet.

Your edits are saved in the same browser when storage is available. When you return, choose "Restore saved work". A different device, private browsing session, or cleared browser data will not contain that saved work. Keep the downloaded HTML file as a separate copy.

Independent practice

Create a page for a small service you would like to build. Give it a descriptive heading, a sentence explaining who it helps, and a link to a relevant public resource. Show it to someone and ask what they think the page is for. Improve the wording based on the answer.

Troubleshooting

The checkpoint will not pass: check that the heading and paragraph contain text, that the link has a complete http or https destination, and that you selected href. Code inside an HTML comment does not create an element.

Narration is unavailable: read the full transcript shown in the lesson. Audio is optional; the text and exercise remain usable.

Saved work is missing: check that you are using the same browser and device. Clearing browser data removes the saved draft. Use your downloaded file as the backup.

What comes next

The next learning goals are document structure, semantic HTML, CSS styling, and JavaScript interaction. The wider programme is being prepared; this pilot does not yet teach those topics.

Keep learning:
MDN learning resources
Aurum Academy

Questions: academy@aurumdigitalconsulting.com