

Making Your Own Website

A beginner's guide to HTML, CSS, and JavaScript — the three building blocks behind every page on the web, including this one.

What a website actually is

A website is just a folder of text files that a browser knows how to read. There's no hidden magic beyond that — this portfolio, for example, is a handful of plain files sitting in a folder. HTML gives the page its structure, CSS gives it a look, and JavaScript gives it behavior. That's the entire stack.

HTML — the structure

HTML (HyperText Markup Language) describes what's on the page — headings, paragraphs, images, buttons — using tags that wrap around content.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My First Page</title>
  </head>
  <body>
    <h1>Hello, world!</h1>
    <p>This is my very first web page.</p>
  </body>
</html>
```

Save that as `index.html` and double-click it. No installation, no account, no server required — that's a real website already.

CSS — the look

CSS (Cascading Style Sheets) controls color, spacing, fonts, and layout — everything about how a page looks, without touching the HTML itself.

```
body {
  background-color: #12151C;
  color: #F1ECE1;
  font-family: sans-serif;
}
h1 {
  color: #C9A227;
}
```

Link a CSS file from the <head> of your HTML with one line:

```
<link rel="stylesheet" href="styles.css">
```

JavaScript — the behavior

JavaScript is what makes a page *do* things — respond to clicks, show and hide content, fetch data, validate a form. It's the only one of the three that can actually make decisions while the page is open.

```
document.querySelector("button").addEventListener("click", () => {
  alert("You clicked me!");
});
```

Link it near the very end of the page, right before </body>:

```
<script src="script.js"></script>
```

Seeing it in the browser

You don't need a server or a build tool to get started — just open the HTML file directly in a browser. For something a little closer to how a real site is served, run this from the folder in a terminal:

```
python3 -m http.server 8080
```

then visit `http://localhost:8080`. This is, quite literally, how this portfolio works: plain files, no framework, no build step.

Where to go from here

- Change colors or text in an existing page and refresh to see it update — the fastest way to learn is to break something small and figure out why.

- Learn one tag or property at a time. You don't need to memorize HTML/CSS; you'll pick up what you actually use.
- Right-click any page you like and choose "View Page Source" to see how it's built.
- Read the JavaScript guide next, once a page can stand on its own without moving.

"This portfolio was built exactly this way — no frameworks, one HTML file per page, plain CSS and JS. If you can read this document, you can read that code."

From the Notes & Guides section of leocebillo.github.io — written to explain, not just to answer.