<!DOCTYPE html>

English edition



ZERO TO BUILDER

The HTML + CSS book that shows you the **result of every line**, and teaches you to **build** real websites.

WHAT'S INSIDE

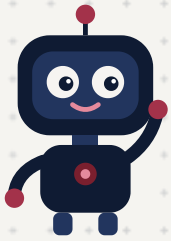
- Every line explained
- Live result under each example
- 8 real projects
- Quizzes in every chapter
- Exercises at 3 levels
- Flexbox, Grid & responsive
- Final exam + answer key
- Oops corners: common mistakes
- Source code + solutions

Elmasryx

Beginner friendly · No experience needed · English + Arabic editions

CHAPTER 00 · MEET HTML & CSS

Every website you've ever seen starts here.



BIT · YOUR GUIDE

Hi, I'm **Bit**. I'll be on every page with you. In this book, you'll **see** the result of every piece of code right under it. Deal?

A web page is built from **three** languages working together. This book teaches the first two, the ones **every** web page needs:

Language	Its job	Think of it as…
HTML	The **content and structure**: headings, text, images, links, buttons, forms.	The **skeleton** of a body
CSS	The **look**: colors, fonts, spacing, layout, animations, mobile design.	The **clothes and style**
JavaScript	The **behavior**: what happens when you click, type, or scroll.	The **muscles** (another book)

Fact	Details
HTML	Created by **Tim Berners-Lee** at CERN. The first website went live on 6 August 1991.
CSS	Proposed by **Håkon Wium Lie** in 1994. The first official CSS standard came in 1996.
Are they programming languages?	Not exactly. HTML is a **markup** language (it labels content). CSS is a **style sheet** language. You don't write logic, you **describe**.

97.7% of all websites use HTML, and almost all of them (99.4%) use CSS too.

Source: W3Techs, September 2026 (HTML) and August 2026 (CSS among HTML5 sites)

CHAPTER 00 · WHAT YOU CAN BUILD

What you can do with HTML & CSS, and what you can't.

Websites
Personal pages, company sites, portfolios, blogs, landing pages.

Online stores
Every product page and checkout is HTML + CSS underneath.

Emails
Newsletters and designed emails are written in HTML.

Documents
Even this book was designed with HTML + CSS, then saved as PDF.

App screens
Many desktop and mobile apps show their screens with web tech.

Simple games
With JavaScript added, pages become games and tools.

✅ With HTML & CSS you can...

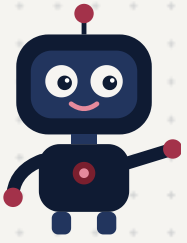
- Build **any page you see** on the internet: text, images, menus, cards.
- Make it **beautiful**: colors, fonts, shadows, animations.
- Make it work on **phones, tablets, and laptops** (responsive design).
- Start **freelancing**: landing pages and simple sites are always in demand.

⚠️ Without other tools, you can't...

- **Calculate or decide** things (if/else, math): that's JavaScript.
- **Save data** (users, orders, passwords): you need a back-end and a database.
- **Log users in** or take payments by yourself.
- Build a full app alone. HTML + CSS are the **first step**, and the most important one.

BIT'S TIP

Good news: HTML and CSS are the **easiest** place to start coding. You see the result instantly, in your browser.



CHAPTER 00 · HOW A PAGE APPEARS

From a text file to a page on the screen.

HTML and CSS are just **text files**. The **browser** (Chrome, Edge, Safari, Firefox) reads them and draws the page. Nothing to install, nothing to compile.

`index.html`
1. You write
Content and structure, in a text file ending in `.html`.

→

`style.css`
2. You style
Colors, sizes, and layout, in a file ending in `.css`.

→

`Browser`
3. It reads
It reads the HTML from top to bottom and applies the CSS rules.

→

`Page`
4. You see it
Change the file, refresh (F5), see the change.

The page is a family tree

The browser turns your HTML into a **tree** called the **DOM**. Every tag sits **inside** another one, like boxes inside boxes:

```
<html>
  <head>
    <title>My site</title>
  </head>
  <body>
    <h1>Welcome</h1>
    <p>Hello there.</p>
  </body>
</html>
```

Part	What it holds	Visible?
`<html>`	Everything. The root of the tree.	—
`<head>`	Information **about** the page: title, CSS link, settings.	No (the title shows on the tab)
`<body>`	Everything the visitor **sees**.	Yes

FAMILY WORDS

`<h1>` is a **child** of `<body>`. `<h1>` and `<p>` are **siblings**. CSS and JavaScript use this family language a lot.

CHAPTER 00 · SETUP

Ready in 5 minutes.

You need only two free things: a **browser** (you already have one) and an **editor** to write code.

1

Install VS Code

Download it free from **code.visualstudio.com** and install it.

2

Add Live Server

In VS Code, open **Extensions** (the 4 squares icon), search **Live Server**, and install it. It refreshes your page **automatically** every time you save.

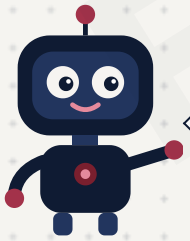
3

Make your project folder

Create a folder called `my-site`. Open it in VS Code (**File → Open Folder**). Inside, create a file named `index.html`. Right-click it → **Open with Live Server**.

Your best friend: DevTools

In Chrome or Edge, right-click any part of any website → **Inspect**. You see its HTML and CSS, and you can change them live to test ideas. Nothing is saved, so you can't break anything.



BIT'S TIP

Type `!` in an empty `.html` file in VS Code and press **Enter**. It writes the whole page skeleton for you.

NO LAPTOP RIGHT NOW?

Search "CodePen" or "online HTML editor". You can run every example in this book from your phone.

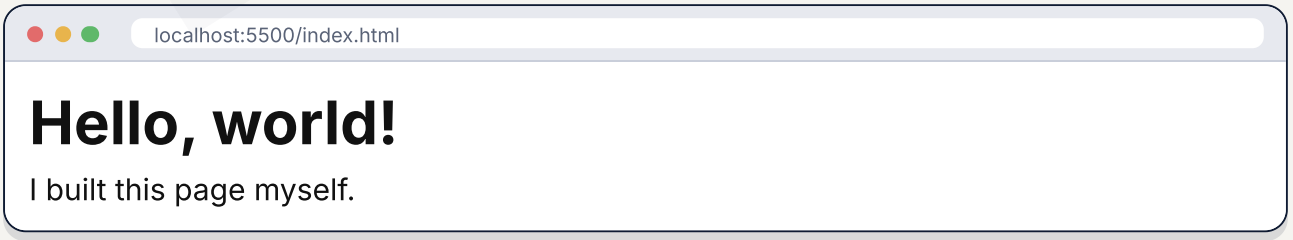
Your first page. Every line explained.

INDEX.HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My first page</title>
</head>
<body>
  <h1>Hello, world!</h1>
  <p>I built this page myself.</p>
</body>
</html>
```

LINE BY LINE

1 `<!DOCTYPE html>`	Tells the browser: “this is modern HTML (HTML5)”. Always the very first line.
2 `<html lang="en">`	The root of the page. `lang` = the page language (helps Google and screen readers).
3 `<meta charset="UTF-8">`	Lets the page show **every** character correctly, including Arabic and emoji.
4 `<title>My first page</title>`	The text on the browser **tab** and in Google results.
5 `<h1>Hello, world!</h1>`	The main heading. `<h1>` **opens**, `</h1>` **closes** (note the `/`).
6 `<p>I built this page myself.</p>`	A paragraph of text.



OOPS CORNER



Forgot to close a tag? HTML doesn’t show an error. It just **guesses**, and the page looks strange.

```
<h1>Hello, world! (missing </h1>: the rest of the page becomes a giant heading)
```

Fix: every tag that opens must close, except a few “empty” ones like `<img>` and `<br>`.

CHAPTER 01

01 HTML Basics

Everything in HTML is an **element**. Learn the shape of one element, and you can read any web page.

ANATOMY OF AN ELEMENT	
`1 <a href="about.html">About me</a>`	The whole thing is one **element**.
`2 <a`	**Opening tag**: the element's name (`a` = link) between `< >`.
`3 href="about.html"`	An **attribute**: extra information. Always `name="value"`, inside the opening tag.
`4 About me`	The **content**: what the visitor sees.
`5 </a>`	**Closing tag**: same name with a `/`.

Headings and paragraphs

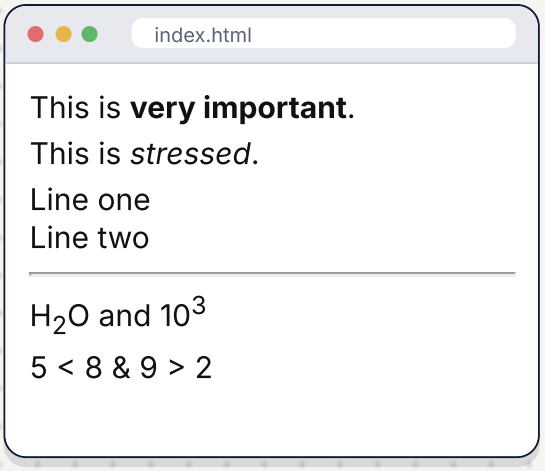
```
<h1>Main title</h1>
<h2>Section title</h2>
<h3>Smaller title</h3>
<p>A paragraph is a
block of text.</p>
<p>Each one starts on
a new line.</p>
```



1. There are 6 heading levels: `<h1>` (biggest) to `<h6>`. Use **one** `<h1>` per page, for the main title. Google reads it to understand your page.
2. The browser **ignores** extra spaces and line breaks in your code. The line break inside the `<p>` doesn't show.

Bold, italic, and line breaks.

```
<p>This is <strong>very important</strong>.</p>
<p>This is <em>stressed</em>.</p>
<p>Line one<br>Line two</p>
<hr>
<p>H<sub>2</sub>O and 10<sup>3</sup></p>
<p>5 &lt; 8 &amp; 9 &gt; 2</p>
<!-- A note for you. Not shown. -->
```



Tag	Meaning	Shows as
`<strong>`	Important text	**bold**
`<em>`	Stressed (emphasis)	*italic*
` `	Line break inside text. **Empty**: no closing tag.	new line
`<hr>`	A topic change. Empty tag.	a line across
`<sub> <sup>`	Below / above the line	H_2O, 10^3
`<!-- -->`	Comment: a note for developers	nothing

SPECIAL CHARACTERS

`<` and `>` are special in HTML (they start tags). To **show** them, write `<` and `>`. And `&` shows `&`. These are called **entities**.

PRO RULE

Use `<strong>` for **meaning**, not for looks. To make text bold just for style, use CSS (Chapter 4).