Learn to Code with Your AI Sidekick: Building Your First Website

Welcome to the exciting world of web development! You're about to learn how to build websites from scratch. But you won't be doing it alone. This guide is built around a revolutionary idea: using Artificial Intelligence (AI) like Gemini or ChatGPT as your personal tutor, coding partner, and creative assistant.

Think of your AI as a sidekick who is available 24/7, never gets tired of your questions, and can explain complex topics in a thousand different ways until you get it. Throughout this book, we'll not only learn the essential languages of the web—**HTML, CSS, and JavaScript**—but we'll also learn *how* to ask an AI for help. This skill is just as valuable as the code itself.

Let's get started! 🚀

Chapter 1: HTML - The Skeleton of Your Website

Every website you've ever visited is built on a foundation of HTML. It stands for **HyperText Markup Language**. Think of it as the **skeleton** of a webpage. It provides the structure and content, like headings, paragraphs, images, and links.

Objectives

- Understand what HTML is and its role.
- Learn to write basic HTML tags for text, images, and links.
- Create your very first HTML file and view it in a web browser.
- Learn how to use AI to explain HTML concepts and generate code.

Core Concepts: Tags and Elements

HTML uses "tags" to define different types of content. Most tags come in pairs: an opening tag like `<p>` and a closing tag like `</p>`. Everything between them is an **element**.

- `<h1>` to `<h6>`: Headings (1 is the most important, 6 is the least).
- `<p>`: A paragraph of text.
- `<img>`: An image. This one is self-closing!
- `<a>`: A link (the 'a' stands for anchor).

🧠 AI-Powered Learning: HTML

Your AI assistant is fantastic for clarifying concepts. Don't just ask it for code; ask it to *explain*. This is how you truly learn.

Pro AI Tips & Prompts:

- **Ask for analogies:** "Explain the difference between an HTML tag and an element using a house analogy."
- **Explore options:** "What are the most common HTML tags a beginner should know? List them in a table with a simple explanation for each."
- **Get code snippets:** "Give me the HTML code for a link that opens in a new tab and links to Google."

Exercise: Create Your Personal Portfolio Page

We'll build a simple `index.html` file. This will be the foundation of our project.

Step 1: Set Up Your Files

1. Create a new folder on your computer named `my-first-website`.
2. Open a plain text editor (like Notepad on Windows or TextEdit on Mac; VS Code is a great free option for coders).
3. Create a new file and save it inside your new folder as `index.html`. Make sure the file extension is `.html`, not `.txt`.

Step 2: Write the HTML Code

Copy and paste the following code into your `index.html` file.

HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Laurence's Portfolio</title>
</head>
<body>

  <h1>Welcome to My Website</h1>

  <p>Hi, I'm Laurence. I'm learning to build amazing websites with HTML, CSS, and JavaScript. This is my very first project!</p>

  <img src="https://via.placeholder.com/150" alt="A placeholder image">

  <p>You can find my work on <a href="#">my GitHub profile</a>.</p>

</body>
</html>
```

Step 3: View Your Creation

Find the `index.html` file in your folder and **double-click it**. It will open in your web browser, and you'll see your first webpage!

Code Explanation

- `<!DOCTYPE html>`: This tells the browser it's an HTML5 document. It's always the first line.
- `<html>`: The root element that wraps everything else.
- `<head>`: Contains meta-information about the page, like the title that appears in the browser tab.
- `<title>`: Sets the title of the browser tab.
- `<body>`: Contains all the visible content of the page.
- `<h1>`: Your main heading.
- `<p>`: A paragraph of text.
- `<img>`: An image tag. `src` is the **source** (where the image comes from) and `alt` is the **alternative text** for screen readers or if the image fails to load.
- `<a>`: A link tag. `href` is the **hyperlink reference** (the destination URL). We've used `"#"` as a placeholder for now.

Chapter 2: CSS - Styling Your Masterpiece

Your website has a skeleton (HTML), but it has no style! CSS, or **Cascading Style Sheets**, is the language we use to add color, fonts, layout, and personality. It's the **clothing and appearance** of your website.

Objectives

- Understand what CSS is and how to link it to your HTML.
- Learn to write CSS rules using selectors and properties.
- Style your portfolio page with colors, fonts, and spacing.
- Use AI to debug CSS issues and get design ideas.

Core Concepts: Selectors and Properties

A CSS rule has two main parts: a **selector** (which HTML element to target) and **declarations** (the styles to apply). Declarations are made of a `property` and a `value`.

CSS

```css
/* This is a CSS comment */
```

```
selector {
  property: value;
}


/* Example: Make all paragraphs red */
p {
  color: red;
}
```

We can also use **classes** to style specific elements. In HTML, you'd add `class="my-class"`, and in CSS, you'd target it with `.my-class`.

🧠 AI-Powered Learning: CSS

CSS can be tricky. Centering elements is a classic challenge. Your AI is an expert at solving these problems.

Pro AI Tips & Prompts:

- **Debugging:** "My image is too big for the page. Give me the CSS to make sure it never goes outside its container."
- **Design Ideas:** "I want a modern and clean look for my portfolio. Suggest a color palette and font pairing for me."
- **Understanding Concepts:** "Explain the CSS box model like I'm building with LEGOs."
- **Code Generation:** "Write the CSS to style a button with a blue background, white text, rounded corners, and no border."

Exercise: Add Style to Your Portfolio

Let's make our page look much better.

Step 1: Create Your CSS File

1. In the same `my-first-website` folder, create a new file named `styles.css`.
2. This is where all our style rules will go.

Step 2: Link CSS to HTML

You have to tell the HTML file where to find the styles. Add this line of code inside the `<head>`

section of your `index.html` file, right after the `<title>` tag.

HTML

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

Step 3: Write the CSS Code

Copy and paste the following code into your `styles.css` file.

CSS

```
/* Style the entire body of the page */
body {
   font-family: Arial, sans-serif;
   background-color: #f4f4f4;
   color: #333;
   line-height: 1.6;
   margin: 20px;
}

/* Style the main heading */
h1 {
   color: #0056b3;
}

/* Add a class for our container to center content */
.container {
   max-width: 800px;
   margin: auto;
   background: white;
   padding: 20px;
   border-radius: 8px;
}
```

```
/* Style links */
a {
  color: #0056b3;
  text-decoration: none; /* Removes the underline */
}

/* Change link color on hover */
a:hover {
  color: #007bff;
  text-decoration: underline;
}
```

Step 4: Update the HTML to Use the Class

We created a `.container` class in our CSS. Let's use it! In your `index.html` file, wrap all the content inside the `<body>` tag with a `<div>` that has this class. Your `<body>` should now look like this:

HTML

```
<body>
  <div class="container">
    <h1>Welcome to My Website</h1>

    <p>Hi, I'm Laurence. I'm learning to build amazing websites with HTML, CSS, and JavaScript. This is my very first project!</p>

    <img src="https://via.placeholder.com/150" alt="A placeholder image">

    <p>You can find my work on <a href="#">my GitHub profile</a>.</p>
  </div>
</body>
```

Step 5: See the Difference!

Save both files and refresh your `index.html` page in the browser. It should look much more professional!

Chapter 3: JavaScript - Bringing Your Site to Life

We have structure (HTML) and style (CSS). Now we need behavior. JavaScript (JS) is the **brain** of your website. It handles interactivity, like responding to button clicks, validating forms, and creating animations.

Objectives

- Understand what JavaScript is for.
- Learn how to add JavaScript to your webpage.
- Write a simple function to react to a user action (a click).
- Use AI to write and explain simple JavaScript functions.

Core Concepts: Functions and DOM Manipulation

- **Functions:** A block of code designed to perform a particular task. A function is "called" or "invoked" when you want it to run.
- **The DOM (Document Object Model):** When a webpage is loaded, the browser creates a "model" of the page. JavaScript can interact with this model to change HTML and CSS content on the fly. We can grab an element by its ID and change it.

🧠 AI-Powered Learning: JavaScript

JavaScript introduces logic, which can be a hurdle. Use your AI to break down problems into smaller steps.

Pro AI Tips & Prompts:

- **Explain Errors:** If you see an error in the browser console, copy it and ask: "I'm getting this JavaScript error: [paste error here]. What does it mean and how can I fix it in this code: [paste code here]?"
- **Generate Functions:** "Write a JavaScript function that takes two numbers as arguments and returns their sum."
- **DOM Questions:** "Explain what `document.getElementById()` does in JavaScript."

- **Step-by-Step Logic:** "I want to make a button that shows a hidden message when clicked. What are the steps I need to take in HTML, CSS, and JavaScript to make this work?"

Exercise: Add an Interactive Button

Let's add a button that reveals a "secret" message when clicked.

Step 1: Create Your JavaScript File

1. In your `my-first-website` folder, create a new file named `script.js`.

Step 2: Link JS to HTML

Just like with CSS, we need to connect our JS file. Add this line to your `index.html` file right before the closing `</body>` tag. It's best practice to put scripts at the end so the page content loads first.

HTML

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

Step 3: Add the HTML for the Button and Message

In your `index.html` file, add this code right before the `<p>` with the link.

HTML

```
<button id="toggleButton">Show Contact Info</button>


<p id="contactInfo" style="display: none;">My email is example@email.com</p>
```

- We give each element an **ID** so JavaScript can find it easily.
- We use an inline CSS style `display: none;` to hide the contact info paragraph initially.

Step 4: Write the JavaScript Code

Copy and paste the following code into your `script.js` file.

JavaScript

```
// This tells the browser to wait until all HTML is loaded before running the script
```

```
document.addEventListener('DOMContentLoaded', function() {

  // Get the button and the contact info paragraph by their IDs
  const toggleButton = document.getElementById('toggleButton');
  const contactInfo = document.getElementById('contactInfo');

  // Add a "click" event listener to the button
  toggleButton.addEventListener('click', function() {

    // Check if the contact info is currently hidden
    if (contactInfo.style.display === 'none') {
      // If it's hidden, show it
      contactInfo.style.display = 'block';
      toggleButton.textContent = 'Hide Contact Info';
    } else {
      // If it's visible, hide it
      contactInfo.style.display = 'none';
      toggleButton.textContent = 'Show Contact Info';
    }

  });

});
```

Code Explanation

1. `document.addEventListener(...)`: This is a safety measure to ensure our code doesn't try to find HTML elements before they exist.
2. `document.getElementById(...)`: This is the key DOM method. It finds the HTML element with the specified `id`.
3. `toggleButton.addEventListener('click', ...)`: This tells the browser to "listen" for a click on our button. When it happens, it runs the function we provide.
4. `if / else`: This is a simple conditional. We check the `display` style of our

paragraph. If it's `none`, we change it to `block` (to make it visible) and update the button's text. Otherwise, we do the opposite.

Step 5: Test the Interactivity!

Save all your files and refresh `index.html` in your browser. Click the button—you should see the message appear and disappear! You've just made an interactive webpage!

Final Quiz: Test Your Knowledge!

See how much you've learned. Choose the best answer for each question.

1. What is the primary purpose of HTML? a) To style the webpage. b) To add interactivity to the webpage. c) To provide the structure and content of the webpage. d) To manage the web server.

---

2. Which tag would you use to create the most important heading on a page? a) `<heading>` b) `<h1>` c) `<p>` d) `<main>`

---

3. How do you correctly link a stylesheet named `theme.css` to an HTML file? a) `<script href="theme.css"></script>` b) `<link rel="stylesheet" href="theme.css">` c) `<style src="theme.css"></style>` d) `<css link="theme.css">`

---

4. Which CSS selector would you use to style an element with `class="highlight"`? a) `#highlight` b) `highlight` c) `.highlight` d) `*highlight`

---

5. What does the CSS property `color` change? a) The background color of an element. b) The border color of an element. c) The text color of an element. d) The shadow color of an element.

---

6. Where is the best place to link your `script.js` file in your HTML? a) Inside the `<head>` section. b) Just before the closing `</body>` tag. c) At the very top, before `<!DOCTYPE`

`html>`. d) Inside a `<p>` tag.

---

7. What is the JavaScript method to find an HTML element with `id="logo"`? a) `document.getElementByName('logo')` b) `document.querySelector('#logo')` c) `document.getElementById('logo')` d) Both b and c are correct.

---

8. What does "DOM" stand for? a) Document Object Model b) Data Object Manipulation c) Direct Operation Method d) Document Orientation Marker

---

9. You want to run a function named `showAlert` when a button is clicked. Which is the correct JavaScript? a) `button.onClick(showAlert);` b) `button.listen('click', showAlert);` c) `button.addEventListener('click', showAlert);` d) `button.whenClicked = showAlert;`

---

10. An AI prompt like "Explain the CSS box model using a pizza box analogy" is a good way to: a) Get the AI to write your whole website for you. b) Understand a complex concept in a simpler, more memorable way. c) Debug an error in your code. d) Choose a color scheme.

Quiz Answers & Explanations

1. **C: To provide the structure and content of the webpage.**
   - **Explanation:** HTML is the skeleton. CSS handles styling (a), and JavaScript handles interactivity (b).
2. **B: `<h1>`**
   - **Explanation:** HTML provides heading tags from `<h1>` to `<h6>`, with `<h1>` being the most important for structure and SEO.
3. **B: `<link rel="stylesheet" href="theme.css">`**
   - **Explanation:** The `<link>` tag is used in the `<head>` to connect external resources. `rel="stylesheet"` tells the browser it's a CSS file, and `href` provides the path.
4. **C: `.highlight`**
   - **Explanation:** In CSS, a period (`.`) is used to select elements by their class name. A hash (`#`) is used to select elements by their ID.
5. **C: The text color of an element.**
   - **Explanation:** The `color` property specifically sets the color of the text content. To change the background, you would use `background-color`.

6. **B: Just before the closing `</body>` tag.**
   - **Explanation:** Placing scripts at the end allows the HTML and CSS to load and render first, so the user sees the page content as quickly as possible. This prevents a blank page from showing while a large script loads.
7. **D: Both b and c are correct.**
   - **Explanation:** `getElementById('logo')` is the traditional and most direct way. `querySelector('#logo')` is a more modern and versatile method that uses CSS selectors to find elements (the `#` indicates an ID). Both work perfectly for this task.
8. **A: Document Object Model**
   - **Explanation:** The DOM is the browser's tree-like representation of a webpage's elements, which JavaScript can manipulate.
9. **C: `button.addEventListener('click', showAlert);`**
   - **Explanation:** `addEventListener` is the standard, modern way to handle events in JavaScript. It allows you to attach multiple functions to the same event if needed.
10. **B: Understand a complex concept in a simpler, more memorable way.**
    - **Explanation:** This is a key skill for learning with AI. Asking for analogies, simple explanations, and different perspectives helps solidify your understanding far more than just asking for code.

Congratulations! You've built your first interactive webpage and learned how to use AI as your personal coding tutor. This is just the beginning of an amazing journey. Keep experimenting, keep asking questions, and keep building