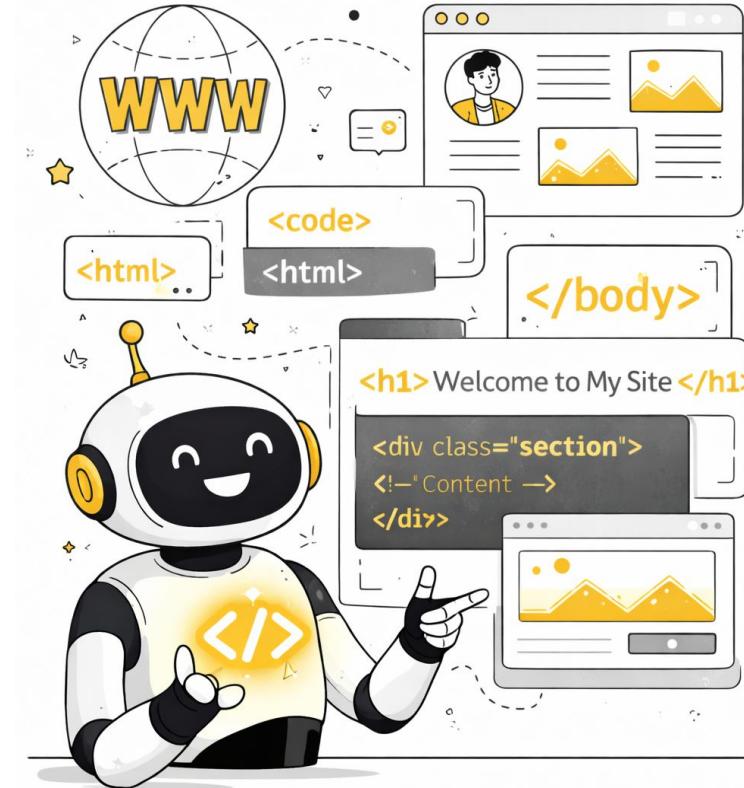


Lesson 03: Introduction to AI-Code Generation

Start

Understand what AI generates, how to read it so you can make small changes confidently in the future



Learning Objectives



Describe a website structure using clear, non-technical language



Identify common website sections (header, navigation, content, footer)



Compare different layout types and their use cases



Choose an appropriate layout based on a portfolio goal



Write a clear structural description suitable for an AI prompt

 These objectives focus on clarity of thinking, not visual design or styling.

Deliverables



✓ A written description of a website structure (no code)



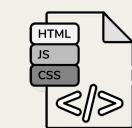
✓ A selected layout type justified by the website's goal



✓ A clear, well-structured AI prompt describing the layout



✓ At least one committed update to the website



✓ A conceptual understanding of how the structure maps to HTML sections



If you can explain your website structure clearly before generating code, you have succeeded.





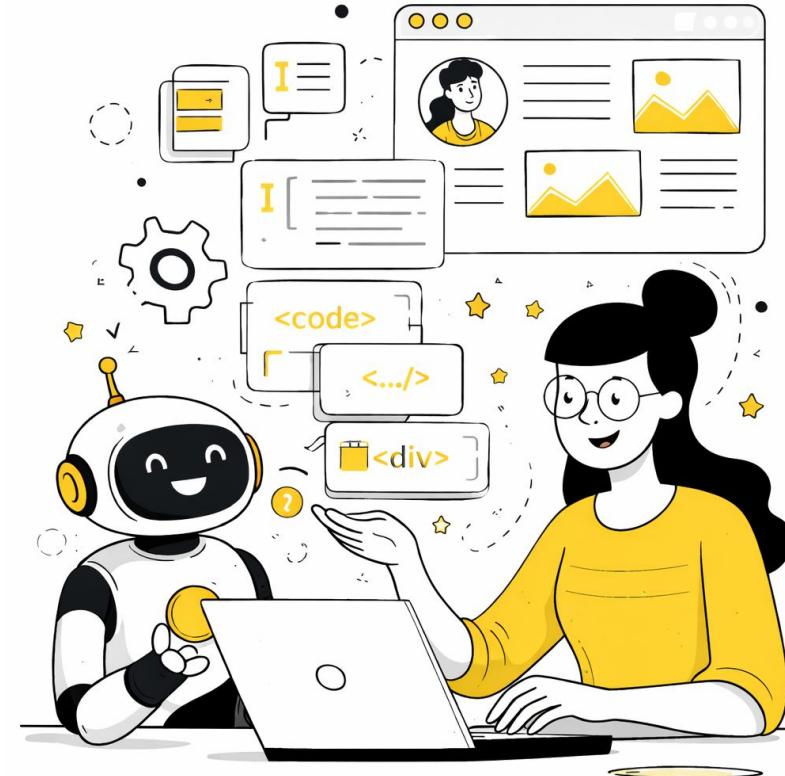
AI Learning Support for This Lesson

In this lesson, the AI tutor will help you think before generating code.

A guided prompt is available to support you while defining your website structure. You can use it to help you:

- Identify the main sections of a website
- Compare different layout types and their use cases
- Describe structure clearly without writing code
- Turn your ideas into a clear structural description
- Prepare better prompts for generating HTML later

 Use the AI to clarify structure and intent — not to generate code yet. **Lesson Prompt**.



Why Structure Matters

Before writing (or generating) code, we must:

- Know what the website is for
- Understand which parts it has
- Decide how content is organized on screen

💡 AI generates better code when:

- The description is clear
- The structure is well defined
- The goal is explicit

Next





Define the Context

Every website starts with a **purpose**

Example context:

- Personal portfolio website
- Hosted using GitHub Pages
- Main goal: showcase professional experience
- First focus: structure of index.html

The **prompt** would be:

"I would like to create a website using Github pages to use it as my personal portfolio to showcase my experience. First I would like to create the overall structure of the index.html."

- ✓ This context guides all future decisions

Next



Thinking in Sections

Most websites are made of **sections**:

- Header / Banner
- Content areas
- Navigation menus
- Footer / Contact section

💡 Think of a website like a **document**:

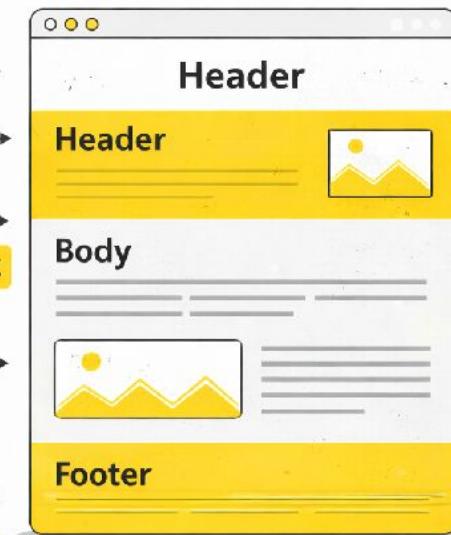
- Title
- Chapters
- Subsections
- Closing information

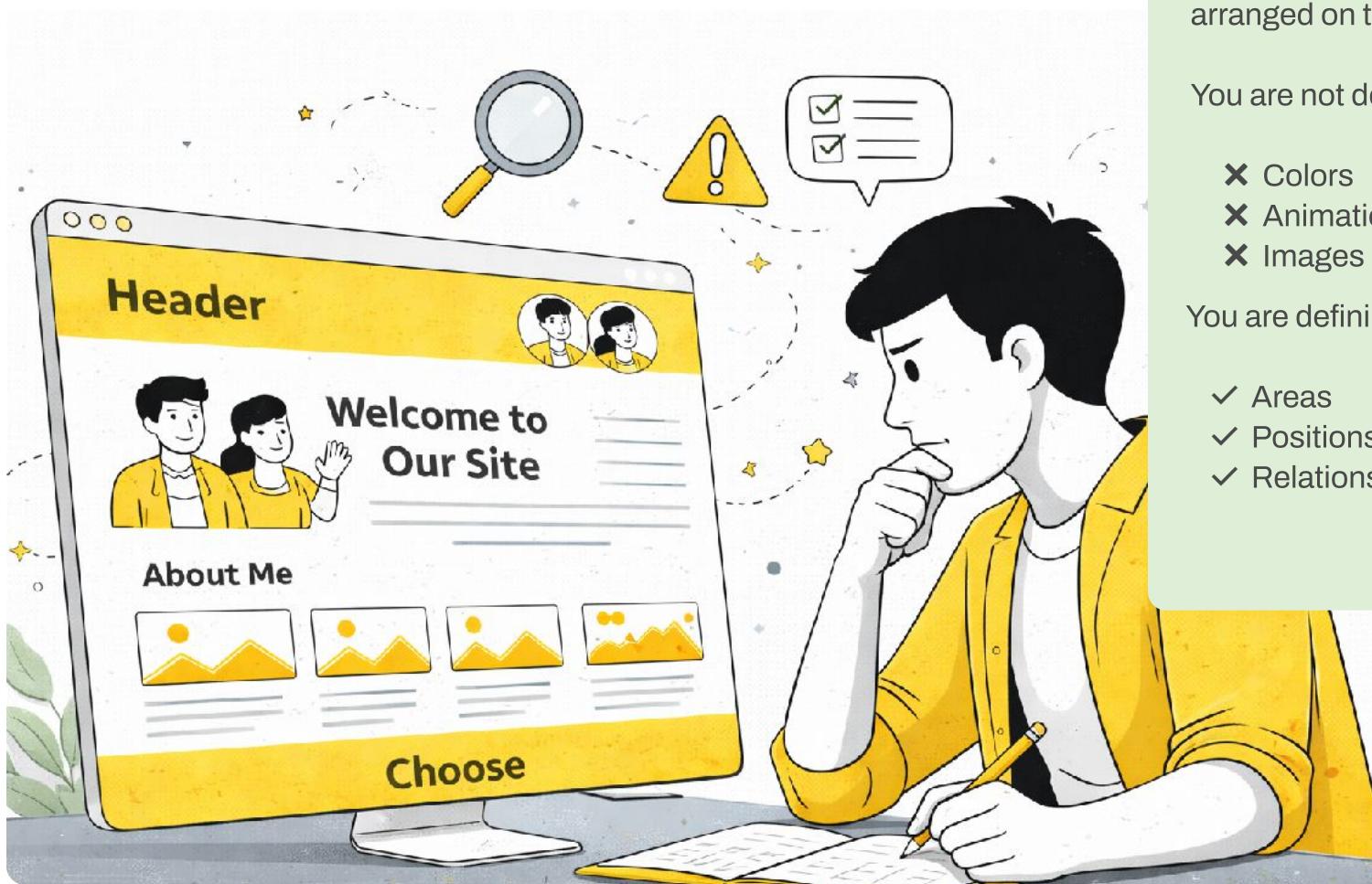
Next

Text Document



Website





Define the Website Layout

A layout describes how sections are arranged on the screen.

You are not defining:

- ✗ Colors
- ✗ Animations
- ✗ Images

You are defining:

- ✓ Areas
- ✓ Positions
- ✓ Relationships between sections

Next

<

>

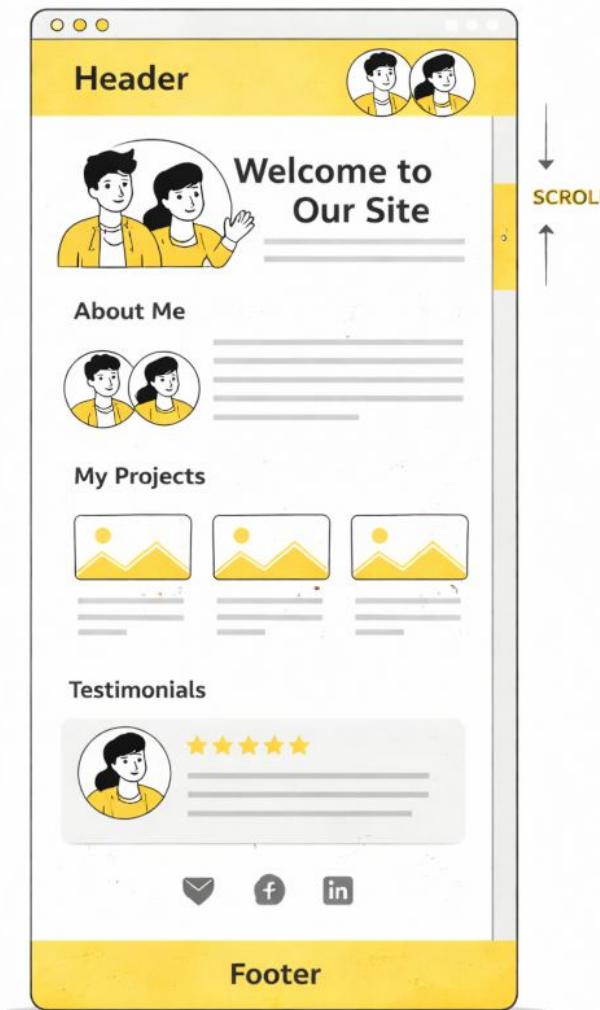


Layout Example 1: Single Area Layout

- A top banner with:
 - Your name
 - Short professional description
- A fixed horizontal menu above the banner
- Four vertical sections below:
 - Area1 Experience
 - Area2 Experience
 - Area3 Experience
 - Area4 Experience

💡 Menu links scroll to each section

Next



A small black circular button with a white 'X' inside, located in the top right corner of the white box.

The prompt would be:

“At the top use a banner where I can present the website using my name “Name Surnames” and a description “I'm a XXXXX XXXXXX specialized in Area1, Area2, Area3 and Area4”

Create 4 separated sections that I should be able to customize later. They are going to be named:

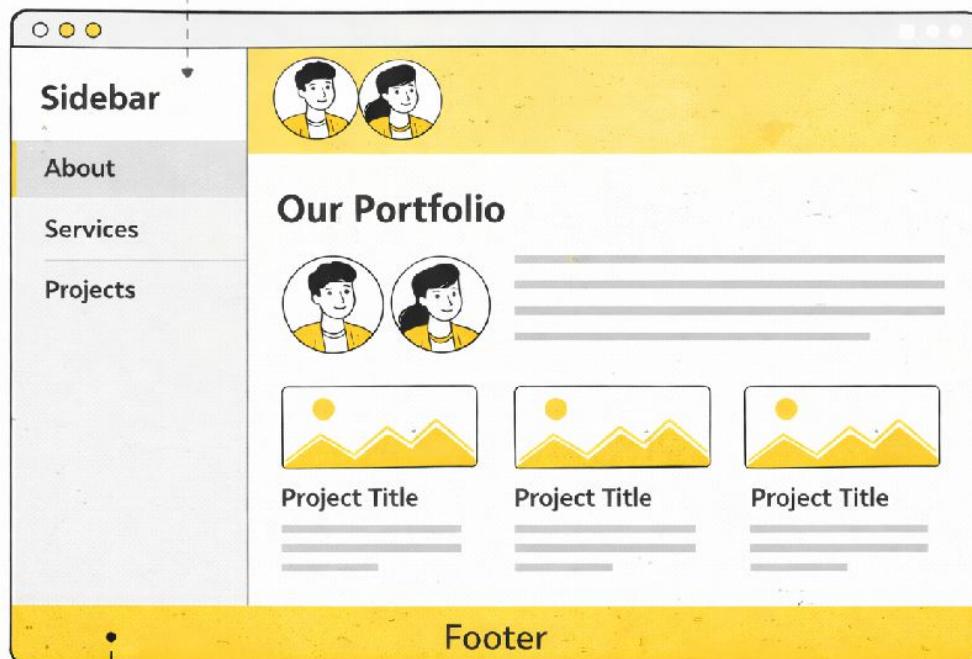
- 1. Area1 Experience*
- 2. Area2 Experience*
- 3. Area3 Experience*
- 4. Area4 Experience*

Define an horizontal fixed menu above the banner that allows us to jump to the different sections.”



Header

Body



Footer

Layout Example 2: Two-Area Layout

Description:

- Fixed top banner with name and description
- Below the banner:
 - Left panel (20% width): navigation menu
 - Right panel (80% width): experience content

Experience sections:

1. Area1 Experience
2. Area2 Experience
3. Area3 Experience
4. Area4 Experience

Next





The prompt would be:

“Define a layout with 2 areas.

A fixed horizontal area at the top will be a banner where I can present the website using my name “Name Surnames” and a description “I'm a XXXXX XXXXXX specialized in Area1, Area2, Area3 and Area4”

Below the banner will be the other area which is composed by the next 2 panels:

On the lateral fixed panel to the left around 20% of the screen place a menu that will allow to scroll between the areas of experience.

On the panel to the right, the remaining 80% of the width of the screen, create 4 different areas of experience:

1. *Area1 Experience*
2. *Area2 Experience*
3. *Area3 Experience*
4. *Area4 Experience””*

Layout Example 3: Grid Layout (5 Areas)

Description:

- Fixed horizontal banner at the top
- Below it:
 - A grid of 4 panels
 - Each panel represents one expertise area

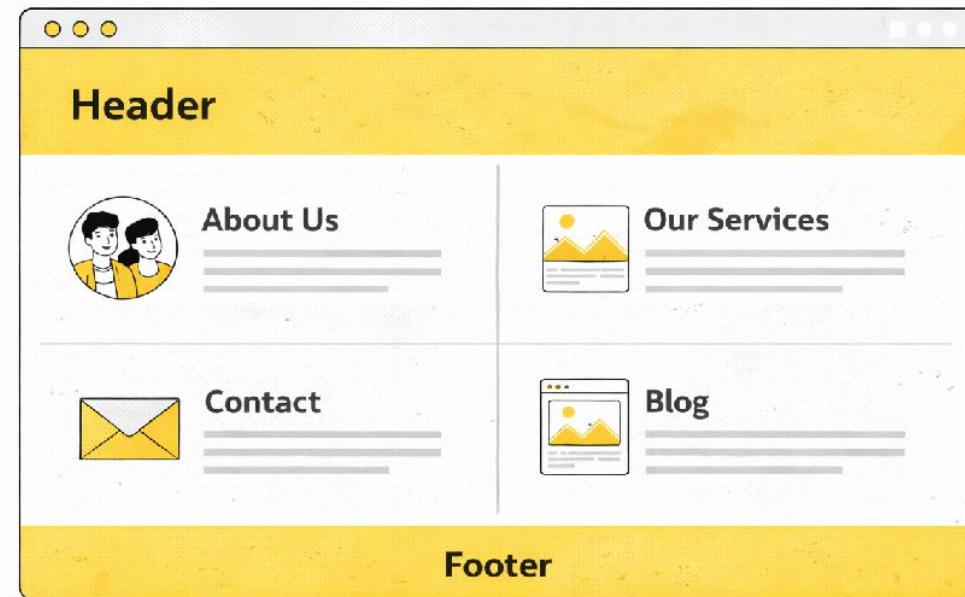
Panels:

1. Area1 Experience
2. Area2 Experience
3. Area3 Experience
4. Area4 Experience

💡 Screen space is used simultaneously, not vertically

Next

Header



A small black circular button with a white 'X' inside, located in the top right corner of the white box.

The prompt would be:

“Define a layout with 5 fixed panels.

A fixed horizontal panel at the top will be a banner where I can present the website using my name “Name Surnames” and a description “I'm a XXXXX XXXXXX specialized in Area1, Area2, Area3 and Area4”

Below the previous panel define 4 panels in a grid taking the rest of the space where we are going to place descriptions for 4 different areas of expertise:

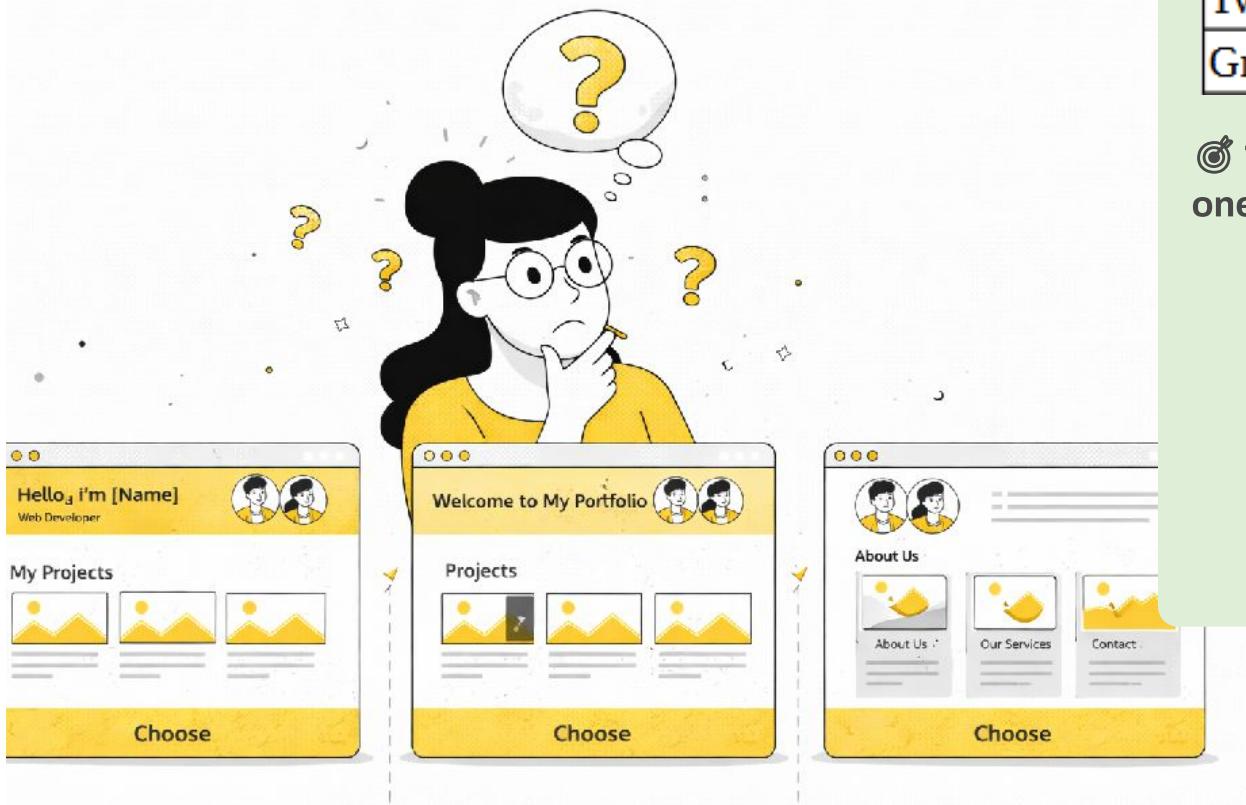
- 1. Area1 Experience*
- 2. Area2 Experience*
- 3. Area3 Experience*
- 4. Area4 Experience”*



Comparing Layout Types

Layout Type	Best For
Single Area	Simple storytelling
Two Areas	Navigation-heavy portfolios
Grid Layout	Visual comparison of skills

⌚ There is no “best” layout — only the right one for your goal



Next



Define a Closing Section

Most websites end with a footer or contact area.

Example:

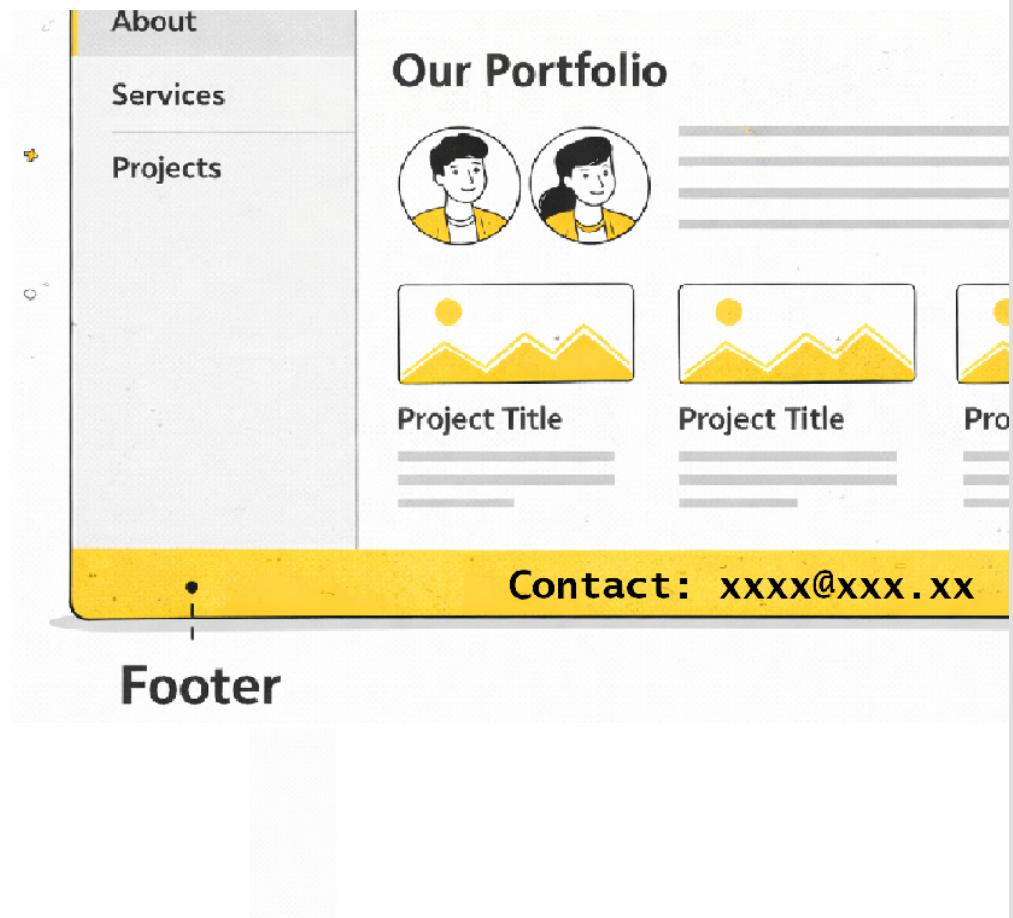
- Contact form
- Email address displayed
- Simple, clear purpose: “Allow visitors to reach out”

This reinforces completeness of the page

The **prompt** would be:

“Next create a contact form so they can reach out to my with the email address “xxxxxx@xxxxxx.xxx”

Next





Visual Inspiration

Before asking AI to generate code:

- Search for portfolio websites you like
- Observe:
 - Main areas
 - Layout structure
 - Content grouping

 Try to describe what you see, not how it's animated

Next



Ask AI to create the layout you want



Description

Describe Designs to AI



Generation

Using AI to Generate
index.html



Commit

Time to update the
index.html in your
repository



A dark circular button with a white 'X' inside, located in the top right corner of the white box.

Once your description is ready, you can use:

- ChatGPT: <https://chatgpt.com/>
- Claude: <https://claude.ai/>
- Gemini: <https://gemini.google.com/>
- Mistral: <https://chat.mistral.ai/>
- DeepSeek: <https://chat.deepseek.com/>

You will:

1. Paste your description
2. Ask for a basic index.html
3. Compare results
4. Choose the one you understand best



Good descriptions focus on:

- Main sections
- Layout type
- Content purpose

Avoid:

- ✗ Animations
- ✗ Icons
- ✗ Colors
- ✗ Interactions

⌚ **AI needs structure, not decoration (yet)**



Key Takeaways:

- Understand website structure conceptually
- Know how to describe layouts clearly
- Be able to read AI-generated HTML without fear
- Be ready for the next lesson: understanding the code itself



Structure first. Code second.

Knowledge Test Questions



Importance of Structure

[Question](#)

Structure vs Design

[Question](#)

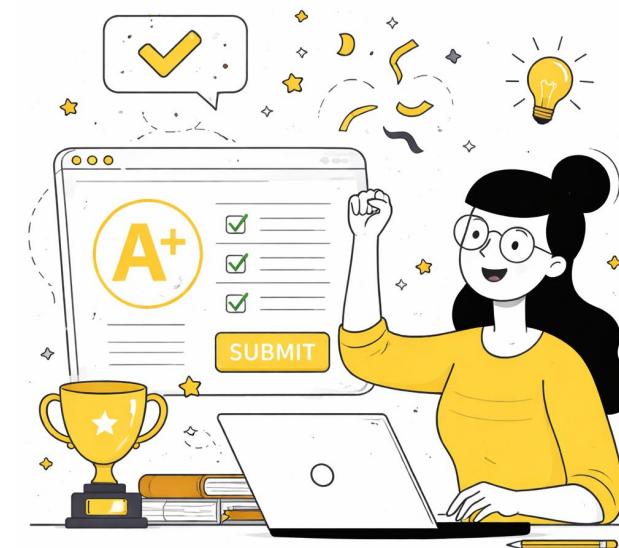
Layout Choice

[Question](#)

AI Prompt Quality

[Question](#)

Understanding Website Structure Before Generating Code



<

>



Why is it important to define a website's structure before generating code with AI?

Because structure automatically defines the final visual design

Because AI cannot generate HTML without colors and images

Because a clear structure leads to better, more usable AI-generated code



Which prompt is more likely to generate clear and understandable HTML from an AI?

“Create a colorful website with animations”

“Create a page with a header, a navigation menu, and four content sections below”

“Make my website look modern and cool”



Which of the following describes website structure rather than visual design?

Adding animations and transitions

Choosing background colors and fonts

Defining sections like header, content areas, and footer



Which layout is generally best suited for a navigation-heavy portfolio website?

A single vertical storytelling layout

A grid layout with multiple panels visible at once

A two-area layout with a navigation panel and content area

Help us to improve

With your feedback we will keep improving for you to get the best results



Which concept in this lesson was the most difficult to understand or apply, and why?

Write your answer here.

Send

Did you feel prepared to describe your own website structure clearly after this lesson?

Write your answer here.

Send

What change would most improve this lesson for future students?

Write your answer here.

Send

<

>

