

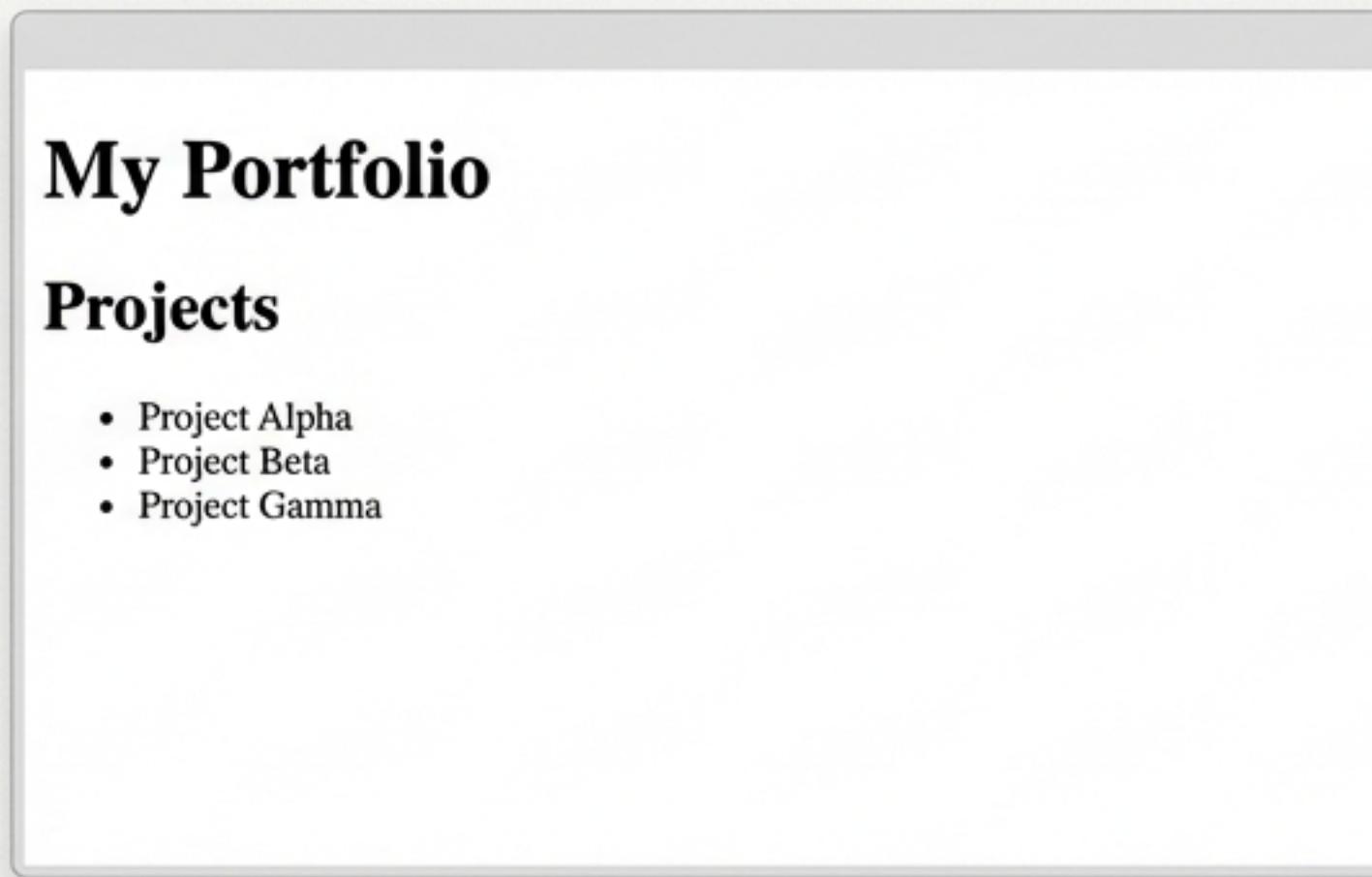
From Structure to Style: An Introduction to CSS

For IIT67-272: Web Application Development

The goal: Make your website beautiful with CSS.

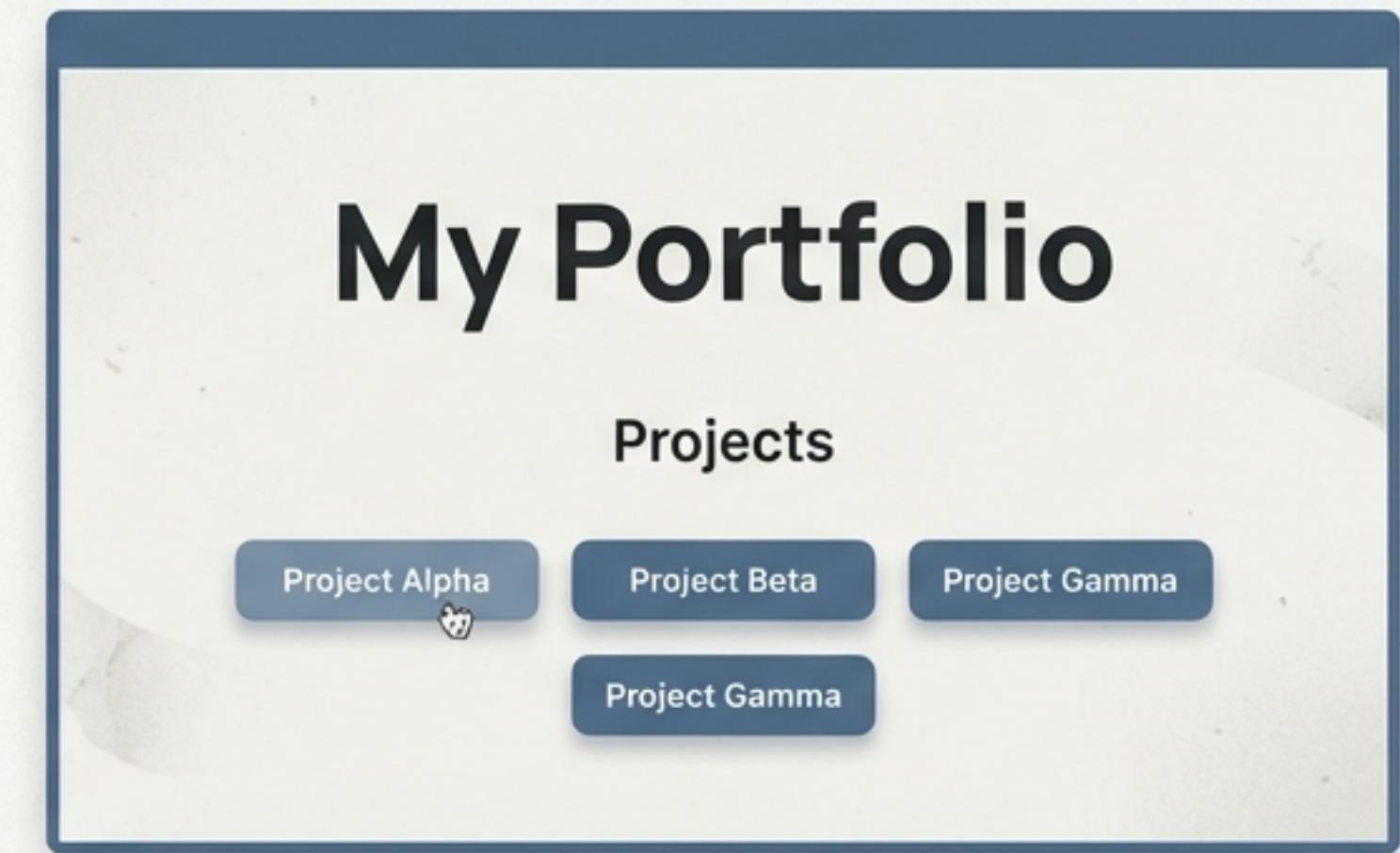
The Difference is Design.

BEFORE - HTML ONLY



The Blueprint.

AFTER - WITH CSS



The Final Design.

HTML gives your website structure. CSS gives it life.

Your Learning Objectives for This Week

- 1.** Explain the distinct roles of HTML and CSS.
- 2.** Use CSS to style fonts, colors, backgrounds, and spacing.
- 3.** Select specific elements for styling using classes and IDs.
- 4.** Connect an external CSS file to your HTML document.
- 5.** Apply these skills to transform your Week 2 project.

The Architect and The Interior Designer

The Architect

HTML



Role: Creates the structure of the site.

```
<p>Welcome</p>
```

The Interior Designer

CSS

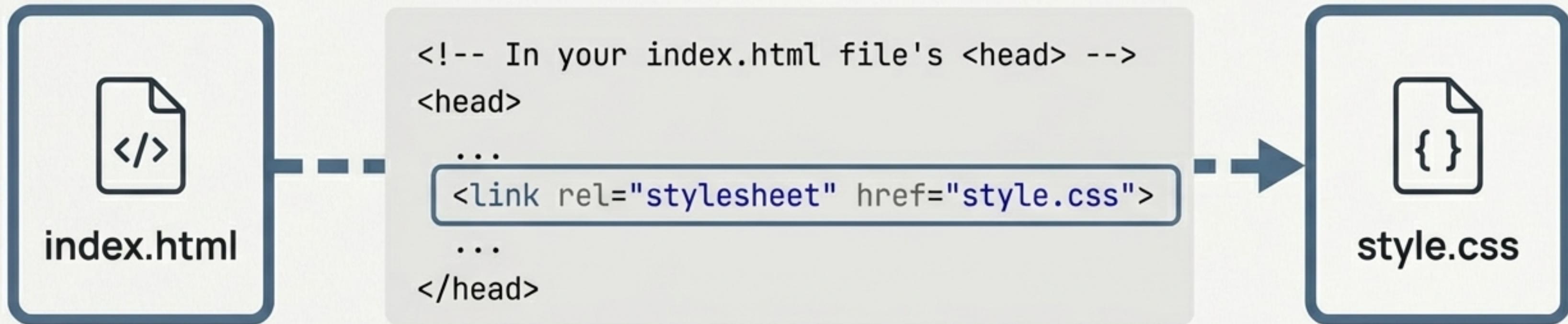


Role: Styles the appearance and layout.

```
p {  
    color: blue;  
}
```

HTML builds the house. CSS decorates the rooms.

How the Designer's Plans Connect to the Blueprint



The `<link>` tag in your HTML file tells the browser where to find the CSS rules, connecting the structure to its style.

Three Methods for Applying CSS

Inline CSS

Style is applied directly to a single HTML element. Good for quick tests, but not for overall site styling.

Inter Regular

```
<h1 style="color:  
blue;">My Heading</h1>
```

Internal CSS

CSS rules are placed inside a `<style>` tag within the HTML file's `<head>`. Affects only that single page.

Inter Regular

```
<style>  
  h1 { color: blue; }  
</style>
```

RECOMMENDED

External CSS

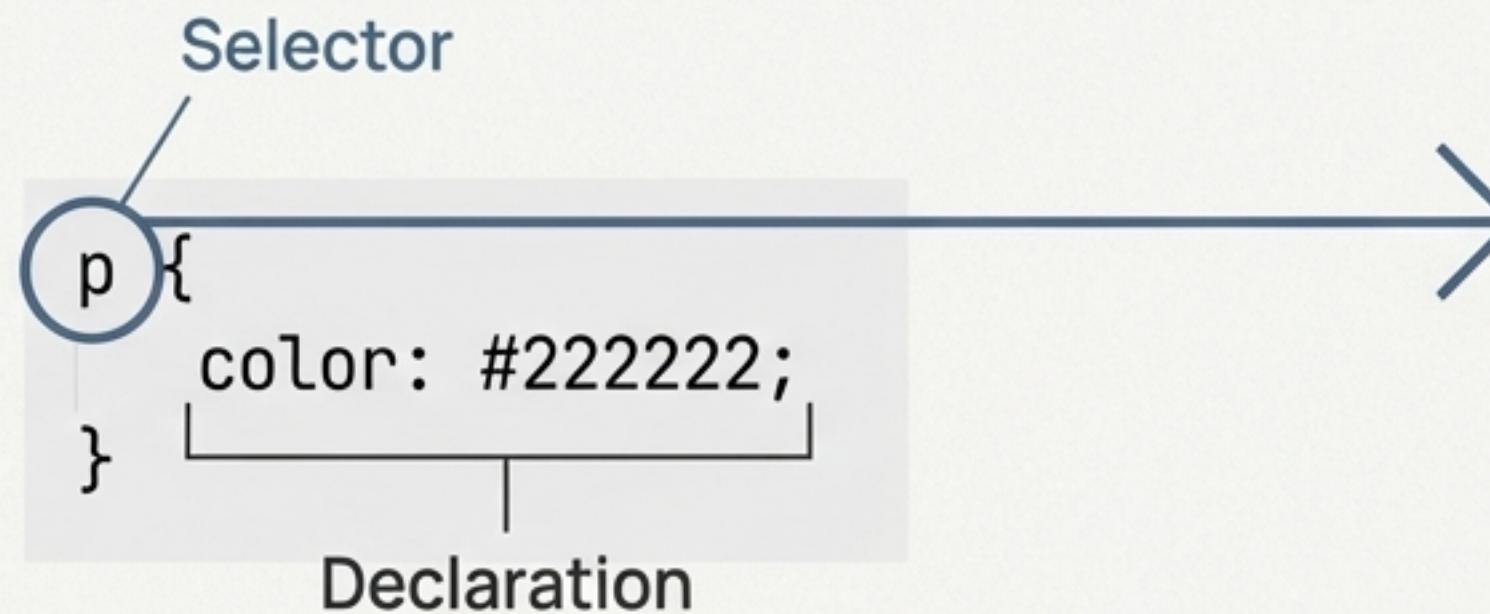
All CSS rules are in a separate ``.css`` file. This is the most organized, reusable, and efficient method.

Inter Regular

```
/* In style.css */  
h1 { color: blue; }
```

Selectors: Precisely Targeting What You Want to Style

“Think of selectors as the address on a letter. They tell the browser exactly which HTML element should receive the styling instructions.”



```
<body>
  <h1>A Title</h1>
  <p>This paragraph is selected.</p>
  <div>
    <p>This one too.</p>
  </div>
</body>
```

The Three Core Selector Types

Selector Type	Example Syntax	How It Works
Tag	p { ... }	Selects all elements of a specific tag (e.g., all paragraphs).
Class	.highlight { ... }	Selects any element with class="highlight". Can be used on many elements.
ID	#header { ... }	Selects the one unique element with id="header". An ID must be unique per page.

In Action

```
<header id="header">
  <h1>Welcome!</h1>
</header>
<p>This is a standard paragraph.</p>
<p class="highlight">This paragraph is special.</p>
```



Rendered Output

Welcome!

This is a standard paragraph.

This paragraph is special.

Typography: Giving Your Content a Clear Voice

Good typography is the foundation of good design. With CSS, you have complete control over how your text looks and feels, dramatically improving readability and user experience.

Welcome.

Welcome

Welcome.

Welcome

Welcome.

welcome.

Welcome

Welcome

Welcome

Welcome

Welcome

Welcome

Welcome.

A Practical Example: Styling a Heading

```
h1 {  
    font-size: 56px;  
    color: black;  
    text-align: center;  
    line-height: 1.1;  
}
```

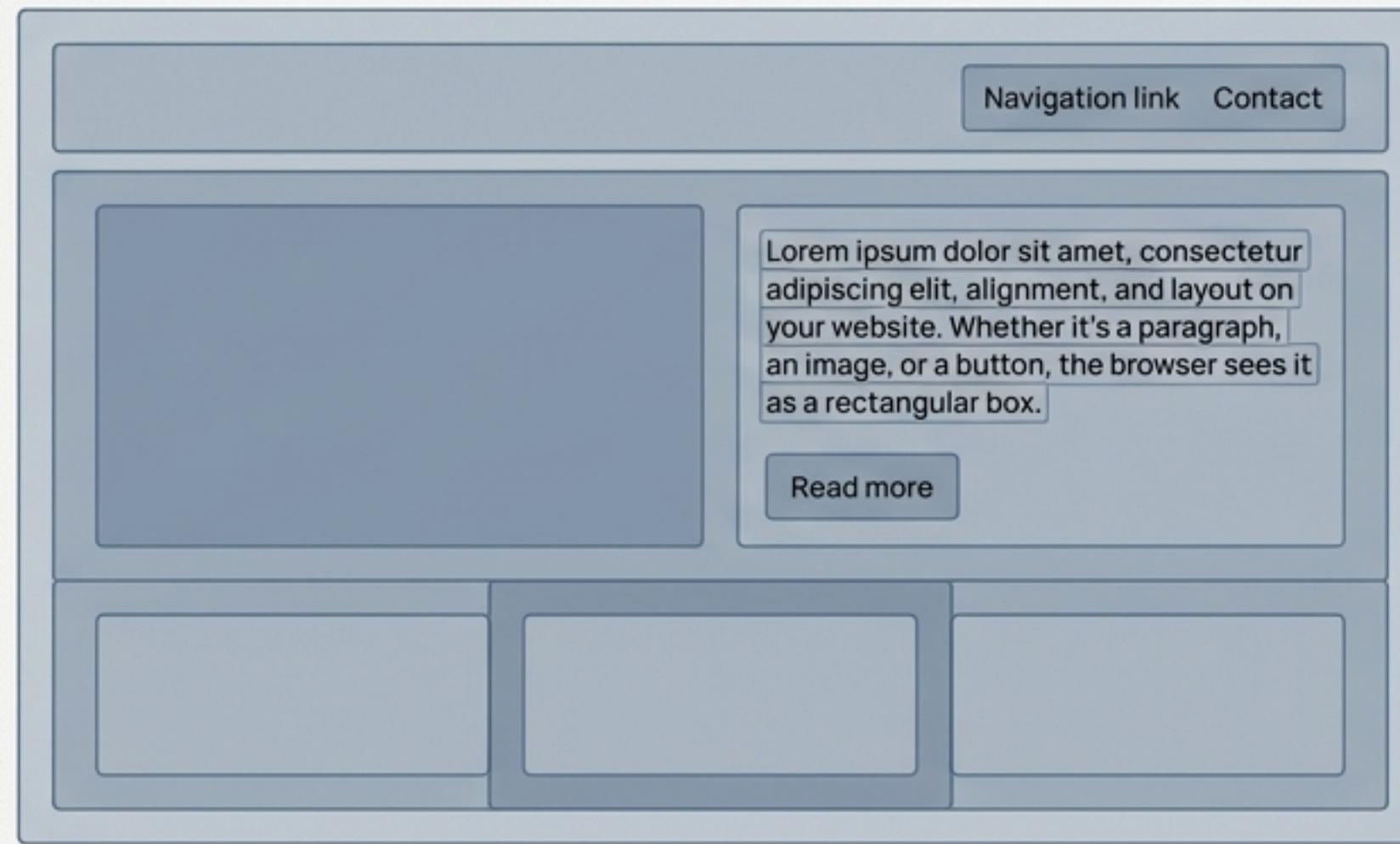
 Before

My Website

 After

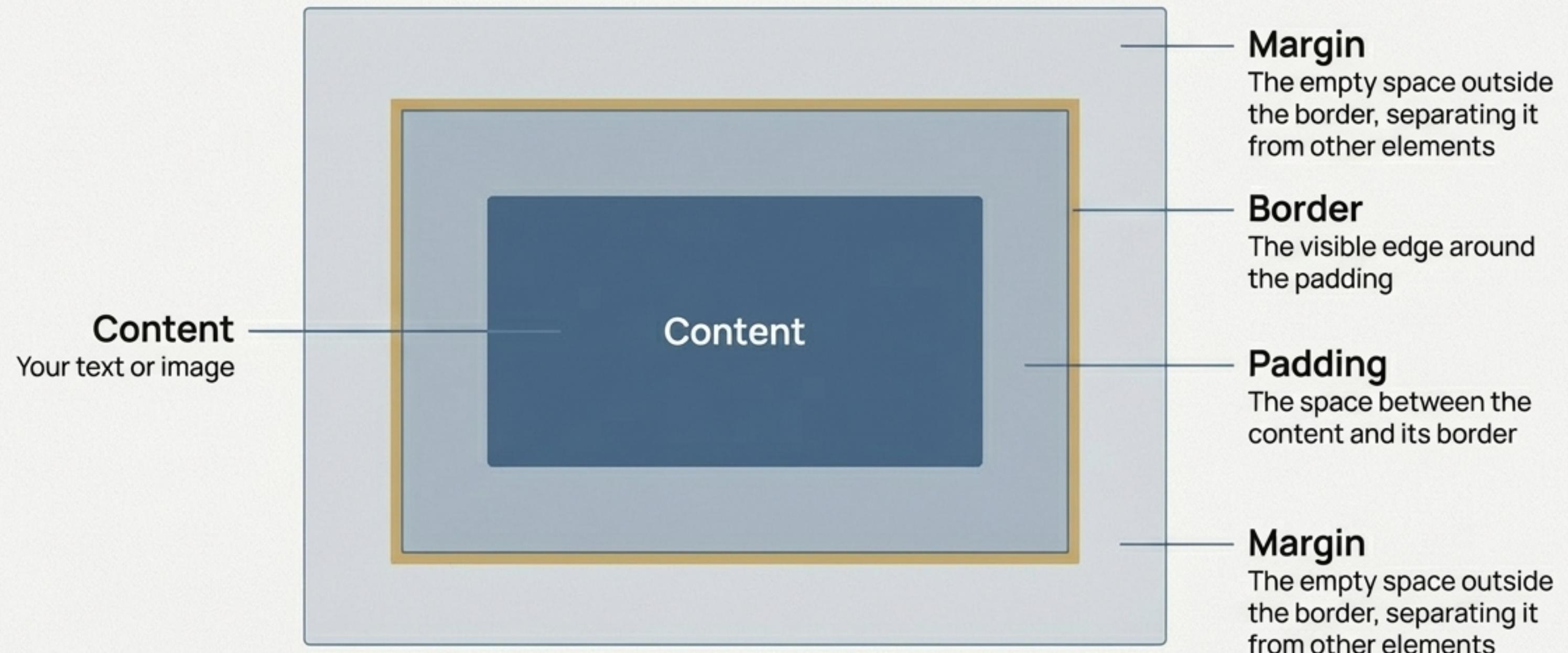
My Website

The Foundational Rule of CSS Layout: Every Element is a Box



Understanding this single concept is the key to controlling space, alignment, and layout on your website. Whether it's a paragraph, an image, or a button, the browser sees it as a rectangular box.

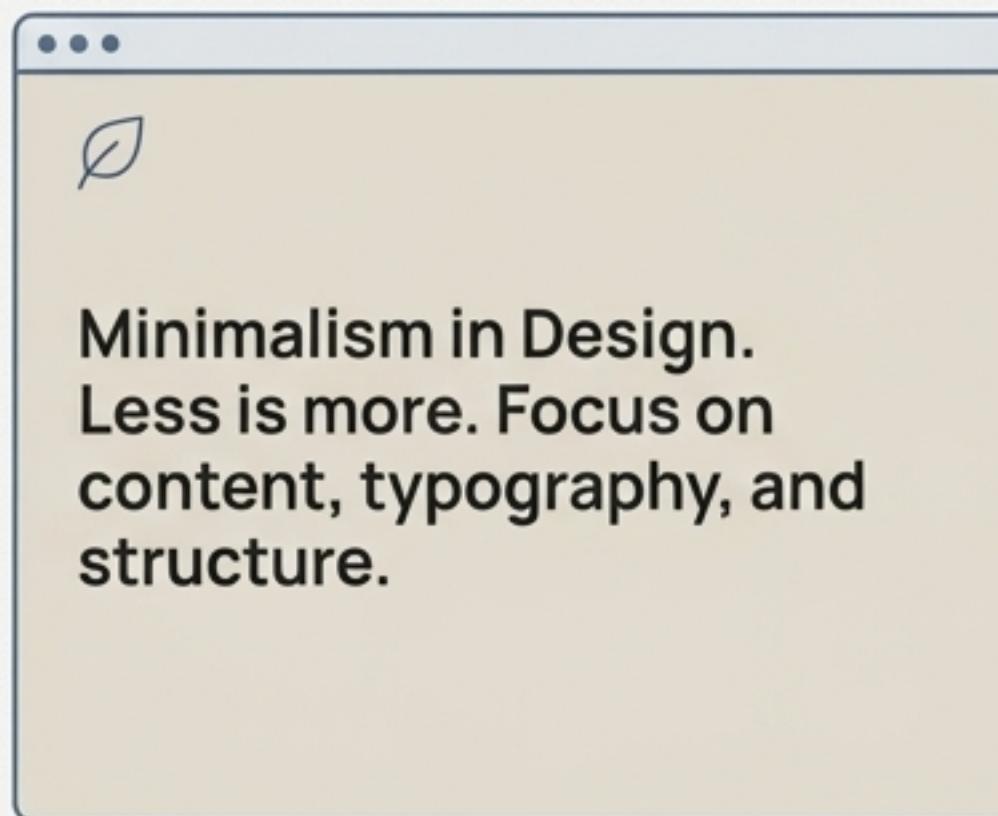
Anatomy of the Box Model



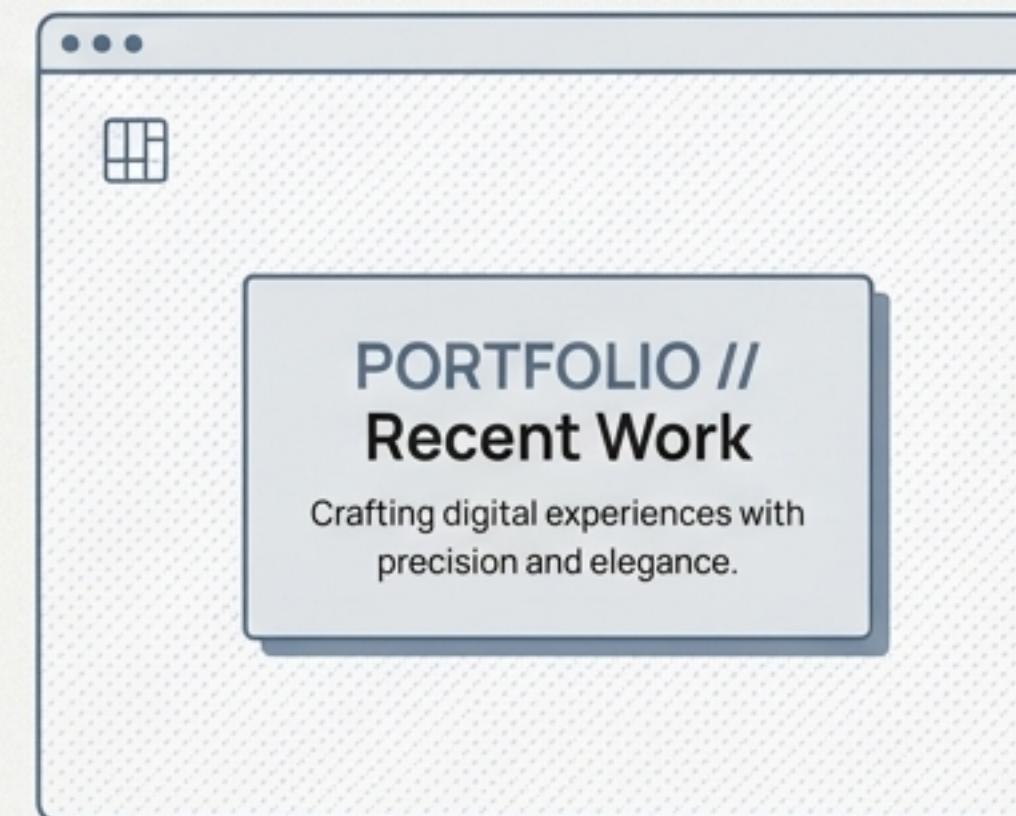
Backgrounds: Setting the Stage for Your Content

A well-chosen background can unify your design, improve readability, and create a specific mood. CSS gives you the power to use both solid colors and complex images.

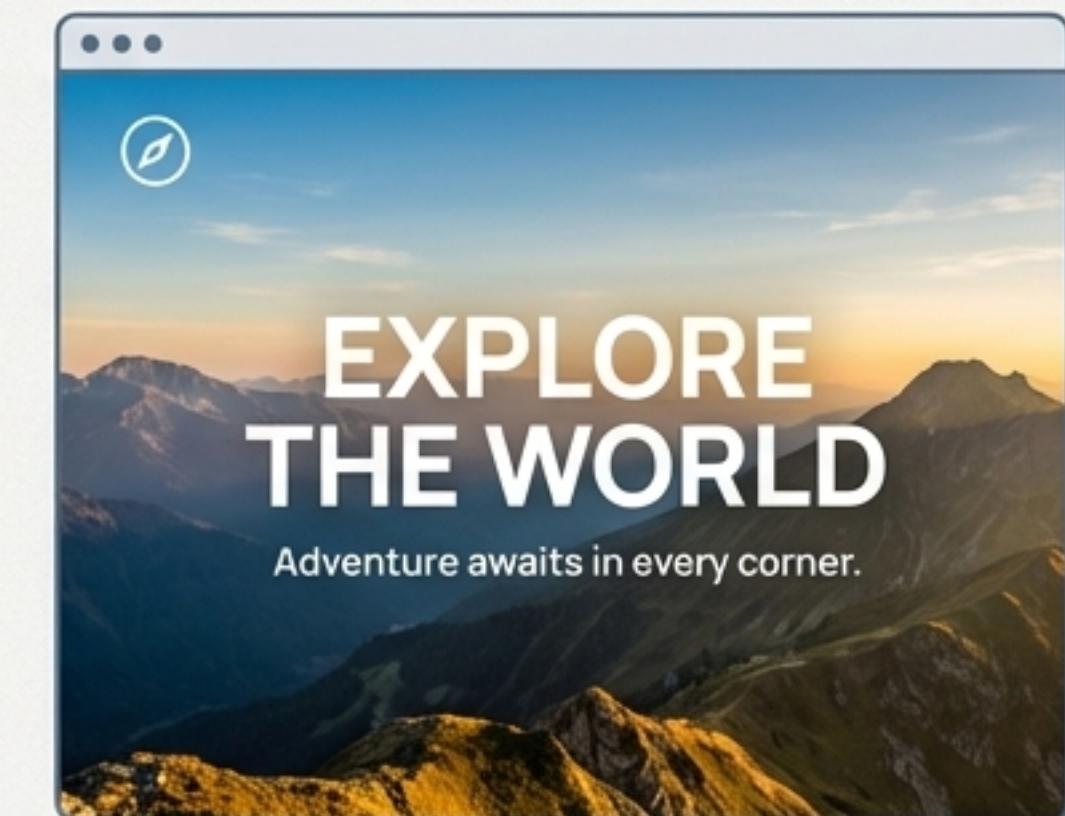
Solid Color



Subtle Pattern



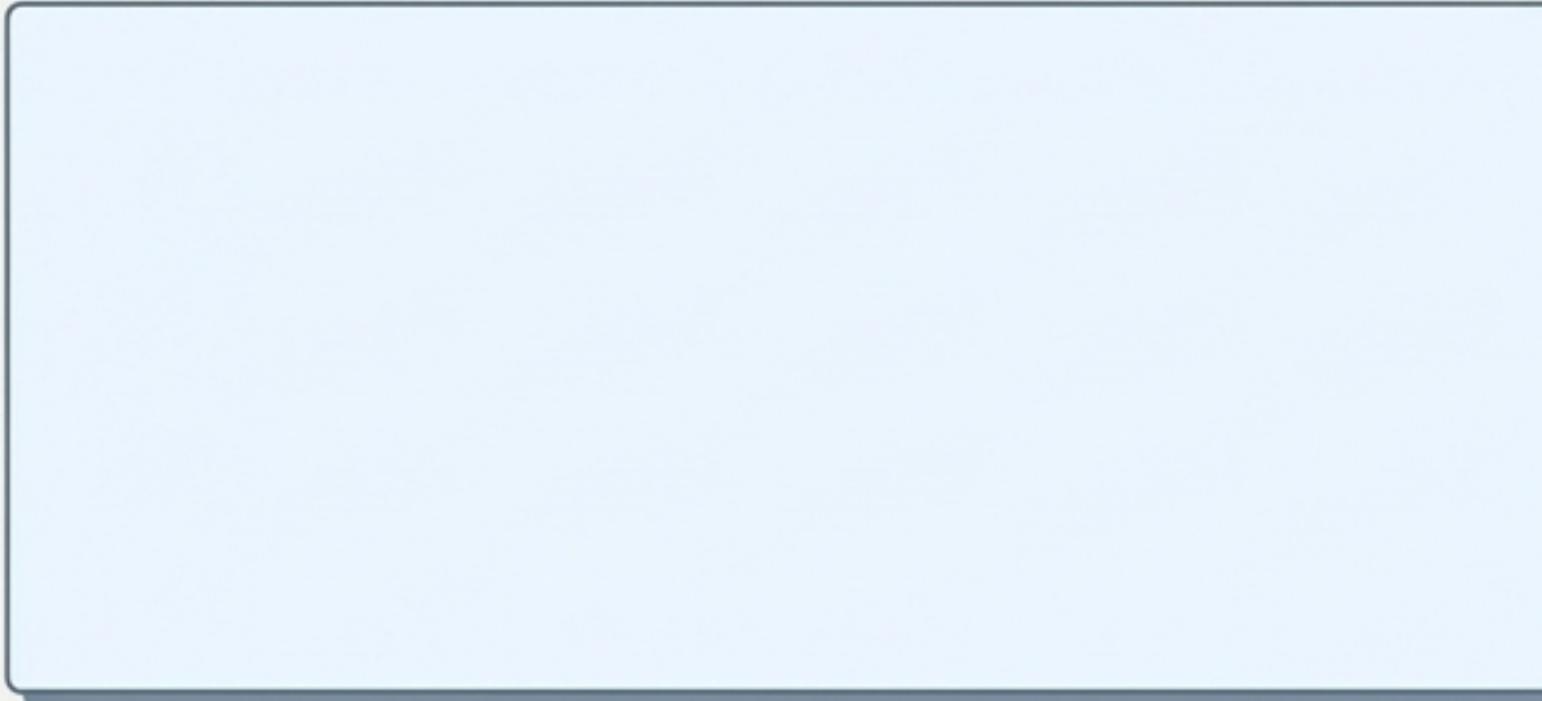
Full-Bleed Photograph



Applying Colors and Images

Solid Color

```
div {  
  background-color: #f0f8ff; /* Alice Blue */  
}
```



Background Image

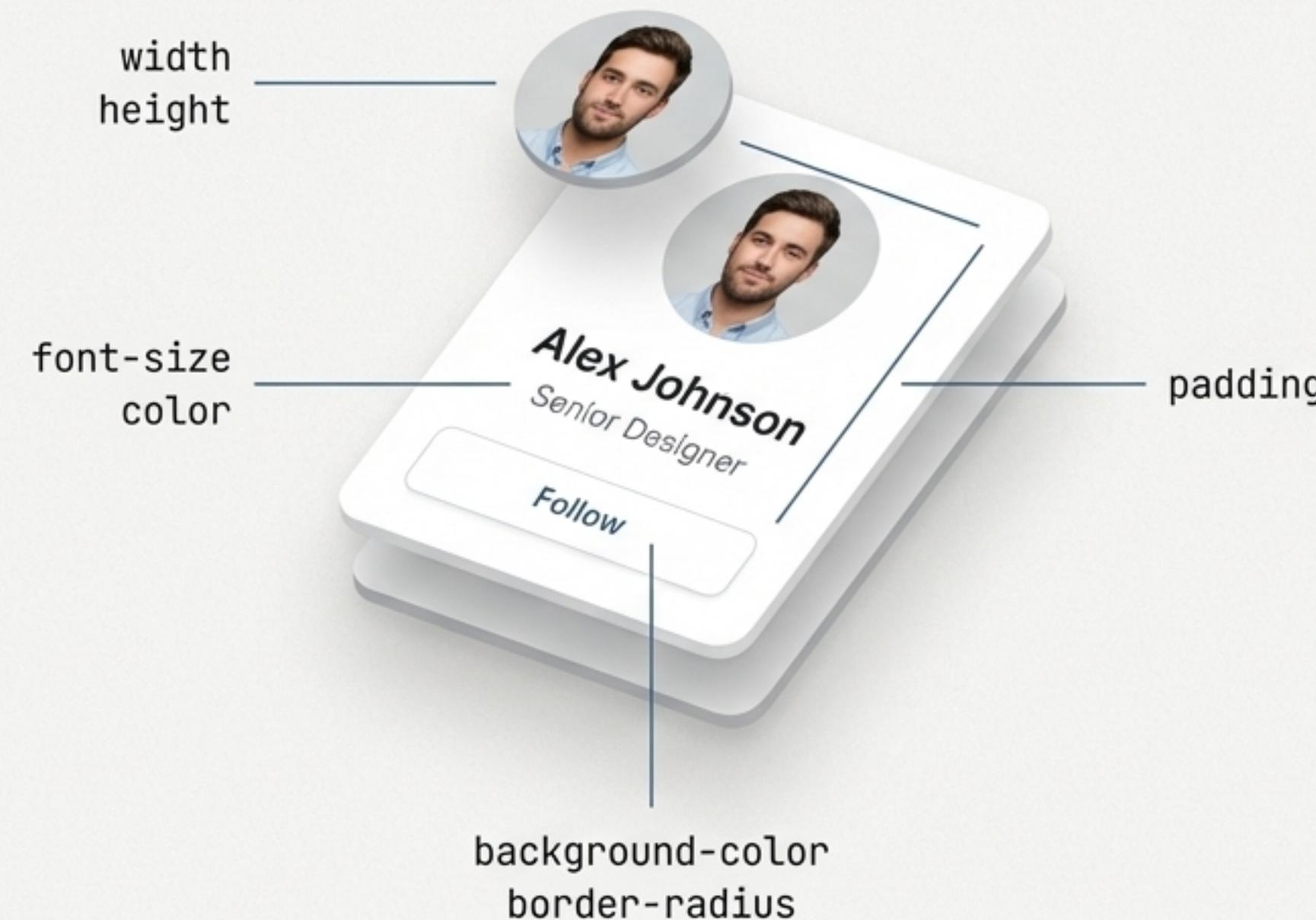
```
body {  
  background-image: url('wallpaper.jpeg');  
}
```



Performance Tip: Use optimized images like `jpg` or `webp` for backgrounds to ensure your website loads quickly.

From Individual Tools to a Cohesive Design

You've now collected the core tools: selectors for targeting, typography for voice, the box model for space, and backgrounds for mood. The next step is to see how they work together to build real components.

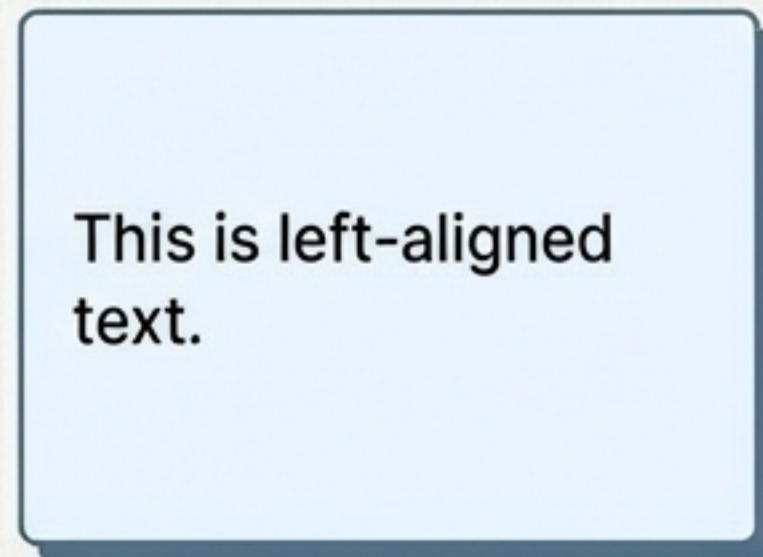


Two Essential Layout Techniques

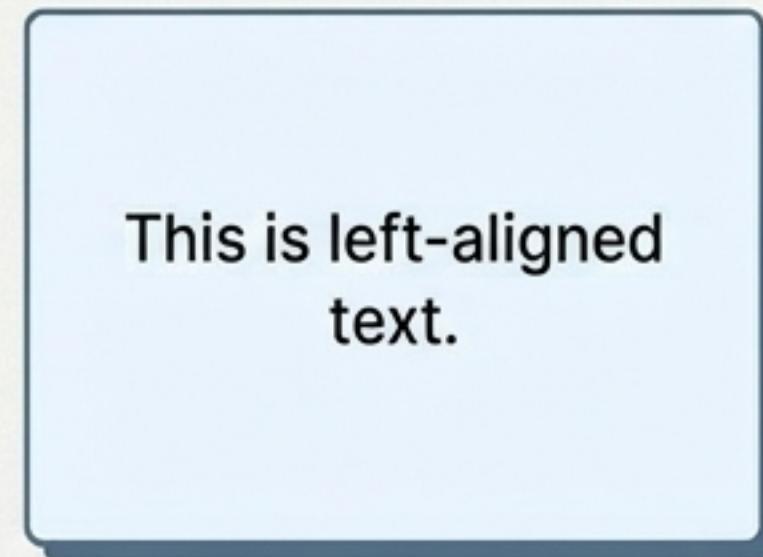
Centering Text

Use `text-align: center;` in Inter Regular.

Before



After

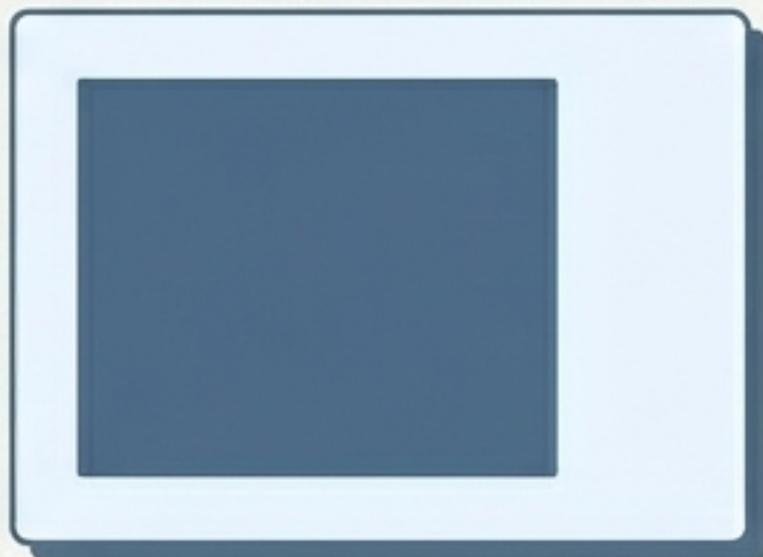


```
p {  
  text-align: center;  
}
```

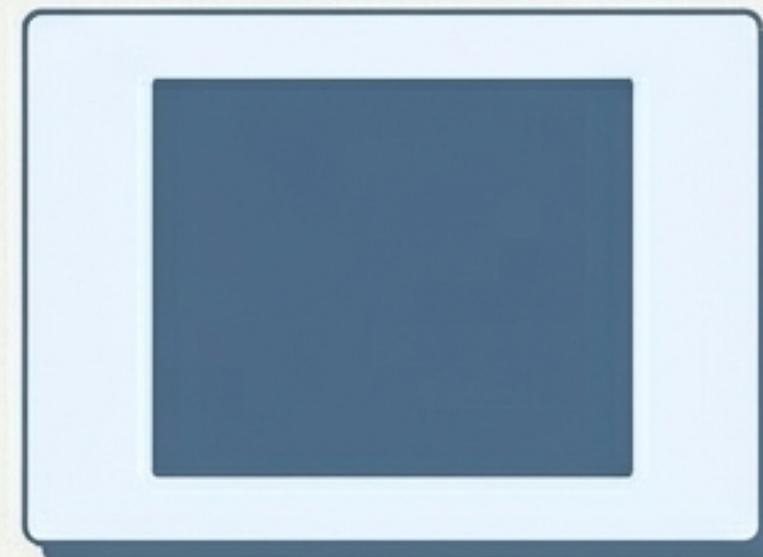
Centering a Box / Image

Use `margin: auto;` (on an element with a defined width). in Inter Regular.

Before



After



```
div {  
  width: 80%;  
  margin: auto;  
}
```

Capstone Project: Building a Navigation Menu

Let's apply everything we've learned to transform a basic HTML list into a professional navigation menu.

Before

- [Home](#)
- [About](#)
- [Services](#)
- [Contact](#)

The Blueprint (HTML)

```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Services</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</nav>
```

Step 1: Basic Structure and Spacing

First, we target the `` element to remove the default bullets and add a background color. We'll use padding to give the links some breathing room.

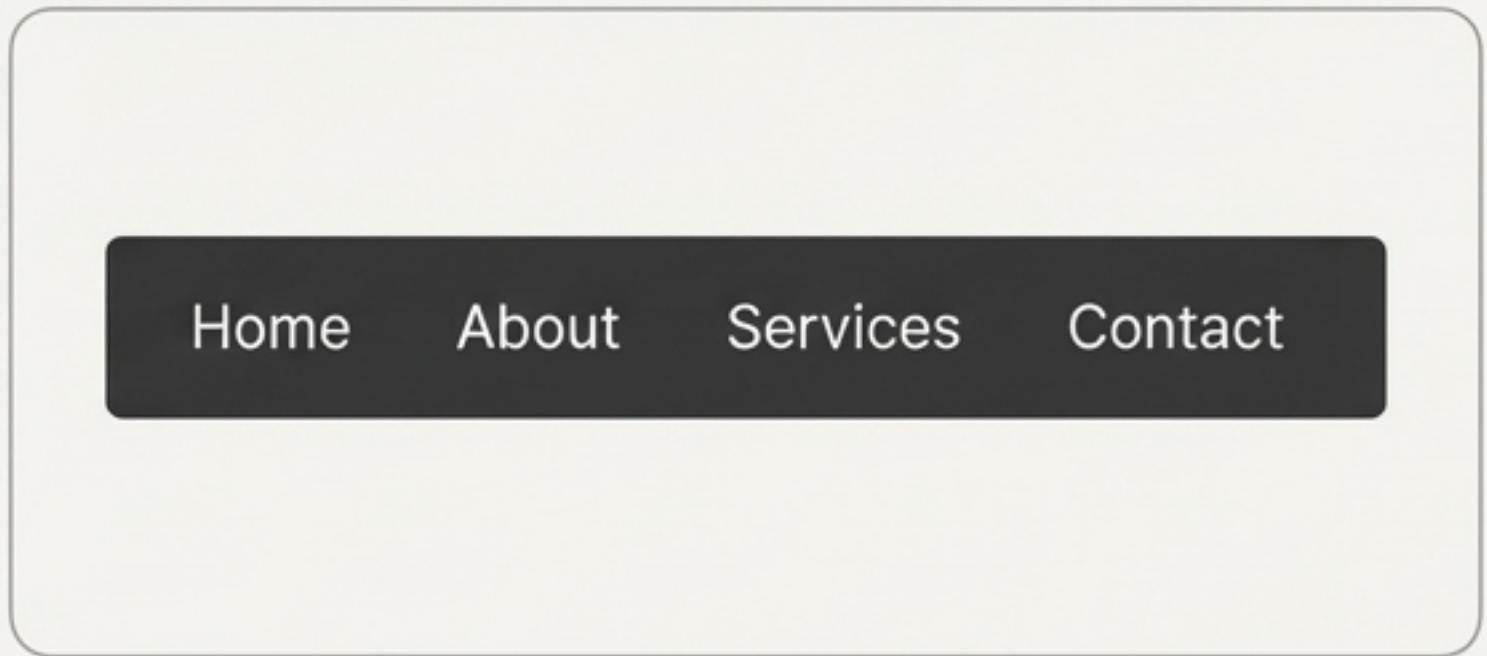
```
/* Step 1 */
nav ul {
  list-style-type: none; /* Removes bullets */
  background-color: #333;
  padding: 10px;
}
```

[Home](#) [About](#) [Services](#) [Contact](#)

Step 2: Styling the Links

Next, we'll style the `<a>` tags. We need to change the color to be visible on the dark background, remove the underline, and add padding to make the clickable area larger.

```
/* Step 2 */
nav ul li a {
  color: white;
  text-decoration: none; /* Removes underline */
  padding: 8px 16px;
}
```

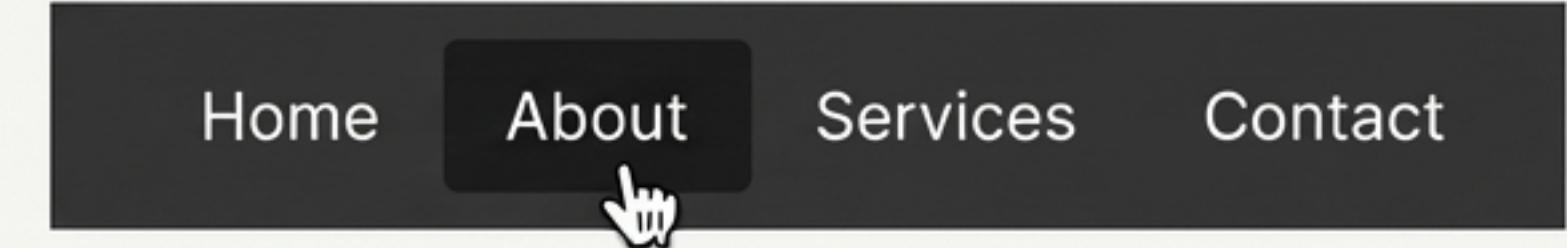


Home About Services Contact

Step 3: Adding an Interactive Hover Effect

A great menu provides visual feedback. We'll use the `:hover` pseudo-class to change the background color when a user's mouse is over a link.

```
/* Step 3 */  
nav ul li a:hover {  
  background-color: #111;  
}
```



The Transformation is Complete

The Blueprint

My Awesome Website

Welcome to the home page.

- Home
- About
- Services
- Contact

The Final Design

Home About Services Contact

My Awesome Website

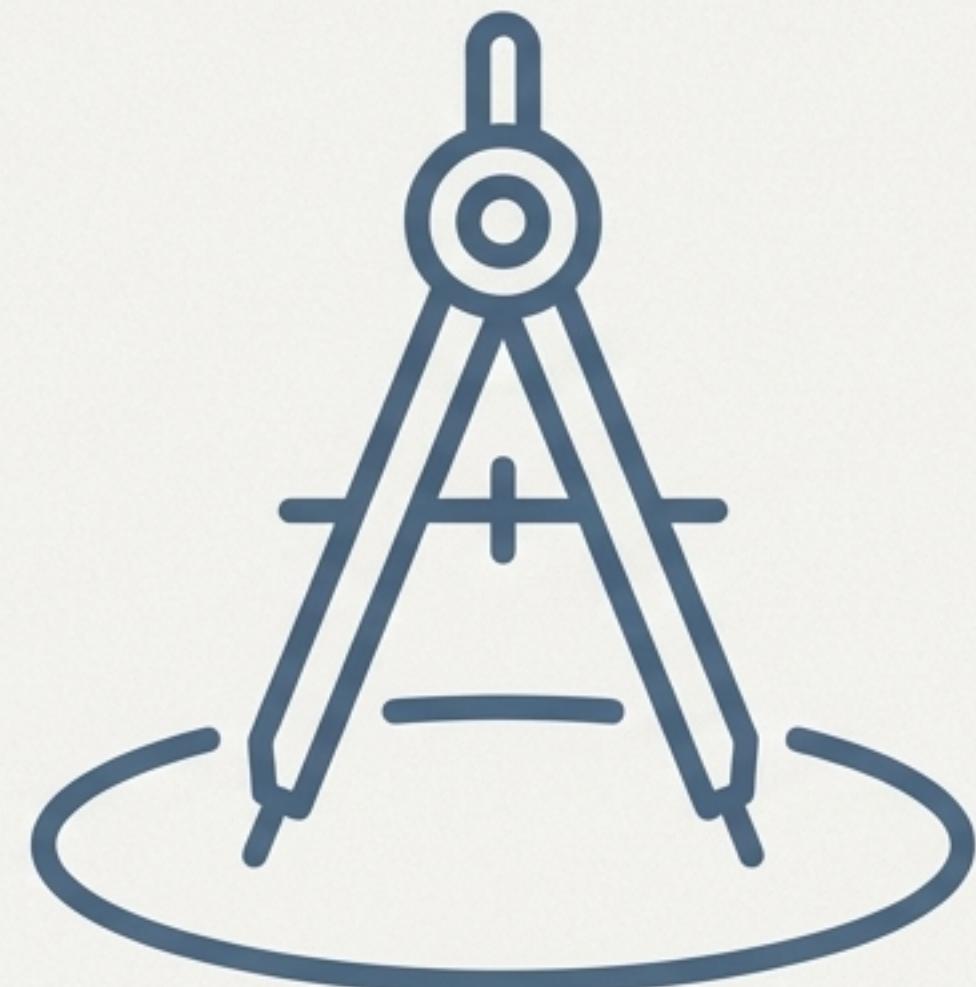
Welcome to the home page. This is the culmination of our journey through CSS, combining selectors, the box model, and more.

By combining selectors, typography, the box model, and backgrounds, you have transformed a plain document into a designed experience.

Your Mission: Apply Your New Skills

You now have the fundamental tools of a web designer. Your task is to take your project from Week 2 and use CSS to make it more beautiful, readable, and professional.

- What font will best represent your content?
- What color scheme will you choose?
- How can you use spacing to guide the reader's eye?



Your CSS Core Concepts Checklist



Manrope SemiBold **Connection**

- Linking an external `style.css` file



Manrope SemiBold **Selectors**

- Tag: `h1`
- Class: `.highlight`
- ID: `#header`



Manrope SemiBold **The Box Model**

1. Margin
2. Border
3. Padding
4. Content



Manrope SemiBold **Key Properties**

`color`, `font-size`,
`text-align`

`background-color`,
`background-image`

`width`, `margin`