From Structure to Style

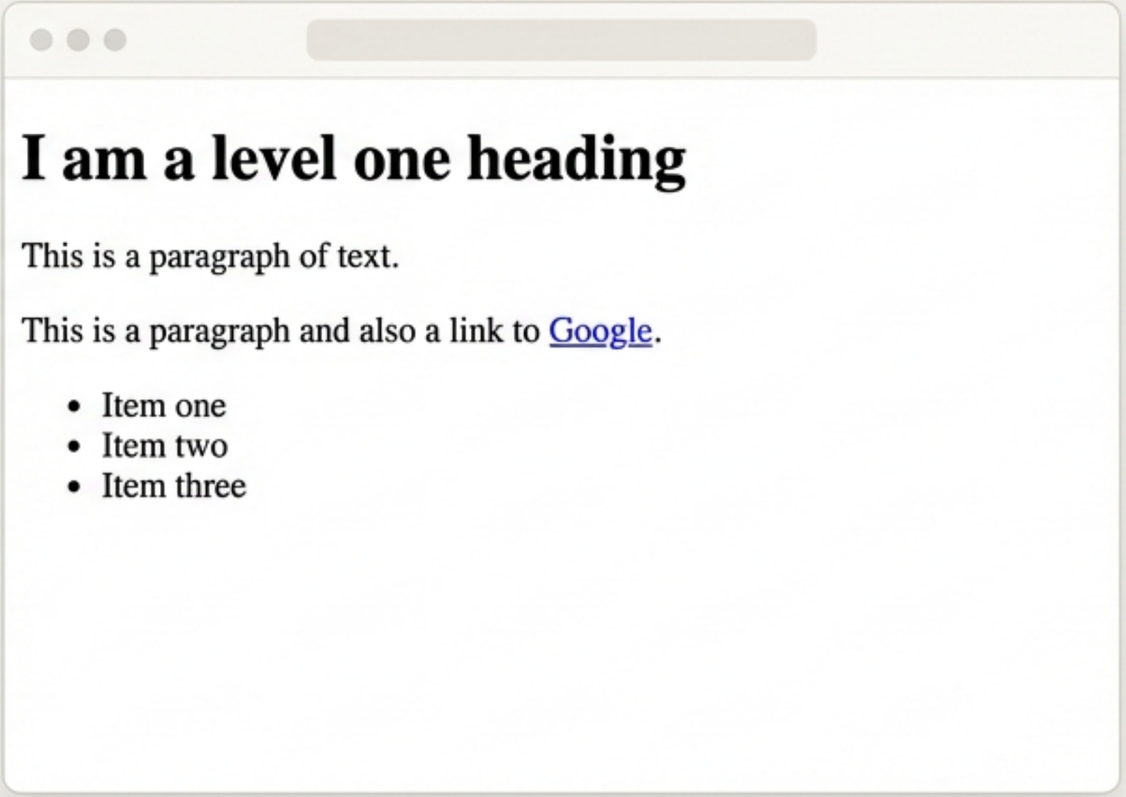
Your First Steps in Transforming Web Pages with CSS

The Web's Default Appearance

HTML gives our documents structure, but browsers apply only the most basic styles to make them readable. Headings are larger, paragraphs have space, and links are underlined. Without CSS, the web would be a very uniform and boring place.

Browser Default Styles | Manrope Semi-Styles

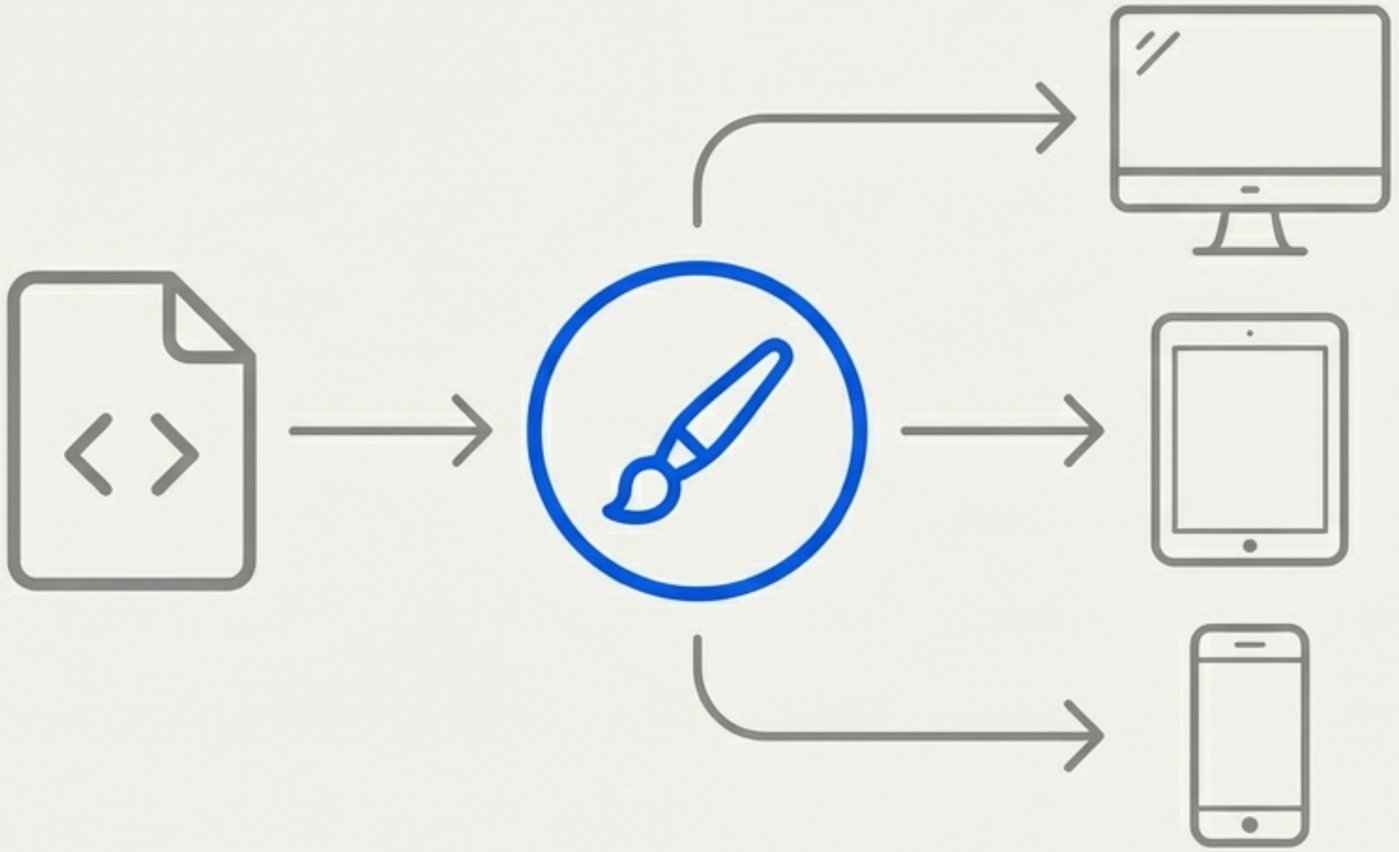
```html
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Getting started with CSS</title>
</head>
<body>
    <h1>I am a level one heading</h1>
    <p>This is a paragraph of text.</p>
    <p>This is a paragraph and also a link to <a
       href="https://google.com">Google</a>.</p>
    <ul>
       <li>Item one</li>
       <li>Item two</li>
       <li>Item three</li>
    </ul>
</body>
</html>
```



"The web would be a boring place if all websites looked like that."

The Power to Present: What is CSS for?

CSS (Cascading Style Sheets) is a language for specifying how documents are presented to users. It's how we control the styling, layout, and even animation of our HTML.



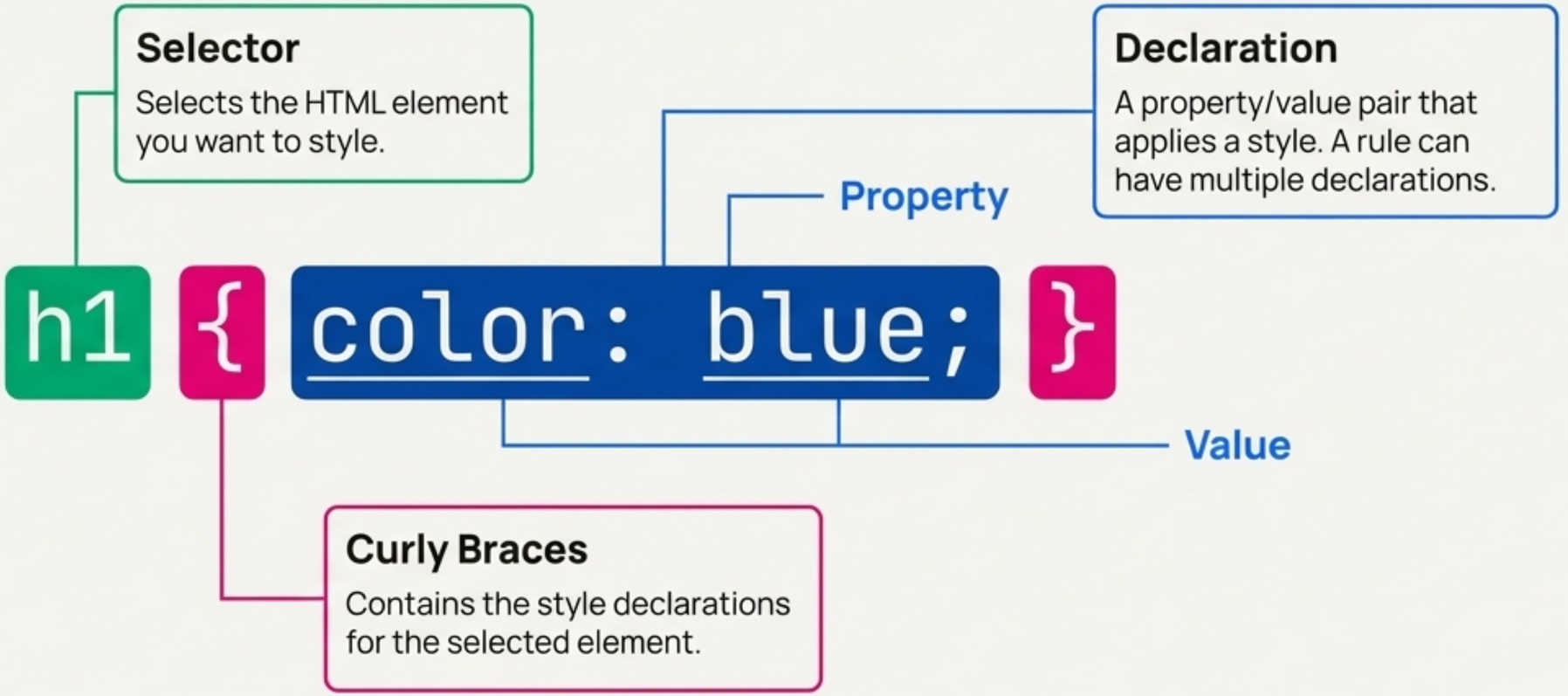
Text Styling:
Change colors, sizes, and fonts of headings, links, and text.

Layout Creation:
Turn a single column of text into complex layouts with sidebars and content areas.

Dynamic Effects:
Add animations and other interactive effects to engage users.

Speaking the Language of Style: The Anatomy of a CSS Rule

CSS is a rule-based language. You define rules that specify which styles should apply to which elements.



Making the Connection

To apply your styles, you must tell the HTML document where to find your CSS file. The most common and useful method is linking an external stylesheet from the <head> section of your HTML.

```
<head>
  <meta charset="utf-8">
  <title>Getting started with CSS</title>
  <!-- This line links our stylesheet -->
  <link rel="stylesheet" href="styles.css">
</head>
```

rel="stylesheet":
Tells the browser it's a stylesheet.

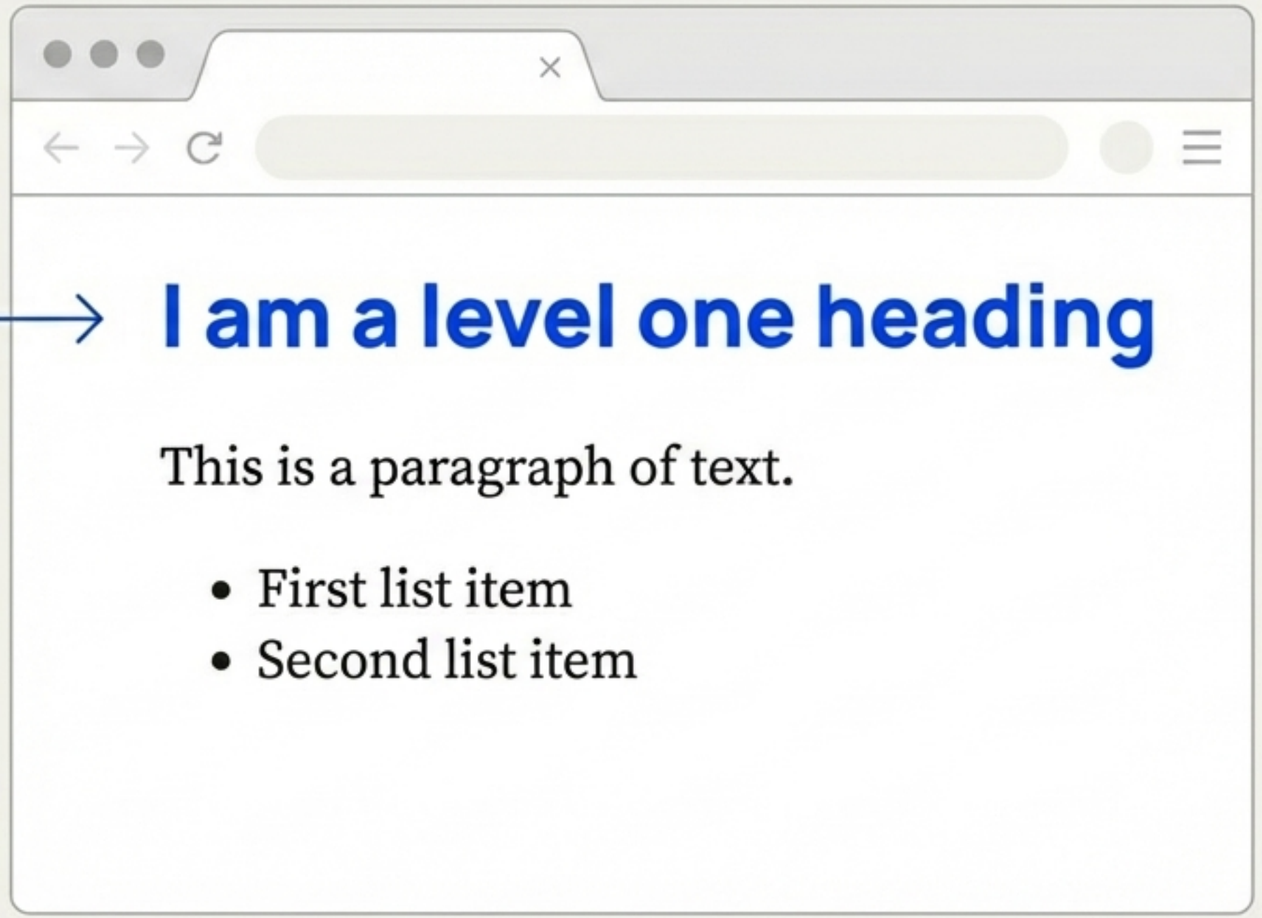
href="styles.css":
Provides the path to the CSS file.

Your First Transformation

styles.css

```
/* styles.css */

h1 {
  color: blue;
}
```



Expanding Your Reach

Targeting by Element Selector

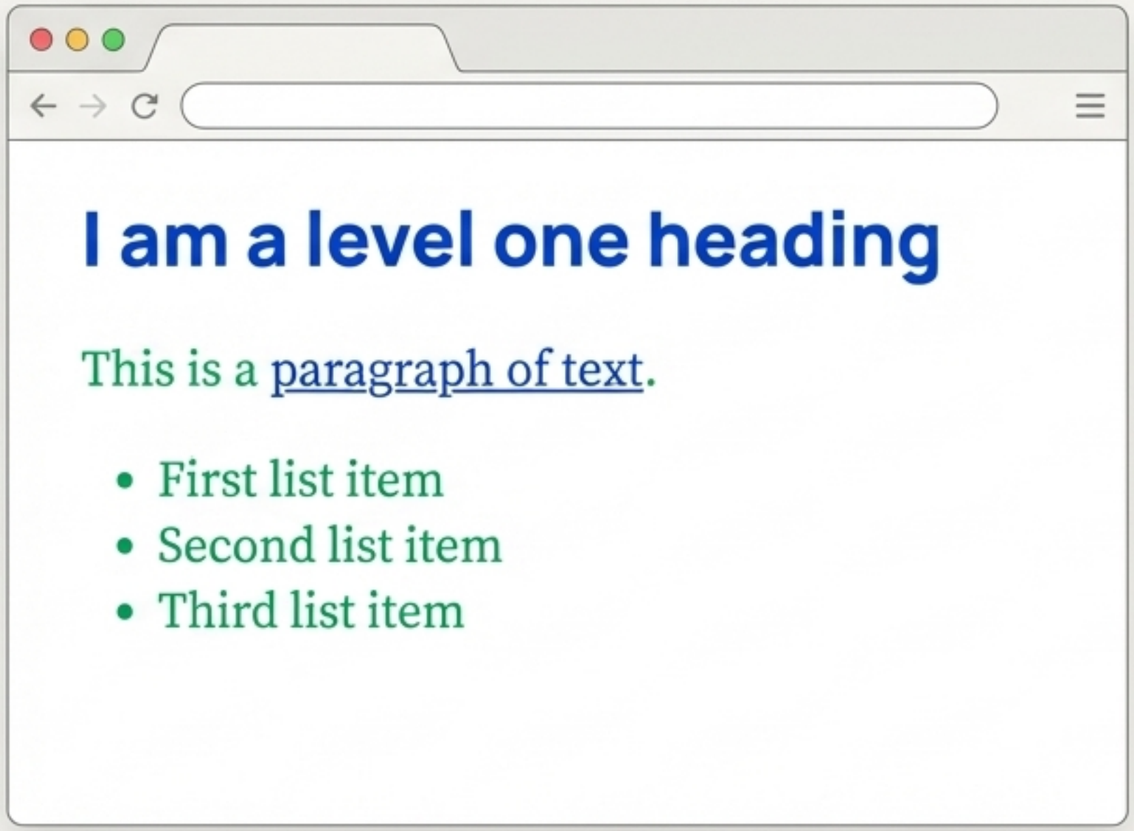
You can target any HTML element by its name. To style all paragraphs, use the `p` selector.

```
p { color: green; }
```

Targeting Multiple Elements

To apply the same style to different elements, separate the selectors with a comma.

```
p, li { color: green; }
```



Styling with Context: Responding to User Actions

CSS can target elements based on their state, such as a link that has been visited or is being hovered over. These are called pseudo-classes.

```
a:link { color: pink; } /* For unvisited links */
```

```
a:visited { color: green; } /* For visited links */
```


```
a:hover { text-decoration: none; } /* For when the mouse is over the link */
```

Accessibility Note: Be mindful of user experience. While you *can* remove the underline from links, it's a crucial visual cue that users expect. Ensure your styled links are still clearly identifiable as interactive elements.

Taking Control: Changing Default Behavior

Browsers have internal stylesheets that create the default look of HTML elements like list bullets. You can easily override these defaults with your own CSS rules.

To remove the default bullets from an unordered list (`<ul>`), use the `list-style-type` property.

Browser Default

- Item one
- Item two
- Item three

With CSS

Item one
Item two
Item three

The Power of Precision: Introducing Classes

The Problem: Styling by element name (`h1`, `p`, etc.) affects *every* element of that type. What if you want different styles for different paragraphs or lists?

The Solution: Add a `class` attribute to your HTML elements. This lets you create a specific "hook" that your CSS can target.

HTML Example

```
<ul class="important-list">
  <li>Item one</li>
</ul>
```

CSS Syntax

In your CSS, you target a class by using its name preceded by a period (`.`).

```
.important-list { color: red; }
```

Classes in Action: Creating Variation

By defining multiple classes, you can create a library of styles to apply throughout your document.

```
.square-list {
  list-style-type: square;
}
```

- Item one
- Item two

```
.roman-numerals {
  list-style-type: upper-roman;
}
```

I. Item one
II. Item two

```
.circle-list {
  list-style-type: circle;
}
```

- Item one
- Item two

```
.greek-letters {
  list-style-type: lower-greek;
}
```

α. Item one
β. Item two

From Document to Design: The Transformation

BEFORE

A Simple Document

This is a basic paragraph. It has no specific styling applied and uses the default font and color. The spacing is also standard.

- Item one
- Item two
- Item three

Learn more about web design

AFTER

A Simple Document

This is a basic paragraph. It has no specific styling applied and uses the default font and color. The spacing is also standard.

- Item one
- Item two
- Item three

Learn more about web design

Continue Your Journey: Practice Your Skills

The best way to learn is by doing. Apply your new CSS knowledge to the HTML exercises from previous units.

Style the headers, paragraphs, and text formatting from Units 1 & 2.

Change the colors and states of the hyperlinks from Unit 3.

Customize the colors, bullets, and numbering of the lists from Unit 4.

Need color ideas? Find a full list of color keywords on the MDN Web Docs for `color_value`.

Test Your Knowledge

Ready to check your understanding? Solidify what you've learned with these short quizzes.

Quiz: Basic Concepts

Quiz: Colors and Simple Selectors