

Introduction to CSS

CSS (Cascading Style Sheets) is used to style and layout web pages — including colors, fonts, spacing, and positioning of elements. While HTML gives structure to a web page, CSS makes it look beautiful and usable.

Why CSS?

Without CSS, all websites would look plain, like unstyled documents. CSS helps you:

- Change colors and fonts
- Add spacing and layout
- Make responsive designs for mobile
- Animate and transition between states
- Separation of concerns: HTML for structure, CSS for style

How CSS Works with HTML

CSS can be applied to HTML in **three main ways**:

1. Inline CSS

CSS written inside an HTML tag using the `style` attribute.

```
<p style="color: blue; font-size: 18px;">This is a blue paragraph.</p>
```

2. Internal CSS

CSS written inside a `<style>` tag within the `<head>` section of the HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p {
        color: green;
        font-weight: bold;
      }
    </style>
  </head>
  <body>
    <p>This is a green bold paragraph.</p>
  </body>
</html>
```

3. External CSS (Best Practice)

CSS written in a separate file and linked to the HTML file. This is the **most recommended** method for real-world projects.

style.css

```
h1 {
  color: darkred;
  text-align: center;
}
```

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Welcome to CSS</h1>
```

```
</body>
</html>
```

The “Cascading” in CSS

If there are multiple rules targeting the same element, CSS uses the **cascade** to decide which rule to apply. This depends on:

- **Specificity** (How specific the selector is)
- **Order of appearance**
- **Importance** (`!important`)

Example:

```
<p style="color: red;">This will be red because inline CSS wins.</p>
```

Anatomy of a CSS Rule

```
selector {
  property: value;
}
```

Example:

```
p {
  color: black;
  font-size: 16px;
}
```

- `p` → Selector (targets all `<p>` elements)
- `color` , `font-size` → Properties
- `black` , `16px` → Values

What You'll Learn in CSS

As we move forward, you'll learn how to:

- Style text, backgrounds, and borders
- Control layout with Flexbox and Grid
- Make your website responsive and mobile-friendly
- Animate elements and transitions
- Write modern, maintainable CSS