

Basics of CSS

≡ Week 2

Adib Sakhawat
IUT CSE '21

CSS (Cascading Style Sheets) is a style sheet language used to control the presentation of HTML content, including layout, colors, fonts, and spacing. It allows developers to separate content (HTML) from design (CSS), improving maintainability and flexibility.

Basic Structure of a CSS File

A CSS file is a plain text file with the extension `.css`. It consists of a collection of rules that define how HTML elements should be styled.

Syntax of a CSS Rule

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

- **Selector:** Specifies the HTML element(s) the rule applies to (e.g., `p`, `h1`, `.class`, `#id`).
- **Property:** Defines the style aspect to modify (e.g., `color`, `font-size`, `margin`).
- **Value:** Specifies the style setting (e.g., `red`, `16px`, `10px`).

Example of a CSS File

```
/* Basic CSS Example */  
  
/* Apply styles to all paragraphs */  
p {  
    color: darkblue;  
}
```

```
    font-size: 16px;
    line-height: 1.5;
}

/* Style a specific class */
.card {
    background-color: #f4f4f4;
    border: 1px solid #ccc;
    padding: 10px;
    border-radius: 5px;
}

/* Style an element by ID */
#header {
    font-family: 'Arial, sans-serif';
    font-size: 24px;
    text-align: center;
}

/* Style multiple selectors */
h1, h2, h3 {
    color: #333;
    margin-bottom: 10px;
}

/* Add a hover effect */
a:hover {
    color: red;
    text-decoration: underline;
}
```

How to Use a CSS File

1. Inline CSS

CSS is written directly within an HTML element using the `style` attribute.

Example:

```
<p style="color: red; font-size: 18px;">This is an inline-styled paragraph.</p>
```

2. Internal CSS

CSS is written inside a `<style>` tag in the `<head>` section of the HTML file.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <style>
    p {
      color: blue;
      font-size: 18px;
    }
  </style>
</head>
<body>
  <p>This is an internally styled paragraph.</p>
</body>
</html>
```

3. External CSS

CSS is written in a separate `.css` file and linked to the HTML document using a `<link>` tag.

Example:

CSS File (`styles.css`):

```
body {
  background-color: #f0f0f0;
  font-family: Arial, sans-serif;
}

h1 {
  color: navy;
```

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

HTML File:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="stylesheet" href="styles.css">
  <title>Example</title>
</head>
<body>
  <h1>Welcome</h1>
  <p>This is a paragraph styled with external CSS.</p>
</body>
</html>
```

CSS Selectors

1. Universal Selector

Selects all elements.

```
* {
  margin: 0;
  padding: 0;
}
```

2. Element Selector

Selects all instances of a specific element.

```
h1 {
  color: blue;
}
```

3. Class Selector

Selects elements with a specific class. Use `.` before the class name.

```
.card {  
  background: white;  
}
```

4. ID Selector

Selects an element with a specific ID. Use `#` before the ID name.

```
#header {  
  text-align: center;  
}
```

5. Group Selector

Applies the same style to multiple selectors.

```
h1, h2, h3 {  
  font-weight: bold;  
}
```

6. Pseudo-classes

Apply styles based on a specific state.

```
a:hover {  
  color: red;  
}
```

CSS Properties Overview

1. Text and Fonts

- `color`: Sets text color.
- `font-size`: Specifies text size.
- `font-family`: Sets the font style.
- `text-align`: Aligns text horizontally.

- `text-decoration`: Adds or removes text decorations (e.g., underline).

2. Box Model

The box model governs the size and spacing of elements.

- `margin`: Space outside the element.
- `padding`: Space inside the element, around content.
- `border`: Border around the element.
- `width` and `height`: Control the size of the element.

Example:

```
.box {  
  width: 200px;  
  height: 100px;  
  margin: 20px;  
  padding: 10px;  
  border: 2px solid black;  
}
```

3. Backgrounds and Colors

- `background-color`: Sets the background color.
- `background-image`: Adds a background image.

4. Positioning and Layout

- `position`: Specifies the element's position (`static`, `relative`, `absolute`, `fixed`).
- `display`: Determines the element's display type (`block`, `inline`, `flex`, etc.).
- `flexbox` and `grid`: Modern methods for layout design.

Best Practices for Writing CSS

1. Use External CSS for Reusability:

- Keep styles separate from content for cleaner code and easier maintenance.

2. Organize CSS with Comments:

- Use comments to describe sections of your CSS file.

```
/* Header Styles */
```

3. Minimize Specificity:

- Use class selectors over ID selectors to make the code easier to override and maintain.

4. Use Shorthand Properties:

- Simplify your CSS by combining related properties.

```
margin: 10px 20px 30px 40px; /* top, right, bottom, left  
*/
```

5. Optimize for Performance:

- Minify CSS files for production to reduce file size and loading times.