

# 1. What is CSS: Purpose and Benefits of CSS, CSS Syntax and Selectors

## Learning Objectives

By the end of this lesson, students will:

1. Understand what CSS is, its purpose, and its benefits in web development.
  2. Gain knowledge of CSS selectors, including element selectors, class selectors (.class), ID selectors (#id), and the universal selector (\*).
  3. Understand how to add CSS to HTML using inline styles, internal (embedded) styles, and external stylesheets.
  4. Use CSS to style text, including changing fonts, sizes, and aligning text.
- ## 2. Introduction

CSS (Cascading Style Sheets) is a language used to describe the presentation of a webpage, such as its layout, colors, fonts, and overall appearance. While HTML provides the structure and content of a webpage, CSS controls its style. By separating content from design, CSS allows for more flexible and efficient web development.

In this lesson, we will cover the purpose and benefits of CSS, its basic syntax, and how to use CSS selectors to target and style HTML elements.

### 1. What is CSS?

#### Purpose of CSS

CSS is used to style and layout web pages. It defines how HTML elements should be displayed on the screen, allowing developers to control aspects such as colors, fonts, spacing, and positioning. By using CSS, you can create visually appealing and consistent web designs.

#### Key Benefits of CSS

1. **Separation of Content and Design:** By separating content (HTML) from design (CSS), developers can manage both independently, making it easier to maintain and update websites.
  2. **Consistency:** With CSS, you can apply consistent styling across multiple pages or elements, ensuring a unified look and feel for a website.
  3. **Efficiency:** CSS enables developers to style entire websites by applying styles to multiple elements at once, reducing the need to repeat styling rules within HTML.
  4. **Responsive Design:** CSS allows you to create layouts that adapt to different screen sizes, making websites more mobile-friendly.
- ## 3. Example: HTML without CSS vs. HTML with CSS

Without CSS (Plain HTML):

```
<!DOCTYPE html><html lang="en">
<head>
  <title>Simple Web Page</title>
</head>
<body>
  <h1>Welcome to My Website</h1>
  <p>This is a paragraph of text.</p>
</body>
</html>
```

4.

5. With CSS:

```
<!DOCTYPE html><html lang="en">
<head>
  <title>Styled Web Page</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
      color: #333;
    }
    h1 {
      color: navy;
      text-align: center;
    }
    p {
      font-size: 18px;
      text-align: justify;
    }
  </style>
</head>
<body>
  <h1>Welcome to My Styled Website</h1>
  <p>This is a paragraph of text with styling applied using CSS. Notice the background color,
font style, and text alignment.</p>
</body>
</html>
```

6.

7. Key Takeaways:

1. CSS enhances the visual presentation of a webpage, making it more aesthetically pleasing.
2. It promotes code reusability and consistency across multiple pages.
3. CSS is essential for creating responsive and mobile-friendly designs.

8.

---

## 2. CSS Syntax

CSS is written in rulesets that contain selectors and declarations. The selector targets the HTML element(s) you want to style, and the declaration block contains one or more declarations that specify the style to be applied.

CSS Syntax Example:

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

9.

10. Example Explanation:

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

11.

1. `p`: The selector targets all `<p>` elements (paragraphs).
2. `color: blue;`: The property is `color`, and the value is `blue`. This changes the text color of paragraphs to blue.
3. `font-size: 16px;`: The property is `font-size`, and the value is `16px`, changing the font size of paragraphs.

12. Declaration Block:

1. Declarations are placed inside curly braces `{}`.
2. Each declaration consists of a property (what you want to style) and a value (how you want to style it).
3. Declarations are separated by semicolons `;`.

13. Common CSS Properties:

1. `color`: Changes the color of text.
2. `background-color`: Sets the background color of an element.
3. `font-size`: Changes the size of the text.
4. `text-align`: Aligns text (e.g., `left`, `center`, `right`).
5. `margin` and `padding`: Adds space around or inside elements.

14.

---

### 3. CSS Selectors

CSS selectors define which HTML elements the styling rules apply to. There are different types of selectors, each used to target specific elements or groups of elements.

#### 1. Element Selector

The element selector targets all instances of a specified HTML element. For example, using `p` will select all `<p>` elements (paragraphs) in the document.

Example:

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

15.

16. This will make all paragraphs on the webpage green.

#### 2. Class Selector (`.class`)

Class selectors target elements that have a specific class attribute. Class names in CSS are preceded by a dot (`.`).

Example:

HTML:

```
<p class="highlight">This is highlighted text.</p>
<p>This is normal text.</p>
```

17.

18. CSS:

```
.highlight {  
  background-color: yellow;  
}
```

19.

20. In this example, only the paragraph with the `class="highlight"` will have a yellow background.

3. ID Selector (`#id`)

ID selectors target a single element with a specific `id` attribute. In CSS, ID selectors are preceded by a hash symbol (`#`). IDs should be unique and used only once per page.

Example:

HTML:

```
<h1 id="main-title">Main Title</h1>
```

21.

22. CSS:

```
#main-title {  
  font-size: 36px;  
  color: darkred;  
}
```

23.

24. The ID selector `#main-title` applies the styling to the element with `id="main-title"`.

4. Universal Selector (```)

The universal selector (`*`) selects all elements on the page. It is commonly used to reset or apply global styles.

Example:

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

25.

26. This rule removes all default margins and padding from every element on the page.

---

Full Example: Combining CSS Selectors

Here is an example that combines different types of selectors:

HTML:

```
<!DOCTYPE html><html lang="en">
```

```

<head>
  <meta charset="UTF-8">
  <title>CSS Selectors Example</title>
  <style>
    /* Universal selector */
    * {
      margin: 0;
      padding: 0;
    }

    /* Element selector */
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
    }

    /* Class selector */
    .highlight {
      background-color: yellow;
      font-weight: bold;
    }

    /* ID selector */
    #main-title {
      color: darkblue;
      text-align: center;
      font-size: 36px;
    }
  </style>
</head>
<body>
  <h1 id="main-title">Welcome to My Website</h1>
  <p>This is a paragraph of normal text.</p>
  <p class="highlight">This is highlighted text with a yellow background.</p>
</body>
</html>

```

27.

## 28. CSS Breakdown:

1. ``: The universal selector removes the default margin and padding from all elements.
2. body: The element selector applies a global font family and background color.
3. .highlight: The class selector adds a yellow background and bold text to any element with the class="highlight".
4. #main-title: The ID selector styles the main title with dark blue color, centered alignment, and a large font size.

29.

---

## Practice Exercises

1. Styling with Element Selectors:  
Create an HTML page with paragraphs, headings, and a list. Use CSS to style all paragraphs to have a blue text color and all headings to be centered.
2. Using Class Selectors:

Create an HTML document with several paragraphs and images. Assign a class to some of the paragraphs (e.g., `class="highlight"`) and use CSS to change their background color.

3. Working with ID Selectors:

Create a webpage with multiple headings. Use an ID selector to style one specific heading differently from the others (e.g., larger font size and a different color).

4. Combining Selectors:

Write an HTML document and apply global styles using the universal selector (`*`), element selectors, class selectors, and ID selectors in CSS.

30.

---

### Summary

In this lesson, we learned about CSS and its purpose in web development. CSS allows developers to style HTML documents by defining how elements should be displayed on the page. We explored the CSS syntax, which consists of selectors and declarations, and learned about different types of CSS selectors, including element selectors, class selectors, ID selectors, and the universal selector.

### Key Takeaways:

1. CSS separates content (HTML) from design, making websites easier to manage and style.
2. CSS syntax is made up of selectors and declarations that apply specific styles to HTML elements.
3. Selectors target specific HTML elements: element selectors target all elements of a type, class selectors target elements with a specific class, and ID selectors target one unique element.

31. Additional Resources

1. [MDN Web Docs: CSS Syntax](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_syntax/Syntax)
2. [CSS Selectors - MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_selectors)

[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_syntax/Syntax](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_syntax/Syntax)

[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_selectors](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_selectors)

32.

---

### Adding CSS to HTML, CSS Rules, Colors and Backgrounds, and Text Styling

#### Introduction

CSS (Cascading Style Sheets) allows you to add style and visual design to your web pages. There are multiple ways to apply CSS to HTML, and knowing when and how to use each method is crucial for creating efficient, maintainable web designs. Beyond applying basic styles, CSS offers powerful tools to control colors, backgrounds, and text styling to make web pages visually engaging.

In this lesson, we will explore how to add CSS to HTML, understand the structure of CSS rules, and learn how to apply colors, backgrounds, and text styling.

## 1. Adding CSS to HTML

There are three main ways to add CSS to an HTML document: inline styles, internal (embedded) styles, and external stylesheets.

### 1.1. Inline Styles

Inline styles apply CSS directly to HTML elements using the `style` attribute. This method is not ideal for large projects, but it can be useful for quick changes or for styling a single element.

Example of Inline Styles:

```
<p style="color: red; font-size: 20px;">This is a red paragraph with larger text.</p>
```

33.

34. In this example, the `style` attribute is used to change the text color to red and increase the font size of the paragraph.

### 1.2. Internal (Embedded) Styles

Internal or embedded styles are placed within the `<style>` tag inside the `<head>` section of the HTML document. This method is good for applying styles to a single page.

Example of Internal Styles:

```
<!DOCTYPE html><html lang="en">
<head>
  <style>
    body {
      background-color: lightblue;
    }
    h1 {
      color: navy;
      text-align: center;
    }
  </style>
  <title>Internal Styles Example</title>
</head>
<body>
  <h1>Welcome to My Website</h1>
  <p>This page has internal styles applied.</p>
</body>
</html>
```

35.

### 1.3. External Stylesheets

External stylesheets are the best way to apply CSS for larger projects. This method involves linking a separate CSS file to your HTML document using the `<link>` tag. External stylesheets allow you to apply consistent styles across multiple pages.

Example of External Stylesheet:

HTML:

```
<!DOCTYPE html><html lang="en">
<head>
  <link rel="stylesheet" href="styles.css">
  <title>External Stylesheet Example</title>
</head>
<body>
  <h1>Welcome to My Website</h1>
  <p>This page uses an external stylesheet.</p>
</body>
</html>
```

37.

38. CSS (styles.css):

```
body {
  background-color: #f0f0f0;
}
```

```
h1 {
  color: darkgreen;
  font-size: 36px;
}
```

```
p {
  font-size: 18px;
  color: #333;
}
```

39.

40. When to Use Each Method

1. Inline styles: For quick, one-time style changes.
2. Internal styles: For styling a single HTML page.
3. External stylesheets: For larger projects and consistent styling across multiple pages.

41.

---

2. CSS Rules: Declarations, Properties, and Values

CSS Rule Structure

CSS rules are made up of selectors, declarations, properties, and values.

1. Selector: Targets the HTML element(s) to style.
2. Declaration block: Contains one or more declarations enclosed in curly braces {}.
3. Declaration: Each declaration consists of a property (what you want to style) and a value (how you want to style it), separated by a colon :.

42. Example of a CSS Rule:

```
h1 {
  color: blue;
  font-size: 24px;
}
```

43.

1. Selector: h1 (applies the styles to all <h1> elements).
2. Declaration block: { color: blue; font-size: 24px; }



3. Properties: color, font-size

4. Values: blue, 24px

#### 44. Multiple Declarations

You can apply multiple declarations to a single element by separating them with semicolons.

```
p {  
  color: gray;  
  font-size: 18px;  
  text-align: center;  
}
```

45.

46.

---

### 3. Colors and Backgrounds in CSS

CSS provides several ways to apply color, background images, and even gradients to HTML elements, helping to enhance visual appeal.

#### 3.1. Applying Colors

You can specify colors in CSS using:

1. Named colors: (e.g., red, blue, green).
2. Hexadecimal values: (e.g., #ff0000 for red).
3. RGB values: (e.g., rgb(255, 0, 0) for red).

47. Example:

```
h1 {  
  color: #ff0000; /* Red using hex */  
}  
  
p {  
  color: rgb(0, 128, 0); /* Green using RGB */  
}
```

48.

#### 49. 3.2. Background Colors

The background-color property changes the background color of an element.

Example:

```
body {  
  background-color: #f0f0f0;  
}  
  
div {  
  background-color: lightyellow;  
}
```

50.

#### 51. 3.3. Background Images

The background-image property is used to set an image as the background of an element.

Example:

```
body {
  background-image: url('background.jpg');
  background-size: cover;
  background-repeat: no-repeat;
}
```

52.

1. `background-size: cover;`: Ensures the background image covers the entire area of the element.
2. `background-repeat: no-repeat;`: Prevents the image from repeating.

### 53. 3.4. Gradients

CSS also supports gradients, which are smooth transitions between colors.

Example of a Linear Gradient:

```
body {
  background: linear-gradient(to right, red, yellow);
}
```

54.

55. 

---

## 4. Text Styling in CSS

CSS allows you to style text in various ways, including changing fonts, sizes, and alignment.

### 4.1. Font Family

The `font-family` property specifies the font of text. You can provide multiple fonts as fallback options in case the browser does not support the first one.

Example:

```
p {
  font-family: Arial, sans-serif;
}
```

56.

### 57. 4.2. Font Size

The `font-size` property controls the size of text. You can use units like `px`, `em`, or `%` to define font sizes.

Example:

```
h1 {
  font-size: 36px;
}
```

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

58.

### 59. 4.3. Text Alignment

The `text-align` property is used to align text. Common values include `left`, `center`, `right`,

and justify.

Example:

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

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

60.

#### 61. 4.4. Font Weight

The `font-weight` property controls the thickness of the text, with values like `normal`, `bold`, or numerical values such as `400` or `700`.

Example:

```
p {
  font-weight: bold;
}
```

62.

#### 63. 4.5. Text Color

You can change the color of the text using the `color` property.

Example:

```
p {
  color: darkblue;
}
```

64.

65.

---

Full Example: Combining Styles for Colors, Backgrounds, and Text

```
<!DOCTYPE html><html lang="en">
<head>
  <link rel="stylesheet" href="styles.css">
  <title>Styled Webpage</title>
</head>
<body>

  <h1>Welcome to My Styled Web Page</h1>
  <p>This is a sample paragraph styled using CSS for color, font, and alignment.</p>

  <div>
    <p>This div has a background color and styled text inside it.</p>
  </div>

</body>
```

</html>

66.

67. CSS (styles.css):

```
body {
    background: linear-gradient(to bottom, #ffcccc, #ffffff);
    font-family: Arial, sans-serif;
}

h1 {
    color: navy;
    text-align: center;
    font-size: 36px;
}

p {
    color: darkgray;
    font-size: 18px;
    text-align: justify;
}

div {
    background-color: lightyellow;
    padding: 20px;
    margin: 10px;
    border: 1px solid #ccc;
}
```

68.

69.

---

#### Practice Exercises

1. Add Inline Styles:

Create an HTML page with a paragraph and use inline styles to change the text color, font size, and background color.

2. Use Internal Styles:

Write an HTML document with a heading and paragraph. Use internal styles in the <head> to change the background color of the body and the font size of the text.

3. External Stylesheet:

Create a webpage that links to an external stylesheet. Style the background, headings, and paragraphs using the external stylesheet.

4. Experiment with Colors and Backgrounds:

Apply a background image to a webpage and use the background-size and background-repeat properties to control its behavior.

5. Text Styling:

Create a webpage with various text elements (headings, paragraphs, links) and style them using different font sizes, colors, alignments, and font weights.

70.

---

#### Summary

In this lesson, we covered how to add CSS to HTML using inline styles, internal (embedded) styles, and external stylesheets. We explored the structure of CSS rules, including declarations, properties, and values. Additionally, we learned how to apply colors, background images, and gradients, and how to style text using fonts, sizes, and alignment.

**Key Takeaways:**

1. Inline styles, internal styles, and external stylesheets are the three main methods for adding CSS to HTML.
2. CSS rules consist of selectors, properties, and values that define how elements are styled.
3. CSS allows for rich text styling, including fonts, sizes, colors, and alignments.
4. You can use background images and gradients to enhance the visual appeal of web pages.

**71. Additional Resources**

1. [MDN Web Docs: CSS Basics](#)
2. [CSS Color and Background - MDN](#)

[https://developer.mozilla.org/en-US/docs/Learn\\_web\\_development/Getting\\_started/Your\\_first\\_website/Styling\\_the\\_content](https://developer.mozilla.org/en-US/docs/Learn_web_development/Getting_started/Your_first_website/Styling_the_content)

[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_backgrounds\\_and\\_borders](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_backgrounds_and_borders)