

# CSS Getting Started

## Getting Started with CSS

In this tutorial you'll learn how easy it is to add style and formatting information to the web pages using CSS. But, before we begin, make sure that you have some working knowledge of HTML.

Without further ado, let's get started with the Cascading Style Sheets (CSS).

## Including CSS in HTML Documents

CSS can either be attached as a separate document or embedded in the HTML document itself. There are three methods of including CSS in an HTML document:

- **Inline styles** — Using the `style` attribute in the HTML start tag.
- **Embedded styles** — Using the `<style>` element in the head section of a document.
- **External style sheets** — Using the `<link>` element, pointing to an external CSS file.

In this tutorial we will cover all these three methods for inserting CSS one by one.

**Note:** The inline styles have the highest priority, and the external style sheets have the lowest. It means if you specify styles for an element in both *embedded* and *external* style sheets, the conflicting style rules in the embedded style sheet would override the external style sheet.

## Inline Styles

Inline styles are used to apply the unique style rules to an element by putting the CSS rules directly into the start tag. It can be attached to an element using the `style` attribute.

The `style` attribute includes a series of CSS property and value pairs. Each `"property: value"` pair is separated by a semicolon ( `;` ), just as you would write into an embedded or external style sheets. But it needs to be all in one line i.e. no line break after the semicolon, as shown here:

### Example

```
<h1 style="color:red; font-size:30px;">This is a heading</h1>
<p style="color:green; font-size:22px;">This is a paragraph.</p>
<div style="color:blue; font-size:14px;">This is some text content.</div>
```

Using the inline styles are generally considered as a bad practice. As style rules are embedded directly inside the HTML tag, it causes the presentation to become mixed with the content of the document; which makes the code hard to maintain and negates the purpose of using CSS.

**Note:** It's become impossible to style [pseudo-elements](#) and [pseudo-classes](#) with inline styles. You should, therefore, avoid the use of style attributes in your code. Using [external style sheets](#) is the preferred way to add styles to the HTML documents.

## Embedded Style Sheets

Embedded or internal style sheets only affect the document they are embedded in.

Embedded style sheets are defined in the `<head>` section of an HTML document using the `<style>` element. You can define any number of `<style>` elements in an HTML document but they must appear between the `<head>` and `</head>` tags. Let's take a look at an example:

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>My HTML Document</title>
  <style>
    body { background-color: YellowGreen; }
    p { color: #fff; }
  </style>
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph of text.</p>
</body>
</html>
```

**Tip:** The type attribute of the `<style>` and `<link>` tag (i.e. `type="text/css"`) defines the language of the style sheet. This attribute is purely informative. You can omit this since CSS is the standard and default style sheet language in HTML5.

## External Style Sheets

An external style sheet is ideal when the style is applied to many pages of the website.

An external style sheet holds all the style rules in a separate document that you can link from any HTML file on your site. External style sheets are the most flexible because with an external style sheet, you can change the look of an entire website by changing just one file.

You can attach external style sheets in two ways — *linking* and *importing*.

### Linking External Style Sheets

Before linking, we need to create a style sheet first. Let's open your favorite code editor and create a new file. Now type the following CSS code inside this file and save it as "style.css".

### Example

```
body {
  background: lightyellow;
  font: 18px Arial, sans-serif;
}
h1 {
  color: orange;
}
```

An external style sheet can be linked to an HTML document using the `<link>` tag. The `<link>` tag goes inside the `<head>` section, as you can see in the following example:

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>My HTML Document</title>
  <link rel="stylesheet" href="css/style.css">
</head>
```

```
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph of text.</p>
</body>
</html>
```

**Tip:** Among all the three methods, using external style sheet is the best method for defining and applying styles to the HTML documents. As you can clearly see with external style sheets, the affected HTML file require minimal changes in the markup.

## Importing External Style Sheets

The `@import` rule is another way of loading an external style sheet. The `@import` statement instructs the browser to load an external style sheet and use its styles.

You can use it in two ways. The simplest is within the header of your document. Note that, other CSS rules may still be included in the `<style>` element. Here's an example:

### Example

```
<style>
  @import url("css/style.css");
  p {
    color: blue;
    font-size: 16px;
  }
</style>
```

Similarly, you can use the `@import` rule to import a style sheet within another style sheet.

### Example

```
@import url("css/layout.css");
@import url("css/color.css");
body {
  color: blue;
  font-size: 14px;
}
```

**Note:** All `@import` rules must occur at the start of the style sheet. Any style rule defined in the style sheet itself override the conflicting rules in the imported style sheets. However, importing a style sheet within another style sheet is not recommended due to performance issue.