

Working with CSS and HTML Together

by Devmountain Tutorials



WHAT'S COVERED

This section will explore how CSS and HTML work together by discussing:

1. INCLUDING CSS IN AN HTML PAGE
2. SELECTING HTML ELEMENTS

1. INCLUDING CSS IN AN HTML PAGE

Now that you understand some basics about how CSS works, we can start integrating CSS into HTML and talk about how they work together. There are a couple different ways to apply CSS to an HTML page.

You can write CSS rules in a file separate from your HTML page. These files are known as **external stylesheets** and they usually have a **.css** file extension (though having a certain file extension is not a requirement). Then, you can reference any external stylesheet in your HTML with a link element. You can add the link anywhere in your HTML, but it usually goes in the page's head element.

Here's an example HTML document with a link that references an external stylesheet called **styles.css**.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Yay, CSS</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <p>Ah, how nice.</p>
  </body>
</html>
```

And here's an example of the CSS in **styles.css**:

```
p {
  font-size: 40px;
```

```
font-family: sans-serif;
}
```

Another way to apply CSS to HTML is to use an **internal stylesheet**. Unlike external stylesheets where you write CSS in a separate file, an internal stylesheet is written directly in your HTML inside a style element. Again, like with external stylesheets, you can place internal stylesheets anywhere in your HTML but it usually goes in the **head** element.

Here's our example again, this time with an internal stylesheet in place of an external one:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Yay, CSS</title>
    <style>
      p {
        font-size: 40px;
        font-family: sans-serif;
      }
    </style>
  </head>
  <body>
    <p>Ah, how nice.</p>
  </body>
</html>
```

You can also add style declarations to an element through its style attribute. Below, we've removed the internal stylesheet and moved the **font-size** and **font-family** declarations into the **p** element's style attribute:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Yay, CSS</title>
  </head>
  <body>
    <p style="font-size: 40px; font-family: sans serif;">
      Ah, how nice.
    </p>
  </body>
</html>
```

CSS declarations in an element's style attribute are known as **inline styles**; so-called because they appear "in line" with the HTML element being styled. Inline styles only apply to one element at a time. Best practices suggest avoiding using inline styles as best you can. Inline styles might make your page harder to change and update, especially since they only apply to one element at a time rather than a group of elements. They also make code harder to read since they mix CSS syntax with HTML syntax.

You have a variety of techniques for applying CSS to HTML — you can reference an external stylesheet with a link element, write CSS directly into the HTML as an internal stylesheet in a style element, or by styling one element at a time with inline styles. There are also different ways to write CSS rules so they select a wider or more narrow group of elements. That's what we'll cover next.



TERMS TO KNOW

External stylesheet

CSS rules in a file separate from your HTML page.

Internal stylesheet

An internal stylesheet is written directly in your HTML inside a style element.

Inline styles

CSS declarations in an element's style attribute. so-called because they appear “in line” with the HTML element being styled. Inline styles only apply to one element at a time.

2. SELECTING HTML ELEMENTS

So far, you've only seen examples of selectors being used to select a type of element, like using the `h1` selector to apply styles to all level-one headings but there are many other types of selectors you have at your disposal. Some selectors select elements with a more narrow degree of specificity — for example, “select only paragraphs that are part of the important class” — and others can select elements based on where they appear relative to other elements — for example, “select the first paragraph of within a section”.

Let's say you have a couple paragraphs to style:

```
<p>Let's put CSS in our HTML.</p>
```

```
<p>We can use an external stylesheet.</p>
```

```
<p>There are also internal stylesheets and inline styles.</p>
```

```
<p>Wow, exciting!</p>
```

You've already seen how to select all `p` elements and change their font.

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

The selector above and the selectors we used in the examples so far are all type selectors. Type selectors are simple to write — they just look exactly like the name of the element you want to select. For example, `h1` will select all `h1` elements and `p` will select all `p` elements.

However, type selectors aren't sufficient when you want to select a more specific subset of elements. For example, we want some paragraphs to have a different color to show that they contain important information. First, it'd be easier for us to distinguish between info-heavy paragraphs and regular paragraphs if we categorized them in a class. We can do that by adding a class to the info-heavy paragraphs.

You can make up whatever class name you'd like — we decided to use `info` since it's short and descriptive:

<p>Let's put CSS in our HTML.</p>

<p class="info">We can use an external stylesheet.</p>

<p class="info">There are also internal stylesheets and inline styles.</p>

<p>Wow, exciting!</p>

Adding the **info** class does two things — it helps us remember that some paragraphs have important information in them and it enables us to use the **class selector** to select and style the info elements so they stand out. **Class selectors** start with a period (.) followed by the name of a class and will select all elements belonging to that class. For example, to select all the elements that are members of the info class, start with the class selector:

```
.info
```

Then we can list style declarations in a pair of curly braces like usual:

```
.info {  
  color: blue;  
}
```

Let's make those paragraphs even more exciting. Edit the HTML and CSS code in the *Editable Code* section to make the last paragraph seem "loud". You'll use HTML to give the last paragraph a class and CSS to select that paragraph and give it a larger font size. Like usual, *Show Solution* will show you the solution and you can start over with *Reset*.



TRY IT

Follow these steps in the box below:

1. In the HTML, give the last paragraph a class attribute and set its value to **"loud"**.
2. In CSS, add a new rule. Start with the class selector **loud** followed by an opening curly brace (**{**).
3. Add a style declaration **font-size: 40px;**.
4. Don't forget the closing curly brace (**}**)!

There are many other types of selectors you can take advantage of. Unfortunately, we won't be able to cover all of them in this course but there are plenty of excellent, free resources you can check out to learn more about different selectors. CSS Tricks is a great website for learning CSS, HTML, and JavaScript and they have an easy-to-follow breakdown on how CSS selectors work, which you can view [by clicking on this link](#). If you like interactive educational materials, CSS Diner is a game where you can learn about different selectors and use them to select plates and food on a table; [you can play the game here](#).



TERMS TO KNOW

External stylesheet

CSS rules in a file separate from your HTML page.

Inline styles

CSS declarations in an element's style attribute, so-called because they appear "in line" with the HTML element being styled. Inline styles only apply to one element at a time.

Internal stylesheet

An internal stylesheet is written directly in your HTML inside a style element.