**Intro to CSS**

While [HTML](https://www.codecademy.com/learn/learn-html) is the fundamental structure of every web page, it can be visually unappealing on its own. Cascading Style Sheets or *CSS* is a language web developers use to style the HTML content on a web page. If you’re interested in modifying colors, font types, font sizes, images, element positioning, and more, CSS is the tool for the job!

In this lesson, you’ll learn how to set up your CSS file structure and select which HTML elements you wish to style.

**Instructions**

**1.**

Take a look at the code in **index.html** and observe how it displays in the browser to the right. This is plain HTML without any styling. Let’s take a quick look at the power of CSS.

Copy and paste the following line of code onto line 5:

<link href="style.css" rel="stylesheet">

What happens when you press the Run button?

Take some time to explore and experiment with the code in the **style.css** file.

**CSS Anatomy**

The diagram on the right shows two different methods, or syntaxes, for writing CSS code. The first syntax shows CSS applies as a ruleset, while the second shows it written as an inline style. Two different methods of writing CSS may seem a bit intimidating at first, but it’s not as bad as it looks!

Notice how both syntaxes contain a declaration. Declarations are the core of CSS. They apply a style to the selected element. Here, the <p> element has been selected in both syntaxes and will be styled to display the text in blue.

Understanding that a declaration is used to style a selected element is key to learning how to style HTML documents with CSS! The terms below explain each of the labels in the diagram on the right.

**Ruleset Terms:**

* Selector—The beginning of the ruleset used to target the element that will be styled.
* Declaration Block—The code in-between (and including) the curly braces ({ }) that contains the CSS declaration(s).
* Declaration—The group name for a property and value pair that applies a style to the selected element.
* Property—The first part of the declaration that signifies what visual characteristic of the element is to be modified.
* Value—The second part of the declaration that signifies the value of the property.

**Inline Style Terms:**

* Opening Tag—The start of an [HTML element](https://www.codecademy.com/courses/learn-html/lessons/intro-to-html/exercises/anatomy-html). This is the element that will be styled.
* Attribute—The style [attribute](https://www.codecademy.com/courses/learn-html/lessons/intro-to-html/exercises/attr-html) is used to add CSS inline styles to an HTML element.
* Declaration—The group name for a property and value pair that applies a style to the selected element.
* Property—The first part of the declaration that signifies what visual characteristic of the element is to be modified.
* Value—The second part of the declaration that signifies the value of the property.

Don’t worry about memorizing all of these—you will get acquainted with them more and more as the course progresses! Feel free to come back and use this exercise as a reference later on.

### Instructions

Study the diagrams to become familiar with the CSS syntax and the new terms that will be used throughout the course.

Click the “Next” button when you are ready to write some code!



**Inline Styles**

Although CSS is a different language than HTML, it’s possible to write CSS code directly within HTML code using inline styles.

To style an HTML element, you can add the style attribute directly to the opening tag. After you add the attribute, you can set it equal to the CSS style(s) you’d like applied to that element.

<p style='color: red;'>I'm learning to code!</p>

The code in the example above demonstrates how to use inline styling. The paragraph element has a style attribute within its opening tag. Next, the style attribute is set equal to color: red;, which will set the color of the paragraph text to red within the browser.

If you’d like to add more than one style with inline styles, simply keep adding to the style attribute. Make sure to end the styles with a semicolon (;).

<p style='color: red; font-size: 20px;'>I'm learning to code!</p>

It’s important to know that inline styles are a quick way of directly styling an HTML element, but are rarely used when creating websites. But you may encounter circumstances where inline styling is necessary, so understanding how it works, and recognizing it in HTML code is good knowledge to have. Soon you’ll learn the proper way to add CSS code!

### Instructions

**1.**

In **index.html**, use inline styling to set the color of the first paragraph (the first <p> element) to green.

**Internal Stylesheet**

As previously stated, inline styles are not the best way to style HTML elements. If you wanted to style, for example, multiple <h1> elements, you would have to add inline styling to each element manually. In addition, you would also have to maintain the HTML code when additional <h1> elements are added.

Fortunately, HTML allows you to write CSS code in its own dedicated section with a [<style> element](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/style) nested inside of the <head> element. The CSS code inside the <style> element is often referred to as an internal stylesheet.

An internal stylesheet has certain benefits and use cases over inlines styles, but once again, it’s not best practice (we’ll get there, we promise). Understanding how to use internal stylesheets is nonetheless helpful knowledge to have.

To create an internal stylesheet, a <style> element must be placed inside of the <head> element.

<head>  
  <style>  
   
   
  </style>  
</head>

After adding opening and closing <style> tags in the head section, you can begin writing CSS code.

<head>  
  <style>  
    p {  
      color: red;  
      font-size: 20px;  
    }  
  </style>  
</head>

The CSS code in the example above changes the color of all paragraph text to red and also changes the size of the text to 20 pixels. Note how the syntax of the CSS code matches (for the most part) the syntax you used for inline styling. The main difference is that you can specify which elements to apply the styling.

### Instructions

**1.**

Let’s move the inline style that was added to the paragraph into an internal stylesheet.

Start by adding an empty <style> element in the head of **index.html**.

Stuck? Get a hint

**2.**

Inside of the <style> element, add a CSS ruleset targeting the paragraph (the <p> element). You can leave the declaration block empty for now.

Stuck? Get a hint

**3.**

Next, place just the declaration from the inline style into the empty declaration block in the inline stylesheet.

Stuck? Get a hint

**4.**

Finally, delete the inline style from the <p> element.

Notice how the styling works the same in the stylesheet as it did in the inline style!

**External Stylesheet**

Developers avoid mixing code by storing HTML and CSS code in separate files (HTML files contain only HTML code, and CSS files contain only CSS code).

You can create an external stylesheet by using the **.css** file name extension, like so: **style.css**

With an external stylesheet, you can write all the CSS code needed to style a page without sacrificing the readability and maintainability of your HTML file.

### Instructions

**1.**

Take a look at **index.html**. Cut the CSS code ruleset in between the opening and closing <style> tags and paste it directly into the file called **style.css**.

Delete the remaining <style> element (now empty) from **index.html** and press the Run button.

Hmm, the font changes didn’t take effect? Click the Next button to find out why.

**Linking the CSS File**

Perfect! We successfully separated structure (HTML) from styling (CSS), but the web page still looks bland. Why?

When HTML and CSS codes are in separate files, the files must be linked. Otherwise, the HTML file won’t be able to locate the CSS code, and the styling will not be applied.

You can use the <link> element to link HTML and CSS files together. The <link> element must be placed within the head of the HTML file. It is a self-closing tag and requires the following attributes:

1. href — like the anchor element, the value of this attribute must be the address, or path, to the CSS file.
2. rel — this attribute describes the relationship between the HTML file and the CSS file. Because you are linking to a stylesheet, the value should be set to stylesheet.

When linking an HTML file and a CSS file together, the <link> element will look like the following:

<link href='https://www.codecademy.com/stylesheets/style.css' rel='stylesheet'>

Note that in the example above, the path to the stylesheet is a URL:

https://www.codecademy.com/stylesheets/style.css

Specifying the path to the stylesheet using a URL is one way of linking a stylesheet.

If the CSS file is stored in the same [directory](https://en.wikipedia.org/wiki/Directory_(computing)) as your HTML file, then you can specify a [relative path](https://www.w3schools.com/html/html_filepaths.asp) instead of a URL, like so:

<link href='./style.css' rel='stylesheet'>

Using a relative path is very common way of linking a stylesheet.

**Instructions**

**1.**

Let’s link the stylesheet **style.css** to the HTML file **index.html**.

First, add a <link> element within the <head> section.

Stuck? Get a hint

**2.**

Next, add the href attribute to the <link> element and set it equal to style.css.

Take a look at the web page in the browser to the right. Do you notice any changes yet?

Stuck? Get a hint

**3.**

Finally, add the rel attribute and set it to the correct value. Keep an eye on the first paragraph’s font—it should appear different from the destinations’ descriptions when your stylesheet is linked correctly.

Stuck? Get a hint

**Review**

Great work so far! By understanding how to incorporate CSS code into your HTML file, as well as learning some of the key terms, you’re on your way to creating spectacular websites with HTML and CSS.

Let’s review what you learned so far:

* The basic anatomy of CSS syntax written for both inline styles and stylesheets.
* Some commonly used CSS terms, such as *ruleset*, *selector*, and *declaration*.
* CSS inline styles can be written inside the opening HTML tag using the style attribute.
* Inline styles can be used to style HTML, but it is not the best practice.
* An internal stylesheet is written using the <style> element inside the <head> element of an HTML file.
* Internal stylesheets can be used to style HTML but are also not best practice.
* An external stylesheet separates CSS code from HTML, by using the “.css”.file extension.
* External stylesheets are the best approach when it comes to using HTML and CSS.
* External stylesheets are linked to HTML using the <link> element.

Take this knowledge to the next lesson, where you start learning how to select HTML elements to style!

Here are a few more resources to add to your toolkit:

* [Codecademy Docs: CSS](https://www.codecademy.com/resources/docs/css)
* [Codecademy Workspaces: CSS](https://www.codecademy.com/workspaces/new)

Make sure to bookmark these links so you have them at your disposal.