Basic CSS Structure & Syntax

**CSS**

You've learned how to separate HTML and CSS into two files and link them together, but you haven't learned how CSS syntax is used to style elements on a web page.

In this lesson, you'll learn the basic structure and syntax of CSS so that you can start styling web page elements.

Let's begin!

**Element Selectors**

We've discussed styling HTML elements using CSS, but how is it actually done?

To style an HTML element using CSS, you must first *select* that element in the CSS file. For example, to style a <p> element, the syntax to select it using CSS is:

p {

}

In the example above, all paragraph elements are selected using a CSS *selector*. The selector (in this case) is p. Note that the CSS selector essentially matches the HTML tag for that element, but without the angle brackets.

**Note:** The p selector in the example above will select *all* <p> elements on the web page. Later in this course, you'll learn how to use more specific CSS selectors so that you can select any element you want.

**CSS Declarations: Anatomy**

It's not enough to simply select an HTML element in a CSS file. To actually style the element, you need to specify two things inside the body of the selector:

1. Property - the property you'd like to style of that element (i.e., size, color, etc.).
2. Value - the value of the property (i.e., 18px for size, Blue for color, etc.).

h1 {

color: Blue;

}

In the example above, the <h1> element has been selected. Inside of the selector's body, the heading's color property is set to a value of Blue. The heading will now appear the color blue in the browser.

The line color: Blue; is referred to CSS *declaration*. A CSS declaration consists of a property and a value. Note that a semicolon (;) ends all declarations.

Finally, the entire snippet of code in the example above is known as a CSS *rule*. A CSS rule consists of the selector and all declarations inside of the selector.

# Multiple Element Selectors

Styling with CSS would be very inefficient if you were forced to manually style the same property across many elements.

For example, what if you wanted to change the color of 10 different elements to Aquamarine in CSS?.

Fortunately, you can select multiple elements at once so that you can save time styling a shared property.

h1, h2, p {

color: Green;

}

In the example above, the <h1> heading, the <h2> heading, and the paragraph have all been styled to appear Green using a multiple element selector. A multiple element selector can save you time when you want to style the same property across many elements.

# The Universal Selector

There's a special selector that can instantly select every single element on the web page: the universal selector.

\* {

font-family: Arial;

}

In the example above, the universal selector, \*, is used to select every element on the page and set the font to Arial.

What makes the universal selector so special? When all elements on a web page require the same styling, it's often more efficient to set that styling using the universal selector. Afterwards, you can modify (or remove) that styling for specific elements that don't require it.

In a later lesson, you'll use the universal selector to help you understand how HTML elements are laid out within the browser and how to position those elements.

# Indentation & Spacing

Just like HTML, CSS follows certain best practices for spacing and indentation.

h1 {

color: blue;

}

p {

color: red;

}

1. One space should be used between the selector and the opening curly brace ({).
2. No extra spaces should exist between opening and closing curly braces ({ and }) and CSS declarations (as in the example above).
3. Two spaces of indentation should be used for CSS declarations.
4. One line of spacing should exist between CSS rules. In the example above, there is one line of spacing between the CSS rule for the heading and the CSS rule for the paragraph.

CSS files can become lengthy as styling is added to a web page. Proper spacing and indentation helps keep CSS code maintainable and readable for you and other developers.

**Comments**

Just like HTML, you can also leave *comments* in your CSS file. CSS comments begin with /\* and end with \*/, like so:

/\* This is a comment in CSS! \*/

Including comments in your code is helpful for many reasons:

1. They help you (and others) understand your code if you decide to come back and review it at a much later date.
2. They allow you to experiment with new code, without having to delete old code.

/\* Paragraph Styling \*/

p {

color: Blue;

}

In the example above, a comment is used to specify CSS styling for paragraphs.

/\*

h1 {

color: Red;

}

\*/

In the example above, a valid CSS rule has been "commented out." This practice is useful when you want to experiment with new code without having to delete old code.