**CSS (Cascading Style Sheets)**

****

**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/resources/docs/html/elements?page_req=catalog). This is the element that will be styled.
* Attribute—The style [attribute](https://www.codecademy.com/resources/docs/html/attributes?page_req=catalog) 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.

**Conclusion**

* 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.

**Pseudo-class**

ou may have observed how the appearance of certain elements can change, or be in a different state, after certain user interactions. For instance:

* When you click on an <input> element, and a blue border is added showing that it is in *focus*.
* When you click on a blue <a> link to *visit* to another page, but when you return the link’s text is purple.
* When you’re filling out a form and the submit button is grayed out and *disabled*. But when all of the fields have been filled out, the button has color showing that it’s *active*.

These are all examples of pseudo-class selectors in action! In fact, :focus, :visited, :disabled, and :active are all pseudo-classes. Factors such as user interaction, site navigation, and position in the document tree can all give elements a different state with pseudo-class.

A pseudo-class can be attached to any selector. It is always written as a colon : followed by a name. For example p:hover.

p:hover {  
  background-color: lime;  
}

In the above code, whenever the mouse hovers over a paragraph element, that paragraph will have a lime-colored background.

**Selectors**

* CSS can select HTML elements by type, class, ID, and attribute.
* All elements can be selected using the universal selector.
* An element can have different states using the pseudo-class selector.
* Multiple CSS classes can be applied to one HTML element.
* Classes can be reusable, while IDs can only be used once.
* IDs are more specific than classes, and classes are more specific than type. That means IDs will override any styles from a class, and classes will override any styles from a type selector.
* Multiple selectors can be chained together to select an element. This raises the specificity but can be necessary.
* Nested elements can be selected by separating selectors with a space.
* Multiple unrelated selectors can receive the same styles by separating the selector names with commas.