**My Notes on HTML5 and CSS3**

* HTML, HyperText Markup Language, gives content structure and meaning by defining that content as, for example, headings, paragraphs, or images.
* CSS, or Cascading Style Sheets, is a presentation language created to style the appearance of content—using, for example, fonts or colors.
* As a rule, HTML will always represent content, and CSS will always represent the appearance of that content.
* HTML is a [**markup language**](http://webdesign.about.com/od/htmlxhtmltutorials/p/what-are-markup-languages.htm), which means that it is written with codes that can be read by a person, without needing to be compiled. In other words, the text on a web page is “marked up” with these codes to tell the browser how to display the text.
* **Understanding Common HTML Terms**

### Elements

Elements are designators that define the structure and content of objects within a page. Some of the more frequently used elements include multiple levels of headings (identified as <h1> through <h6> elements) and paragraphs (identified as the <p>element); the list goes on to include the <a>, <div>, <span>, <strong>, and <em>elements, and many more.

Elements are identified by the use of less-than and greater-than angle brackets, < >, surrounding the element name.

### Tags

The use of less-than and greater-than angle brackets surrounding an element creates what is known as a tag. Tags most commonly occur in pairs of opening and closing tags.

The content that falls between the opening and closing tags is the content of that element. An anchor link, for example, will have an opening tag of <a> and a closing tag of </a>. What falls between these two tags will be the content of the anchor link.

### Attributes

Attributes are properties used to provide additional information about an element.

Attributes are defined within the opening tag, after an element’s name. Generally attributes include a name and a value. The format for these attributes consists of the attribute name followed by an equals sign and then a quoted attribute value.

Ex:



* What is the Difference between HTML element and tag?

HTML tag is just opening or closing entity. For example:

<p> and </p> are called HTML tags

HTML element encompasses opening tag, closing tag, content (optional for content-less tags) Eg:

<p>This is the content</p> : This complete thing is called a HTML element

## HTML Document Structure

## HTML documents are plain text documents saved with an .html file extension rather than a .txt file extension.

## All HTML documents have a required structure that includes the following declaration and elements: <!DOCTYPE html>, <html>, <head>, and <body>.

## The document type declaration, or <!DOCTYPE html>, informs web browsers which version of HTML is being used and is placed at the very beginning of the HTML document. Because we’ll be using the latest version of HTML, our document type declaration is simply <!DOCTYPE html>.

## Following the document type declaration, the<html> element signifies the beginning of the document. Inside the <html> element, the <head> element identifies the top of the document, including any metadata (accompanying information about the page).

## The content inside the <head> element is not displayed on the web page itself. Instead, it may include the document title (which is displayed on the title bar in the browser window), links to any external files, or any other beneficial metadata.

## All of the visible content within the web page will fall within the <body> element. A breakdown of a typical HTML document structure looks like this:

## 

## 

## 

* HTML Validator (W3C):
  + http://validator.w3.org/
* CSS Validator (W3C):
  + http://jigsaw.w3.org/
* Hypertext is text displayed on a [computer display](https://en.wikipedia.org/wiki/Computer_display) or other [electronic devices](https://en.wikipedia.org/wiki/Electronic_devices) with references ([hyperlinks](https://en.wikipedia.org/wiki/Hyperlinks)) to other text which the reader can immediately access, or where text can be revealed progressively at multiple levels of detail (also called[StretchText](https://en.wikipedia.org/wiki/StretchText)).

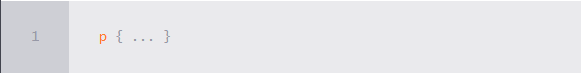
## Understanding Common CSS Terms

### Selectors

A selector designates exactly which element or elements within our HTML to target and apply styles (such as color, size, and position) to.

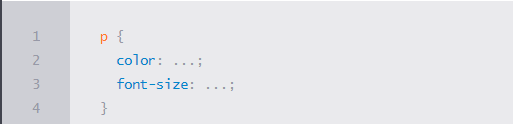
Selectors generally target an attribute value, such as an id or class value, or target the type of element, such as <h1> or <p>.

Within CSS, selectors are followed with curly brackets, {}, which encompass the styles to be applied to the selected element. The selector here is targeting all <p> elements.



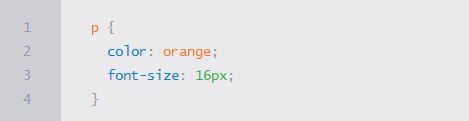
### Properties

Once an element is selected, a property determines the styles that will be applied to that element. Property names fall after a selector, within the curly brackets, {}, and immediately preceding a colon, :. There are numerous properties we can use, such asbackground, color, font-size, height, and width, and new properties are often added.

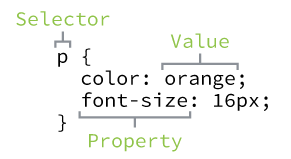


### Values

So far we’ve selected an element with a selector and determined what style we’d like to apply with a property. Now we can determine the behavior of that property with a value. Values can be identified as the text between the colon, :, and semicolon, ;. Here we are selecting all <p> elements and setting the value of the color property to be orange and the value of the font-size property to be 16 pixels.



In CSS our rule set begins with the selector, which is immediately followed by curly brackets. Within these curly brackets are declarations consisting of property and value pairs. Each declaration begins with a property, which is followed by a colon, the property value, and finally a semicolon.

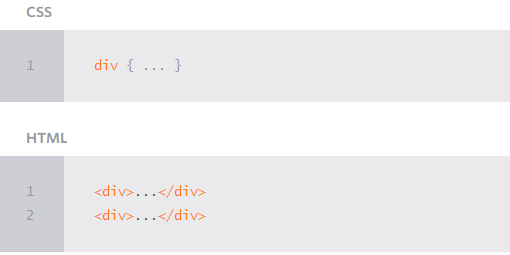


## Working with Selectors

## Selectors, indicate which HTML elements are being styled.

### Type Selectors

Type selectors target elements by their element type. For example, should we wish to target all division elements, <div>, we would use a type selector of div.



### Class Selectors

Class selectors allow us to select an element based on the element’s class attribute value. Class selectors are a little more specific than type selectors, as they select a particular group of elements rather than all elements of one type.

Class selectors allow us to apply the same styles to different elements at once by using the same class attribute value across multiple elements.

Within CSS, classes are denoted by a leading period, ., followed by the class attribute value.



### ID Selectors

ID selectors are even more precise than class selectors, as they target only one unique element at a time. Just as class selectors use an element’s class attribute value as the selector, ID selectors use an element’s id attribute value as a selector.

Regardless of which type of element they appear on, id attribute values can only be used once per page. If used they should be reserved for significant elements.

Within CSS, ID selectors are denoted by a leading hash sign, #, followed by the idattribute value.

## 

## Referencing CSS

## The best practice for referencing our CSS is to include all of our styles in a single external style sheet, which is referenced from within the <head> element of our HTML document. Using a single external style sheet allows us to use the same styles across an entire website and quickly make changes sitewide.

## Within the <head> element of the HTML document, the <link> element is used to define the relationship between the HTML file and the CSS file. Because we are linking to CSS, we use the rel attribute with a value of stylesheet to specify their relationship. Furthermore, the href (or hyperlink reference) attribute is used to identify the location, or path, of the CSS file.

## 

## Using CSS Resets

## Every web browser has its own default styles for different elements. How Google Chrome renders headings, paragraphs, lists, and so forth may be different from how Internet Explorer does. To ensure cross-browser compatibility, CSS resets have become widely used.

## CSS resets take every common HTML element with a predefined style and provide one unified style for all browsers. These resets generally involve removing any sizing, margins, paddings, or additional styles and toning these values down. Because CSS cascades from top to bottom—more on that soon—our reset needs to be at the very top of our style sheet. Doing so ensures that those styles are read first and that all of the different web browsers are working from a common baseline.

## There are a bunch of different resets available to use, all of which have their own fortes. One of the most popular resets is [Eric Meyer’s reset](http://meyerweb.com/eric/tools/css/reset/), which has been adapted to include styles for the new HTML5 elements.

## If you are feeling a bit more adventurous, there is also [Normalize.css](http://necolas.github.io/normalize.css/), created by Nicolas Gallagher. Normalize.css focuses not on using a hard reset for all common elements, but instead on setting common styles for these elements. It requires a stronger understanding of CSS, as well as awareness of what you’d like your styles to be.

## 

**Resources:**

**HTML & CSS BOOK:** The book Learn to code HTML and CSS - <http://learn.shayhowe.com/>

* This Tutorial prepared based on the above book/link.

**Common HTML Terms**: <http://www.scriptingmaster.com/html/HTML-terms-glossary.asp>

**CSS Terms & Definitions:** <https://www.impressivewebs.com/css-terms-definitions/>