HTML Jargon

As you learn and read about HTML/CSS, a few terms may come up over and over again. The following is a short list of some of these terms and what they mean.

Semantic

Literally, it refers to the *logical* meaning of an HTML element. All HTML tags/elements have a **meaning** and are used, not only to structure your content, but to *classify* your content. For example, the P element (<p> tag) represents a **paragraph**. You should not use this element for random words (e.g. "Date: 09/09/2016") if those words are not a paragraph because this is **semantically** incorrect (i.e. it is logically **NOT** a paragraph).

Do not fall into the habit of using *any* expedient HTML element to structure your content layout. Classifying your content illogically has a negative impact on accessibility.

REMEMBER: All HTML elements have a logical or semantic meaning and represent different things. Use appropriate semantic tags to classify your content.

Element

This refers to one part or *piece* of HTML. For example, the paragraph *element* is represent by <p> and is enclosed in <p></p>.

Tag

All elements have tags. A tag is the piece enclosed in the left angle bracket (<) and the right angle bracket (>). Some elements have an **opening** and **closing** tag.

Example of a paragraph *element* **with opening and closing** *tags***:**

```
<p>This is a paragraph.</p>
```

Some element tags are *self-closing*.

Example of an image element with a self-closing tag:

```
<img src="image.jpg" alt="A random image" />).
```

Self-closing tags are single elements without a closing tag. In other words, they close themselves by ending in "/>". The slash (/) denotes the closing. In straight HTML (not XHTML), you technically do not need the slash but I always include it because it doesn't hurt and is a good habit. Other elements which are self-closing are line break (<br />), meta data (<meta ... />), horizontal rule (<hr />).
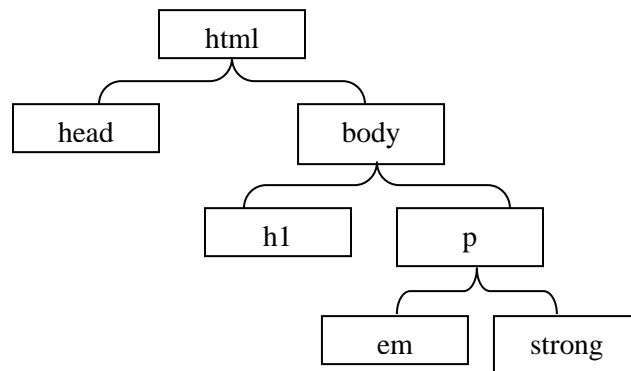
Nesting

Nesting refers to elements within elements. See the code below:

```
<p>This is a <strong>paragraph</strong>.</p>
```

The <strong> element is <u>*nested*</u> within the <p> element.

Descendant

HTML documents can be viewed as trees similar to a family tree (see figure below) so, many times, the locations of elements in HTML documents *in relation to other elements* are often described using genealogical terms.

```
              ┌────────┐
              │  html  │
              └────────┘
           ┌──────┴──────┐
      ┌────────┐     ┌────────┐
      │  head  │     │  body  │
      └────────┘     └────────┘
                   ┌─────┴─────┐
               ┌──────┐    ┌──────┐
               │  h1  │    │  p   │
               └──────┘    └──────┘
                        ┌─────┴─────┐
                    ┌──────┐   ┌────────┐
                    │  em  │   │ strong │
                    └──────┘   └────────┘
```

Child

Take a look at the HTML document tree figure above. Keeping in mind the family tree analogy, **p** is a child of **body** but is <u>not</u> a child of **html**.

Parent

Looking at the HTML document tree example, **body** is a *parent* of **h1** and **p** but is not a parent of **em** or **strong**. (You would not call your grandmother your parent. She is your grandparent but not your parent.)

Ancestor

Again, using the figure above, **html** is an ancestor of **p** and of **strong**.

Attribute

Attributes are extra behaviours of info you can define about an HTML element. This is usually denoted by

```
attribute_name="attribute_value"
```

within an **element's opening tag**. (In the case of a self-closing tag, attributes will be defined in the self-closing tag.

Attributes allow you to add or modify default behaviours for an element.

Take a look at the following example:

```
<a href="https://www.google.ca">Google link</a>
```

In the above example, there is an attribute defined called **href**. This attribute is defined to point to Google's home page.