HTML - *Hyper Text Markup Language* Programming

This document will teach you the basic rules of HTML along with a few common tags. Keep in mind that this does not define every HTML tag out there; quite a few HTML tags exist, and plenty of books and websites such as http://www.w3schools.com/html/default.asp are devoted to HTML and provide comprehensive command lists. *Later on and as you search the web, you will learn more about XHTML which is "a reformulation of the three HTML 4 document types as applications of XML 1.0". The World Wide Web Consortium (W3C) maintains that HTML 4.01 is the recommendation for hypertext markup.* If you learn the basic rules of using HTML tags, you'll be able to use any of the tags you discover online or in HTML books. *You can use any text editor such as Notepad, **Brackets**, Word, WordPad, etc. to write HTML. You MUST know HTML and how to use a text editor in order to have flexibility and full control over your pages* even when using a professional web editor such as Dreamweaver. There will be times when using Dreamweaver that you will want to do something and will have to modify the code itself to make it work. *One of the biggest advantages of using a professional web editor is that it is much faster and lets you see the web page during development.* As you learn, you will begin to discover how to *use additional languages like CSS (Cascading Style Sheets) to apply formatting to multiple pages.*

(Basic HTML Example)

```
<!DOCTYPE html>
<html>
<head>
<title>A Simple HTML Example</title>
</head>
<body>
<h1>Heading sizes range from h1 through h6, with h1 being
the largest heading size.</h1>
<p>This is the first paragraph. Notice that most tags have
a beginning and ending indicated by a forward slash (/) and
are called two-sided.</p>
<p>This is the second paragraph.</p>
</body>
</html>
```

The <!DOCTYPE> declaration must be the very first thing in your HTML document, before the <html> tag. The <!DOCTYPE> declaration is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.

This is the basic code that makes up a web page. This first document, the home page, will be named *index.html* and saved in your public_html folder that you copied to the desktop. *Note that the <html></html> tags tell the browser where the page starts and stops and all tags are enclosed in angle brackets < and >.*

(Center Alignment via CSS)

```
<html>
<head>
<title>A Simple HTML Example</title>
</head>
<body>
<h1>HTML is Easy To Learn</h1>
<p>This is the first paragraph. Notice that
most tags have a beginning and ending indicated by a forward
slash (/) and are called two-sided.</p>
<p style="text-align:right">This is the second paragraph.</p>
</body>
</html>
```

The code that you see in bold is all that you will add. When added, save your work and look at the page in your browser to see the changes. You might have to hit F5 to refresh the page and see the new content. Note that you are using CSS instead of HTML and this is called an inline style because it was added to the HTML p tag.

(Forced Line Break)

```
This<br>
will<br>
create forced line breaks WITHOUT white space/blank lines.<br>

</body>
</html>
```

This tag will allow you to push content down the page without having a blank line between it and whatever is below as you saw with the <p> tag. Using the <p> tag is exactly like hitting the Enter key. Using the <br> tag is exactly like using Shift-Enter. Notice that this tag is one-sided. The code that you see in bold will go after your last paragraph and before where the body ends. Only add what you see in bold! </body> and </html> will always be at the end and are shown here as an example.

(HTML Styles for text color, font family, alignment, and text size)

In this section you are working with HTML styles/formatting and again using CSS.

```
<p style=”color:blue”>Paragraph 1</p>
<br>
<p style=”font-family:courier”>Paragraph 2</p>
<br>
<p style=”text-align:center”>Paragraph 3</p>
<br>
<p style=”font-size:50px”>Paragraph 4</p>

</body>
</html>
```

(Lists)

(Unnumbered, Bulleted List)

```
<ul>
<li>oranges</li>
<li>peaches</li>
</ul>

</body>
</html>
```

(Numbered List)

```
<ol>
<li>oranges</li>
<li>peaches</li>
</ol>

</body>
</html>
```

(List w/Squares)

```
<ul style=”list-style-type:square”>
<li> oranges</li>
<li> peaches</li>
</ul>

</body>
</html>
```

(List w/Letters)

```
<ol type=”A”>
<li> oranges</li>
<li> peaches</li>
</ol>

</body>
</html>
```

Here you are creating four different lists. When you finish the first one (unnumbered lists), hit the enter key and add the next, etc.

(Definition List – Definition Term, Definition Definition)

```
<dl>
<dt> Coffee </dt>
<dd> - black hot drink </dd>
<dt> Milk </dt>
<dd> - white cold drink </dd>
</dl>

</body>
</html>
```

(Comments)

```
<!-- This is a comment -->
<p>This is a paragraph.</p>

</body>
</html>
```

(Iframes)

```
<iframe src=”https://www.google.com” width=”500” style=”border:medium solid red”></iframe>

</body>
</html>
```

(Nested List)

```
<ul>
<li> A few New England states:
    <ul>
    <li> Vermont
    <li> New Hampshire
    <li> Maine
    </ul>
<li> Two Midwestern states:
    <ul>
    <li> Michigan
    <li> Indiana
    </ul>
</ul>

</body>
</html>
```

(Blockquote)

```
<blockquote> This will offset a paragraph from the regular body text, usually by indenting the paragraph’s left and right margins. </blockquote>
```

(Preformatted Text)

```
<pre>This text will k e   e p the formatting y   o  u specify.</pre>

</body>
</html>
```

This tag will allow you to format text however you wish and have more control over your page. If you were to add spacing between letters without this tag, the spaces will be reduced to one.

(Horizontal Rules)

```
<hr size=4 width="50%">

</body>
</html>
```

This creates a separator for content within a page. You can increase or decrease the width by changing the %.

(Bold, Italic, Underline, Highlight, Subscript, Strikeout)

```
<b>Bold</b>
<br>
<i>Italic</i>
<br>
<u>Underline</u>
<br>
<em>Italic</em>
<br>
<strong>Bold</strong>
<br>
<mark>Highlight</mark>
<br>
<sub>Subscript</sub>
<br>
<del>Strikeout</del>
<br>

</body>
</html>
```

Remember that these are two-sided tags and MUST have an ending (/) or they will affect everything below them. *Note that <em></em> can be used for italic text (emphasize) and <strong></strong> can be used for bold (extra importance) as well.*

(Links/URLs - *Uniform Resource Locator*)

```
<a href=http://www.google.com target=”_blank”>Click here to go to Google’s Website!</a>

</body>
</html>
```

This makes an external link to Google’s website. *Using the target=”_blank” command will cause a link to open into a new window* which should be used for ALL external links.

```
<nav>
  <a href="index.html">Home</a> | <br>
  <a href="about.html">About Us</a> | <br>
  <a href="staff.html">Our Staff</a> | <br>
  <a href="contact.html">Contact Us</a>
</nav>

</body>
</html>
```

The nav tag should be used for a group of links such as the navigation for all of the internal links/pages in your website.

(Change Link Colors)

```
<style>
/* unvisited link */
a:link {
    color: red;
}

/* visited link */
a:visited {
    color: green;
}

/* mouse over link */
a:hover {
    color: hotpink;
}

/* selected link */
a:active {
    color: blue;
}
</style>
```

You will notice that links are blue by default and turn purple after being clicked. You can control their colors by using these CSS codes.

- a:link - a normal, unvisited link
- a:visited - a link the user has visited
- a:hover - a link when the user mouses over it
- a:active - a link the moment it is clicked

Note that the primary colors are red, green and blue. T**hese codes are enclosed within <style> tags and appear in the heading (<head></head>) of your document. You will have to scroll back up to the top and add them.** The heading tags do not need to be added again.

(Audio)

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio tag.
</audio>
```

(Video)

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

(Editable Content)

```
<p contenteditable="true">This is an editable paragraph.</p>
```

(SVG Graphics - Paths, Boxes, Circles, Text, etc.)

```
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>
```

(Mailto)

```
<br>
<a href="mailto:you@youremailaddress.com">E-mail me!</a>

</body>
</html>
```

This code will open a local e-mail application such as Microsoft Outlook. The problem with that is the majority of people use web-based e-mail such at Hotmail or Yahoo. This will not work for them when they click the link. What should you do?

(Images/Pictures)

```
<br>
<img src="image.jpg" alt=”HC Football” width=”250” height=”400”>

</body>
</html>
```

I have already placed a file in your public_html folder named image.jpg. If you have another image that you want to use, replace the file name in the code. *The alt command provides a text label about the image in case it does not download.* Generally, images will not be in the root folder, but will be placed into a sub-folder called images. *Standard images/photos have an extension like .jpg. Animated images, banners, buttons, clipart, etc. typically have an extension of .gif. Gif and Jpg are the two main types of images. Note that you can also change the size of an image in HTML by using height and width as seen above. Hspace can be used to add space to the left and right of your image, separating it or providing a gap between it and text.*

(Insert a Table)

```
<table style=”width:300px” border="5px solid red”>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>

</body>
</html>
```

This will insert a simple HTML table, containing two columns and two rows. Tables are a must to properly display content within your pages. *th defines a header cell (column), tr defines a table row and td defines a table cell where data is contained. You can also add background colors to tables which allow you to add colors to certain areas of the page and not the entire background.* You should research adding colors to tables and changing their width.

(Background Color)

```
<html>
<head>
<title>A Simple HTML Example</title>
</head>
<body style=”background-color:yellow">
<h1>HTML is Easy To Learn</h1>
```

*You will now be working **back at the top of your document** and adding this code to your body tag to change the background color.* You can use a hexadecimal color instead of a primary color name such as red, blue, or green. For example, the hex code #ff0000 will represent the color red. *#000000 will represent black.* You need to do a Google image search for ‘hexadecimal color code chart’ and save it to your workstation. It will have a number of hex codes that you will use later. *Note that there are 216 browser safe colors.*

(Background Image)

```
<html>
<head>
<style>  body {
    background-image: url("background.jpg");
    }
</style>
<title>A Simple HTML Example</title>
</head>
<body>
<h1>HTML is Easy To Learn</h1>
```

Now you are using a file for your background. I have an image, background.jpg, in your public_html folder already. You are going to notice that it will not fill the entire screen and will tile itself in order to fill the gaps. This is something that you can’t use in the real world and must be corrected. What can you do to fix the issue?

Also, notice that you are using a CSS inline style.

```
<html>
```

The title indicates what the page is about and is displayed in the title bar located at the very top of the browser. It, along with meta tags are used by search indexers to find your page(s) when someone searches via Google, etc. Scroll back up to the top of your document to change the title.

```
<head>
<title>My Webpage!</title>
</head>
```

(Forms)

```
<form action="mailto:youremail@outlook.com" method="post" enctype="text/plain"
name="sampleform" target="_self" id="sampleform">
  <label>Name
        <input type="text" name="Name" id="Name" />
  </label> <br>
  <label>
        <input type="radio" name="Yes/No" value="yes" id="Yes/No_0" />Yes
  </label> <br>
  <label>
          <input type="radio" name="Yes/No" value="no" id="Yes/No_1" />No
  </label> <br>
  <label>Comments
        <textarea name="Comments" id="Comments" cols="45" rows="5"></textarea>
  </label> <br>
  <label>Drop Down
        <select name="Drop Down" id="Drop Down">
        <option value="1" selected="selected">Item 1</option>
        <option value="2">Item 2</option>
        </select>
   </label> <br>
   <label>
        <input type="submit" name="button" id="button" value="Submit" />
        <input type="reset" name="button2" id="button2" value="Reset" />
   </label>
</form>

</body>
</html>
```

(Footer)

```
<footer>
  <p>Posted by: John Smith</p>
  <p>Contact information: <a href="mailto:someone@example.com">
  someone@example.com</a>.</p>
</footer>

</body>
</html>
```

(Meta Tags)

```
<hmtl>
<head>
<title>My Webpage!</title>
<meta name="description" content="Tutorial on how to build an HTML website">
<meta name="keywords" content="HTML, CSS, Coding, Website, Tags, Links">
<meta http-equiv="refresh" content="30">
<meta charset="UTF-8">
</head>
```

HTML meta tags are officially page data tags that lie between the open and closing head tags in the HTML code of a document. The text in these tags is not displayed, but parsable and tell the browsers (or other web services) specific information about the page. Simply, it "explains" the page so a browser can understand it. *Metatags are specifically used to store information usually relevant to browsers and search engines. Note that HTML 4.01 and HTML 5 handle meta tags differently.* More information can be found at http://www.w3schools.com/tags/tag_meta.asp. This example also provides a tag to refresh or reload the page every 30 seconds which is useful if you have a dynamic page that receives new content regularly.

*This completes your first page

Now that you know the basics of HTML and adding CSS styles to HTML tags, you are required to build a second page and create links between it and the homepage. Name the second page CSS.html. Use CSS to define all of the styles for the page which will appear within the heading and not 'inline' with the HTML tags. Feel free to change or add styles and content as you wish. When finished, let me see your work. This will be your first graded assignment and will lay the foundation for other HTML/CSS projects that we will complete going forward.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: lightblue;
    font-family: "Courier New";
}

h1 {
    color: white;
    text-align: center;
    font-family: Verdana;
}

h2 {
    color: green;
    margin-left: 20px;
    font-family: Verdana;
}

p {
    font-family: verdana;
    font-size: 20px;
    border-style: dotted;
    border-width: 4px;
    font-style: italic;
    font-weight: bold;
}

/* unvisited link */
a:link {
    background-color: lightblue;
}

/* visited link */
a:visited {
    background-color: lightblue;
}

/* mouse over link */
a:hover {
    background-color: yellow;
}

/* selected link */
a:active {
    background-color: yellow;
}
</style>
</head>
<body>
```

```
<h1>My First CSS Example</h1>
<p>This is the second page of my HTML website and will demonstrate how to use CSS to define the styles for my page. In my home page, I used CSS in a number of spots as inline styles. I understand that an inline style loses many of the advantages of a style sheet (by mixing content with presentation) and should be used sparingly.</p>
<h2>Navigation</h2>
<nav>
  <a href="index.html">Home</a> |
  <a href="CSS.html">CSS</a>
</nav>
<p>This is the second and final paragraph of this page.</p>
</body>
</html>
```