HTML Tags (tutorial 1):

At the end of this tutorial, you should know the html tags for a template, a paragraph, different headers, lists, and tables.

Starting Out:

Open your **template** file.

Save it under another name (as an html file). Now you should have your template file and a new file that is an exact copy of the template file. This new file is the file you will edit (in either notepad++ or textwrangler).

Template Tags:

A few notes on the template:

1. Every template tag MUST OCCUR ONLY ONCE IN YOUR WEB PAGE (opening and closing)
2. Every template tag MUST OCCUR IN THE ORDER YOU SEE THEM IN THE TEMPLATE
3. You make place ONLY TEXT between the opening and closing <title> tag.
4. You may place other information between the opening and closing <head> tag (but we won't until later in this course.
5. All other html tags MUST GO BETWEEN THE OPENING AND CLOSING BODY TAGS.

To summarize: Add a title between the opening and closing title tags, and add other html tags between the opening and closing body tags. Otherwise, don't touch the template.

HTML Body Tags:

Here is an example of a simple web page using the template, with a title added at the top and other html tags added inside the body. I will explain the tags below.

Note that:

- we only added tags between the opening and closing body tag, and
- the tags added can be added more than once.

For instance, I've got 3 <p> tags in the body and two <h2> tags in the body as well.

```
<!doctype html>
<html lang="en-US">
   <head >
      <meta charset="utf-8">
      <title>My First Web Page</title>
   </head>
   <body>
      <h1> This is the header of my first web page </h1>
      <h2> This is a subheader on my web page </h2>
      <p> This is a lovely paragraph about anything I want.  Perhaps it's
          about puppies, or maybe kittens, or possibly monster trucks.  It
          could even be about zombies.  It can be on anything I want.
      </p>
      <p> This is a second paragraph. It is equally lovely. The browser knows
          it’s a paragraph because I’ve put the p tags around it.
      </p>
      <h2> This is a second subheader, indicated by the h2 tag </h2>
      <p> And this is a third paragraph.  Short and sweet, largely because
          I’ve run out of things to say.
      </p>

   </body>
</html>
```

Try this by adding the content between the opening and closing body tag to your new html file. Save it. Bring it into your browser by double-clicking on it. Now you can see what these basic html tags look like in your browser.

Note that with most tags (but not all tags) ***if you open it, you must close it.*** In other words, <p> is an opening tag that goes before a paragraph, and </p> is a closing tag that goes at the end of the paragraph.

Paragraphs:

The tags you used in the above web page were:

<p> </p>

This tag should go around all paragraphs in your web page.

Headers:

<h1></h1>

This tag goes around any major header you want on your page. Note that you can have as many h1 tags on your page as you want.

<h2></h2>

This tag goes around any subheader you want on your page. NOTE that you can have as many subheaders (h2) tags on your page as you want.

There are also:

- <h3></h3> For subsubheaders
- <h4></h4> For subsubsubheaders
- <h5></h5> For subsubsubsubheaders; and
- <h6></h6> For subsubsubsubsubheaders.

(There is no <h7></h7> tag).

Note that any of these tags can be used as many times as you want.

Lists:

To add a numbered (ordered) list, include the following tags:

```
<ol>
   <li> list item 1 </li>
   <li> list item 2 </li>
   <li> list item 3 </li>
 </ol>
```

To add a bulleted (unordered) list, include the following tags:

```
<ul>
    <li> list item 1</li>
    <li> list item 2</li>
    <li> list item 3</li>
</ul>
```

The <ol></ol> indicates that the list items will be numbered (ordered), and the <ul></ul> tags indicate that the list items will be bulleted (unordered).

The **ONLY THING** that can go inside the <ol></ol> tag or the <ul></ul> tag are <li></li> (or list items).

Now, inside the <li></li>, you can put text, images (I'll show you how later), links (again, I'll show you how later), or anything else you want. For now, just add text.

NOTE: while this is technically CSS, you can choose from different styles of bullets by specifying the list-style-type. The predefined style types include: disc, circle, square, and none. To try these, do the following:

```
<ul style = "list-style-type:square">
    <li> list item 1</li>
    <li> list item 2</li>
    <li> list item 3</li>
</ul>
```

Tables:

We did and still do amazing, creative things with tables. While CSS is used for most styling and layout, sometimes it's faster and easier to use a table for your web page's layout. In addition, tables are just useful as tables. For instance, the University's web site uses tables throughout, e.g., for calendars, for layout, for menus, etc.

There are a few things to remember about tables. Tables are rectangles. They are never oddly shaped entities with bumps sticking out. They're rectangles. As such, **each row in a table must have the same number of columns (data cells)**. (we will discuss how to combine rows and cells inside a table later) The total number of columns for each row in a table must be the same.

Second, we specify the table row (with the tr tag), and then we specify each data cell in the row.

And third, the only place text (or lists, or images, or links, or even other tables) can go is inside of a data cell.

To make a table:

- Put <table> </table> around entire table
- For each **row** we want, we put <tr> </tr> around the entire row.
- For each **column within the row** (or data cell) within the row, we put <td> </td> around each data

Example (Try):

```
<table >
   <tr>
      <td> row1, col1 </td>
      <td> row1, col2 </td>
      <td>row1, col3</td>
   </tr>
```

```
     <tr>
       <td>row2,col1</td>
       <td>row2,col2</td>
       <td>row2,col3</td>
     </tr>
   </table>
```

Notice that this table has 2 rows (surrounded by the tr tags). Inside each row are 3 data cells (surrounded by the td tags). It is inside those td tags that all the content (text, etc.) goes.

NOTE: Tables are rectangles with columns and rows, but right now you can't clearly see the borders of the columns and rows. In order to more clearly see how a table works, you might want to add borders to the cells within the table. Unfortunately that is a style (CSS) element, not an HTML element. Style (CSS) comes later. However, if you want to see the table and its cell's borders now, you may add to the top of your web page (in between the <head> tags) the following code (the code you're adding is in purple):

```
<!doctype html>
<html lang="en-US">
   <head>
      <meta charset="utf-8">
      <title>My First Web Page</title>
        <style>
            td {
            border-style:solid; border-width: 2px; border-color:black;
            }
            table {
            border-style:solid; border-width: 2px; border-color:black;
            }
        </style>
   </head>
   <body>
```

Adding headings in the table:

For each column, you might want a heading. You can do this using the **th** tag inside your row. Note that, again, the row must contain the same number of headings as the number of columns in each of the other rows.

```
   <table>
     <tr>
       <th>head for col1</th>
       <th>head for col2</th>
       <th>head for col3</th>
     </tr>
     <tr>
       <td> row1, col1 </td>
       <td> row1, col2 </td>
       <td> row1, col3 </td>
     </tr>
     <tr>
       <td> row2, col1 </td>
       <td> row2, col2 </td>
       <td> row2, col3 </td>
     </tr>
   </table>
```

Combining two columns (data cells):

While every table row MUST have the same number of columns, you can visually combine two columns combine two columns (data cells) together. When you do this, it still counts as 2 data cells in a row even though it appears as one larger data cell. To do this, inside the td tag, add colspan = “2” (or 3, or however many columns you want to combine). When you do this, then that’s how many columns the data cell will take up.

```
<table >
   <tr>
      <td colspan = “2” > row1, col1  and 2 </td>
      <td> row1, col3 </td>
   </tr>
   <tr>
      <td> row2, col1 </td>
      <td> row2, col2 </td>
      <td> row2, col3 </td>
   </tr>
</table>
```

Combining two rows:

You can combine two rows together as well. When you do this, it counts as a data cell in both the first and the second row that are combined. Again, INSIDE the td tag, add rowspan = “2” (or 3, or however many rows you want to combine). When you do this, then that’s how many rows the data cell will take up (and thus each of those rows needs to have one less data cell).

```
<table >
   <tr>
      <td colspan = “2” > row1, col1  and 2 </td>
      <tdrowspan = “2”> row 1 and 2, col3 </td>
   </tr>
   <tr>
      <td> row2, col1 </td>
      <td> row2, col2 </td>
   </tr>
</table>
```

Putting a table into a cell in a table::

The easiest way to put a table inside of a table is to create both tables separately first. Then when you are sure you’ve formatted both tables properly, take the first table and paste it inside of a data cell
(<td> *new table goes here*</td>)
in the other table.

So create a new table:

```
<table>
   <tr>
      <td> table2 c1</td>
```

```
                   <td>table2 c2</td>
               </tr>
               <tr>
                    <td> t2c3</td>
                    <td>t2c4</td>
              </tr>
          </table>
```

Then copy and paste this new table into a data cell in your old table:

```
<table>
    <tr>
        <td colspan = “2” > row1, col1  and 2 </td>
        <tdrowspan = “2”> row 1 and 2, col3 </td>
    </tr>
    <tr>
        <td>
            <table >
               <tr>
                   <td> table2 c1</td>
                   <td>table2 c2</td>
               </tr>
               <tr>
                    <td> t2c3</td>
                    <td>t2c4</td>
              </tr>
          </table>
       </td>
        <td> row2, col2 </td>
    </tr>
</table>
```

That’s pretty much it for tables, although you’d be amazed at what all you can do with tables when you start getting creative.

Horizontal line across page:

This tag is one we used to use more frequently than we use now , but it is still a nice way to separate sections on a page by including a horizontal line. To add a horizontal line, include the following tag (**Note** that it’s only an opening tag with no closing tag. It’s a horizontal line, so the tag doesn’t need to go around anything else. As a result, there’s only the one tag.):

```
<hr>
```

Strong and br tags

Another example of a tag that must go inside other tags is the **<strong> tag**, which is for text you want emphasized within a paragraph. The importance of the strong tag is that browsers and search engines consider what is between strong tags to have stronger importance.

To include this tag, you’d do the following:

```
<p><strong>This is even more emphasized</strong></p>
```

To include a line break (or new line), you can use the **<br> tag**. This is another tag that only has an opening, with no closing. To see what this tag does, try including the following on your web page:

```
        <p>         I do not like them here nor there.
                    I do not like them anywhere.
                    I do not like green eggs and ham.
                    I do not like them Sam I am.
        </p>
```

Include this in your web page, then save it and view it in the browser. Does this look like you'd hope?

Now modify the code by adding line breaks:

```
        <p>         I do not like them here nor there. <br>
                    I do not like them anywhere. <br>
                    I do not like green eggs and ham. <br>
                    I do not like them Sam I am. <br>
        </p>
```

Save it and view it in the browser. See the difference?

pre tags

Another way to format a poem, or possibly code, or anything in which the line returns make a difference, is to use <pre> tags. Anything between the <pre> tags is displayed with a fixed width font and the spaces and line breaks retained. So, for instance, if you had:

```
<p> I
                    Am
                            Falling


                                    Downwards</p>
```

It would be displayed in the browser as:

```
I Am Falling Downwards.
```

Whereas if you had,

```
<pre> I
                    Am
                            Falling


                                    Downwards</pre>
```

It would be displayed as:

```
I
        Am
```

```
Falling



       Downwards
```

End of Tutorial 1