

# HTML Tags (Tutorial, Part 1):

---

## Starting Out:

Open your template file and save it under another name (possibly firstwebpage.html – note that there are no spaces in the name of your file. This is on purpose. Spaces in names make life difficult (and later, in javascript, aren't allowed). So get into the habit of naming things with names that have no spaces in them). This will be the file you edit (in either notepad++ or textwrangler). Wherever you saved the file on your computer, double-click on it. It should come up in your browser as a completely blank page. We're going to change that by adding text, images, links, and html tags.

## Template Tags:

A few rules about html: You have a template with the tags that are absolutely necessary for a minimal web page (a blank web page). These tags **MUST** occur only once inside a web page (not a web site, which can consist of many web pages, but each web page may have these tags only once, and they must occur in the order they occur in your template.

The following tags can **occur only once** in your web page:

- `<!doctype html>`
- `<html> </html>`
- `<head></head>`
- `<title></title>`
- `<meta charset = "utf-8">`
- `<body></body>`

## HTML Body Tags:

The following tags are tags that occur only between the two body tags. These tags may occur as many times as desired, as long as they're between the two body tags. For example, if you wanted a web page with a header, two subheaders, and 3 paragraphs, you'd have a page that looks like this:

```
<!doctype html>
<html>
  <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 these tags (and almost 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 goes around any paragraph you want 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). In other words, the `ol` and `ul` indicate a list. The only thing in a list can be 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.

Try adding both lists to your web page by placing both types of list inside the body of your web page. Save the file and bring it up in your browser. See what the lists look like?

## Tables:

We used to use tables to lay out our web pages. We did amazing things with tables, and got incredibly creative. Now CSS allows us to do a lot more in terms of laying out a web page, but sometimes a table is just easier. In addition, tables are just useful on their own. 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 can join data cells together, but then we must specify that a data cell is 2 columns in width, and, again, 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.

So a simple table would be formatted as follows:

Put `<table></table>` around entire table

For each row we want, we put `<tr></tr>` around the entire row.

For each column (or data cell) within the row, we put `<td></td>` around each data

### Example (Try):

```
<table border = "1">
  <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 (with a border width of 1), 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.

Add this table (*by typing it in, not copying and pasting*) to your web page, save it and view it in the browser.

#### 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 border = "1">
  <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>
```

Try modifying your previous table by adding table headers. Save the web page and view it in the browser.

#### Adding a caption to your table:

Captions are usually short text paragraphs that describe what the table is. The caption is placed outside of the table, so it doesn't need to have the same number of cells as other rows within the table. Different browsers place the caption in different places (some above, some below). To include a caption for a table, you'll include the <caption></caption> tag

```
<table border = "1">
  <caption>This is the caption that tells about the table </caption>
  <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>
```

Modify your table to include a caption, then save the web page and view it in the browser.

### Combining two columns:

You can combine two columns (data cells) together. When you do this, it counts as 2 columns in a row. 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 border = "1">
  <caption>This is the caption that tells about the table </caption>
  <tr>
    <th>head for col1</th>
    <th>head for col2</th>
    <th>head for col3</th>
  </tr>
  <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>
```

Modify your table to include the colspan, then save the web page and view it in the browser.

### 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 border = "1">
  <caption>This is the caption that tells about the table </caption>
  <tr>
    <th>head for col1</th>
    <th>head for col2</th>
    <th>head for col3</th>
  </tr>
  <tr>
    <td colspan = "2" > row1, col1 and 2 </td>
    <td rowspan = "2"> row 1 and 2, col3 </td>
  </tr>
  <tr>
    <td> row2, col1 </td>
    <td> row2, col2 </td>
  </tr>
</table>
```

Modify your table to include the rowspan, then save the web page and view it in the browser.

### Controlling width:

You can control the width of the table by specifying the width, either in the number of pixels wide the table should be, or as a percentage. Inside the table tag, include the width option as follows:

```
<table width = "500" border = "1">
  <caption>This is the caption that tells about the table </caption>
  <tr>
    <th>head for col1</th>
    <th>head for col2</th>
    <th>head for col3</th>
  </tr>
  <tr>
    <td colspan = "2" > row1, col1 and 2 </td>
    <td rowspan = "2"> row 1 and 2, col3 </td>
  </tr>
  <tr>
    <td> row2, col1 </td>
    <td> row2, col2 </td>
  </tr>
</table>
```

Or, alternatively, use a percentage to specify the width:

```
<table width = "50%" border = "1">
  <caption>This is the caption that tells about the table </caption>
  <tr>
    <th>head for col1</th>
    <th>head for col2</th>
    <th>head for col3</th>
  </tr>
  <tr>
    <td colspan = "2" > row1, col1 and 2 </td>
    <td rowspan = "2"> row 1 and 2, col3 </td>
  </tr>
  <tr>
    <td> row2, col1 </td>
    <td> row2, col2 </td>
  </tr>
</table>
```

Modify your table to include the width in pixels, then save the web page and view it in the browser. Resize your browser and watch what happens. Then modify it so that the width is specified in %. Save the web page and view it in your browser. Resize the page. See how this is different?

### 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 one and paste it inside of a data cell (<td> new table goes here </td>) in the other one.

### So create a new table:

```
<table border = "2">
  <tr>
```

```

        <td> table2 c1</td>
        <td>table2 c2</td>
    </tr>
    <tr>
        <td> t2c3</td>
        <td>t2c4</td>
    </tr>
</table>

```

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

```

<table width = "50%" border = "1">
  <caption>This is the caption that tells about the table </caption>
  <tr>
    <th>head for col1</th>
    <th>head for col2</th>
    <th>head for col3</th>
  </tr>
  <tr>
    <td colspan = "2" > row1, col1 and 2 </td>
    <td rowspan = "2"> row 1 and 2, col3 </td>
  </tr>
  <tr>
    <td>
      <table border = "2">
        <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>

```

Modify your table to include a table within a table, then save the web page and view it in the browser.

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.

## Preformatted Text:

Sometimes you want things to look like you want them to look. You don't want the browser to be messing with it.

For instance, I might

```

    want my
        text to
            slowly
                slowly
                    slide

```

across the  
page.

The easiest way to do this is with the preformat tag. It tells the browser to leave stuff where you put it. So, for instance, I could do the following:

```
<pre>
    I might
      want my
        text to
          slowly
            slowly
              slide
                across the
                  page.
</pre>
```

Add the above (or something similar) to the bottom of your web page., then save and view in the browser.

### 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>
```

Add that tag to your web site (maybe between two paragraphs, below a header, or below a table), then save and view in your browser. See the horizontal line?

### Tags that need to go inside other tags:

Certain tags must go inside of other tags. The tags they most frequently go inside are:

- the p tags
- the h1 through h6 tags
- the li tags, or
- the td tags.

An example of this type of tag is the <em> tag, which is for text you want emphasized within a paragraph. To include this tag, you'd do the following:

```
<p>This is <em> emphasized </em> </p>
```

Another example is the strong tag, for text you want strongly emphasized:

```
<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?

## Images:

Images belong among the tags that must go inside the other tags (specified above). To include an image, you must specify not only that you're including an image, but also the name and location of the image, the width and height of the image, an id for the image, and alternative text that describes the image for people who are unable to download or see the image. So an image tag will look as follows:

To include an image, follow these steps (**Important:** use only jpeg, gif, or png files):

**Note:** Like `<hr>` and `<br>`, the `<img >` tag does not have a separate opening and closing tag

**Step1:** Find an image and place it in the same folder as the web page you are working on. For now, the image MUST be in the same folder as your web page, or this won't work. Make sure the image is either a .jpg, a .gif, or a .png file. This ensures it will show up in all browsers.

(For this example I shall be using the picture "kittens.jpg". You will be using whatever image you've chosen.)

**Step 2:** in the folder, hover your mouse over the image. You should see something like this:



Notice the Dimesions. The width is 422 pixels, and the height is 317 pixels.

(On the mac, click on the image, then under the file menu, choose get info. You should see the dimensions in the menu that pops up.)

**Step 3:** Now open your web page for editing.

**step 4:** Add the image (note that it's inside of a p tag here):

```
<p><img src = "kittens.jpg" width = "422" height = "317" alt = "a picture of itty bitty baby kittens" id =  
"kittenpic"> Way cute!!! </p>
```

- The **src** is the name of the picture we want to show up.
- The **width** and the **height** are the width, and height in pixels, respectively.
- The **alt** is alternative text describing the picture for people who either can't download or can't see the image (critical for individuals who are blind); and
- The **id** is a unique id for this particular picture. Every id on a web page must be different.

Save the web page and view it in the browser. Your web page should now have a picture in it.

### To create a folder for images:

**Step 5:** Navigate to the folder containing your saved web page.

**Step 6:** Inside that folder, create a new folder called "Images".

**Step 7:** Move your image (kittens.jpg) into the Images folder.

**Step 8:** Reload your web page. You should no longer see your image. The browser can't find it because it is looking for the image in the same directory where your web page is, but now the image is located in the Images folder.

**Step 9:** Edit your web page. Change the image tag to:

```
<p><img src = "Images/kittens.jpg" width = "422" height = "317" alt = "a picture of itty bitty baby kittens" id = "kittenpic"> Way cute!!! </p>
```

The change: **src** now is set to "images/kittens.jpg". That is telling the browser to look inside the images folder for the picture kittens.jpg.

Save the web page and view it in the browser. Your web page should have a picture in it again.

### Links to other web pages:

Note: links must go inside one of the other tags mentioned above.

There are two types of links: **Absolute and Relative**.

#### Absolute Links:

Absolute links must include the transfer protocol (usually http), the domain name, and the name of the web page (html file). In other words, an absolute link is a link to a web page on another server.

An example of a link using an absolute path:

```
<p> This is a link: <a href = "http://www.eecis.udel.edu/~yarringt/103/index.html" id = "103link"> link to 103 Web page </a> </p>
```

In this example

- **href** is what we're linking to:
  - **http** is the transfer protocol
  - **www.eecis.udel.edu** is the domain name
  - **~yarringt/103** are the folders in which the web page is located, and
  - **index.html** is the web page
- the **id** is a unique id for the link
- the text between **<a... id="103link">** and **</a>** is the text you click on to go to the web page, and
- **</a>** ends the text that can be clicked on.

Add the absolute link to your web page. Save the page and view it in your browser. Click on “link to 103 Web page”. Did it take you to the class’s web site? Now modify the link (and the text that takes you to that link) so that it is a link to amazon (or espn, or cnn – whatever you like). Save your web page and make sure it works.

## Relative links

You use relative links for pages within your web site (so they are on the same server as the web page you’re editing).

With a relative link, the browser starts looking in the folder where your web page is saved, and moves around from there. To move up a folder, you’d use ../ To move down into another folder, you specify the name of that folder.

Let’s start by creating a relative link to another web page that is located in the same folder as the web page you’re editing. You’ll first need to create another web page from your template file.

**step1.** Using the template.html file, create a new web page. Add a paragraph, and possibly a header. Save the new web page as *samedirectory.html*. Save it in the same folder as the web page you’ve been working on so far.

**step2.** In the web page you’ve been working on, add the following code :

```
<p> This is a relative link: <a href = “samedirectory.html” id = “samedir”> link to web page in the same directory </a></p>
```

Save your web page and open it in a browser. Click on “link to same directory page”. The browser should open the *samedirectory.html* web page you created. This is because we didn’t specify another folder, or a transfer protocol and a domain, so the browser looked for *samedirectory.html* in the same folder where your web page is located. Unless told to look elsewhere, the browser will always start looking for the linked web page in the same folder where the web page you’re working on is saved.

### ***To create a relative link to a file in another folder:***

**step3.** Create a new folder in the same place where you’ve saved your web page (and now, *samedirectory.html*). Name this new folder *OtherWebPages*

**step4.** Use the template to create another new web page. Give your new web page a paragraph and save your new web page in the folder *OtherWebPages*. Call your new web page *tba.html*.

**step5.** In the web page you’ve been working on add the following link:

```
<p> This is a link to a web page in a folder (remember, we always start looking in the folder this web page is in and move from there): <a href = “OtherWebPages/tba.html” id = “tbalink”> link to tba </a></p>
```

Notice that in this case, the href specifies the folder first, and then the file name. Now the browser starts looking in the same folder where the web page you’re working on is saved, looks for a folder called *OtherWebPages*, and then, inside that folder, it looks for the file *tba.html*.

Save and view in browser. Click on “link to tba”. Your browser should bring up the *tba.html* page.

Now you’re going to create a link from the web page “*tba.html*” back to the web page you’ve been working on. To do that, you must tell the browser to look in the folder above the one you’re currently in. You’ll use the ../ to specify this.

**step 6.** Navigate into the *OtherWebPages* folder and open *tba.html*.

**step 7.** create a link to the page you'd been working on. Add the following to tba.html:

```
<p> This is a link back to your main page: <a href = "../firstwebpage.html" id = "yourpagelink"> link to main page</a></p>
```

Save tba.html. Open it in the browser. Click on "link to main page". It should take you back to the page you'd been working on.

**step8.** Inside OtherWebPages create a new folder called DeeperPages.

**step9.** Use the template to create yet another new web page. Give your new web page a paragraph and save your new web page in the folder DeeperPages. Call your new web page *wba.html*.

**step10.** Back in the web page you've been working on add the following link:

```
<p> This is a link to a web page in a folder in a folder (remember, we always start looking in the folder this web page is in and move from there): <a href = "OtherWebPages/DeeperPages/wba.html" id = "wbalink"> link to wba </a></p>
```

Save and view in browser. Click on "link to wba". Your browser should bring up the wba.html page.

**step 11.** Navigate into the OtherWebPages folder and then into the DeeperPages folder and open wba.html.

**step 12.** create a link to the page you'd been working on. Add the following to wba.html:

```
<p> This is a link back to your main page: <a href = "../../firstwebpage.html" id = "yourpagelink"> link to main page</a></p>
```

Save wba.html. Open it in the browser. Click on "link to main page". It should take you back to the page you'd been working on.

You've now created an absolute link on your web page, as well as a number of relative links to web pages in different folders. In essence, you've created a web site.

### Turning an image into a link:

You may want to make a picture be something you can click on to take you to another web page. You've seen this on numerous pages – buttons you click on, logos you click on, etc. To turn an image into a link is fairly simple. Create the link:

```
<p><a href = "http://www.lolcats.com" id = "lolcats"> what you click on goes here </a></p>
```

Then add the image:

```
<img src = "Images/kittens.jpg" width = "422" height = "317" alt = "a picture of itty bitty baby kittens" id = "kittenpic">
```

**Now copy the image into the area that says, "what you click on goes here" in the link:**

```
<p><a href = "http://www.lolcats.com" id = "lolcats"> <img src = "Images/kittens.jpg" width = "422" height = "317" alt = "a picture of itty bitty baby kittens" id = "kittenpic"> </a></p>
```

Now when you save the file and bring it up in the browser, you can click on the kittens picture and you should go to the lolcats web site.



**Step 1.** First add a marker at the bottom of the firstwebpage.html. So your code might now look like:

```
...  
<br >  
<a href = "#topofpage"> Back to top  
</a>  
</p>  
<p><a id = "bottom"></a></p>
```

**Step 2:** Open the file samedirectory.html. Within that web page, add the following link:

```
<p><a href = "firstwebpage.html#bottom">link to bottom of firstwebpage </a>  
</p>
```

**Step 3.** Save the file and click on the link. Does it take you to the bottom of your first web page?

## End of Part 1

**For more information on html tags:**

<http://www.w3schools.com/tags/default.asp>