More HTML
Images and Links

Links

- An **anchor** tag (a) is used to define a link, but you also need to add something to the anchor tag – the **destination** of the link.

- <a href="http://www.htmldog.com">HTML Dog</a>

- The destination of the link is defined in the href attribute of the tag.
- The link can be absolute, such as “http://www.htmldog.com”, or it can be relative to the current page.
- So if, for example, you had another file called “flyingmoss.html” in the same directory then the line of code would simply be <a href="flyingmoss.html">The miracle of moss in flight</a> or something like this.

- A link does not have to link to another HTML file, it can link to any file anywhere on the web.
- A link can also send a user to another part of the same page they are on. You can add an id attribute to just about any tag, for example <h2 id="moss">Moss</h2>, and then link to it by using something like this:<a href="#moss">Go to moss</a>. Selecting this link will scroll the page straight to the element with that ID.

Images

- The img tag is used to put an image in an HTML document and it looks like this:
- <img src="http://www.htmldog.com/badge1.gif" width="120" height="90" alt="HTML Dog">
- The src attribute tells the browser where to find the image.

- The width and height attributes are necessary because if they are excluded, the browser will tend to calculate the size as the image loads, instead of when the page loads, which means that the layout of the document may jump around while the page is loading.
- The alt attribute is the **alternative description**. This is an accessibility consideration, providing meaningful information for users who are unable to see the image (if they are visually impaired, for example).
- Note that, like the br tag, because the img element does not enclose any content, no closing tag is required.

- The most commonly used file formats used for images are **JPEGs**, **GIFs**, and **PNGs**. They are compressed formats, and have very different uses.
- **JPEGs are typically used for images such as photographs.**
- **GIFs are typically used for images with solid colors, such as icons or logos.**
- **PNGs are typically used for versatile images in more complex designs BUT they are not fully supported by some older browsers.**

Tables

- There are a number of tags used in tables.

```
<table>
    <tr>
        <td>Row 1, cell 1</td>
        <td>Row 1, cell 2</td>
        <td>Row 1, cell 3</td>
    </tr>
    <tr>
        <td>Row 2, cell 1</td>
        <td>Row 2, cell 2</td>
        <td>Row 2, cell 3</td>
    </tr>
    <tr>
        <td>Row 3, cell 1</td>
        <td>Row 3, cell 2</td>
        <td>Row 3, cell 3</td>
    </tr>
    <tr>
        <td>Row 4, cell 1</td>
        <td>Row 4, cell 2</td>
        <td>Row 4, cell 3</td>
     </tr>
</table>
```

- The table element defines the table.
- The tr element defines a table **row**.
- The td element defines a **data cell**. These must be enclosed in tr tags, as shown above.
- If you imagine a 3x4 table, which is 12 cells, there should be four tr elements to define the rows and three td elements within each of the rows, making a total of 12 td elements.

Forms

- **Forms** are used to collect data inputted by a user. They can be used as an interface for a web application, for example, or to send data across the web.
- On their own, forms aren’t usually especially helpful. They tend to be used in conjunction with a programming language to process the information inputted by the user.

- The basic tags used in the actual HTML of forms are form, input, textarea, select, submit and option.

form

- form defines the form and within this tag, if you are using a form for a user to submit information (which we are assuming at this level),
  an action attribute is needed to tell the form where its contents will be sent to.
- The method attribute tells the form how the data in it is going to be sent and it can have the value get, which is default, and latches the form information onto a web address, or post, which (essentially) invisibly sends the form's information.

```
<form action="processingscript.php" method="post">


</form>
```

input

The input tag is the daddy of the form world. It can take a multitude of guises, the most common of which are outlined below:

- <input **type="text“ name=“id”**> or simply <input> is a standard textbox. This can also have a value attribute, which sets the initial text in the textbox.
- <input **type="password“ name=“id”**> is similar to the textbox, but the characters typed in by the user will be hidden.
- <input **type="checkbox“ name=“id”**> is a checkbox, which can be toggled on and off by the user. This can also have a checked attribute (<input type="checkbox" **checked**> - the attribute doesn’t require a value), and makes the initial state of the check box to be switched on, as it were.
- <input **type="radio“ name=“id**> is similar to a checkbox, but the user can only select one radio button in a group. This can also have a checked attribute.
- <input **type="submit"**> is a button that when selected will submit the form. You can control the text that appears on the submit button with the value attribute, for example **<input type="submit" value="Ooo. Look. Text on a button. Wow">**.

textarea

- textarea is, basically, a large, multi-line textbox. The anticipated number of rows and columns can be defined
  with rows and cols attributes

```
<textarea name=“id” rows="5" cols="20">
      A big load of text
</textarea>
```

select

- The select tag works with the option tag to make drop-down select boxes.

```
<select name=“id”>
        <option>Option 1</option>
        <option value=“second option”>Option 2</option>
        <option value="third option">Option 3</option>
</select>
```

- When the form is submitted, the value of the selected option will be sent. This value will be the text between the selected opening and closing option tag unless an explicit value is specified with the value attribute, in which case this will be sent instead. So, in the above example, if the first item is selected, “Option 1” will be sent, if the third item is selected, “third option” will be sent.

- Similar to the checked attribute of checkboxes and radio buttons, an option tag can also have a selected attribute, to start off with one of the items already being selected,

eg.

<option selected>Rodent</option>

would pre-select “Rodent” from the items.

Names

- All of the tags mentioned above will look very nice presented on the page but if you hook up your form to a form-handling script, they will all be ignored. This is because the form fields need **names**. So to all of the fields, the attribute name needs to be added, for example
- <input type="text" name="talkingsponge">.

Sample Form (formatted)

```
<form action="contactus.php" method="post">
        <p>Name:</p>
        <p><input type="text" name="name" value="Your name"></p>
        <p>Comments: </p>
        <p>
                <textarea name="comments" rows="5" cols="20">
                        Your comments
                </textarea>
        </p>
         <p>Are you:</p>
        <p>
                <input type="radio" name="areyou" value="male"> Male
        </p>
        <p>
                <input type="radio" name="areyou" value="female"> Female
        </p>
        <p>
                <input type="radio" name="areyou" value="hermaphrodite"> An hermaphrodite
        </p>
        <p>
                <input type="radio" name="areyou" value="asexual"> Asexual
        </p>
        <p><input type="submit"></p>
</form>
```

Exercises

By now you have learned how to use a large number of different HTML tags. Each, of course, serves a different purpose in organizing the structure of content on a web page. Use these tags to further modify your class website by doing the following:

1. Using the **images** folder in your website's root folder, find and add images that represent each of your first four units in the course.
2. Add these images to each unit page (something that represents the content of that unit).
3. Go to http://avachara.com/avatar/ or some other free site and create your avatar. Save it to your images folder and place it on your homepage near your paragraph where you discussed yourself.
4. Create a list of links from your home page to each of your unit pages.
5. Find a pdf or doc online that provides info on links and images in HTML. Add it to your class website **Resources folder and then link to it via your Unit 1 page under a heading titled "Unit Resources".**
6. On your home page include a blurb about yourself (if you haven't already) and create a table of your first and second semester course schedule. Describe the reasons for taking this course (again if you haven't already). *Also include a list of links to the software you will be using in the class as well as to at least 2 site references on using HTML (again place this under a heading titled "Class Software".*
7. Finally create a new page and a link to it from your homepage (use "**User Registration**" as the link). In it add a form that provides a visitor to your site with a form that allows them to register as a user of your site. Make sure to include boxes for the visitor's first and last names, their gender (use a set of **radioButtons**), age and the school they attend. Include a submit button. Also include a **textarea** with a label prompting the user to write a little blurb about themselves. Use the following names to identify the form components:

firstName, lastName, age, gender, school, aboutMe

Use POST as the method. Download and use the **echo.php** script to test your form (it should "echo" the form entries). Create a folder called **scripts (place this folder in your websites root folder)** and place this file in it. You will only be able to test your script from a webserver so you'll have to upload everything to your remote site.