

# AOS 452 - Lab 7 Handout

## HTML, CSS and Your Webpage

### INTRODUCTION

The influence of the Internet on today's society would be very difficult to overestimate. From its more secretive beginnings in the United States Department of Defense in the late 1960s, the Internet has grown into an entity that has revolutionized the way the world operates. The Internet took off in mainstream society in the mid 1990s with the advent of the World Wide Web (invented in 1989) and suitable Internet browsers (such as Netscape, first available in 1994) to view text and images posted online. Internet savvy is much desired in today's job market. One computer-related skill that especially attracts employers is web page development.

Web pages are generated from plain-text files written in HyperText Markup Language, or HTML. There are many ways to create web pages and learn HTML. A multitude of "point and click" HTML editors are available that will allow even the most novice of computer users to create a web page. However, hackers agree that the worst way to learn HTML is to use one of these "user friendly" editors, which are prone to producing bloated, buggy, and impossible to understand HTML. Instead, you will gain a more comprehensive understanding of how web pages are constructed by getting your hands dirty with the raw HTML code.

**For your third WxDiscussion you will be *required* to incorporate your webpage into the discussion** (primarily via hosting GEMPAK images you will create for use during the weather discussion). You will find that your discussion will be much more organized if you make effective use of your web page.

The purpose of today's lab is to introduce you to the basics of HTML. At the end of this lab, you will be directed to online resources with which you can further your understanding of HTML. For those of you with experience using HTML, this lab should serve as a good refresher<sup>1</sup>. For those of you first being exposed to HTML today, get ready to learn a skill that can be enjoyable to apply.

HTML has steadily evolved through the years, and today it is widely recognized that the best way to organize web pages is to use two files per page. The first file is in HTML (or XHTML to be exact), and it contains that page's content. The second file is in CSS (Cascading Style Sheets), and it informs the browser how your content should be displayed. This distinction will become clearer as we go through the lab. Note also that the same CSS file can be applied to many different HTML files.

### GETTING STARTED

Space for your web pages has already been set aside on an AOS server. The first thing I would like you to do is start Firefox and type in the following URL<sup>2</sup> (Universal Resource Locator, i.e., "web address"):

`http://www.aos.wisc.edu/~your_username`

For instance, my web page is found here: <http://www.aos.wisc.edu/~mbreeden>

---

<sup>1</sup> Note that this lab presents cutting edge HTML (i.e., XHTML+CSS), not your grandfather's HTML from 1999.

<sup>2</sup> Some folks use the acronym URI, for Universal Resource Identifier.

(For the purposes of this lab, you will want to use this as a guide. Just go my 452 webpage above, right-click to see a menu, then select view page source to get some hints. Do not copy my code and use it as your own. I will be able to tell - I did write it.)

If you have not worked on your AOS-provided web page, many of you should see some text describing how to send mail to you. Other may see a message about the location not being found. Soon your own web page will be found at this location.

Check to see if you have a subdirectory called **public\_html** in your home directory. If no such directory exists, then type the following at the prompt:

```
mkdir public_html
```

Now move to this new directory by entering the following command into the terminal:

```
cd public_html
```

The HTML code that controls the content you see (or will see) on your homepage will be found in a file called *index.html*. If your **public\_html** directory already existed, you may find the *index.html* file in that directory. If you just made the **public\_html** directory, then you will need to create the *index.html* file. You will be building your web page using raw HTML and CSS code written in files created in a text editor. On the Room 1411 workstations, you can use the text editor **gedit** to create and edit the HTML and CSS files. (The text editors **vi** and **emacs** are also available).

Open **gedit**. Once you have a text box type in the following:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <title>Enter the title of your page here</title>
    <link rel="stylesheet" type="text/css" href="web.css" />
  </head>
  <body>
  </body>
</html>
```

The text above is a template containing the HTML code that is needed for every web page you create<sup>3</sup>. Save this file as *index.html*. If you are asked to overwrite a pre-existing file, just click OK. (If you already have a web page created on your AOS-provided site, then ignore the above instructions and save the file to some other .html location!)

---

<sup>3</sup> Strictly speaking, lines 1 and 2 are not necessary; however, they will be needed to validate your HTML, described later. Line 6 is only necessary if you wish to apply CSS (web.css in this example) to your HTML.

## THE BASIC STRUCTURE OF AN HTML FILE

In the template you created, you see text like `<html>` and `<head>`. These pieces of text are referred to as *tags*. Two types of tags exist, container tags and empty tags. Container tags have a starting tag and a closing tag, and tell the web browser something about all of the text in between them. To make a closing tag, just add a forward slash (/) to the starting tag. For example `<html>` is a container tag (closed by `</html>`) that tells the browser that everything contained within that tag is, in fact, HTML code. Empty tags, the second type, are distinguished by the fact that they do not have closing tags. They tell the browser to do something at one spot only. Empty tags have the format `<tag />`.

Later in the lab, you will learn about attributes. Attributes go inside starting and empty tags and tell the browser additional information about those tags.

According to the latest standard, HTML tags must be lowercase.

### HTML tag

The `<html>` tags tell the browser that the text between the tags is going to be part of an HTML document.

### HEAD and TITLE tags

Everything located between the `<head>` and `</head>` tags will not actually show up on your web page. The main purpose of these tags is to provide information about your web page, such as a title. You can define a title for your web page by entering text between the `<title>` and `</title>` tags, this title will show up in the title bar of your browser window.

### BODY tag

Everything that you want to appear in the browser display window must be contained within the `<body>` tags. In addition, you can specify a background color or image as a CSS property applied to the `<body>` tag.

*You can find information on how to specify a background color or image in a tutorial that I will refer you to at the end of this lab.*

## ENTERING AND FORMATTING TEXT

It is very easy to include text in your web page. All text that you want to appear in your web page must be located inside the `<body>` tags. One important point to remember is that by default, browsers do not recognize extra spaces or carriage returns in the HTML document. However, you can use the following HTML code to control space and carriage return formatting in your web page:

<code>&amp;nbsp;</code>	Means “space” to the browser
<code>&lt;br /&gt;</code>	Line break (essentially what ENTER does in a word processor)

<code>&lt;p&gt; &lt;/p&gt;</code>	The “paragraph” tag isolates the block of text. These tags should enclose much of your regular web text.
<code>&lt;pre&gt;&lt;/pre&gt;</code>	Text will appear in the browser as shown in the HTML document

CSS provides much greater control over text formatting, *see the tutorial on the webpage at the end of the lab.*

Edit *index.html* to look like the following:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

  <head>
    <title>This is my AOS 452 page!</title>
    <link rel="stylesheet" type="text/css" href="web.css" />
  </head>

  <body>
    <p>Welcome to my AOS 452 web page! I will be using a web page for my second map
    discussion. Hooray!</p>
  </body>
</html>
```

Save the changes to the HTML document. Reload the web page to see the changes. Experiment with the other space and carriage return formatting tags.

### Text styles

It is easy to add text styles. Text styles that you might use are listed below:

<code>&lt;strong&gt;&lt;/strong&gt;</code>	<b>Bold (usually)</b>
<code>&lt;em&gt;&lt;/em&gt;</code>	<i>Italic (usually)</i>
<code>&lt;big&gt;&lt;/big&gt;</code>	<b>Larger Font</b>

For example, you could have all of your text in your web page in bold letters:

```
<p><strong>
  Welcome to my AOS 452 web page! I will be using a web page for my second
  map discussion.
</strong></p>
```

Other tags, like `<u></u>` for underlining, are considered obsolete. CSS should be used instead. CSS also provides the means through which you can apply additional styles, change colors, use different fonts, etc. You can read more about CSS on the tutorial page given at the end of this lab.

## INSERTING IMAGES

Images can be added to your web page by using the empty tag `<img />`. In the `<img />` tag, you must specify the source of the image using the `src` attribute. For instance:

```

```

The browser will place the image where it would normally place the next piece of text. Be sure to enclose the image name in double quotes (`src` is an attribute).

Here are some other attributes that can be used with the `<img />` tag:

<code>height</code>	Controls the height of the image
<code>width</code>	Controls the width of the image
<code>alt</code>	Substitute for the image when the user is using a browser that is not displaying images. <b>This attribute is required by the HTML specification!</b>
<code>title</code>	In modern browsers, the value of <i>title</i> often appears next to the mouse arrow when the arrow is over the image.

Here is an example that uses these attributes:

```

```

The image *tornado.gif* will have a size of 300 x 300 pixels. If you move the mouse cursor over the image, "Big Tornado" may appear in a little box next to the cursor. If you've disabled images in your browser, "F5" will appear where the image would have been. The original image may have different dimensions than the values given above, so changing the *height* and *width* attribute could distort your image. The appearance of your image also depends on the monitor resolution.

NOTE: The *src* attribute specifies not only what image, but also the location of the image. The above source, "tornado.gif", will tell the browser to look for the image *tornado.gif* in the same directory as the HTML document itself. If the image is in another location (a different directory on your account, another web page), you will need to adjust the *src* attribute value accordingly (full pathname or web address). The online tutorial will provide more information on this issue.

ANOTHER NOTE: UNIX is case sensitive with regard to filenames. For example, take the following three filenames: *lab7.gif*, *LAB7.GIF*, *Lab7.gif*. UNIX treats these as three distinct filenames, whereas a Windows-based server would consider these the same filename. To avoid this case problem, I recommend using lower case letters for filenames.

STILL ANOTHER NOTE: The UNIX command `file` will tell you the pixel size of an image, e.g. `file tornado.gif <Enter>` when entered into a terminal.

## INSERTING LINKS

One of the best features of web pages is the ability to link to images and other pages within or outside your web page. Links are contained within the `<a>` tags. You can choose to add text links or graphical links to your web page.

Text link

The following is the general format for adding a text link:

```
<a href="URL">Text describing the link</a>
```

Let's say you wanted to insert a text link to the National Weather Service's website. You would enter something like the following:

```
<a href="http://www.nws.noaa.gov/">National Weather Service homepage</a>
```

Graphical link

The following is the general format for adding a graphical link:

```
<a href="http://www.nws.noaa.gov/"></a>
```

Let's say you have a GIF image of Bucky Badger in the same directory as the HTML document and you want to use the image as a link to the UW-Madison homepage. You would enter something like this:

```
<a href="http://www.wisc.edu/"></a>
```

**TABLES**

Tables can greatly enhance the appearance and organization of your web page. As with other basic features of web pages, tables are relatively easy to create. The creation of tables only involves (at minimum) three tags:

<code>&lt;table&gt;</code>	The main <code>&lt;table&gt;</code> tags.
<code>&lt;tr&gt;</code>	The Table Row tag defines a horizontal row of cells
<code>&lt;td&gt;</code>	The Table Data tag specifies an individual cell in a table row

A template of a table composed of two rows with two cells is given below:

```
<table>
  <tr>
    <td>Cell 1 of Row 1</td>
    <td>Cell 2 of Row 1</td>
  </tr>
  <tr>
    <td>Cell 1 of Row 2</td>
    <td>Cell 2 of Row 2</td>
  </tr>
</table>
```

**MISCELLANEOUS INFORMATION**

- Comments can be added to the HTML document by placing text between the "starting tag" `<!--` and the "ending tag" `-->`

**Example:**

```

<head>
  <title>AOS 452 homepage</title>
  <link rel="stylesheet" type="text/css" href="web.css" />
</head>

<body>
  <p>Making web pages is fun!</p>
  <!-- Hi Mom! I made my first web page! -->
</body>

```

*Making web pages is fun!* would appear on the screen, but *Hi Mom! I made my first web page!* would not.

- After you edit and save your HTML document, or any CSS being applied to your HTML document, you can view the changes by clicking the RELOAD (or REFRESH) button on your browser. If your page does not update, you can force the browser to update by holding down the SHIFT key and clicking on the RELOAD (or REFRESH) button.
- You can view the HTML code for a web page by using the “View Page Source” option on the browser. You can find this option under the “View” menu in most web browsers.
- You can check that your HTML is free of bugs by using the following validation service:  
<http://validator.w3.org>  
 Enter your URL in the “address” form, and if everything is OK, it will tell you that you have valid HTML code. If not, it will tell you what’s wrong. Note that the first error may lead to spurious errors later in your HTML. My advice is to fix the first error first, revalidate, and if necessary, repeat the process.
- Once you know your HTML is correct, you can use a similar process to validate your CSS at:  
<http://jigsaw.w3.org/css-validator/>

**HTML DOG**

One of the better HTML tutorials online is available from HTML Dog. Upon successful completion of the tutorial, you will have the basic skills necessary to create a quality web page. The tutorial is done with humor, which only enhances the tutorial’s quality.

You will find the tutorial at the following URL:

**<http://www.htmldog.com>**

If you click on the “HTML Beginner’s Guide” link, you will be taken to the first page of the basic HTML tutorial. All of the tutorials available to you can be found along the left side of the page.

Here you can get info on background colors and images, titles, CSS style sheets and more...

Work through these four tutorials in this order:

1) HTML Beginner, 2) CSS Beginner, 3) HTML Intermediate, 4) CSS Intermediate

You need not worry about the “Forms” or “Javascript” sections.