CAS 111D: XHTML Tutorial

HTML, XML, XHTML, Oh My!

You may, or may not, have heard the terms: HTML, XML, or XHTML. What are they and what is the difference between them? Listed below are some basic definitions. If you'd like more information about the differences between these terms, go to: http://www.webstandards.org/learn/articles/askw3c/oct2003/.

HTML: Stands for "Hypertext Markup Language". This is the original language used to create documents for viewing on the web. HTML was designed to display data. The latest version is HTML 4.01 This is the last version because HTML has been superseded by a more versatile language called XHTML.

XML: Stands for "Extensive Markup Language". XML was designed to describe data and to focus on what data is XML is a cross-platform, software and hardware independent tool for transmitting information. XML allows data to be exchanged and shared with many different types of playforms, software and hardware. XML makes data more useful. HTML is limited and can not do this. HTML is about displaying information only, while XML is about describing information. XML allows easy conversion of data regardless of where it is being used. HTML can not do this.

XHTML: Stands for "Extensible Hypertext Markup Language". This is a combination of HTML and XML. XHTML is a newer, stricter and cleaner version of HTML. XHTML was designed to replace HTML. It is now the standard language for creating web pages. XHTML is more versatile and can be used with all new browser versions. XHTML, unlike HTML, can display on different platforms such as PDAs and other portable devices. Many web pages written with HTML do not use the correct tags and often have "sloppy" code. XHTML does not allow sloppy code and therefore will result in information that is universally accessible.

The main difference between HTML and XHTML:

- All XHTML tags must be lowercase except the "DOCTYPE" tag because it is simply a comment and not a true XHTML tag
- All tags must be "closed" including empty elements such as
, <hr />.
- All tags must be properly "nested"
- All documents must be properly formatted
- All documents must have a "DOCTYPE"

Why do I need to know this?

You may be wondering why all of this matters since Dreamweaver does all the coding for you. A basic understanding of XHTML is important to fully understand what is going on as you design your web pages. Remember, web pages are simply text documents that include specific tags that tell a Web browser how to display the elements. Throughout this course, you will be examining the "CODE" behind what you see on the Dreamweaver Design page. Learning these basic tags will help you better understand what you are looking at and what purpose each tag has in the overall design of your site.

If you're serious about Web design, you should consider taking a class on HTML. CAS206 is a class devoted to XHTML and is a great follow-up to your Dreamweaver course.

Your first web page

It is now time for you to create a web page using XHTML. You will be using a text editor rather than Dreamweaver for this one exercise. The purpose is to give you some practice working with the codes. Remember, Dreamweaver does all the coding for you but it is still important to understand why the program does what it does.

You will be working through a beginner's XHTML tutorial taken from *HTML Dog created by Patrick Griffiths*. This Beginner Tutorial assumes that you have no previous knowledge of HTML or CSS. You can reference the tutorial online at **http://htmldog.com/guides/htmlbeginner/** but all of the instructions are in this handout.

You will need to be at a computer connected to the Internet. You will be using your browser and a text editor such as NOTEPAD. You can have your browser and your text editor open simultaneously which will allow you to switch back and forth easily.

Getting Started:

Most of the stuff on the web is no different than the stuff on your computer – it's just a whole load of files sorted into a whole load of directories. HTML files are nothing more than simple text files, so to start writing in HTML, you need nothing more than a simple text editor.

Notepad is a common text editor (on Windows this is usually found under the Programs > Accessories menu).

1. Type this in to your text editor:

   My name is _____ (type your name) and this is my first web page.

2. Save the file as "myfirstpage.html". Be sure to save the file in the "Additional Exercises" folder that you created on your USB drive. It is important that the extension ".html" is typed as part of the file name - some text editors, such as Notepad, will automatically save it as ".txt" otherwise.

3. To look at HTML files, they don't need to be on the web. Open your web browser, either **Firefox** or **Internet Explorer** and in the address bar, where you usually type web addresses, type in the location of the file you just saved (for example, "e:/cas111d/additional exercises/myfirstpage.html") and hit return. You can also go to the File menu of the browser, select Open, and browse for the file.

Pow. There it is. Your first web page. How exciting. And all it took was a few typed words.

Tags, Attributes, and Elements:

Although the basics of HTML are plain text, we need a bit more to make it a valid HTML document. The basic structure of an HTML document includes **tags**, which surround content and apply meaning to it.

1. Change your document so that it looks like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<body>
My name is _____ (type your name) and this is my first web page.
</body>
</html>
```

2. Now save the document again, go back to the web browser and select "refresh" (which will reload the page).

The appearance of the page will not have changed at all, but the purpose of HTML is to apply structure or meaning, not presentation, and this example has now defined some fundamental elements of a web page.

The first line on the top that starts <!DOCTYPE... is to let the browser know that you know what you're doing. You may think that you don't actually know what you're doing yet, but it's important to stick this in. If you don't, browsers will switch into "quirks mode" and act in a very peculiar way. It may seem ridiculously long to have to type this in at the start of every web page. But guess what? Dreamweaver does this for us – so once we start creating web pages using Dreamweaver, we won't have to type this in ever again!

Tags

All XHTML tags have an opening tag and a closing tag which are indicated with brackets <>. The first tag we see is the **<html> tag** which kicks things off and tells the browser that everything between that and the **</html> closing tag** is an HTML document. The stuff between **<body>** and **</body>** is the main content of the document that will appear in the browser window.

Not all tags have closing tags like this (<html></html>). Some tags, which do not wrap around content will close themselves. The horizontal rule tag for example, looks like this : <hr />. We will come across more of these examples later. All you need to remember is that all tags must be closed and most (those with content between them) are in the format of opening tag → content → closing tag.

Attributes

Tags can also have **attributes**, which are extra bits of information. Attributes appear inside the opening tag and their value is always inside quotation marks. They look something like <tag attribute="value">Margarine</tag>. We will come across tags with attributes later.

Elements

Tags tend not to do much more than mark the beginning and end of an **element**. Elements are the bits that make up web pages. You would say, for example, that everything that is in-between and includes the <body>

and </body> tags is the body element. As another example, whereas '<title>' and '</title>' are *tags*, '<title>Rumple Stiltskin</title>' is a title *element*.

Page Titles

All XHTML pages should have a page title. This title does not appear on the web page itself when viewed in a browser, but it does appear in the title bar of the browser window (or the tab title if using tabbed browsing). It is also the title that appears when a user bookmarks the page. An engaging and descriptive page title may entice a visitor to revisit your page. **Web sites should always have short and meaningful page titles!**

1. To add a title to your page, change your code so that it looks like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
 <html>
<head>
<title>My first web page</title>
</head>
 <body>
My name is _____ (type your name) and this is my first web page.
</body>
</html>
```

2. Now save the document again, go back to the web browser and select "refresh" (which will reload the page). When you look at this document in the browser, you will see that "My first web page" will appear on the title bar of the window (not the actual canvas area). The text that you put in between the title tags has become the title of the document (surprise!).

We have added two new elements. The head element (that which starts with the <head> opening tag and ends with the </head> tag) appears before the body element (starting with <body> and ending with </body>) and contains information about the page. The information in the head element does not appear in the browser window.

We will see later on that other elements can appear inside the head element, but the most important of them is the title element.

Paragraphs

1. Go back to your text editor and add another line to your page:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
 <html>
<head>
<title>My first web page</title>
</head>
 <body>
My name is _______ (type your name) and this is my first web page.
This is the most exciting thing I have done!
</body>
</html>
```

   Look at the document in your browser. You might have expected your document to appear as you typed it, on two lines, but instead you should see something like:

   My name is _____ and this is my first web page. This is the most exciting thing I've done!

   This is because web browsers don't usually take any notice of what line your code is on. It also doesn't take any notice of spaces. If you want text to appear on different lines, you need to explicitly state that.

2. Change your two lines of content so that they look like this:

   ** My name is _____ and this is my first web page. **
   ** This is the most exciting thing I have done! **

   The p tag is for **paragraph**. Look at the results of this. The two lines will now appear on two lines. Think of the HTML content as if it were a book - with paragraphs where appropriate.

Another important note about HTML – unlike Word Processors, HTML automatically adds space above and below content inside tags and all heading tags. To remove this spacing you will need to modify the margin and padding settings of the element. Don't worry – we'll work with this in Dreamweaver later this term!

Emphasis

You can emphasize text in a paragraph using em (emphasis) and strong (strong emphasis). These are two ways of doing pretty much the same thing, although traditionally, browsers display em in italics and strong in bold.

3. Add one more line of text to your document:

   My name is _____ and this is my first web page
   <p**>**This is the most exciting thing I have done!**</**p**>**
   ** Yes, that is what I said. How very exciting. **

Line breaks

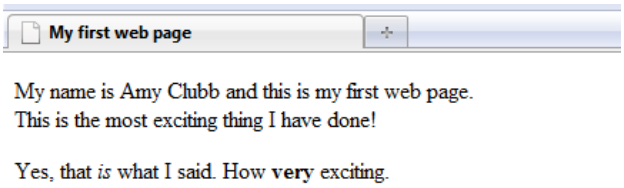
The line-break tag can also be used to separate lines like this:

This is my first web page**<**br **/>** How exciting

However, this method is over-used and shouldn't be used if two blocks of text are intended to be separate from one another (because if that's what you want to do you probably want the p tag). You should use the
 tag when you simply want to start a new line within a paragraph.

Note that because there's no content involved with the line-break tag, there is no closing tag and it closes itself with a "/" after the "br".

4. Now save the document again, go back to the web browser and select "refresh" (which will reload the page). Your document should look like this:



Headings

If you have documents with genuine headings, then there are HTML tags specifically designed just for them. They are h1, h2, h3, h4, h5, and h6. h1 is the largest of the group and h6 is the smallest.

1. Add the following text after the opening <body> tag:

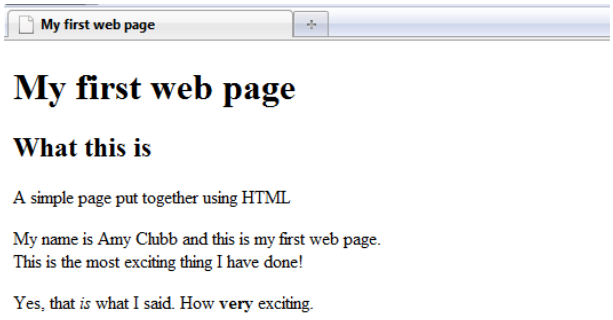
```
<h1>My first web page</h1>
<h2>What this is</h2>
<p>A simple page put together using HTML</p>
```

Note that the h1 tag is only used once – it is supposed to be the main heading of the page and shouldn't be used multiple times. h2-h6, however, can be used as often as you desire, but they should always be used in order, as they were intended. For example, an h4 should be a sub-heading of an h3, which should be a sub-heading of an h2.

Note about Headings and Accessibility:

Heading tags can help to make your pages more accessible and usable. Visually challenged visitors who are using a screen reader can configure the software to display a list of the headings used on a page to focus on the topics that interest them. A well-organized page will be more usable for every visitor to your site, including those who are visually challenged.

2. Now save the document again, go back to the web browser and select "refresh" (which will reload the page). Your document should look like this:



Lists

There are three types of lists: unordered lists, ordered lists, and definition lists. In this tutorial, we will look at the first two types of lists.

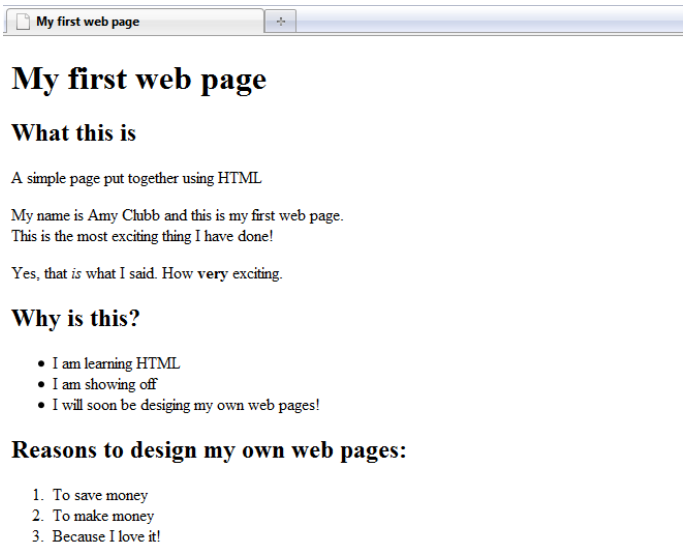
Unordered lists and **ordered** lists work the same way in HTML, except that the first is used for non-sequential lists with list items – such as bulleted lists; and the latter is for sequential lists – such as numbered lists.

The **ul** tag is used to define unordered lists and **ol** tag is used to define ordered lists. Inside the lists, the **li** tag is used to define each list item.

1. Add the following text after your last </p> tag:

```
<h2>Why is this?</h2>
<ul>
        <li>I am learning HTML</li>
        <li>I am showing off</li>
        <li>I will soon be designing my own web pages!</li>
</ul>
<h2>Reasons to design my own web pages:</h2>
<ol>
        <li>To save money</li>
        <li>To make money</li>
        <li>Because I love it!</li>
</ol>
```

2. Now save the document again, go back to the web browser and select "refresh" (which will reload the page). Your document should look like this:



Links

So far you've been making a stand-alone web page, which is all very well and nice, but what makes the internet so special is that it all links together! The 'H' and 'T' in 'HTML' stand for 'hypertext', which basically means a system of linked text. An **anchor** tag is used to define a link, but you also need to add something to the anchor tag – the **destination** of the link.

1. Add the following text below your ordered list:

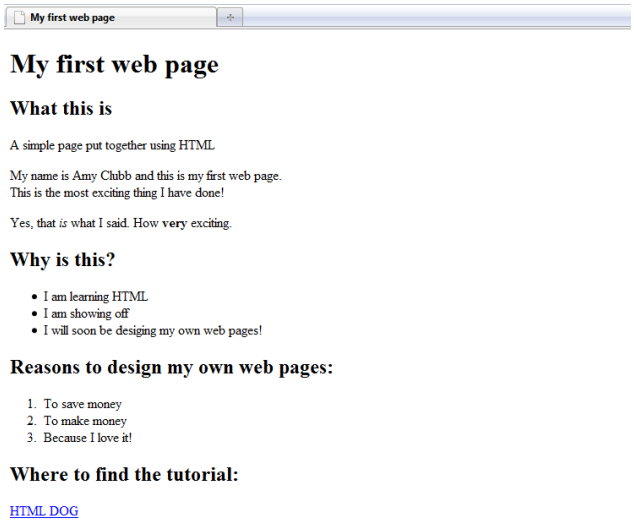
```
<h2>Where to find the tutorial:</h2>
<p><a href="http://www.htmldog.com">HTML Dog</a></p>
```

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. We will discuss the differences between absolute and relative links further once we begin working in Dreamweaver.

A link does not have to link to another HTML file. It can link to any file anywhere on the web. The a tag allows you to open the link in a newly spawned window, rather than replacing the web page the user is on, which at first thought may sound like a good idea. However, there are a number of issues to consider when doing this:

From a usability point of view, this method breaks navigation. The most commonly used navigation tool on a browser is the "back" button. Opening a new window disables this function. On a wider, more general usability point, users do not want new windows to be popping up all over the place. If they want to open a link in a new window then they can choose to do so themselves.

2. Now save the document again, go back to the web browser and select "refresh" (which will reload the page). Your document should look like this:



Images

Things might seem a little bland and boring with all of this text formatting. Of course, the web is not just about text, it is multi-media and the most common form of media is the **image**.

The **img** tag is used to put an image in an HTML document. It has several attributes that go with it. The **src** attribute tells the browser where to find the image. Like the a tag, this can be absolute, but is usually relative. For example, if you create your own image and save is as “alienpie.jpg” in a directory called “images” then the code would be <img src=”imgages/alienpie.jpg”…

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. The is used for people who cannot or choose not to view images. This is a requirement in XHTML.

Note that, like the br tag, because the img tag does not have a closing tag, it closes itself, ending with “/>”

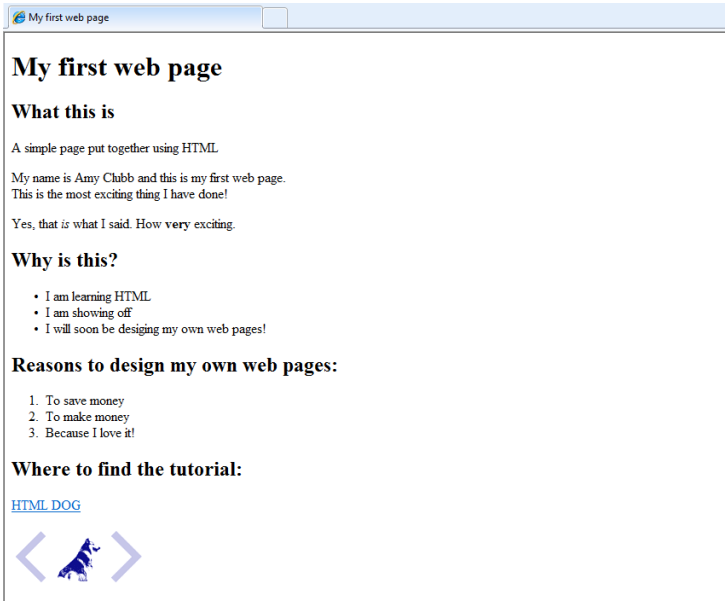
1. Add the following img directly above the closing BODY tag. :

```
<img src="http://www.htmldog.com/images/logo.gif" width="157"
height="70" alt="HTML Dog logo" />
```

The construction of images for the web is a little outside of the remit of this tutorial, but it is worth noting a few things...

- The most commonly used file formats used for images are GIFs and JPEGs. They are both compressed formats, and have very different uses.
- GIFs can have no more than 256 colours, but they maintain the colours of the original image. The lower the number of colours you have in the image, the lower the file size will be.
- GIFS SHOULD BE USED FOR IMAGES WITH SOLID COLORS.
- JPEGs on the other hand use a mathematical algorithm to compress the image and will distort the original slightly. The lower the compression, the higher the file size, but the clearer the image.
- JPEGS SHOULD BE USED FOR IMAGES SUCH AS PHOTOGRAPHS.
- Images are perhaps the largest files a new web designer will be handling. It is a common mistake to be oblivious to the file size of images, which can be extremely large. Web pages should download as quickly as possible, and if you keep in mind that most people still use modems that download at less than 7Kb a second (realistically it is less than 5Kb), you can see how a large file will greatly slow down the download time of a full page.
- You need to strike a balance between image quality and image size. Most modern image manipulation programs allow you to compress images and the best way to figure out what is best suited for yourself is trial and error.

2. Now save the document again, go back to the web browser and select "refresh" (which will reload the page). Your document should look like this:



Congratulations on your first web page! After all that typing of code, you will really come to appreciate the power of using Dreamweaver and having it do all the coding for you. To submit this tutorial assignment, complete the following:

Submission Instructions:

1. Upload the completed file (myfirstwebpage.html) to the Student Web Server (SWS). The file should be uploaded to the **Additional Exercises** folder you created earlier. You will be using Filezilla to FTP your file.