WEEK 1 NOTES

|  |
| --- |
| **"Hello, World"** |

**"Hello World"**

Traditionally, a beginner's book for a programming language starts out with a lesson on how to create a program that does nothing other than create a simple "Hello World" Message. HTML is not a programming language; it's a mark-up language (more on the distinction later). But, the principle is the same. To get started, let's create a simple page that displays a "Hello World" message. This should help you become familiar with the tools used to create a simple page, and get the logistics out of our way first.

**Step 1: Find a good text editor**

The first thing you will need is a good text editor. If you are using a Macintosh (or any computer running the Mac OS), you probably have a copy of SimpleText already installed on your hard drive. That would do the job, but I would **strongly** recommend using something better--check out the listing [at About.com's Webdesign page](http://webdesign.about.com/od/macintoshhtmleditors/tp/free-macintosh-editors.htm). My preference is [Komodo Edit](http://komodoide.com/komodo-edit/" \o "Komodo Edit" \t "_blank) (it's also available for other platforms), because it's free and open-sourc, although Alpha is also pretty good.

If you are using a Windows, NotePad comes with your system. That does the job fairly well. [HTML-Kit (the free version)](http://www.htmlkit.com/download/) is a dated but effective editor, and, as I mentioned, [Komodo Edit](http://komodoide.com/komodo-edit/" \o "Komodo Edit" \t "_blank) is a great editor that lets you choose to create a new HTML5 document off the bat.

BTW, just about any word processor can also create plain text files. For example, Word 5.1 for Macintosh allows you to Save As Text. When you do this, you lose any formatting or pictures inserted (because these aren't text). This is OK if you want, but usually a text editor will give you more options relavant to creating simple text files--and I recommend using them instead of your word processor. The choice is yours, though.

The one thing I do **not** want you to do right now is use a WYSIWYG HTML editor, or to Save As HTML from a word processor. In general, there is nothing wrong with doing either of these, but what we're doing right now is learning HTML. And, as we'll be discussing during this course, HTML is not a WYSIWYG language. Later on in the course we can use these--but we have to understand the HTML basics first!

**Step 2: Create a new text document**

Once you have your text editor installed on your computer and working, we're all set to go. Now, create a new file with your text editor.

**Step 3: Make it an HTML document**

The first thing we want to do with our document is make sure it's an HTML document. To do that, put the following at the very beginning of your file:  
  
<!DOCTYPE html>--tells the browser what version of HTML (HMTL5, in our case) we're using. This is the first line of all your pages. We'll just place it here for now, but know that every HTML5 will begin with the DOCTYPE. (Believe me, this is much simpler than earlier DOCTYPEs!)

<html>

And then put the following at the very end of your file:

</html>

**Step 4: Give your document a header and a body**

Every HTML document consists of two parts, the **header** and the **body**. The header is the part of the document that contains information about the page that the browser (Netscape, Internet Explorer, etc.) might use. The body is the document itself--and contains the text that the reader will see, links to other pages, pictures, etc. Basically, the vast majority of what we'll be discussing after this lesson will go in the body.Â

The header generally goes first, within the two HTML tags. It begins with and ends with tags, as follows:

<html>

<head>

</head>

</head>

HTML5 has other tags like <atricle>, which we'll investigate next week.   
  
There is one thing we need to put in the Header that is required--that is the Title. The Title goes inside the Header, begins with . What the Title is goes between those two tags. I'll give this page the title: "Hello World" Page. Now, the page looks like the following:

<html>

<head>

<title>"Hello World" Page</title>

</head>

</html>

Last, but not least, we need to put in the Body. The Body begins with and ends with . Once we insert that, we have:

<html>

<head>

<title>"Hello World" Page

</title>

</head>

<body>

</body>

</html>

**Step 5: Make the page display "Hello World"**

Our purpose of this simple page was for it to display "Hello World." Therefore, "Hello World" will be put in the Body of the page. We're going to put this text into a Paragraph. A paragraph starts with <p> and ends with </p>. The text of the paragraph goes between those two markers, as follows:

<html>

<head>

<title>"Hello World" Page</title>

</head><body>

<p>Hello World</p>

</body>

</html>

**Step 6: Save the file**

Now you need to save this file to your hard drive. There are a few simple rules you need to follow when choosing a file name for an HTML document:

* The filename should end with either ".htm" or ".html". Generally, using ".html" is best, but some older operating systems (Windows 3.1 and earlier, for example) don't allow for four-character file-naming extensions. So, if you're using Windows 95 to XP, Mac OS, or Linux/UNIX, end the filename with ".html". If you're using Windows 3.1 or earlier, use ".htm" instead. We'll discuss why this is necessary later.
* It is best not to put any fancy characters into an HTML page filename. No spaces, no special characters, no periods other than the one that ".html" or ".htm" start. Just letters and numbers, and then the extension. We'll discuss why this is later on. Also, be sure to keep your filenames all lower case! Different server operating systems treat case differently, so, to be safe, use all lowercase letters.

I think I'll name this file "hello.html."Â

**Step 7: View this page in your Browser**

OK, now you want to see what we've done. Open up your browser (Internet Explorer, Netscape, whatever) and open the file. To do that in most browsers, go to the "File" menu and select either "Open" or "Open File." Then, find the file you created on your hard drive and select it. You should see the page show up, looking something like this:

Notice two things:

1. The document says "Hello World."
2. The Title bar of the window says "'Hello World' Page." A browser doesn't have to work this way: Netscape displays "Netscape: 'Hello World' Page, and Internet Explorer just displays "'Hello Word' Page." Other browsers might do something completely different, or nothing. If you are using Netscape or Internet Explorer, though, and don't see either "'Hello World' Page" or "Netscape: 'Hello World' Page" in the Title bar, though, there may be something wrong with the syntax.

There are always browser differences and operating system differences, but if "Hello World" doesn't show up as the page, or if something looks really wrong to you, check to make sure your page markup looks exactly what I wrote (computers are picky about typos), and if so then post in the discussion list.

**Step 8: Pat yourself on the Back!**

Congratulations, you just created an HTML document. Pat yourself on the back, grab a beer, and then go on to the next step: putting a picture in the page. :-)

|  |
| --- |
| **Adding an image** |

**Adding an Image**

The next thing we are going to learn to do is to put a picture in the page.

**Step 1: Find Graphics Software that can make a GIF.**

For the Macintosh, I recommend [GIFConverter](http://www.kamit.com/" \t "_blank).

For Windows, I recommend [Paint Shop Pro](http://www.jasc.com/psp.html).

There are many other programs that do this as well, but these two are inexpensive and do everything we'll need. PhotoShop is ideal--but PhotoShop is really expensive, so I don't have it and I don't recommend going out and buying it unless you really think you'll benefit from all of its features.

BTW, GIF stands for Graphics Interchange Format; this is the image format that just about any web browser can read--although alternatives include JPEG and PNG; we'll be talking about the differences between formats and what browsers support which formats later in the course. For now, let's stick to GIFs.

**Step 2: Find a picture, any picture--and copy it to the Clipboard**

The first thing you want to do is to find a picture. I don't care what the picture is of, just something that you can use to put on a web page. If you're using a Macintosh, you might just want to use a folder icon. To select a folder icon and copy it to the clipboard, just select it in the Finder, select "Get Info" from the "File" Menu, select the picture of the icon in the Info window, and select "Copy" from the "Edit" Menu.

Also, many word processors come with ready-made ClipArt to be inserted into a program--you might want to use one of those and copy it to the clipboard. Anyway, it doesn't matter what you're picture is of, just find a picture and copy it to the Clipboard.

**Step 3: Make it a GIF**

Paste the image into your new graphics program and save it **in the same folder or directory as your "Hello World" Page**. Again, there are some special rules for saving this file:

* The filename should end with ".gif."
* Just like with the HTML file--It is best not to put any fancy characters into an HTML page filename. No spaces, no special characters, no periods other than the one that ".gif."

Make sure save as a GIF; most graphics programs let you save to many different graphics formats; GIF is only one of them. Putting the "gif" extension on the name alone does not make it a GIF--it has to be saved in the proper format as well.

**Step 4: Reference it from your "Hello World" Page.**

The tag used to put an image in the page is the following:

<img />

Between the two quotes we'll put the filename. I named my GIF file "folder.gif", so I'll need to use the following tag:

<img src="folder.gif" />

Now, let's create a paragraph for this picture:

<p><img src="folder.gif" /></p>

And let's insert this picture into the Body of the "Hello World" Page:

<html><head>

<title>"Hello World" Page</title>

</head><body>

<p>Hello World</p>

<p><img src="folder.gif" /></p>

</body></html>

**Be sure to save this file now.** If you don't, the browser will look at the file before the changes we just made.

**Step 5: View this page in your Browser**

As you did before, open the page in your browser.

If the page looks exactly as it did before, use the "Reload" or "Refresh" button or menu item in your browser. What is happening if you get this result is that the browser has saved a copy of your file for quick loading the last time it loaded it, because it assumed it was loading the file from the network. Most browsers do this in order to reduce network traffic, reduce the load on the computer serving the page, and to improve performance for you. Of course, since the file is on your hard drive now, it didn't benefit you this time.

The page should look something like this:

My picture is of a folder icon; obviously yours will be different if you used a different picture, but you get the idea...

If everything worked, good job! If not, post a note to the Web Board and I'll give you a hand in troubleshooting. Make sure that the image and the HTML file are in the same directory. If you've posted the file and get a broken-image icon, your image link is probably something like C:/myimages/myimage.gif. That means your system defaulted to your hard drive, and you need to change the reference to just myimage.gif.

**Recommendation: An Index Page**

As you complete your assignments, you would do well to create a main, index page to list your work; that way, you can refer just to that page, with the updated list and links of projects/assignments.

|  |
| --- |
| **Filenames** |

**Important: Filenames**

I know I covered this in previous pages, but I feel this issue is one that warrants its own page and repeating.

**It is really important that filenames for your web document files have the following characteristics:**

The filename should have **no spaces,** **no special characters**. The only characters in the filename should be letters and digits--and there should be **only one** period.

The period should start the **suffix** of the file. If you're using Windows, your suffix might be a three-character one. Otherwise, you should use the preferred one for your file type. The suffixes to use are as follows:

|  |  |  |
| --- | --- | --- |
| **File Type** | **Preferred Suffix** | **Three-character Suffix** |
| HTML File | .html | .htm |
| GIF Image | .gif | .gif |
| JPEG Image | .jpeg | .jpg |
| PNG Image | .png | .png |

**Note also that all file names and suffixes MUST BE IN LOWERCASE**. If you do not use these suffixes, or if you put odd characters in your filenames, you will create problems for yourself. Many times, these problems won't appear when you're editing the files on your local computer, but they will when you upload them to the server.

**Case Sensitivity**

The server you put your files on for this course, uml.edu, does not have a case-sensitive file system. Therefore, you can name a file "myfile.html" and refer to it from another file as "MyFile.html," and have no problem. This is not a good idea, though. The first reason is that this is not how all servers work--and if you ever want to move your files from this server to one that is case-sensitive, you will run into problems.

The second reason is that browsers do treat URLs as being case-sensitive, because some servers have case-sensitive file systems (e.g., Linux and UNIX systems). The right file will always be retrieved, but if the browser loads http://profbill.uml.edu/myfile.html and later finds a reference to http://profbill.uml.edu/MyFile.html, a few things will not go as expected:

* Because the browser sees these as two unrelated URLs, it will display the link with the second URL as unread, even though the first URL has been read.
* If you load the second URL while the first has been loaded and cached, the second URL loading will not take advantage of the caching, and the page will load slower.

|  |
| --- |
| **Reading Assignment: HTML5, Browsers, and The Web** |

# Reading Assignment: HTML5, Browsers, and The Web

* As I suspect some of you will get through those first four pages **much** more quickly than many others. This is actually next week's assignment, but I don't want anyone to be bored, so some of you may want to start on it early.
* This reading will give you a good overview of writing an HTML5 document. You already have a template, now you'll have stuff to use it for (plain ol' paragraphs get pretty boring rather quickly!).
* Read 20 *Things I've Learned About Browsers And The Web*, included in our Misc. Info module at the bottom.
* Also, I want you to read *HTML5: An Introduction*, also in the Misc. Info module. Remember my mentioning DOCTYPE? You'll find a bit more information here.
* Both ebooks are brief and very entertaining.
* In that module, you'll also see some great stuff on JavaScript and Cascading Style Sheets!
* **Optional Videos**
* One take-away we'll see repeatedly in this course is that the Web is very good at teaching us about itself. If you like, search YouTube or any other video site for coding or design tips. Post any recommendations to our discussion lists!

WEEK 2 NOTES