**BUA 235 lab #1 (Sept. 1): create a hypertext web site with HTML**

HTML is a “markup” language, not a programming language. It uses “tags” to add information to ordinary text. For example, the <B> tag means “boldface”. The matching tag </B> ends the boldface font. So this code:

<B>These three words</B> are in bold.

Would appear on screen as:

**These three words** are in bold.

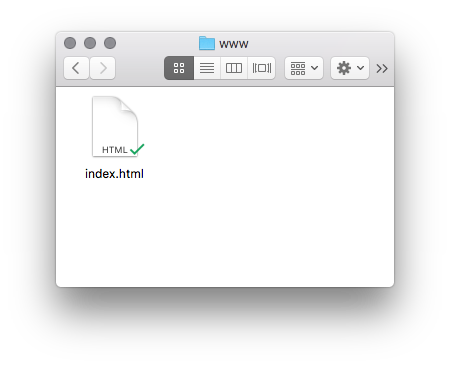
**Getting Started**

An HTML file is just a “plain text” file that uses HTML markup. You can create and edit a plain text file with Notepad (on Windows) or TextEdit (on Mac), but these are really not ideal. There are several free choices that are much better for writing code:

* Notepad++ (for Windows): <https://notepad-plus-plus.org/>
* TextWrangler (for Mac): <http://www.barebones.com/>
* Sublime Text (both): <http://www.sublimetext.com/>

The advantages of these tools include syntax highlighting, which makes code much more readable and helps you find mistakes, as well as line numbers, find-and-replace, and other handy features.

You could also download an integrated development environment (IDE) specially designed for web development. Atom is a popular free one. JetBrains makes a commercial IDE called WebStorm that’s free for students. I don’t recommend this option for today, because there are too many bells and whistles to learn. You might try an IDE at home later.

One other alternative I’ve experimented with is a “development environment in the cloud” provided by Cloud9 (<https://c9.io)>. They set everything up for you so you don’t have to install anything on your computer, just access it through a web browser. This might be a solution for those of you who don’t have the ability to install new programs on your laptop. I don’t know it well, though, so I can’t easily help you with it.

Whichever tool you use, start by creating a new folder and a plain-text file in that folder called “index.html”.

Note: *Do not use a word processor like Microsoft Word or Apple Pages to create this code file. Word processors save lots of “non-text” information, like fonts, layouts, and settings, so they are not plain-text files.*

**Hello World!**

It is a tradition among computer programmers to test out new technology by making it say “Hello world!”. You can make this happen by coding your new web page like so:

<html>

Hello world!

</html>

Then double-click the file, or drag it onto a web browser window, to see it in the browser.

**A more proper beginning**

The above isn’t exactly kosher HTML, but it works because the browsers are smart enough to fill in what’s missing. The bare minimum for a “proper” web page looks like the following example. Many IDEs will give this to you automatically. Here it is for your reference:

<!DOCTYPE html>

<html>

<head>

<title>Hello world</title>

</head>

<body>

Hello world!

</body>

</html>

In this document, the <body> tag indicates the main content of the page. Your “Hello world” and any other web page content should go in between <body> and </body>. The <title> tag is also worth noticing. It sets the words that will appear in the bar at the top of the window.

**A personal web site**

Back in the days before Facebook, yes, even before MySpace and LiveJournal, college students would create their own personal web sites. This is a little “retro” now, but a good choice for this lab because none of you will lack things to talk about – yourselves!

This is the first step in an assignment that will be due on October 13, to create a personal website. The requirements for that assignment are:

* minimum 4 pages or screens of information
* effective hypertext links for navigation between screens
* appealing visual style and layout

The assignment will be graded on style and effectiveness; more details will come later.

**Start with a homepage**

Begin by creating the text for your homepage. Do this in plain text, and I’ll show you how to add HTML markup later. I suggest starting with your name, a subtitle of some kind, a paragraph of introduction, and placeholders for images and links. Here’s mine. (Remember, this goes between the <body> and </body> tags.)

Joseph W. Clark, Ph.D.

Business Faculty at UMaine, Computer and Database Geek

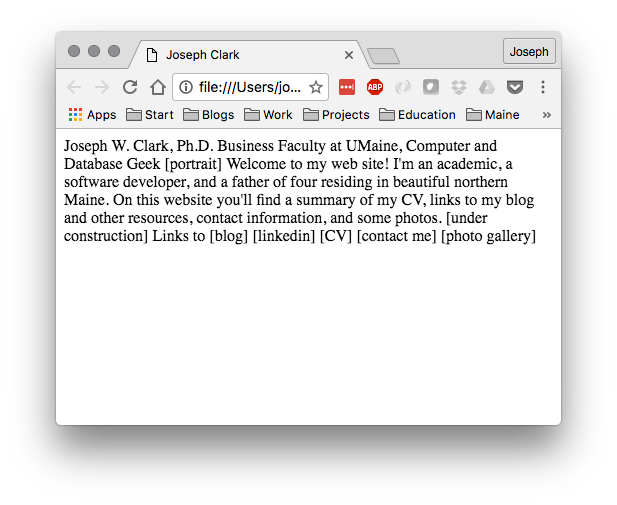
[portrait]

Welcome to my web site! I'm an academic, a software developer, and a father of four residing in beautiful northern Maine. On this website you'll find a summary of my CV, links to my blog and other resources, contact information, and some photos.

[under construction]

Links to [blog] [linkedin] [CV] [contact me] [photo gallery]

Now, there’s no HTML markup in here yet, so if you look at it in the browser it will appear jumbled, as you can see:



**The “ML” in “HTML” is Markup Language**

The next step is to mark up the code with HTML tags. Here are some of the first ones that will be useful:

<H1> Heading </H1>

Makes the text large, bold, and on a line of its own.   
There are also H2, H3, H4, H5, and H6 tags for sub-headings.

<P> Paragraph </P>

Wrap a block of text with this tag and it will be separated from preceding and following text by line breaks.

<DIV> Text block </DIV>

Like <P>, but with no extra whitespace between the blocks above and below.

<B>Bold</B> <I>Italic</I>

Markup to style text as bold or italic.

<IMG SRC="portrait.jpg">

Includes the specified image into the page. (Assuming it’s in the same folder.)

<UL>

<LI>list item 1</LI>

<LI>list item 2</LI>

<LI>list item 3</LI>

</UL>

An “unordered list” (bullet list) with three items. Use <OL> for a numbered list.

<A HREF="https://umaine.edu/"> UMaine </A>

A hyperlink to another web address. “UMaine” will be seen by the user. Clicking the link takes them to the address in HREF.

<A HREF="cv.html"> CV </A>

A hyperlink to another page in the same website. “cv.html” must be in the same folder.

This should be enough to get you started. For more HTML tags, take a look at <http://www.w3schools.com/>, by far the most popular online reference site for web developers. One of the next more advanced things you might want to try is a <TABLE> for laying out text in a grid of rows and columns.

**Instructions for today’s Lab**

Today, lay the foundation for a personal web page by creating 4-5 HTML files with some placeholder text. (You can add more later.) The most important thing is to create the links between pages, making this a “hypertext” document. The homepage should have links to all the other pages, and every page should have links back to the homepage (at least). You can add links to your blog, your LinkedIn profile, and any other sites that interest you.

Before the next lab you should:

* have 4-5 pages
* add at least one picture, just to make sure you know how
* have links between them
* fix any bugs or broken HTML tags so the pages work

Don’t worry about the fact that the visual style of your site is pretty plain. In the next lab, I’ll introduce you to CSS, the “style sheet” language for adding fonts, colors, and styles.