Unit 1 – Introduction to Web Development

Objectives	- Learn how to create and view a web page on your local hard drive. - Learn the structural tags of a web page, headings, paragraphs, and lists.
Required Reading	- Lesson 1 - Navigating the World Wide Web, Lemay, p. 9 - Lesson 3 - Introducing HTML and XHTML, Lemay, p. 47 - Lesson 4 - Learning the Basic of HTML, Lemay, p. 65 - How Web Server Works http://computer.howstuffworks.com/web-server.htm - How Web Pages Work http://computer.howstuffworks.com/web-page.htm - How Internet Infrastructure Works http://computer.howstuffworks.com/internet-infrastructure.htm - How Domain Name Servers Works http://computer.howstuffworks.com/dns.htm - Chapter 1 - Understanding the Web Design Environment, Sklar, p. 2
Assignment	Create a basic web page that you will use as your assignments.html page. You will learn how to upload the page in Unit 2.

***Warning:** There's a lot of important information coming up. It is important to read, re-read, and maybe re-read 5 more times.* :-) I'll be honest... the information to follow is sometimes confusing, but it's important for you to develop a good understanding about how this stuff works. If there are parts that you don't understand, go to the Assignment Page (Unit 1 & 2) discussion board and ask for help. I guarantee, in fact... I promise, you're not the only one who's lost. If you suck it up, swallow your pride, and start asking questions, I know your fellow classmates will love you for it.

Being the first person to ask questions is tough, but something that can be even harder is being the person who answers the questions. Don't rely on the people in the ZONE to answer all of your questions. Help out your classmates and yourself by jumping in there an answering the question. I guarantee (wow, I've guaranteed twice in one document) that you will develop a better understanding of this material by trying to explain it to someone else.

There is a considerable amount of reading for this unit. It may seem overwhelming, but the load will get lighter after the first couple of units. We need to be sure that everyone is starting with the information that they need to be successful in this course. So, bear with me... :)

After reading the text, you should create a web paged named **assignments.html** and view this web page offline. You will upload this file in Unit 2. Read Exercise 3.2: Viewing the Result on Lemay, page 56 for more information on viewing a web page offline. For detail instruction of the Assignment Page, please refer to **Assignment Page Instruction** under *[Resources / 4. Guidelines for Assignments / Assignment Page].*

I. Important Concepts

1. Basic structure of HTML web page. (Lemay, p. 66)

Below is the basic structure for a HTML file. Each **.html** file needs to include these tags. You may save it as a template for further use. You can always start with this file without typing the basic tags every time.

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

<html>
<head>
<title>This is a testing page </title>
</head>
<body>
 This is where to put your page content.
</body>
</html>
```

2. How to name your file? (Lemay, p. 56)

2.1 Always use lower case:

UNIX is **case sensitive**, which means that you'll make your life much easier by always naming your files and referring to them in the link tags you'll create in **LOWER CASE**. (Yes, I used upper case to emphasize that!) Let's say that you create a HTML file that will be a web page about gardening. You save it as garden.html. Then you make another page about cooking with fresh vegetables and use an anchor tag (<a href="">) to link this page back to the gardening page. If the file has been saved in lowercase (garden.html) and your tag references the file like this: <a href="GARDEN.HTML">, the server won't find that file because it can only recognize the lowercase "garden.html." The <a href> should be: <a href="garden.html"> (note the matching case). It's easiest to consistently use lower case and short file names.

2.2 Use small and simple names:

Never use a space or special characters in the file name because UNIX will not recognize spaces and special characters (e.g. $, #, %, etc.) in a file name.

3. Relative and Absolute Paths (Lemay, p. 101)

When we talk about paths, we're referring to just that -- the path to the server, directories, and files. A **RELATIVE PATH** is one that points to another file as its location relates to the file containing the link. When you learn the linking tag, you'll learn more about paths. For now, just be aware that an **ABSOLUTE PATH** contains the full pathname (e.g. http://zone.missouri.edu/index.html)

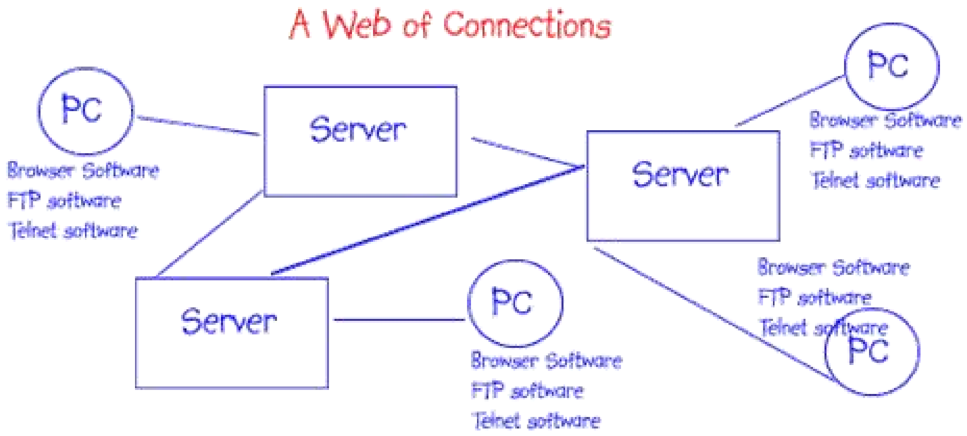
You may notice that although your individual files are nested within a "www" directory in your home directory (your directory is named your pawprint), the "www" doesn't appear in the pathname of the URL. Just remember that all the web files you create, whether text, graphic, audio, video, etc., need to be placed in your "www" directory to show up on the web. The file names show; the "www" part of the pathname doesn't.

4. Bengal Server Space

Every mizzou student has **a bengal server space** up to 200 MB for web files storage. Once you have your web page done and upload (we will learn how to upload files using a Secure FTP program in unit 2) it to the bengal server, you are accessing a computer running a UNIX operating system. Most of us are used to working with either a Macintosh or Windows operating system (OS). UNIX is quite a bit different; fortunately you'll only need to know a few things to successfully use it for storing your web files.

5. Browsers, Servers, & Web Pages

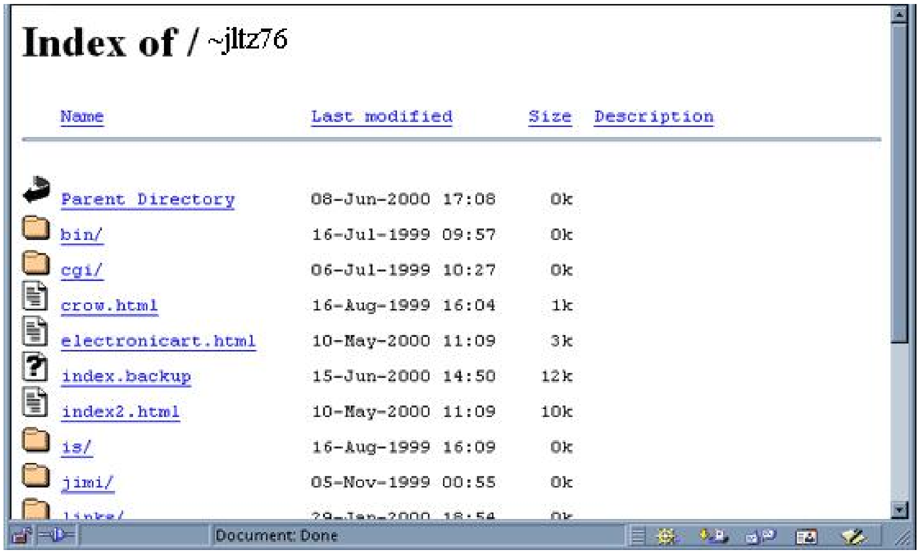
As you've also learned from Lesson 1, browsers and servers work together to deliver information on the World Wide Web. Here's a picture that may help you visualize the interactions between them. Imagine a web of connections between the servers and individual computers.



When you've created a web page, you usually name the first one "**index.html**." That's because when the most web servers receive a http request from a web browser, index.html is the name of the file that they **automatically look for first** in your www directory (or in any subdirectory) it encounters. If the server can't find this file, then it will display a list of all files in that directory. The reason for saving your initial html file with the name of "index" is that the server will, by default, display the index.html file even if all someone types in the location box is: http://www.missouri.edu/~yourpawprint/ Of course, you can always access any web page in your www directory by typing in the complete URL, such as http://www.missouri.edu/~yourpawprint/assignments.html.

Again, if you have no index.html file in your www directory (for example, you name your initial file "biography.html"), the user would have to type in the entire pathname (http://bengal.missouri.edu/~youraccountname/biography.html). If he did not type it all in, and you had no index.html file, the server would simply display the contents of your directory. It would look something like this:

A www/ directory of a user with no index.html file when URL is entered.



Index of / ~jltz76

```
 Name                     Last modified       Size  Description

 Parent Directory         08-Jun-2000 17:08     0k
 bin/                     16-Jul-1999 09:57     0k
 cgi/                     06-Jul-1999 10:27     0k
 crow.html                16-Aug-1999 16:04     1k
 electronicart.html       10-May-2000 11:09     3k
 index.backup             15-Jun-2000 14:50    12k
 index2.html              10-May-2000 11:09    10k
 is/                      16-Aug-1999 16:09     0k
 jimi/                    05-Nov-1999 00:55     0k
 links/                   29-Jan-2000 18:54     0k
```

You can think of the server's displaying your directory contents as an index in a book. When you create an index.html file, you are replacing what the server would display as the index with what you create within the index.html file for it to display.

For the purposes of this course, you will **NOT** have an index.html file in your www directory. This allows us as instructors to look in this directory if needed to provide you feedback. When you complete your final project, your first page will need to be named "index.html" and will be in a subdirectory named "final".