

## FIT 100: Lab 4

# Constructing HTML Documents

### Introduction

The previous labs introduced the Dante account and navigating remote and local directory structures. Today is an introduction to HTML and how to create HTML documents to publish to your web site. We will use a reference to XHTML to ensure a very well-formed HTML document that follows the rules about tags VERY EXPLICITLY. This allows the document to match the specifications of newer markup languages and standards like XML. **What you need:** You should already have your email account and Student web page enabled.

### Reading

#### **Required:**

The [Reference](#) section of this lab.

#### **Helpful References(review/use in lab):**

- W3 Schools HTML Introduction [http://www.w3schools.com/html/html\\_intro.asp](http://www.w3schools.com/html/html_intro.asp)
- W3 Schools XHTML Introduction  
[http://www.w3schools.com/xhtml/xhtml\\_why.asp](http://www.w3schools.com/xhtml/xhtml_why.asp)  
[http://www.w3schools.com/xhtml/xhtml\\_html.asp](http://www.w3schools.com/xhtml/xhtml_html.asp)  
[http://www.w3schools.com/xhtml/xhtml\\_syntax.asp](http://www.w3schools.com/xhtml/xhtml_syntax.asp)
- *Review* Chapter 4 of the FIT book, pp. 62-74.
- **Chapters 1 – 7, 13 and Appendix A** of the *HTML For The World Wide Web with XHTML and CSS*

Use the table of contents and index to find reference information

### Objectives

- To use a text editor such as Notepad or Pico to create a simple HTML document
- To gain an understanding of basic tags used for simple web page formatting.
- Create and post a page to your student web site, either as the default web page or as an additional page in your site.
- To begin using your debugging skills.

### To Do

1. Create an HTML document
2. Save an HTML document
3. Add to the body of the HTML document
4. Create links to email, images and other pages
5. Publish your HTML document

## To Submit

Email the URL of your lab 4 Web page to your TA no later than 6 PM Friday night, January 16<sup>th</sup>.

## Details

### 1. Create an HTML document

1.1. Have you ever used Notepad before? It is a simple text editor that comes with Windows. To open it, click on the Start button, then go to Programs, then Accessories: (**Start > Programs > Accessories**) and click on the Notepad entry. If Notepad is not on the program menu, go to **Start > Run** and type in **Notepad** and click ok. You will be presented with a blank window and a flashing cursor in the top left corner.

1.2. All HTML documents, in order to be well-formed, must follow some very specific rules. Every new HTML file you create should begin with this text:

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

Type this in at the top of your Notepad document *exactly*. This will mean that you will need to pay attention to detail when adding HTML tags so that they are well-formed.

1.3. After the <!DOCTYPE...> tag, add open and close tags to indicate HTML as the markup language. Remember, all tags in HTML are identified with these angle bracket symbols <tag>. Enter the <html> tags that tell the browser that this page is to be interpreted using Hyper Text Markup Language:

```
<html>
    Put all other tags and text in-between the HTML beginning and ending
    tag.
</html>
```

1.4. Add <head> tags and include a title tag (with a title for your page. ex. **My first FIT 100 Web page**). Make sure to close the head and title tags. All new tags to be added to your document are in bold.

```
<html>
    <head>
        <title>My first FIT 100 Web page</title>
    </head> The rest of the tags and content will go here...
</html>
```

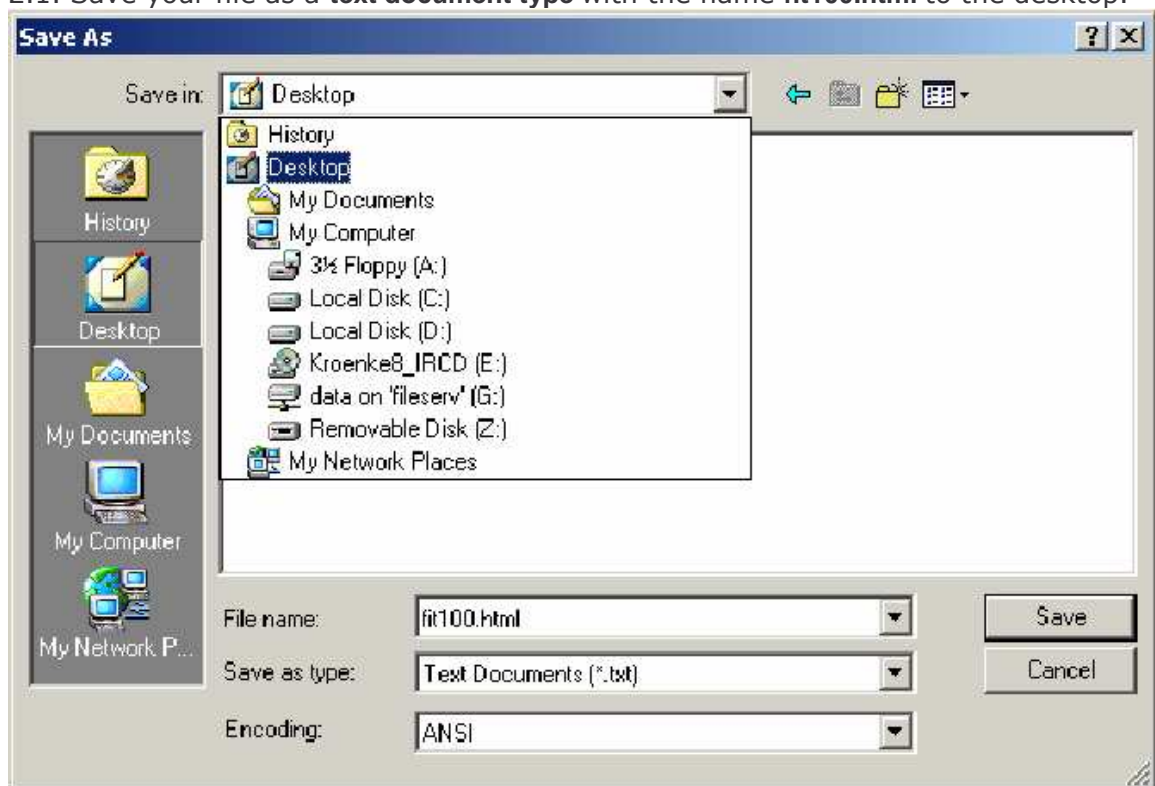
1.5. Just below the </head> tag and above the </html> tag, add in the body tags that will include the bulk of your content

```
<html>
    <head>
        <title>My FIT 100 Webpage</title>
    </head>
    <body>
        <p>This is my first web page for FIT 100! </p>
    </body>
</html>
```

## 2. Save an HTML document

**A good way to work:** When you are putting together web pages, a good way to work to keep track of what you do is to make one change and then check to see if it is working. Save what you have done up to this point and then open the file in a web browser to see if it works. Checking as you go is the first step in learning how to debug.

2.1. Save your file as a **text document type** with the name **fit100.html** to the desktop.



2.2. Open up your preferred web browser (Netscape or Explorer).

2.3. Browse (navigate) to the Desktop or the disk holding your HTML document.

2.4. Open this document to see your code displayed as a Web page. (**File>Open>** then **Browse** to your HTML file)

**Checking your work:** You should see the title of your Web page displayed in the top-left corner of the browser window and a single line of text in the content window.

Each time you make changes to your text file and save them, you will need to press the **Refresh**, or **Reload** button on your browser so that your new changes will appear.

Your code so far will look something 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 FIT 100 Web page</title>
  </head>
  <body>
    <p>This is my first web page for FIT 100! </p>
  </body>
</html>

```

### 3. Add to the body of the HTML document

3.1. If you need to, write out the proper HTML tags on the lab print out or some other paper, then add them to your file. Use Appendix A of your FIT book or the HTML attribute list at [http://www.w3schools.com/xhtml/xhtml\\_reference.asp](http://www.w3schools.com/xhtml/xhtml_reference.asp) Include inside your body tag the following additional tags:

- 3.1.1. A welcome sentence using an H1 or H2 tag. (Header 1 and Header 2 are tags that will display your text in a certain size, they have an open and close to them).
- 3.1.2. A line (horizontal rule) following the header end tag to separate the header information from the following paragraph.
- 3.1.3. Another paragraph tag.
  - 3.1.3.1. Include a short description of yourself and why you are in this class.
  - 3.1.3.2. Make part of the paragraph bold
  - 3.1.3.3. Make another part italicized
  - 3.1.3.4. Underline a third part
- 3.1.4. Create an ordered (numbered) list of your favorite colors

3.2. Now, use the font tags to make one of the color names the appropriate

**\*\*Remember\*\* Frequently save and check your work! Your motto should be: Save early and save often! Check to see what your html file looks like in the Web Browser window.**

color. Only do one for now, we will be learning how to use Style tags to do this in the next lab. You will need to use either hexadecimal or RGB color codes. Here are three links to web pages that give you the codes for colors:

- [http://www.w3schools.com/html/html\\_colors.asp](http://www.w3schools.com/html/html_colors.asp)
- <http://www.geocities.com/SiliconValley/Network/2397/>
- <http://www.lynda.com/hexh.html>

3.3. Create an unordered (bulleted) list of your favorite foods.

**3.4. REMEMBER TO PROPERLY CLOSE ALL TAGS!**

### 4. Create Links to Email, Images, and Other Pages

4.1. Using the anchor tag, add in the following three hyperlinks, or three of your favorite links:

- <http://www.washington.edu>
- <http://www.cs.washington.edu/100/>

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

4.2. Use anchor text for the links that will properly identify them on your page.

4.3. Use the "mailto" attribute and add a link to your page so that people can send you mail. *This link opens up an email message addressed to you and allows the user to fill in the subject and message.*

4.4. Add an image to your fit100 file.

To grab the full-size photo, not the thumbnail:

4.4.1. Click on the image you want to get the larger size, then

4.4.2. Right click on the image

4.4.3. Select **Save Picture As**, save the image in the same folder that is holding your HTML file. Give the image a short name (ex. "sun" or "dog"). It will be saved with the proper file extension (such as .jpg or .gif)

**\*\*\*Use images located at this URL: <http://freestockphotos.com/>**

**OR go out on the Web and find your own. Make sure you have permission to use the image if you just go searching \*\*\***

4.5. Open the text file of your web page in NotePad. (If you have already closed NotePad, open it again and open your HTML file.)

4.6. Add the image you have just saved into your web page by using the **** tag. By just listing the file name, you are using a relative path. If you were to give the entire link to get to the image on your account, like **<http://students.washington.edu/uwnetid/image.jpg>** then that would be an absolute path. Try to use relative paths as much as possible.

## 5. Publish your HTML document

5.1. Save your HTML file again and prepare it for transfer to the public\_html folder.

### What is public\_html or student\_html?

**public\_html** (called **student\_html** for some students) is the name given to the folder placed in every student's remote account on Dante once they have enabled web publishing. It is a directory space where you publish your HTML pages and is provided to you by the University. *The public\_html folder itself is a shortcut to server space set up for you on another server called Virgil. It is a convenience to be able to move files from your file space directly to your web server space with out having to make another network connection. For convenience, simply think of public\_html as your web folder.* All files that you wish to see up on your web page at the UW must be placed inside of this folder. If those files are not in this folder, then they cannot be seen by someone browsing the Internet and looking at your web site. The web documents that you create today are sent to this folder using the SSH file transfer client.

5.2. Open up the SSH client that we used in Lab1. Look for this icon on the Desktop:



You will need to log in with your UWNNetID and password. Make sure that the host computer name is **dante.u.washington.edu**

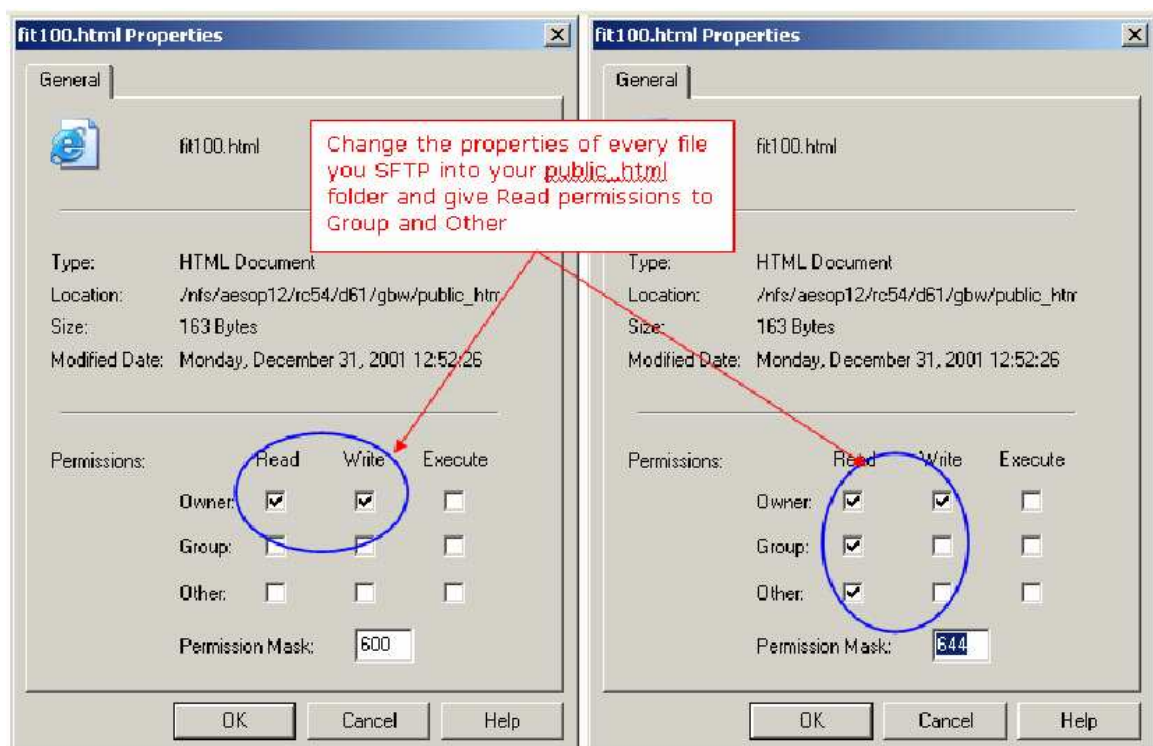
**NOTE:** You will need to select "All Files" from the Files of type menu in order to view your HTML file in the window.

5.3. **SFTP your document** inside of your **public\_html** folder.

5.3.1. When you SFTP your Web document over to your Dante account, be sure to transfer your image(s) as well, or they won't display.

5.3.2. IF, by default, all file permissions are changed to Owner read and write only, then you will have to change the permissions to make them readable by others.

5.3.3. Select the file, fit100.html, from the right side of the SFTP window. Right click and select Properties:



5.4. Once you have sent your file to your public\_html folder and changed the permissions on the file and any images, view the page from your browser. The URL to get to your web page should be:  
<http://students.washington.edu/YourUWNNetID/fit100.html>

(Replace **YourUWNnetID** with your own.) URL addresses are case sensitive. If you use capital letters in your file name, they need to be capitalized in the URL.

5.5. Log off the machine before leaving.

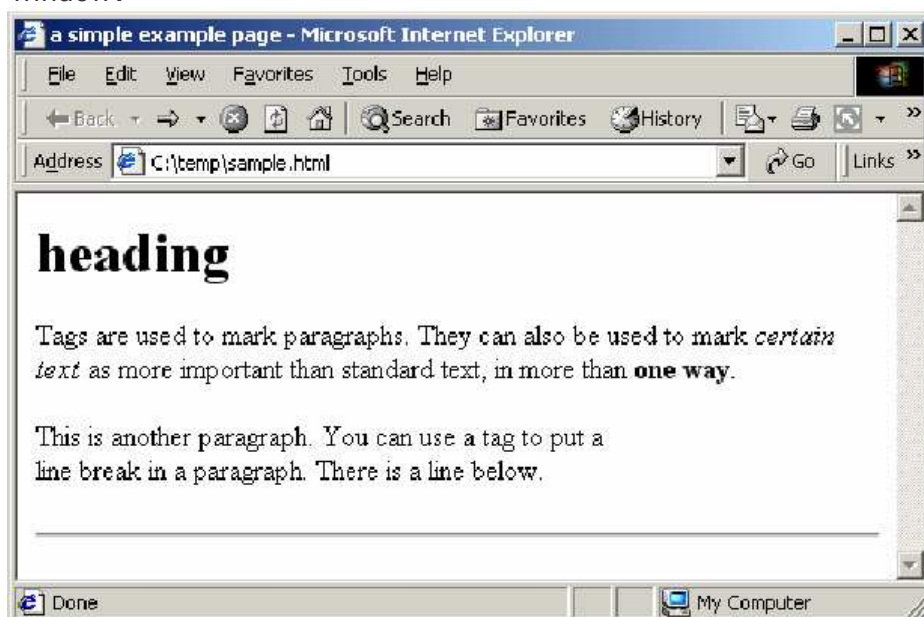
## Reference

### Basic HTML Syntax

**What HTML is and isn't.** *HTML* stands for Hypertext Mark-up Language and is used to produce web pages. "Hypertext" is text viewed on a computer that includes connections called "hyperlinks" or, more commonly, just *links*, which can be selected to jump to other documents, images, and other files. Hypertext existed before the web in the form of documents that were linked to other documents on the same computer (without using the network), but it became much more popular with the advent of the Internet and web browsers.

HTML has some characteristics of a *programming language* but is not one, strictly speaking. The main difference is that HTML is used to specify the structure of a web page (title, paragraphs, and images, for instance) and how it should appear in a web browser, whereas a programming language is used to specify what the computer should do, in the form of instructions. Like a programming language, HTML has special words and punctuation that must be very carefully used for the computer to understand them. These special words and punctuation, combined with rules for their proper usage, are known as *syntax*, a word we will see again when we discuss programming. Syntax is a lot like the computer equivalent of grammar for a language like HTML or a programming language.

**Two views of HTML: web page and source.** A web page is a file consisting of plain text that follows the syntax rules of HTML. Before we get started writing HTML ourselves, we will look at an example. Here is a very simple example web page, as viewed in a browser window:



*a simple example HTML file as viewed in a web browser*



(Note that the address of the page is shown as a pathname, not a URL. This is because the web page `sample.html` is being viewed locally, from the `temp` folder on Drive C, instead of from a web server. Although web pages are conventionally viewed via the network from a remote computer called a web server, web pages are files and can be stored on disks, just as with any other file.)

The file `sample.html` contains the following text, which is written in HTML and follows the rules of XHTML and is called the web page's *source*, to distinguish it from the web page as seen in the browser:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<head>
<title>a simple example page</title>
</head>
<body>
<p>Tags are used to mark paragraphs. They can also be used to mark
<em>certain text</em> as more important than standard text, in more than
<strong>one way</strong>.</p>
<p>This is another paragraph. You can use a tag to put a <br> line break in
a paragraph. There is a line below.</p>
<hr>
</body>
</html>
```

*HTML for the web page shown above*

The web browser's job is to read HTML source and display the page it describes.

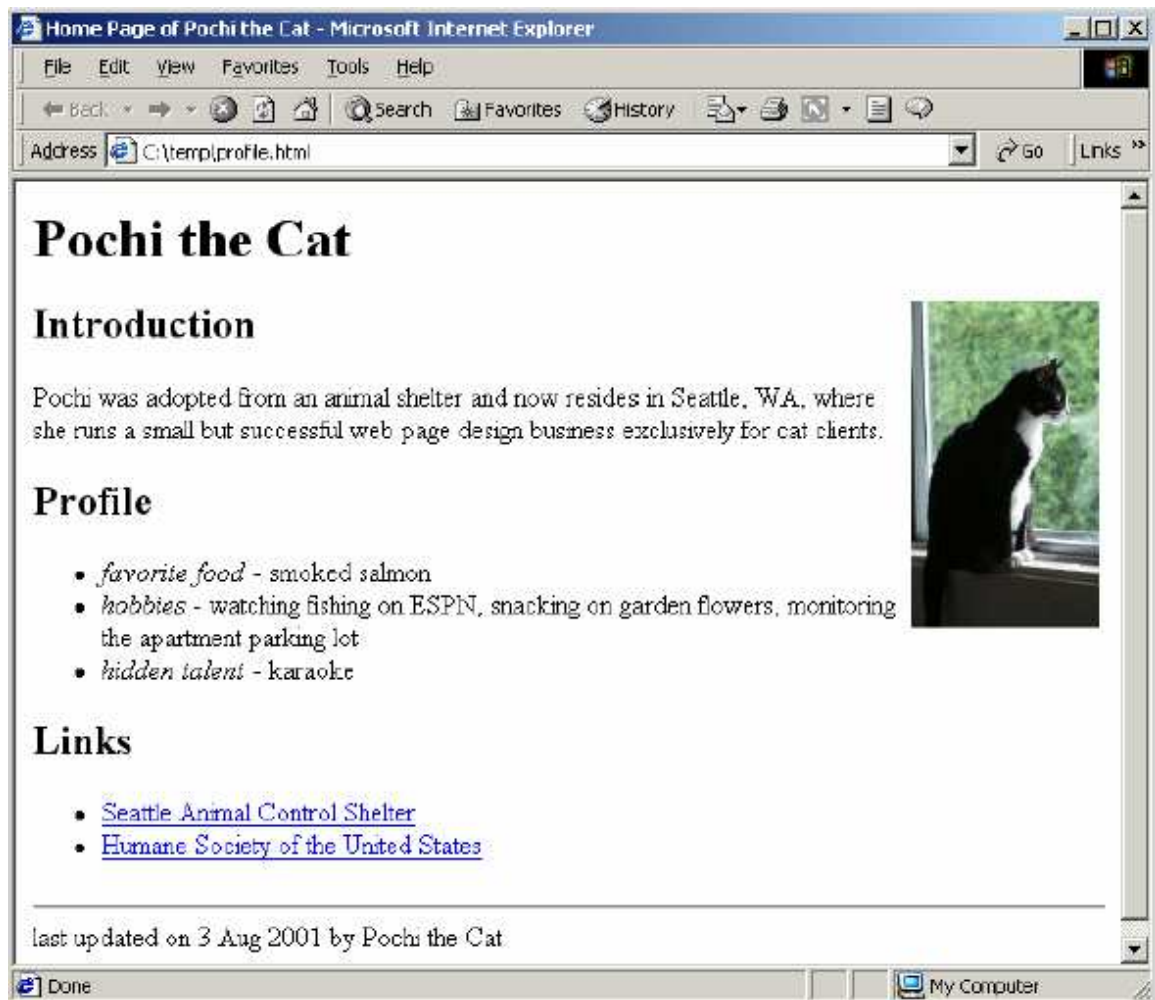
**Tags: web page building blocks.** You do not need to understand the whole HTML file now, but there are a number of important features to notice. In addition to the text to be displayed (including the page title, "a simple example page," which appears in the browser window's title bar), the source contains HTML *tags* placed carefully in the text. Tags are special markers that control text formatting (e.g., bold face, in a table, centered) or allow inclusion of non-text items such as images and links. Tags are enclosed in "pointy braces," i.e., the "less than" and "greater than" symbols `<` and `>`. These words (such as **body**, **h1**, **p**, and **em**) do not appear in the browser, but they affect how the text they surround is displayed.

If you look carefully at the tags in this example, you will see the two kinds of tags that exist in HTML. Some tags are "paired tags" and are used to mark sections of text, as opposed to just a specific point in the text. For example, text between a `<p>` tag and the next `</p>` tag is treated as a single paragraph. Since the first tag (without the slash) denotes the start of a paragraph and the other tag (with the slash) denotes the end of a paragraph, in general, we call the first a "start tag" and the other an "end tag." Other paired tags used above include **em** and **strong**, which are used to indicate that certain sections of text should be emphasized.

The other kind of tag is the "unpaired tag," which is generally used to denote the insertion of some non-text item in the web page. In the example above, the `<hr>` tag is used to insert a "horizontal rule," a thin horizontal bar stretching the full width of the window, in the page. There is no `</hr>` tag, because it does not make sense for a horizontal rule to "contain" text, the way it does for a heading or paragraph.



**Building your web page.** In this lab, starting from a very simple HTML file, you will gradually add tags and enhance a personal home page. (See the example below.)



example page

Although web pages are usually viewed from a web server via the network, your web browser can also view local web pages—pages stored on the computer you are using now. Until your web page is basically complete, you should view it locally.

### Default Web Pages

Default web pages, or home pages, are usually a way to indicate the very top level of your web site. However, every folder within your web site can also have a default page. At the UW, the name given to default web pages within a directory is **index.htm** or **index.html**. Other web servers have different names, such as *default.html* or *default.asp*, etc. So, if you want a particular page to come up automatically when someone enters the URL to your site:

**[http://students.washington.edu/your\\_uw\\_id/](http://students.washington.edu/your_uw_id/)**

then you must place an HTML document in your **public\_html** folder called **index.html**. When you created your account here at the UW, an index page was created

automatically for you. Until you add to that page, or replace it with an index page of your own, it will serve as your default home page. For every folder (directory) you create inside of your public\_html account, you have the option of placing an index document to serve as the default page when that folder is opened. The page that comes up when you enter this address:

**<http://www.washington.edu/home/departments/>**

is named **index.html**. You would see the same page if you were to type in the full path name:

**<http://www.washington.edu/home/departments/index.html>**

It acts as the default web page for the directory **departments**. If a web directory does not have a default web page, then you will see a listing of all folders and files in that directory. There is no **index.htm** or **index.html** page (or some equivalent) in that directory to act as the default.