IS 360 Lab 1  
Introduction to XHTML  
Ekedahl

In this lab, you will apply the following skills:

* Create an HTML 5 document in Visual Studio
* Create various tags and content in that document
* Create links between documents and to the Web
* Download and render images
* View Web pages in a Web browser
* Upload a simple site to the Web and test it

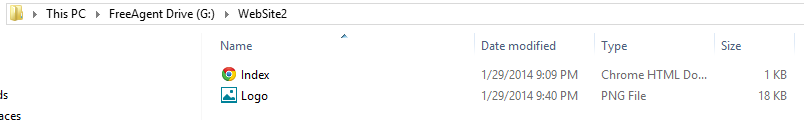
Note that you might not complete this lab in the class time allotted. You can certainly finish the lab outside of our class time. If you do finish the lab, feel free to work on homework assignments.

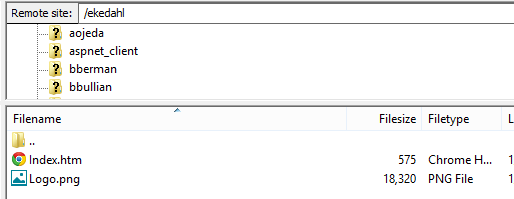
Use the following table as a simple tag reference.

|  |  |
| --- | --- |
| Tag name | Description |
| <title> | The document title that appears in the browser’s title bar. |
| <a> | Anchor (link) href contains the internal / external link or bookmark |
| <b> | Bold |
| <code> | Use a monospace courier (code) font. |
| <h1>, <h2>, … | Document headings |
| <hr /> | Horizontal rule (line) |
| <i> | Italics |
| <img> | Image src contains the internal / external link to the image alt contains the alternate text that will appear if the image cannot be displayed (also used to improve accessibility. |
| <pre> | Preformatted text |
| <ul> | Unordered list |
| <ol> | Ordered list |
| <strong> | Semantic emphasis tag |

**Creating a Web Site and HTML 5 Document:**

A Web site can be nothing more than one or many files that reside in a folder on a Web server. The folder name corresponds to the URL. As you develop a Web site, the folder structure that you implement on the local computer mirrors the structure of the Web site as it will appear on the Web server. For example, suppose that I have a Web site with two files named Index.html and Logo.png as follows:



I upload these files to the remote site using FileZilla as follows. In other words, the two files appear in the folder Ekedahl.   
  


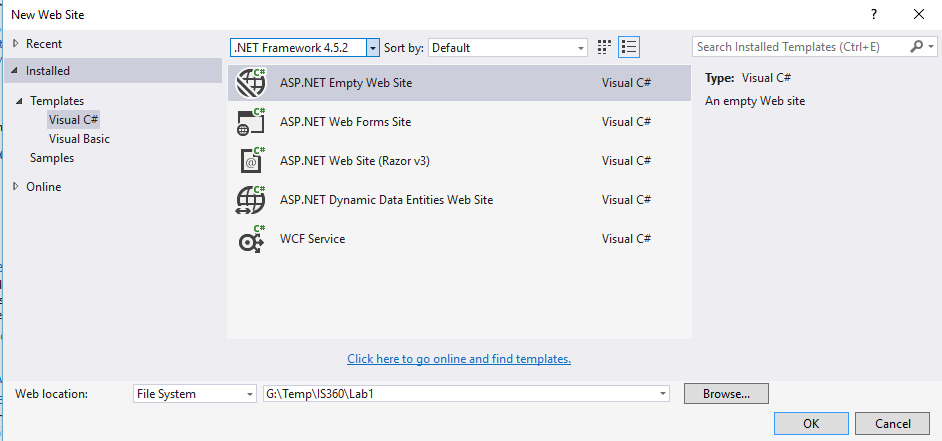
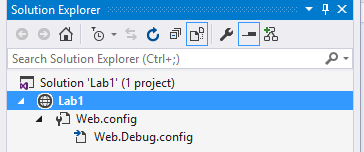
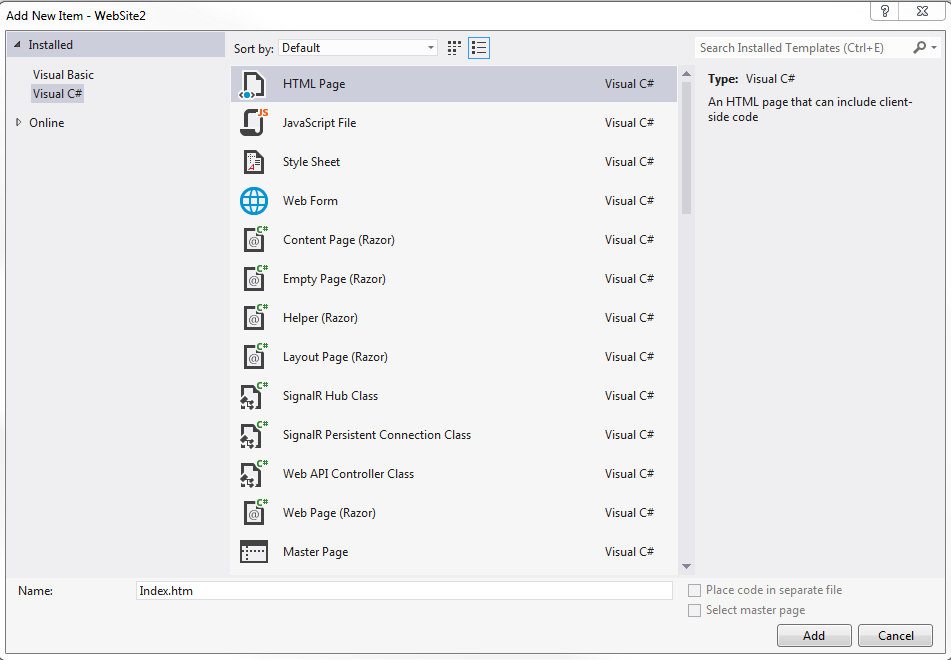
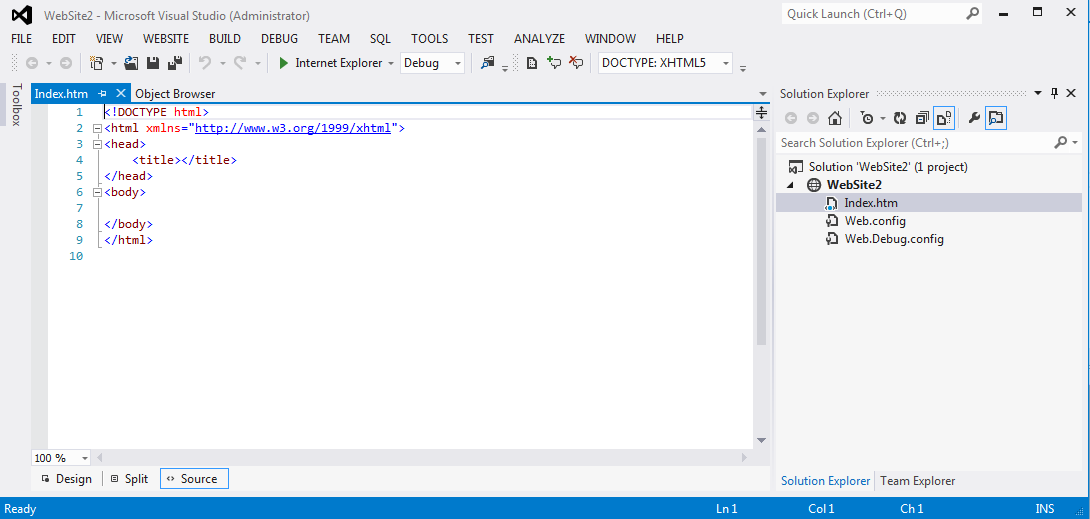
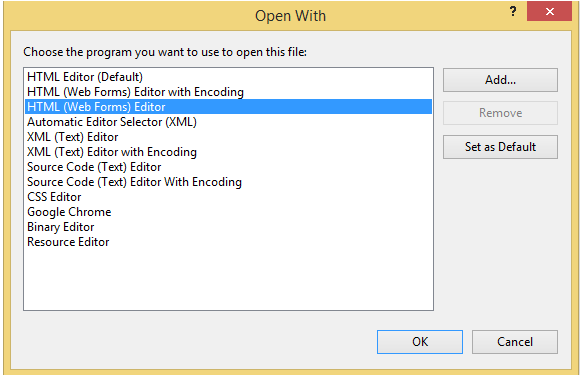
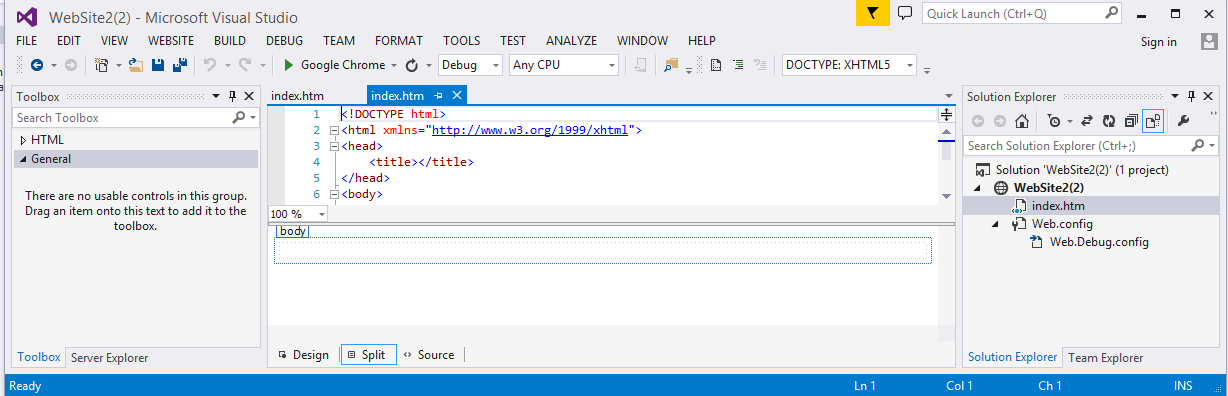
You reference a Web page with a URL made up of a domain, folder, and file.

Since the host is swww.coba.unr.edu, and the folder is Ekedahl, the URL would be <http://swww.coba.unr.edu/ekedahl>.

The page named index.htm is the default Web page, so in need not be specified explicitly in the url. However, you could fully qualify the name as in:

<http://swww.coba.unr.edu/ekedahl>/index.htm,

**To create a local Web site and add a Web page to it.**

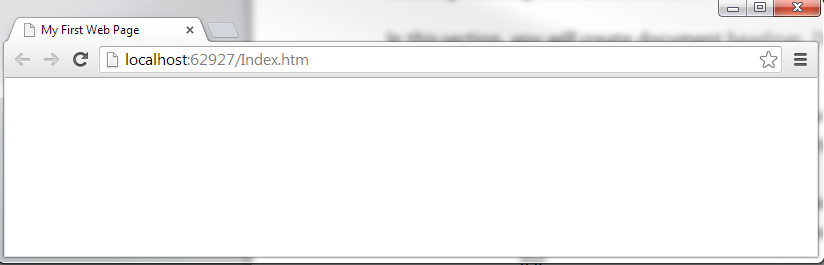
1. Start Visual Studio 2013.
2. Create a new Visual Studio Web site by clicking **File**, **New Web Site**. I suggest that you create the project on a USB drive or in your StudentHome directory (U: drive). Create an **ASP.NET Empty Web Site** and name the project as you see fit as shown in the following figures. Specify a directory that does not exist in the Choose Location  *Make sure that you create an empty Web site. If you create the other Web site types, a vast amount of code will be added to the site that is beyond the scope of the class.* ***Do not put spaces in the folder name. It will cause problems later! Do not put spaces in file names either. They make for very ugly URLS.***It does not matter whether you specify a Visual C# or Visual Basic template as we will not be creating any server code. Click **OK** to create the project.
3. The Solution Explorer will appear and displays the files that make up the solution. At this point, there are two files in the folder. The file web.config is the application configuration file. The file web.Debug.config is a debugging version of the web configuration file. These files are used by ASP.NET and are needed by simple HTML 5 pages. For now, you can ignore them.  
     
   **When you upload your Web site to the Web server, DO NOT upload the Web.config file. If you do, the Web server will consider this to be an ASP.NET application. However, the Web server is not configured to run your ASP.NET application. The page will not load and your application will not run.**  
     
   
4. Next, add an HTML 5 file to the project by clicking **Website**, **Add New Item**. The following dialog box will appear:  
     
   
5. Select **HTML Page**. Name the page **index.htm**. *Web pages (HTML documents) typically have a file extension of .htm or .html. Furthermore, index.htm typically designates a page as a default page for Web servers.* ***Again, don’t create Web pages having spaces in their file names****.* Click **Add** to create the page. It appears in the Visual Studio designer window as shown in the following figure:  
     
     
   **Remember that the page named index.html is magical. When a file named Index.htm or Index.html appears in a folder, the Web server will automatically render a file of this name when no file name is specified. On Windows servers, the case of the file is not relevant.**  
   The HTML 5 file is created and is made the active file in the Code Editor. The file also appears in the Solution Explorer. You can see that the file named index.htm is created. Visual Studio, using a template, copied some default text into the document for you.   
     
   **Visual Studio 2013, introduced a new default HTML Editor, which replaced the ASP.NET Web Forms editor. However, the new editor does not support a previewer. In the following steps, you will associate Visual Studio 2013 with the Web Forms Editor. You may or may not have to perform this step.**
6. In the Solution Explorer, right-click the index.htm file that you just created, and click **Open With** in the context menu. In the dialog that follows, click **HTML (Web Forms) Editor,** and then click **OK.** *If the document is open, a warning message will appear asking you to close it.*  
   Note the tabs along the bottom of the screen. The Design tab provides a WYSIWYG (What you see is what you get) interface to develop the page. The Split tabsplits the window so that you can see both the HTML 5 code and the WYSIWYG representation. The final tab, Source, shows only the source (HTML 5) code. The following figure shows the window Code Editor in Split mode.   
     
   Note that line numbers appear. If you want line numbers, click **Tools, Opensions**. Expand the **Text Editor** folder. Expand the **All Languages** folder. Select the **General** Node, and then check the **Line Numbers** box.  
     
     
     
   *Note that Visual Studio used a template to create this file. The DTD reference was automatically inserted to the HTML 5 DTD. Default <head>, <title>, and <body> tags were also added. Visual Studio uses templates for many different types of files.*
7. Click the **Split** tab so that you can see both the HTML 5 and rendered markup. The Design tab will be blank because you have not yet created any content.
8. Change the document title to **My First Web Page**. This title will appear in the browser’s title bar and selected tab. That is, edit the text for the <title> tag.   
     
     
     
   *Note that if you edit the content in the Source view, you will need to save the file for the changes to be synchronized in the Design view. In addition, note that the Visual Studio previewer is not perfect. It will not correctly render complex documents.*

**Running the Web Page**

Visual Studio allows you to test your Web pages from inside of Visual Studio itself. When a page is selected and run, it is rendered into the default browser that you have selected. Visual Studio will allow you to select from any browsers installed on your system.

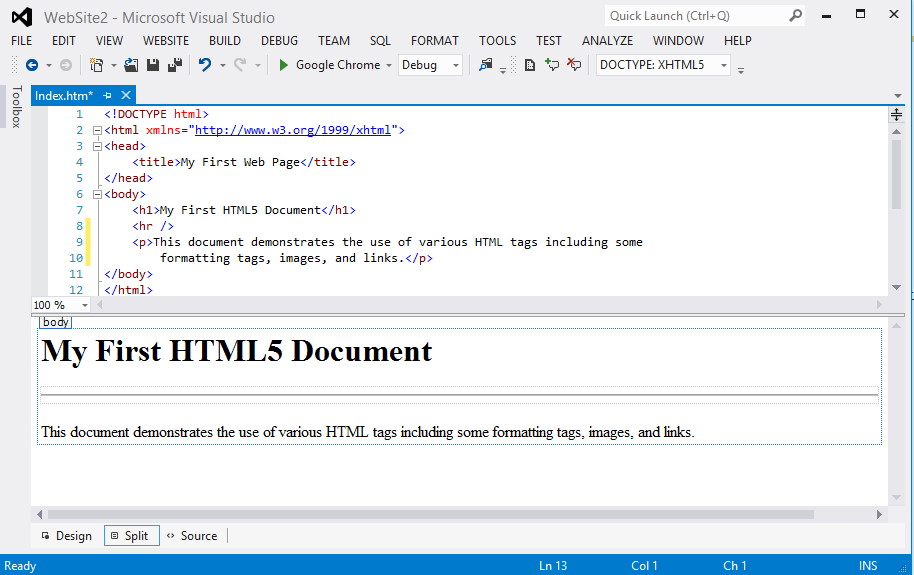
For reasons that will become clear when you begin to develop JavaScript code, I **strongly suggest** that you use Chrome as the default Browser. It has the best debugging tools and supports a great CSS development tool called FireBug.

There are a couple of ways to run the program, thereby displaying the Web page in a browser. Select the page that you want to view in the Solution Explorer, and then click the **Run** button. You can also press **F5**.   
  
*Make sure to select the page to run in the Solution Explorer.NET does not inherently know which page to run. Also, the following toolbar item allows you to select the default browser:*  


1. In the Solution Explorer, select the file named **Index.htm**. Press **F5** or click the **Run** button. The page appears in the selected Web browser (in this case Chrome)  
     
     
     
   If you look at the above URL, localhost indicates that the page is being run from local file. The value 62927 is the temporary port used to communicate with the local web server. Here, Visual Studio has a small built-in Web server which it uses to send your page(s) to a browser.
2. There is no content in the page because you have not created any content yet. Close the Browser.
3. Press the **End**  button to stop running the program. Visual Studio returns to design mode.

**Creating Headings and Basic Content:**

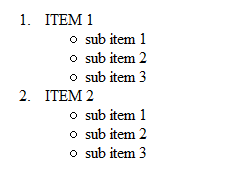
In this section, you will create document headings. Document headings are marked with the <h1>, <h2>, <h3> … tags.

1. In the <body> section of the Web page that you just created, create an <h1> tag so that the text **My First HTML5 Document** appears as the heading text. You can create the HTML in two ways.
   * ***Using Source View:*** *as you create this first tag, pay attention to the Intellisense technology*. *The ending tag is created automatically. Possible tags appear in the popup list.*
   * ***Using Design View:*** *the Formatting ToolBar will create headings and other formatting elements. To use the formatting toolbar, select the text to be formatted and click the desired formatting selections.*
2. Create a horizontal line just below the <h1> tag that you just created. Use the <hr /> tag. Remember that this is the shorthand notation to combine both a starting and ending tag.
3. Insert a paragraph <p> tag after the <h1> tag and <hr /> tags. Then enter the following content:  
     
   **This document demonstrates the use of various HTML tags including some formatting tags, images, and links.**  
     
   At this point, your document should resemble the following figure. Remember that it does not matter where you put the carriage returns as HTML performs whitespace normalization (removes extra tabs, spaces, carriage returns, etc…).  
     
   
4. Right-click somewhere in the region of the HTML 5 document and click **View in Browser** or press the **Run** button. The document should appear in the default browser (typically IE or Chrome in the College of Business labs).

**Basic Formatting:**

1. Create a level 2 heading <h2> at the end of the document. The text for the heading should be **Formatting**.
2. Next, create three paragraphs so that they appear formatted as follows: If you cannot remember the tag names, then refer to the table at the beginning of this document. You might also want to look at <http://www.w3schools.com/html/default.asp>  
     
   This text is **bold**.  
   This text is *italics*.   
   This text appears in a **monospace courier** font.  
   This text is both *bold and italic*.  
     
   Note that the second to the last paragraph uses a monospace courier font appearing in a bold typeface. Thus, you need to nest the <strong> and <code> tags.
3. Insert a copyright symbol in another paragraph using the following named entity: (&copy;) so that it appears as:  
     
   **© Michael V. Ekedahl**
4. Create an unordered list that will display the following content. Make sure that every other line has bold text. Again, you will need to use nested tags to do this.

* Item1
* **Item2**
* Item3
* **Item4**

1. Next, create a second list as an ordered (numbered) list.
2. Now, try to create a list within a list as follows. Here, you have unordered lists within an ordered list.  
     
   
3. Create the following paragraph after the one just you created so as to demonstrate the use of named entities:  
     
   <p>The <code>&lt;br /&gt;</code> tag is used to insert a horizontal rule</p>
4. Write the HTML 5 to render the following text:  
     
   The ampersand (&) character precedes the less-than (<) sign when referencing a **named entity**.

**Inserting Links:**

1. Create another level two heading after the content that you just created. The content for the heading should be **Links**.
2. Create an anchor <a> tag, as follows, so that a link to http://www.unr.edu is created. Again, remember that attributes consist of key/value pairs. In the following, the attribute named href has the value ="http://www.unr.edu". Attribute values are always quoted in HTML 5. Single quotes and double quotes are acceptable. The element content (what the user sees) is University of Nevada, Reno  
     
   <a href="http://www.unr.edu">University of Nevada, Reno</a>
3. Create a second anchor tag that will link to Google.com. Display the text Google.
4. Next, create a link to [www.business.unr.edu](http://www.business.unr.edu). The link should appear as an <h2> tag. Use text as you see fit.
5. Create a second Web page in the project named **Images.htm.** Click **Website**, **Add New Item** to do this. Create a level 1 heading <h1> on this page with the content **Image Demo**. Create a horizontal rule after the <h1> tag. In other words, both of the pages will have similar headings.
6. On the first page (index.htm), create a *relative link* to the images page that you just created as follows. This relative link tells the system that a page named Image.htm appears in the same folder as the page containing the link.  
     
   <a href="Images.htm">Images Page</a>
7. Preview the home page (index.htm) in the browser. Verify that all of the links that you created are working correctly.

**Images:**

In this exercise, you will download images and create links to them. There is much more to know about images than what is covered here. Web browsers can render many types of images. And those different types of images have different characteristics (compression, fidelity, vector or rasterized, and much more). In this lab, we will only render images and change their size.

* To download an image, right-click the image a browser window (IE), and then click **Save Picture As**.
* Using Chrome, right-click the image and click **Save Image As**. Save the picture to a folder of your choosing but make sure that you put image in the same folder.

1. Visit <http://www.unr.edu/marcom/university-identity/logos/downloading-logos>. From this Web page, download one of the Nevada logos as you see fit.  
     
     
     
   Download the image of your choice and save it to the project directory (the directory containing the HTML 5 files that you have created so for). Name the image **Logo.png**. Depending on the image that you chose, you might save a gif file instead of a .png file. Today’s browsers want process a wide range of image types.
2. On the Images.htm Web page, insert the following tag to reference the image. *Note that this is a relative reference*. *Think closely about this link. The image tag operates as the link and appears inside of the <a> tag.*  
     
   <a href="http://www.unr.edu"><img src="logo.png"   
    alt="University of Nevada, Reno" /></a>
3. Next, create the following image link using an absolute reference. *This file exists in my Web site*:  
     
   <img src="http://business.unr.edu/faculty/ekedahl/Heavenly.jpg" alt="Pricture of person skiing" />
4. Preview the page in a browser window to verify that both of the images are displayed. Check that the link to <http://www.unr.edu> is working.
5. Insert a second image of your choosing.
6. View the page in the browser to make sure that it is working correctly.

**Stretching and Resizing Images:**

You can change the size and shape of images by setting the width and height attributes. By default, values are expressed in pixels.

1. Change the image tag as follows to explicitly set the width and height as follows.

<a href="http://www.unr.edu"><img src="logo.png" width="300" height="300"

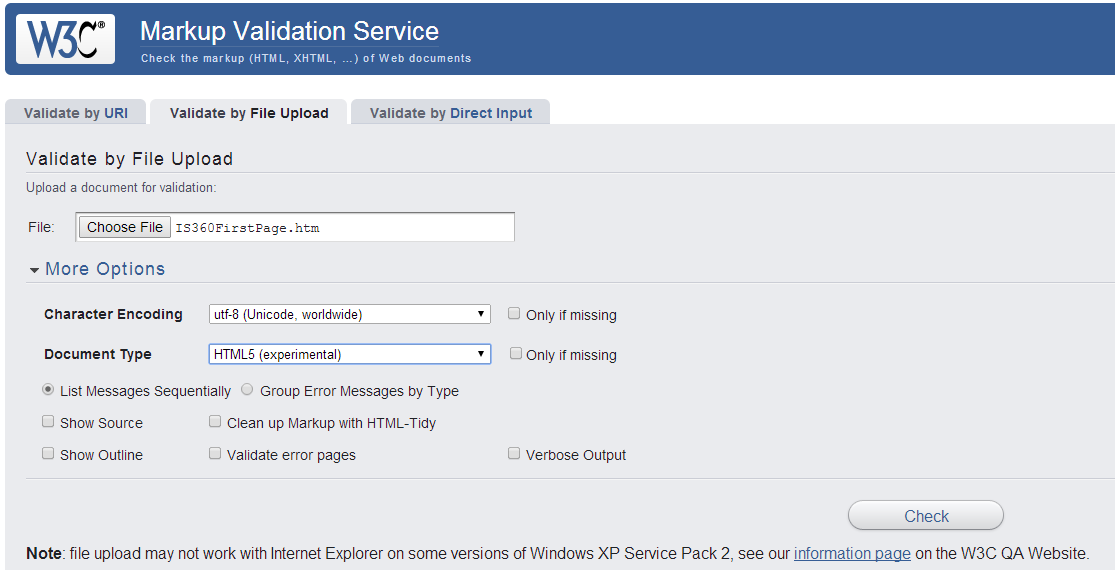
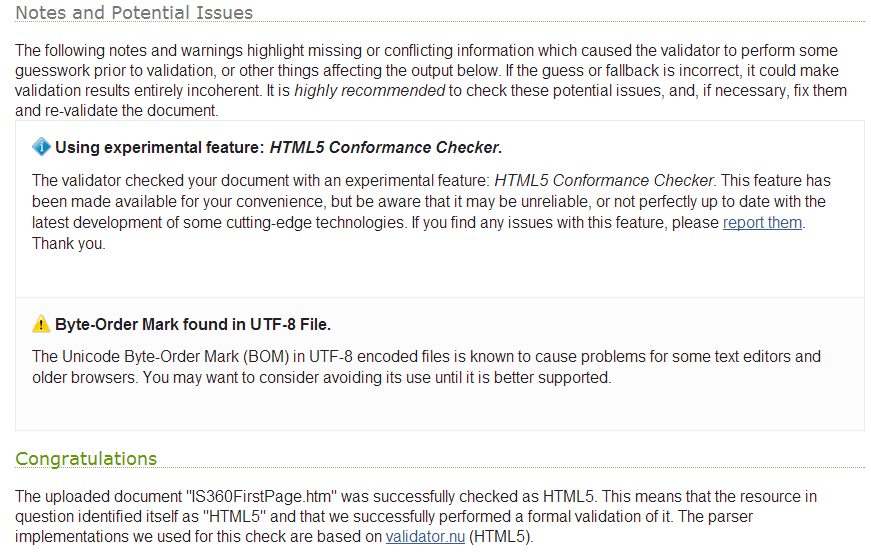
alt="University of Nevada, Reno" /></a>

1. Display the page in a Browser and note how the image is horizontally compressed.  
     
   
2. Close the browser and end the program.

**Creating Links Between Pages**

1. Add a third page to the web site.
2. On this page, create a relative link back to the first page. On the first page, create a link to this page. Use relative references for both links.
3. Test and make sure that both links work correctly.

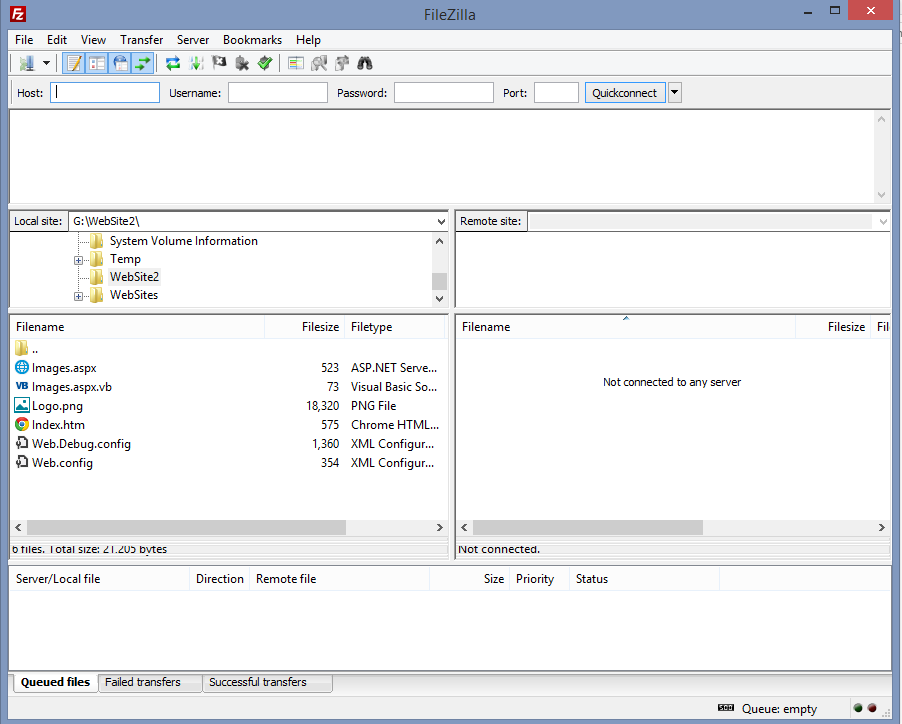
**In the following exercise, you will use the W3C validation service.**

1. Make sure that you have saved the Web pages that you created in the previous exercise.
2. In the browser of your choosing, visit the W3C Markup Validation Service at <http://validator.w3.org>.
3. Click the **Validate by File Upload** tab so that you can upload the Web page.
4. Click the **Browse** button to select the file to upload.
5. Click the **More Options** button to select the DTD. The selections you make should be similar to those in the following figure:  
     
   
6. Click **Check** to run the validator. The errors and warnings appear at the bottom of the form. The following two errors are acceptable.   
     
   
7. Review the errors.

**Posting the Site**

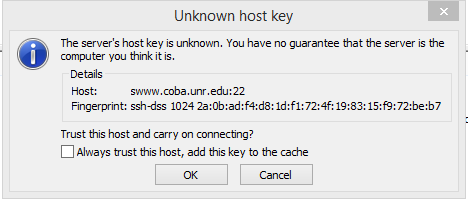
In this section, you will post the Web site that you have created to the Web. In this class, we will use FileZilla. It’s free and widely available. You can download at <http://sourceforge.net>.  
  
***Note that FileZilla is already installed in all of the COBA labs.***

1. **Start FileZilla**. Your screen should appear similar to the following one.



* The FTP requests and responses appear at the top of the screen.
* On the left side of the screen is the local file system.
* On the right side, is the remote file system. Nothing appears in the above figure because we are not connected.
* Status and log information appears in the lower window.

Next, you configure the host to which you will upload your web site. The host is named **swww.coba.unr.edu**.

1. Click **File**, **Site** **Manager**. The following dialog box appears.
2. Enter the host named **swww.coba.unr.edu**. Make sure to select the SFTP – SSH File Transfer Protocol. We run secure FTP. For the user, use your NetId. Our student Web server authenticates using your NetId. I suggest prompting for passwords! Click **Connect**. You will be prompted for a password if you did not save one. Because we don’t have a certificate, the following error message appears.  
   
3. Click **OK** to continue. The top screen panel should display the successful connection as follows. In addition, you should see and empty home directory in the right panel (server).  
     
   
4. Next you will upload the Web site. In the rightmost window, select the drive and folder that contains your local Web site.
5. Drag each individual file from the left window to the right window. As you do, the command and status windows are updated.
6. Finally, try to see if the Web site works. In the browser of your choosing, enter the following URL [http://swww.coba.unr.edu/*netid*](http://swww.coba.unr.edu/netid) where *netid* is your actual NetId. If you did everything correctly, your Web page should appear.