

Assignment 1:

Throughout this semester, our assignment will walk us through how to build a website for our resume. Some parts of the assignment will be explained in more details than others, so be prepared to either follow directions or complete tasks on your own. Although we may not use all pages we make right away, we will probably be using them at some point. Therefore, you should keep your files highly organized with intuitive names so you can locate them easily for later use.

Based on the readings and the information below, create a webpage that outlines your resume and has the following elements:

Note: this assignment may take a while, and if you want to stop in the middle and pick it up later, that may be a good idea. Directions on how to save your work can be found starting on the top of page 4. This area also describes how to preview your webpage, which may also be helpful to you while creating the page itself, so you may want to read that section first or as you create your page.

Creating the page:

- Open your text editor program (Notepad++ for PC users and textedit or textwrangler for a Mac) and begin typing your code (see next steps).
 - We use a text editor and not a program like Microsoft Word because Word will autocorrect some tags and formatting into what it thinks we are trying to spell and capitalize things that should not be capitalized.
- You should base the structure of your document off of that explained in the reading: <http://reference.sitepoint.com/html/page-structure>
 - We use XHTML transitional in this course, so make sure to choose the correct doctype. If you read through the examples the XHTML one will be obvious.
- Within the Body, create a **heading** for your name. Headings are a great way to highlight different text throughout your page.
 - For information on the heading tag, click here: http://www.w3schools.com/tags/tag_hn.asp
- Under that, write a brief **paragraph** of what this website is about. Paragraphs are alright for simply putting some text on a page.
 - For information on the paragraph tag, click here http://www.w3schools.com/tags/tag_p.asp
- Next, create a **table** that outlines your professional experiences. Create a column for place (where you can put the name of the employer or school), degrees earned or job title, and another column for the years you were there. Find a way to use either

colspan or *rowspan*, which will have a **table header in it** that *spans* all of the columns or rows and put some text in it to let the user know what the table is (i.e. résumé highlights).

- Directions for tables: http://www.w3schools.com/html/html_tables.asp
- http://www.w3schools.com/tags/att_td_colspan.asp, this link was found by searching in Google “HTML Column Span” and this link: http://www.w3schools.com/tags/att_td_rowspan.asp was found by searching for “row span.” You will only need to use one of these, it is up to you to decide which one.
- A way to look at spanning things within the table is that we are adding more information to a tag or in this case, modifying what the default output of the table tag is. To modify the default output or settings of a tag (as we will do a lot more for things beyond spanning), we use an attribute.
- Using the attribute that will add a border to the table, add a border. We can find out how to do this by performing a Google search on “HTML Table Border”
- Underneath that, create another **heading** and then a **definition list** of your education, where the term is the degree and the definition is a hyperlink to the school’s website that is shown as the school’s name.
 - For list information, see: http://www.w3schools.com/html/html_lists.asp
 - Although all hyperlinks work the same way and redirect a user to a location on the web, there are two ways of creating links, both use the **anchor** tag. They allow us to link to other documents on the web.
 - One way is by using an absolute link, or absolute path. An absolute path is the full URL to a website (i.e. <http://www.albany.edu/myualbany.html>). This link will always work and link to the page’s specified location on the internet. So regardless of the location of the page that the absolute link is on, it will always work, but it is lengthy to type out.
 - The second way is to use a relative path. This path links to another page based on the page’s (that is being linked to) relative location to the page that the relative link is on. For example, if you have two HTML documents in the same folder, the relative path link is just the name of the html (in the above example, if SUNY Albany had their homepage in the same folder as the myualbany page, the relative path would just be [myualbany.html](#)). Another example, if SUNY

Albany had their homepage in a folder named “home” and within that home folder there was another folder named “students” and within that “students” folder was the myualbany.html, the relative path to the myualbany page from the SUNY Albany’s homepage would be “students/myualbany.edu”.

- Using the target attribute, ensure all pages open in a new window or tab:
http://www.w3schools.com/tags/att_a_target.asp
- We can use a special type of a relative link even though we only have one page, by using a special type of link: the intrapage link. The links that we created above are between different pages, or interpage links. The intrapage link links to a specific part within the same page.
 - To exemplify how this works, add a lot of space (enough space so you need to scroll down to see your list) between the definition list and your table. Notice that you cannot just keep hitting return, as HTML does not recognize whitespace. Instead we can add a bunch of paragraph tags that have no content within them and separate the paragraph tags by the break tag (i.e. paragraph break paragraph break, etc.). The break tag is
. Note that it looks a little different than our other tags because it opens and closes in the same tag. This is called a self closing tag.
 - Follow these directions,
[http://www.brainbell.com/tutorials/HTML and CSS/Intrapage Link ing.htm](http://www.brainbell.com/tutorials/HTML_and_CSS/Intrapage_Link_ing.htm) to put a link at the top of your page that links to the table at the bottom.
 - Note that although the directions are for intrapage, you can apply the same methods to link from one of your pages to a specific location on another page.

QUESTION 1: What does this intrapage link do after you add it and click it?

- Lastly, create another **heading** and then an **unordered list** or **ordered list** of your skills.
 - There are three types of lists: Ordered, unordered, and definition.

Question 2: which type of list did you choose and why?

Saving your work

- It is simple to save your work. Click file and save as. Select to save as type “all types.” After this is done, save your file by typing the name that you want to save it as followed by .html.
 - Everything after the last “.” in a file name is the extension (see http://en.wikipedia.org/wiki/Filename_extension if you want to learn more). So in this case .html is the file extension which specifies the type of the file, or in this case an HTML file. You do not always need to manually enter in a file extension for all files. Many programs will ask what type of file you want to save it as and then automatically put the file extension in after the name.
- You can now preview your page by navigating to where you saved it and double clicking it to open it, just as you would any other file. If you closed out of Notepad++ and want to edit more, right click the file and choose edit with Notepad++

Validating your work

- It is good practice to ensure that your code validates according to the W3C standards. Visit the W3C validator (Google W3C validator and it will be the first result) to make sure your code validates
- Copy and paste your code into the dialog in the “validate by direct input” tab (on the top), then click check
- If there are any errors (symbolized by a red circle with an x), read through what they are. This is a very useful tool for you. The errors will list the line number that the error occurs and what the error is. If you do not know what the error is after reading the description of the error, copy and paste the error message into a Google search and see what others say about it.
 - Sometimes the validator will give an error for a certain line even though that line is correct. In this case, it is typically the line right above it that has an issue, so if you do not see the error right away on the line that the validator says, check the lines above and below.
 - You may also have many warnings. You do not need to fix these for this class, but can do so if you would like.
- Fix all of the errors so that your page validates, resave the page.
 - Sometimes if your page is not doing what you want it to do, there is an error in the code and if that is fixed then it will do what you want it to do. So if you are

ever having problems, the first thing that you should do is get your page to validate. Sometimes fixing one error will actually fix multiple errors (i.e. your page has 22 errors and after fixing just one of them you only have 10 errors left) but sometimes fixing one error also allows the validator to pick up on other errors that it did not see before and the number will go up. My personal preference when using the validator is to work down from the top by fixing the error that appears first in the code, revalidating, and repeating as necessary. As I get more comfortable with the code, I may fix several or all errors before revalidating.

Uploading to the World Wide Web – Creating a public folder

- We first need to create a folder that has permissions that allow the contents of the folder to be viewed on the web. **This will be the only time that you need to create a folder for your HTML this way.**
 - If you have already created a public HTML for another class, you can skip these steps. If you are not sure if you have, you can try the steps and it will say that the public HTML already exists when you attempt to create it (or read below to see how you know if one is successfully created), at which point you can exit the program.
- Open up putty itsunix.albany.edu (found under SSH on the school computers or virtual commons)
 - This software uses Unix commands to do things like create directories, move and delete files, etc.
- Login using your net id and then enter your password
 - When entering your password, it appears as though nothing is being typed. It is actually typing in your password, it is just not displaying it so the person sitting next to you cannot see your password
- Press enter
- Type startweb, press enter
 - Unix may prompt you with a question after this step. If it does, read it and enter what it tells you to in order to proceed
- Type setweb, press enter
 - This may take a few minutes to process, so be patient
- Type exit and press enter

- Go to your S drive and make sure that a folder is created that is named public_html. This is the only folder in your S drive that has the permissions which allow the web to view its contents, so all of your HTML documents will be transferred (not saved directly to though) to this folder

Uploading to the World Wide Web – Transferring your HTML to the public_html and setting permissions

- We will be using FileZilla to transfer the HTML file that you created above to the public_html folder (see below for screenshot)
 - FileZilla is an open source FTP. It allows for us to upload files to and from our public folder and also set permissions to those files. It allows us to do all of this remotely.
- Log into Filezilla (under SSH also, or you can download the program for free for either a Mac or PC)
- Enter the host (itsunix.albany.edu), your username (netid), your password, and use port 22
- The screenshot below shows the layout of FileZilla for the local side. The local side refers to the local storage, or the storage that you physically have right in front of you (flash drive, hard drive, etc.). The remote side is on the right and follows the same layout and refers to some remote storage, in this case the UAlbany server where your S drive is.
- By clicking through the local directory tree on the local side, navigate just as you would on a PC to the location of where you saved your HTML. Click once on the folder where the public HTML is saved and all of the contents of that folder should appear below.
- Then do the same on the remote side, navigating to your public_html.
- Then, from the contents of the local directory, drag and drop your HTML to the contents of the remote directory.
- Right click the html file in the remote side and choose file permissions, set them to 755 (see below for screenshot)
 - These file permissions give different users different access to the file. By choosing 755, we do not allow anyone except the owner (yourself) to write to (edit) the file but still allow all other users to read (view) and execute (run) it.
 - This is the reason why we need to use FileZilla to transfer files over to the public_html and not save them directly to the public_html. It ensures that

people have the access that they need to the files, therefore you will do this every time you want to put anything on the web (html, images, audio, etc.)

- Also right click the public_html folder itself and set the file permissions for that to 711 and then try to access the folder on the web. This removes read access for most users. The url to your public folder is www.albany.edu/~username. Make note of what happens
- Now right click the public_html and set the file permissions to 755. Try to access the folder again

QUESTION 3: What happens differently when you try to access your public_html from the web when the file permissions for the folder at 711 vs 755?

- Leave the public_html folder's permissions set to 755
- Check to make sure your webpage works properly on the web by trying to use the web to view it. The URL to that html is:
www.albany.edu/~username/nameofhtml.html
- In Blackboard, submit ONLY your hyperlink and the answers to the questions.

***note, if you are having trouble with this assignment due to a lack of professional experience please talk with me and we can find a workaround. I expect at least 2 employers, 2 different educational institutions, and two skills.

Example page (note your definition lost will be separated from the table):

Norman Gervais

Please see the description below outlining my resume

Employment History		
Employer	Job title	Dates
Employer #1	My title	month year-month year
Employer #2	My title	month year-month year
Employer #3	My title	month year-present

Education

High school

Location

College 1

College name

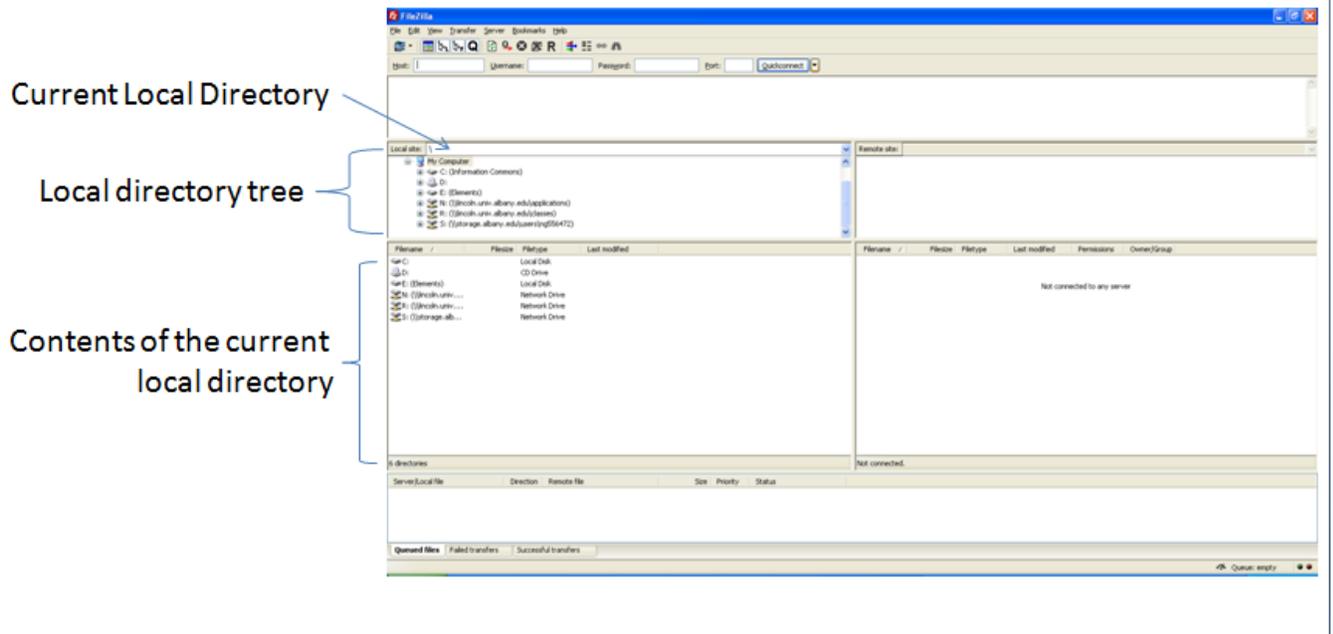
College 2

College 2 name

Skills

- Critical thinking
- Statistical analysis
- Web design and creation
- Image editing

Screenshot of FileZilla with labels:



Screenshot of permissions in FileZilla:

