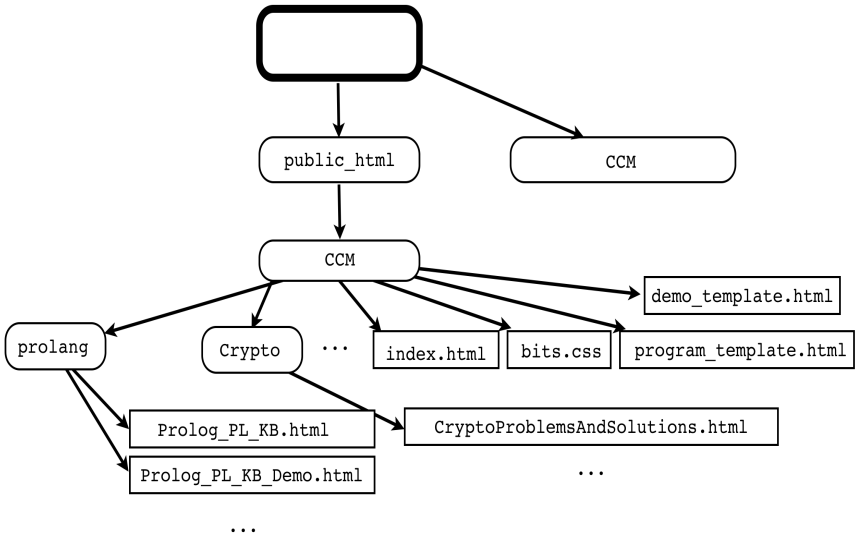
Optional CCM Web Site Lab
Establishing Your Work Portfolio

About this Lab

As I mentioned in the CCM Web Site Assignment document, you are welcome to use any suitable Web development software that you like for the Web assignment. Since I am getting some questions about this assignment, I thought I would provide a generic response to "How might I build a site?" in the form of this optional Lab that you are most welcome to work through. It is premised on the notion that you will be working on the CS Department machines, which are running the Ubuntu operating system. **Please note that I am not requiring that anyone do this lab!** I merely offer it as an option should you, for whatever reason, like to adopt the approach to developing your Web site that is described herein. This lab is adapted from one I give to my CS1 students each year. It seems that nearly all of the roughly 30 students who enroll in the course each year successfully used the lab from which this one is derived to establish a basic Web site.

This lab is designed to help you to commence the activity of building a Web site dedicated to presenting your work in this course. This lab affords you the opportunity to engage in processes of directory creation, downloading files, placing files in appropriate directories, edited files, and viewing files in a browser. It will also point you to a document that provides you with what you need to know to make your site public for all the world to see.

Suggested Directory Structure



Learning Outcomes

As you work through this Lab, you will commence to build a Web site by:

1. Creating a directory structure for your work site.
2. Downloading files to particular locations in your newly created directory structure.
3. Regularly engaging in a constructionist process of:
    1. Editing **`.html`** files using the **emacs** text editor.
    2. Viewing **`.html`** files in a browser
    3. Creating and saving **`.html`** "program" and "demo" files to particular locations in your in your directory structure.

Tasks

1. **Get ready to do some work.**

    1. Log on to a machine in the Computer Science Lab.
    2. Open a **Terminal** window for Web development. I found one in the column of icons on the left of my screen, and opened it by double clicking on it.
    3. Open the **emacs** text editor. Shrink it.
    4. Open a good **browser**. Shrink it.

2. **Add some folders to your directory structure.**

    1. In the terminal widow, make sure that you are in your home directory by issuing two commands:
        1. Issue the most basic version of the "change directory" command by typing **`cd`** followed by the enter key.
        2. Issue the "print working directory" command by typing **`pwd`** followed by the enter key. Observe.
    2. Create a folder for your Web work. This folder must have a particular name in order for the CS servers to, eventually, make it available on the Web.
        1. Issue the "make directory" command to create the required "public html" folder within your home directory by mindfully thinking underscore (rather than dash) and typing the **`mkdir public_html`** command. If the directory already exists, no worries, just keep going.
        2. Issue the "list files" command by typing **ls** (that is "elle" followed by "esss") followed by the enter key. Observe that the folder exists.
    3. Create a folder within your public html area for your course work site. Be careful to name it as specified.
        1. Change directories so that your are working within your **`public_html`** directory by typing the **`cd public_html`** command.
        2. Just to assure that your are where you want to be, issue the command to print your working directory, **`pwd`**, and observe.
        3. Issue the "make directory" command to create the folder within which you will build your work site for this course by typing the **`mkdir CCM`** command.
        4. Issue the "list files" command by typing **`ls`** (that is "elle" followed by "esss") followed by the enter key. Observe that your folder was created.
    4. Have you learned how to enter a directory, check your location, make a new directory within the current directory, verify that you have accomplished the task? Let's see! Here is what you should do next: Create a directory called **`proglang`** within the **`CCM`** directory. (Just be sure to work by analogy with the four steps in the immediately preceding task.)
    5. Create a directory called **`Crypto`** within the **`CCM`** directory. (Just be sure to work by analogy with the four steps in the immediately preceding task.)

3. **Download four files.**

    1. Expand the browser and find your way to the course page at:

        `www.cs.oswego.edu/~blue/oswego/green/2017/CCMCourseSite.html`

    2. Download the following files:

        1. Download **`index.html`** to the **`CCM`** directory.
        2. Check, in the terminal window, to be sure that you were successful by listing the files in the **`CCM`** directory and observing.
        3. Download **`bits.css`** to the **`CCM`** directory.
        4. Check, in the terminal window, to be sure that you were successful by listing the files in the **`CCM`** directory and observing.
        5. Download **`program_template.html`** to the **`CCM`** directory.
        6. Check, in the terminal window, to be sure that you were successful by listing the files in the **`CCM`** directory and observing.
        7. Download **`demo_template.html`** to the **`CCM`** directory.
        8. Check, in the terminal window, to be sure that you were successful by listing the files in the **`CCM`** directory and observing.

4. **Load the `index.html` file of the `CCM` directory into a browser. This file will serve as your main work site page.**

    1. One way to do this is to simply open the **Files** program among the icons, navigate to the appropriate **`index.html`** file icon, and then double click on it. These acts should bring the file up in your browser.
    2. Observe (do some clicking) that the "review" links do not work.
    3. Observe that the links in final block of text/hypertext do work.

5. **Edit the contents of the `index.html` file so that you are represented in the identifying information.**

    1. I prefer to use the **emacs** text editor to perform text editing. Begin by activating **emacs**. You will find it among the application icons.
    2. Open the **`index.html`** file. There are various ways to do this. My favorite is to use my hands and type **`CONTROL-x`** then **`CONTROL-f`** and then interact (type characters to identify the file name) in the minibuffer.
    3. Change the text, replacing my name with with your name.
    4. Save the file.
    5. Reload the file in the browser and take a look.
    6. Change the text, getting rid of **Sample**.
    7. Save the file.
    8. Reload the file in the browser and take a look.
    9. Shrink the **emacs** editor.

6. **Operationalize your work site links for the first of your Prolog assignments.**

    1. I like to split my screen with **`CONTROL-x`** then **`2`**.
    2. Operationalize the source file link:
        1. Activate **emacs** and then open load your Prolog program (the 9 fact KB) into the top buffer. (I like to think **`CONTROL-x`** then **`CONTROL-f`**)
        2. Load **`program_template.html`** into the bottom **emacs** buffer. (I like to think **`CONTROL-x`** then **`CONTROL-f`**!)
        3. Write the **`program_template.html`** buffer to a file named **`Prolog_PL_KB.html`** so that your template won't get destroyed and your work site will properly reference the source program.
        4. Copy the Prolog program text from the one buffer so that it replaces the **`#TEXT-COPIED-FROM-PROGRAM-FILE-GOES-HERE#`** tag in the other buffer.
        5. Appropriately replace the **`#IDENTIFIER-GOES-HERE#`** tag in the **`Prolog_PL_KB.html`** file.
        6. Save the **`Prolog_PL_KB.html`** file. (I like **`CONTROL-x`** then **`CONTROL-s`**.)
        7. From your Web browser, check to see that the link to the source program is working properly.
    3. Operationalize the link to the demo:
        1. Run your first Prolog program, however you do that. (I would do it in an **emacs** shell in the top buffer. **`Meta-x`** then **`shell`** in the minibuffer.)
        2. Load **`demo_template.html`** into the bottom **emacs** buffer.
        3. Write the **`demo_template.html`** buffer to a file named **`Prolog_PL_KB_Demo.html`** so that your template won't get destroyed and your work site will properly reference the demo file.
        4. Copy the Prolog program output so that it replaces the **`#TEXT-COPIED-FROM-PROGRAM-EXECUTION-GOES-HERE#`** tag in the **`Prolog_PL_KB_Demo.html`** buffer.
        5. Appropriately replace the **`#IDENTIFIER-GOES-HERE#`** tag in the **`Prolog_PL_KB_Demo.html`** file.
        6. Save the **`Prolog_PL_KB_Demo.html`** file. (I like **`CONTROL-x`** then **`CONTROL-s`**.)
        7. From your Web browser, check to see that the link to the demo page for the first Prolog program is working properly.

7. **Working by analogy with the previous task, operationalize your work site links for the second Prolog programming assignment files.**

    1. Do the deed for the source program.
    2. Do the deed for each of the two demos

8. **Add the Crypto problem solutions to your work site**

    1. Place an appropriately typed file (pdf or html) containing your solutions in the **`Crypto`** directory within your **`html`** work area.
    2. Assure that it loads properly into your browser when you click on the relevant link.

9. **Add the requisite links to external sites.**

    1. Find the sites.
    2. Add the links.

10. **Make your work site available to the world!**

    1. Making your site available to the world will involve changing some permissions on files and directories. You will find information on how to do this (see Step 2.2), together with other information that will be useful to you should you end up doing your Web work on your own machine, embedded in the document at the following Web address:

       `http://www.cs.oswego.edu/~agraci2/wt/`

    2. After you have done what needs to be done, anyone on the Web should be able to view your work site for this course. Where will it be? Try the following, after instantiating the question mark (?) with your usercode:

       `http://www.cs.oswego.edu/~?/CS2/index.html`