1. Creating and using a local repository

**If you have not installed the git tools or GitHub Desktop skip this exercise. Just create a folder on your desktop called ITM352\_F20\_repo**

* Go to GitHub.com, sign in using your github account and select your hello-world repo (that you created from the [Hello World GitHub guide](https://guides.github.com/activities/hello-world/)). Go to Settings and change the repo name to ITM352\_F20\_repo. Change the README.md if you wish.
* Clone this repo and open it in VS Code. You can clone it in GitHub or use GitHub desktop (or any other way you wish that works).

*What is the URL for your GitHub ITM352\_F20\_repo?*

<https://github.com/alvinalmira/ITM352_F20_repo>

*What is the path to the local ITM352\_F20\_repo?*

/Users/alvinalmira/Documents/GitHub

2. Using VS Code to create a web page

VS Code is a file editor designed to help build applications (code). Web applications are built from files with instructions that the server and browser process. Let try making a web page and viewing it in a browser.

* In the repo folder, create a new folder inside called “Lab1”, create file in this folder <your Last\_First name>\_hello.html. Edit this file “Hello from <your first name>!”. **Don’t forget to save this file after you make changes!**
* Open a browser, navigate to the file you created or drag and drop it into the browser from a file explorer (you may find it easier to right-click on the file in VS Code and Open in Finder/Explorer).

*Explain here why you see text in the browser window. Is this a web page?*

* I see text in my browser because the file is saved locally on my mac; it is web page but it is only accessible through my computer.
* Edit the file to add your **last name** and enclose your full name between <b> tags and then enclose all the text in between <html> tags. Save a refresh your browser.

*Explain here why you do not see the code for the tags in the browser window:*

* Since the file is saved as an HTML, the browser will render what is in the document in HTML format.
* In VS Code file manager right-click, in terminal, or in explorer/finder, change the file extension to .txt. In your browser window change the .html in the filename in the address bar to .txt and reload the window.

*Explain here why you****do****see the code for the tags in the browser window:*

* The file is saved and read as a .txt file instead of a .html file, so the browser is going to read it as a .txt file.
* In VS Code, in front of Hello type <i

*Explain why you do not get the Intellisense help for HTML anymore:*

* The intellisense is not popping up because the file is in a format that doesn’t read code.
* Change the file extension back to .html. Find an HTML tag that will italicize text. Use this tag to italicize your entire greeting. Save and reload your browser window.

*Explain why you****do****get the Intellisense help for HTML now and why you****do not****see the html in the browser window:*

- I get the intellisense help because the file is read and saved as a .html rather a “.txt” file. The reason I cannot see the html in my browser window because the file name in the address bar ends in “.txt” instead of “.html”.

3. Installing and using http-server as a local web server

For convenience and speed we will test applications on our own machine before deploying to a server. Applications that need only serve static pages can use a simple web server that accesses the static pages (documents) from a single directory (the document root). The http-server package will do this for us without any configuration or coding.

* Open a terminal in VS Code, check that you have node.js node --version
* Add the http-server package npm install http-server -g *Note for Mac users: you may need to add sudo to the front of this to override file permissions.*
* Check the run options with http-server --help
* Start the local http-server and use the current directory as the document root by typing http-server .
* Open a browser to http://localhost:8080/<your Last\_First name>\_hello.html and verify it is served rather than loaded directly into the browser (you should see http://localhost:8080/<your Last\_First name>\_hello.html rather than the file path)

**NOTE:** If the server fails to run and you get a port already in use error you may have another process using port 8080. Try changing the port to something else like 8081 and try running again. \*\*

*Go to the terminal window in VS Code and copy and paste here the output after you started http-server. Explain what this output is:*

alvinalmira@chipmunk ITM352\_F20\_repo % http-server

Starting up http-server, serving ./

Available on:

http://127.0.0.1:8080

<http://192.168.231.43:8080>

It’s going to look in the current directory for a certain file.

*Explain how the page in the browser window was loaded. Why is the URL path not the same as the filepath for the file in your Lab1 directory?*

* The page was loaded through a localhost rather loaded as a local file. The file path is not the same because it’s being read through a localhost server.
* Hit ctrl-c to stop http-server. Reload or refresh the page in your browser.

*Explain why you do not see the page anymore:*

* The page cannot be read anymore because the server went offline.

*Explain here how this is different than what you did previously. Is your file now a web page?*

- This process is different because the file was read through a localhost server rather than read from my local disk.

4. Local vs Global web servers

There really is no difference between the web server such as you installed on your laptop for class and any other web server on the internet. It’s really a matter of accessibility. A local web server is accessed though the URL http://localhost where localhost is always set to the IP address 10.0.0.1 or 127.0.0.1 (which is also known as the “local loopback”). Anytime you try to connect to localhost you will always be connecting to your own machine. So if you set your web server’s address to be localhost it then can only be accessed from your machine regardless if it is connected to the internet (or any network). A global web server simply has it’s address set to some globally accessible IP address (and possibly an internet registered domain name).

You should always develop your web applications locally and then “publish” them–after careful testing–to a global web server (that is if you want the outside world to be able to access it). For our class we have made available to you the global web server <http://itm-vm.shidler.hawaii.edu/itm352student> and this is where you should test all your applications if they are intended to be used non-locally.

Testing the class web server as your global Web server

Locate the file you created to test your local web server from exercise 3 above.

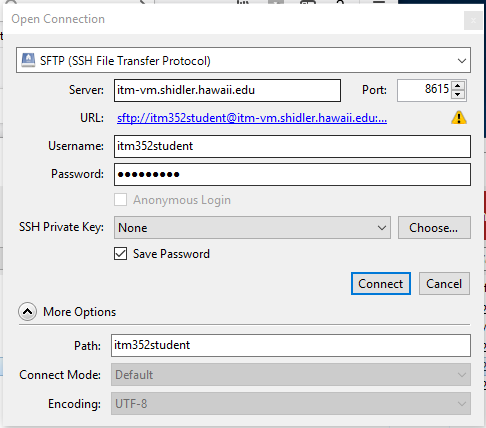
1. Use SFTP or SSH or a file transfer tool such as [CyberDuck](https://cyberduck.io/?l=en) to upload it to the ITM 352 student class webserver at itm-vm.shidler.hawaii.edu/itm352student/. The "htdocs" or document root folder to store your web files on the class server is actually called `public\_html`, but when you log in it will take you directly to this location so you won't see it. You cannot write files to this directory so you will need to go to `Section\_Port` or `Section\_Kazman` depending on which class section you are in. Make a note of this as you will need to add this to URL to access your file.

* username is <ask instructor> with password <ask instructor> you must use port 8615 You can try this link or copy and paste it into the Server textbox <sftp://itm352student@itm-vm.shidler.hawaii.edu:8615/>

See <http://www.hawaii.edu/askus/692> for more information on SSH/sFTP

1. Access your web page by typing in http://itm-vm.shidler.hawaii.edu/itm352student/<your class section>/<your Last\_First name>\_hello.html For example http://itm-vm.shidler.hawaii.edu/itm352student/Section\_Port/Port\_hello.htm

If you are using Cyberduck here is what your open connection should look like this:



Answer the following questions:

*How do you make websites?*

You make a website my creating and .html text file, type opening and closing html tags, type a few words, and save the file. To open it, you would have to drag it into your browser, or right-click it and open with your browser. From this point on, you can add your header, body, paragraph, and other tags to make a full-blown website.

*What’s the difference between a local and global webserver?*

A local webserver is a server only read by the current device that is broadcasting the server. A global webserver is a server that can be reached by anyone who has the IP.

*How do webservers, local or global, work with VS Code?*

VS Code works with webservers by being the editing tool for text and stylesheets. Webservers, local or global, show us the output of our text on the document from VS Code, or any text editor. If changes are need on a webserver’s page, we can use VS Code to carry out the necessary edits.