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_S25_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). Go to Settings and change the repo name to ITM352_S25_repo. Change the README.md if you wish. Also in the Settings page, go to the "Danger Zone" and change the repo "Make private" if it is not already set this way. Add the instructor and TA as contributors.
- 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_S25_repo?
https://github.com/stolson4821/ITM352_S25_repo

What is the path to the local ITM352_S25_repo?
/Users/spencerolson/Documents/GitHub/ITM352_S25_repo

2. Using a terminal (or command line) to access the operating system

Sometimes you will need to do more specific or detailed tasks that would be difficult or inconvenient to do through a GUI. A terminal window provides a command line interface to your operating system. There are many different types of *shell environments* such as sh, csh,bash, zsh,cmd, powershell that run in a terminal. These all have similar functionality and similar commands, but they may vary in the particular command language and syntax they use. You will find it very useful to be familiar with the basic shell commands. We will explore a few basic file system commands here needed for this class. make sure you are in the Lab1 folder.

a) Open a terminal in VS Code (go to the Terminal menu -> New Terminal). Identify what shell is being used: **Terminal running zsh shell, can be identified by improperly inputting commands shows up that the command is not available in (current) shell environment.**

b) Try each of the commands below, copy the result you get and explain what the command does: **See below**

- pw Prints the current working directory, showing the full path of the directory$
- ls Lists the files and directories in the current directory.
- mkdir newdir Creates a new directory named `newdir`.
- cd new Changes the current directory to one matching the pattern `new*`, which coul$
- touch test Creates an empty file named `test` or updates its last modified timestamp if$
- mv test test.txt
  Renames the file `test` to `test.txt`.
- rm test.txt Deletes the file `test.txt`.
- echo hello > hello.txt
  Writes the string "hello" into a new file named `hello.txt`. If the file exi$
- cat h*.txt Concatenates and displays the contents of files matching the pattern `h*.txt$
- cd .. Changes the directory to the parent directory of the current one.
- rm -r newdir Recursively deletes the directory `newdir` and its contents.
- history Displays a list of previously executed commands in the terminal session.

NOTE: if a command doesn't work for your particular shell, look up (or guess) what the command is supposed to do and then do a Google search to find out how to do it in your shell.

c) Most shells support command history. What happens if you press the uparrow key? downarrow key?

Pressing the up arrow uses the previous command that was correctly inputed.
Pressing the down arrow does nothing.

d) Most shells support file name expansion. Try touch xxxx.txt then rm xx then hit the TAB key. What happened?

- touch xxxx.txt creates a text file within the current directory
- rm xx (tab) then auto populates the xxxx.txt file created, once enter filled and enter is hit its deleted.

3. Using VS Code to create a Python program

VS Code is a file editor designed to help build applications (code). Applications are built from files with instructions that the server and browser process. Let try making a simple "hello world" program and viewing it in a terminal.

- In the repo folder, create a new folder inside called Lab1, create file in this folder <your Last_First name>_hello.py. Edit this file and put
- print("Hello from <your first name>!")
  Don't forget to save this file after you make changes!

- Open an integrated terminal window in VS Code (you may find it easier to right-click on the file in VS Code and Open in Integrated Terminal). Type the following in the terminal:

python <your Last_First name>_hello.py

Explain here why you see text in the terminal window. Where did this come from?

- Edit the file and delete the closing quote mark. Click on the PROBLEMS tab next to the TERMINAL tab in bottom VS Code window.

Explain why you see the code underlined in red and some messages in the PROBLEMS :

- Click on the TERMINAL window and try executing the program again.

  Explain why you see the Hello from ! and there are errors messages

-Put the closing quote mark back. In VS Code file manager right-click in explorer/finder, change the file extension from .py to .txt.

Explain why the color of the code changed and what you can do to get it back to what you saw opreviously

- In the terminal, execute the program again
- python <your Last_First name>_hello.txt

Explain why the code did not run and what you need to do to fix it

- Change the file extension back to .py

*Explain why you **do** get the Intellisense help for Python now*

The text that shows up for me is the following when I attempt the action
python <your Last_First name>_hello.py is zsh: command not found: python

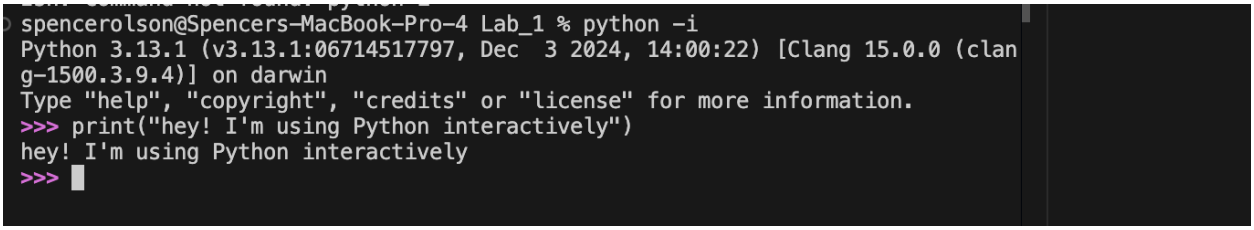
- This was because I did not properly pathe the command, after doing so it showed up as expected with the text in quotations.
- The red lines appear when deleting the quotations because without them the program takes them as individual variable names and does not aptly name them as a whole. You get a name error.
- The benefit of working locally is so that when working on a project you are not tying up your network, nor are you overloading a server with changes from multiple people. This allows efficiency. Once completed with your changes you can commit them to main or let them sit to be reviewed prior to committing them to the main.

4. Interactive Python

In the terminal window type the following:

python -i
Now try

print("hey! I'm using Python interactively")
You can quit interactive Python with



exit()

5. Remote Repo

In GitHub Desktop view the changes made in you local repo. Commit and “push” these to your repository on GitHub. Go to github.com and verify your commits have been added.

Explain the benefits of working locally and pushing changes to a remote repository

Working from a local drive is beneficial so that you don’t bog down a server or platform with multiple users. This allows numerous people to work on a project at once and then once tasks are complete that is when they can submit to the cloud source repo. Reduces bandwidth hindrances and network slowdowns. Enables the user to also work without a network connection which can be invaluable in many scenarios.exit

5. Join Discord and introduce yourself

We will use Discord for most class communication and for seeking help. Use email only for private matters. In your profile you must use use your first and last name as your nickname so that others can recognize who they are talking to! The instructor will email you a Discord invitation. Join the class server and introduce yourself:

- Your name
- Your major(s)
- Where are you from?
- Why are you taking ITM 352?
- Things that you are nervous about ITM 352

Completed and already asked a question on the TA page