Github introduction

1. Create a Github.com account
2. Add your account to google sheet
3. read the directions on how to run visual studio - this is linked through your github account and allows you to run via the web or have visual studio on your laptop but connect to the hpcc server for file saving and github push/pull/commit commands as well. The goal of homework 1 is for you to practice using these tools even though they are new and maybe a little confusing. You can also just use jupyter notebook for your editing of code if you don't want to try these other things out.

There are two homeworks for you to accept:

1. github intro - this is to learn and pratice about github
2. HW1 UNIX- this is to do very basic UNIX and test checkout/commit/push to github plus issuing one or two commands

Github practice

1. From first link you'll read and follow some of the videos and instructions in the github intro repository.

UNIX practice

On the UNIX command line. Go into your bigdata folder. If you have not used the cluster before then you will be in the `gen220` project. Or you may have your own lab bigdata folder.

```
cd ~/bigdata # this should work but if it doesn't
cd /bigdata/gen220/$USER # will go into your bigdata folder for the class
# if the above doesn't work you are likely already in a lab group on HPCC
cd /bigdata/$GROUP/$USER # should work since $USER is your login and $GROUP is your primary
```

For your homework: 1. Accept the homework 1 problem - (see link in Canvas). 2. Create a folder to work in on HPCC (or your own computer): `mkdir -p ~/bigdata/gen220/homework` and then `cd  ~/bigdata/gen220/homework` 3. Checkout the GEN220 data folder `git clone https://github.com/biodataprog/GEN220_data.git` Now you want to make a folder for your work for this class

```
# you can make a folder for GEN220
mkdir gen220
go into that folder
cd gen220
# now use git to checkout the class data folder
git clone https://github.com/biodataprog/GEN220_data.git
```

```
# now go into this folder
cd GEN220_data
```

Look around in the folder. Go into the `tabular` folder where I've stored some tab or comma delimited data. You will later need to copy a file from this folder into your homework folder.

4. Checkout the homework 1 github repository created in step 2. (if you setup SSH keys in github

```
git clone git@github.com:biodataprog/2023-hw1-YOURGITHUBID
```

OR for the https will need to create a token as your password)

```
git clone https://github.com/biodataprog/2023-hw1-YOURGITHUBID.git
```

6. Go into your folder (`cd 2023-hw1-YOURGITHUBID`).
7. Edit a script in there called `filesize.sh`; you can do this in jupyter on web, you can edit on the command line with `nano`, `vi`, or `emacs`, or you can use ssh with aisual studio tunnel or
8. Add some code to this script which achieve the directions at the bottom of this
9. Test it out (run the `./filesize.sh`).
10. To submit your homework (and you can do this more than once), this requires doing `git commit` and then `git push`

```
# this step saves a version of the code
git commit -m "This is a homework 1 solution" filesize.sh
# this step will push the data from HPCC or your computer UP to the github site
# this step will request your username (YOURGITHUBID) and your password (that TOKEN I mentio
# if you have setup github account with SSH keys then it will ask you for your SSH key passw
git push
```

9. you can repeat doing edits to the file, commit, and push to github.

Tasks for Homework 1

1. copy the `threatened-species.csv.gz` file - see info here HW1 or you can just run the included `./setup.sh` script to download. but also encourage you to practice with `cp` command.
2. Update a script called `filesize.sh`, that script should do the following:

- print out the size of the threatened-species.csv.gz using `du` or `ls -l`
- Uncompress the file with `gunzip` while leaving the original alone by adding the '-k' option (`gunzip -k threatened-species.csv.gz`), add a `-f` option so it will overwrite the new file as well in case you run this more than one time.
- Print out the size of the new uncompressed file with `du` or `ls -l`
- print out the number lines in the uncompressed file size

- BONUS: if you can, print out the compression ratio (compressed filesize / uncompressed size), you can do this if you capture the sizes in two variables and run this - requires using cut/awk to capture the filesizes.

```
python -c "print($COMPRESSED / $UNCOMPRESSED)"
```

3. Check in your changes with `git commit -m 'a message'` and `git push` to save the changes to github.