

Software Practice 1

- This miniproject is meant to get you started with the tech we are using like Github, Python, and Linux Shell.
- The instructions will be provided for using the GUI and Command Line for Python and Git, so this should not require the use of a terminal or Linux if not wanted.
- **LINUX SHELL NOTE:** for the shell commands the naming will not be exactly the same for you. Use `ls` in linux to show what is currently in your directory and find your way from there. If you ever get lost, the command `cd ~` will take you to your home dir.

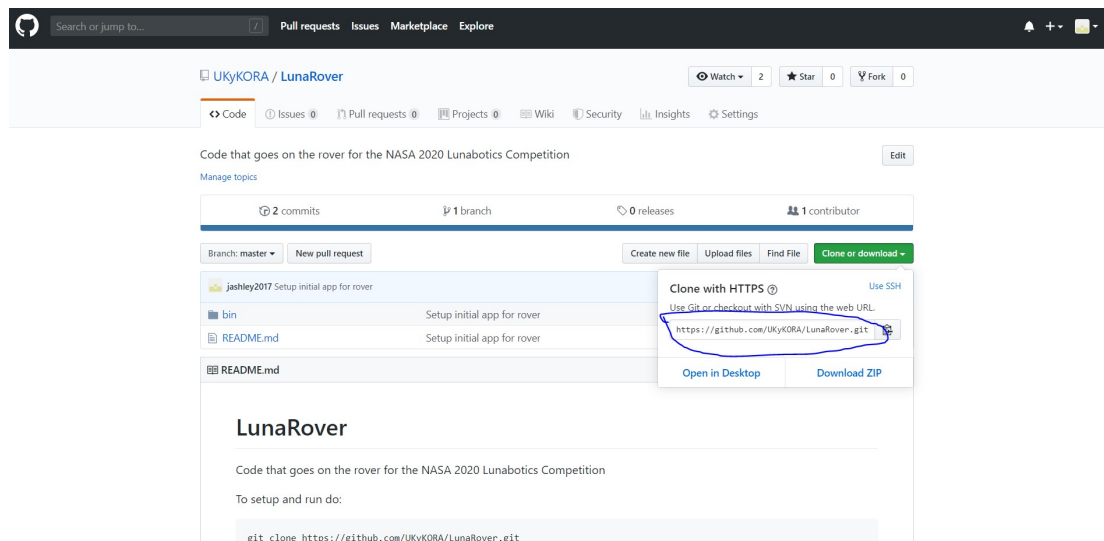
Step 1: Initialize Git

Through Command Line

1. Navigate to where you want your git stuff to end up. I recomend making a folder something like UKYKORA.

```
mkdir ~/UKYKORA  
cd ~/UKYKORA
```

2. Retreive a copy of the git repository from github using the git clone command.
 - NOTE: you are going to want the URL of the specific repo which can be found here



```
git clone {github url here}
```

3. You now have a working copy of the repository! In the directory git clone just made you can now edit project files for the team!

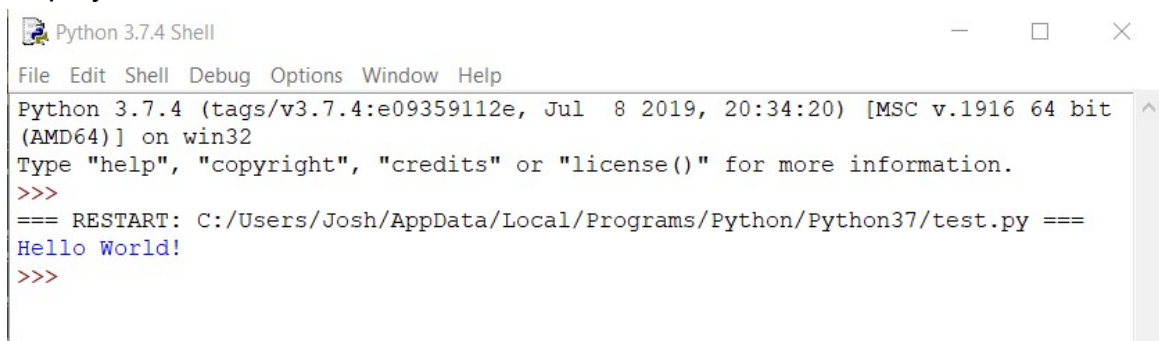
```
ls          <- Lists things in your current folder
cd LunaRover <- changes directory to the git-tracked one (might have a different
ls          <- Shows you what is in the git directory
```

Step 2: Write and Execute your Python Script

With Python App for Windows

1. Make a new file in your git folder titled "{name}_practice.py" and open it in your preferred text editor (Notepad, Notepad++, etc.)
2. Write the code for "Hello World" in python in your text editor (Hint: google howto print out text in python).
3. Press the Windows key to open start menu and type in IDLE.
4. Select the app that says something like "IDLE (Python 3.7)"
5. In the program top menu, select File -> Open and navigate to your Python .py file and open it.
 - o IDLE should now be displaying your .py file
6. Now, in the top menu, select Run -> Run Module
7. Your "Python Shell" Window that opened on IDLE start should now have

displayed "Hello World!"

A screenshot of a Python 3.7.4 Shell window. The title bar says "Python 3.7.4 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area shows the following: "Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32", "Type 'help', 'copyright', 'credits' or 'license()' for more information.", a prompt ">>>", a command line "=== RESTART: C:/Users/Josh/AppData/Local/Programs/Python/Python37/test.py ===", the output "Hello World!", and another prompt ">>>".

```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:/Users/Josh/AppData/Local/Programs/Python/Python37/test.py ===
Hello World!
>>>
```

8. BONUS: Do more programming problems to warmup to Python

- Here is a good website for discovering programming problems:

https://adriann.github.io/programming_problems.html

- There are many good references for Python, my favorite is [TutorialsPoint](#)

With Python linux command

1. Make a new file in your git folder titled "{name}_practice.py" and open it in your preferred text editor (nano, vim (hard), etc.)

```
cd path/to/git/repo/LunaRover/practice
nano joshashley_practice.py
```

2. Write the code for "Hello World" in python in your text editor (Hint: google howto print out text in python).
3. Save and Exit your file and in the terminal type:

```
python3.7 {filename}.py
```

- * HINT: if this doesn't work, python3.7 is likely not installed. To fix this type
- * Your terminal should have printed "Hello World!" at you.

4. BONUS: Do more programming problems to warmup to Python

- Here is a good website for discovering programming problems:

https://adriann.github.io/programming_problems.html

- There are many good references for Python, my favorite is [TutorialsPoint](#)

Step 3: Submitting your changes to Git

Gitbash (Incomplete)

- In your downloaded git folder on your windows explorer, right click and do “Git bash here”
 - Note: only there if git bash installed
 - From here you can do all of your command line things (see below)

Through the Command Line

1. Go to the top level of your local copy of the git repo

```
cd ~/UKYKORA/LunaRover
```

2. Pull down from the remote repository to refresh your files to thhe newest version.

```
git pull
```

3. Check the status of the files in your repository.

- This will show you the files you have added or made changes to and what their current status is for the next commit.

```
git status
```

4. (Optional) Run a git diff program to show you what changes you have made to your repository.

- This is optional right now as it might take some additional setup that you might not be comfortable with. It will be nessecary at a later date but can be setup by Josh Ashley or Cole Gerdemann

```
git difftool
```

5. Add your hello_world program to be committed to the git repo.

```
git add practice/{name}_practice.py
```

6. Commit your change to your local repository

- After this command, the change will be solidified in YOUR repository, but will not show up on github.

```
git commit -m "Commit Message, tell concisely what you added or fixed in the cod
```

7. Push your changes to the remote repository on Github

- This command will send your changes to github, and it will now be publicly available
- If it has not already, at this program will ask you to enter your credentials. Use the login information you gave to github because it is trying to verify that you are permitted to change this repo.

```
git push
```

8. You did it! Congratulations on your first git commit!

- Please make sure you understand what each command is doing at each step, some of these commands when used ignorantly can cause some damage.
- If you are ever confused about what a command does, you can type: *git help {command}* to give you info on it.
- At *any* point in time you can type *git status* in the directory and it will inform you about what has changed in your directory and in the remote, so it is not a bad idea to be frequently using that command to check yourself.