1. Cd From git.bash open working folder.
2. git init
3. git clone https://....

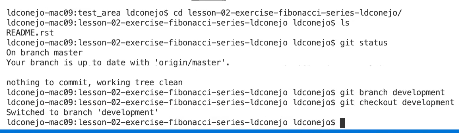
<https://github.com/libgit2/libgit2>

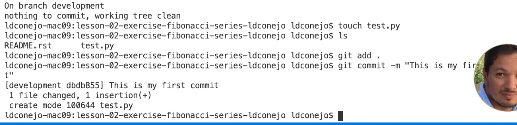
<https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/cloning-a-repository> - clonning

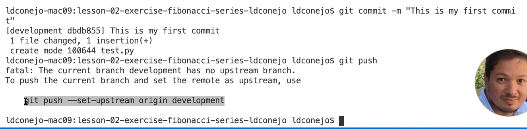
1. git status (need to work not at master branch). Open clonned folder.
2. git branch development
3. git checkout development

To commit the assignment.

1. Cd From git.bash open working folder.
2. Git status
3. “Git add .” (with dot)
4. git commit -m "Commit message"
5. git push
6. copy suggested command: git push --set-upstream origin development
7. Open GitHub and pull request

git







Git statunjkn,m ,.ks

Touch test.py – create new file

Git add . -> add to repository

Git commit –m “This is my first commit” -> -m (message)gi

Git push

git add –h

git help config

git help, git status, git add README

echo %cd% - print the absolute pathname of your current working directory

cd *directory* - change your working directory to the given directory

cd .. - change your working directory to the parent of the current working directory

dir - list the contents of the current directory ("dir" is short for "directory")

python - run

the Python interpreter

python *program.py* - run the Python program that is stored in the *program.py* file

rename  file\_path  new\_name

mkdir name of the folde

pwd – check full path

***echo. >*** *location with name (C:\\_PythonClass\Advanced>echo. > C:\\_PythonClass\Advanced\emptyfile.txt)*

vi – The trick here is that it's difficult to open hidden files using most editors that rely on a typical finder. UNIX comes with a text editor called VI built in that you can access from the Terminal. Open File: $ vi name of the file

To exit VI mode, type :wq. I know, all a little weird, but that's how it works. Welcome to VI.

Also vi adder.py -> create new adder.py file

To save changes: press “esc” and type “:wq” to exit from vi mode

Ok, for the fun part. If you do a ls -p here, there will be no change. To get updated to the .bash\_profile to take effect, you have to do what is called 'sourcing'. To do this, run. 

Now, enter ls -p and your directories will be a different color from your files

"mv" moves the original file, "cp" creates a second file with the same contents

Deleting files is an even easier command (for better or for worse.) The syntax is simply "rm "

$ cat adder.py -> show a script content

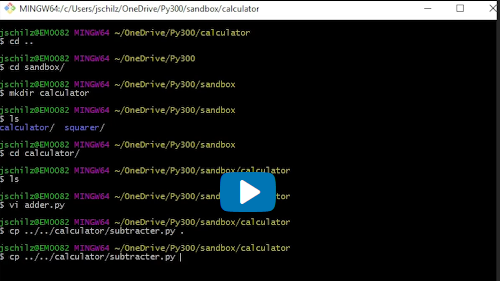
Python – i start python

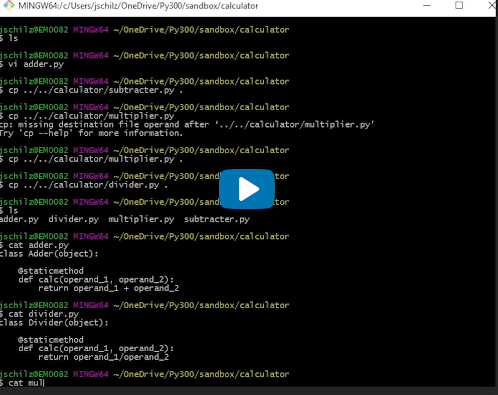
Python – m run a module in python. Like: python –m unites test.

ls the **-a** switch lists every file and folder in the directory. We need to use option ‘**-a**‘ (list hidden files) with command ‘**ls**‘.

We need to use option -‘**l**‘ (long format) with command ‘**ls**‘.

<https://www.lifewire.com/uses-of-linux-ls-command-4054227> -> can be useful commands.





To ignore some type of errors in Pylint -> need to create Pylint configuration file:

Pylint –generate-rcfile > .pylintrc

