Week 2 Lab Assignment Goals

- Clone a Git repository to your local machine
- Learn about opening, reading, and searching files
- Add and commit your changes to the Git repository

Step 1 : Install Git

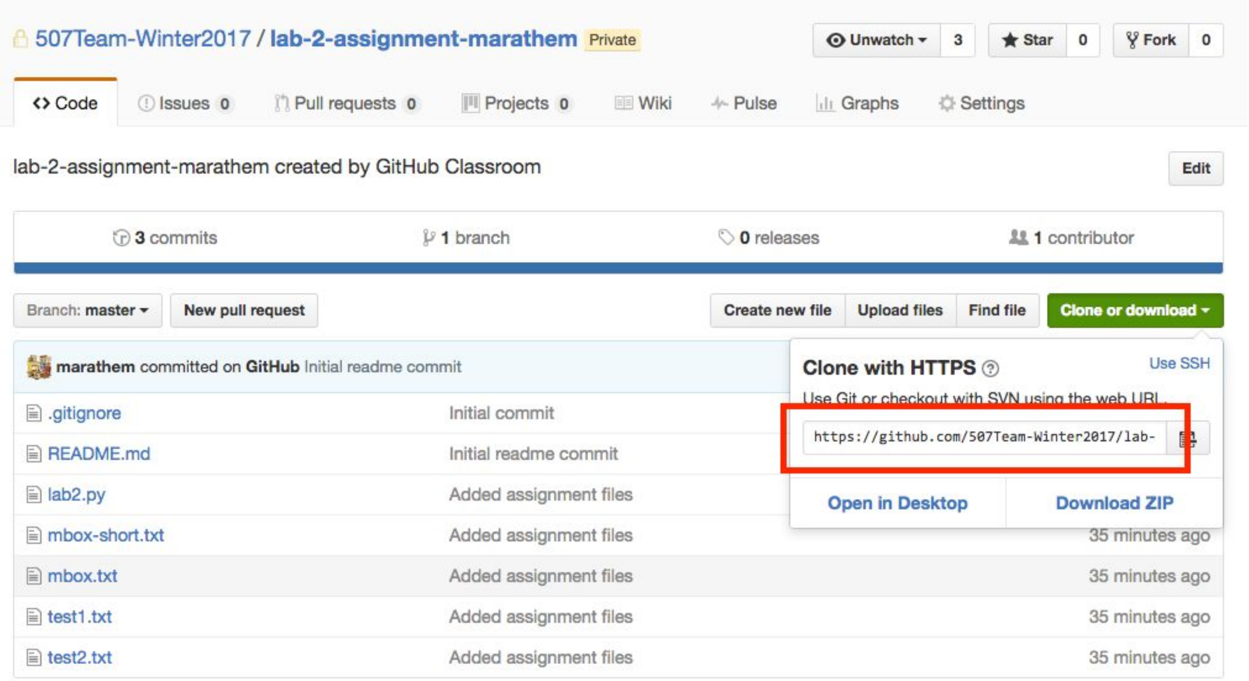
- Go to https://help.github.com/articles/set-up-git/
- Follow the instructions to download and setup Git on your machine

Step 2 : Create a GitHub repository

- Go to https://classroom.github.com/assignment-invitations/6e12ebd000b1586e2c2f14234323d383
- Accept the assignment invite
    - you will need to create a GitHub account if you don't already have one
- This will create a GitHub repository for you containing some starter code

Step 3 : Get starter code onto your machine

- Click the Clone or Download button on your GitHub repository
- Copy the repository URL



lab-2-assignment-marathem created by GitHub Classroom

marathem committed on GitHub Initial readme commit

Use Git or checkout with SVN using the web URL.

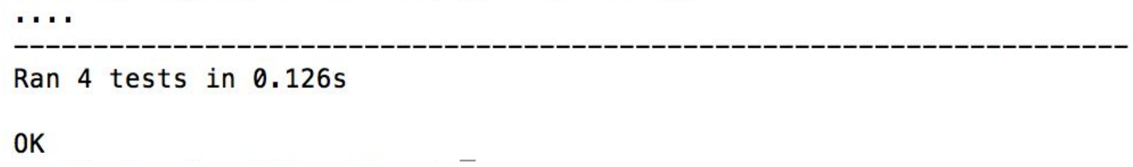
```
https://github.com/507Team-Winter2017/lab-
```

.gitignore	Initial commit	
README.md	Initial readme commit	
lab2.py	Added assignment files	
mbox-short.txt	Added assignment files	35 minutes ago
mbox.txt	Added assignment files	35 minutes ago
test1.txt	Added assignment files	35 minutes ago
test2.txt	Added assignment files	35 minutes ago

- Open a Terminal window or command prompt and cd to the desired folder
- Type 'git clone *repository-URL*'
- Type 'ls'. You should now see your cloned repository. Now 'cd' into this repository.
- Type 'ls' or 'dir' to list the files included in the starter code

Step 4 : File I/O

- Modify the parseData function in lab2.py
    - parseData takes a filename and uses it to calculate the average value of all X-DSPAM-Confidence values in that file
    - parseData should return a string of the form “Average spam confidence is XXX.”
    - If the file does not contain any X-DSPAM-Confidence values, parseData should return “No data to report.”
- Verify whether your code works by typing ‘python lab2.py’ in Terminal or command prompt.
- Your code works if your output looks like the following:



```
....
----------------------------------------------------------------------
Ran 4 tests in 0.126s

OK
```

Step 5 : Commit your changes

- Type ‘git status’
- This will show you all the files you have modified
- Type ‘git add *filename1*’ for every file that you would like to include in your commit
- Type ‘git commit -m “*commit-message*”’ to commit your added files
- At this point, the files are committed locally, but not yet to your GitHub repository

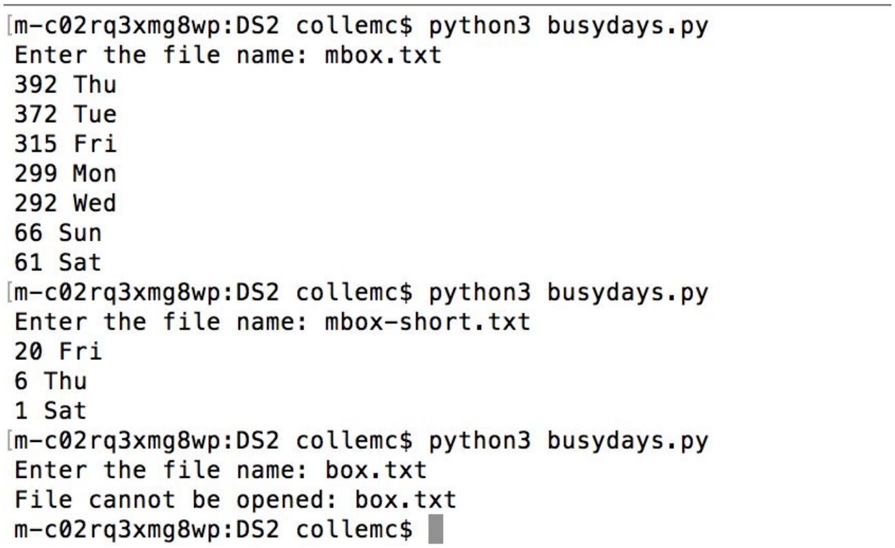
Step 6 : Push your changes

- Type ‘git push’ to push your local changes to the GitHub repository
- Do not forget to run this command!

Step 7 : Additional File I/O

- Write a script busydays.py that
    - Prompts the user for a filename
    - Searches the file to determine on which days most emails are sent. The field that contains the timestamp, including day of the week, is called: “X-DSPAM-Processed”
    - Prints out results in sorted order

- Your code works if your output looks like the following:



```
[m-c02rq3xmg8wp:DS2 collemc$ python3 busydays.py
 Enter the file name: mbox.txt
 392 Thu
 372 Tue
 315 Fri
 299 Mon
 292 Wed
 66 Sun
 61 Sat
[m-c02rq3xmg8wp:DS2 collemc$ python3 busydays.py
 Enter the file name: mbox-short.txt
 20 Fri
 6 Thu
 1 Sat
[m-c02rq3xmg8wp:DS2 collemc$ python3 busydays.py
 Enter the file name: box.txt
 File cannot be opened: box.txt
 m-c02rq3xmg8wp:DS2 collemc$ 
```

- Commit and push busydays.py to your GitHub repository