1. GPL licence, 2017
2. Git init
3. git add ./readme.txt
4. git status - to find the status of the file
5. git commit -m "Completed readme.txt file

You can create a file using a touch. But we can either add or commit to the git.

Steps-MacBook-Pro:my\_project MacBook4$ touch file1.py file2.py

Steps-MacBook-Pro:my\_project MacBook4$ echo "print 'hello world' " > file1.py

Steps-MacBook-Pro:my\_project MacBook4$ echo "print 'this is file2' " > file2.py

Steps-MacBook-Pro:my\_project MacBook4$ git status

On branch master

Untracked files:

(use "git add <file>..." to include in what will be committed)

file1.py

file2.py

nothing added to commit but untracked files present (use "git add" to track)

Steps-MacBook-Pro:my\_project MacBook4$ ls

file1.py file2.py readme.txt

Steps-MacBook-Pro:my\_project MacBook4$ echo "this file also contain two python hello world programs. " >> ./readme.txt

Steps-MacBook-Pro:my\_project MacBook4$ cat readme.txt

2017 (c) This is my first repository, I am using this for training practice, to install this software.

this file also contain two python hello world programs.

Steps-MacBook-Pro:my\_project MacBook4$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: readme.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)

file1.py

file2.py

no changes added to commit (use "git add" and/or "git commit -a")

Steps-MacBook-Pro:my\_project MacBook4$ git add readme.txt file1.py file2.py

Steps-MacBook-Pro:my\_project MacBook4$ git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: file1.py

new file: file2.py

modified: readme.txt

git commit -m "This is my second commit with other 2 python file"

Git log - this is to check the whole log

Method to clone the file from any repository

git clone https://github.com/git/git.git

<https://github.com/ujjwalkarn/DataSciencePython> - good repo to study data science

git branch -v

This shows all the branches in the system

Steps-MacBook-Pro:my\_project MacBook4$ ls

file1.py file2.py readme.txt

Steps-MacBook-Pro:my\_project MacBook4$ git branch-v

git: 'branch-v' is not a git command. See 'git --help'.

Steps-MacBook-Pro:my\_project MacBook4$ git branch -v

\* master 5d77cab This is my second commit with other 2 python file

Steps-MacBook-Pro:my\_project MacBook4$ git branch testing

Steps-MacBook-Pro:my\_project MacBook4$ git branch -v

\* master 5d77cab This is my second commit with other 2 python file

testing 5d77cab This is my second commit with other 2 python file

Steps-MacBook-Pro:my\_project MacBook4$ git checkout testing

Switched to branch 'testing'

Steps-MacBook-Pro:my\_project MacBook4$ git branch -v

master 5d77cab This is my second commit with other 2 python file

\* testing 5d77cab This is my second commit with other 2 python file

git checkout master

git merge testing

git branch -v

Steps-MacBook-Pro:my\_project MacBook4$ git branch -v

\* master ecefb39 Added a line to test

testing ecefb39 Added a line to test

Steps-MacBook-Pro:my\_project MacBook4$ git branch testing2

Steps-MacBook-Pro:my\_project MacBook4$ touch test\_file.py

Steps-MacBook-Pro:my\_project MacBook4$ git add test\_file.py

Steps-MacBook-Pro:my\_project MacBook4$ git commit -m "I added a new file to develop in my testing2 branch"

git checkout testing2

git log --oneline --decorate --graph --all

Vim editor :q to exit

Steps-MacBook-Pro:my\_project MacBook4$ git add ahmad\_test.py

Steps-MacBook-Pro:my\_project MacBook4$ git commit -m "renamed test\_file.py to amad\_file.py"

[master d3438fc] renamed test\_file.py to amad\_file.py

Committer: Step Function <MacBook4@Steps-MacBook-Pro.local>

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly. Run the

following command and follow the instructions in your editor to edit

your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)

rename test\_file.py => ahmad\_test.py (100%)

Steps-MacBook-Pro:my\_project MacBook4$ git log --oneline --decorate --graph --all

\* d3438fc (**HEAD -> master**) renamed test\_file.py to amad\_file.py

\* a98aa81 Merge branch 'testing2'

|\

| \* beabdda (**testing2**) I created a new test file, to begin testing feature X

\* | fa47e49 I added a new file to develop in my testing2 branch

|/

\* ecefb39 (**testing**) Added a line to test

\* 5d77cab This is my second commit with other 2 python file

\* 3faea12 Completed readme.txt file

Steps-MacBook-Pro:my\_project MacBook4$

git branch -d testing2

git branch -d testing

git log --pretty=oneline

git checkout 233d5000

\*\* Always use git move, git remove rather than directly removing stuff \*\*

To check the global user name

git config --global user.name getamar

git config --global user.email getamar@gmail.com

This is the final command required until following command is done:

======================

git push -u origin master

======================

=====================

git push origin master

https://www.open.collab.net/media/pdfs/ClearCase-and-the-journey-to-Git.pdf