All the git commands

First initial the git in the working directory:

* git init

For showing all the files or folders in the directory:

* ls
* showing for hidden files: ls -a

For show the directory’s status:

* git status

Stagging the changes in the directory:

* one file: git add file\_name
* all file: git add .

For showing the changes into a file:

***we can see the changes from working area, it means before commit***

* git diff

For moving previous state: (without add the file)

* git restore file\_name

For stagging stage to move local repo:

* git commit -m “add message related to the changes”

For showing the committed details from the local repo:

* git log
* git log –oneline (show committed data in very shortly)

For show all the changes of the file:

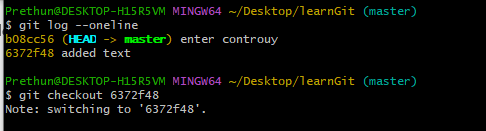
***we can see the changes from stagging area, it means after commit***

* git show
* git show head

Suppose we commit 2 times. Now we have 2 branch. Where head stay your current branch is also here.

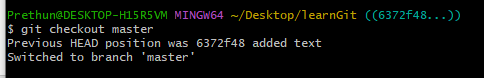
For moving the previous branch:

* git checkout commit\_ID



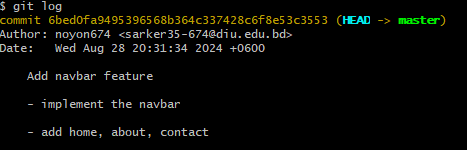
For moving updated version:

* git checkout master (If you have done 10 commits then updated one is master)



For adding multiple line into the one commit:

* git commit -m “Adde navbar feature” -m “- implement the navbar” -m “- add home, about, contact”



**HEAD:**

After every commit git HEAD is changed. It moves to the latest version. Every commit has different ID.

We can move to the specific HEAD using the commit id. The command is: *git checkout commit\_Id.*

git commit

Local Repository

Stagging

Area

git add

Working

Directory

If you want to go *local repository* to staging area:

* git reset -soft HEAD^

If you want to go back *local repository* to working area with recent changes:

* git reset HEAD^

If you want to go back *local repository* to working area with delete all the recent changes:

* git reset –hard HEAD^

Adding shortcut of git command: shortcut for git commit command:

* git config –local alias.ci “commit”
* now, we can write git ci -m “message”