Git & Github

**-------------------------------------------getting started with git ---------------------------------------------------**

git config --global user.name or user.email "your name or your email".

git config --list (list all the config settings)

git config <user.name> or <user.email>(to check spacefic setting).

git help <var> (to check a specific commands).

git help (view the commands).

-------------------------------init git------------------------------------

echo "text" README.md(add readme file).

git add . (add file to add files to staging area).

git commit -m"desc of the commit" (commit changes from the stage area).

git remote add <name> <link>(name usually origin).

------------------------------github---------------------------------------------------

git push [-u]<remote name> <branch name>.(-u creats a tracking reference for every that you

successfully push onto the remote the local branch is aautomaticlly linked with the remote branch

this allows the user to use command such as pull with out arguements).

git pull (pulls the rep).

------------------------------------------------------------git-----------------------------------------------------------

git status (gets info of the rep like branch , the state of the branch if its up to date or notand staged & changed files(if there is files to commit)).

touch <file name>(only for linux to creat a new file).

git status -s / --short (short term of status command).

git diff --staged(compare between different version of the staged files).

git diff (to see the not staged modified/added files and compare staged with not staged).

git commit [-a] -m""(commit changes | -a skips the add to stage area step).

git log [#](commit history the # is to limit the number of commits shown).

Git log –oneline/status/patch (oneline shows the history in one simble line/status detailed history with the file changes in details/patch show history detailed with file changes and what line changed in which file in details(press q to quit))

Git rm [--cached]<file name> (to stop git from tracking file with<file name> and deletes it| cached same as before but don’t delete it)

Git mv <file name> <new file name> (to rename a file using git).

------------------------------------------------------------------------branches--------------------------------------------------------------------------

Git branch <branch name > (create new branch).

Git checkout [-b] <branch name>(to go from branch to another | -b is to create new branch).

Git stash (stashes the changes to work with a clean branch).

Git stash pop (brings the changes back).

Git merge <name of the branch to merge with current branch>(merges two branches (you need the merge commit msg using vem tool first press I then insert the msg w to save changes q to quit)).

----------------------------------------------------------------reset commits-----------------------------------------------------------------------------

Git reset [--soft / -- mixed / --hard] <commit to move to>(--mixed or just reset to move the changes back to working directory / --soft moves the changes back to the staged directory / --hard move the changes to trash including the changes in staged area)

------------------------------------------------------------- pull work from github--------------------------------------------------------------------

Git clone <url> (clones the repostry).

**---------------------------------------------working with branches ------------------------------------------------**