Git & github notes

-git means distributed Version control system

-Git is a free and open source

-github is source for project and sources

-github simplify using git

-you can use git without github

-git has GUI

Why learn git?

-developers contribute to the same project

-you can revert the changes

-you can collaborate to fix bugs

- you can collaborate to create new features

-solve conflict

-oraganize features

Words you will hear

-repository(repo)

-branch

-local repo

-remote repo

-commit(checkpoint in your local repo)

-clone(from local or remote)

- push(upload local changes to remote)

-pull(pull changes from remote to local)

-pull request

Git commands

-git clone +(repo link) >> Git clone is a command for downloading existing source code from a remote repository

-git status >> gives us all the necessary information about the current branch

-git add +(file name)/or gitt add * → -means add all-

-git reset head

-git commit -m “message” >> Takes the staged snapshot and commits it to the project history

-git branch

-git remote -v

-git push +remote name +branch name

-git pull +remote name >> is used to get updates from the remote repo

-git config -l >> A convenient way to set configuration options for your Git installation
-git help config
-git config --global user.email
-git config --global user.name
-git config -l --show-origin
-git config --global --unset user.email
- git config --global --edit
-ssh-keygen -t rsa -b 4096 -C “email”
- cat /.ssh/id_rsa.pub
-git init >> Initializes a new Git repository
- git push -u <remote> <branch-name>
- git config --global alias.st status
- git checkout 〈branchname〉 〉〉 Switching branches
- git branch -d localBranchName >> delete branch
- git branch -m <new name> >> rename branch
-git merge >> combine multiple sequences of commits into one unified history
-git stash>> is a built-in command with the distributed Version control tool in Git that locally stores all the most recent changes in a workspace and resets the state of the workspace to the prior commit state.
- git stash pop >> Takes the files in a stash, places them back into the development workspace and deletes the stash from history
- git stash list >> Displays the stash history in chronological order
- git stash apply >> Takes the files in a stash and places them back into the development workspace, but does not delete the stash from history
-git stash drop
-git stash show
-git stash clear>> Removes all entries in the git stash history.
- git restore --staged <file> >> to undo changes in the staging area
- git clean –n >> just show what would be done
- git tag <tag name>
- git tag –a <tag name> -m "< Tag message>
- git tag –l
- git tag -d <tagname>