git -

1.version Control System (VCS),

2.Source control System(SCM),

3.Source code System

\*Directory-> files irukum

\*Respositary-> files, directory, history irukum.

git –version //to know git version

git config --global user.name "srimuthurajesh"

git config --global user.email "srimuthurajesh@gmail.com"

git config –list

git init //to make current folder git

git init –base //important for remote respo

git checkout <url> //use to return back when the changes are not \*/added

git clone <url>

git add . //save this point in time

git add hello.html //save the specific file

git remote add origin /path or<url>

git remote rm origin //to remove current origin

git push origin master

git pull //update

git status //to know status

git commit -m ""this is out first comment"

git commit -am ""this is out first comment" //moving directly to staging

git rm –cached . //remove entire folder from tracking

git rm –cached index.html // it wil not delete file, just it stop tracking the file

git log //this will show the previous commits

git rm filename //remove file from staging

git reset –hard //move to last commit given,

git reset HEAD //reset to recent changes r

git diff //looking for different between staging and working copy

git diff --staged //difference between working and respository

git diff branchname1 branchname2

git diff commithash 1 commithash2

**Branching:**

git branch //to know the current branch

git branch branch\_name // create a branch

git checkout -b branch\_name //shifts to branch to branch

git checkout --force branch\_name //shifts when code are not yet commited

git merge branch\_name //merge to current branch

git branch –d branch\_name // delete branch

git branch –D branch\_name //force delete branch while it not merged yet

git diff branch\_name //comparing branches

git diff commit\_hashcode1 comit\_hashcode2 //compare two commits

**Git stash:**

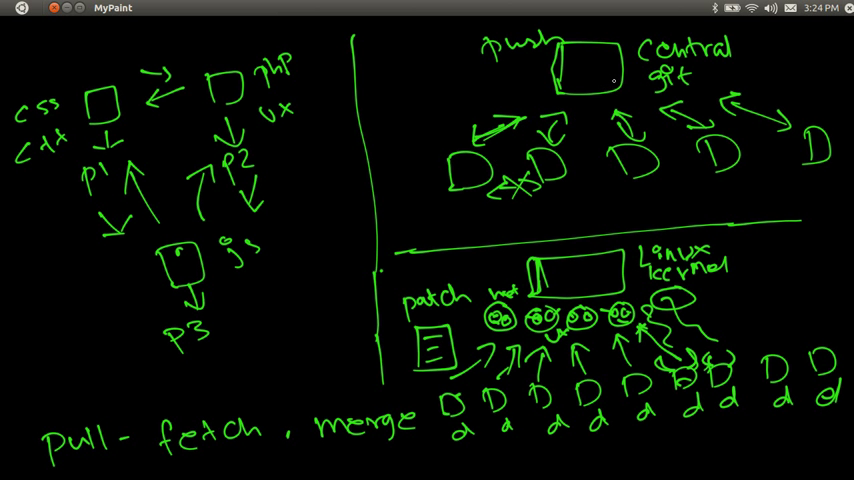
git stash commit stash\_name //temporary commit, but it is not actual commit

git stash list //list the stashes in branch

git stash apply stash\_code //now the stash will apply or used

git stash pop //recent stash will apply or used

git stash clear //remove all stash

git stash drop stash\_codeid //remove specified stash

**Git tags:**

git tag –a tagname –m “message”

git tag –l //list tags

git show tagname //to show current tag

git tag –d tagname //delete tag

**Git Patch: a text file, git diff along with code**

git format-patch –stdout master>branch

git am patchfilename.patch //apply patch

git am –signoff patchfilenam.patch // add username

git help //list

git help topic\_name

pwd // git folder

git log --author="srimu" //list only srimuthurajesh's comments

working copy > staging area > respositary

**renaming file is nothing but remove old file and adding new file name**

git mv filename.txt filename2.txt //move

git mv filename.txt foldername/filename2.txt //move to new folder

git checkout -- filename.html //undo changes old version back from respositary

git reset HEAD filename.html //undo file back to unstage from working area

git checkout commitcode -- filename // move to previous versions based on code

git remote add srimuthurajesh http://github.com/srimuthurajesh

git remote

git push -u nickname master //push project to respository, it will ask for usename password