**GIT COMMANDS**

* Git init : create new local respo in order to use git.
* Gitk : to open git UI.
* Git add :
  + Git add [filename] [filename] [filename]
  + Git add . \\ add all changes into staging area
* Git status : check any Modified, Untrack, ....
* Git commit –m “ .... “ : commit to files which are added to local respository
* Git push [origin master]: push/ up all files in local respo to remote respo.
* Git log : show all commit versions.
* Git show [code of commit] : show any change in this commit.
* Git diff : see changes in **modified** files
* Git checkout -- [filename] : discharge changes in working dir (undo that file)
* Git reset : to unstaged changes from staging area
* Git branch : to see the branch we are working in.
* Git checkout –b [branch-name] : create new branch and move to that branch.
* Git merge [branch-name]: merge/combine all changes from branches.
  + Step 1: checkout the branch we want to merge (usually, branch **master**)
  + Step 2: git merge the branches.
* Git branch –D [branch-name]: delete the branch.
* Git reset
  + --soft
  + --hard [code-name]: delete all after that
* .gitignore : add file name in this file to prevent pushing to remote repo. (usually, node\_package,node\_module,...). Note: ignore the files that have never pushed before or you will fix them.
* Git push origion [branch\_name]: to push a branch to remote repo.
* Git pull: use it before doing anything. For update new things
* Git remote –v : to see name of origion.
* Git fetch [branch\_name]: to checkout the branch without seeing. Usually use to test code before merging request.
* How to resolve Conflict:
  + Use rebase
  + Use merge master.