**GIT :-**

* Git –version ----🡪 to see the version of git
* Git init : initializing
* Git config –global user.name “Meghanath315”
* Git config –global user.mail “[meghanath55555@gmail.com](mailto:meghanath55555@gmail.com)”
* Git clone **(GIT HUB REPOSITRY)** ---🡪 to download files from repository and connect with bash to repository

**or**

* Git remote add origin **(GIT HUB REPOSITRY)** --🡪 to connect git bash to repository
* Git pull origin master -----🡪 to download the files from master branch
* Git push origin master -----🡪 to push the all local repository files into global repository
* Git status --🡪 to check the status of git files
* Git add (file name) -----🡪 to move the file into staging area
* Git add . ------🡪 to copy the all files(modified / created) into staging area
* Git commit –m “description” ----🡪 to copy the staging area files into local repository
* Git commit –am “description” ---🡪 to copy the modified global repository files.
* Git push –u origin master ---🡪 to copy the files from local repository to global repository
* Git branch --------🡪 list of branches
* Git branch (branch name) --------🡪 creating a new branch
* Git checkout (branch name) ------🡪switching into new branch
* Git checkout –b (feature branch name) ---🡪 creates new feature branch and switched into it
* Git merge (feature branch name) -----🡪merging feature branch with master branch
* Git branch –d (feature branch name) ----🡪 to delete the feature branch
  + It will show unmerged files if not merged with master branch
* Git branch –D (feature branch name) ----🡪 to delete the feature branch
  + It will not show unmerged files if not merged with master branch
* Git mkdir (folder name) --🡪 to create folder
* Cd (folder name) ------🡪 to enter into folder
* Git touch (file name with extension) ----🡪 to create file
* Git vi (file name) -----🡪 to edit the existing file if it’s not there then it will create file and it will move into editing mode of file
  + Esc ---🡪 i

to enter into editing mode

* + Esc ---🡪 : wq!

To exit editing mode

* + Esc ---🡪 : q!

To exit editing mode forceful

* Git stash –u -----🡪 to save uncomplete files in working directory
* Git stash list -----🡪 list of files in stash
* Git stash show ---🡪 show the list of files in stash
* Git stash apply ---🡪 to apply the stash repository files
* Git log ---🡪 to show the committing history
* Git log --author=”(branch name)” ----🡪 it will show the (branch name) committing history
* Git log –before=”date” --🡪 to show the commits before the date
* Git log --oneline ------🡪 to show the commit history
* Git revert (shawn number) --🡪 it will back to file into shown number commit
* Git revert HEAD --🡪 file role back to new version
* Differences between GIT and SVN
  + Decentralized:
    - GIT : have 3 repositories
      * Local area
      * Staging area
      * Central area

We don’t need always to connect central repository

* + - SVN: Should always connect with central repository if you are not then we can’t push or pull code.
  + Complex to learn
    - When compare to SVN, GIT bit difficult to learn
  + Unable to handle binary files
    - Git is slow when the large binary files are frequently changing
    - SVN can easy to handle binary files
  + Internal directory
    - Git only creates only one .git file for all folders
    - SVN creates .svn file in each folder
  + User interface
    - Git doesn’t have good UI
    - SVN have good UI