1. git init (command executed in current dir) ----- will convert curent dir into local repository
2. git clone <repository-url> ----- will clone repository to your workspace
3. git fetch <repository-url> <branch-name> ----- fetches particular branch in that repository

**CONFIGURING**

1. git config --global user.name “saurabh” ----- will configure user name
2. git config --global user.email “saurabhsatpute2gmail.com” ----- will configure email
3. git config --list ----- will list down configurations

**STAGING AREA**

1. git add <file-name> ----- brought file to staging area and is ready to commit
2. git status ----- will show which files are staged or unstaged
3. git add . ----- brought all files to staging area ready to comit
4. git rm --cached <file-name> ----- will bring back file form staging area to working dir
5. git restore --staged <file-name> ---- will unstage the fie from staging area ( Note:- if a file is staged you cannot see it in your working dir)

**LOCAL REPO AREA**

1. git commit -m “your-commit-message” ----- commits file which is in staging area to local repo
2. git reset --soft <commit-id> ----- will bring back file from local repo to staging area
3. git reset --soft HEAD^ ----- will delete recent commit with file
4. git log ----- shows logs that are commited
5. git log --oneline ----- shows logs in oneline
6. git reset --hard <commit-id> ----- will delete all commits above that commit-id

**PUSHING LOCAL REPO TO CENTRAL REPO**

1. git clone <url of repository> ---- will connect to repository
2. git remote add origin <url-of-repository> -----will connect cli with repository (for this you have to manually crate repository on github)
3. git remote -v ----- to check whether repository added or not
4. git pull origin master ---- to pull all the changes from master branch
5. git fetch origin master ---- to pull specific updated change from master branch
6. git push origin master ----- pushed master branch to github
7. git push origin master --set-upstream -- will create master branch & push data

**NOTE : Authentication error - authenctication can be done in two ways**

1. **TOKEN BASED** :- settings > developer settings > personal access tokens > Tokens (classic) Dont forget to list down all the permissions required to get access otherwise you will not be able to access . save this token as it acts as password.
2. **KEY\_BASED**:- generate key pair in your server and add public key in gihub console

**BRANCHES**

1. git branch <branch-name > ------ creates branch
2. git checkout <branch-name> ------ enter inside that branch
3. git diff <branch-1> <branch2> ----- differentiates between two branches
4. git merge <branch-1> ----- merged branch -1 with branch-3 (currently present in branch -3)
5. git checkout -b <branch-4> ----- created branch & checked out