To initialize local directory as git repository

**Git init**

If you really want to remove all of the repository, leaving only the working directory then it should be as simple as this.

**rm -rf .git**

To work on repo

**Git clone** <”repo link”>

**git clone** "git@github.com:sunilangadi2/Data\_Strcture-and-Algorithm-with-problem-solving.git"

The git remote command lets you create, view, and delete connections to other repositories.

**git remote**

List the remote connections you have to other repositories.

**git remote -v**

Same as the above command, but include the URL of each connection.

Add the git remote repo

**git remote add origin** git@github.com:sunilangadi2/Data\_Strcture-and-Algorithm-with-problem-solving.git

To create new branch

**Git checkout -b <branch name>**

To delete any branch

**Git branch -d <branch name>**

Note: You can delete the current branch by moving to another branch

To See branches

**Git branch**

should show all the local branches of your repo. The starred branch is your current branch.

If you want to retrieve only the name of the branch you are on, you can do

**git branch | grep \\* | cut -d ' ' -f2**

To add or modify the master branch repo

**Git add \***

* **git add -A stages all changes**
* **git add . stages new files and modifications, without deletions**
* **git add -u stages modifications and deletions, without new files**

**add \*** means add all files in the current directory, except for files whose name begin with a dot. This is your shell functionality and Git only ever receives a list of files.

**add .** has no special meaning in your shell, and thus Git adds the entire directory recursively, which is almost the same, but including files whose names begin with a dot.

To update change on specific file

**Git add <file name>**

Notes: The git add command adds a change in the working directory to the staging area. It tells Git that you want to include updates to a particular file in the next commit.

To know what changes are done

**Git diff**

To know the status of your branch

**Git status**

To remove files or folder

**Git rm -r <file/folder name>**

commit the changes using

**git commit -m** ( give appropriate message in commit )

E.g --- git commit -m "remove folder"

push the changes to the current branch repository

**git push --set-upstream origin <branch\_name>**

when you want to record the current state of the working directory and the index, but want to go back to a clean working directory. The command saves your local modifications away and reverts the working directory to match the HEAD commit.

**Git stash**