# Command Line specific:

|  |  |
| --- | --- |
| Commands | Functions |
| * cd <directory name> | Changes the current directory (navigation) |
| * ls | Gives you all the list of files/folders present in the directory that you are inside of |
| * mkdir <name> | Creates a folder of name |
| * touch <name> | Creates a file of name |
| * cat <file name> | To read the file in command line |
| * vi <file name> | To edit the file in vim |
| * How to exit vim ? | LOL HaHaHa (:wq) |
| * rm -rf <filename> | To force delete the specific file |
|  |  |

# Git specific:

|  |  |
| --- | --- |
| Commands | Functions |
| * git init | Create an empty git repository or reinitialize an existing one |
| * git add -A | Stages all the untracked files |
| * git status | Shows the status of tracked, modified, and untracked files |
| * git commit -m “<message>” | Commits all the staged files with a message |
| * git log | Provides logs of all the previous commits |
| * git reset <commit code from logs> | It resets the commit status equivalent to the status of the commit of which you provided the code, The rest of the commits goes in untracked status (but it’s still there) |
| * git stash | It basically sends the **staged files** to backstage such that the tree becomes clean again (Important when u want to delete some previous commits) |
| * git stash pop | It brings back all the backstage files that you have sent, and its status is untracked |
| * git stash clear | It deletes all the files that are in backstage |
| * git branch <name> | Creates a new branch of name:name |
| * git checkout <branch\_name> | It makes you to move to the branch specified, now changes will be added in this branch |
| * git merge <branch\_name> | It merges the branch with main |

# GitHub specific:

|  |  |
| --- | --- |
| Commands | Functions |
| * git remote add origin <repo URL> | It connect your local git project to your github repo |
| * git push origin <branch name> * git push origin <branch\_name> -f (Force Push) | (branch name – default : main/master) It push all your commits on github |
| * git clone <repo URL> | It clones/downloads the repo from github |
| * git remote add upstream <repo\_URL\_from where you have forked> | You can add multiple URLs, origin/upstream are just names |
| * git remote -v | To get all the linked URLs |
| * git pull {origin,upstream (URL)} main (branch) | It pulls all the commits from URL to your branch |