



Cheat Sheet For Git & GitHub Commands

Essential Commands For
DevOps

Revise before an Interview

Chaitannyaa Gaikwad

Basics Commands:

\$ git init – initializes a new Git repository.

\$ git add [filename] – adds changes to the staging area.

\$ git commit -m "commit message" – creates a new commit with the changes from the staging area.

\$ git status – displays the status of the working directory and staging area.

\$ git log – displays the commit history.

\$ git restore [filename] – get the previous file version back to the file system.

\$ git branch – lists existing branches or creates a new branch.

\$ git checkout -b [branch name] – create a new branch from the current branch

\$ git checkout [branch name] – switches to a different branch or a specific commit.

\$ git merge [branch name] – combines changes from different branches into the current branch.

\$ git remote -v – check connections to the remote repository.

\$ git remote add origin [remote repo url] – create a remote repo access named origin

\$ git push origin [branch name] – sends committed changes to a remote repository.

\$ git pull origin [branch name] – fetches changes from a remote repository and merges them into the local branch.

\$ git clone [remote repo url] – creates a local copy of a remote repository.

\$ git config – sets or retrieves Git configuration options.

\$ git config --global user.name "username" – username for git accountability.

\$ git config --global user.email "user email id" email id git accountability.

\$ git diff [branch name] – displays differences between the current branch and provided branch.

chaitannyaagaikwad@gmail.com

Advanced Commands:

\$ git revert [commit]: Creates a new commit that undoes the changes made by a previous commit.

\$ git reset [commit]: Moves the current branch to a specified commit, by resetting or removing changes made after that commit.

\$ git cherry-pick [commit]: Applies the changes made in one or more commits onto the current branch.

\$ git merge [branch]: Combines two or more branches into a single branch, creating a new commit that represents the merge.

\$ git rebase [branch]: Moves the current branch to a new base commit, replaying any changes made after the old base commit onto the new base commit.

\$ git stash: Temporarily stores changes that aren't ready to be committed yet, allowing you to switch branches without committing incomplete work.

\$ git stash list -----> displays stash jobs list

\$ git stash apply stash@{0} ---> get back you incomplete job to working directory

\$ git stash clear -----> remove all stash jobs permanently from the stash

\$ git stash pop -----> opens recent stash to working directory and removes from stash

\$ git stash drop -----> removes the most recent stash job from the stash without applying it to the working directory.

\$ git squash ----> git rebase -i [commit]: Combines multiple commits into a single. Note - this Squash command is pro version of rebase command

\$ git submodule add [repository-URL], git submodule update: Allows you to include another Git repository as a subdirectory within your repository.

\$ git blame [Filename]: Shows who last modified each line of a file and when.

\$ git reflog: Shows a log of all the changes to the repository's HEAD over time, even if they have been undone or lost.

chaitannyaagaikwad@gmail.com

*Thank
you!*

[Chaitannyaa Gaikwad's Blog_\(hashnode.dev\)](#)

[linkedin.com/in/chaitannyaa-gaikwad-b16965115](https://www.linkedin.com/in/chaitannyaa-gaikwad-b16965115)

chaitannyaagaikwad@gmail.com