

Git init	Initialize git into workspace.
rm -rf. git	Remove git vcs from repository.
git -v or git help	show git version or show help list on terminal.
git remote -v	show remote URLs that git has stored for shortname.
git config --global user.name "ashish"	Set name to git configuration
git config --global user. email "example@gmail.com"	Set email to git configuration
git add file.html file2.css folder/file3.js	Add individual files to staging area to track by git vcs
git add .	Add all files of the current repo to staging area
git reset or git reset files...	Unstage all files or Unstage files individually.
git status	To make sure we've prepared the right stuff and to verify current branch.
git log or git log -p	Used to display the project's commit history, note- press space to see more. Git log -p for detailed commit history.
git checkout <filename's>...	To discard changes in working directory.
git commit -m "short description"	Wraps up your changes done only of the files in the staged
git commit --amend -m "new description"	Correct recent commit message
git diff	Display all changes occurred in staged files.
git diff branch1..branch2	See all the changes from the "branch2" that we don't have in "branch1", yet.
git remote add <remote_name> <remote_url>	Adding remote repo in local workspace.
git remote rm <remote_name>	Delete a remote repo from the local workspace.
git push origin	Push the commit's to remote repository name as "origin"
git push origin <branch_name>	Push to specified remote branch, if not exist than create new.
git push origin --delete <old_name>	Delete a remote branch.
git push origin -u <branch_name>	Push the new local branch and reset the upstream.
git branch <branch_name>	Create a new local branch.
git branch	Show all the local branches note- * denotes current branch
git branch -r	Show all the remote branches.
git branch -a	Show all local & remote branches simultaneously.
git branch -m <new_branch_name>	Rename the local branch.
git branch -d <branch_name>	Delete the local branch from your workspace.
git checkout <branch_name>	Move to another branch in the workspace.
git checkout -b <branch_name>	Make a branch and move to it.
git checkout -b <new_branch_name> --track <remote_name>/<remote_branch_name>	Create a new local tracking branch from a remote git branch with a different name.
git merge <branch_name>	Merge the specified branch in the current branch. Checkout the branch in which you want to merge the other branch.
git merge --abort	Undo recent merge.
git ls-tree -r --name-only <branch_name>	List all the files of the repo related to the specified branch.
git rebase <branch_name>	Rebase a specified branch on to the current branch.

Git clone <remote_url>	Clone a remote repo. Firstly, make sure you are in desired folder.
Git fetch <remote_name>	To update the information about remote, it just downloads data from remote but not integrate into your workspace.
Git checkout --track <remote_name>/<remote_branch>	After fetch if we want to integrate the latest work of remote into local workspace but make sure push all commits before do it.
Git pull <remote_name> <remote_branch>	This command downloads new commits from remote and directly integrate into your work space. And if tracking connection is stabilized already than one can use "Git Pull" only.
Git Pull	
Git push -f <remote_name> <remote_branch>	Force overwrites a remote branch with your local branch.
Git push --force-with-lease	Is a safer option that will not overwrite any work if more commit were added to remote branch.
Git pull --rebase <remote_name> <remote_url>	If only taking in remote changes was your requirement and remote already exists in your local you can use this command
Git reset HEAD~<no_of_commit>	Git will reset recently no of commit. Note- There is no functionality to reset an arbitrary/specified commit.
Git reset --hard <hash_id_of_the_revision>	The fastest way to restore the older version is to use "reset".
Git reset --soft <hash_id_of_the_revision>	Git will reset by keep all undone commit as local modification.
Git revert <revision_id>	Git revert all changes from remote without changing history.

Git Assist Sheet prepared by Ashish Sharma

This sheet contains all basic and useful commands of Git version control system.

Please don't be blind sure while referring it and don't panic if you see any unexpected behavior of commands in terminal, just copy the error and check it out in internet.

You will surely find out the various ways to tackle your issue.

Have a cool day ☺ You can contact me through my portfolio- [Aeshtech.com](https://Aeshtech.com)