

GIT CHEET SHEET

- 1) Verifying the installation
 - a. `git --version`
- 2) Change the defaultbranch name of the Main from Master use:-
 - a. `git config --global init.defaultBranch main`
- 3) Configuring the username and email ID
 - a. `git config --global --list`
 - b. `git config --global user.name "Username"`
 - c. `git config --global user.email "Email"`
- 4) Initialize repo
 - a. `git init`
- 5) Move the file from untracked/unstaged to staged area
 - a. single file= `git add <file name>`
 - b. multiple files= `git add .`
- 6) Commit
 - a. New file= `git commit -m "message"`
 - b. modified file= `git commit -am "message"`
- 7) Remote Repo
 - a. Connect Local repo with Remote repo:- `git remote add origin <repo url>`
 - b. Verify remote connections:- `git remote -v`
 - c. Remove remote repo:- `git remote remove <name of the remote repo>`

8) Log commands

- a. Commit list in details= `git log`
- b. Commit list in one line= `git log --oneline`
- c. List of all the authors within the project (s=summary and n = numbered) = `git shortlog -s -n`
- d. Commits list by author= `git log --author <Author Name>`
- e. OR = `git log --author=<2/3 starting letters>`
- f. Logs of specific file= `git log --author <Author Name> <filename>`
- g. Logs for a specific file= `git log -p <fileName>`
- h. Decorated view of log= `git log --all --decorate --oneline --graph`
- i. OR= `git log --graph --oneline`
- j. Logs from remote repository - `git log reponame/branchname`
- k. Log view of specific commit= `git show <comitid> --stat`
- l. View last few commits= `git log -<no of commits you want to view>`
- m. View logs between specific dates= `git log --after <YYYY-MM-DD> --before <YYYY-MM-DD>`
- n. Detailed view of logs in one line= `git log --stat --oneline`
- o. To read the total count of commits in the repo:- `git log --oneline | wc -l`

9) Reflog Command

- a. To view the backend log with even the hidden steps we can use:- `git reflog`
- b. To see all the details of reflog use - `git reflog show --all`

10) Renameing a file from git

- a. `git mv <old name> <new name>`

11) View the list of configuration

- a. `git config --global --list`

12) Alias:-

- a. `git config --global alias.br branch`
- b. `git config --global alias.ci commit`
- c. `git config --global alias.st status`
- d. `git config --global --unset "alias.name"`
- e. `git config --global --unset-all`

13) Diff commands:-

- a. Difference between two commits = `git diff <old commit> <new commit>`
- b. Modified VS Committed = `git diff <filename>`
- c. Staged VS Committed = `git diff --staged <Filename>`
- d. Staged VS Specific Commit = `git diff <commit ID>`
- e. Remote VS Local = `git diff origin/master`

14) Pushing Changes to remote Repo

- a. 1st time push = `git push -u origin <remote branch>`
- b. multiple times push = `git push`

15) Cloning a repo

- a. Complete repo = `git clone <URL of the repo>`
- b. Only a branch from the repo = `git clone -b <Branch name> <Repository URL>`

16) View List of Files

- a. List all files – `git ls-files`
- b. List staged files – `git ls-files --stage`
- c. List of file in a directory – `git ls-files --directory <dir name>`

17) Configure git ignore to your system

- a. `git config --global core.excludesFile <PATH TO .GITIGNORE FILE>`

18) Checkout Commands:-

- a. To switch to a branch = `git checkout <branchname>`
- b. To switch to a remote branch = `git checkout <remotebranch>`
- c. To switch to a commit `git checkout commitid`
- d. To switch to a tag `git checkout tagname`

19) Clean Command

- a. To view the implication of the clean command - `Git clean --dry-run`
- b. To force delete the file from the repo - `Git clean -f`
- c. To delete dir - `Git clean -f -d`
- d. To delete all the files and dir - `Git clean -f -d`
- e. To delete all the ignored file in one go - `Git clean -f -x`
- f. To delete all the files and dir including ignored files - `Git clean -f -d -x`
- g. To delete a specific file - `git clean -i >> option 4 >> use yes or no for selecting right file for deletion.`

20) RM Command

- a. Removes the file from the git and the file system - `git rm -f <filename>`
- b. Removes the file only from git repo but actual file will still be there in the project –
`git rm --cached <filename>`

21) Reset Command

- a. Move the file to Staging area= `git reset --soft <commit-id>`
- b. Move the file to unstaged area= `git reset --mixed <commit-id>`
- c. Move the files out of records or reverts the file to the stage of previous committed stage `git reset --hard <commit-id>`

22) Revert Command

- a. `git revert <commitID>`
- b. Note:-- when you want to revert from 5th commit to 3rd commit then pass the commit id of 4th Commit to revert to the 3rd commit stage.

23) Amend Command

- a. `git commit --amend -m "message"`

24) Tag Command

- a. Annotated tag= `git tag -a <tagname> <commit-id> -m "Message here"`
- b. Lightweight tag= `git tag <tag name> <commit-id>`
- c. Get the list of tags= `git tag --list`
- d. Get the list of tags with their messages= `git tag -n`
- e. Delete a tag= `git tag -d <tagname>`
- f. View a specific tag= `git show <tagname>`
- g. Pushing tags to remote repo= `git push --tags`
- h. Delete tag from remote= `git push --delete origin <tagname>`

25) Stashing Commands

- a. To stash all the files – git stash
- b. To show the list of files stashed – git stash list
- c. To show the latest stash or files stashed at 0th position – git stash show
- d. To show the stashed records at position 1 – git stash show {1} or use git stash show <stash@{0}>.
- e. To stash with a message – Git stash save “message”
- f. To get the files back to work:- Git stash pop
- g. To get back the stashed files at position 1:- git stash pop {1}
- h. This will remove the files from the position 1 and it will also delete the change completely and u cannot undo this:- git stash drop {1} / {stash-id}
- i. Remove all the stashed files in one go:- git stash clear
- j. Stash Show more details:- git stash show -p
- k. Stashing a single file:- git stash -- <filename>

26) Git Branching

- a. I want to see the branches available in this repository then:- Git branch
- b. I want to see all the branches including local and remote branches:- Git branch -a
- c. I want to see all the branches and their last/latest commits along with their message:- Git branch -v
- d. Delete a branch:- Git branch <branchname> -d
- e. Rename a branch:- Git branch -m <oldName> <newName>
- f. If you are on the branch you want to rename:- git branch -m new-name
- g. to check the parent of a branch:- git log -g <branch name>
- h. To view only the remote branch list:- git branch -r
- i. To create a branch and move inside the newly created branch in one go:- git checkout -b <branch name>
- j. Change the default branch:- git push --set-upstream origin <branchname>

27) Git Merging

- a. To merge the branch:- git merge <branch name>

28) Git rebase

- a. To rebase the branch from current parent to new parent commit:- git rebase <branch name>