Introduction

In this article, we will see what are the list of commonly used commands in . These are the commands which are really useful while working on any project. Hope this will be useful.

1) To initialize GitHub Repository in your local machine project folder:

git init

2) To get the status of files:

git status

3) To configure Username and Password:

git config –global user.name “user-name”

git config –global user.email “email-id”

4) To clone the repository to your local machine:

git clone URL

Note: Here URL is GitHub Repository URL

5) To add a single file to GitHub:

git add file-name

6) To add all the modified file to GitHub:

git add .

git add -A

7) To commit the changes to GitHub:

git commit -m “commit-message”

8) Get the latest code from the main branch:

git pull

9) To pull the latest code from a particular branch:

git pull branch-name

10) To list all the branches including local and remote:

git branch -a

11) To create a new branch name:

git branch branch-name

12) To delete a branch:

git branch -d branch-name

13) Delete remote branch:

git push origin --delete branch-name

14) To create a new branch and switch to the new branch created:

git checkout -b branch-name

15) Rename branch (local):

git branch -m old-branch-name new-branch-name

16) To switch a branch:

git checkout branch-name

17) To discard changes of a particular file:

git checkout -- file-name.txt

18) Merging branch (active branch):

git merge branch-name

19) Merging to a target branch:

git merge source-branch target-branch

20) Stash your changes:

git stash

21) Remove stash:

git stash clear

22) Show difference:

git diff source-branch target-branch

23) To view the log/changes:

git log

24) Detailed view of log/changes:

git log --summary

25) Brief view of log/changes:

git log --online

26) Get the help:

git help

27) To go back to the previous commit/changes:

git reset --hard

28) List stashed files:

git stash list

29) Come out of stash (write working from the top of stash stack):

git stash pop

30) Discard stashed changes:

git stash drop

31) Going back to HEAD:

git reset --soft HEAD

32) Going back to the commit before HEAD:

git reset --soft HEAD^

33) Equivalent to "^":

git reset --soft HEAD~1

34) Going back two commits before HEAD:

git reset --soft HEAD~2

35) To see the list of available tags:

git checkout v0.0.1

36) To set the current tag to v0.0.1:

git tag -a v0.0.3 -m “version 0.0.3”

37) To create a new tag:

git push –tags

38) To delete the file from your working directory:

git rm file-name

39) To show the metadata and content changes of the specified commit:

git show commit-name

40) To delete a branch:

git branch -d branch-name

41) To delete a branch forcefully:

git branch -D branch-name

Note: This will force deletion of the branch, even if it contains unmerged/unpushed commits.

42) Search the working directory for "add()":

git grep "add()"

43) List of branches:

git branch

git branch --list

44) Undo the changes:

git log --oneline

git revert commit-id

45) To delete a file forcefully:

git rm -rf file-name

To get the version:

git --version

How to add a project to GitHub Repository?

* Create a project on your local machine.
* Create a GitHub Repository
* Open the Bash go to the path where the project folder exists and type these commands,
* git init
* git remote add origin <https://github.com/username/repository-name.git>
* [Note : Make sure you are in correct repository using : git remote -v]
* git add .
* git commit -m "commit-message”
* git push origin master --force

[provide your username and password]