Cmd: git --version

# to check the git version in local system

Set Config Values

$ git config --global user.name “Prabha1909”

$ git config --global user.email [prabha.svm@gmail.com](mailto:prabha.svm@gmail.com)

$ git config --list

Need help?

$ git verb --help

$git help verb

$git config --help

$git help config

Getting Started…

Two scenarios – 1. Existing project that we want to track using git or the project which is remotely…

Initializing a repository from existing code

$ git init

Command to stop tracking using git we have to use the below command

$ rm –rf .git # in linux and mac

$ del .git # in windows

Before our first commit let us check the status of git

$ git status

And before committing a file we should add them

$ git add –A # to all the file in the directory to the staging area

$ git add filename # to add the particular file before commit

To ignore a particular file we can add those to gitignore so that they will not be visible to others.

$ touch .gitignore

To remove the files in the staging area

$ git reset #to remove all the files from staging area

$ git reset filename # to remove only the particular file from the staging area

To see the history of the commit we can use the below command

$ git log

Cloning a Remote repository

$ git clone <url> where to clone

Viewing information about the remote repo

$ git remote –v

$ git branch –a #list all the branches in our repo

Pushing Changes

$ git diff

Steps – Summarizing

1. Creating a branch

$ git branch <branch name>

1. Checking out the branch from master

$ git checkout <branch name>

1. Make the changes to the file and check the status

$ git status

1. Then followed by add and commiting the changes

$ git add –A

$ git commit –m “message name” #By doing so we commit the changes to the local branch repo…

1. Next step is to push the changes made from local repo to remote repo(branch)

$ git push –u origin <branch name>

1. Now, if everything works in branch, we need to merge it to master. Thus, checking out the master.

$ git checkout master

1. To make sure that no changes were made to the remote master. We need to first pull the master remote repo. If no changes found we are good to merge the branch.

$ git pull origin master

$ git merge <branch name>

1. Now, we are good to push the changes to master in the remote repo

$ git push origin master

Now, we have pushed the changes to the master in remote repo. We can delete the master in local as well as remote repo

* To delete the local repo below command is useful

$ git branch –d <branch name>

* To delete the remote repo use the below command

$ git push origin --delete <branch name>