GIT MIXED COMMAND

**GIT-INIT**

To create a blank repository

$git init

**GIT-STATUS**

To check file status Tracked or untracked

$git status

(OR)

$git status -vv

**GIT-ADD**

To add untracked files to staged area

$git add <file-name>

If multiple files

$git add **.**

**Stage deleted files**

$git rm <filename>

**To delete the file from git without removing it from disk, use the –cached flag**

$git rm –cached filename

**GIT-COMMIT**

To move files from staged area to local repository

$git commit -m “<commit-message”>

**HOW TO SELECTIVELY PICK UP THE COMMIT’S FROM CHILD BRANCH**

$git cherry-pick <commit-ID/message>

**GIT-RESTORE**

To move file from stage to working directory (i.e untrack file)

$git restore –staged <file>

**GIT-LOG**

If you what to know how many commits done

View the history of commits for the repository:

$git log

To know last 2 commits

$git log -2

**GIT-LOG-ONELINE**

* Will show all of commits with only the first part of the hash and commit messages.
* Each commit will be in a single line, which is usefull if we are looking at a lot of commits

$git log –-oneline

$git log -2 --oneline

**GITCONFIG**

* 1. $git config --global user.name “Username”
  2. $git config --global email “email-id”
  3. $git config --global --list
  4. $git config --global –edit
  5. $git config --global --unset-all user.name

**GIT-LOG-GREP**

To pick commit based on commit message

$ git log --grep <” Any word of committing message”> ‘

$ git log --all --grep < “Search Content”>

Will show all commits that do not contain "*add file*"

$ git log --grep =” add file” --invert-grep

$ git log --after ‘2 days ago’

$ git log --after YYYY-MM-DD

$ git log --author=author

Log showing committed files

$ git log --stat

**GIT log branch level**

$git log <branch-name>

**GIT-SHOW**

To see content of particular commit

$git show <Commit-ID>

**GIT-DIFF**

To know difference between two branches

To view difference between branches

$git diff <Baranch-name1><Branch-name2>

To view difference between commit-ids

$git diff <commit-id> <commit-id>

To view summary of changes

$git diff --stat <branch/commidID>

$git diff

To show all staged and upstaged file

$git diff HEAD

$git diff –help

$git help status

**GIT-SHORTLOG**

* List all contributions grouped by author name

git shortlog summarizes git log and groups by author

$git shortlog

* To simply see number of commits and suppress the commit description, pass in the summary option

$git shortlog -s

$git shortlog -e

Show the possible options for the status command in command line:

$ git status -help

Show all git possible commands in command line:

$ git help –all

Stage all new, modified, and deleted files. Use the shorthand command:

$ git add -A

Check the compact version of the status for repository:

$ git status --short

**To find out who changed a file**

$ git blame <file-name>

**Shows the author email and commit per line of specified**

$ git blame -e <file-name>