**GIT COMMANDS:**

**1.GIT INIT:**

# git init

<create new git repository>

# git init --help

**2.GIT ADD:**

# git add <file name>

<file goes to staged/index state>

# git add .

<all current directory files goes to staged/index state>

# git add <filename> <filename>…….

#git add --all

**3.GIT COMMIT:**

# git commit

< open default editor>

# git commit -m “write comment”

<staged file/s goes to local repository>

# git commit -am “write comment”

<all tracked file/s goes local repository with comment>

**4.GIT CONFIG:**

# git config –global user.name “write user name”

<configure user name>

# git config –global user.email “write user email”

<configure user email>

# git config –global user.name

<print user name>

# git config –global user.email

<print email>

# git config --list

**5.GIT REMOTE:**

# git remote add origin <url of repository>

<configure remote repository url>

# git remote add origin master

**6.GIT PUSH:**

# git push -u origin/master

<push repository from local to remote>

# git push –delete origin <branch name>

<deleted remote branch from local>

# git push

<push repository if already exists remote repository>

**7.GIT PULL:**

# git pull origin master

<pull the latest changes from remote>

# git pull

<pull the latest changes from remote if already exists remote>

**8.GIT BRANCH:**

# git branch

<print the branches>

# git branch -r

<print the remote branches>

# git branch -a

<print the local and remote branches>

# git branch <branch name>

<created new branch>

# git branch -d <branch name>

<deleted branch if already merged>

# git branch -D <branch name>

<deleted branch irrespective of merged>

# git checkout -b <branch name>

<create and switch new branch>

# git branch -m <exist branch> <new branch>

<rename the branch>

#git branch –no-merged

<print the non-merged branches>

# git branch –merged

<print the merged branches>

# git branch -v

<print remote active users>

**9.GIT CHECKOUT:**

# git checkout <file name>

<discards file>

# git checkout <commit id>

<discards commits >

# git checkout <branch name>

<switch the new branch>

# git checkout -b <branch name>

<create and switch to new branch>

**10.GIT MERGE:**

# git merge <branch name>

<merge the changes to current branch>

# git merge <branch name1> <branch name2> …

<merge multiple branches to current branch>

**11.GIT CLONE:**

# git clone <url of repository>

<clone the copy from remote repo>

**12.GIT DIFF:**

# git diff

<print the changes>

# git diff <file1> <file2>

<print the changes between file1 and file2>

# git diff <commit id1> <commit id2>

<print the changes between commit1 and commit2>

# git diff <branch name1> <branch name2>

<print the changes between branch1 and branch2>

# git diff –staged

# git diff –HEAD

**13.GIT RESET:**

# git reset –soft <commit id>

<bring back respective commit from local repo to staged/index state>

# git reset –mixed <commit id>

<bring back the respective commit from local repo to working directory>

# git reset

<same as “git reset –mixed “>

# git reset –hard <commit id>

<discards from working directory>

# git reset <commit id>

**14.GIT REVERT:**

# git revert <commit id>

<same as “git reset” but rewrite new base commit history>

**15.GIT FETCH:**

# git fetch origin/master

<downloads the commits from remote repo>

**16.GIT FORK:**

# git fork origin/master

<fork the exact copy from another upstream repo>

**17.GIT STATUS:**

# git status

<print the status of the files like staged and unstaged>

**18.GIT LOG:**

# git log

<print the commit history>

# git log -p -3

<print the only top of 3 commits history>

# git log –oneline

<print the commit history in one line>

# git log –summary

<print the commit history with clearly>

**19.GIT SHOW:**

# git show <commit id>

# git show <file name>

**20.GIT TAG:**

# git tag

# git tag -a <tag name> <commit id>

# git tag -a <tag name> -m <message>

# git tag -n

# git tag -l v1\*

# git tag -l v2\*

# git tag -l –sort=refname

# git tag -l –sort=refname:version

# git ls -remote tag origin

# git tag describe

**21.GIT CHERRY-PICK:**

# git cherry-pick <commit id>

**22.GIT STASH:**

# git stash

# git stash list

# git stash apply

# git stash pop

# git stash drop

# git stash clear

**23.GIT CLEAN:**

# git clean

# git clean -n

# git clean -dn

# git clean -df

**24.GIT REFLOG:**

# git reflog

# git reflog –show

# git checkout -b <branch name> <commit id>

**25.GIT RM:**

# git rm <file1> <file2>

# git rm –cached <file>

# git rm -r <folder**>**