**1. git config**

command: git config –global user.name “[name]”  
command: git config –global user.email “[email address]”  
The command sets the author's email and name address respectively to be used with your commits.

**2. git init**

command: git init [repository name]  
The command is used to start a new repository.

**3. git clone**

command: git clone [url]  
The command is used to obtain a repository from an existing URL.

**4. git add**

command: git add [file]  
The command adds a file to the staging area.  
command: git add \*  
The command adds one or more to the staging area.

**5. git commit**

command: git commit -m “[ Type in the commit message]”  
The command records or snapshots the file permanently in the version history.  
command: git commit -a  
The command commits any files you’ve added with the git add command and also commits any files you’ve changed since then.

**6. git diff**

command: git diff  
The command shows the file differences which are not yet staged.  
command: git diff –staged  
The command shows the differences between the files in the staging area and the latest version present.

command: git diff [first branch] [second branch]  
The command shows the differences between the two branches mentioned.

**7. git status**

command: git status  
The command lists all the files that have to be committed.

**8. git rm**

command: git rm [file]  
The command deletes the file from your working directory and stages the deletion.

**9. git log**

command: git log  
The command is used to list the version history for the current branch.  
command: git log –follow [file]  
The command lists the version history for a file, including the renaming of files also.

git log - -oneline  
Use: This command will helps to display log in one line.  
git log –grep “message\_word”  
Use: This command will help get commit id if you forgot the commit id.

**10. git show**

command: git show [commit]  
The command shows the metadata and content changes of the specified commit.

**11. git tag**

command: git tag [commitID]  
The command is used to give tags to the specified commit.

**12. git branch**

command: git branch

The command lists all the local branches in the current repository.  
command: git branch [branch name]  
The command creates a new branch.  
command: git branch -d [branch name]  
The command deletes the feature branch.

git branch -D <branch\_name>  
This command is helpfull to delete branch forcefully

**13. git checkout**

command: git checkout [branch name]  
The command is used to switch from one branch to another.

command: git checkout -b [branch name]  
The command creates a new branch and also switches to it.

**14. git merge**

command: git merge [branch name]  
The command merges the specified branch’s history into the current branch.

**12. git remote**

command: git remote add [variable name] [Remote Server Link]  
The command is used to connect your local repository to the remote server.

**13. git push**

command: git push [variable name] master  
The command sends the committed changes of the master branch to your remote repository.  
command: git push [variable name] [branch]  
The command sends the branch commits to your remote repository.  
command: git push –all [variable name]  
The command pushes all branches to your remote repository.  
command: git push [variable name] :[branch name]  
The command deletes a branch on your remote repository.

**14. git pull**

command: git pull [Repository Link]  
The command fetches and merges changes on the remote server to your working directory.

**15. git stash**

command: git stash save  
The command temporarily stores all the modified tracked files.  
command: git stash pop  
The command restores the most recently stashed files.  
command: git stash list  
The command lists all stashed changesets.  
command: git stash drop  
The command discards the most recently stashed changeset.

**16. git reset**

command: git reset [file\_name]

Use: The command is used to undo local changes to the staging area  
git reset \*

Use: The command is used to undo all the local changes to the staging area

**17. git revert**

command: git revert <commit id>  
Use: The command is helpful to undo the exiting commit

**18. git clean**

command: git clean -n (dry run)  
Use: This command will confirm before removing untracked files  
git clean -f (force fully)

Use: This command will remove all the untracked files

**19. adding Tags**

git tag -a <tag\_name> -m <message> <commit-id>  
Use: This command is allow to give meaning fill name to a specific version in the repository  
git tag  
Use: to see the list of tags  
git show<tag\_name>  
Use: To see particular commit content by using tag  
git tag -d <tag\_name>  
Use: to delete the tag