git clone <repository url>

git status

**# 13**

git add <file> // stage the file

git commit –m “<message>”

git restore –staged <file> // unstage given file

git push origin main // origin => remote repo. & main=> default branch in origin

**# 22**

git init <project name>

git commit // open default editor to write comment

#23

git init // initialize git repository in current directory

git add . // stages all files in current directory

#25

git pull origin master

git push origin master

#26

git commit –am “<message>” // add + commit at once

git ls-files // list all tracked files (not important)

#28

git add . // recursive add

#29

git rest head <file> //unstage => reverse add command

git restore <file> // discard changes in working directory (backup last committed)

#30

mv <file name> <new file name>; git add . // long way

git mv <file name> <new file name> //short

#31

git rm <file> == rm <file>; git add –u; //delete

git rm –rf <dir> == rm –rf <dir>; git add –u; // delete all directory

git restore –-staged <file>; git restore <file> // recover

#32

git log

git log --abbrev-commit //short commit id

git log --oneline --graph –decorate //compress logs in one line

git log <start commit id> <end commit id>

git log --since “2 days ago”

git log -- <file> // history of a file

git log --follow -- <file> // list history beyond renames

git show <commit id/ref> //display details autherL..

#33

git config --global alias.hist "log --all --graph --decorate --online"

npp ~/.gitconfig == git config –global –e

#40 setup diff/merge tool

git config --global merge.tool p4merge

git config --global mergetool.p4merge.path "C:/Program Files/Perforce/p4merge.exe"  
git config --global mergetool.prompt false

git config --global diff.tool p4merge

git config --global difftool.p4merge.path "C:/Program Files/Perforce/p4merge.exe"

git config --global difftool.prompt false

#45 compare working directory and the staging area

git diff

git difftool

#46 compare working directory and git repository

git diff head

git difftool head

#47 compare staging area and the git repository

git diff --staged head

git difftool --staged head

#48 limited comparisons to one file

git diff -- <file>

#49 comparing between commits

git diff <commit id/ref> <another commit id/ref>

git difftool <commit id/ref> <another commit id/ref>

git diff head head^ // compare last commit with previous (^ == -1)

git difftool head head^

#50 comparing between local and remote master branches

git diff master origin/master

git difftool master origin/master

#51 Branching basics

git branch // list local branches

git branch –a // list all branches (local and remote)

git branch <new branch name> // create new branch

git checkout <branch name> //switch to new branch

git branch –m mynewbranch newbranch //rename (-m == move)

git brnahc –d newbranch //delete branch (-d == delete)

#52 Happy Path / Fast Forward Merges

git branch // list local branches

git checkout –b title-change // create new branch and checkout in one step

git diff master title-change // compare two brnches

git merge title-change // **fast forward if no new commits in master**

#53 Happy Path / Disable Fast Forward Merges

git merge title-change –no-ff // **no fast forward / maintain temp branch details**