git --version. #

mkdir folder

git init

git config --global help.autocorrect 1

git config --global core.editor "vim"

git config --global user.name "leaddevops"

git config --global user.email "leaddevops@gmail.com"

git config --global --list

git config --list

touch file1

ls -ltr

clear

git status

git add file1

git status

git commit -m "add file1"

git status

touch file2 file3

git add file2

git status

git add .

git status

git commit -m "first commit"

git log

git log --oneline

git commit -a -m "file2 change"

rm file1

git rm file2

git status

git add .

git status

git commit -m "commit name"

git rm --cached file3

ls -ltr

git status

git commit -m "commit name"

fit status

git status

git diff HEAD~1..HEAD

vi .gitignore

git status

git add .

git commit -m "added gitignore"

git tag --a v1.4 -m "new change"

git status

git log

git tag --a v1.4 4b8b12a28de28dc33deea72406c9c92325ad2935 -m "new change"

git tag --a v1.5 4b8b12a28de28dc33deea72406c9c92325ad2935 -m "new change"

git log

git show 4084d6cbac9cc2b4eb5eb8bb22faddda9ddcc6a4

git show v1.5

Git clean -n --> to see unstated files

Git clean -f --> to remove

git clone https://github.com/leaddevops/edu1.git

ls -ltr

cd edu1/

ls -ltr

git remove

git remote -v

git remote -v

git remote add origin https://github.com/leaddevops/edu2.git

git remote -v

git push

git push --set-upstream origin master

clear

Branch and Merge

touch new1 new2

ls -ltr

vi new1

git stash

git status

git add .

git status

git stash

git status

clear

git statsh list

git stash list

git stash apply stash@{0}

touch new2 new3

git add .

clear

git status

git stash

git stash list

git stash pop

git stash list

git commit -m "sfdsf"

clear

git reset --hard HEAD~1

git revert HEAD

=====================================

cat ~/.gitconfig --> to see config details

git add, git status, git rm

check cat .git/index --> to see the index of new file added.

git status -s/-v -- short format / verbose format

git commit -a -m "msg" commint and staging

cat .git/COMMIT\_EDITMSG

Git creates a directory under .git/objects/ by taking first 2 digits of commit id.

.git/info/exclude ---> to see ignored files

Tags:

=====

git tag -a v1.1 -m "msg"

git tag --> to see all tags

git show v1.1

git tag -d v1.1

Rebase:

=======

Replay the changes made to one branch over the top of another.

git checkout newbranch

make changes, add commit

come back to master, git rebase newbranch

to make history of the project clean, use rebase.

git diff --summary cmt1 cmt2

git gc: The git garbage collection command cleans out old objects that can't be useful anymore and compresses contents of .git directory.

git gc --prune -- cleans objects older than 2 weeks.

git config gc.pruneexpire "15 days"

Logs:

======

git log

git log --since="4 days ago"

git log -S "bug " -- to search Specific Data

git log --stat -- summary of each commit

git log --pretty=format:"%h - %an - %ar - %s"

Clone:

======

git clone dir1 dir2 -- it makes another copy of local repo, this is local cloning.

git clone "remote URl"