git init

git config --global user.name "name"

git config --global user.email "email"

git config --list \* show config list

ls -a \* listing

touch file1 \* file adding

git status

git add file1 \*staging

git status

git commit -m "label" \* committing

git log \* check log for modifications

//once committing if we check log it diplays shawcode/iformation for commiited data

//we use shaw code for viewing changes

git show <commiited code> \* for viewing committed data

-------------

>>rcovering from commiitted/staging area<<

--------

git reset head <file name> (stahing to workspace)

\*// recover from staging/indeex area

git reset --soft <committed id> (committed/local repo to staging)

\*//recover from committed files.

git reset --mixed <cid>

\*// recover (local to working directry )

-------------------------------------------

>>>downoad github file to local repo<<

git clone <repo path> \*\*download from github

git push (for push code to github)

git pull (for local repo)

---------------------------------------------

>>LOG CHEKING cmds;;;;

git log

git log --oneline

git log --2 (for 2 last logs)

git log --oneline <n> (for number of logs in one line)

git log --auther=<name> (for getting auther log)

git log --auther=<name> --oneline -2 (filtering auter&listing)

git log --since=yyyy-mm-dd (for perticular from date)

git log --until=yyyy-mm-dd (for until date)

BRANCHING & MERGING

* To check available branches?

$git branch

* How to crate & change branch ?

$git branch<branch name>----for creating branch from master.

$git checkout <branch name>---for changing branch.

$git checkout –b<branc name>----for create and changing branches.

* Merging

Before merging we need change branch to master then merege .

$git merge <branch name>

* Deleing branches

Before we want to delete branch we change to other branch then delete the branch

$git branch –d<branch name >

$git branch –D<branch name> --force full deletion.

REMOVINIG&MOVING

* Add&commit at a single time?

$git commit –am “<commit name>”

* How to change committed message?

$git commit --amend –m ” <commit name>”

* How to remove file?

$rm <file>

* How to get deleted file?

$git checkout <file>--if its ther in any one of workd/stage/locrepo

* How to rename in git?

$git mv <file> <renaming file>

IGNORE & RESET FILE

* Git ignore:- to ignore un necessary files in project

Sys level---- /etc/gitignore

Project level--- $vi .gitignore // after commiting the ignore file..its ignores list .

* If we want to allow particular file from ignore list ?

$git add –f <file name> -- forcefull add

//\* HEAD : head is a pointer that’s indicates latest commits from the index list.

* How to erase history?

There is three ways to delete history

1)$git reset –soft <shaawone num>

Soft reset used for deleting only local reposistory history.

2).$git reset –mixed <shawone num>

Mixed reset is used for deleting both local repository & index/staging area

3).$git reset –hard <shawone num>

Hard reset is used for deleting from all local/index/working directorys.

**GIT STASH AREA**

* What is stash area?

Stash is temporary area under git for storing temporary files.

$git stash list—for viewing in stash

* How we use stash?

Conditions…..

1. The file must and should in staging/cache area

* What are the commands are used to store data in stash?

$git stash save <”message”>

* What are the commands used to copy data into branches?

$git stash apply <stash#> //\* copy from stash and pasting to staging area

$git stash pop <stash#> //\* cut from stash and pasting to staging area.

* How to delete stash area?

$git stash drop <stash#> // for deleting particular stash.

$git stash clear // for delete entire stash area.