UnDoing Changes

Undoing changes is a method of changing or undo a particular operation done in the git terminal, it includes undoing the commits, files and history of commits.

Commands used for Undoing :

1. git checkout
2. git reset
3. git revert
4. git rm

1. git checkout :

git checkout is used to discard changes done to a file in the working directory.

Syntax : git checkout filename.

(Or)

To discard changes of all the files in working directory

git checkout .

2. git reset :

- To Unstage the files from staging area to working directory

  Syntax : git reset filename

- To delete a commit and get back the file from Local repository to Staging area.

  Syntax : git reset --soft HEAD~1

- To delete a commit and get back the file from Local repository to Working Directory.

  Syntax : git reset --mixed HEAD~1 (Or) git reset HEAD~1

- To delete a commit and also delete the file permanently
  Syntax : git reset --hard HEAD~1

Note 1 : git reset is used only for local undoing changes and git reset will not store the history of the deleted commits. So, after using reset we cannot able to send out files to remote repository without the history of commits.

Note 2 : git reset will only delete a current commit or a sequence of commits. It is not possible to delete a particular commit in the history

3. git revert :

This command will delete a particular commit and saves the deleted history of the commit.

Syntax : git revert commit-Id

After using revert if we want to delete the same file in remote repository -----→ git push alias branchname

4. git rm :

To delete a particular file.

Syntax : git rm filename

Copying Changes from Remote Repository to Local Repository

1. git fetch :

It will copy the changes from Remote repository and save it to the Local repository as a hidden branch. If we want to access the hidden branch just use the command git checkout branch name.

Syntax : git fetch alias (or) git fetch branchUrl branch name

2. **git pull :**

   It will copy the changes from Remote repository and merge the changes into Local repository.

   Syntax : git pull alias branchname

3. **git clone :**

   It will copy the remote repository to your system by creating a folder with the same name as remote repository.

   Syntax : git clone Url of remote repo

Few more Commands in Git

1. **git stash** ----> it will save the files in staging area to a temporary location.
2. **git stash --list ----->** To see the stashed files
3. **git stash apply ----->** To get back the stashed files to staging area.
4. **git stash drop ----->** To delete current stashed file
5. **git stash clear ----->** To delete all the stashed files
6. **git log --oneline ----->** Alternate for git log, git log –online will show single line description of commits history
7. **git diff ----->** To see the difference between the branches, commits etc.,

   Syntax : git diff <commit-Id of any file> < commit-Id of different file.

   (Or)

   git diff <branch 1 > <branch 2>