Git Reset

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

This is process of toggling between mutiple commits

Git stores the code in the form of multiple versions

and it allows us to jum between any version depending

on requirement

Reset can be done in 3 ways

1 Hard reset

2 Soft Reset

3 Mixed reset

Hard Reset

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

Here the HEAD will point to an older commit and we will be able

see the data as it was present at the time of the older commit

1 Create a file and store some data

vim file1

Go into insert mode by pressing i

Add a word "one"

To save and quit Esc :wq Enter

2 Send this file into stagging and local repository

git add .

git commit -m "a"

3 Open the same file1 and make some more changes

vim file1

one

two

Save and quit

4 Send this file into stagging and local repository

git add .

git commit -m "b"

5 Open the same file and add some more data

vim file1

one

two

three

Save and quit

6 Send this file into stagging and local repository

git add .

git commit -m "c"

7 Check the commit history

git log --oneline

8 To hard reset from c to b commit

git reset --HARD b\_commitid

9 Open the file1 and we should see the data as it was present at b commit

cat file1

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

Soft reset

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

This will take the repository one step back it if we reset from c to b

commit then HEAD will point to b commit but the files will be shown as they

are present in the stagging area.ie the status of git just before the c commit happened

git reset --soft b\_commitid

Mixed Reset

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

This will take the repository two steps back it if we reset from c to b

commit then HEAD will point to b commit but the files will be shown as they

are present in the untracked/modified area.

git reset --mixed b\_commitid