UNDOING CHANGES

>> Undoing: Reverse the doing/changes.

Types of Undoing changes:

1. checkout/restore
2. reset
3. revert

1. Checkout/restore

>> Discards the made in the Working Directory & restores the state of file from the last state of commit/staged, in the Working Directory itself.

>> Discards the unstaged changes.

>> Changes should not be staged.

-> git restore file-name
-> git checkout file-name

2. Reset:

>> It is a powerful tool in GIT, as it works both in Staging Area as well as the Local Repository.

Undoing changes in Staging Area

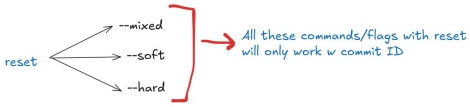
-> git reset file-name

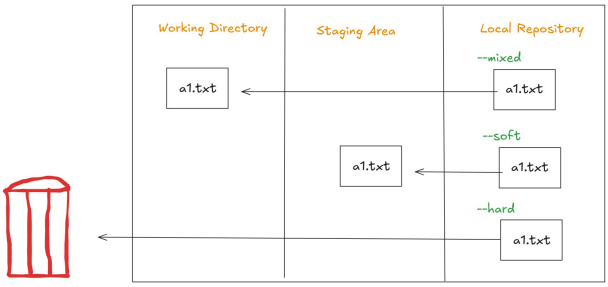
-- Moves the files from Staging area back to the Working Directory.
-- Removes the files from the Staging Area & unstages them.

-> git reset .

-- Removes all the files from the Staging Area & unstages them.

Undoing changes in Local Repository





1. git reset --mixed <commit-id>:

--> Undoing changes by sending the file from the Local Repository back to the Working Directory.

--> Undo the commit & unstages the files AND send the files in the Working Directory.

--> Whatever files ae related to a particular commit, it will effected.

2. git reset --soft <commit-id>:

--> Undoing changes by sending the file from the Local Repository back to the Staging Area.

--> Undo the commit but keeps the files staged in the Staging Area.

--> Whatever files ae related to a particular commit, it will effected.

3. git reset --hard <commit-id>:

--> Undoing changes by sending the file from the Local Repository back to directly deleting the file/changes/versions from the git initialized folder itself.

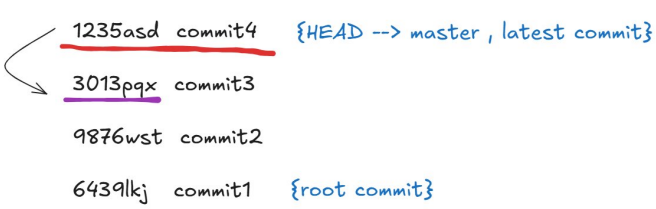
--> Completely erases the files with all the commits & changes.

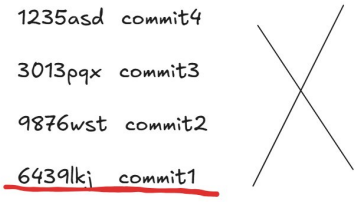
--> Whatever files ae related to a particular commit, it will effected.

NOTE :

1. To reverse any particular commit, we need to use the commit ID of the previous commit.
2. We cannot make changes to the 1st commit of the directory(root commit) using commit ID, because there are no prior commits to that.
3. If we try to make changes in a particular commit, along with that commit, all the above commits will also get effected.

Commit Structue:





3. Revert:

--> Undoing changes to the commits in the Local Repository.

--> It will undo the changes to the specific file associated to that commit ID BUT, it will not delete/remove the commit message from commit history.

--> While reverting a specific commit, use the same commit ID.

-> git revert <commit-id>:

> An editor will open, and you have to specify a 'commit-delete' msg.

--> After the undoing of changes with revert, 1 extra commit will be added which notifies that, changes are done.

TASK

1. Create 4 files a1.txt,a2.txt,a3.txt & a4.txt and add it to the staging area.
2. Try to edit a1.txt and discard the unstaged changes.
3. Commit all the files
4. Make changes to a4.txt using --soft.
5. Commit the a4 file.
6. Make changes to a3.txt using --mixed.
7. Add all the files again to staging area and commit
8. Make changes to a2.txt using --hard.