

10 May 2014 on [git](#) | [changes](#) | [remove](#) | [commits](#) | [branches](#) | [unsynced](#) | [revert](#) | [reset](#) | [hard](#) | [head](#)

How to remove changes from Git

<http://www.ainixon.me/how-to-remove-changes-from-git/>

We work everyday on git. Sometimes we make some accidental move like commit unwanted codes, merge wrong branch with the master, delete a commit accidentally and so more. Recently I have faced similar kind of problems. For finding the solutions I have to check all the cheatsheets. Let's look at some of the issue and it's solutions

Remove uncommitted changes:

If you make changes or you get a conflict in multiple files and you want to remove those changes then you should use

git reset --hard

Careful: It will *delete* your working directory changes.

Remove an unsynced commit:

- If you adds a commit, you want to remove it before syncing it...

git reset --hard HEAD~1

It will remove the last commit in other words it will remove the last commit from the head.

- Now if you want to remove **n** numbers of commit then....

git reset --hard HEAD~n

By running this command you will remove the **n** of commits from the head.

- If you want to remove a specific commit not from the head then you have to run a this command and find the id for that commit

git log

Using that *id* if you run this command

git reset --hard sha1-commit-id

This command will remove the specific commit(of the id) from current branch

Undo delete action of a commit:

If you delete a commit accidentally then to restore that commit you can use

git reflog

Remove last synced commit:

To remove last synced commit you can use two methods,

- Reverting the last commit. For this you need to run..

git revert

- Force push to remove the last commit.

git push origin HEAD --force

But, if others have pulled it, it would be better for you to start a new branch. Because when they pull that, it will just merge it into their branches, and you will get it pushed back again.