**Undo a commit & redo**

<https://stackoverflow.com/questions/927358/how-do-i-undo-the-most-recent-local-commits-in-git>

$ git commit -m "Something terribly misguided" # (0: Your Accident)

$ git reset HEAD~ # (1)

[ edit files as necessary ] # (2)

$ git add . # (3)

$ git commit -c ORIG\_HEAD # (4)

1. This command is responsible for the **undo**. It will undo your last commit while **leaving your working tree (the state of your files on disk) untouched.** You'll need to add them again before you can commit them again).
2. Make corrections to working tree files.
3. git add anything that you want to include in your new commit.
4. Commit the changes, reusing the old commit message. reset copied the old head to .git/ORIG\_HEAD; commit with -c ORIG\_HEAD will open an editor, which initially contains the log message from the old commit and allows you to edit it. If you do not need to edit the message, you could use the -C option.

**Alternatively, to edit the previous commit (or just its commit message)**, commit --amend will add changes within the current index to the previous commit.

**To remove (not revert) a commit that has been pushed to the server**, rewriting history with git push origin master --force is necessary.