

# 12 Days of Git

DAY 6

REWRITING GIT HISTORY

# Rewriting History

## Change most recent Git commit message

There are times when you make a commit but realise you've written the wrong thing within the commit message. What do you do?

This is one of the use cases for the `--amend` option.

```
COMMIT
(base) → PyPi-connectdb git:(main) ✗ git commit --amend -m "updated last commit message"
[main 968cb72] updated last commit message
Author: Arjun Panwar <arjun.panwar@l
Date: Tue Nov 8 22:37:45 2022 +0530
1 file changed, 32 insertions(+)
create mode 100644 update.py
```

```
commit 968cb72afde456eac5c45cbc2611d4643e47101b (HEAD -> main)
Author: Arjun Panwar <arjun.panwar@l
Date: Tue Nov 8 22:37:45 2022 +0530

    updated last commit message
```

It will replace last message

# Add extra changes to a commit

There are times when you complete a commit and then realise you want to add in one more change or you've forgotten something. And it would make more sense to add it into that commit rather than open another one.

This is another use case for the **git commit --amend** command.

Your workflow might be:

1. Make changes to file 1 and file 2
2. Add and commit those changes
3. Realise you've forgotten to add a small change into file 1
4. Make the additional change
5. Use the command ***git commit --amend --no-edit***

The additional option on the command, **--no-edit** takes that last change and puts it into the previous commit, without changing the message. For anyone else looking at this commit, it will look like it was done in one commit.

***I would caution using this option and only using it on your own commits, don't confuse others by amending other people's commits***

For more visit:  
<https://www.techielass.com/rewriting-git-history/>



Other ways to rewriting history will be explored in comming days