



# Squashes the Commits

Learn how to squash commits.

We'll cover the following



- Squashing the squashed commits
- Pull the changed history

Now you (as Alice) are going to squash the commits like you did in the previous chapter

(<https://www.educative.io/collection/page/10370001/6075350136651776/6591426467463168>).

```
1 cd ../lgthw_bare_repo_alice
2 git rebase -i $(git rev-list --max-parents=0 HEAD) HEAD
```

Edit the final commit message so it looks like this:

```
commits:1-10
```

Now that it is done, you need to move the `master` branch pointer to the squashed commit:

```
3 git branch -f master
4 git checkout master
```

**Note:** In this lesson, you are only limited to one terminal.



Terminal



Click to Connect...

## Squashing the squashed commits #

What will happen if Alice pushes? Think about it, then guess:

```
5 git push origin master
```

Yup, it failed. Read the output carefully now, as you are *not* going to follow its advice.

The key phrase above is this one:

```
Updates were rejected because the tip of your current branch is behind  
its remote counterpart
```

But that's what you wanted! You wanted to be “behind” because you've squashed. You don't want to `git pull` as it advises because you want this history to override the previous history that is still stored on the remote.

How do you perform this override? Like this:

```
6 git push -f origin master
```



The `-f` flag forces the push onto the remote repository. This works now because the bare repository is configured to accept it.

This is great, right? Well, if you are Bob, not so much...

## Pull the changed history #

Bob comes into work and wants to get going with his tasks on this repository. First, he wants to get in sync with the remote `master` branch.

```
7 cd ../lgthw_bare_repo_bob
8 git status
```

Bob's a typical user and hasn't read this course, so he just does a `git pull` to pick up any recent changes from the remote.

```
9 git pull
```

Bob gets asked by Git to do a merge. Bob thinks that's strange, but he doesn't really understand what's going on and goes ahead and merges.

You should do as Bob would and accept the merge as it is. Done that?

Where are you now?

```
10 git log
```

Hmmm. It's a complete mess. You have all the commits you originally had. Plus, you have the squashed one that Alice pushed.

Bob doesn't notice that (he really should read this course, shouldn't he?) and pushes the resulting merge up so everything is in sync.

```
11 git push
```

He thinks that all looks good now and carries on with his work; safe in the

knowledge that all is in sync now.



```
12 cd ../lgthw_bare_repo_alice
13 git pull
```

Looks good to Alice, just a fast-forward. That's healthy, right?

```
14 git log
```

Oh, dear. Alice is now wondering what happened to her lovely squashed changes.

[← Back](#)[Next →](#)

Setting up a Bare Repository

Git Pull and Merge Hell



Mark as Completed



Report an  
Issue



Ask a Question

([https://discuss.educative.io/tag/squashes-the-commits\\_\\_bare-repositories\\_\\_learn-git-the-hard-way](https://discuss.educative.io/tag/squashes-the-commits__bare-repositories__learn-git-the-hard-way))