

# Git Pull and Merge Hell

Learn what to do when you find yourself stuck in the cycle of pulling and merging code.

We'll cover the following



- Squash on a branch and locally

What you experienced in the previous lesson is a *simplified* version of the pain that can arise when the history of a branch gets out of sync. Similar things can happen when rebasing changes between a point in the repository or cherry-picking.

If you find yourself doing merge after merge and it seems like you're in a mess, take a step back and think about what may have happened between your repository and the remote.

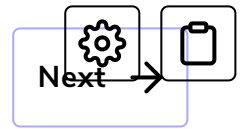
Understanding roughly why this happens is going to put you ahead of 95% of Git users, and can make you very popular with your colleagues when they are stuck.

## Squash on a branch and locally #

If there's a moral to all this, it is that you should squash on a branch and only locally (if possible).

It's yet another reason to take advantage of Git's cheap cost of branching.

If there's a second moral, it's to avoid force-pushing wherever other users are involved. Many Git products have options to forbid or only allow certain privileged users to force-push changes to repositories for this reason.

[← Back](#)[Squashes the Commits](#)[Introduction: Cherry-Picking and Thre...](#)[Mark as Completed](#)[Report an Issue](#)[Ask a Question](#)

([https://discuss.educative.io/tag/git-pull-and-merge-hell\\_\\_bare-repositories\\_\\_learn-git-the-hard-way](https://discuss.educative.io/tag/git-pull-and-merge-hell__bare-repositories__learn-git-the-hard-way))