

12 Days of Git

DAY 9
GIT REBASING

What is Git Rebasing?

Rebasing your Git repository rewrites its history. **It can be a harmful command**, so it is one to watch when you use it.

Typically, you would use the git rebase command for one of the following reasons:

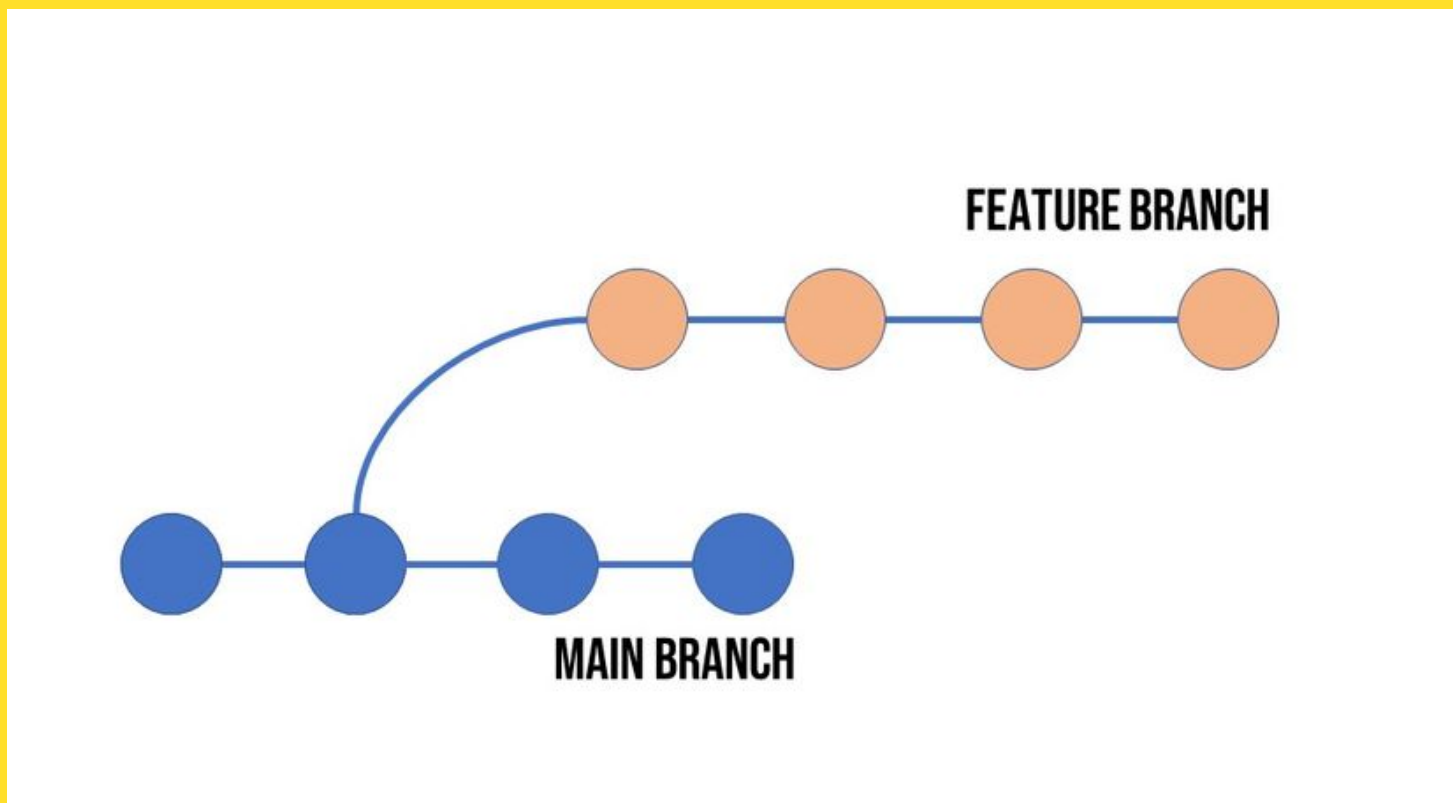
- Edit a previous commit message
- Combine multiple commits into one
- Delete or revert commits that are no longer necessary

Why do we need Git rebase?

Rebasing rewrites the project's history. It gives you a much cleaner project history. Rebasing eliminates the unnecessary merge commits required by git merge. It gives a much linear history when you are looking back at your logs.

Why do we need Git rebase?

We have a repository with the main branch and then a feature branch. We've been working away on different things on this feature branch and have several commits. If we look at the graph of this, it will look something along the lines of this.



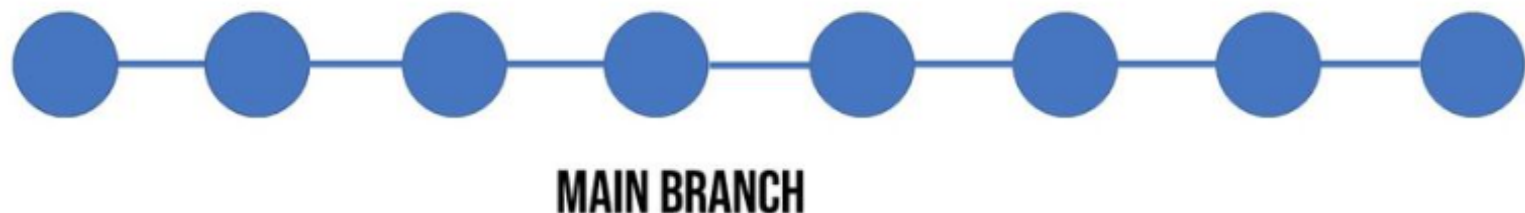
We want to bring all the commits and cool new features we've been creating from our feature branch into our main branch. But we want to make the history as linear as possible. So instead of doing a git merge we are going to use git rebase.

The commands we want to use are:

git checkout main
git rebase featurebranch

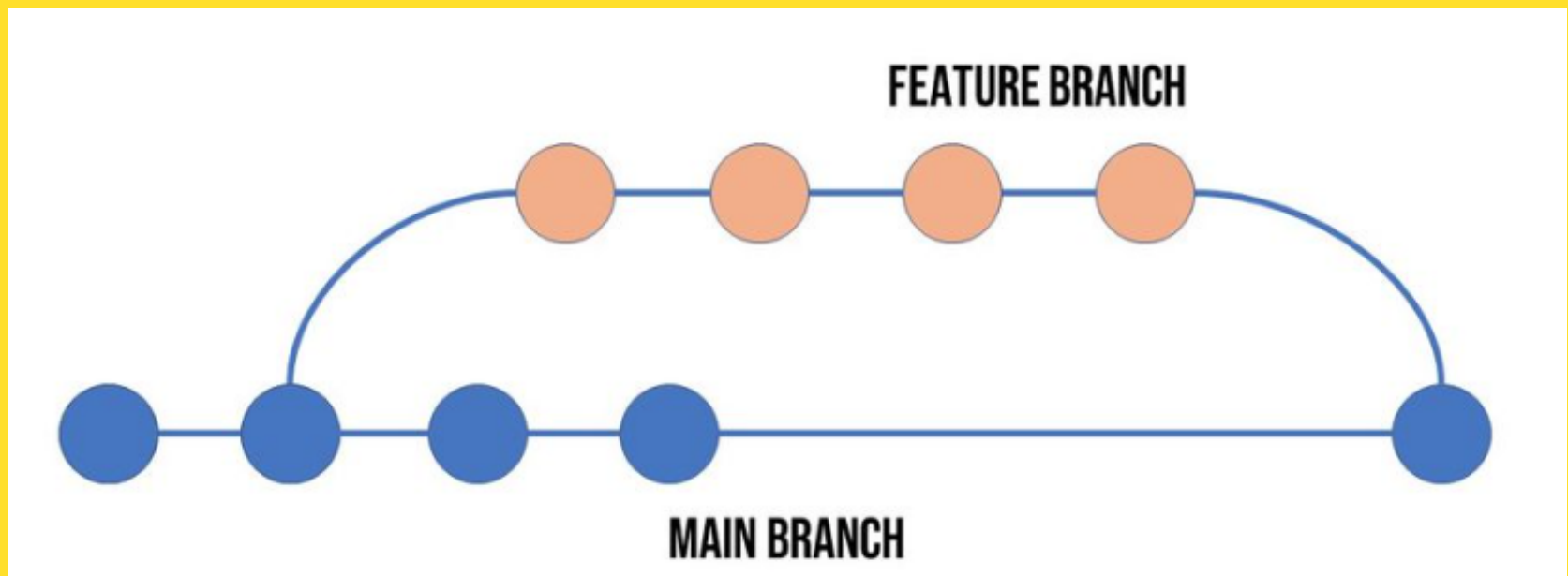
The first command makes sure we are in our main branch and the second command rebases that main branch to include the commits from feature.

Our git graph log would look like this:



You can see our history has been rebased into the main branch in a linear way.

If we'd used the git merge command instead this is what our repository history would look like:



Using the git rebase command gives you a much clearer history, especially if you have a large team of developers working on different features and bugs. It can become a real message if you are only using git merge. The history really becomes hard to read and understand.

Git rebase isn't always the right option though, as it does rewrite history. It's important as a team you understand the right use case for git rebase over git merge and use it appropriately.

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

