

# Git rebase

## What is Git Rebase

`git rebase` is one of the two strategies in git along with `git merge` that allows you to integrate changes from one branch onto another.

The differences between those two is that `git merge` is always a forward moving change record, meaning that the past history is never changed, new changes are tied onto the branch that you want to integrate changes to as a new commit.

On the other hand, `git rebase` is a history rewriting feature that allows you to basically base your branch on the newer commits.

### Branch background

So a little bit of background for branches, whenever you spawn off a branch it is based on a point in the commit history. Let's say you are on commit A of main, then you spawn off another branch called feature-a from main, then feature-a is based on commit A.

A branch is really nothing but a commit in the git history actually. So now when your co-worker pushes changes to main while you're working on feature-a that was based on old base (commit), you want to have those changes because you think it will help with your feature, you can use `git rebase` to change your base (the commit that your branch is spawned off from) to be from the latest commit of main.

Git tutorial: Git rebase

### Why git rebase

The main reason as to why you should use `git rebase` is to maintain a linear project history. If you use `git merge` to integrate changes from main it will result in a new commit in order to tie those new changes into the feature branch, which might look ugly to some minimalist.

Git rebase allows you to avoid the extra merge commit by changing the base commit that your branch is currently basing off from, then replay the commits that you had.

### How does it work

Well the command is simple is just

```
git rebase <base>
```

What this will do it will automatically rebases the current branch onto `<base>`, it will basically take all the commits that you have committed to the current branch so far, then change the branch so that it is basing off from the newer main, then REPLAY those commits that you had committed originally to the feature branch.

Important thing to note that replaying the commits will not result in the same commit hashes. The changes are still the same per commit, but the hash will not be the same as before because the parent of each commit HAS CHANGED!

---

Revision #1

Created 2023-04-01 13:15:51 UTC by Tamarine

Updated 2023-04-01 14:00:40 UTC by Tamarine