

# Git rebase local vs git pull --rebase origin

Asked 5 years, 4 months ago   Modified 2 years, 6 months ago   Viewed 11k times



10



I have a local (never pushed to a remote) feature branch **F1** made from **master** with a few commits. I don't seem to understand a difference between these two actions (**F1** is the current branch):

```
git fetch origin
git rebase master
```

and

```
git pull --rebase origin master
```

My expectation is that they should be equivalent but they aren't - different results are produced.

What's wrong with my thinking?

[git](#) [version-control](#) [git-rebase](#) [branching-and-merging](#)

Share Follow

edited Jul 4, 2017 at 12:57

asked Jul 4, 2017 at 12:51



UserControl

14.4k 20 98 182

2 Answers

Sorted by:

Highest score (default)



14



1. `git fetch origin` and `git rebase master` will apply changes from `F1` to local `master` branch. Assume your commit history looks like below at first (the remote `master` branch has commit `J` in remote):

```
A---B---C---D---E   master
                \
                F---G---H   F1
```

When you execute `git fetch origin` and `git rebase master`, even though `origin/master` points to `J`, it will only rebase `F1` branch on the top of local `master` branch (commit `E` as the graph):

```

A---B---C---D---E(master)---J origin/master
      \
      F---G---H   F1

```

2. The command `git pull --rebase origin master` will pull changes from remote `master` branch at first, then rebase current branch `F1` on the top of it:

```

A---B---C---D---E---J      master,origin/master
      \
      F---G---H   F1

```

In a word, if local `master` branch is sync with remote `master` branch, these two ways have the same result (rebase `F1` branch on the top of `master` branch). If remote `master` branch has new commit(s) which is(are) not exist on local `master` branch, the results are different (one rebases on local `master` branch, the other rebase on `origin/master` branch).

Share Follow

edited May 7, 2020 at 0:16



Joel Christophel

2,526 4 29 49

answered Jul 4, 2017 at 14:38



Marina Liu

35.8k 4 59 72

Fantastic explanation. I'd also suggest adding `--autostash` to automatically stash and pop local changes during this process. In total: `git pull --rebase --autostash origin master` – [odigity](#) Apr 21 at 16:34



`git fetch origin`

-1



only download remote files from a remote repository into your local branch.



`git rebase master`



takes the whole branch (the commits that are in your branch), and put on the top of your local repository. It changes the history of your project.

```
git pull --rebase origin master
```

`git pull` by default, performs a merge, but you can force it with a rebase option.

check this [link](#)

Share Follow

answered Jul 4, 2017 at 16:27



Guru\_Clef

366 2 16

