

<https://github.com/nvarabioff/learning-about-version-control/commits/main/>

```
(base) nicholasvarabioff@Nicholass-MacBook-Pro ~/Documents/UBC-MDS/03-DSCI_522/learning-about-version-control (main) 18:59
$ git push
To github.com:nvarabioff/learning-about-version-control.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'github.com:nvarabioff/learning-about-version-control.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

(base) nicholasvarabioff@Nicholass-MacBook-Pro ~/Documents/UBC-MDS/03-DSCI_522/learning-about-version-control (main) 18:59
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 984 bytes | 246.00 KiB/s, done.
From github.com:nvarabioff/learning-about-version-control
   debadfb..28f6b15  main      -> origin/main
hint: You have divergent branches and need to specify how to reconcile them.
hint: You can do so by running one of the following commands sometime before
hint: your next pull:
hint:
hint:   git config pull.rebase false  # merge
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only       # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
fatal: Need to specify how to reconcile divergent branches.

(base) nicholasvarabioff@Nicholass-MacBook-Pro ~/Documents/UBC-MDS/03-DSCI_522/learning-about-version-control (main) 19:01
$ git pull --rebase
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
error: could not apply 948aff9... update readme from local.
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
Could not apply 948aff9... update readme from local.

(base) nicholasvarabioff@Nicholass-MacBook-Pro ~/Documents/UBC-MDS/03-DSCI_522/learning-about-version-control (main *+IREBASE 1/1) 19:05
$ vim README.md
```

Figure 1: CLI1

```
# Learning about Version Control
Author: Nicholas Varabioff

Solving merge conflicts takes conscious effort, but I can do it!

<<<<<<< HEAD
Solving merge conflicts is easy!
=====
Solving merge conflicts is soooooo hard!
>>>>>>> 948aff9 (update readme from local.)
~
```

Figure 2: Vim1

```
# Learning about Version Control
Author: Nicholas Varabioff

Solving merge conflicts takes conscious effort, but I can do it!
~
~
~
```

Figure 3: Vim2

```

(base) nicholasvarabioff@Nicholass-MacBook-Pro ~/Documents/UBC-MDS/03-DSCI_522/learning-about-version-control (main *+IREBASE 1/1) 19:06
$ git add README.md

(base) nicholasvarabioff@Nicholass-MacBook-Pro ~/Documents/UBC-MDS/03-DSCI_522/learning-about-version-control (main +IREBASE 1/1) 19:07
$ git rebase --continue
[detached HEAD 964b36d] update readme from local.
1 file changed, 1 insertion(+), 1 deletion(-)
Successfully rebased and updated refs/heads/main.

(base) nicholasvarabioff@Nicholass-MacBook-Pro ~/Documents/UBC-MDS/03-DSCI_522/learning-about-version-control (main) 19:07
$ git
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

(base) nicholasvarabioff@Nicholass-MacBook-Pro ~/Documents/UBC-MDS/03-DSCI_522/learning-about-version-control (main) 19:07
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 386 bytes | 386.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:nvarabioff/learning-about-version-control.git
28f6b15..964b36d main -> main

(base) nicholasvarabioff@Nicholass-MacBook-Pro ~/Documents/UBC-MDS/03-DSCI_522/learning-about-version-control (main) 19:07
$

```

Figure 4: CLI2

It is useful to work in a branch-pull-request workflow instead of working by directly committing each change to the main branch for a few reasons. One, the main branch might be deployed in a production environment, so untested changes should not be applied directly to an environment that is shown to users. Another reason is that branches make it easier to solve specific problems, as you can relate a branch directly to a problem. Another useful features of branches is the pull request feature, which makes it easy to share changes with teammates, adding a layer of protection against merging code that might cause errors.