

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

```
(base) josed@JMDMYT25 ~/git/block3/522/learning-about-version-control (main)
$ git push
To github.com:josedmyt/learning-about-version-control.git
 ! [rejected]           main -> main (fetch first)
error: failed to push some refs to 'github.com:josedmyt/learning-about-version-control.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository
      pushing to
hint: the same ref. If you want to integrate the remote changes,
      use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for
      details.

(base) josed@JMDMYT25 ~/git/block3/522/learning-about-version-control (main)
$ git pull origin main
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), 975 bytes | 37.00 KiB/s, done.
From github.com:josedmyt/learning-about-version-control
 * branch            main      -> FETCH_HEAD
   cbdc6f7..acd59c0  main      -> origin/main
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result
.
```

The screenshot shows a GitHub commit interface for a file named 'README.md'. The commit has 5 changes. The commit message is: 't-version-control > README.md > abc # <<<<< HEAD Solving merge conflicts is sooooo hard!' It was made by 'You' 2 minutes ago and has 2 authors (You and one other). The commit content includes:

```
1 Learning about version control
2
3 <<<<< HEAD (Current Change)
4 Solving merge conflicts is sooooo hard!
5 =====
6 Solving merge conflicts is easy!
7 >>>> acd59c0a4d0ae9bf70867320a0c4c46e06a5f3b9 (Incoming Change)
8
9 🌟
10 Solving merge conflicts takes conscious effort, but I can do it!
```

The commit message and the last line of the commit content are highlighted in yellow, indicating they are part of the current branch ('HEAD'). The incoming changes ('Incoming Change') are highlighted in blue.

Working in a branch + pull-request workflow is useful because it keeps the main branch stable and ready while you work on new things. This is useful since you only merge code after it has been reviewed and tested, which helps causing issues to the project under the main branch. This also helps keep track of what has changed, any discussion related to the changes and who submitted and approved it. All this helps make collaboration with others much more organized and keeps from people breaking each others work.