

How to Use Git and GitHub

Lesson 2 > Merging on the Command Line

DASHBOARD

CLASSROOM

MATERIALS

DISCUSSION

OVERVIEW

Get Help

[Discussion Topics](#)

[Report an Issue](#)

Skip to Quiz

Instructor Notes

Checking out the coins branch

If you haven't already checked out the coins branch, you'll need to do so now with the command `git checkout coins` before you'll be able to refer to it. Once you've done that, decide whether you should keep it checked out or check out a different branch before completing the merge.

A note about `git merge`

`git merge` will also include the currently checked-out branch in the merged version. So if you have branch1 checked out, and you run `git merge branch2 branch3`, the merged version will combine branch1 as well as branch2 and branch3. That's because the branch1 label will update after you make the merge commit, so it's unlikely that you didn't want the changes from branch1 included in the merge. For this reason, you should always checkout one of the two branches you're planning on merging before doing the merge. Which one you should check out depends on which branch label you want to point to the new commit.

Since the checked-out branch is always included in the merge, you may have guessed that when you are merging

Downloadables

LESSON

Videos



Transcripts



two branches, you don't need to specify both of them as arguments to `git merge` on the command line. If you want to merge branch2 into branch1, you can simply `git checkout branch1` and then type `git merge branch2`. The only reason to type `git merge branch1 branch2` is if it helps you keep better mental track of which branches you are merging.

Also, since the two branches are merged, the order in which they are typed into the command line does not matter. The key is to remember that `git merge` always merges all the specified branches into the currently checked out branch, creating a new commit for that branch.

Merge conflict

If you get a message like this

```
Auto-merging game.js
CONFLICT (content): Merge conflict in game.js
Automatic merge failed; fix conflicts and then com
```

then your files were not in the same state as Caroline's when you started the merge. To fix this, complete the following steps:

1. Restore your files to their state before you started the merge by running `git merge --abort`
2. Double check the state of your files. If you run `git log` while the master branch is checked out, you should see Caroline's "Add color" commit as the second-most-recent, and the most recent should be your commit fixing the bullet bug. If you use `git diff` to compare your commit to Caroline's, your commit should introduce the line `this.delayBeforeBullet = 10;` on line 424. The line should be indented to the same level as the line below it using only spaces (no tabs), and the line should have no spaces after it.
3. Once your file is in the correct state, create a new commit with your changes.
4. Try the merge again.

Merge conflict (Newline characters between Windows and Unix systems)

Context: Whenever we hit the "Enter" key on the keyboard, we are actually telling the computer to insert an invisible character into our text file to indicate to the computer that there should be a new line. Unix systems adds one character called the "line feed" character or LF or `\n` while Windows systems adds two characters, "carriage return" and "line feed" or CRLF or `\r\n`.

Caroline's files have LF because her files were edited on Mac OSX, which uses LF. If a Windows user were to edit Caroline's files, the Windows text editor might convert all LF to CRLF to make editing files possible. When the Windows user merges her file with Caroline's files, a merge conflict will result due to the different LF and CRLF characters.

To fix this, Windows users should set the global autocrlf attribute to true: `git config --global core.autocrlf true`.

More information can be found here:

<https://help.github.com/articles/dealing-with-line-endings/#platform-all>

Comparing a commit to its parent

The command Caroline mentions to compare a commit to its parent is `git show commit_id`

INFORMATION

[Nanodegree Credentials](#)
[Georgia Tech Program](#)
[Udacity for Business](#)
[Udacity for Veterans](#)
[Help and FAQ](#)
[Feedback Program](#)

COMMUNITY

[Blog](#)
[News & Media](#)
[Developer API](#)

UDACITY

[About](#)
[Jobs](#)
[Contact Us](#)
[Legal](#)
[Service Status](#)

FOLLOW US ON

MOBILE APPS



Nanodegree is a trademark of
Udacity

