Feeling conflicted?

Avoiding and resolving conflicts in git and GitHub

What causes conflicts?

If you make changes to a file on your computer, and you (or someone else) makes changes to the version of that file that's on GitHub, and then you try to either "git pull" or "git push", you're going to encounter a conflict.

How to avoid conflicts

Always "git pull" your repo whenever you start working on it if:

- You made changes to a file directly on GitHub (like the README)
- You or others may have changed files since you last ran "git push"

When conflicts happen

If you "git pull" and there's a conflict, git will try to merge the files together. When you open up the conflicted file, you'll see sections that look like this:

<<<<<HEAD

code from GitHub version here

code from local version here

>>>>>>>>>>> *some long number*

To fix the conflict

You have to fix the code manually, by converting this:

<<<<<HEAD

code from GitHub version here

code from local version here

>>>>>>>>>>> *some long number*

To this:

combined version of code

...by deleting the lines with the <<<<<< and >>>> and any extraneous or overlapping code

Once you've fixed the conflict

Run:

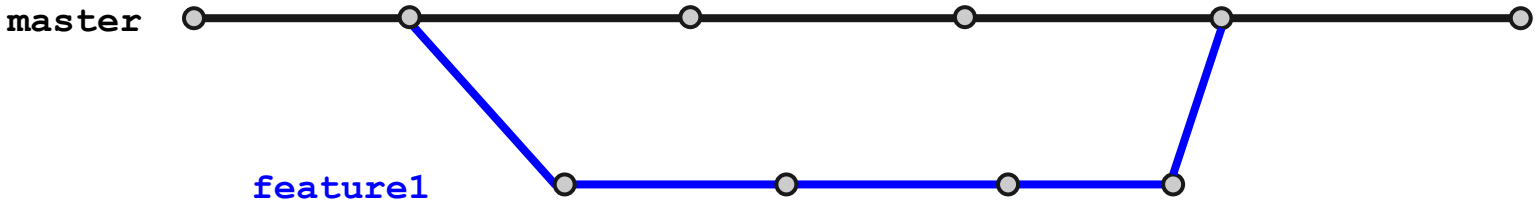
git commit -i *filename* -m "Commit message here."

This will merge the conflicts. Then you can "git push" to GitHub again and everything will be up-to-date.

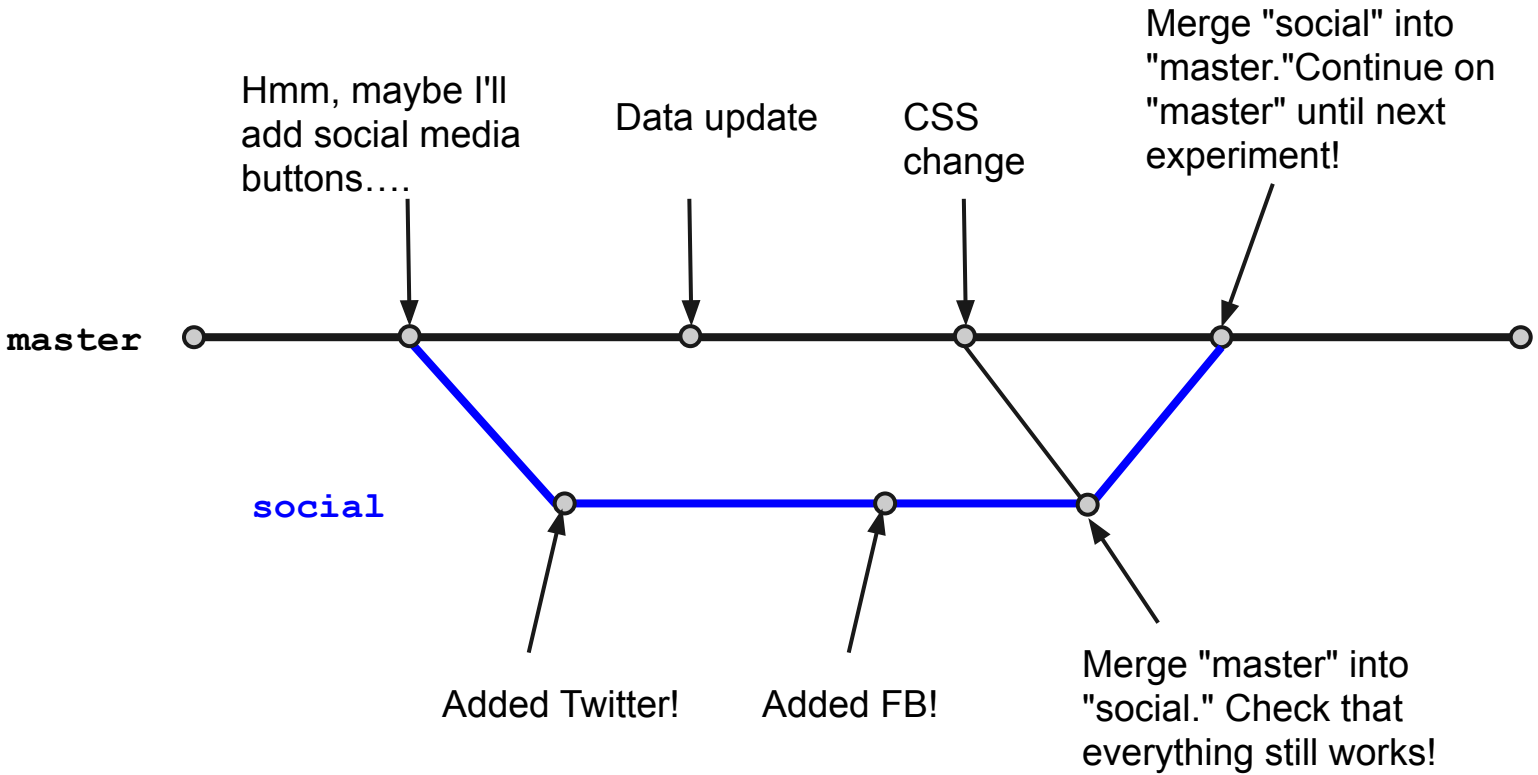
Branching & merging

Fail-safe experimentation

You create a branch when you want to experiment or create a new feature. It gives you the starting point of your main working set of code, and then you can play around with changes. If they work out, you can merge them back into "master", updating your main project. Otherwise, you can just delete or abandon them.



The process of branching and merging



The process of branching and merging

The 5-step program

1. Switch to a new branch, make & commit changes
2. Switch back to master & pull from GitHub to make sure it's all up-to-date
3. Switch to branch and merge master into branch
4. **Test** to make sure everything still works!!!
5. Switch back to master, merge the branch back into master, and then push back up to GitHub