

Understanding branches

Consider the following Terminal output run from a `git` repository:

```
git log --oneline --graph -10
```

```
1 * 8e540b0 (HEAD -> master, origin/master, origin/HEAD) update for Tuesday
2 * 89b0271 updated schedule
3 * 065ac6c added Amanda visit
4 * 2938417 updated ethics codes
5 *   076413e Merge pull request #2 from beanumber/hi
6 | \
7 | * 12a85b0 (origin/hi, hi) project
8 | * a352ecd updated schedule
9 * | 3e88c0e Merge pull request #1 from beanumber/hi
10 | \
11 | * 7b1766e update for Thursday
12 | /
13 * df0e1f4 reordered projects
```

1. How did we get from `df0e1f4` to `7b1766e`? Write out the `git` commands and explain them in your own words.
2. How did we get from `7b1766e` to `3e88c0e`? Write out the `git` commands and/or GitHub activities and explain them in your own words.
3. What happened after `12a85b0`? Did `beanumber` update the `hi` branch to include `3e88c0e`? Why did PR #2 succeed cleanly?
4. If this were a group project, what *faux pas* did `beanumber` commit when he pushed commit `2938417`? Explain why this is problematic in a group project (but not so problematic in a solo project).
5. Suppose you look at the GitHub repository page for the `hi` branch now. It says: “This branch is ____ commits behind ____.” Fill in the blanks.