

Intermediate Git

Branching and Merging

Jude Battista

25 Feb 2020

What we already know

The basic git cycle:

- 1 git pull
- 2 git status
- 3 git add
- 4 git commit
- 5 git push

Fear of commitment

So you committed a file and now regret it, so how **do** we undo a commit?

1 `git rm <filename>`

`git rm dumbFile.py`

... if you don't mind deleting your file.

2 Assuming you want to keep the file, use the `--CACHED` flag

`git rm --CACHED moderatelyBrightFile.py`

... `moderatelyBrightFile.py` will remain intact, but will no longer be part of your commit.

Undercommitment

What if you prematurely committed and neglected to include some files?

1 `git add forgottenFile.py`

2 `git commit --amend`

This updates your existing commit rather than creating another.

Creating a new branch

Branching is one of the most important features of git, allowing developers to work in parallel without stepping on each other's commits. Branching also permits the development of experimental features while concurrently maintaining a functional product. Correspondingly, the branching process is quite involved. The next two slides will summarize the process.

Creating a new branch

1 Initialize the branch.

`git branch shiny`

This is equivalent to: `git checkout -b shiny`

2 Switch to the new branch

`git checkout shiny`

Creating a new branch

Yeah, that's pretty much it.

All the King's horses...

Putting the pieces back together.

1 Switch to the destination branch

`git checkout The-Branch-We-Want-To-Merge-Into`

2 Initiate the merge

`git merge The-Branch-We-Want-To-Incorporate`

3 Celebrate?

Haha. No.

Merge Conflicts.

They are a thing.

Like Vivado and Illinois Nazis are things.

Whiskey Tango Foxtrot, angle brackets?!

```
<<<<<<< HEAD:index.html
```

```
<div id="footer">contact : email.support@github.com</div>
```

```
=====
```

```
<div id="footer">
```

```
please contact us at support@github.com
```

```
</div>
```

```
>>>>>>> iss53:index.html
```

Git some

Yea though I walk through the valley of the shadow of merge, I
shall fear no conflicts for I am the meanest XXXXXXXXXX developer
in the valley.

Whiskey Tango Foxtrot, angle brackets?!

<<<<<<< Indicates the start of the conflict

===== Splits the two sections

>>>>>>> Indicates the peace accord

You just need to pick the bits you want!

But Jude, what if your code is terrible and mine is great?

You can choose to take whole files from one source

To use your whole file:

```
git checkout ConflictedFileName --ours
```

To use the file in the repo:

```
git checkout ConflictedFileName --theirs
```

Klaatu Barada Nihmmhmmhm

<https://git-scm.com/book/en/v2>