

Git(Hub) 401

Intermediate/Advanced git/GitHub Tutorial

.astronomy 2018

Instructors:

Matthew Bourque

Jules Fowler

Outline

- Installation/Setup
- Collaborative Workflows
- Merging vs Rebasing
- Release Workflow
- Exercise: Propose Changes via a Pull Request
- Exercise: Keeping a fork up-to-date (Rebase) (if time allows)
- Exercise: Squashing history (if time allows)

Setup

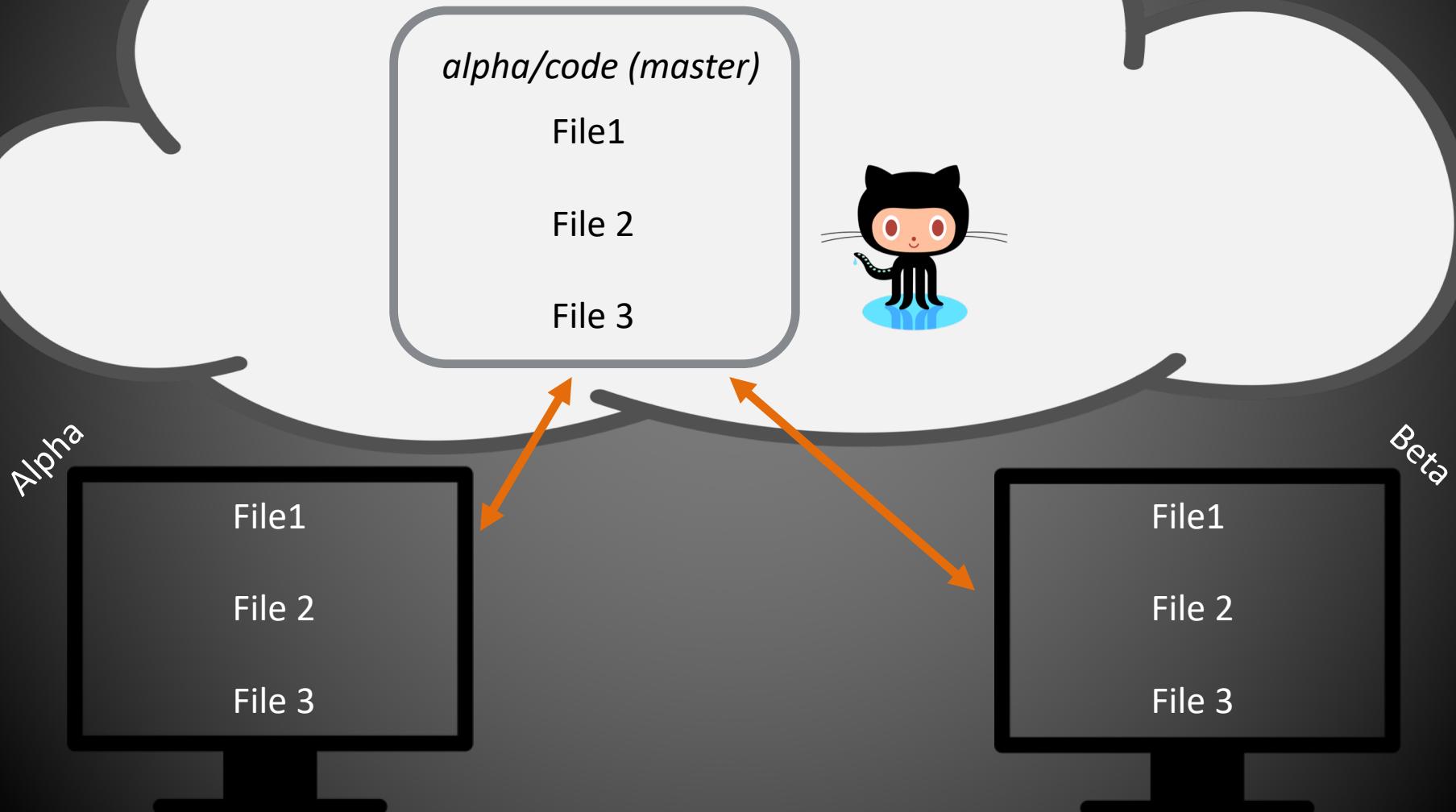
- Slack channel: #github-401
- Post-It notes
- Install git:
 - Linux/Mac (with Anaconda)
 - `conda install git`
 - Linux (without Anaconda)
 - <https://git-scm.com/download/linux>
 - Mac (without Anaconda)
 - <https://git-scm.com/download/mac>
 - Windows
 - <https://gitforwindows.org/>
- Set up SSH key (if using SSH):
 - <https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/>
- Sign up/sign into GitHub Account
- Configure git (if necessary):
 - `git config --global user.name "Your Name"`
 - `git config --global user.email "YourEmail@email.edu"`

Assumed Level of Knowledge

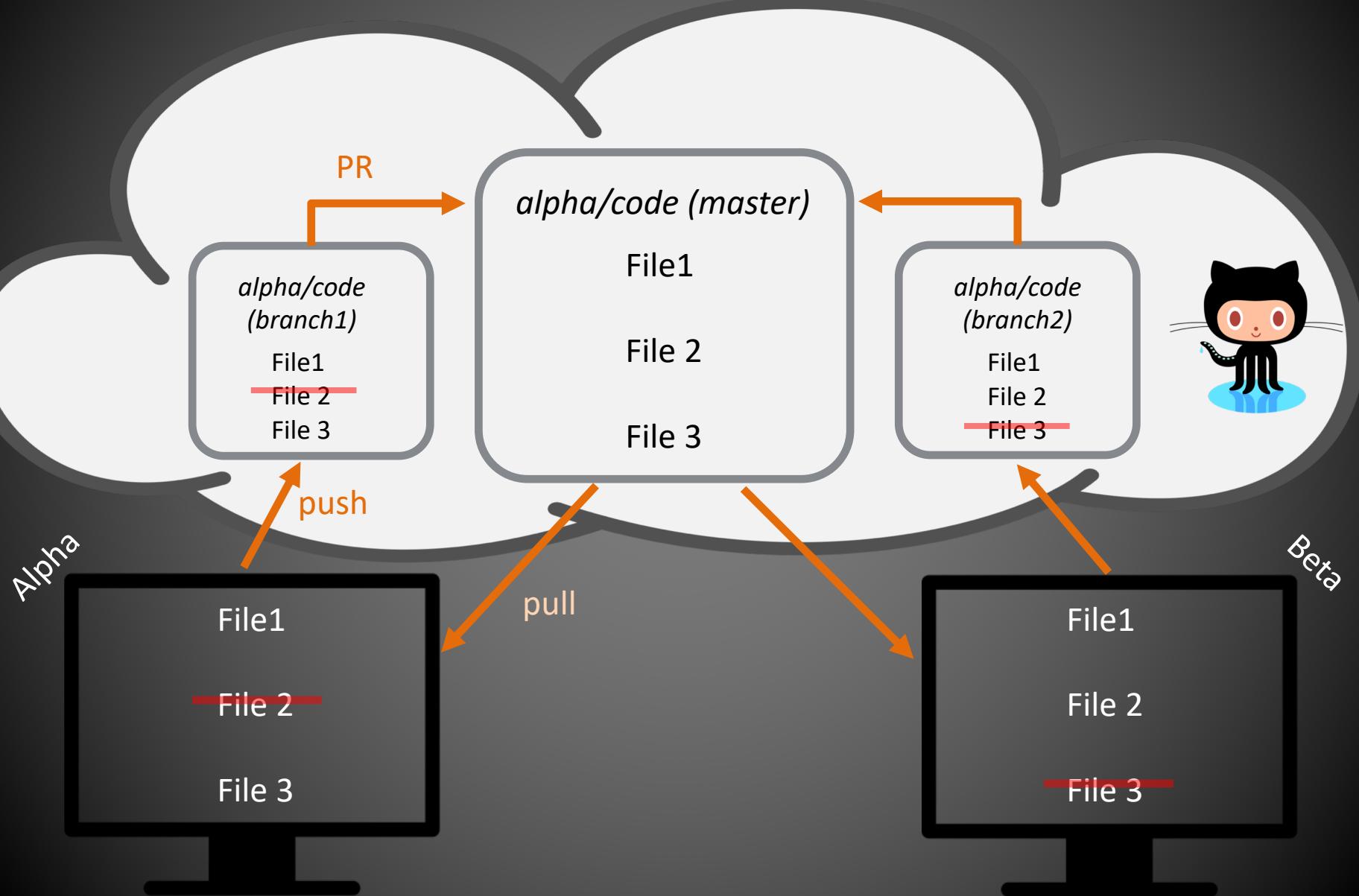
- git clone
- git add/commit
- git push/pull
- git branch/checkout

Git(Hub) 101 class: 15:30 – 16:30 Café Con

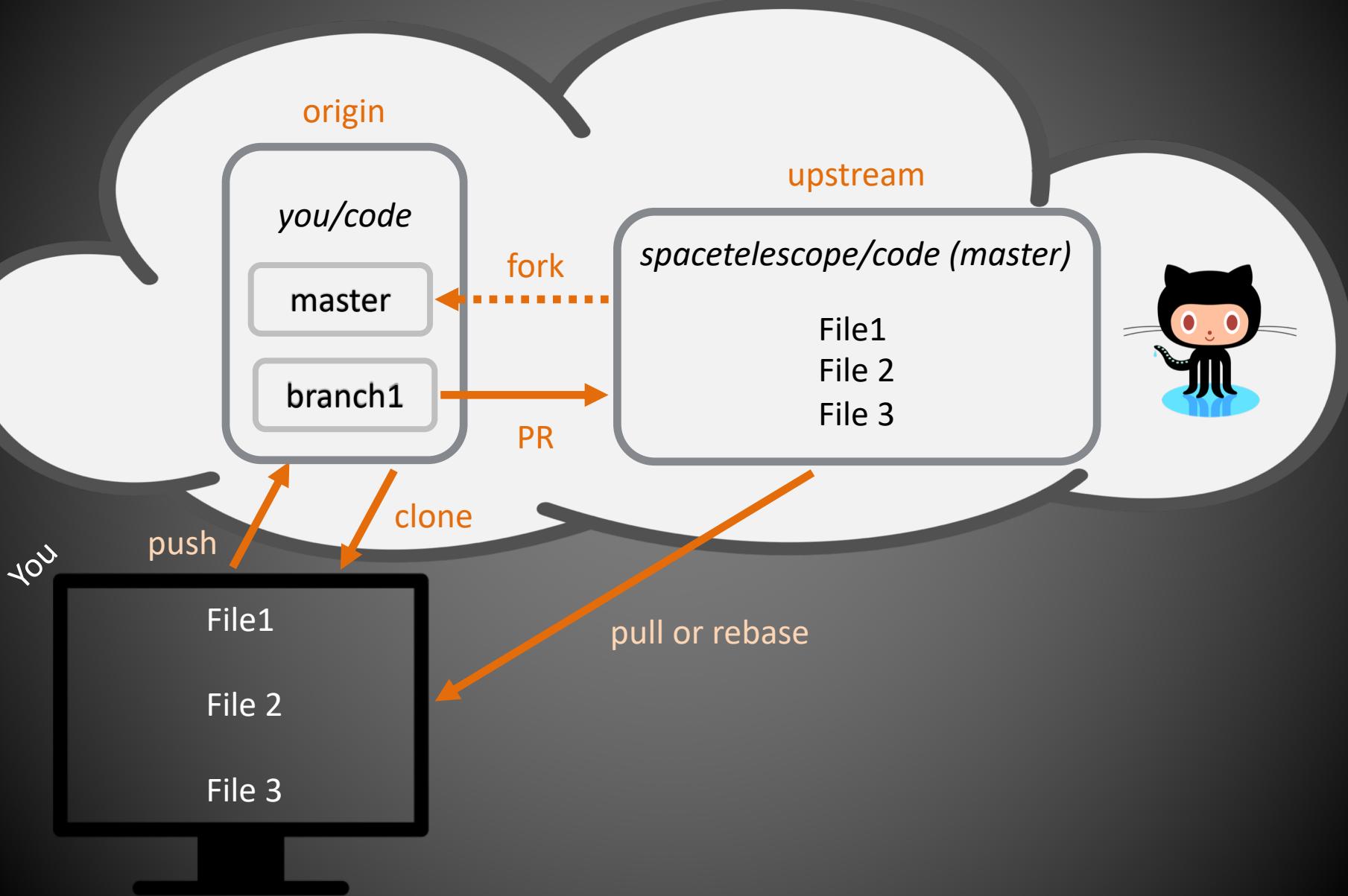
The simplest workflow



A better workflow



A even better workflow



Let's dig down...

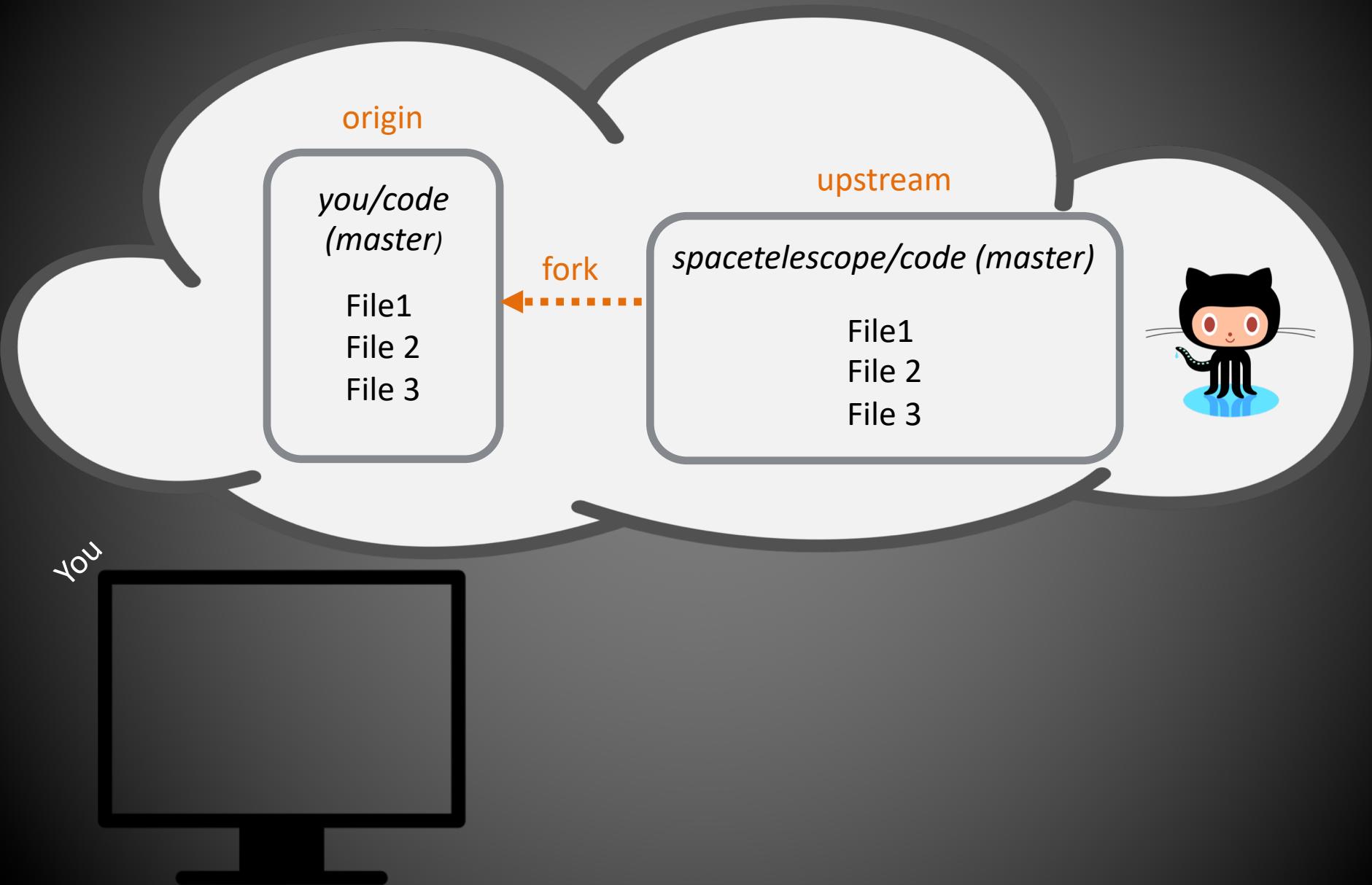
You

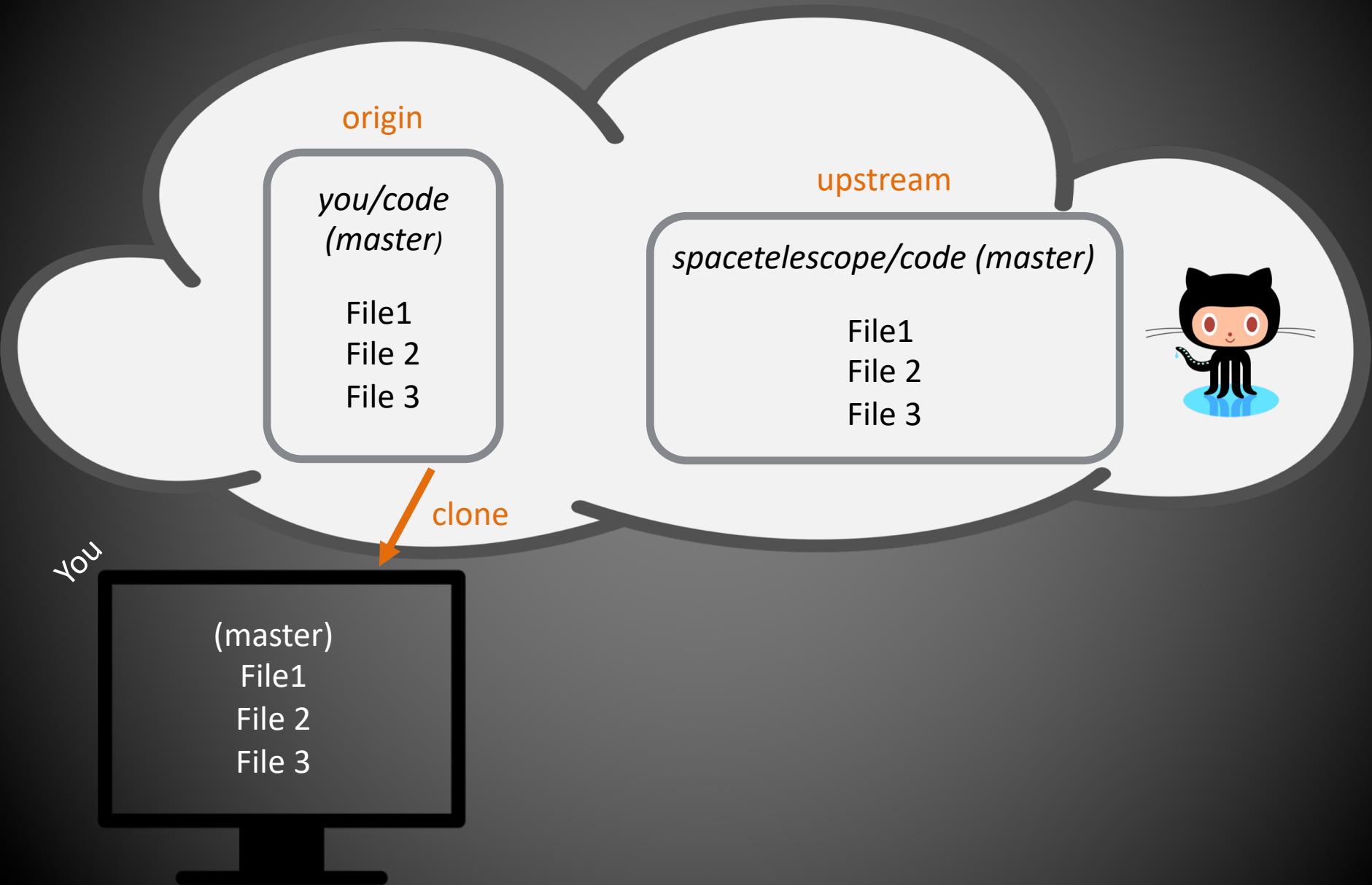


spacetelescope/code (master)

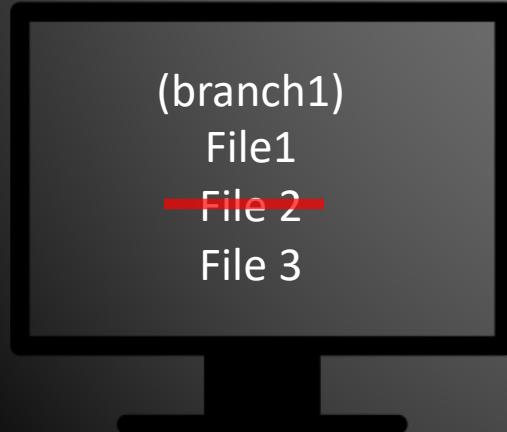
File1
File 2
File 3







You



origin

*you/code
(master)*

File1
File 2
File 3

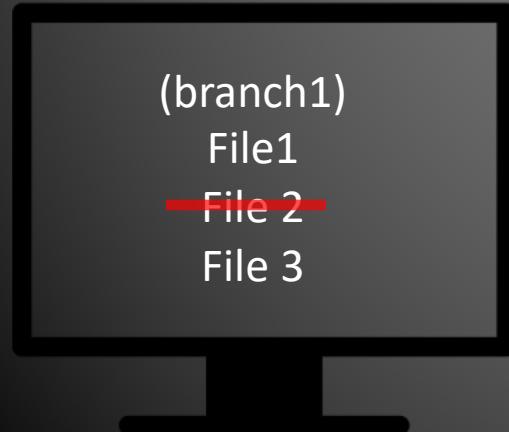
upstream

spacetelescope/code (master)

File1
File 2
File 3



You



push

origin

*you/code
(branch1)*

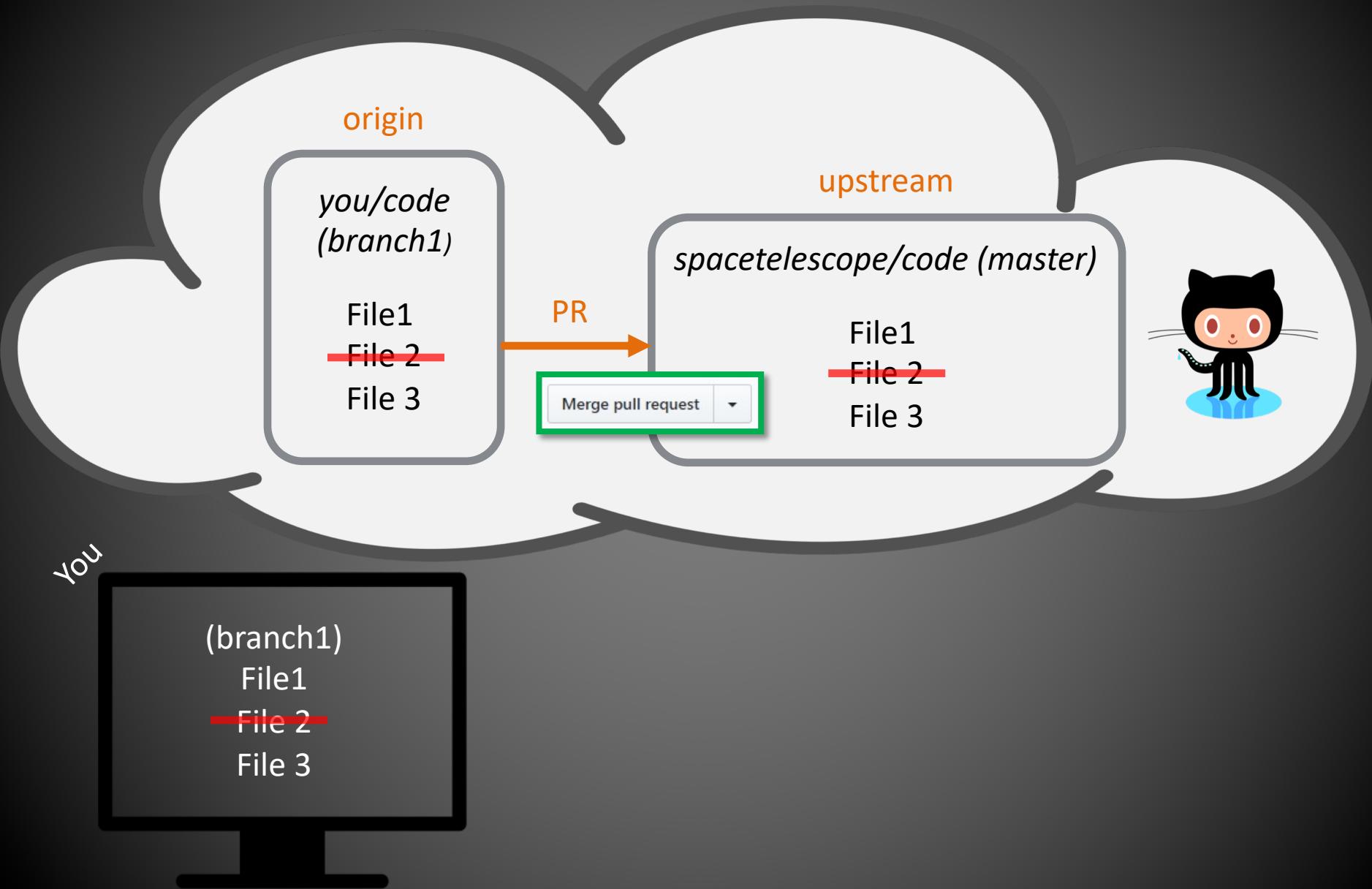
File1
~~File 2~~
File 3

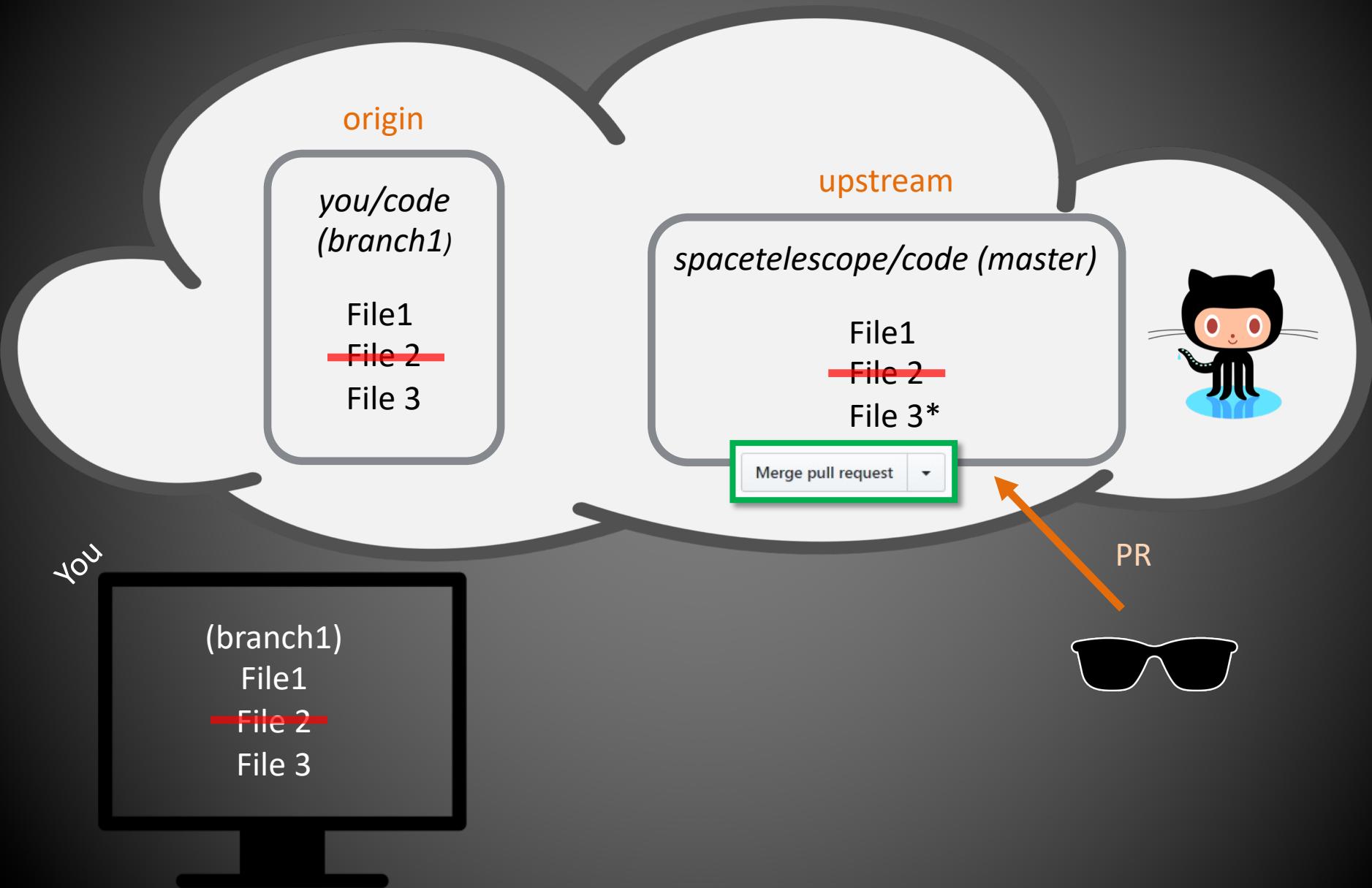
upstream

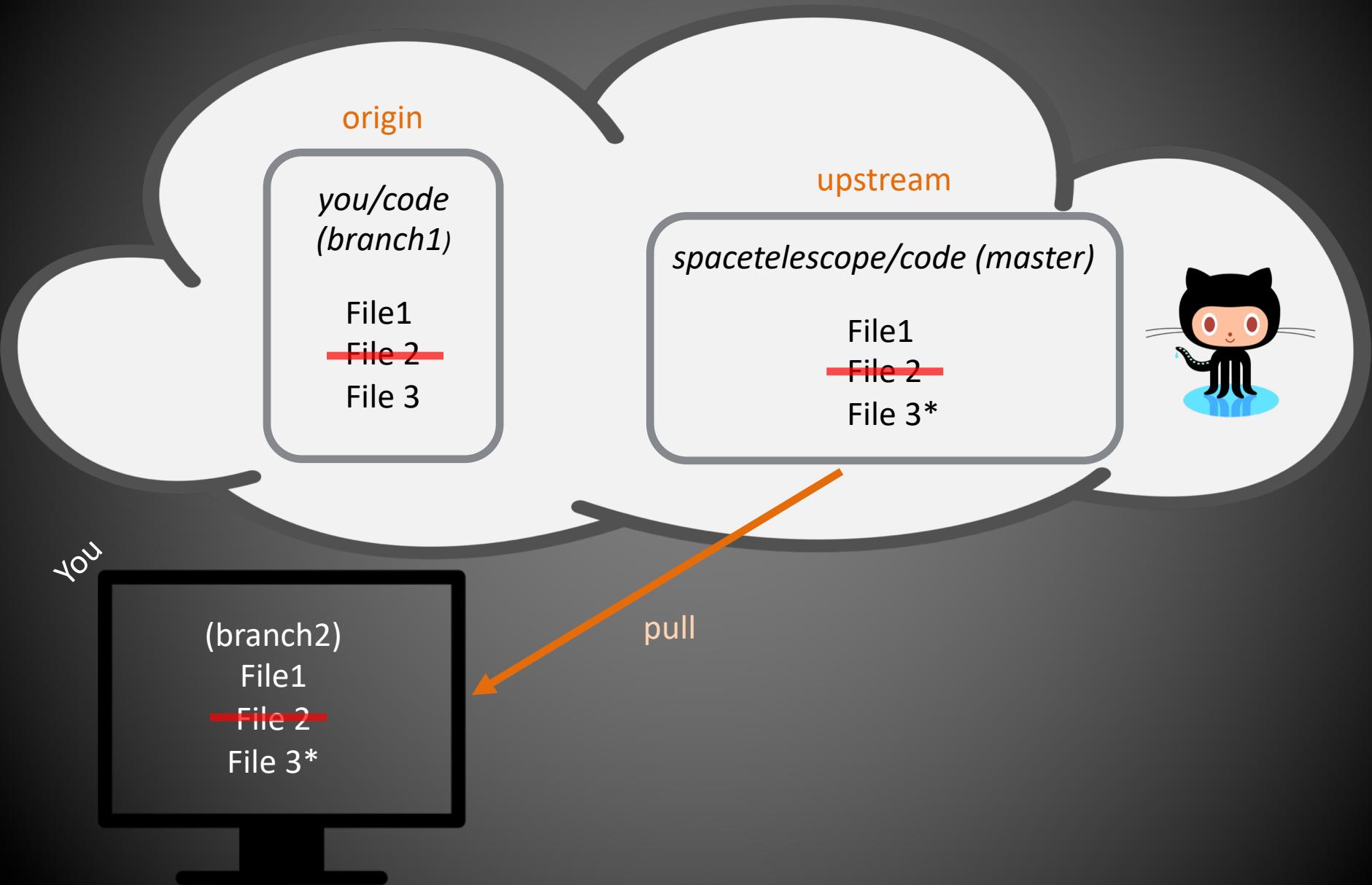
spacetelescope/code (master)

File1
File 2
File 3

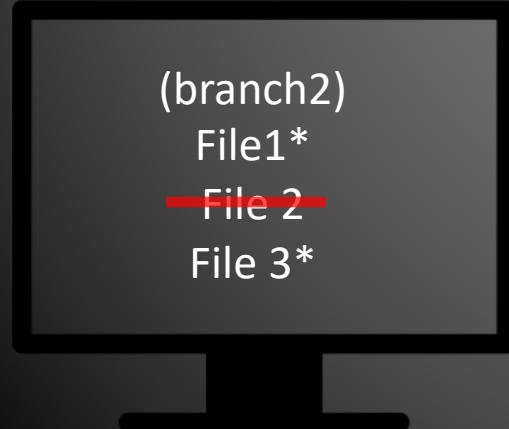








You



origin

*you/code
(branch1)*

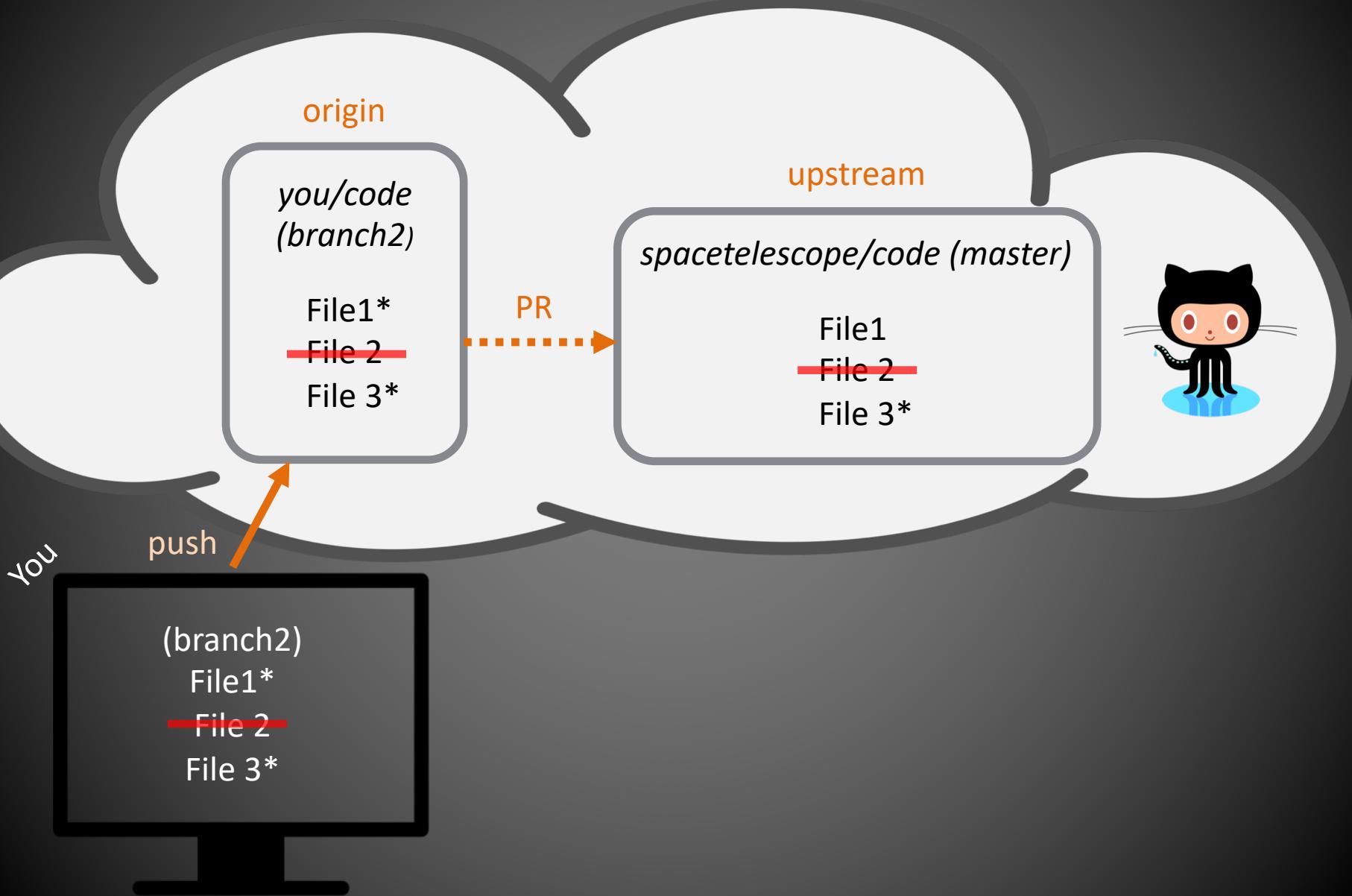
File1
~~File 2~~
File 3

upstream

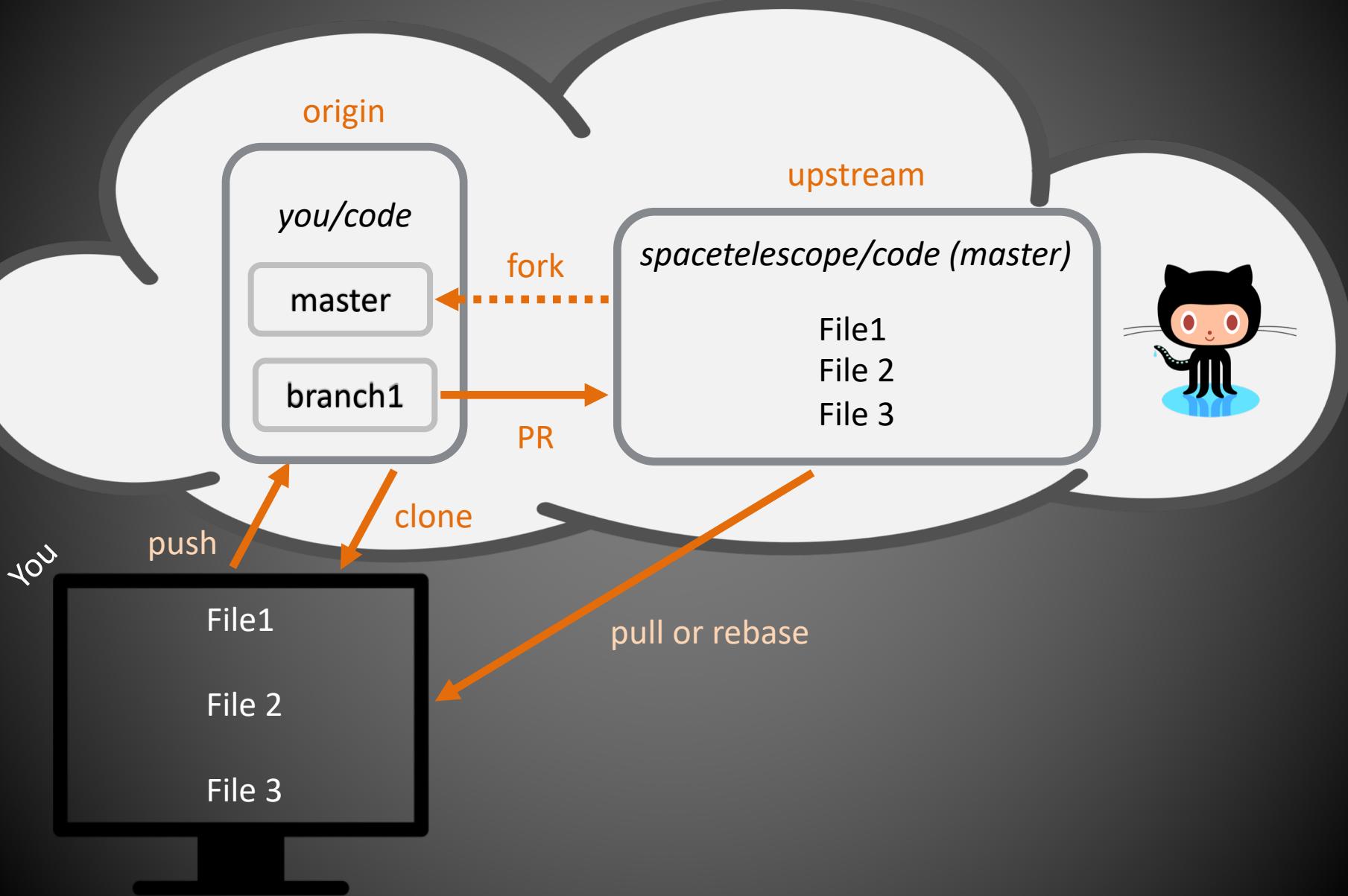
spacetelescope/code (master)

File1
~~File 2~~
File 3*

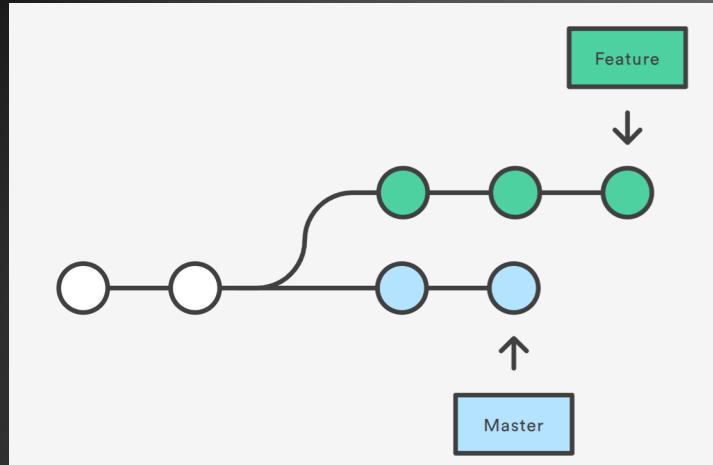




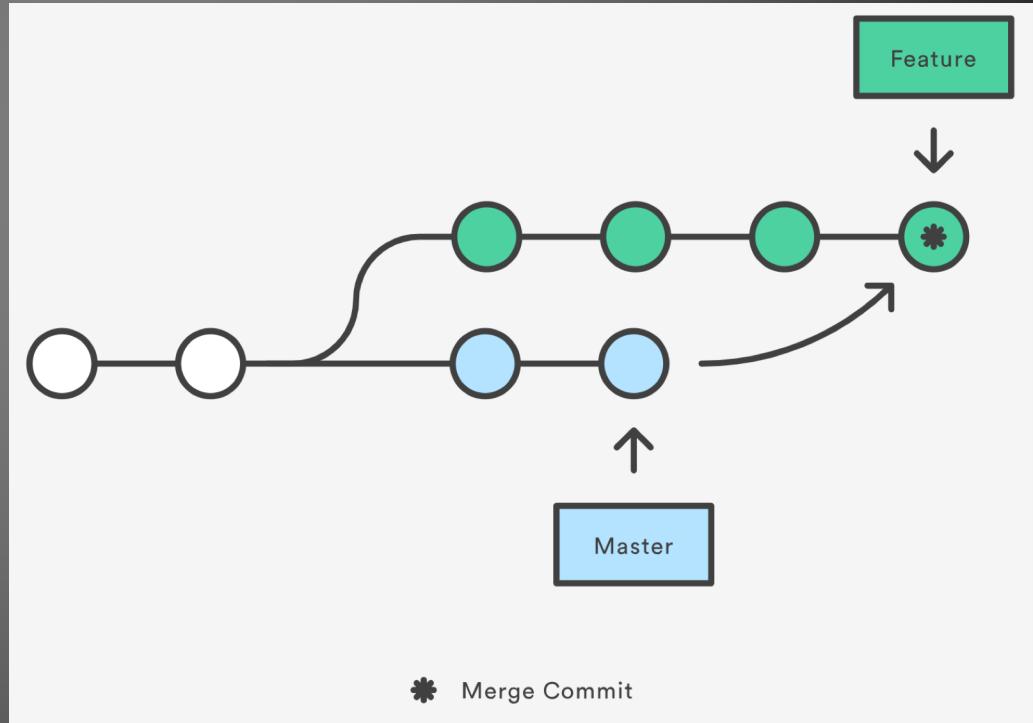
A even better workflow



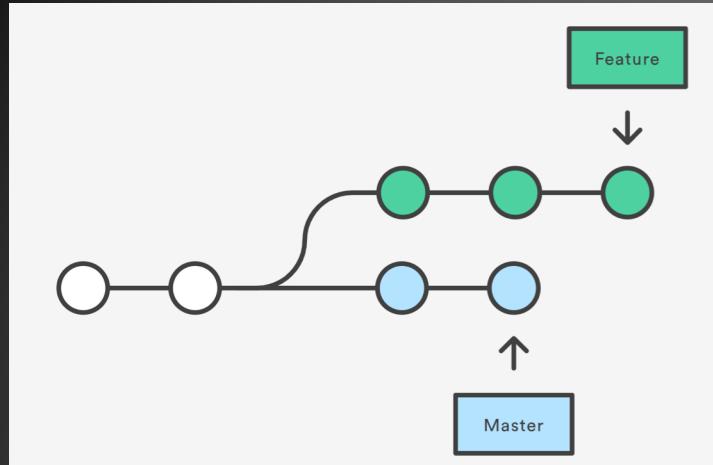
Merging



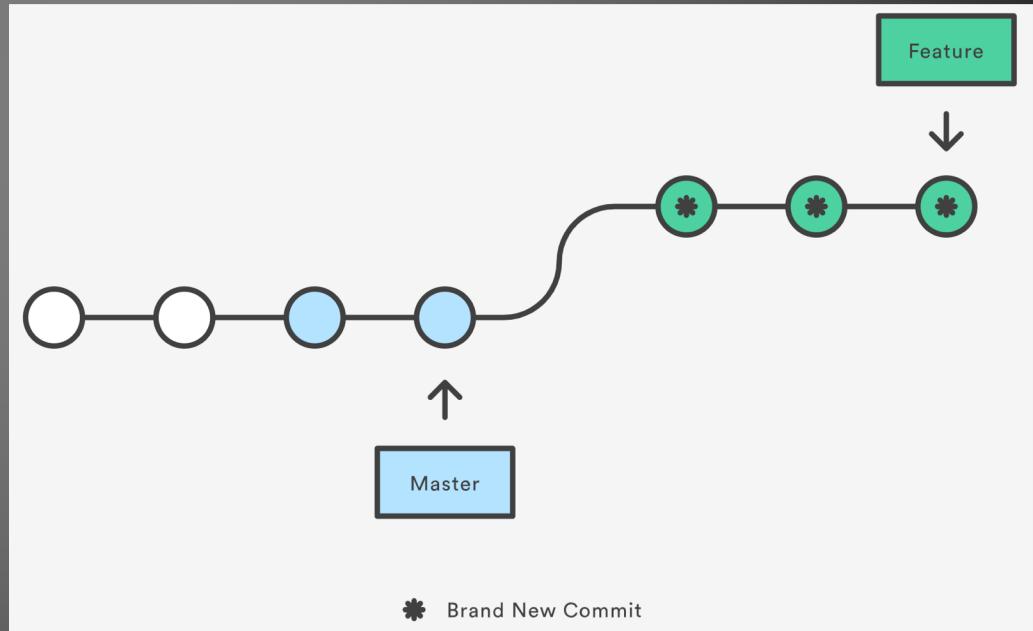
- git checkout feature
- git merge master
- non-destructive – history of existing branches are not changed
- Extra merge commit every time you incorporate upstream changes



Rebasing



- git checkout feature
- git rebase master
- Moves entire feature branch to begin at tip of master branch
- Instead of merge commit, rebasing re-writes history by creating brand new commits
- Linear project history – you can follow tip of feature branch all the way to the beginning of the project



Exercise: Propose Changes via Pull Request

1. Fork the `spacetelescope/github-401` repository to your GitHub `<username>`
2. Clone your `<username>/github-401` repository to your local machine
3. In your local clone, create a branch called `<username>-feature` and switch to it
4. Add your name to the `README.md` file
5. Create a new file called `<username>-time.py`
6. In `<username>-time.py` write a program that prints the current time using your favorite method
7. Commit your changes and push your branch to your fork
8. Open a Pull Request that compares your branch to the `spacetelescope/github-401 master` branch
9. Assign `bourque` as a reviewer
10. Review the pull request of someone else in the class. Make at least one comment on their code with something you would have done differently/suggested improvements
11. BONUS: Make improvements to your code based on the suggestions of your fellow participants and update the pull request

Exercise: Updating your fork via rebase

1. Instructor will commit change to
spacetlescope:master branch
2. Add a new remote that points to the
spacetlescope/github-401 repository:
 - git remote add upstream <url>
3. Perform the rebase:
 - git fetch upstream master
 - git rebase upstream/master
4. Resolve any conflicts
5. You should now see any changes that occurred in
spacetlescope:master branch in your
<username>-feature branch

Exercise: Squashing History

1. Squash your two latest edits into one
 - git rebase -i HEAD~2
 - Follow terminal instructions
2. Push your branch out again (with --force)
3. Inspect your pull request: What has changed?



Special Thanks

- STScI git Training Team
- Software Carpentry
- Azalee Bostroem
- Erik Tollerud