

GitHub Campus Advisors

Teacher training to master Git and GitHub



GitHub Education

Module 1

Basics

Module 2

Individual assignments

Module 3

Group work

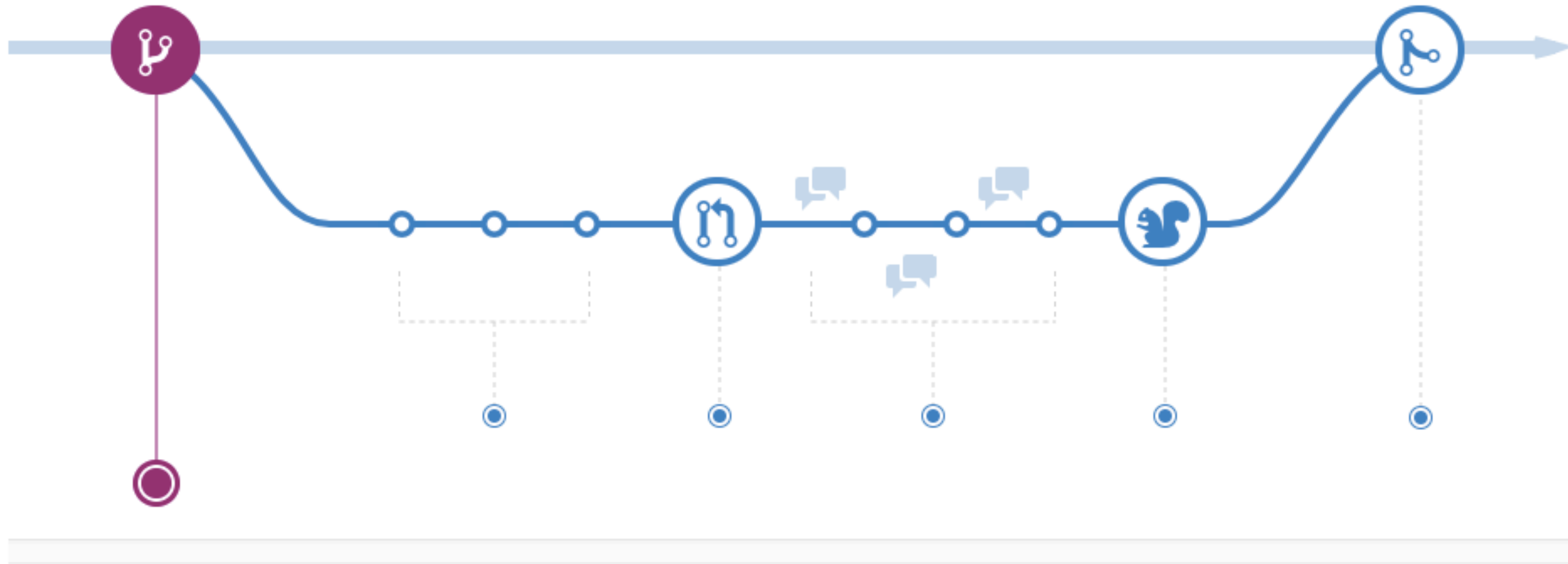
Module 4

Student programs



Group work

made regularly. This guide explains how and why and has a few tricks.



The GitHub Flow

Dr. Root distributes a group exercise

Ok Team 1, here is a shared repository with starter code for your group exercise.

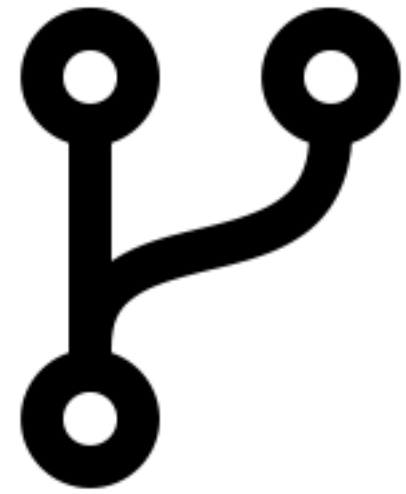


[github.com/ Software-engineering/team-1-setup](https://github.com/Software-engineering/team-1-setup)



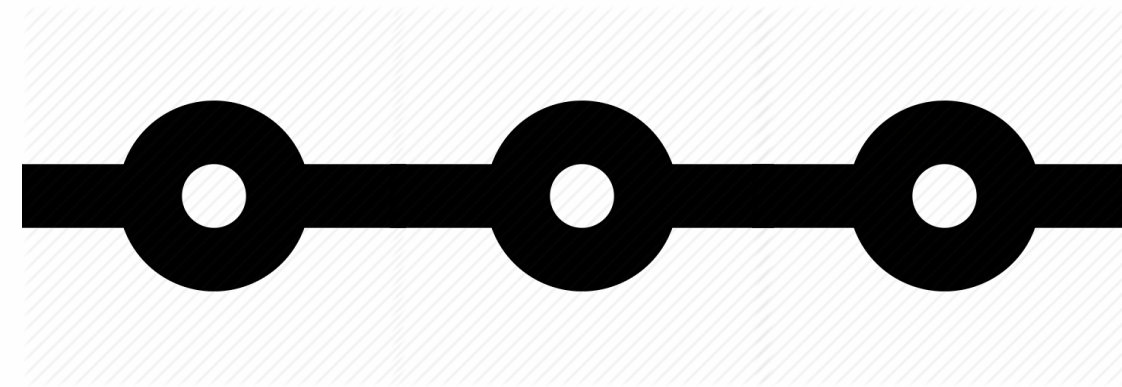
Branch

```
git checkout -b authenticate
```



Commit

```
git checkout -b authenticate
```

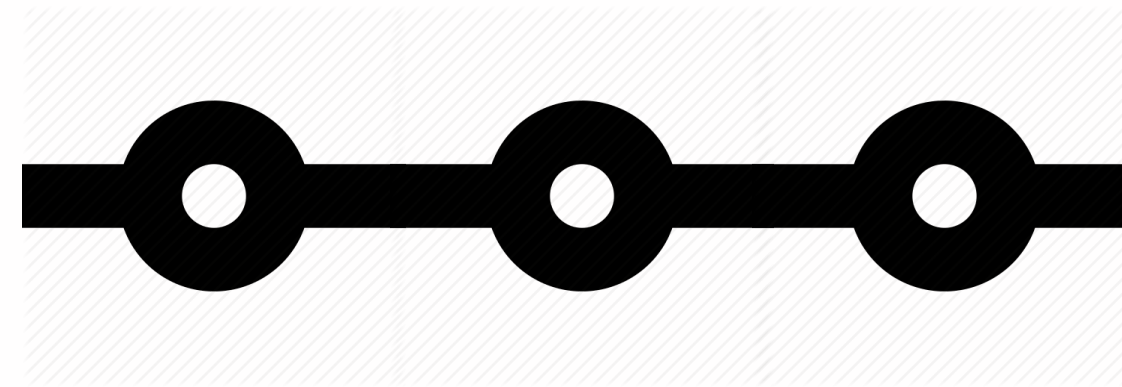


```
git commit -m 'add OAuth handshake'
```



Push to remote

```
git checkout -b authenticate
```

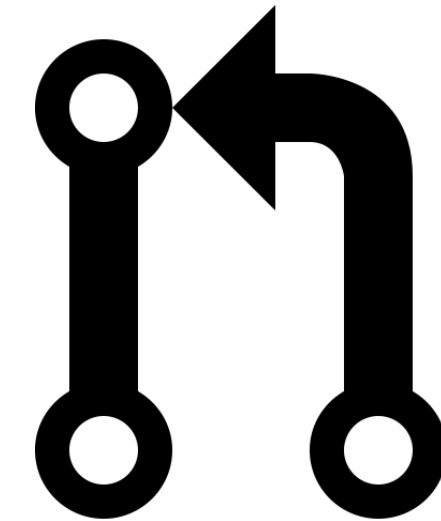
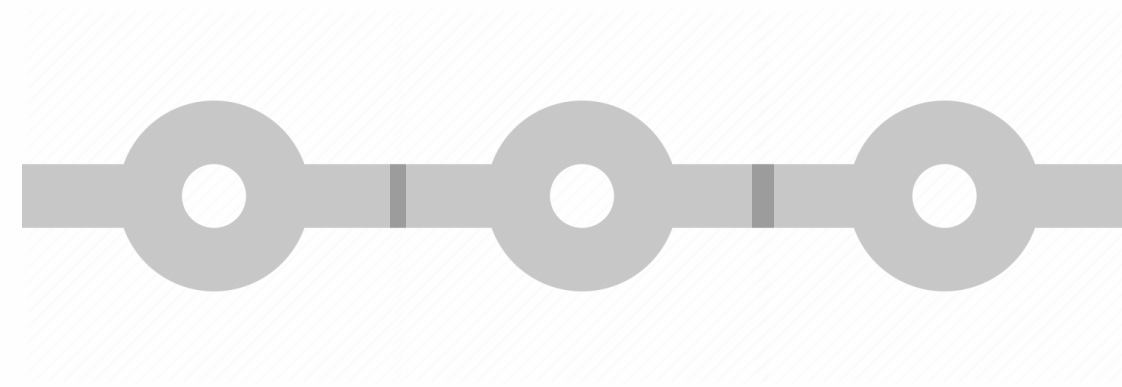


```
git commit -m 'add OAuth handshake'  
git push origin authenticate --set-upstream
```

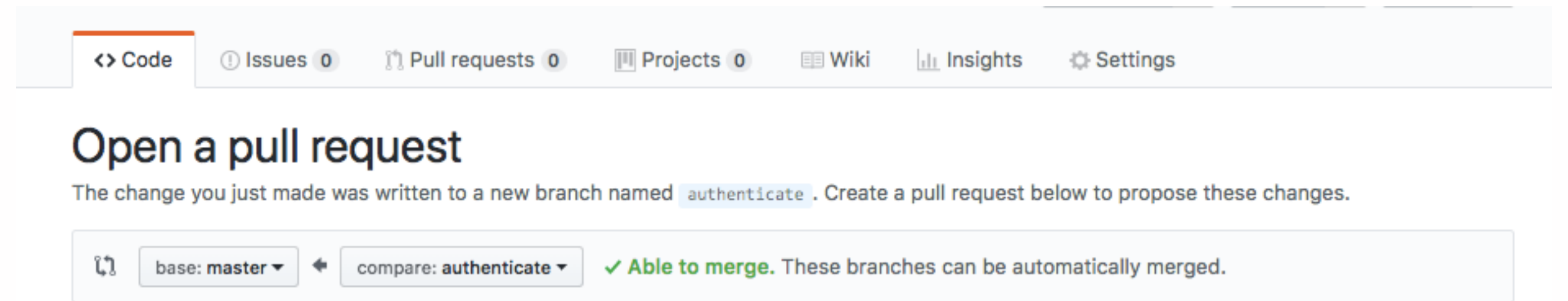


Pull request

```
git checkout -b authenticate
```

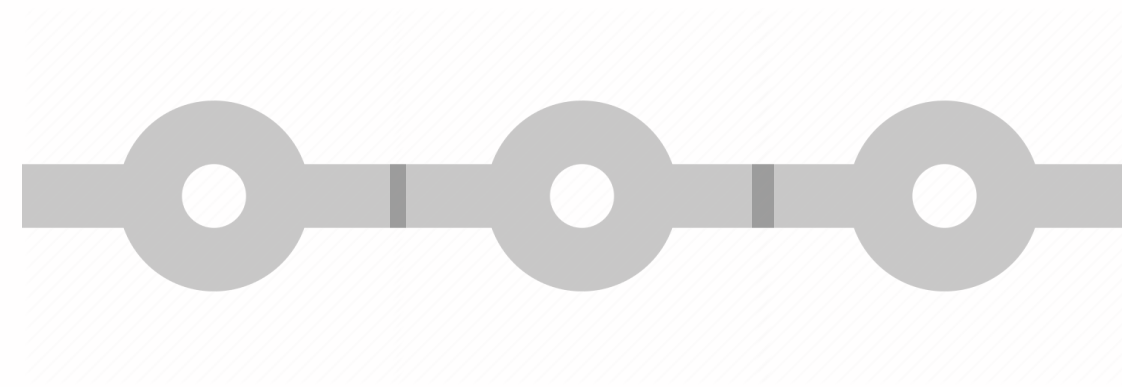


```
git commit -m 'add OAuth handshake'  
git push origin authenticate --set-upstream
```

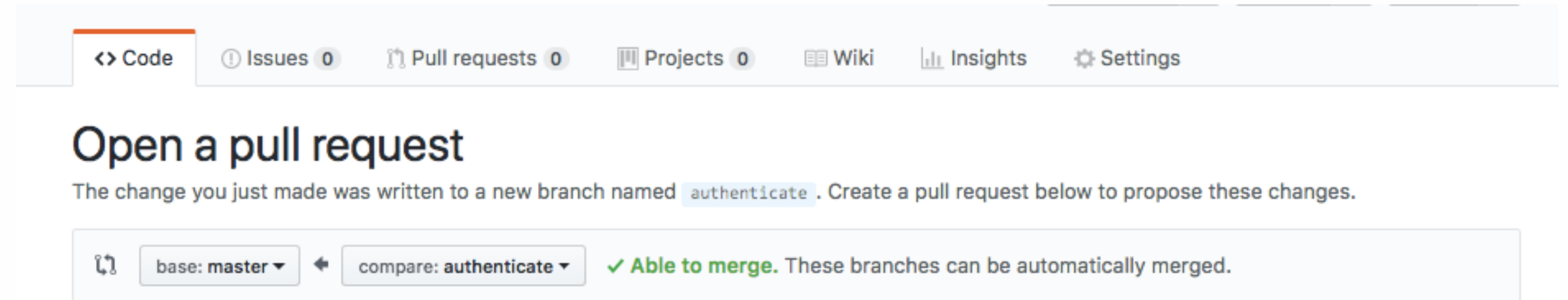
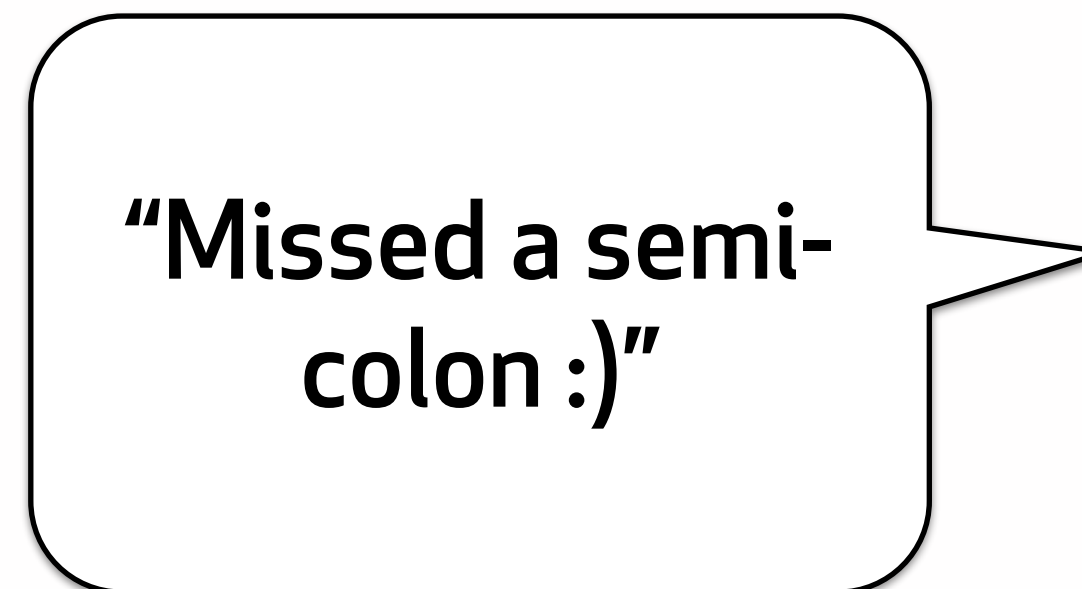


Review

```
git checkout -b authenticate
```

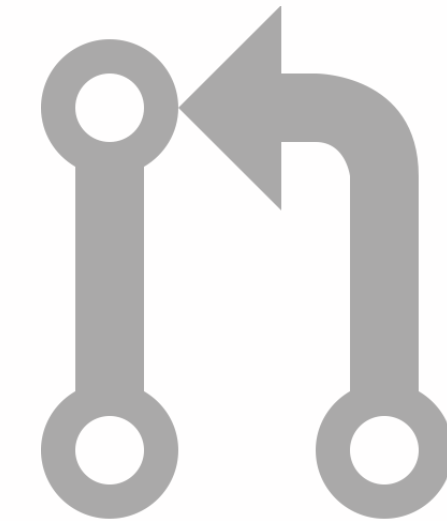
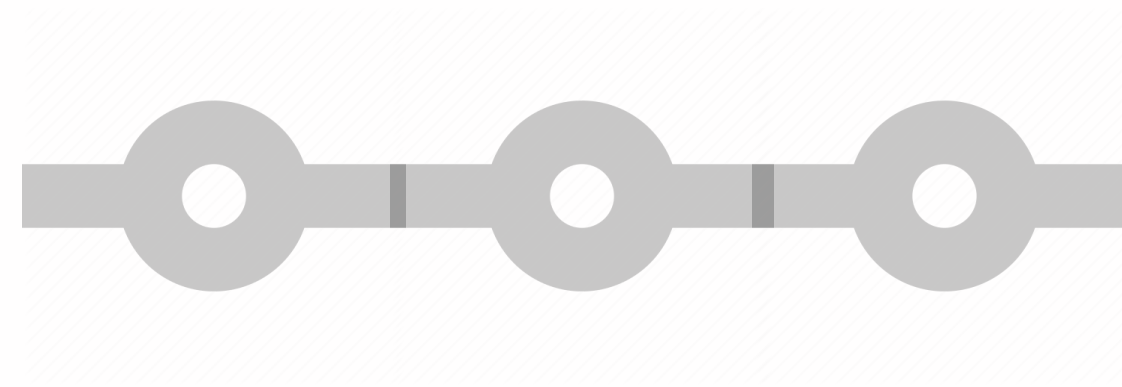


```
git commit -m 'add OAuth handshake'  
git push origin authenticate --set-upstream
```

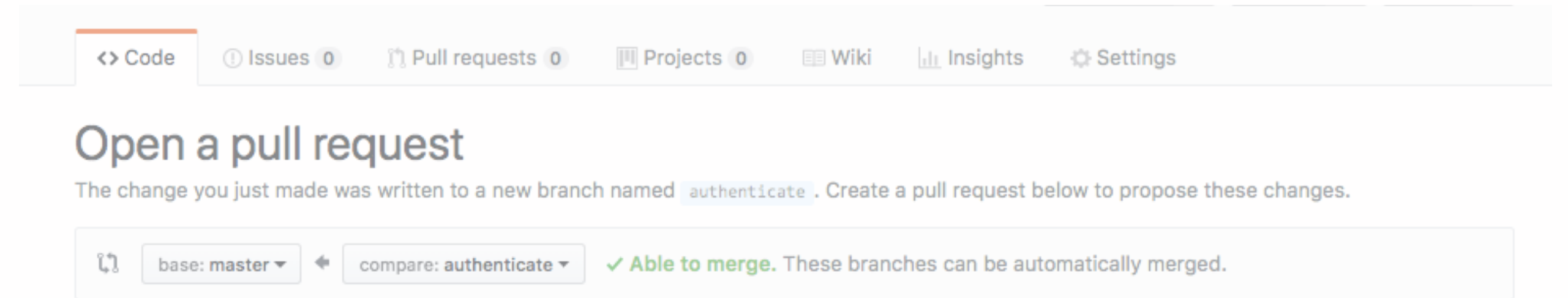


Merged!

```
git checkout -b authenticate
```



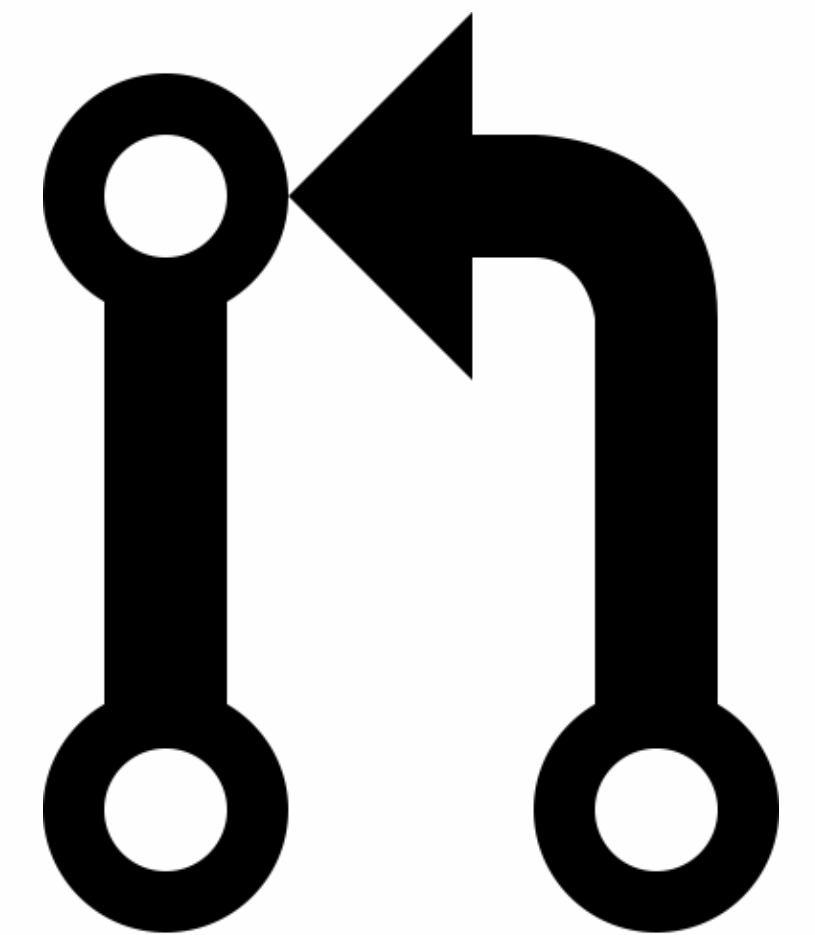
```
git commit -m 'add OAuth handshake'  
git push origin authenticate --set-upstream
```



Pull requests

Pull requests

- Technically: merge one branch into another branch in the repository's history
- Pedagogically: peer discussion and feedback



Ways to use pull requests

- Individually, on your own repository
- For collaboration in shared repositories
- github.com (recommended for beginners)
- using the `hub` package (more advanced)



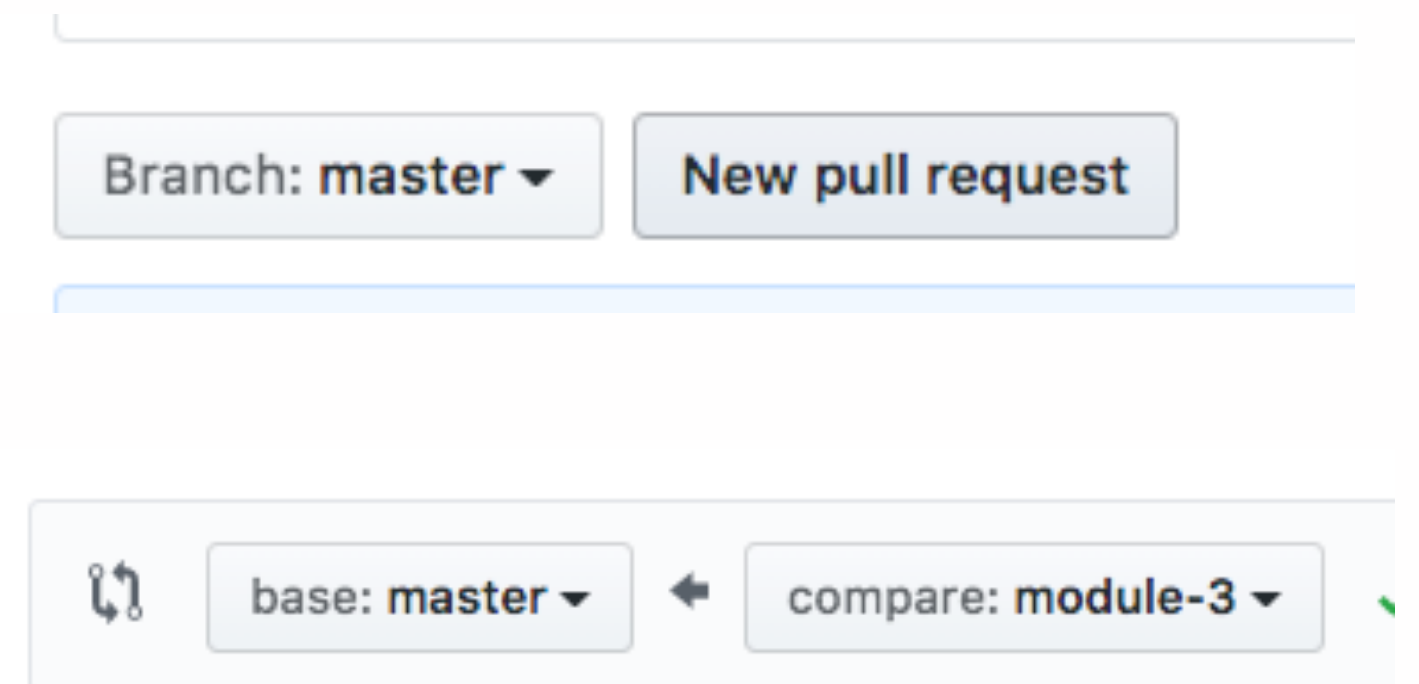
Why use pull requests for solo work?

- Builds the habit of pushing to a remote
- Enables rich, contextual feedback in assessment



Steps to open a pull request

1. In your terminal, create a branch
2. Add a remote tracking branch and push the local branch to the remote
3. In the project repository on github.com, open a new pull request
4. Select the where you'd like the branch to go (base) and the branch you want merged (head)
5. Create the pull request, and describe your changes and select reviewers from your team



Steps to open a pull request

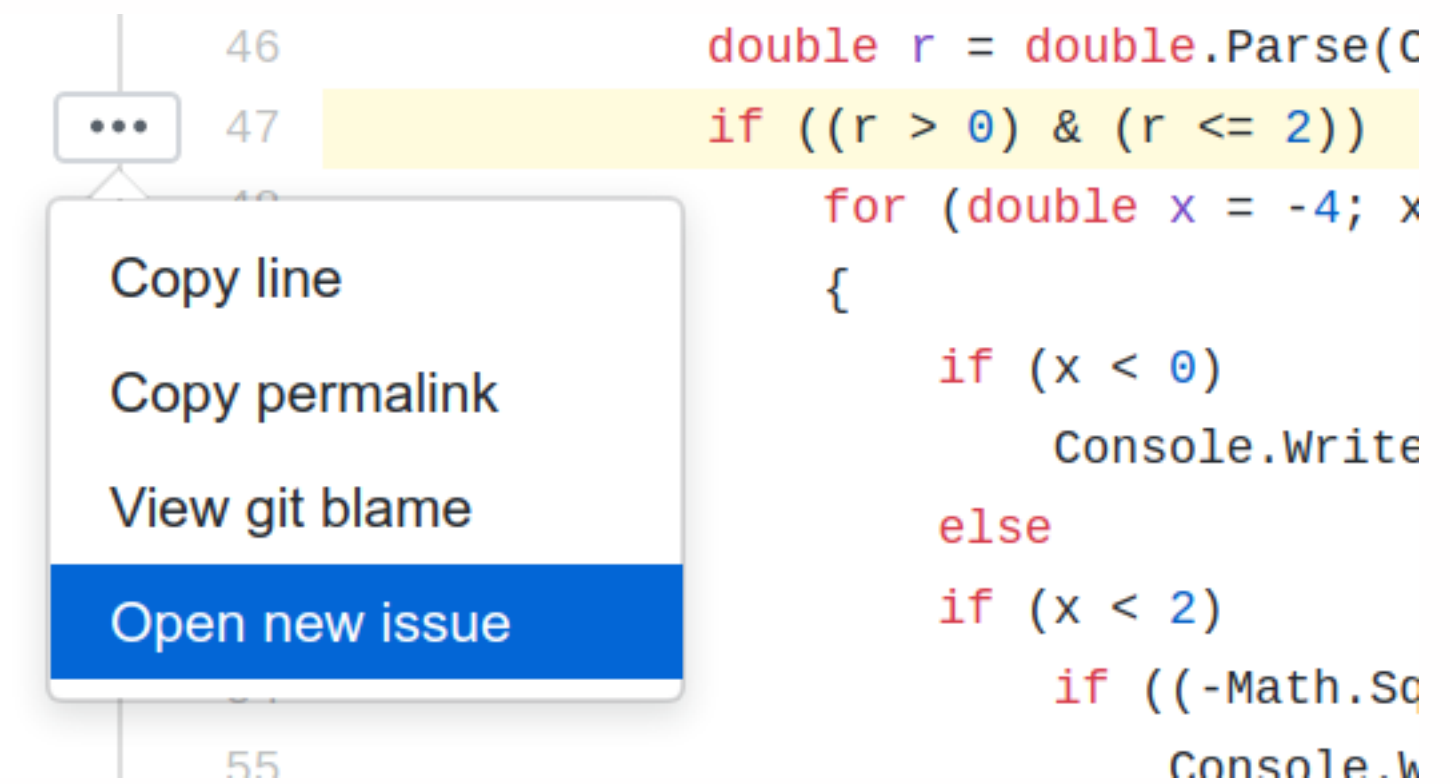


A terminal window with a title bar that reads "billyyumyum2x2.github.io — brianamarie@MacBook-Pro-33 — ..2x2.github.io — -zsh — 99x29". The terminal prompt is "billyyumyum2x2.github.io (start-portfolio) git" followed by a cursor. A small information icon is visible in the center of the terminal area.



Assessing student work on pull requests

- Click on the + sign of the line number to give feedback in-context.
- Supports markdown syntax.
- Automate feedback on consistent mistakes by saving a canned reply (great for syntax errors).
- From the code view, click the three dots on the line and open an issue with your canned reply.



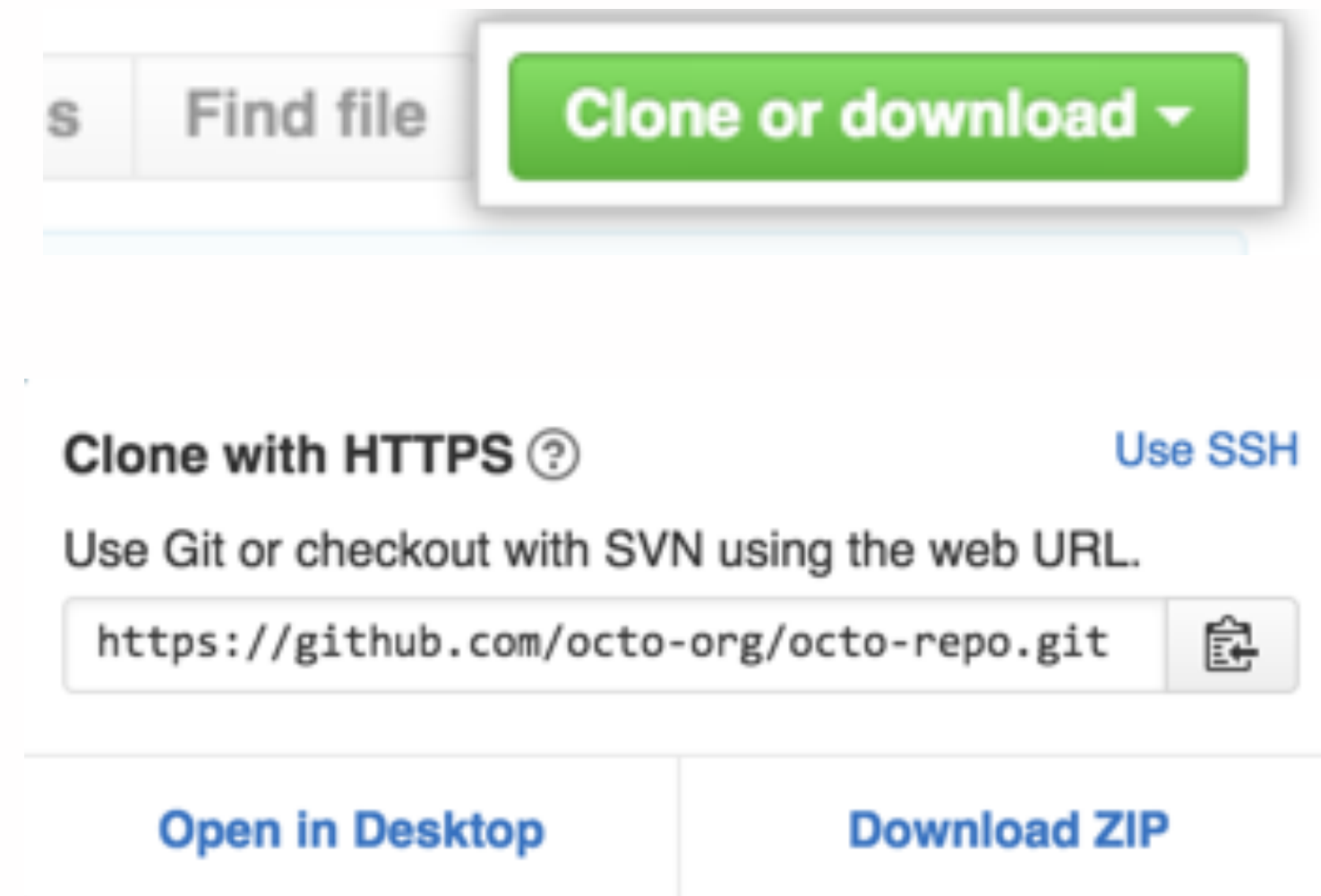
The screenshot shows a code editor with a context menu open over a line of code. The code is in C# and includes a conditional statement and a loop. The context menu has four options: 'Copy line', 'Copy permalink', 'View git blame', and 'Open new issue'. The 'Open new issue' option is highlighted in blue. The code snippet is as follows:

```
46 double r = double.Parse(C
47 if ((r > 0) & (r <= 2))
48 for (double x = -4; x
49 {
50     if (x < 0)
51         Console.Write
52     else
53     if (x < 2)
54         if ((-Math.Sc
55             Console.W
```



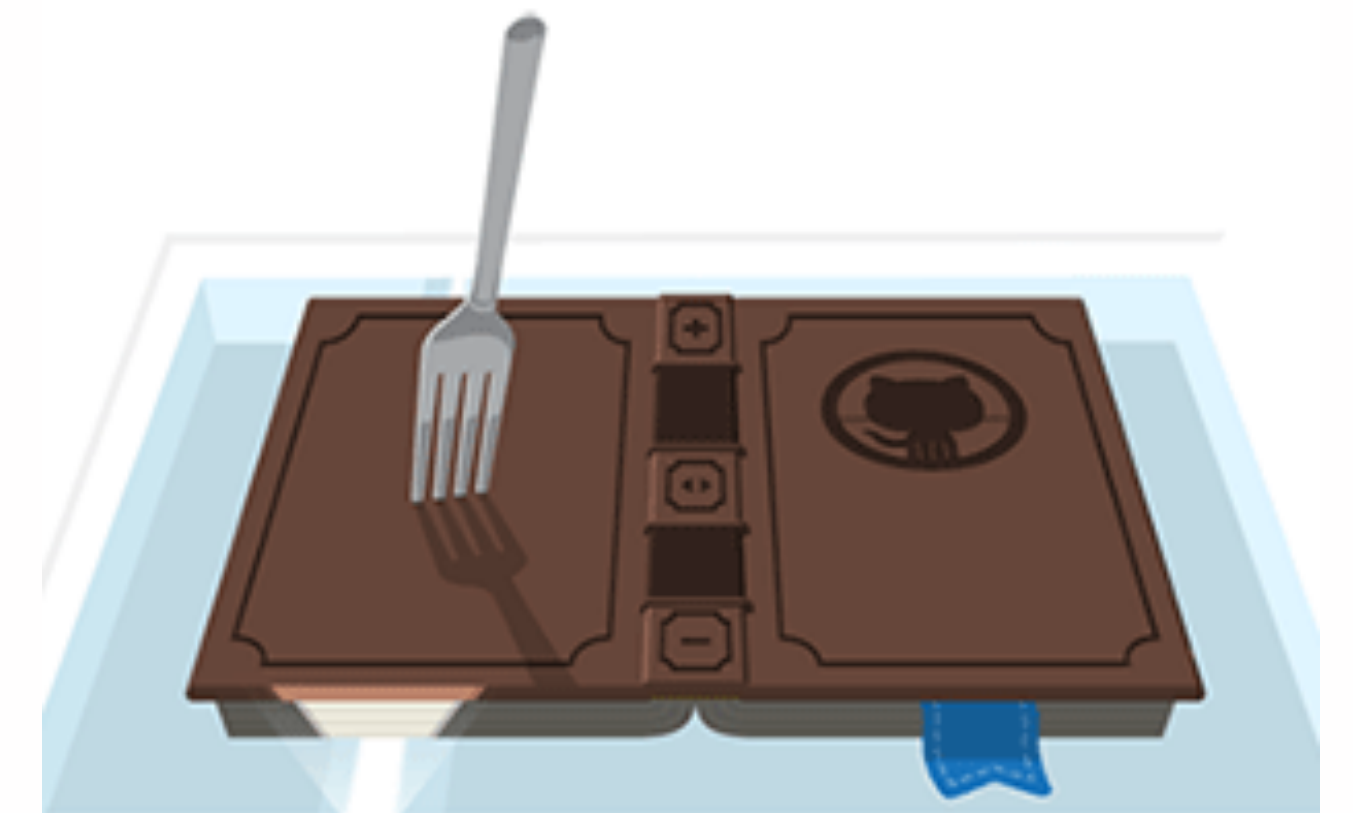
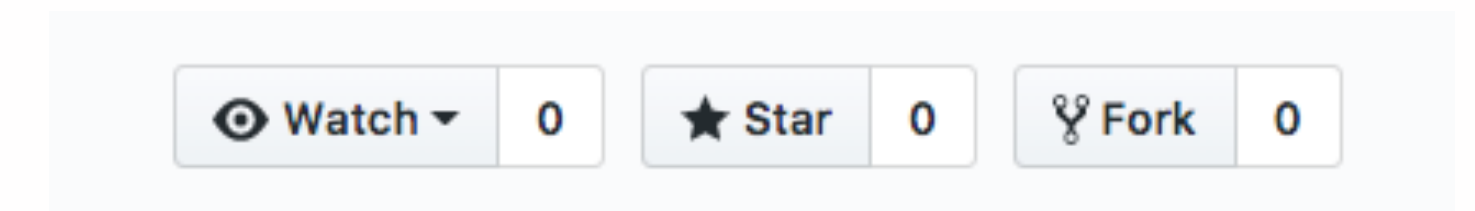
Clone a repository

- We've used the clone method to make a copy of a repository from GitHub.com to your local machine
- Clone sets up a sync between local and remote
- Use the either the web UI or the HTTPS methods on the repository page.



Fork a repository

- Copy of a project that is always associated with the base project.
- Always discoverable, not private.
- Proposing ideas to a project you do not have access to.
- Closer to the open source workflow—used to propose new ideas on a project you don't necessarily own




Exercise:


- On the command line, create repository for Module 3 and a readme file.
- On GitHub.com, create a module 3 repo and add it as a remote.
- In your terminal, create a new branch with a questions.md document, and write down any remaining questions you have about pull requests.
- Add a remote tracking branch and push to it.
- On GitHub.com, open a pull request with the branch that has your questions--you can mention me @mozzadrella as a reviewer or in the comments to answer them.
- Post a screenshot of your pull request to the forums.



<<<The feared merge conflict>>>

Add more commits by pushing to the **my-title** branch on beardofedu/my-resume.




 **This branch has conflicts that must be resolved**
Use the [web editor](#) or the [command line](#) to resolve conflicts.


[Resolve conflicts](#)

Conflicting files

_config.yml

[Merge pull request](#)  You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

beard's information > Conflicts

_config.yml 1 conflict [Prev](#) [Next](#) 

```
1 # Site settings
2 <<<<<< my-title
3 title: "beard's resume"
4 description: "beard's amazing resume."
5 =====
6 title: "beardofedu's resume"
7 description: "beardofedu's excellent resume."
8 >>>>>> master
9
10 # Build settings
11 markdown: kramdown
12 sass:
13   sass dir: sass
```



Merge conflicts are
Git-speak for
“I need a human to
make a decision”



Steps to resolve a merge conflict

1. Navigate to where the conflict occurs, shown by conflict markers <<<<<<<, =====, >>>>>>>.
2. Choose option A, option B, or a blend of the two (option C).
3. Save the file and stage your changes.
4. Commit with a message documenting your choice.

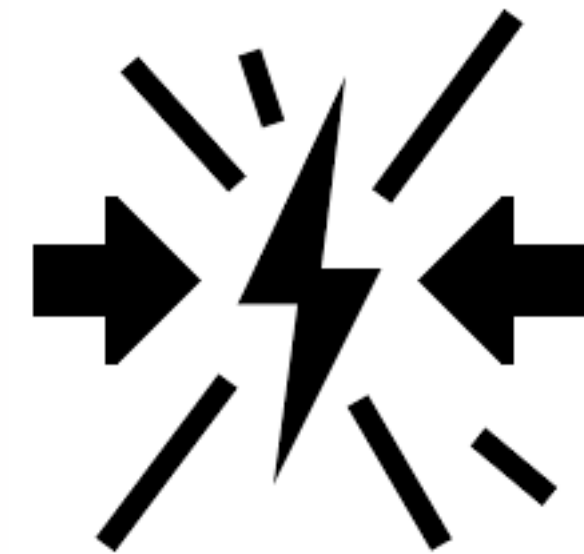
Example: collaborating on a pizza



```
1 # Ingredients
2
3 * Flour
4 * Water
5 * Tomatoes
6 * Yeast
```



Example: collaborating on a pizza



```
ingredients.md — ~/Desktop/pizza
ingredients.md
1 # Ingredients
2
3 * Flour
4 * Water
5 * Tomatoes
6 * Yeast
7
8 <<<<<<< HEAD
9 * Green pepper
10 * Mushroom
11 =====
12 * Pepperoni
13 * Sausage
14 >>>>>> 920b1661c31aa8538486b0d2f2495526981e78a8|
15
```



Epic win

```
[master f74243e] resolved merge conflicts  
Mozzadrella:pizza Mozzadrella$ █
```



Merge conflict tips and tricks

1. Scaffold: push to individual repositories first, then pull requests, then group assignments.
2. Demo: in class, make a demo like this one so they can watch you and then do it themselves.
3. Keep changes small, logically grouped, but not covering 50+ lines or 20 files.
4. Have a Project Manager appointed in each team who merges pull requests quickly.

Exercise:

- Accept the group assignment, join a team, and gain access to a shared repository.
- From the shared repository on GitHub.com, clone the repository down to your local machine.
- Edit the “Group Assignment” markdown file in your text editor.
- Save it out, stage the changes and push to the Module 3 repository.
- Resolve any merge conflicts that arise.



Exercise:

1. Find an active open source project.
2. Point to 2 pieces of evidence that the project is active.
3. These pieces of evidence can be:
 - How quickly pull requests are reviewed
 - Data in Insights
 - Number of forks or stars
 - Some other metric :)

Reflect on how you might use these insights to assess collaboration in group work.

