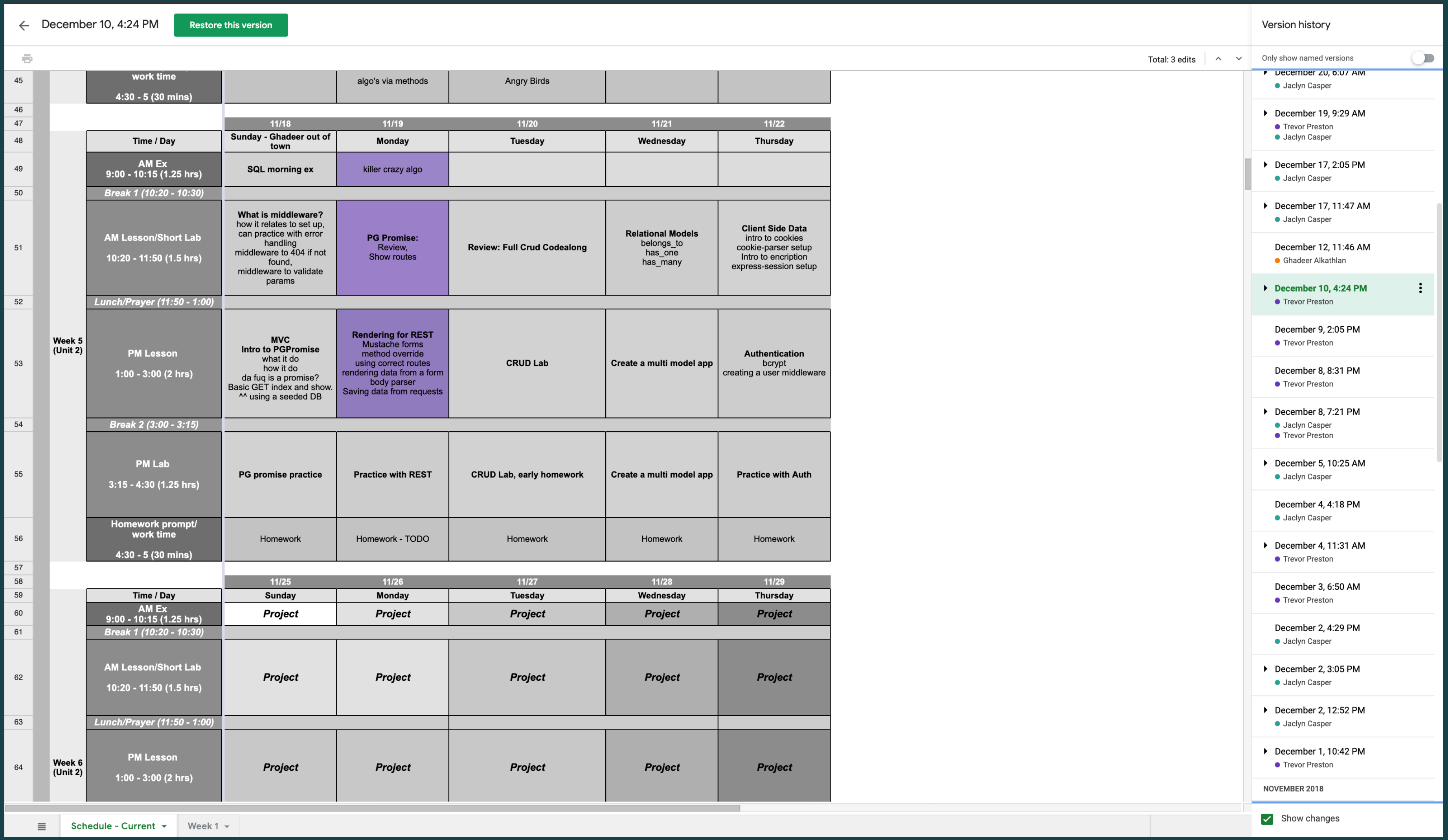
GIT & GITHUB

WHAT IS GIT

- Git is a version control system.
- You probably have used some before, like dropbox or Google docs/sheets.
- Have you ever seen something like this?

VERSION CONTROL



December 10, 4:24 PM

Total: 3 edits

45		work time 4:30 - 5 (30 mins)		algo's via methods	Angry Birds		
46							
47			11/18	11/19	11/20	11/21	11/22
48	Week 5 (Unit 2)	Time / Day	Sunday - Ghadeer out of town	Monday	Tuesday	Wednesday	Thursday
49		AM Ex 9:00 - 10:15 (1.25 hrs)	SQL morning ex	killer crazy algo			
50		Break 1 (10:20 - 10:30)					
51		AM Lesson/Short Lab 10:20 - 11:50 (1.5 hrs)	What is middleware? how it relates to set up, can practice with error handling middleware to 404 if not found, middleware to validate params	PG Promise: Review, Show routes	Review: Full Crud Codealong	Relational Models belongs_to has_one has_many	Client Side Data intro to cookies cookie-parser setup Intro to encription express-session setup
52		Lunch/Prayer (11:50 - 1:00)					
53		PM Lesson 1:00 - 3:00 (2 hrs)	MVC Intro to PGPromise what it do how it do da fuq is a promise? Basic GET index and show. ^^ using a seeded DB	Rendering for REST Mustache forms method override using correct routes rendering data from a form body parser Saving data from requests	CRUD Lab	Create a multi model app	Authentication bcrypt creating a user middleware
54		Break 2 (3:00 - 3:15)					
55		PM Lab 3:15 - 4:30 (1.25 hrs)	PG promise practice	Practice with REST	CRUD Lab, early homework	Create a multi model app	Practice with Auth
56		Homework prompt/ work time 4:30 - 5 (30 mins)	Homework	Homework - TODO	Homework	Homework	Homework
57							
58			11/25	11/26	11/27	11/28	11/29
59	Week 6 (Unit 2)	Time / Day	Sunday	Monday	Tuesday	Wednesday	Thursday
60		AM Ex 9:00 - 10:15 (1.25 hrs)	*Project*	*Project*	*Project*	*Project*	*Project*
61		Break 1 (10:20 - 10:30)					
62		AM Lesson/Short Lab 10:20 - 11:50 (1.5 hrs)	*Project*	*Project*	*Project*	*Project*	*Project*
63		Lunch/Prayer (11:50 - 1:00)					
64		PM Lesson 1:00 - 3:00 (2 hrs)	*Project*	*Project*	*Project*	*Project*	*Project*

Version history

Only show named versions

December 20, 6:07 AM
● Jaclyn Casper

December 19, 9:29 AM
● Trevor Preston
● Jaclyn Casper

December 17, 2:05 PM
● Jaclyn Casper

December 17, 11:47 AM
● Jaclyn Casper

December 12, 11:46 AM
● Ghadeer Alkathlan

December 10, 4:24 PM
● Trevor Preston

December 9, 2:05 PM
● Trevor Preston

December 8, 8:31 PM
● Trevor Preston

December 8, 7:21 PM
● Jaclyn Casper
● Trevor Preston

December 5, 10:25 AM
● Jaclyn Casper

December 4, 4:18 PM
● Jaclyn Casper

December 4, 11:31 AM
● Trevor Preston

December 3, 6:50 AM
● Trevor Preston

December 2, 4:29 PM
● Jaclyn Casper

December 2, 3:05 PM
● Jaclyn Casper

December 2, 12:52 PM
● Jaclyn Casper

December 1, 10:42 PM
● Trevor Preston

NOVEMBER 2018

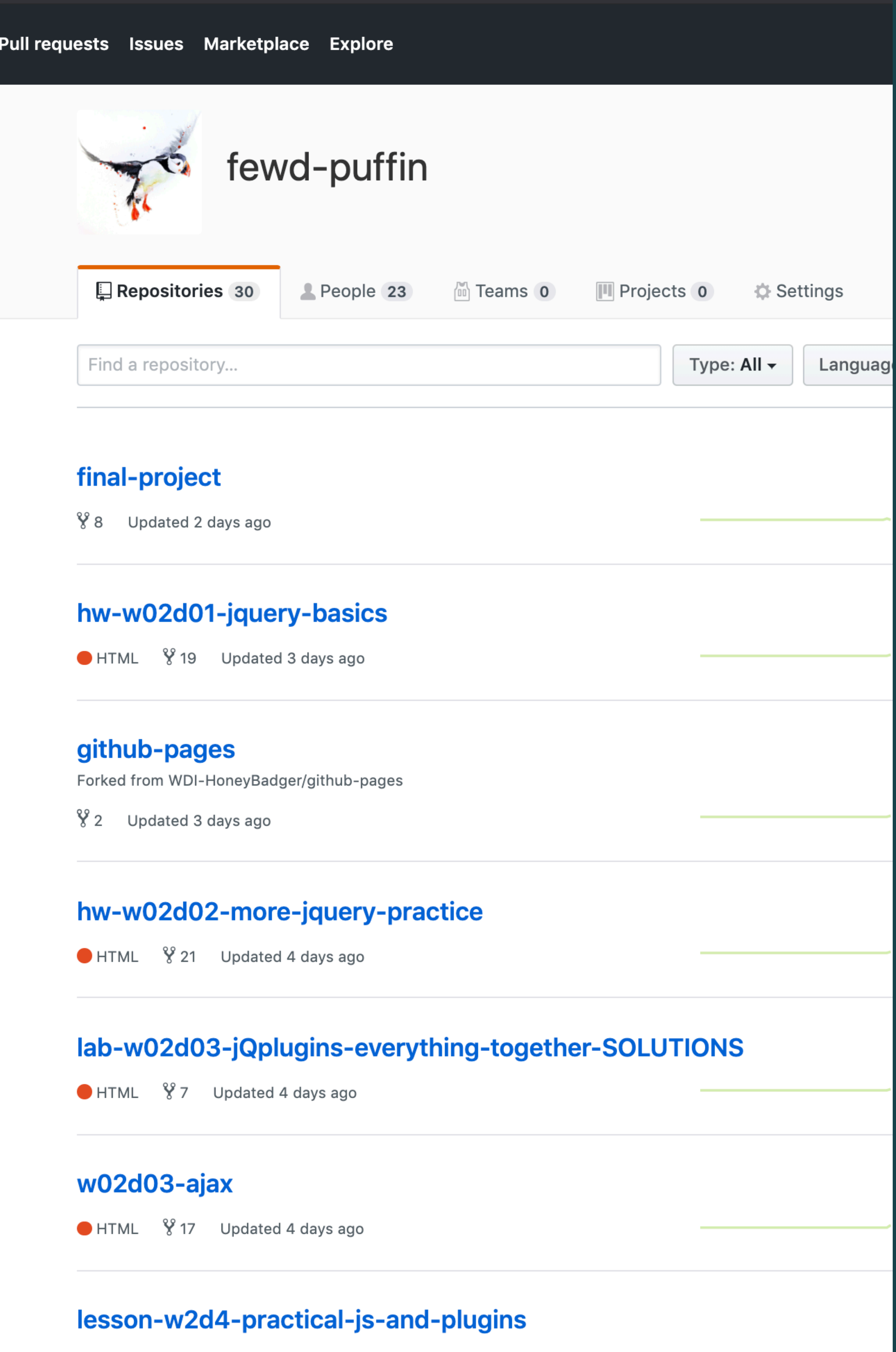
☑ Show changes

- We can look on the right side and see every time anyone has changed the document and what exactly they changed.
- Git is like this, but for software - developers use this to log their changes and roll out new code. And just like this, you can roll back any changes that aren't wanted.

GIT

- The key thing to remember here is that Git is a special type of software for version control.
- It was invented in 2005 by Linus Trovalds, the creator of Linux.
- It is free and open source.

DOES ANYONE KNOW THE DIFFERENCE BETWEEN GIT AND GITHUB?



final-project
8 Updated 2 days ago

hw-w02d01-jquery-basics
● HTML 19 Updated 3 days ago

github-pages
Forked from WDI-HoneyBadger/github-pages
2 Updated 3 days ago

hw-w02d02-more-jquery-practice
● HTML 21 Updated 4 days ago

lab-w02d03-jQplugins-everything-together-SOLUTIONS
● HTML 7 Updated 4 days ago

w02d03-ajax
● HTML 17 Updated 4 days ago

lesson-w2d4-practical-js-and-plugins

GITHUB

- Github is a website that provides an interface for people and teams to interact with projects using Git.
- It’s a VERY handy and VERY popular tool to help manage code for projects.
- Github also lets you host projects in the cloud with something called “repositories”.

GITHUB – SOME VOCAB

- Repository
  - Remote Repository
  - Local Repository
- Forking
- Cloning
- Staging
- Commit
- Push

REPOSITORY (REPO)

- Basic element of Github.
- A repository, or “repo”, means that you are using Git software to track changes in your code.
- A repo can:
  - Be online on something like Github, ***this is called a “remote repo”.***
  - Be on your computer, ***this is called a “local repo”***
- ***The primary job of any repo is to host code and track changes to the code.*** Remote repos are online, local repos are on a computer.

REMOTE REPO'S

- Putting code that you made on your computer into a remote repo gives other people access to that code.
- Remote repo's have many configuration options:
  - You can make them private or public.
  - You can give specific people privileges to change the code.
  - You can manage any requests to change the code.

GENERAL REMOTE REPO RULES

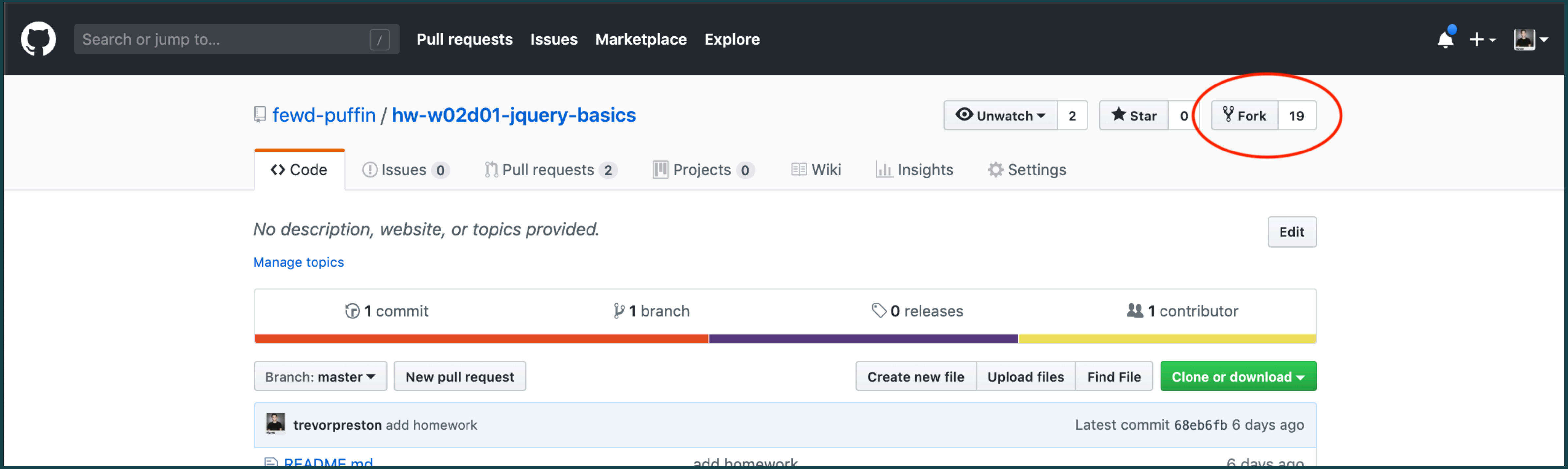
- A remote repo that is owned by someone else won't let you add code to it automatically. They need to approve any additions or make you a co-owner of the repo.
- A remote repo that you own can be updated by you as much as you want.
- Any changes that someone else wants to make to your repo's need to be approved by you.

FORKING

- It can be frustrating if you want to add code to a repo owned by someone else because they have the right to deny your code.
- To solve this, you can fork their code.
- *Forking is a way of copying a repo that belongs to someone else to a new repo (with all the same code) that belongs to you.*

FORKING - HOW TO

- Github makes forking really easy. All you have to do is click the “fork” button in the top right corner of a repo.
- BAM now you own your own version of the original code. You have full access to write to this remote repo.



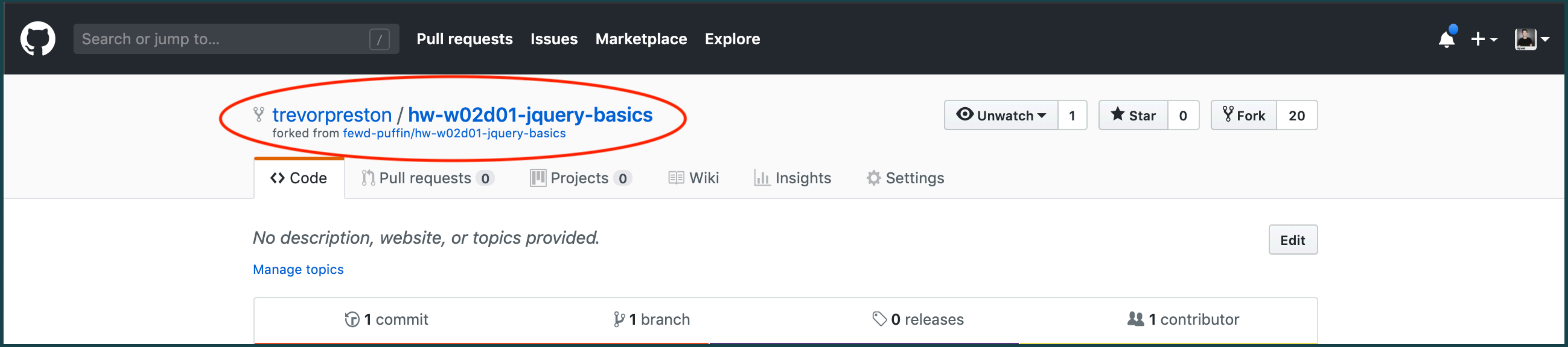
No description, website, or topics provided.

trevorpreston add homework — Latest commit 68eb6fb 6 days ago

README.md — add homework — 6 days ago

FORKING

- You’ll see a pretty little animation of a copying machine, this is how you know it is forking.
- When the fork is done you’ll see this in the top left corner:



trevorpreston / hw-w02d01-jquery-basics

forked from fewd-puffin/hw-w02d01-jquery-basics

No description, website, or topics provided.

- Your username next to the name of the repo, and “forked from” message showing where it came from.

CLONING & LOCAL REPOS

- Now this is great and all, you have some code on GitHub that you can add to. The trouble is that we can't edit code in the browser, so we need to get this on our computers!
- The next thing we need to do is clone the repo.
- ***Cloning is how we copy a remote repository to our computer.***
- This creates a "local repository" which contains all the code from the remote, which we can edit and add to as we wish.
- A local repo is just a folder on your computer that has git version control.

LOCAL REPOS

- A local repo will have all the code from the remote in a folder on your computer.
- A local repo will remember the remote repo that it was cloned from.
- This connection lets us add code to the remote after any work we do.

WHOA THAT WAS A LOT OF INFO. LET'S REFLECT:

- What is Git?
- What is Github?
- What is a repo?
- What does it mean to “fork a repo”?
- What does it mean to “clone a repo”?
- What is the difference between a remote repo owned by someone else and a remote repo owned by you?

ADDING CODE TO YOUR REMOTE – STEP 1: STAGING

- Once a developer (you!) has added code they need to get it online to their remote for sharing.
- There are 3 steps to this, the first one is adding the files you want to staging.
- ***Staging is where your code goes before being remembered by git***
- You can do this by doing:
  - `git add (file names)`
- Or instead of adding each file individually, it's usually faster to add them all at once. To do this go to the root of your local repo and use the command:
  - `git add .`
- Where the `.` represents all files in the directory

ADDING CODE TO YOUR REMOTE – STEP 1: STAGING

- You can add code to staging with the command:

  -
    ```
    git add (file names)
    ```
- Or instead of adding each file individually, it's usually faster to add them all at once. To do this go to the root of your local repo and use the command:

  -
    ```
    git add .
    ```
- Where the `.` represents all files in the directory

ADDING CODE TO YOUR REMOTE – STEP 2: COMMIT

- The next step is to ***commit*** your code.
- ***Making a commit means that you are saving your code in the .git file***
- This .git file is where all your version information is saved.
- Making a commit means you are telling the .git file to remember the code you had staged and save it.

ADDING CODE TO YOUR REMOTE – STEP 2: COMMIT

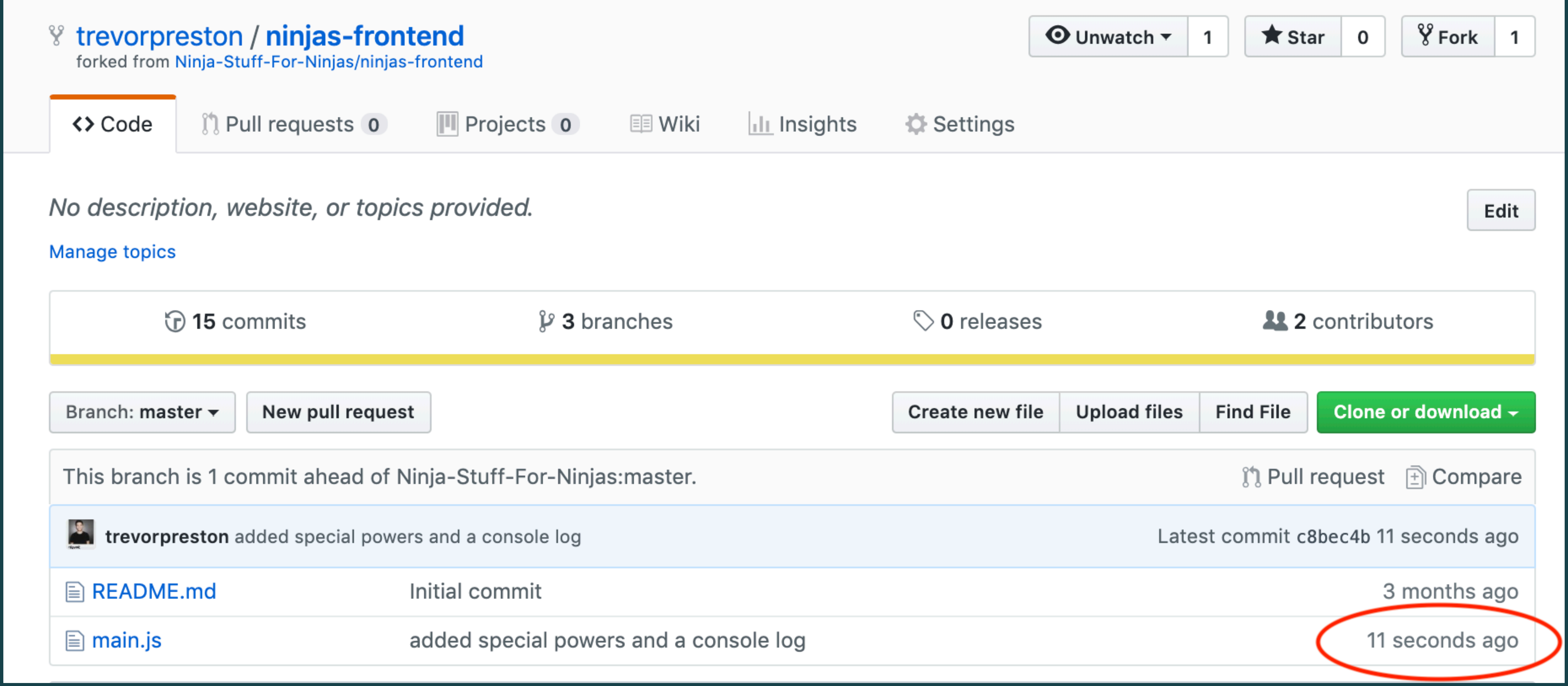
- To write a commit, use the command:
  - `git commit -m “commit message”`
- Let’s break this command apart:
  - ‘git’ - we are using a git command
  - ‘commit’ - we are “committing” code to be remembered
  - ‘-m’ - this means add a short message about the commit
  - “blah blah blah” - some short message about what your commit is for.
    - Good examples are, “add header to index.html”, “add homework w1d1”, “update about me section”, etc.

ADDING CODE TO YOUR REMOTE – STEP 3: PUSHING

- The final step is to ***push*** your code to the remote.
- This is the final step! To do a push use the command:
  - `git push origin master`
- In your console you can see a few messages cataloging the compression and sending of your code to the remote repo.

ADDING CODE TO YOUR REMOTE – STEP 4: VERIFICATION

- It worked! We think. You can tell it worked by going to the remote and seeing that the files changed there recently:



trevorpreston / ninjas-frontend
forked from Ninja-Stuff-For-Ninjas/ninjas-frontend

No description, website, or topics provided.

Manage topics

15 commits · 3 branches · 0 releases · 2 contributors

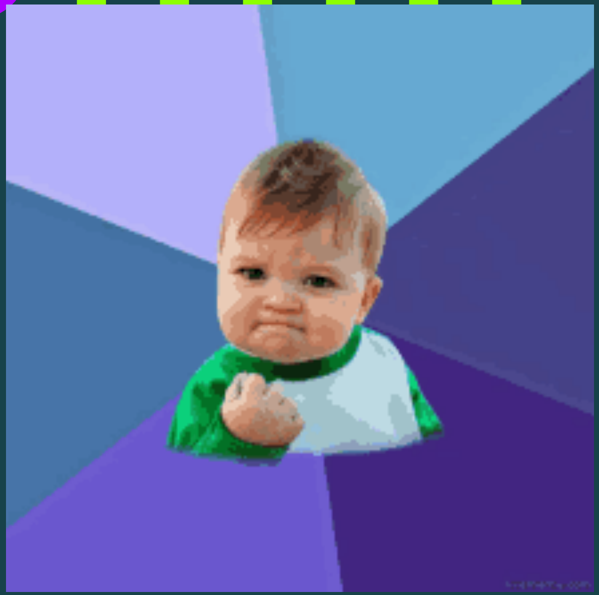
This branch is 1 commit ahead of Ninja-Stuff-For-Ninjas:master.

trevorpreston added special powers and a console log — Latest commit c8bec4b 11 seconds ago

README.md	Initial commit	3 months ago
main.js	added special powers and a console log	11 seconds ago

- Changed 11 seconds ago! It worked!

SUCCESS!!!!!!



GITHUB STEPS:

- Bear in mind that there are times where certain steps will happen in different orders. It’s a situational thing, but this flow is what we’ll be doing in this class.
- Over
- And over
- And over
- And over
- So don’t stress if you didn’t get this right away. We’ll practice this every day!

- What is staging?
- What is a commit?
- What happens when you push to a repo?
- What is the .git file for?
- How can we check to see if our push worked?

WHOA THAT WAS A LOT OF STEPS

Let’s review:

GITHUB STEPS (FOR THIS CLASS AND MORE!)

- Step 1: fork the repo (in the browser)
  - Button on the top right of GitHub
- Step 2: clone the repo (in the command line)
  - `git clone (https code from GitHub)`
- Step 3: update/add code (in VSCode)
  - `git add .`
- Step 4: commit the code (command line)
  - `git commit -m “commit message”`
- Step 5: push the code (command line)
  - `git push origin master`

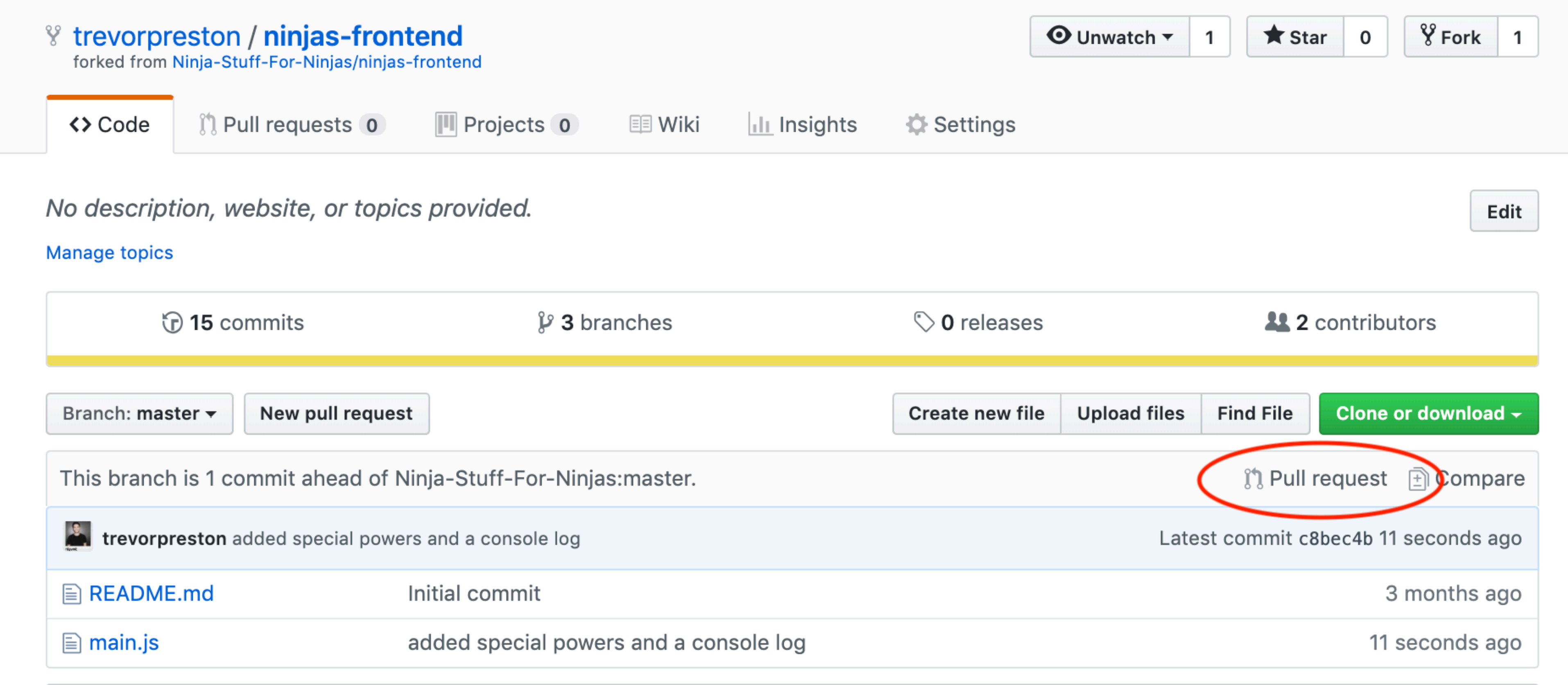
CODEALONG

Let's do this together!

PULL REQUESTS

- So we now know how to:
    - Make a copy of someone’s code to your own repo with forking.
    - How to make a local repo of your remote repo with cloning
    - How to update your remote repo with new code with adding, committing and pushing.
- What if you want to change the code in the original repo you forked?
- You can’t make changes to it directly, but you can request to make updates to their code with a ***pull request***

PULL REQUESTS

- *A pull request is how we ask other GitHub users permission to add code to their repo.*
- To do this, we:
  - Push your code to your repo.
  - Click on the pull request button:
    - 

trevorpreston / ninjas-frontend
forked from Ninja-Stuff-For-Ninjas/ninjas-frontend

No description, website, or topics provided.

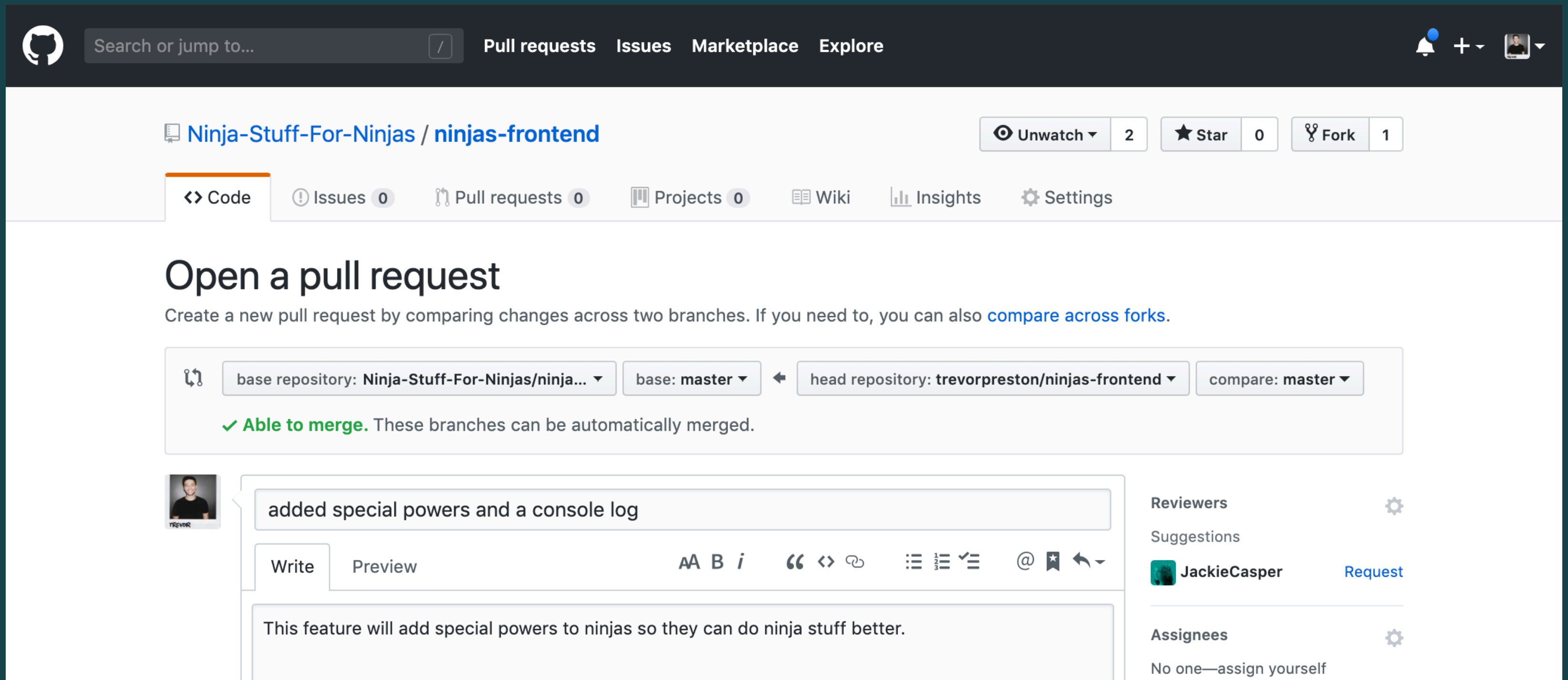
This branch is 1 commit ahead of Ninja-Stuff-For-Ninjas:master.

trevorpreston added special powers and a console log — Latest commit c8bec4b 11 seconds ago

README.md	Initial commit	3 months ago
main.js	added special powers and a console log	11 seconds ago

PULL REQUESTS

- This will then lead you to a screen with a big green button that says “create pull request”. Clicking that button will lead you to a short form for you to fill out.
- This form is where you typically tell the owner of the original repo what your code addition does.
- Once you add a comment, you hit the submit button and the owner can see the code that you want to add.



Ninja-Stuff-For-Ninjas / ninjas-frontend

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base repository: Ninja-Stuff-For-Ninjas/ninja... base: master ← head repository: trevorpreston/ninjas-frontend compare: master

✓ Able to merge. These branches can be automatically merged.

added special powers and a console log

This feature will add special powers to ninjas so they can do ninja stuff better.

Reviewers
Suggestions
JackieCasper Request

Assignees
No one—assign yourself

CODEALONG

Let's do this together!

GITHUB LAB

GITHUB LAB

- Use the repo in slack and do the following:
- Fork
- Clone
- Edit index.html
- Add
- Commit
- Push
- And open a pull request to the original repo.