

# Introduction to Git and Github

Alex Seong

12. 22. 2018

# Objectives:

Today's objectives:

- Explain the basic function and purpose of version control
- Become familiar with the basic/fundamental git commands and what each is used for:
  1. fork
  2. clone
  3. add, commit
  4. push, pull
  5. pull request (less fundamental, but used frequently throughout the program)

# Agenda

Today's plan:

1. Version control and Git/Github Overview
2. Intro to Git Commands

# Why does this matter?

- Version Control:
  - Using some sort of version control system is nearly universal in the tech and data science world, and Git is one of the most common.
- Git/Github:
  - Allows you to take advantage of version control and everything that it offers.
  - Github (built on Git) will allow you to share your personal projects as well as collaborate on projects with others.

# Version Control

- A **version control system** is a repository of files with monitored access
  - Files are primarily source code, but can be of other types
- Version control is useful primarily because every change to a repository is tracked, along with who made it, why they made it, and references to any problems fixed or enhancements made by the change.
  - Provides the ability to track changes over time, and the ability to reverse any of them if necessary
  - Allows for easy collaboration across teams

# Version Control Systems

- Numerous version control systems exist, but four of the most popular are:
  1. Git (distributed)
  2. Mercurial (distributed)
  3. Subversion (centralized)
  4. Concurrent Versions Systems (centralized)
  5. With centralized systems, you have to be connected to the central repository at all times (e.g. you typically need an Internet connection), while with distributed systems there is more freedom to work locally.

# Why Git?

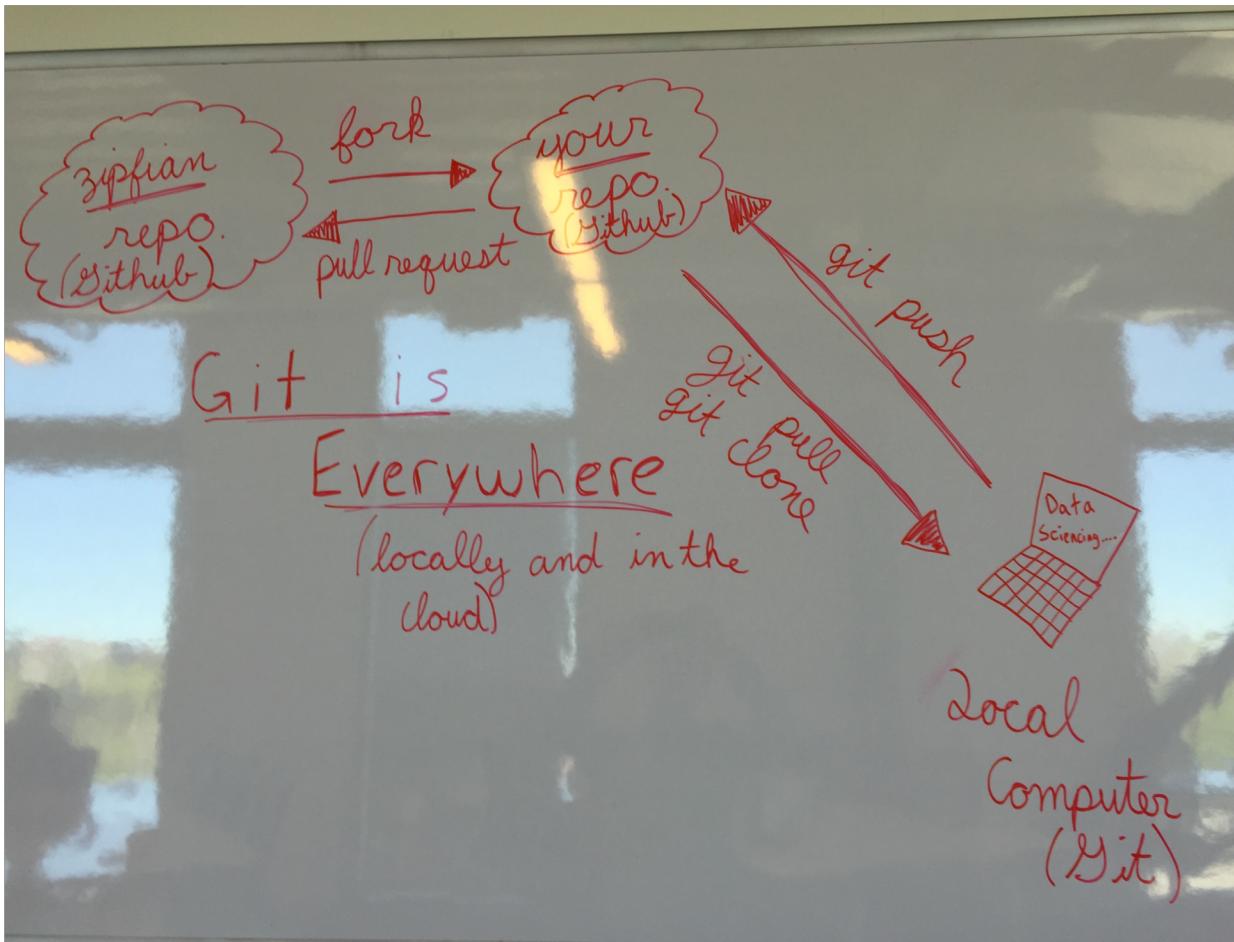
- Why use Git?
  - Distributed (allows more freedom to work locally)
  - Free, open source
  - Collection of hosting services for Git repositories (Bit Bucket, Github)
  - Arguably the most popular

# Git v. Github

- What's the difference between Git v. Github?
  - Git is a version control system, while Github is a web-base hosting service for Git repositories (e.g. Github is “in the cloud”, whereas Git works locally).
  - Git exists independently of Github, while the converse is not true.

# Git v. Github Part Dos

- What if we visualize how Git and Github interact?



# Intro Git Commands

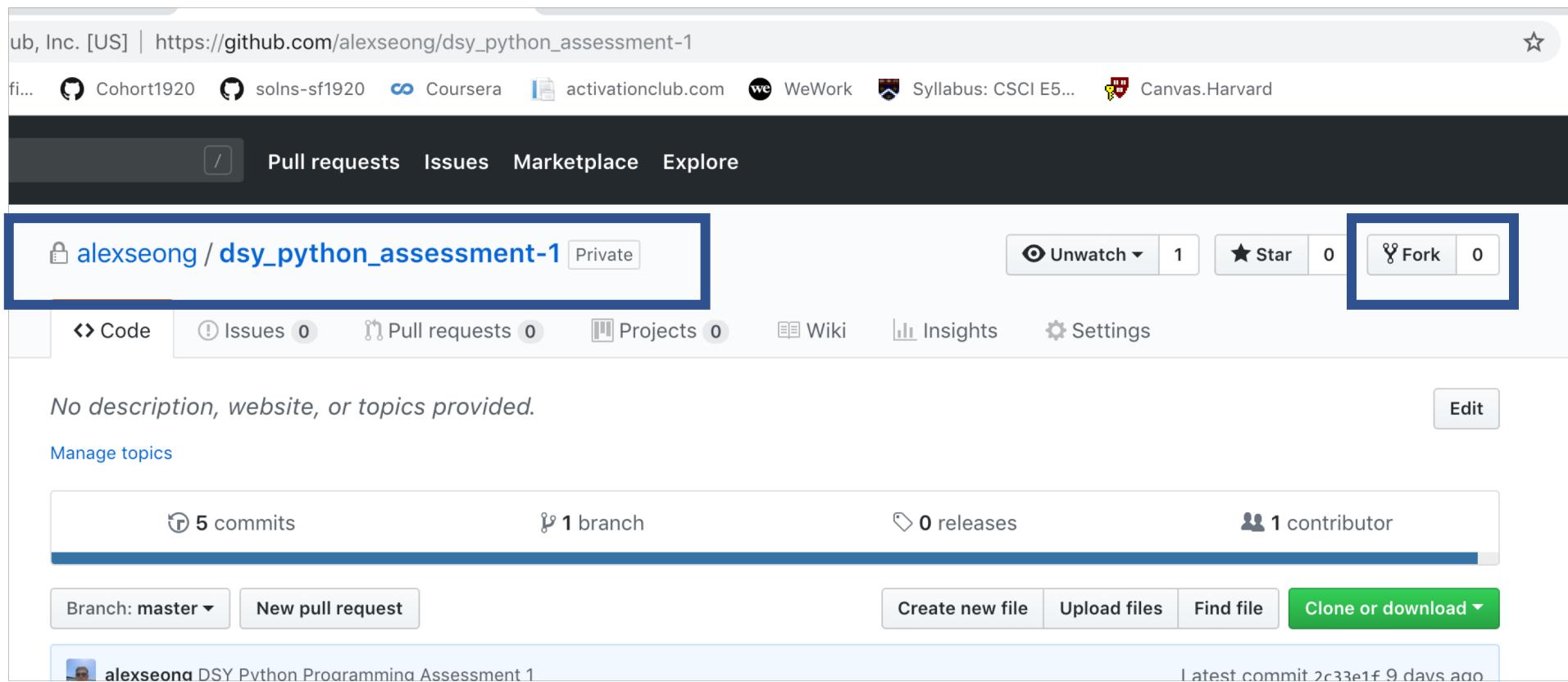
- Major Git commands include:
  - clone: Makes a copy (clone) of a repository into a newly created directory, with a reference still pointing to the original repository.
  - add: Add one or more files to the index (e.g. tell Git to keep track of these files)
  - commit: Commit your changes, creating a “checkpoint” that can then be referenced or reverted back to later
  - push: Updates a remote copy of the repository with local changes
  - pull: Updates a local copy of the repository with remote changes

# Intro Github Commands

- Major Github (issued from the browser) commands include:
  - fork: Makes a copy of a repository onto your personal Github account (it's like the Github version of cloning), but without a reference still pointing to the original repository.
  - pull request: Issued to try to update a repository with changes from another copy of the repository (a fork or branch)

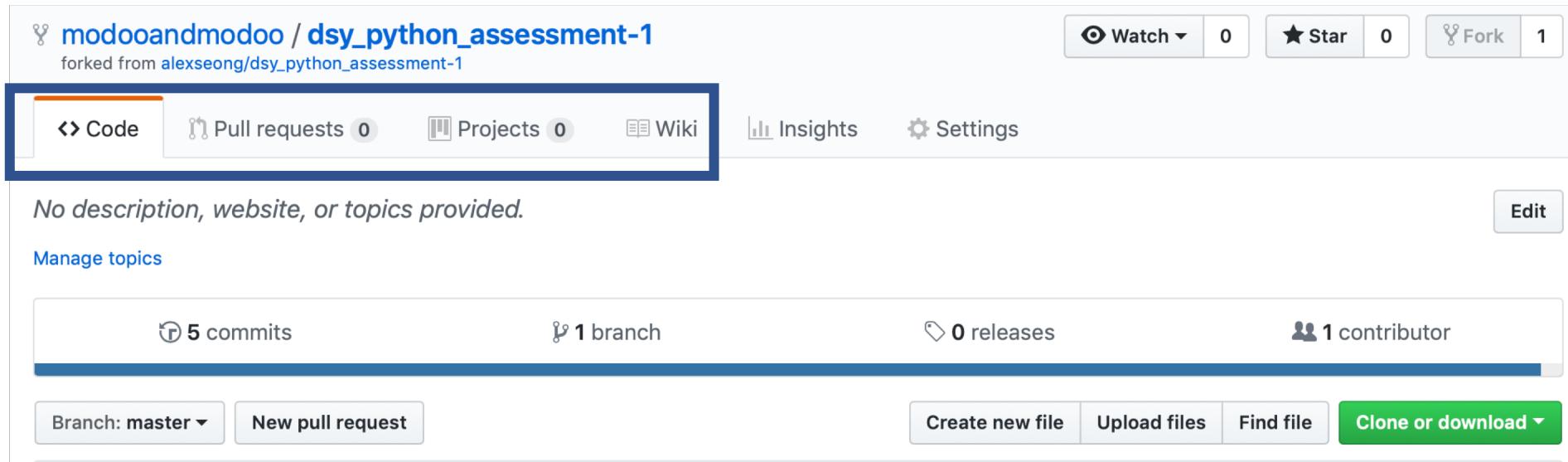
# Classroom Git/Github Workflow

- Let's fork the `dsy_assessment-day1` repository (Note: We're looking at the copy of the repository, and just clicking that Fork button):



# Classroom Git/Github Workflow

- Now that we've forked, let's clone (Note: We're now looking at our personal copy of the repository, issuing this command from our terminal):



- git clone [https://github.com/modooandmodoo/dsy\\_python\\_assessment-1](https://github.com/modooandmodoo/dsy_python_assessment-1)

# Classroom Git/Github Workflow

- So, we issued a fork in our browser to get our own personal copy of some repository (here `dsy_python_assessment-day1`), and then we issue a git clone in our terminal to get that personal copy on our local machine.
- Now, you crush the assessment! Along the way, you'll add, commit, and push any changes you make. Let's say I just finished a part of the assessment in the `assessment.py` file, and I now want to tell Git to make sure that it registers all those changes. . .

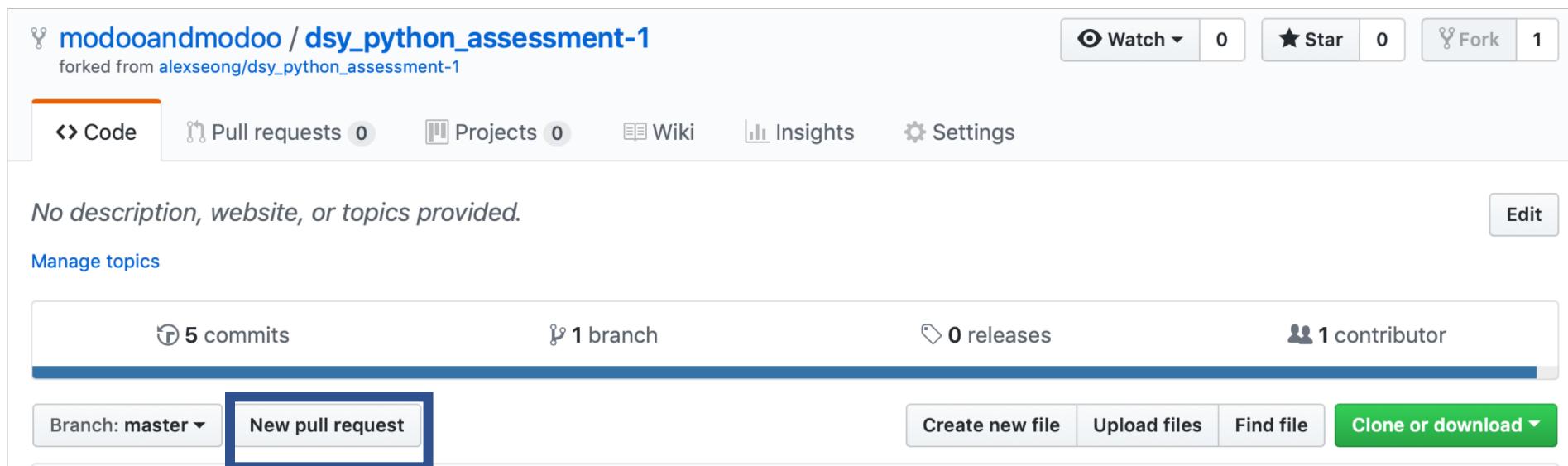
```
git add assessment.py
```

```
git commit -m 'Crush part 1 of assessment.py'
```

```
git push
```

# Classroom Git/Github Workflow

- Now that you've finished the assessment, we'd like to take a look at it. To do so, we're going to have you ask that the changes you've made to your copy of the repository (your forked copy) are used to update our copy (the alexseong copy). This is done via a pull request.



# Classroom Git/Github Workflow

This screenshot shows a GitHub repository page for `zipfian / assessment-day1`. The repository is private, has 55 watchers, 0 stars, and 337 forks. The main navigation bar includes links for Pull requests, Issues, Gist, Unwatch, Star, Fork, Code, Issues (0), Pull requests (1), Wiki, Pulse, Graphs, and Settings.

## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base fork: `zipfian/assessment-day1` base: `master` ... head fork: `sallamander/assessment-d...` compare: `master`

**X Can't automatically merge.** Don't worry, you can still create the pull request.

**Sean Sall - End of assessment #114**  
No description available [View pull request](#)

**Create new pull request** Create another pull request to discuss and review the changes again. [?](#)

# Classroom Git/Github Workflow

- Once you've clicked the Pull Request button, just give the PR a title and a potential message:

The screenshot shows a GitHub pull request interface. At the top, there are dropdown menus for 'base fork' (zipfian/assessment-day1), 'base' (master), 'head fork' (sallamander/assessment-d...), and 'compare' (master). A red 'X' icon indicates 'Can't automatically merge'. Below this, the pull request details are shown for 'Sean Sall - End of assessment #114'. The message input field contains the text 'End of Assessment 1' and 'Sean, I killed this assessment. 👍'. The 'View pull request' button is green. On the right, there are sections for 'Labels' (None yet), 'Milestone' (No milestone), and 'Assignee'.

base fork: zipfian/assessment-day1 ▾ base: master ▾ ... head fork: sallamander/assessment-d... ▾ compare: master ▾

✗ Can't automatically merge. Don't worry, you can still create the pull request.

Sean Sall - End of assessment #114

No description available

View pull request

End of Assessment 1

Write Preview

Sean, I killed this assessment. 👍

Labels

Milestone

Assignee