Github: What, Why, and Welcome

Friendly Github Workshop, UBC - Vancouver

mozilla Science Lab

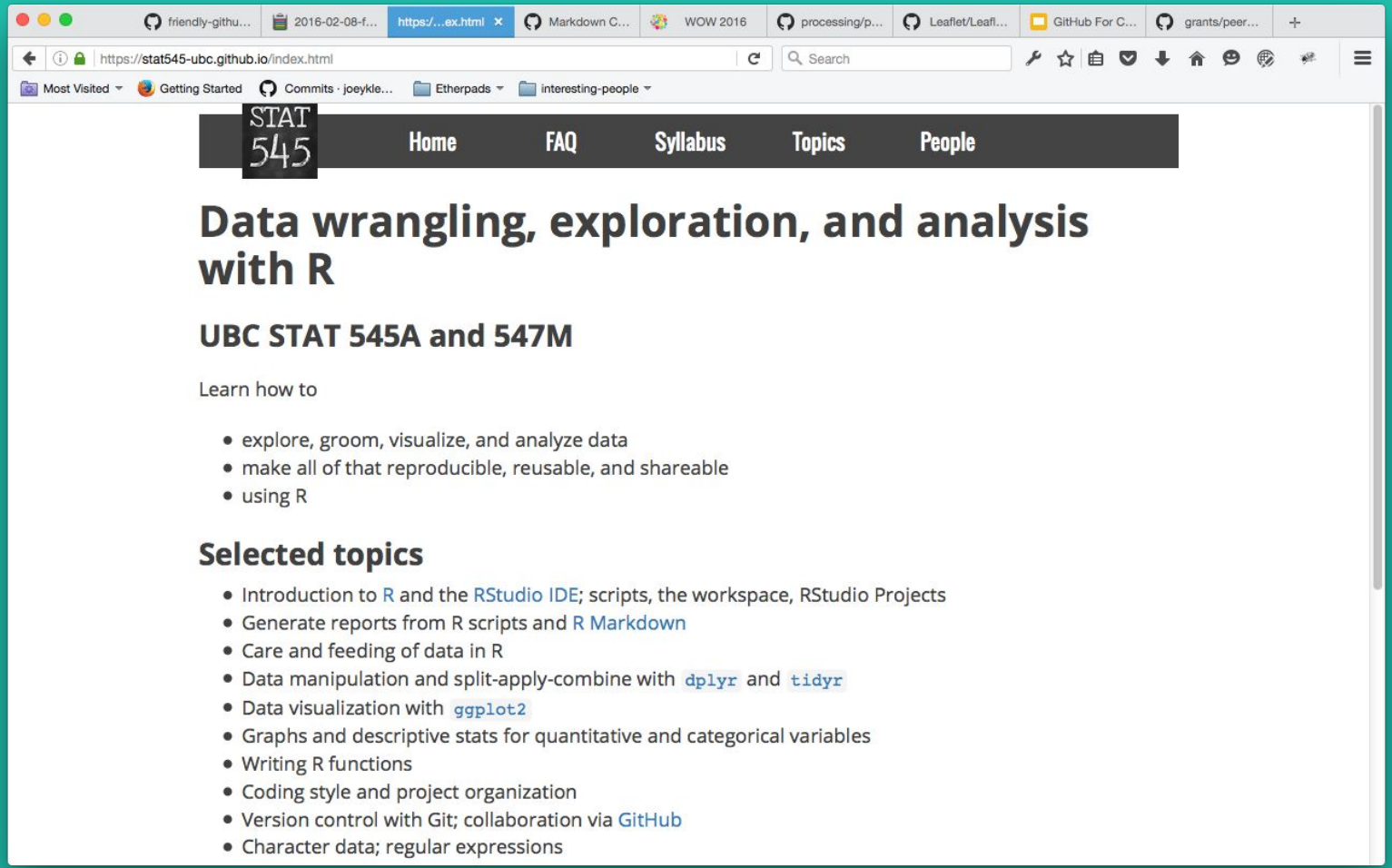
Welcome & Hello!

We’re so happy you’re here!

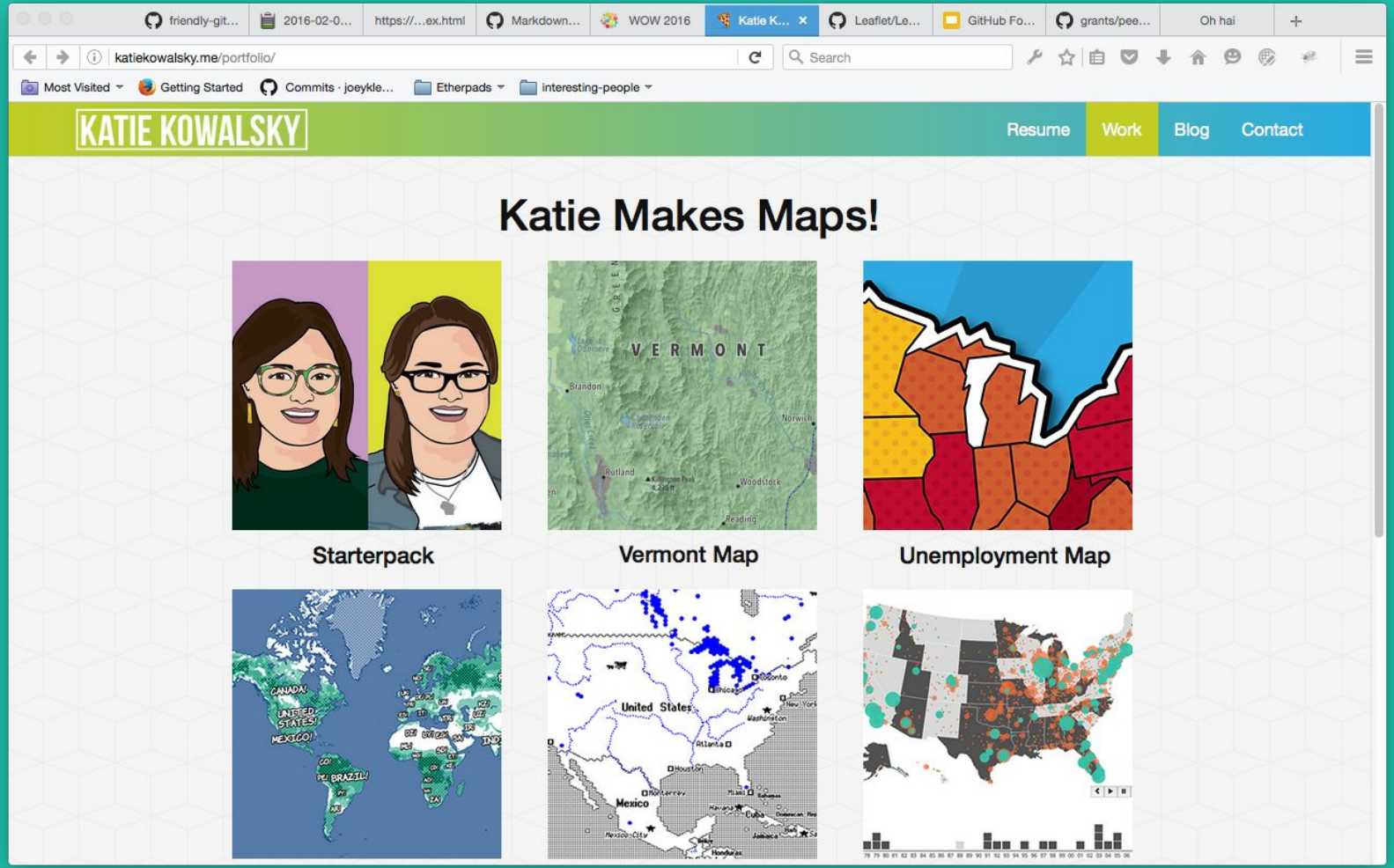
Let’s learn some GitHub together!

Don’t worry if you miss anything - we will be around to clarify.

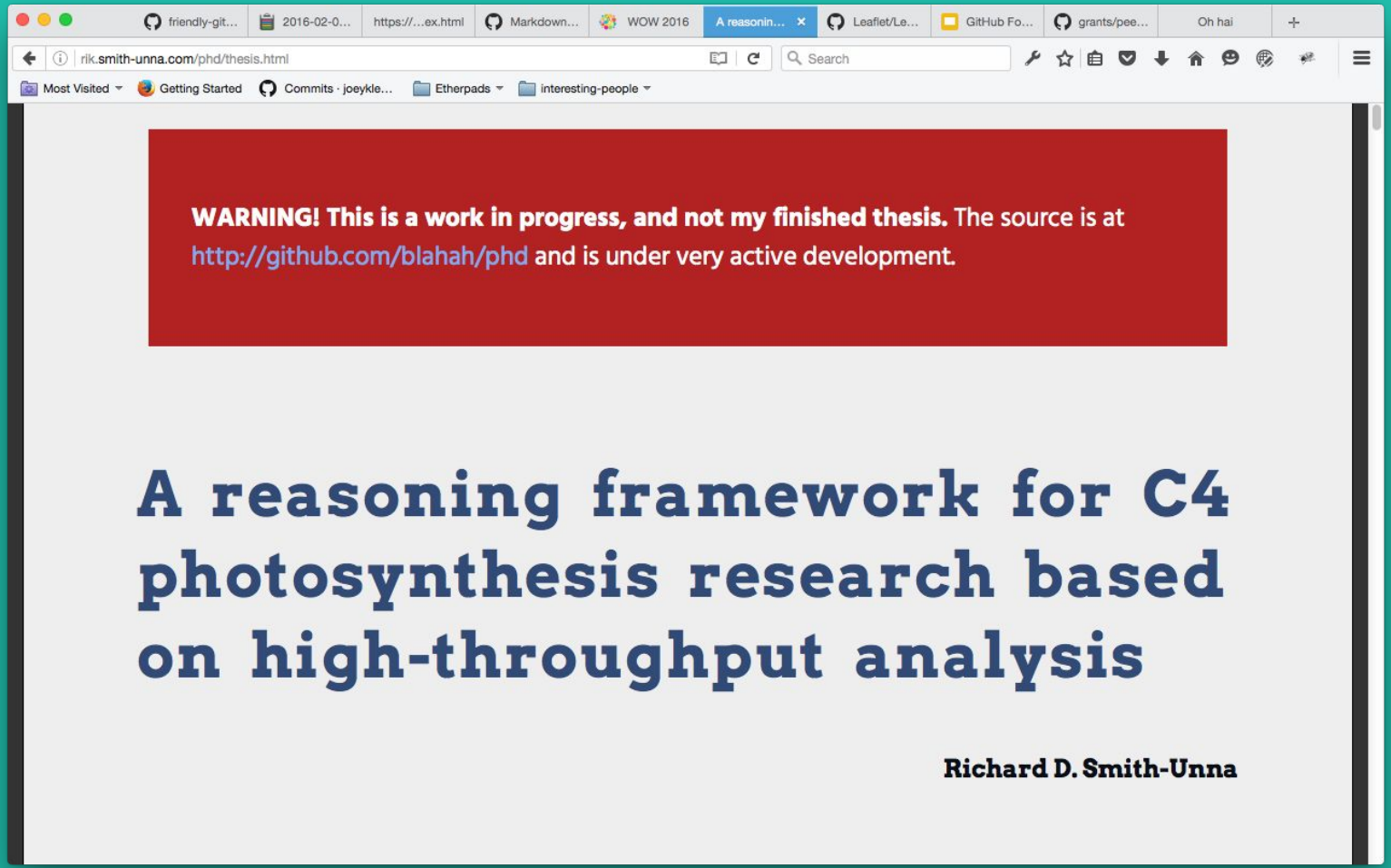
Examples of Git/Github Magic



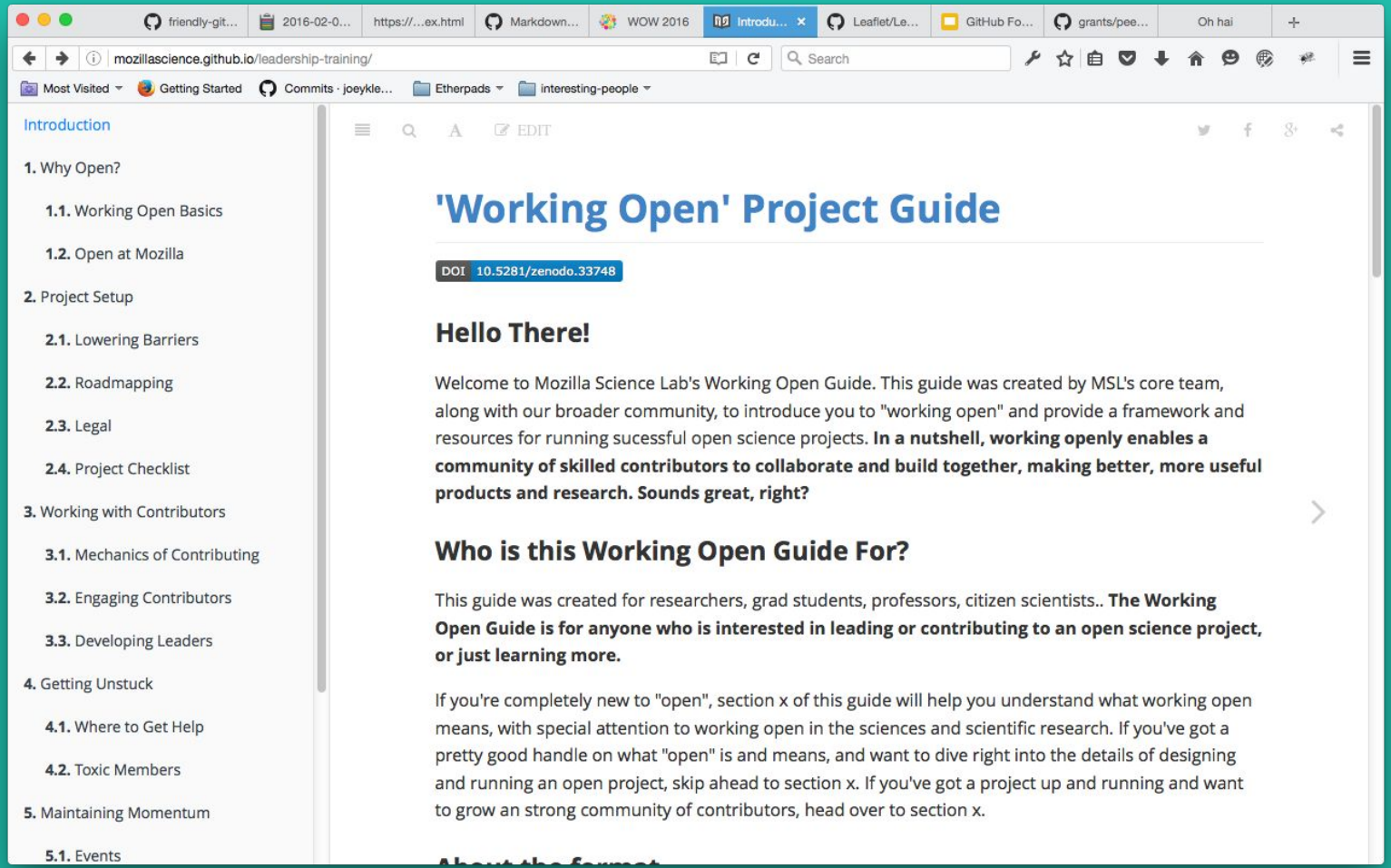
https://stat545-ubc.github.io/faq.html



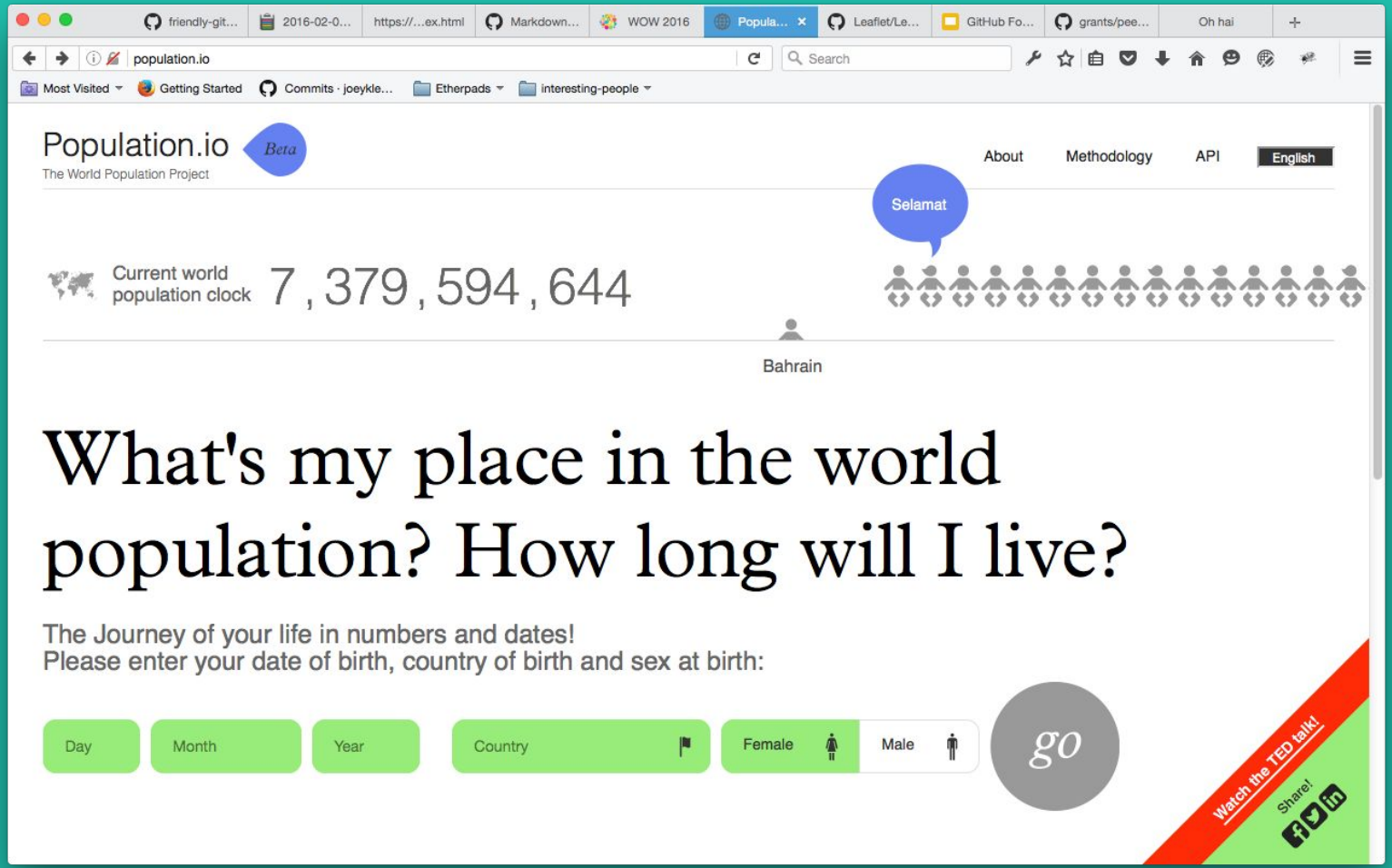
http://katiekowalsky.me/portfolio/



https://github.com/Blahah/phd



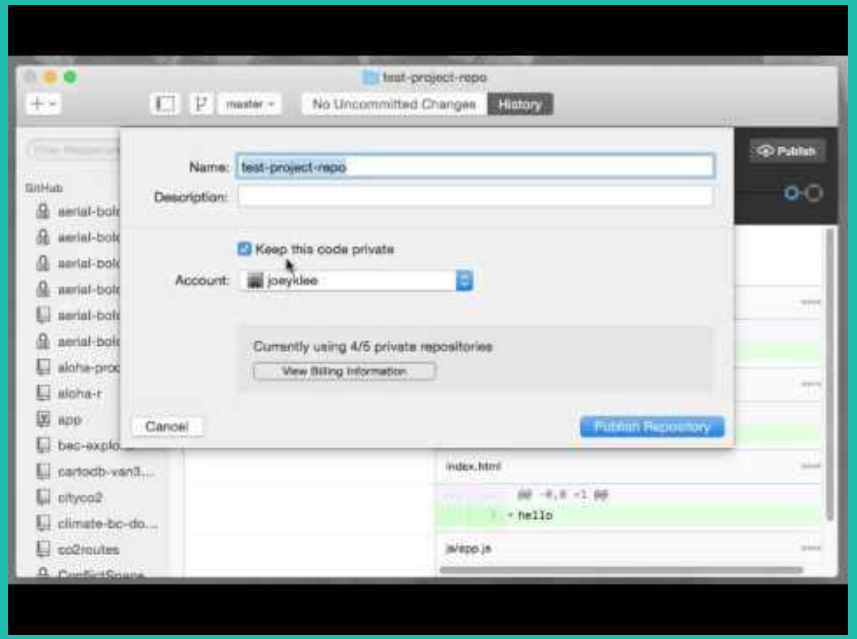
http://mozillascience.github.io/leadership-training/

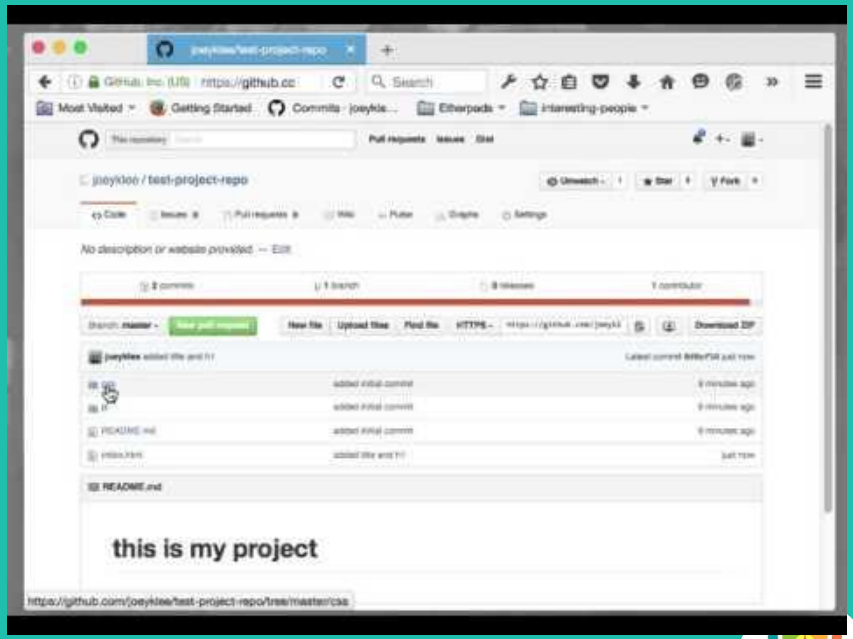


https://github.com/worldpopulation/

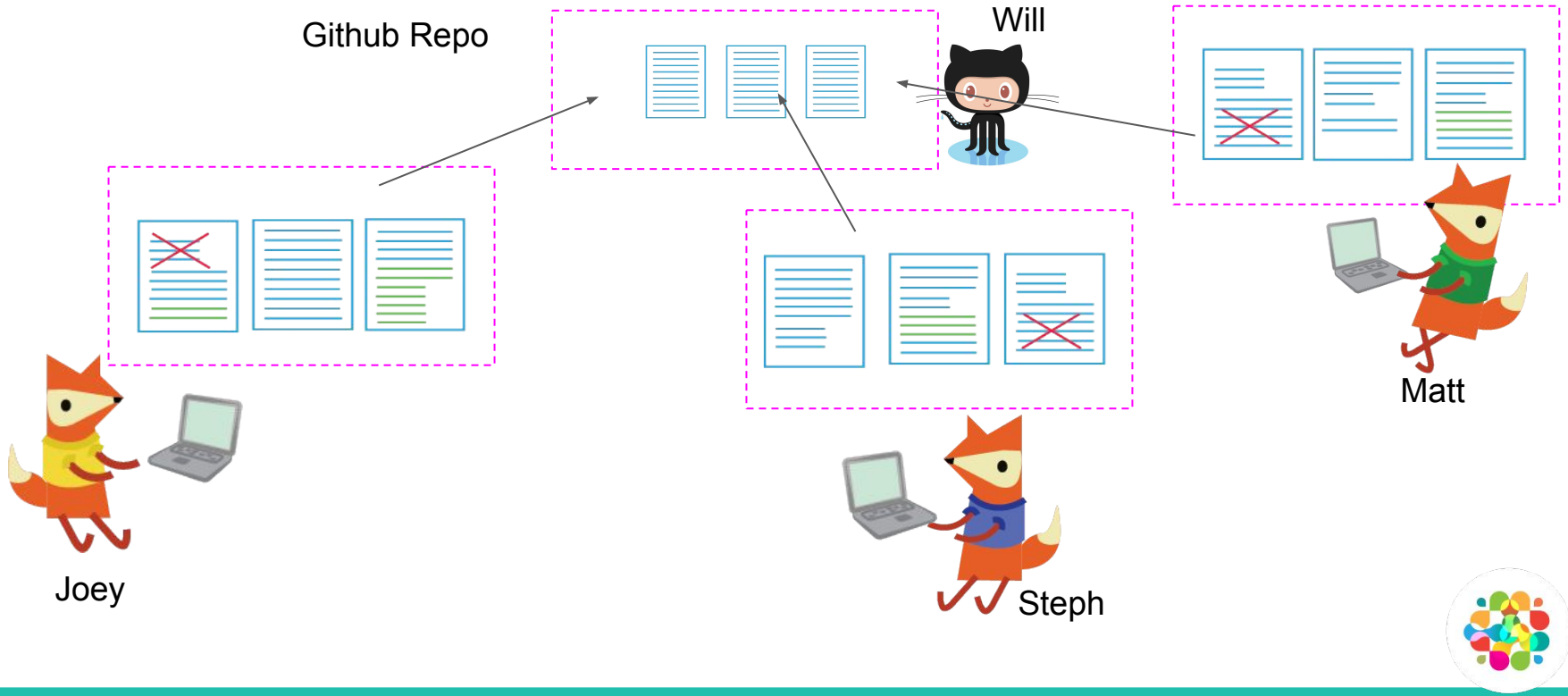
What is Git/GitHub?

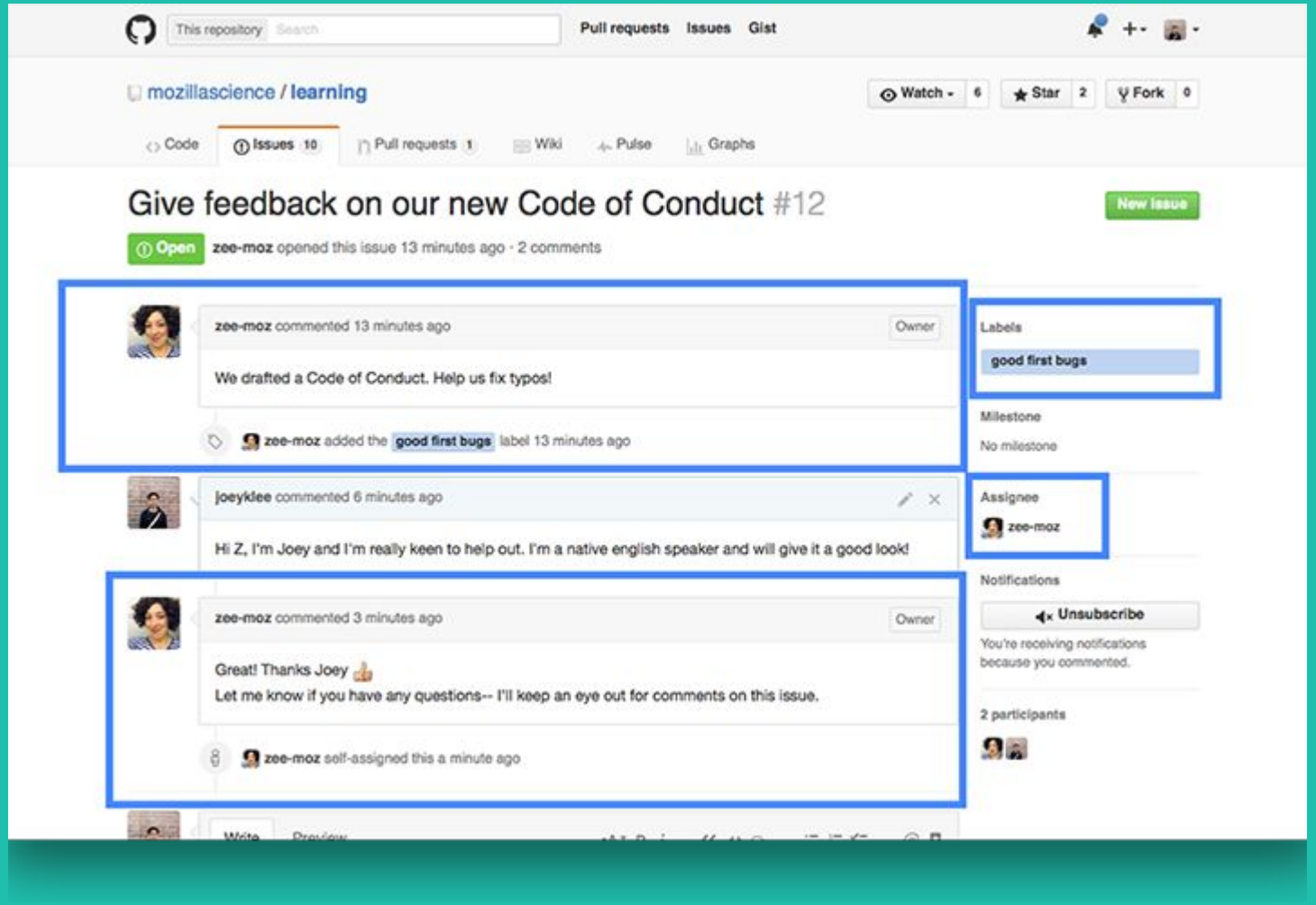
Git/GitHub is… a software!



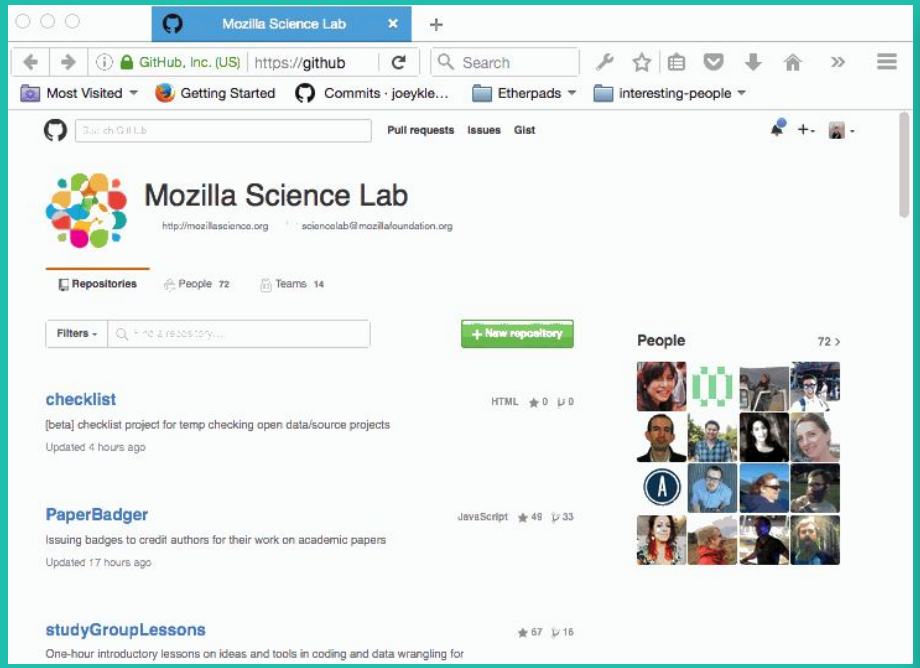


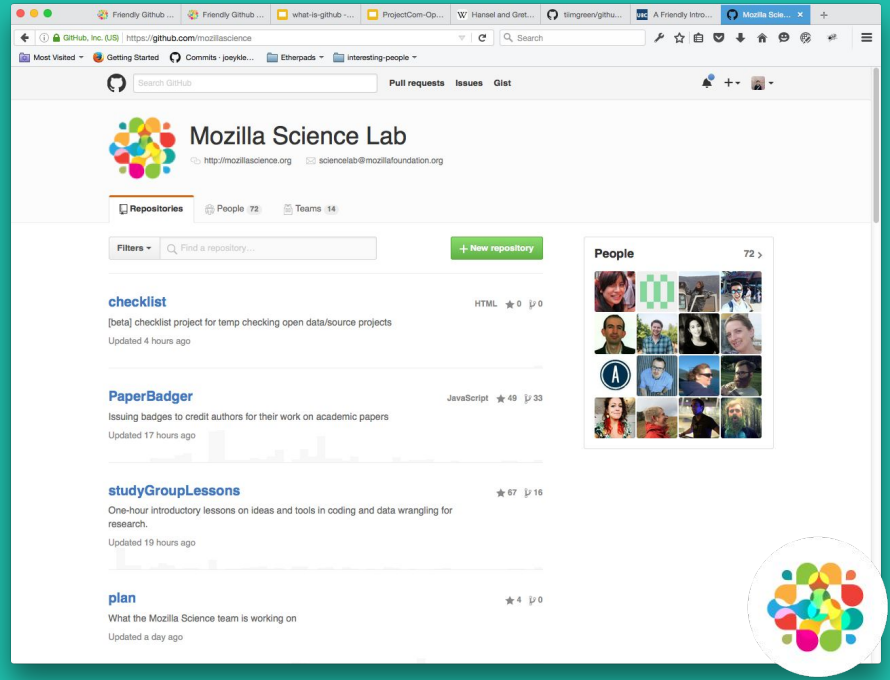
GitHub is… a collaboration tool!




This repository
Search
Pull requests
Issues
Gist
mozillascience / learning
Watch 6
Star 2
Fork 0
Code
Issues 10
Pull requests 1
Wiki
Pulse
Graphs
Give feedback on our new Code of Conduct #12
New issue
Open
zee-moz opened this issue 13 minutes ago · 2 comments
zee-moz commented 13 minutes ago
Owner
We drafted a Code of Conduct. Help us fix typos!
zee-moz added the good first bugs label 13 minutes ago
joeyklee commented 6 minutes ago
Hi Z, I'm Joey and I'm really keen to help out. I'm a native english speaker and will give it a good look!
zee-moz commented 3 minutes ago
Owner
Great! Thanks Joey
Let me know if you have any questions-- I'll keep an eye out for comments on this issue.
zee-moz self-assigned this a minute ago
Write
Preview
Labels
good first bugs
Milestone
No milestone
Assignee
zee-moz
Notifications
Unsubscribe
You're receiving notifications because you commented.
2 participants

GitHub is… an ecosystem of version controlled repositories (projects)!



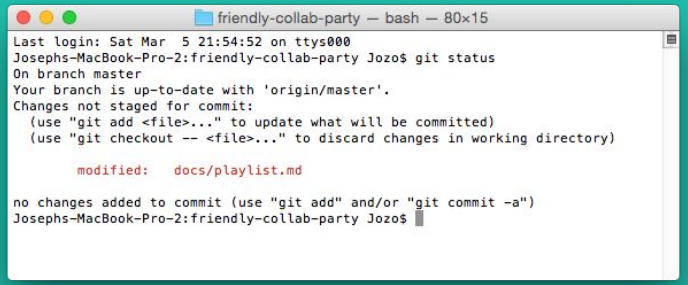


Git vs. Github? Version controlled? Repositories? Whuuuut??

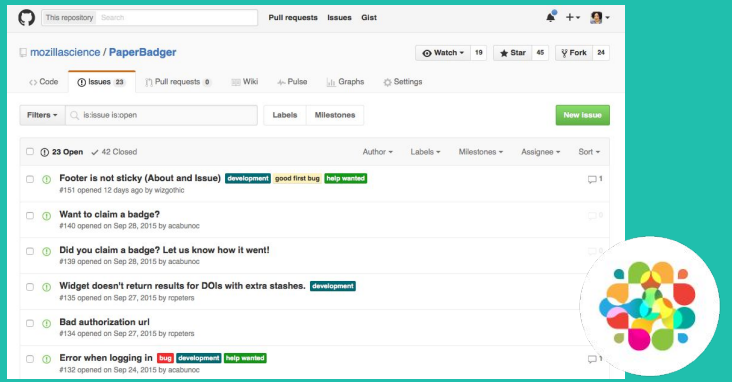
Git/GitHub Terms

Git & Github: How are they related?

- Git is the command line software handles version control on your repository.
- When you use GitHub, Git is always doing its thing behind the scenes.



- GitHub is a service that hosts your repository online and helps you work with contributors/collaborators.
- It's a web interface for version control
- Most often you will interface with Github (online) with the Github Desktop App or via the command line.

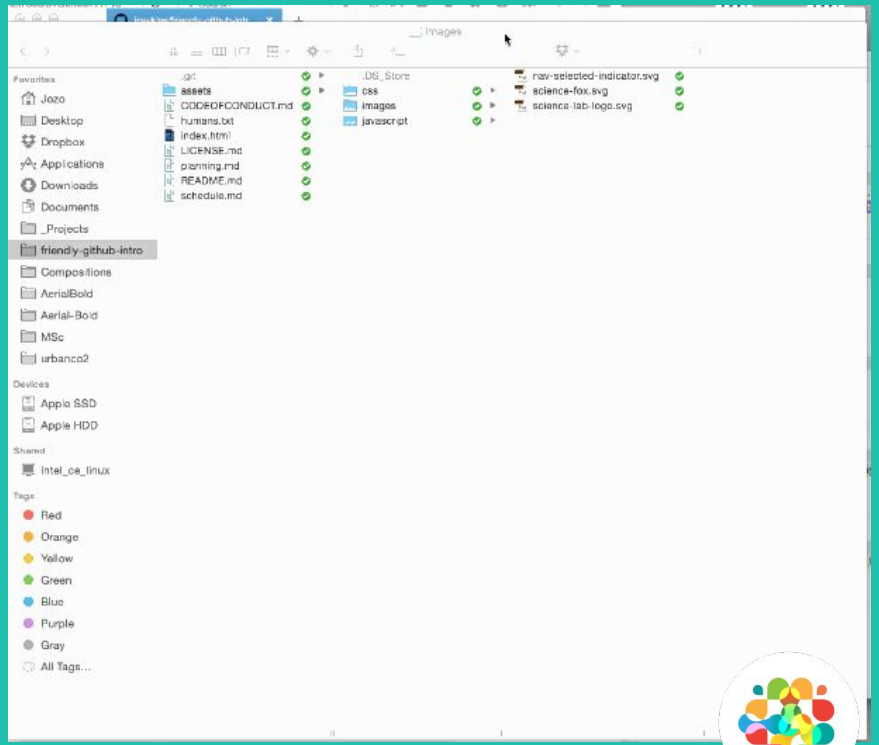


We are going to focus on GitHub online and the GitHub Desktop App.

...but the command line offers full access to git. just fyi.

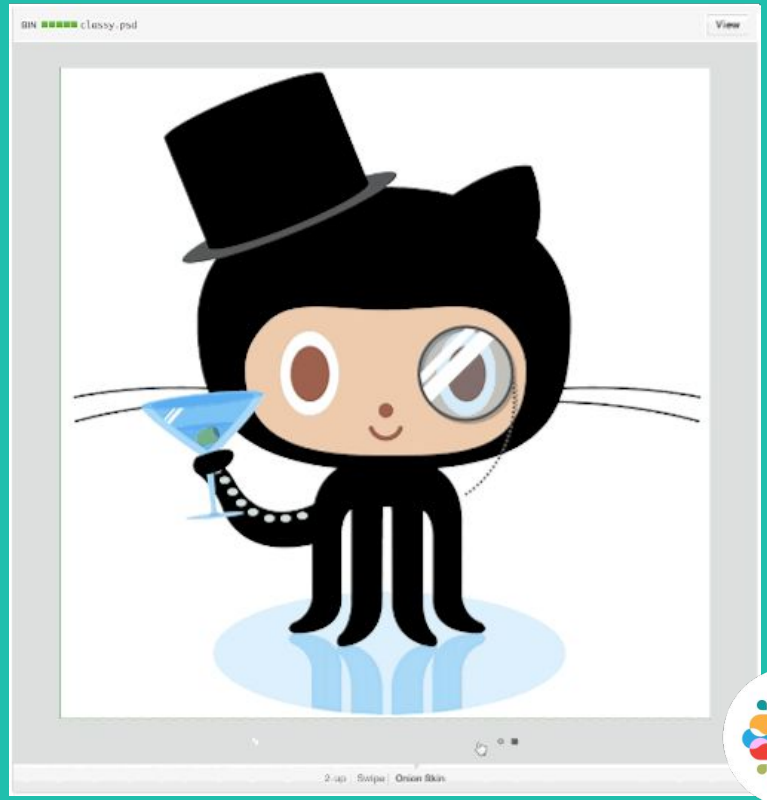
Repository (Repo)

Your project - this is where all your files live. Each file in a Git/GitHub repo is version controlled. With Github your repo can live on your computer and on the web!



Version Control

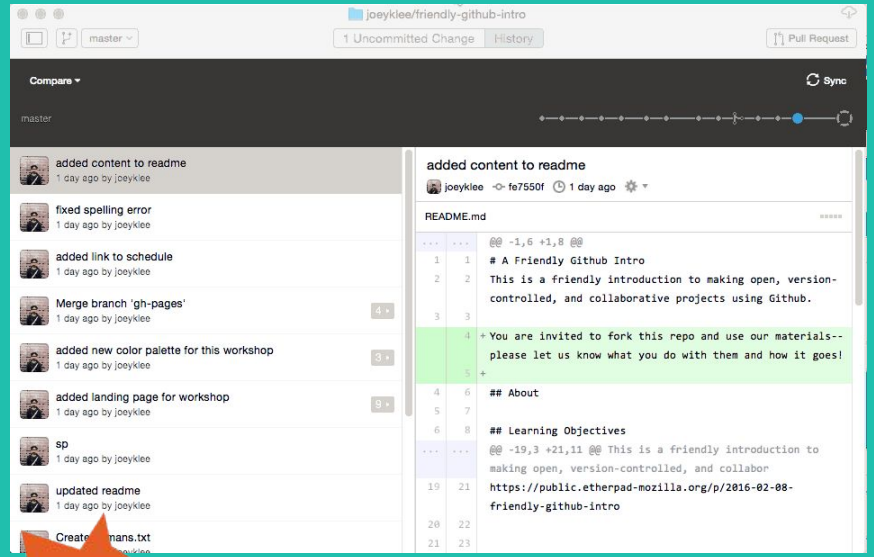
A system for tracking changes (or versions) of a anything (document, file, image, etc). Version control allows you to go back to previous versions of a file or files.



https://github.com/tiimgreen/github-cheat-sheet

Commit

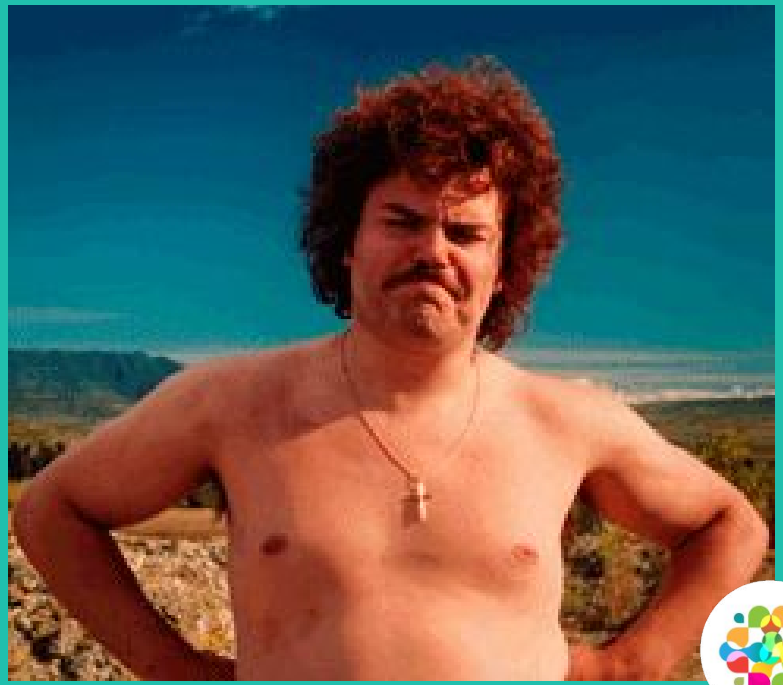
A “save” of a version of a file.
Every time you commit a file or files in Git/GitHub, you are leaving a “breadcrumb” of that file or those files to come back to.



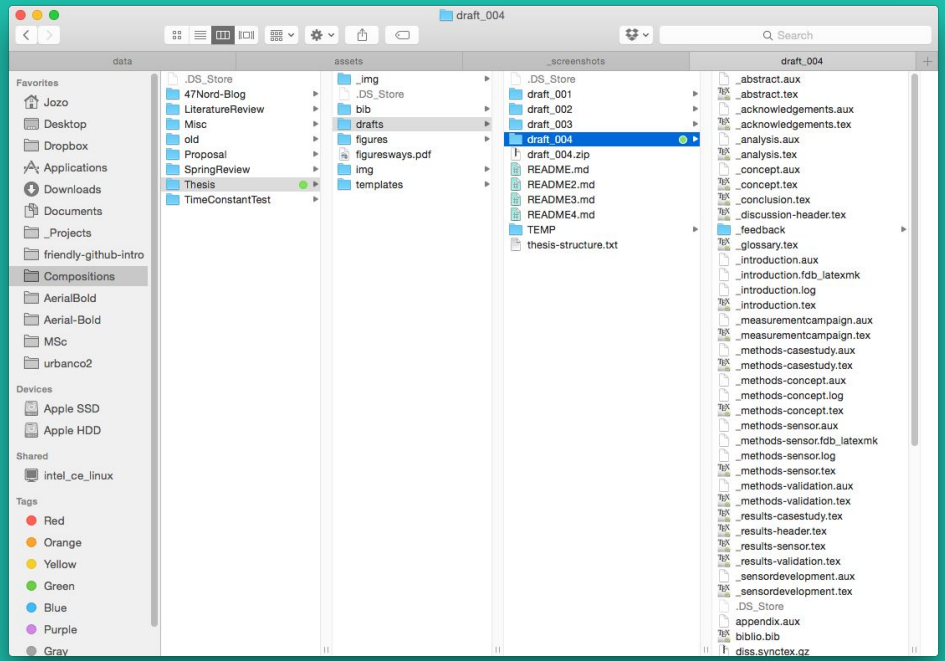
Holy guacamole! You're saying with Github we can:

- sync our projects locally on our computers and online?
- and make "commits" which allow us to track our changes over time?

YES!!!



Scenario 1: You want to try new things without messing up your project.

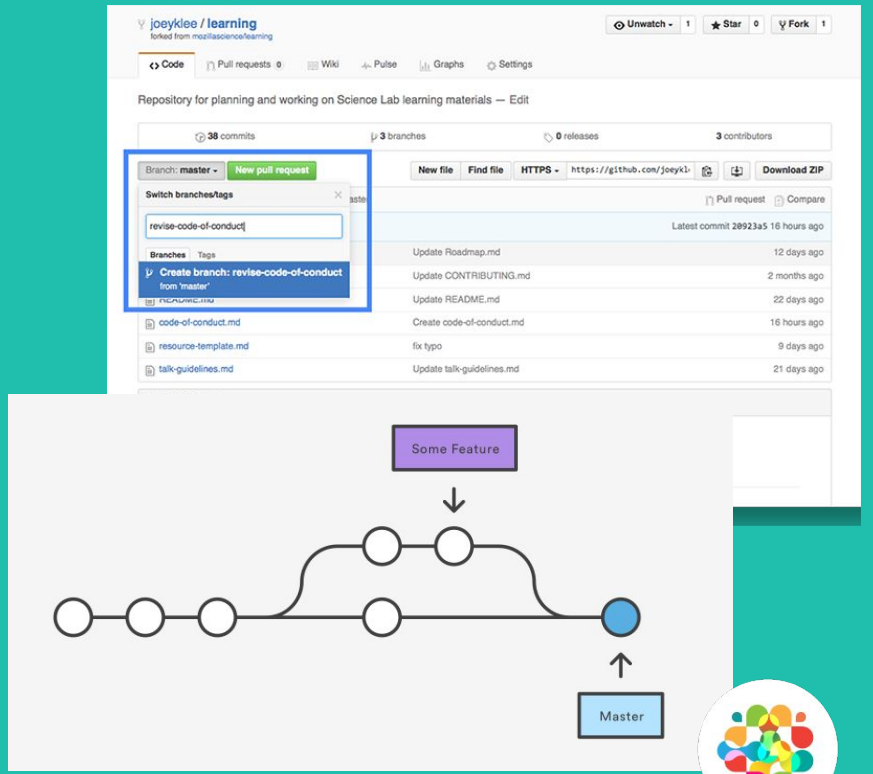


…
thesis-version1.doc
thesis-version1-ACcomments.doc
thesis-version2.doc
thesis-version2-JLComments-ACReview.doc
…

^^ It doesn't have to be like this - Prevent this tragedy!

Branch / Merge

A copy of your project (repo) at a particular point in time that allows you to experiment with new features without affecting the working parts of your project. You can “merge” them into your project later if/when those new features are satisfactory.

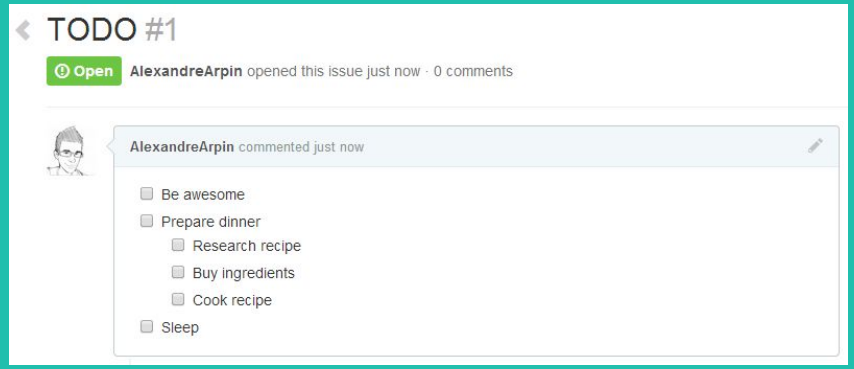


https://www.atlassian.com/git/tutorials/using-branches/git-merge

Scenario 2: There are 20 tasks to do, 3 people working on the project, and no common to-do list / who is doing what.

Issues

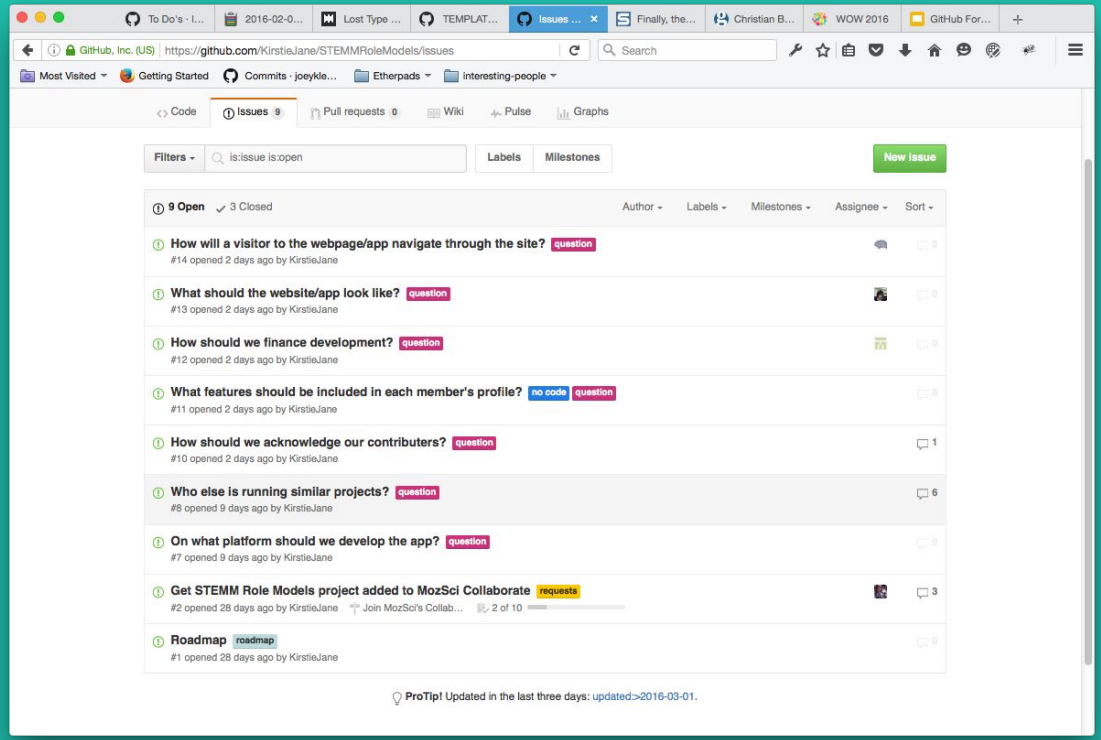
Issues are your “to-do’s” - they are the tasks you need to perform, the bugs you need to fix, and things you want to accomplish.



https://github.com/tiimgreen/github-cheat-sheet

You can “label” issues and “assign” them to collaborators (or yourself) to do.

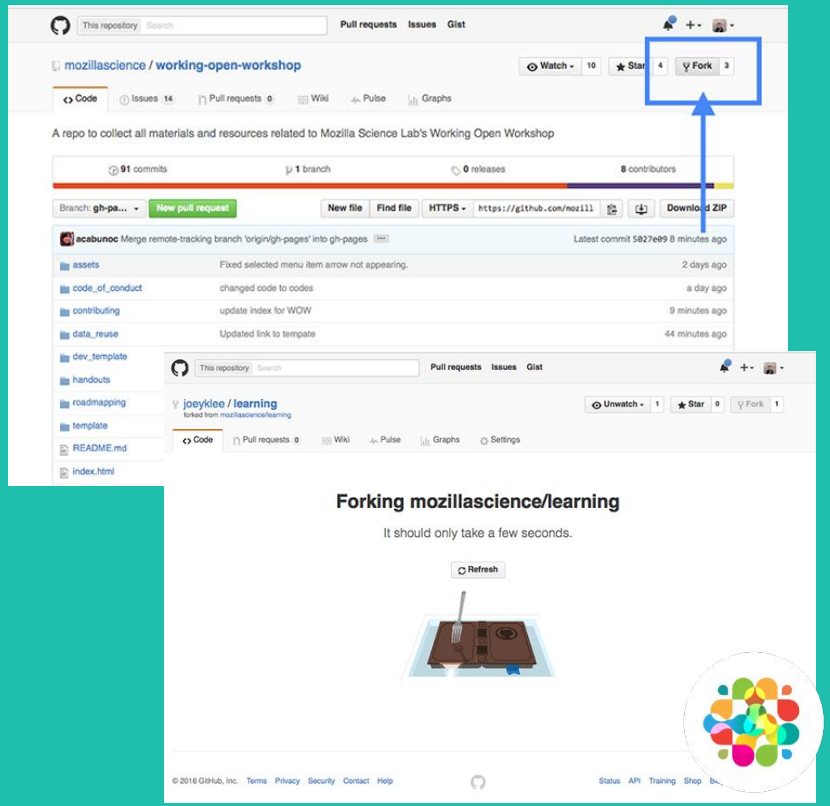
Issue Tracking!



Fork

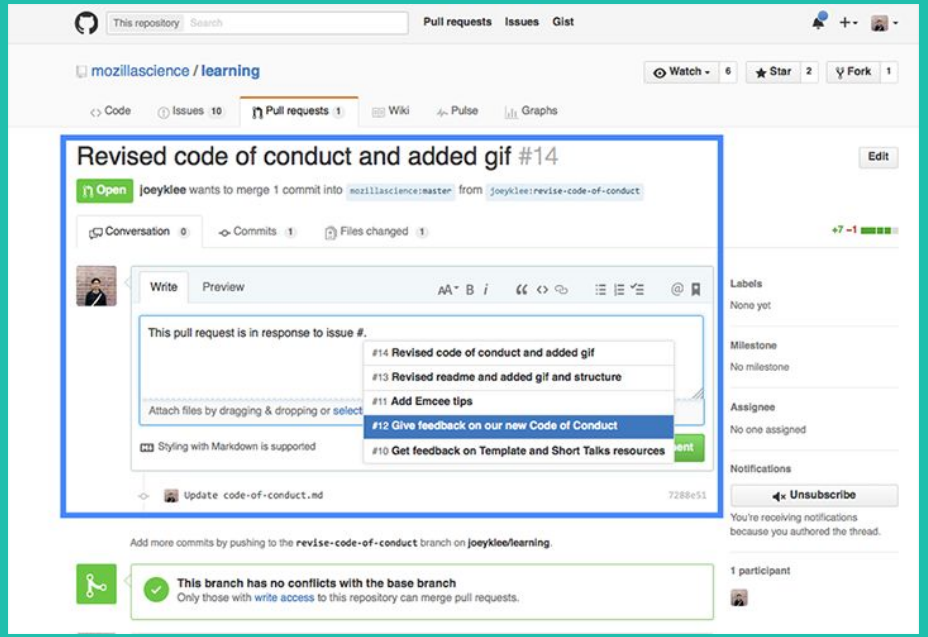
VERB: To fork a repo means you copy someone else's repo with the intention to build on it/improve it and contribute back changes using "pull requests" - you're not an official collaborator on the repo in this case.

NOUN: A copy of someone else's project to which you want to contribute new features. Not to be confused with "clone".

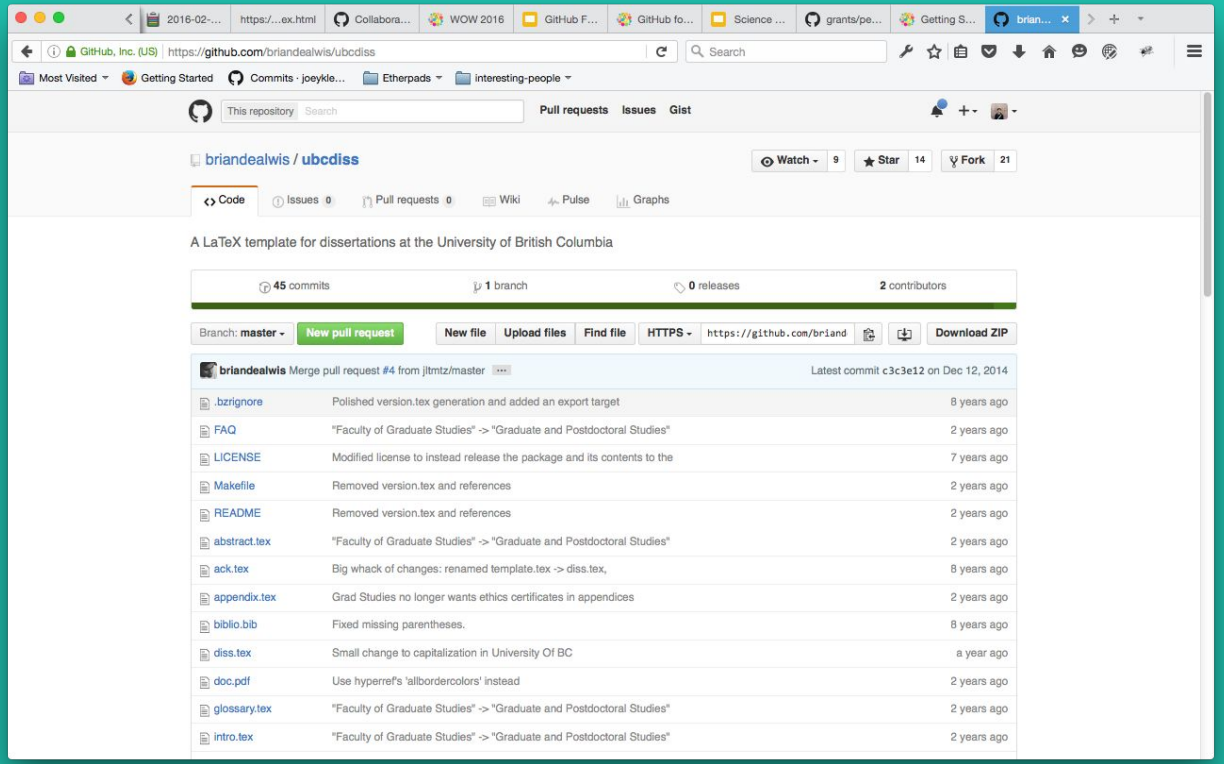


Pull Request

You make a pull request when you've made some awesome changes (commits) to a forked repo or a branch of your own repo and want those changes to be merged into the project either in the "master" branch or branch of interest. It is a "hey, pull these changes into the project".



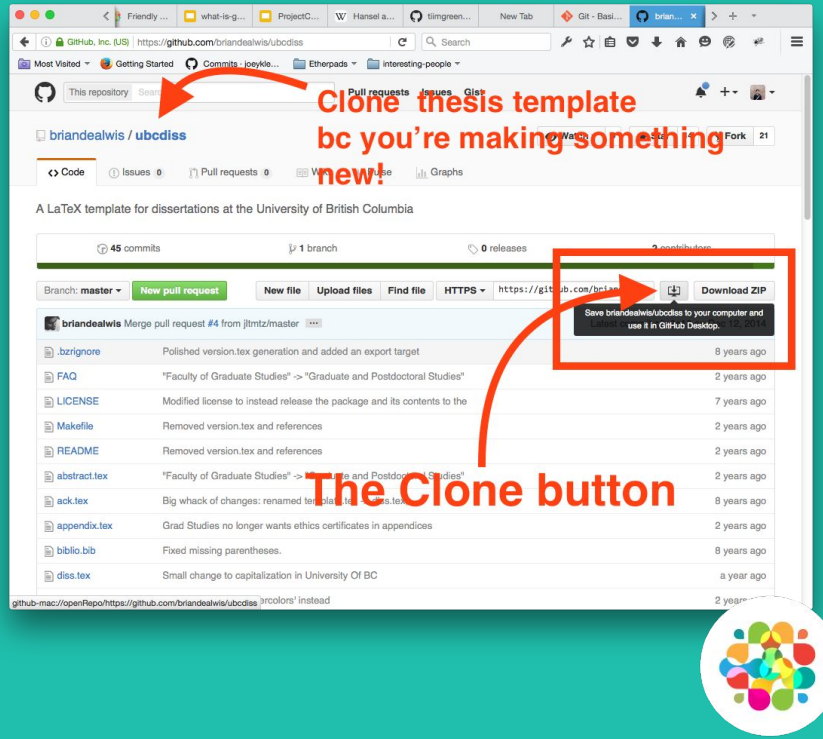
Fork this thesis template & contribute!



Clone

You might “clone” a repo if:
1. you’ve been added as a “collaborator” to a project and want to “clone” or copy the project to your computer

2. you’re “cloning” a project you’ve just forked so that you can pull it from online to your local computer.

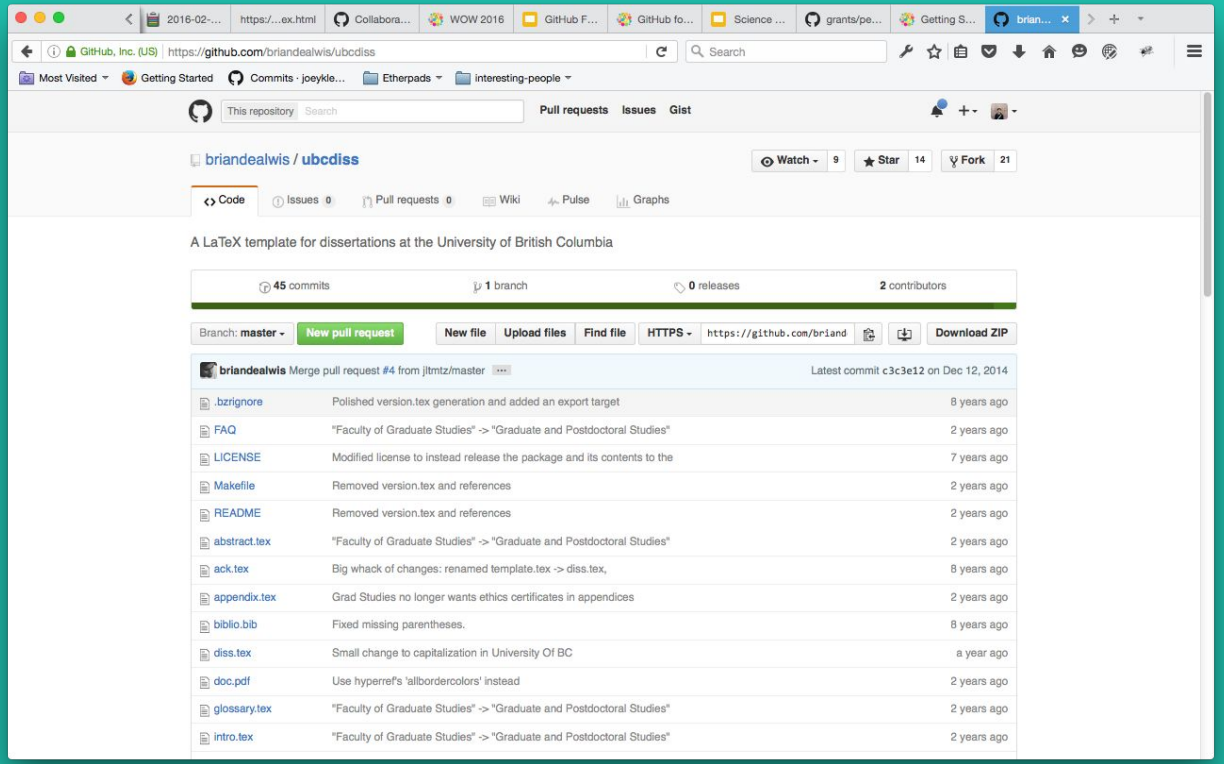


Download Zip

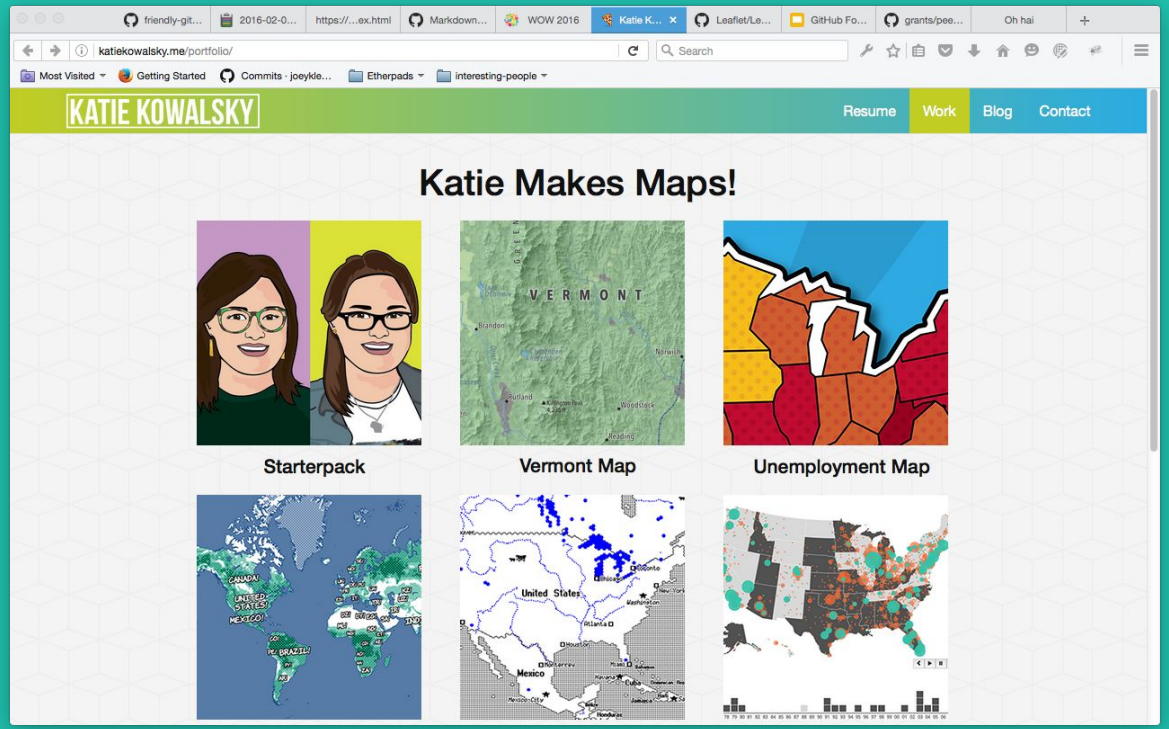
You might “download zip” if:

- You find a cool project online and want to add your own content and remix it into something else.
- it is of course important to attribute where you got your template!
- After downloading, you might upload your new project as a repo to GitHub with a new name/content etc.

Download zip of template and remix!



Download zip of portfolio and remix!



http://katiekowalsky.me/portfolio/

Hold the phone! This means that we can:

- experiment on projects without breaking them. - branching
- make to-do lists and keep track of who is doing them - issues
- we can contribute to projects by forking them, cloning, and making pull requests
- build on existing projects by ourselves or with collaborators by cloning
- download the zip, remixing, and pushing a new repo to GitHub.

Don’t worry if you haven’t gotten all the terms yet - we’ll give you time to play later and clarify any questions.

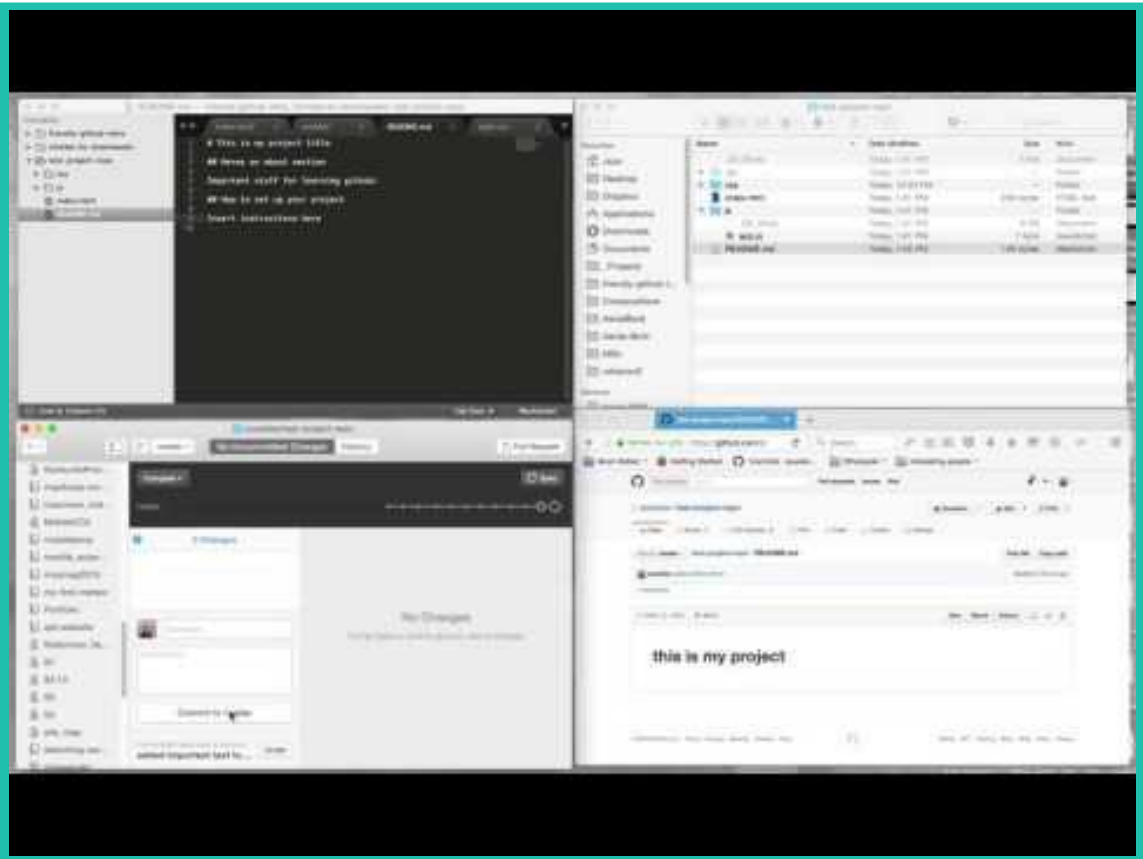
Let’s go through together and set up a project on GitHub to get started!

Thanks!

creds: collab foxes - Z.Marsh @ Mozilla Science Lab.

Scenario 3: You accidentally delete one of your files and need to get it back.

Git/Github Solution: Go back to a commit



friendly-git… | 2016-02-0… | https://…ex.html | Markdown… | WOW 2016 | Katie Kow… | Leaflet/Le… | GitHub Fo… | grants/pee… | Leaflet for … ×

https://rstudio.github.io/leaflet/basemaps.html

Search

Most Visited | Getting Started | Commits · joeykle… | Etherpads | interesting-people

Fork me on GitHub

Leaflet for R

- Introduction
- The Map Widget
- Basemaps
- Markers
- Popups
- Lines and Shapes
- JSON
- Raster Images
- Shiny Integration
- Colors
- Legends
- Show/Hide Layers

Using Basemaps

Leaflet supports basemaps using map tiles, popularized by Google Maps and now used by nearly all interactive web maps.

Default (OpenStreetMap) Tiles

The easiest way to add tiles is by calling `addTiles()` with no arguments; by default, OpenStreetMap tiles are used.

```
m <- leaflet() %>% setView(lng = -71.0589, lat = 42.3601, zoom = 12)
m %>% addTiles()
```

Third-Party Tiles

Alternatively, many popular free third-party basemaps can be added using the

https://rstudio.github.io/leaflet/basemaps.html

There’s a lot of history we’re going to skip over.

Instead we’re going to try to give you an overview of the most important bits.

Let’s go!