

Introduction to Git and GitHub

PPOL 768: Research Design and Implementation

Spring 2023

Georgetown University Initiative

guide^{.2}

on Innovation, Development and Evaluation

What is Git? What is GitHub?

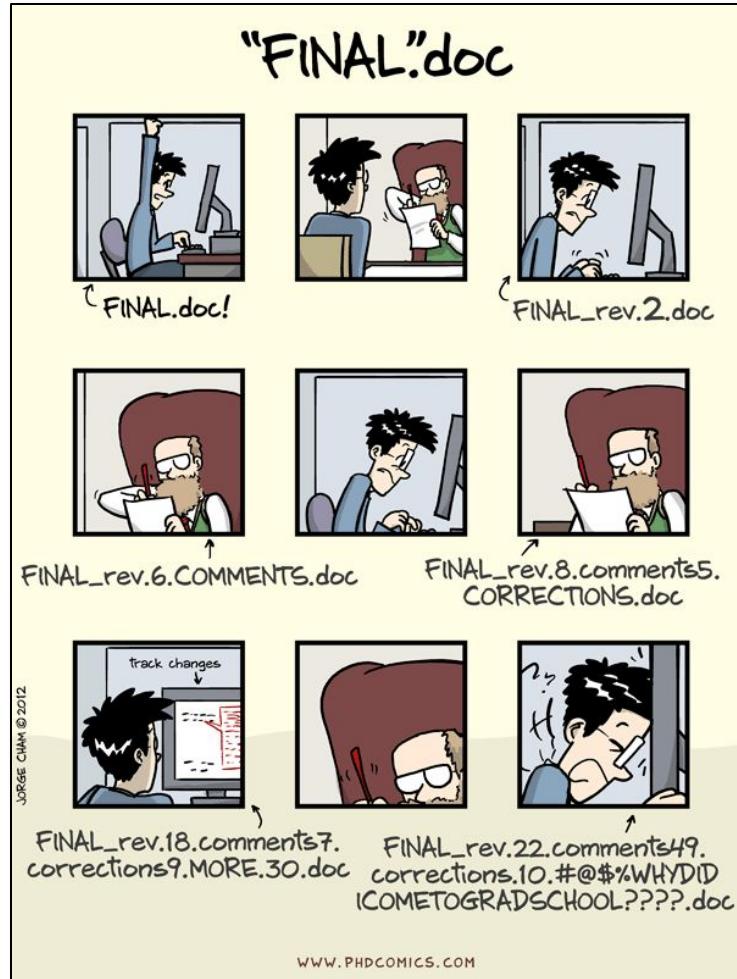
First, Git is not GitHub.

Second, GitHub is not Git.

Git is a **version control software**. It organizes *versions* of every file in a project in an orderly way.

GitHub is a “**development platform**”. It manages *contributions* to projects in an orderly way.

Together, Git and GitHub allow you to avoid the “FINAL”.doc problem.

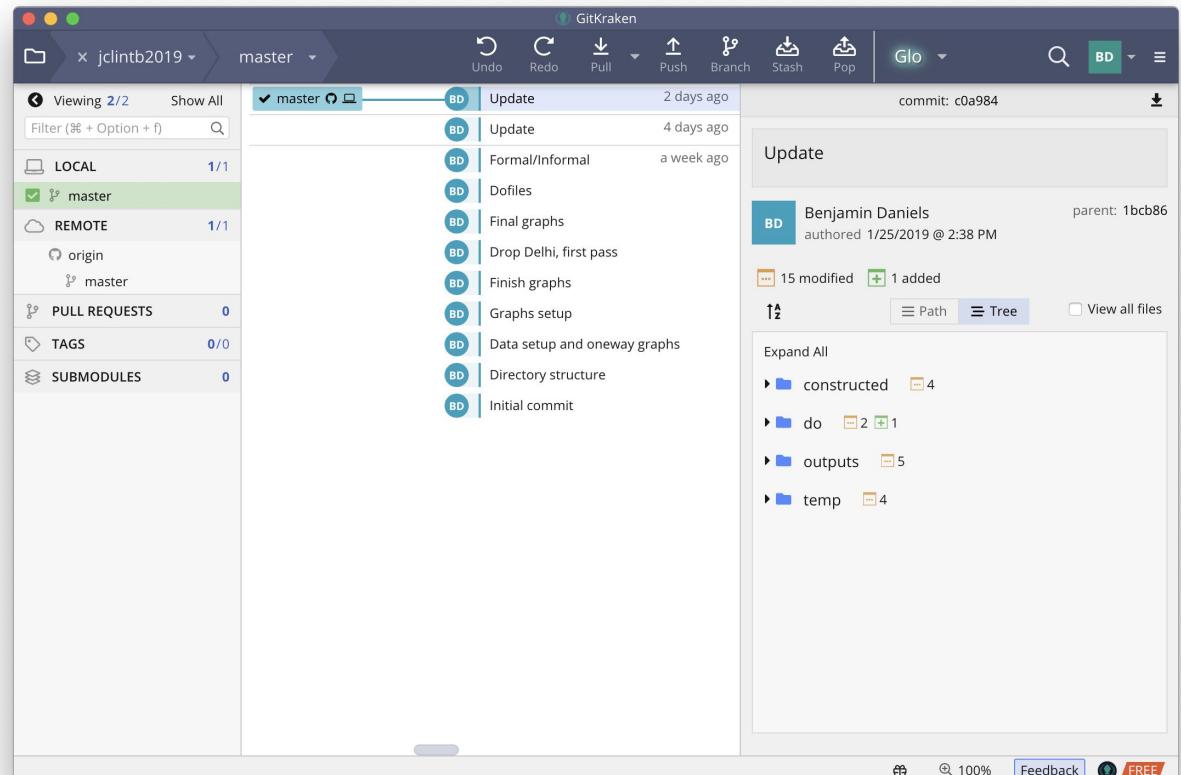


Git: The basics

Git is a software that runs on your computer. It stores “version histories” of your files in a way that is really useful, but that you mainly don’t need to understand.

The complete history of a project is called a “repository”. A repository can be viewed in a desktop “client”, such as [GitKraken](#), shown here.

A very simple repository would look much like this one: it has a series of versions called *commits* that make up its history. Each commit is a record of the changes between itself and the previous commit.

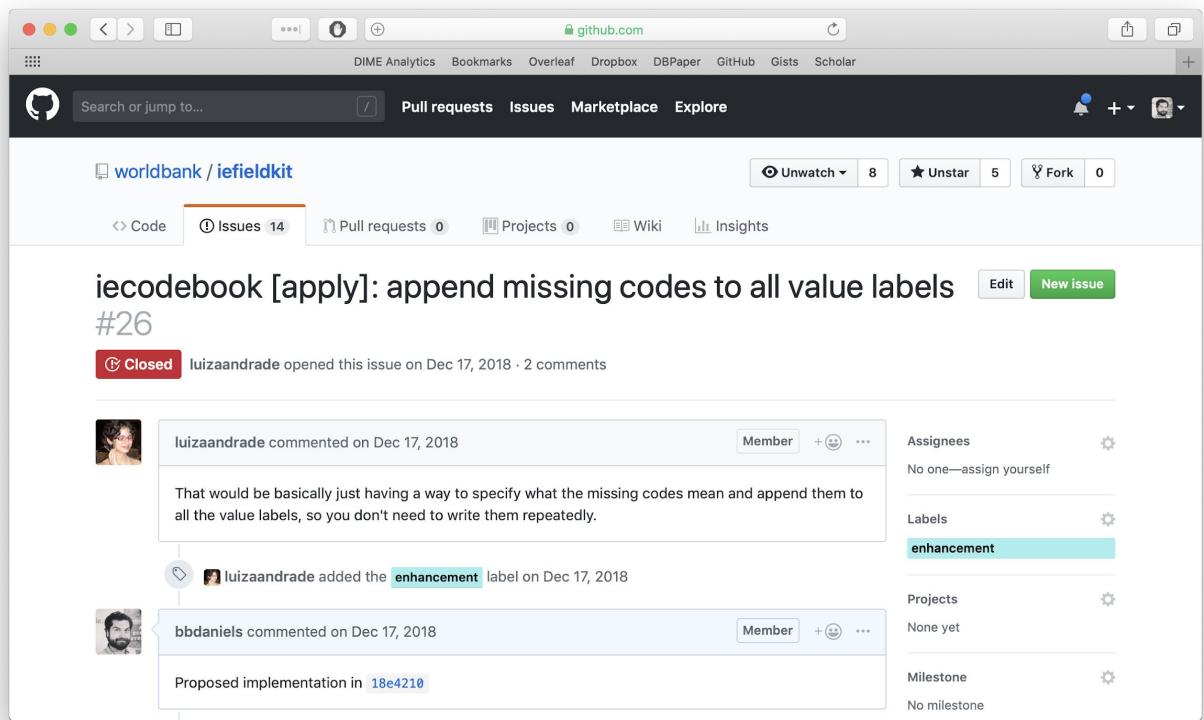


GitHub: The basics

GitHub is a website that you can use to manage collaboration with other people.

This has two main forms: first, managing and tracking tasks (known generally as “issues”), and second, allowing people to submit new materials to the repository.

A very simple issue looks like this one: A problem is posed, and after some discussion a solution is submitted as a “pull request” to the repository.



Git & GitHub: The basics

Mastering Git with GitHub gives you an awesome tool for managing your own team's workflow, as well as for making and releasing big projects.

Having both your data, code, and *code history* available for all authors and the public makes an incredibly useful tool as a product of your work.

Plus, Git protects all your files much better than other software.

(Shown here: The World Bank Atlas of Sustainable Development Goals.)

The screenshot shows a GitHub repository page for 'worldbank / sdgatlas2018'. The repository has 9 commits, 1 branch, 0 releases, and 1 contributor. The latest commit was made by econandrew on May 23, 2018. The commits are listed below:

File	Commit Message
docs	First clean commit
inputs	Final tweak to data license info
.gitignore	First clean commit
INSTALL.md	Update INSTALL.md
README.md	Merge branch 'master' of github.com:worldbank/sdgatlas2018
about_the_atlas.R	First clean commit
make.R	First clean commit
sdg1.R	First clean commit
sdg10.R	Add notes on data terms.
sdg11.R	First clean commit
sdg12.R	First clean commit
sdg13.R	First clean commit
sdg14.R	First clean commit
sdg15.R	Add notes on data terms.
sdg16.R	First clean commit
sdg17.R	First clean commit
sdg2.R	First clean commit
sdg3.R	Add notes on data terms.
sdg4.R	First clean commit
sdg5.R	First clean commit
sdg6.R	First clean commit

The foundation for any evidence is trust: trust that data have been collected, managed, and analyzed responsibly and trust that they have been faithfully presented. The Atlas is the first World Bank publication that sets out to be computationally **reproducible—the majority of its charts and maps are produced with published code, directly from public data sources such as the World Bank's Open Data platform.**

The *Atlas* distills the World Bank's knowledge of data related to the SDGs. I hope it inspires you to explore these issues further so that we can collectively accelerate progress toward achieving the SDGs.

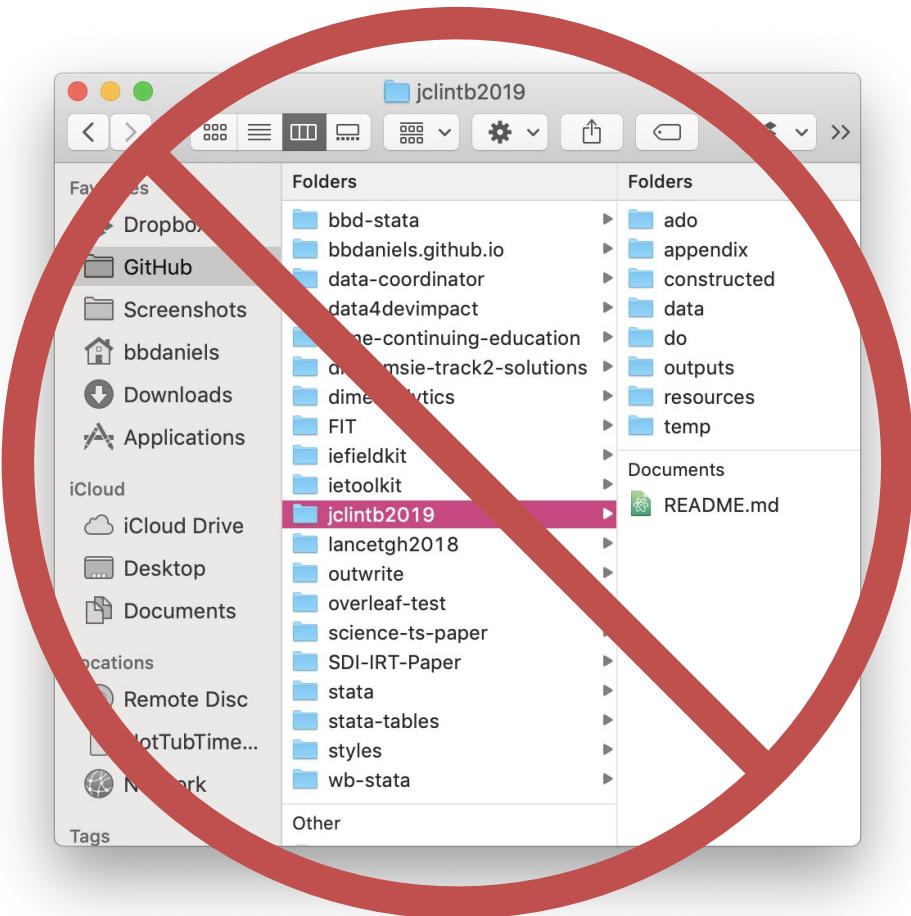
Shanta Devarajan
Senior Director, Development Economics and
Acting Chief Economist
World Bank Group

The files are not the repository

Your Git repositories will probably mainly be used to manage a lot of files. But it is important to remember that although the terms are often used semi-interchangeably, the “repository” is *not* the files themselves.

The files live in what we will call the “working directory” – the project folder (“directory”) where you are working.

A Git repository manages the contents of the working directory.



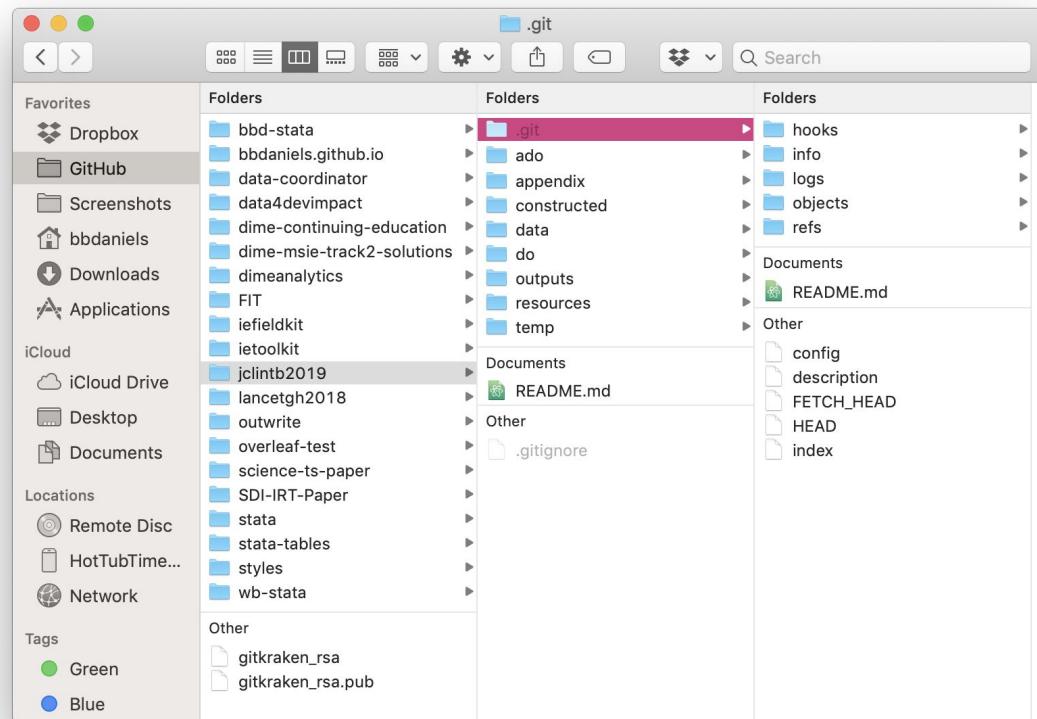
What is the repository then?

You *really* don't need to remember this, but:

The Git repository is a bunch of hidden files *inside* the working directory. This is where the magic happens.

It's hidden because you never need to touch these. NEVER.

Just pretend this doesn't exist – everything important happens entirely inside your Git client.



So how does it *work*?

Part 1: Creating a repository and making commits



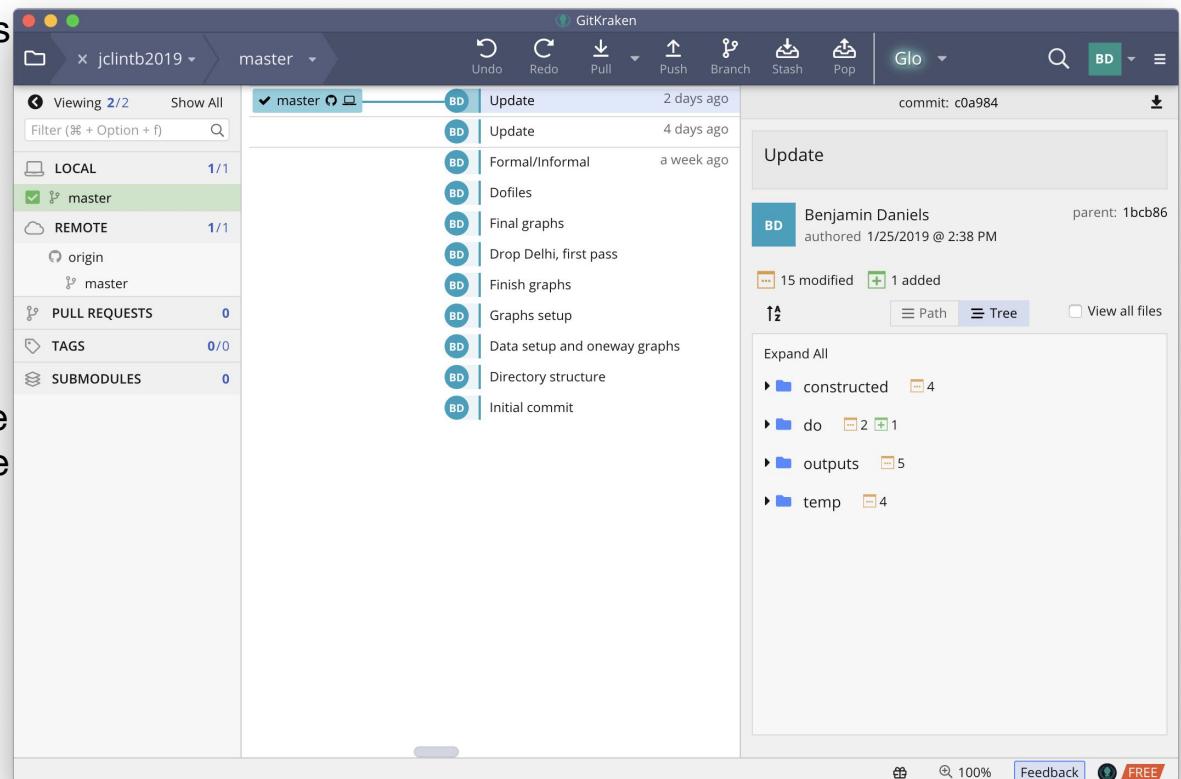
GEORGETOWN UNIVERSITY

Git creates commits

The most important thing Git does is create “commits”.

A commit is a snapshot of the entire working directory. Each commit has a *name*, a *timestamp* and an author.

You have to tell Git when to create new commits. This is probably the biggest difference from whatever you do now.



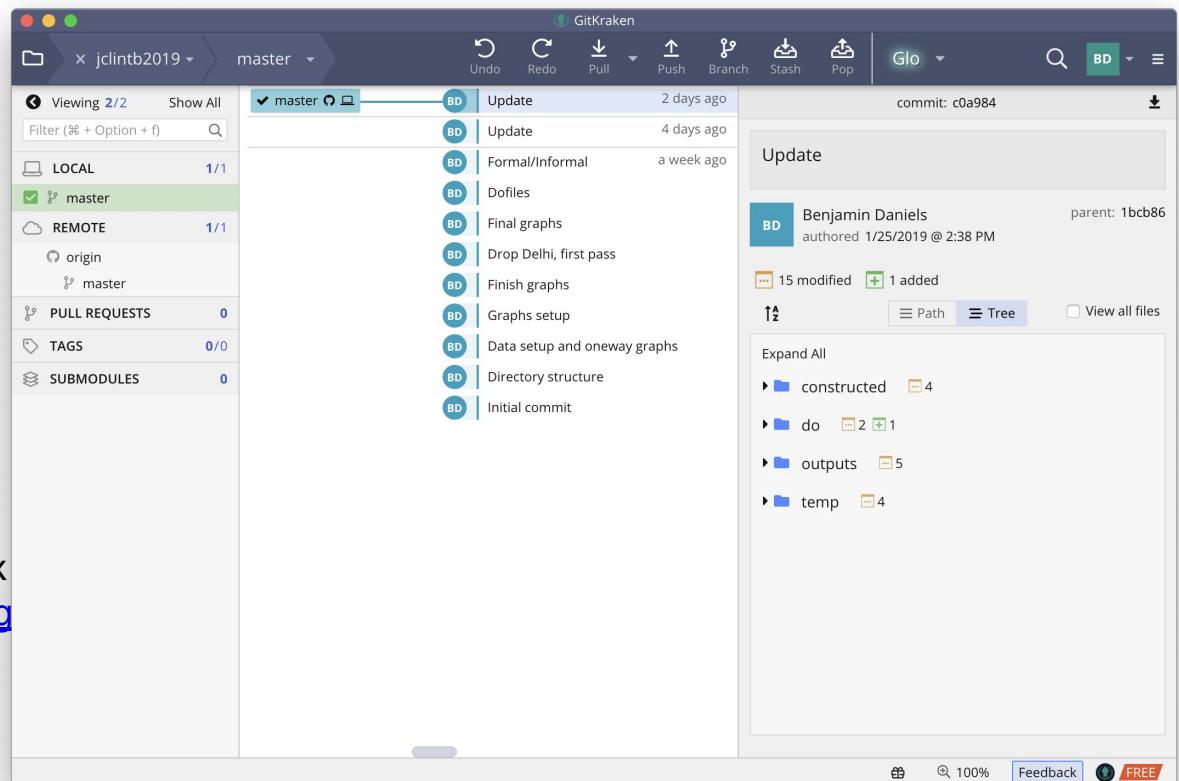
Commits name past versions

Git watches your saved changes

But it only stores them when you tell it to: this is a “commit”

You can name these so that they make sense and are easy to find when you need them in the future (you *always* need them when you least expect, right?)

Required Reading: Git vs Dropbox
<https://michaelstepner.com/blog/git-vs-dropbox>



How is that better than Dropbox?

Dropbox stores a new snapshot *every time you save each file*.

You save your work *often* (right?)

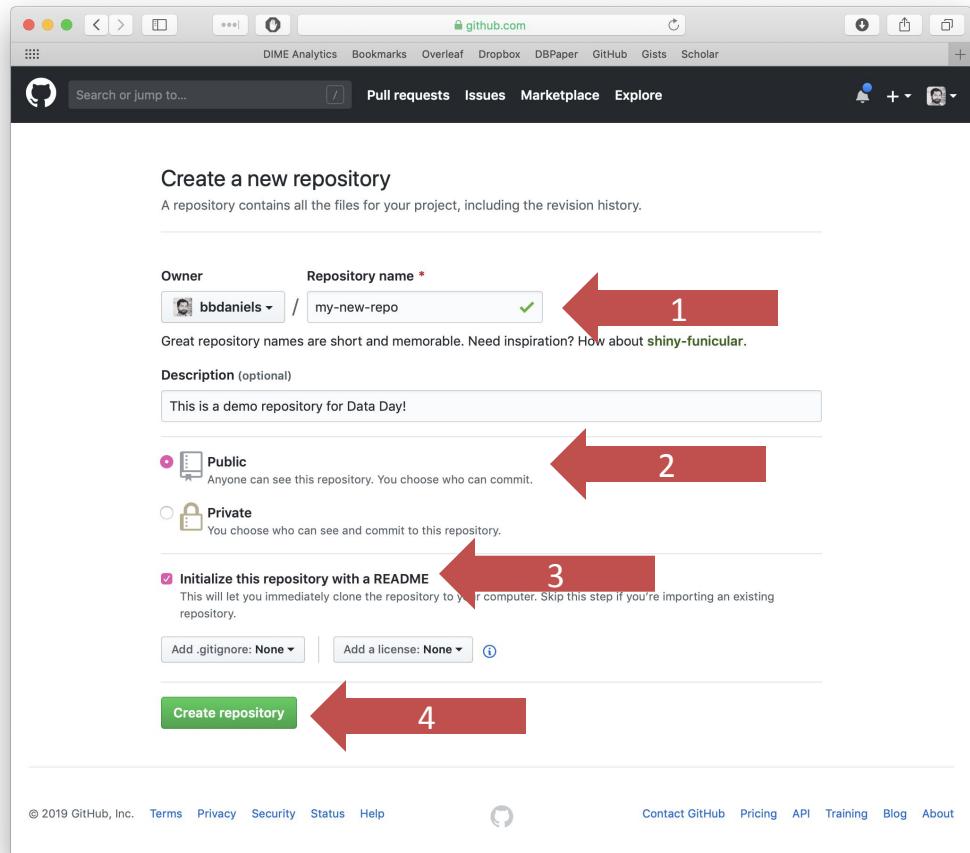
So Dropbox stores *many* similar images of your files

But you can't tell which is which!

The screenshot shows a list of 12 versions of a file named 'analysis_1_m1_summary.do'. The versions are listed from newest at the top to oldest at the bottom. Each entry includes the file name, timestamp, editor (Edited by Benjamin Daniels), device (Desktop), and file size. The newest version is labeled as the 'Current version'.

Timestamp	Editor	Device	File Size
5:30 PM	Edited by Benjamin Daniels	Desktop	10.45 KB
5:29 PM	Edited by Benjamin Daniels	Desktop	10.29 KB
5:29 PM	Edited by Benjamin Daniels	Desktop	10.22 KB
5:28 PM	Edited by Benjamin Daniels	Desktop	10.25 KB
4:13 PM	Edited by Benjamin Daniels	Desktop	10.25 KB
3:07 PM	Edited by Benjamin Daniels	Desktop	10.09 KB
3:02 PM	Edited by Benjamin Daniels	Desktop	10 KB
2:11 PM	Edited by Benjamin Daniels	Desktop	9.02 KB
2:00 PM	Edited by Benjamin Daniels	Desktop	8.96 KB
1:59 PM	Edited by Benjamin Daniels	Desktop	8.97 KB
1:58 PM	Edited by Benjamin Daniels	Desktop	

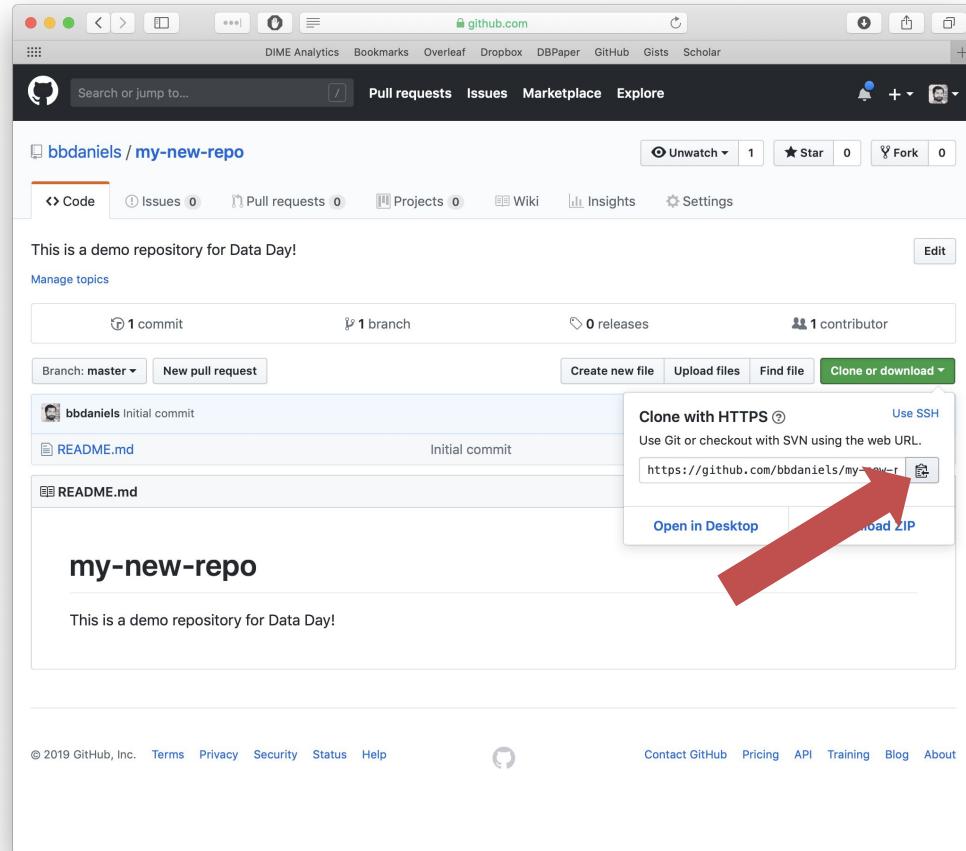
Create a new repository on GitHub



Clone the repo with your desktop client

Cloning the repository makes a complete copy of it at the new location, including the entire history (remember, the repository is the history).

Cloning from GitHub to your local computer is a good way to start a repository.

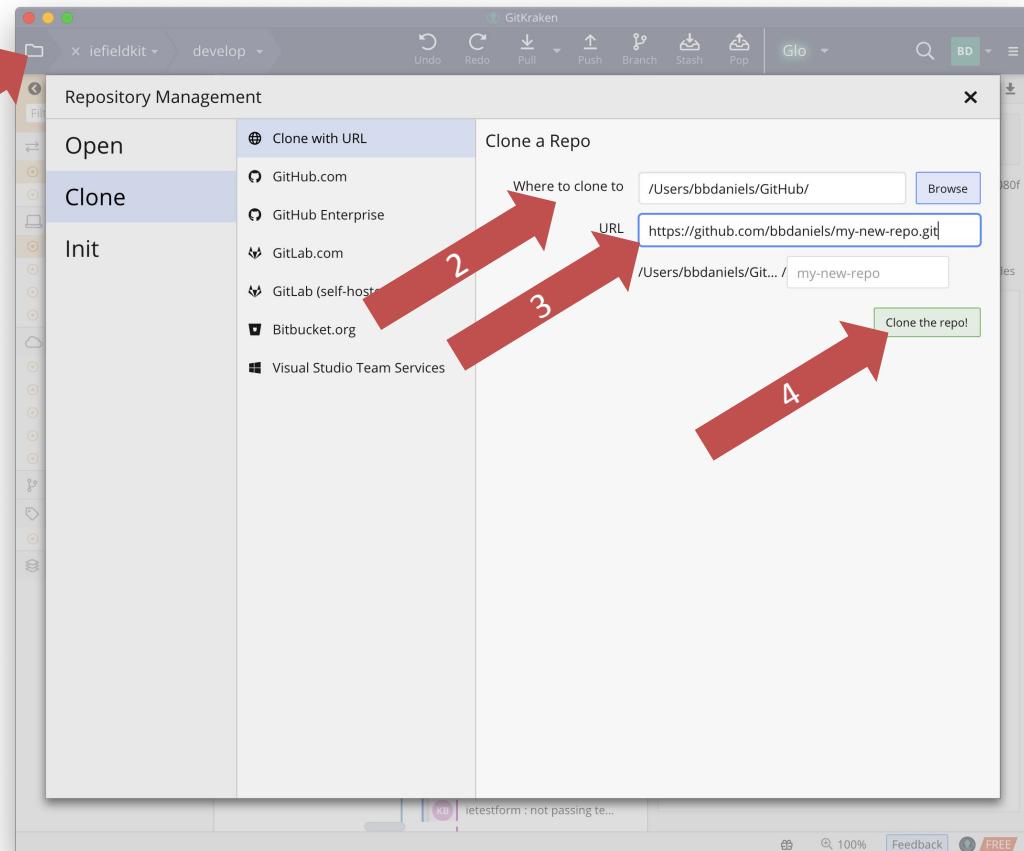


Clone the repo with your desktop client

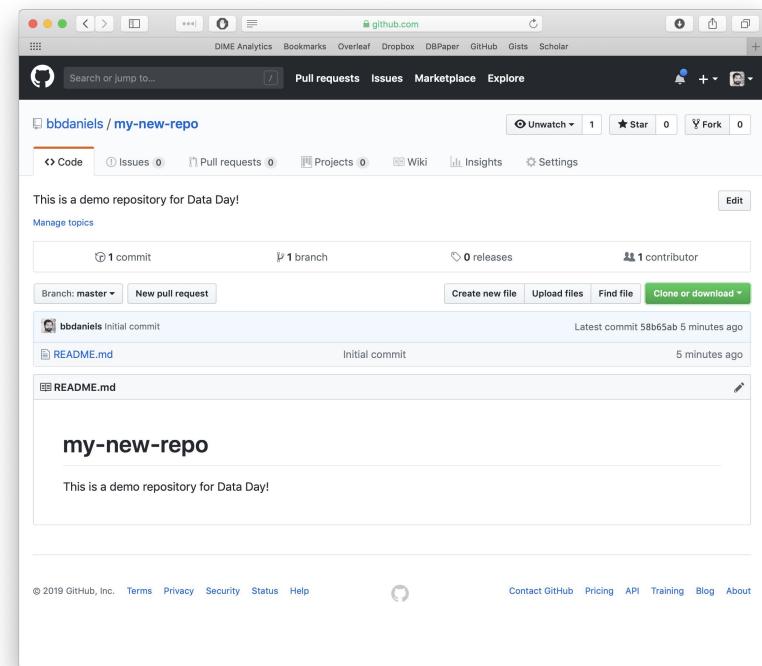
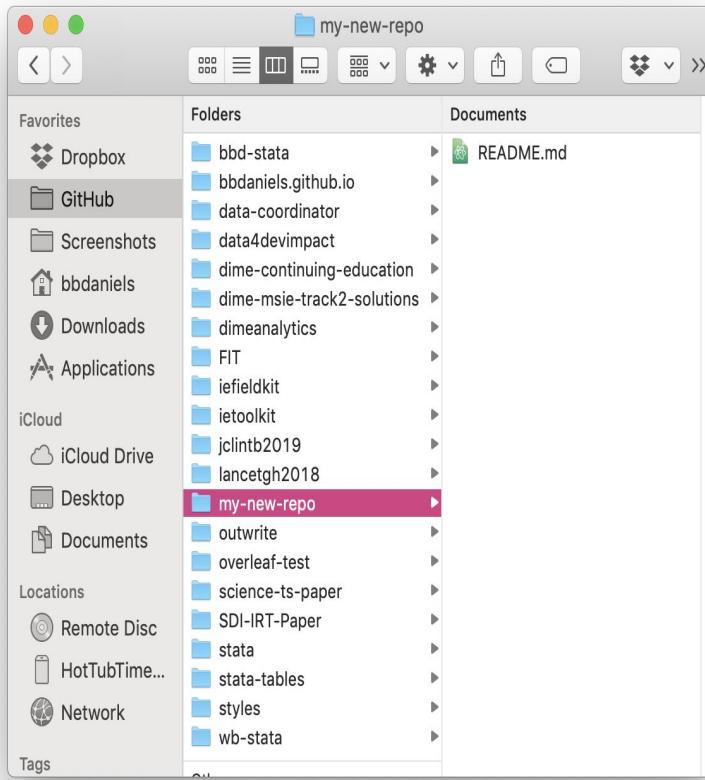
Cloning the repository makes a complete copy of it at the new location, including the entire history (remember, the repository is the history).

Cloning from GitHub to your local computer is a good way to start a repository.

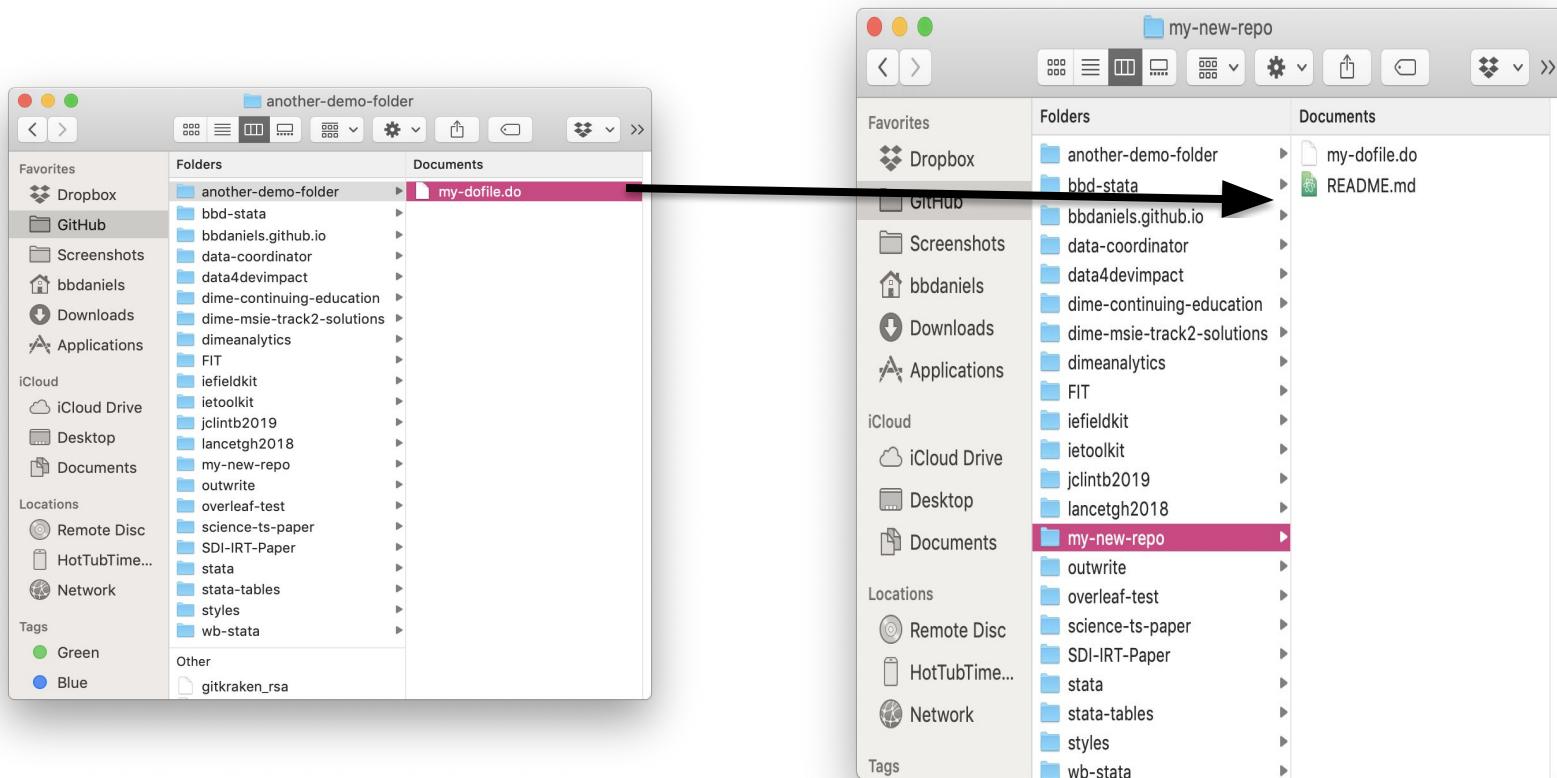
(I know it seems like a lot of steps now, but this becomes second nature. Bear with me, and use this presentation as a guide!)



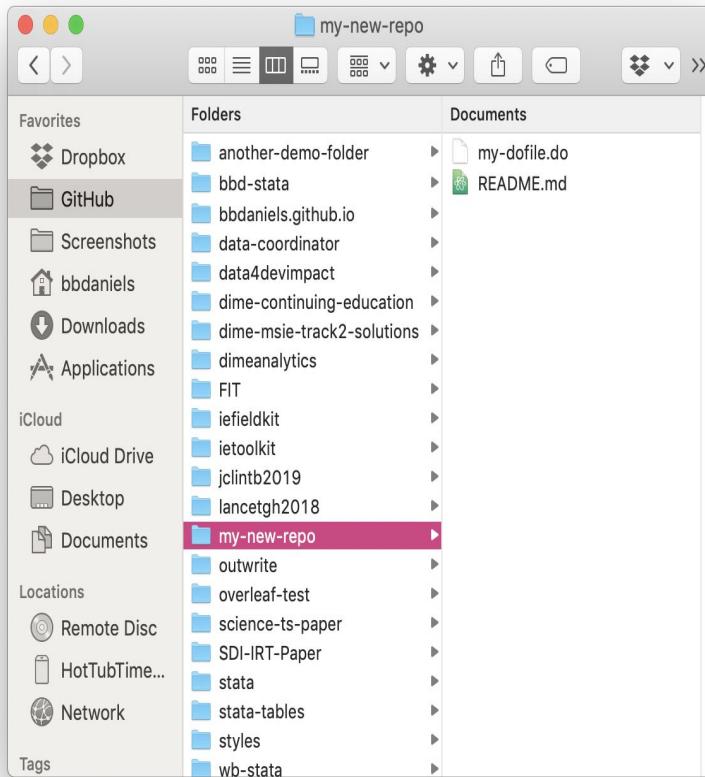
The “local” and “remote” instances are identical



Add some kind of plain-text file locally



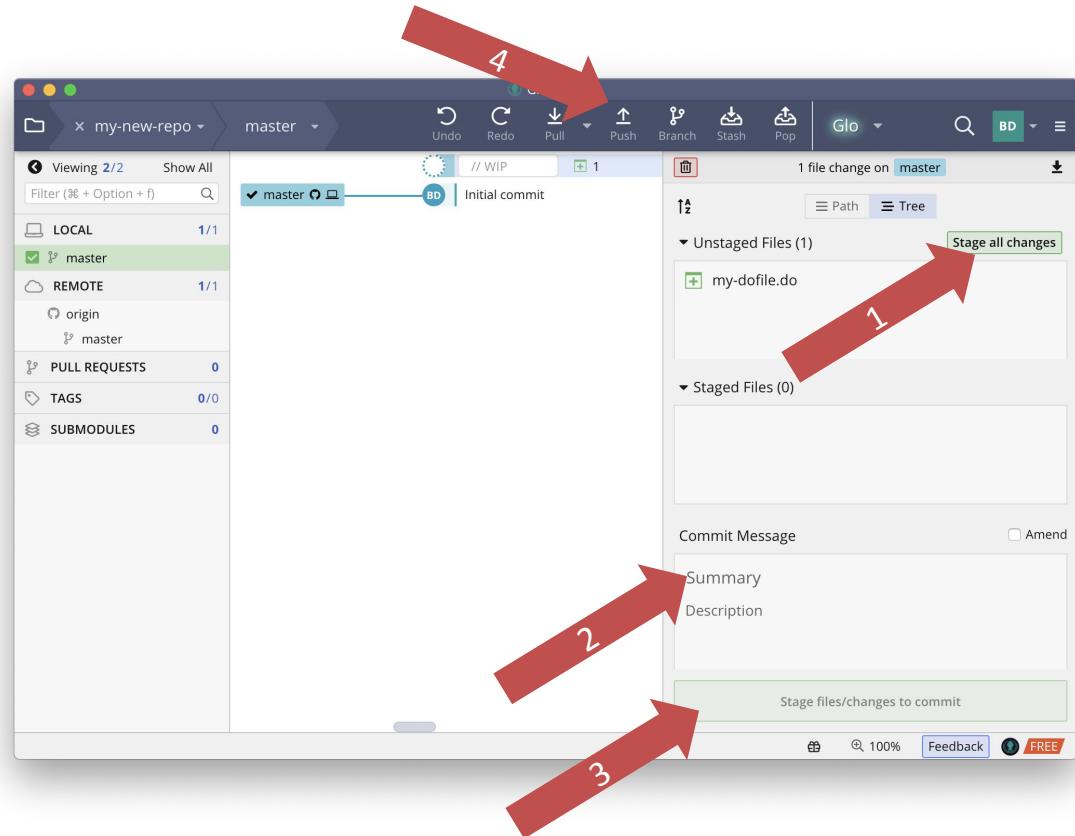
The local and remote are *not* in sync



A screenshot of a GitHub repository page for "my-new-repo" by "bbdaniels". The page header shows "DIME Analytics", "Bookmarks", "Overleaf", "Dropbox", "DBPaper", "GitHub", "Gists", and "Scholar". The repository summary shows 1 commit, 1 branch, 0 releases, and 1 contributor. It lists "bbdaniels Initial commit" and "README.md" (Initial commit, 5 minutes ago). The repository description is "This is a demo repository for Data Day!". The footer includes links for "Contact GitHub", "Pricing", "API", "Training", "Blog", and "About".

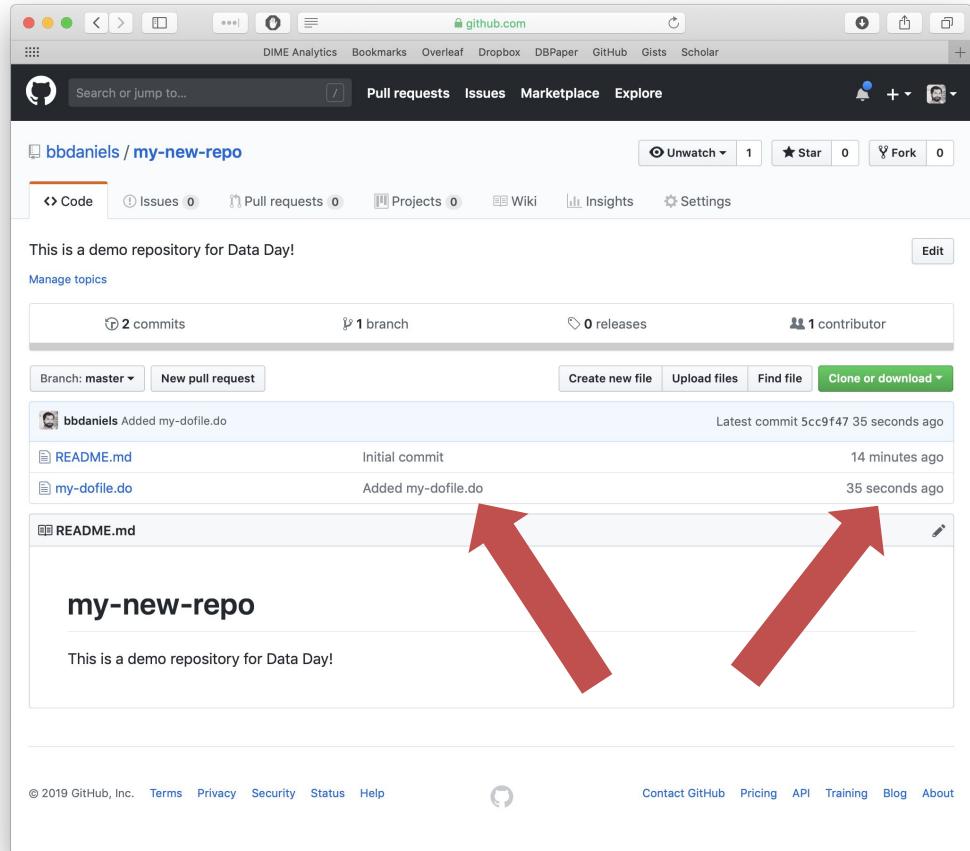
Git notes changes as “work in progress”

1. “Stage” your changes to add them to the queue to be committed.
2. “Name” your changes informatively – a short sentence will do nicely.
3. “Commit” your changes to add them to the version history.
4. “Push” your changes to sync the remote (GitHub) repository with your local copy.



The local and remote are in sync

Each item records when it was last modified – using the corresponding name and timestamp from the commit that modified it.



Georgetown University Initiative
guide
on Innovation, Development and Evaluation

www.gui2de.org