

COGS138: Neural Data Science

Lecture 7

C. Alex Simpkins, PhD

UCSD Dept. of Cognitive Science, Spring 2023

RDPRobotics, LLC

http://casimpkinsjr.radiantdolphinpress.com/pages/cogs138_sp23

rdprobotics@gmail.com | csimpkinsjr@ucsd.edu

Plan for today

- Announcements
- Assignment 1 overview
- Review - Last time
- Data

Announcements

- Final reminder to check on your FinAID status
- A1 - due **a week from release**, which will be tonight or tomorrow
- Reading 1 - Released on canvas and in web site password protected area soon, lecture quiz due **a week from release**, released tonight
- **Group formation** - time to start choosing who you want to work with for your project group

Last time

Course links

Website	http://casimpkinsjr.radiantdolphinpress.com/pages/cogs138_sp23	Main face of the course and everything will be linked from here. Lectures, Readings, Handouts, Files, links
GitHub	https://github.com/drsimpkins-teaching	files/data, additional materials & final projects
datahub	https://datahub.ucsd.edu	assignment submission
Piazza	https://piazza.com/ucsd/spring2023/cogs138_sp23_a00/home (course code on canvas home page)	questions, discussion, and regrade requests
Canvas	https://canvas.ucsd.edu/courses/44897	grades, lecture videos
Anonymous Feedback	Will be able to submit via google form	If I ever offend you, use an example you are uncomfortable with, or to provide general feedback. Please remain constructive and polite

That's a lot of data!

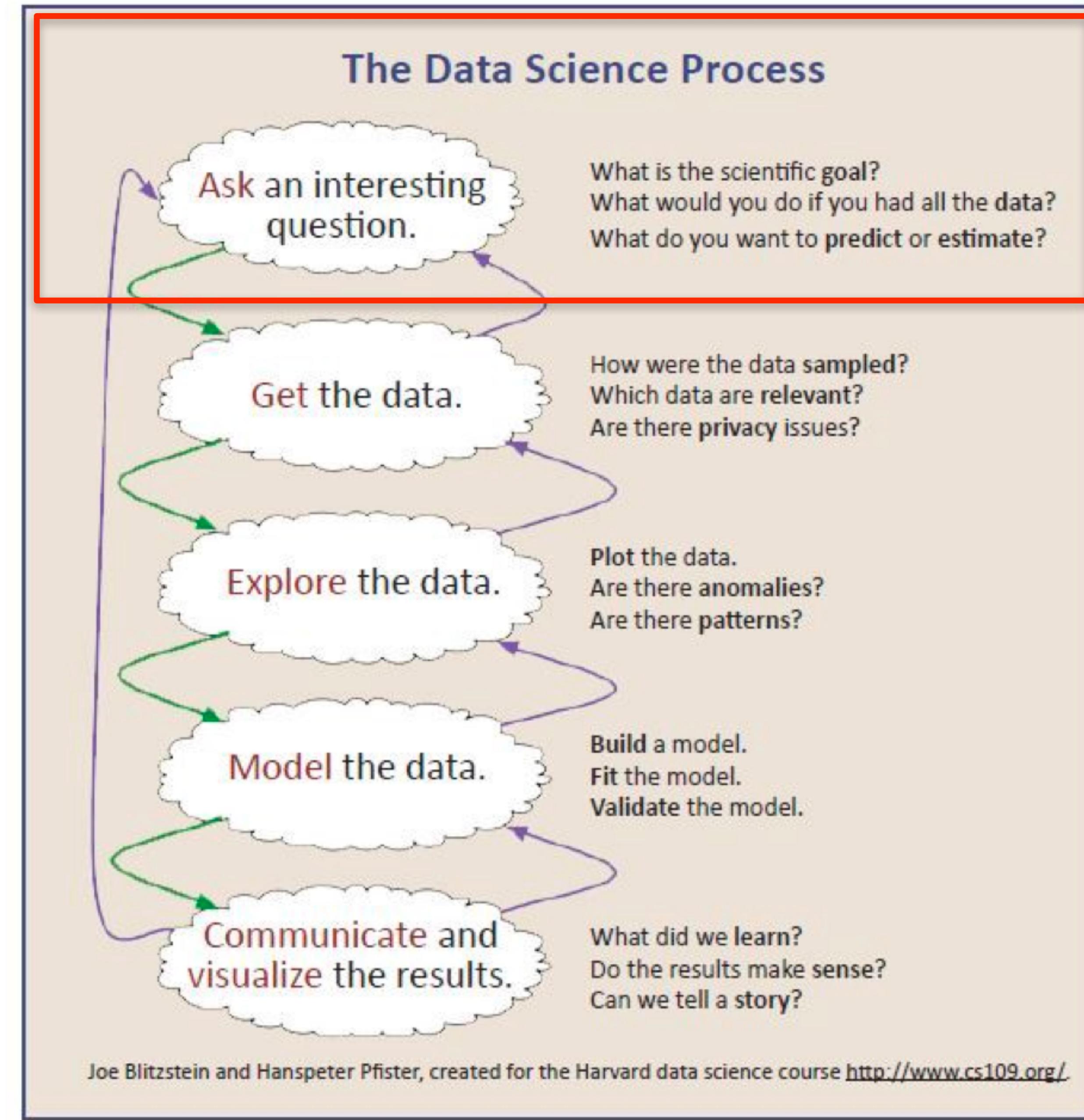
- How do you deal with it all, standardize, organize, communicate it?
- How can you talk across disciplines?
- How do you collaborate and work in teams with this?
- How can you ask questions with all that data and the results generated?

Data science questions, hypothesis generation
(automated), Genes/gene expression, animal
models, FAIR, Neurodata Without Borders (NWB),
Brain Imaging Data Structure (BIDS), DANDI

Formulating Data Science Questions

When you and your group sit down to figure out what you're going to do for your final project in this class, you'll have to formulate a strong question. It should be:

1. **Specific,**
2. Can be answered with **data,**
3. And makes **clear what** exactly **is** being **measured.**



adapted from Chris Keown

Neurodata Without Borders

(N.W.B.)

Introduction, tools, definitions and relevance

Use NWB for

- Use this for cellular neurophysiology, such as electrophysiology and optical physiology

NWB Definition

- <https://www.nwb.org/>
- “**Neurodata Without Borders (NWB)** is a ***data standard*** for neurophysiology, providing neuroscientists with a common standard to share, archive, use, and build analysis tools for neurophysiology data. NWB is designed to store a variety of neurophysiology data, including data from intracellular and extracellular electrophysiology experiments, data from optical physiology experiments, and tracking and stimulus data.” [www.nwb.org]

NWB Introduction

- <https://www.nwb.org/>
- <https://nwb-overview.readthedocs.io/en/latest/>
- So essentially
 - A data format for sharing/archiving
 - Standardized (set of rules and best practices)
 - Packages Data and Metadata together so human- and machine-readable

NWB Introduction

- Take advantage of established techniques for processing, analysis, visualization tools
- Makes data easier to reuse - additional scientific insights
- Essential step to getting data into the DANDI archive (<https://dandiarchive.org/>)

Brain Imaging Data Structure

(B.I.D.S.)

Introduction, tools, definitions and relevance

Use **BIDS** for

- Use for neuroimaging data such as MRI

Brain Imaging Data Structure

- <https://bids.neuroimaging.io/>
- A second data standard

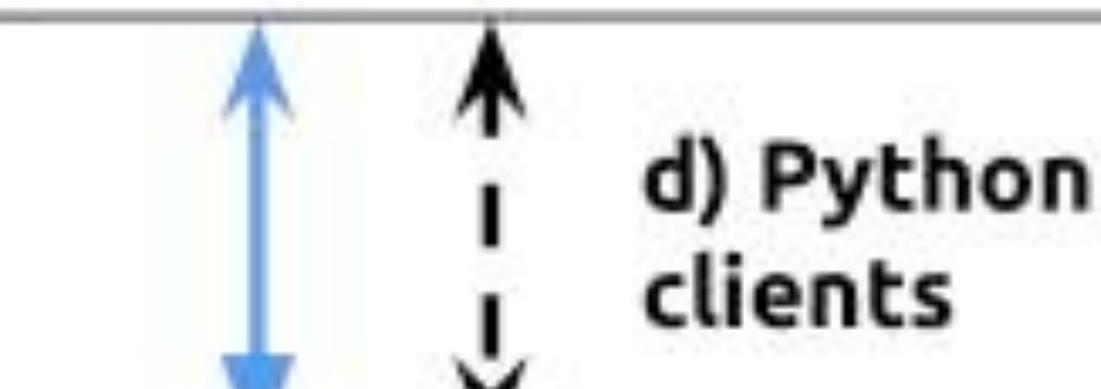
**Distributed Archives for
Neurophysiology Data Integration
(D.A.N.D.)**

What is DANDI?

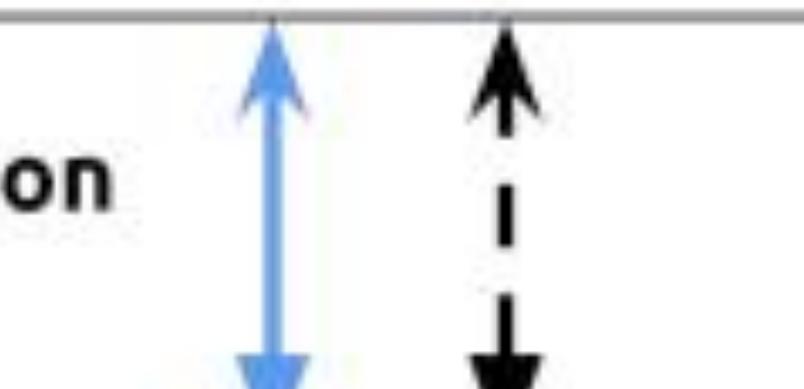
- The BRAIN Initiative archive for publishing and sharing neurophysiology data including
 - Electrophysiology, Optophysiology, Behavioral time-series, Images from immunostaining experiments.
- A persistent, versioned, and growing collection of standardized datasets
- A place to house data to collaborate across research sites
- Supported by the BRAIN Initiative and the AWS Public dataset programs

a) Web application

The screenshot shows the homepage of The DANDI Archive. At the top, there is a navigation bar with links: DANDI (with a brain icon), WELCOME, PUBLIC DANDISETS, MY DANDISETS, ABOUT, DOCUMENTATION, HELP, NEW DANDISET, and a user profile icon. Below the navigation bar, the title "The DANDI Archive" is displayed in large blue text, followed by a subtitle: "The BRAIN Initiative archive for publishing and sharing neurophysiology data including electrophysiology, optophysiology, and behavioral time-series, and images from immunostaining experiments." A search bar is present with the placeholder "Search Dandisets by name, description, identifier or contributor name". Below the search bar, there are three dark grey boxes showing statistics: "138 datasets", "311 users", and "157 TB total data size".



Collaborator(s)



Lab Member(s)

b) Supported standards



c) Analysis platform



Benefits of DANDI

- A FAIR (Findable, Accessible, Interoperable, Reusable) data archive to house standardized neurophysiology and associated data
- Rich metadata to support search across data
- Consistent and transparent data standards to simplify data reuse and software development.
 - Uses NWB, BIDS, Neuroimaging Data Model (NIDM), and other BRAIN Initiative standards to organize and search the data.
 - The data can be accessed programmatically allowing for software to work directly with data in the cloud
- The infrastructure is built on a software stack of open source products, thus enriching the ecosystem

DANDI compatibility

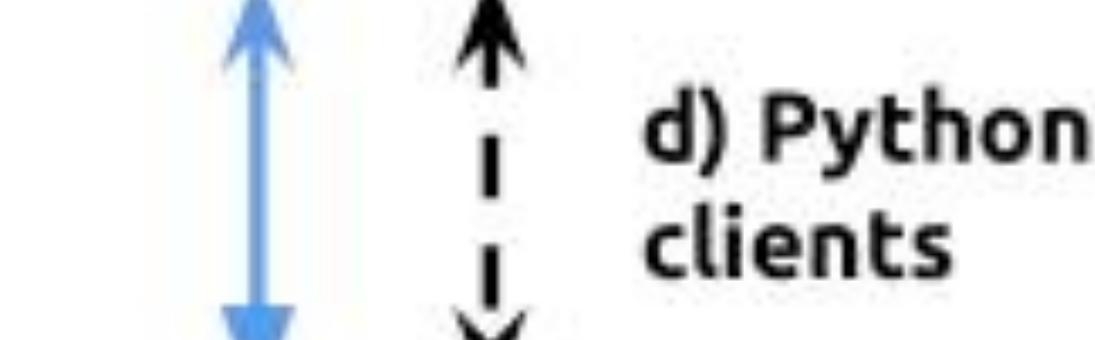
- Uses NWB for core data language
- “Dandisets” - DANDI datasets - collection of NWB files recorded over multiple sessions, organized together
- Viewable from a web browser
- Can interact through Jupyterhub interface for exploring, visualizing and analyzing the data stored in the archive

DANDI python client

- Organize data locally into the required structure
- Download/upload data from/to the DANDI archive

a) Web application

The screenshot shows the homepage of The DANDI Archive. At the top, there is a navigation bar with links: DANDI (with a brain icon), WELCOME, PUBLIC DANDISETS, MY DANDISETS, ABOUT, DOCUMENTATION, HELP, NEW DANDISET, and a user profile icon. Below the navigation bar, the title "The DANDI Archive" is displayed in large blue text, followed by a subtitle: "The BRAIN Initiative archive for publishing and sharing neurophysiology data including electrophysiology, optophysiology, and behavioral time-series, and images from immunostaining experiments." A search bar with the placeholder "Search Dandisets by name, description, identifier or contributor name" is present. Below the search bar, three dark grey boxes show statistics: 138 datasets, 311 users, and 157 TB total data size. A large, semi-transparent brain image serves as the background for the page.



Collaborator(s)

Lab Member(s)

b) Supported standards



c) Analysis platform



Web Browser, Shell, API

JSON/JSON-LD, NWB, NIfTI, TIFF

DANDI archive

- **Public DANDI sets:** <https://dandiarchive.org/dandiset>
- **Documentation:** https://www.dandiarchive.org/handbook/10_using_dandi/

DANDI Properties

- **Data identifiers:** The archive provides persistent identifiers for versioned datasets and assets, thus improving reproducibility of neurophysiology research
- **Data storage:** Cloud-based platform on AWS. Data are available from a public S3 bucket. Data from embargoed datasets are available from a private bucket to owners only
- **Type of data:** The archive accepts cellular neurophysiology data including electrophysiology, optophysiology, and behavioral time-series, and images from immunostaining experiments and other associated data (e.g. participant information, MRI or other modalities)
- **Accepted Standards** and Data File Formats: NWB (HDF5), BIDS (NIfTI, JSON, PNG, TIF, OME.TIF, OME.BTF, OME.ZARR) (see Data Standards for more details)

Neurophysiology Informatics Challenges and DANDI Solutions

Challenges	Solutions
Most raw data stays in laboratories.	DANDI provides a public archive for dissemination of raw and derived data.
Non-standardized datasets lead to significant resource needs to understand and adapt code to these datasets.	DANDI standardizes all data using NWB and BIDS standards.
The multitude of different hardware platforms and custom binary formats requires significant effort to consolidate into reusable datasets.	The DANDI ecosystem provides tools for converting data from different instruments into NWB and BIDS.
There are many domain general places to house data (e.g. Open Science Framework, G-Node, Dropbox, Google drive), but it is difficult to find relevant scientific metadata.	DANDI is focused on neurophysiology data and related metadata.
Datasets are growing larger, requiring compute services to be closer to data.	DANDI provides Dandihub, a JupyterHub instance close to the data.
Neurotechnology is evolving and requires changes to metadata and data storage.	DANDI works with community members to improve data standards and formats.
Consolidating and creating robust algorithms (e.g. spike sorting) requires varied data sources.	DANDI provides access to many different datasets.

DANDI archive

- <https://elifesciences.org/articles/78362>
- **Oliver Rübel, Andrew Tritt, Ryan Ly, Benjamin K Dichter, Satrajit Ghosh, Lawrence Niu, Pamela Baker, Ivan Soltesz, Lydia Ng, Karel Svoboda, Loren Frank, Kristofer E Bouchard (2022) The Neurodata Without Borders ecosystem for neurophysiological data science eLife 11:e78362**
- <https://doi.org/10.7554/eLife.78362>

Version control, git, github

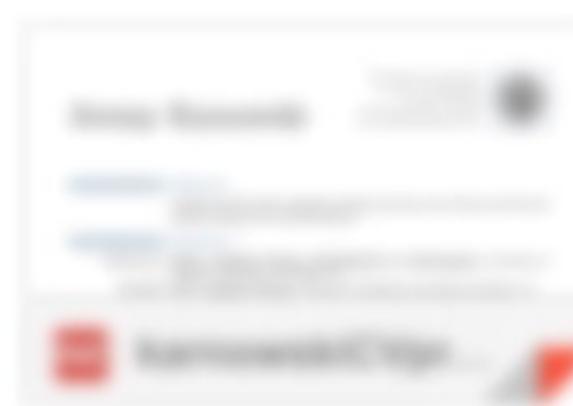
This sucks

 main_simple_bak9-pretty-good.c	Aug 1, 2008, 1:01 AM	33 KB	C Source
 main_simple_bak9-pretty-good.o	Aug 1, 2008, 1:00 AM	303 KB	object code
 main_simple_bak9-pretty-goodv2.c	Aug 2, 2008, 1:16 AM	33 KB	C Source
 main_simple_bak10.c	Sep 28, 2008, 1:16 PM	33 KB	C Source
 main_simple_bak11-workingUART_correctspeed.c	Aug 30, 2008, 2:49 AM	27 KB	C Source
 main_simple_bak11-workingUART_correctspeed.o	Aug 2, 2008, 1:17 AM	303 KB	object code
 main_simple_bak12_willspin.c	Aug 2, 2008, 1:30 AM	28 KB	C Source
 main_simple_bak12_willspin.o	Aug 2, 2008, 2:35 AM	301 KB	object code
 main_simple_bak13-worksA-D-nonoise-spins.c	Aug 7, 2008, 12:57 PM	26 KB	C Source
 main_simple_bak14-widersinefunctionsworkingrotation.c	Aug 8, 2008, 5:02 PM	26 KB	C Source
 main_simple_bak15-spins-stillneedsquadrantfixed.c	Aug 15, 2008, 7:32 PM	30 KB	C Source
 main_simple_bak16-15backup-spins-needs-improvement.c	Oct 15, 2008, 8:54 PM	31 KB	C Source
 main_simple_bak17-smoother-stillnostandingstart.c	Aug 16, 2008, 6:50 PM	30 KB	C Source
 main_simple_bak17-smoother-stillnostandingstart.o	Aug 18, 2008, 9:41 PM	305 KB	object code
 main_simple_bak18-notgood.c	Aug 18, 2008, 9:42 PM	31 KB	C Source
 main_simple_bak20SIMPLE-DCnotbrushless.c	Sep 17, 2009, 11:02 PM	27 KB	C Source
 main_simple_bak20WORKS_PWM_COMMAND_CONTROL.c	Aug 19, 2008, 12:54 AM	29 KB	C Source
 main_simple_timer_intrpt_bak.c	Aug 12, 2008, 12:16 AM	13 KB	C Source
 main_simple_timer_intrpt_bak2.c	Aug 12, 2008, 2:00 PM	13 KB	C Source
 main_simple_timer_intrpt_bak3.c	Aug 18, 2008, 12:14 AM	13 KB	C Source
 main_simple_timer_intrpt.c	Aug 18, 2008, 12:17 AM	13 KB	C Source
 main_simple_workingHWPWM.c	Aug 18, 2008, 7:19 PM	15 KB	C Source
 main_simple.c	Sep 17, 2009, 11:02 PM	29 KB	C Source

Thanks for chatting with me earlier today. I added the link to the visualization project into my resume and attached the resume. Thanks for any connections you can make for me. I'd love to know where you send it, so I can keep track of that. Thanks again!

Best,

Yup, this sucks too.



✉ May 11 ⭐ ↺ ↻



✉ May 11 ⭐ ↺ ↻

Actually, please use this one. I fixed a typo that was previously missed. Thanks!

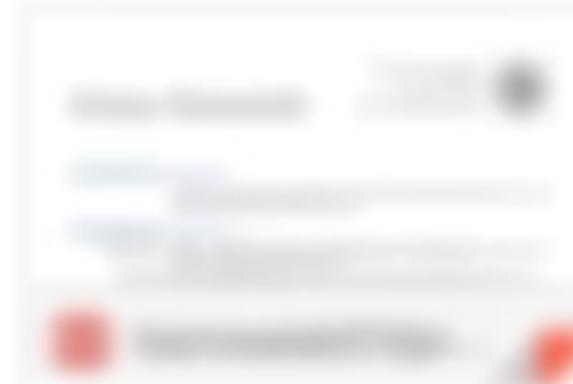
...



✉ May 11 ⭐ ↺ ↻

Final copy, I swear. Thanks for helping out.

...



adapted from Brad Voytek

This is a step in the right direction

Total: 9 edits | ^ v Only show named versions

SDSS Teacher Workshop

Considering how to incorporate data science into your high school STEM classroom?

~~The goal of this workshop is for you to leave with data science skills and applicable examples that can be used in your classroom.~~

The goal of this workshop is for you to leave with data science skills and applicable examples that can be used in your classroom.

This workshop will answer questions like—

- ~~What is data science?~~
- ~~How can high schoolers prepare for data science courses in college?~~
- ~~What does a career in data science involve? Dd~~

~~discuss answer questions like:~~

- ~~What is data science?~~
- ~~How can high schoolers prepare for data science courses in college?~~
- ~~What does a career in data science involve? what data science is, what high schoolers can do to best prepare for data science courses in college, and what a career in data science involves.~~
-

We will walk through how data scientists carry out projects using RStudio, introduce the basics of the R programming language, and work with real datasets to generate visualizations and analyze data. ~~The goal of this workshop is for you to leave with data science skills and applicable examples that can be used in your classroom.~~

Version history

MARCH

▶ March 4, 7:27 AM Current version ● Shannon Ellis

▶ March 3, 9:47 AM ● Donna LaLonde ● Shannon Ellis

FEBRUARY

▶ February 27, 6:29 AM ● Shannon Ellis

February 26, 5:44 PM ● Shannon Ellis

▶ February 26, 4:57 PM ● Shannon Ellis

▶ February 26, 3:50 PM ● Kelly McConville

▶ February 25, 3:53 PM ● Shannon Ellis

February 25, 3:33 PM ● Shannon Ellis

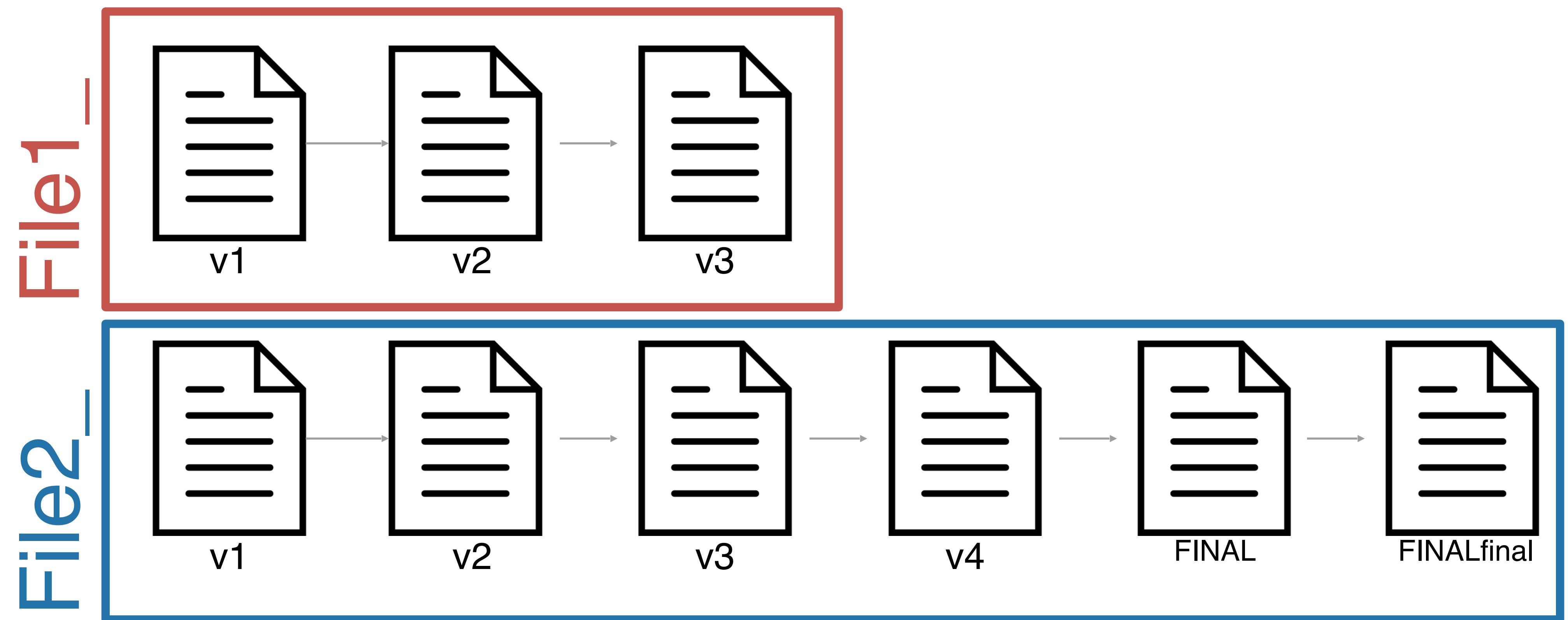
Show changes

Version Control

- Enables multiple people to simultaneously work on a single project.
- Each person edits their own copy of the files and chooses when to share those changes with the rest of the team.
- Thus, temporary or partial edits by one person do not interfere with another person's work

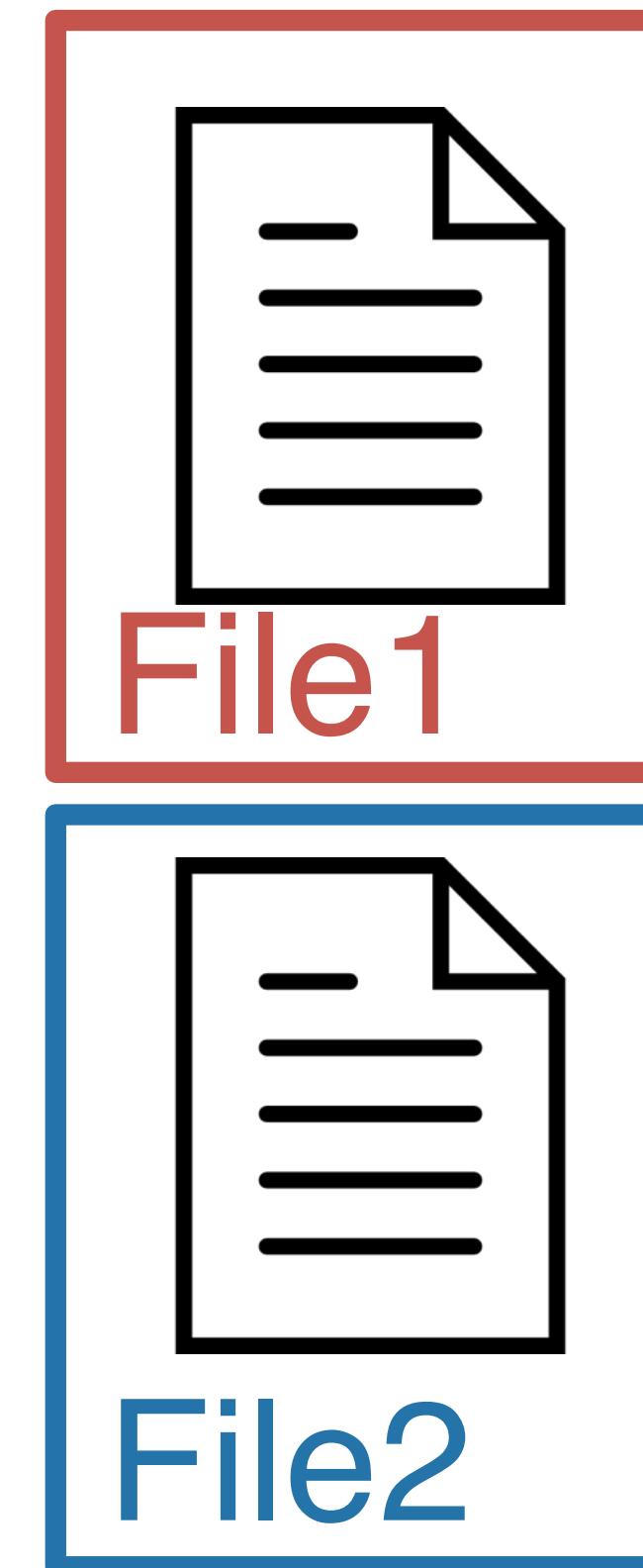
What is version control?

A way to manage the evolution of a set of files



What is version control?

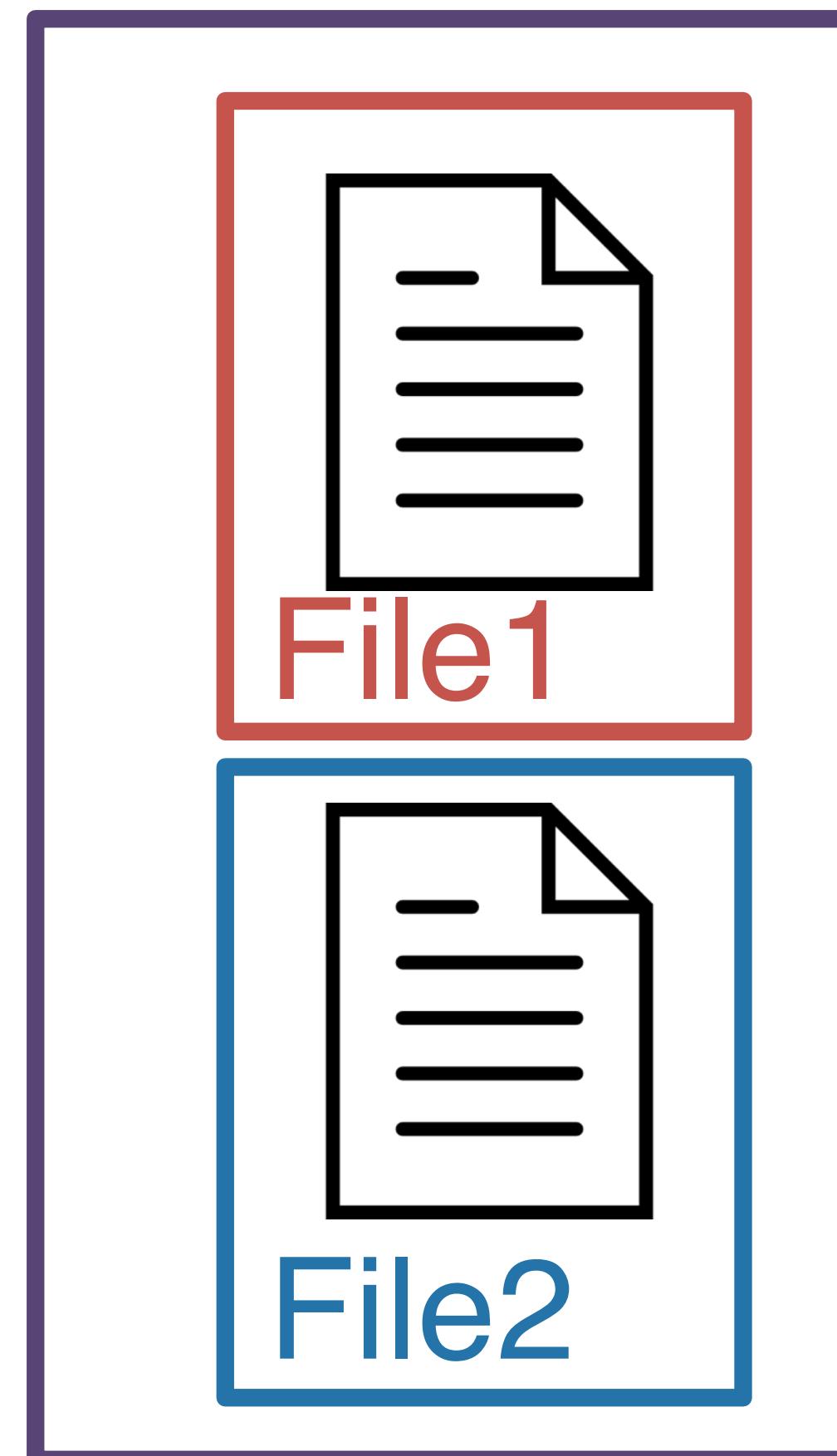
A way to manage the evolution of a set of files



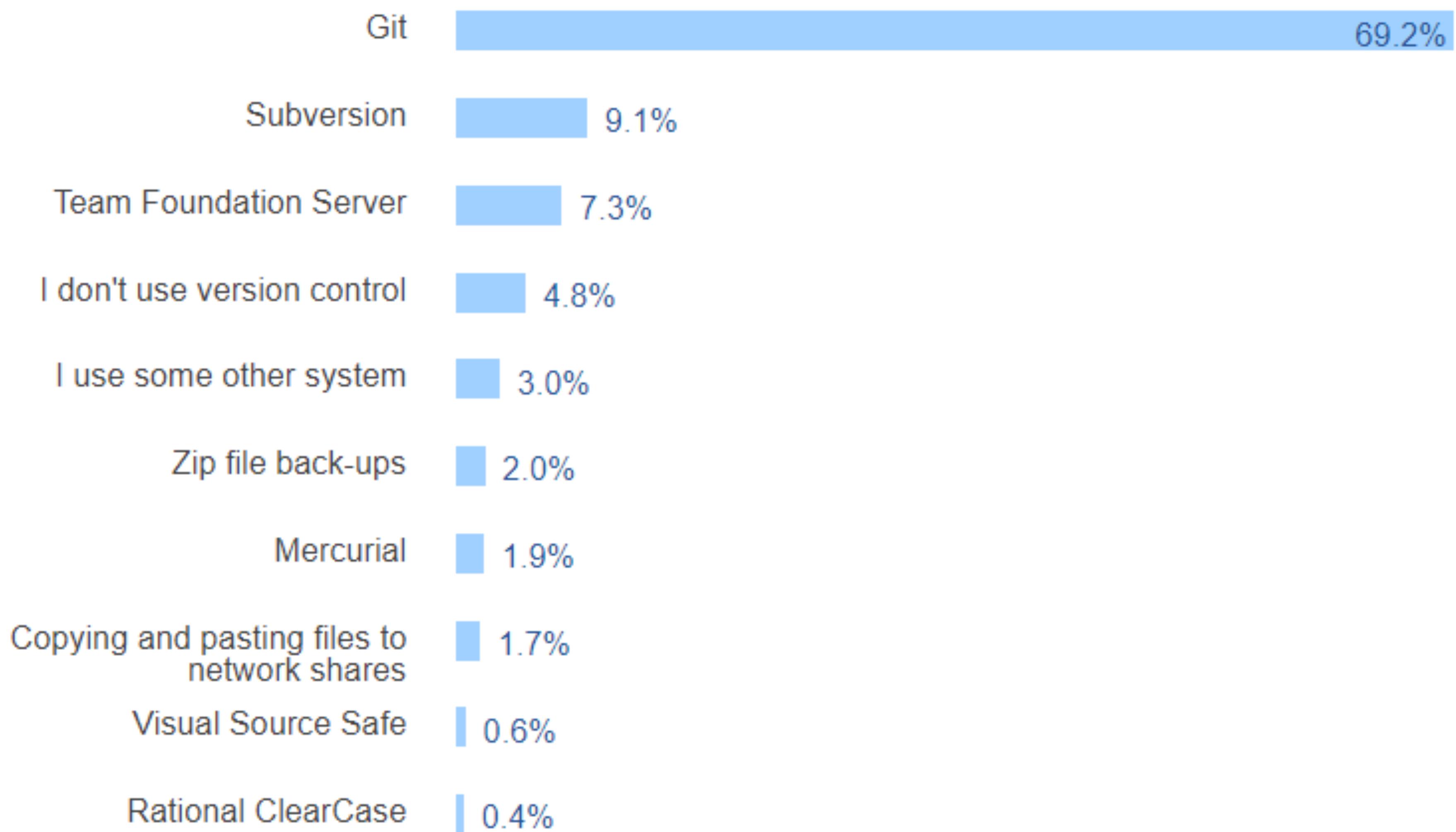
When using a version control system,
you have **one copy of each file** and
the *version control system tracks the
changes* that have occurred over time

What is version control?

A way to manage the evolution of a set of files



The set of files is referred to as a **repository** (**repo**)

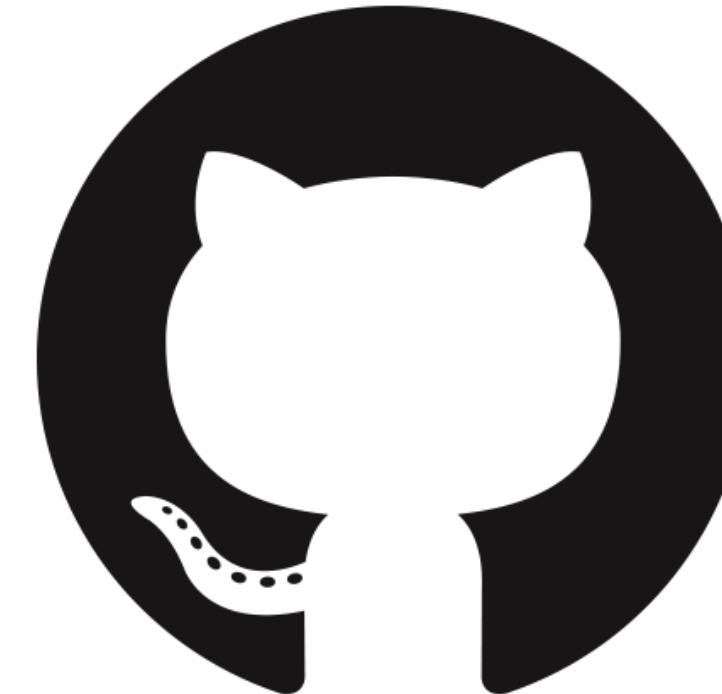


git & GitHub

git

the version control system

~ Track Changes
from Microsoft
Word....on
steroids



GitHub (or Bitbucket or
GitLab) is the home **where**
your git-based projects live

on the Internet.

~ Dropbox....but
way better

What version control looks like

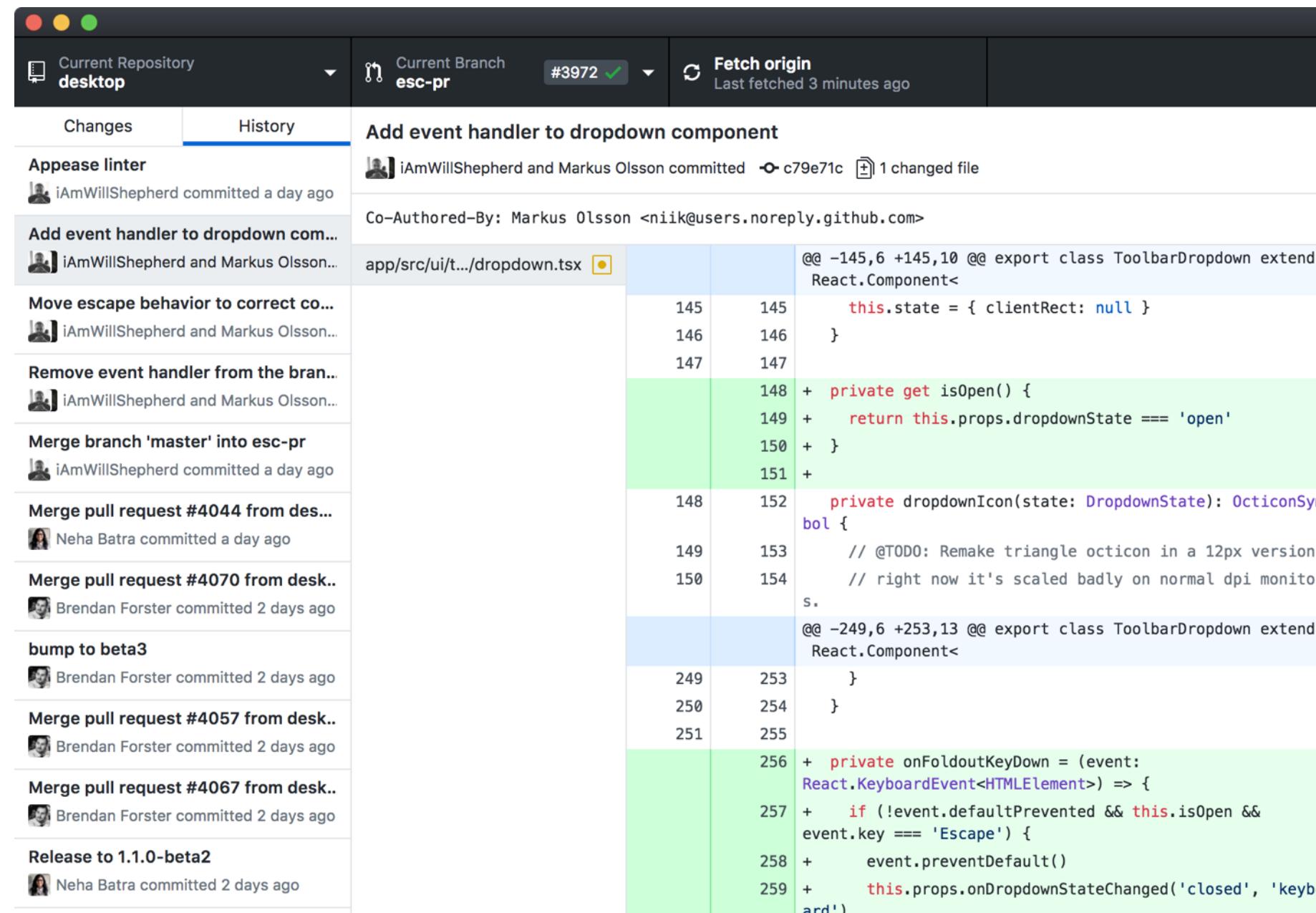
```
$ git clone https://www.github.com/username/repo.git  
$ git pull  
$ git add -A  
$ git commit -m "informative commit message"  
$ git push
```

Terminal
git

The screenshot shows the GitHub organization page for COGS108. At the top, there's a sidebar for 'Hands-On Data Science' with options for Repositories (22), People (7), Teams (2), Projects (0), and Settings. The main area is titled 'COGS108 - Data Science in Practice' and describes it as 'Course materials for Hands-On Data Science'. It lists pinned repositories: 'Overview' (Overview and map of the organization, which services COGS108: Hands-On Data Science, from UCSD.), 'Lectures-Sp19' (Slides and Notebooks used in Lecture for Sp19 COGS108), 'Section_Workbooks' (Workbooks for practice during discussion section), 'Tutorials' (Tutorial notebooks for hands-on data science, following along with the course topics), 'Projects' (Final Project materials and description), and 'Readings' (A curated list of suggested reading materials). Below these are sections for 'MyFirstPullRequest' (To be used for the assignments in Cogs 108) and 'Overview' (Overview and map of the organization, which services COGS108: Hands-On Data Science, from UCSD). A 'Top languages' section shows Jupyter Notebook and Python, and a 'Most used topics' section shows data-science, python, and tutorial.

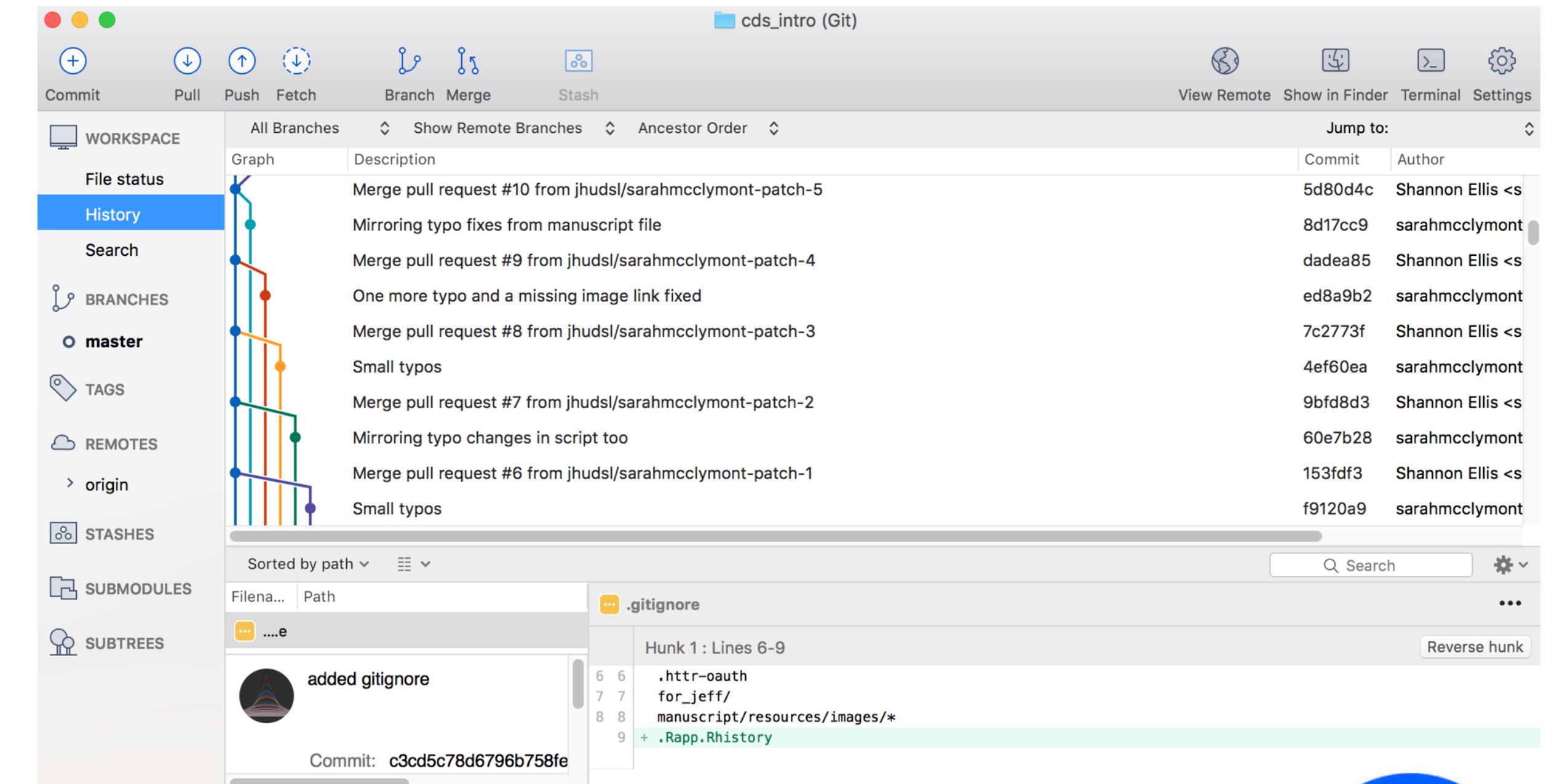


GUIs can be helpful when working with version control



The screenshot shows the GitHub Desktop application interface. At the top, it displays the current repository ("desktop") and branch ("esc-pr #3972"). Below this, a "Changes" tab is selected, showing a commit titled "Add event handler to dropdown component". The commit details show two authors: iAmWillShepherd and Markus Olsson. The commit message includes a Co-Authored-By line for Markus Olsson. The code diff shows several lines of TypeScript code for a "ToolbarDropdown" component, with changes highlighted in green and red. The commit history on the left lists various pull requests and merges from the master branch into esc-pr.

GitHub Desktop

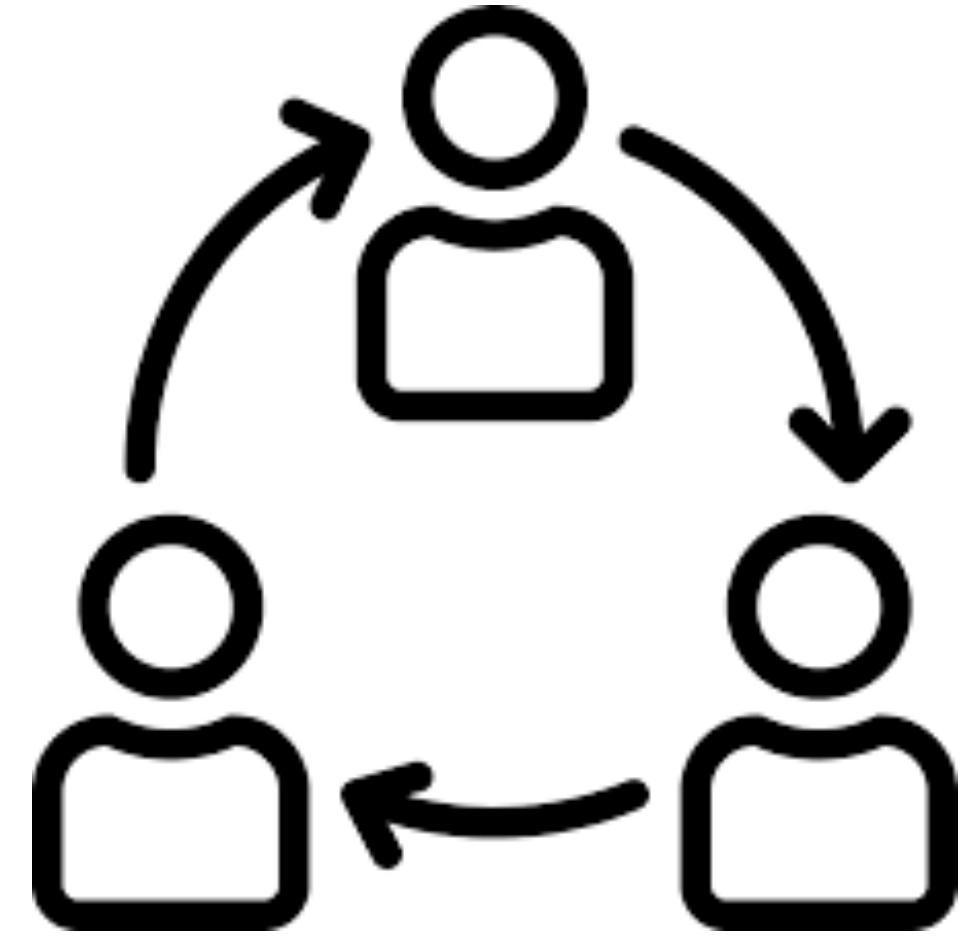


The screenshot shows the SourceTree application interface. At the top, it displays the current repository ("cds_intro (Git)"). Below this, a "History" tab is selected, showing a detailed git log. The log includes commits from various authors, such as Shannon Ellis and Sarah McClymont, with commit messages like "Merge pull request #10 from jhudsl/sarahmcclymont-patch-5" and "Small typos". The left sidebar shows sections for WORKSPACE, BRANCHES (with "master" selected), TAGS, REMOTES, STASHES, SUBMODULES, and SUBTREES. The bottom pane shows a file named ".gitignore" with its contents displayed. A large, colorful commit graph is visible on the right side of the main window.

SourceTree



Why version control with git and GitHub?



Collaboration



Returning to
a safe state

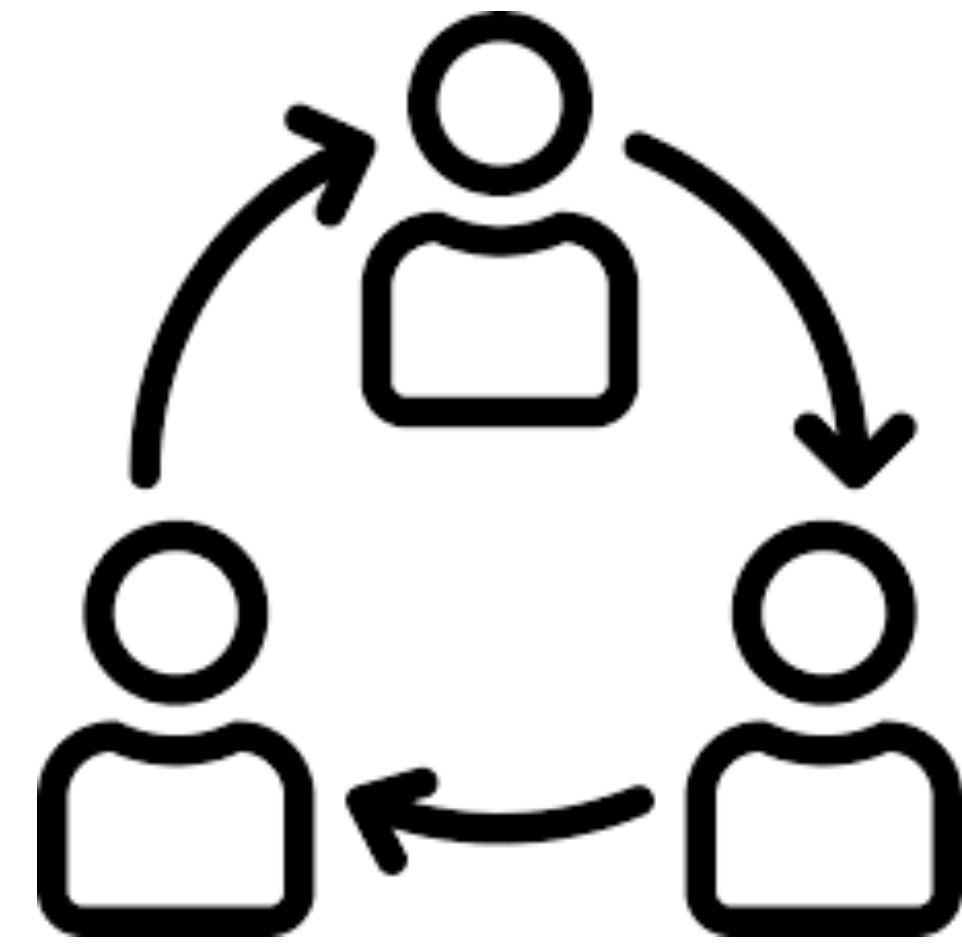


Exposure
for your
work

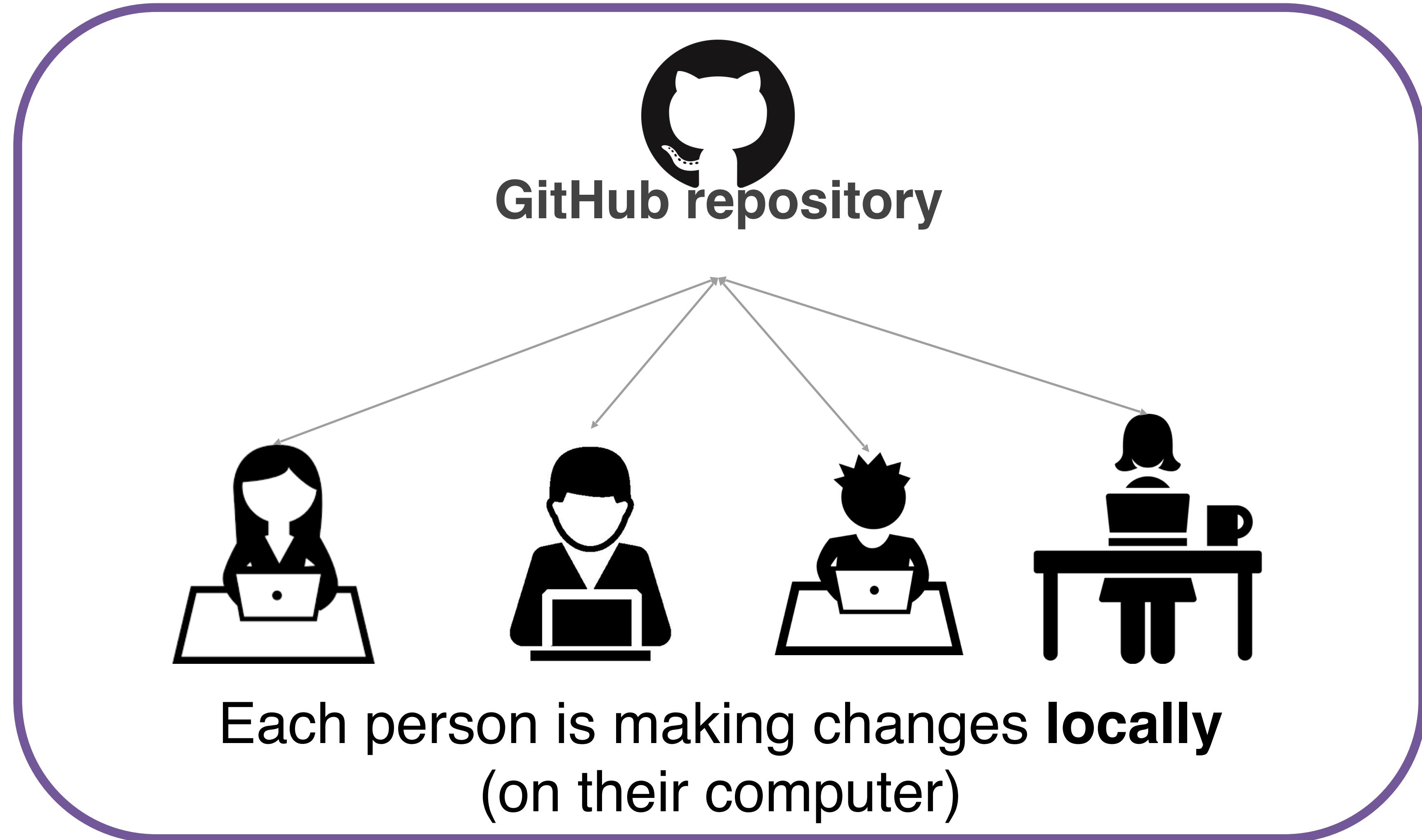


Tracking
others' work

Collaborate like you do with Google Docs



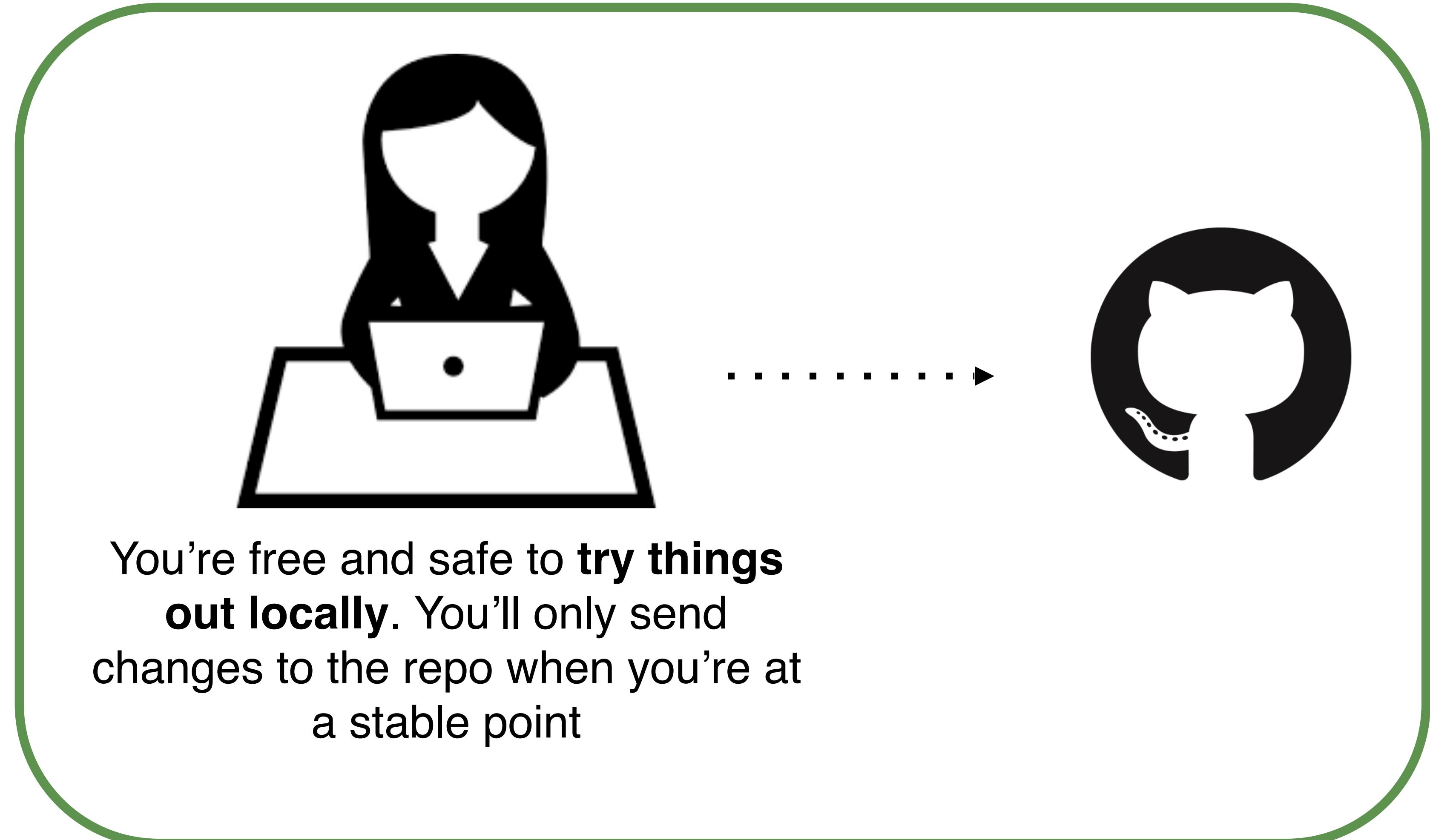
Collaboration



Make changes locally, while knowing a stable copy exists



Returning to a safe
state



Your repositories will be visible to others!



Exposure
for your
work

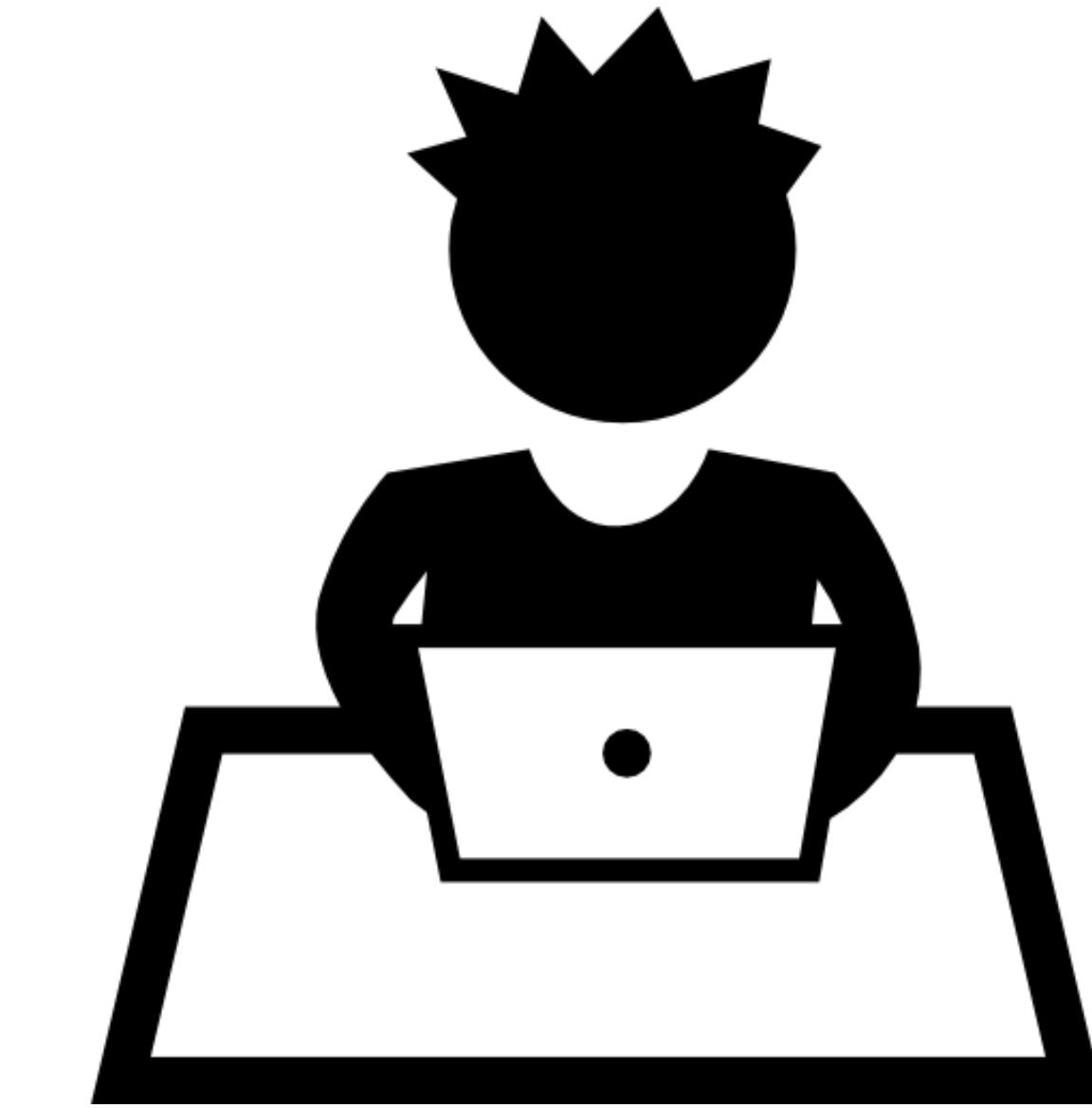


Your public GitHub repos
are your coding social
media

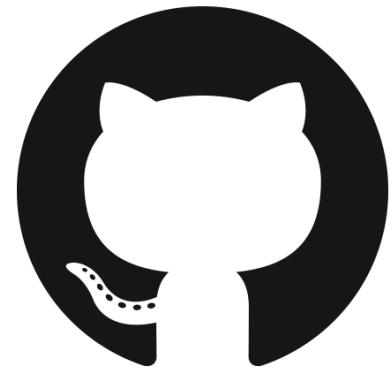
Keep up with others' work easily



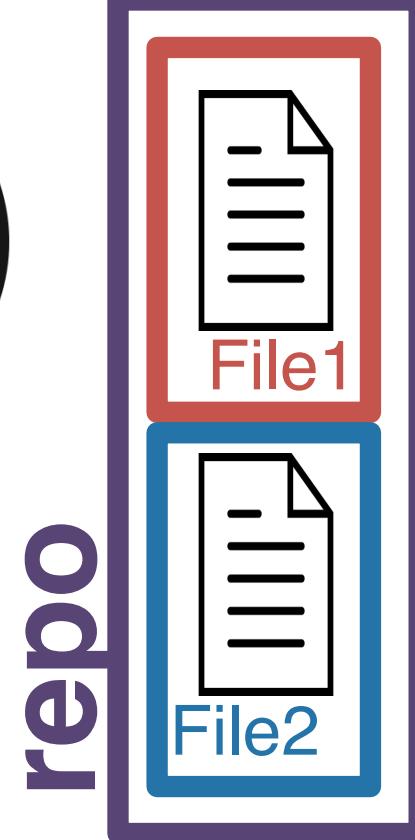
Tracking
others' work



As a social platform, you
can see others' work too!

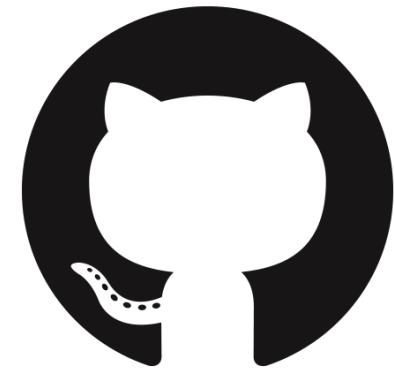


repo

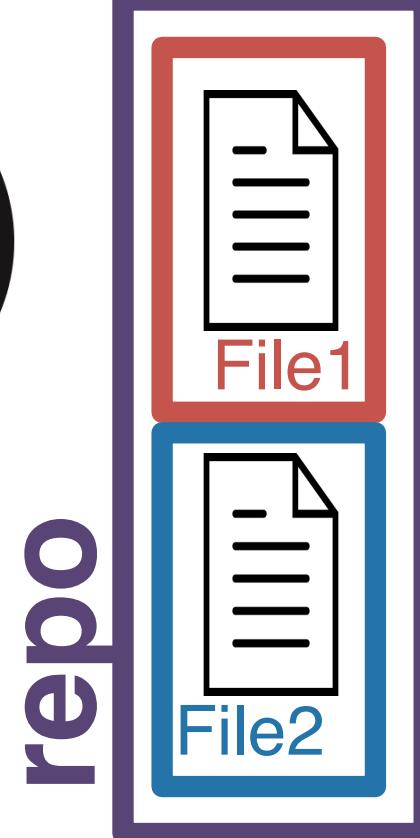


A **GitHub repo** contains all the files and folders for your project.

GitHub is a **remote host**. The files are geographically distant from any files on your computer.



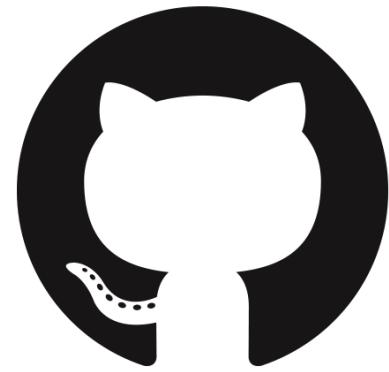
repo



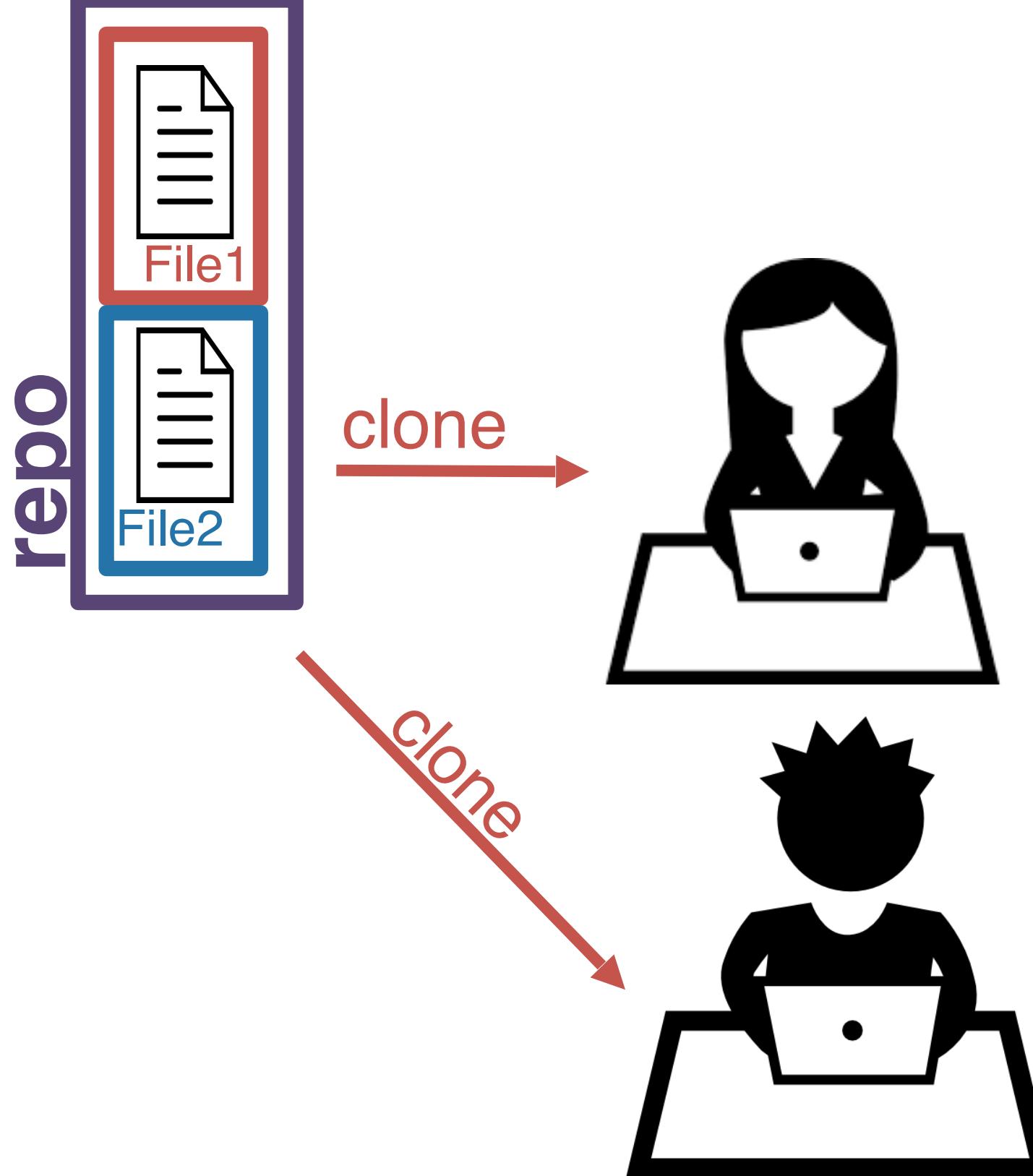
clone



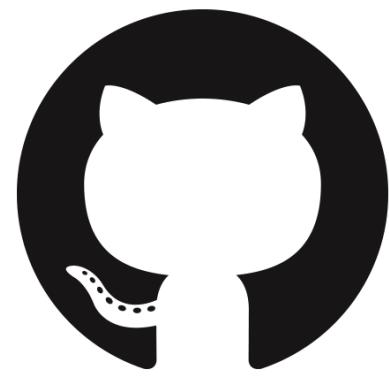
When you first make a copy onto your local computer (read: laptop), you **clone** the repository.



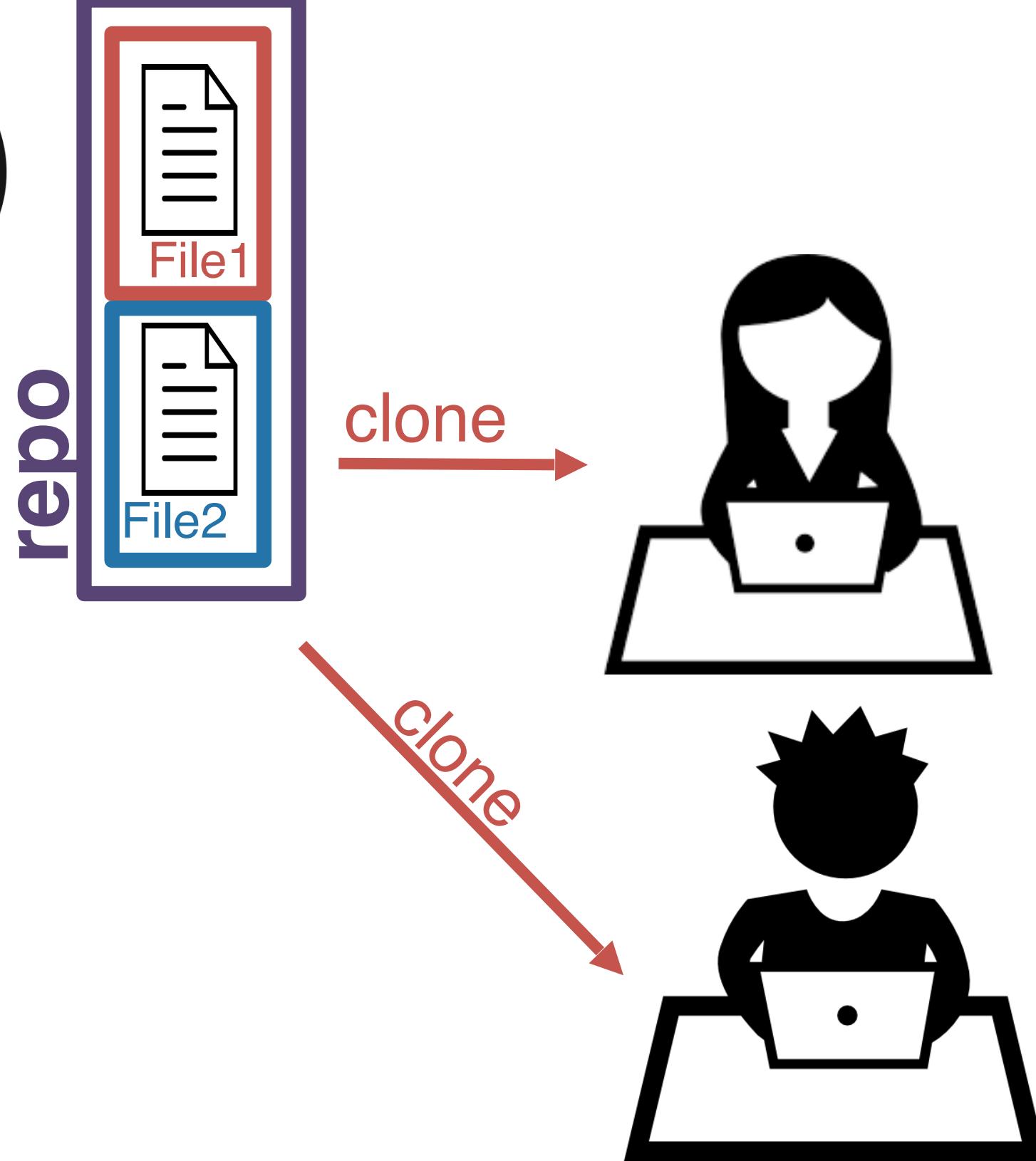
repo



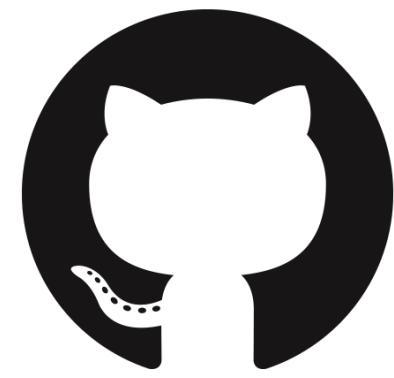
If someone else on your project cloned the repo at the same time, you would have identical copies of the project on each of your computers.



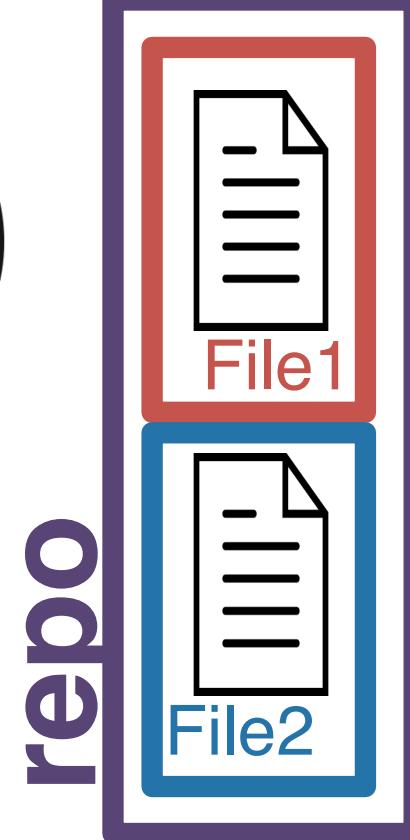
repo



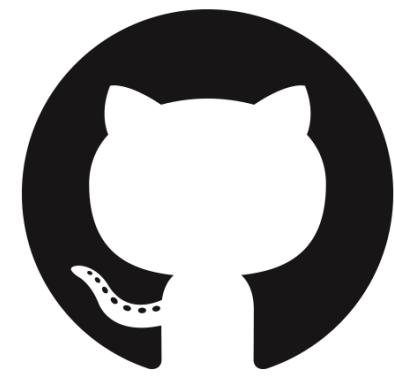
Yay! Everyone can
work on the project!



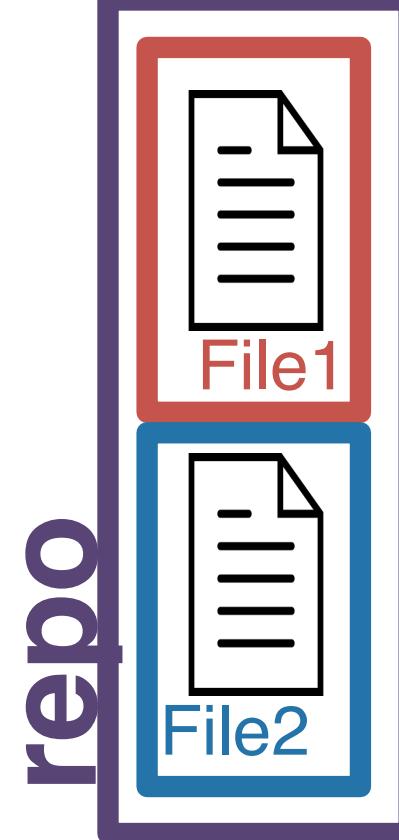
repo



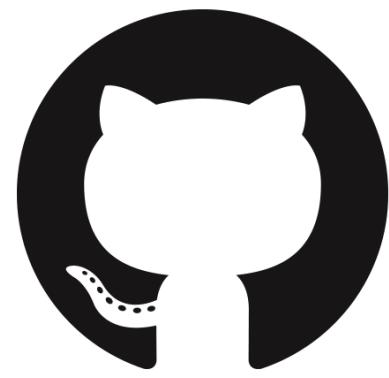
You decide you want to
change some of the text
in the project.



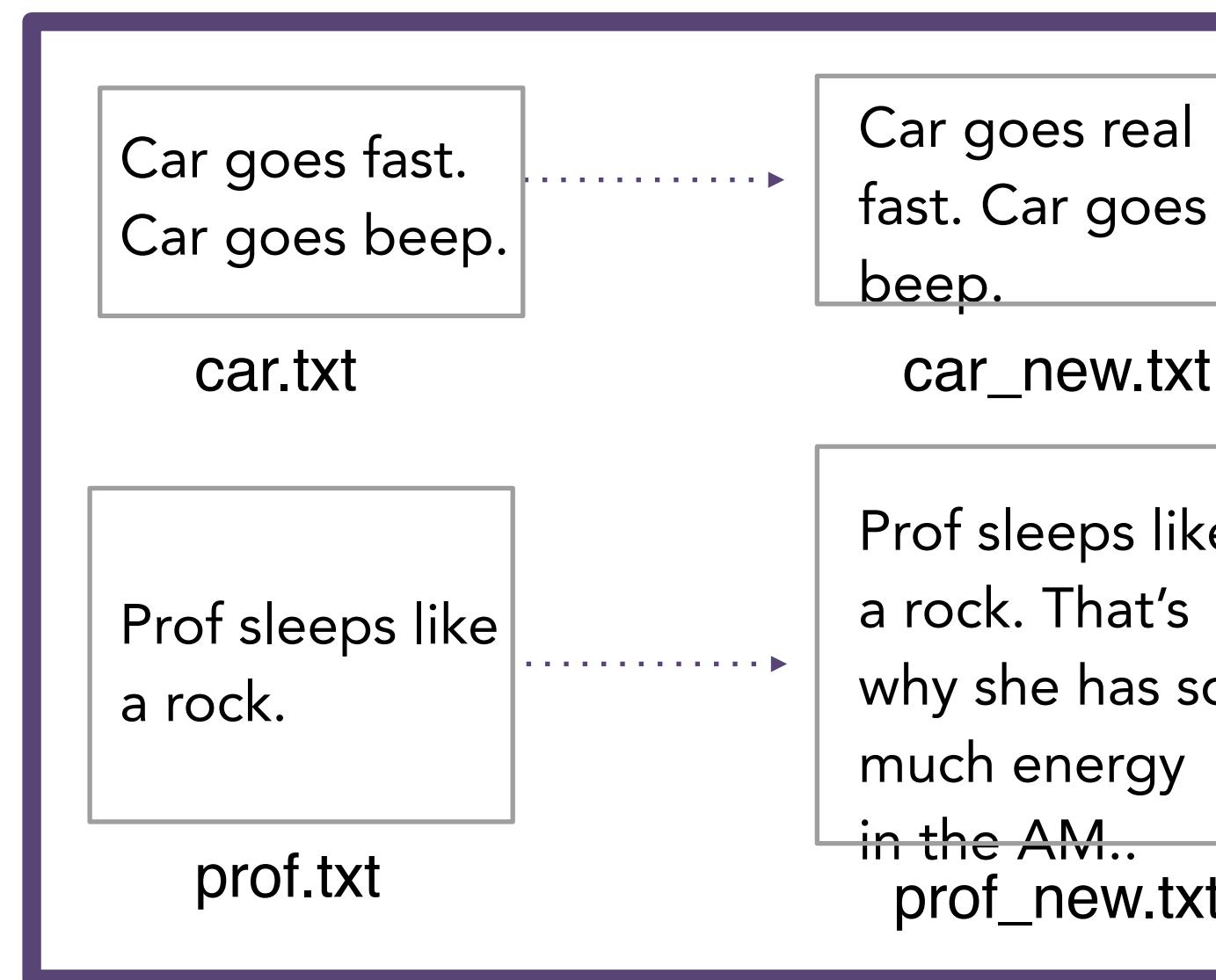
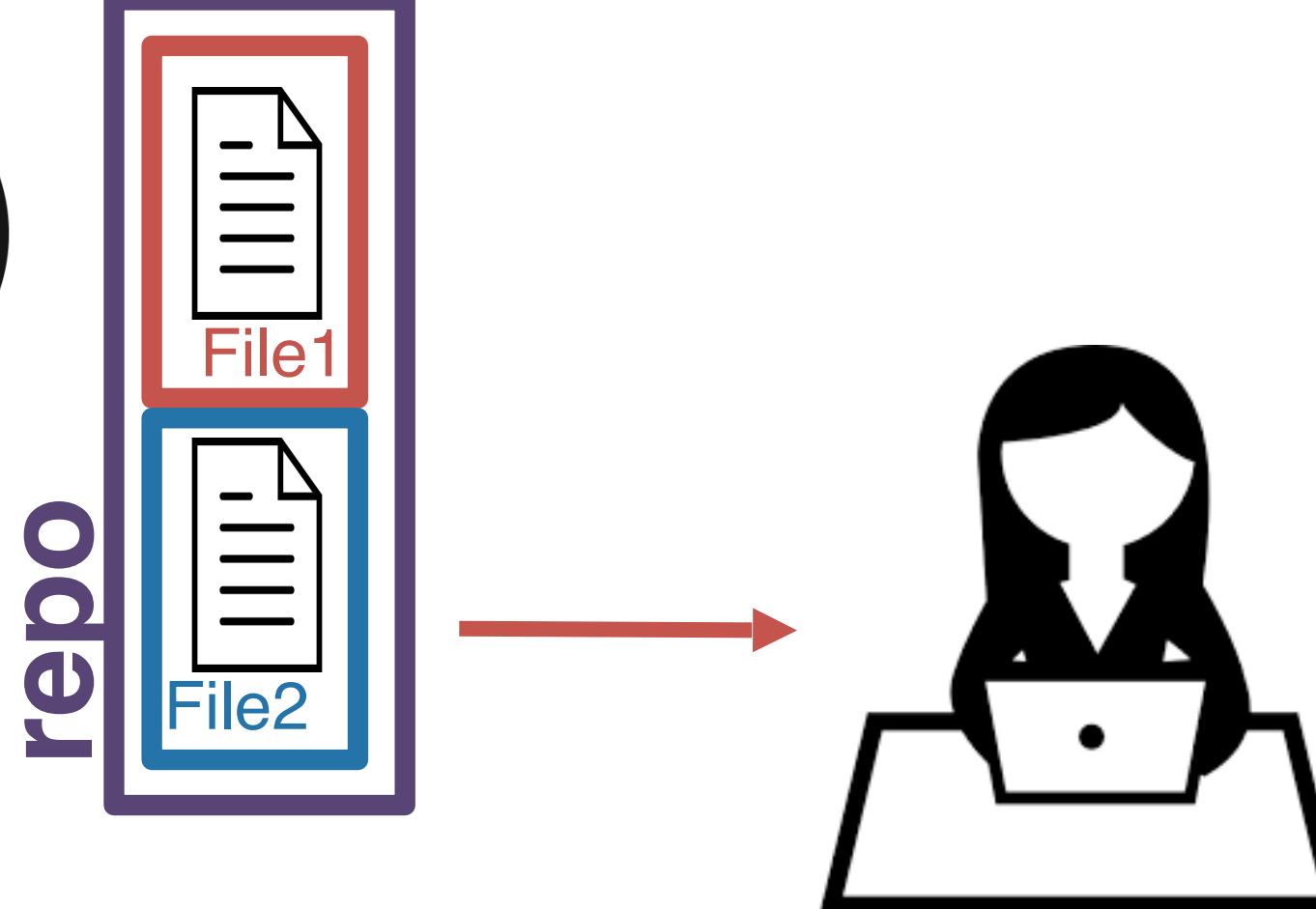
repo



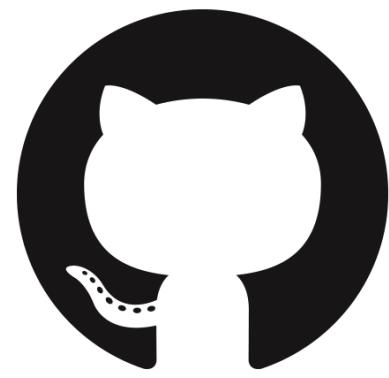
You decide you want to
change some of the text
in the project.



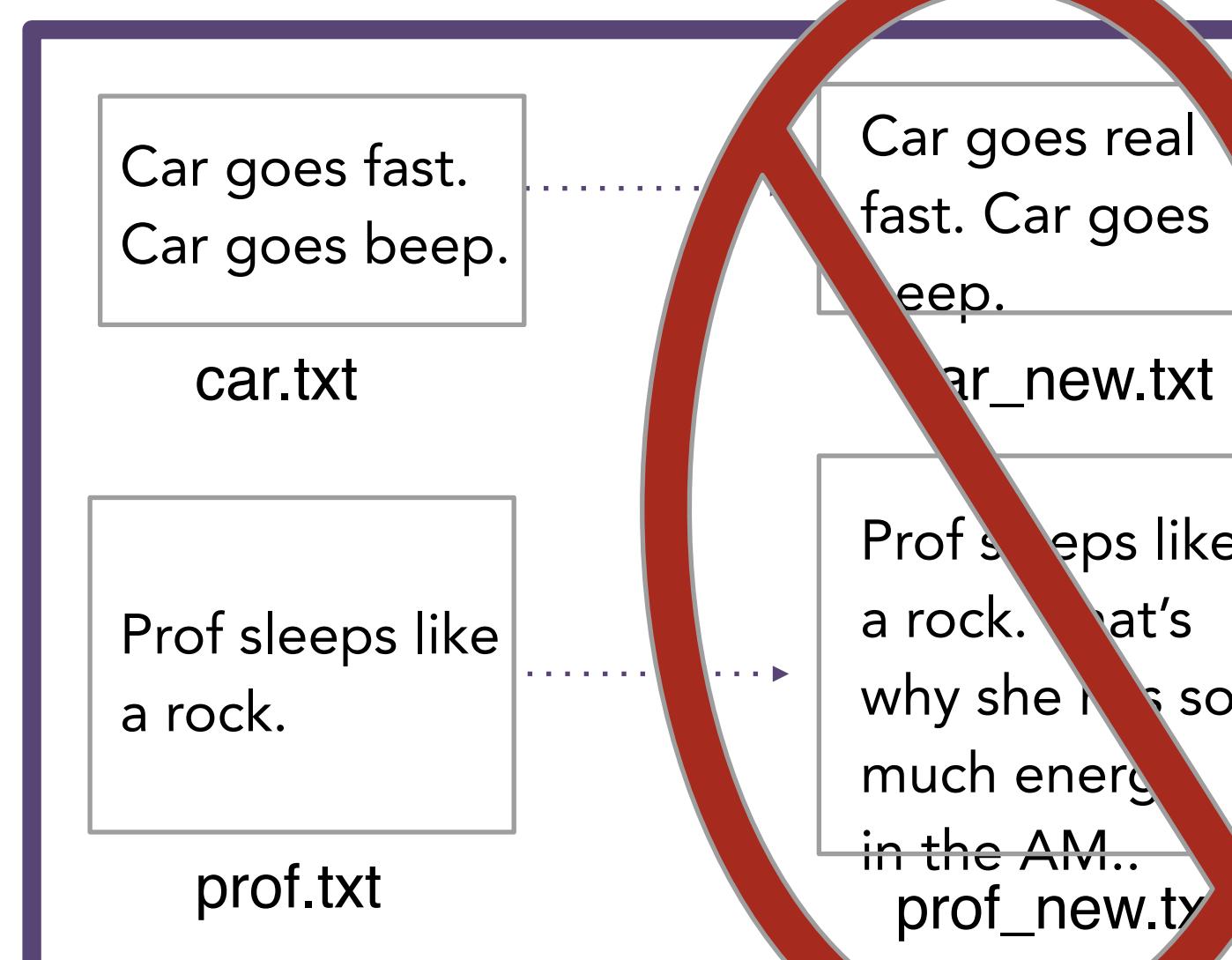
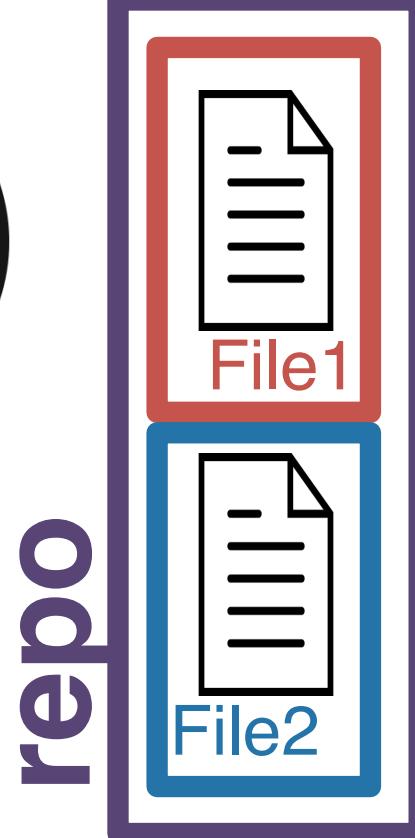
repo



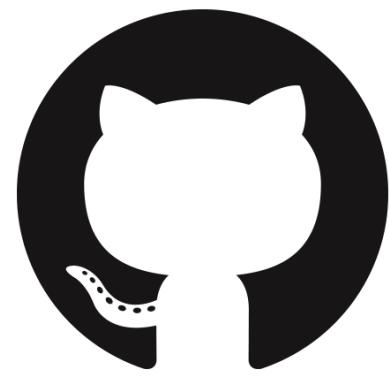
without git...you'd
likely rename
these files....



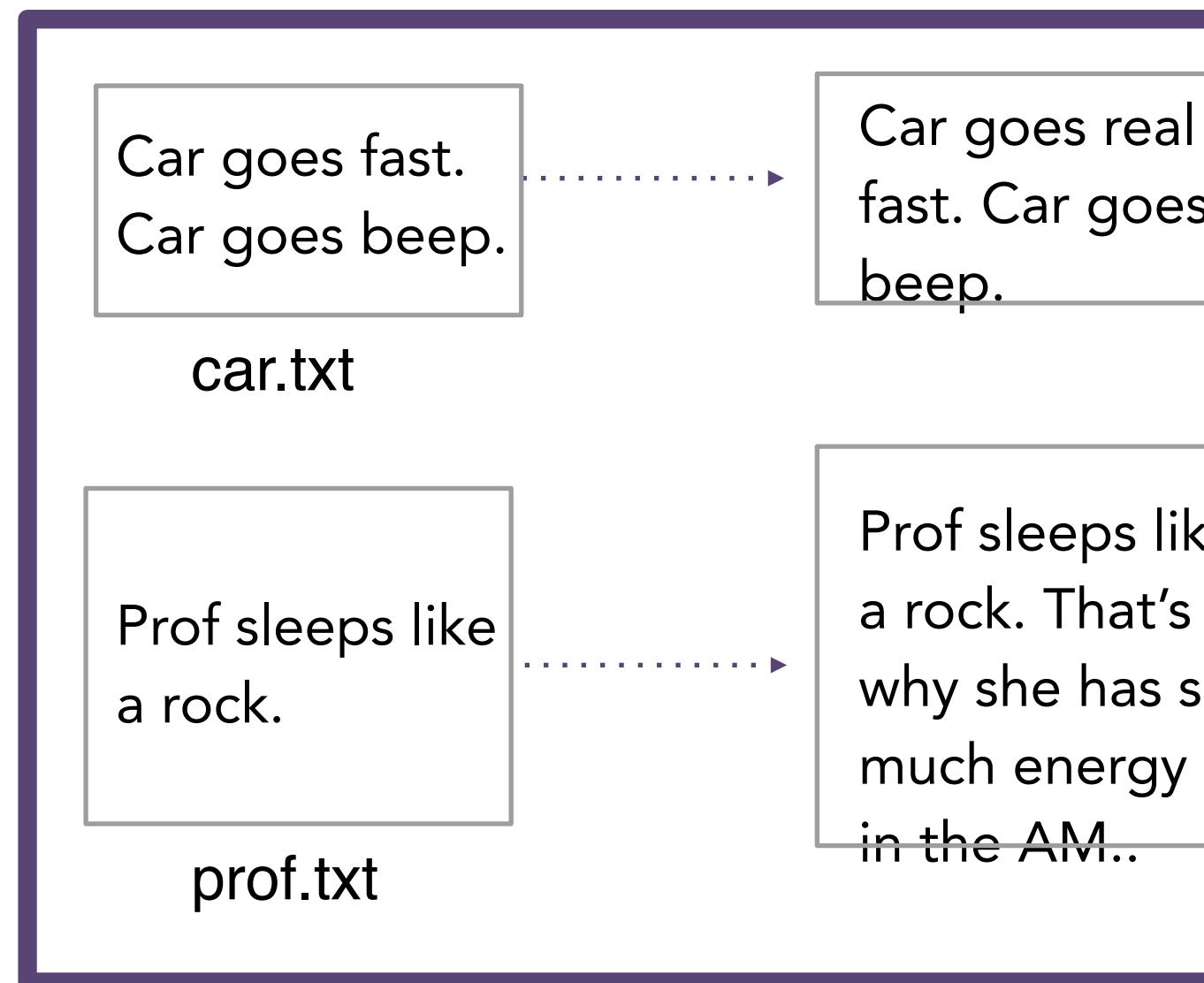
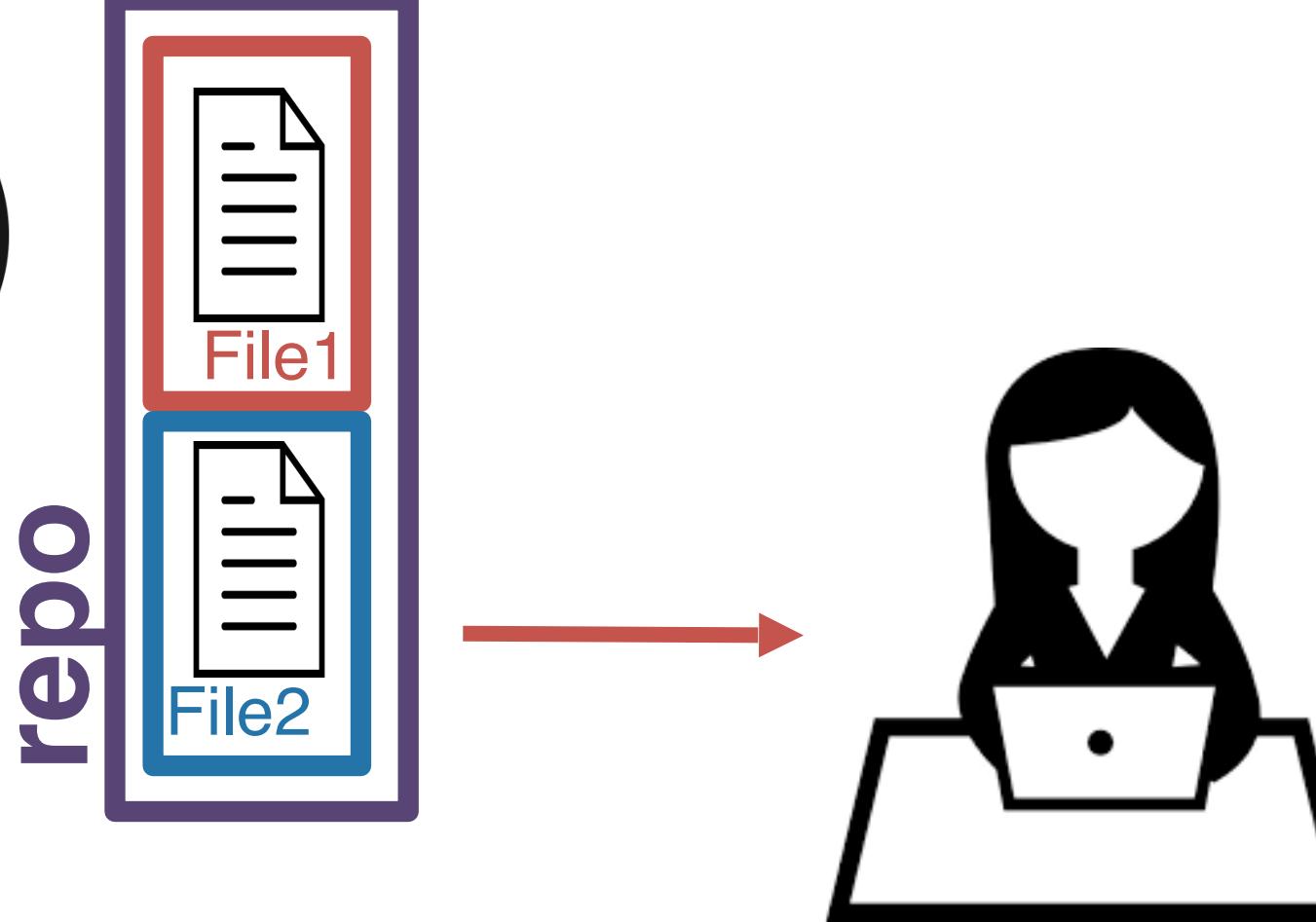
repo



Thank
goodness those
days are over!



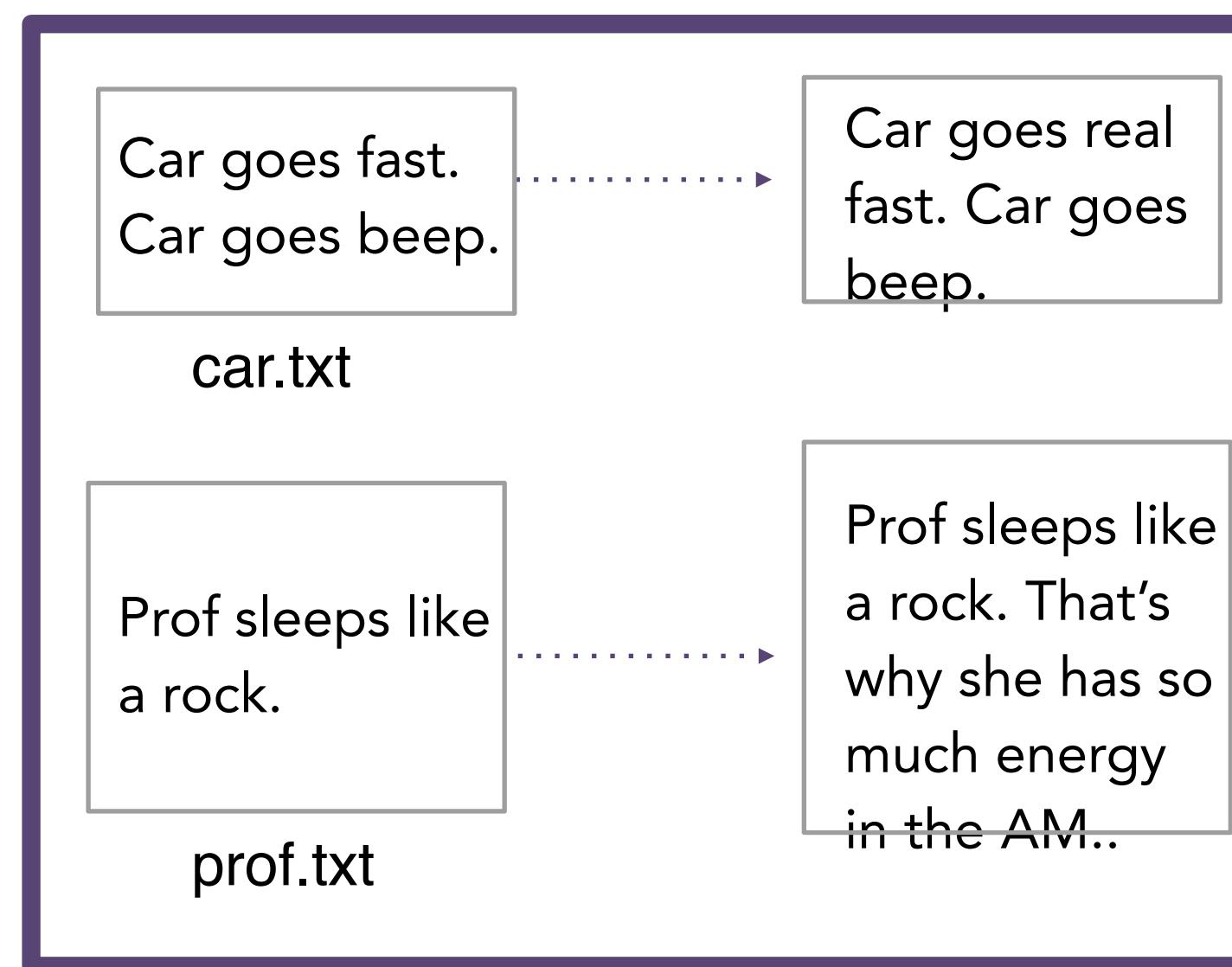
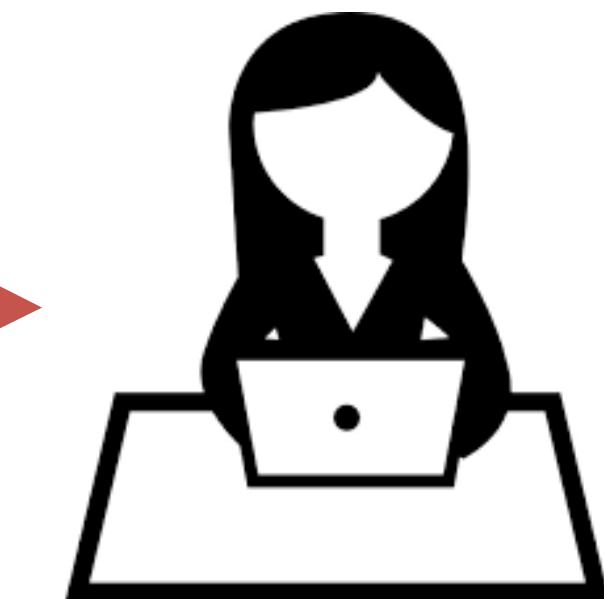
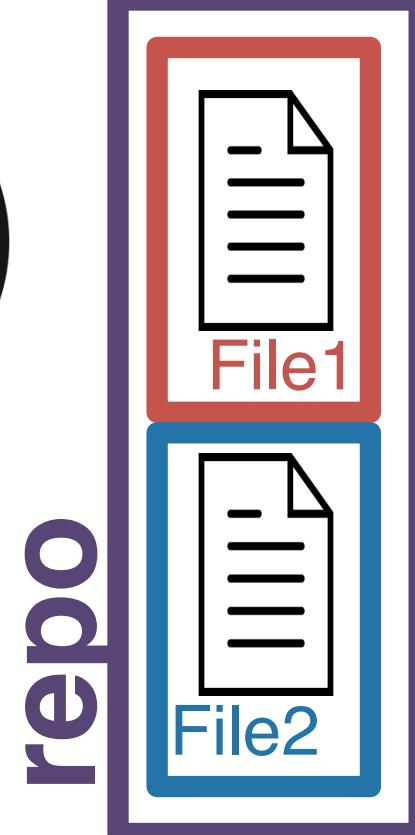
repo



Instead, you tell git which files you'd like to keep track of using **add**. This process is called *staging*.

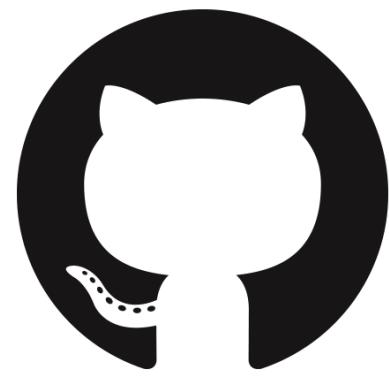


repo

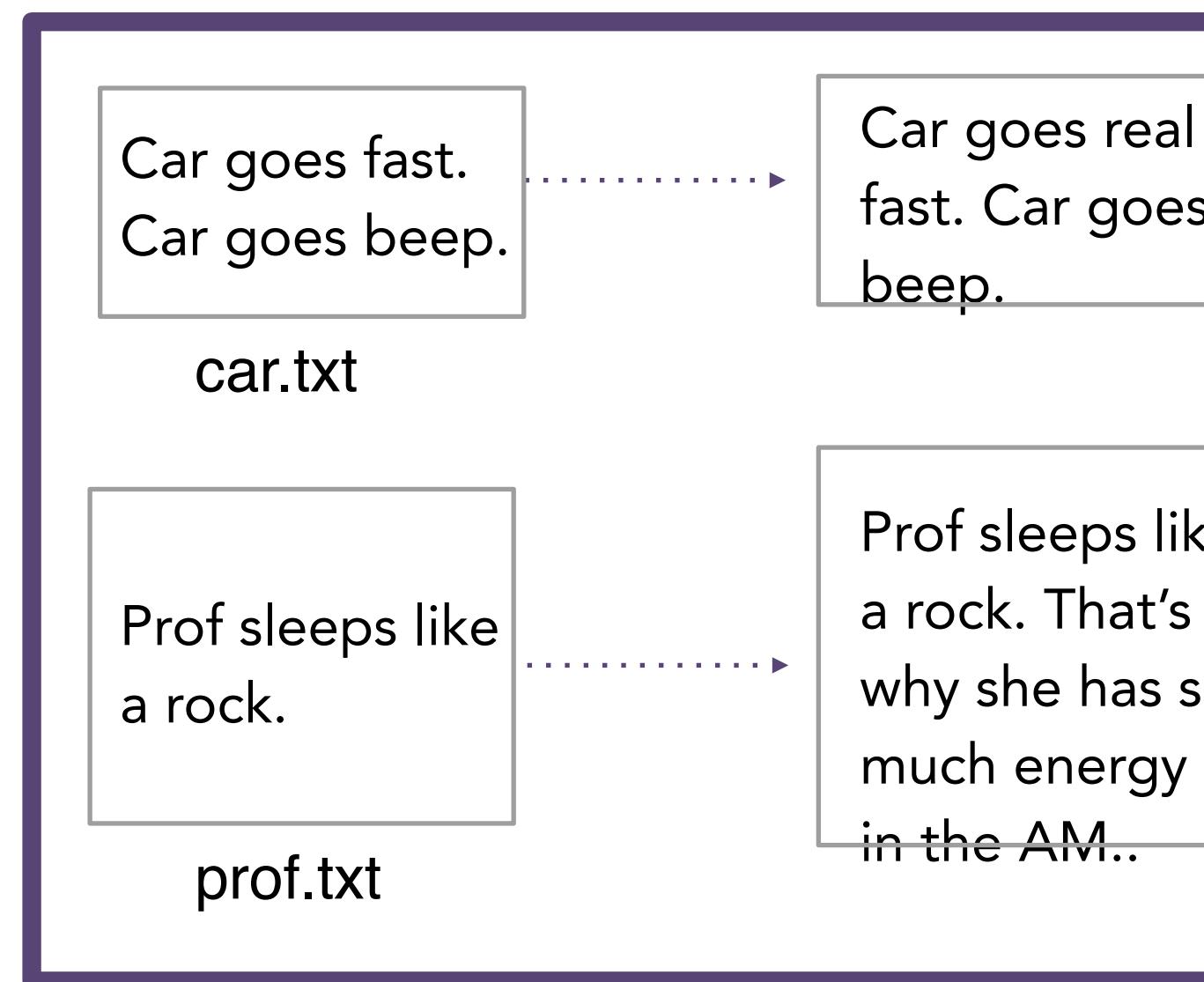
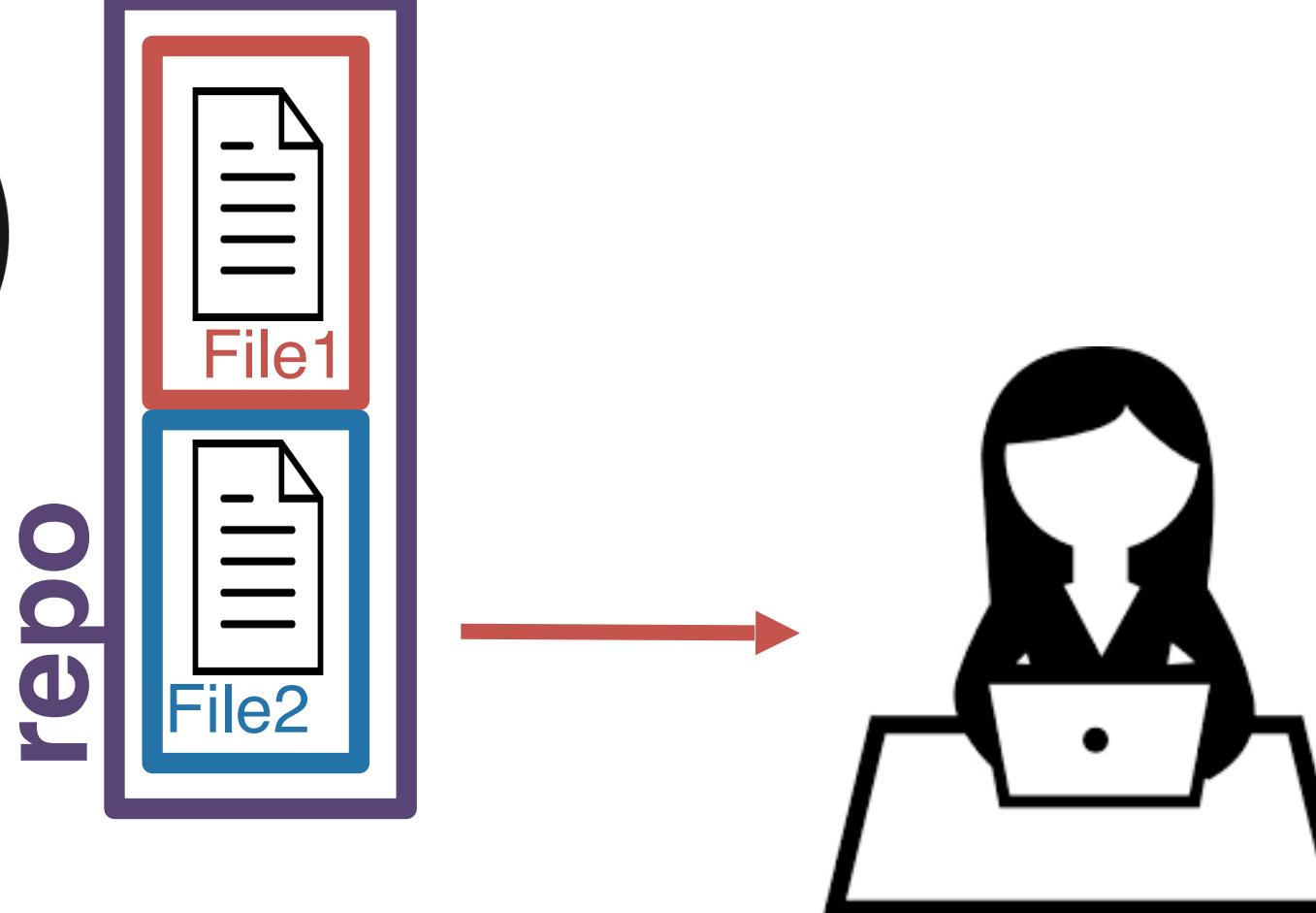


git add file	stages specified file (or folder)
git add .	stages new and modified files
git add -u	stages modified and deleted files
git add -A	stages new, modified, and deleted files
git add *.csv	Stages any files with .csv extension
git add *	Use with caution: stages everything

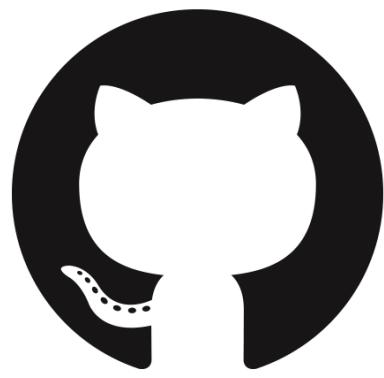
Instead, you tell git which files you'd like to keep track of using **add**. This process is called *staging*.



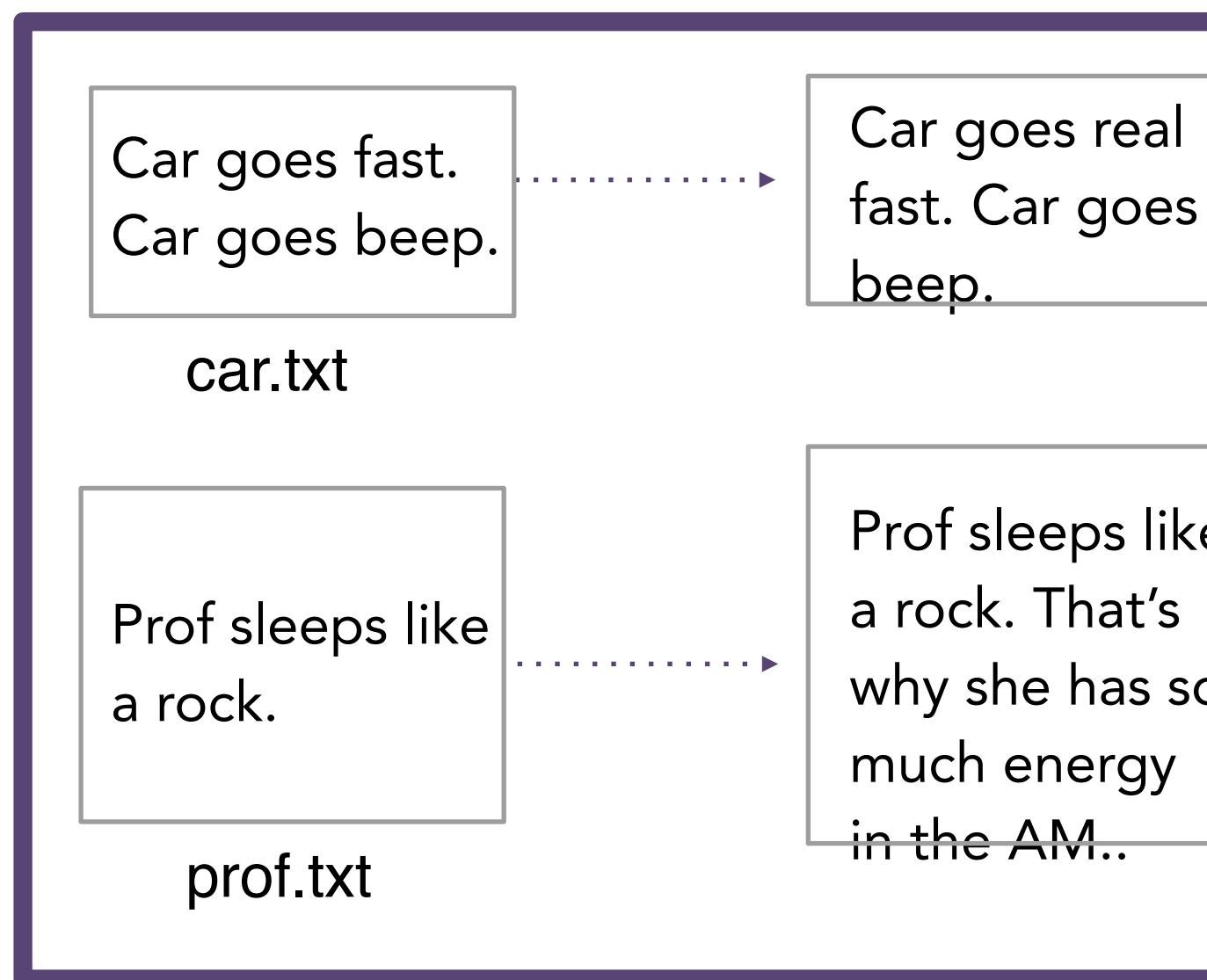
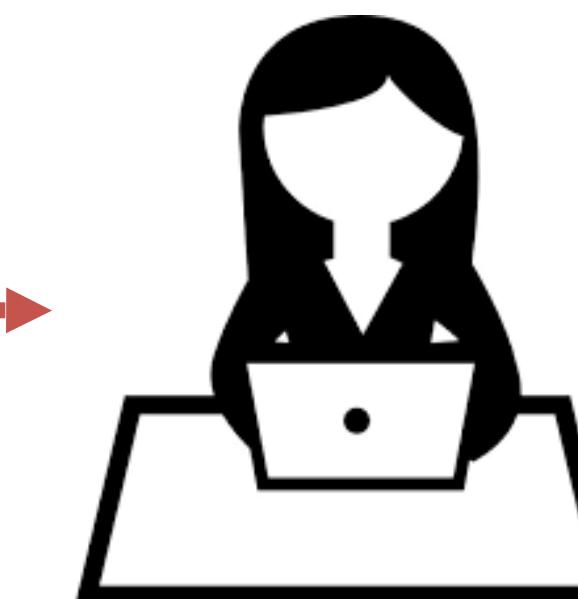
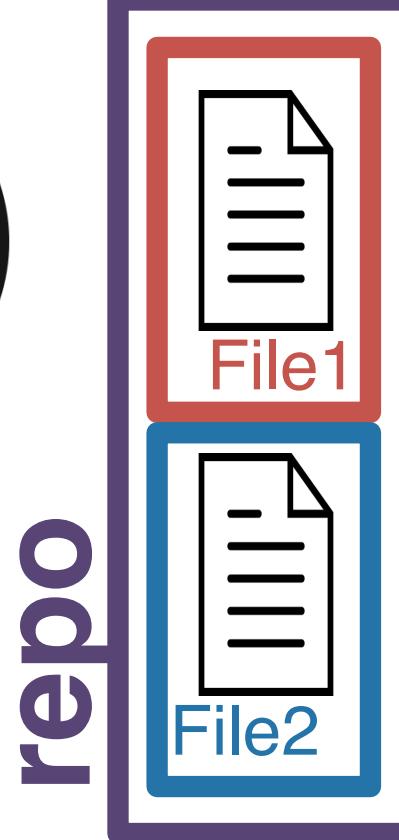
repo



Then, you create a snapshot of your files at this point. This snapshot is called a **commit**.



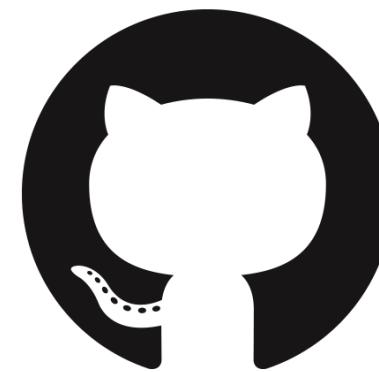
repo



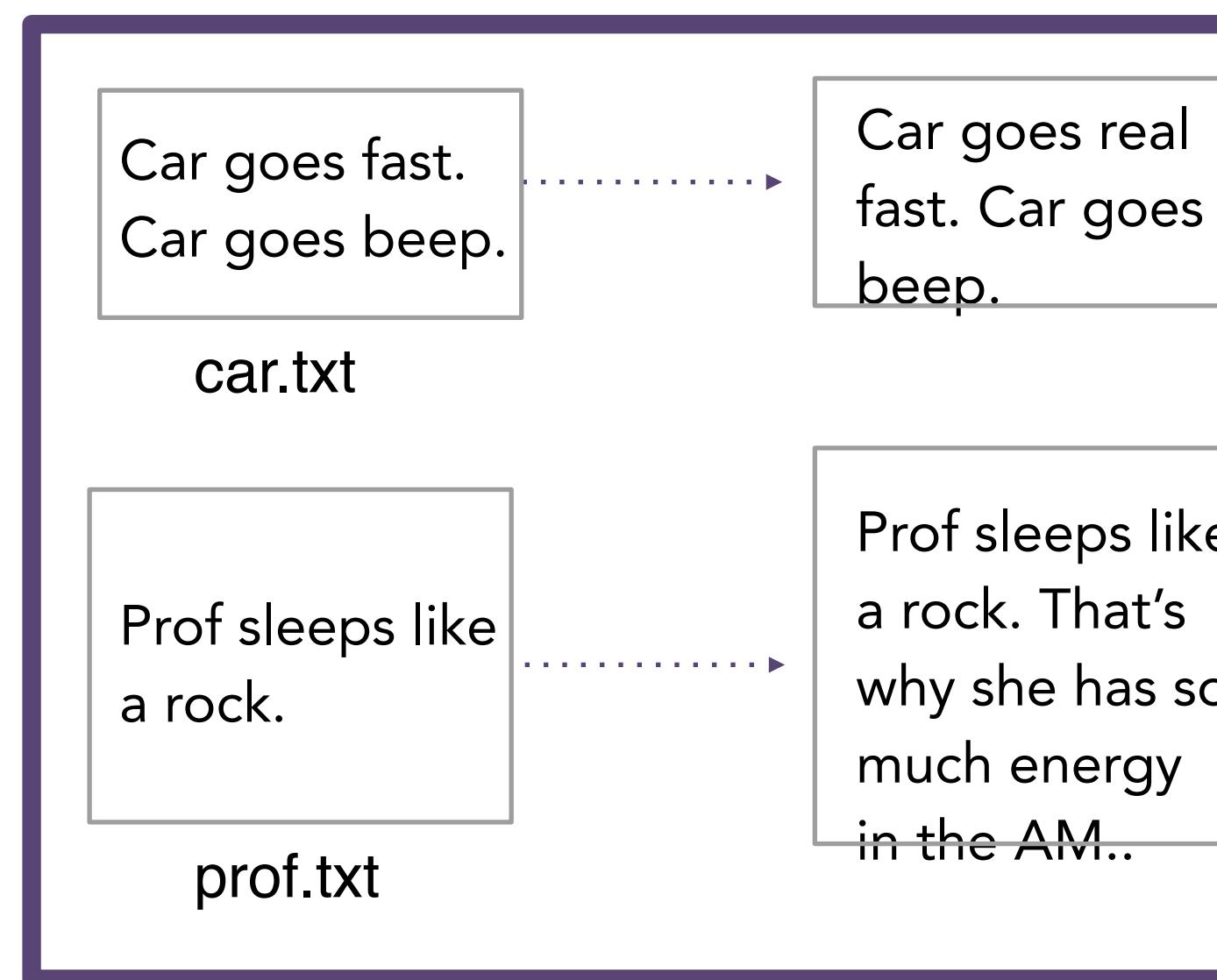
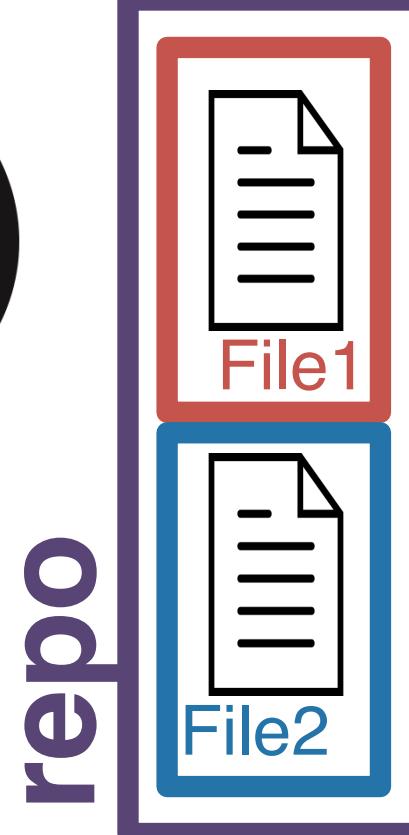
Then, you create a snapshot of your files at this point. This snapshot is called a **commit**.



A **commit** tracks
who, what, and
when



repo

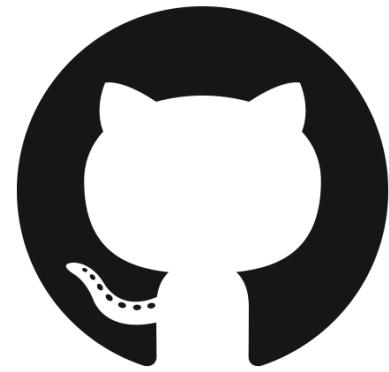


A **commit** tracks
who, what, and
when

You can make commits more informative by adding a **commit message**.

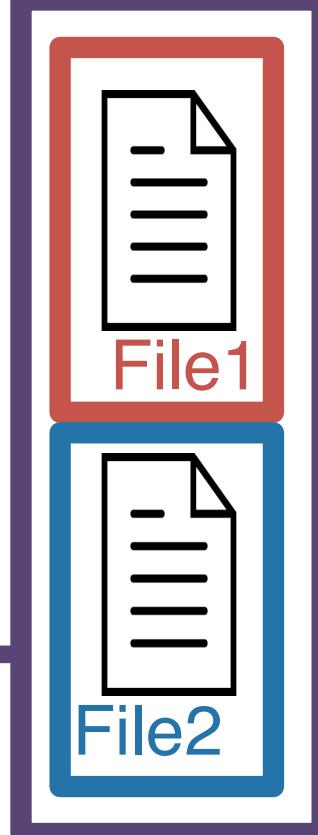
Example: `git commit -m 'fix typos in car and prof'`

Then, you create a snapshot of your files at this point. This snapshot is called a **commit**.



repo

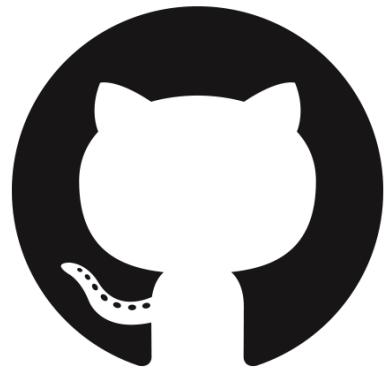
repo



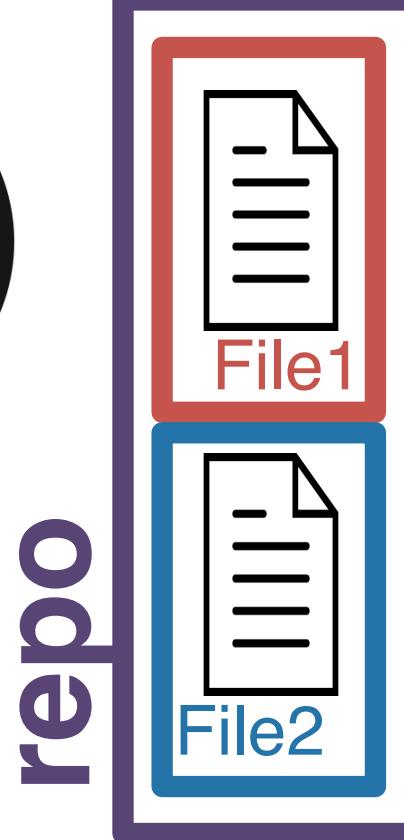
Shannon Ellis

3/28/21 3:28pm

fix typos in car and prof



repo



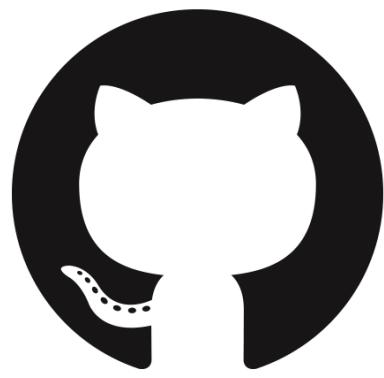
push

Remember, you're not the only one working on this project though! You want your teammates to have access to these changes! You **push** these changes back to the remote.

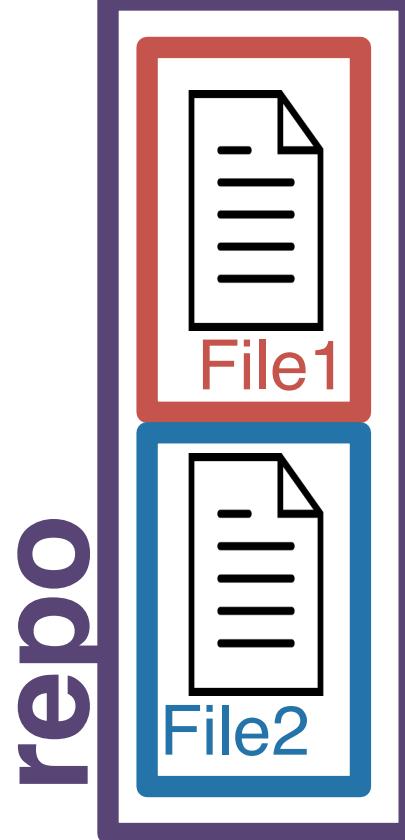


Shannon Ellis
3/28/21 3:28pm

fix typos in car and prof

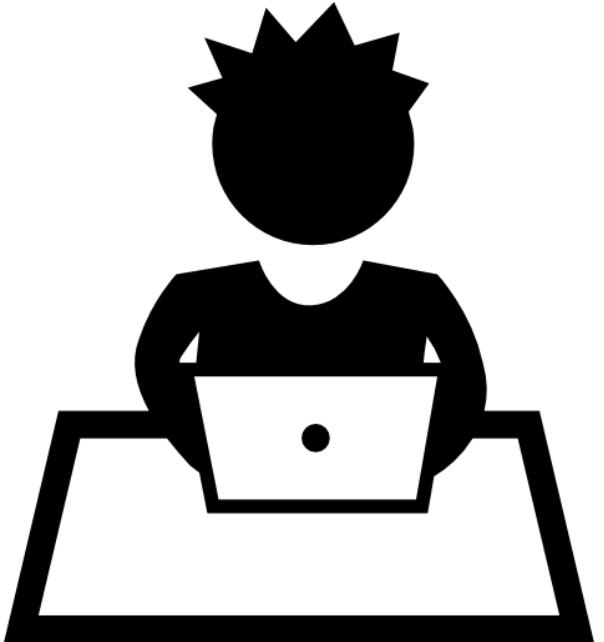


repo



Shannon Ellis
3/28/21 3:28pm

fix typos in car and prof

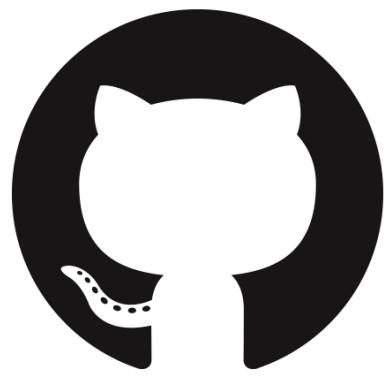


Your teammate is still
working with the (out-
of-date) copy he
cloned earlier!

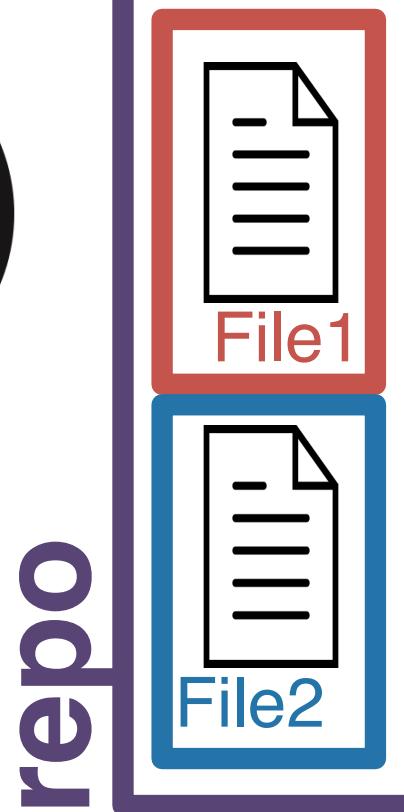


Shannon Ellis
3/28/21 3:28pm

fix typos in car and prof



repo



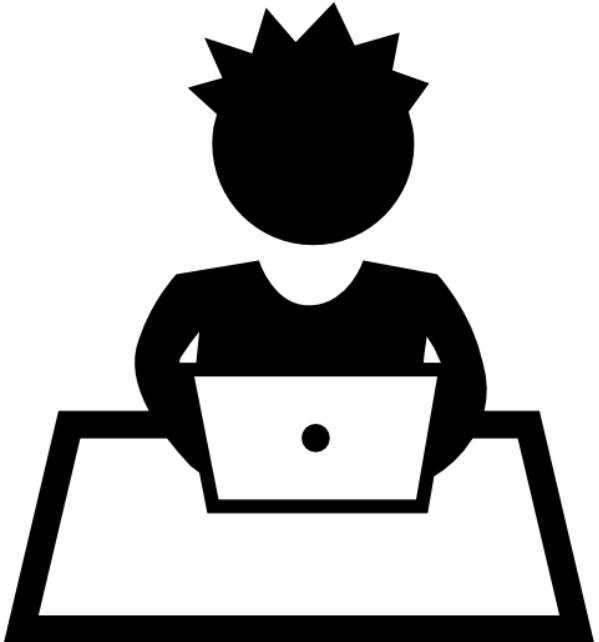
To catch up, your teammate will have to
pull the changes from GitHub (remote)



Shannon Ellis

3/28/21 3:28pm

fix typos in car and prof



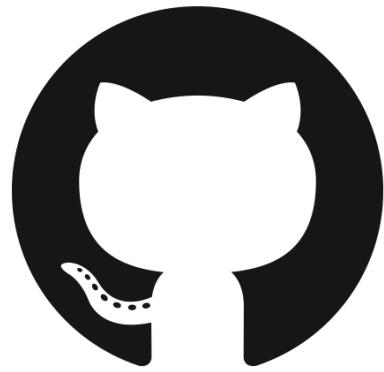
Your teammate is still
working with the (out-
of-date) copy he
cloned earlier!



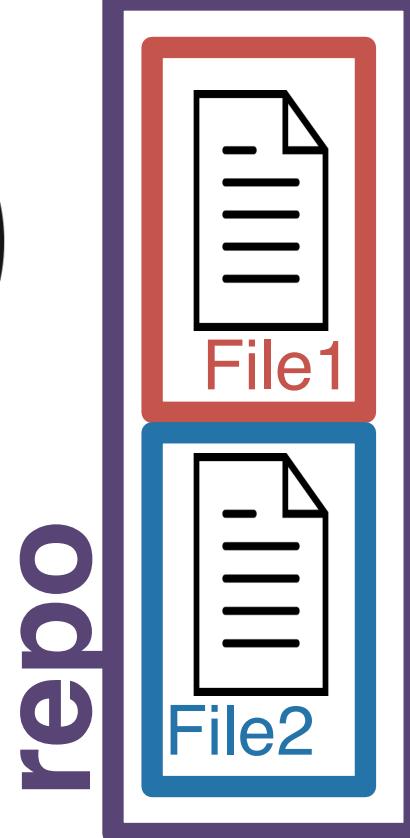
Shannon Ellis

3/28/21 3:28pm

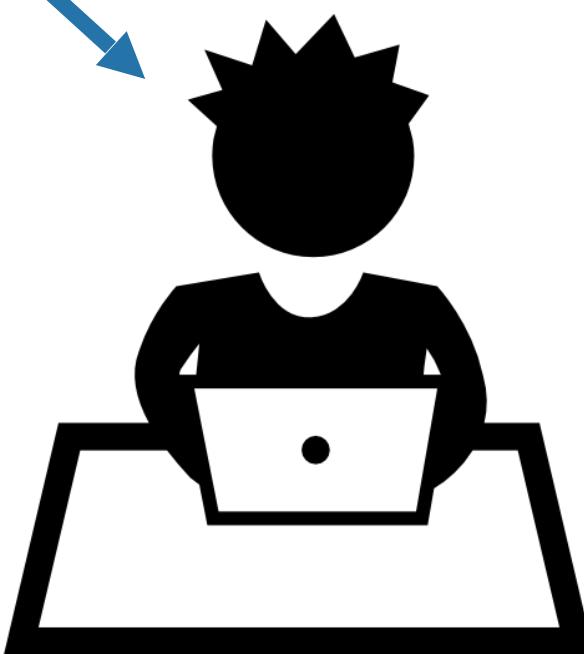
fix typos in car and prof



repo



pull

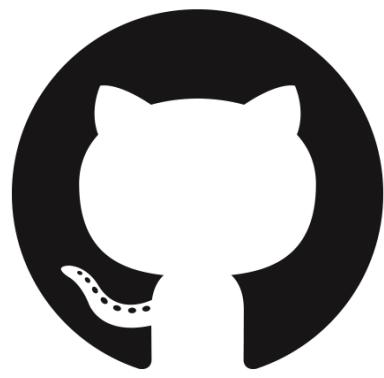


Your teammate pulls
from remote and is
now up-to-date!

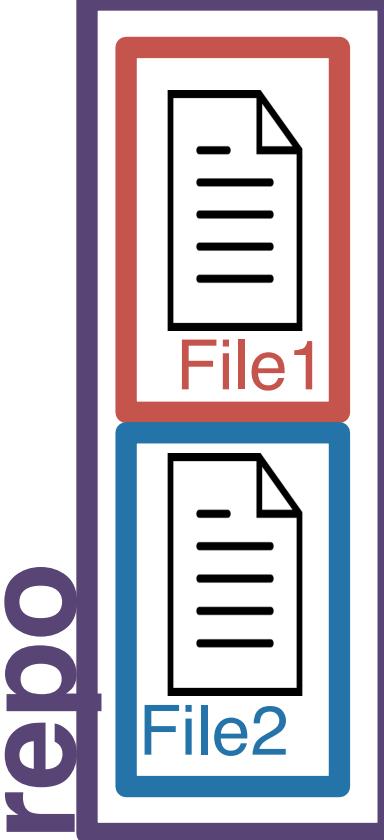


Shannon Ellis
3/28/21 3:28pm

fix typos in car and prof



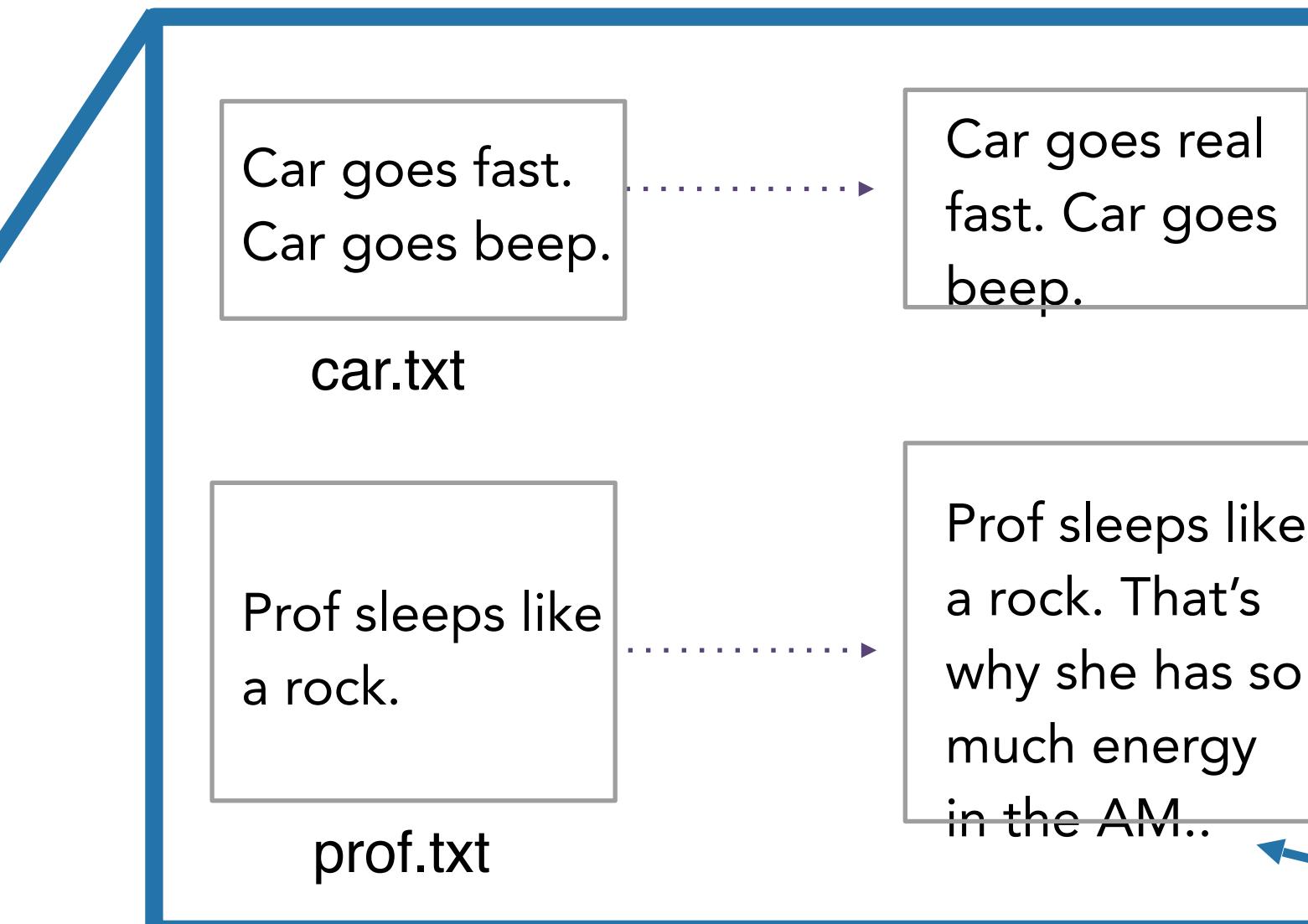
repo



pull



Your teammate pulls
from remote and is
now up-to-date!



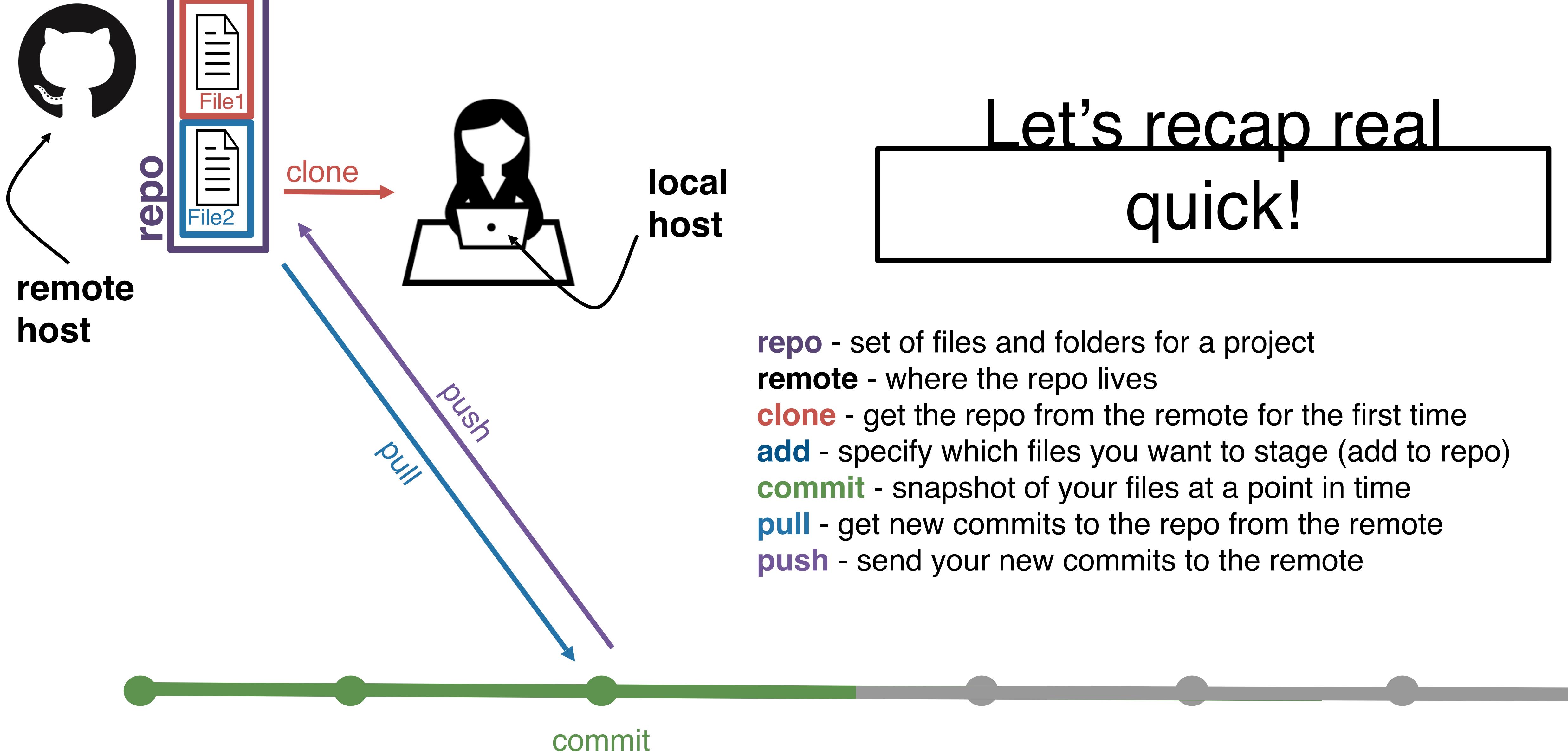
The files in his project
locally will now have
the updated files



Shannon Ellis

3/28/21 3:28pm

fix typos in car and prof

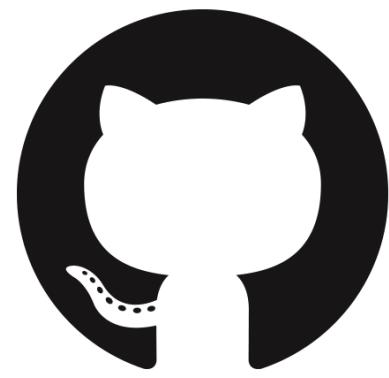


```
(base) sellis:Projects shannonellis$ git status
On branch master
Your branch is up to date with 'origin/master'.

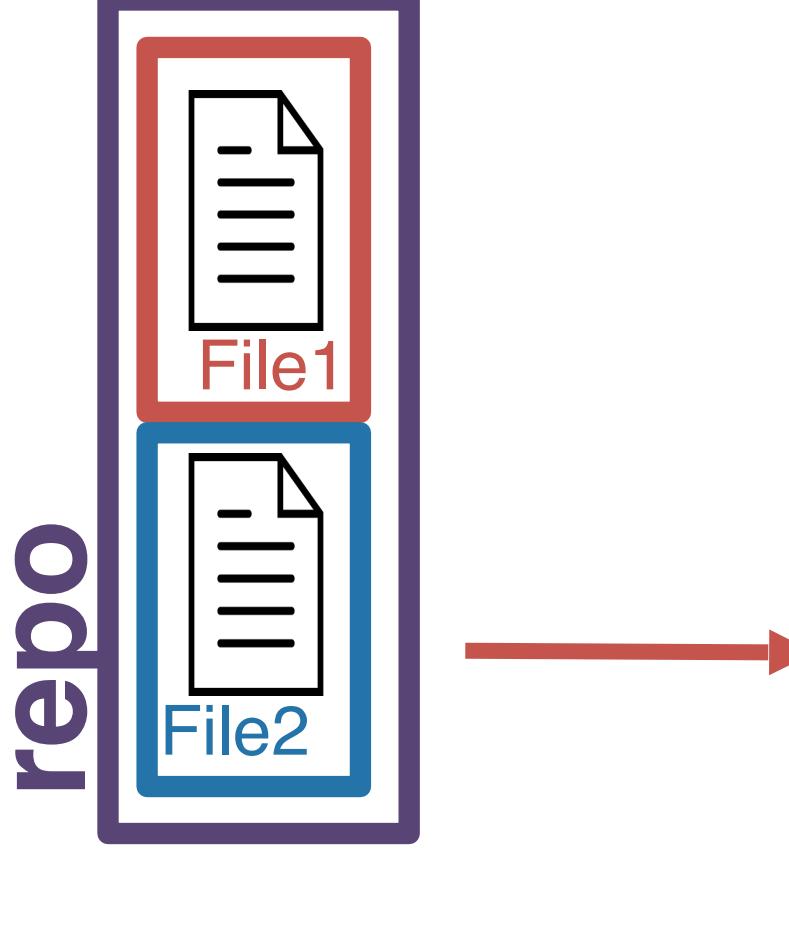
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    FinalProject_Guidelines.pdf

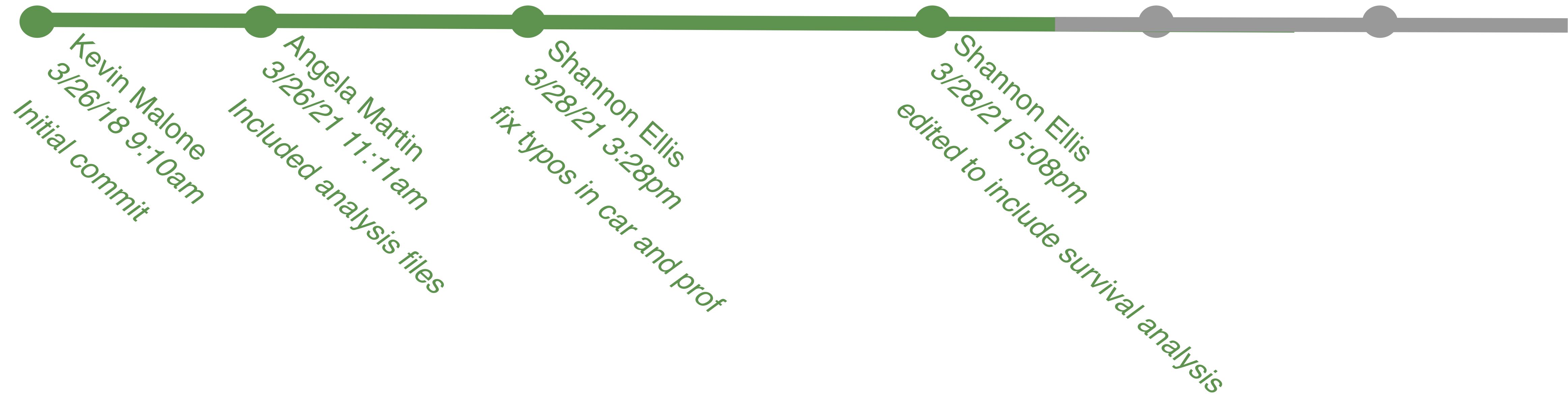
nothing added to commit but untracked files present (use "git add" to track)
(base) sellis:Projects shannonellis$ git add FinalProject_Guidelines.pdf
(base) sellis:Projects shannonellis$ git commit -m "update Project Guidelines"
[master 264e91a] update Project Guidelines
  1 file changed, 0 insertions(+), 0 deletions(-)
   create mode 100644 FinalProject_Guidelines.pdf
(base) sellis:Projects shannonellis$ git push
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 148.21 KiB | 29.64 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/COGS108/Projects.git
  6931768..264e91a  master -> master
```

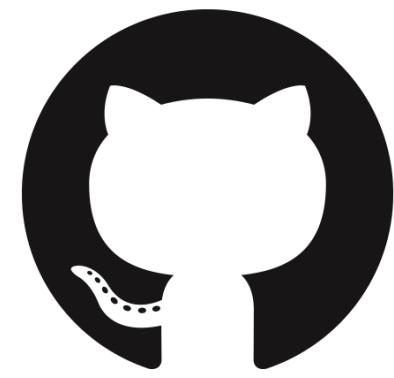


repo



Each time you create a commit, git tracks the changes made automatically.



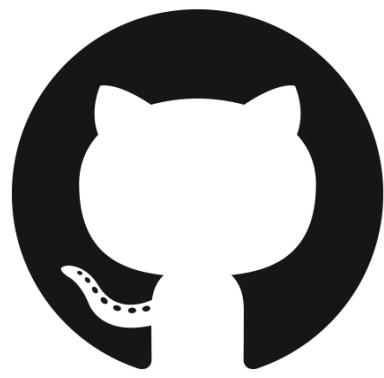


repo



By committing each time you make changes, git allows you to time travel!





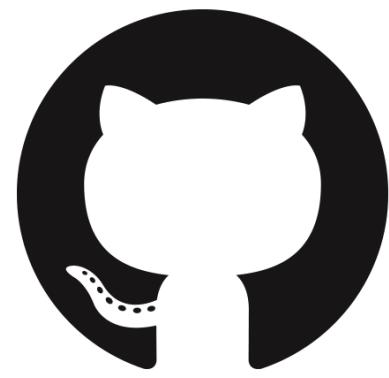
repo



By committing each time you make changes, git allows you to time travel!



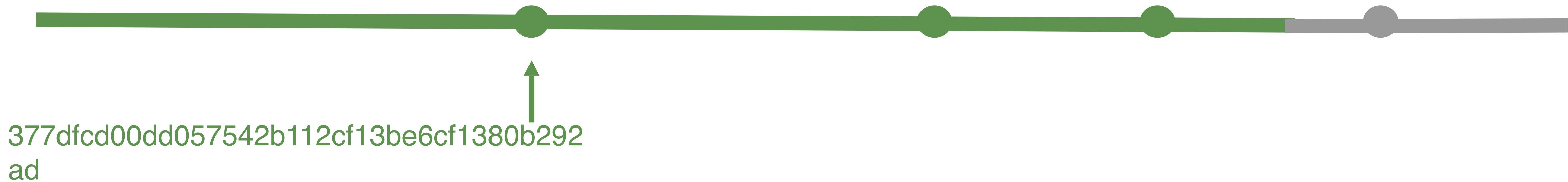
There's a unique id, known as a **hash**, associated with each commit.

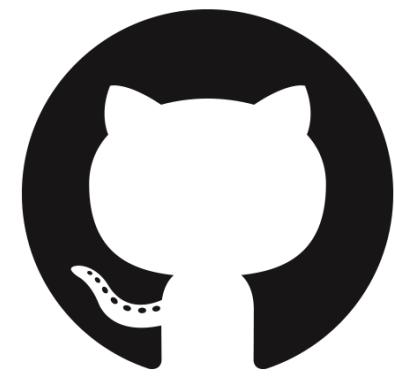


repo

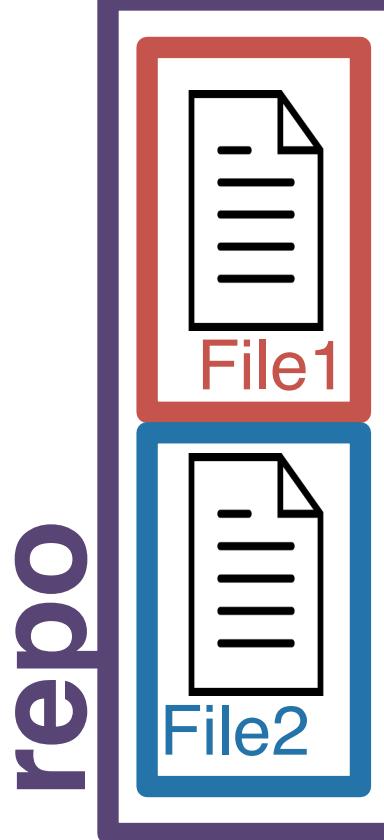


You can return to the state of the repository at any commit. Future commits don't disappear. They just aren't visible when you **check out** an older commit.



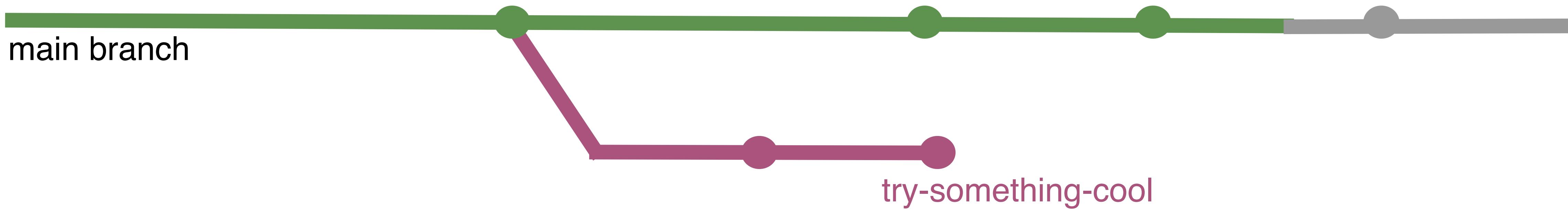


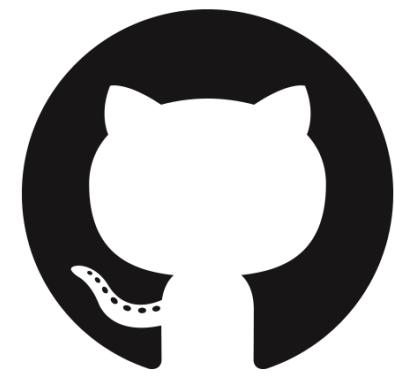
repo



main branch

But...not everything is always linear.
Sometimes you want to try something out
and you're not sure it's going to work. This
is where you'll want to use a **branch**.



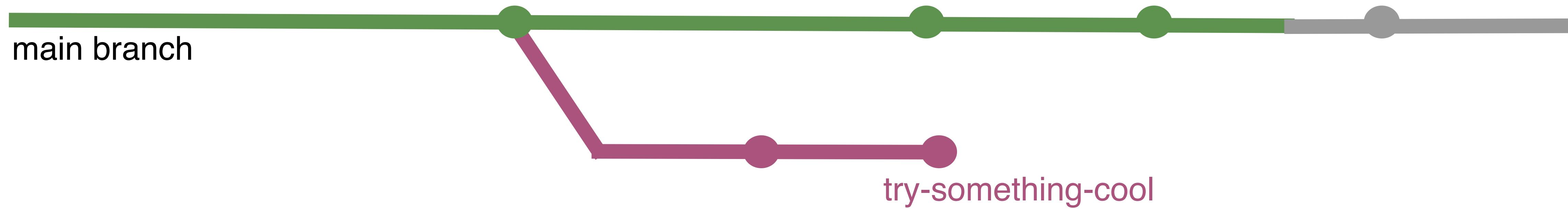


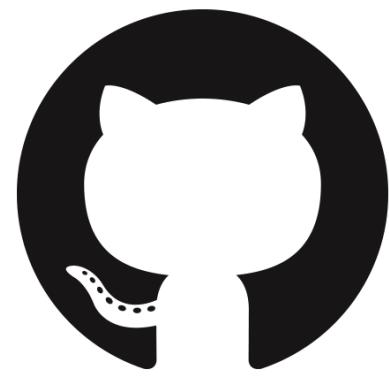
repo



main branch

It's a good way to experiment. It's pretty easy to get rid of a branch later on should you not want to include the commits on that branch.





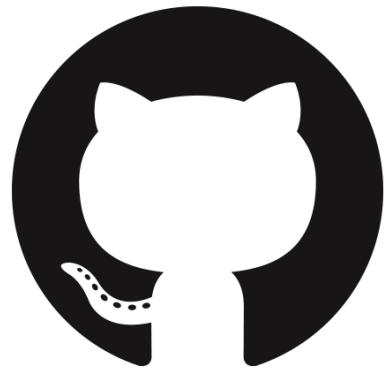
repo



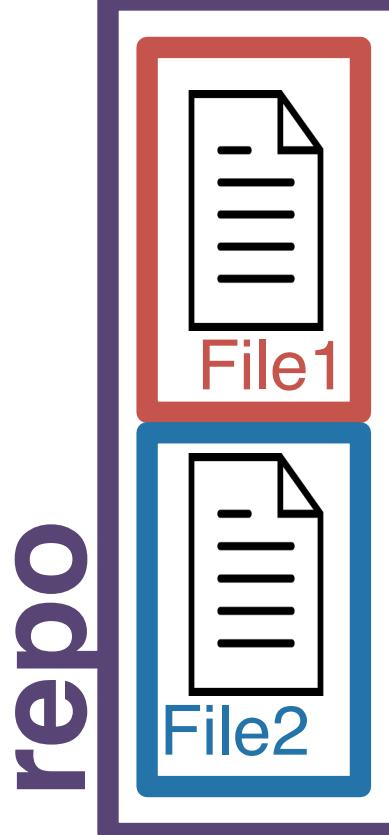
main

But...what if you DO want to include the changes you've made on your try-something-cool branch into the main branch?

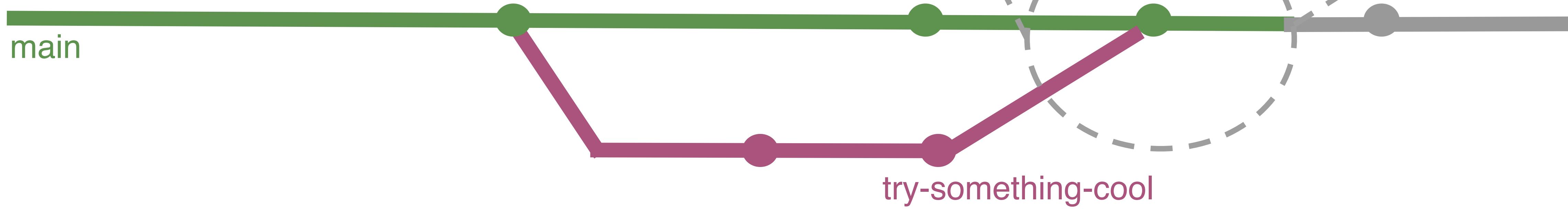
try-something-cool



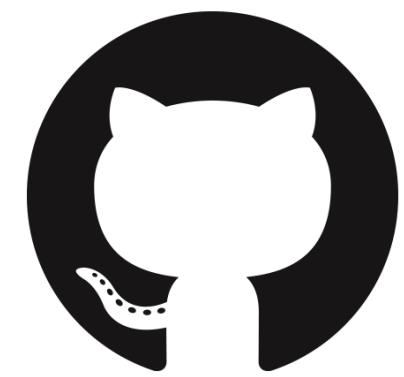
repo



main



A merge allows you to combine the commits from a branch back into the main.

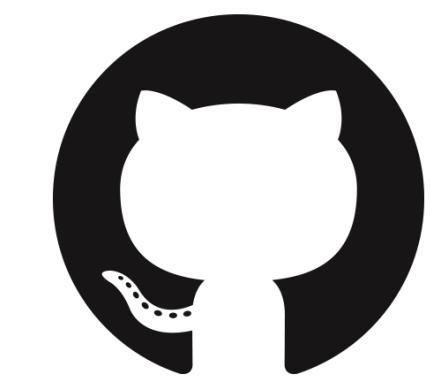


someone
else's
repo

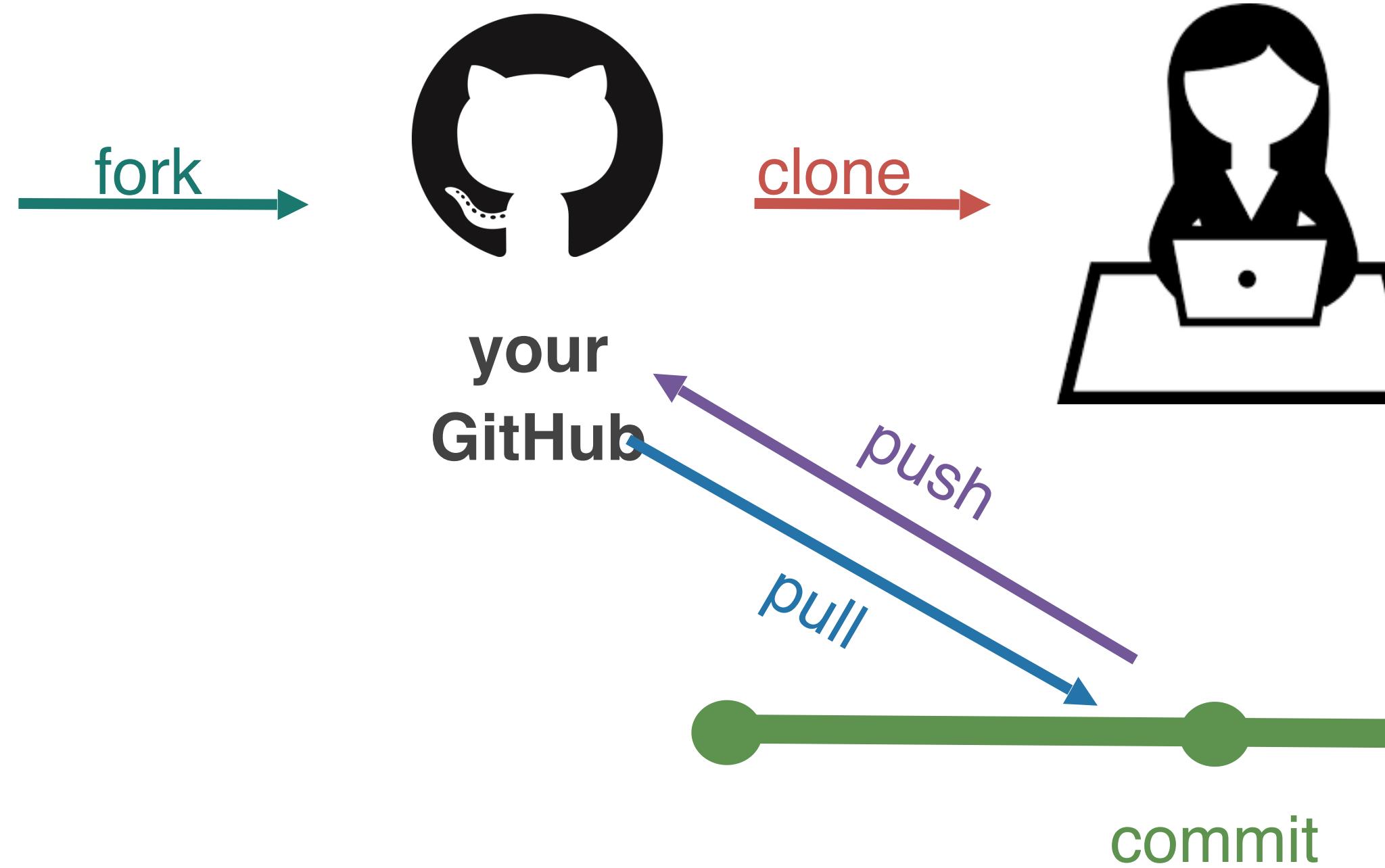


your
GitHub

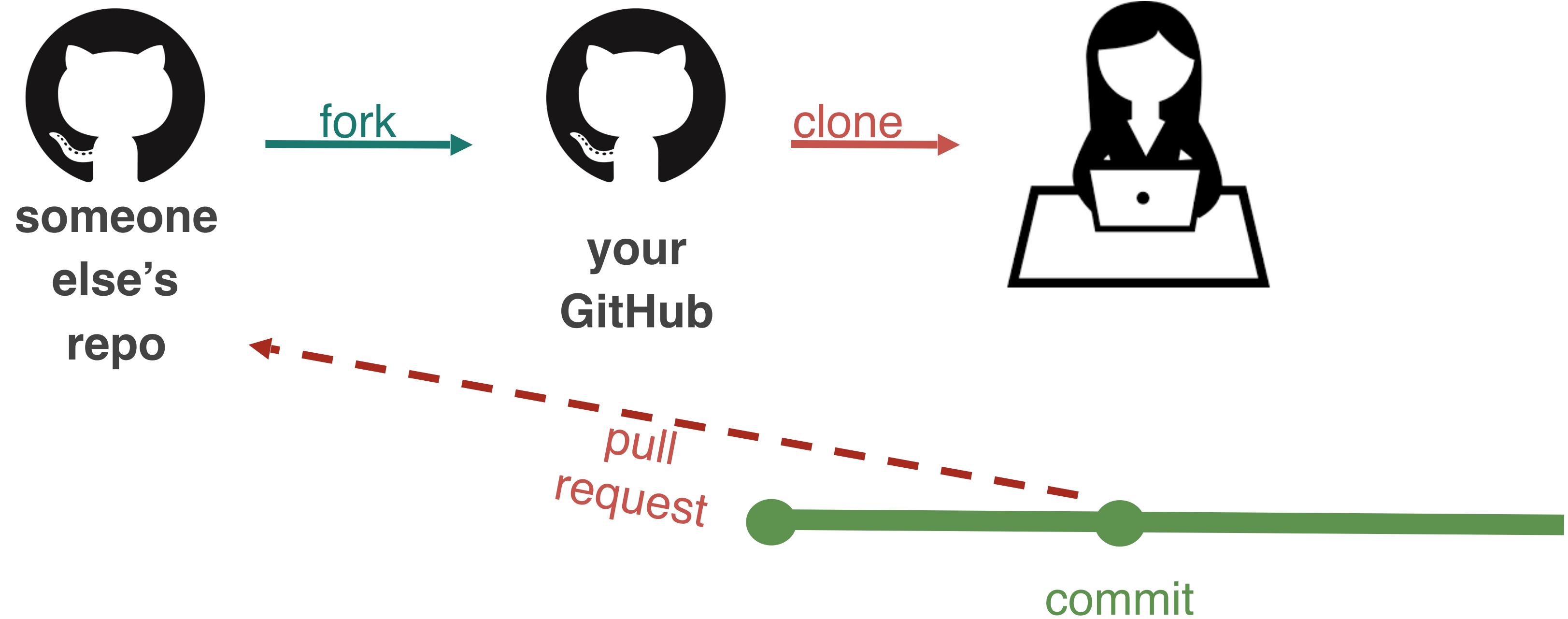
What if someone else is working on something cool and you want to play around with it? You'll have to **fork** their repo.



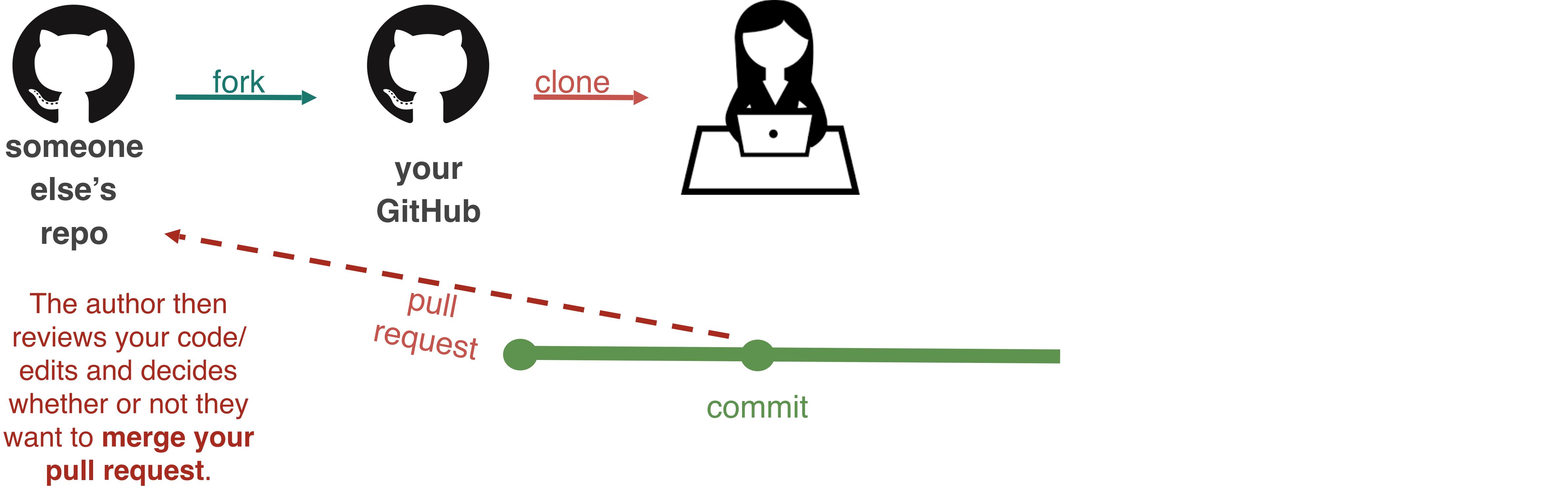
someone
else's
repo



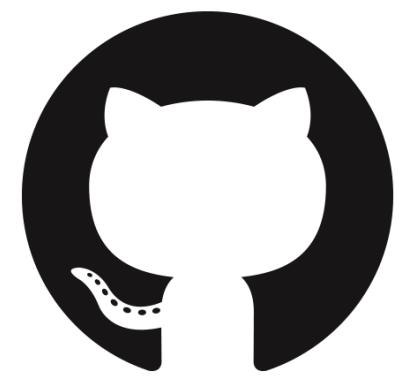
After you fork their repo, you can play around with it however you want, using the workflow we've already discussed.



But what if you think you've found a bug in their code, a typo, or want to add a new feature to their software? For this, you'll submit a **pull request** (aka **PR**).

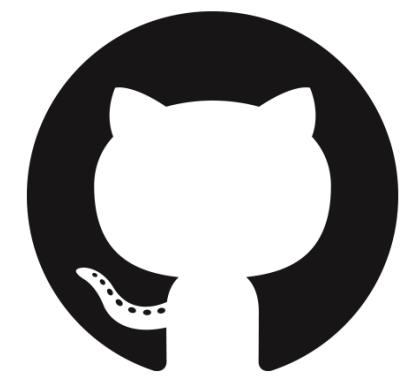


But what if you think you've found a bug in their code, a typo, or want to add a new feature to their software? For this, you'll submit a **pull request** (aka **PR**).



someone
else's
repo

Last but not least...what if you find a bug in someone else's code OR you want to make a suggestion but aren't going to submit a suggestion with a PR. For this, you can file an **issue** on GitHub.



someone
else's
repo

Last but not least...what if you find a bug in someone else's code OR you want to make a suggestion but aren't going to submit a suggestion with a PR. For this, you can file an **issue** on GitHub.

Issues are *bug trackers*. While, they can include bugs, they can also include feature requests, to-dos, whatever you want, really!

They can be assigned to people.

They can be closed once addressedor if the software maintainer doesn't like the suggestion



377dfcd00dd057542b112cf13be6cf1380b292
ad

commits allow you to time travel
because each commit is assigned
a unique **hash**

One more git recap...

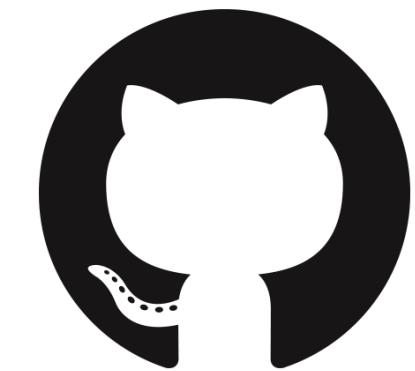


One more git recap...

377dfcd00dd057542b112cf13be6cf1380b292
ad

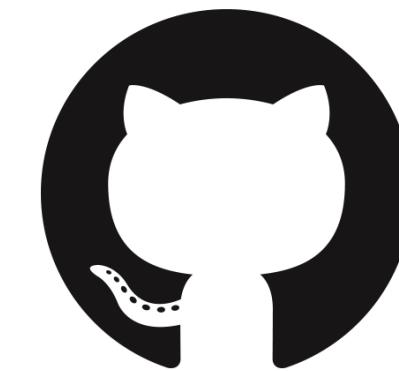
commits allow you to time travel
because each commit is assigned
a unique **hash**

main branch



someone
else's
repo

fork



your
GitHub

You can work on others'
repos by first **forking** their
repository onto your GitHub

try-something-cool

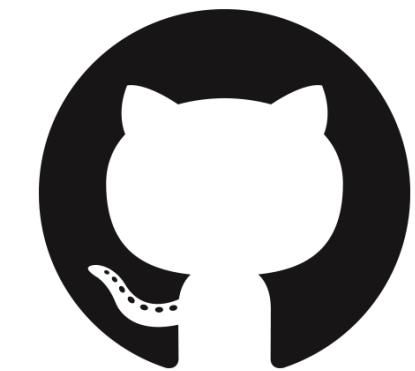
branches allow you to
experiment. branches can be
abandoned or **merged**

One more git recap...

377dfcd00dd057542b112cf13be6cf1380b292
ad

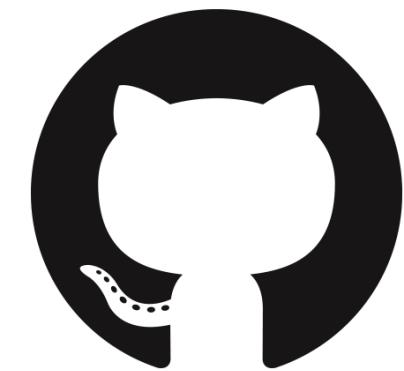
commits allow you to time travel
because each commit is assigned
a unique **hash**

main branch



someone
else's
repo

fork



your
GitHub

You can work on others'
repos by first **forking** their
repository onto your GitHub

try-something-cool

branches allow you to
experiment. branches can be
abandoned or **merged**

Pull requests allow you to make
specific edits to others' repos

Issues allow you to make general
suggestions to your/others' repos

One more git recap...

On to today . . .

Data structures (Types, Tidy Data,
Data Intuition), Data Cleaning

Neural data and structures

- Neural data science generates and processes large amounts of data
- Data must be stored in some organized way for analysis - “Structure”
 - There are three classes of data storage we will discuss - *structured, semi-structured, unstructured*

Data Structures Review

Structured data

- Can be stored in database SQL
- Tables with rows and columns
- Requires a relational key
- 5-10% of all data

Semi-structured data

- Doesn't reside in a relational database
- Has organizational properties (easier to analyze)
- CSV, XML, JSON

Unstructured

- Non-tabular data
- 80% of the world's data
- Images, text, audio, videos

Question

- Why do we do this? What do you think?
- Could we perform neural data science without understanding data structure or giving it any thought?

(Semi-)Structured Data

Data that is stored in such a way that it is easy to search and work with. These data are stored in a particular format that adheres to organization principles imposed by the file format. These are the data structures data scientists work with most often.

CSV files

Each column
separated by a
comma

Has the
extension “.csv”

Example CSV - Sheet1 — Notatnik				
Plik	Edycja	Format	Widok	Pomoc
Email	First Name	Last Name	Company	Snippet 1
example1@domain.com	John	Smith	Company 1	Snippet Sentence1
example2@gmail.com	Mary	Blake	Company 2	Snippet Sentence 2
example3@outlook.com	James	Joyce	Company 3	Snippet Sentence 3

Each row is
separated by a
new line

CSV



Example CSV



File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive



fx

	A	B	C	D	E	F
1	Email	First Name	Last Name	Company	Snippet 1	
2	example1@domain.com	John	Smith	Company 1	Snippet Sentence1	
3	example2@gmail.com	Mary	Blake	Company 2	Snippet Sentence 2	
4	example3@outlook.com	James	Joyce	Company 3	Snippet Sentence 3	

CSV file

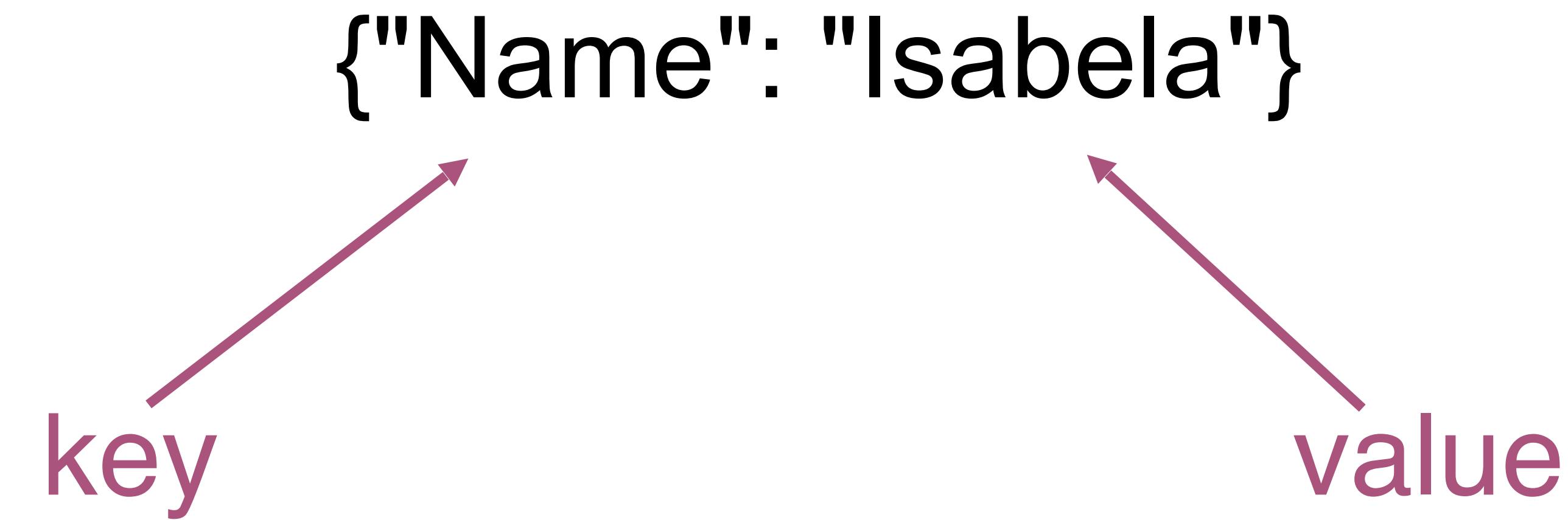


Example CSV - Sheet1 — Notatnik
Plik Edycja Format Widok Pomoc
Email,First Name,Last Name,Company,Snippet 1
example1@domain.com,John,Smith,Company 1,Snippet Sentence1
example2@gmail.com,Mary,Blake,Company 2,Snippet Sentence 2
example3@outlook.com,James,Joyce,Company 3,Snippet Sentence 3

CSV

JSON: key-value pairs

nested/hierarchical data



JSON

JSON

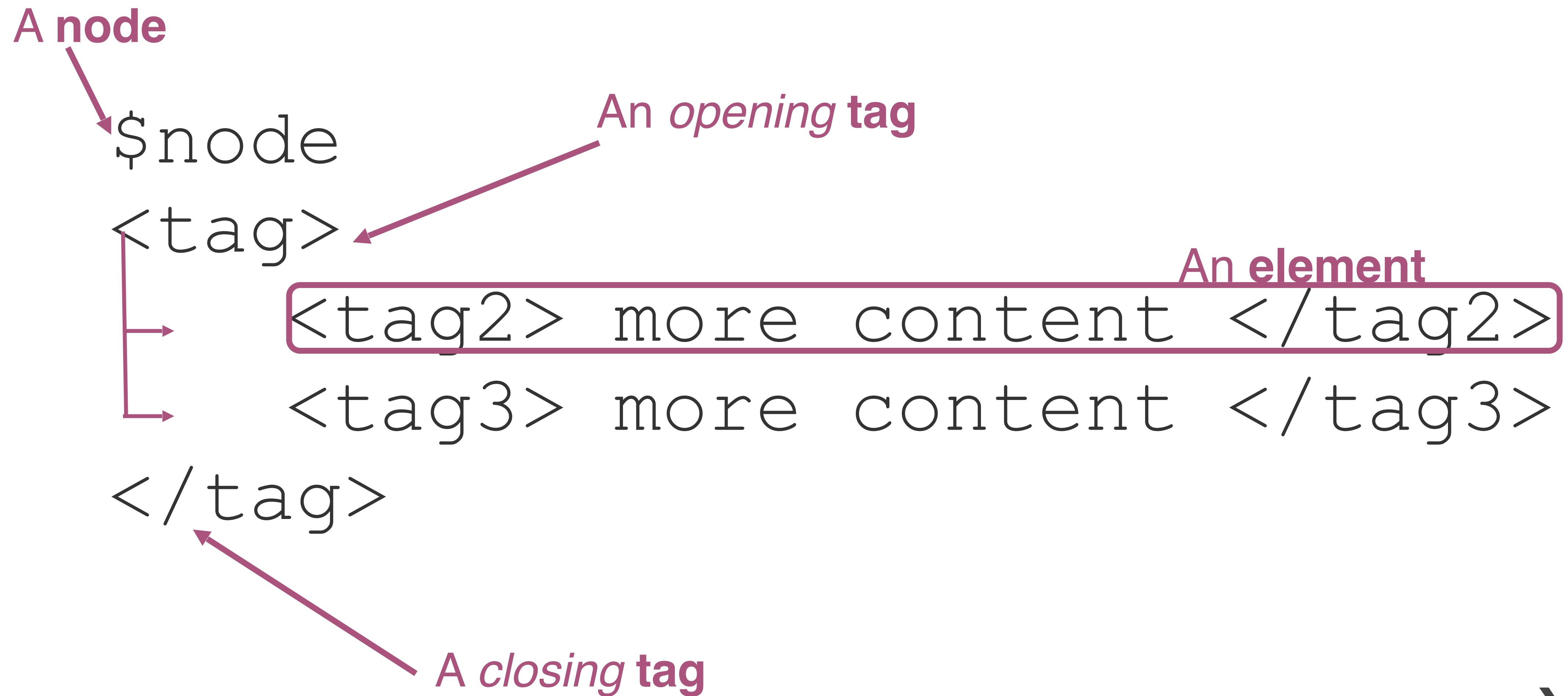
These are all nested within attributes

```
"attributes": {  
    "Take-out": true,  
    "Wi-Fi": "free",  
    "Drive-Thru": true,  
    "Good For": {  
        "dessert": false,  
        "latenight": false,  
        "lunch": false,  
        "dinner": false,  
        "breakfast": false,  
        "brunch": false  
    },
```

These are all nested within "Good For"

Extensible Markup Language (XML): nodes, tags, and elements

nested/hierarchical data



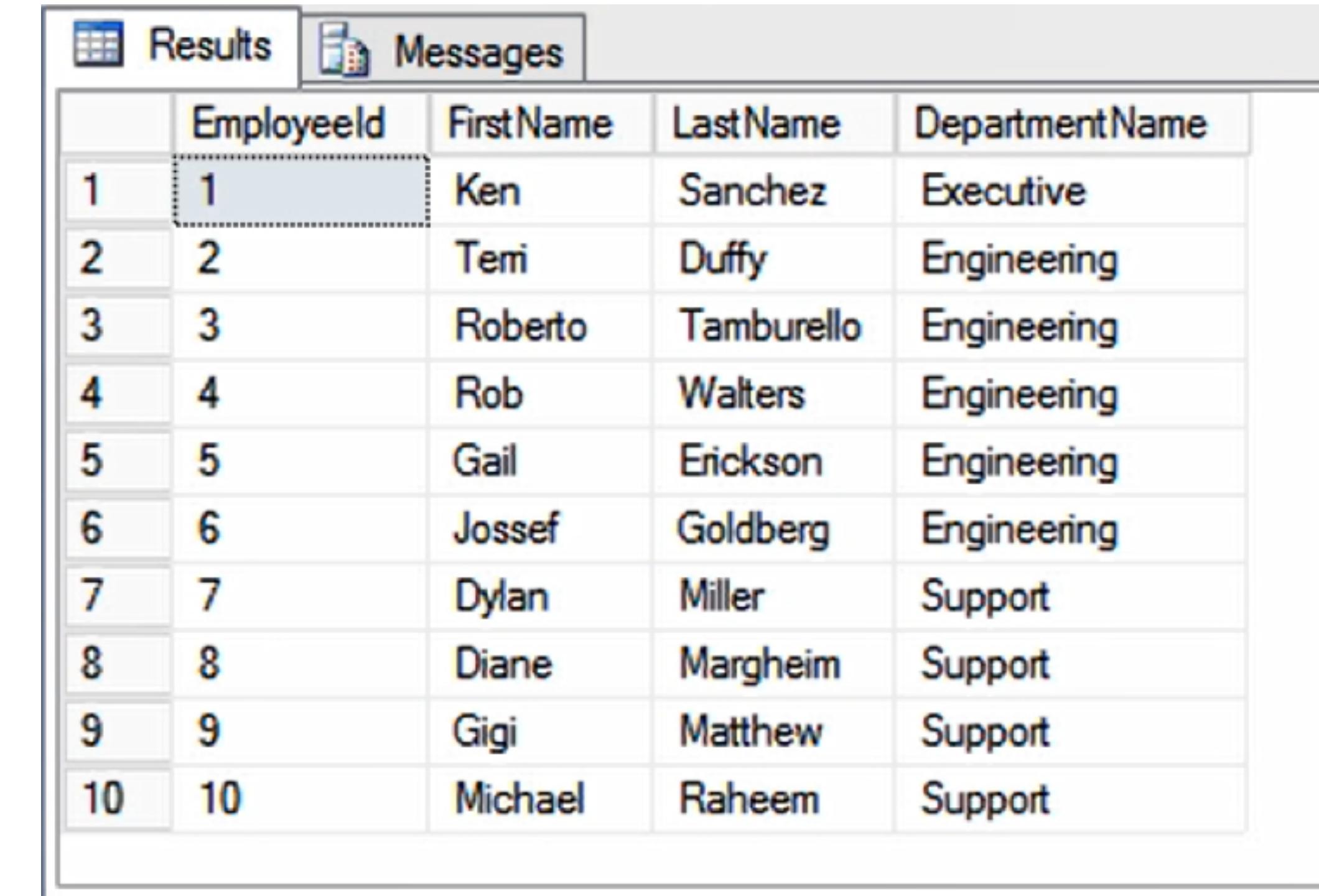
```
<?xml version="1.0" encoding="UTF-8"?>
<customers>
    <customer>
        <customer_id>1</customer_id>
        <first_name>John</first_name>
        <last_name>Doe</last_name>
        <email>john.doe@example.com</email>
    </customer>
    <customer>
        <customer_id>2</customer_id>
        <first_name>Sam</first_name>
        <last_name>Smith</last_name>
        <email>sam.smith@example.com</email>
    </customer>
    <customer>
        <customer_id>3</customer_id>
        <first_name>Jane</first_name>
        <last_name>Doe</last_name>
        <email>jane.doe@example.com</email>
    </customer>
</customers>
```

XML

adapted from Chris Keown

Relational Databases: A set of interdependent tables

1. Efficient Data Storage
2. Avoid Ambiguity
3. Increase Data Privacy



The screenshot shows a Microsoft SQL Server Management Studio (SSMS) interface with the 'Results' tab selected. The results grid displays a table with four columns: EmployeeID, FirstName, LastName, and DepartmentName. The data consists of ten rows, each representing an employee. The first row, where EmployeeID is 1, has a dashed border around it, indicating it is the current row selected or highlighted.

	EmployeeID	FirstName	LastName	DepartmentName
1	1	Ken	Sanchez	Executive
2	2	Temi	Duffy	Engineering
3	3	Roberto	Tamburello	Engineering
4	4	Rob	Walters	Engineering
5	5	Gail	Erickson	Engineering
6	6	Jossef	Goldberg	Engineering
7	7	Dylan	Miller	Support
8	8	Diane	Margheim	Support
9	9	Gigi	Matthew	Support
10	10	Michael	Raheem	Support

relational
database

Information is stored across tables

unique_identifier
AH13JK
JJ29JJ
CI21AA

unique_identifier
AH13JK
JJ29JJ
JJ29JJ
XJ11AS
CI21AA

unique_identifier
AH13JK
SE92FE
CI21AA

entries are *related* to one another by their unique identifier

relational database

restaurant

name	id	address	type
Taco Stand	AH13JK	1 Main St.	Mexican
Pho Place	JJ29JJ	192 Street Rd.	Vietnamese
Taco Stand	XJ11AS	18 W. East St.	Fusion
Pizza Heaven	CI21AA	711 K Ave.	Italian

health inspections

id	inspection_date	inspector	score
AH13JK	2018-08-21	Sheila	97
JJ29JJ	2018-03-12	D'eonte	98
JJ29JJ	2018-01-02	Monica	66
XJ11AS	2018-12-16	Mark	43
CI21AA	2018-08-21	Anh	99

rating

id	stars
AH13JK	4.9
JJ29JJ	4.8
XJ11AS	4.2
CI21AA	4.7

relational
database

restaurant

name	id	address	type
Taco Stand	AH13JK	1 Main St.	Mexican
Pho Place	JJ29JJ	192 Street Rd.	Vietnamese
Taco Stand	XJ11AS	18 W. East St.	Fusion
Pizza Heaven	CI21AA	711 K Ave.	Italian

health inspections

id	inspection_date	inspector	score
AH13JK	2018-08-21	Sheila	97
JJ29JJ	2018-03-12	D'eonte	98
JJ29JJ	2018-01-02	Monica	66
XJ11AS	2018-12-16	Mark	43
CI21AA	2018-08-21	Anh	99

rating

id	stars
AH13JK	4.9
JJ29JJ	4.8
XJ11AS	4.2
CI21AA	4.7

Two different restaurants with the same name will have different unique identifiers

relational database

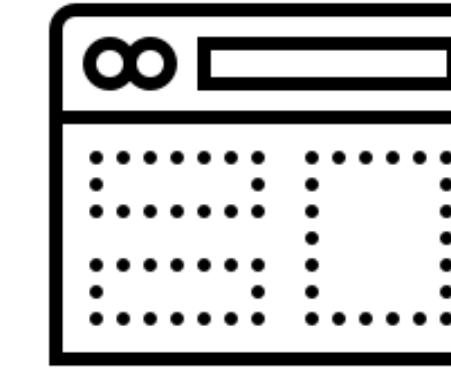
Unstructured Data

Some datasets record information about the state of the world, but in a more heterogeneous way. Perhaps it is a large text corpus with images and links like Wikipedia, or the complicated mix of notes and test results appearing in personal medical records.

Unstructured Data Types



Text files
and
documents



Websites
and
applications



Sensor
data



Image
files



Audio
files



Video
files



Email
data



Social
media
data

Tidy Data

"Good data scientists understand, in a deep way, that the heavy lifting of cleanup and preparation isn't something that gets in the way of solving the problem: it is the problem." - DJ Patil

untidy data

Australian Bureau of Statistics											
1800.0 Australian Marriage Law Postal Survey, 2017											
Released on 15 November 2017											
Table 5 Participation by Federal Electoral Division(a), Males and Age Gender apartheid											
Table junk											
Yeah NA	18-19 years	20-24 years	25-29 years	30-34 years	35-39 years	40-44 years	45-49 years	50-54 years	55-59 years	60-64 years	
Lingia(c)	Total participants	292	1,058	1,465	1,653	1,515	1,516	1,710	1,730	1,753	1,574
	Eligible participants	572	2,910	3,789	3,996	3,607	3,506	3,645	3,331	2,960	2,456
Primary keynotes	Participation rate (%)	51.0	36.4	38.7	41.4	42.0	43.2	46.9	51.9	59.2	64.1
Merged cells	Comma on										
Solomon	Total participants	442	1,461	2,066	2,357	2,188	2,057	2,224	2,108	2,134	1,772
	Eligible participants	750	2,991	3,994	4,155	3,634	3,398	3,427	3,066	2,931	2,355
	Participation rate (%)	58.9	48.8	51.7	56.7	60.2	60.5	64.9	68.8	72.8	75.2
Northern Territory (Total)	Total participants	734	2,519	3,531	4,010	3,703	3,573	3,934	3,838	3,887	3,346
	Eligible participants	1,322	5,901	7,783	8,151	7,241	6,904	7,072	6,397	5,891	4,811
	Participation rate (%)	55.5	42.7	45.4	49.2	51.1	51.8	55.6	60.0	66.0	69.5
Australian Capital Territory Divisions	Covariate as Subheading	Summary of data inside data									
Canberra(d)	Total participants	1,764	4,789	4,817	4,973	4,626	4,453	5,074	4,826	5,169	4,394
	Eligible participants	2,260	6,471	6,448	6,509	5,983	5,805	6,302	5,902	6,044	5,057
	Participation rate (%)	78.1	74.0	74.7	76.4	77.3	76.7	80.5	81.8	85.5	86.9
Fenner(e)	Total participants	1,477	4,687	5,178	5,786	6,025	5,463	5,191	4,208	3,948	3,465
	Eligible participants	1,904	6,354	7,121	7,822	7,960	7,155	6,480	5,206	4,692	3,945
	Participation rate (%)	77.6	73.8	72.7	74.0	75.7	76.4	80.1	80.8	84.1	87.8
Australian Capital Territory (Total)	NA Yeah	3,241	9,476	9,995	10,755	10,051	9,916	10,203	9,034	9,117	7,659
	Eligible participants	4,164	12,825	13,569	14,331	13,943	12,960	12,782	11,108	10,736	9,002
	Participation rate (%)	77.8	73.9	73.7	75.1	76.4	76.5	80.3	81.3	84.9	87.3
Australia											
Total	Total participants	151,297	438,166	441,658	460,548	462,206	479,360	524,620	517,693	543,449	506,799
	Eligible participants	201,439	635,909	646,916	665,250	656,446	660,841	693,850	659,150	664,720	597,386
	Participation rate (%)	75.1	68.9	68.3	69.2	70.4	72.5	75.6	78.5	81.8	84.8
(a) The Federal Electoral Divisions are current as at 24 August 2017	Return of the table junk										
(b) Includes those whose age is unknown											
(c) Includes Christmas Island and the Cocos (Keeling) Islands											
(d) Includes Norfolk Island											
(e) Includes Jervis Bay											
	MS Excel or Die										

tidy data

data → wrangling

area	gender	age	State	Area (sq km)	Eligible participants	Participation rate (%)	Total participants	Total Paticipants
Adelaide	Female	18-19 years	SA	76	1341	83.5	1120	1120
Adelaide	Female	20-24 years	SA	76	4620	81.2	3750	3750
Adelaide	Female	25-29 years	SA	76	4897	81.8	4004	4004
Adelaide	Female	30-34 years	SA	76	4784	79.8	3820	3820
Adelaide	Female	35-39 years	SA	76	4319	79	3411	3411
Adelaide	Female	40-44 years	SA	76	4310	80.6	3472	3472
Adelaide	Female	45-49 years	SA	76	4579	81.4	3728	3728
Adelaide	Female	50-54 years	SA	76	4475	84.7	3791	3791
Adelaide	Female	55-59 years	SA	76	4622	87.3	4033	4033
Adelaide	Female	60-64 years	SA	76	4342	89.3	3879	3879
Adelaide	Female	65-69 years	SA	76	3970	90.7	3602	3602
Adelaide	Female	70-74 years	SA	76	3009	90.3	2716	2716
Adelaide	Female	75-79 years	SA	76	2156	88.5	1908	1908
Adelaide	Female	80-84 years	SA	76	1673	85.1	1423	1423

Tidy Data

1. Each **variable** you measure should be in a single column

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

2. Every **observation** of a variable should be in a different row

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

3. There should be one table for each type of data

Demographic Survey Data

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2		1004	Smith	Jane	female	Frederick	MD
3		4587	Nayef	Mohammed	male	Upper Darby	PA
4		1727	Doe	Janice	female	San Diego	CA
5		6879	Jordan	Alex	male	Birmingham	AL

Doctor's Office Measurements Data

	A	D	E	F	G
1	ID	Height_inches	Weight_lbs	Insulin	Glucose
2	1004	65	180	0.60	163
3	4587	75	215	1.46	150
4	1727	62	124	0.72	177
5	6879	77	160	1.23	205

4. If you have multiple tables, they should include a column in each *with the same column label* that allows them to be joined or merged

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

	A	D	E	F	G
1	ID	Height_inches	Weight_lbs	Insulin	Glucose
2	1004	65	180	0.60	163
3	4587	75	215	1.46	150
4	1727	62	124	0.72	177
5	6879	77	160	1.23	205

Tidy data == rectangular data

A

	A	B	C	D	E
1	id	sex	glucose	insulin	triglyc
2	101	Male	134.1	0.60	273.4
3	102	Female	120.0	1.18	243.6
4	103	Male	124.8	1.23	297.6
5	104	Male	83.1	1.16	142.4
6	105	Male	105.2	0.73	215.7

Tidy Data Benefits

1. Consistent data structure
2. Foster tool development
3. Require only a small set of tools to be learned
4. Allow for datasets to be combined

Data Intuition



[https://www.youtube.com/watch?
v=0YzvupOX8ls](https://www.youtube.com/watch?v=0YzvupOX8ls)

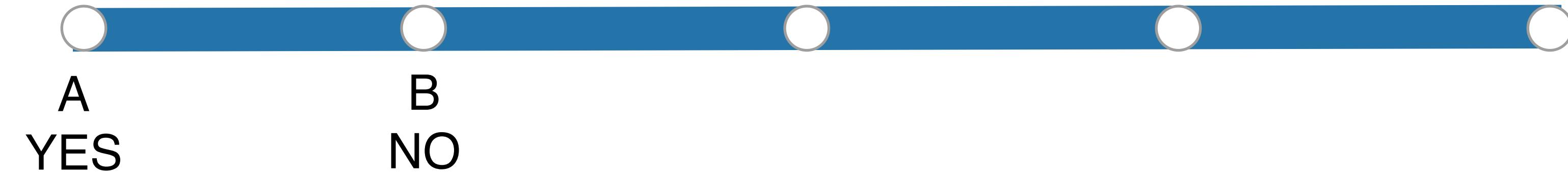
**Has humanity produced enough
paint to cover the entire land area of
the Earth?**

—Josh (Bolton, MA)



Fermi Estimation

Has humanity produced enough paint to cover the entire land area of the Earth?



This answer is pretty straightforward. We can look up the size of the world's paint industry, extrapolate backward to figure out the total amount of paint produced. We'd also need to make some assumptions about how we're painting the ground. Note: When we get to the Sahara desert, I recommend not using a brush.



But first, let's think about different ways we might come up with a guess for what the answer will be. In this kind of thinking—often called **Fermi estimation**—all that matters is getting in the right ballpark; that is, the answer should have about the right number of digits. In Fermi estimation, you can round [1] all your answers to the nearest order of magnitude:



FACTS ABOUT ME

AGE: 10
HEIGHT: 10 FEET
NUMBER OF ARMS: 1
NUMBER OF LEGS: 1
TOTAL NUMBER OF LIMBS: 10
AVERAGE DRIVING SPEED: 100 MPH

Let's suppose that, on average, everyone in the world is responsible for the existence of two rooms, and they're both painted. My living room has about 50 square meters of paintable area, and two of those would be 100 square meters. 7.15 billion people times 100 square meters per person is a little under a trillion square meters –an area smaller than Egypt.

NOT ENOUGH	EXACTLY ENOUGH	MORE THAN ENOUGH
/		

Let's make a wild guess that, on average, one person out of every thousand spends their working life painting things. If I assume it would take me three hours to paint the room I'm in,^[2] and 100 billion people have ever lived, and each of them spent 30 years painting things for 8 hours a day, we come up with 150 trillion square meters ... just about exactly the land area of the Earth.

NOT ENOUGH	EXACTLY ENOUGH	MORE THAN ENOUGH
/	/	

How much paint does it take to paint a house? I'm not enough of an adult to have any idea, so let's take another Fermi guess.

Based on my impressions from walking down the aisles, home improvement stores stock about as many light bulbs as cans of paint. A normal house might have about 20 light bulbs, so let's assume a house needs about 20 gallons of paint.^[3] Sure, that sounds about right.

The average US home costs about \$200,000. Assuming each gallon of paint covers about 300 square feet, that's a square meter of paint per \$300 of real estate. I vaguely remember that the world's real estate has a combined value of something like \$100 trillion,^[4] which suggests there's about 300 billion square meters of paint on the world's real estate. That's about one New Mexico.

NOT ENOUGH	EXACTLY ENOUGH	MORE THAN ENOUGH
//	/	

Of course, both of the building-related guesses could be overestimates (lots of buildings are not painted) or underestimates (lots of things that are not buildings [5] are painted) But from these wild Fermi estimates, my guess would be that there probably isn't enough paint to cover all the land.

So, how did Fermi do?

According to the report [**The State of the Global Coatings Industry**](#), the world produced 34 billion liters of paints and coatings in 2012.

There's a neat trick that can help us here. If some quantity—say, the world economy—has been growing for a while at an annual rate of n —say, 3% (0.03)—then the most recent year's share of the whole total so far is $1 - \frac{1}{1+n}$, and the whole total so far is the most recent year's amount times $1 + \frac{1}{n}$.

If we assume paint production has, in recent decades, followed the economy and grown at about 3% per year, that means the total amount of paint produced equals the current yearly production times 34.^[6] That comes out to a little over a trillion liters of paint. At 30 square meters per gallon,^[7] that's enough to cover 9 trillion square meters—about the area of the United States.

So the answer is no; there's not enough paint to cover the Earth's land, and—at this rate—probably won't be enough until the year 2100.

Data Intuition

1. Think about your question and your expectations
2. Do some Fermi calculations (back of the envelope calculations)
3. Write code & look at outputs <- think about those outputs
4. Use your gut instinct / background knowledge to guide you
5. Review code & fix bugs
6. Create test cases - “Sanity checks”

What is data cleaning?

- Fixing/removing incorrect, corrupted, incorrectly formatted, duplicate, incomplete, data within a dataset
- Many issues combining data sources and types, researcher styles, standards, recording errors, etc

Consequences of poorly cleaned data

- Unreliable outcomes and algorithms
- Difficult to detect these issues
- Biased results
- Failure to process algorithms (for example NaNs causing errors)

Variability in cleaning

- There is no one process to clean data
- Varies from set to set, project to project, software to software
- But can establish a ‘template’ procedure/process of ‘check-offs’ to make sure you’ve done your best to address it

Methods can be

- Interactive through ‘wrangling tools’
- Automated through scripts, programs or other software (batch processing)

