PSY 503: Foundations of Statistical Methods in Psychological Science

Github

Suyog Chandramouli

311 PSH (Princeton University)

10th November, 2025

Version Control

- Practice of tracking and managing changes to files
    - Primarily for code
    - Version control systems (VCS) are software that help with version control

- A complete long-term change history of every file

Why Version Control?

- Organization, documentation, and dissemination is part of the work, not separate from it.
- Helps with asynchronous collaboration
  - With other humans
  - With your future self
- Makes web presence easy
- Version control is not just for code, but also other artifacts
  - Figures, Data, Reports, Code..
- Reproducibility

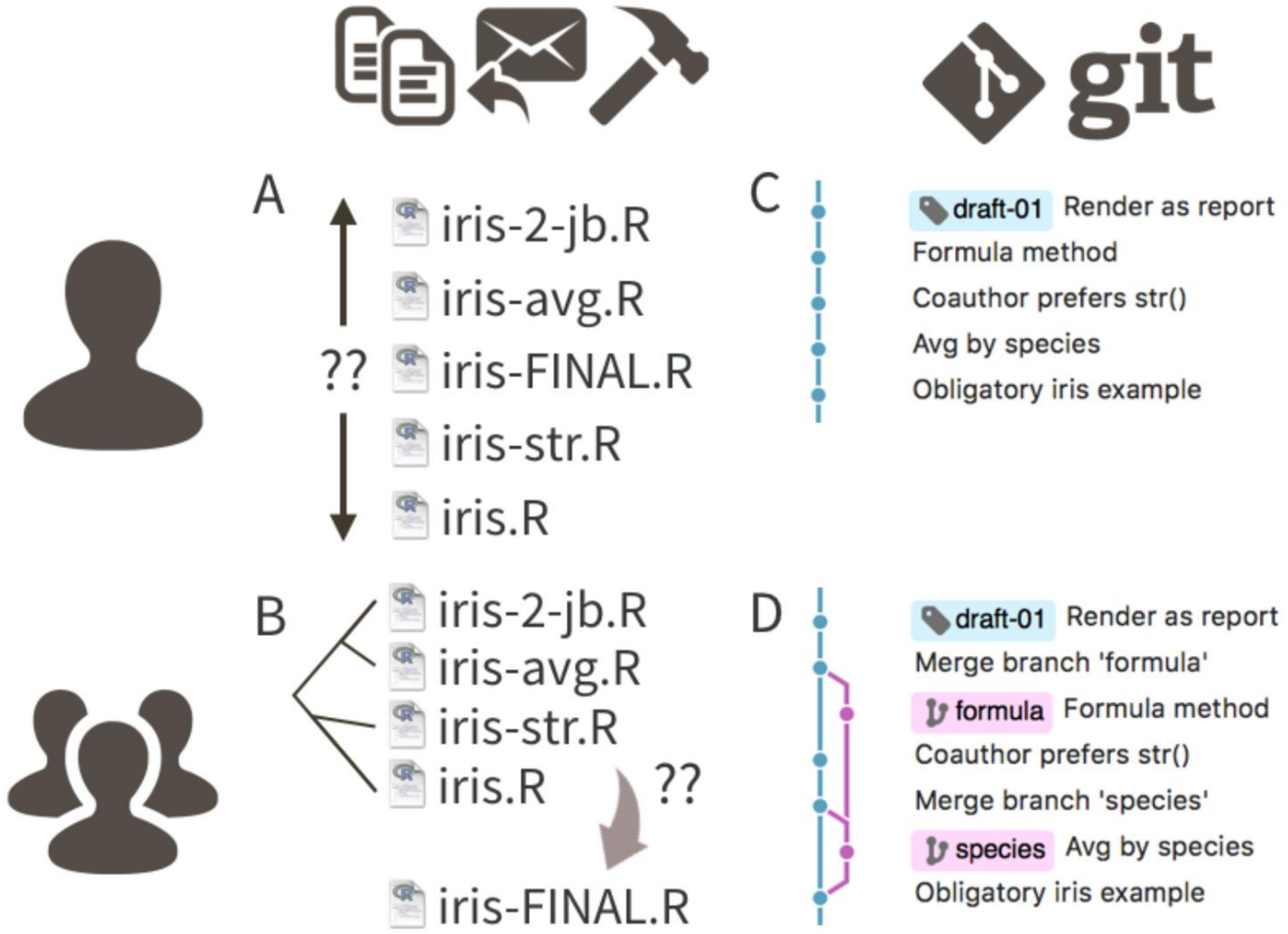
Some version control systems

- Github (based on Git)
- GitLab
- Bitbucket
- Apache Subversion (SVN)
- …

Git vs Github: What is git?

- Git is a version control system (VCS)
- Tracks and manages the evolution of a set of files (in a “repository”)
- Original intent
  - So that developers collaborate on large software projects
- Like google docs
  - But more functionality
    - More features
    - Or filetypes

Workflows


A
iris-2-jb.R
iris-avg.R
??
iris-FINAL.R
iris-str.R
iris.R
B
iris-2-jb.R
iris-avg.R
iris-str.R
iris.R
??
iris-FINAL.R
git
C
draft-01 Render as report
Formula method
Coauthor prefers str()
Avg by species
Obligatory iris example
D
draft-01 Render as report
Merge branch 'formula'
formula Formula method
Coauthor prefers str()
Merge branch 'species'
species Avg by species
Obligatory iris example

What is Github?

- Complementary to git
- Git is local, Github is a remote hosting service
- Github has a web interface
  - Edit
  - Add
  - Hyperlink to specific locations in code

Solo work

- Advantages to keeping a synced copy on Github
  - Backup & Recovery
    - “Clone” a repository that you had “pushed”
    - a safety-net
  - Safer to experiment with code
  - Organized structure
- Web Interface
  - Visualize change over time
  - Navigation
  - Web presence

Collaborative work

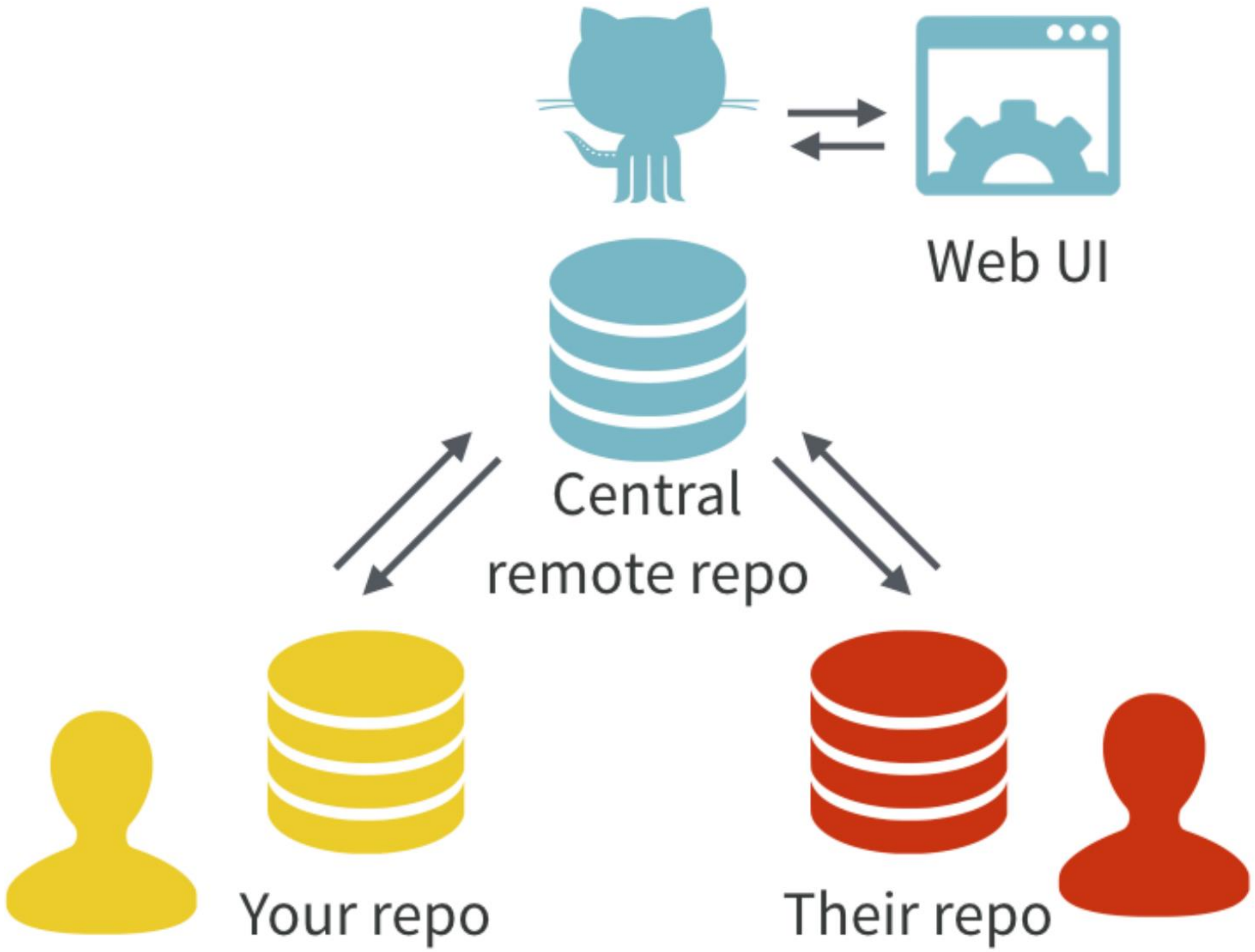
- Different permission levels
- Issues
  - Code review
  - Bug reporting
  - Simultaneous work
    - Merge

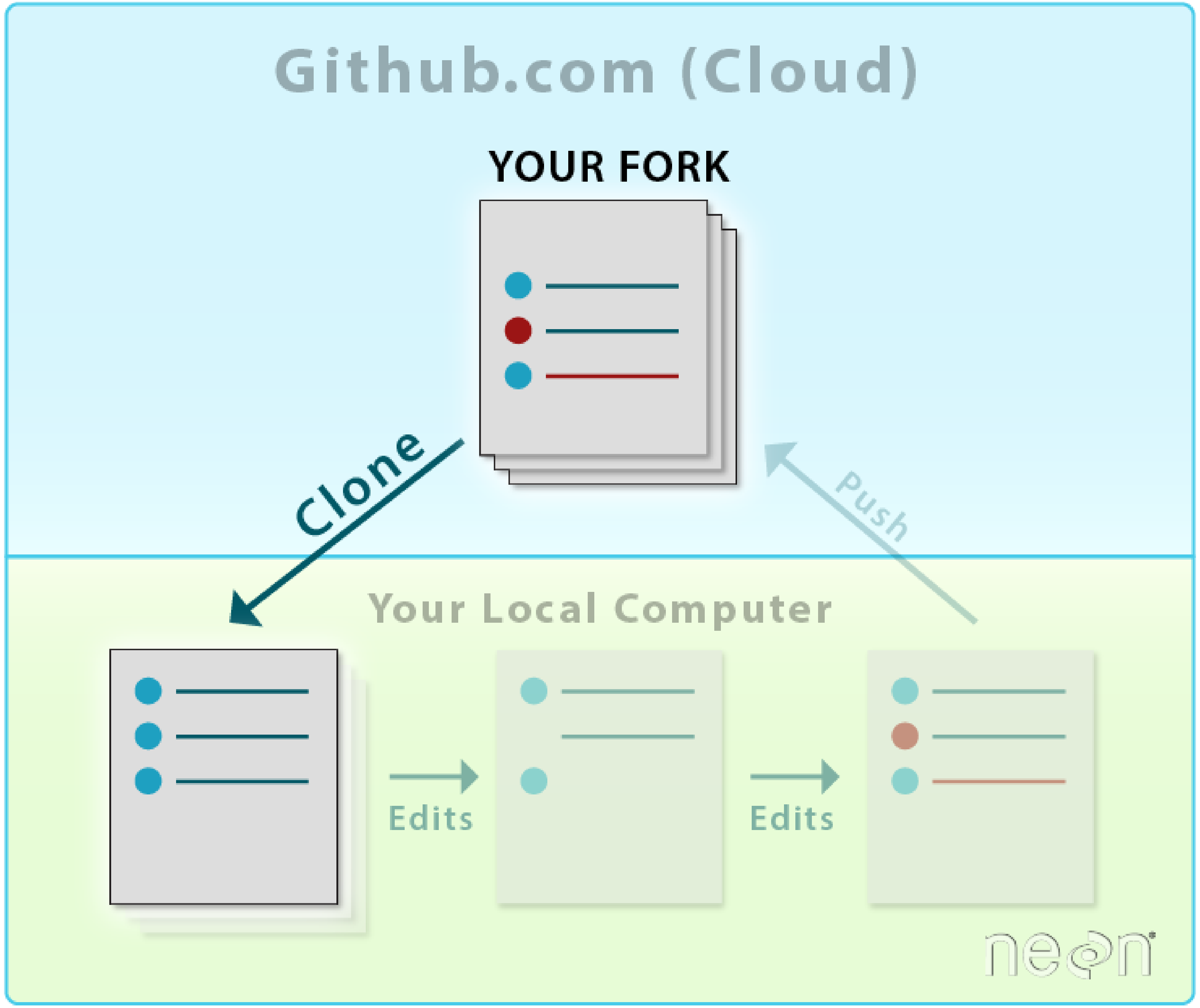
Repositories

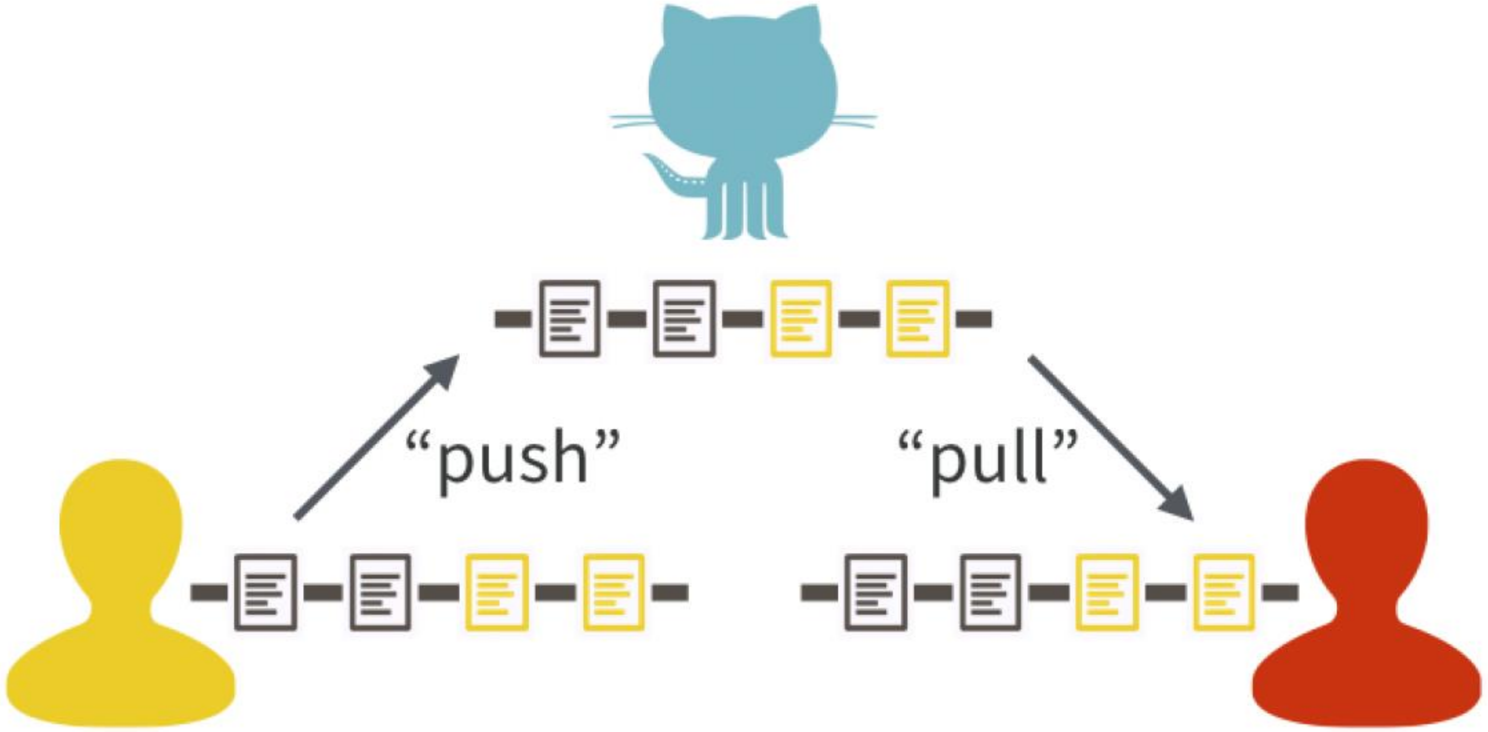
- A repository (repo) is a storage location that contains your project files
  - Its history is tracked
  - Local repository is on your computer
  - Remote repository is on GitHub’s servers

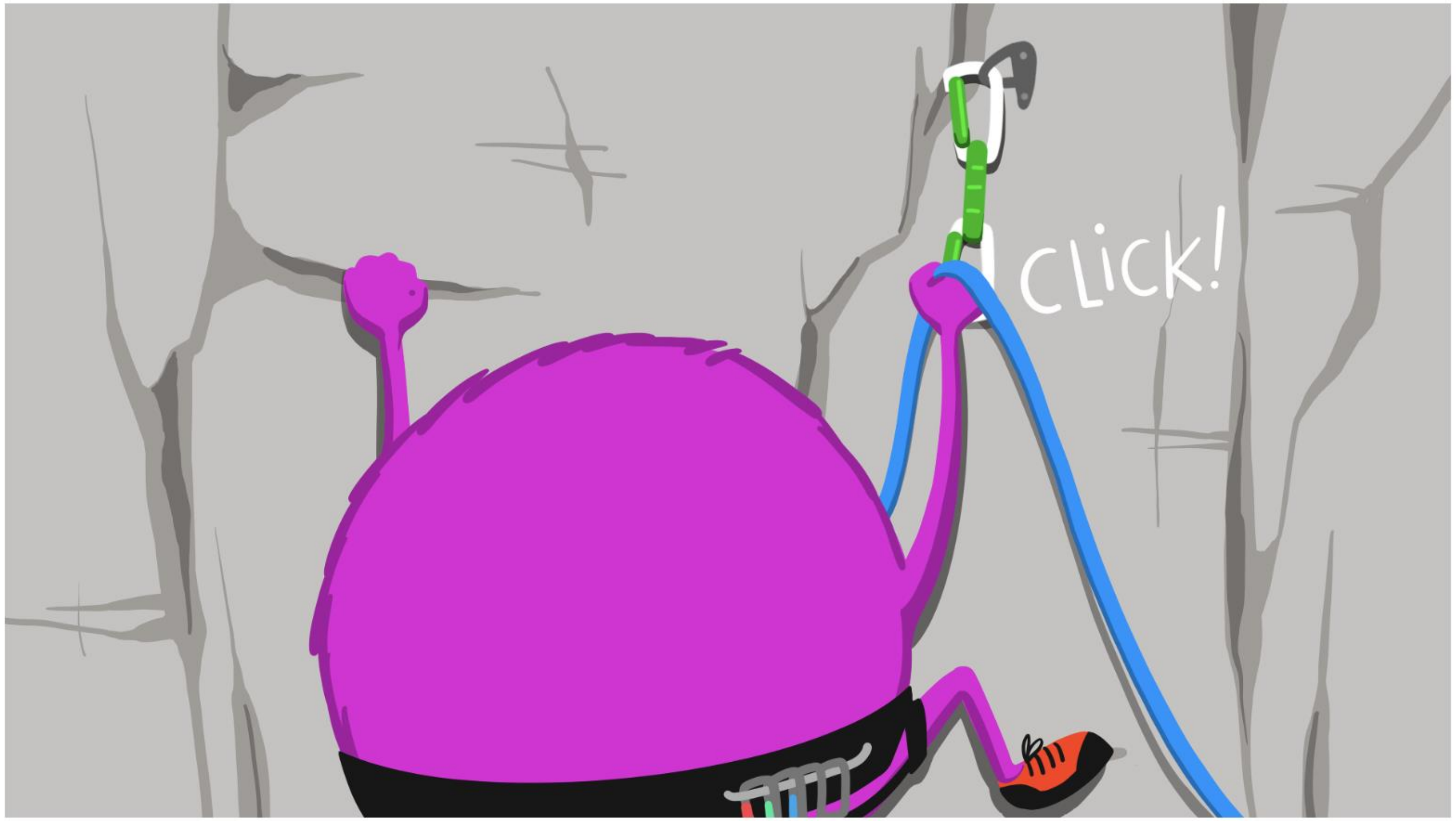
- Public vs private

Github

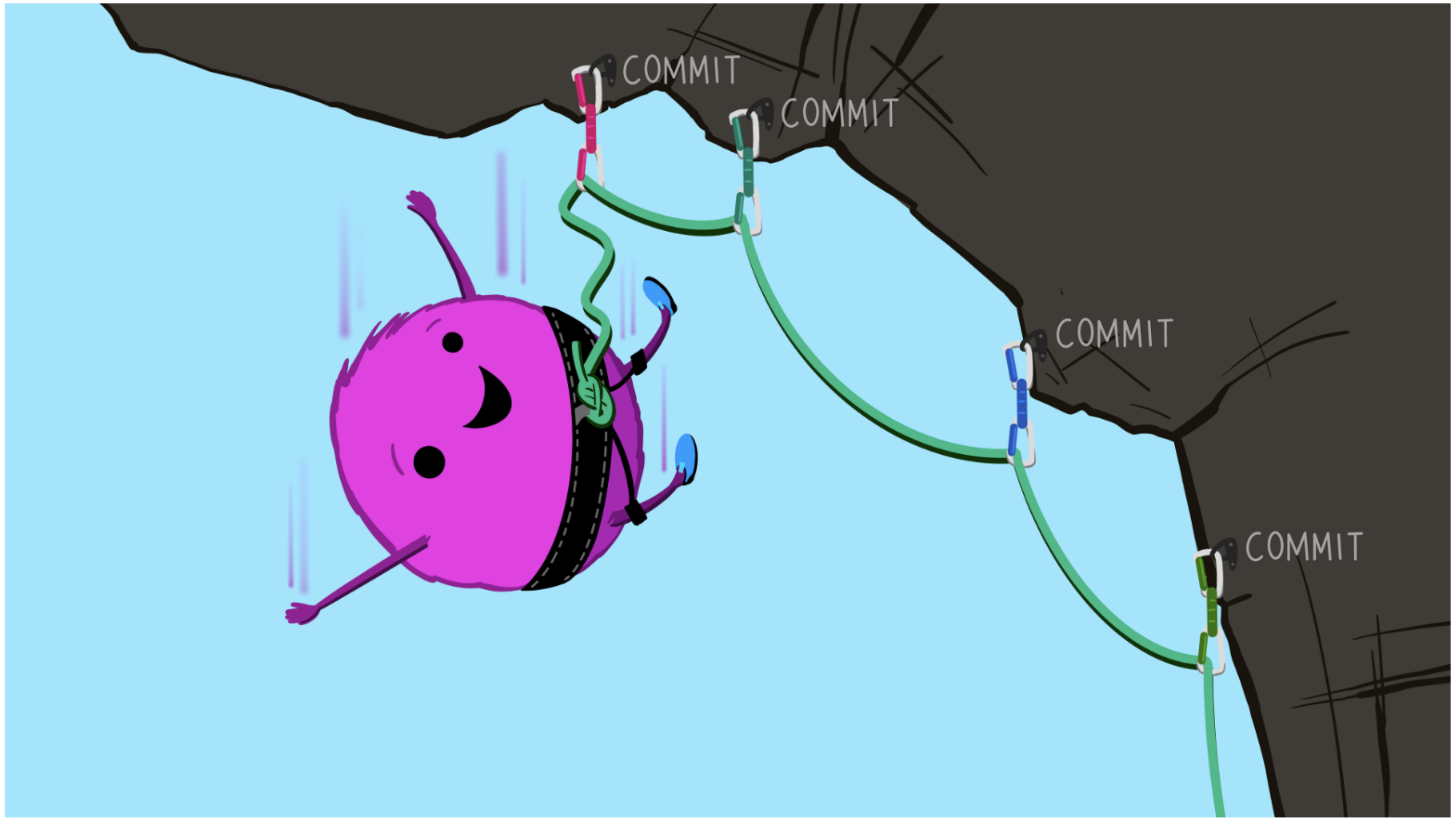



Github.com (Cloud)
YOUR FORK
Clone
Push
Your Local Computer
Edits
Edits
neon


"push"
"pull"

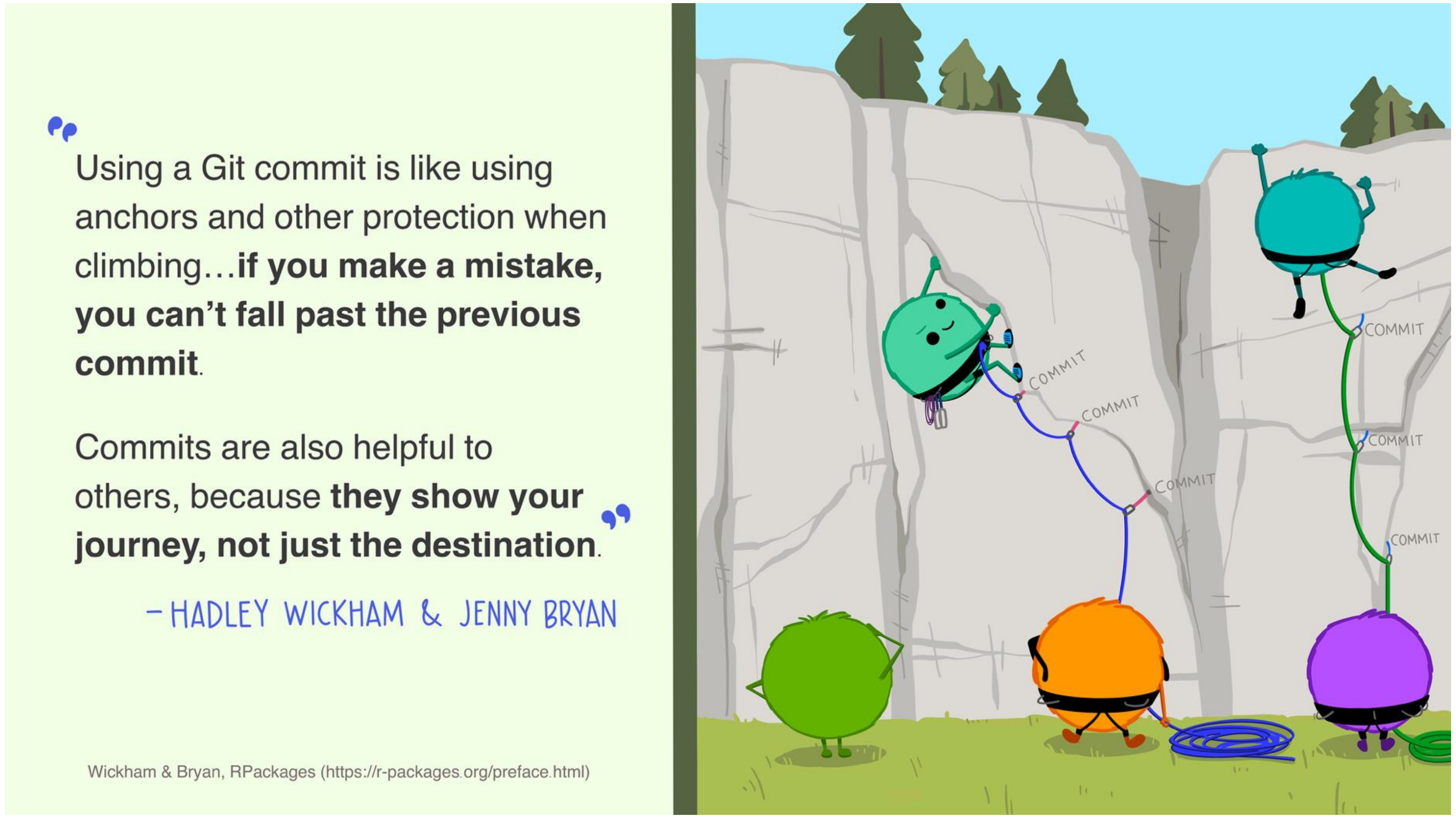


A close-up rear view of a monster climbing a rock face, clicking into an anchor point, with the word “Click!”



A confidently smiling monster is falling from a rock overhang, while secured by four anchors, each labeled “Commit”.

commit

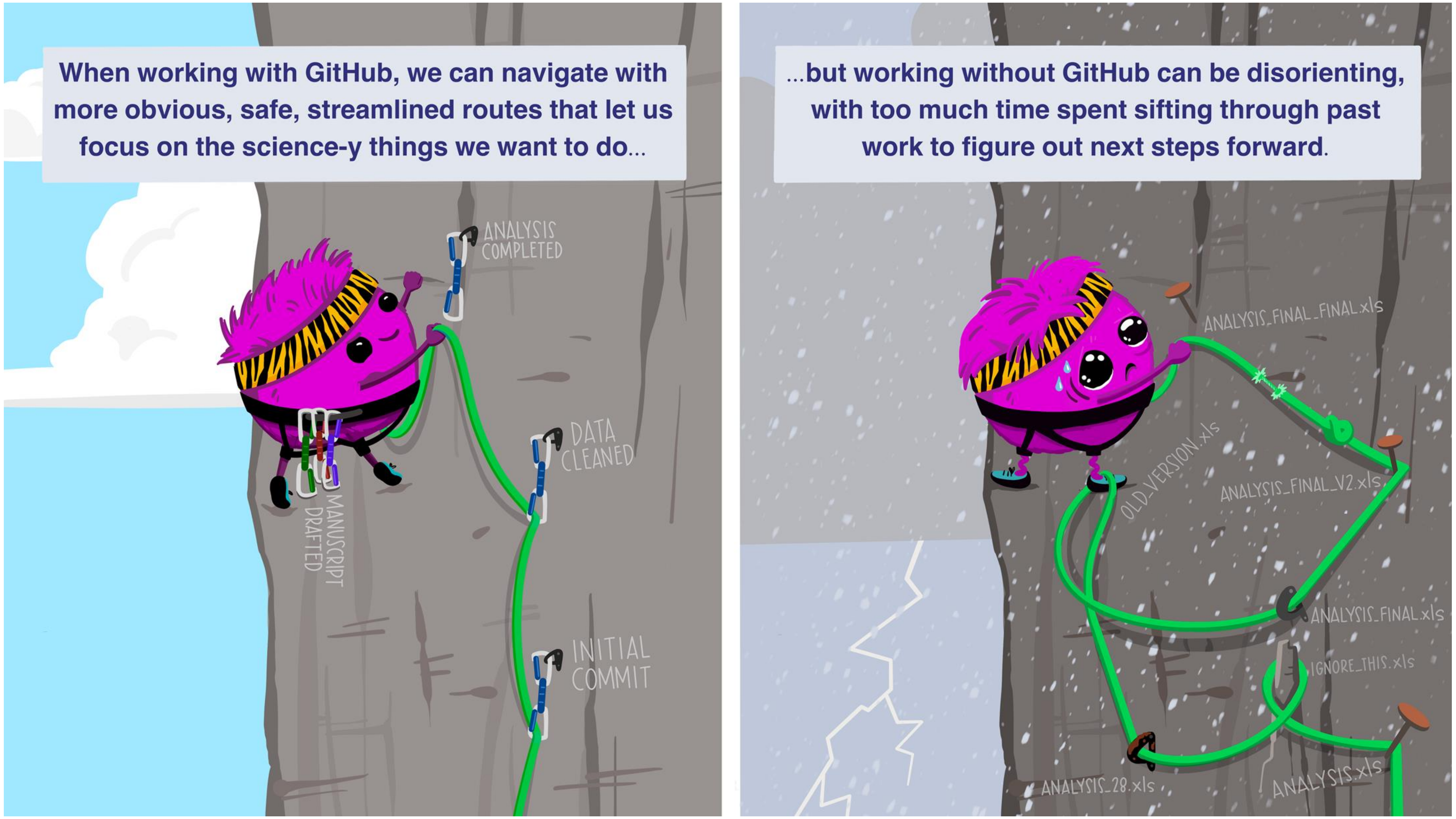


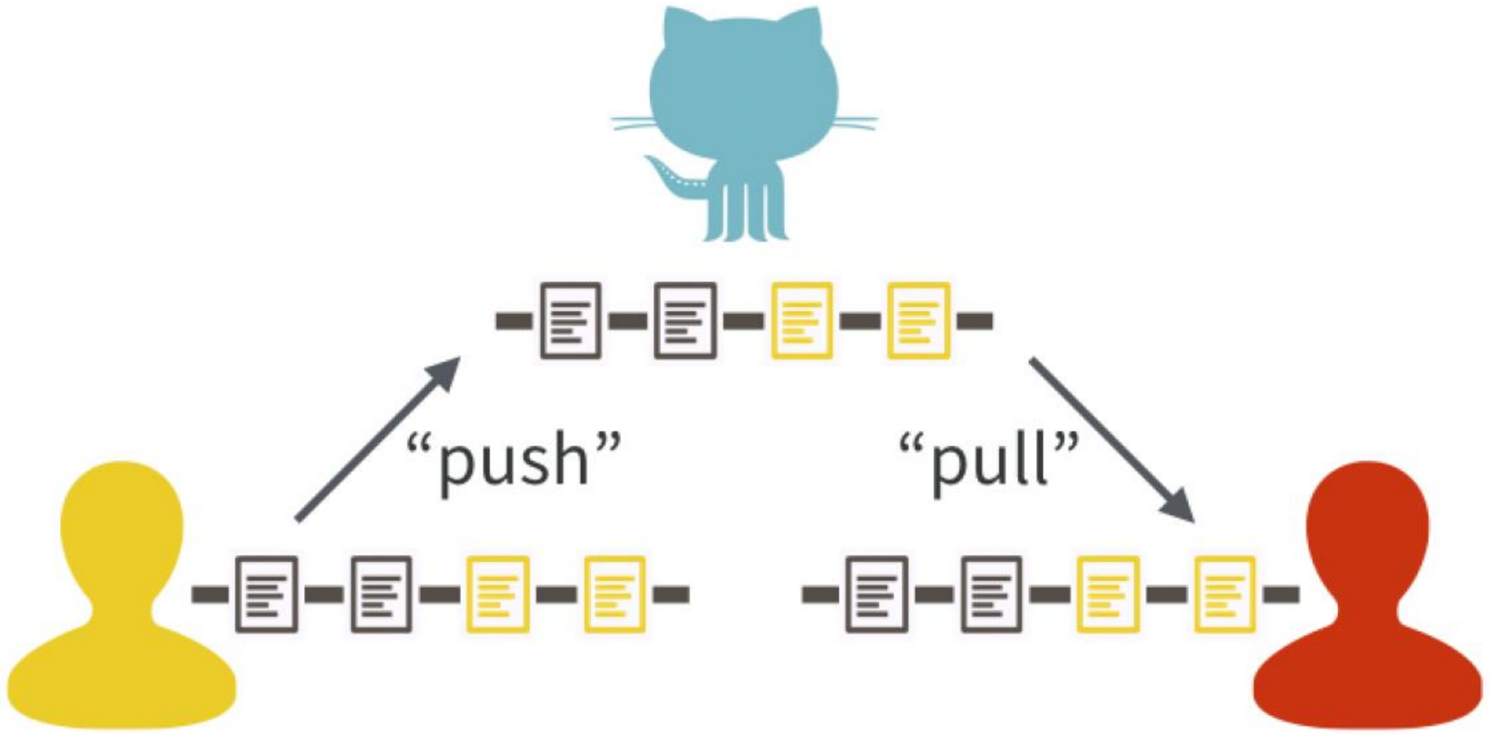
"Using a Git commit is like using anchors and other protection when climbing…**if you make a mistake, you can't fall past the previous commit**.

Commits are also helpful to others, because **they show your journey, not just the destination**."

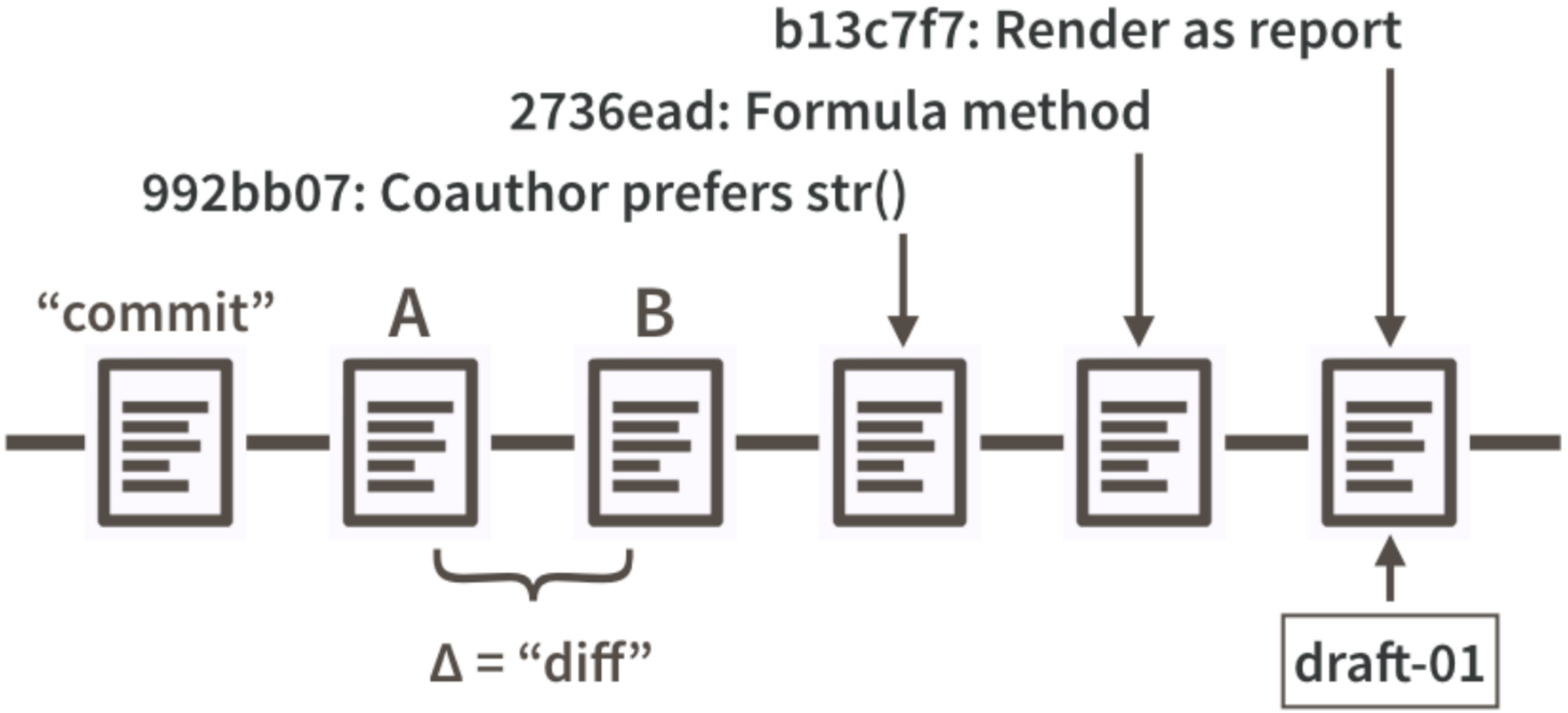
– HADLEY WICKHAM & JENNY BRYAN

Wickham & Bryan, RPackages (https://r-packages.org/preface.html)


When working with GitHub, we can navigate with more obvious, safe, streamlined routes that let us focus on the science-y things we want to do...
ANALYSIS COMPLETED
MANUSCRIPT DRAFTED
DATA CLEANED
INITIAL COMMIT
...but working without GitHub can be disorienting, with too much time spent sifting through past work to figure out next steps forward.
ANALYSIS_FINAL_FINAL.xls
OLD_VERSION.xls
ANALYSIS_FINAL_V2.xls
ANALYSIS_FINAL.xls
IGNORE_THIS.xls
ANALYSIS_28.xls
ANALYSIS.xls


"push"
"pull"

Under the hood

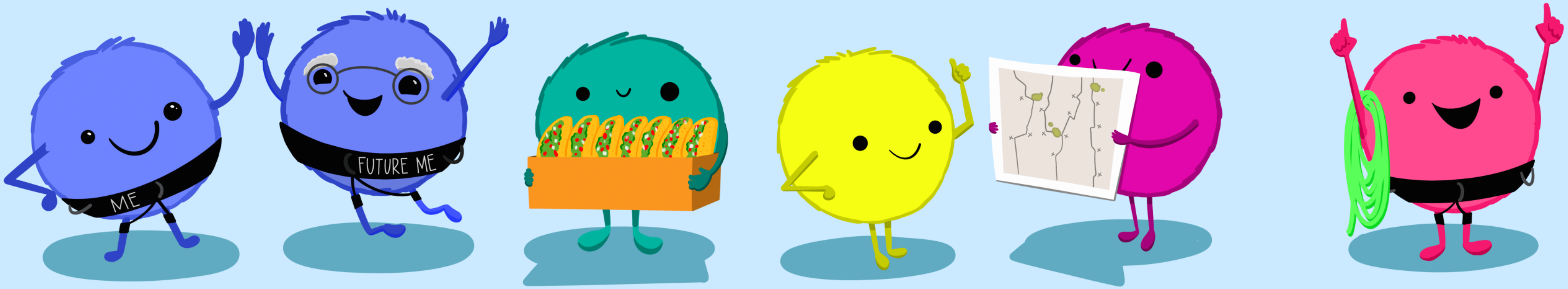


How does your workflow change given Git?

- Do what you normally do
  - But
    - Instead of just saving individual files, <u>periodically</u> make a commit of the repository (takes a snapshot of all files, so that it can be reverted to)
      - Snapshot is saved locally
      - Is this a meaningful version you might want to revert to at some point?
    - Push these commits to GitHub periodically

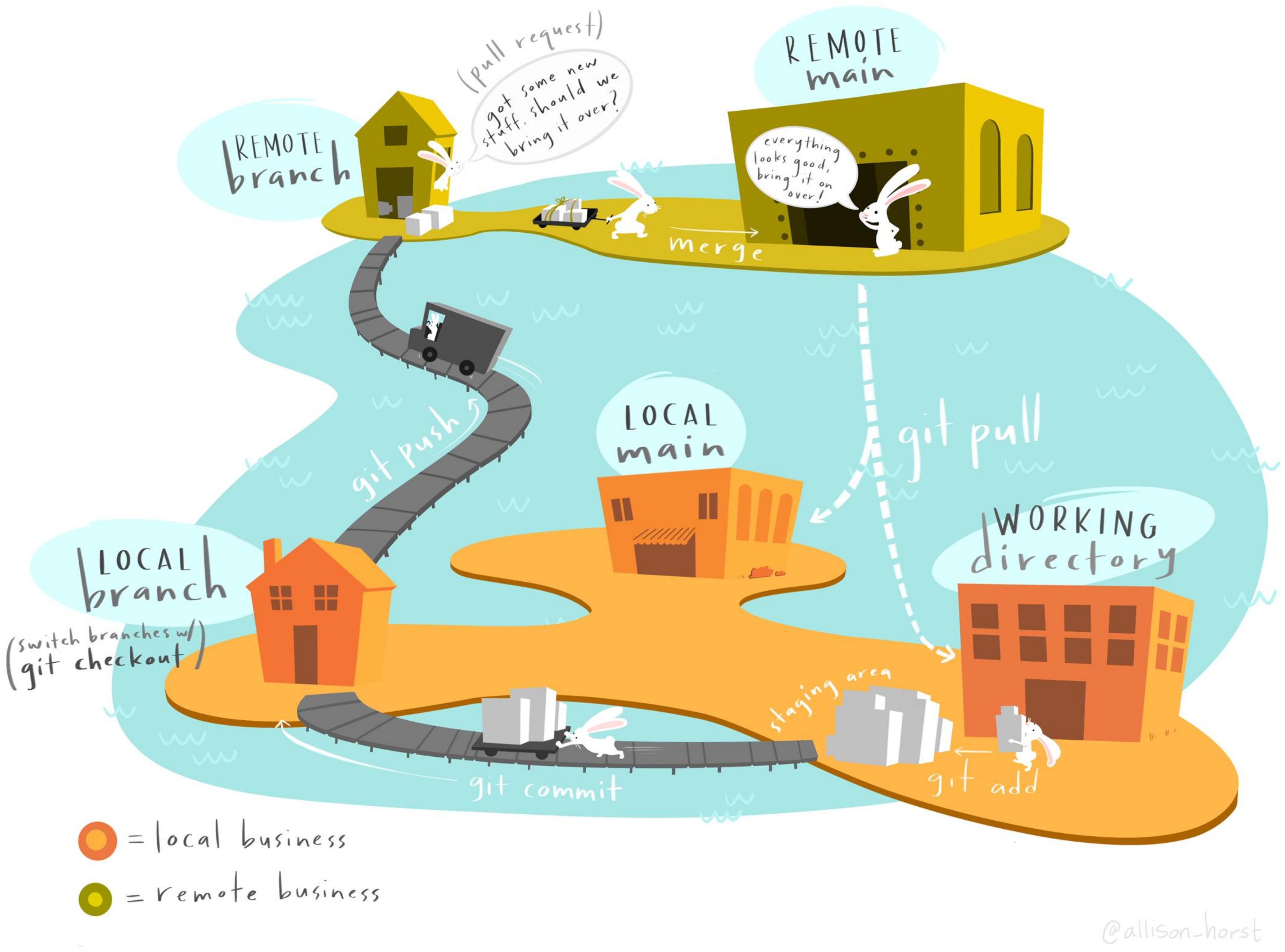
“**Collaboration is the most compelling reason to manage a project with Git and GitHub**. My definition of collaboration includes hands-on participation by multiple people, including your past and future self, as well as an asymmetric model, in which some people are active makers and others only read or review.”

-JENNY BRYAN



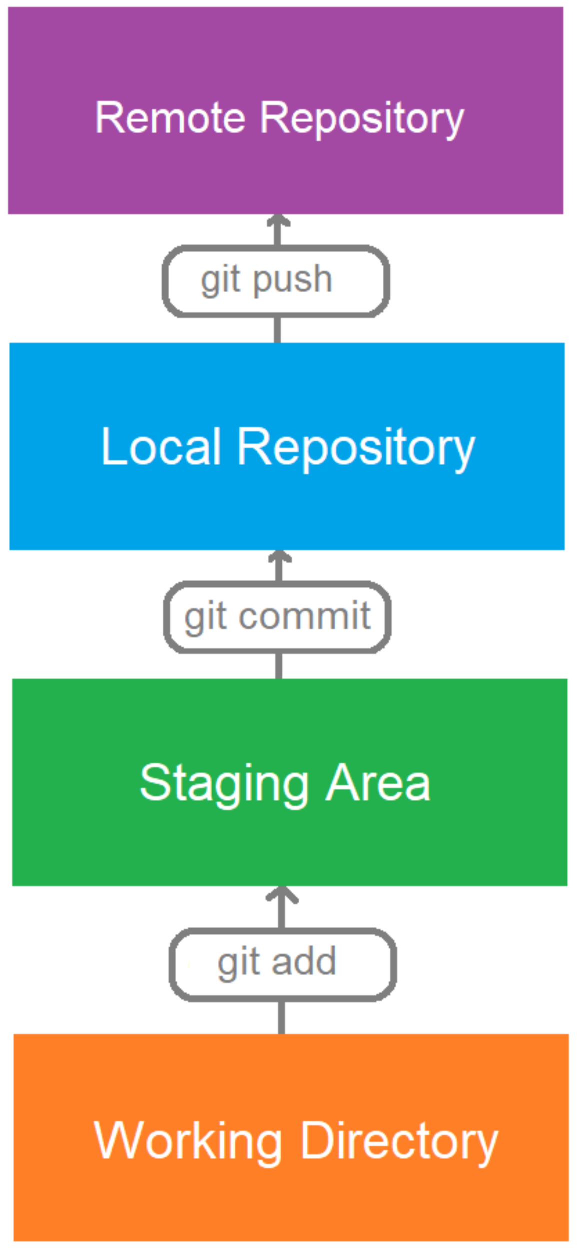
Bryan, J. 2017. Excuse me, do you have a moment to talk about version control? PeerJ Preprints. 5:e3159v2. DOI: 10.7287/peerj.preprints.3159v2

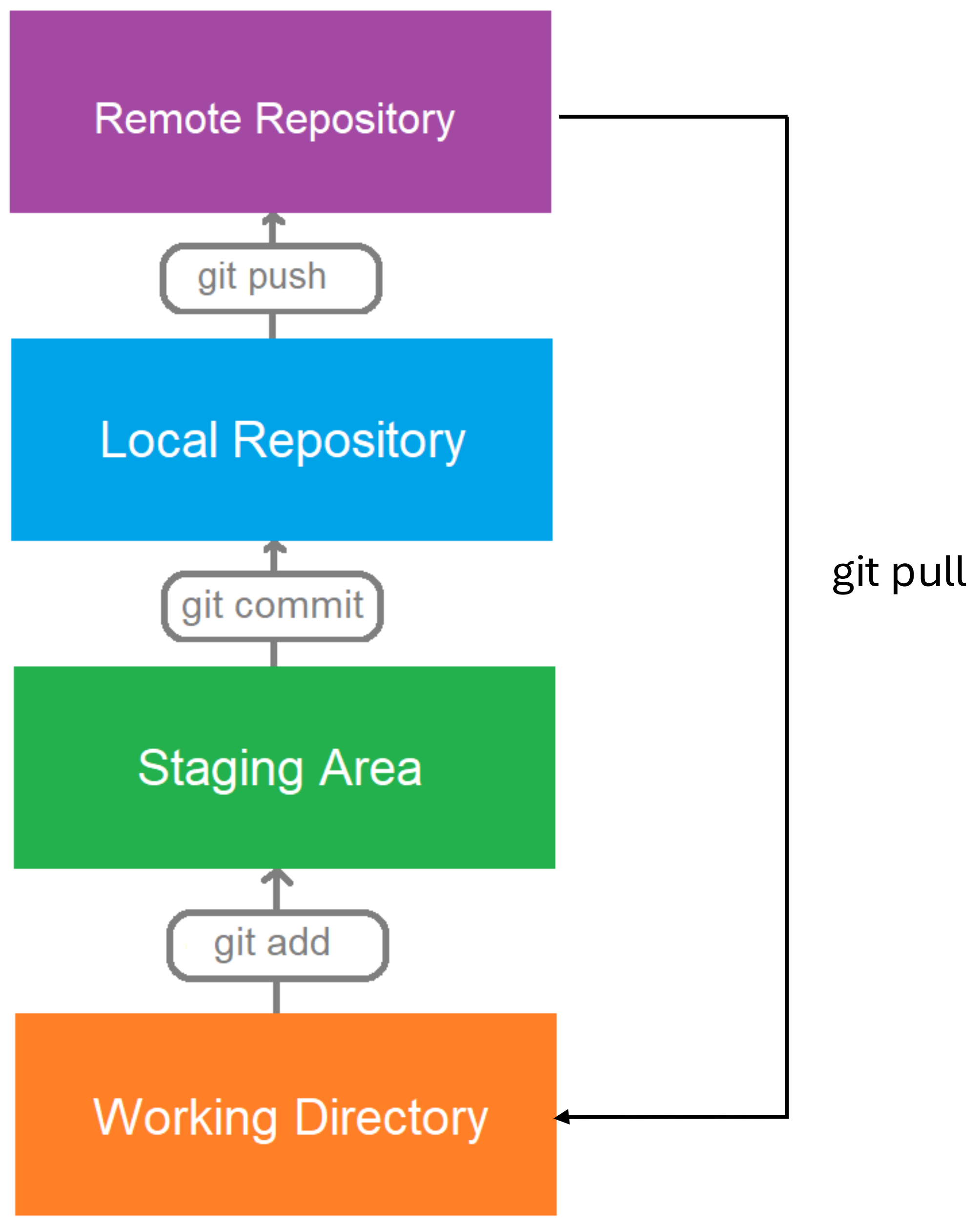
branches

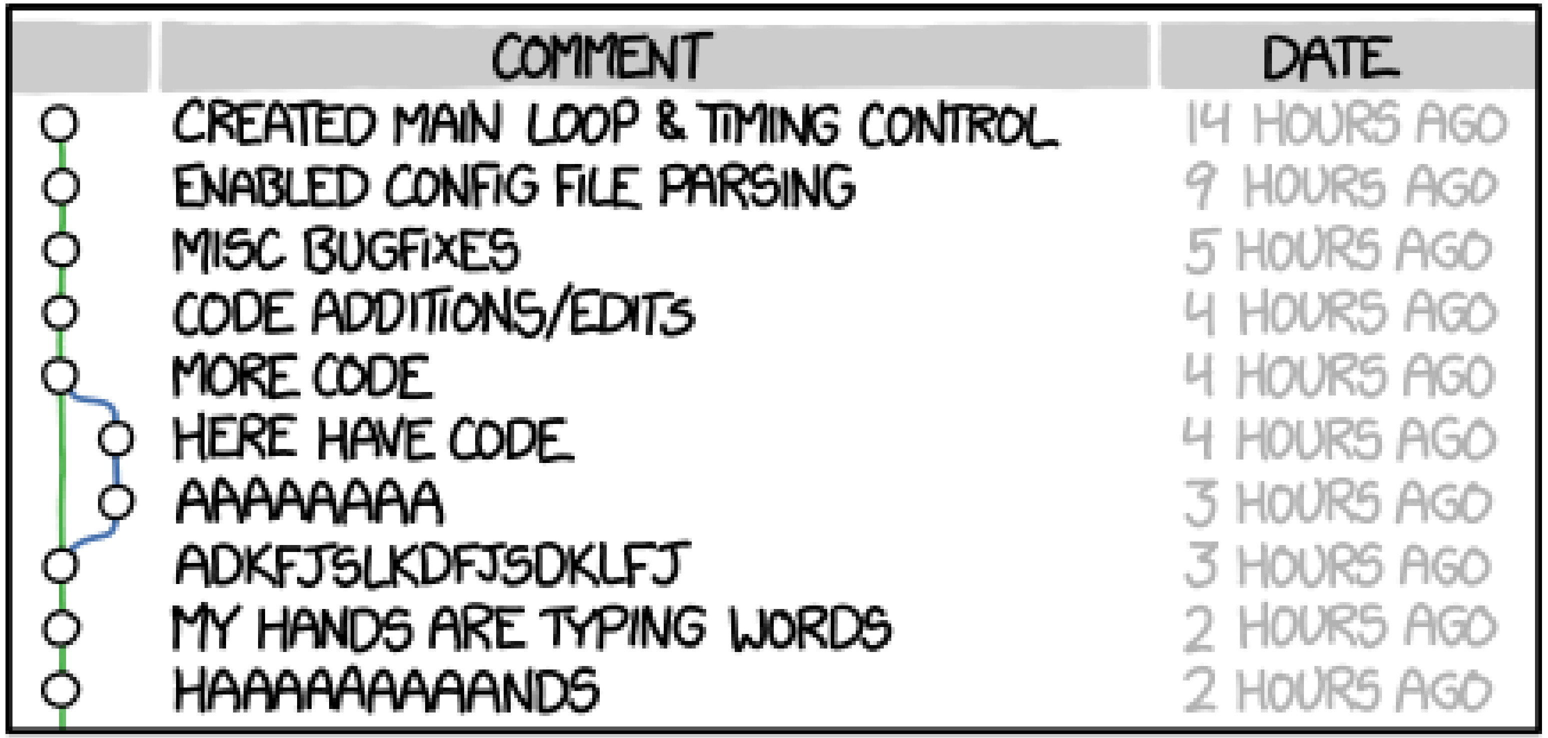

REMOTE branch
(pull request)
got some new stuff, should we bring it over?
REMOTE main
everything looks good, bring it on over!
merge
git push
LOCAL main
git pull
WORKING directory
LOCAL branch
(switch branches w/ git checkout)
staging area
git add
git commit
= local business
= remote business
@allison_horst

git HUG
MERGE CONFLICT

EXISTING ROUTES
OUR ROUTE
WARNING!
NOTES:
!!


Remote Repository
git push
Local Repository
git commit
Staging Area
git add
Working Directory


Remote Repository
git push
Local Repository
git commit
Staging Area
git add
Working Directory
git pull



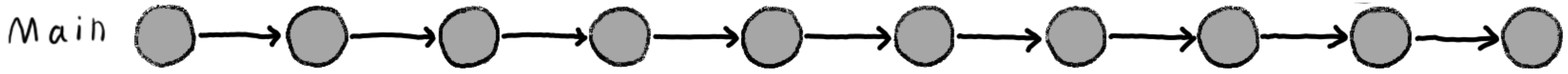
	COMMENT	DATE
	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
	ENABLED CONFIG FILE PARSING	9 HOURS AGO
	MISC BUGFIXES	5 HOURS AGO
	CODE ADDITIONS/EDITS	4 HOURS AGO
	MORE CODE	4 HOURS AGO
	HERE HAVE CODE	4 HOURS AGO
	AAAAAAAA	3 HOURS AGO
	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
	MY HANDS ARE TYPING WORDS	2 HOURS AGO
	HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

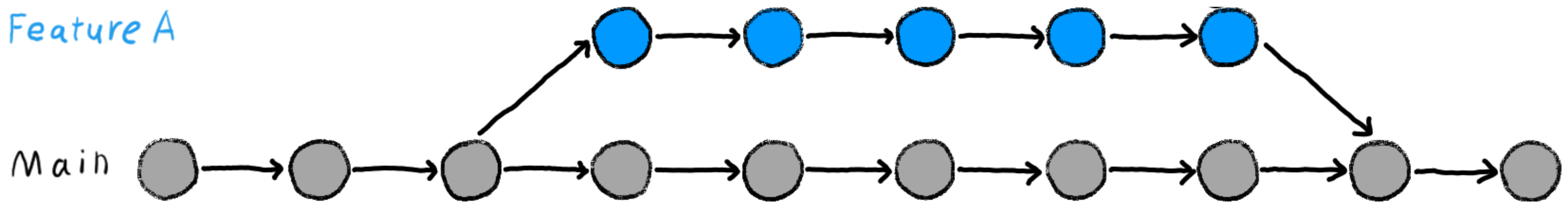
Ways to start a project

- Clone an existing one in Github, and take it from there
- Add a project that is already on your computer

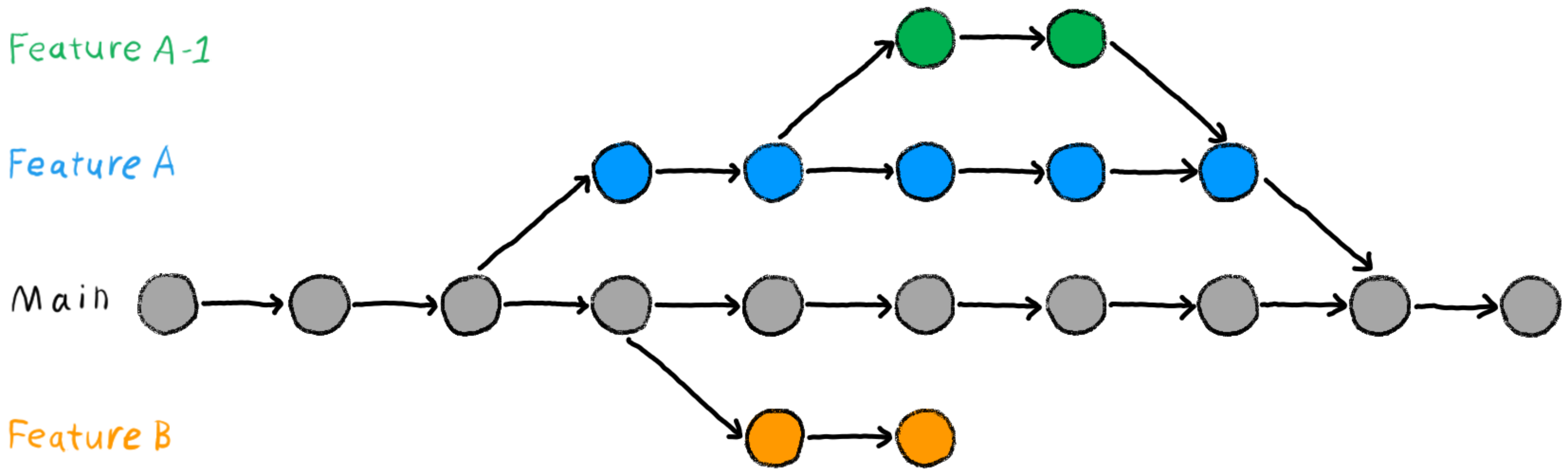
Version control (time travel)



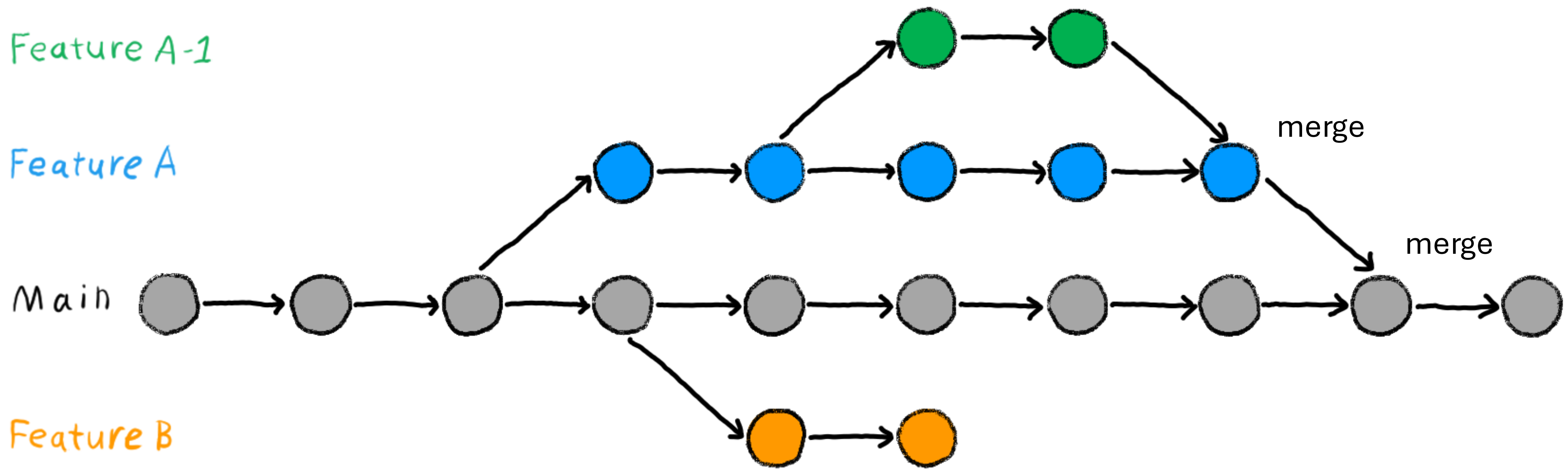
Version control (parallel universes)



Version control (parallel universes)



Version control (parallel universes)



Version control

- Let’s you rollback changes (Undo)
- Let’s you navigate through different version of the project
- Let’s you browse the history and timeline of changes

Git – Basic usage

- Basics
  - Make changes to a document
  - Save version history
  - Commit = snapshot of all files

- Advanced
  - Branching
  - Merging, etc.

Git & Github Setup

- Steps:
  1) Register for account with Github
  2) Install Git
  3) Connect Git to Github (using username/e-mail)

  4) Install a local git client (a GUI for git)
     - Good to get used to the workflows, and get confident with the basics

  5) Create Repository – test_repo
  6) Check connection -- clone

Steps

1) Creating Account

- Visit https://github.com/
- Sign up
  - Email
  - Username

2) Install git

- See instructions at https://git-scm.com/install/

3) Connect git to Github

- git config --global user.name "*username*"
- git config --global user.email "*email*"

4) Install git client

- https://desktop.github.com/download/

Task

- Create a repository for your class project on GitHub, with a brief description.
  - Clone the repository to your computer
  - Populate the repository with your files ***(e.g. papaja file.pdf)***
  - To push to your repository you can use GitHub Desktop (or Rstudio).
  - Add suyoghc (suyoghc@gmail.com) as a collaborator

See

Excuse Me, Do You Have a Moment to Talk About Version Control?