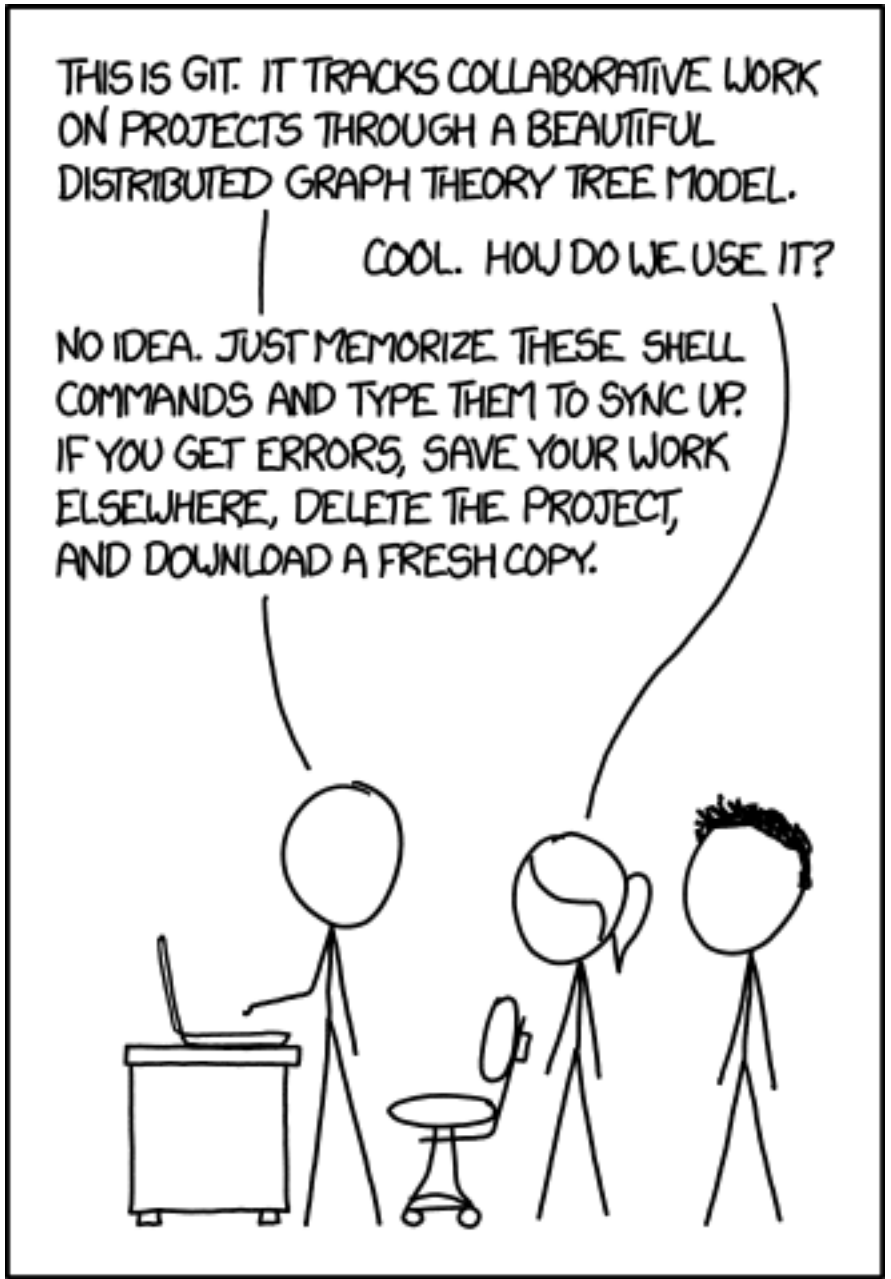
Intro to Git for Turner Group

Eliot Kim, Eric Mei

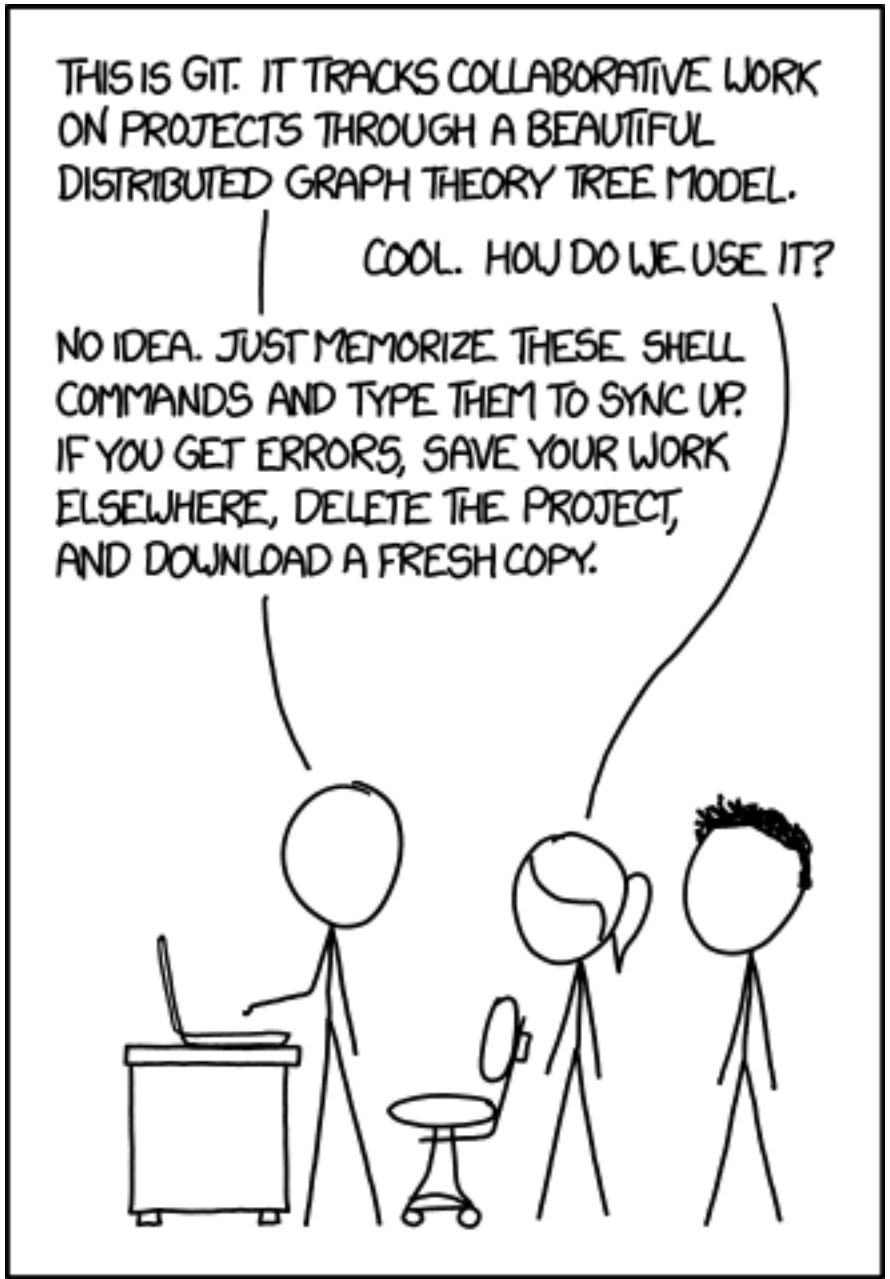
August 27th, 2025

Why git?


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Why git?

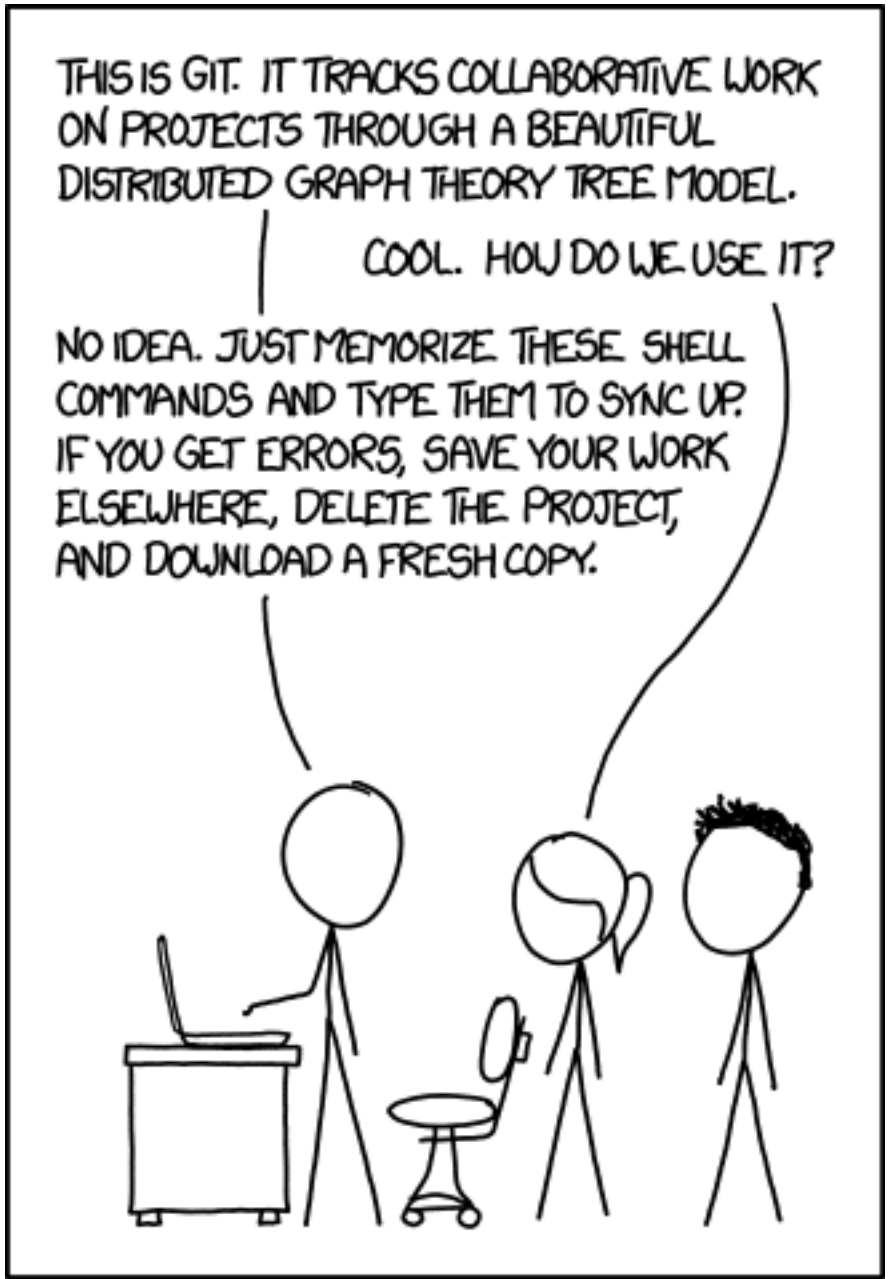
- **Version control:** Saves prior code versions, can revert if needed
- **Collaboration**: Team members can edit same codebase in parallel
- **Accountability**: Write code that others can read, run, and edit!



Why git?

- **Version control:** Saves prior code versions, can revert if needed
- **Collaboration**: Team members can edit same codebase in parallel
- **Accountability**: Write code that others can read, run, and edit!

- ***Git vs. GitHub***
  - **Git**: Software for managing LOCAL repositories
  - **GitHub**: Web service with graphical interface for hosting REMOTE repositories

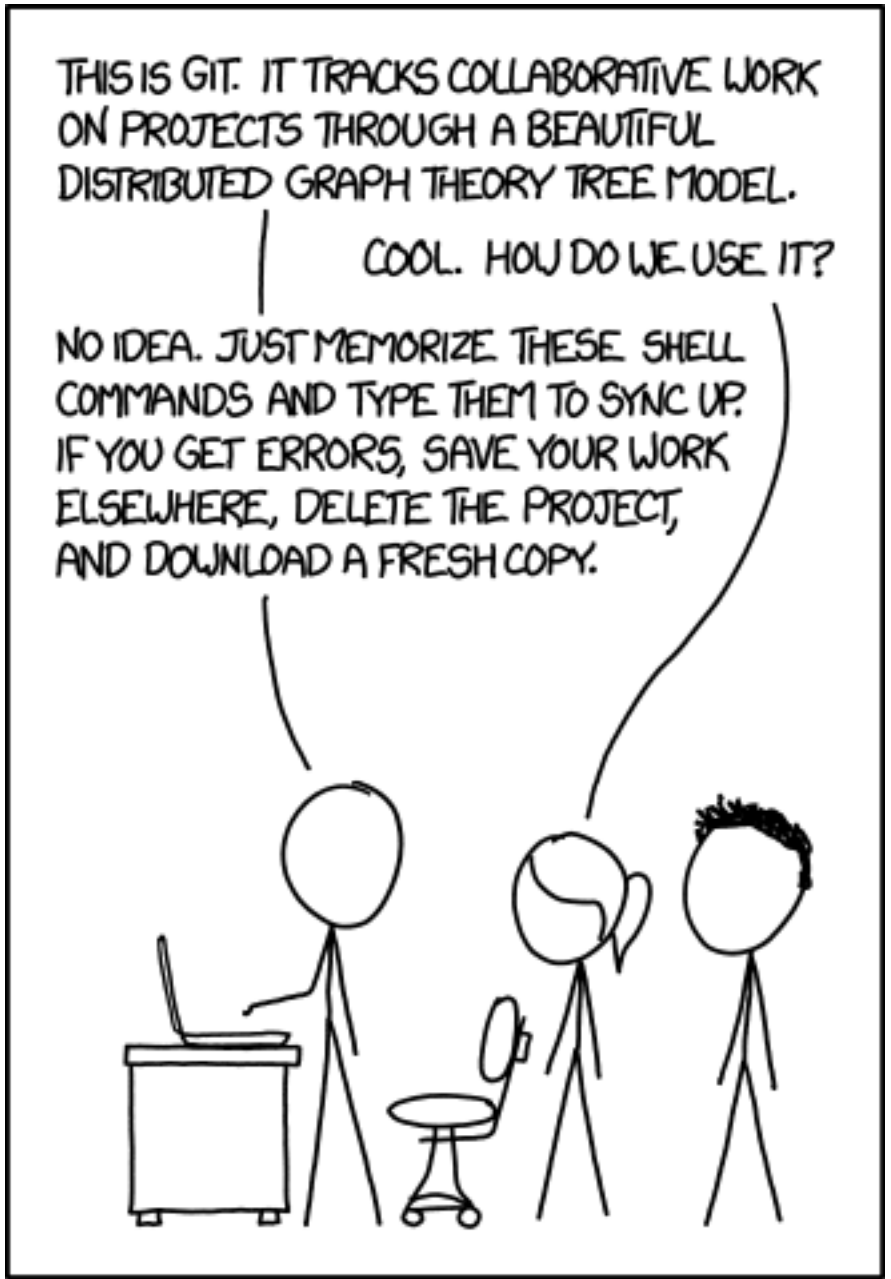


Why git?

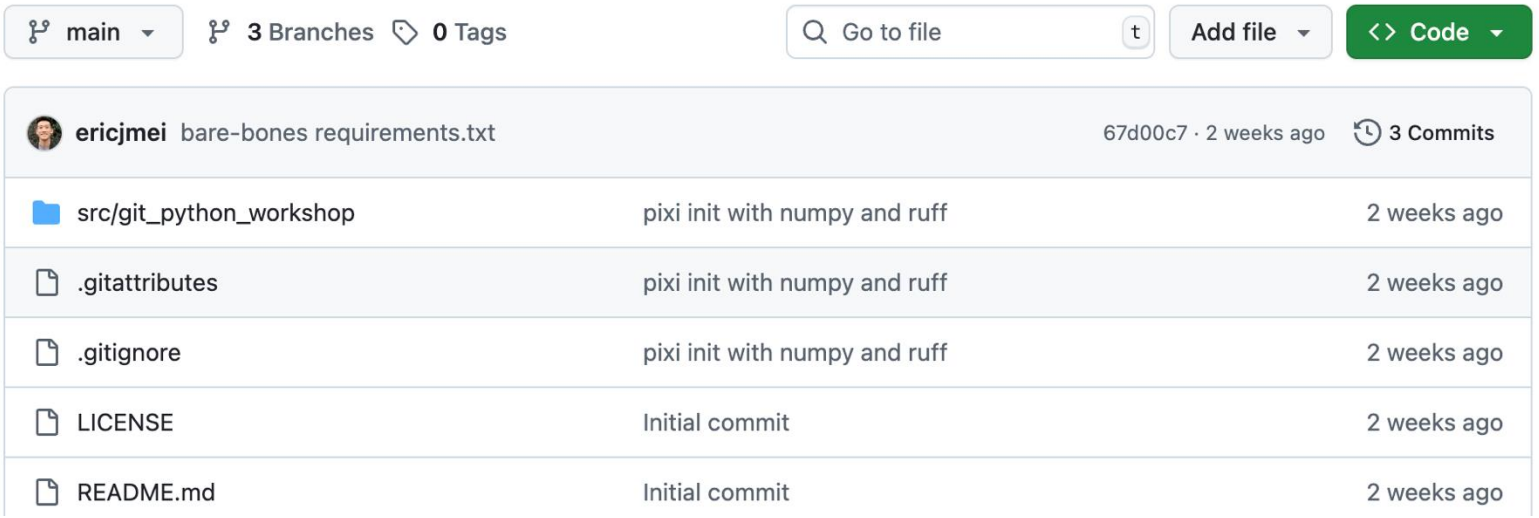
- **Version control:** Saves prior code versions, can revert if needed
- **Collaboration**: Team members can edit same codebase in parallel
- **Accountability**: Write code that others can read, run, and edit!

- ***Git vs. GitHub***
  - **Git**: Software for managing LOCAL repositories
  - **GitHub**: Web service with graphical interface for hosting REMOTE repositories

- **Future Career:** Many industry jobs request links to GitHub profiles!



How does git work?



- **Repositories:** Where the code is stored
  - **README**: Directions to run code
  - **.gitignore**: Specific filetypes to ignore when updating the repo
- **Branches:** Make parallel edits to codebase, i.e. adding a new feature
  - **Main**: The “production” branch
- **Commits:** Repo edit history — “versions”
- **Pull Requests:** Merging edits made in a branch with the “main” branch

How to get, edit, and save code?

git clone <REPO LINK>

- *Copy repo to LOCAL*

git status

- *Check which files have been changed / unstaged / staged*

git add <FILEPATH>

- *Stage file(s)*

git commit –m <MESSAGE>

- *Add staged file(s) to repo's commit LOCAL history*

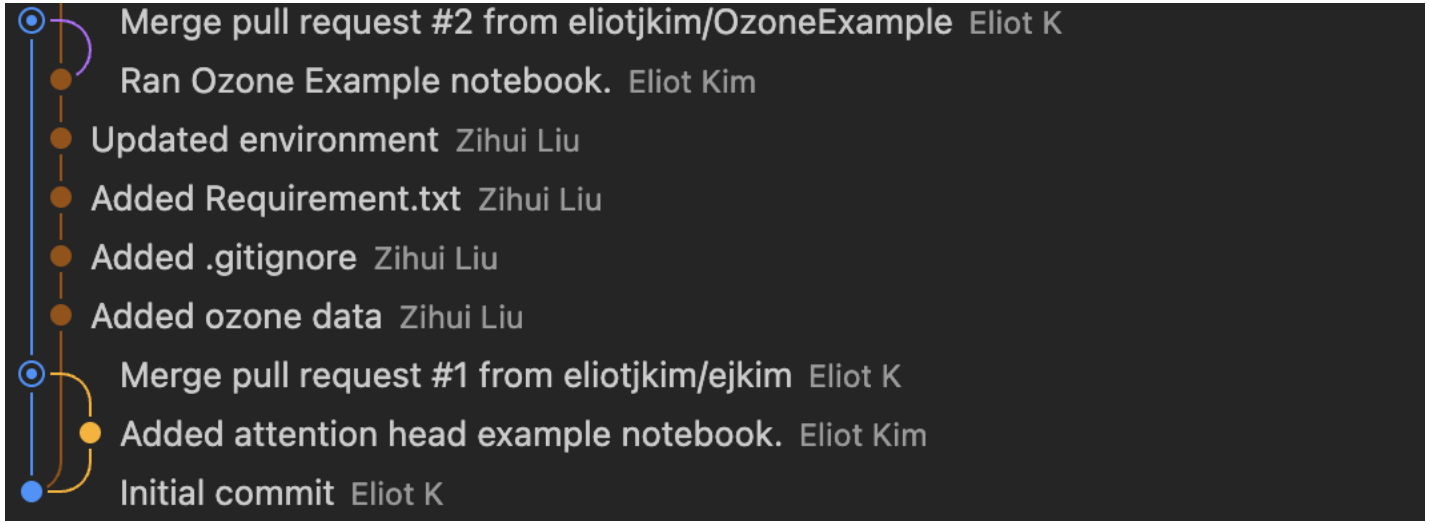
git push

- *Update REMOTE codebase with committed file(s)*

git pull

- *Get codebase updates from REMOTE to LOCAL*

How to collaborate using GitHub?



git branch

- *Check current branch and other available branches*

git branch <BRANCH NAME>

- *Make a new branch*

git checkout <BRANCH NAME>

- *Hop to the desired branch*

Group Exercise!

1. **Clone repo!**

github.com/ericjmei/git_python_workshop

2. **Make a branch (include your name / username in branch name)!**
3. **Write and push edits to a function in *git_intro.py***
   - James: add
   - Iana: subtract
   - Sydney: divide
   - Alex: multiply
   - Eric: exponent
   - Eliot: mod
4. **Create a pull request for your branch!**
5. **Merge the pull requests into the main branch (as a group)**