Git

What is Git?

- Git is a **distributed version-control application**

- **Version Control:**
  - Managing source code, changes in source code, backups, etc

- **Distributed**
  - Built for (1) collaboration, and (2) sharing code across different machines, over the web
  - Distributed also means there's no universal "master copy" of the code

Git vs GitHub

- Git:
  - A program you run on your computer
  - Each git installation also acts as a mini "server" that you can push/pull code from (with the correct setup)

- GitHub:
  - A website that provides a nice interface to a git installation hosted on GitHub servers
  - Think of it as a service that helps run a git installation, except on the server instead of on your local computer
  - Because it's online, it always comes with accounts, sharing, etc
  - E.g. GitLab is a very similar alternative

Git Overview

- Goal: We want to be able to:
  - Keep/restore old versions code when we make changes
  - Not have to worry about accidentally deleting code
  - Switch between different "branches" when we are working on different features
    - E.g. working on a bug fix vs. adding a new feature, we don't want the two changes to interact
  - Merge different branches
    - And resolve any conflicts between them
  - Pull and push changes
    - Pull from others' work, or push your copy to be shared to others

All this can be done very easily with Git: but don't rush in!

Git Concepts

- Repository
- Fork
- Add
- Reset
- Commit
- Branch
- Remote
- Merge
- Rebase
- Push/Pull
- Checkout

Windows Users: I'm sorry

(Also, use git bash for everything we're about to do)

Git

- Git is **extremely** complex.
- Even most veterans software developers will only use a small fraction of its functionality.
- Do not get intimidated by git commit trees and rebasing and the theoretical foundational concepts of git
- We will start small

(1) Assume you just have a local copy of code that you want to record a history of changes

We're going to:

1. Create a local git project
2. Add a file
3. Commit it (with a message)
4. Look at the log

(2) Assume that there is a static, external copy of the code that you're going to make a copy of

(3) You just want to get a local copy of some external code

Git beginner commands

- git init (create a git repository using the current folder)
- git status (look at the status of your repository)
- git add (adding a file for git to track)
- git commit -m "my message" (making a commit)
- git log (viewing the git log)
- git clone (cloning a repository)
- git rm (removing files)
- git mv (renaming files)

Vim

- Vim is a terminal-based text editor
- Vim is a **modal** editor, which means it has different modes (insert, normal, visual) modes to shift between
    - That's why when it shows up, it can be confusing. You need to switch into the right mode to use it.

Vim

Cheatsheet:

- Pressing ESC will put you back into Normal Mode
- In Normal mode, Type
  :q to exit.
  :q! to exit without saving.
  :wq to exist with saving
- In Normal mode, type **i** to enter Insert mode. Now you can move around and type as usual!
- In Insert mode, press ESC to go back to Normal Mode