Intro to Git

Jonathan Tannen

What is Git?

A version control system for coordinating work across people and tracking changes.

Tons of resources and tutorials.

E.g.

- https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners
- Github's extensive documentation and tutorials.

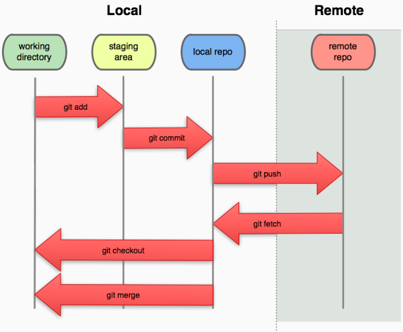
Git terminology

- **Version Control**: Stores history of files ("versions")
- **Distributed Version Control**: Remote source of truth, with local copies.
- **GitHub**: A private company (now Microsoft) with fancy tools integrating git.
- **Repository ("Repo")**: A self-contained project.

Getting Started

- Sign up at github.com
- Install and set up git
    - https://docs.github.com/en/get-started/quickstart/set-up-git
- Install GitHub CLI or UI.

Your local workflow



from https://kevintshoemaker.github.io/StatsChats/GIT_tutorial.html

Your local workflow

Do once:

- `git init`: Initiates git in whatever folder you're in, and all subfolders.

Repeat:

- Make changes to files.
- `git status`: See what files you've changed.
- `git diff file1`: See what changes you've made to file1.
- `git add file1 file2`: Stage changes to be committed for file1, file2.
  - `git add .`: Stage *all* changes.
- `git commit -m "Informative description here"`: Commit the changes.

and now. . .

Demo

Your remote workflow

First time you connect your Repo

- `git init` your local repo.
- Create the repo in github
  - Option 1: In the github.com webview.
  - Option 2: `gh repo create`
- `git push --set-upstream origin HEAD`
  - Only need set-upstream the first time.

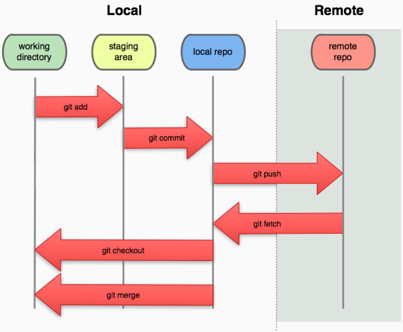
In general. . .

- `git pull`
- `git push`
- `git merge`

Some terminology. . .

- `origin/master`

Your local workflow



from
https://kevintshoemaker.github.io/StatsChats/GIT_tutorial.html

Git Rules

1. Commits should be small (do one thing).
2. Commits should be complete (they should work).

Homework for next week

Goal:

- A GitHub Repo in our Organization
- Named `lastname_firstname_projectnickname`
- With…
  - A folder `raw_data/` with raw data.
  - A folder (`R`, `py`, `src`, `scripts`) with code to explore the data.
  - A `README.md` file describing the project and folder organization.

CPLN 680 GitHub Organization

- Log in to GitHub
- I've invited all of your upenn.edu email addresses.
  - If you have another account, let me know.
- Navigate to https://github.com/CPLN-680-Spring-2022
- Accept the Invitation to join.
- Create a new Repo named `lastname_firstname_projectnickname`
- Link your local repo (see previous slide).

Advanced git

- `.gitignore`
- `git branch`
- `git merge`