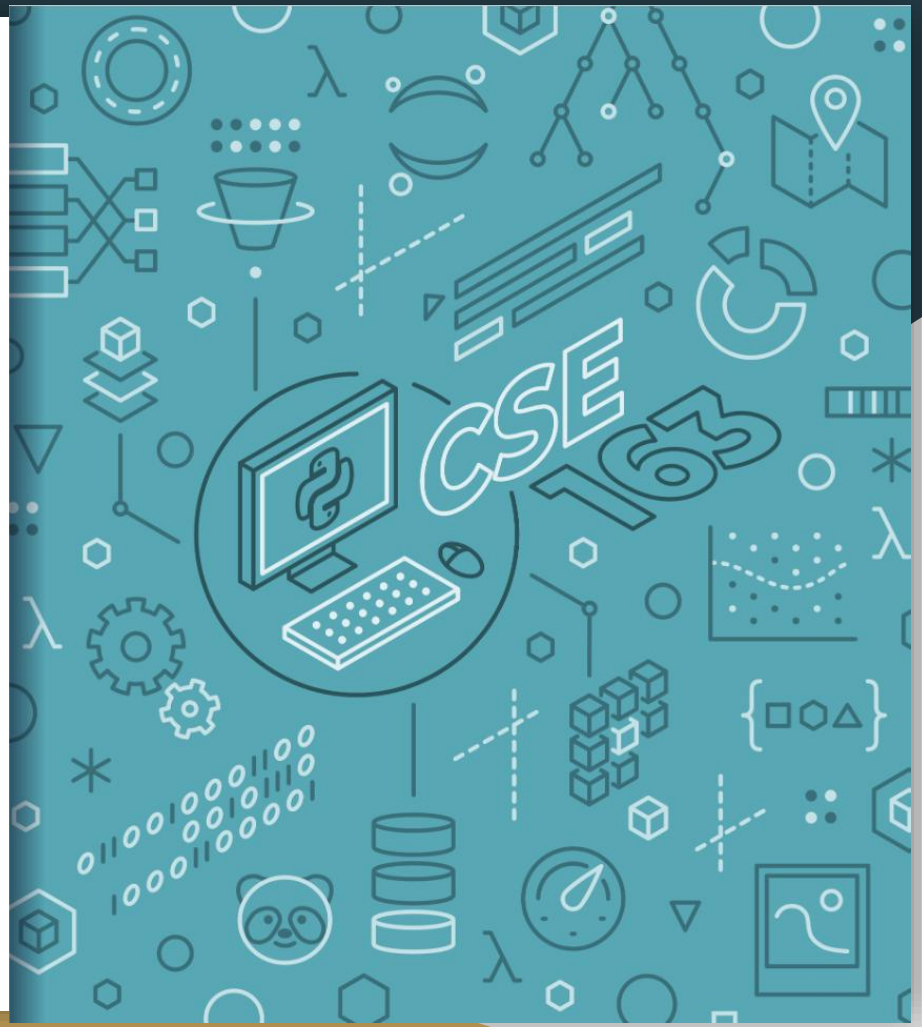
CSE 163
Introduction to Git

Adrian Salguero
Spring 2026

💭 Icebreaker (discuss with neighbors):
What are you most excited about for summer?

slido.com
#cse163

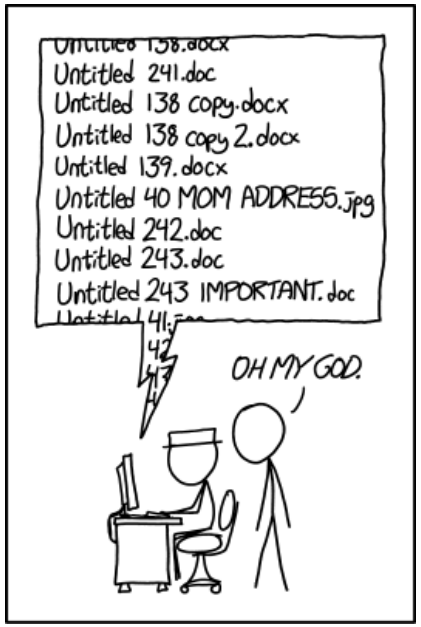


Announcements

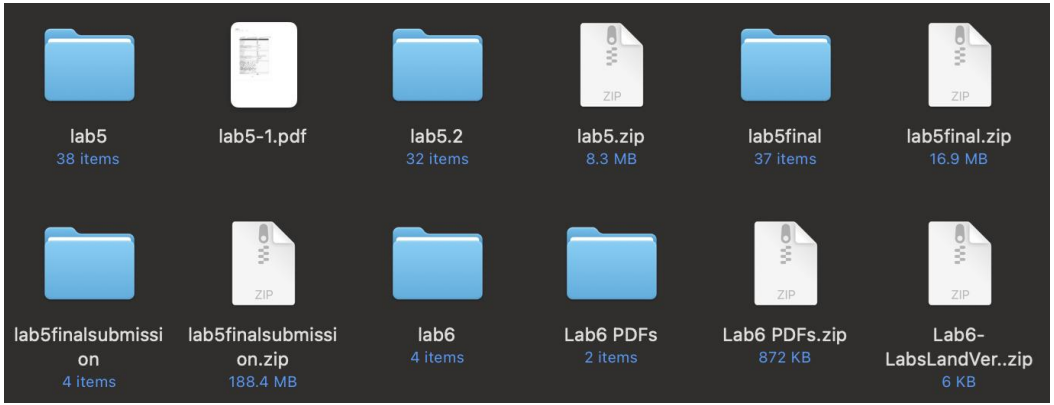
- **THA 5 Peer Reviews** due Monday June 1st at 11:59pm
- **Checkpoint 5** due Tuesday, June 2nd at 11:59pm!
- **Lesson 25 Quiz** due Sunday, May 31st at 11:59pm
- **Project Part 3 (Final Deliverables)** due June 5th at 11:59pm on Gradescope!
- **Presentations** next week during Section and Friday's lecture (June 5th)
- **Final Reflection** on Monday, June 8th at 8:30am in KNE 220

Version Control

- Allows for collaboration, makes reverting mistakes easier, accountability



Google Docs Google Slides



History of Git

- Created by Linus Torvalds in 2005
- Desired an efficient version control system that supported large scale coding projects
- Centralized vs. distributed version control:
  - ***Centralized***: a central server processes, maintains, and shares all changes
  - ***Distributed***: all users maintain a working copy (local repository) on their own machine

https://git-scm.com/

Git vs. GitHub vs. GitLab vs...

- Git is the tool that allows this version control system to function locally
- GitHub and GitLab are hosting services for git's remote functionalities (e.g. hosting remote repositories)

https://github.com/

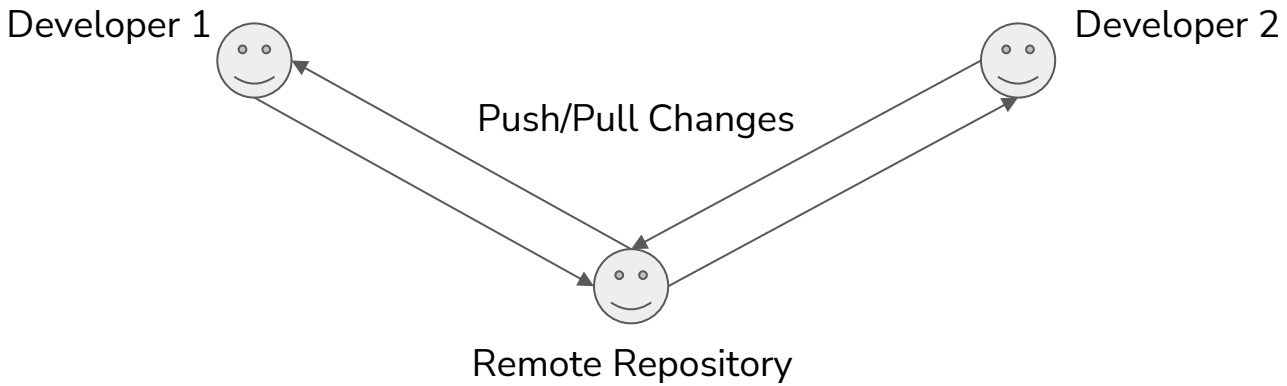
https://about.gitlab.com/

Repositories

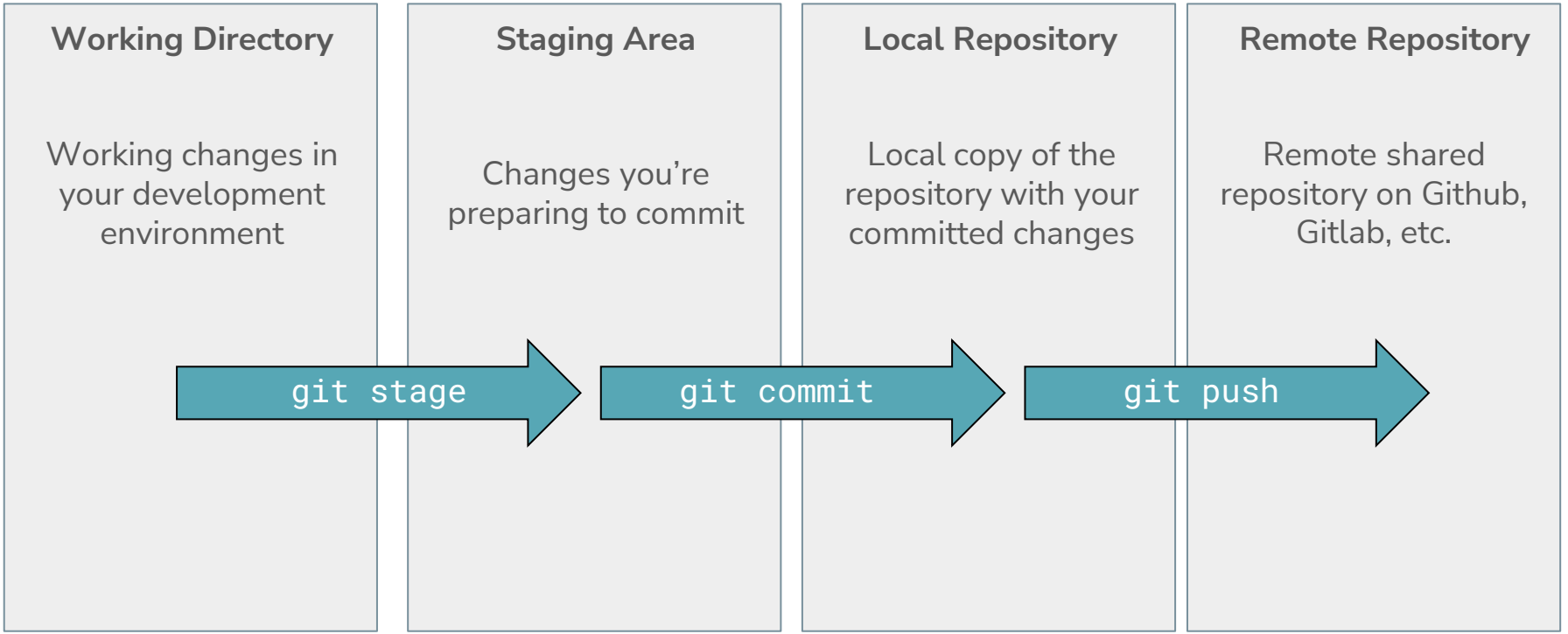
- A repository (e.g. “repo”) is a location that stores a copy of all your files
- A repo can be considered local or remote
- A git repo generally contains:
  - Source files such as Python files (.py), Jupyter Notebooks (.ipynb), etc.
  - General resource files such as images
  - README.md files and notes (generally instructions on what the repo contains and how to use it)
- A repo ***should not*** contain:
  - Executable or generate files
  - Data (e.g. .csv files, .shp files, etc.) or files that take up a lot of space
  - .gitignore file - specify which files to not commit to repo

Repositories

- With git, everyone working on the project has a complete version of the repo
  - There is a remote repository, which is the central repository → “source of truth”
  - Remote repositories are hosted on services like GitHub and GitLab
  - Everyone has a local copy of the repository, which is what we use to send our own changes to the remote repository



The 4 Stages of Git



Git Clone

```
git clone <URL>
```

- Used to clone an already existing remote repository onto your local machine
- For private repositories, requires authentication (e.g. SSH keys, etc)
- After cloning, you can collaborate with others who have access to the remote repository

Git Push

```
git push
```

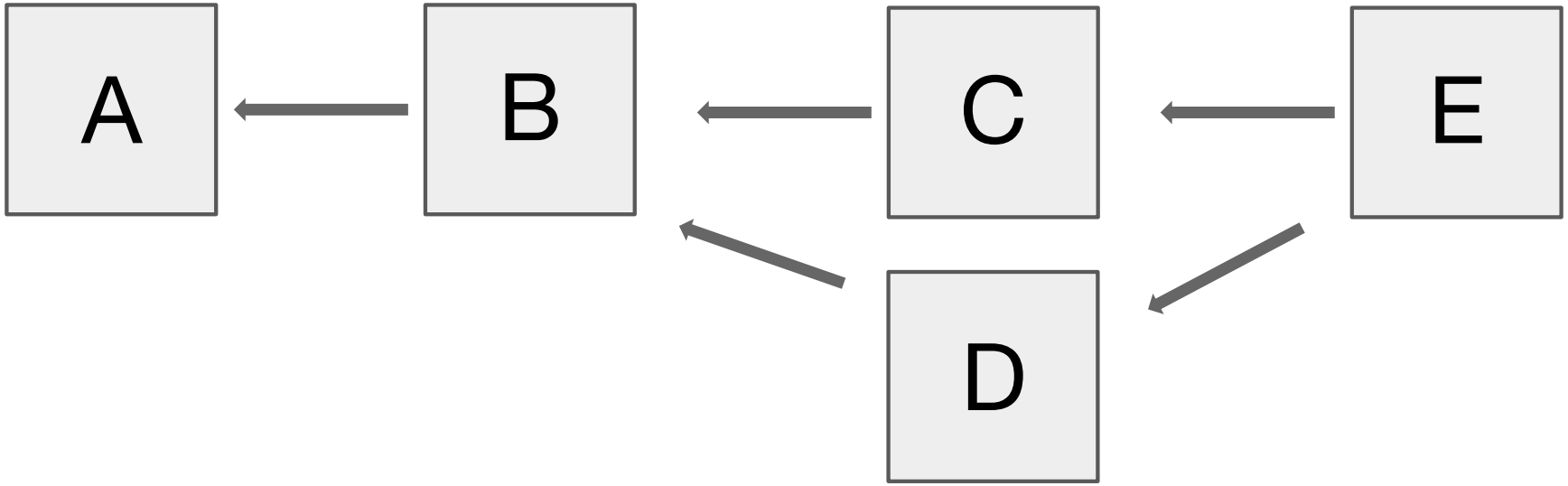
- Used to push changes from your local repo to the remote repo
- Once you push changes, they are visible to all others!
- If someone else wants to grab the changes you pushed, they will use the command on the next slide…

Git Pull

```
git pull
```

- Used to update your working directory with changes from the remote repository
- Combination of the commands ***fetch*** and ***merge***
- ***Fetch*** retrieves the changes from the remote repository to your local repository
- ***Merge*** combines the changes in your local repository with your working directory

Branching and Diverging Branches



- A developer working on their own branch (with commit “D”) separate from branch “main”
- A developer trying to pull changes, but encountering a diverging branch due to conflicting changes from the remote repo

Git Reset

```
git reset HEAD filename
```

- Used to remove a file previously added to the ***staging*** area
- Does not affect changes in your working directory

Git Revert

```
git revert HEAD
```

- Deleting a commit involves altering the entire commit history and can cause issues
- While it is possible to “delete” commits, it’s best practice to avoid doing so
- Think carefully of what you want to commit instead of going through the process of reverting