CS134:
Lab 01

Today's Plan

Tools & resources for doing CS Labs!

After this lesson, you should be able to:

1. Explain the difference between VS Code, Terminal, and python
2. Navigate your file structure with Terminal
3. Submit your lab assignments to be graded

Lab Deadlines

- Depend on your lab session day!

- Mondays —> Wednesday @10p
- Tuesdays —> Thursday @10p

Lab Instructions on Course Website!

- https://bit.ly/csi34f24

Labs

Lab assignments are due Wednesday 10 pm EST for students in Monday lab sections, and Thursday 10 pm EST for students in Tuesday lab sections. To request a 12-hour extension, please fill out this form *before* your lab's submission deadline.

Lab Date	Topic
Sept 09/10	Lab 1: Python/Git Workflow
Sept 16/17	Lab 2: Day of the Week
Sept 23/24	Lab 3: Madlibs
Sept 30/Oct 1	Lab 4: Voting
Oct 21/22	Lab 5: What's In A Name?
Oct 28/29	Lab 6: Thinking Recursively
Nov 4/5	Lab 7: AutoComplete
Nov 11/12	Lab 8: Boggle! [Part A] [Part B] [Graphics module]
Dec 2/3	Lab 9: Sorting

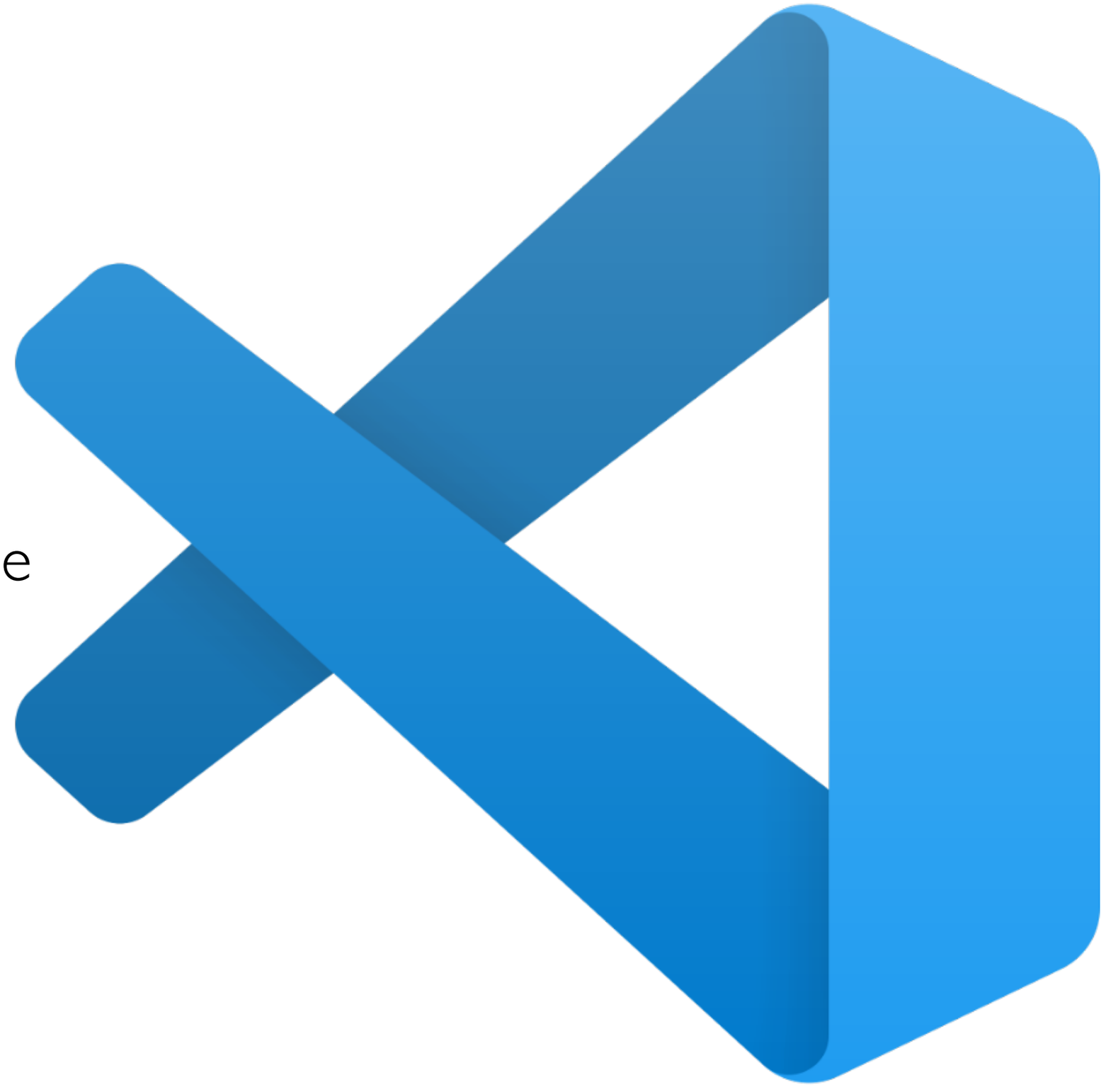
What do these 4 applications have in common?

Google Docs

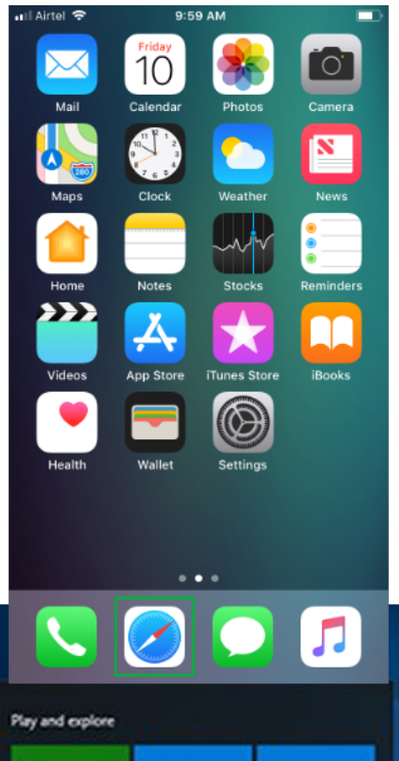
What do these 4 applications have in common?

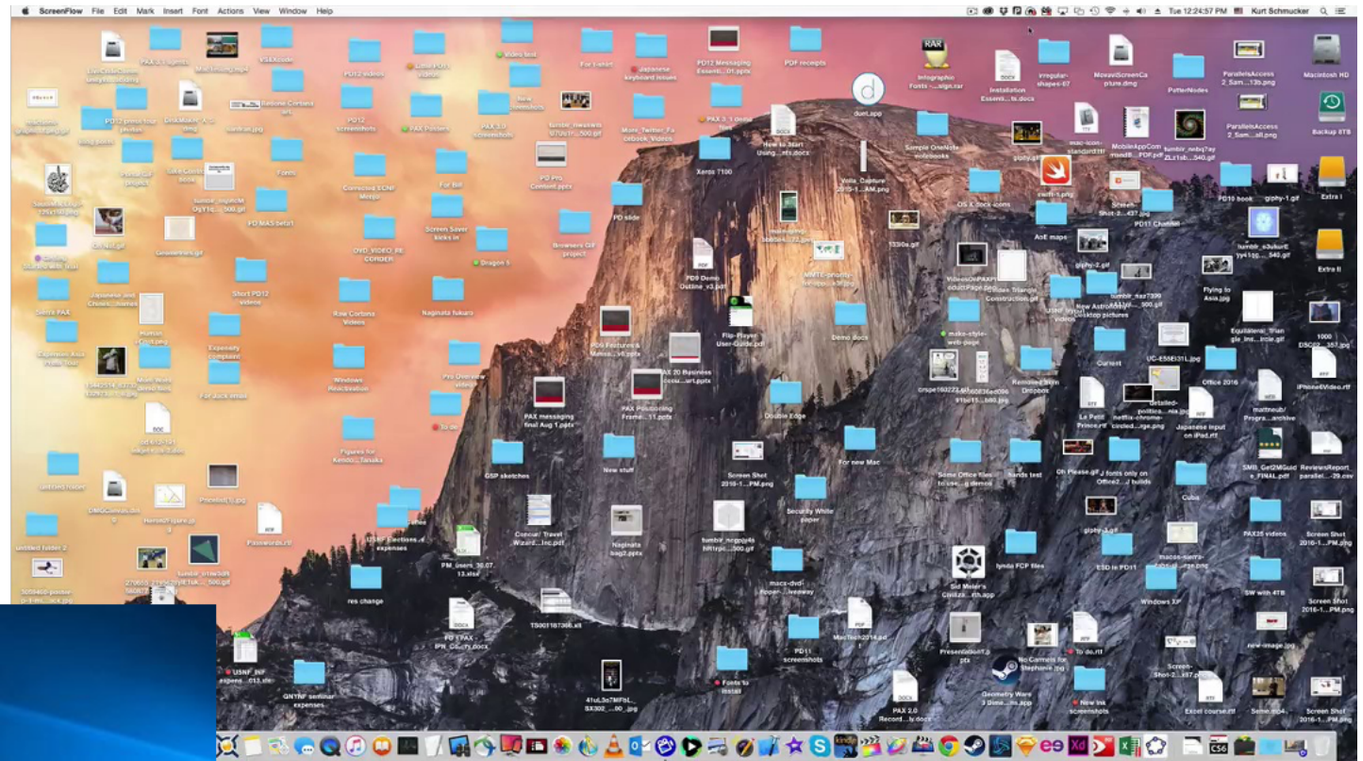
VS Code

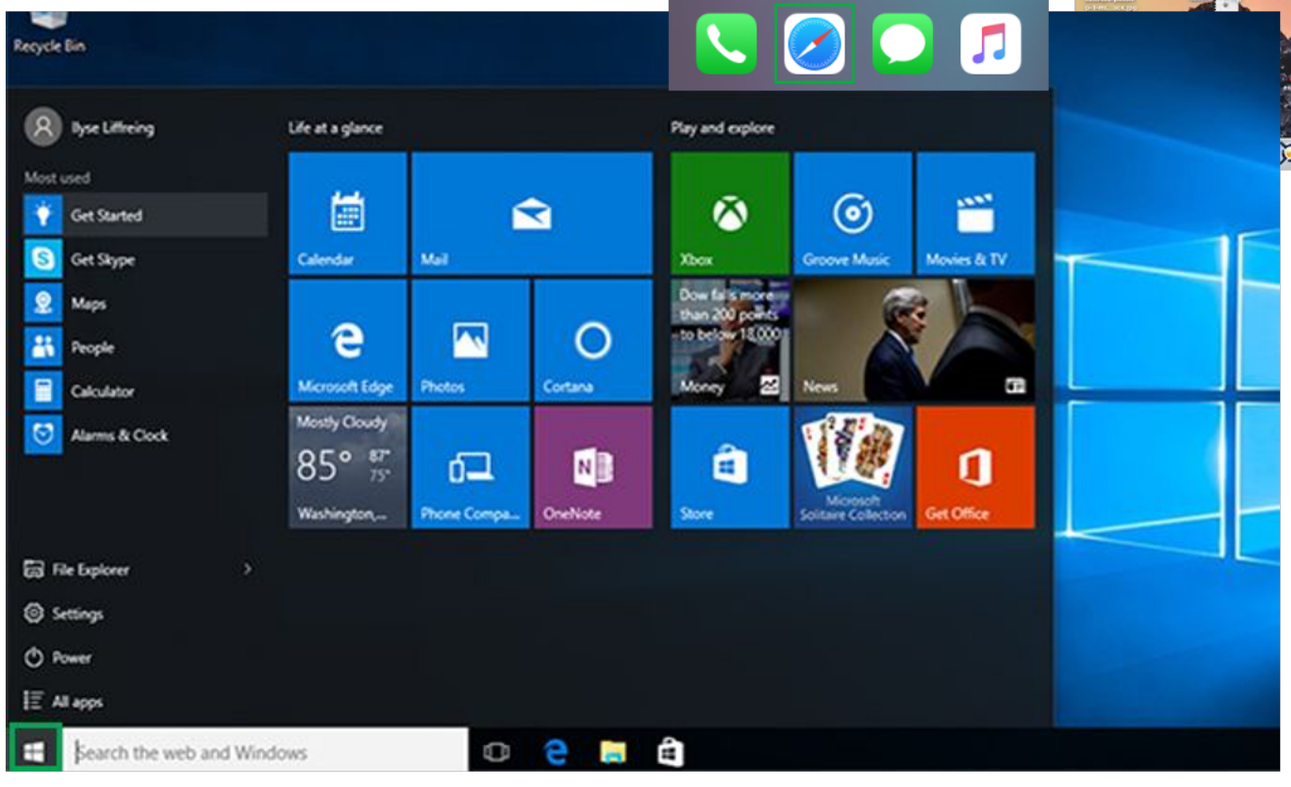
Visual Studio Code

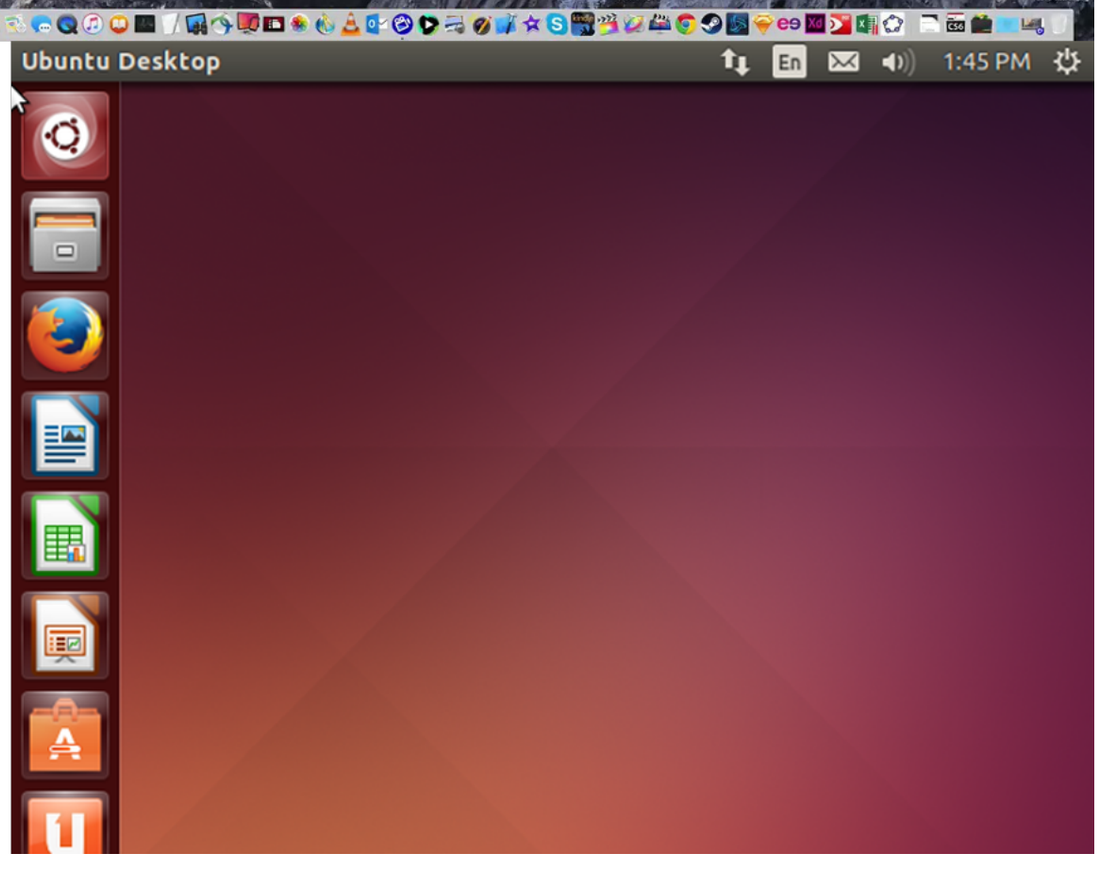


Similar…

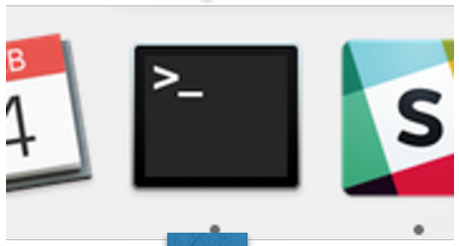

Airtel
9:59 AM
Mail
Calendar
Photos
Camera
Maps
Clock
Weather
News
Home
Notes
Stocks
Reminders
Videos
App Store
iTunes Store
iBooks
Health
Wallet
Settings


ScreenFlow File Edit Mark Insert Font Actions View Window Help
Tue 12:24:57 PM Kurt Schmucker


Recycle Bin
Ilyse Liffreing
Most used
Get Started
Get Skype
Maps
People
Calculator
Alarms & Clock
Life at a glance
Calendar
Mail
Microsoft Edge
Photos
Cortana
Mostly Cloudy
85°
87°
75°
Washington,...
Phone Compa...
OneNote
Play and explore
Xbox
Groove Music
Movies & TV
Money
News
Store
Microsoft Solitaire Collection
Get Office
File Explorer
Settings
Power
All apps
Search the web and Windows

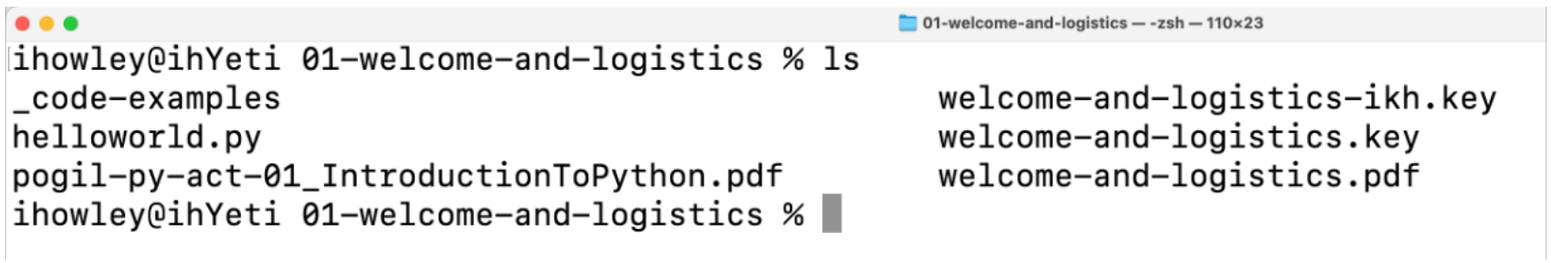

Ubuntu Desktop
En
1:45 PM

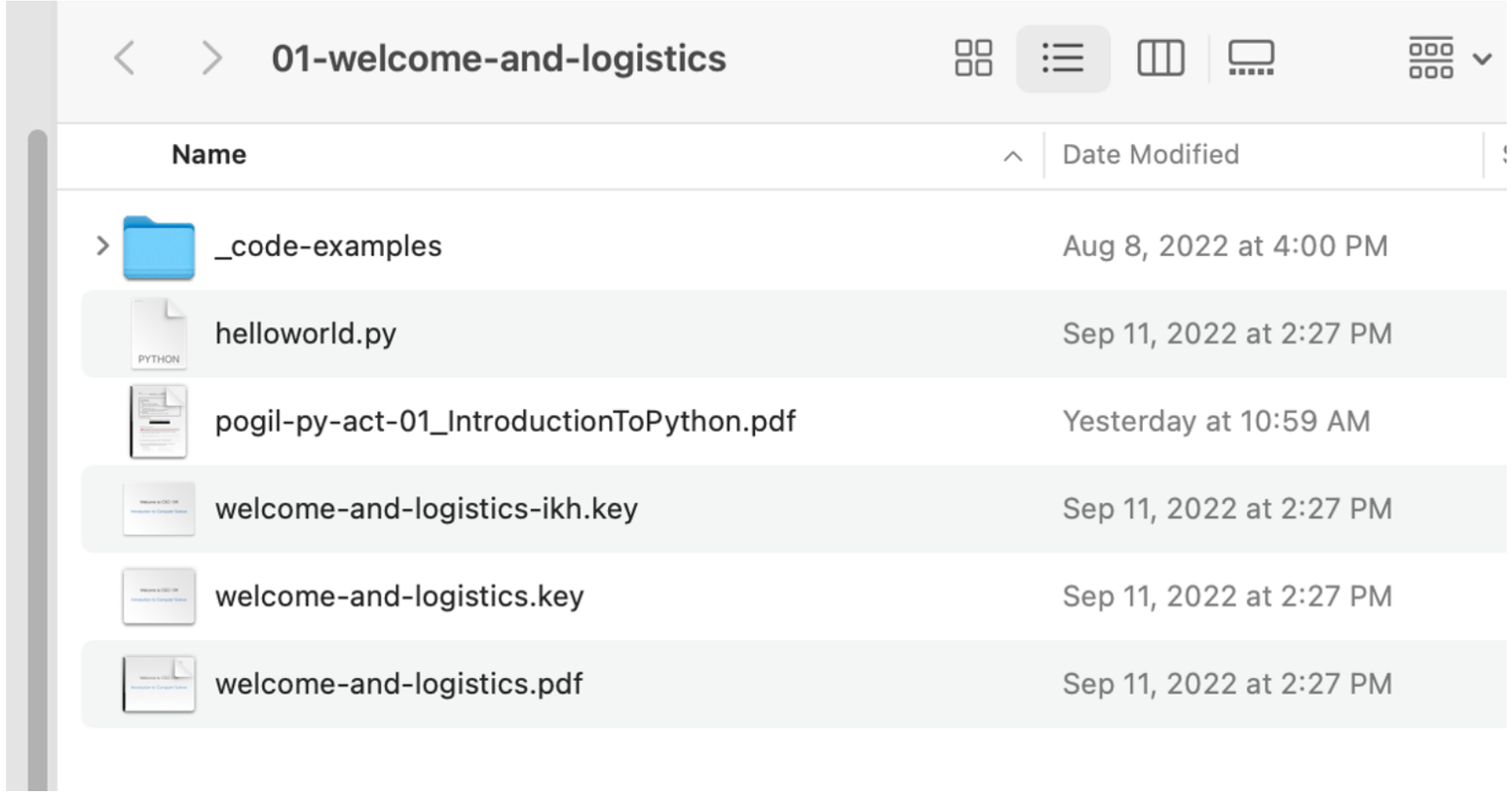
Terminal



Terminal vs. File Explorer

- Terminal is a text-based view of your **directory** structure
  - Like File or Explorer / Windows Explorer, but in text



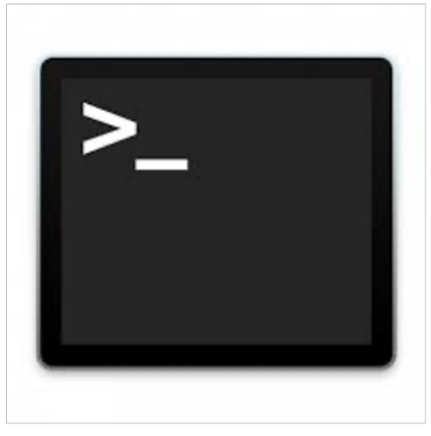


Python

- Written into a **script** and then interpreted by Python
  - `python3 hello.py`

- **Interactive** mode
  - `python3`
  - `>>> print(“hello!”)`

- Terminal & Python are two separate apps!



Terminal Keyboard Shortcuts

- Type a letter or two, then hit 

–> Tab Complete!

- 

—> Cycles through all previous Terminal commands!

Useful Keyboard Shortcuts

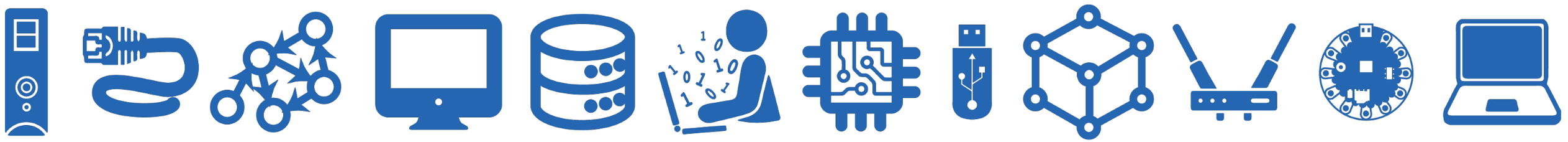
- ⌘ (command) + C → Copy
- ⌘ (command) + X → Cut
- ⌘ (command) + V → Paste
- ⌘ (command) + A → Select All
- ⌘ (command) + S → Save
- ⌘ (command) + R → Refresh
- ⌘ (command) + Z → Undo

- Ctrl+ A → First character of line
- Ctrl + E→ End of line
- Option +  → Right one word
- Option +  → Left one word

Useful Terminal/Unix Commands

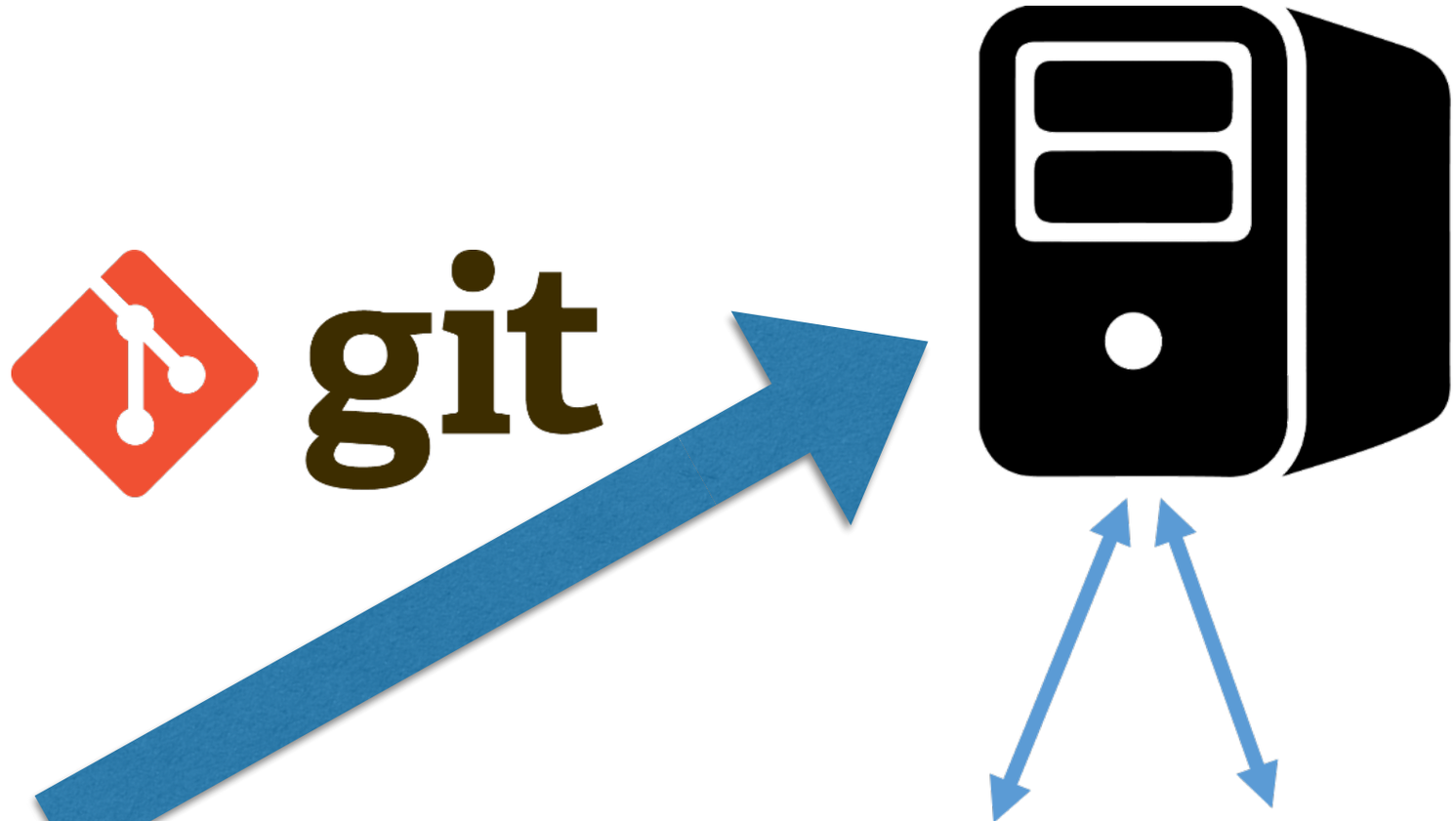
- `pwd` - print working directory
- `mkdir <dir name>` - make new directory (or folder)
- `cd <dir name>` - change directory (like moving into a folder)
- Special directory names in Unix
  - single dot, current directory
  - `..` two dots, parent directory
  - `~` tilde, home directory
- `cd ..` - takes you to the parent directory
- `cd` - takes you "home"
- `ls` - shows contents of current directory

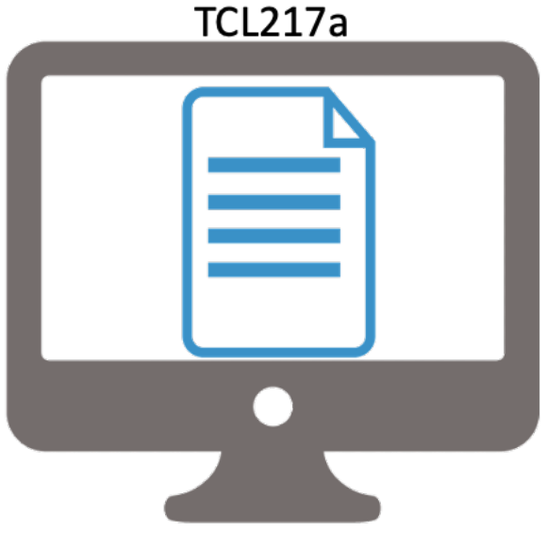
git



Working on Labs

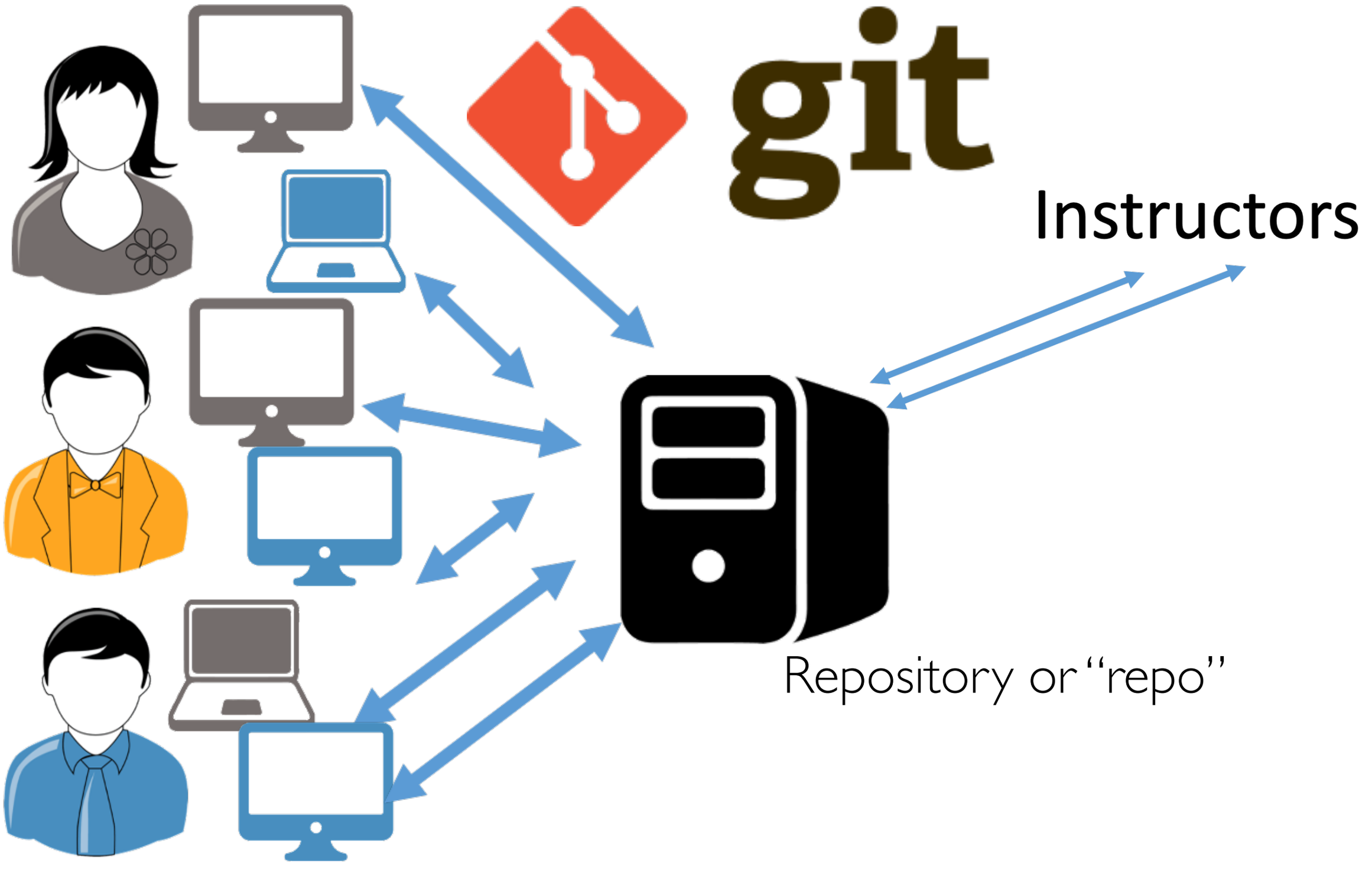
Repository or "repo"





Instructors

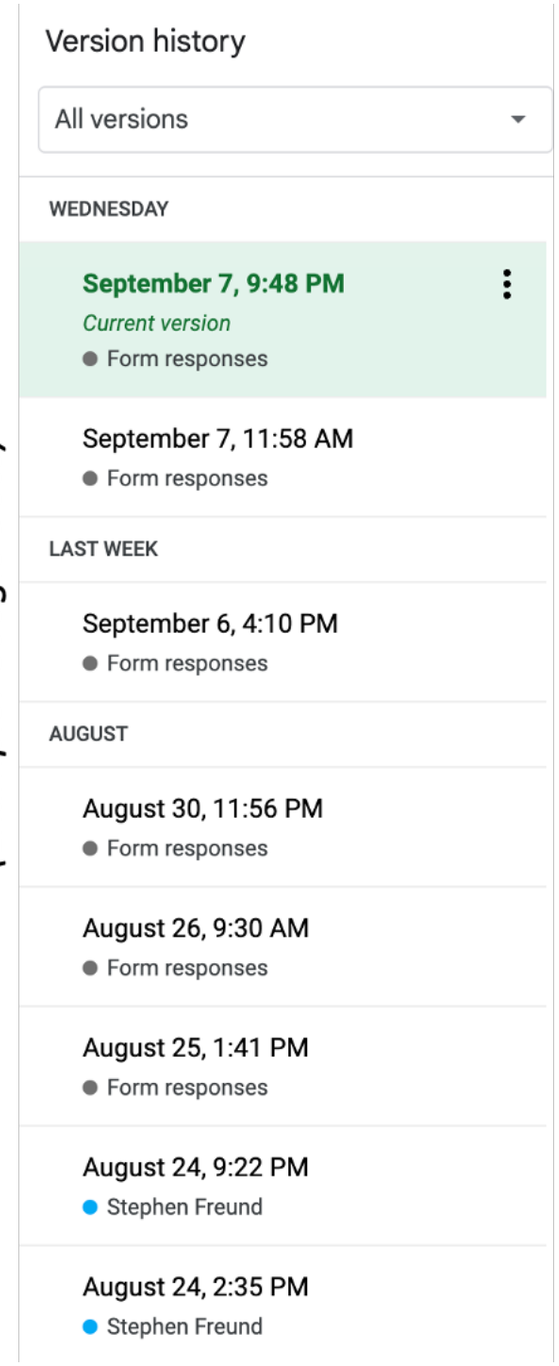
Working on Labs



Why git?!

- Version history!
- Access files from anywhere!
- Great for collaboration!
- Great for maintaining large code bases!

(Example: Google Docs)



`git`:: Get the starter code first

- Copying files —> "I need to get a copy of the lab files my instructor made."

```
git clone https://URL-here.git
```

`git`:: 3 Steps: `git add`

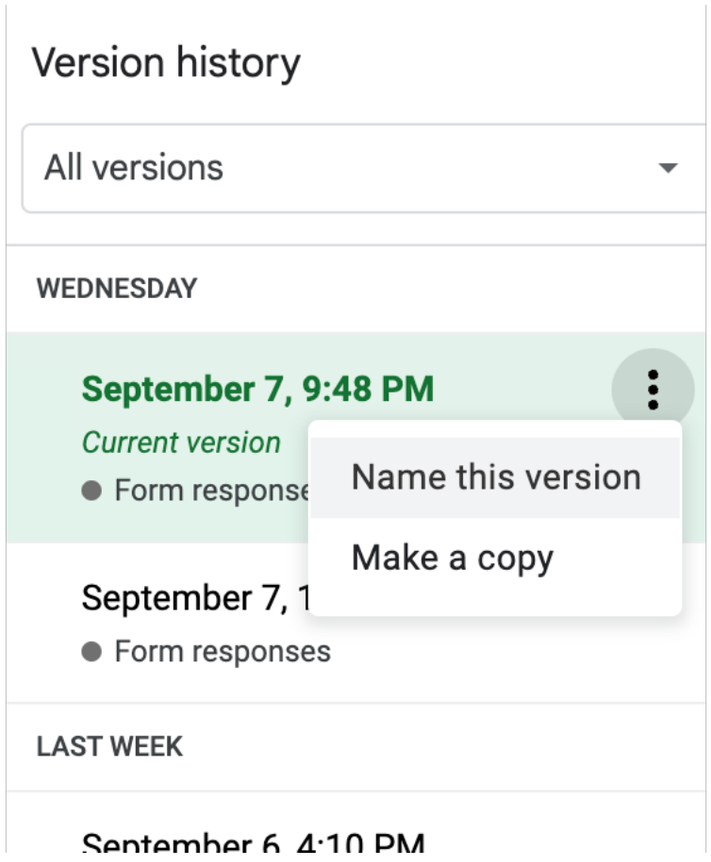
- Staging files —> “I edited this file, I want to include it in the next snapshot of my code .”

```
git add myfilename.py
```

`git`:: 3 Steps: `git commit -m`

- Committing files —> “Take a snapshot of my code that I’ve added so far and assign it a version number.”

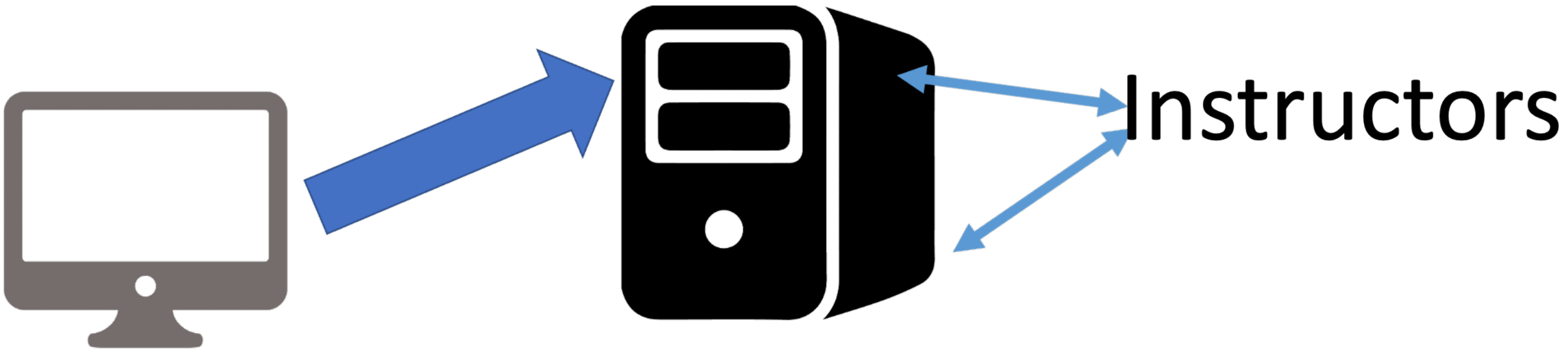
```
git commit -m “Message here”
```



`git`:: 3 Steps: `git push`

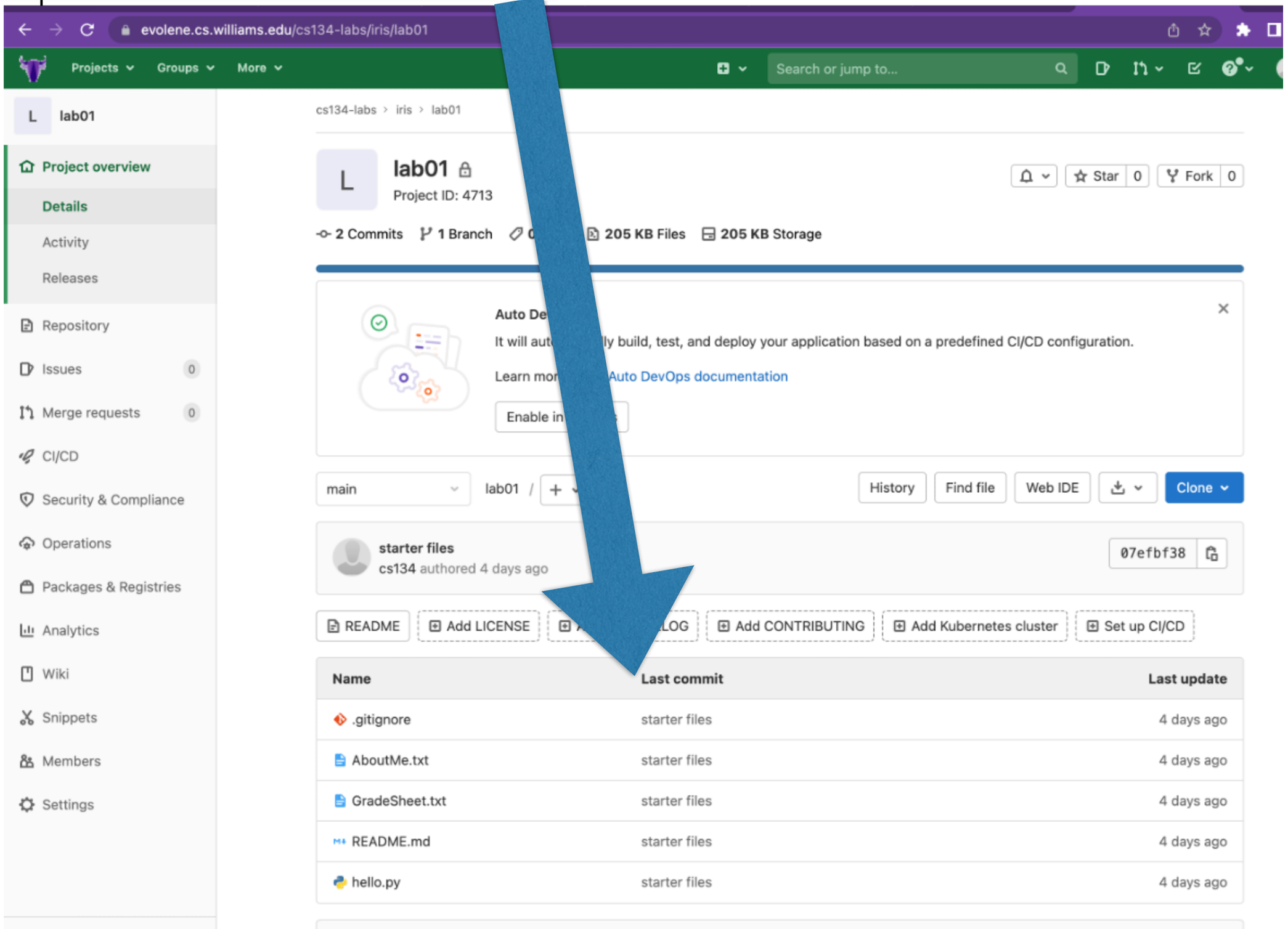
- Pushing files —> “Send all my snapshots up to the server!”

`git push`



`git`:: Check the Website to See Your Edits

- https://evolene.cs.williams.edu



`git`:: 3 Steps

- `git clone`: copy code from server to a **new** machine for the first time. Only run this once for each assignment on each machine!

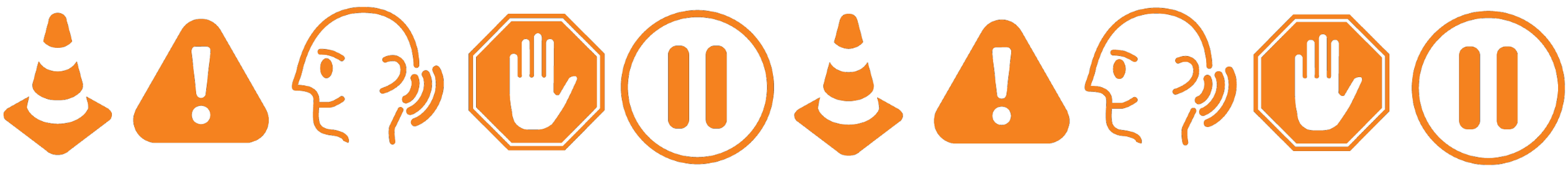
- `git add <files>`: add new or modified files to the next commit (this basically allows you to choose which files you plan to commit)
- `git commit -m "<message>"`: create a local snapshot of the added files (this does **not** copy anything back to the server!)
- `git push`: copy changes from your machine back to our server

- `git pull`: copy latest version of code from our server to your local machine (this can only be done **after** you have run `git clone` on this machine)
- `git commit -am "<message>"`: commits an already added file (a shortcut)

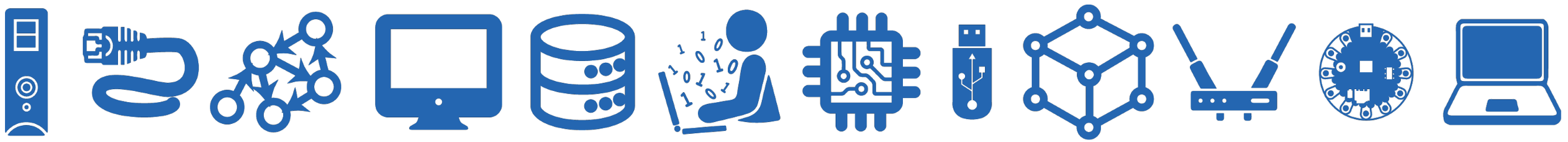
Notes

- We use `git` commands in Terminal
- You need your CS account to log-in to evolene (the gitlab server with the lab starter files)
- Always `git add/commit/push` before you leave lab!

- Lab instructions are on the course Website
- You'll *submit* your assignments through GradeScope.



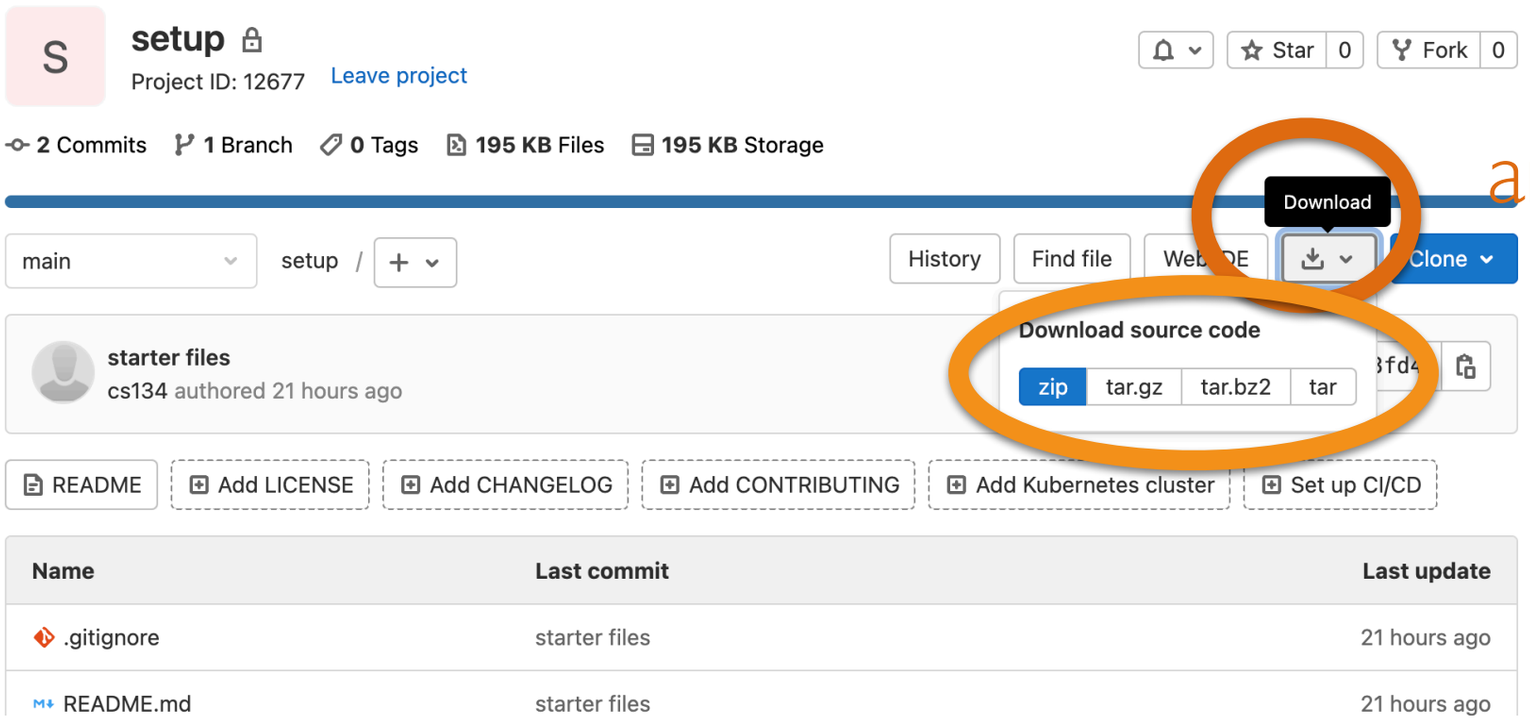
Submitting Your Lab Assignment Through Gradescope



Gradescope

- First, you'll need to go to evolene/gitlab and download your lab assignment as a .zip file:

  (This screenshot is for a different lab, but the buttons should be the same!)



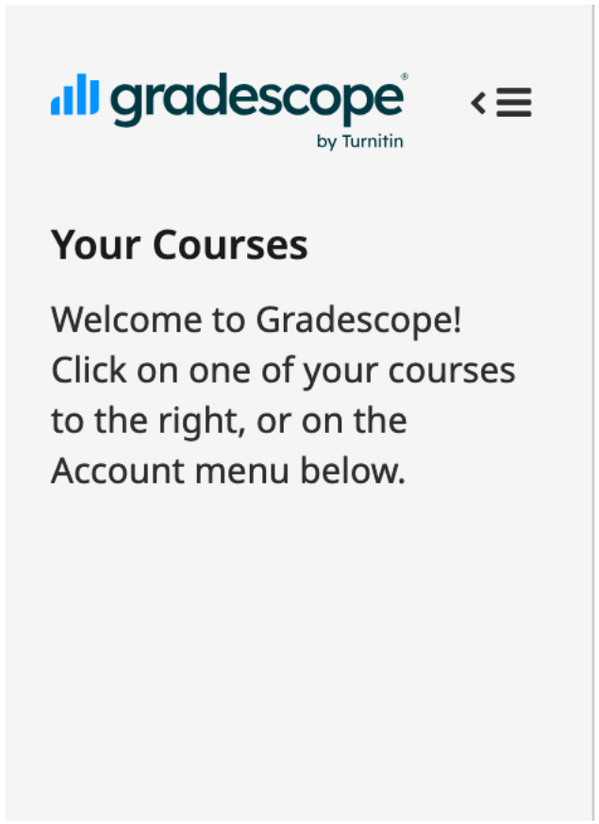
Select the downward arrow Download button

Then select 'zip' as the Download source code option

Remember where you've downloaded the **`lab01-main.zip`** file!

Gradescope

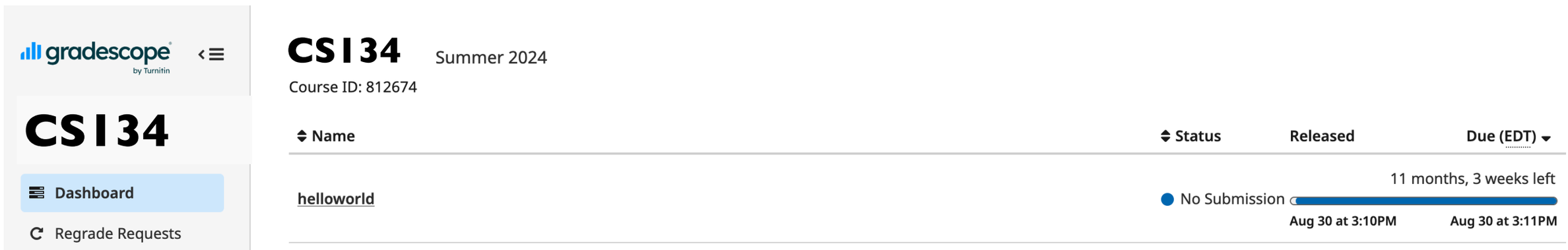
- Use the Gradescope invitation email you receive to create an account associated with CS134
- From the Gradescope Dashboard, select CS134



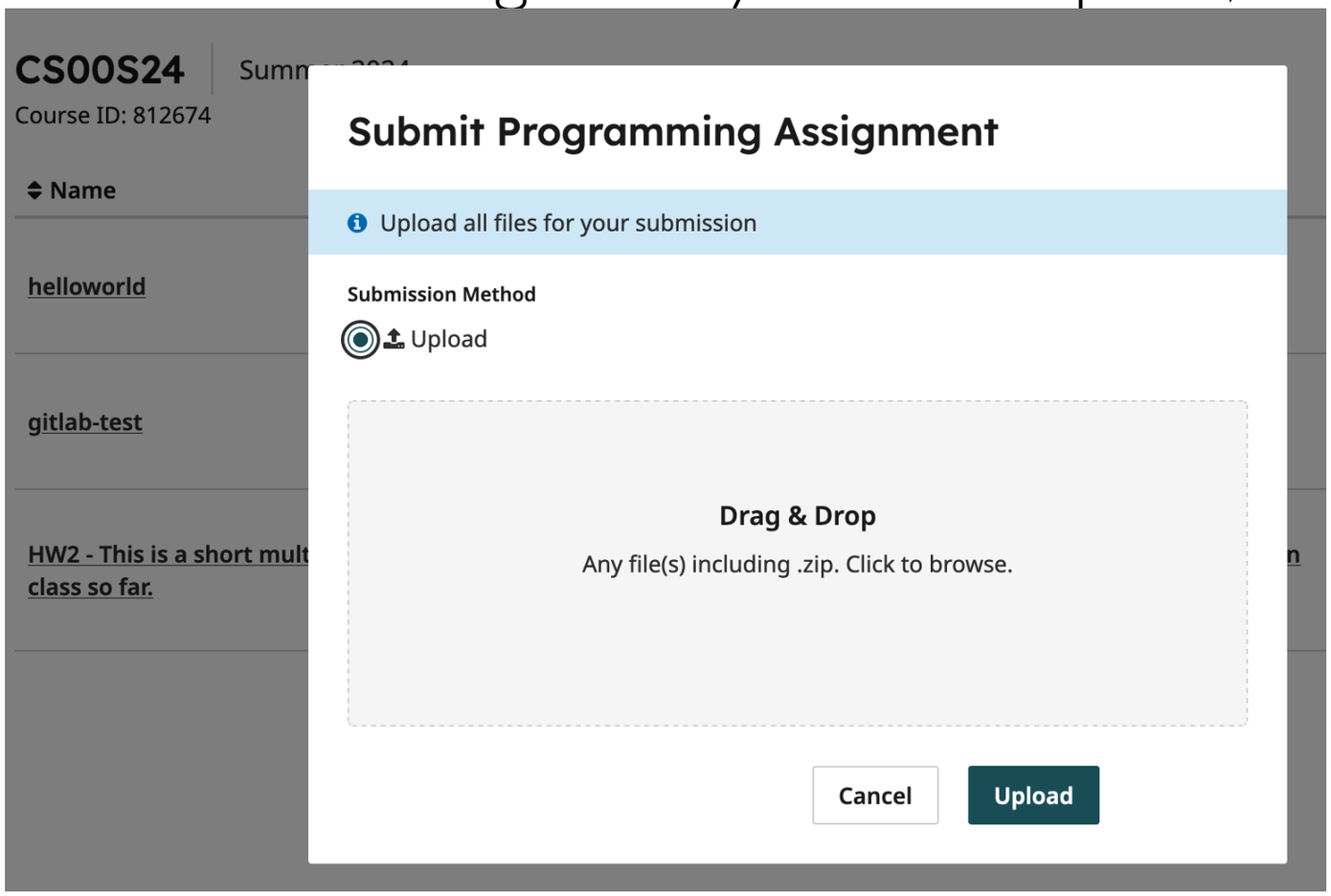
Your Dashboard will look *something* like this!

Gradescope

- Select the appropriate assignment (for this week, lab 1)



- You should navigate to your Lab .zip file, and upload it



Remember to press 'Upload!

The end!