Lecture 3: Processing Linguistic Data, Git/GitHub

LING 1340/2340: Data Science for Linguists

Na-Rae Han

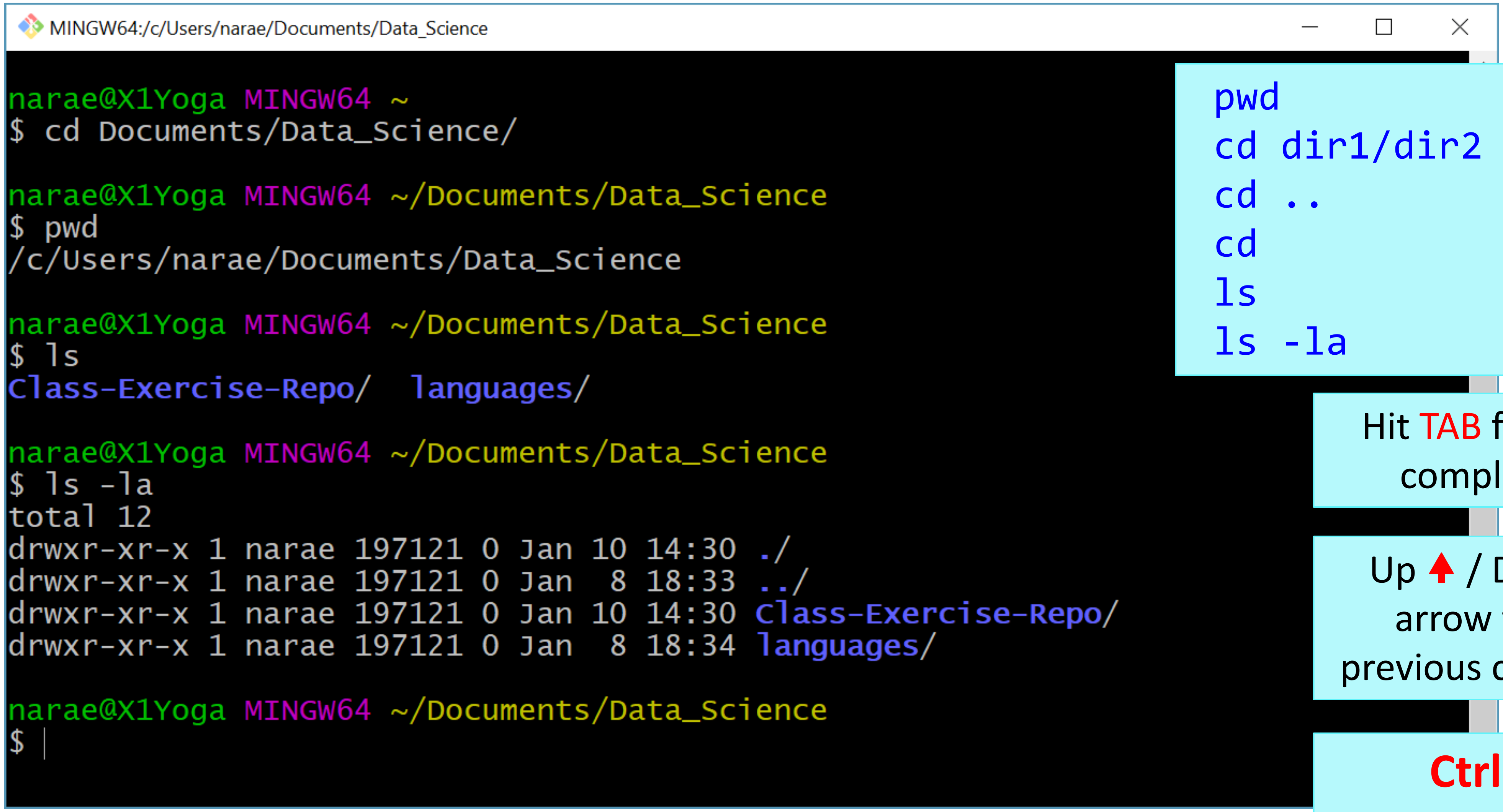
Objectives

- To-do 1: What linguistic data did you find?
- HW1: What did you process?
- GitHub: completing the fork triangle
- DataCamp tutorials

- Tools:
  - Git and GitHub
  - Jupyter Notebook

You should be taking NOTES!

First thing to do every class



MINGW64:/c/Users/narae/Documents/Data_Science

```
narae@X1Yoga MINGW64 ~
$ cd Documents/Data_Science/

narae@X1Yoga MINGW64 ~/Documents/Data_Science
$ pwd
/c/Users/narae/Documents/Data_Science

narae@X1Yoga MINGW64 ~/Documents/Data_Science
$ ls
Class-Exercise-Repo/  languages/

narae@X1Yoga MINGW64 ~/Documents/Data_Science
$ ls -la
total 12
drwxr-xr-x 1 narae 197121 0 Jan 10 14:30 ./
drwxr-xr-x 1 narae 197121 0 Jan  8 18:33 ../
drwxr-xr-x 1 narae 197121 0 Jan 10 14:30 Class-Exercise-Repo/
drwxr-xr-x 1 narae 197121 0 Jan  8 18:34 languages/

narae@X1Yoga MINGW64 ~/Documents/Data_Science
$
```

```
pwd
cd dir1/dir2
cd ..
cd
ls
ls -la
```

Hit TAB for auto-completion.

Up ↑ / Down ↓ arrow to use previous command

Ctrl + c to cancel

To-do #1

- What linguistic data sets did you look at?
  - Corpus data?
  - Non-corpus data?

- What makes a dataset a corpus?

Back to Class-Exercise-Repo

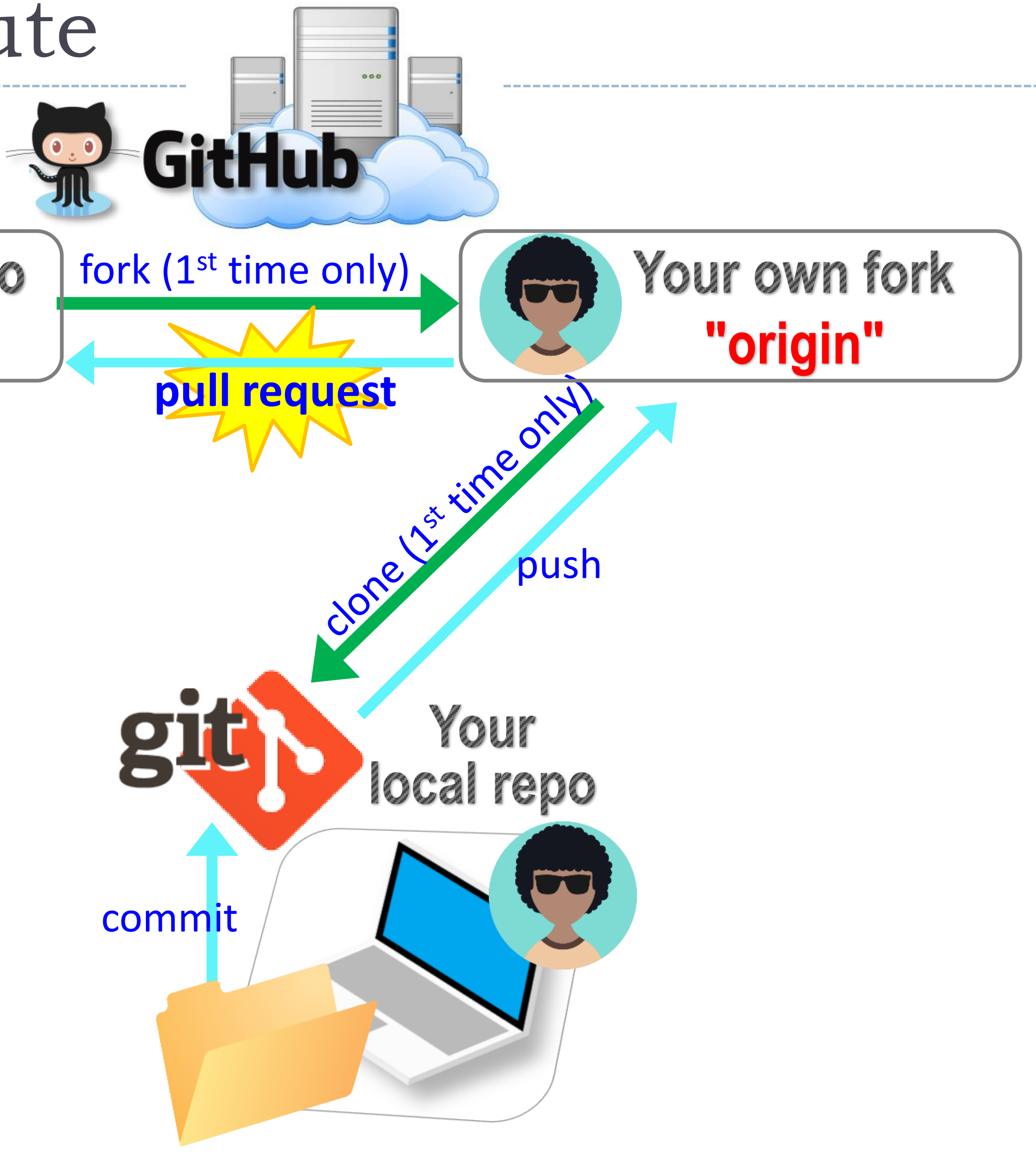
https://github.com/Data-Science-for-Linguists-2021/Class-Exercise-Repo

- Todo1
  - Your To-do 1 submissions

- Lots of files -- I have merged in everyone's contributions.
- **But! Your own fork does not have those.**

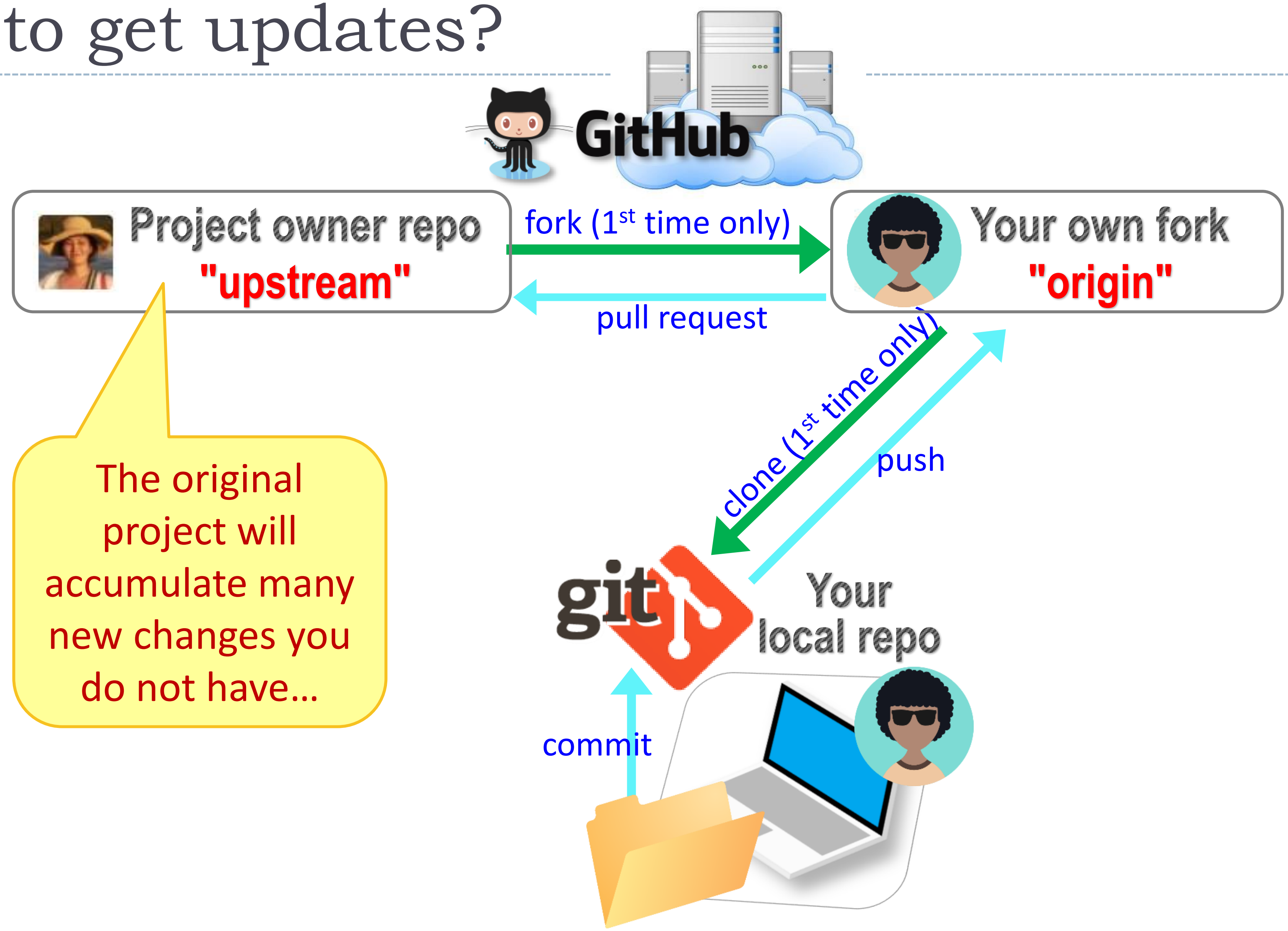
*Avatar icons by FLATICON

Offering to contribute



How to get updates?

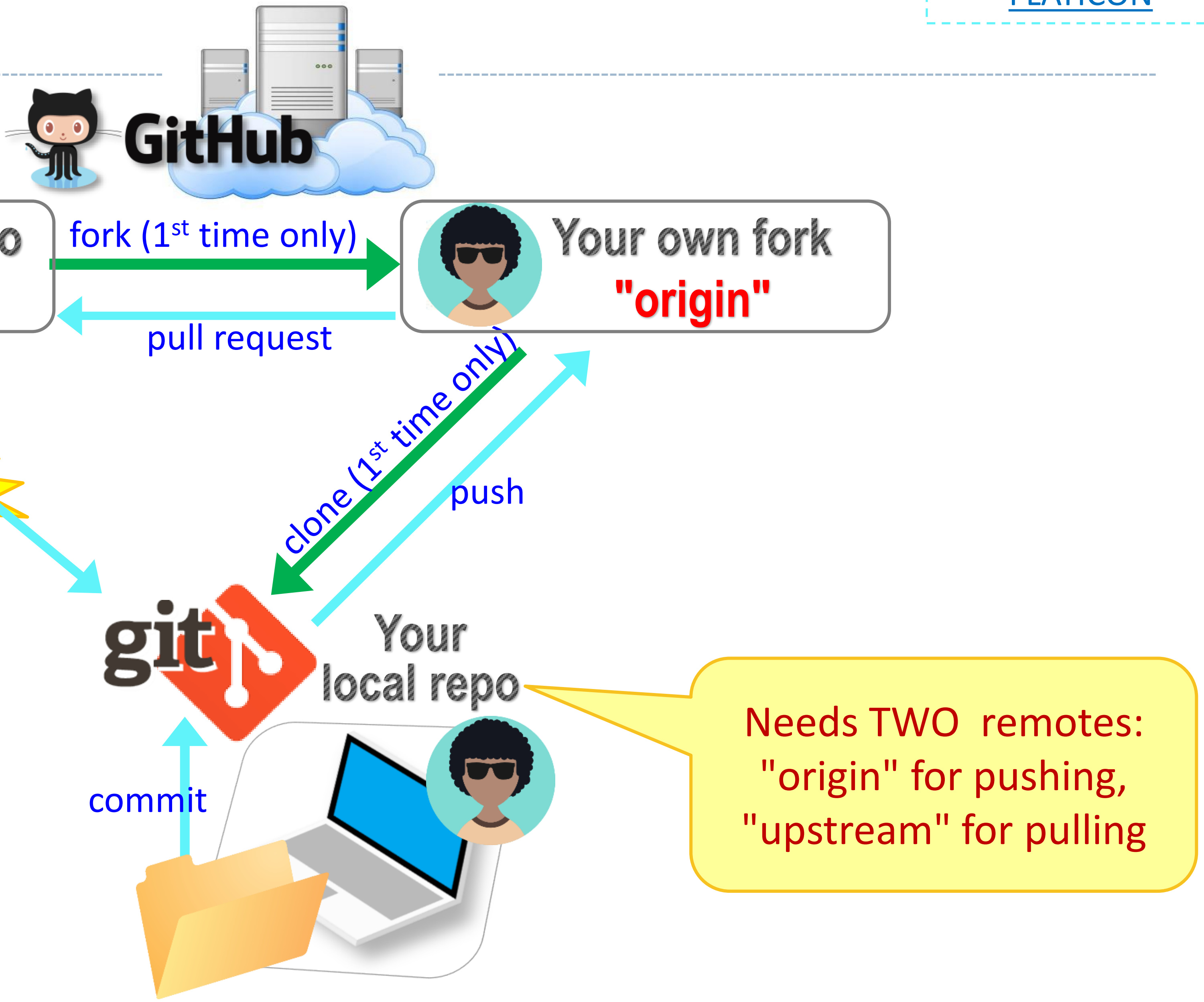
*Avatar icons by FLATICON



The original project will accumulate many new changes you do not have…

The fork triangle, complete

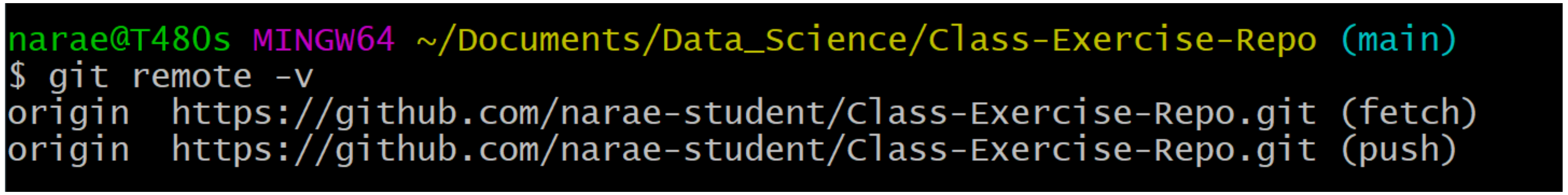
*Avatar icons by FLATICON



- Solution: you should **pull** from "**upstream**".

Keeping your fork up-to-date

- The original repo ("upstream") will keep changing.
  - How to keep your copies (GitHub fork and local repo) up-to-date?
- Cloning already configured your GitHub fork as "origin":

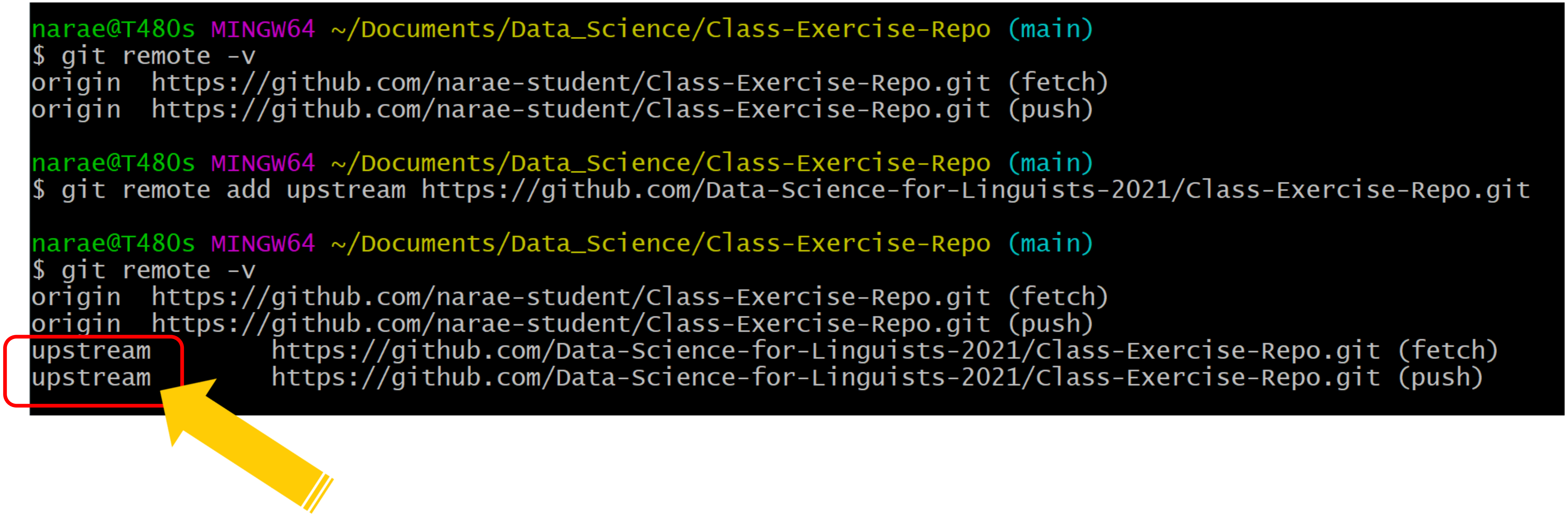


```
narae@T480s MINGW64 ~/Documents/Data_Science/Class-Exercise-Repo (main)
$ git remote -v
origin  https://github.com/narae-student/Class-Exercise-Repo.git (fetch)
origin  https://github.com/narae-student/Class-Exercise-Repo.git (push)
```

- Configure the original repo as another remote: "upstream"
  - `git remote add upstream <GitHub-repo-URL.git>`
- When it's time to sync, pull from upstream:
  - `git pull upstream main`
- Pushing should be done to your GitHub fork ("origin").
  - `git push origin main`

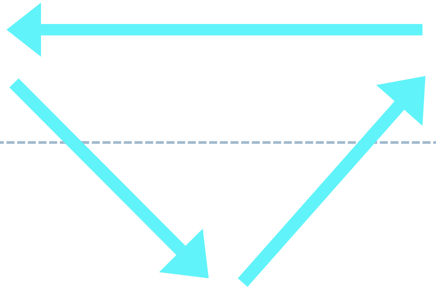
You might be able to leave out "origin main".

Two remotes: "origin", "upstream"



```
narae@T480s MINGW64 ~/Documents/Data_Science/Class-Exercise-Repo (main)
$ git remote -v
origin  https://github.com/narae-student/Class-Exercise-Repo.git (fetch)
origin  https://github.com/narae-student/Class-Exercise-Repo.git (push)

narae@T480s MINGW64 ~/Documents/Data_Science/Class-Exercise-Repo (main)
$ git remote add upstream https://github.com/Data-Science-for-Linguists-2021/Class-Exercise-Repo.git

narae@T480s MINGW64 ~/Documents/Data_Science/Class-Exercise-Repo (main)
$ git remote -v
origin  https://github.com/narae-student/Class-Exercise-Repo.git (fetch)
origin  https://github.com/narae-student/Class-Exercise-Repo.git (push)
upstream        https://github.com/Data-Science-for-Linguists-2021/Class-Exercise-Repo.git (fetch)
upstream        https://github.com/Data-Science-for-Linguists-2021/Class-Exercise-Repo.git (push)
```

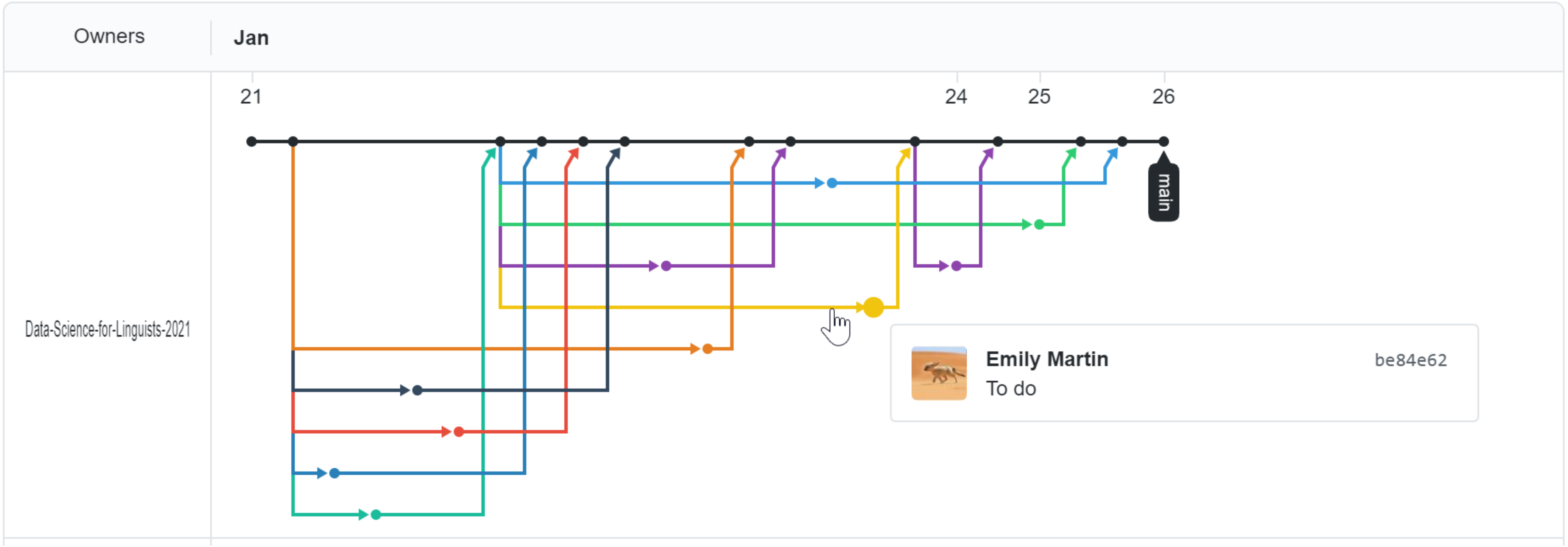
The fork triangle: workflow



- On your **laptop**
  1. Check your local repo's status: `git status`. Get it to a clean state.
  2. Pull from "upstream", syncing your local repo: `git pull upstream main`. Your local repo now has all latest changes.
     - If there is a merge conflict, you will need to resolve it. (fingers crossed)
  3. Do your work! New files, edits, etc.
  4. Do your usual local Git routine: `git add` and `git commit`.
  5. Push new versions to your own GitHub fork ("origin"): `git push origin main`
- On **GitHub**
  1. Check your forked repo. It should have your new work.
  2. Create a **pull request** for the original repo ("upstream") owner.
  3. Give it some time, and check back on the status of your pull request.

Many forks and merges

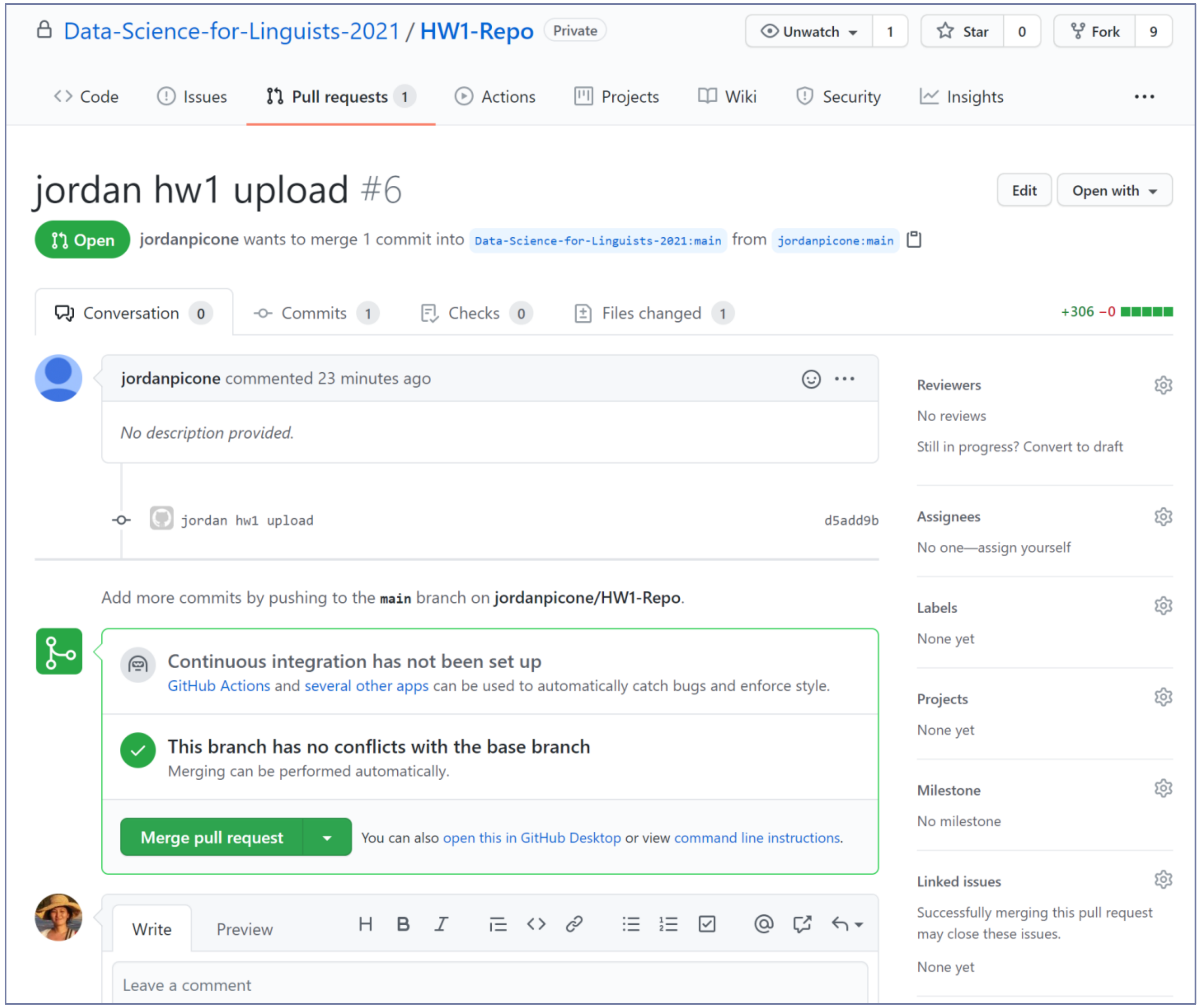
- https://github.com/Data-Science-for-Linguists-2021/Class-Exercise-Repo/network



Emily Martin be84e62
To do

HW1: processing pull request, merging

- With everyone working on their own files/folders, merging is conflict-free:



Data-Science-for-Linguists-2021 / HW1-Repo Private

Unwatch 1 Star 0 Fork 9

Code Issues Pull requests 1 Actions Projects Wiki Security Insights

jordan hw1 upload #6

Edit Open with

Open jordanpicone wants to merge 1 commit into `Data-Science-for-Linguists-2021:main` from `jordanpicone:main`

Conversation 0 Commits 1 Checks 0 Files changed 1 +306 −0

jordanpicone commented 23 minutes ago

No description provided.

`jordan hw1 upload` d5add9b

Add more commits by pushing to the **main** branch on **jordanpicone/HW1-Repo**.

Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request — You can also open this in GitHub Desktop or view command line instructions.

Write Preview

Leave a comment

Reviewers
No reviews
Still in progress? Convert to draft

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked issues
Successfully merging this pull request may close these issues.
None yet

HW1: sync your HW1-Repo

1. Configure "upstream" remote:

```
git remote add upstream https://github.com/Data-Science-for-Linguists-2021/HW1-Repo.git
```

2. Pull from upstream:

```
git pull upstream main
```

3. Push to your GitHub fork:

```
git push origin main
```

Everyone's repos are synced.

Now, everyone has everyone's homework submission.

HW1: Review

- What did you all work on?
- You wish list: what new skills would you like to learn?
- What is the `.gitignore` file?
- Why did we exclude data files from Git?
- What is up with that "your_file_here.txt" blank file? What is `git rm`?
- Jupyter Notebook: do you like it?

Git and GitHub are complicated.

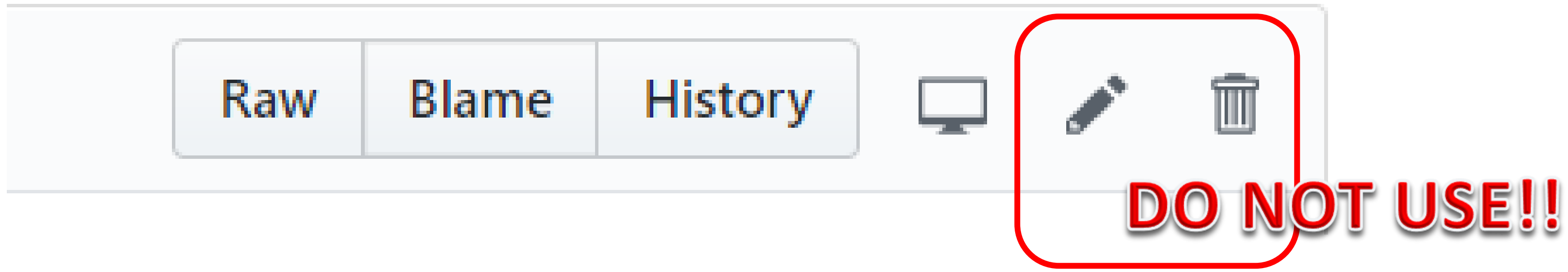
- They are powerful tools.
- There are a lot of abstract, high-level concepts involved.
- Concepts do not make sense before you get hands-on.
- You cannot get hands-on without the right context.

🡐 We will learn slowly, learning various pieces as we go.

🡐 You need to be patient, careful and methodical. Make sure you don't rush, and follow instructions.

Git and GitHub are complicated.

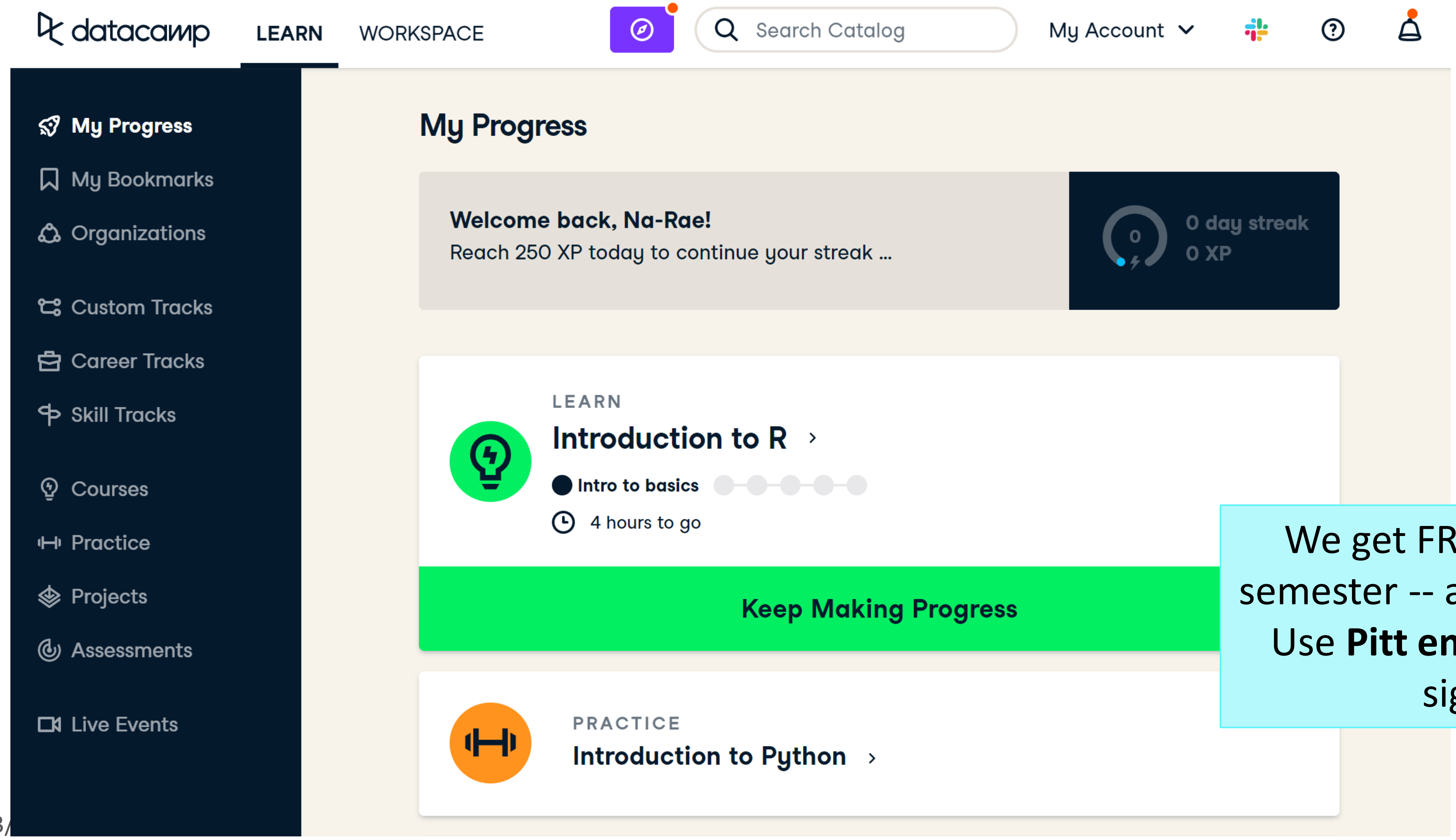
- We will follow some ground rules.
- DO NOT EDIT A REPOSITORY'S CONTENT THROUGH GITHUB.



- Don't accidentally commit a file! Be mindful of what you add. Avoid using:
  - `git add .`
  - `git add *`
- For now, do not **delete** or **re-name** any previously committed file.
  - If you must: use `git rm` and `git mv`.

Course Group on DataCamp

- Video-based, interactive tutorials



My Progress

Welcome back, Na-Rae!
Reach 250 XP today to continue your streak ...

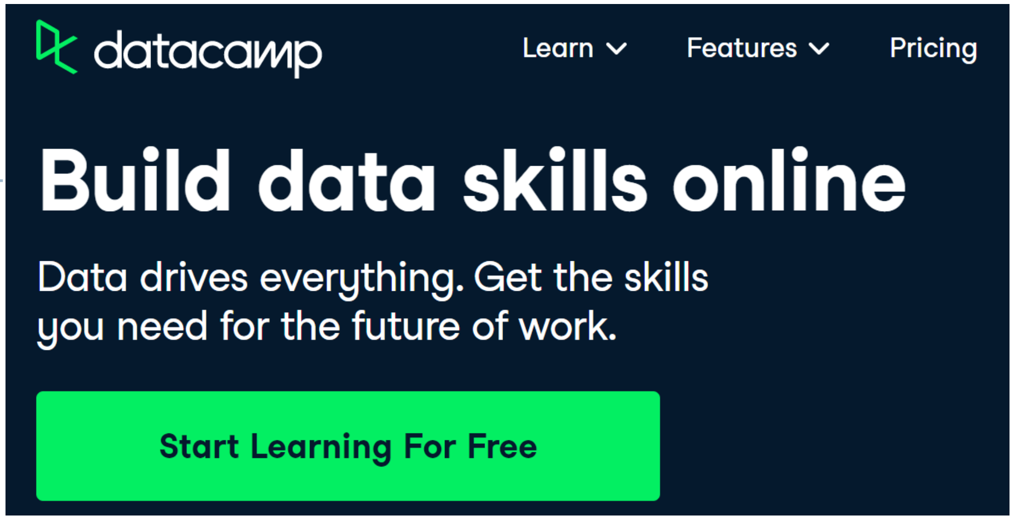
0 day streak
0 XP

LEARN
Introduction to R ›
Intro to basics
4 hours to go

PRACTICE
Introduction to Python ›

We get FREE access this semester -- all you can learn! Use **Pitt email address** to sign up.

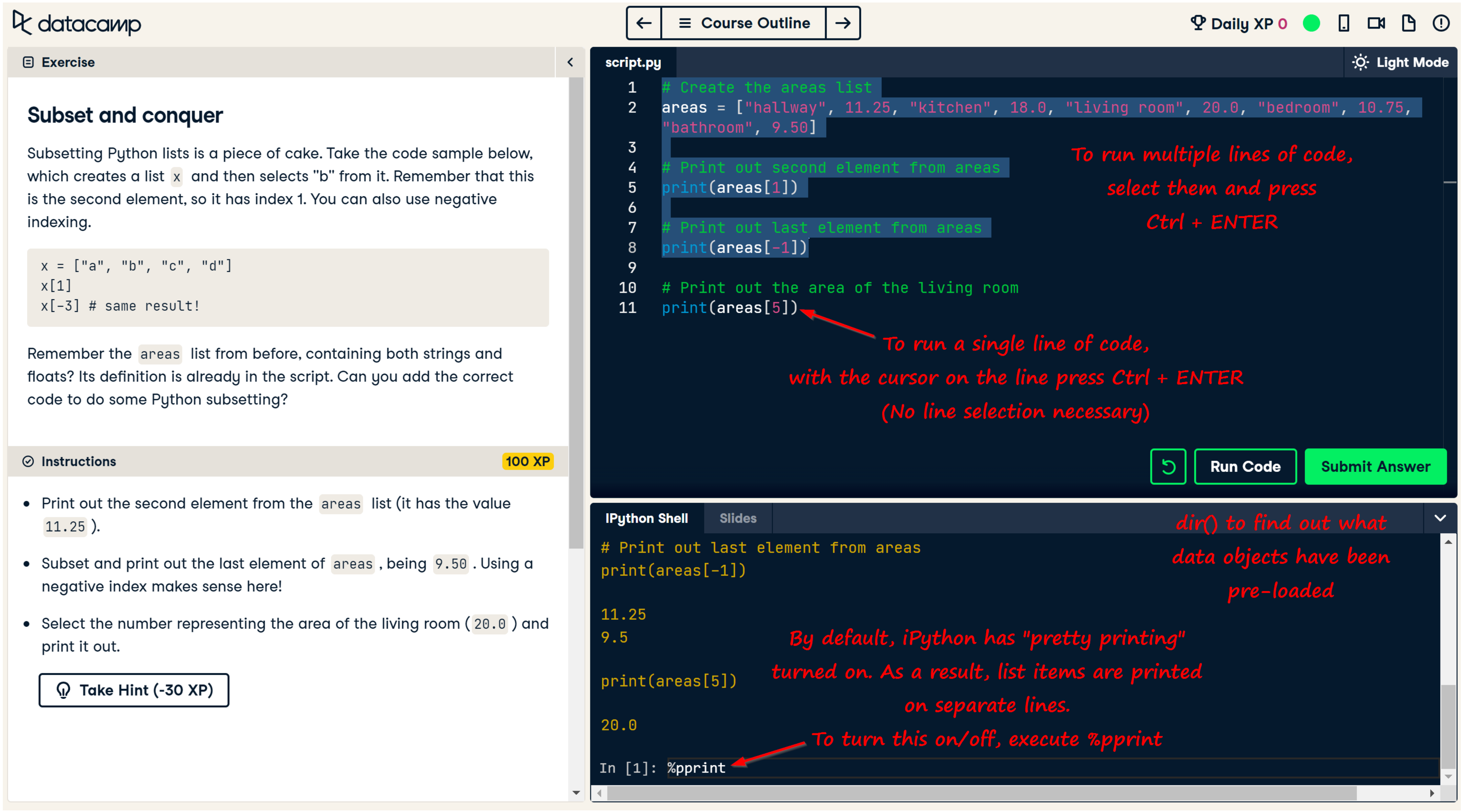
How to use DataCamp



- Topics for the next couple of weeks:
  - numpy library
  - pandas library
  - visualization libraries such as `matplotlib`
- The video tutorials are linked as "assignments"
  - Great learning resource, but not mandatory.
  - They *complement* the textbook nicely.
- Online exercise interface needs some getting used to.

  ➔ next slide

https://campus.datacamp.com/courses/intro-to-python-for-data-science/chapter-2-python-lists?ex=7



Subset and conquer

Subsetting Python lists is a piece of cake. Take the code sample below, which creates a list `x` and then selects "b" from it. Remember that this is the second element, so it has index 1. You can also use negative indexing.

```
x = ["a", "b", "c", "d"]
x[1]
x[-3] # same result!
```

Remember the `areas` list from before, containing both strings and floats? Its definition is already in the script. Can you add the correct code to do some Python subsetting?

Instructions

- Print out the second element from the `areas` list (it has the value `11.25`).
- Subset and print out the last element of `areas`, being `9.50`. Using a negative index makes sense here!
- Select the number representing the area of the living room (`20.0`) and print it out.

```
# Create the areas list
areas = ["hallway", 11.25, "kitchen", 18.0, "living room", 20.0, "bedroom", 10.75,
"bathroom", 9.50]

# Print out second element from areas
print(areas[1])

# Print out last element from areas
print(areas[-1])

# Print out the area of the living room
print(areas[5])
```

To run multiple lines of code, select them and press Ctrl + ENTER

To run a single line of code, with the cursor on the line press Ctrl + ENTER (No line selection necessary)

```
# Print out last element from areas
print(areas[-1])

11.25
9.5

print(areas[5])

20.0

In [1]: %pprint
```

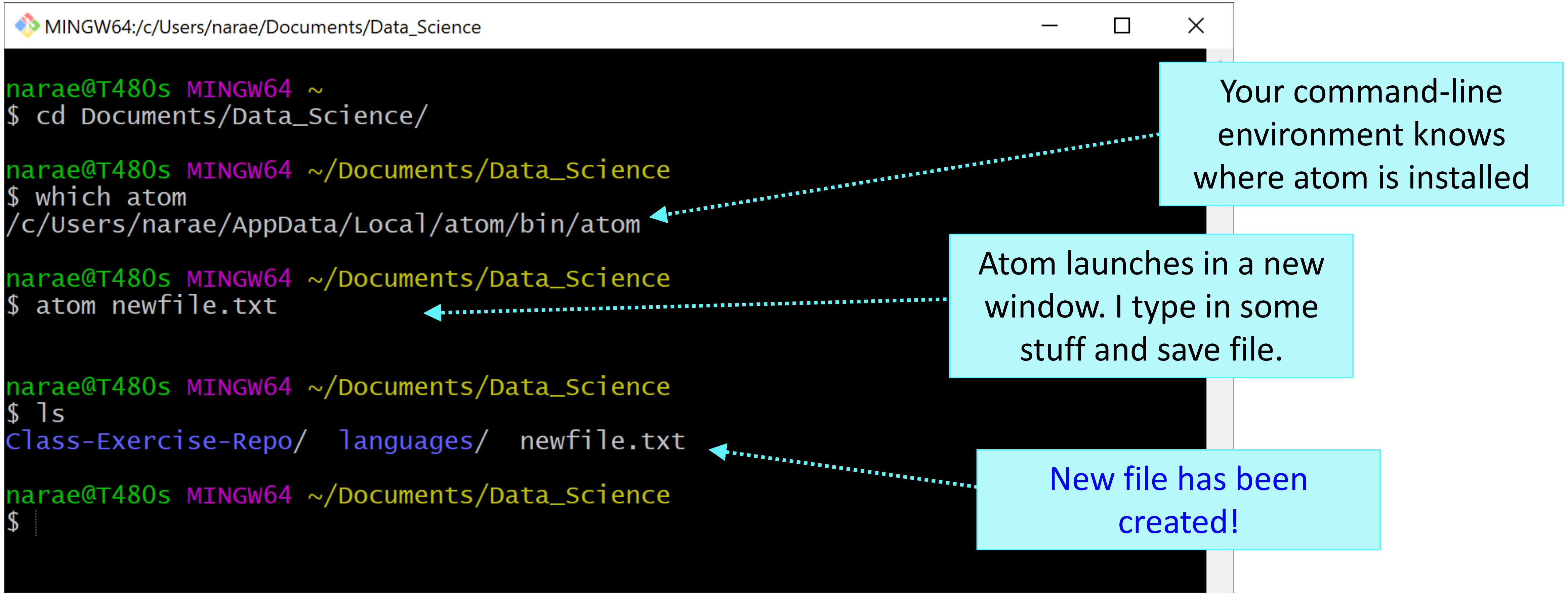
dir() to find out what data objects have been pre-loaded

By default, iPython has "pretty printing" turned on. As a result, list items are printed on separate lines.

To turn this on/off, execute %pprint

Your text editor in shell

- You should be able to launch your text editor from shell and create a new text file in the directory.



MINGW64:/c/Users/narae/Documents/Data_Science

```
narae@T480s MINGW64 ~
$ cd Documents/Data_Science/

narae@T480s MINGW64 ~/Documents/Data_Science
$ which atom
/c/Users/narae/AppData/Local/atom/bin/atom

narae@T480s MINGW64 ~/Documents/Data_Science
$ atom newfile.txt


narae@T480s MINGW64 ~/Documents/Data_Science
$ ls
Class-Exercise-Repo/  languages/  newfile.txt

narae@T480s MINGW64 ~/Documents/Data_Science
$
```

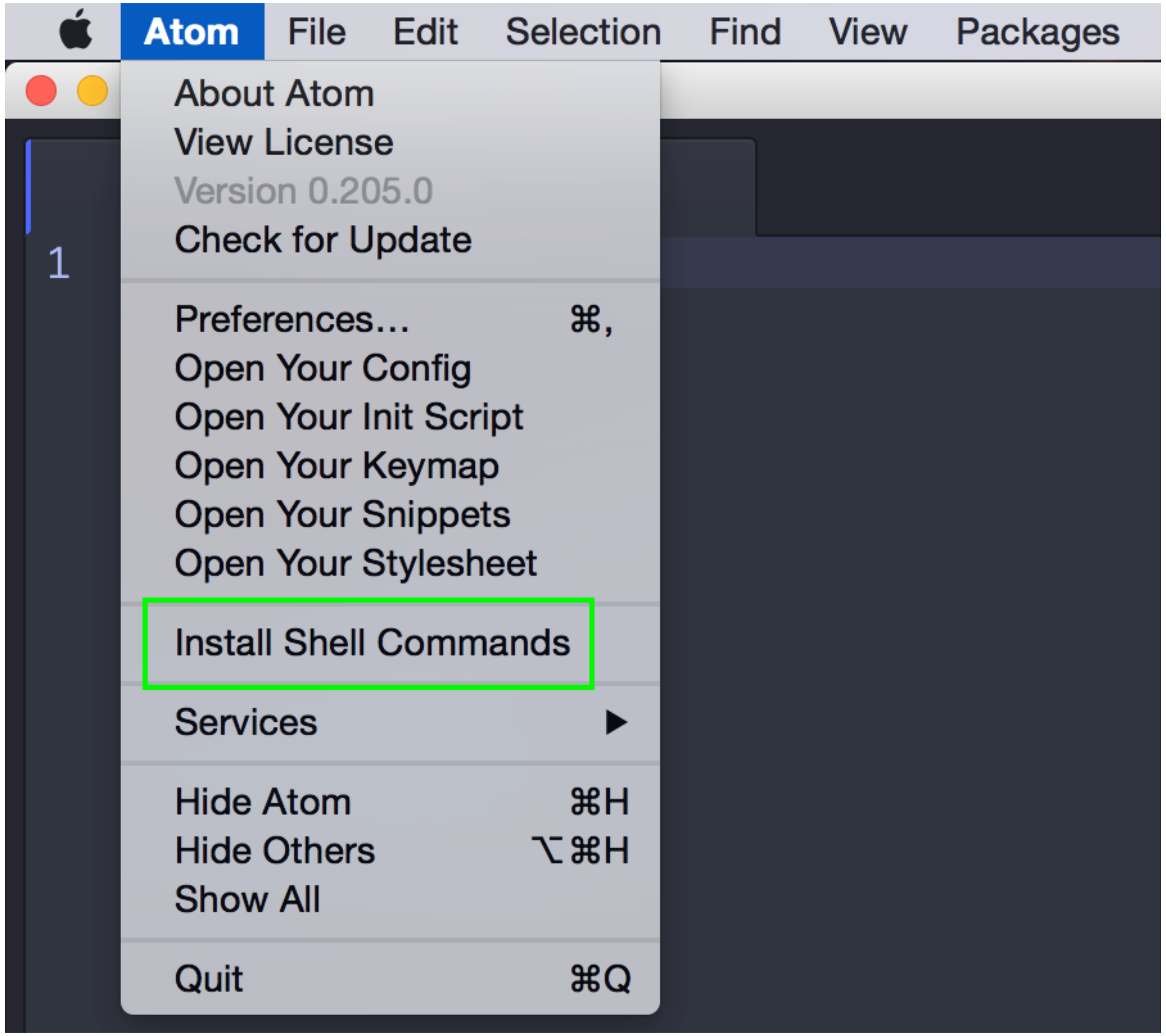
Your command-line environment knows where atom is installed

Atom launches in a new window. I type in some stuff and save file.

New file has been created!

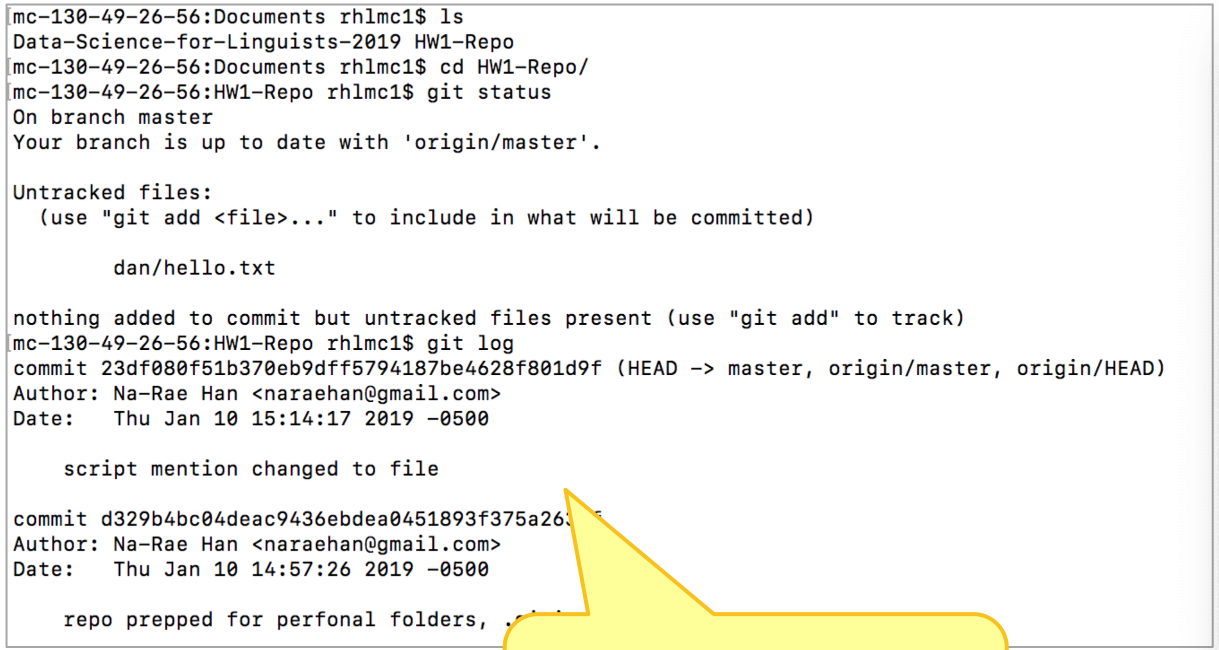
Mac users: configure Atom for shell

- https://stackoverflow.com/questions/22390709/how-to-open-atom-editor-from-command-line-in-os-x
  - "Install Shell Commands"
  - After this, you can launch atom directly from your Terminal (bash or zsh shell).



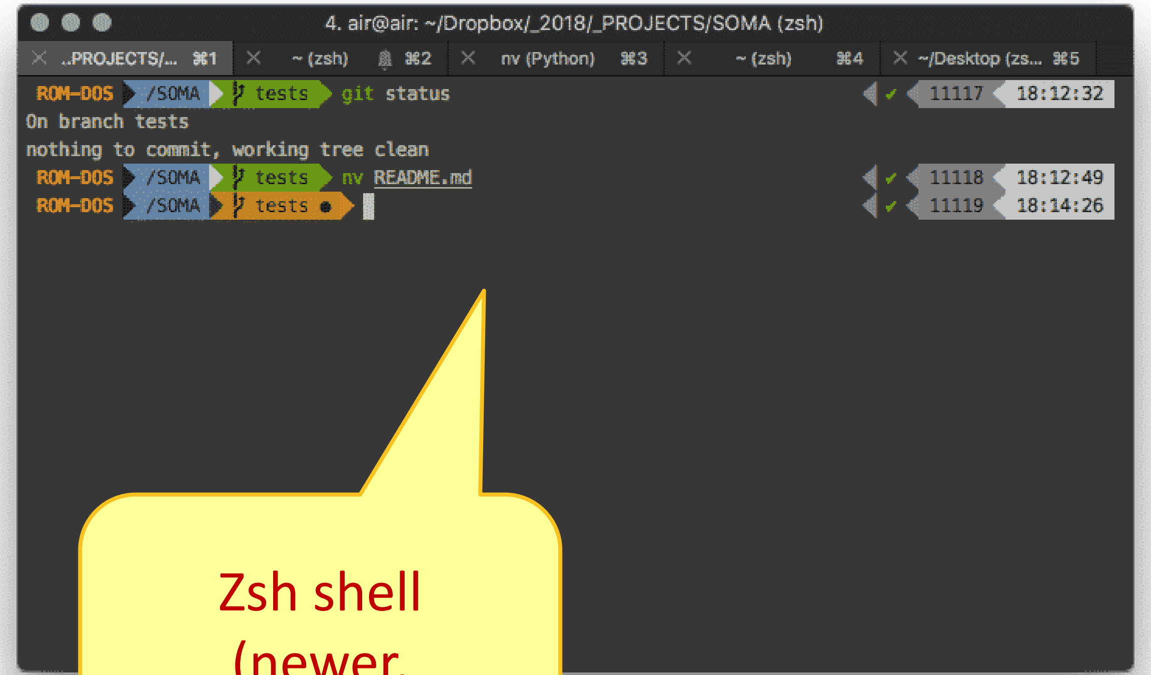
Mac only: Bash vs. Zsh

- **Windows** folks are using Git-bash, which has nice colorized Git output
- **Mac**: new default shell is zsh, older versions will have bash
  - In your terminal, execute `echo $0`



```
mc-130-49-26-56:Documents rhlmc1$ ls
Data-Science-for-Linguists-2019 HW1-Repo
mc-130-49-26-56:Documents rhlmc1$ cd HW1-Repo/
mc-130-49-26-56:HW1-Repo rhlmc1$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        dan/hello.txt

nothing added to commit but untracked files present (use "git add" to track)
mc-130-49-26-56:HW1-Repo rhlmc1$ git log
commit 23df080f51b370eb9dff5794187be4628f801d9f (HEAD -> master, origin/master, origin/HEAD)
Author: Na-Rae Han <naraehan@gmail.com>
Date:   Thu Jan 10 15:14:17 2019 -0500

    script mention changed to file

commit d329b4bc04deac9436ebdea0451893f375a26
Author: Na-Rae Han <naraehan@gmail.com>
Date:   Thu Jan 10 14:57:26 2019 -0500

    repo prepped for perfonal folders,
```

Bash shell (older)



```
4. air@air: ~/Dropbox/_2018/_PROJECTS/SOMA (zsh)
..PROJECTS/...  ⌘1 | ~ (zsh)  ⌘2 | nv (Python)  ⌘3 | ~ (zsh)  ⌘4 | ~/Desktop (zs...  ⌘5
ROM-DOS  /SOMA  tests  git status                                ✓  11117  18:12:32
On branch tests
nothing to commit, working tree clean
ROM-DOS  /SOMA  tests  nv README.md                              ✓  11118  18:12:49
ROM-DOS  /SOMA  tests ●                                          ✓  11119  18:14:26
```

Zsh shell (newer, recommended)

Wrapping up

- To-do #2 is out: due Thu.
  - Study numpy, make your own study notes as JNB. Submit via Class-Exercise-Repo.

- Try out DataCamp tutorials! Especially the numpy chapter.
- Learn:
  - Git, GitHub
  - Jupyter Notebook
  - numpy
  - pandas