Lecture 12:
Collaboration on Git/GitHub

LING 1340/2340: Data Science for Linguists

Na-Rae Han

Objectives

▶ **Git/GitHub**
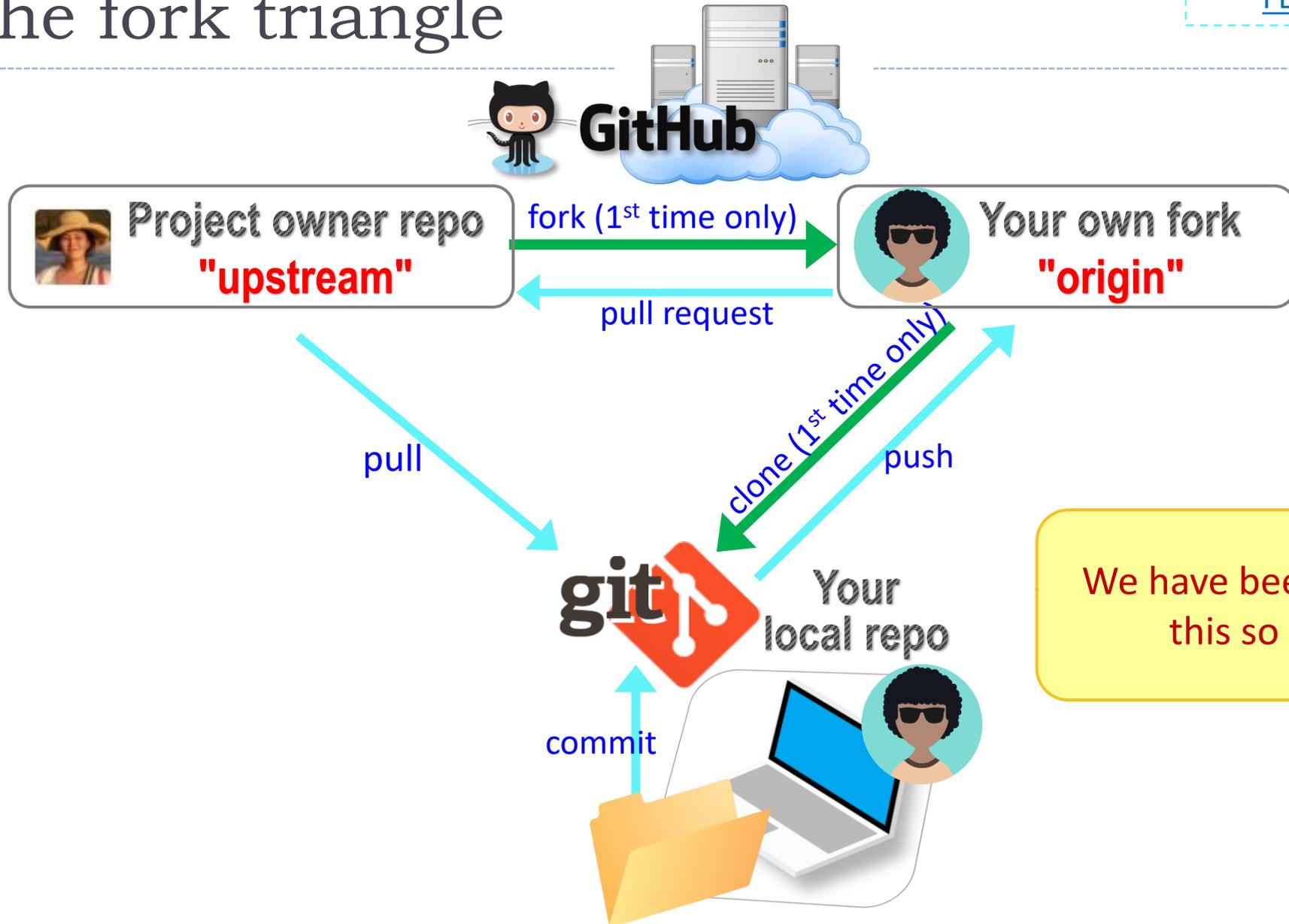
  ◆ Collaborating on the *same* repo – push access

▶ **HW #3 review**

  ◆ Task 2 prompt identification (wrap)

  ◆ Task 1 L1 identification

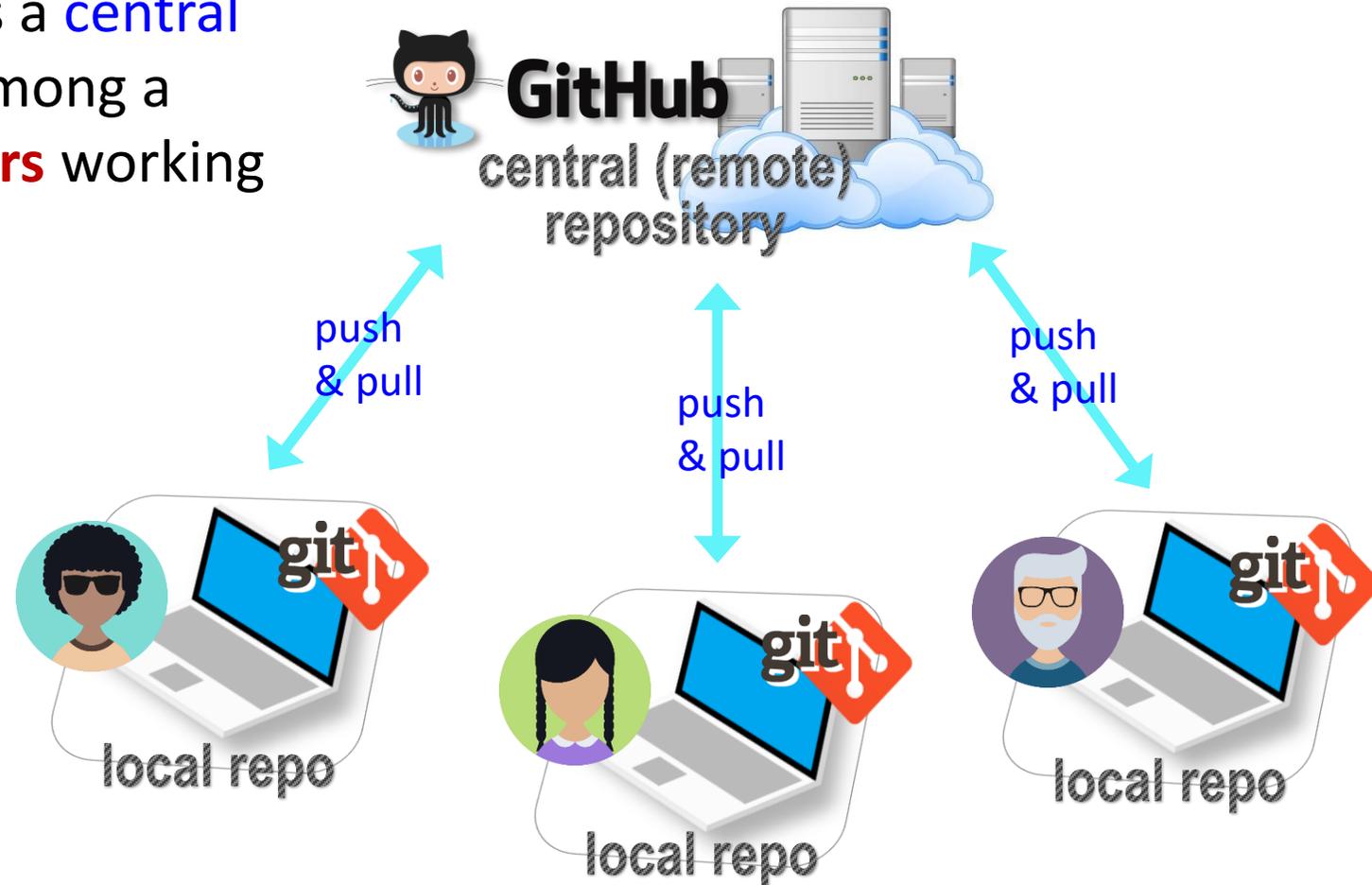# The fork triangle

GitHub: a *social*, remote repository

▶ GitHub also works as a central remote repository among a group of **collaborators** working on a shared project.

- ◆ Everyone works on their own *local* copy of the repository, making changes.
- ◆ Git is able to keep track and merge changes submitted by everyone.
- ◆ Everyone is an **equal collaborator** with push (=write) access.

GitHub central (remote) repository

push & pull

push & pull

push & pull

local repo

local repo

local repo

We are now ready!

3/14/2025

4

Introducing... "Class-Lounge"

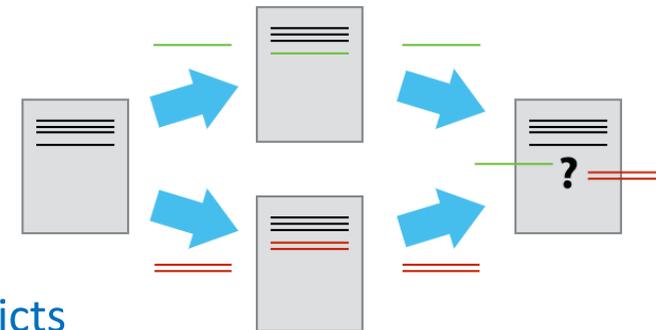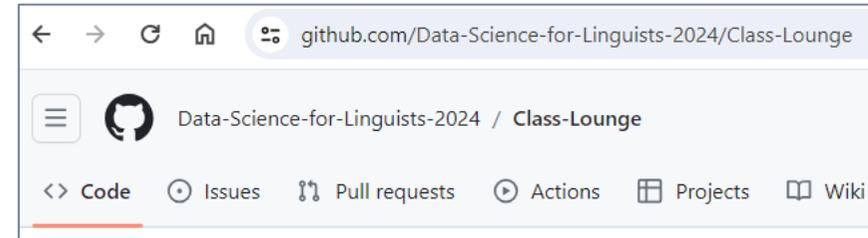▸ https://github.com/Data-Science-for-Linguists-2025/Class-Lounge

▸ Public!

▸ Everyone is listed as a "collaborator".

- ◆ Meaning, everyone has push access.
- ◆ No need to fork: pull and push directly.
- ◆ We will also truly collaborate: **edit shared files**.

▸ This means: **CONFLICTS**

- ◆ Na-Rae's tutorial on Git conflicts:
  - ◆ https://github.com/mcdonn/LSA2019-Reproducible-Research/blob/master/linking_git_and_github.md#conflicts

When there is a conflict

▸ After you pull, Git **changes your file content**, which now looks like:

```
<<<<<<< HEAD
There was copyright information here.


=======
"RTlexdec","RTnaming","Familiarity","Word","AgeSubject","WordC
tenFrequency","WrittenSpokenFrequencyRatio","FamilySize","Deri
y","InflectionalEntropy","NumberSimplexSynsets","NumberComplex
thInLetters","Ncount","MeanBigramFrequency","FrequencyInitialD
>>>>>>> c954b23f86b629c569223e2c0c38e32a0d870d22
```

You must **manually edit the file** and tidy up these lines.
(aka resolve conflict)

A GitHub race: our favorite animals

1. Everyone was already added to the repo as a collaborator.

2. Clone the repo to your laptop.

3. Edit "favorite_animals.md", add your line.

4. Do your usual local git routine: adding, committing.

5. Try pushing. It is likely you run into a conflict (someone else pushed in the meantime) and Git tells you to pull first.

6. Pull to receive the new updates.

   ◆ Pay attention to the Git log message! It tells you how the file was changed.

7. Open "favorite_animals.md", edit to resolve conflict.

8. Go back to step 4. Hope you were quick enough this time!

Wrapping up

- To-do #13: visit your classmates

- Next week: supercomputing on CRC

- Progress report #2 is coming up. Work on your project!

- PyLing next Wed!
  - Introduction to HCI (Human-Computer Interaction) by Alejandro Ciuba
  - 3/19 (Wed) 6pm, @ 2818 CL (ling conference room)