LESSON 4 · 29 OCTOBER 2026

Applied Systems Sciences 1

Working with git and GitHub

A Python warm-up, then your team's first shared repository

David Maier
Department of Environmental Systems Sciences
University of Graz

How far is your setup?

Vote for the last step that works. The laptop part of the next lessons needs git and VS Code.

How far does your setup work?

1. Nothing yet
2. uv works
3. uv and VS Code work
4. uv, VS Code and git work
5. Everything works, and the Education application is sent

Python challenges: half a point each

Solve a challenge within two tries, and it earns half a participation point

½ point

Predict the output

Type what the code prints and check it. Right in your first or second try, it earns the half point. After a miss you get a hint, not the answer.

½ point

Finish the code

Replace the `...` and press ▶ Run. Solved in your first or second run, it earns the half point. Hints cost nothing.

Five challenges today, so up to 2.5 points. Log in with GitHub to take part. Under each challenge you see whether you got the point.

Before and after the catch

```
stock = 10
before = stock
catch = 4

if catch > stock:
    print("not enough fish")
else:
    stock = stock - catch

print(stock, before)
```

What will be printed?

Your turn: the catch cannot exceed the stock

```
stock = 3
catch = 5

if catch > stock:
    ...
else:
    stock = stock - catch

print(stock, catch)
```

Finish the if: a fisher who wants more than the lake holds gets what is left, and the lake is empty.

GOAL

```
0 3
```

Looping over the catches

```
catches = [3, 1, 2]
total = 0

for catch in catches:
    catch = catch * 2
    total = total + catch

print(catches, total)
```

What will be printed?

Your turn: what is left after each catch

```
stock = 10
catches = [4, 5, 1]
left = []

for catch in catches:
    stock = stock - catch
    ...

print(left)
```

Finish the loop so that left records how many fish are left after each catch.

GOAL `[6, 1, 0]`

Counting the big catches

```
catches = [3, 1, 2]
catches.append(4)
big = 0

for catch in catches:
    if catch > 2:
        big = big + 1

print(big, len(catches), catches[1])
```

What will be printed?

Why version control

Without git

```
model_final.py
model_final_v2.py
model_final_v2_really_final.py
model_backup_old.py
model_DO_NOT_DELETE.py
```

Which one made the figure in your report?

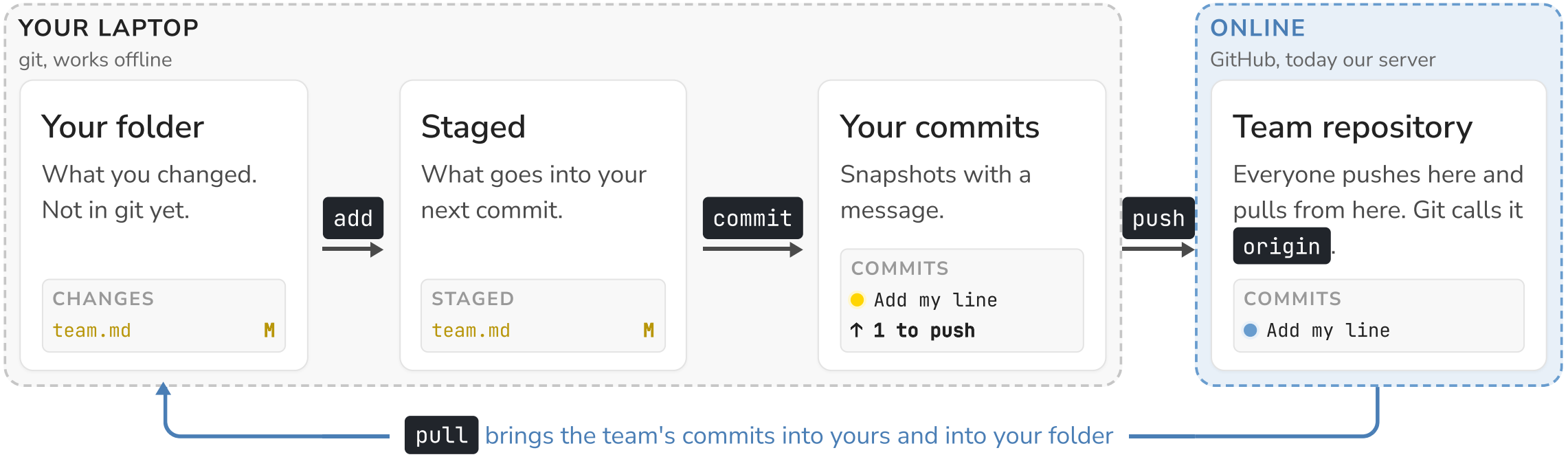
With git

```
e4a1c2f Brake when the gap is small
9b77d10 Add random slowdowns
3f02e88 First traffic model
```

One file. Every change is saved with a message, an author and a time, and you can go back to any of them.

> In the projects, three of you change one model. Git keeps each change apart and brings them together.

Where a change lives



VS Code's Source Control shows the same places: **Changes**, **Staged Changes**, and **Sync Changes** for pull and push.

Your task: add yourself to the team file

First, a team of up to three: one of you presses **Start a team** and reads out the code, the others type it in and press **Join**. In the wrong team? **Leave team**.

1 Edit

Add one line to team.md: your name and what you would like to research.

```
edit team.md
```

→

2 Stage

Choose what goes into your next commit.

→

3 Commit

Save a snapshot and say what you changed.

→

4 Push

Send your commit to the team.

Rejected? Detour: Pull

The team has commits you do not have. Get them first, then push again.

The git commands: look them up in the git cheatsheet.

A rejected push is **expected**: a teammate was faster.

Stuck? Use the **Hint** and **Explain** buttons next to your terminal.

Try it: your team's repository

Open this slide on your phone or laptop to get your team's terminal.