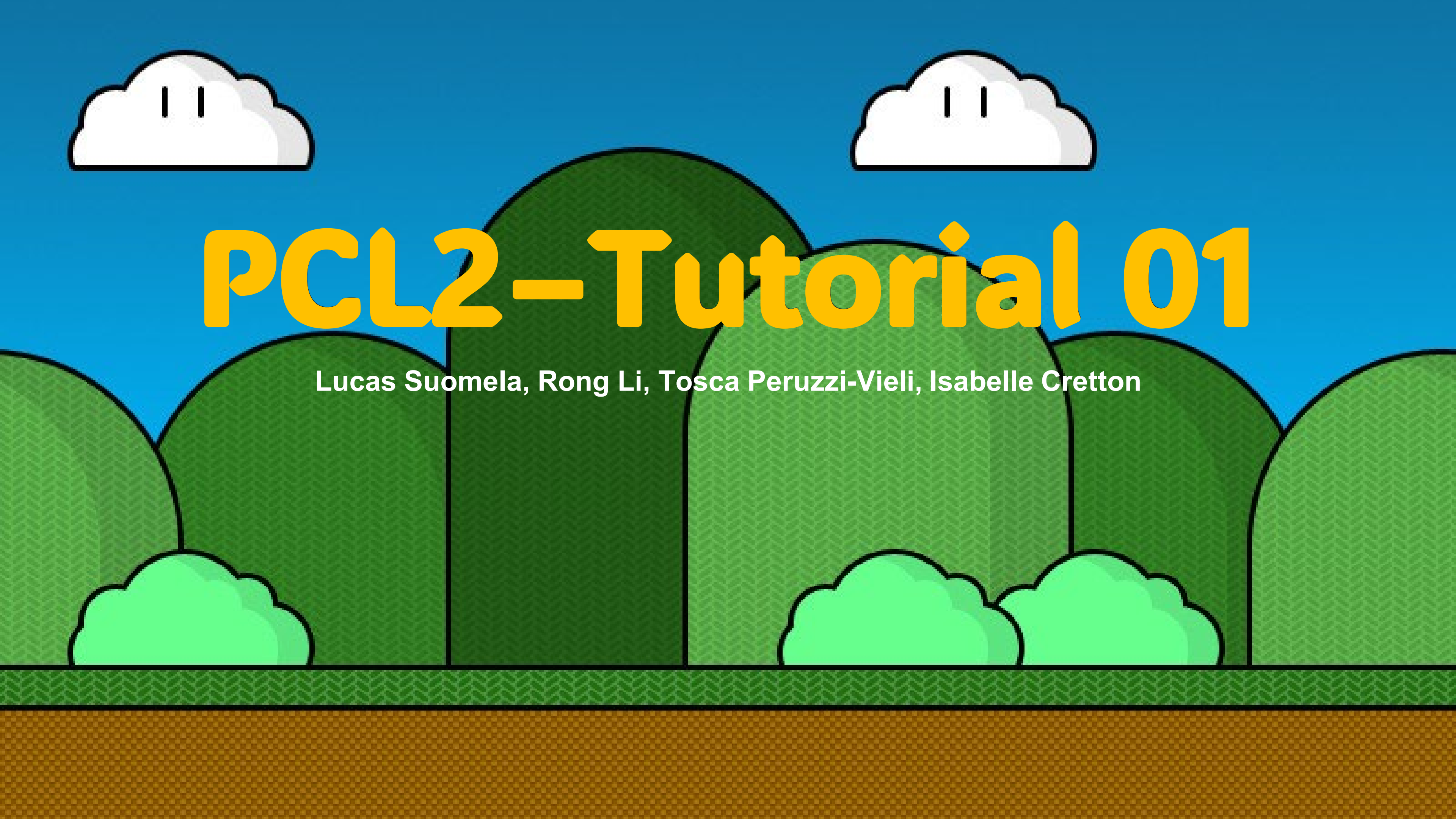
PCL2-Tutorial 01

Lucas Suomela, Rong Li, Tosca Peruzzi-Vieli, Isabelle Cretton



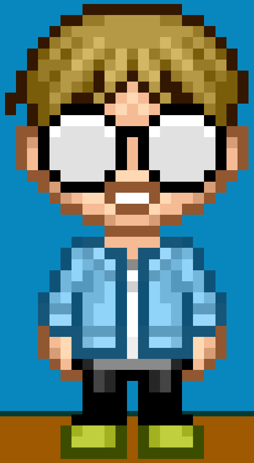
Who we are...

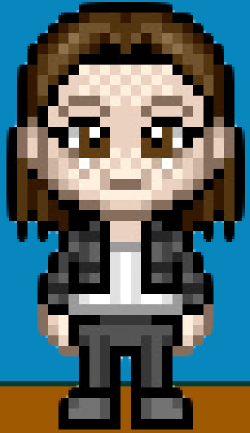
Isabelle AKA Izzie
4th semester, CL Major BA
Computational Science Minor
Speaks 4.5 languages

Lucas
4th semester, CL Major BA
Japanese Studies Minor
Recently developed addiction to keyboards

Rong 榕 🌳
1st (?) semester, CL Major MA
Neuroinformatics Minor
Though my name sounds like "wrong", I'm trying my best to do things right

Tosca
6th semester, German Major BA, CL Minor
Cat person, owns a dog





Tell us about yourself!

(click here)

Organizational Stuff

Weekly exercises (most are **NON-GRADED**)

- **release**: Wednesdays at mid-day on GitLab
- **deadline**: Tuesdays at 23:59 via GitLab <u>if you want feedback</u>
- <u>"autograde" on GitLab</u>

"Project" Style exercise + 2 of the weekly exercises are **GRADED**

- Exercises 2 & 6 : 1 point each
- Exercise 8: 3 points.
- **count 25% of the grade**

On Paper Midterm Exam: 24/04

Exam: 12.06.2024, from 10:00am (24 hours)

What's covered?

- Lecture recaps, look into exercises, answer your questions…
- Questions? Ask your neighbour, ask us, ask in the OLAT-forum…
- **<u>DO NOT WAIT UNTIL RIGHT BEFORE HAND IN IF YOU CAN AVOID IT</u>**

Organizational Stuff

Tutorial **every week**

- A "big" tutorial will be held the weeks a graded exercise, as well as before the midterm
- Q&A sessions the other weeks

Big Tutorial

- All Tutors present
- Explanation of graded exercise, midterm
- Recap of topics from the lecture, focus on exercise-relevant topics, summary of FAQs

Small Tutorial

- 2 Tutors present
- Q&A
- Slides covering the basics will be available on OLAT

Tutorials by week

Monster tutorial TBD!!

week 1 (23.02): Big
week 2 (1.3): Small
week 3 (8.3): Big
week 4 (15.3): Small
week 5 (22.3): Small
week 6: (29.3): Small
week 7: (12.4.): Break

week 8 (12.4): Big
week 9 (19.4): Big
week 10 (26.04): Break
week 11 (10.5): Big
week 12 (17.5): Small
week 13 (24.5): Small

Write these dates down somewhere!

Contacting the Tutors

OLAT Forum
"@Tutors" on OLAT
During the Tutorial

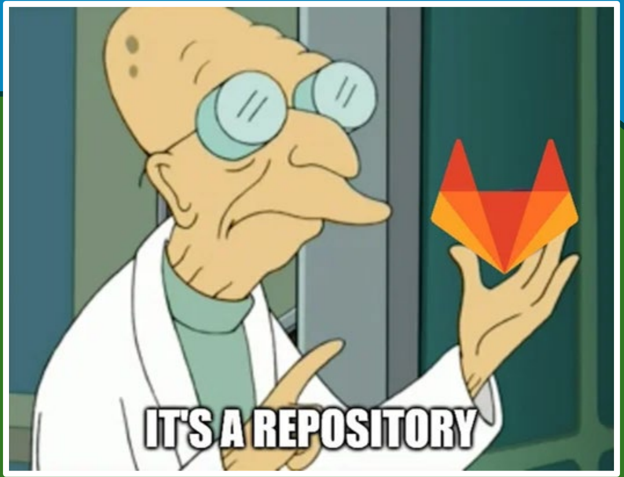
Please contact us if you are struggling with errors or strange (code) behavior

Rule of Thumb: If you spend more than an hour working on/fixing a single error, please write us an email

What the Hell Is GitLab?

Like Google Drive…

… but for your code... and better.



- Store your code in a repository
- Track previous versions and restore them
- Work on the same code in a team easily

GitLab will come in handy throughout your studies (and beyond), especially for large r group projects.

How does GitLab Work?

GIT = VERSION CONTROL SYSTEM (VCS)

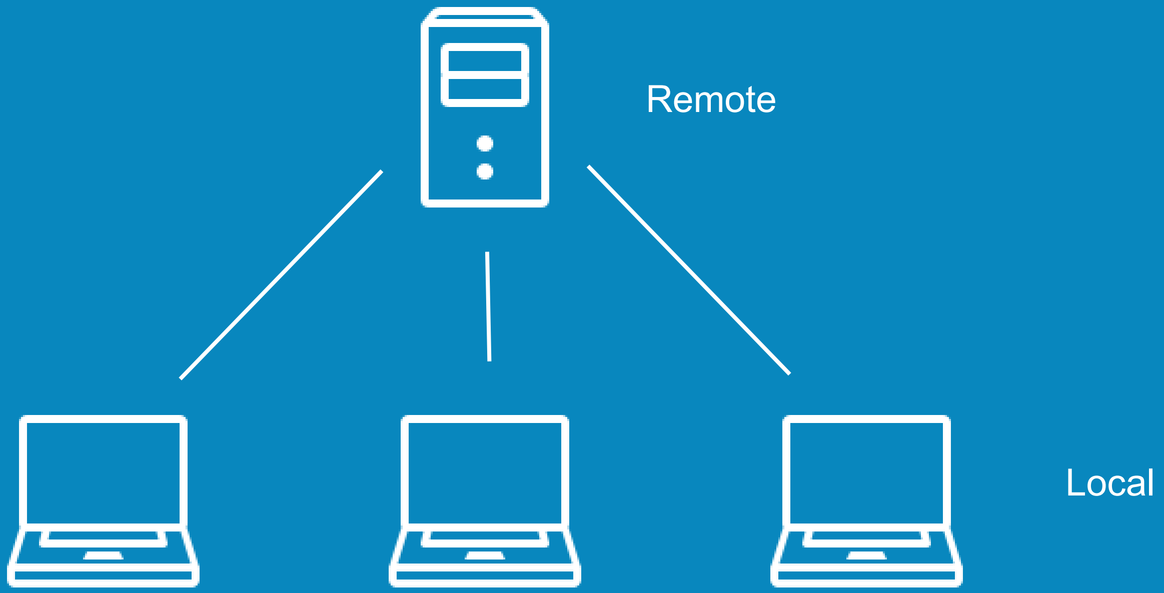
Which changes were made?

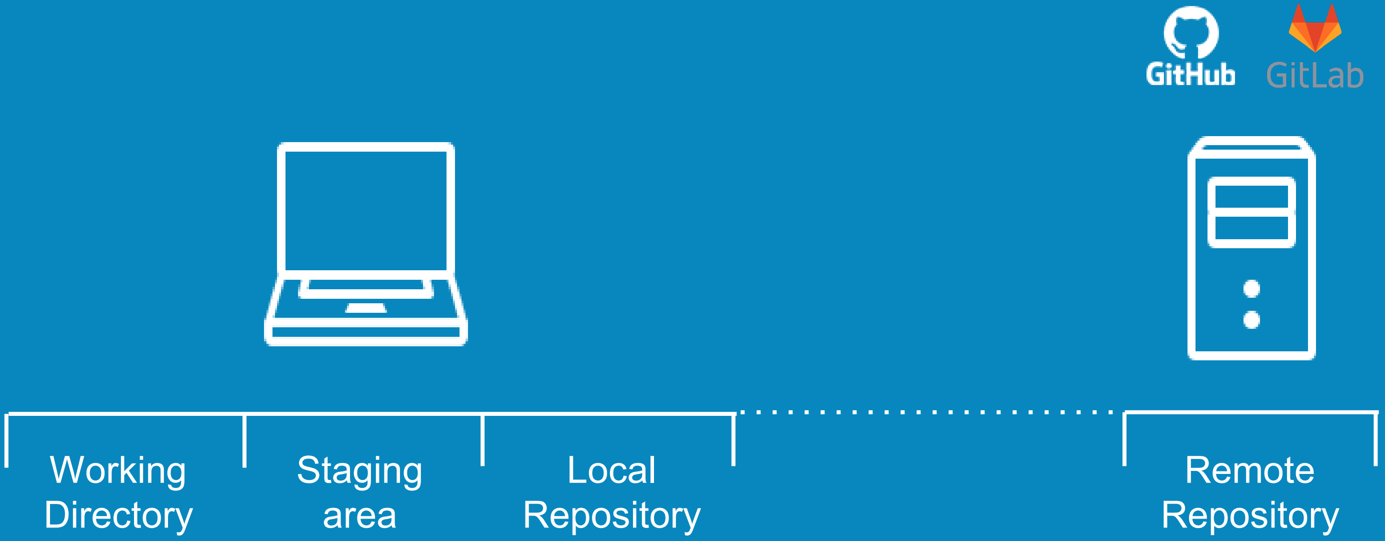
Who changed what?

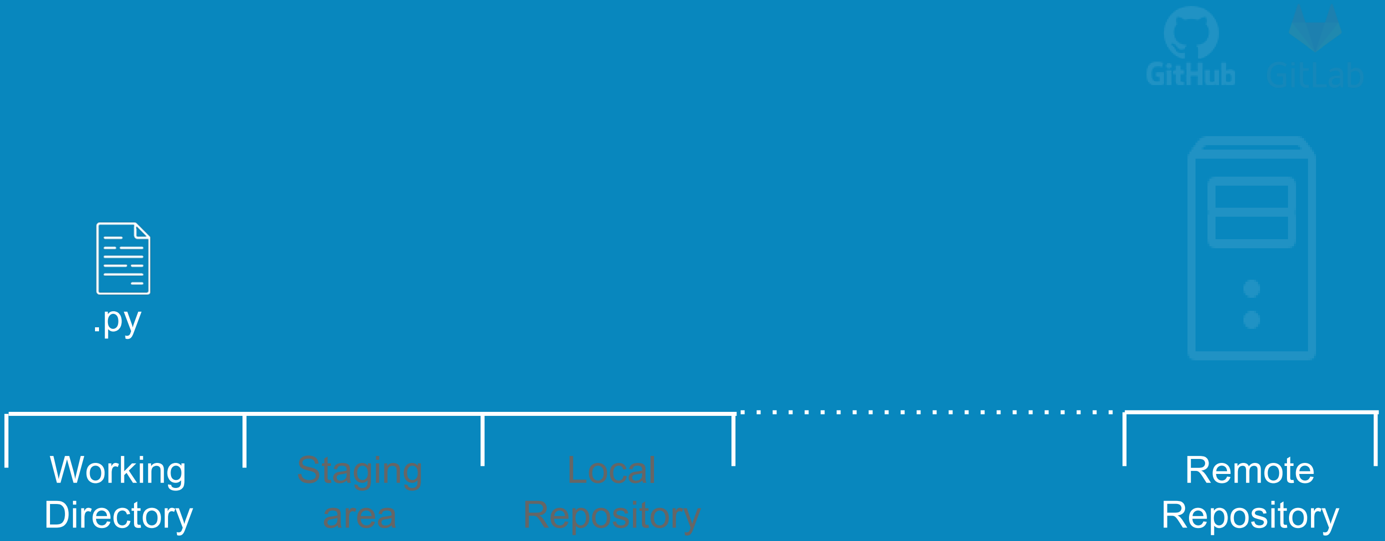
When was something changed?

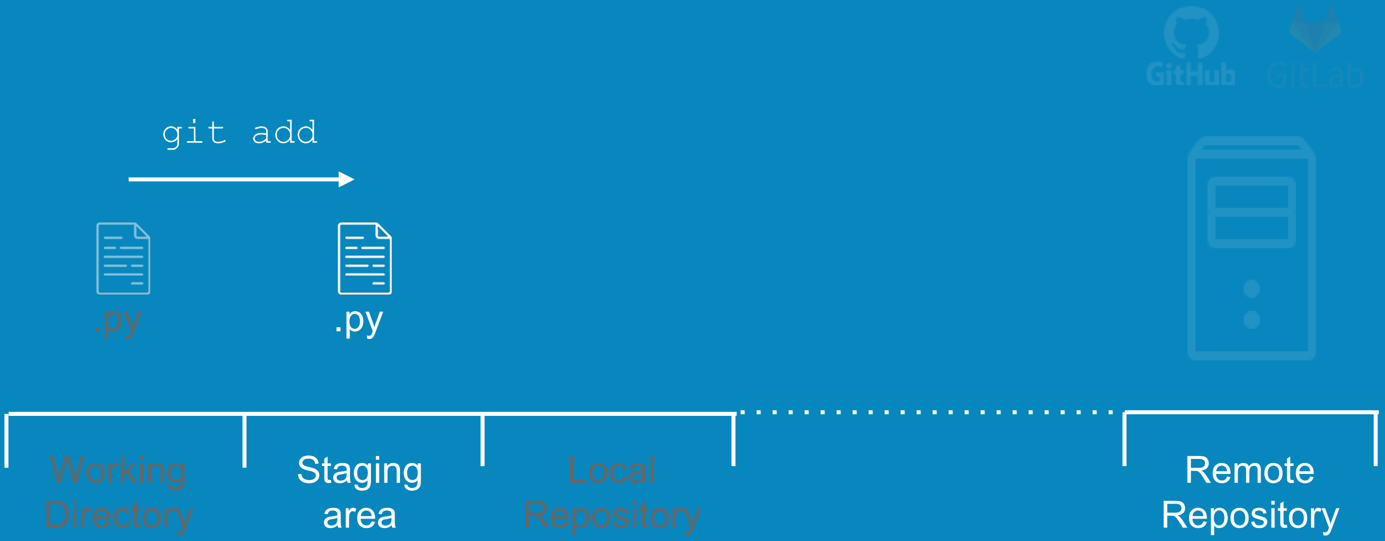
Why was it changed?

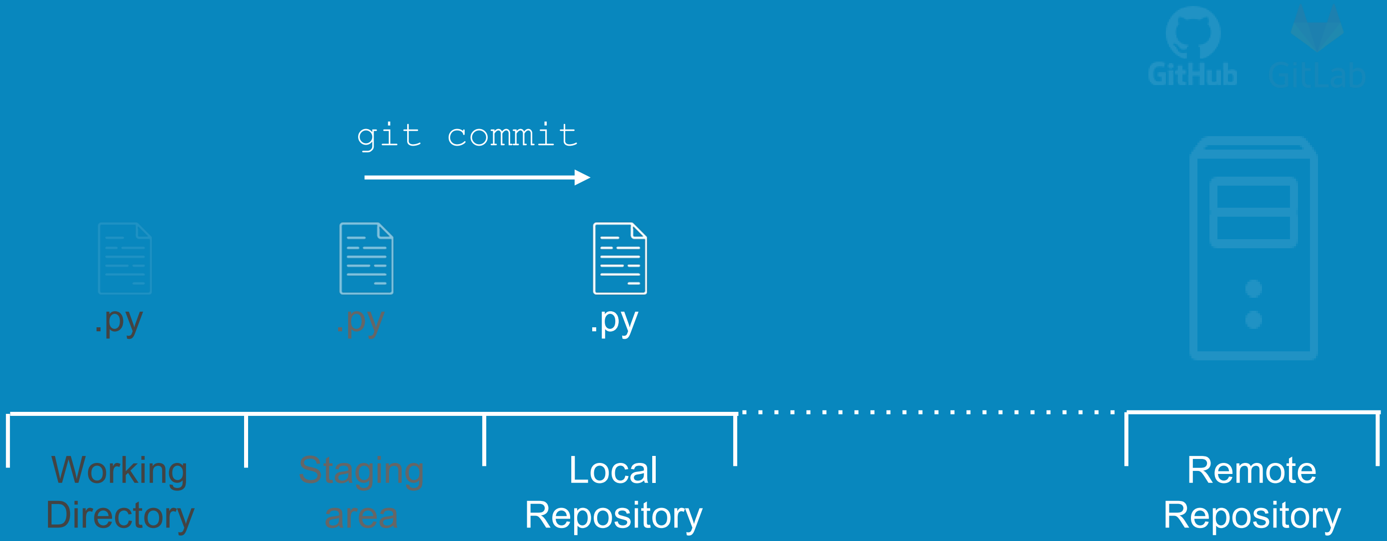
How does GitLab Work?



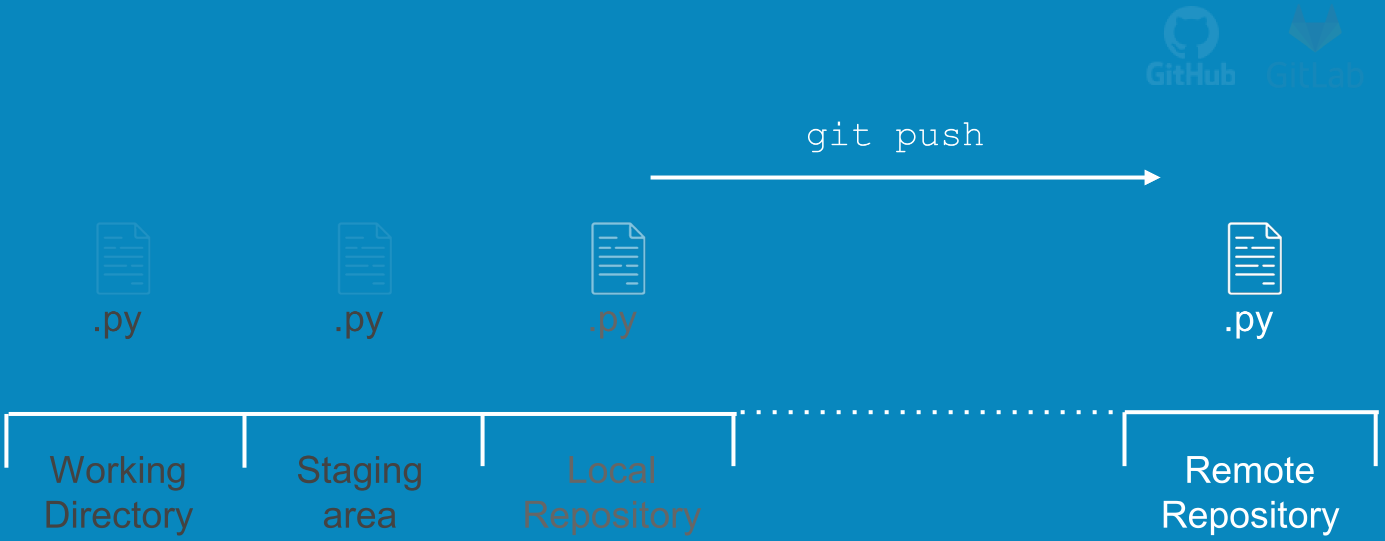

GitHub
GitLab
Working Directory
Staging area
Local Repository
Remote Repository

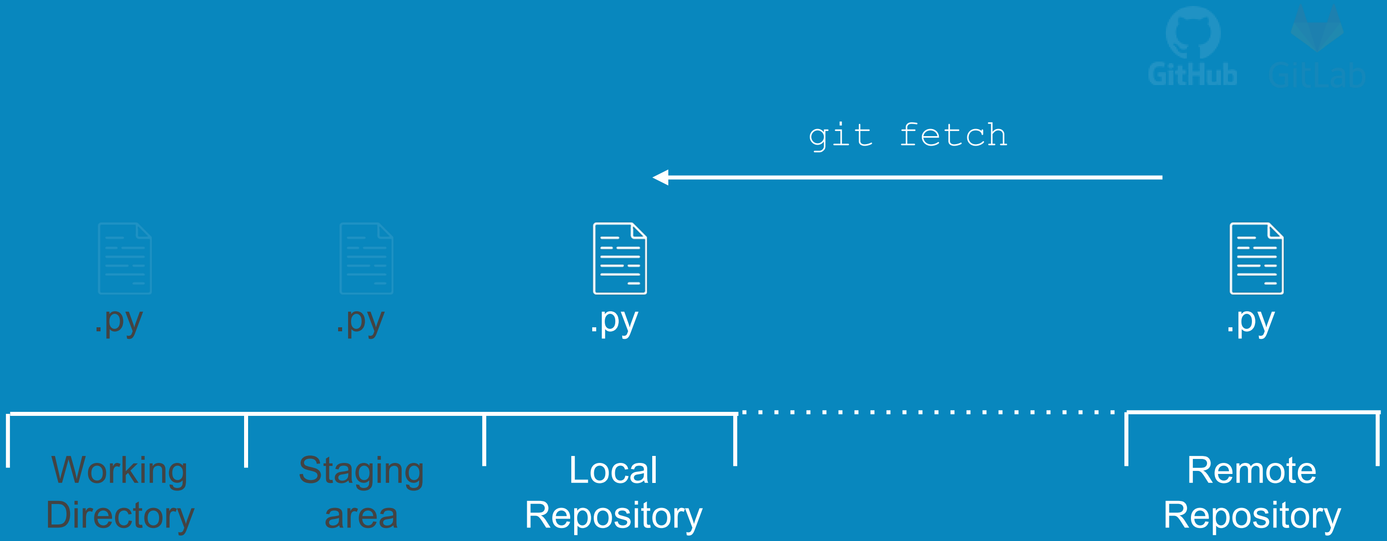

.py
GitHub
GitLab
Working Directory
Staging area
Local Repository
Remote Repository

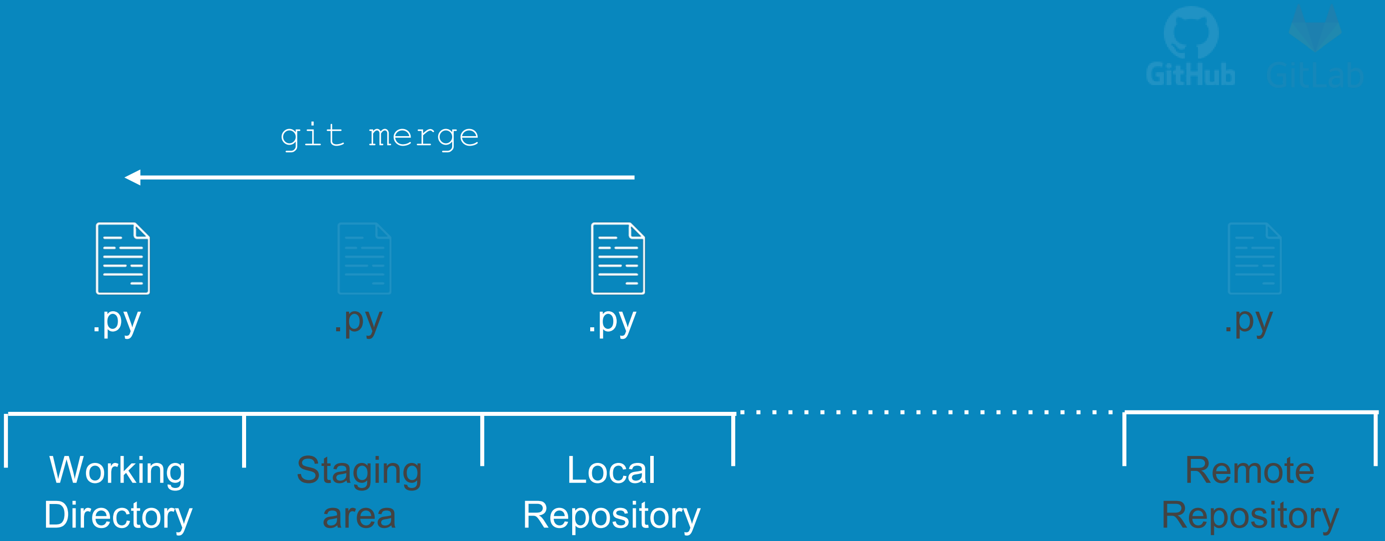

git add
.py
.py
GitHub
GitLab
Working Directory
Staging area
Local Repository
Remote Repository



GitHub
GitLab
git commit
.py
.py
.py
Working Directory
Staging area
Local Repository
Remote Repository


GitHub
GitLab
git push
.py
.py
.py
.py
Working Directory
Staging area
Local Repository
Remote Repository


GitHub
GitLab
git fetch
.py
.py
.py
.py
Working Directory
Staging area
Local Repository
Remote Repository

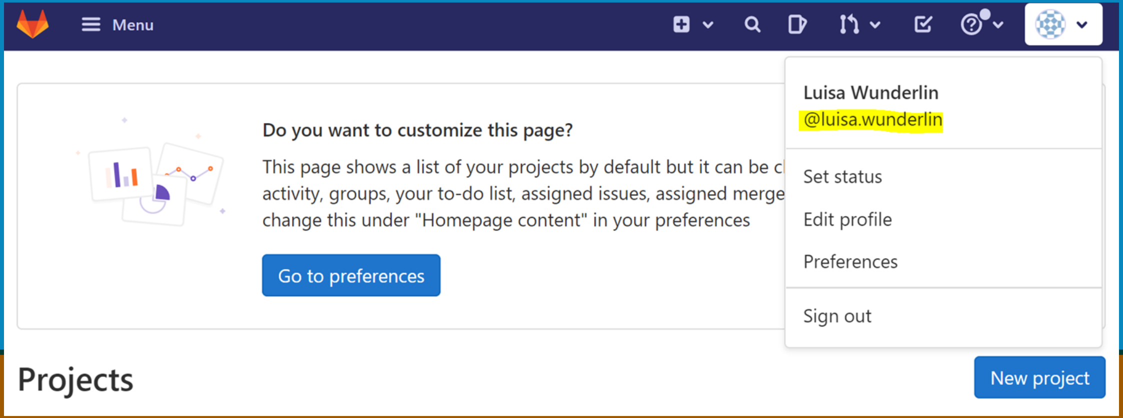


GitHub
GitLab
git merge
.py
.py
.py
.py
Working Directory
Staging area
Local Repository
Remote Repository

Sign Up to GitLab

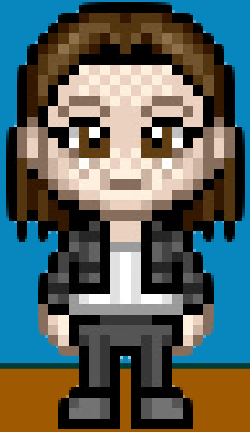
Go to https://gitlab.uzh.ch/

Sign in with SWITCH edu-ID (UZH email and password)



Do you want to customize this page?

This page shows a list of your projects by default but it can be c
activity, groups, your to-do list, assigned issues, assigned merge
change this under "Homepage content" in your preferences



Important Commands

Command	Description
git clone [project url]	download repository/project to your computer
git status	display status of your current directory and latest commit
git log	check commit history
git add [filename]	stages file for commit
git commit -m ‘[message]’	commit all the files you’ve staged and give your changes a title
git push	push your committed local changes to the remote repository (uploads them)
git pull	fetches changes and updates your local repository

Before You Can Start Working...

1. Create a GitLab account
2. Install git on your local machine (Source)
   - $ git version → is it installed already?
   - Mac & Windows: https://git-scm.com/downloads
   - Linux: $ sudo apt-get install git-all
3. git config --global user.name "Your Name"
   - git config --global user.email "youremail@domain.com"
4. Verify yourselves using an access token

What is SSH?

- SSH → Secure Socket Shell, or Secure Shell
- Generates a pair of unique keys, one private and one public
- Used to set up an encrypted channel between client and remote
- Recommended practice for git

Git Authentication Via SSH - Windows

1. Open a Terminal (Powershell on Windows)
2. Run the command **"ssh-keygen"**(ssh-keygen.exe on windows)
3. Confirm the key location and passphrase by hitting Enter
4. Use the command **cat .ssh\id_rsa.pub** to get your public key
5. Open Gitlab and navigate to the "preferences" menu by clicking on your prof ile icon
6. In the menu on the left side, select "SSH keys"
7. Paste your public key and give it a name

- There will be a short demonstration of this process later

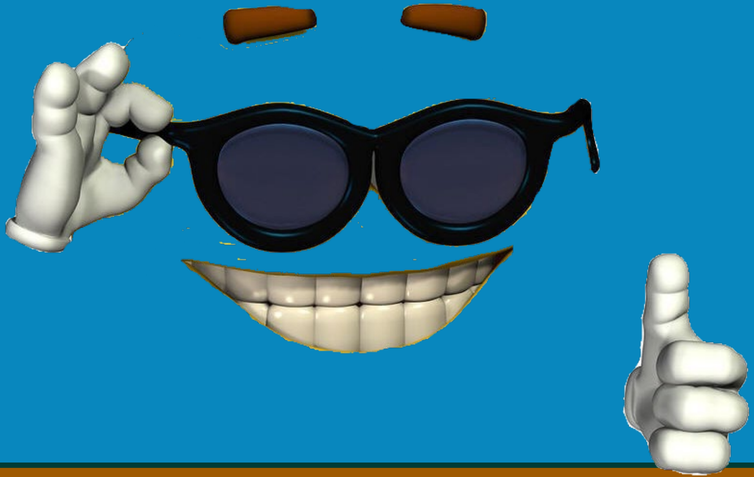
Git Authentication Via SSH - Mac

(copy paste the commands in 2., 4. and 8.)

1. Open a Terminal (Terminal on Mac OSX and Linux)
2. Run the command: **ssh-keygen -t ed25519 -C "<your.email@uzh.ch>"**
3. Confirm the key location and passphrase by hitting Enter
4. Use the command: **tr -d '\n' < ~/.ssh/id_ed25519.pub | pbcopy** to get your public key
5. Open Gitlab and navigate to the "preferences" menu by clicking on your profile icon
6. In the menu on the left side, select "SSH keys"
7. Paste your public key and give it a name
8. Run the command: ssh -T git@gitlab.uzh.ch
9. You shall see: Welcome to GitLab, @your.username!

- There will be a short demonstration of this process later

How do you edit your code?



git pull
JEJUCODING BASE CAMP
JEJU
git add .
JEJU
git commit
JEJUCODING BASE CAMP
git push
JEJUCODING BASE CAMP

Visual Studio Code (Source)

1. On Mac

   Code > Preferences > Settings > type ‘Git: Enabled’
   > make sure box is ticked

1. On Windows

   File > Preferences > Settings > type ‘Git: Enabled’ > make sure box is ticked

1. Configure your account

```
git config --global user.name "Your Name"
git config --global user.email "youremail@domain.com"
```

For any other IDEs, if you're unsure please reach out :-)

Git Command Line (Source)

1. Make sure git works
2. Clone into desired directory on your local machine.

Git Exercise (Exercise 0)

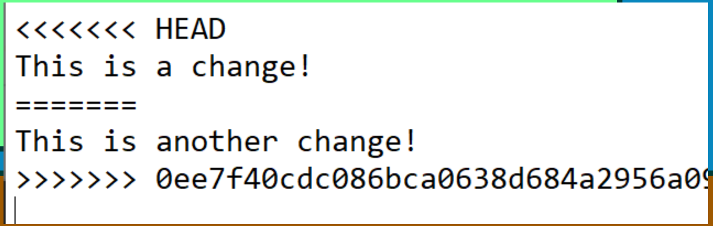
Once we've added you to the project (you should already be in)

1. Go to Projects > Your Projects > pcl2-2024-exercises > **Exercise 0**
2. Follow the steps in the description (readme-file)
3. Have fun with it, you cannot break anything!

How to avoid & resolve merge conflicts when working as a group

1. Pull the most recent version
2. Edit the code
3. Add and commit your changes
4. Pull it again

   → If there are merge conflicts, you will need to resolve them now.
   Most basic option: Just go into the file and delete what you don't want.
   Then, add and commit your updated file again.
5. Push your changes



```
<<<<<<< HEAD
This is a change!
=======
This is another change!
>>>>>>> 0ee7f40cdc086bca0638d684a2956a0
```

Now It's Your Turn...

#TODO:

- ❏ Create GitLab account
- ❏ Install git on your local machine
- ❏ Set up git
- ❏ Try out the git-exercise
- ❏ Start working on the first exercise
- ❏ Have a good weekend
- ❏ We'll be here for any questions and to help you set up everything