Boot Camp Preparation - step by step

SRH Fernhochschule - The Mobile University

Nice to meet you!

You need to prepare several things before you can start the course documents:

1. **You need a computer**
2. **Please log in to github (if you do not have an account yet)**
3. **Install "miniconda"**
4. **Fork and clone the course repository from github**
5. **Prepare the Python environment for the course**
6. **Start the course**

I have prepared these step-by-step instructions for you.

Your laptop
Minimum requirements

01

Your laptop

We want to carry out data science and machine learning ourselves with the help of Python and Python frameworks. You will need a laptop (Windows, Mac or Linux). A tablet is not sufficient. Ideally, the **CPU should not be older than 3 years** and your laptop should have **8 GB** (preferably 16 GB or more) **of main memory**.

A graphics card is useful to speed up the training of the models but not necessary for this course.

Github Login (Create account)

02

Login to github

github is a website where programmers can store their code - i.e. the programs they have created.

We will exchange files via GitHub if necessary.

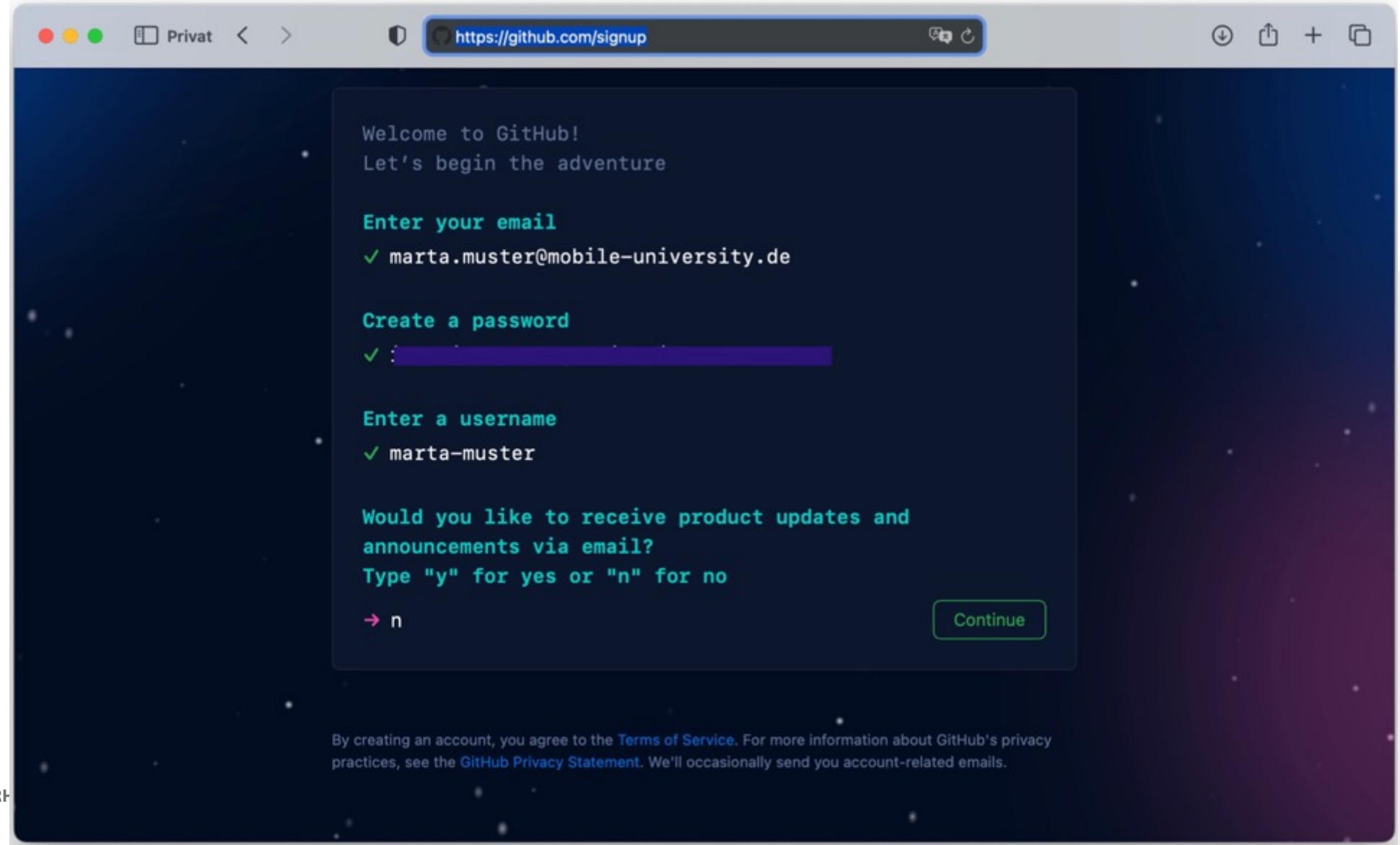
So that you can take part in the course, I would like to ask you to create an account there, i.e. to register.

The best way to do this is to use your **Mobile University e-mail address**

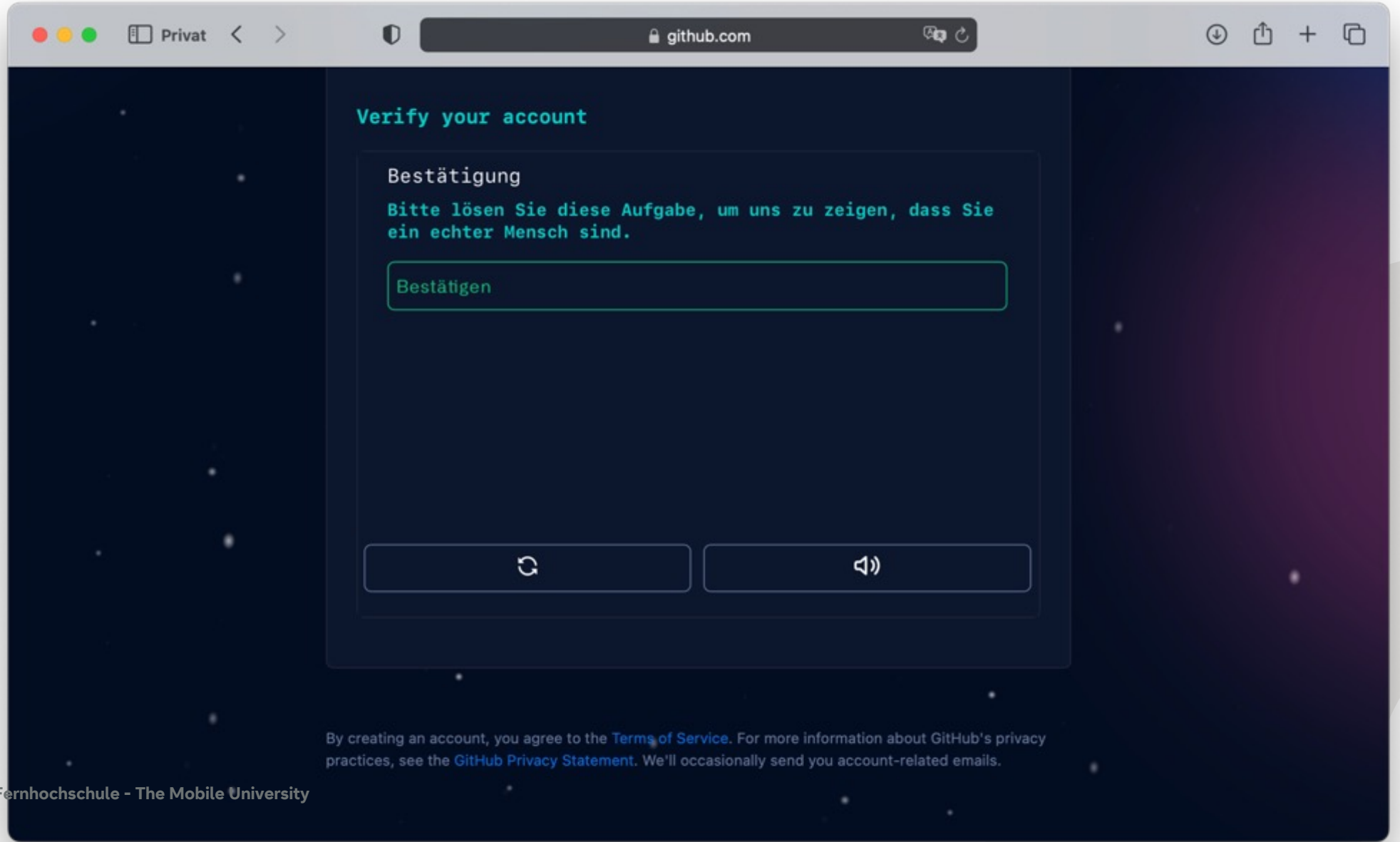
ihr.name@stud.mobile-university.de

If you already have a github account, you do not need to create a new one.

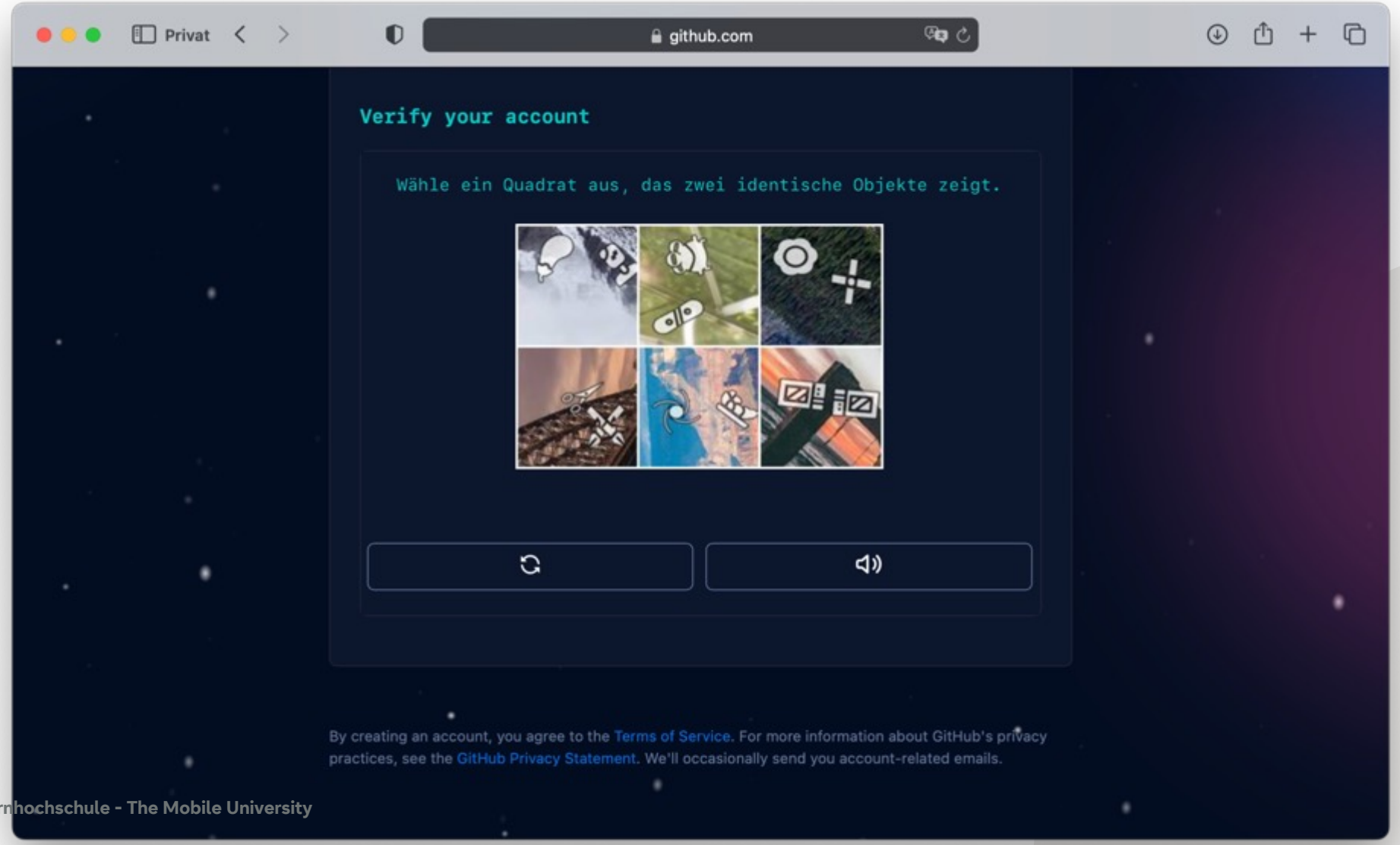
https://github.com/signup



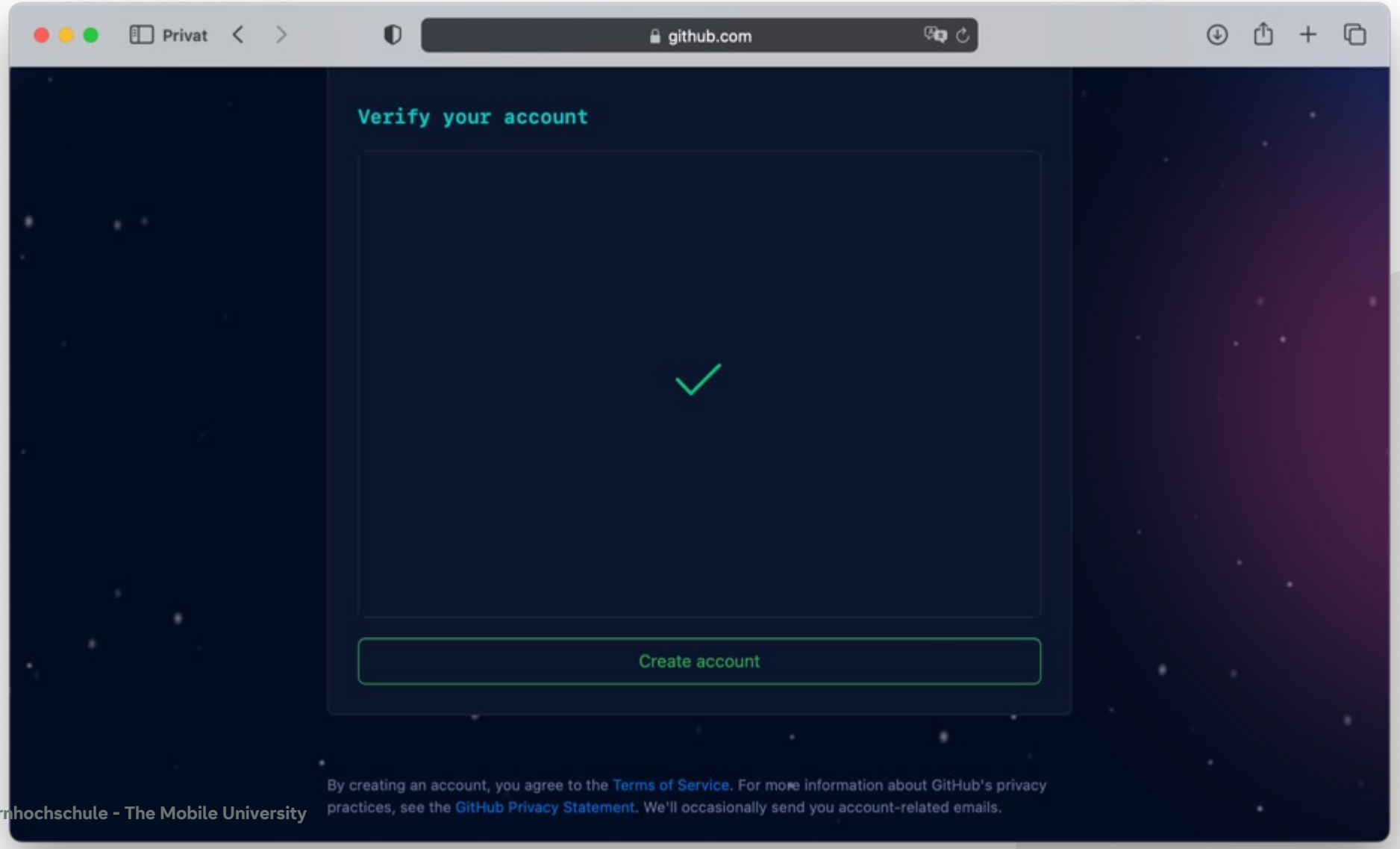
Confirm" button



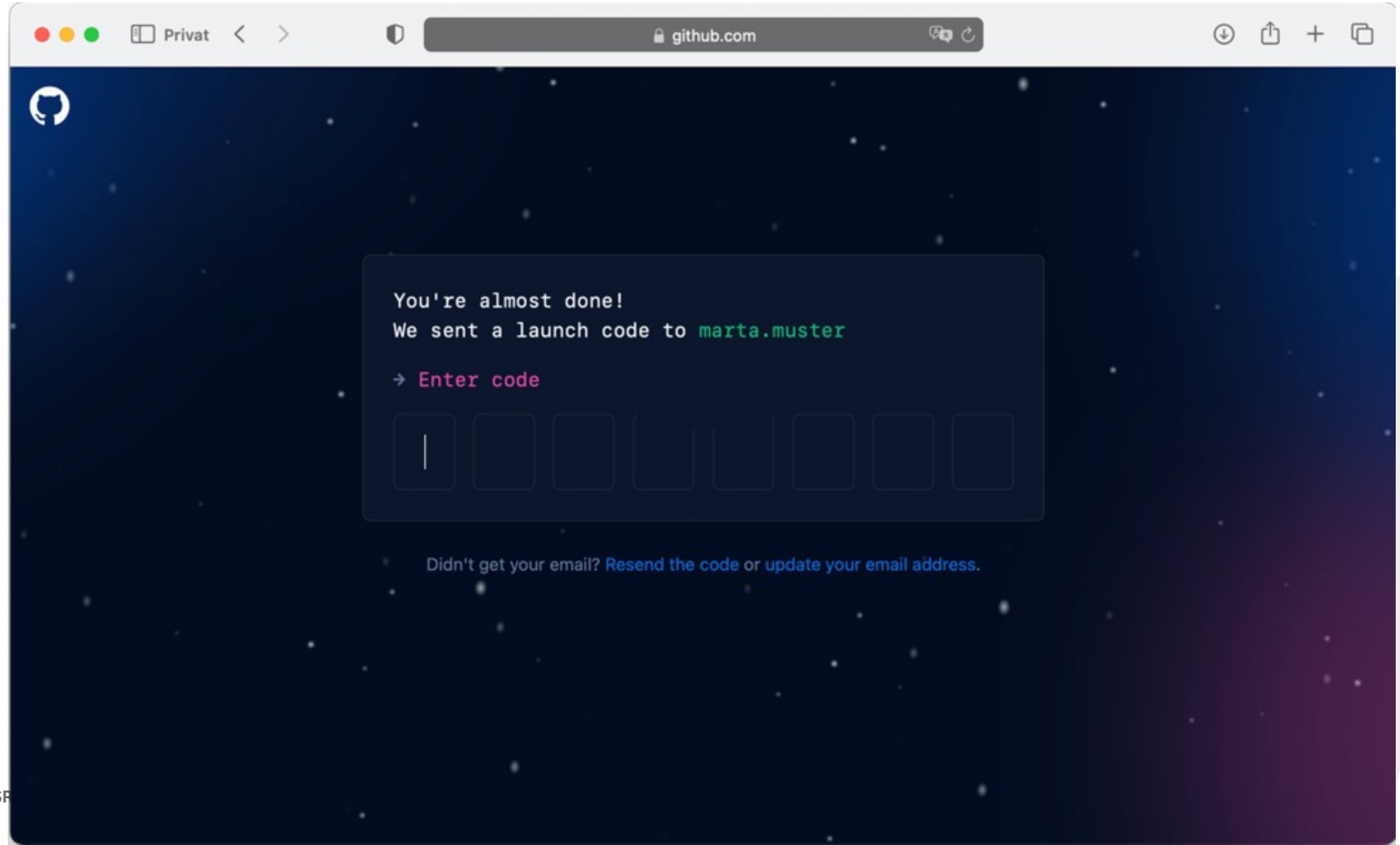
"Solve "puzzles



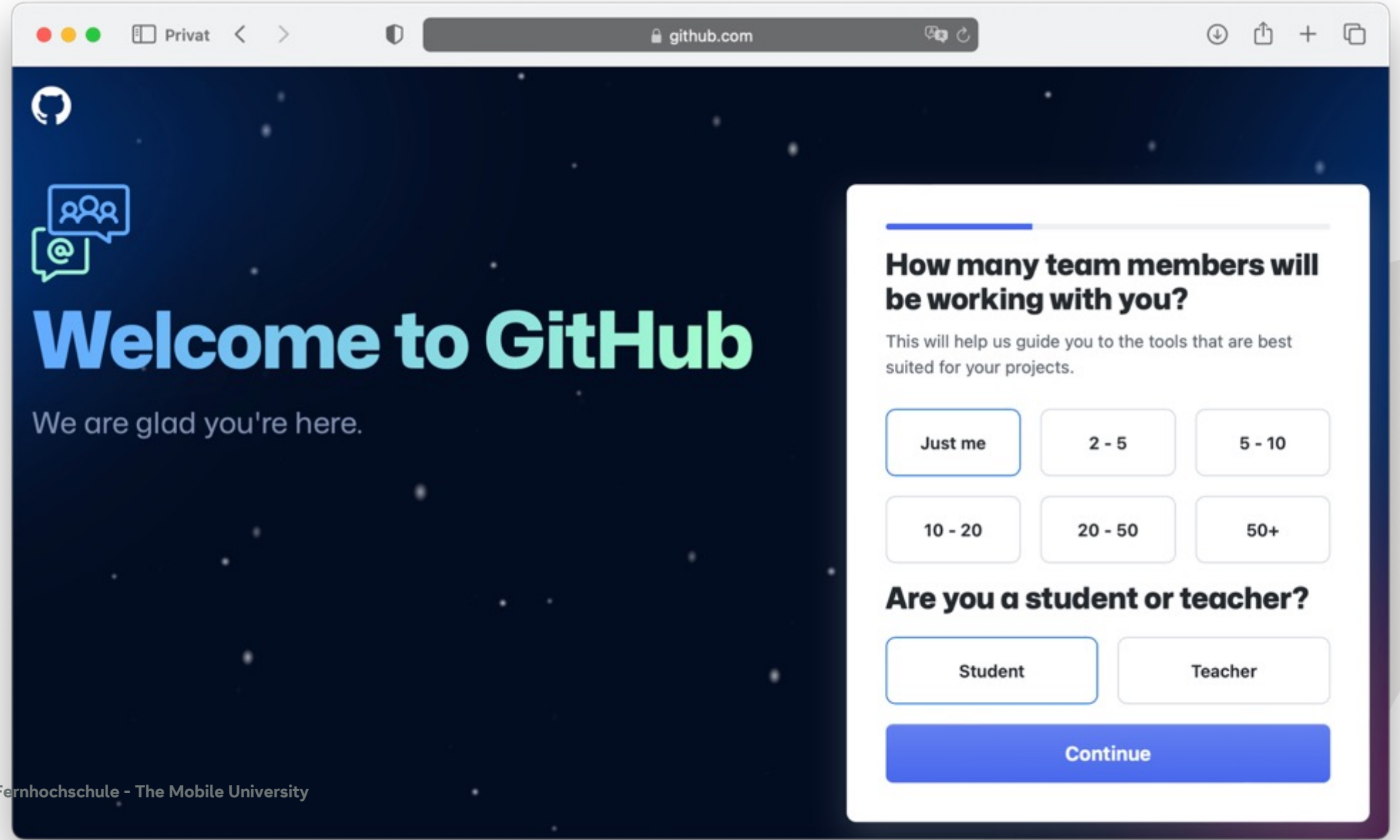
Create account - Click on "Create account"



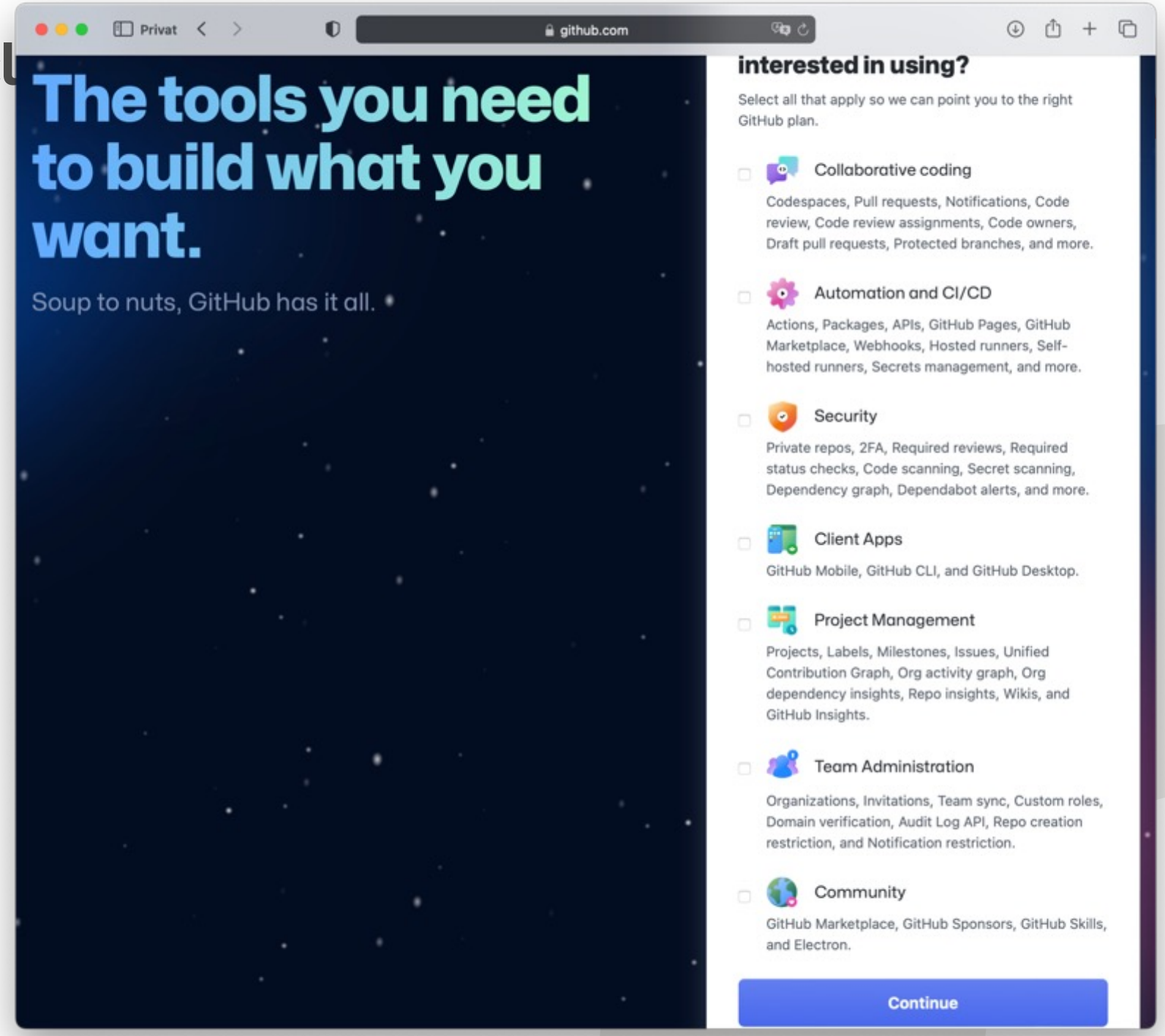
Confirm the code you received by e-mail



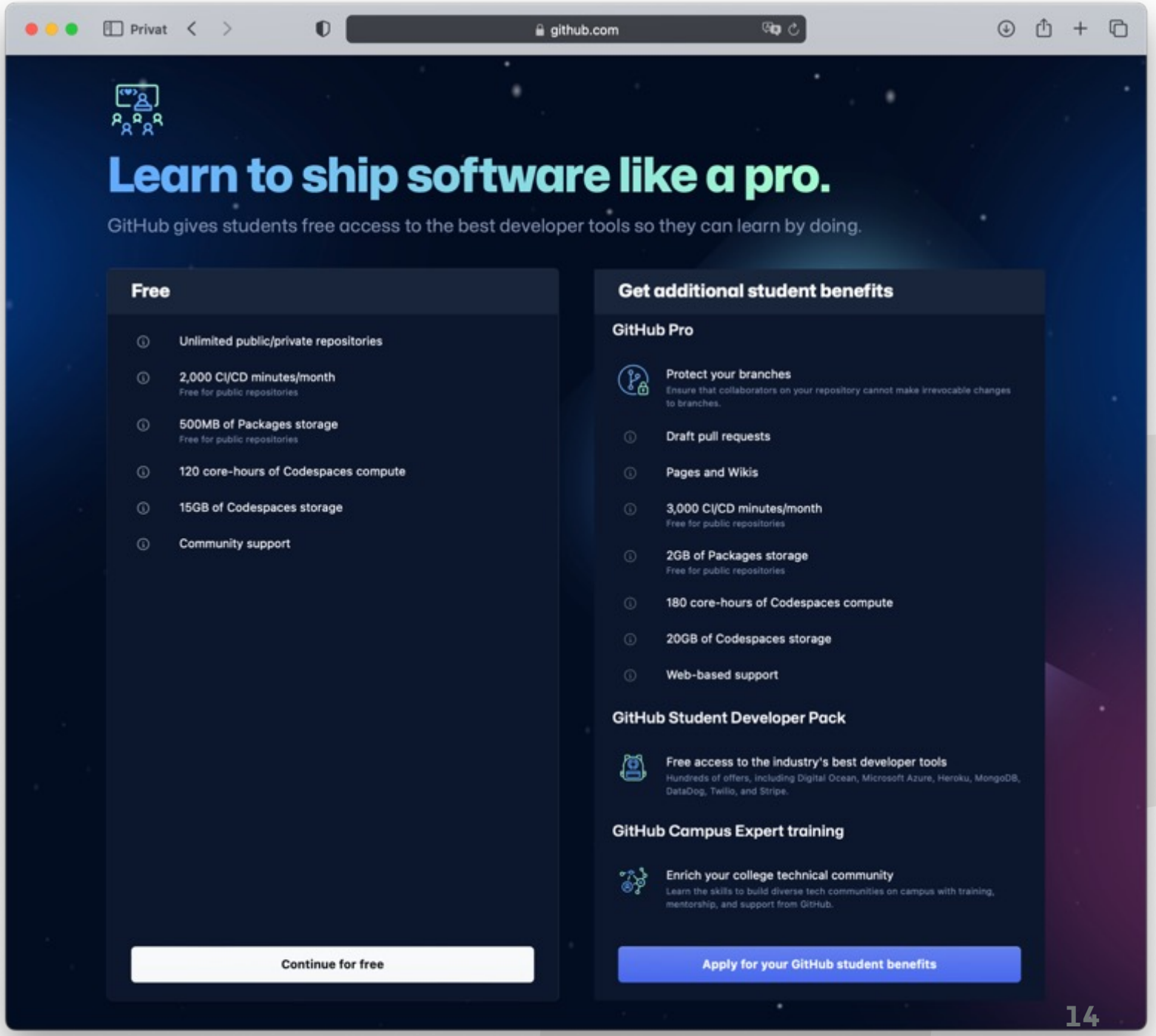
Answer further questions if necessary



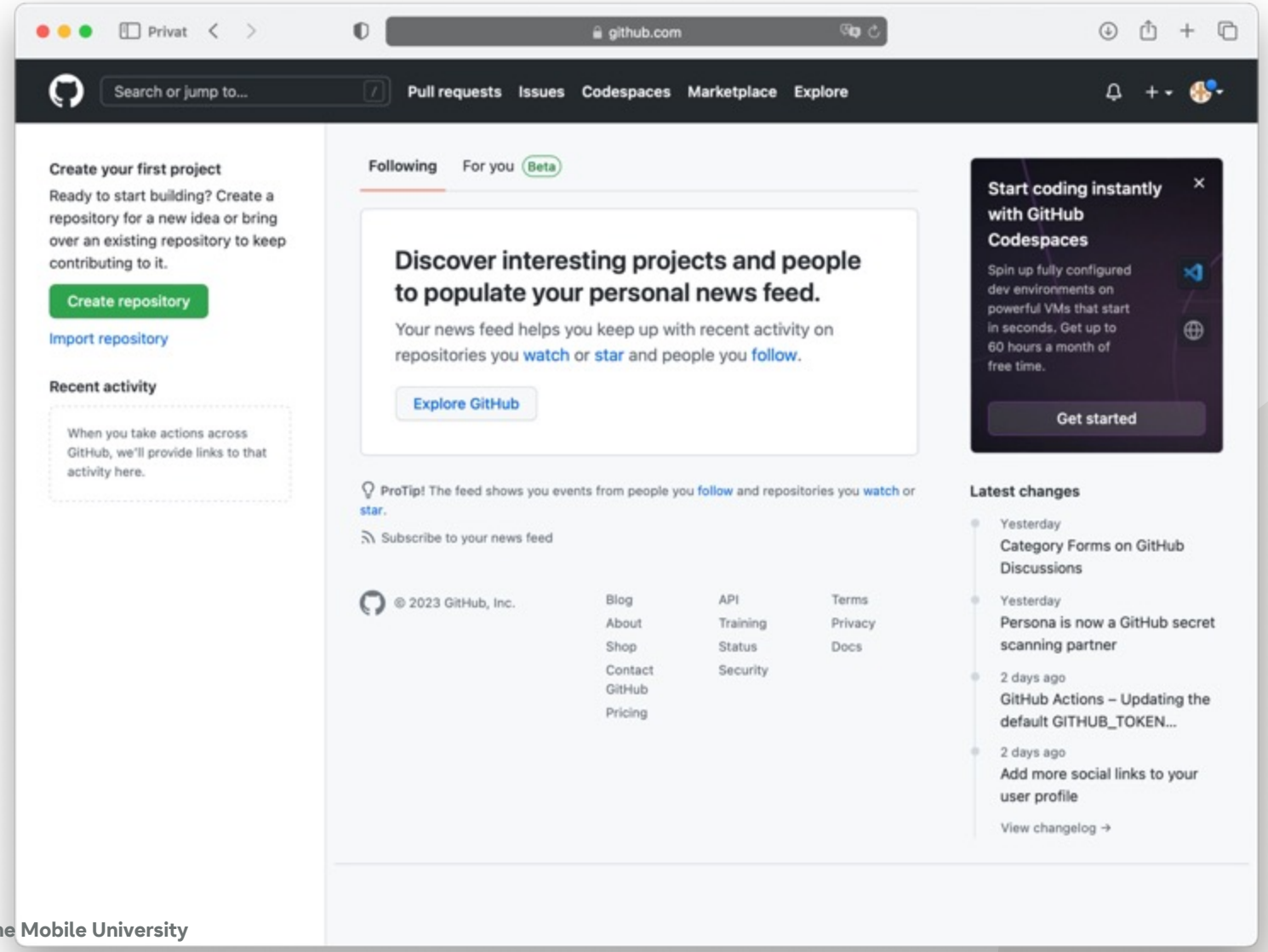
Here it is sufficient to cl
"continue" to click



select
"continue for free"



You have made it!



Install miniconda

03

What is miniconda?

We will create, train and apply data science and machine learning algorithms using the Python programming language.

So-called packages are used for this. These are prepared programs and functions that make our work easier. However, these additional packages have to be installed in advance. This can be quite challenging.

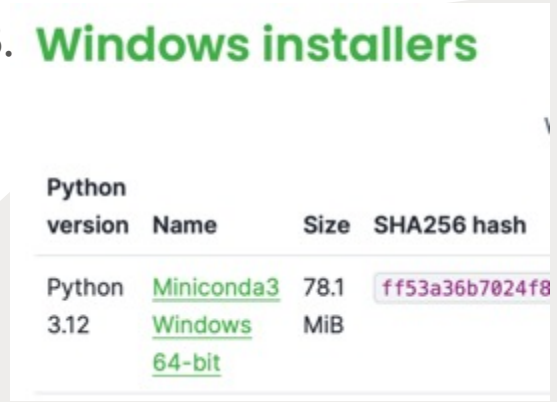
So-called package managers are used to make this easy. These facilitate the installation of these packages and ensure that everything runs smoothly.

"Anaconda" is such a package manager for Python. We use the "small" version "miniconda".

Install miniconda

You can find the required installers at https://docs.anaconda.com/free/miniconda/miniconda-other-installer-links/.

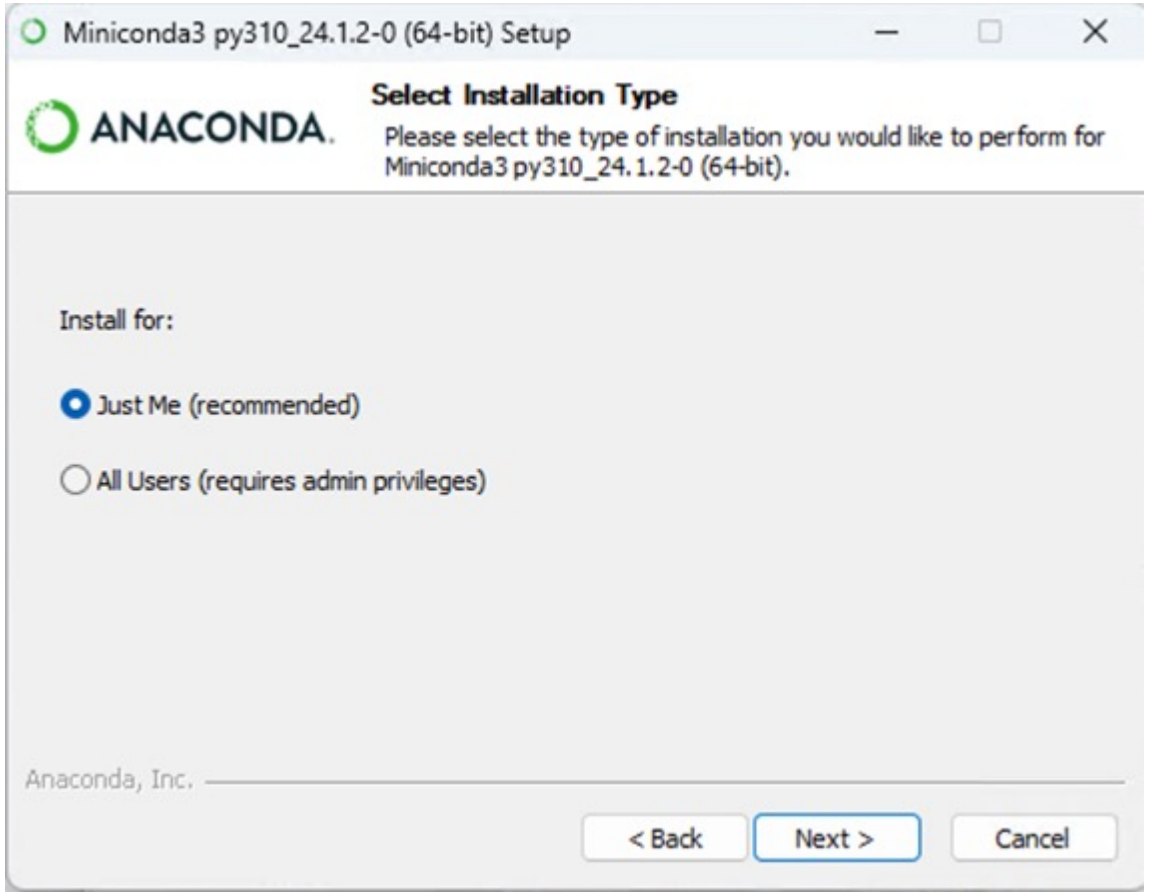
z. B. **Windows installers**



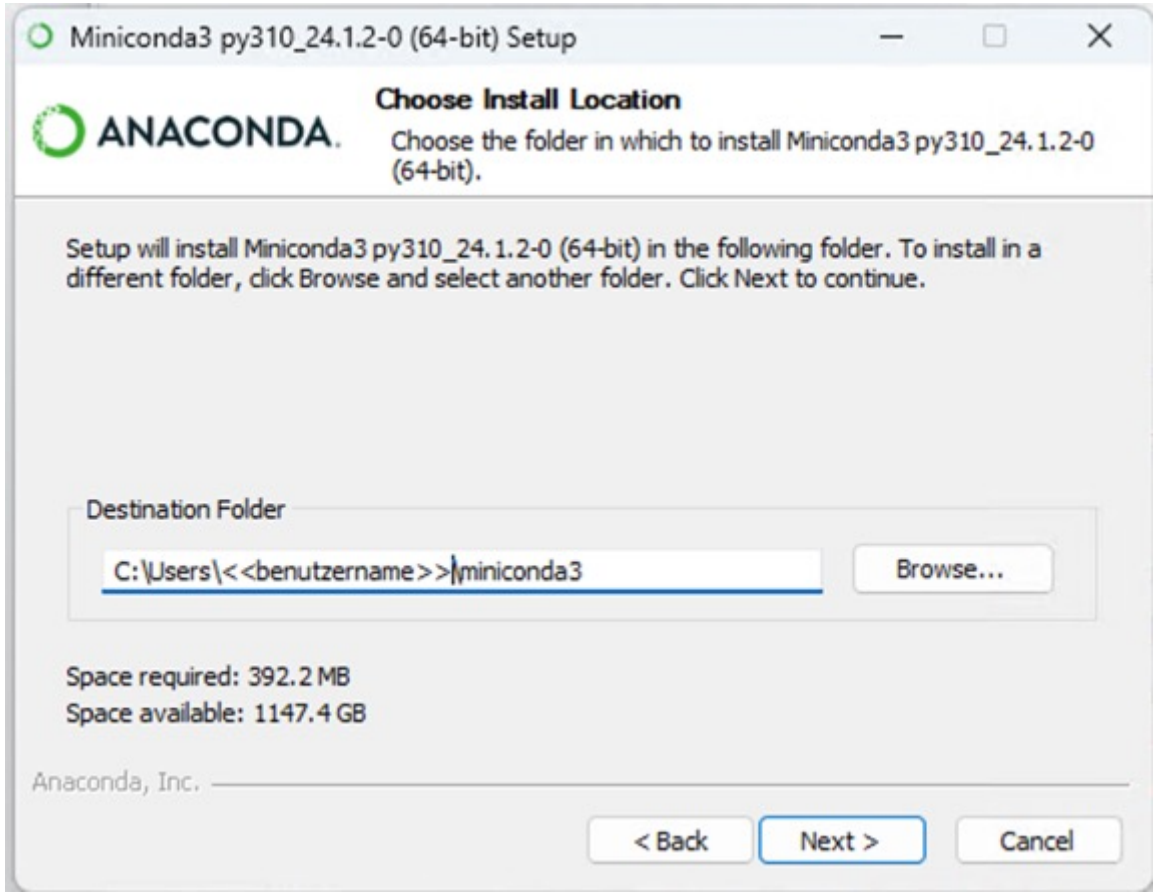
Python version	Name	Size	SHA256 hash
Python 3.12	Miniconda3 Windows 64-bit	78.1 MiB	ff53a36b7024f8

- Select the **miniconda** installer for your operating system
- Select the installer for Python 3.**12**
- Follow the instructions on the following pages

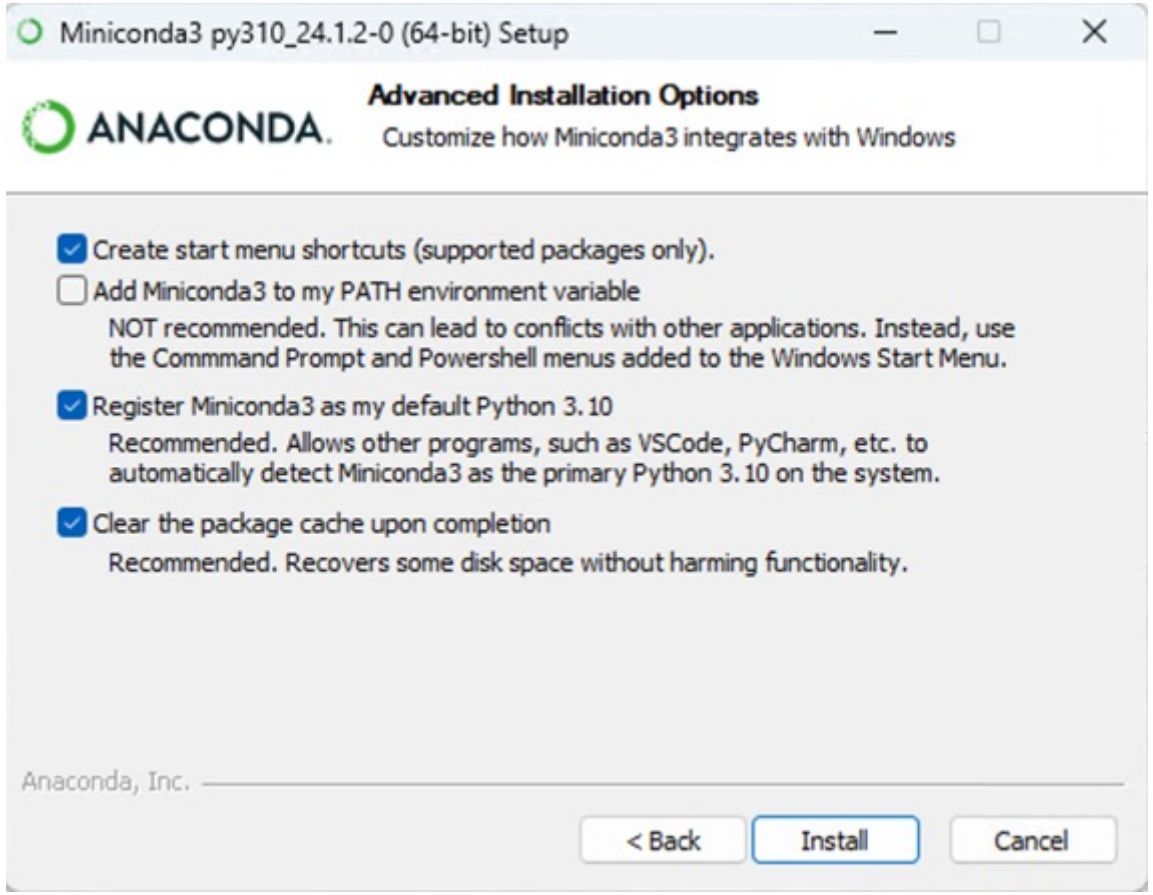
It is best to install miniconda "only for your account"



The installation takes place in your profile



Select the options shown



Complete the installation

You can carry out all further steps according to the default settings.

Wait until the installation is complete.

This activates miniconda. I will show you how to use it in section 5 of these instructions.

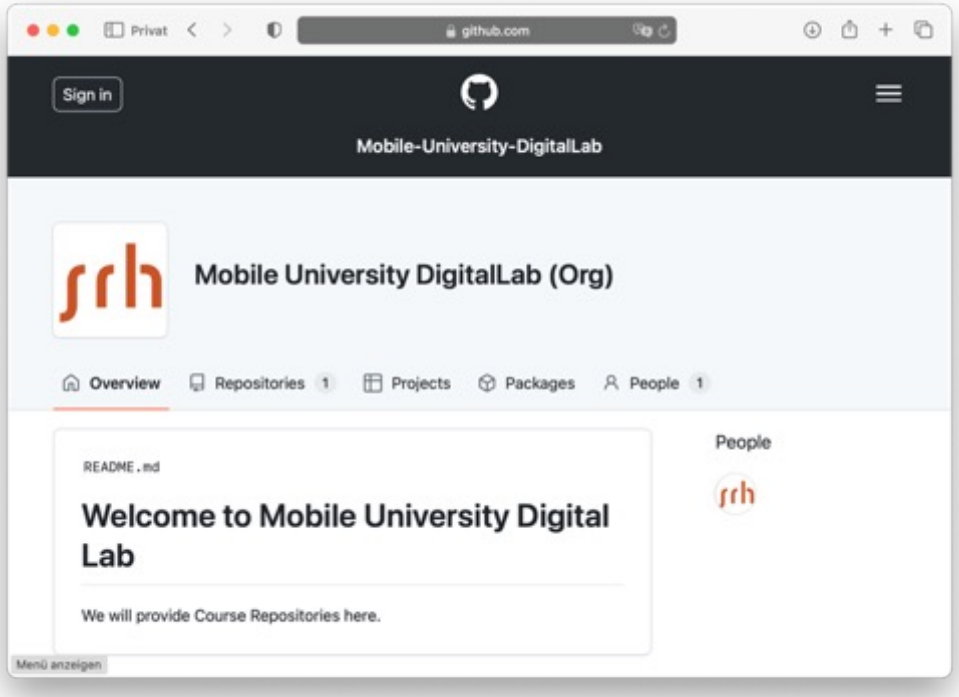
Fork and clone the course repository from github

04

Invitation to the MU github

The course materials are stored in the github repository of the Mobile University:

https://github.com/Mobile-University-DigitalLab



The repositories are not public.

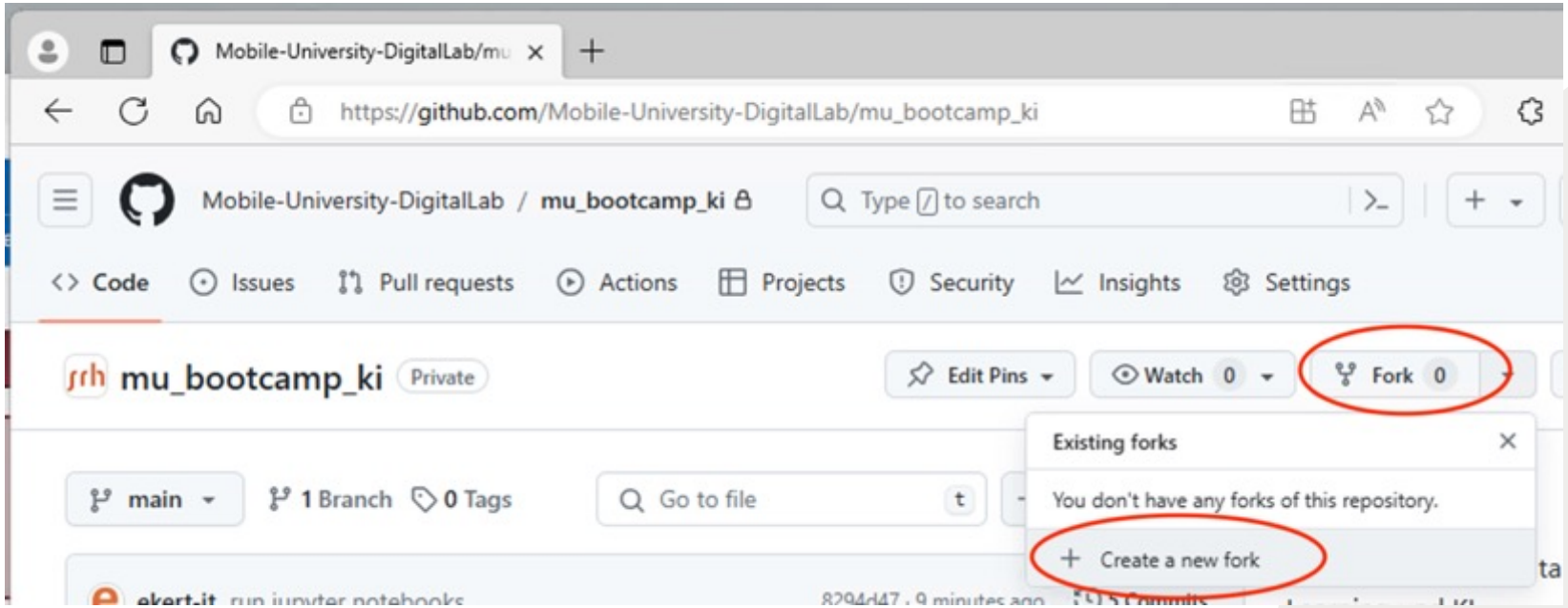
Ask your course instructor (or module leader) to **invite you to the github repository.**

You can find the repository here:

https://github.com/Mobile-University-DigitalLab/mu_bootcamp_ki

Create a fork of the repository

In order to receive your *own* copy of the course materials, you must first create your own fork of the repository. You can do this via the github website.



Create a local copy ("clone") of the repository

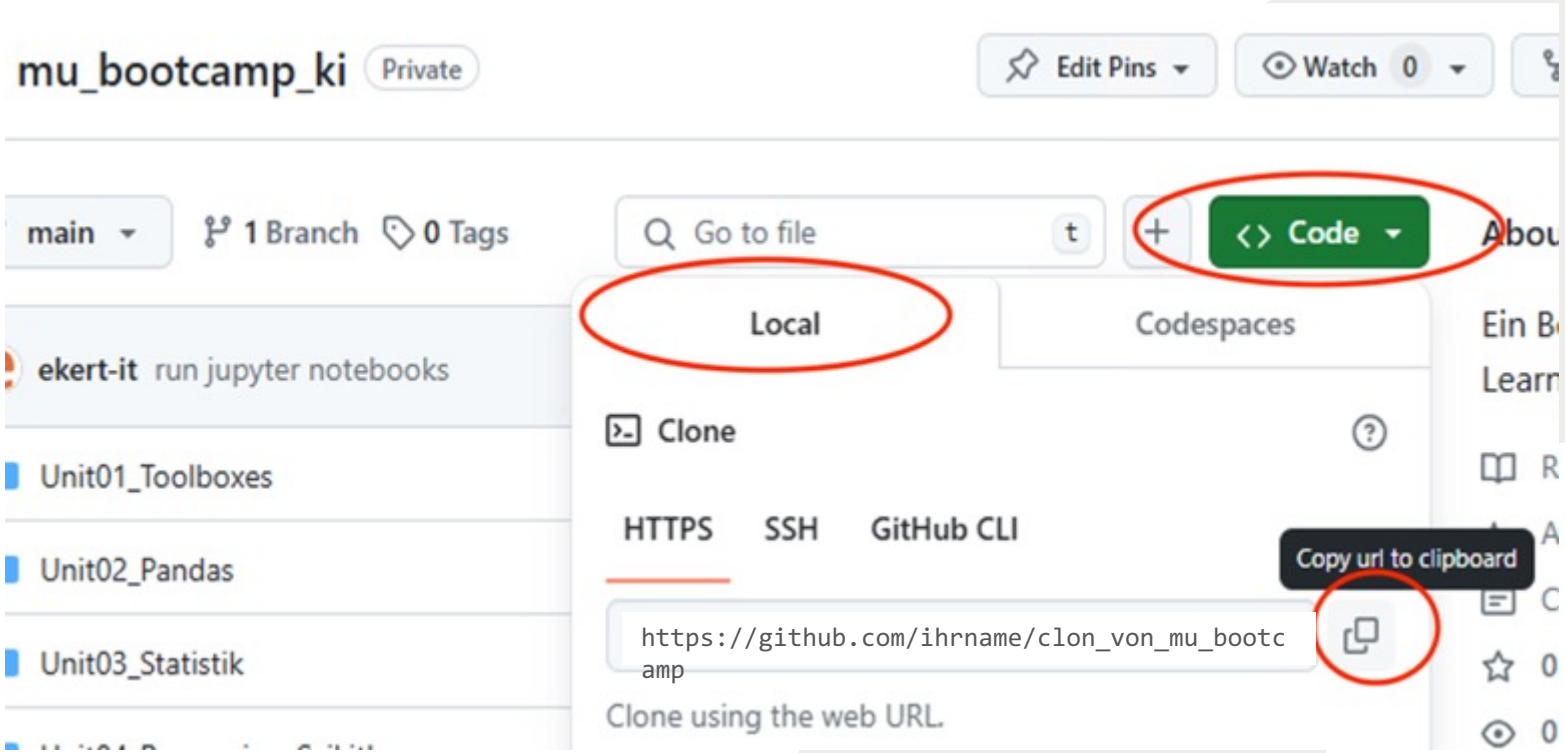
So that you can work locally on your computer, you can now clone the project you have just "forked" (from your personal github account). When cloning, all data is loaded onto your local computer.

The address of your forked repository is required for this.

You will receive these as shown

in the screen.

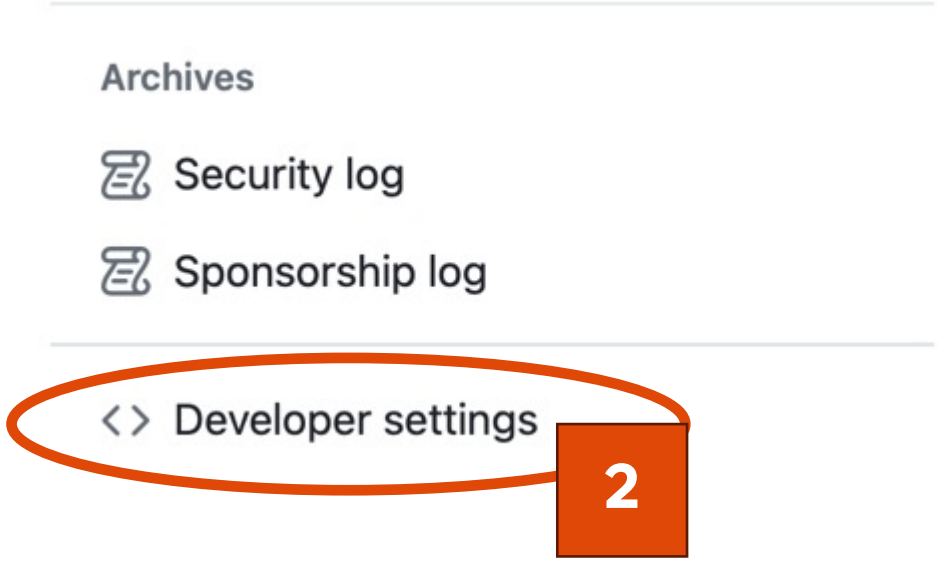
SAVE this URL.

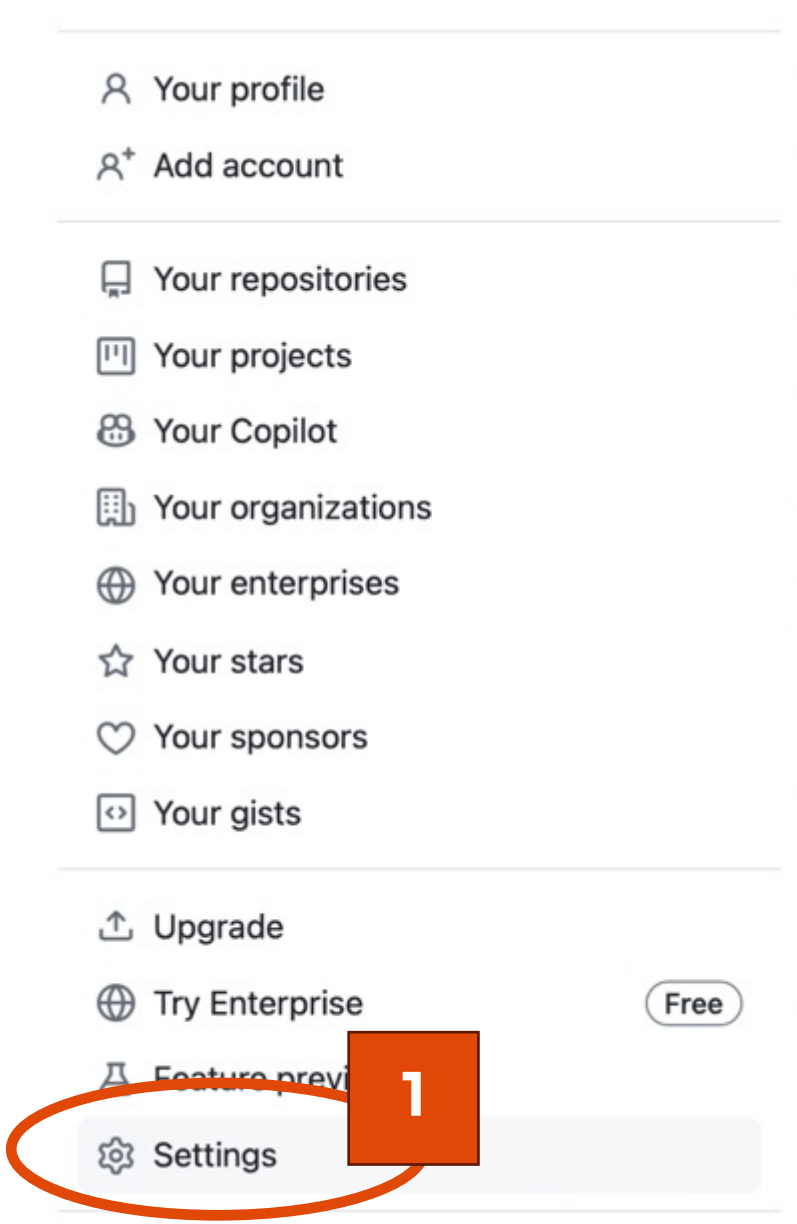


Allow access

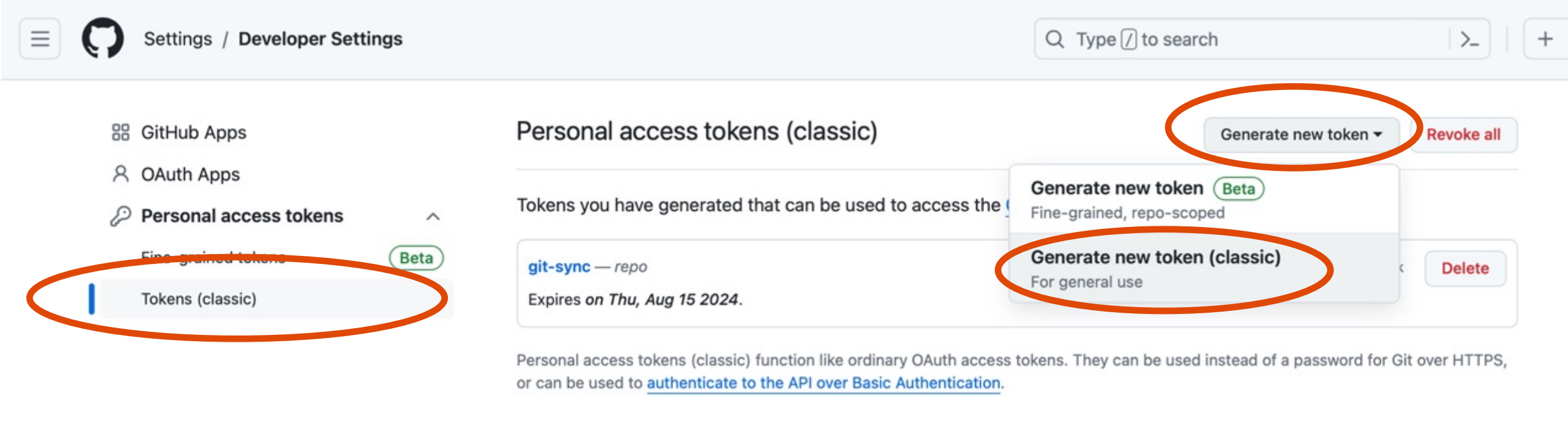
You need a token to enable github access.

You create this via the settings of your account.

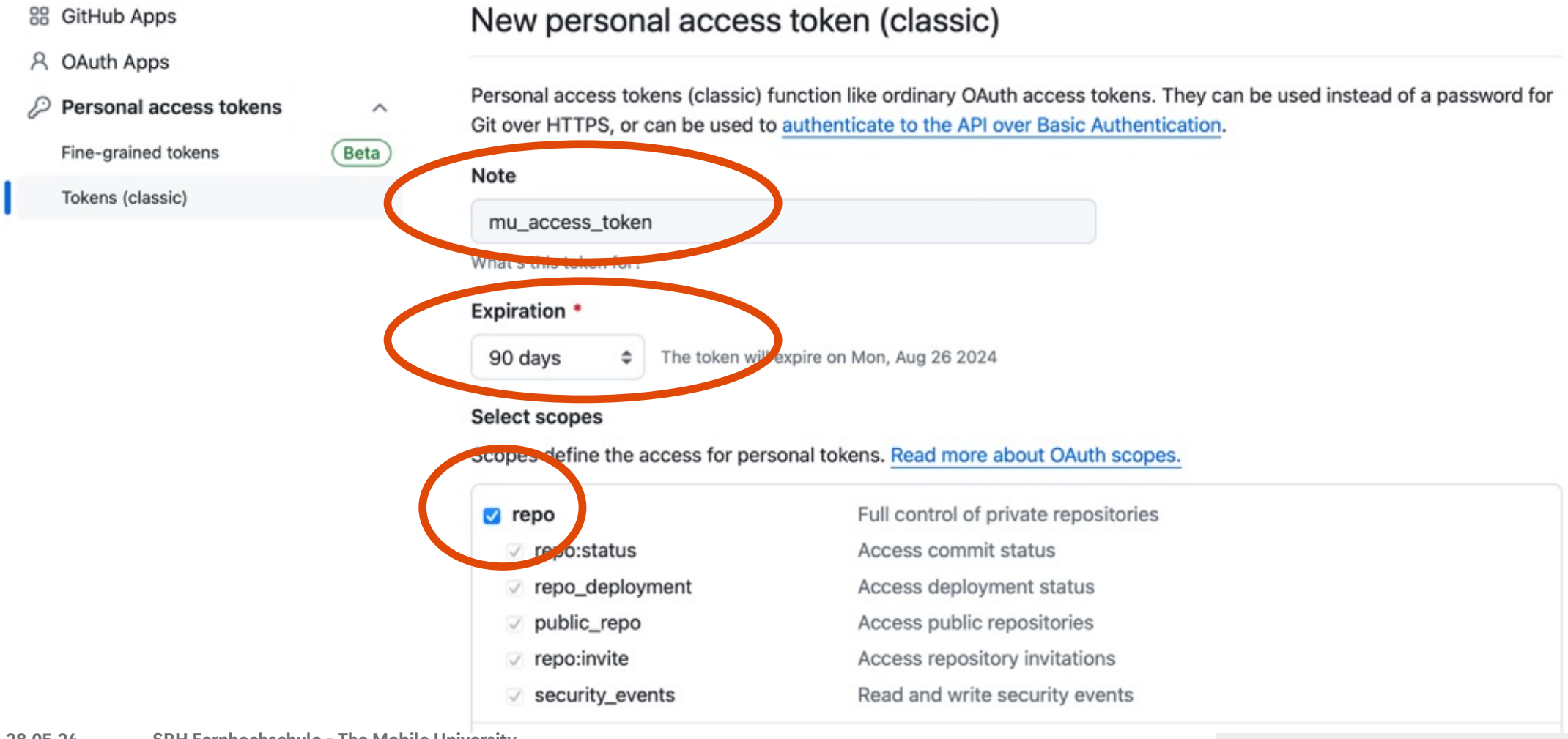




Create new token



Delegate rights to token

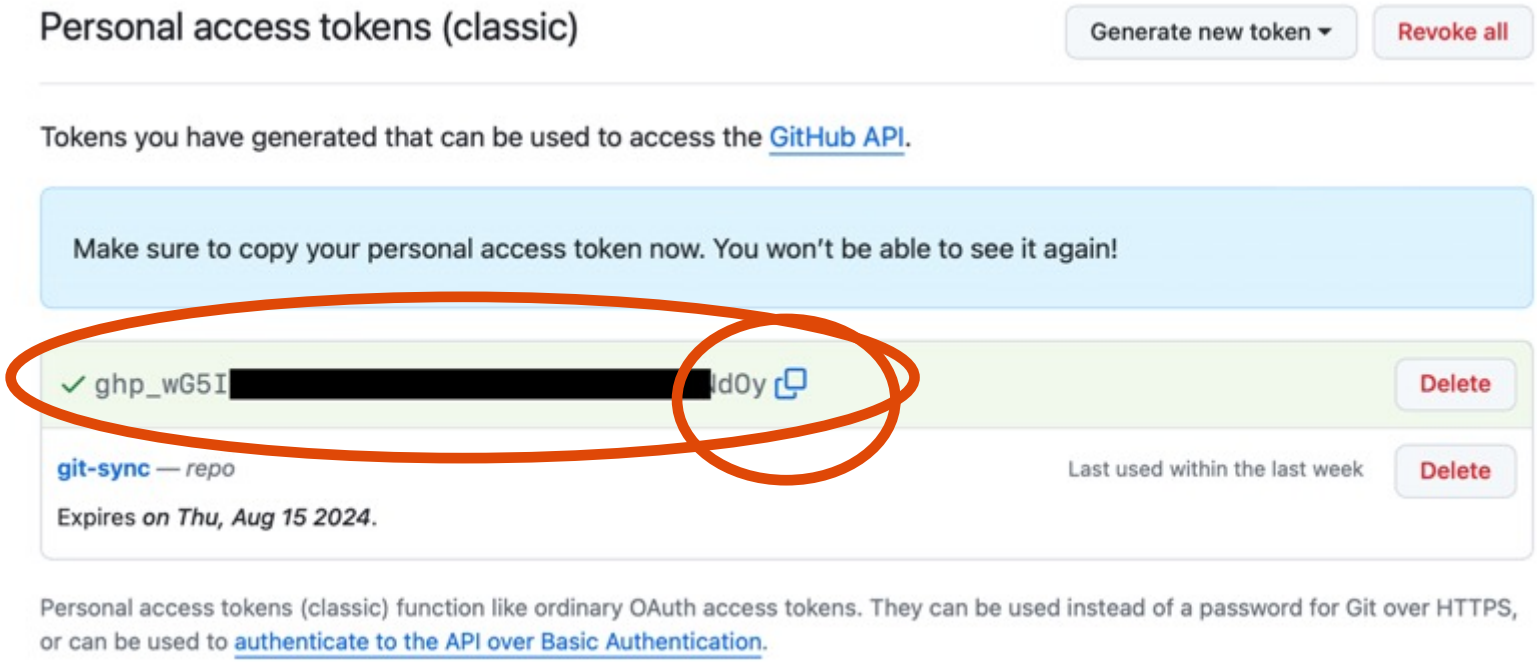


Copy your token - Save it like a password

Your new token **will be** used **like a password** together with your github **username.**

So copy this password and save it.

Protect the password from access by third parties.

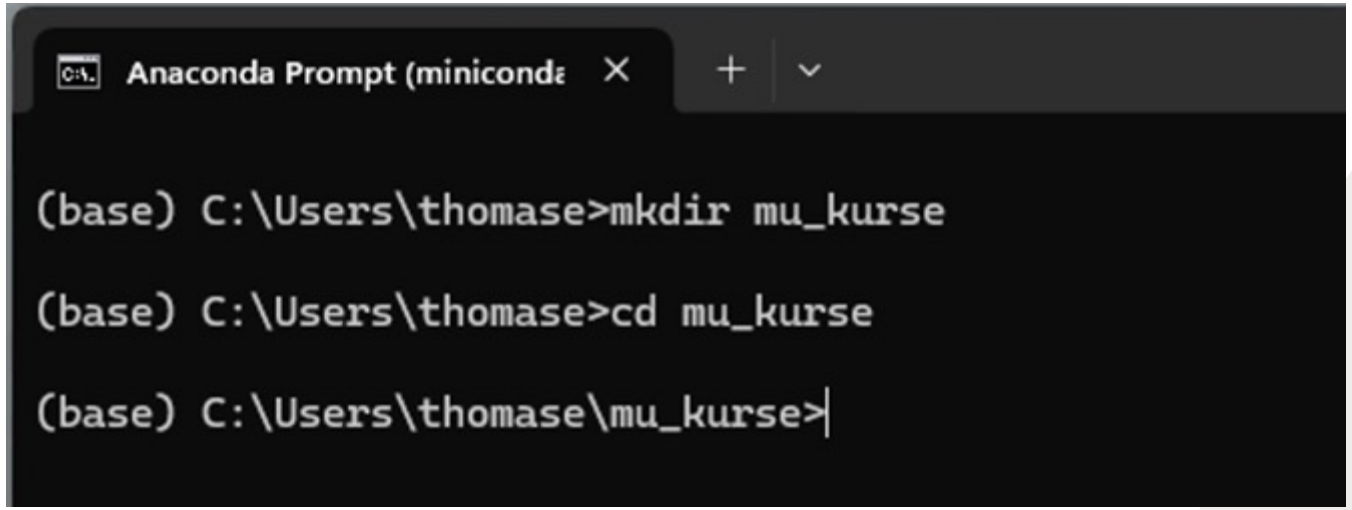


Clone repository

You first need an empty folder for your course repositories.

You can create these either via Windows Explorer or via a console:

- `mkdir mu_courses`
- `cd mu_courses`

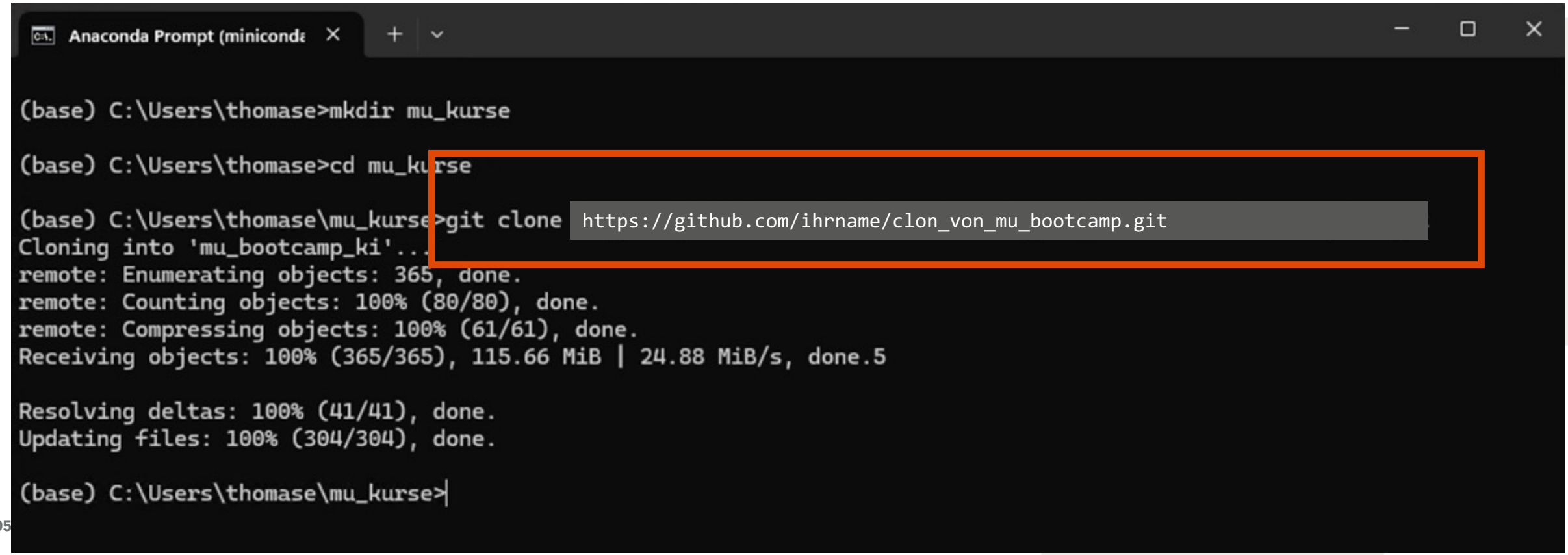


Clone repository

Now you need the URL of your (forked) repository (see slide 26).

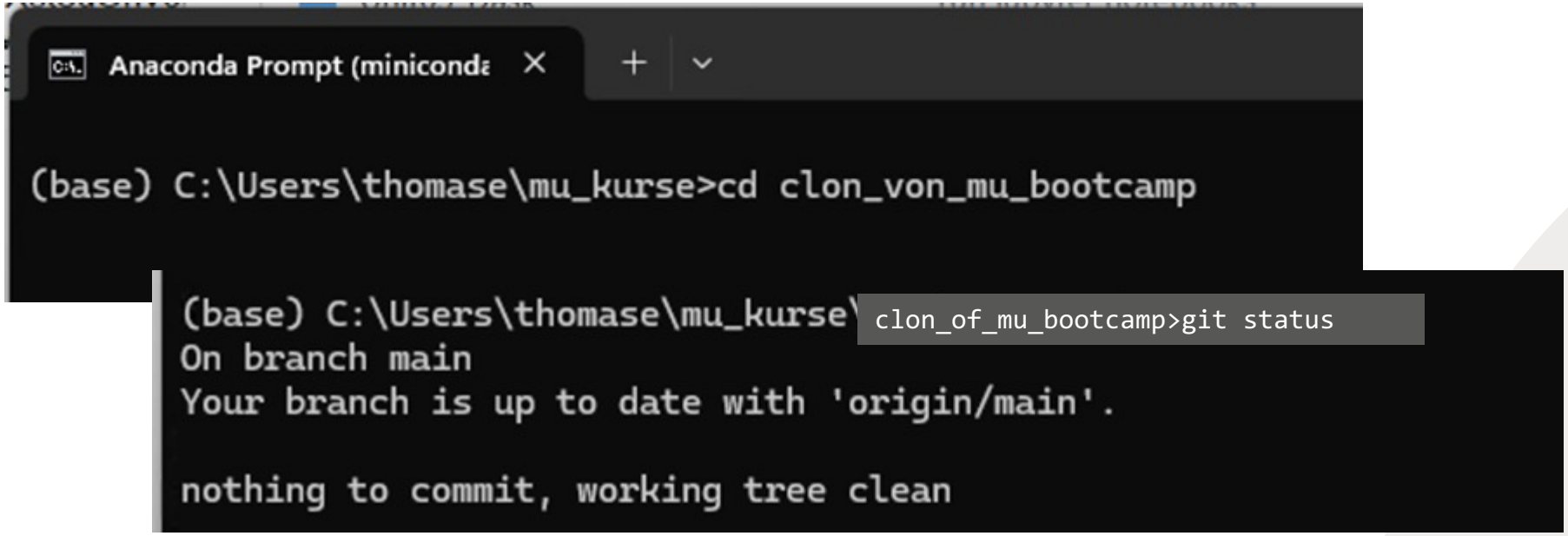
Enter the command `git clone <URL of your repo>.`

You will now be asked for your **user name.** Enter **the token created above as the password**.



Check git status

Now you can check the status of the repository with "`git status`".



Important git commands

`git add <filename>` or `git add *`

➔ transfer changed or new files to the "control" of git

`git commit -m "A commit message"`

➔ Check in the files transferred in this way. Checked-in files represent a comprehensible intermediate status in git.

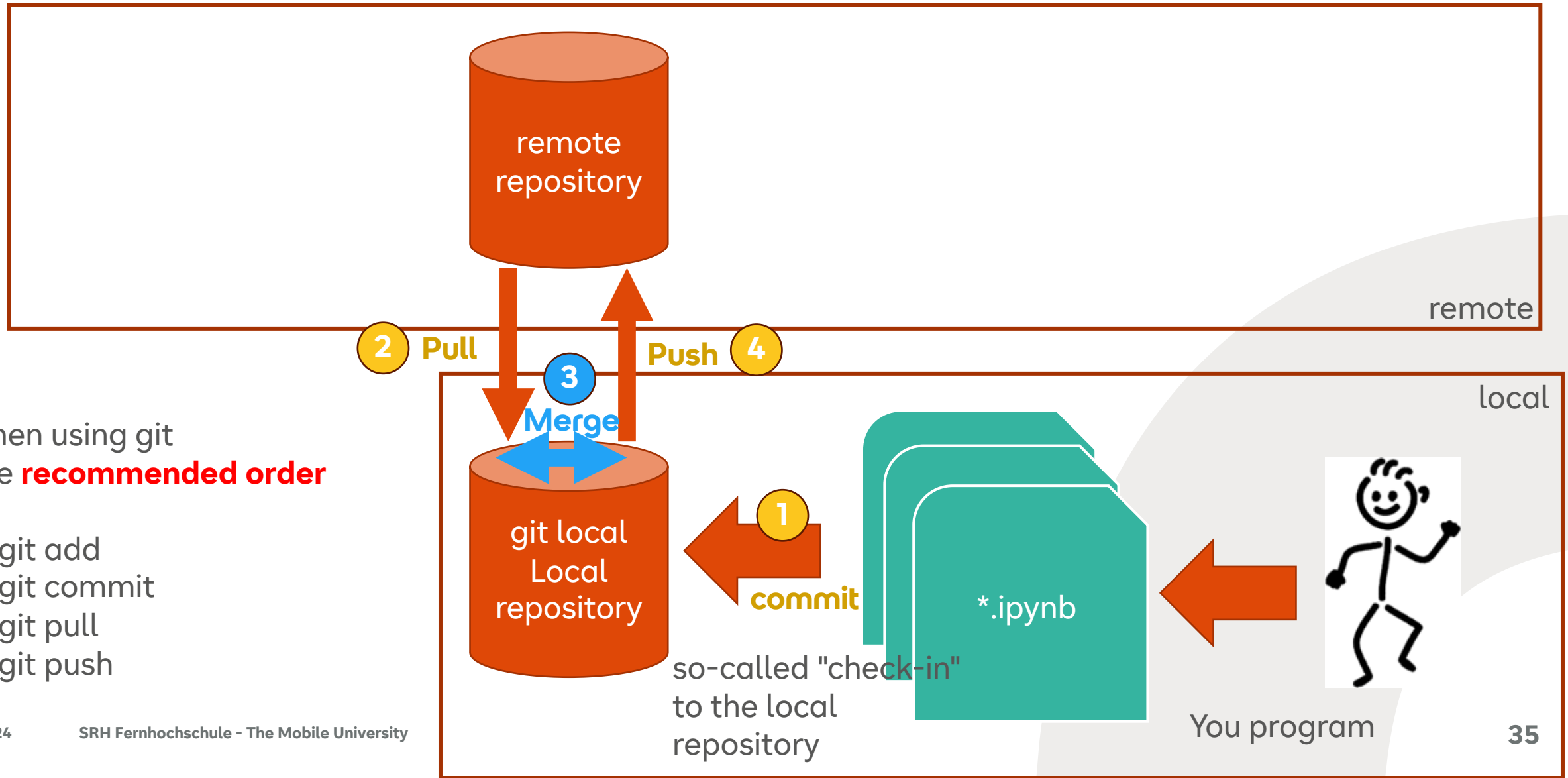
`git pull`

➔ download files from your github repository (new or modified).

`git push`

➔ upload the data checked in this way to your github repository

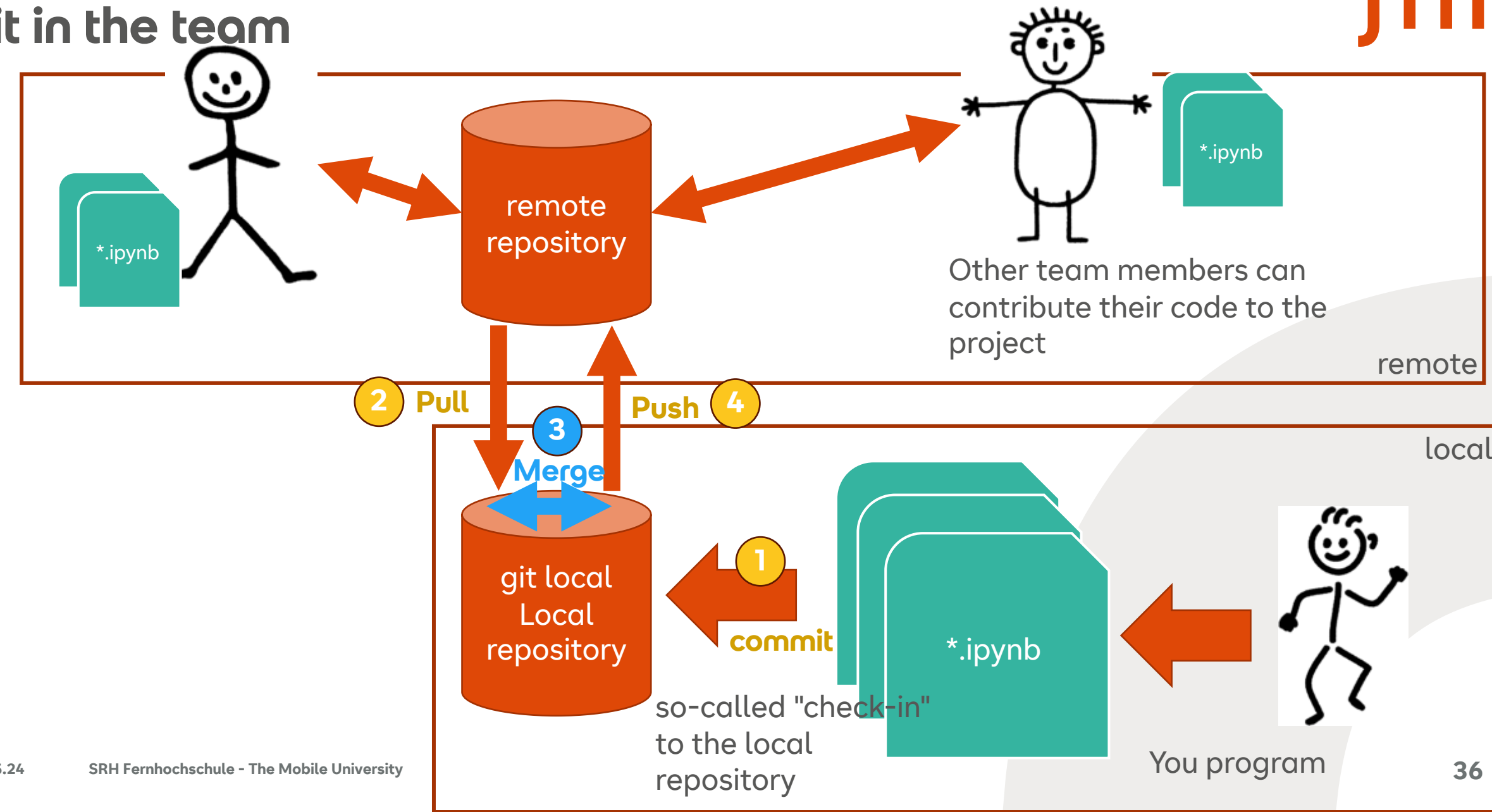
Best practice: commit -> pull -> (merge) -> push



When using git
the **recommended order** is:

- git add
- git commit
- git pull
- git push

git in the team



Prepare environment

05

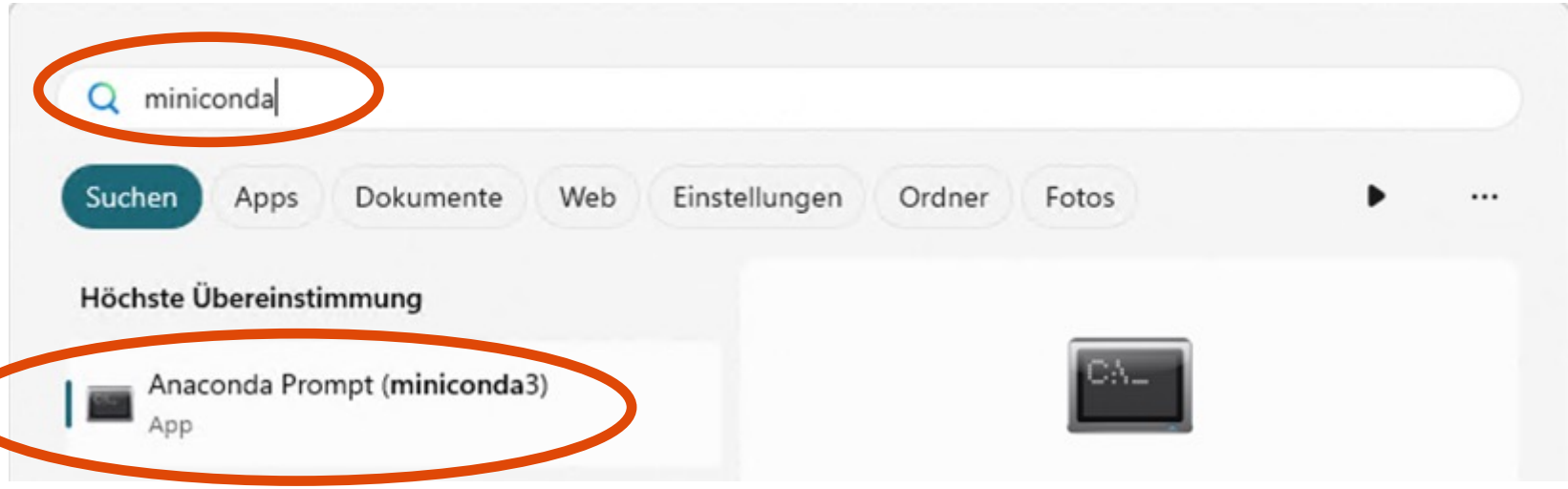
Prepare environment

As mentioned at the beginning, you need a package manager to work with Python.

You have installed miniconda for this purpose.

I have prepared various "*.yml" files for you, which you can use to create all the required packages. This takes some time, but works fully automatically.

*.yml



You will find several *.yml files in the cloned repository

Now start a "miniconda-console":

You will find a corresponding entry in your Windows Start menu.

On Mac or Linux, start a terminal.

Mac: Programs -> Utilities -> Terminal.app

Linux: Depends on the distribution.

cd

Navigate to the environment folder in the miniconda Console (or in the terminal on a Mac)

Windows:

```
cd "C:\Users\<<your_profile_name>>\mu_courses\<name_of_course>"
```

or on Mac or Linux

```
cd ~/mu_courses/<name_of_course>
```

Install environment

Für Unit 1-12, 14:

```
conda env create -f mu_bootcamp_base.yml
pip install graphviz
pip install svgling
```

Für Unit 13 Dashboards

```
conda env create -f mu_bootcamp_bokeh.yml
```

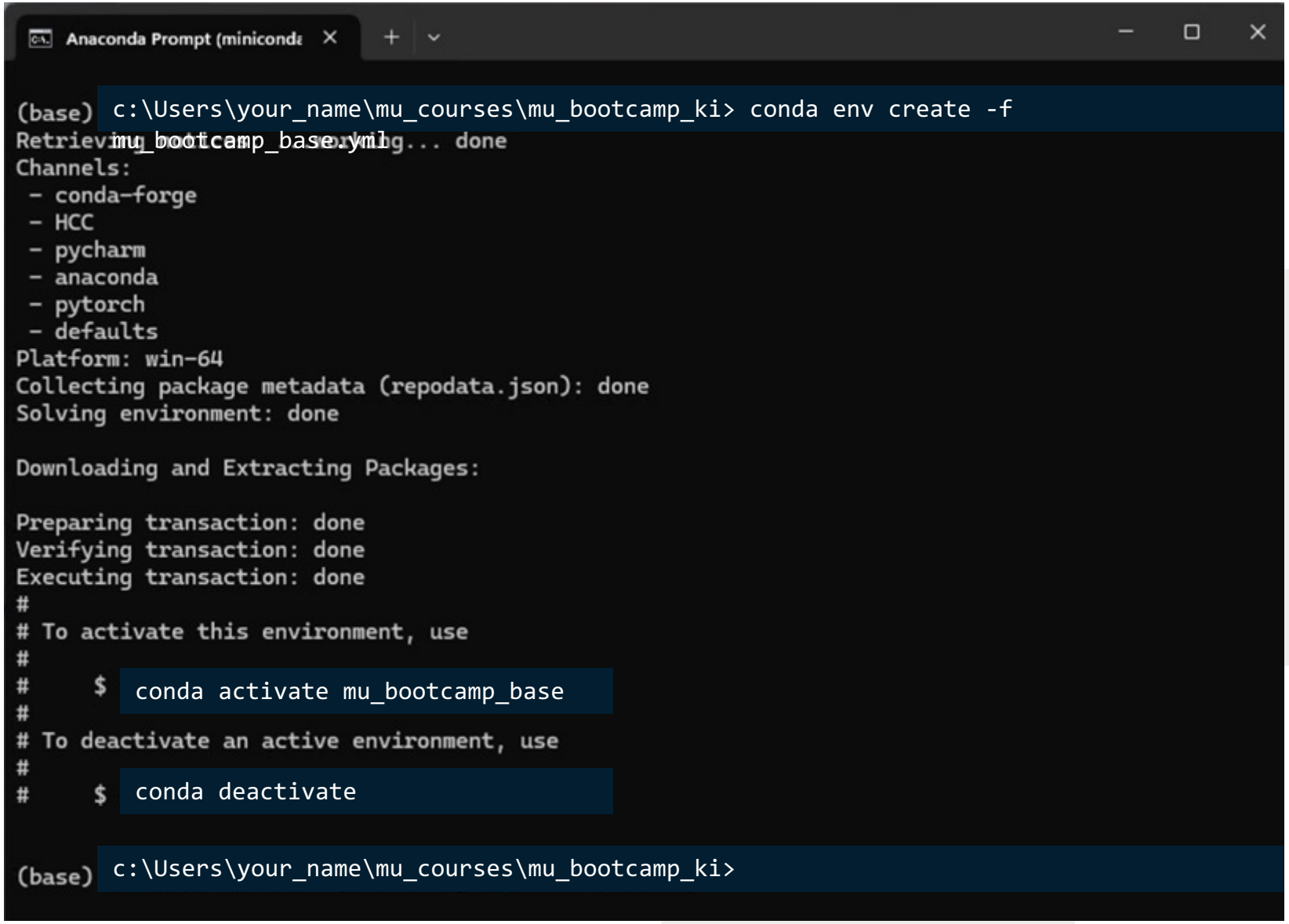
Für Unit 15 Xplainable AI

```
conda env create -f mu_bootcamp_xai.yml
```

conda env create -f <name>.yml

The installation of the environment may take several minutes.

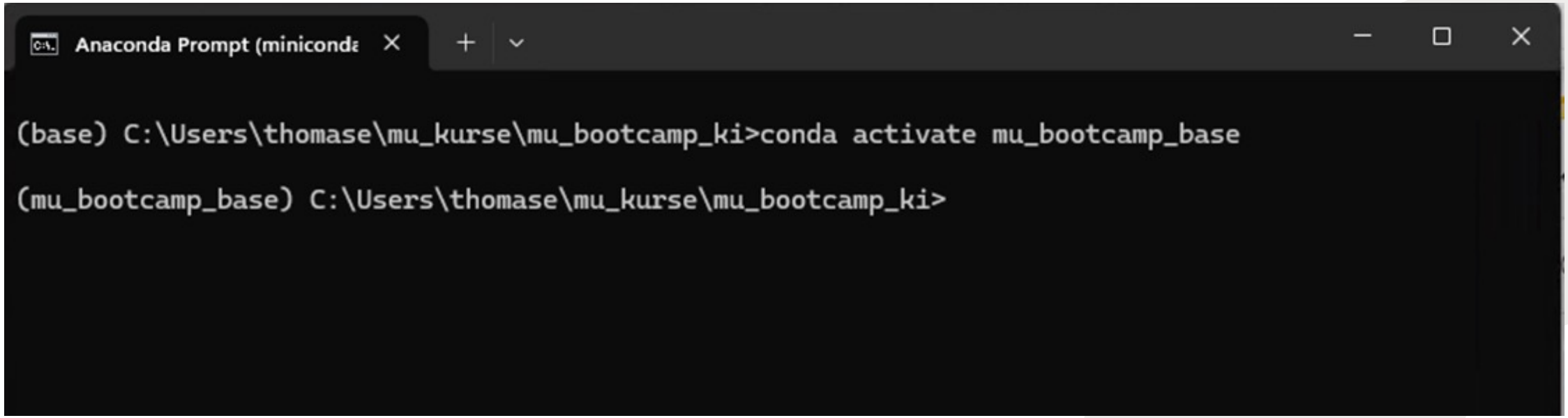
You will then see the screen shown on the right.



conda activate mu_bootcamp_base

You can now activate the environment as a test.

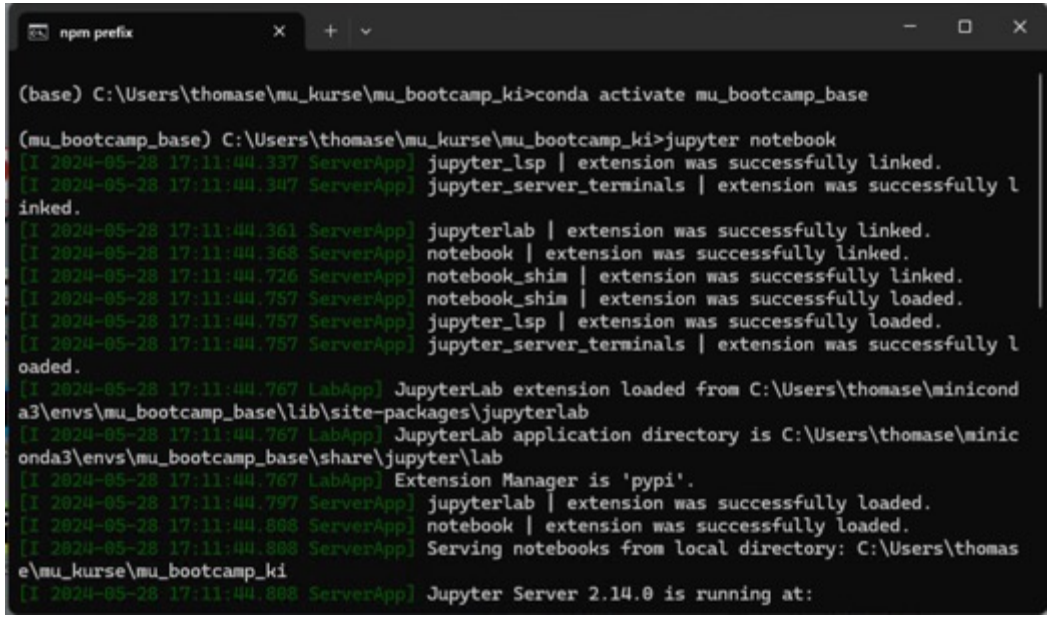
```
conda activate mu_bootcamp_base
```

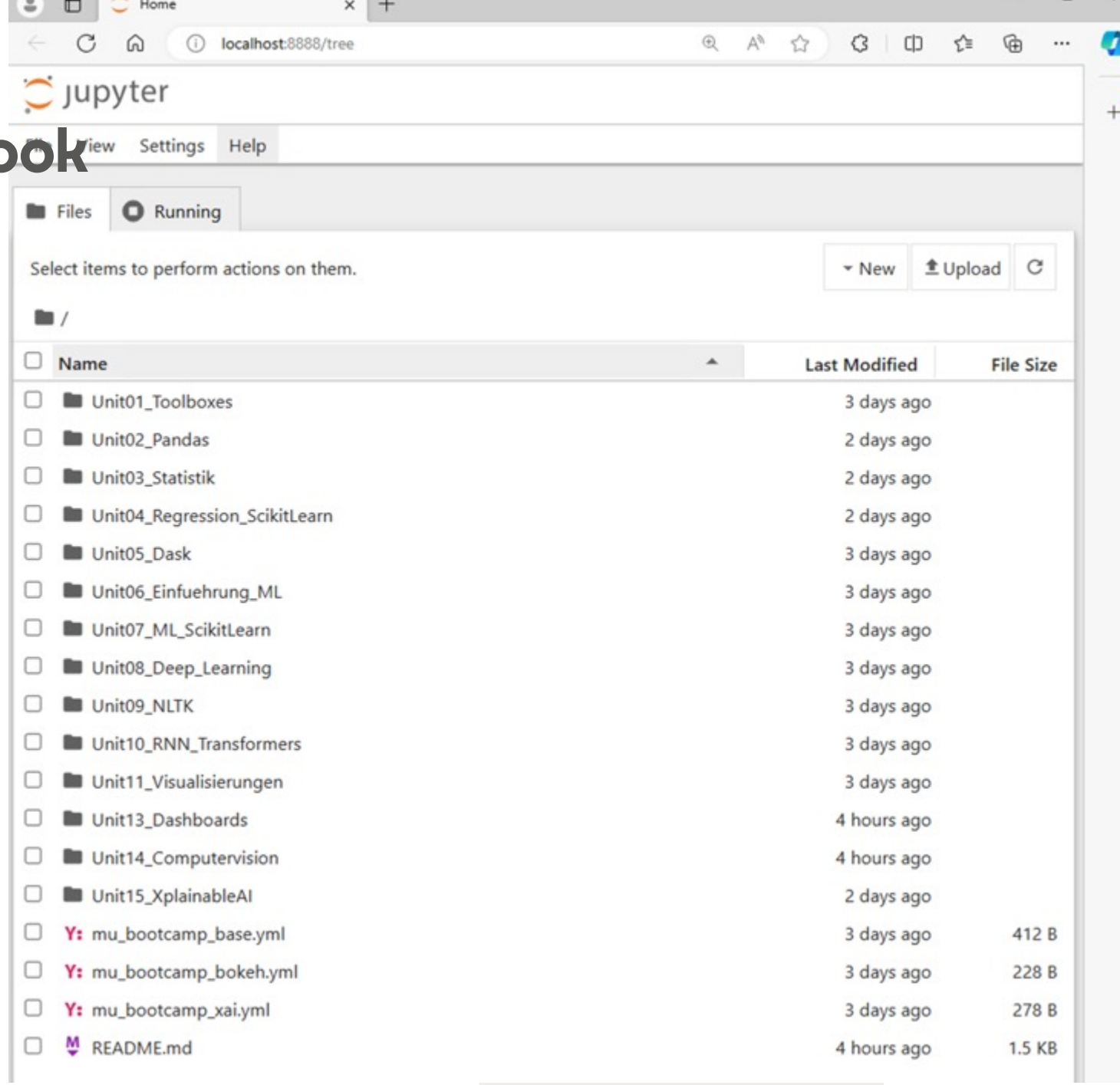


Start course - jupyter notebook

Now start the course with the command "jupyter notebook":

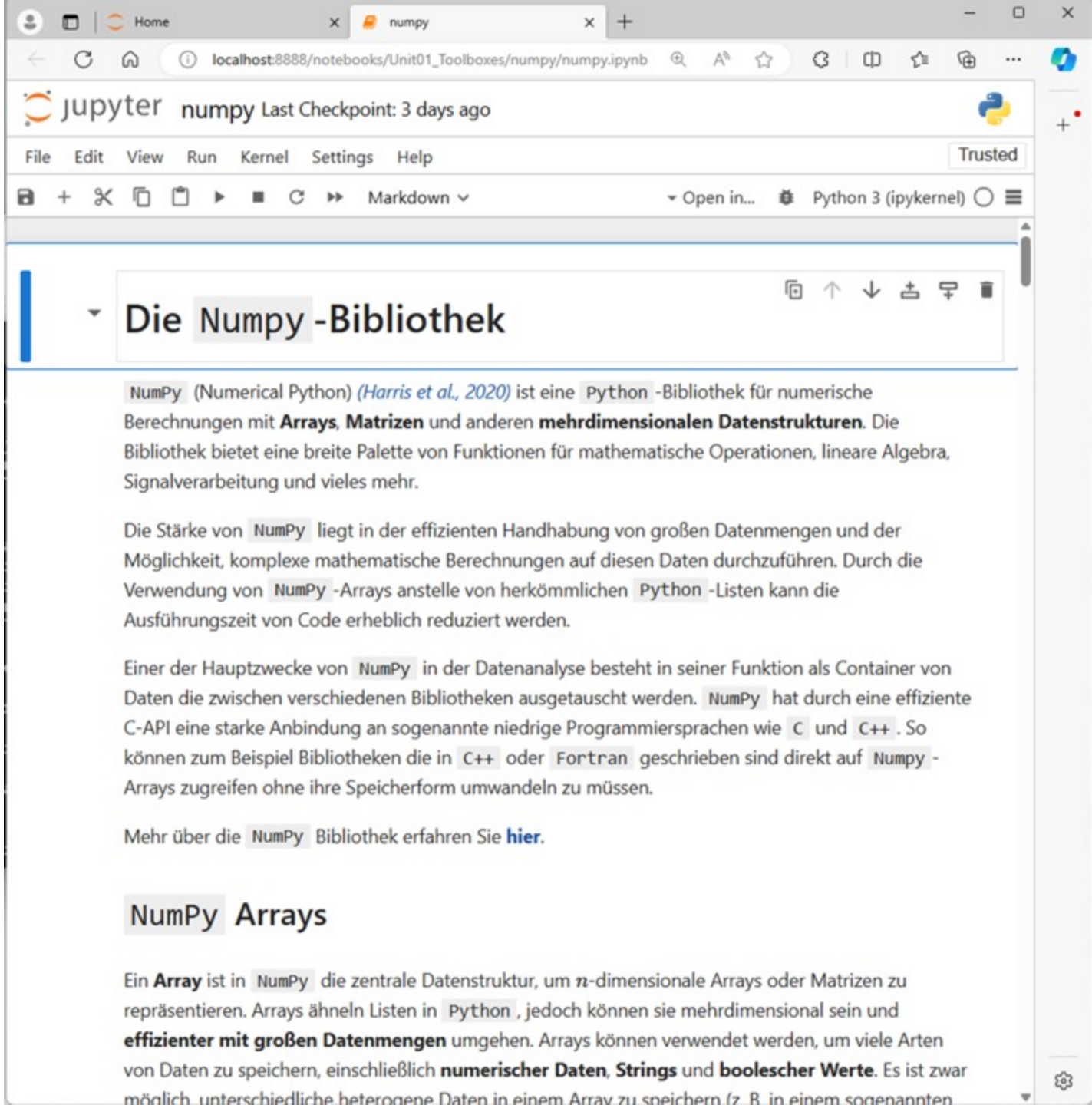
```
jupyter notebook
```





Start Jupyter Notebook

You can now start the course files by double-clicking on the folders and the contained Jupyter notebooks.



Run Python

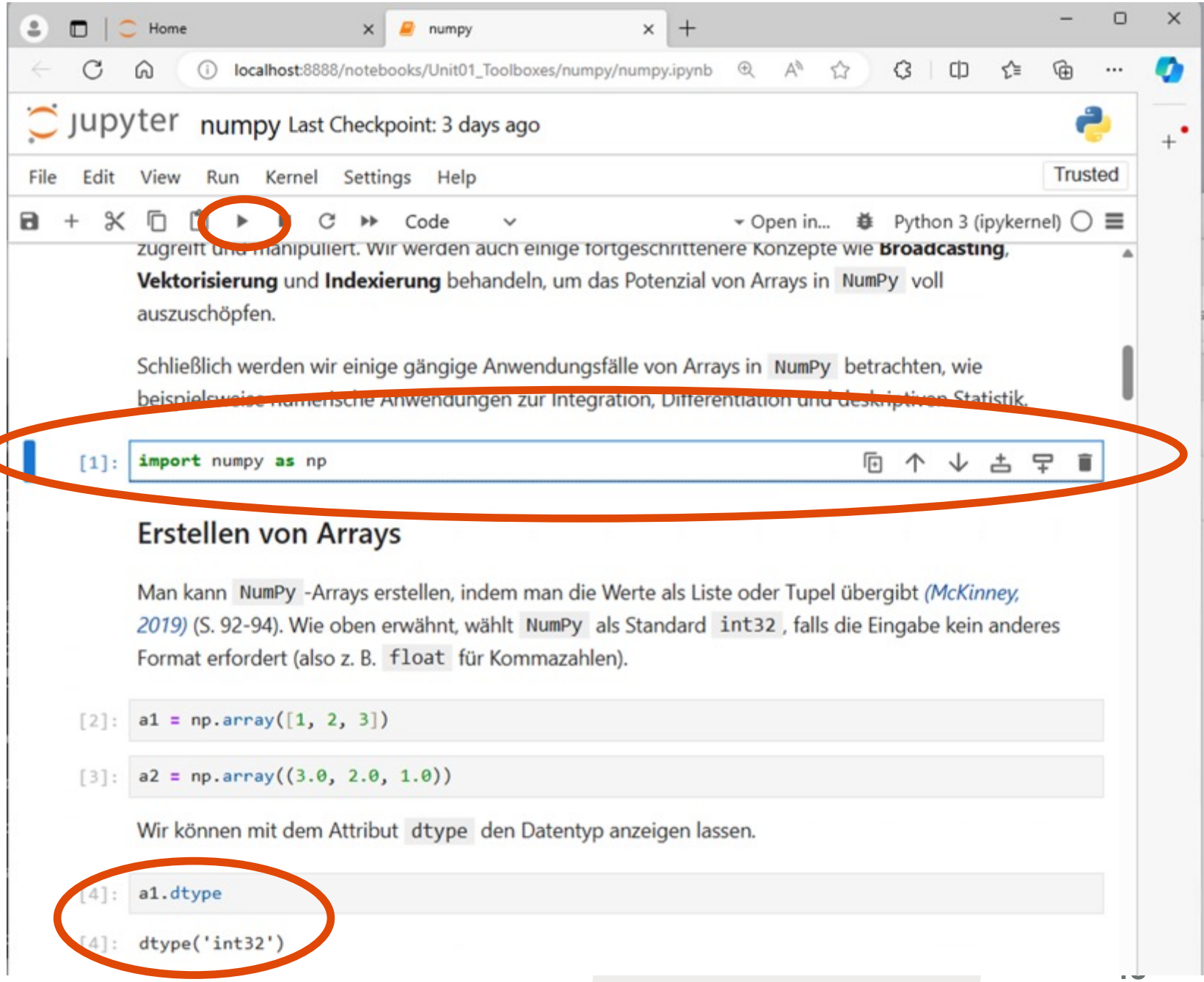
You can now execute the programming code in the notebooks.

First click in the so-called cell.

The code is executed via the "Play" button.

Some cells do not generate any output (cells 1..3 in the screenshot). However, these must still be executed.

Cell 4 has an output.



Important: conda deactivate

Before you switch to a different environment (e.g. for another unit), you must first stop the current environment and deactivate the jupyter notebook.

CTRL-C (stops the notebook)

conda deactivate (deactivates the environment)

You did it!

Thank you very much. The necessary preparations have been completed.

Have fun with the courses!