



Passion
for life.

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.

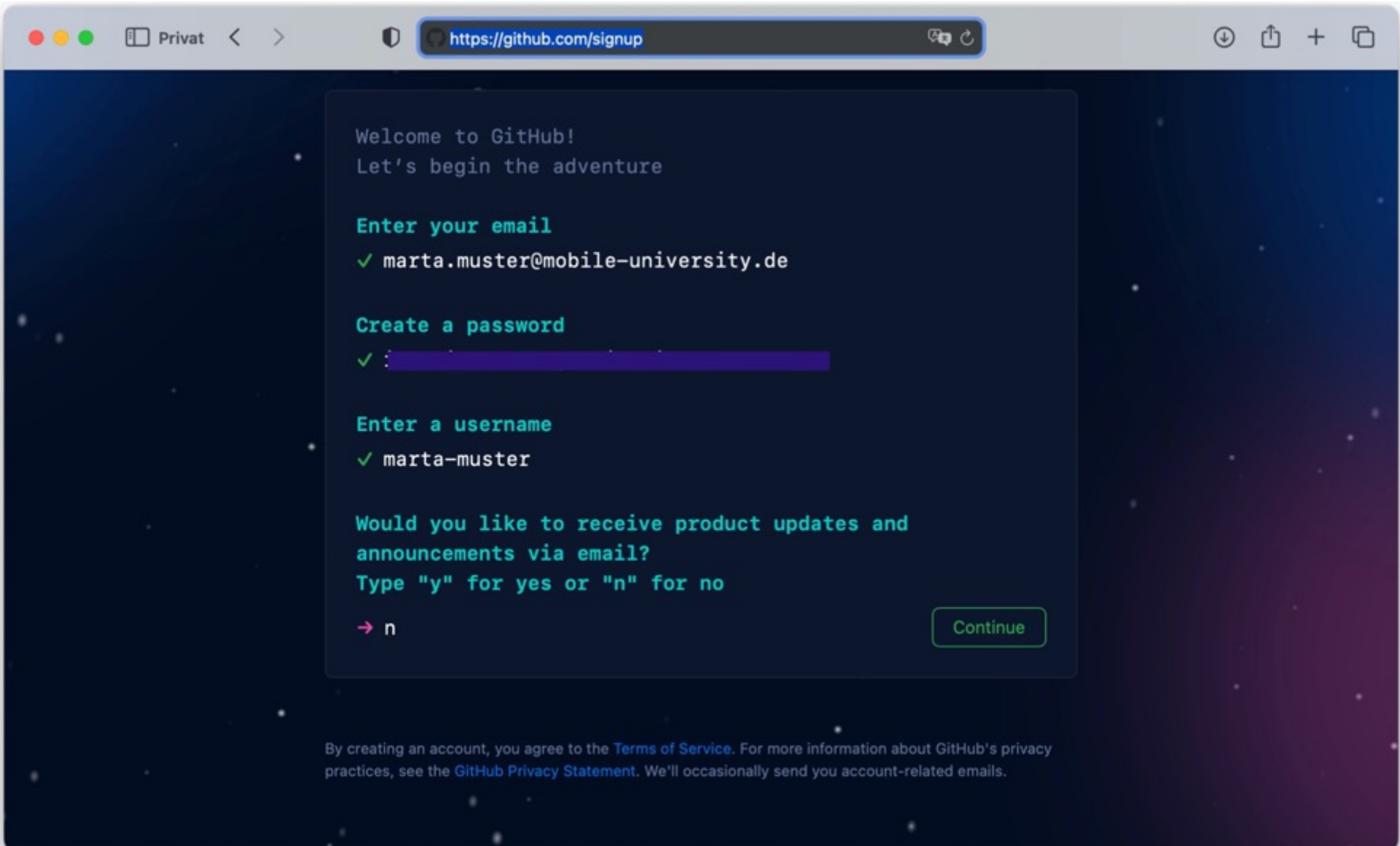
So that you can take part in the course, I would like to ask you to create an account there, i.e. to register.

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



The screenshot shows a web browser window with the URL `https://github.com/signup`. The page has a dark blue background with a starry pattern. A central white box contains the following text:

Welcome to GitHub!
Let's begin the adventure

Enter your email
✓ `marta.muster@mobile-university.de`

Create a password
✓ [password field]

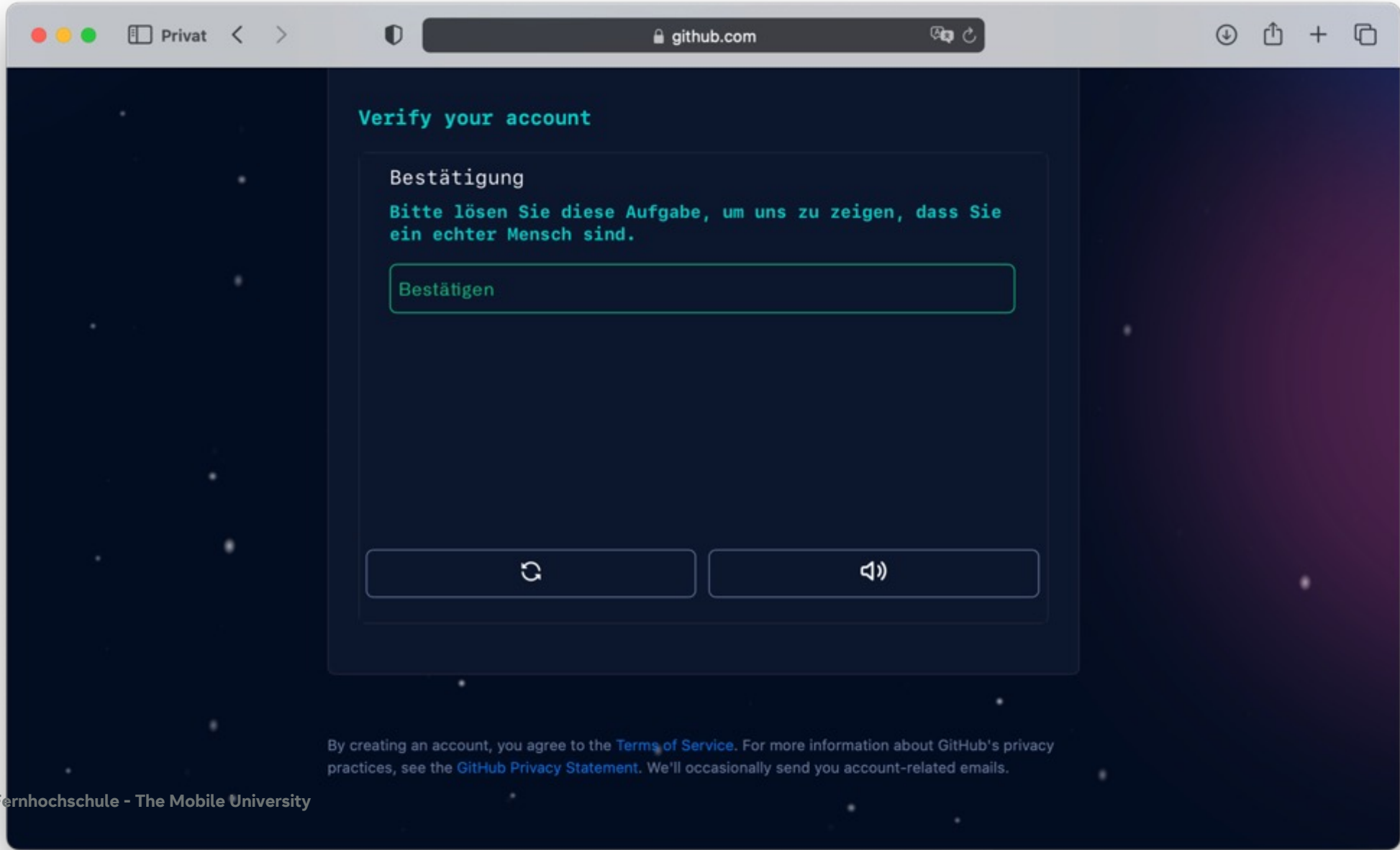
Enter a username
✓ `marta-muster`

Would you like to receive product updates and announcements via email?
Type "y" for yes or "n" for no
→ `n`

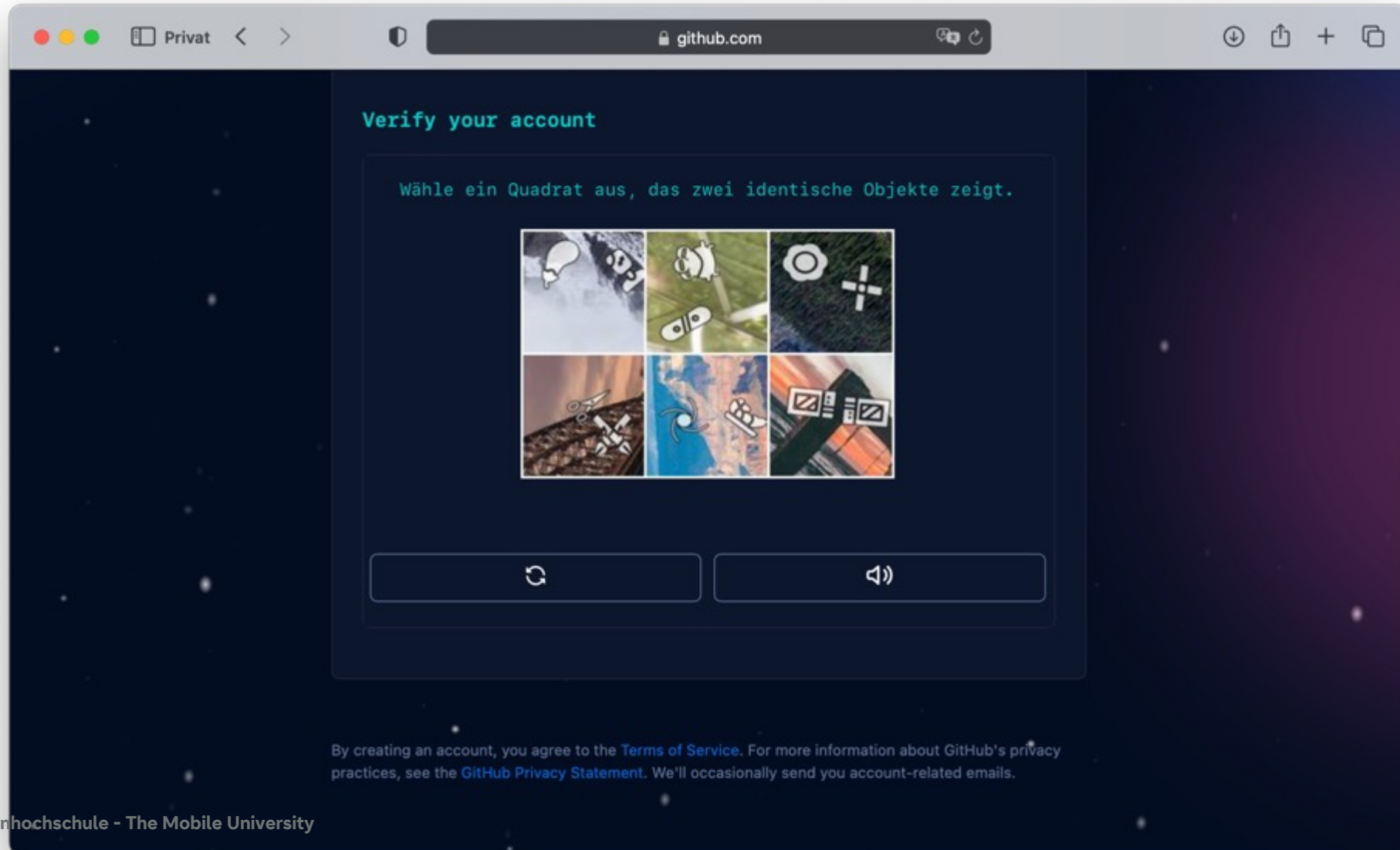
[Continue](#)

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.

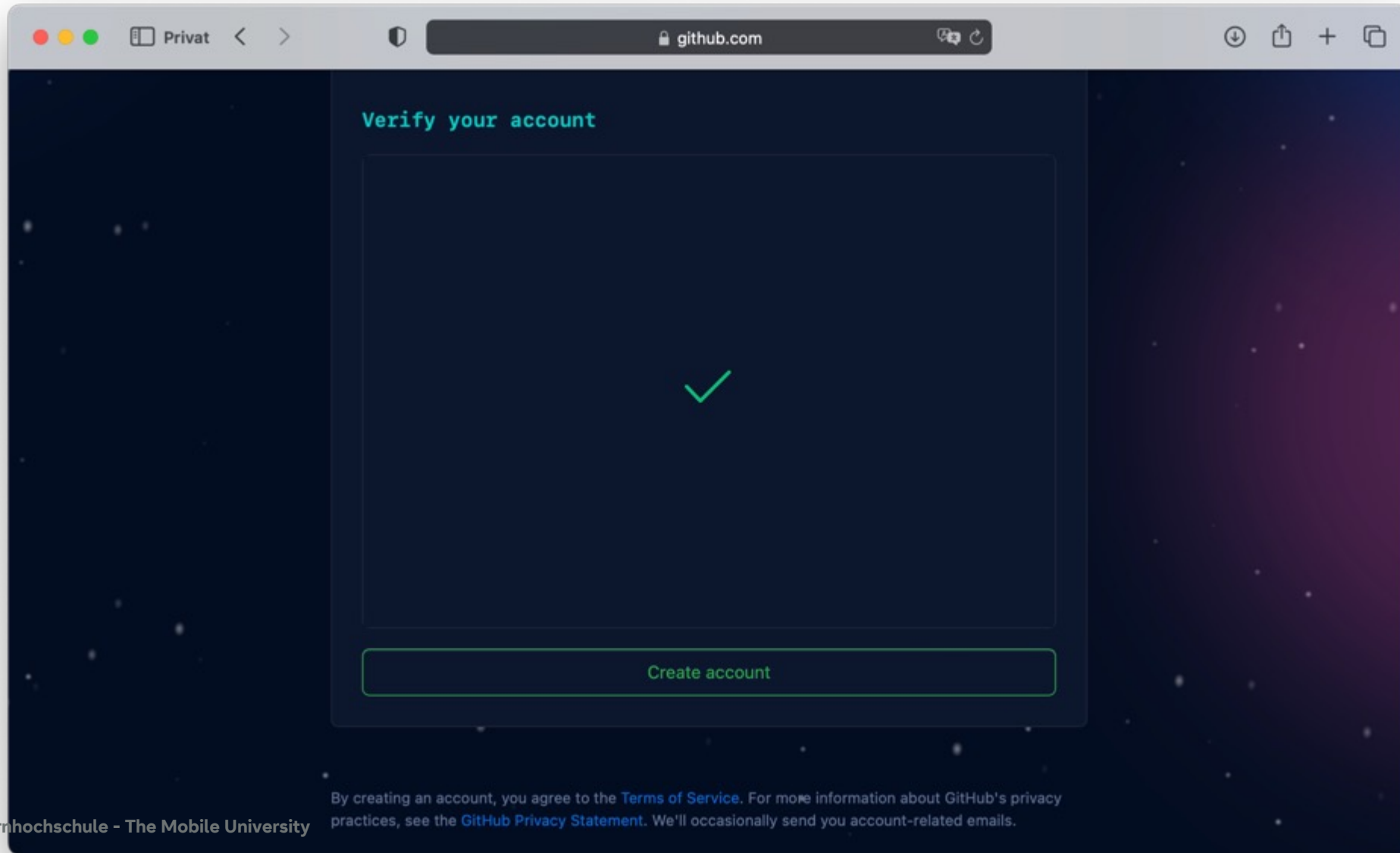
Confirm" button



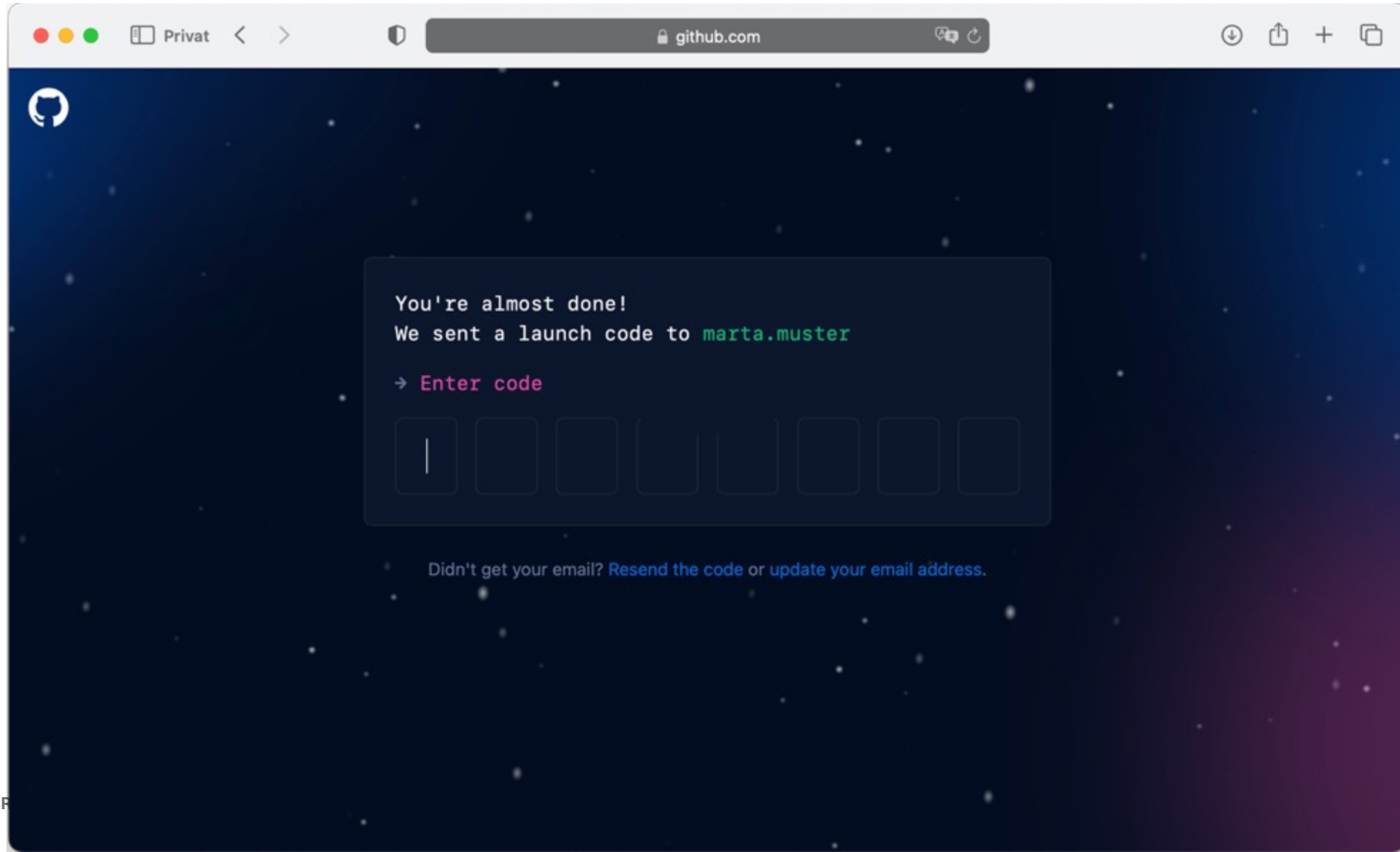
"Solve "puzzles



Create account - Click on "Create account"

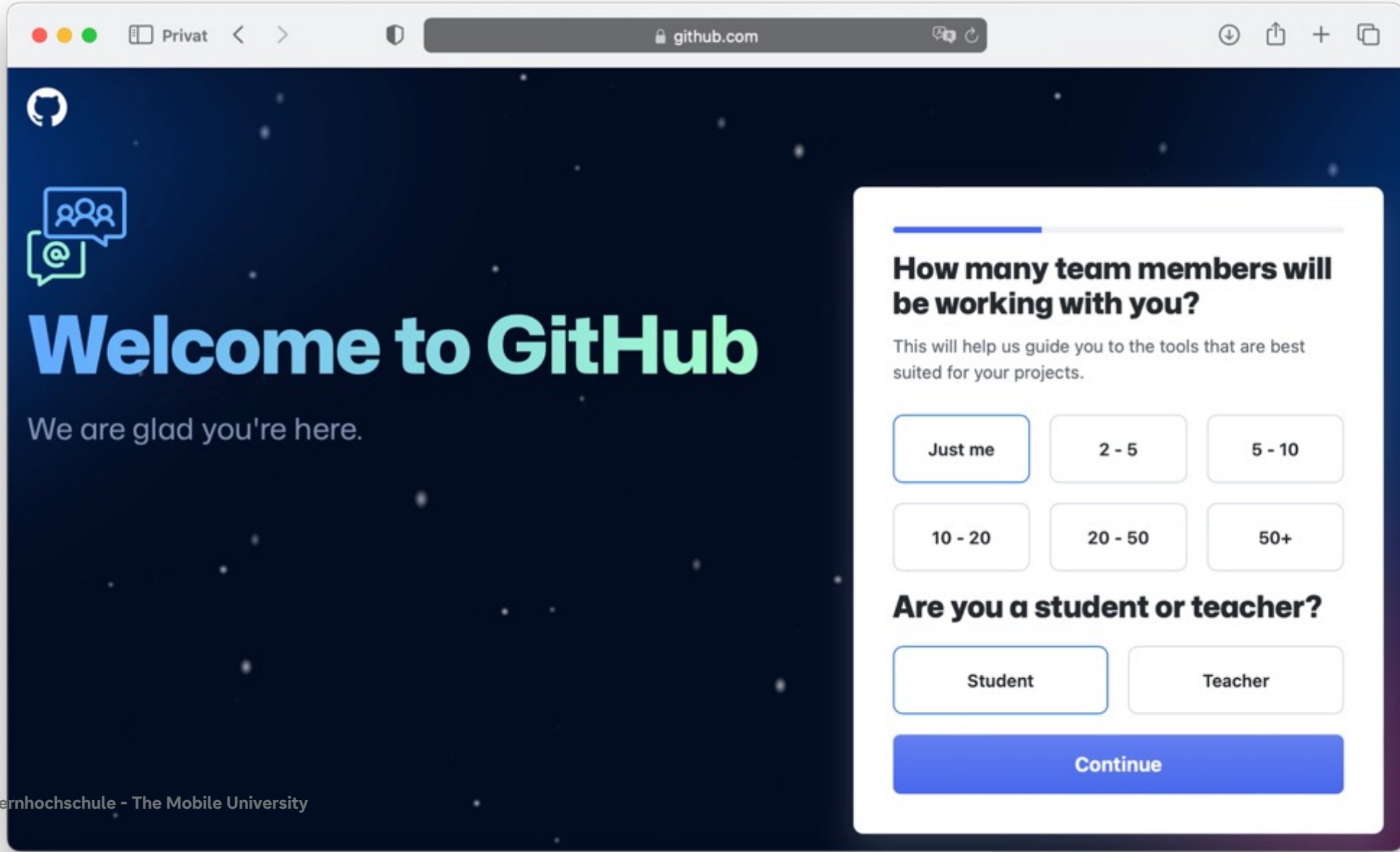


Confirm the code you received by e-mail



The screenshot shows a web browser window with the address bar displaying 'github.com'. The page has a dark blue background with a starry pattern. In the top left corner is the GitHub logo. A central white box contains the following text: 'You're almost done!', 'We sent a launch code to `marta.muster`', and '→ Enter code'. Below this text is a row of eight input boxes, with the first one containing a vertical cursor. At the bottom of the white box, it says 'Didn't get your email? [Resend the code](#) or [update your email address](#)'.

Answer further questions if necessary



The image shows a web browser window displaying the GitHub homepage. The browser's address bar shows 'github.com'. The main content area has a dark blue background with the GitHub logo and the text 'Welcome to GitHub' in large, light blue letters. Below this, it says 'We are glad you're here.' To the right, a white survey overlay is displayed. The survey has a progress bar at the top. The first question is 'How many team members will be working with you?' with a subtext 'This will help us guide you to the tools that are best suited for your projects.' There are six buttons for team size: 'Just me', '2 - 5', '5 - 10', '10 - 20', '20 - 50', and '50+'. The second question is 'Are you a student or teacher?' with two buttons: 'Student' and 'Teacher'. At the bottom of the survey is a large blue 'Continue' button.

github.com

Privat

How many team members will be working with you?

This will help us guide you to the tools that are best suited for your projects.

Just me 2 - 5 5 - 10

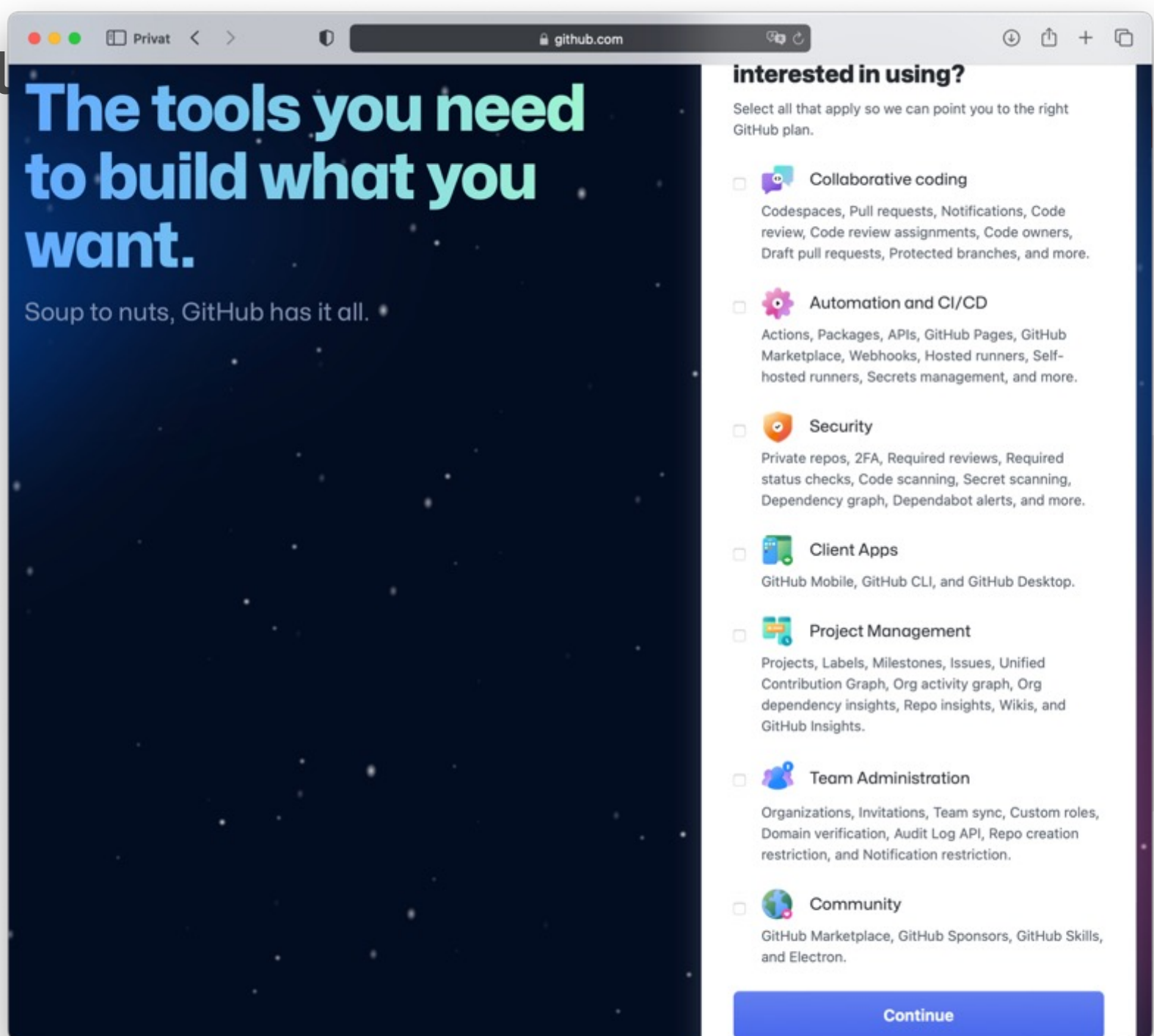
10 - 20 20 - 50 50+

Are you a student or teacher?

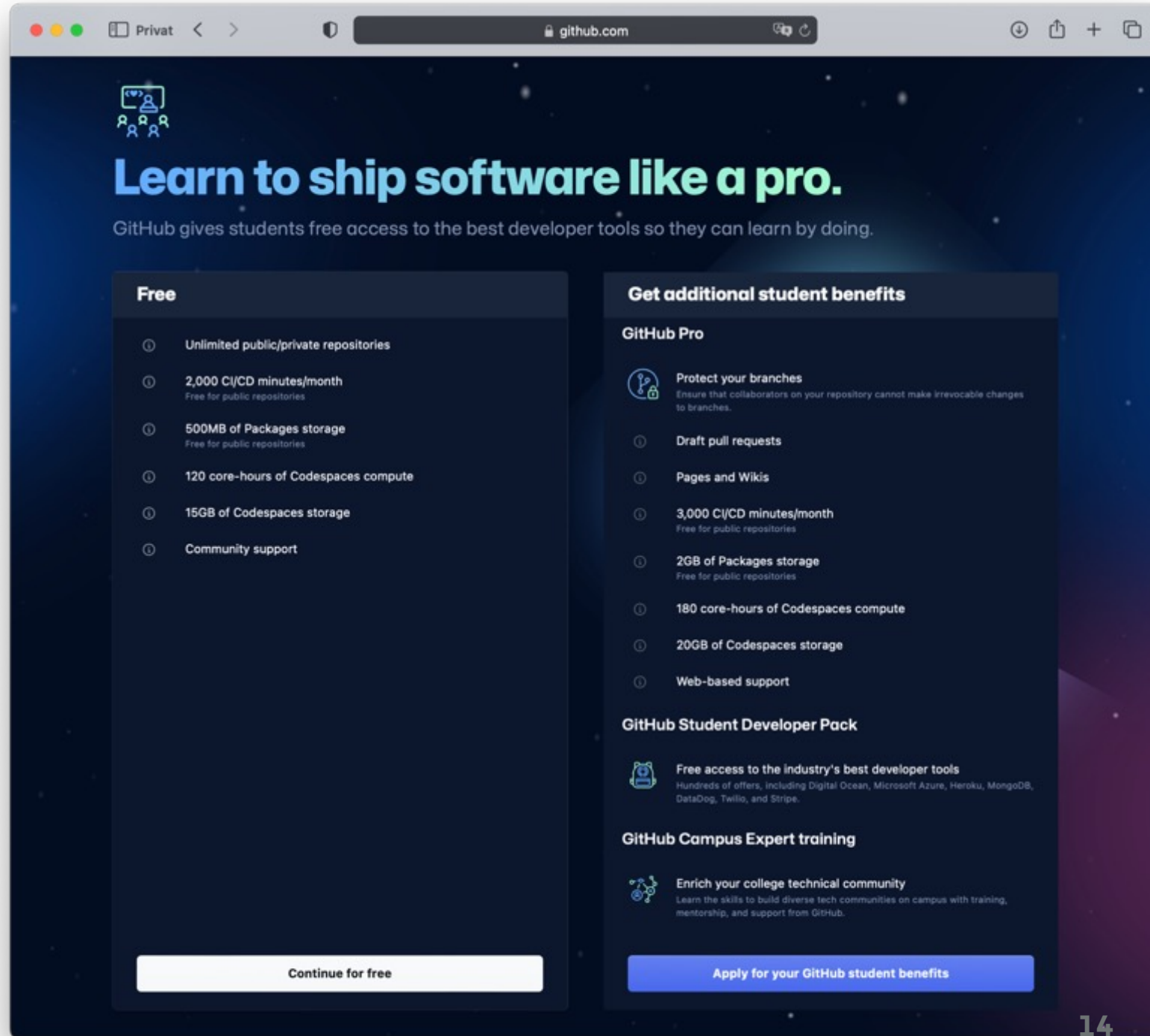
Student Teacher

Continue

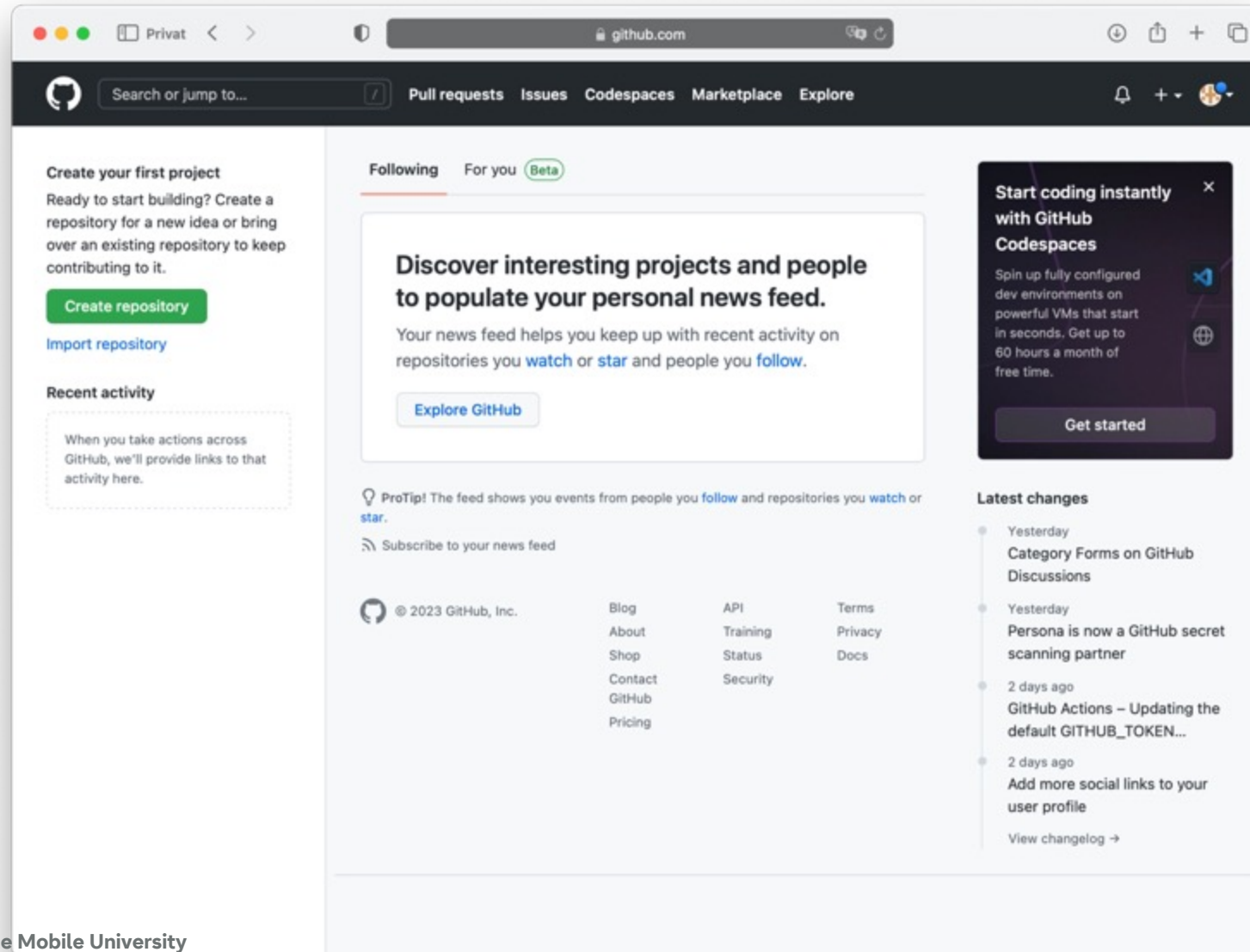
Here it is sufficient to click
"continue" to click



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

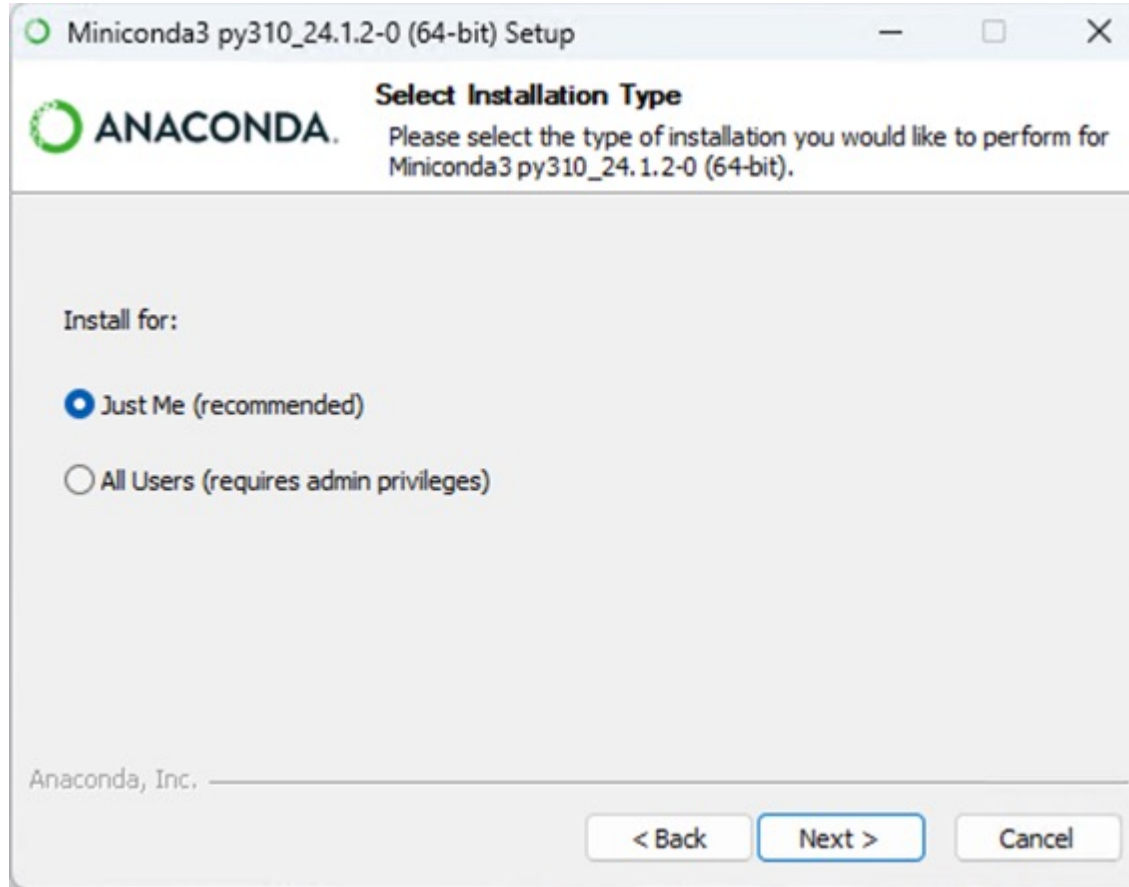
You can find the required installers at <https://docs.anaconda.com/free/miniconda/miniconda-other-installer-links/>.

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

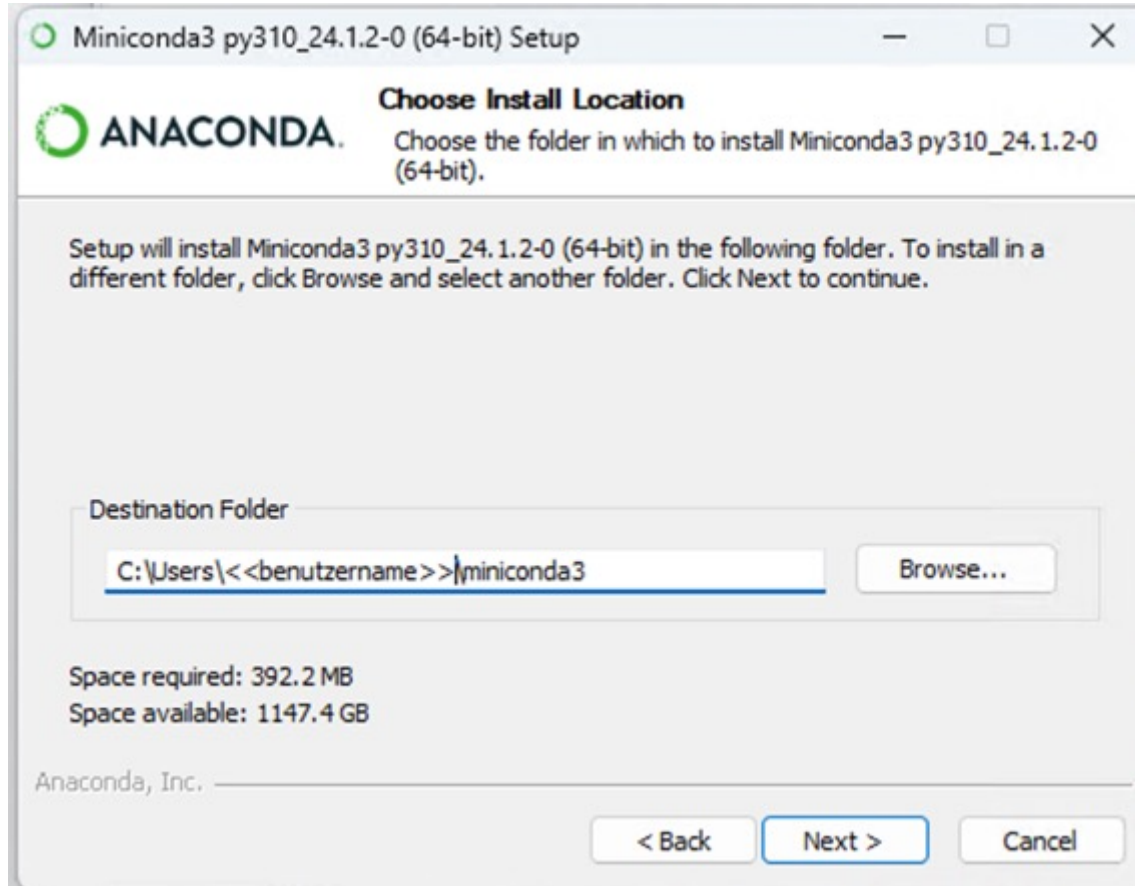
z. B. **Windows installers**

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

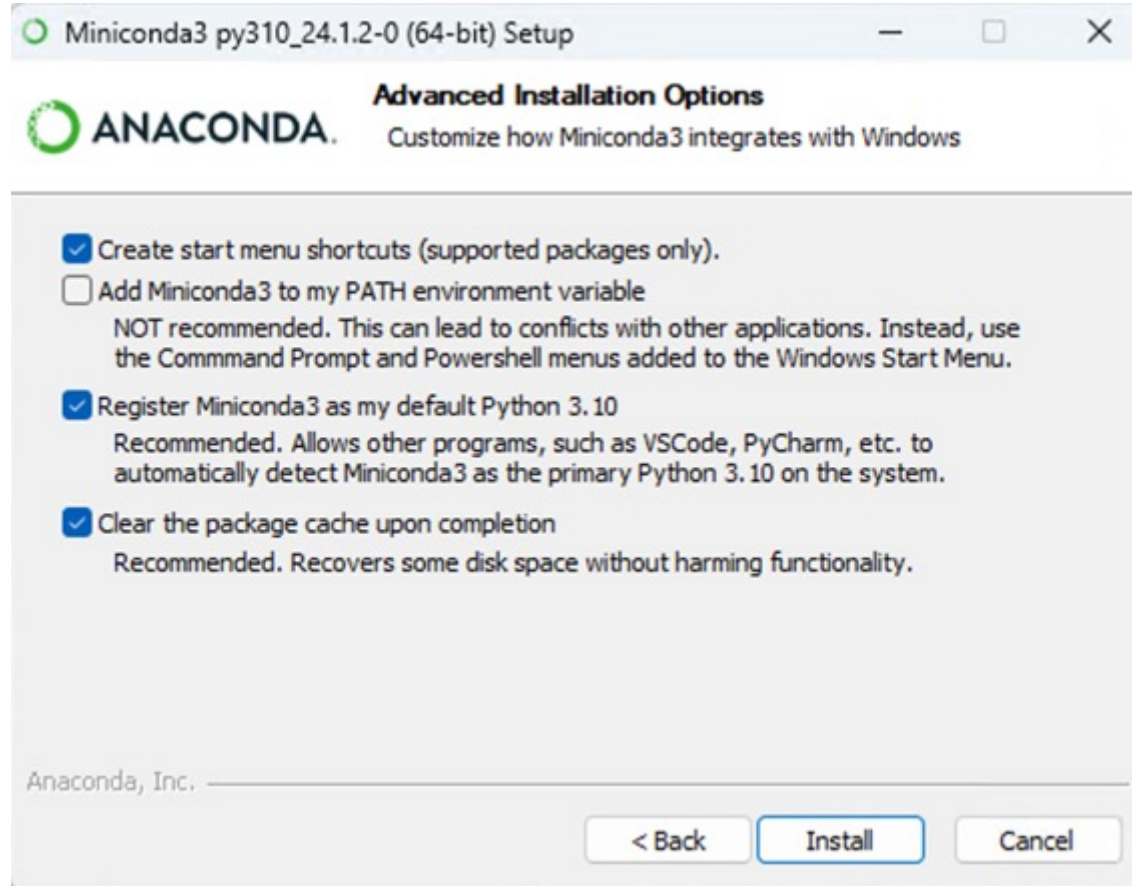
It is best to install miniconda "only for your account"



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.

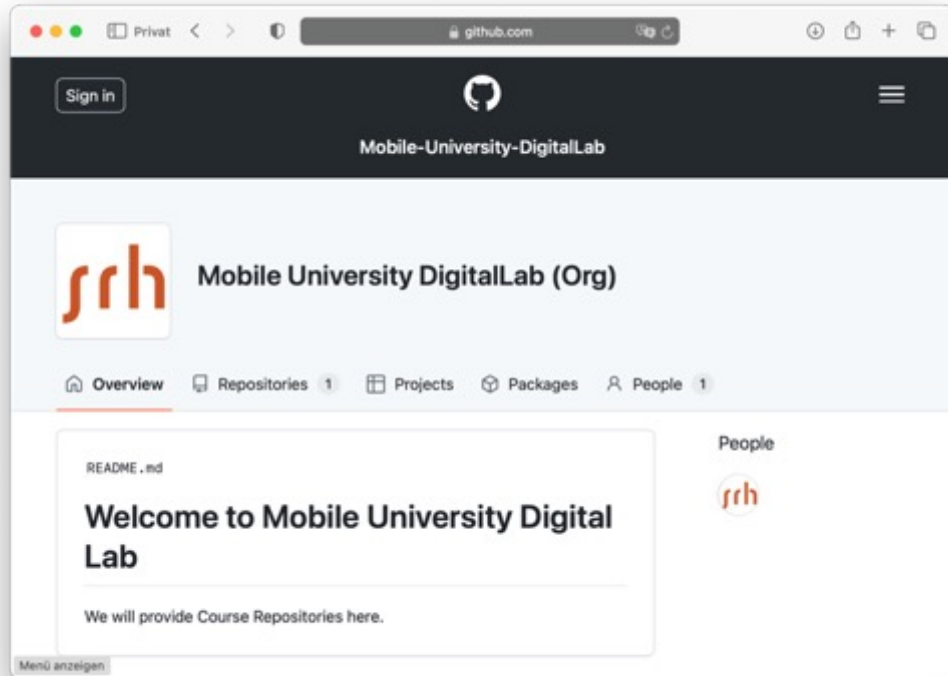
This activates miniconda. I will show you how to use it in section 5 of these instructions.

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.

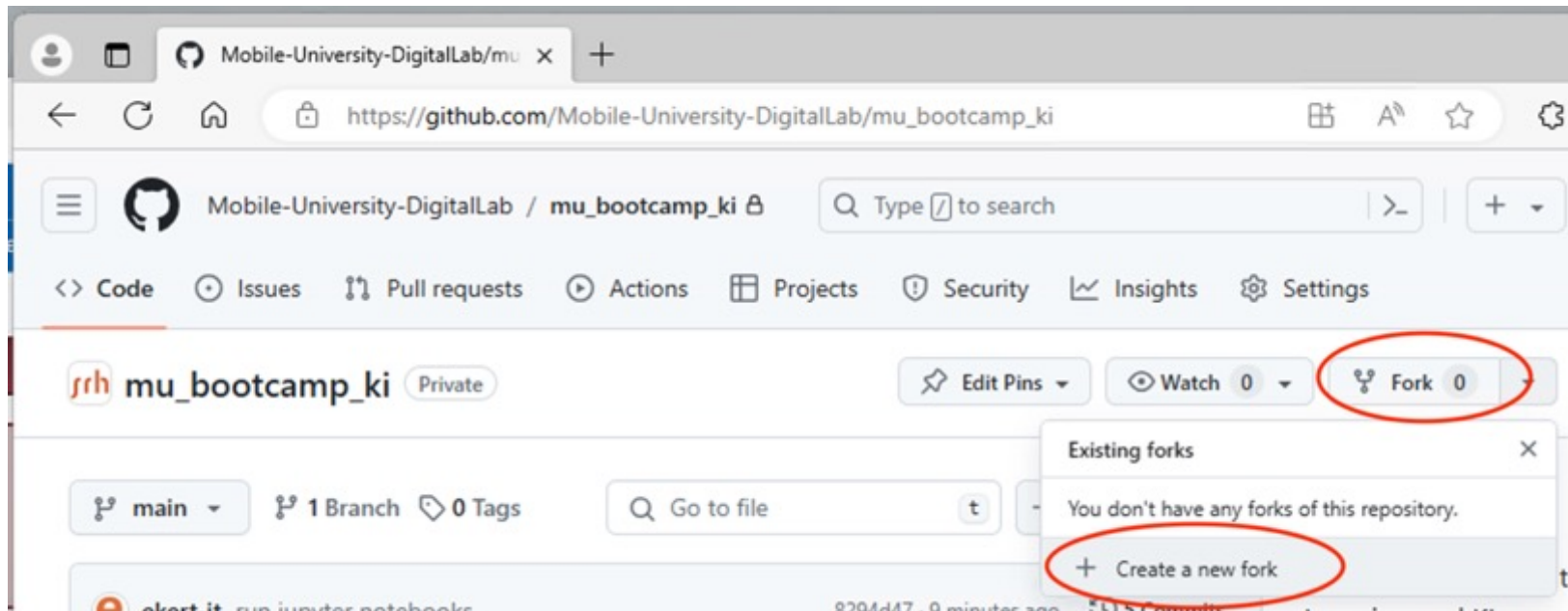
Ask your course instructor (or module leader) to **invite you to the github repository.**

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.



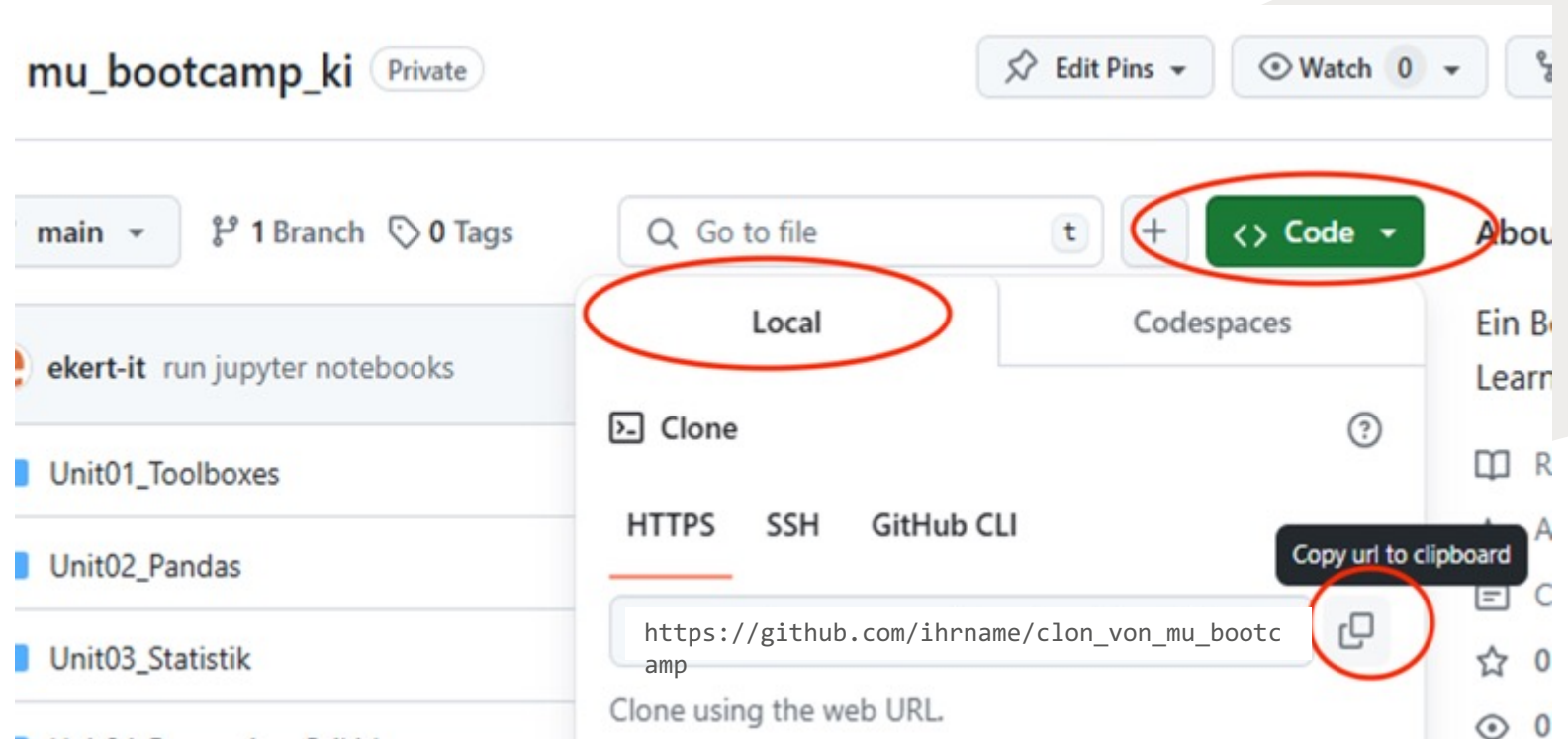
Create a local copy ("clone") of the repository

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.

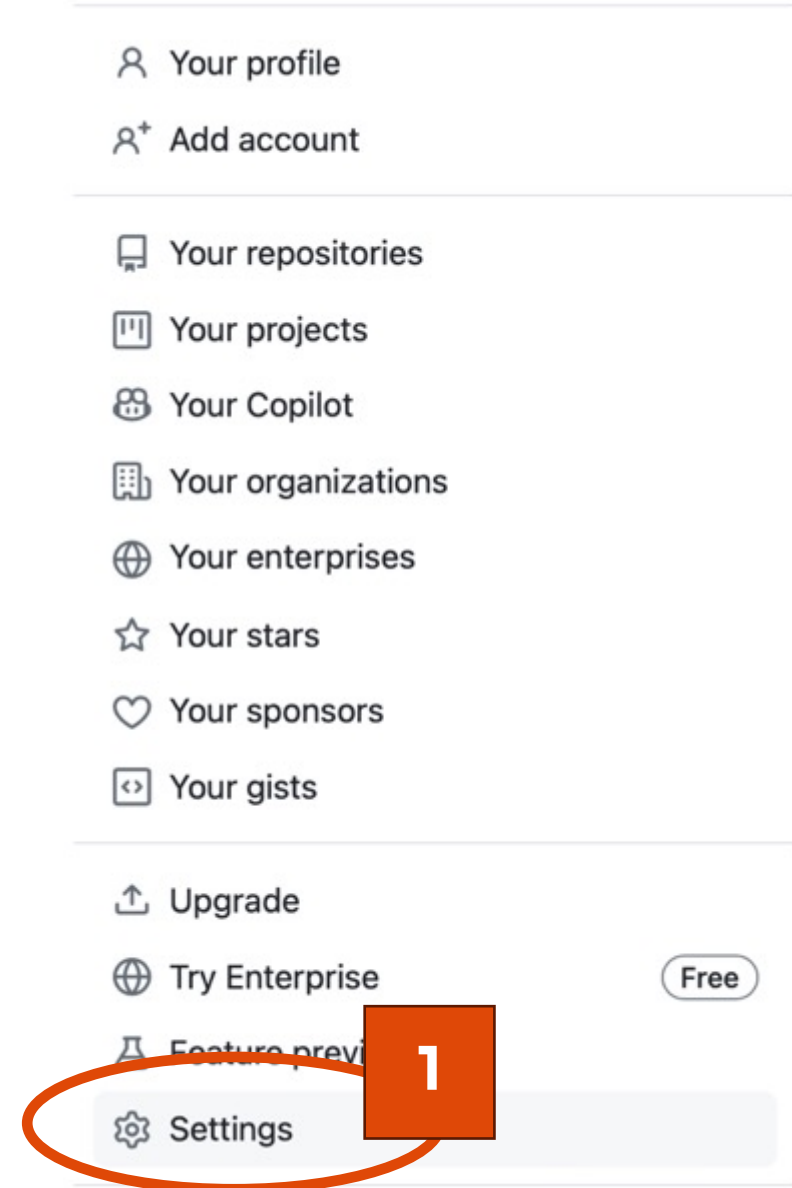
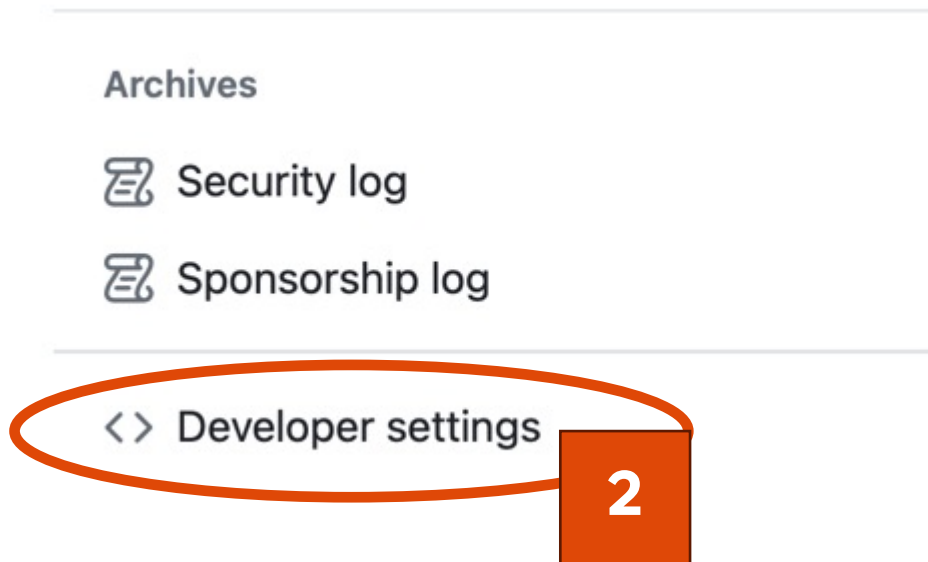
SAVE this URL.



Allow access

You need a token to enable github access.

You create this via the settings of your account.



Create new token

A screenshot of the GitHub Developer Settings page. The left sidebar shows the navigation menu with 'Personal access tokens' selected and 'Tokens (classic)' highlighted. The main content area is titled 'Personal access tokens (classic)' and shows a list of tokens. A dropdown menu is open for the 'git-sync' token, showing two options: 'Generate new token' (marked as Beta) and 'Generate new token (classic)'. The 'Generate new token (classic)' option is highlighted. The 'git-sync' token entry shows it is a repo-scoped token for 'git-sync' and expires on 'Thu, Aug 15 2024'.

Settings / Developer Settings

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens (Beta)

Tokens (classic)

Personal access tokens (classic)

Tokens you have generated that can be used to access the repository.

git-sync — repo

Expires on *Thu, Aug 15 2024*.

Generate new token (Beta)

Fine-grained, repo-scoped

Generate new token (classic)

For general use

Delete

Revoke all

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Delegate rights to token

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

Beta

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

mu_access_token

What's this token for:

Expiration *

90 days The token will expire on Mon, Aug 26 2024

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

☒ repo

Full control of private repositories

☒ repo:status

Access commit status

☒ repo_deployment

Access deployment status

☒ public_repo

Access public repositories

☒ repo:invite

Access repository invitations

☒ security_events

Read and write security events

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.

Personal access tokens (classic) Generate new token ▼ Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_wG5I [REDACTED] Id0y 	Delete
git-sync — repo	Last used within the last week
Expires on Thu, Aug 15 2024.	Delete

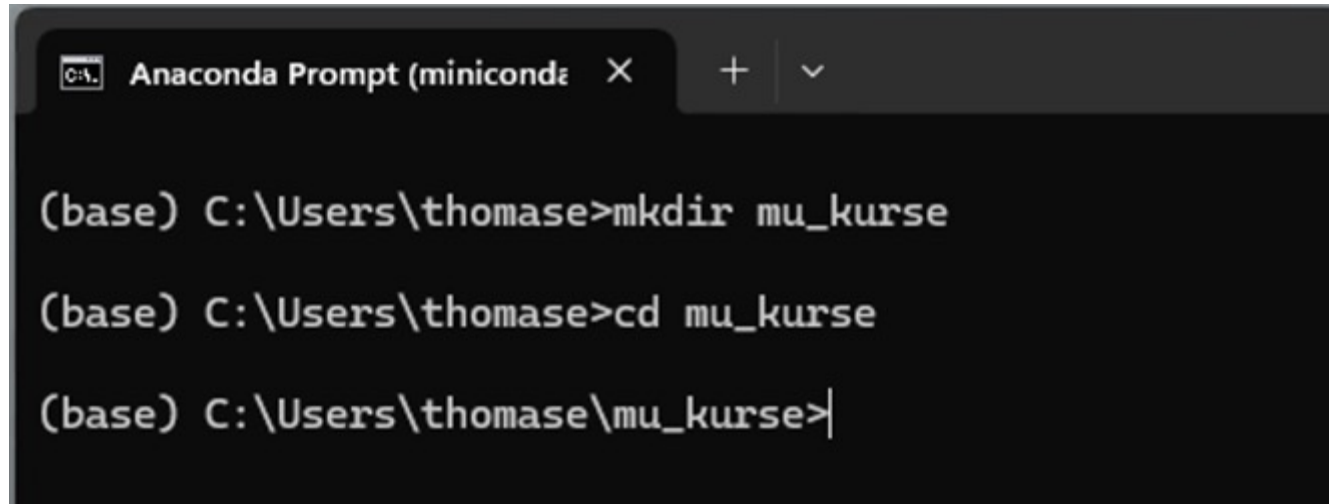
Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

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`



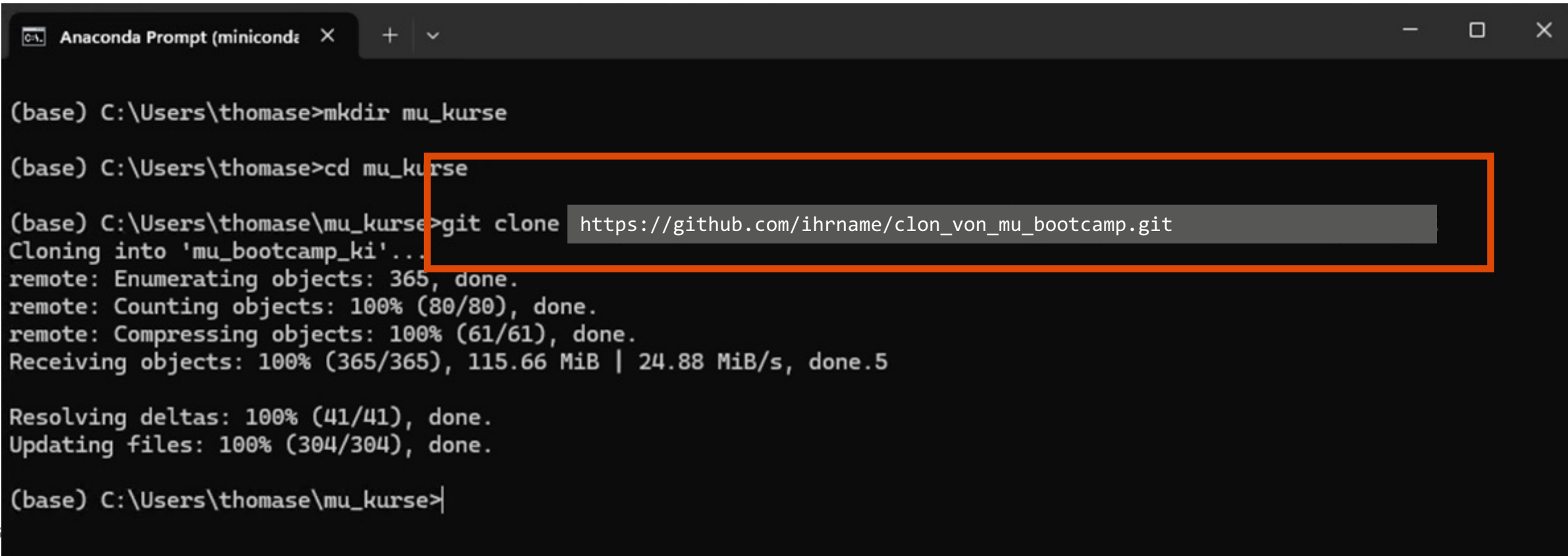
```
Anaconda Prompt (miniconda) X + v
(base) C:\Users\thomase>mkdir mu_kurse
(base) C:\Users\thomase>cd mu_kurse
(base) C:\Users\thomase\mu_kurse>
```

Clone repository

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

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

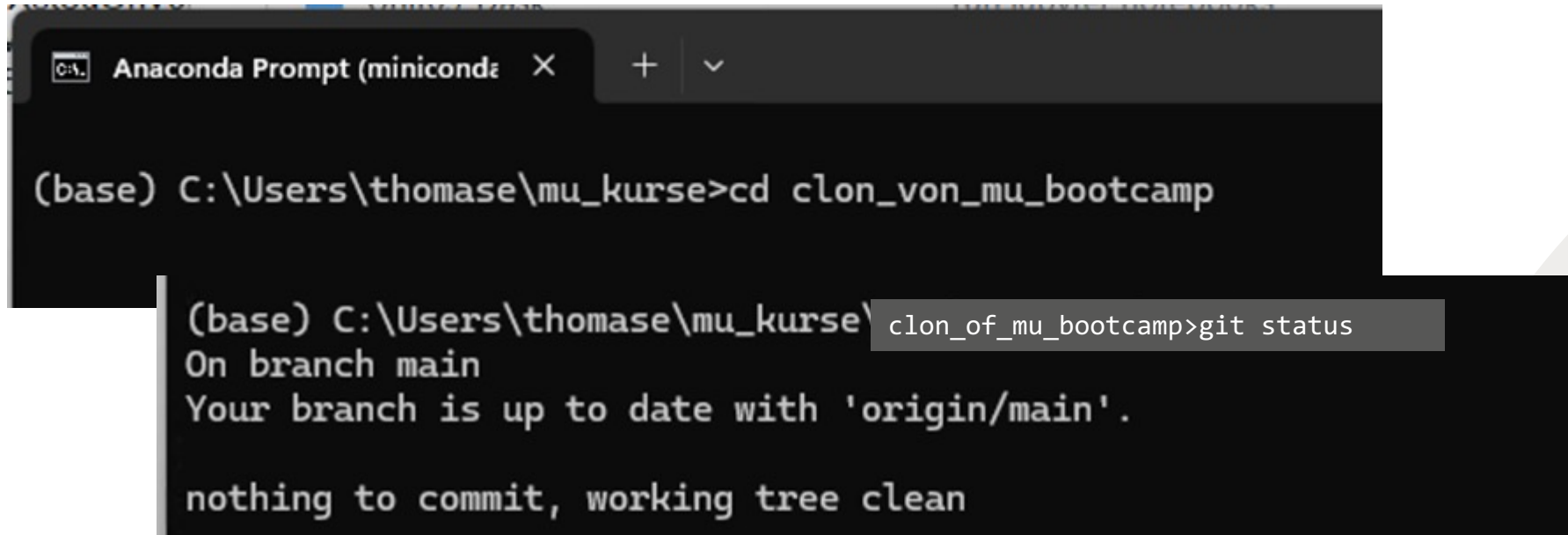
You will now be asked for your **user name**. Enter **the token created above as the password**.



```
Anaconda Prompt (miniconde) x + v
(base) C:\Users\thomase>mkdir mu_kurse
(base) C:\Users\thomase>cd mu_kurse
(base) C:\Users\thomase\mu_kurse>git clone https://github.com/ihrname/clon_von_mu_bootcamp.git
Cloning into 'mu_bootcamp_ki'...
remote: Enumerating objects: 365, done.
remote: Counting objects: 100% (80/80), done.
remote: Compressing objects: 100% (61/61), done.
Receiving objects: 100% (365/365), 115.66 MiB | 24.88 MiB/s, done.5
Resolving deltas: 100% (41/41), done.
Updating files: 100% (304/304), done.
(base) C:\Users\thomase\mu_kurse>
```

Check git status

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



```
Anaconda Prompt (miniconda) X + v
(base) C:\Users\thomase\mu_kurse>cd clon_von_mu_bootcamp

(base) C:\Users\thomase\mu_kurse\clon_of_mu_bootcamp>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

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).

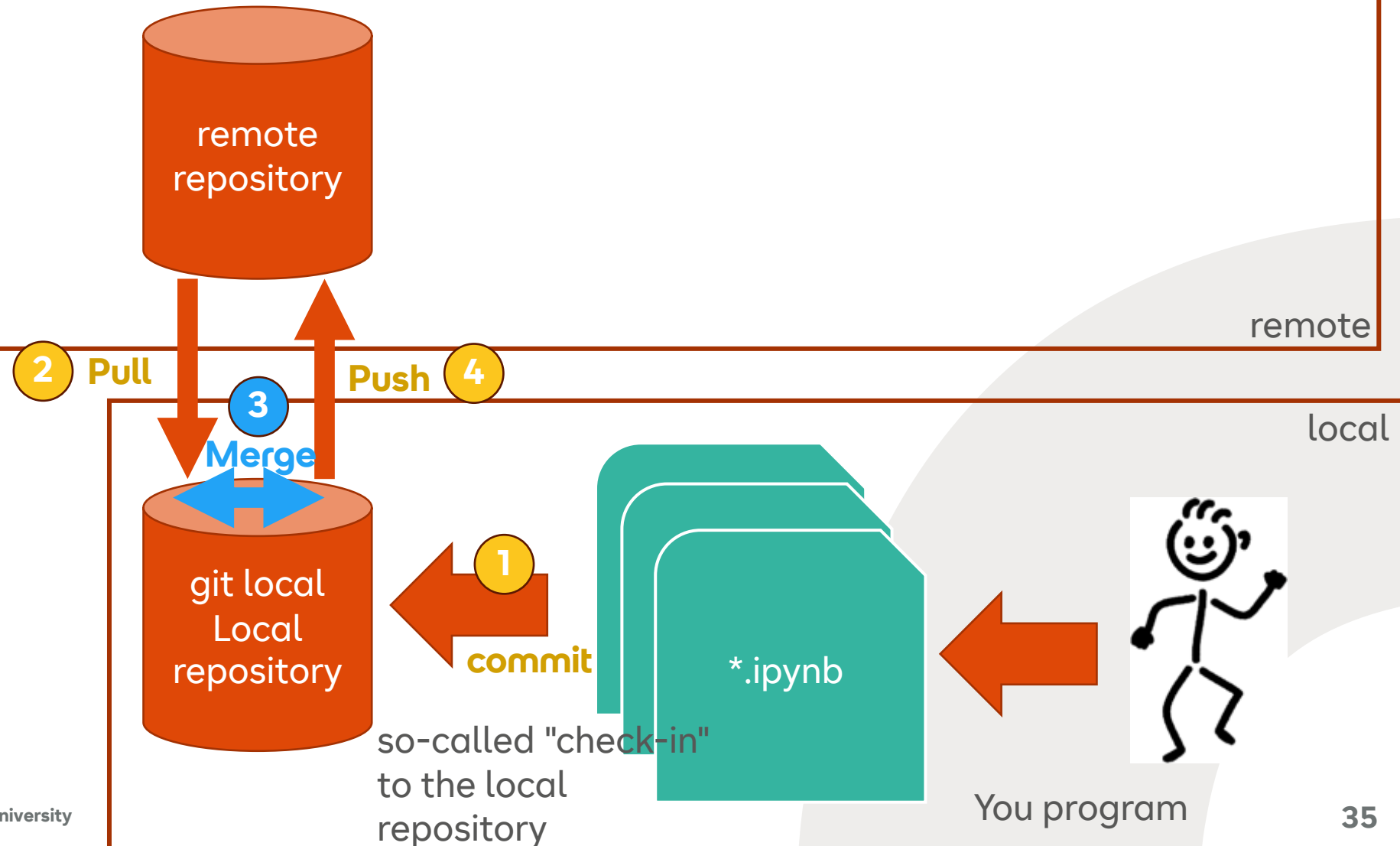
`git push`

→ 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





Prepare environment

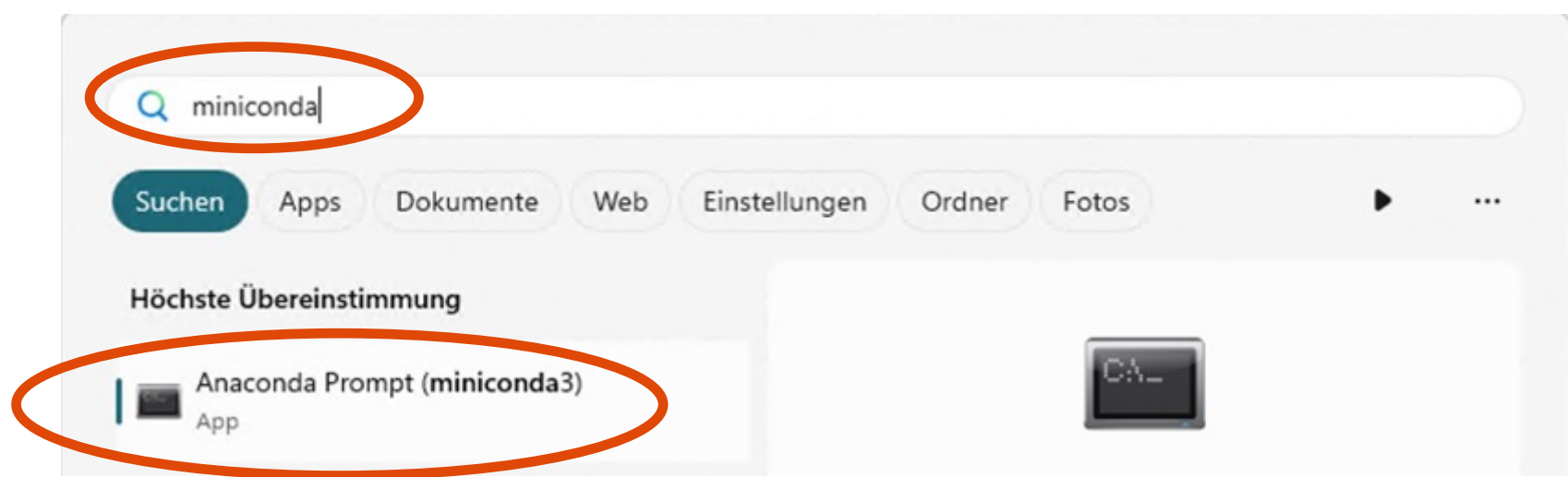
05

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

*.yaml



You will find several *.yaml 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>
```

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>.yaml

The installation of the environment may take several minutes.

You will then see the screen shown on the right.

```
Anaconda Prompt (miniconda) X + v
(base) c:\Users\your_name\mu_courses\mu_bootcamp_ki> conda env create -f
mu_bootcamp_base.yaml
Retrieving mu_bootcamp_base.yaml... done
Channels:
- conda-forge
- HCC
- pycharm
- anaconda
- pytorch
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

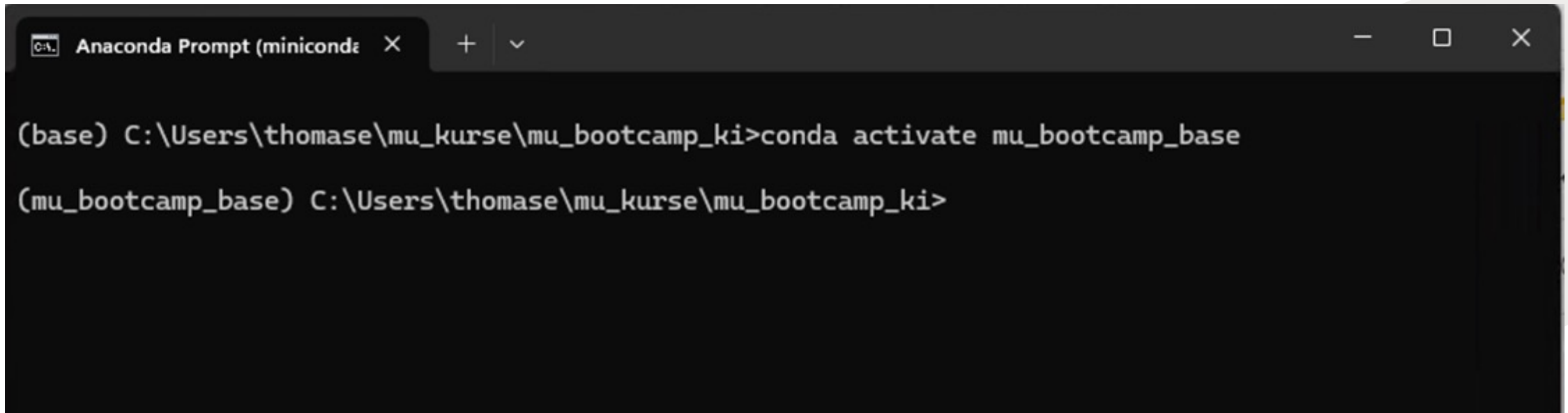
Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate mu_bootcamp_base
#
# To deactivate an active environment, use
#
#   $ conda deactivate
#
(base) c:\Users\your_name\mu_courses\mu_bootcamp_ki>
```

conda activate mu_bootcamp_base

You can now activate the environment as a test.

conda activate mu_bootcamp_base



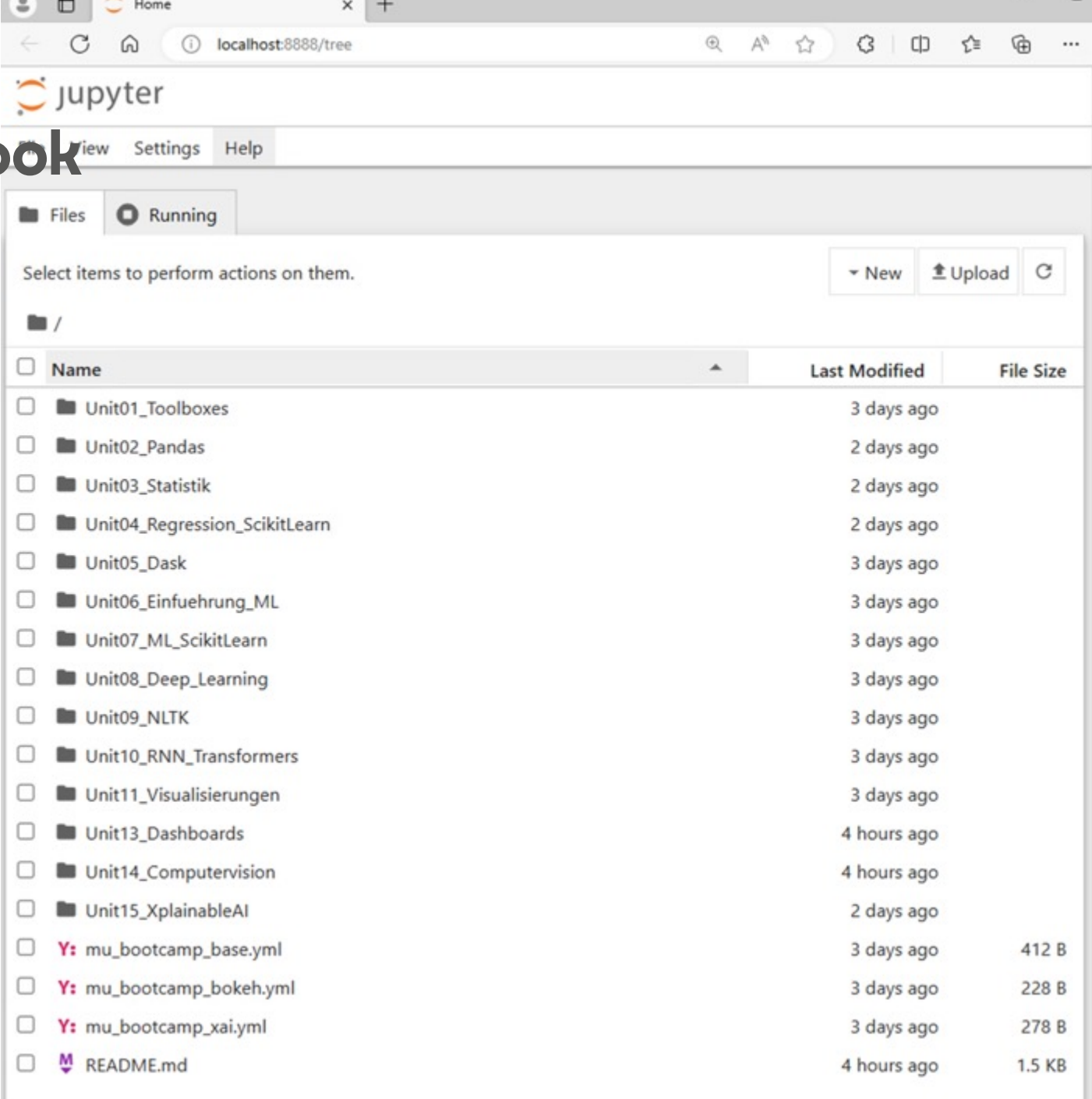
```
Anaconda Prompt (miniconde) C:\Users\thomase\mu_kurse\mu_bootcamp_ki>conda activate mu_bootcamp_base
(mu_bootcamp_base) C:\Users\thomase\mu_kurse\mu_bootcamp_ki>
```

Start course - jupyter notebook

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

jupyter notebook

```
npm prefix
(base) C:\Users\thomase\mu_kurse\mu_bootcamp_ki>conda activate mu_bootcamp_base
(mu_bootcamp_base) C:\Users\thomase\mu_kurse\mu_bootcamp_ki>jupyter notebook
[I 2024-05-28 17:11:44.337 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2024-05-28 17:11:44.347 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2024-05-28 17:11:44.361 ServerApp] jupyterlab | extension was successfully linked.
[I 2024-05-28 17:11:44.368 ServerApp] notebook | extension was successfully linked.
[I 2024-05-28 17:11:44.726 ServerApp] notebook_shim | extension was successfully linked.
[I 2024-05-28 17:11:44.757 ServerApp] notebook_shim | extension was successfully loaded.
[I 2024-05-28 17:11:44.757 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-05-28 17:11:44.757 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2024-05-28 17:11:44.767 LabApp] JupyterLab extension loaded from C:\Users\thomase\miniconda3\envs\mu_bootcamp_base\lib\site-packages\jupyterlab
[I 2024-05-28 17:11:44.767 LabApp] JupyterLab application directory is C:\Users\thomase\miniconda3\envs\mu_bootcamp_base\share\jupyter\lab
[I 2024-05-28 17:11:44.767 LabApp] Extension Manager is 'pypi'.
[I 2024-05-28 17:11:44.797 ServerApp] jupyterlab | extension was successfully loaded.
[I 2024-05-28 17:11:44.808 ServerApp] notebook | extension was successfully loaded.
[I 2024-05-28 17:11:44.808 ServerApp] Serving notebooks from local directory: C:\Users\thomase\mu_kurse\mu_bootcamp_ki
[I 2024-05-28 17:11:44.808 ServerApp] Jupyter Server 2.14.0 is running at:
```



Start Jupyter Notebook

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

jupyter numpy Last Checkpoint: 3 days ago

File Edit View Run Kernel Settings Help Trusted

Markdown ▾ Open in... Python 3 (ipykernel)

Die Numpy-Bibliothek

NumPy (Numerical Python) (*Harris et al., 2020*) ist eine Python-Bibliothek für numerische Berechnungen mit **Arrays**, **Matrizen** und anderen **mehrdimensionalen Datenstrukturen**. Die Bibliothek bietet eine breite Palette von Funktionen für mathematische Operationen, lineare Algebra, Signalverarbeitung und vieles mehr.

Die Stärke von NumPy liegt in der effizienten Handhabung von großen Datenmengen und der Möglichkeit, komplexe mathematische Berechnungen auf diesen Daten durchzuführen. Durch die Verwendung von NumPy-Arrays anstelle von herkömmlichen Python-Listen kann die Ausführungszeit von Code erheblich reduziert werden.

Einer der Hauptzwecke von NumPy in der Datenanalyse besteht in seiner Funktion als Container von Daten die zwischen verschiedenen Bibliotheken ausgetauscht werden. NumPy hat durch eine effiziente C-API eine starke Anbindung an sogenannte niedrige Programmiersprachen wie C und C++. So können zum Beispiel Bibliotheken die in C++ oder Fortran geschrieben sind direkt auf NumPy-Arrays zugreifen ohne ihre Speicherform umwandeln zu müssen.

Mehr über die NumPy Bibliothek erfahren Sie [hier](#).

NumPy Arrays

Ein **Array** ist in NumPy die zentrale Datenstruktur, um n -dimensionale Arrays oder Matrizen zu repräsentieren. Arrays ähneln Listen in Python, jedoch können sie mehrdimensional sein und **effizienter mit großen Datenmengen** umgehen. Arrays können verwendet werden, um viele Arten von Daten zu speichern, einschließlich **numerischer Daten**, **Strings** und **boolescher Werte**. Es ist zwar möglich, unterschiedliche heterogene Daten in einem Array zu speichern (z. B. in einem sogenannten

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.

The screenshot shows a Jupyter Notebook interface in a web browser. The browser address bar shows `localhost:8888/notebooks/Unit01_Toolboxes/numpy/numpy.ipynb`. The Jupyter interface has a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar with icons for file operations and execution. A red circle highlights the "Run" button (a play icon) in the toolbar. Below the toolbar, the notebook content is displayed. The first cell contains the code `[1]: import numpy as np` and is circled in red. The second cell, titled "Erstellen von Arrays", contains two code snippets: `[2]: a1 = np.array([1, 2, 3])` and `[3]: a2 = np.array((3.0, 2.0, 1.0))`. The third cell contains the code `[4]: a1.dtype` and its output `[4]: dtype('int32')`, both of which are circled in red. The notebook interface also shows a "Trusted" status and a "Python 3 (ipykernel)" environment.

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!