Introduction: Part 2

Scientific Programming in Python

Installation instructions: Miniconda & Linux

- Download your version from https://docs.conda.io/en/latest/miniconda.html
- *bash Miniconda3-latest-Linux-x86_64.sh* (saying yes when it asks you to add it to the terminal)
  - (*which python* should now answer .../anaconda3/bin/python)
  - (alternatively: *conda update --all*)
- If you don’t have git already (try by running *which git*), install it using *sudo apt-get install git*

To have jupyterlab globally on your system:

- *conda install jupyter*
- *conda install jupyterlab*
- *conda install conda-forge::nodejs*
- *jupyter labextension install @lckr/jupyterlab_variableinspector*

To create the environment for the class:

- *git clone https://github.com/scientificprogrammingUOS/lectures.git*
- *conda env create -f lectures/environment.yml*
- *conda activate scientific_programming*
- *jupyter labextension install @lckr/jupyterlab_variableinspector*

Installation instructions: Miniconda & Windows

- Download your version from https://docs.conda.io/en/latest/miniconda.html
- Run the graphical installer.
  - **Make sure to add conda to your PATH**[1], such that you can use it from your standard terminal.
- Afterwards, open the command-prompt as Administrator (hit Win-Key, type “cmd”, right-click “Command Prompt”, select “as Admin”)
  - Test if your installation was correct by running *where python*
    → it should return a path containing .../Anaconda3/…
  - Update your Conda installation by running *conda update conda*
- Install git from https://git-scm.com/downloads
  - Make sure that git **can** be used from your command-line[2]
  - Make sure that you use one of the two options **committing unix-style**[3]
  - Leave everything else the way it is
- Afterwards you should have the commands *python*, *conda* and *git* as registered commands[4]

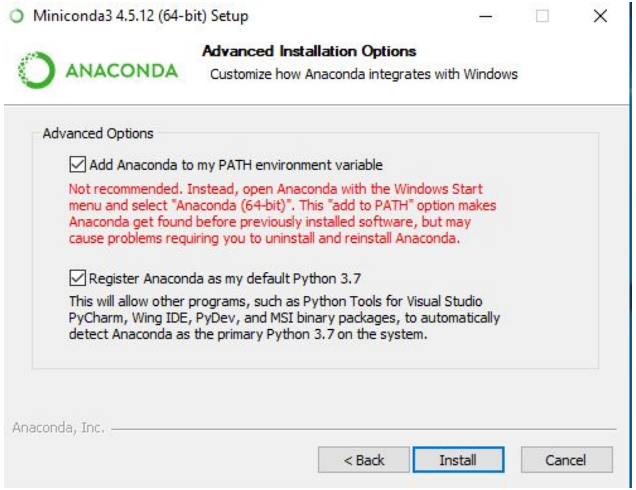
To have jupyterlab globally on your system:

- *conda install jupyter*
- *conda install jupyterlab*
- *conda install conda-forge::nodejs*
- *jupyter labextension install @lckr/jupyterlab_variableinspector*[5]

To create the environment for the class:

- *git clone https://github.com/scientificprogrammingUOS/lectures.git*
- *conda env create -f lectures/environment.yml*
- *conda activate scientific_programming*
- *jupyter labextension install @lckr/jupyterlab_variableinspector*[5]

1)



Miniconda3 4.5.12 (64-bit) Setup

Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

☑ Add Anaconda to my PATH environment variable

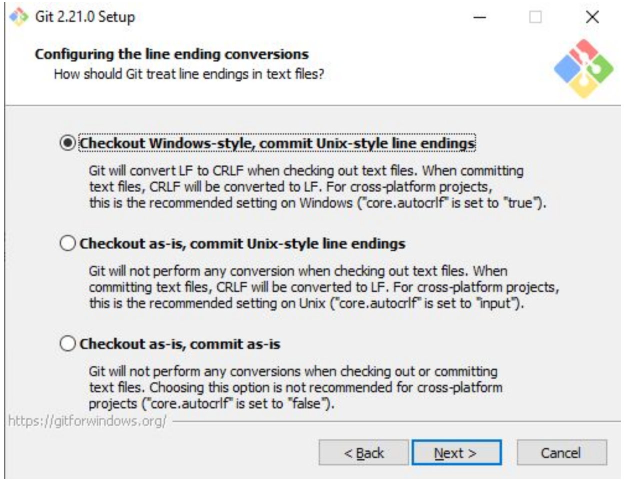
Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☑ Register Anaconda as my default Python 3.7

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.7 on the system.

Anaconda, Inc.

2)



Git 2.21.0 Setup

Configuring the line ending conversions

How should Git treat line endings in text files?

● **Checkout Windows-style, commit Unix-style line endings**

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

○ **Checkout as-is, commit Unix-style line endings**

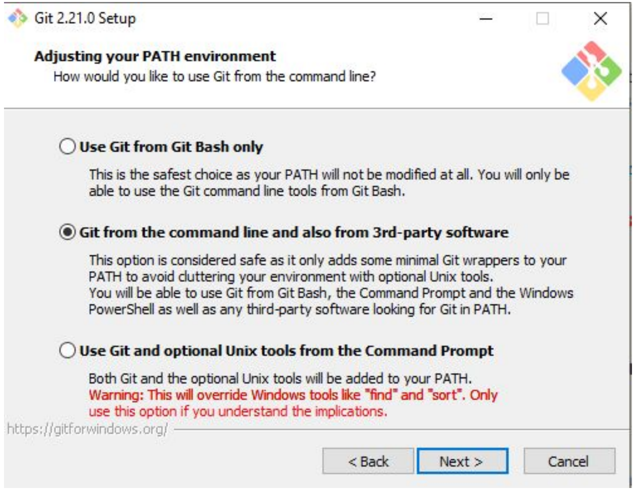
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

○ **Checkout as-is, commit as-is**

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

https://gitforwindows.org/

3)



Git 2.21.0 Setup

Adjusting your PATH environment

How would you like to use Git from the command line?

○ **Use Git from Git Bash only**

This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

● **Git from the command line and also from 3rd-party software**

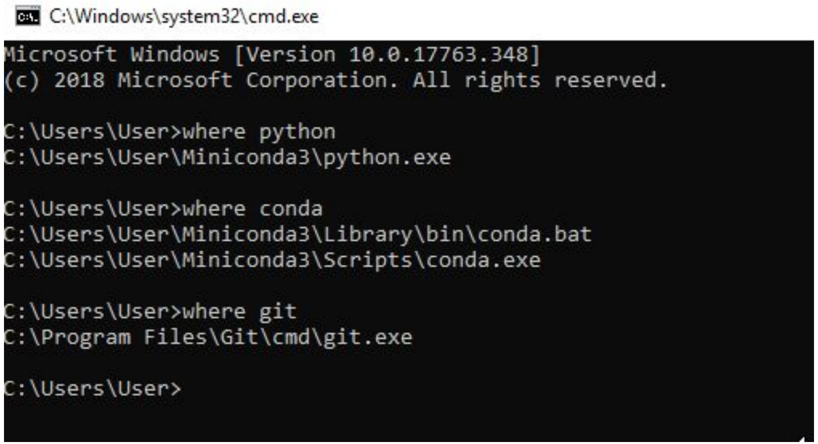
This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

○ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH. Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

https://gitforwindows.org/

4)



C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 10.0.17763.348]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\User>where python
C:\Users\User\Miniconda3\python.exe

C:\Users\User>where conda
C:\Users\User\Miniconda3\Library\bin\conda.bat
C:\Users\User\Miniconda3\Scripts\conda.exe

C:\Users\User>where git
C:\Program Files\Git\cmd\git.exe

C:\Users\User>
```

Windows: If the previous didn’t work

- Sorry for the huge mess that was today’s lecture - we know that for many of you, the previous installation process doesn’t work like that.
- If you are sure you followed the previous steps as described, and you still get an error containing something along the lines of *“qt linking failed”,* you can instead not install the complete environment all in once, but start with what’s needed for the first few lectures and install the rest on demand

```
conda create -n scientific_programming python=3.7
conda activate scientific_programming
pip install jupyter
pip install jupyterlab
pip install pytest
conda install nodejs -y
jupyter labextension install @lckr/jupyterlab_variableinspector[5]
```

- Afterwards, you can enable this environment just as if you installed it using the environment.yml file.

After Installation

- Once you activate your environment using *conda activate scientific_programming,* your shell should indicate that you're inside this environment
- Note that you have to **activate your this environment every time you work on the exercises!**
- To test if all packages are installed successfully, run *conda list* and check if all demanded packages are indeed listed.
- To start working inside jupyter lab, navigate to the correct directory using *cd,* and then start jupyterlab by executing *jupyter lab .* (yes, including the dot)[5]

[5] If the commands involving jupyter don't work on Windows, try using a hyphen instead of a space:
→ *jupyter-labextension install @lckr/jupyterlab_variableinspector*
→ *jupyter-lab .*

Intro: git

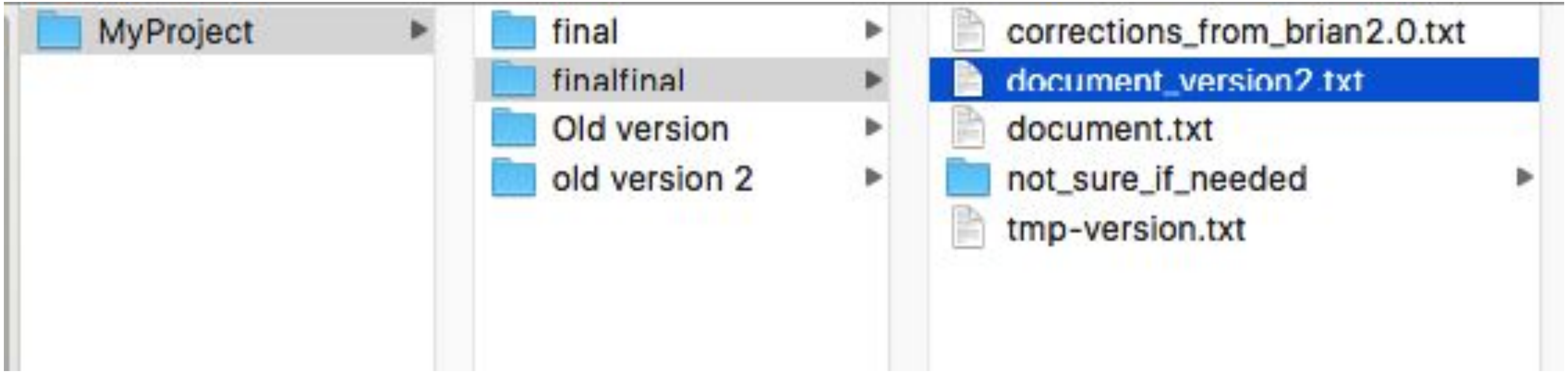
[These slides are basing on the ones created by Lukas Kalbertodt, all credits to him]

Getting started with git

[7]

- Git is a free and open source VCS (version control system)
- It allows you to track changes to files over time, compare new versions to old versions, have multiple versions in parallel, and work simultaneously on projects
- VCS are commonly used for programming projects, but can also be useful for any other project

Why version control?



MyProject

final
finalfinal
Old version
old version 2

corrections_from_brian2.0.txt
document_version2.txt
document.txt
not_sure_if_needed
tmp-version.txt

Version control

Tracks & logs changes in your files with…

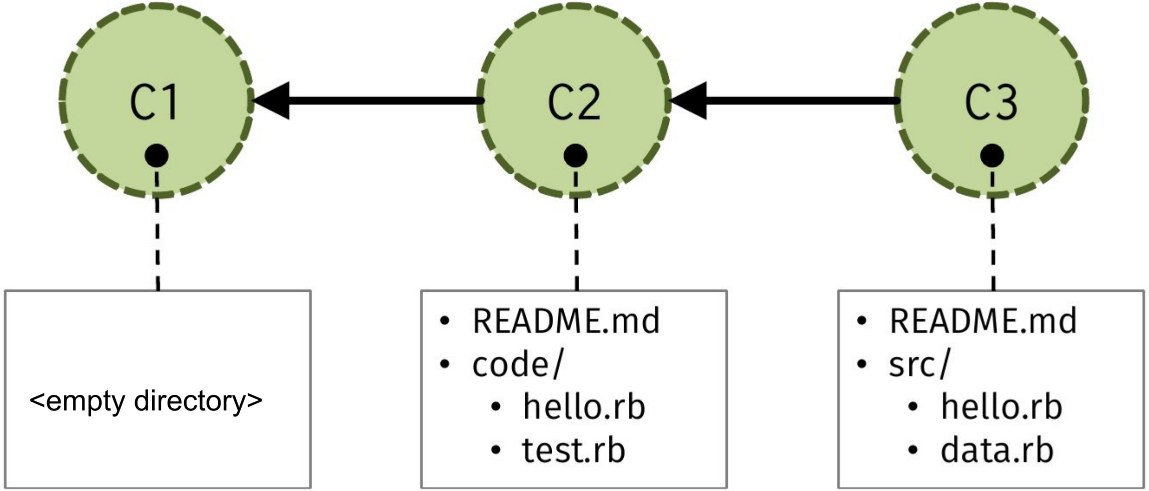
- Author
- Timestamp
- Description

Allows…

- Restoring old versions
- Having multiple parallel versions
- Analyzing your code
- Collaborating on code

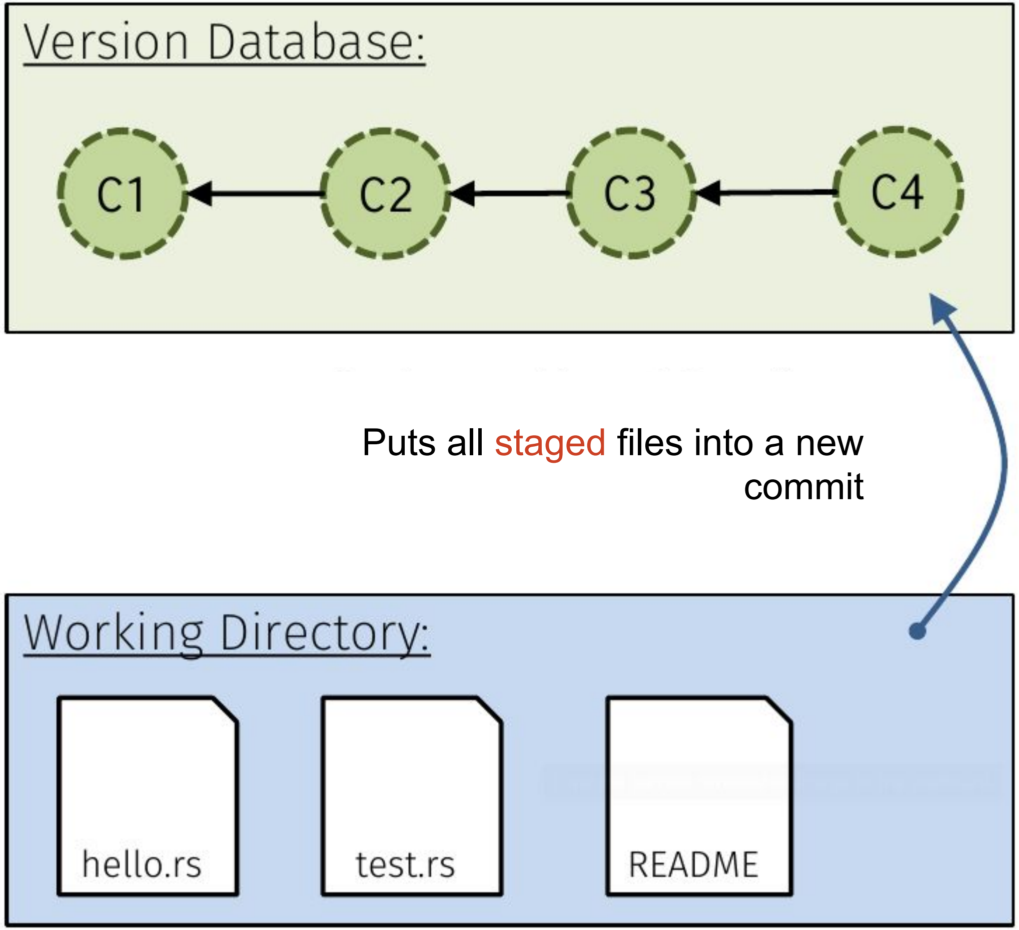
A commit

- Snapshot of the whole project at certain time
- A commit saves…
  - Its predecessor
  - Changes in the files (delta from predecessor)
  - Author, time, commit message
- Identified by Hash (eg. C1, C2, ..)



The repository

- Contains all commits
- Saved in a hidden folder called .git



Puts all staged files into a new commit

```
git commit
```

File status

Staged

File will be committed with the next commit

Modified

File is registered for git and was changed since the last commit

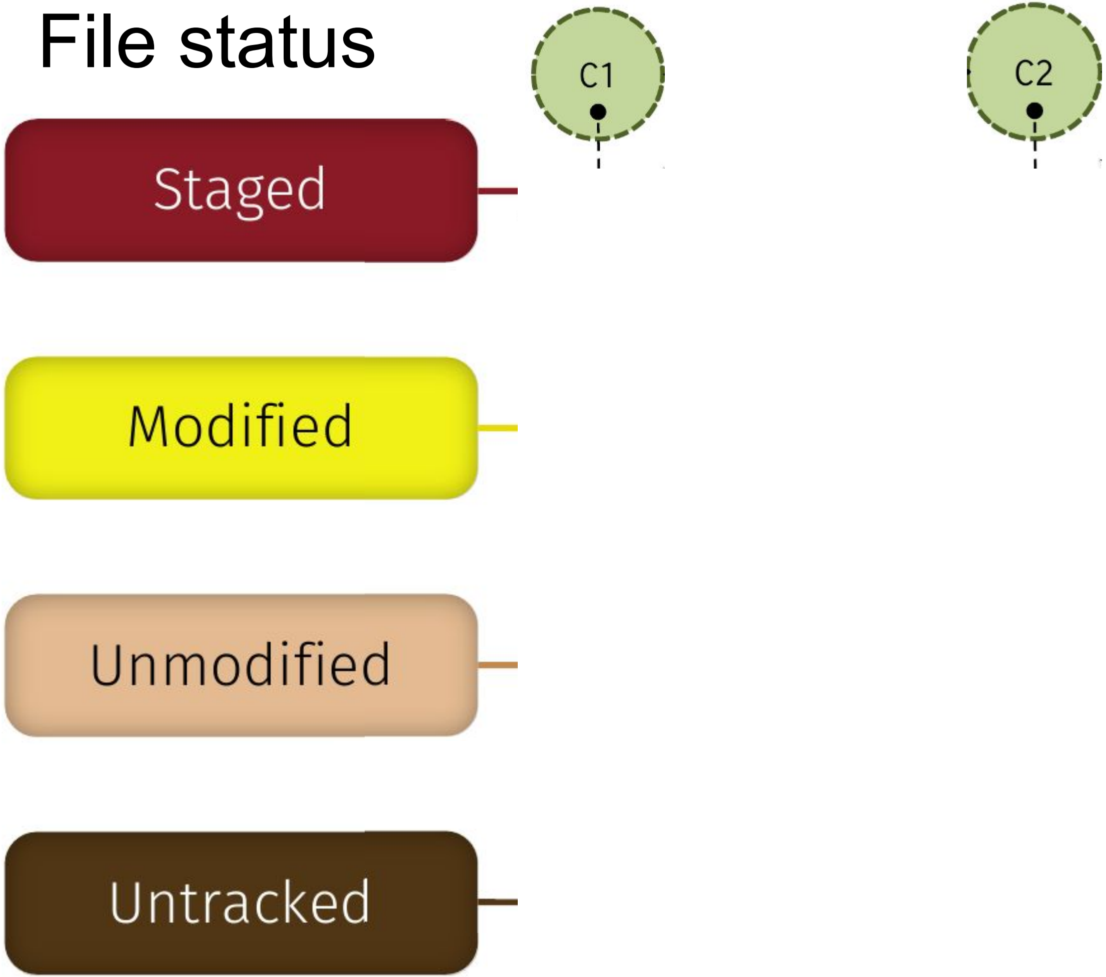
Unmodified

File is registered in git, but equal to the last commit

Untracked

Git knows the file exists, but won’t do anything with it

File status



Staged

Modified

Unmodified

Untracked

C1

C2

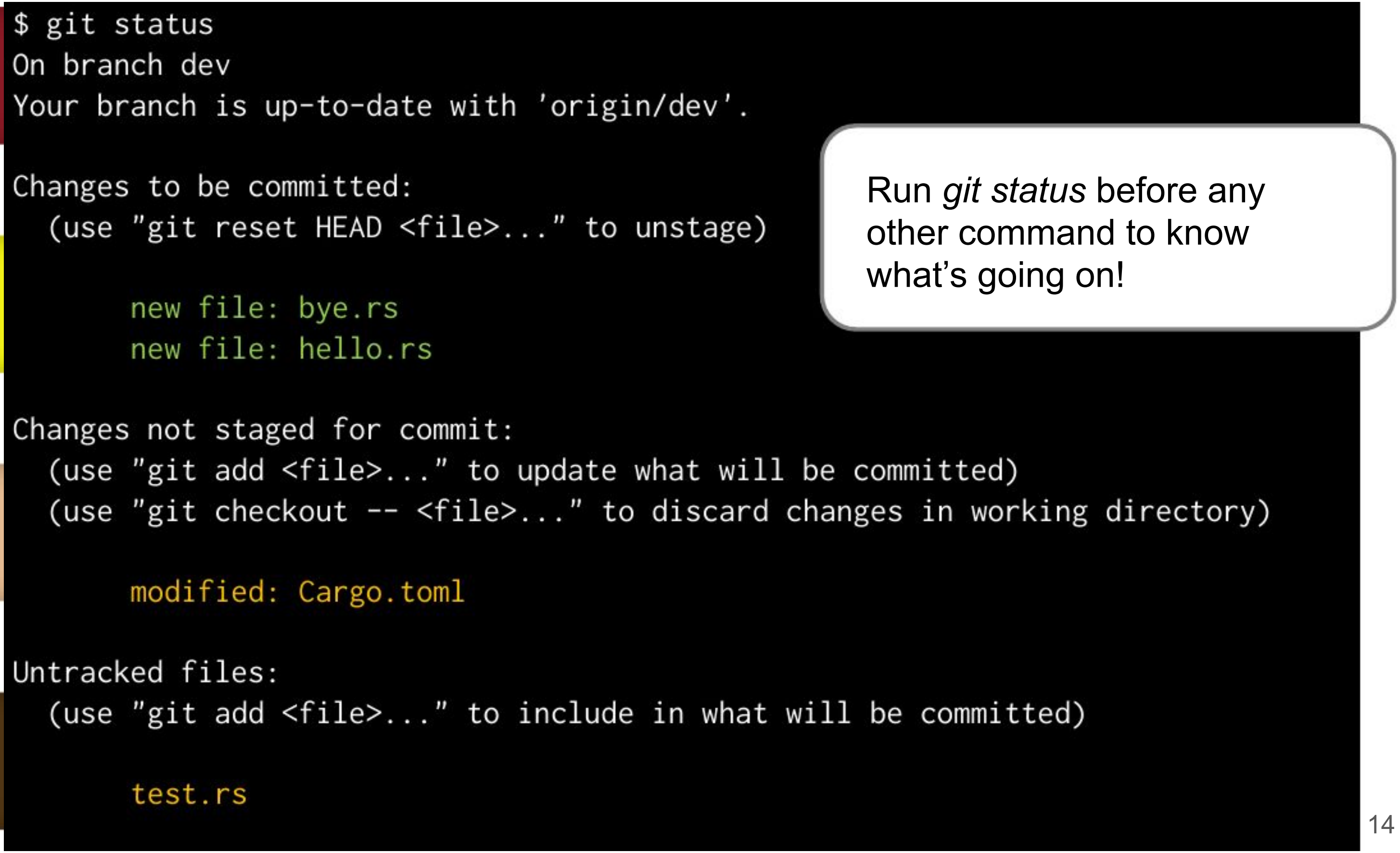
git status

Staged

Modified

Unmodified

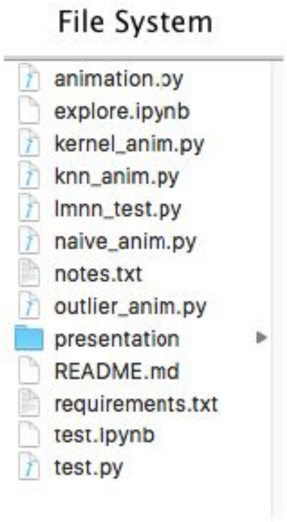
Untracked



```
$ git status
On branch dev
Your branch is up-to-date with 'origin/dev'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   bye.rs
        new file:   hello.rs

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Cargo.toml

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.rs
```

Run *git status* before any other command to know what's going on!

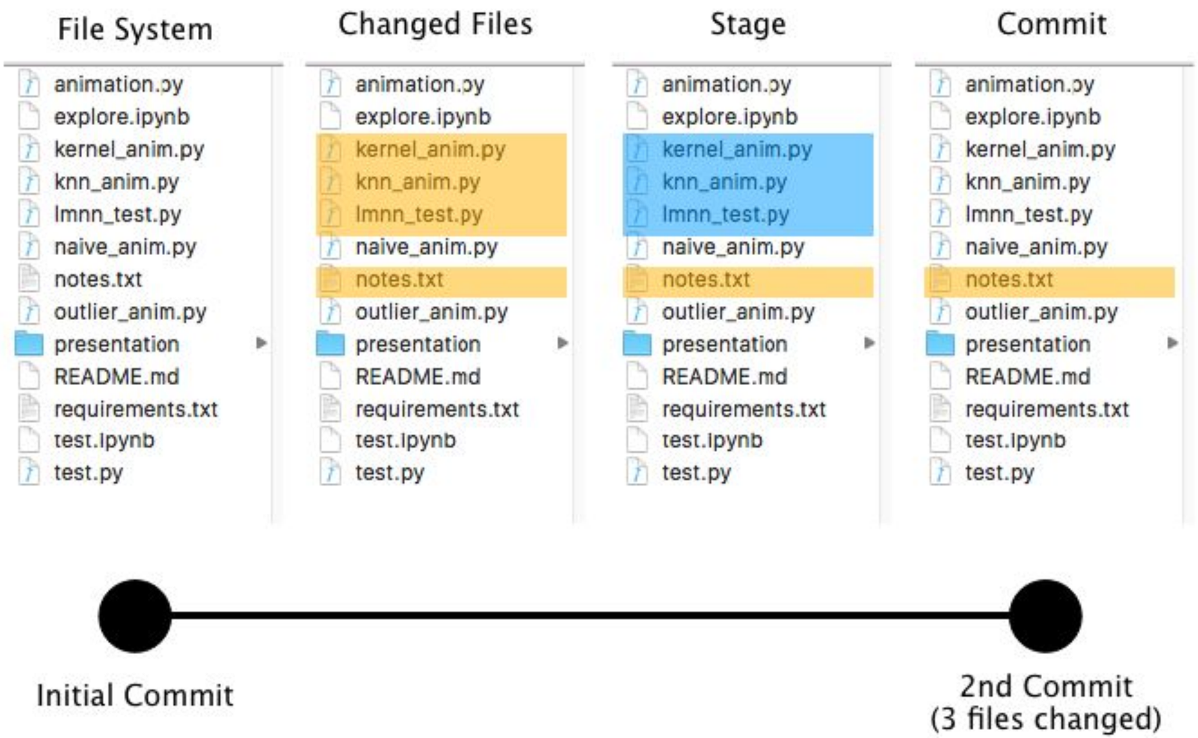
Git workflow



File System

animation.py
explore.ipynb
kernel_anim.py
knn_anim.py
lmnn_test.py
naive_anim.py
notes.txt
outlier_anim.py
presentation
README.md
requirements.txt
test.ipynb
test.py

Git workflow



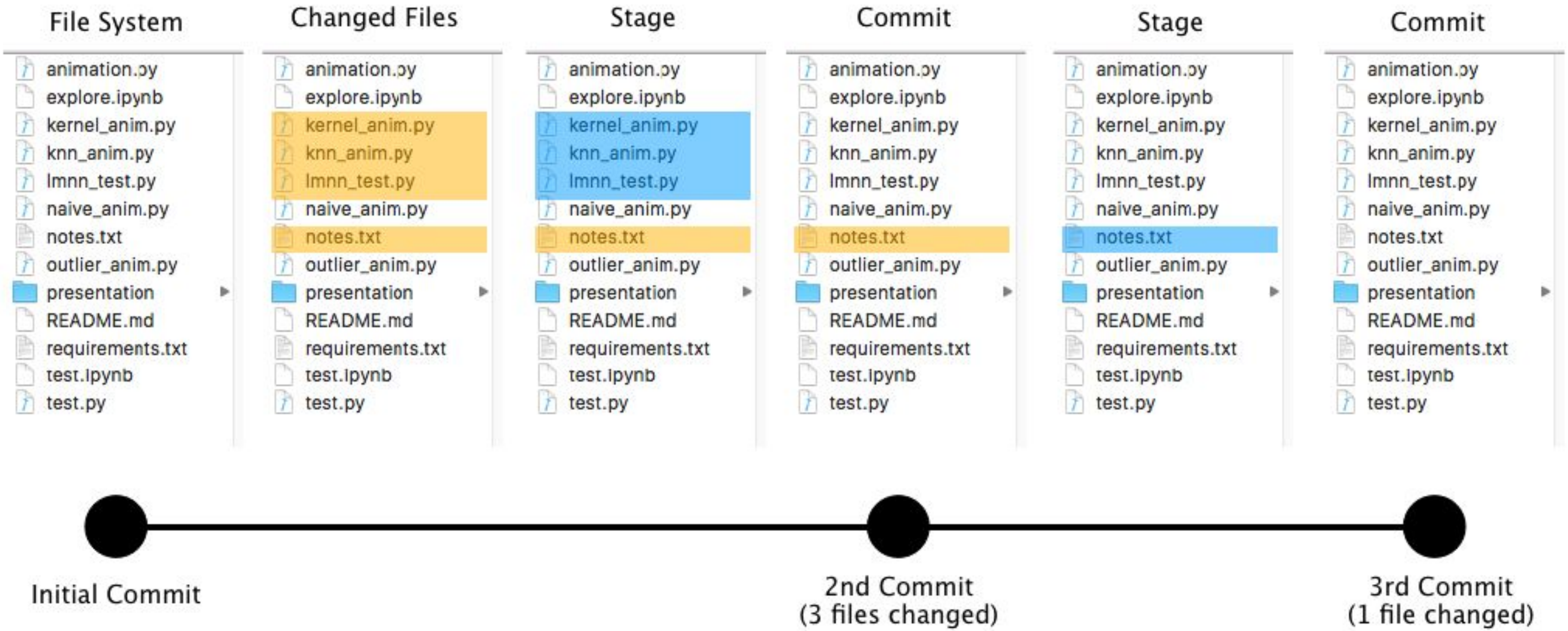
File System	Changed Files	Stage	Commit
animation.py	animation.py	animation.py	animation.py
explore.ipynb	explore.ipynb	explore.ipynb	explore.ipynb
kernel_anim.py	kernel_anim.py	kernel_anim.py	kernel_anim.py
knn_anim.py	knn_anim.py	knn_anim.py	knn_anim.py
lmnn_test.py	lmnn_test.py	lmnn_test.py	lmnn_test.py
naive_anim.py	naive_anim.py	naive_anim.py	naive_anim.py
notes.txt	notes.txt	notes.txt	notes.txt
outlier_anim.py	outlier_anim.py	outlier_anim.py	outlier_anim.py
presentation ▸	presentation ▸	presentation ▸	presentation ▸
README.md	README.md	README.md	README.md
requirements.txt	requirements.txt	requirements.txt	requirements.txt
test.ipynb	test.ipynb	test.ipynb	test.ipynb
test.py	test.py	test.py	test.py

Initial Commit

2nd Commit
(3 files changed)

https://try.github.io/

Git workflow



File System	Changed Files	Stage	Commit	Stage	Commit
animation.py	animation.py	animation.py	animation.py	animation.py	animation.py
explore.ipynb	explore.ipynb	explore.ipynb	explore.ipynb	explore.ipynb	explore.ipynb
kernel_anim.py	🟨 kernel_anim.py	🟦 kernel_anim.py	kernel_anim.py	kernel_anim.py	kernel_anim.py
knn_anim.py	🟨 knn_anim.py	🟦 knn_anim.py	knn_anim.py	knn_anim.py	knn_anim.py
lmnn_test.py	🟨 lmnn_test.py	🟦 lmnn_test.py	lmnn_test.py	lmnn_test.py	lmnn_test.py
naive_anim.py	naive_anim.py	naive_anim.py	naive_anim.py	naive_anim.py	naive_anim.py
notes.txt	🟨 notes.txt	🟨 notes.txt	🟨 notes.txt	🟦 notes.txt	notes.txt
outlier_anim.py	outlier_anim.py	outlier_anim.py	outlier_anim.py	outlier_anim.py	outlier_anim.py
presentation ▸	presentation ▸	presentation ▸	presentation ▸	presentation ▸	presentation ▸
README.md	README.md	README.md	README.md	README.md	README.md
requirements.txt	requirements.txt	requirements.txt	requirements.txt	requirements.txt	requirements.txt
test.ipynb	test.ipynb	test.ipynb	test.ipynb	test.ipynb	test.ipynb
test.py	test.py	test.py	test.py	test.py	test.py

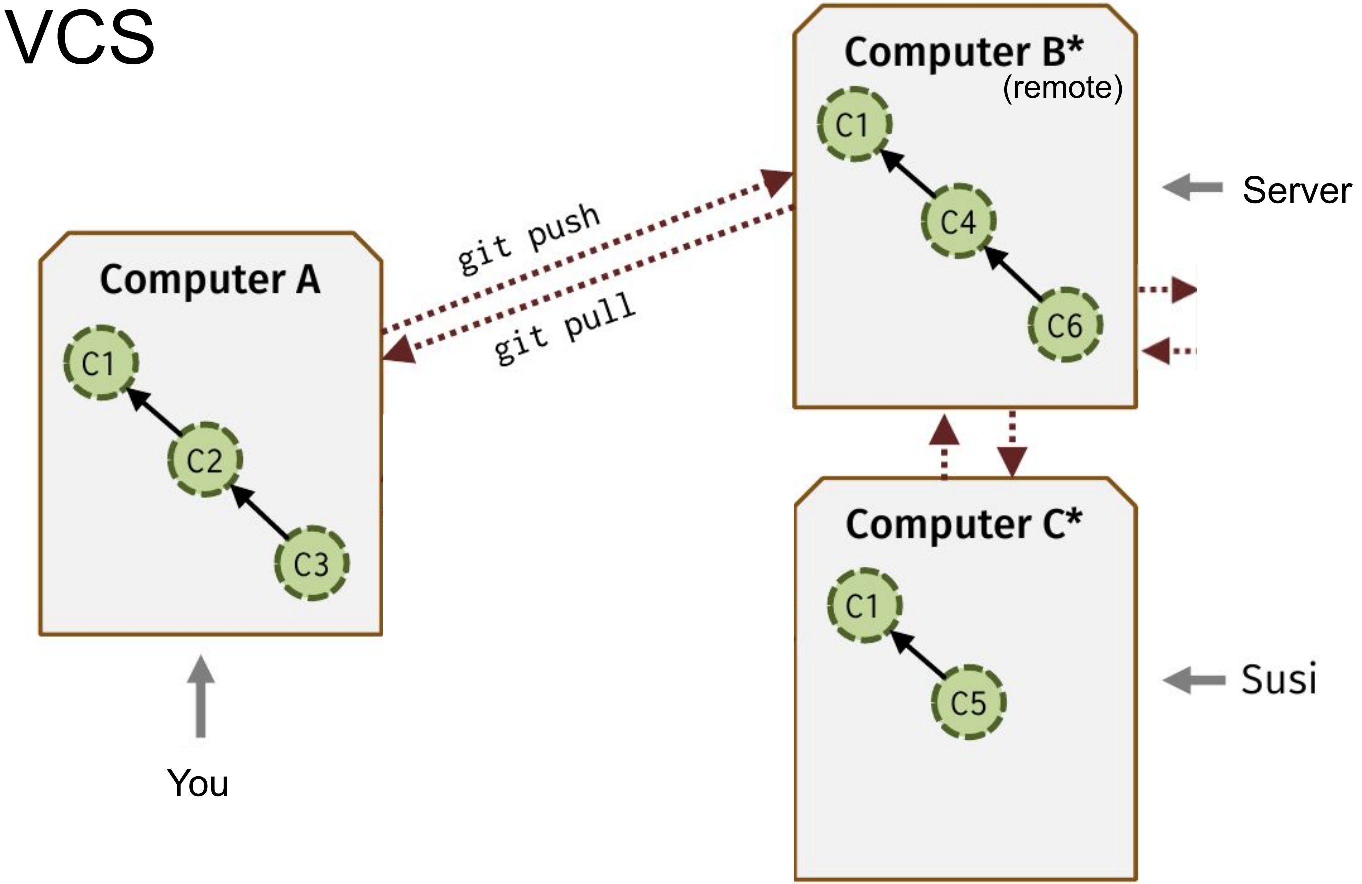
Getting started with GitHub

- Github is a website, providing a server to store your projects and also a web-interface to easily access them and to collaborate with others
- GitHub adds many additional features, like Pull-Requests with Code review, Issues, Wikis, …

- Github is only an *external storage (+UI)* for git projects
- Git works perfectly fine locally, without GitHub
- Github allows for easy access from multiple computers
- You need to manually synchronize your local directory with GitHub!

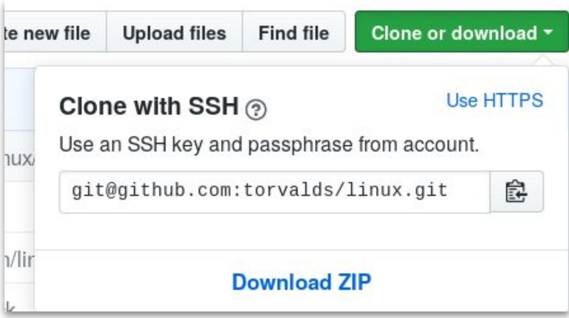
[7,8]

Distributed VCS



Demo: git

Demo: shells



Clone with SSH

Use an SSH key and passphrase from account.

```
git@github.com:torvalds/linux.git
```

What to do

1. Create a new repository
   - *git init* PROJECT1
   - Creates a new folder PROJECT1
   - *cd* PROJECT1
2. Add some files
   - *echo "hallo" > file.txt*
   - *ls/dir*
   - *git status*
   - *git add file.txt*
   - *git status*
   - *git commit -m “added a file”*
   - *git status*

Folder contains *file.txt*

Untracked files: “file.txt”

Changes to be committed: “file.txt”

nothing to commit, working tree clean

OR

1. Clone a repository
   - *git clone GIT_URL*
   - Creates a local copy of the repository and adds URL as *remote* **origin**

3. Push local changes
   - (Create repo: https://github.com/new
   - *git remote add origin* GIT_URL)
   - *git push origin master*

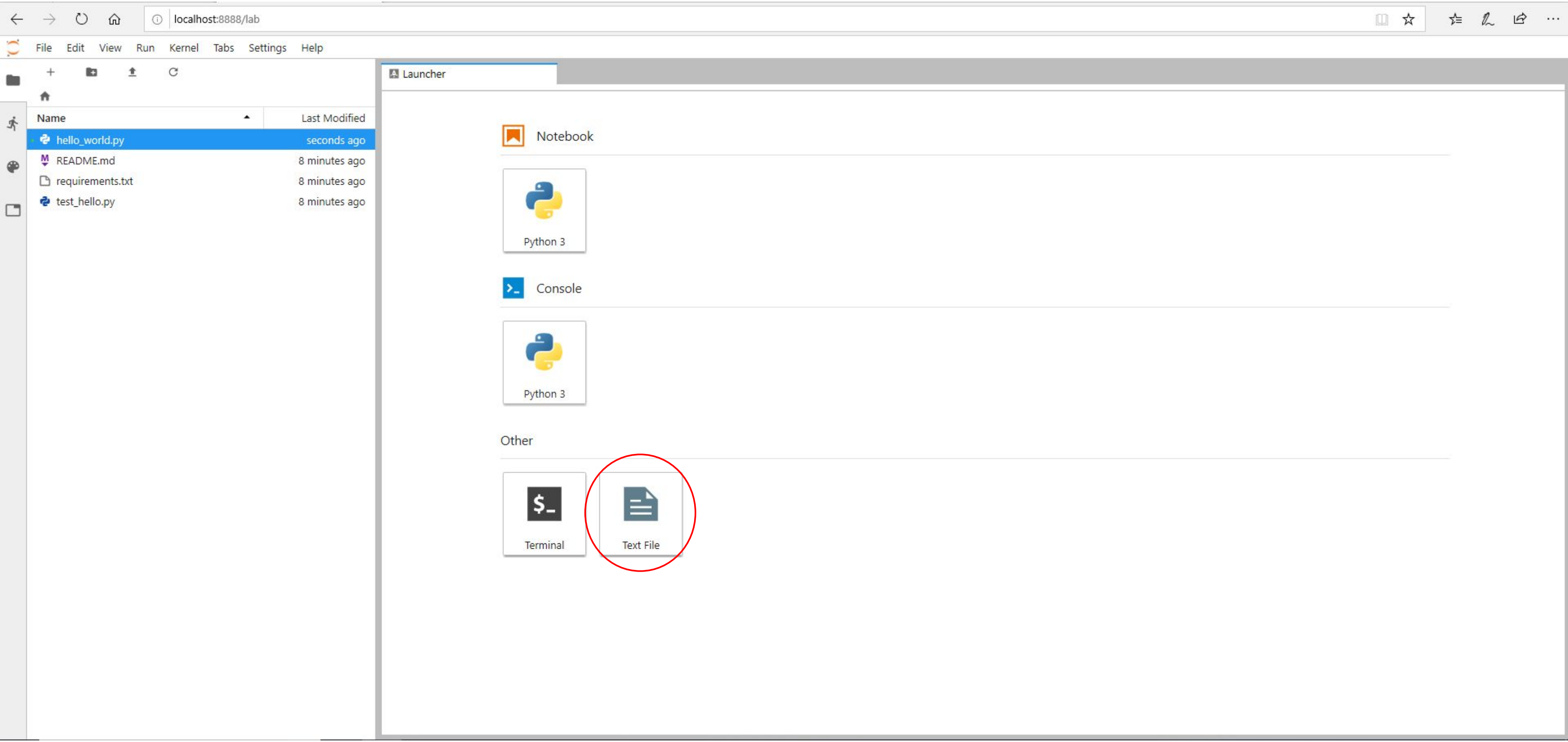
Git commands cheat-sheet

- `git init` to create a new project
- `git clone <url>` to copy an existing project from eg. GitHub or BitBucket
- `git status` to view which files changed in status
- `git diff` to view each file’s difference to the last commit (or also between commits)
- `git checkout <file>` to reset a file to the last commit
- `git checkout -b <branch> <hash>` to completely restore an older commit (to a branch)
- `git checkout <branch>` to switch branches (eg. back to master)
- `git log` to view your latest commits incl. messages
- `git pull` to update your local repository to the state of the one on GitHub/BitBucket
- `git push` to update the repository on GitHub/BitBucket to your local version
- `git add <file>` such that git will stage the file to be considered in the next commit
- `git rm <file>` to delete a file from filesystem and also stop tracking it
- `git commit -m “<message>”` to create a commit of the currently staged files

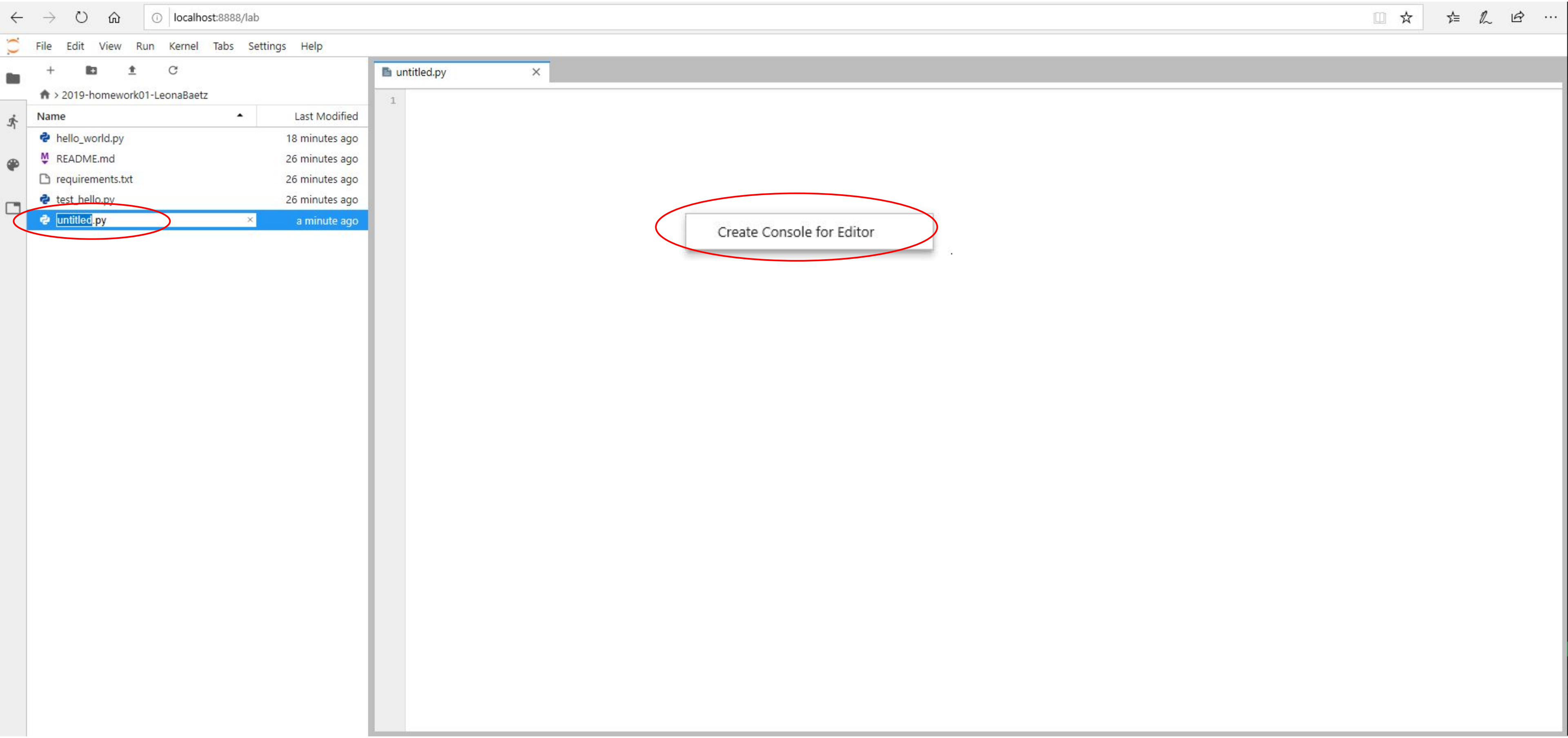
Final remarks on git

- Git is made for source-code and is no dropbox! → Add only text-based (diff’able) files
- Gitignores may help to not add unnecessary files. Add a “.gitignore” file to your repository and write filename-masks you want git to completely ignore into it. A useful start is https://github.com/github/gitignore/blob/master/Python.gitignore
- Modern editors all come with git integration, however we advise to not use it until you’re really familiar with git! Learning the console always helps!
- If you are not familiar with git https://try.github.io/
- For deeper looks https://git-scm.com/book/en/v2

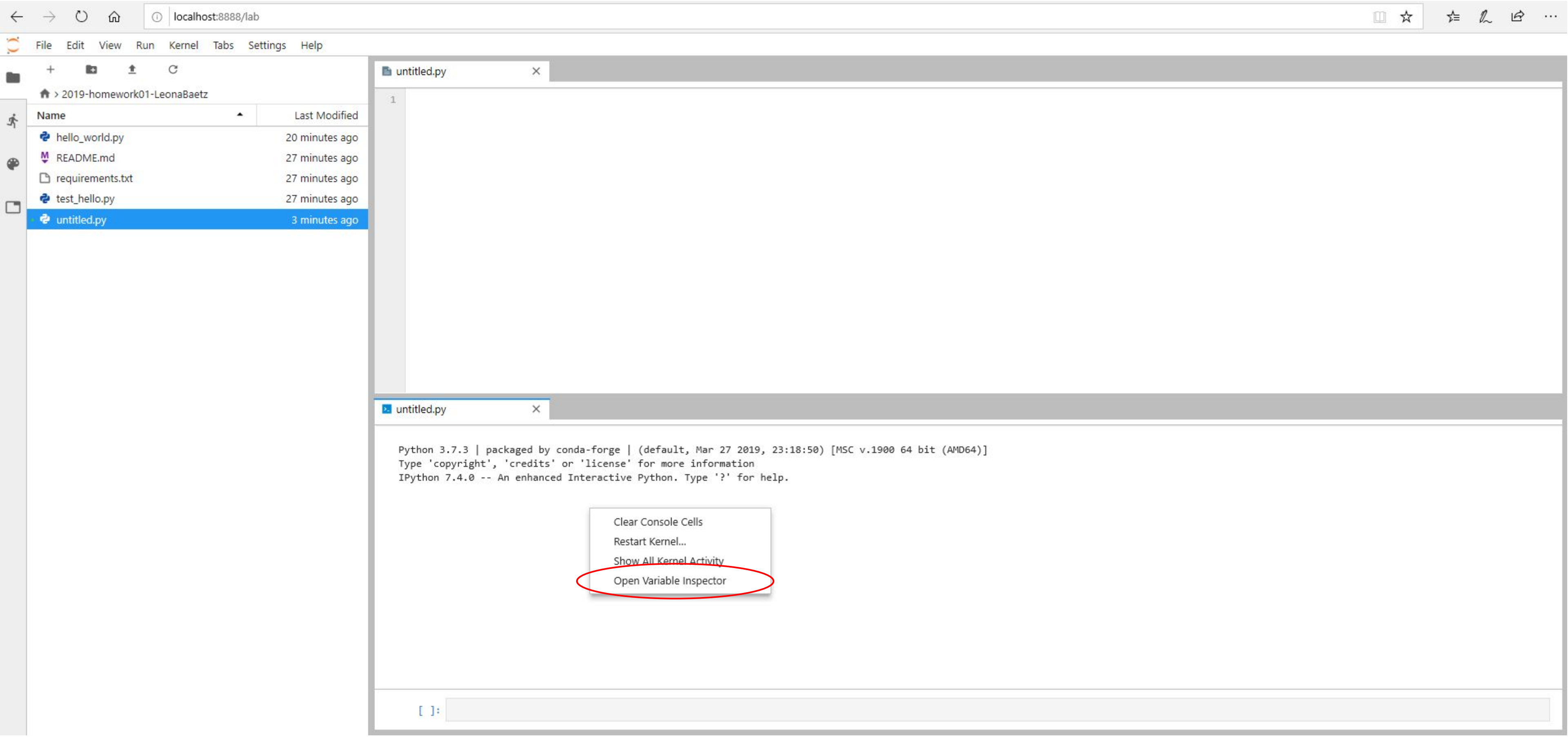
Howto: Interactive Kernel



localhost:8888/lab
File
Edit
View
Run
Kernel
Tabs
Settings
Help
Launcher
Name
Last Modified
hello_world.py
seconds ago
README.md
8 minutes ago
requirements.txt
8 minutes ago
test_hello.py
8 minutes ago
Notebook
Python 3
Console
Python 3
Other
Terminal
Text File



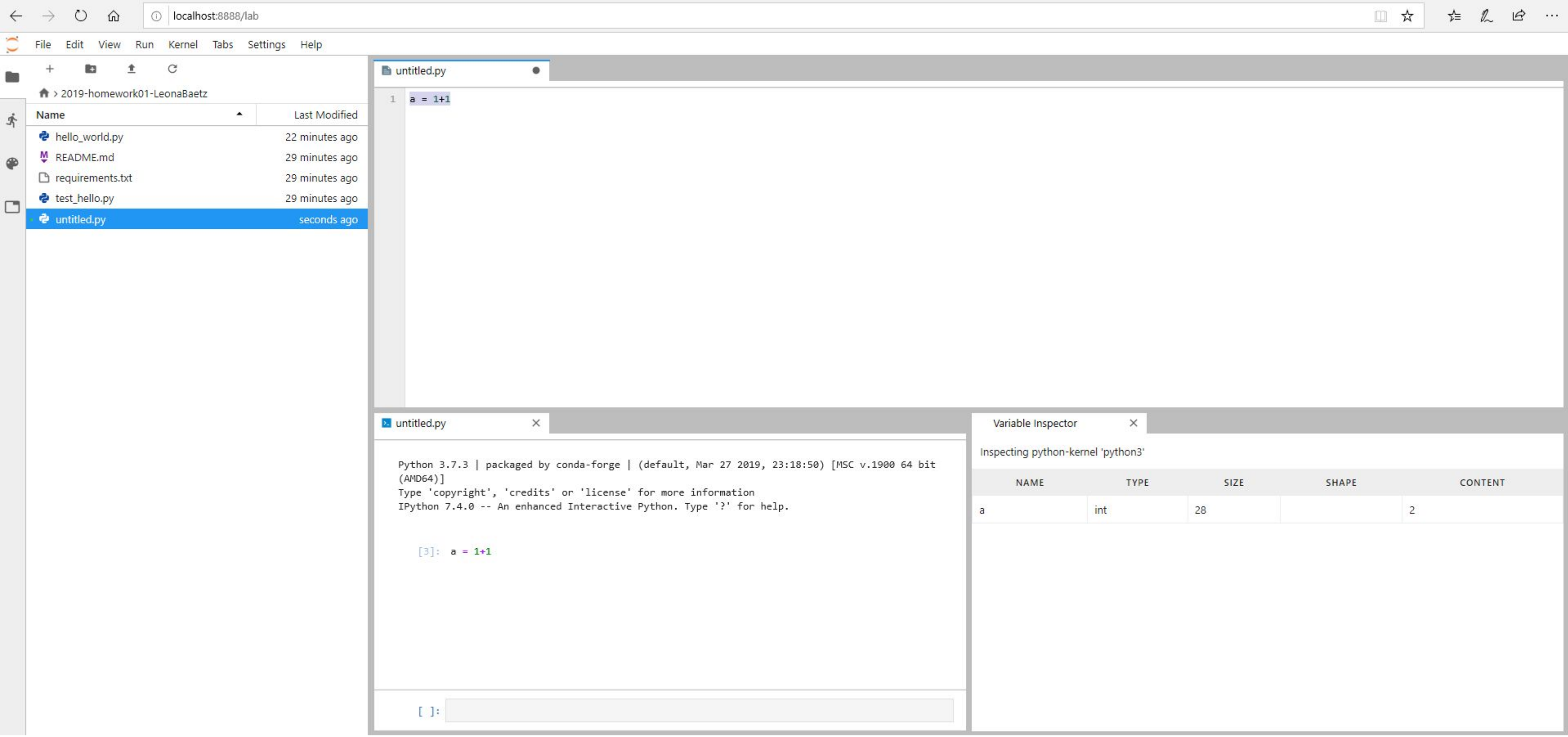
localhost:8888/lab
File
Edit
View
Run
Kernel
Tabs
Settings
Help
2019-homework01-LeonaBaetz
Name
Last Modified
hello_world.py
18 minutes ago
README.md
26 minutes ago
requirements.txt
26 minutes ago
test_hello.py
26 minutes ago
untitled.py
a minute ago
untitled.py
Create Console for Editor



2019-homework01-LeonaBaetz

Name	Last Modified
hello_world.py	20 minutes ago
README.md	27 minutes ago
requirements.txt	27 minutes ago
test_hello.py	27 minutes ago
untitled.py	3 minutes ago

```
Python 3.7.3 | packaged by conda-forge | (default, Mar 27 2019, 23:18:50) [MSC v.1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.4.0 -- An enhanced Interactive Python. Type '?' for help.
```



Name	Last Modified
hello_world.py	22 minutes ago
README.md	29 minutes ago
requirements.txt	29 minutes ago
test_hello.py	29 minutes ago
untitled.py	seconds ago

```
a = 1+1
```

```
Python 3.7.3 | packaged by conda-forge | (default, Mar 27 2019, 23:18:50) [MSC v.1900 64 bit
(AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.4.0 -- An enhanced Interactive Python. Type '?' for help.


  [3]:  a = 1+1

  [ ]:
```

Inspecting python-kernel 'python3'

NAME	TYPE	SIZE	SHAPE	CONTENT
a	int	28		2

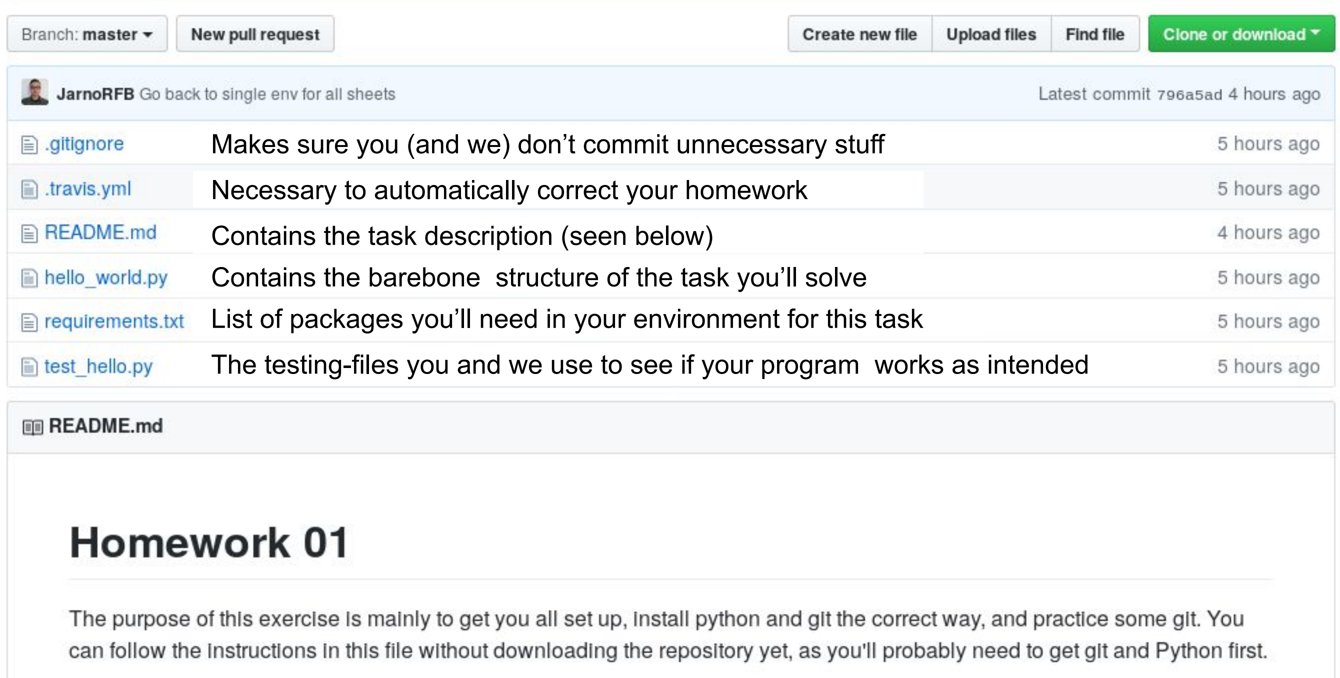
Getting used to Jupyter lab

- Jupyter Lab Cheat-sheet: https://www.cheatography.com/weidadeyue/cheat-sheets/jupyter-notebook/pdf_bw/
- Markdown-Commands Cheat-Sheet: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet

Homework

Homework

- Homework is distributed via *Github classroom*
- You need a github-account to submit the homework!
- If there are mistakes in the homework, we will announce that and update the repositories, so regularly pull!



JarnoRFB Go back to single env for all sheets Latest commit 796a5ad 4 hours ago

.gitignore	Makes sure you (and we) don’t commit unnecessary stuff	5 hours ago
.travis.yml	Necessary to automatically correct your homework	5 hours ago
README.md	Contains the task description (seen below)	4 hours ago
hello_world.py	Contains the barebone structure of the task you’ll solve	5 hours ago
requirements.txt	List of packages you’ll need in your environment for this task	5 hours ago
test_hello.py	The testing-files you and we use to see if your program works as intended	5 hours ago

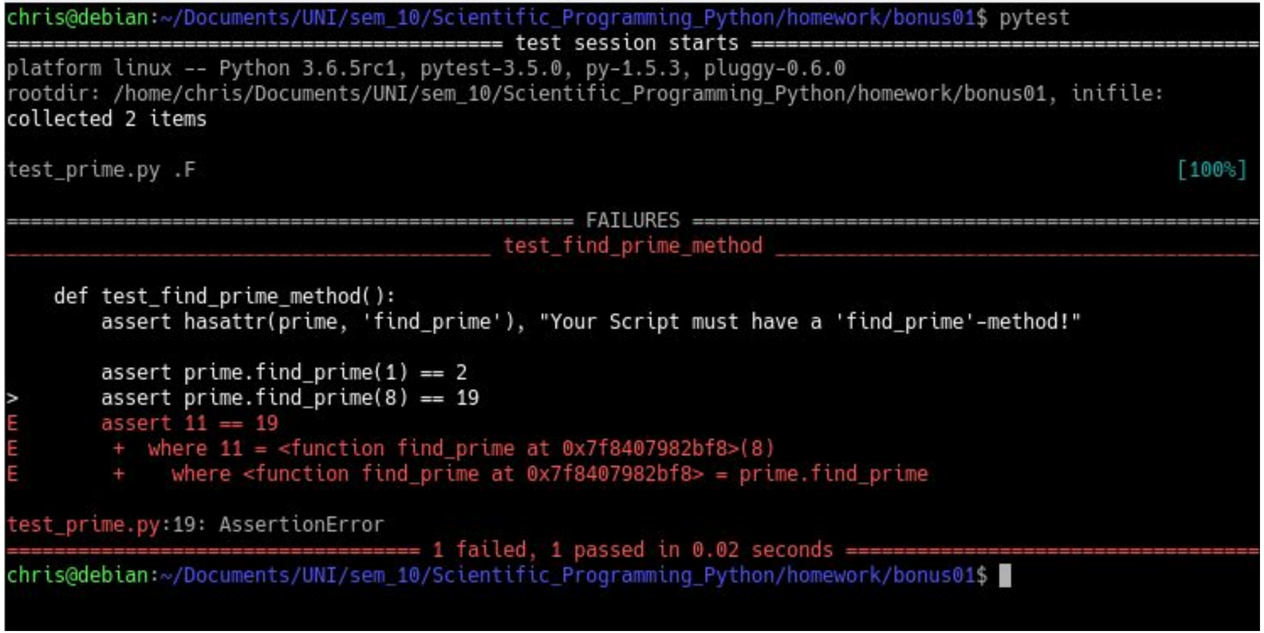
README.md

Homework 01

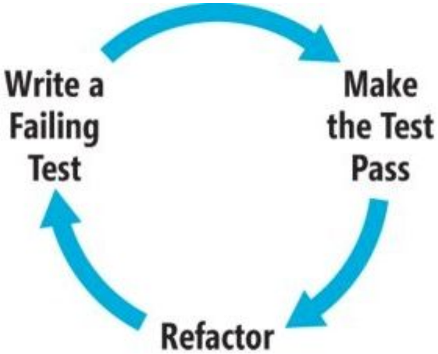
The purpose of this exercise is mainly to get you all set up, install python and git the correct way, and practice some git. You can follow the instructions in this file without downloading the repository yet, as you'll probably need to get git and Python first.

Pytest and Test-Driven Development

- `pytest` is a testing library included with python
- It will grab all `test_`-functions in all `test_`-files, execute them, and check for errors
- To do so, it uses assertions: `assert prime.find_prime(1)==2`

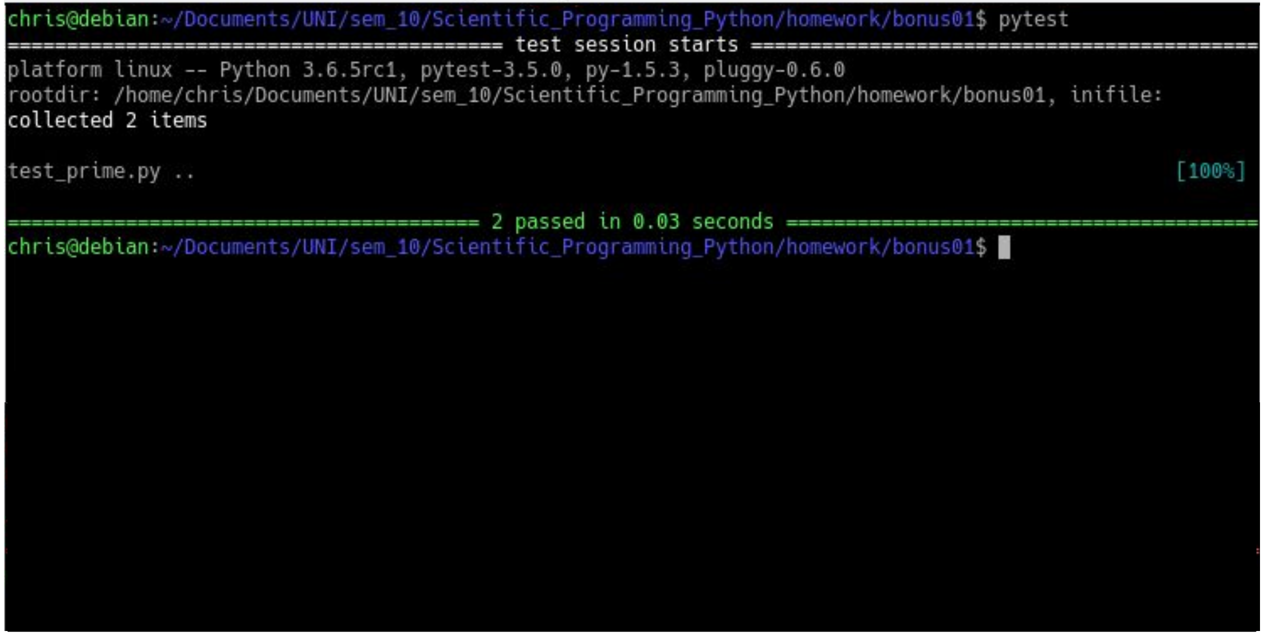


```
chris@debian:~/Documents/UNI/sem_10/Scientific_Programming_Python/homework/bonus01$ pytest
=========================================== test session starts ============================================
platform linux -- Python 3.6.5rc1, pytest-3.5.0, py-1.5.3, pluggy-0.6.0
rootdir: /home/chris/Documents/UNI/sem_10/Scientific_Programming_Python/homework/bonus01, inifile:
collected 2 items

test_prime.py .F                                                                                    [100%]

================================================= FAILURES =================================================
_________________________________________ test_find_prime_method ___________________________________________

    def test_find_prime_method():
        assert hasattr(prime, 'find_prime'), "Your Script must have a 'find_prime'-method!"

        assert prime.find_prime(1) == 2
>       assert prime.find_prime(8) == 19
E       assert 11 == 19
E        +  where 11 = <function find_prime at 0x7f8407982bf8>(8)
E        +    where <function find_prime at 0x7f8407982bf8> = prime.find_prime

test_prime.py:19: AssertionError
==================================== 1 failed, 1 passed in 0.02 seconds ====================================
chris@debian:~/Documents/UNI/sem_10/Scientific_Programming_Python/homework/bonus01$ █
```

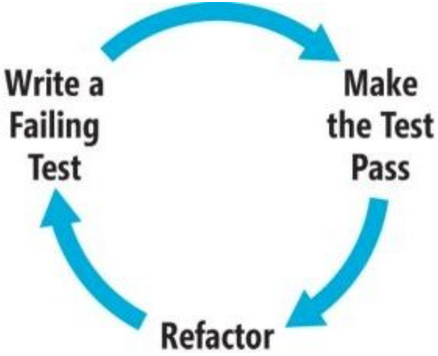


Pytest and Test-Driven Development

- `pytest` is a testing library included with python
- It will grab all `test_`-functions in all `test_`-files, execute them, and check for errors
- To do so, it uses assertions: `assert prime.find_prime(1)==2`



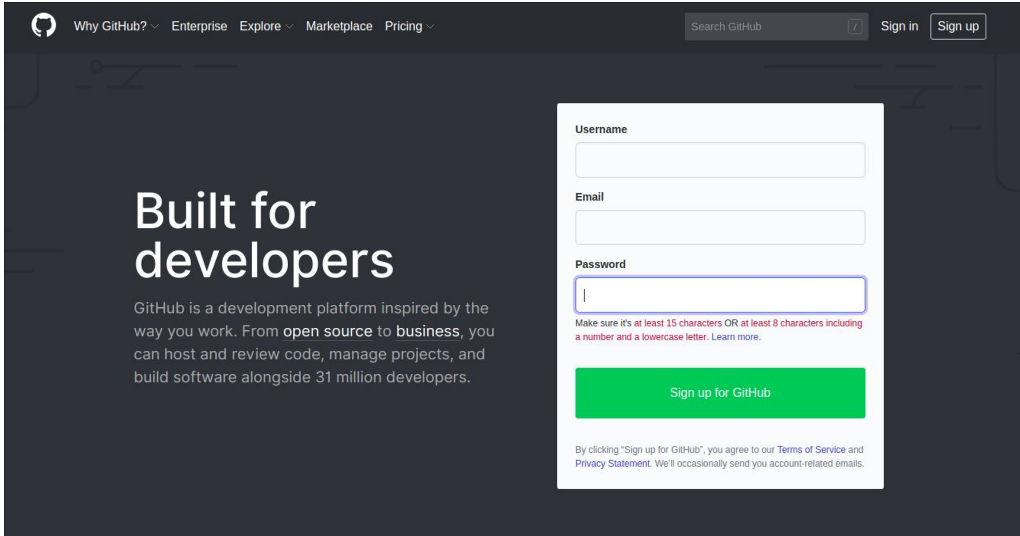
```
chris@debian:~/Documents/UNI/sem_10/Scientific_Programming_Python/homework/bonus01$ pytest
========================================== test session starts ==========================================
platform linux -- Python 3.6.5rc1, pytest-3.5.0, py-1.5.3, pluggy-0.6.0
rootdir: /home/chris/Documents/UNI/sem_10/Scientific_Programming_Python/homework/bonus01, inifile:
collected 2 items

test_prime.py ..                                                                                [100%]

======================================= 2 passed in 0.03 seconds ========================================
chris@debian:~/Documents/UNI/sem_10/Scientific_Programming_Python/homework/bonus01$
```



Once this is the result of pytest, your homework will pass!

Create a GitHub account

- Go to https://github.com/
- Your GitHub account is like your portfolio for programming, so use something you can show to others instead of xXOmqItzPotatoXx



Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 31 million developers.

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

By clicking "Sign up for GitHub", you agree to our Terms of Service and Privacy Statement. We'll occasionally send you account-related emails.

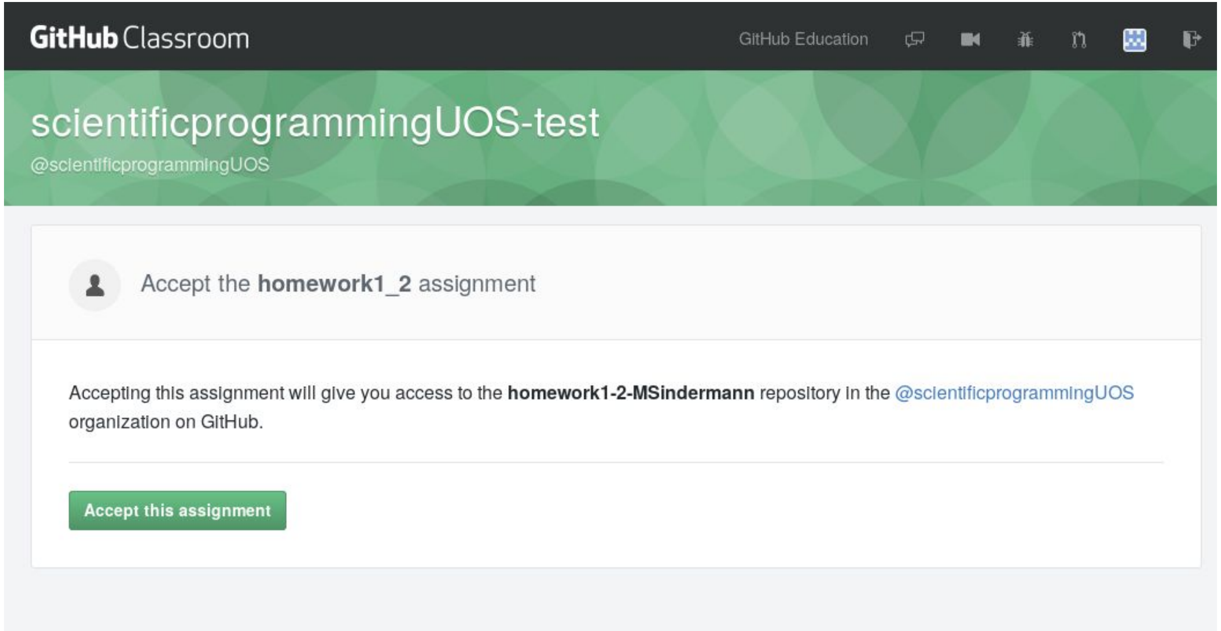
Your homework

- Homework-link:
  - https://classroom.github.com/a/2RXLdM_r

<demo: homework>

How to download and work on Homework

- Homework-link:
  - https://classroom.github.com/a/2RXLdM_r



scientificprogrammingUOS-test

@scientificprogrammingUOS

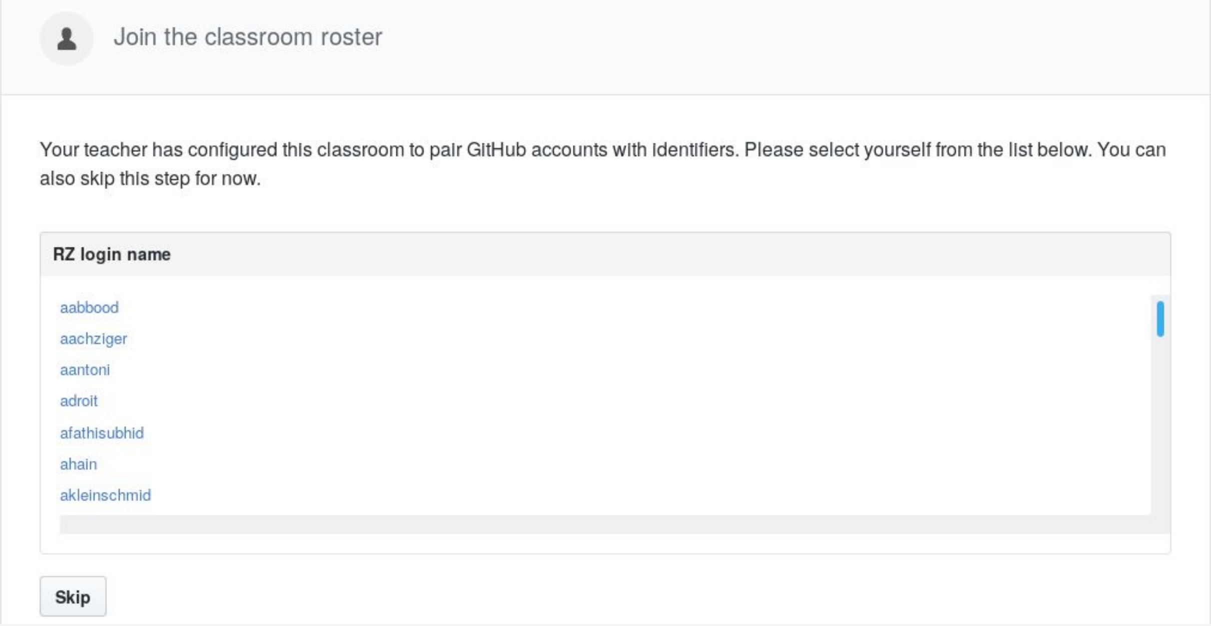
Accept the **homework1_2** assignment

Accepting this assignment will give you access to the **homework1-2-MSindermann** repository in the @scientificprogrammingUOS organization on GitHub.

- You need to sign in to github
- Use your @uos-mail to get unlimited free repositories! https://education.github.com/pack/offers

How to download and work on Homework

- Homework-link:
  - https://classroom.github.com/a/2RXLdM_r



Join the classroom roster

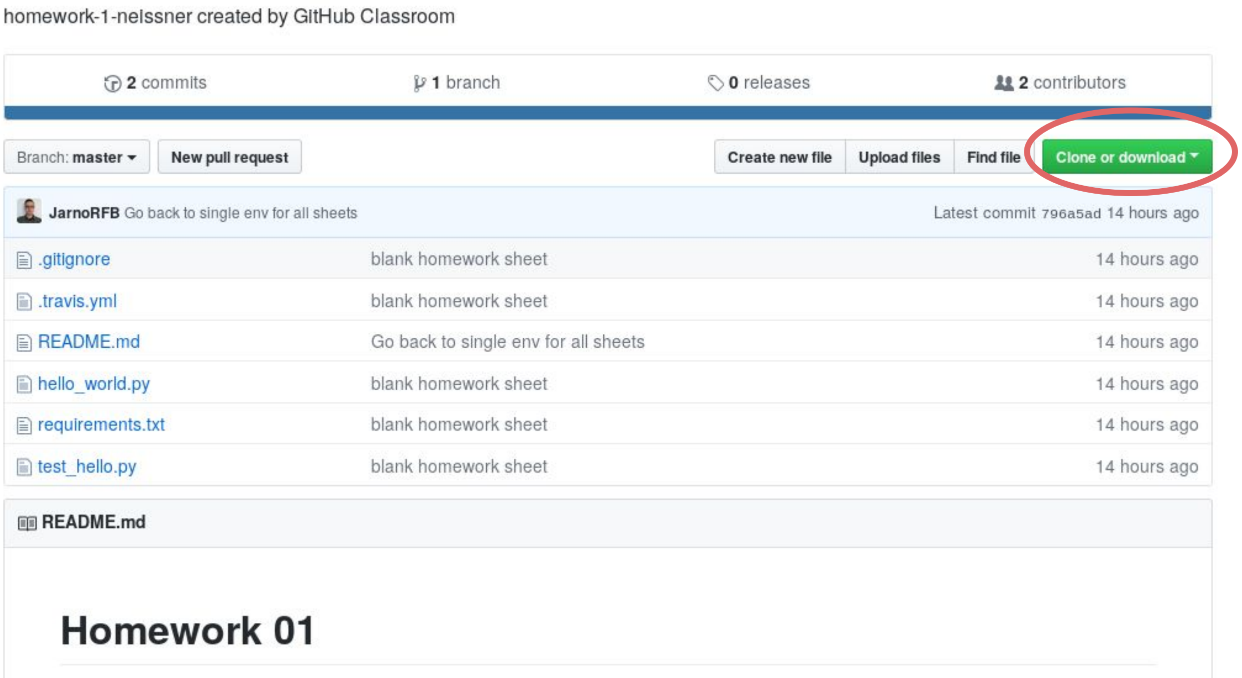
Your teacher has configured this classroom to pair GitHub accounts with identifiers. Please select yourself from the list below. You can also skip this step for now.

RZ login name
aabbood
aachziger
aantoni
adroit
afathisubhid
ahain
akleinschmid

- If your RZ-login is not listed here or spelled incorrectly, please write us an email!
- When it says *“preparing your new repository, there is no need to keep this window open, we'll email you when the import is done”*, just hit F5 after 5 seconds

How to download and work on Homework

- Homework-link:
  - https://classroom.github.com/a/2RXLdM_r



homework-1-neissner created by GitHub Classroom

JarnoRFB Go back to single env for all sheets — Latest commit 796a5ad 14 hours ago

.gitignore	blank homework sheet	14 hours ago
.travis.yml	blank homework sheet	14 hours ago
README.md	Go back to single env for all sheets	14 hours ago
hello_world.py	blank homework sheet	14 hours ago
requirements.txt	blank homework sheet	14 hours ago
test_hello.py	blank homework sheet	14 hours ago

README.md

Homework 01

- If your RZ-login is not listed here or spelled incorrectly, please write us an email!
- When it says *“preparing your new repository, there is no need to keep this window open, we'll email you when the import is done”*, just hit F5 after 5 seconds

How to download and work on Homework

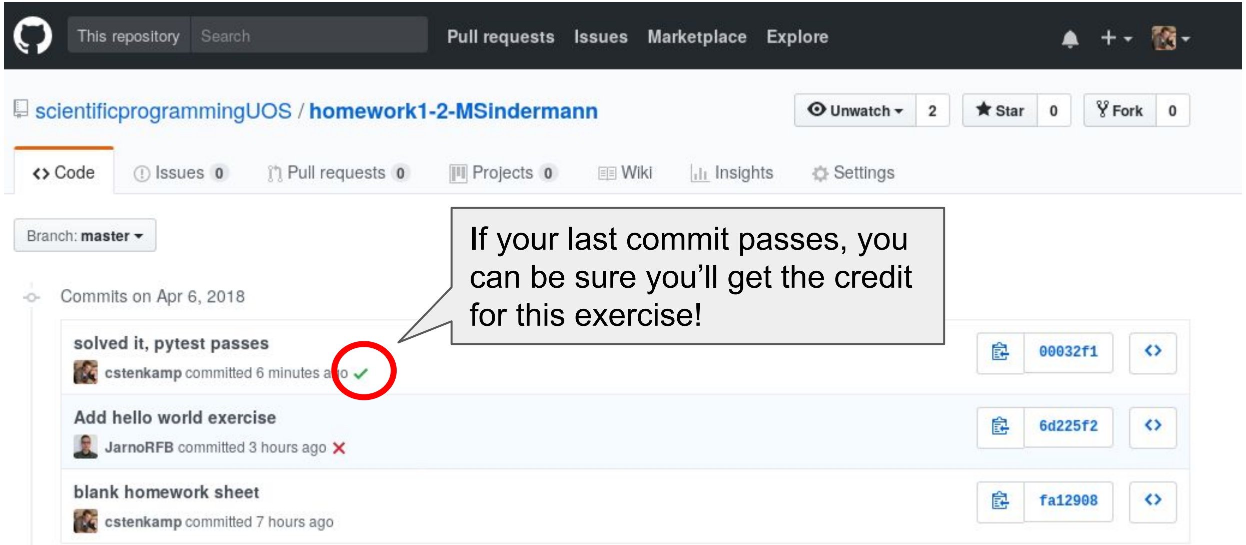
- Homework-link:
  - https://classroom.github.com/a/2RXLdM_r

```
git clone REPOSITORY_URL
cd YOUR_REPOSITORY_PATH
conda activate scientific_programming
jupyter lab .
...
pytest
...
git status
git add CHANGED_FILE
git commit -m “solved the exercise”
git push origin master
conda deactivate
```

How to download and work on Homework

- Homework-links:
  - https://classroom.github.com/a/2RXLdM_r

- `git config --get remote.origin.url` tells you the domain of your remote repository



scientificprogrammingUOS / **homework1-2-MSindermann**

Commits on Apr 6, 2018

solved it, pytest passes **cstenkamp** committed 6 minutes ago ✓	00032f1
Add hello world exercise **JarnoRFB** committed 3 hours ago ✗	6d225f2
blank homework sheet **cstenkamp** committed 7 hours ago	fa12908

If your last commit passes, you can be sure you'll get the credit for this exercise!

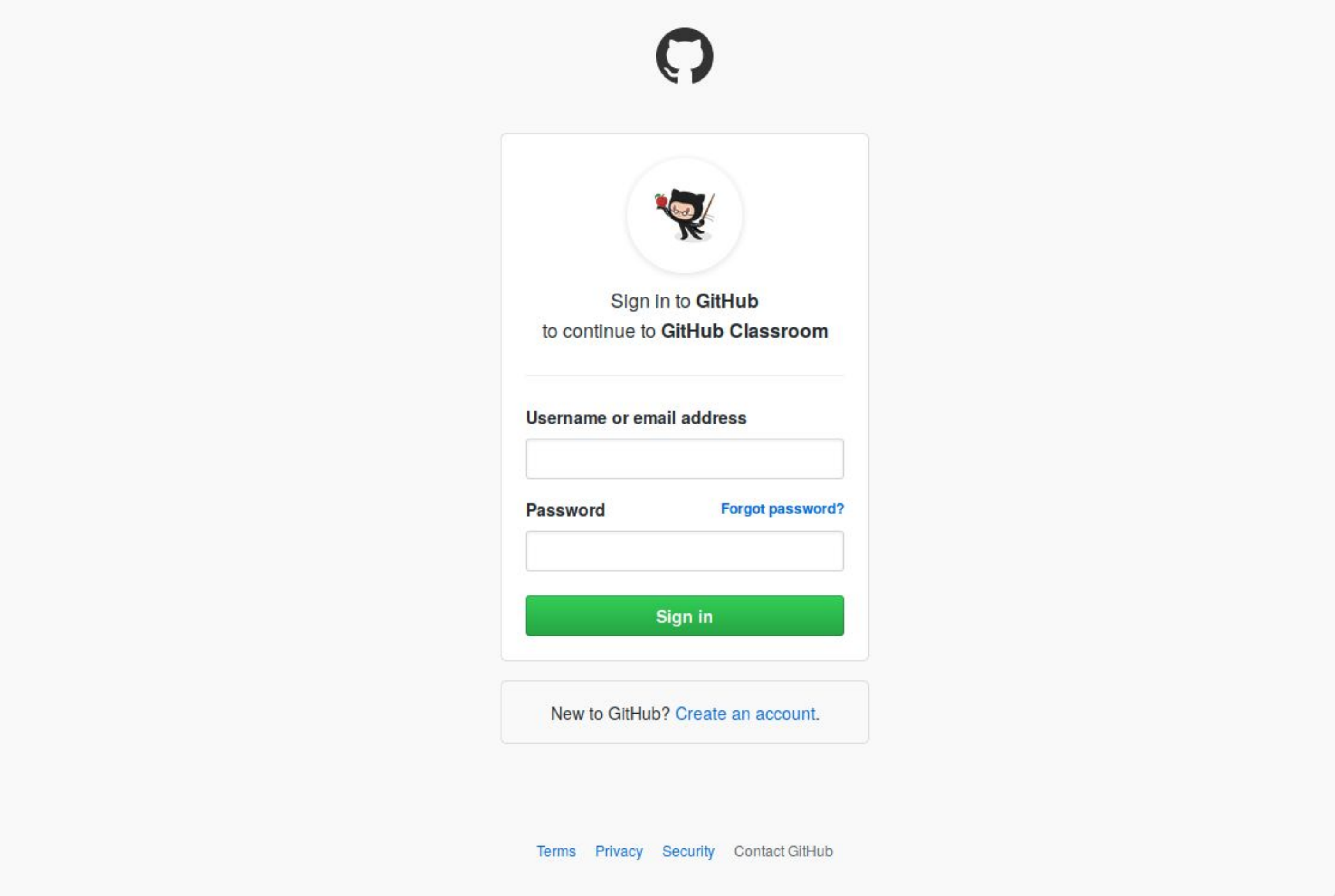
- Note that the first check will be performed 5-15 minutes after accepting the exercise, from then on ASAP

How to download and work on Homework

- Homework-links:
  - https://classroom.github.com/a/2RXLdM_r

- Note that the first check will be performed 5-15 minutes after accepting the exercise, from then on ASAP
- First deadline is **next Tuesday, 12:00** (hard)
- Clone the repository >15 minutes before the deadline passes, make the last commit before the deadline passes
- Yes, you get the test-scripts, but don’t change them, we check for that!
- You’ll get an email telling you if you passed or failed, including your overall pass/fail-count right after the next lecture!

<Screenshots of Presentation>

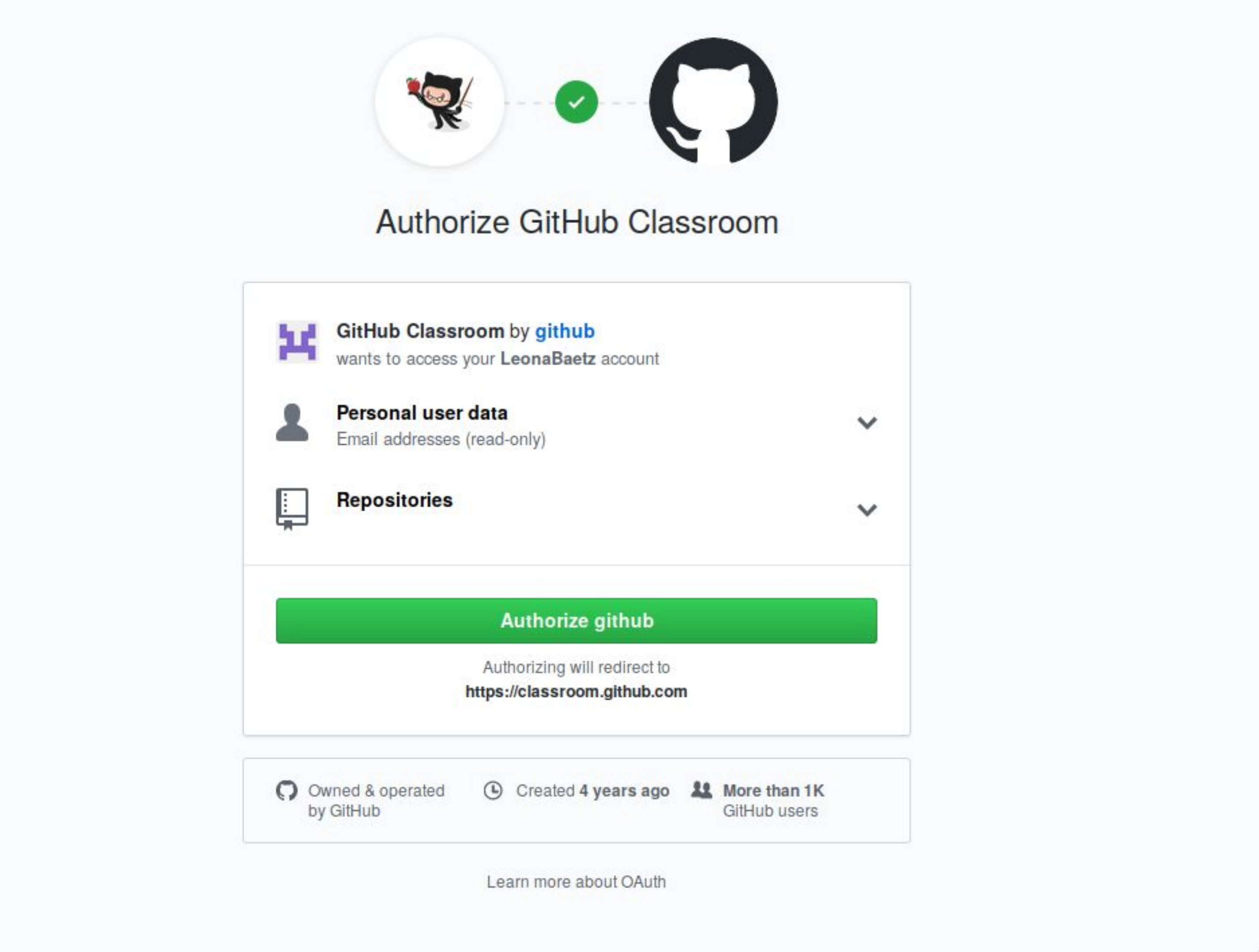


Sign in to **GitHub**
to continue to **GitHub Classroom**

Username or email address

Password Forgot password?

New to GitHub? Create an account.



Authorize GitHub Classroom

GitHub Classroom by **github**
wants to access your **LeonaBaetz** account

Personal user data
Email addresses (read-only)

Repositories

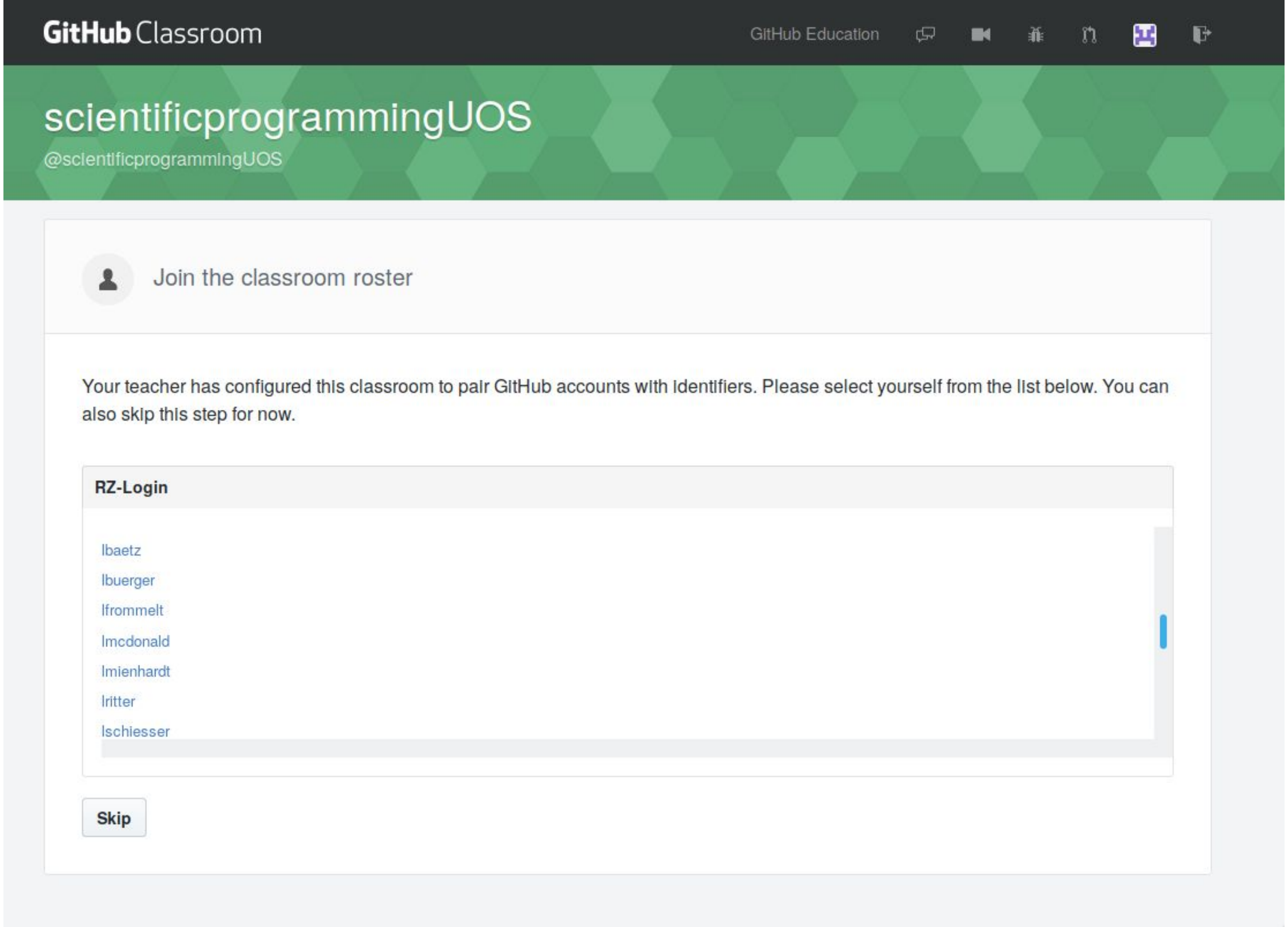
Authorizing will redirect to
https://classroom.github.com

Owned & operated by GitHub

Created **4 years ago**

More than 1K GitHub users

Learn more about OAuth



scientificprogrammingUOS

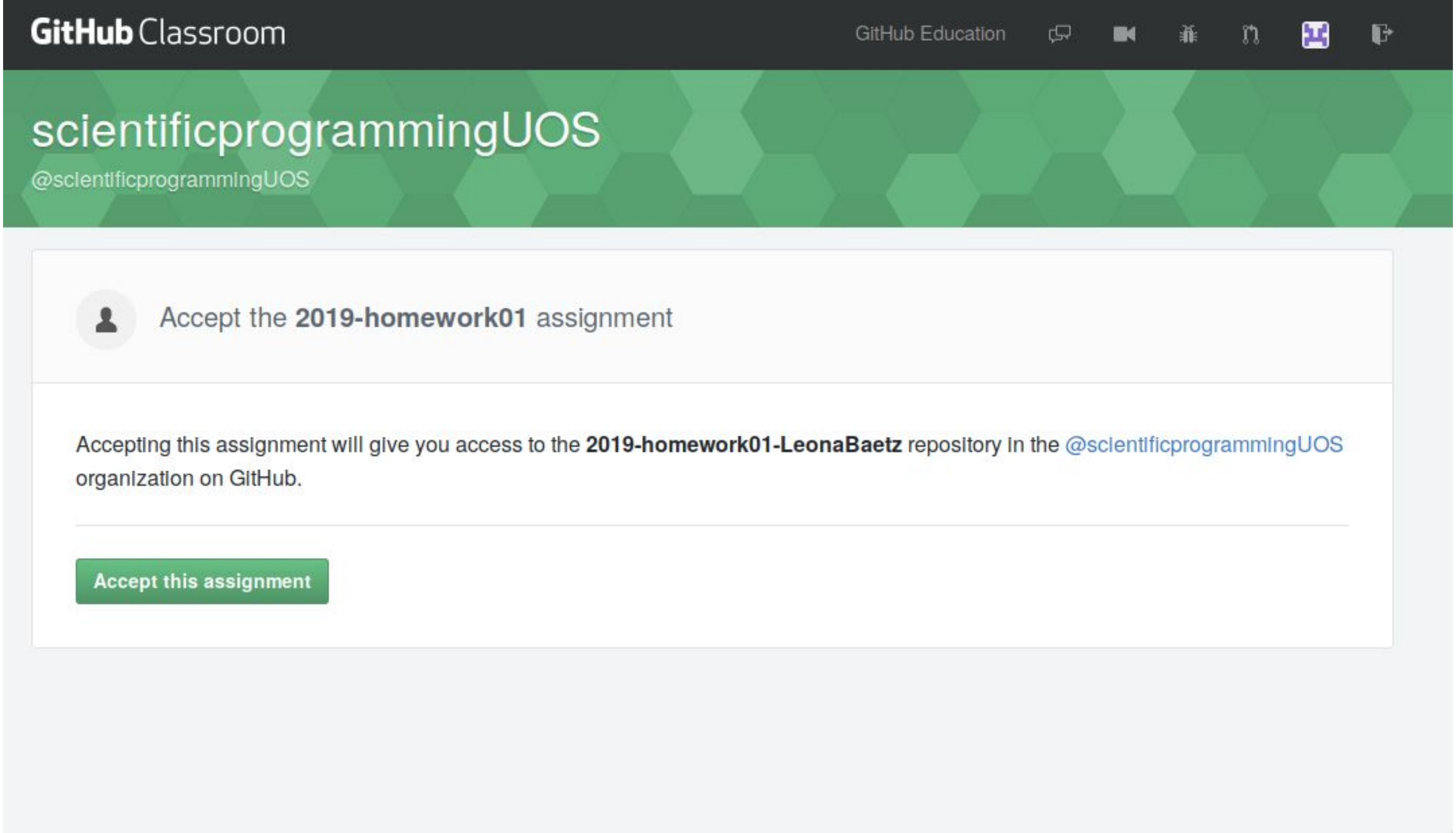
@scientificprogrammingUOS

Join the classroom roster

Your teacher has configured this classroom to pair GitHub accounts with identifiers. Please select yourself from the list below. You can also skip this step for now.

RZ-Login

lbaetz
lbuerger
lfrommelt
lmcdonald
lmienhardt
lritter
lschiesser

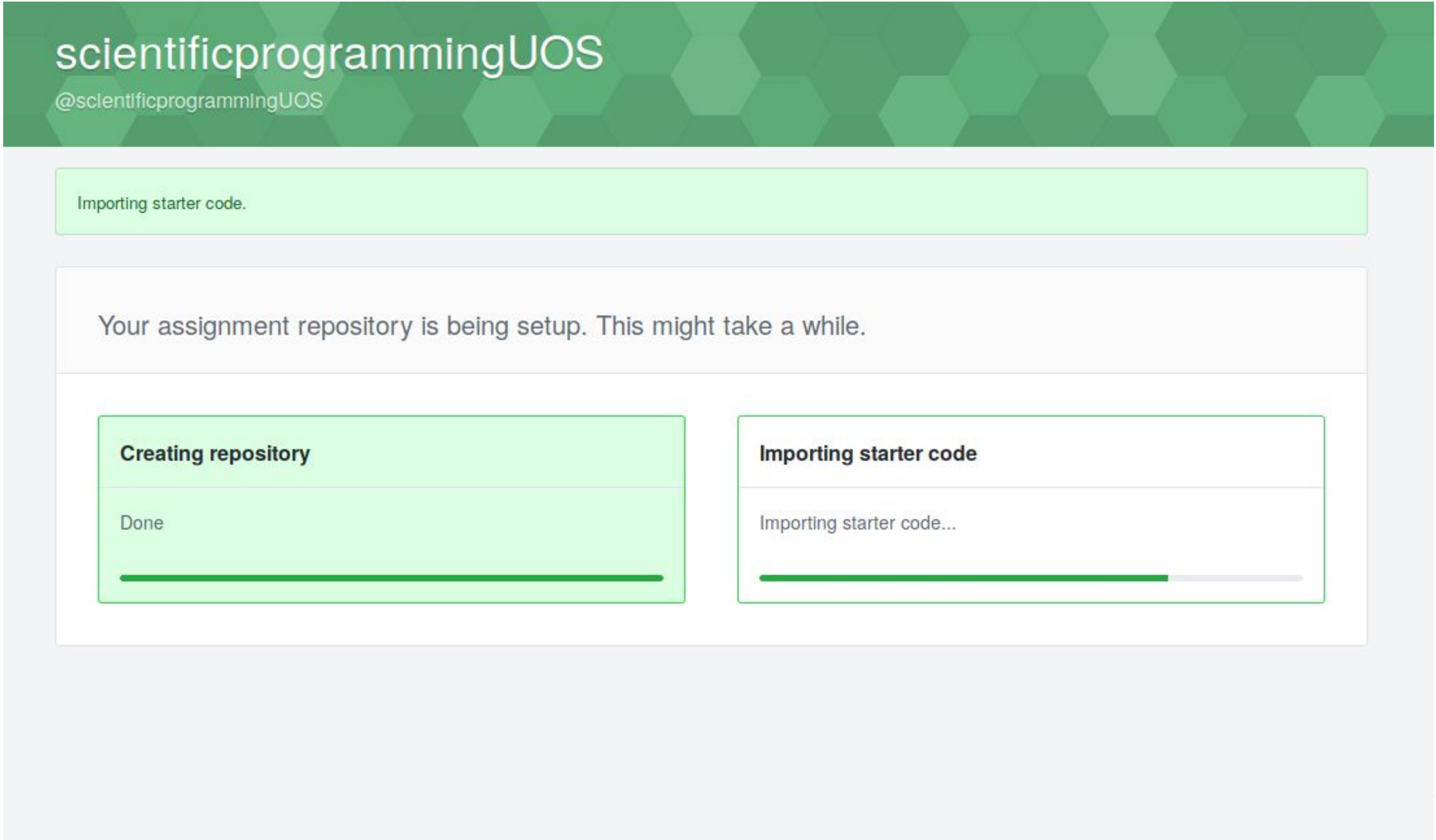


scientificprogrammingUOS

@scientificprogrammingUOS

Accept the **2019-homework01** assignment

Accepting this assignment will give you access to the **2019-homework01-LeonaBaetz** repository in the @scientificprogrammingUOS organization on GitHub.



scientificprogrammingUOS

@scientificprogrammingUOS

Importing starter code.

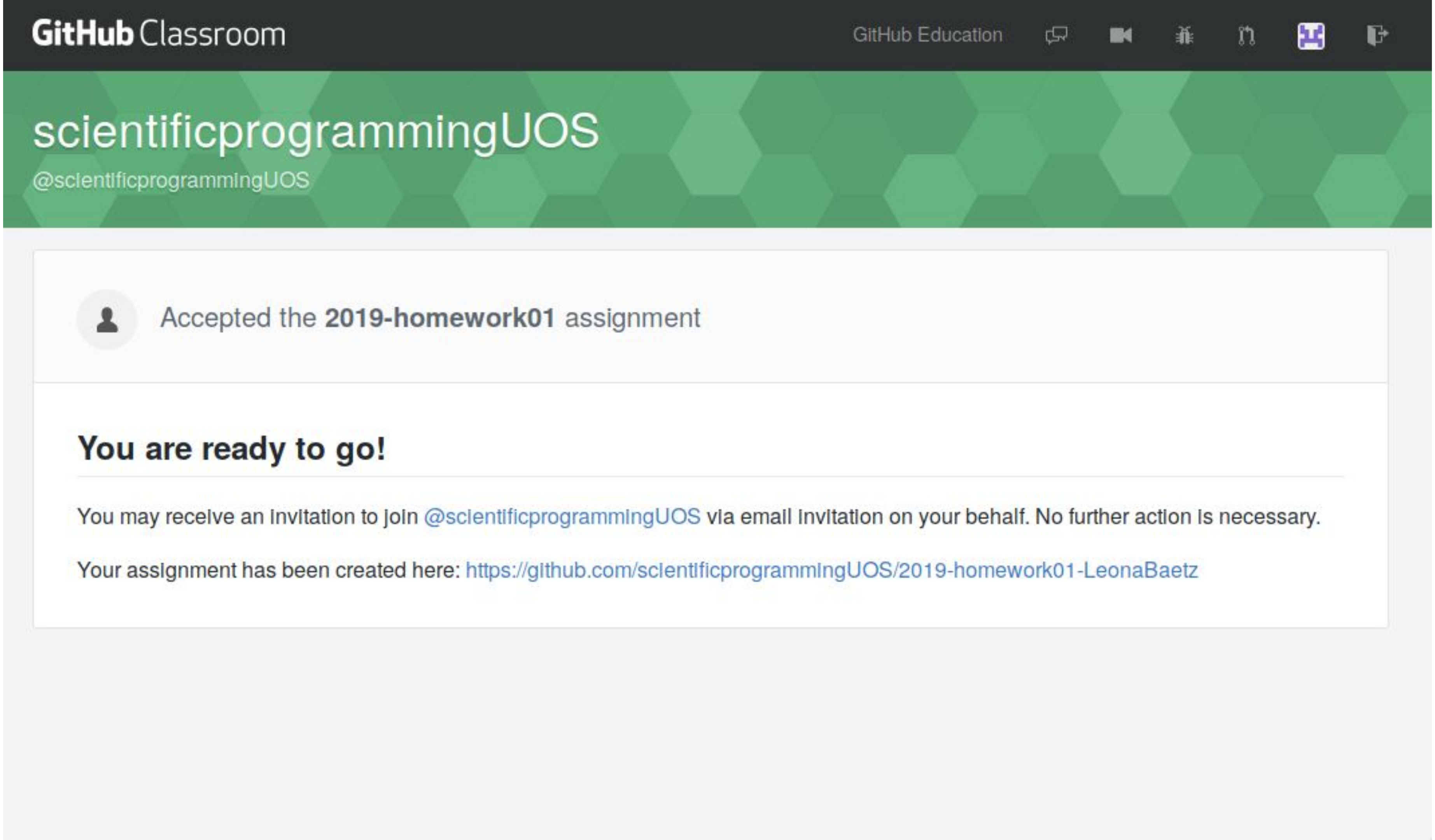
Your assignment repository is being setup. This might take a while.

Creating repository

Done

Importing starter code

Importing starter code...



scientificprogrammingUOS

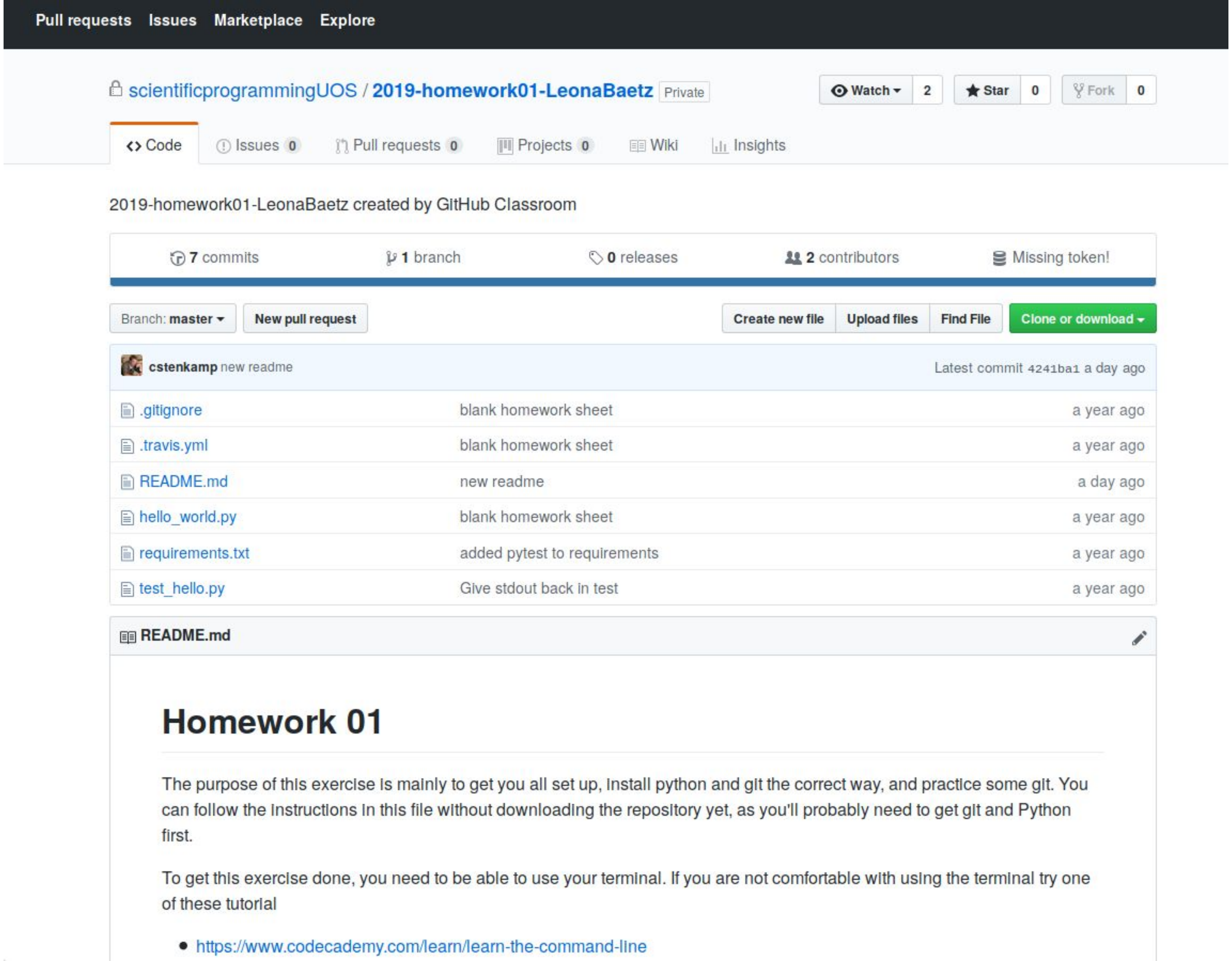
@scientificprogrammingUOS

Accepted the **2019-homework01** assignment

You are ready to go!

You may receive an invitation to join @scientificprogrammingUOS via email invitation on your behalf. No further action is necessary.

Your assignment has been created here: https://github.com/scientificprogrammingUOS/2019-homework01-LeonaBaetz



scientificprogrammingUOS / 2019-homework01-LeonaBaetz Private

2019-homework01-LeonaBaetz created by GitHub Classroom

7 commits · 1 branch · 0 releases · 2 contributors · Missing token!

cstenkamp new readme — Latest commit 4241ba1 a day ago

.gitignore	blank homework sheet	a year ago
.travis.yml	blank homework sheet	a year ago
README.md	new readme	a day ago
hello_world.py	blank homework sheet	a year ago
requirements.txt	added pytest to requirements	a year ago
test_hello.py	Give stdout back in test	a year ago

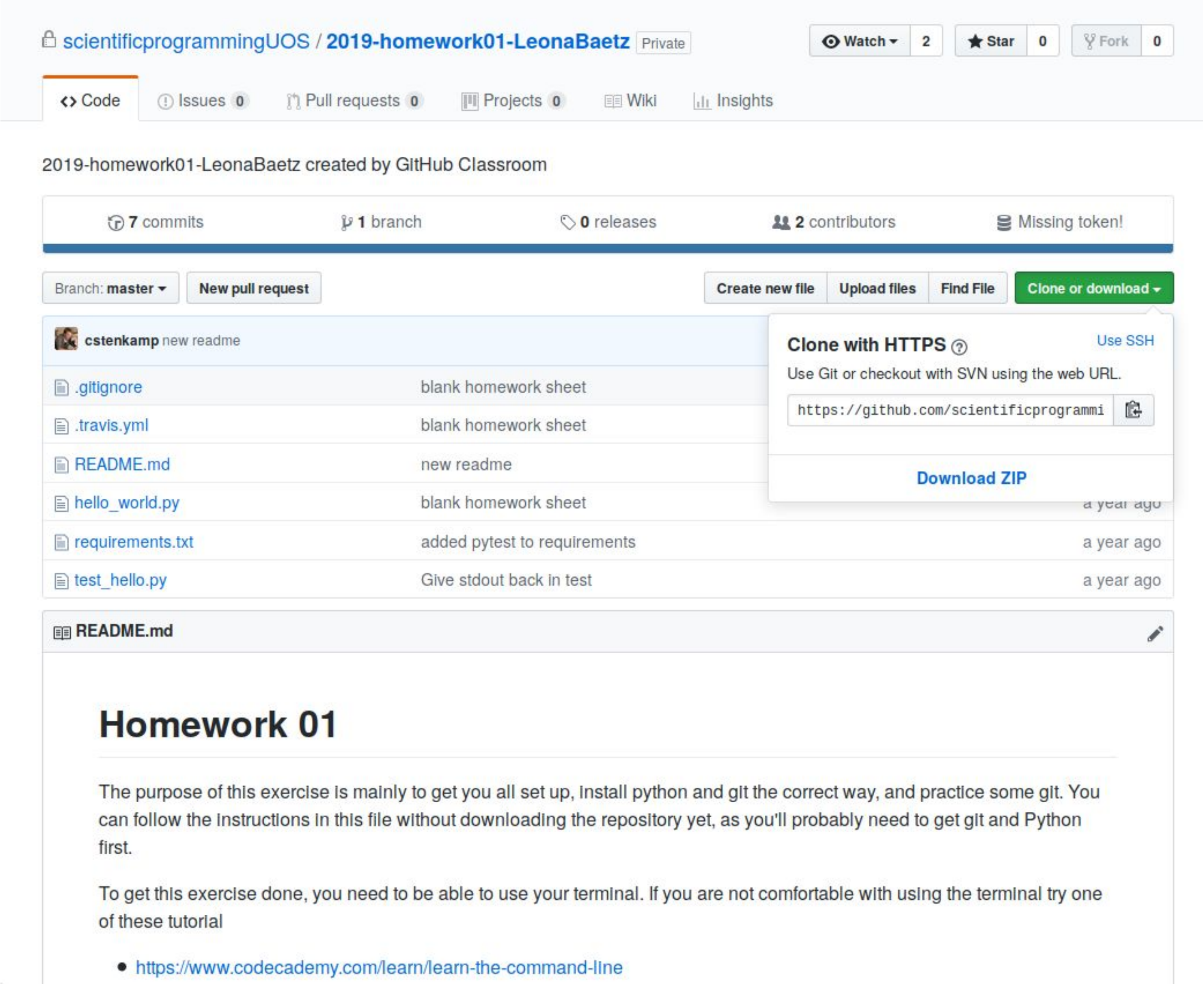
README.md

Homework 01

The purpose of this exercise is mainly to get you all set up, install python and git the correct way, and practice some git. You can follow the instructions in this file without downloading the repository yet, as you'll probably need to get git and Python first.

To get this exercise done, you need to be able to use your terminal. If you are not comfortable with using the terminal try one of these tutorial

- https://www.codecademy.com/learn/learn-the-command-line



2019-homework01-LeonaBaetz created by GitHub Classroom

cstenkamp new readme

.gitignore	blank homework sheet	
.travis.yml	blank homework sheet	
README.md	new readme	
hello_world.py	blank homework sheet	a year ago
requirements.txt	added pytest to requirements	a year ago
test_hello.py	Give stdout back in test	a year ago

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

```
https://github.com/scientificprogrammi
```

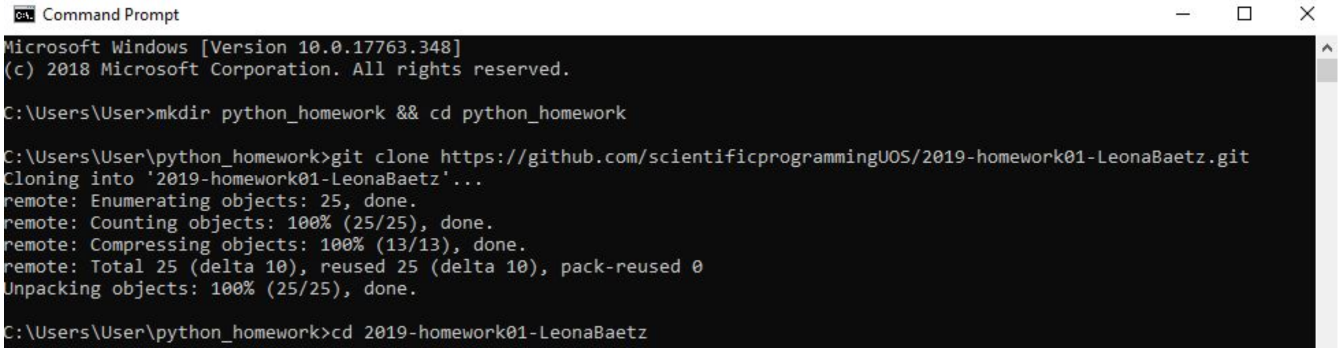
README.md

Homework 01

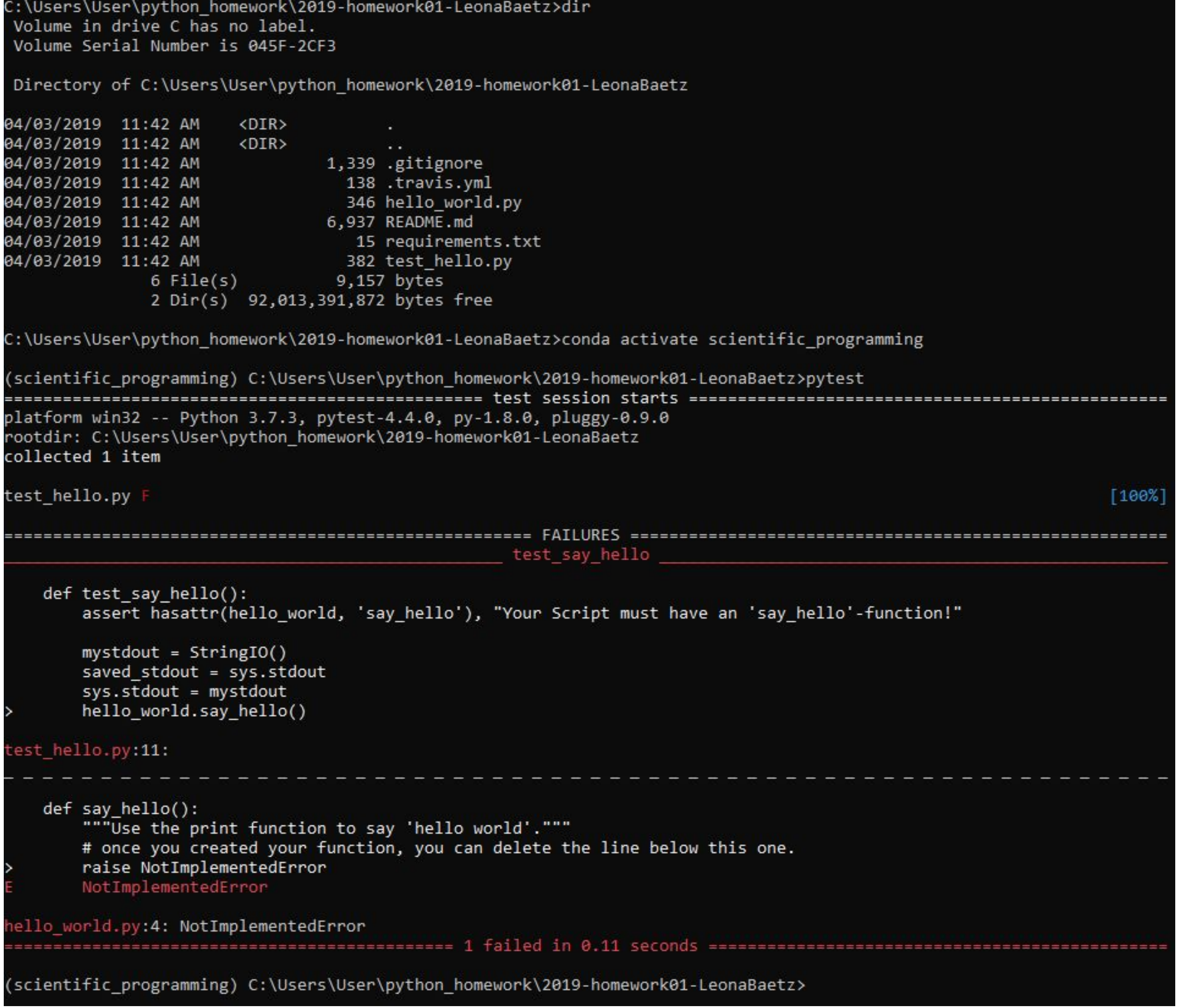
The purpose of this exercise is mainly to get you all set up, install python and git the correct way, and practice some git. You can follow the instructions in this file without downloading the repository yet, as you'll probably need to get git and Python first.

To get this exercise done, you need to be able to use your terminal. If you are not comfortable with using the terminal try one of these tutorial

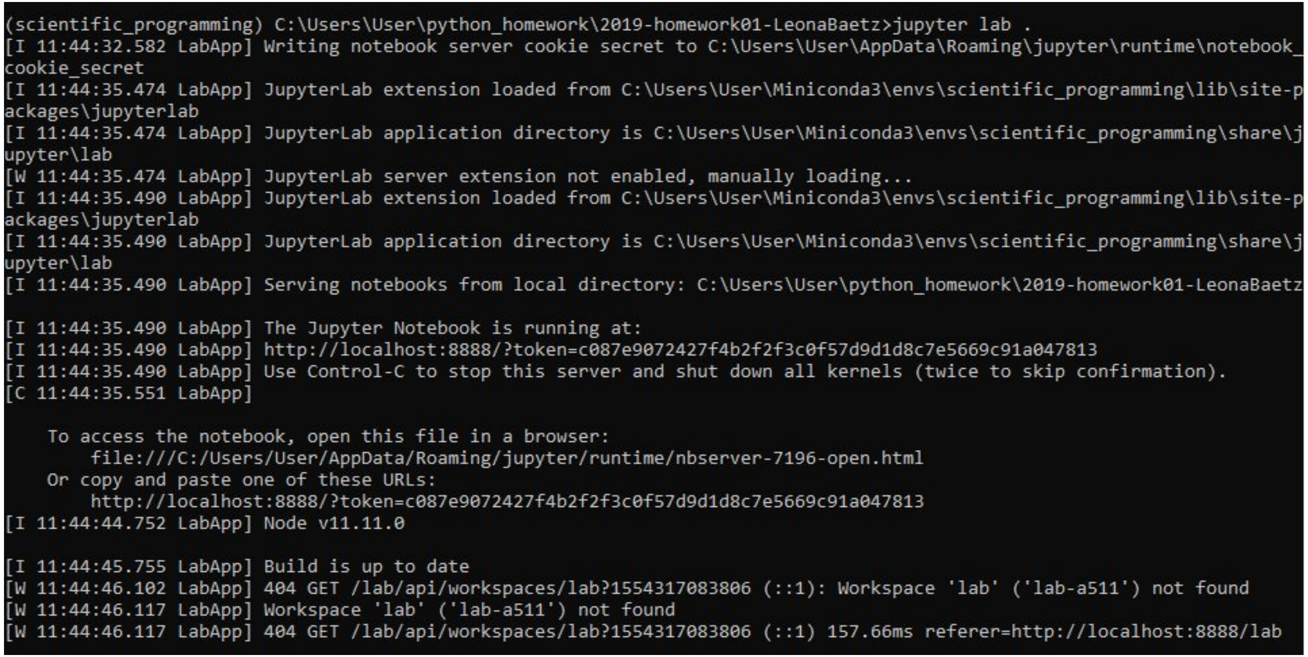
- https://www.codecademy.com/learn/learn-the-command-line



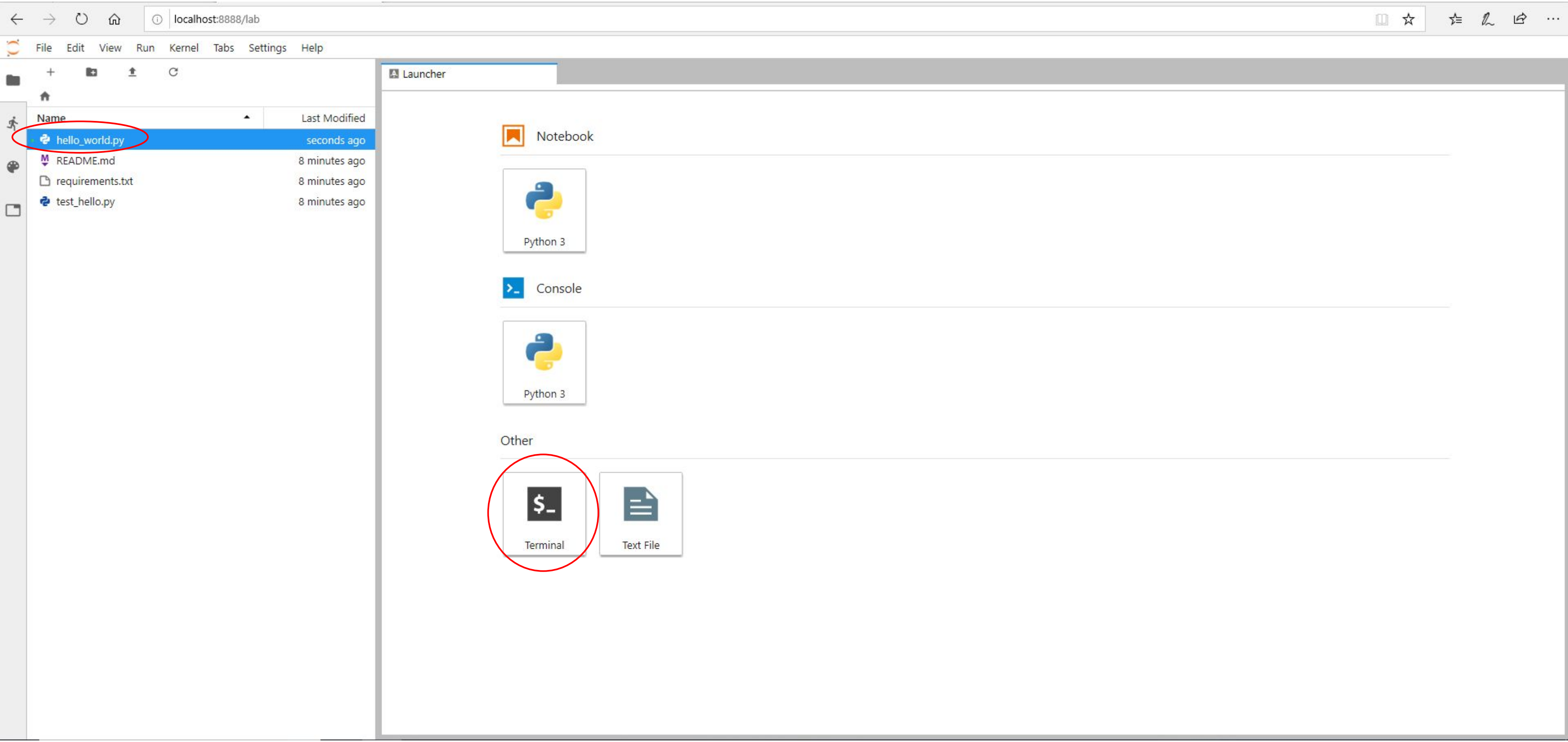
```
Microsoft Windows [Version 10.0.17763.348]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\User>mkdir python_homework && cd python_homework

C:\Users\User\python_homework>git clone https://github.com/scientificprogrammingUOS/2019-homework01-LeonaBaetz.git
Cloning into '2019-homework01-LeonaBaetz'...
remote: Enumerating objects: 25, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 25 (delta 10), reused 25 (delta 10), pack-reused 0
Unpacking objects: 100% (25/25), done.

C:\Users\User\python_homework>cd 2019-homework01-LeonaBaetz
```



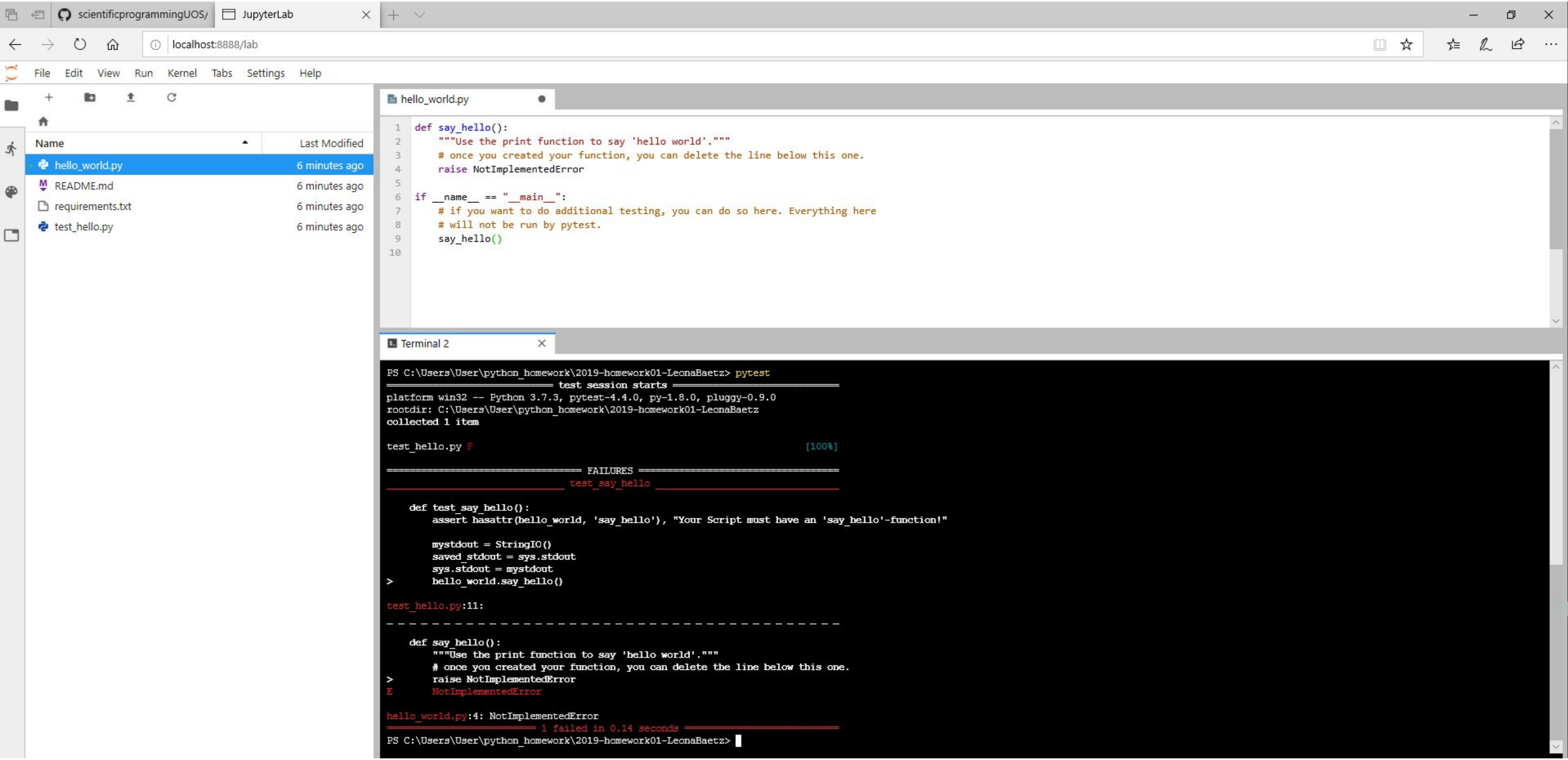
```
C:\Users\User\python_homework\2019-homework01-LeonaBaetz>dir
 Volume in drive C has no label.
 Volume Serial Number is 045F-2CF3

 Directory of C:\Users\User\python_homework\2019-homework01-LeonaBaetz

04/03/2019  11:42 AM    <DIR>          .
04/03/2019  11:42 AM    <DIR>          ..
04/03/2019  11:42 AM             1,339 .gitignore
04/03/2019  11:42 AM               138 .travis.yml
04/03/2019  11:42 AM               346 hello_world.py
04/03/2019  11:42 AM             6,937 README.md
04/03/2019  11:42 AM                15 requirements.txt
04/03/2019  11:42 AM               382 test_hello.py
               6 File(s)          9,157 bytes
               2 Dir(s)  92,013,391,872 bytes free

C:\Users\User\python_homework\2019-homework01-LeonaBaetz>conda activate scientific_programming

(scientific_programming) C:\Users\User\python_homework\2019-homework01-LeonaBaetz>pytest
================================================= test session starts =================================================
platform win32 -- Python 3.7.3, pytest-4.4.0, py-1.8.0, pluggy-0.9.0
rootdir: C:\Users\User\python_homework\2019-homework01-LeonaBaetz
collected 1 item

test_hello.py F                                                                                                  [100%]

====================================================== FAILURES =======================================================
___________________________________________________ test_say_hello ____________________________________________________

    def test_say_hello():
        assert hasattr(hello_world, 'say_hello'), "Your Script must have an 'say_hello'-function!"

        mystdout = StringIO()
        saved_stdout = sys.stdout
        sys.stdout = mystdout
>       hello_world.say_hello()

test_hello.py:11:
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

    def say_hello():
        """Use the print function to say 'hello world'."""
        # once you created your function, you can delete the line below this one.
>       raise NotImplementedError
E       NotImplementedError

hello_world.py:4: NotImplementedError
============================================== 1 failed in 0.11 seconds ===============================================

(scientific_programming) C:\Users\User\python_homework\2019-homework01-LeonaBaetz>
```



```
(scientific_programming) C:\Users\User\python_homework\2019-homework01-LeonaBaetz>jupyter lab .
[I 11:44:32.582 LabApp] Writing notebook server cookie secret to C:\Users\User\AppData\Roaming\jupyter\runtime\notebook_
cookie_secret
[I 11:44:35.474 LabApp] JupyterLab extension loaded from C:\Users\User\Miniconda3\envs\scientific_programming\lib\site-p
ackages\jupyterlab
[I 11:44:35.474 LabApp] JupyterLab application directory is C:\Users\User\Miniconda3\envs\scientific_programming\share\j
upyter\lab
[W 11:44:35.474 LabApp] JupyterLab server extension not enabled, manually loading...
[I 11:44:35.490 LabApp] JupyterLab extension loaded from C:\Users\User\Miniconda3\envs\scientific_programming\lib\site-p
ackages\jupyterlab
[I 11:44:35.490 LabApp] JupyterLab application directory is C:\Users\User\Miniconda3\envs\scientific_programming\share\j
upyter\lab
[I 11:44:35.490 LabApp] Serving notebooks from local directory: C:\Users\User\python_homework\2019-homework01-LeonaBaetz

[I 11:44:35.490 LabApp] The Jupyter Notebook is running at:
[I 11:44:35.490 LabApp] http://localhost:8888/?token=c087e9072427f4b2f2f3c0f57d9d1d8c7e5669c91a047813
[I 11:44:35.490 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 11:44:35.551 LabApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/User/AppData/Roaming/jupyter/runtime/nbserver-7196-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=c087e9072427f4b2f2f3c0f57d9d1d8c7e5669c91a047813
[I 11:44:44.752 LabApp] Node v11.11.0

[I 11:44:45.755 LabApp] Build is up to date
[W 11:44:46.102 LabApp] 404 GET /lab/api/workspaces/lab?1554317083806 (::1): Workspace 'lab' ('lab-a511') not found
[W 11:44:46.117 LabApp] Workspace 'lab' ('lab-a511') not found
[W 11:44:46.117 LabApp] 404 GET /lab/api/workspaces/lab?1554317083806 (::1) 157.66ms referer=http://localhost:8888/lab
```



Name	Last Modified
hello_world.py	seconds ago
README.md	8 minutes ago
requirements.txt	8 minutes ago
test_hello.py	8 minutes ago

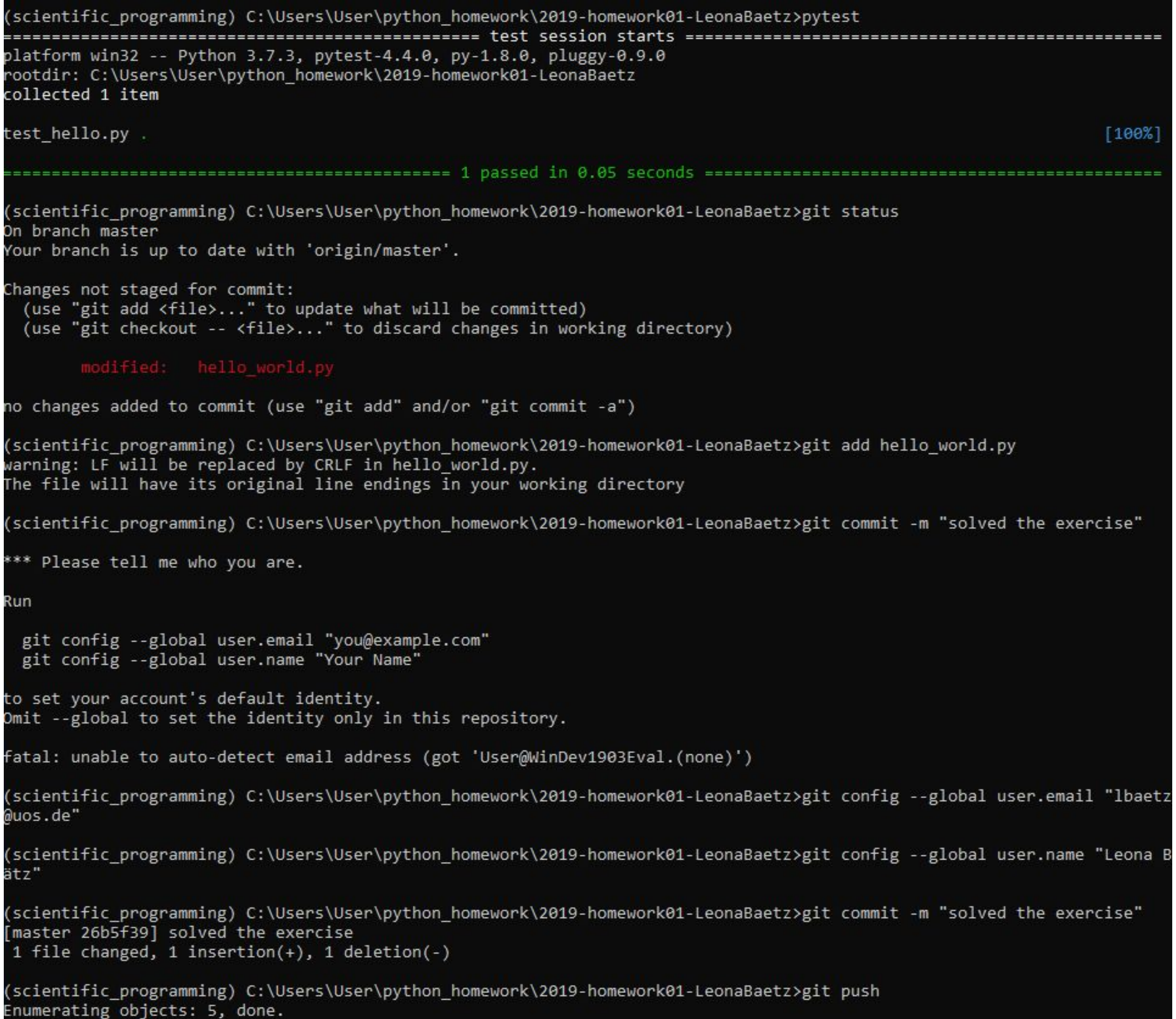


Name	Last Modified
hello_world.py	6 minutes ago
README.md	6 minutes ago
requirements.txt	6 minutes ago
test_hello.py	6 minutes ago

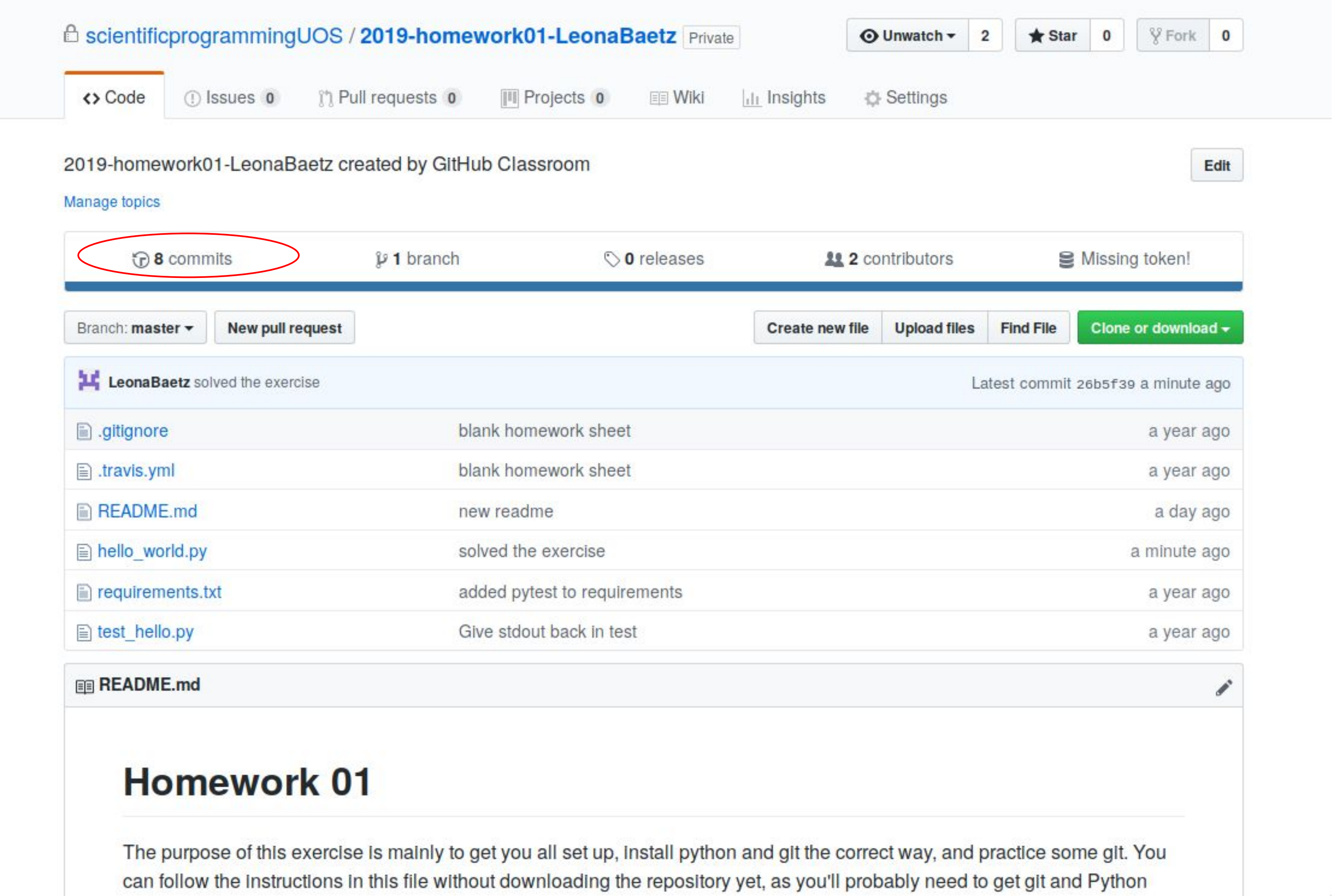
```python
def say_hello():
    """Use the print function to say 'hello world'."""
    # once you created your function, you can delete the line below this one.
    raise NotImplementedError

if __name__ == "__main__":
    # if you want to do additional testing, you can do so here. Everything here
    # will not be run by pytest.
    say_hello()

```

```
PS C:\Users\User\python_homework\2019-homework01-LeonaBaetz> pytest
========================= test session starts ==========================
platform win32 -- Python 3.7.3, pytest-4.4.0, py-1.8.0, pluggy-0.9.0
rootdir: C:\Users\User\python_homework\2019-homework01-LeonaBaetz
collected 1 item

test_hello.py F                                                  [100%]

================================ FAILURES ================================
______________________________ test_say_hello ______________________________

    def test_say_hello():
        assert hasattr(hello_world, 'say_hello'), "Your Script must have an 'say_hello'-function!"

        mystdout = StringIO()
        saved_stdout = sys.stdout
        sys.stdout = mystdout
>       hello_world.say_hello()

test_hello.py:11:
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

    def say_hello():
        """Use the print function to say 'hello world'."""
        # once you created your function, you can delete the line below this one.
>       raise NotImplementedError
E       NotImplementedError

hello_world.py:4: NotImplementedError
========================= 1 failed in 0.14 seconds =========================
PS C:\Users\User\python_homework\2019-homework01-LeonaBaetz>
```

(in between here you should solve the given task)



```
(scientific_programming) C:\Users\User\python_homework\2019-homework01-LeonaBaetz>pytest
================================================ test session starts =================================================
platform win32 -- Python 3.7.3, pytest-4.4.0, py-1.8.0, pluggy-0.9.0
rootdir: C:\Users\User\python_homework\2019-homework01-LeonaBaetz
collected 1 item

test_hello.py .                                                                                                 [100%]

============================================= 1 passed in 0.05 seconds ==============================================

(scientific_programming) C:\Users\User\python_homework\2019-homework01-LeonaBaetz>git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   hello_world.py

no changes added to commit (use "git add" and/or "git commit -a")

(scientific_programming) C:\Users\User\python_homework\2019-homework01-LeonaBaetz>git add hello_world.py
warning: LF will be replaced by CRLF in hello_world.py.
The file will have its original line endings in your working directory

(scientific_programming) C:\Users\User\python_homework\2019-homework01-LeonaBaetz>git commit -m "solved the exercise"

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'User@WinDev1903Eval.(none)')

(scientific_programming) C:\Users\User\python_homework\2019-homework01-LeonaBaetz>git config --global user.email "lbaetz
@uos.de"

(scientific_programming) C:\Users\User\python_homework\2019-homework01-LeonaBaetz>git config --global user.name "Leona B
ätz"

(scientific_programming) C:\Users\User\python_homework\2019-homework01-LeonaBaetz>git commit -m "solved the exercise"
[master 26b5f39] solved the exercise
 1 file changed, 1 insertion(+), 1 deletion(-)

(scientific_programming) C:\Users\User\python_homework\2019-homework01-LeonaBaetz>git push
Enumerating objects: 5, done.
```



2019-homework01-LeonaBaetz created by GitHub Classroom

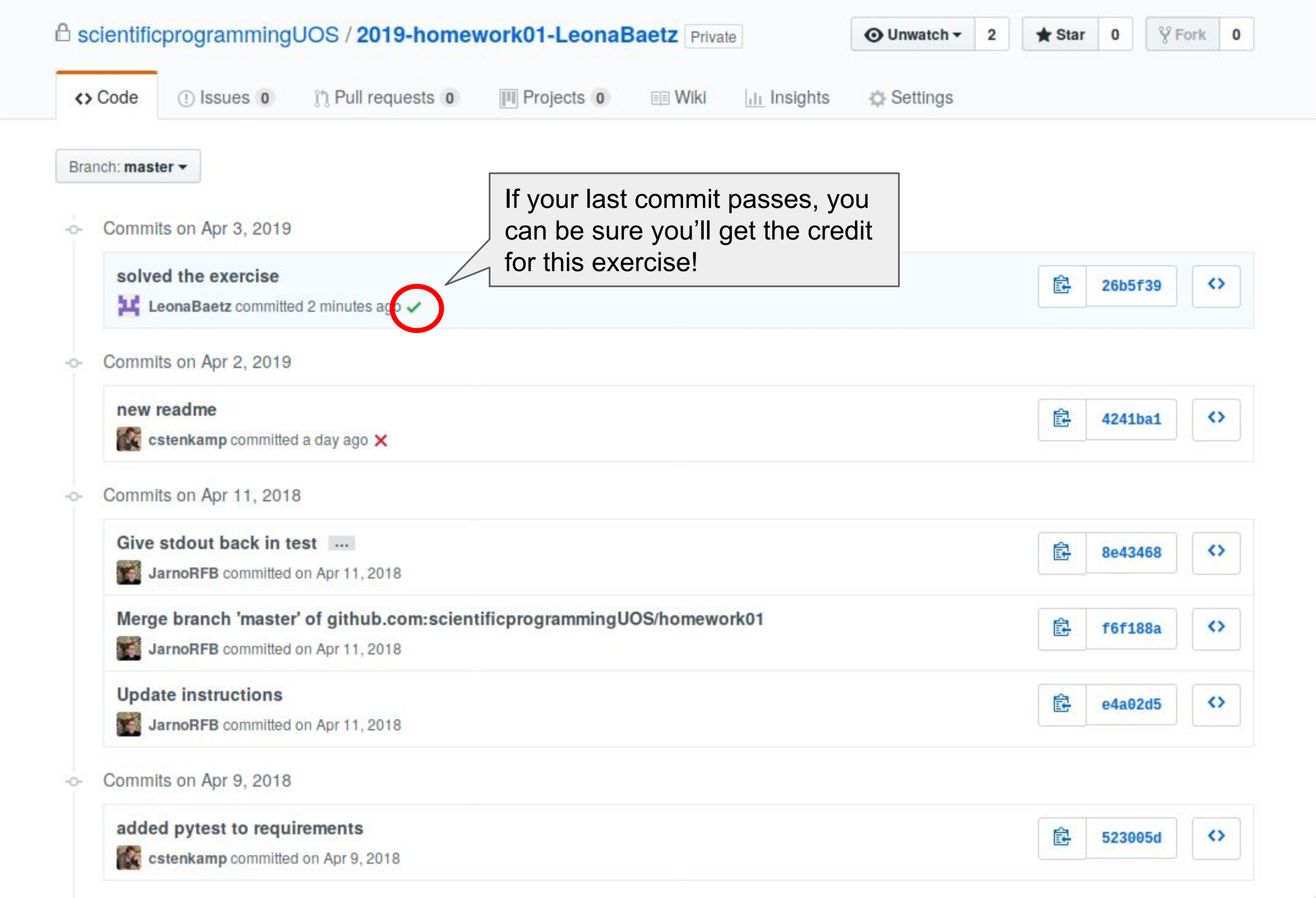
LeonaBaetz solved the exercise — Latest commit 26b5f39 a minute ago

.gitignore	blank homework sheet	a year ago
.travis.yml	blank homework sheet	a year ago
README.md	new readme	a day ago
hello_world.py	solved the exercise	a minute ago
requirements.txt	added pytest to requirements	a year ago
test_hello.py	Give stdout back in test	a year ago

README.md

Homework 01

The purpose of this exercise is mainly to get you all set up, install python and git the correct way, and practice some git. You can follow the instructions in this file without downloading the repository yet, as you'll probably need to get git and Python



Commits on Apr 3, 2019

solved the exercise
LeonaBaetz committed 2 minutes ago ✓ 26b5f39

Commits on Apr 2, 2019

new readme
cstenkamp committed a day ago ✗ 4241ba1

Commits on Apr 11, 2018

Give stdout back in test …
JarnoRFB committed on Apr 11, 2018 8e43468

Merge branch 'master' of github.com:scientificprogrammingUOS/homework01
JarnoRFB committed on Apr 11, 2018 f6f188a

Update instructions
JarnoRFB committed on Apr 11, 2018 e4a02d5

Commits on Apr 9, 2018

added pytest to requirements
cstenkamp committed on Apr 9, 2018 523005d

If your last commit passes, you can be sure you'll get the credit for this exercise!

Thanks for your attention!

- We will have have a feedback-questionnaire after 4-5 sessions
- Any questions and remarks please via email!
- Content-suggestions are always welcome!

Thanks to Patrick Faion and Brian Lewis for the old *Scientific Programming in Python* lecture
Thanks to Lukas Kalbertodt for the git lecture

Further reading

Review of scientific languages http://flow.byu.edu/posts/sci-prog-lang

Sources

1. https://commons.wikimedia.org/wiki/File:Python.svg
2. https://pixabay.com/vectors/swiss-army-knife-pocket-knife-blade-154314/
3. https://en.wikipedia.org/wiki/R_(programming_language)
4. https://commons.wikimedia.org/wiki/File:Matlab_Logo.png
5. https://commons.wikimedia.org/wiki/File:Images_200px-ISO_C%2B%2B_Logo_svg.png
6. https://pt.wikipedia.org/wiki/Julia_(linguagem_de_programa%C3%A7%C3%A3o)
7. https://commons.wikimedia.org/wiki/File:Git_icon.svg
8. https://farm2.staticflickr.com/1482/24588096069_59a0513790_z.jpg