

Python

Automatisierung, Web-Scraping,
Bildbearbeitung

1. Termin

- Web-Scraping, Datenbereinigung und Datenaufbereitung
 - Identifizieren und Auswählen relevanter Elemente im HTML-Quelltext
 - Web-Scraping (z.B. mit BeautifulSoup)
 - Daten bereinigen und aufbereiten

2. Termin

- Daten analysieren, Daten in andere Dateiformate speichern, Daten präsentieren
 - Grundlegende Techniken der Datenanalyse mit Python
 - Speichern von Daten in unterschiedlichen Dateiformaten
 - Erstellen von Diagrammen und Visualisierungen
 - Erstellen von interaktiven Dashboards (z.B. mit Dash oder Flask)

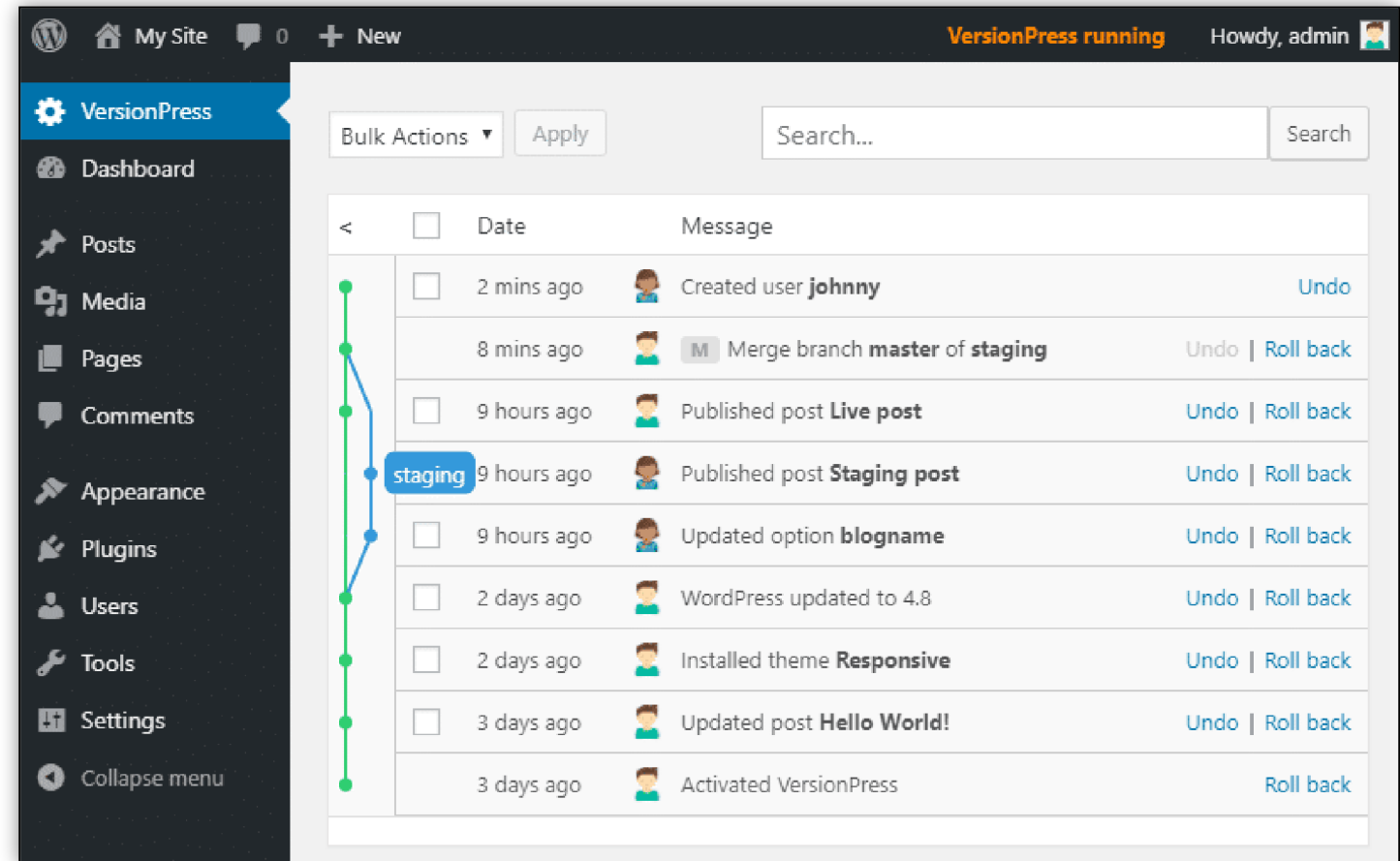
3. Termin

- Bildbearbeitung und Automatisierung mit Python
 - Einführung in die Bildverarbeitung mit Python (z.B. mit OpenCV)
 - Einfache Manipulation von Bildern
 - Python verwenden, um Aufgaben und Workflows zu automatisieren

Vorbereitung

- Um erfolgreich zu sein, braucht es ein wenig Vorbereitung
 - Python installiert
 - Eine Python IDE (Pycharm Community/Edu oder VSCode)
 - Git installiert (<https://www.git-scm.com/downloads>)*
 - Rechte, um Pakete mit PIP zu installieren

Git



Git [git] ist eine freie Software zur verteilten Versionsverwaltung von Dateien, die durch Linus Torvalds initiiert wurde.

Warum Git

Neben Git wäre auch der Austausch via ZIP-File jeweils möglich (und wird auch angeboten).

Dies bietet aber den Nachteil, dass bei jeder Kursanpassung / mehr Inhalt ein neues Paket heruntergeladen werden muss + Daten abgeglichen werden müssen.

=> Dies ist in diesem Kurs nicht schlimm, da es Tageweise aufgebaut ist, aber warum nicht Git eine Chance geben?

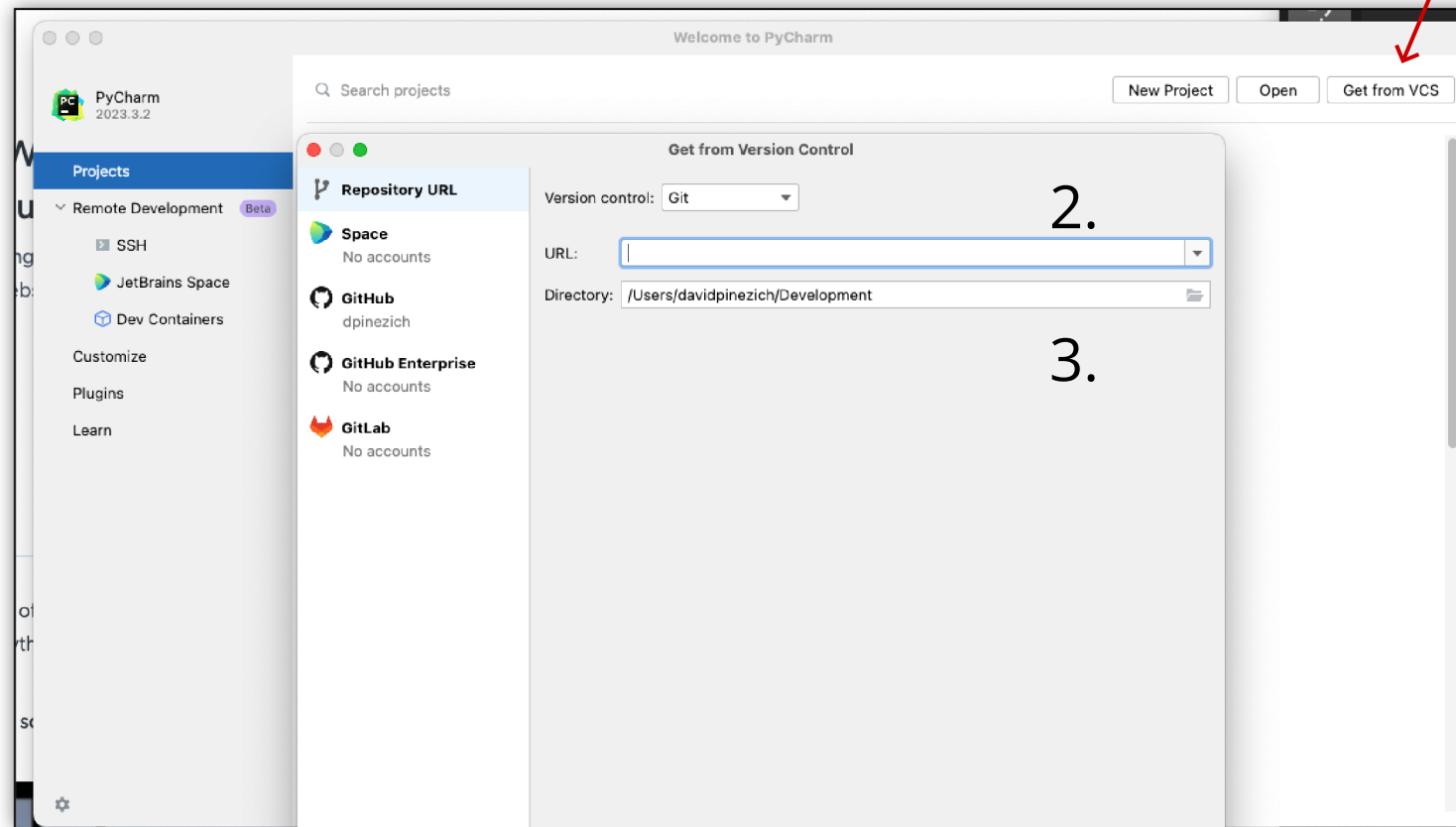
Wie verwenden wir Git?

Obwohl es bei Git um kollaboratives Arbeiten geht, verwenden wir es einseitig.

Ich stelle die Materialien zur Verfügung, ihr könnt diese einfach via **git pull** beziehen und aktualisieren.

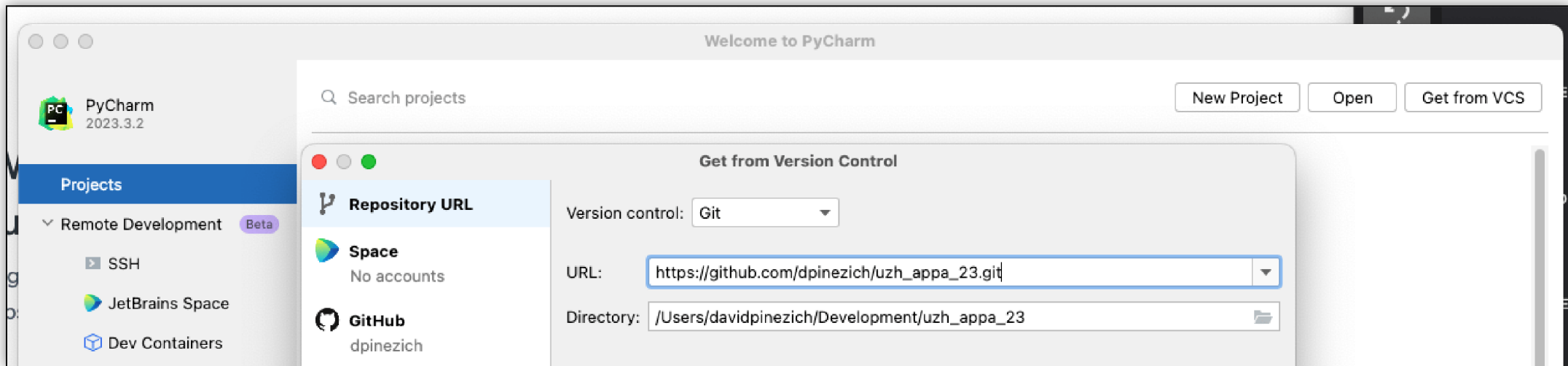
Git verwendet dabei einen Delta-Algorithmus, das heisst es ist wichtig, dass wir nicht im gleichen Ordner arbeiten – dazu aber später mehr.

Git in Pycharm



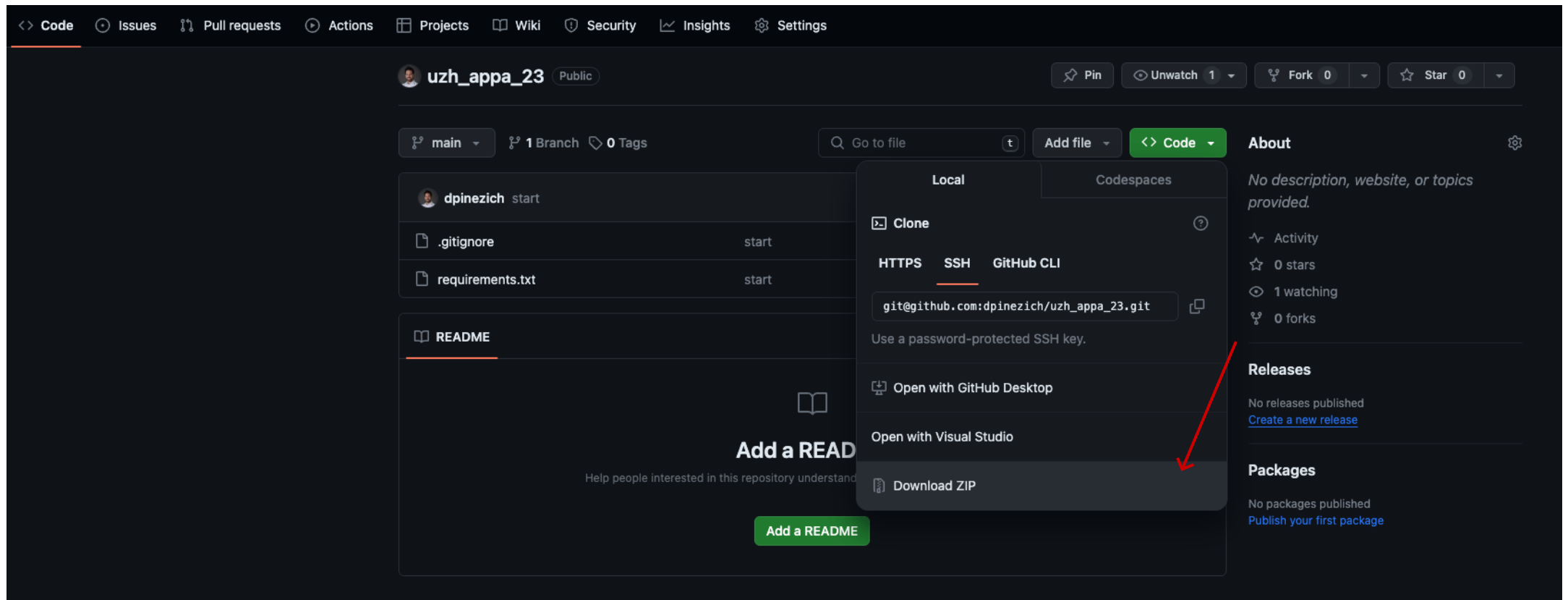
Git in Pycharm

https://github.com/dpinezich/uzh_appa_23.git



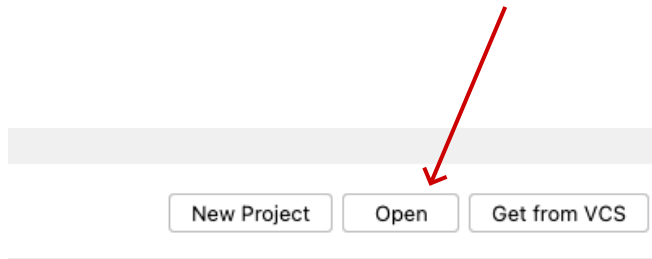
ZIP

https://github.com/dpinezich/uzh_appa_23

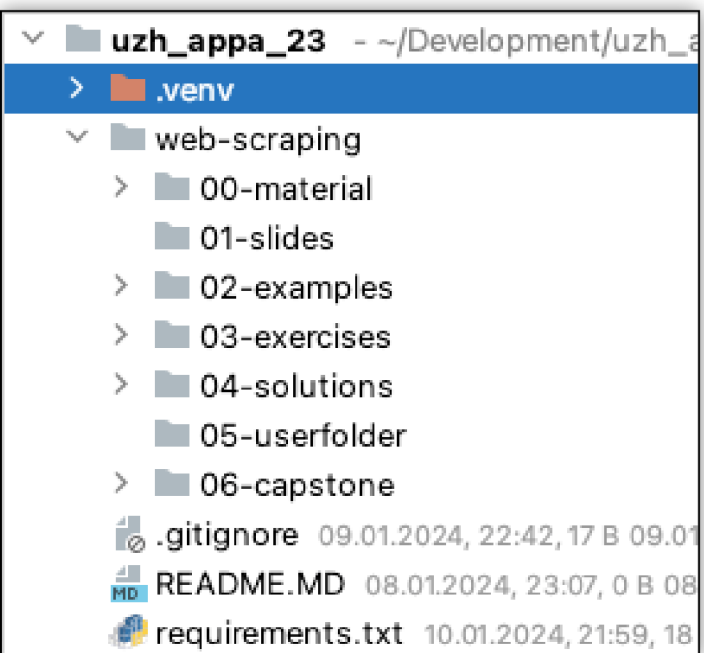


ZIP

Zip auf dem Desktop entpacken (bitte nicht in Downloads)



Typische Struktur



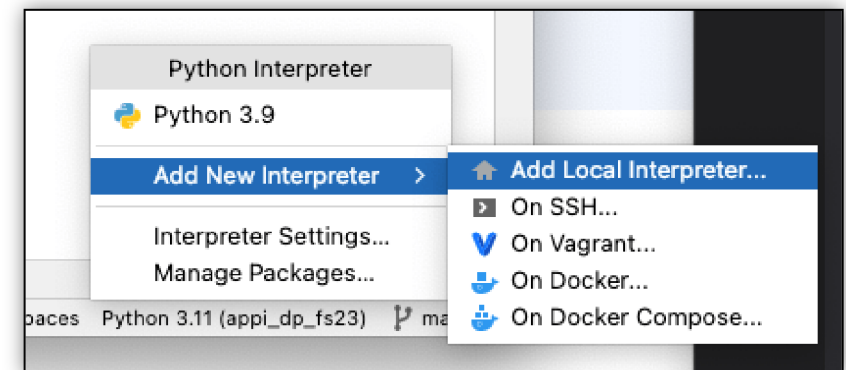
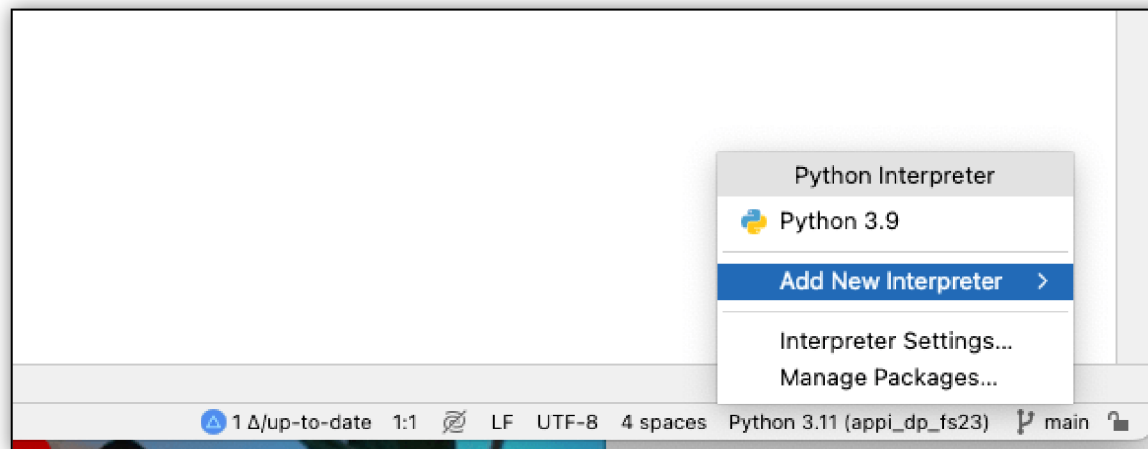
- 00-material: wird für zusätzliches Material, Datensätze, Spickzettel verwendet
- 01-slides: enthält alle gezeigten und zusätzlichen Folien
- 02-examples: enthält fertige Beispiele
- 03-exercises: enthält Beispiele enthalten, die einen Ausgangspunkt für Übungen darstellen
- 04-solutions: enthält alle Lösungen zu den in den Folien gestellten Aufgaben
enthält auch Lösungen zu nicht beendeten 03-exercises
- 05-userfolder: IHR **Spielplatz**, um Ihre **Lösungen**, **Versuche**, **Zweitmeinungen**, etc. zu speichern :-)
- 06-capstone: Ein etwas aufwändigeres, reales Abschlussprojekt

Checkpoint

Wir haben nun sowohl mit Git als auch mit ZIP denselben Checkpoint erreicht.

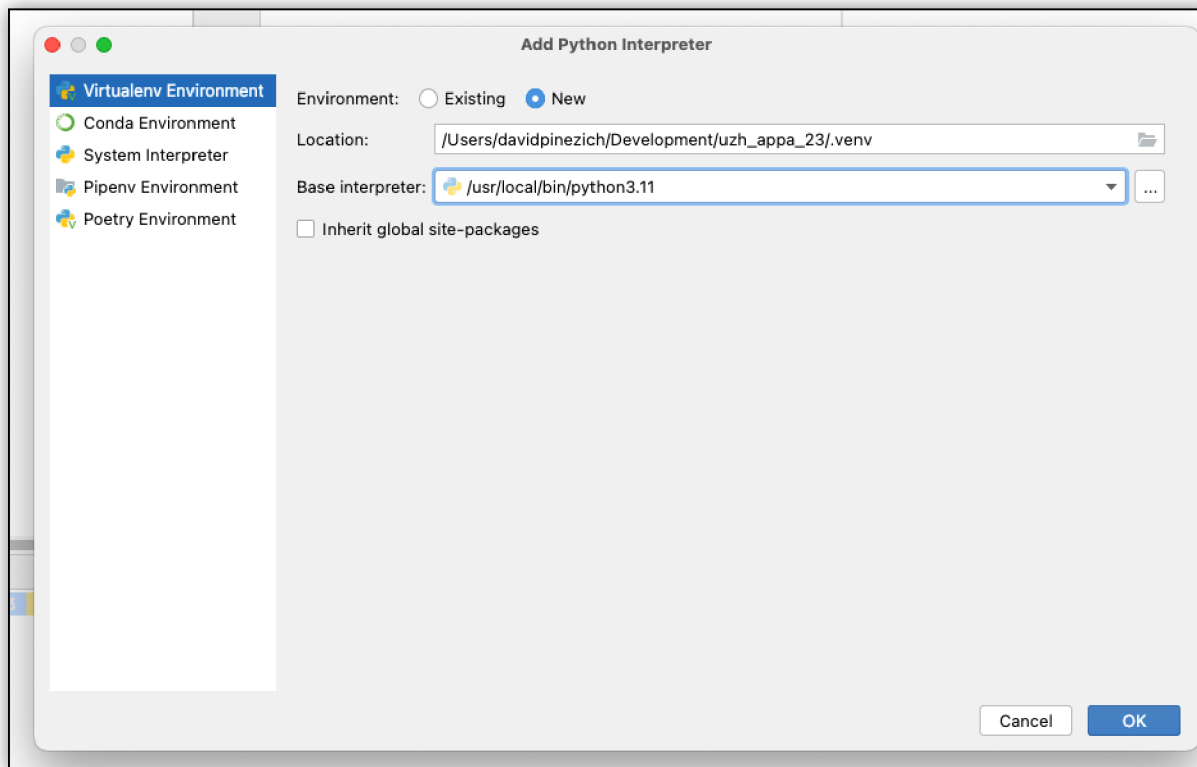
Als Nächstes brauchen wir ein virtuelles Environment, damit wir Pakete ohne schlechtes Gewissen installieren können.

Kleiner Tipp: Richtig viele Details zu VENV gibts im APPI Kurs :-)



VENV

Python ab 3.8 ist in Ordnung, 3.11/.12+ ist zu empfehlen.



inherit global site-packages bitte nicht auswählen, wir wollen einen "frischen" Start.



Pakete installieren

Pakete installieren können wir auf drei Arten:

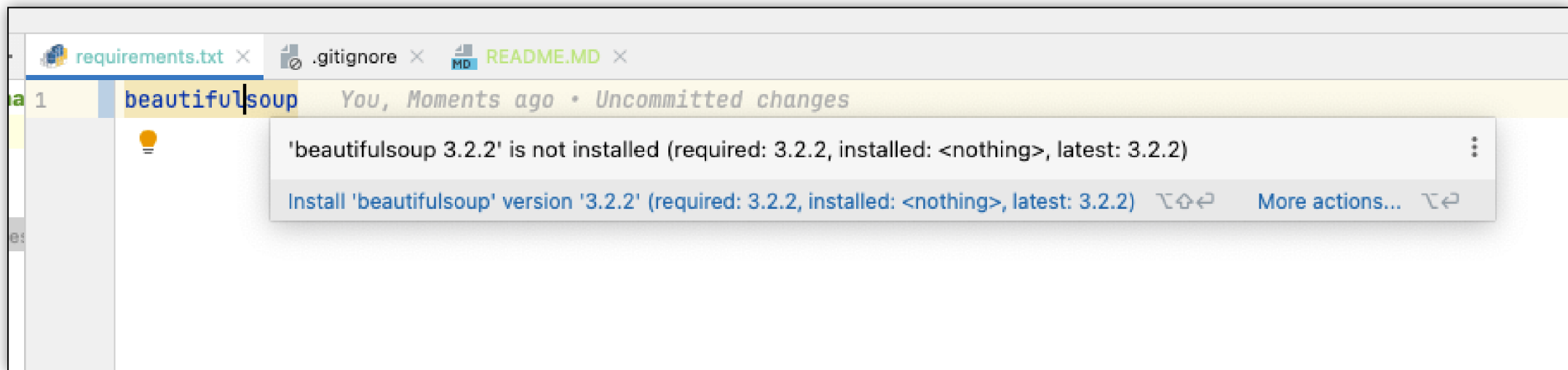
- PyCharm Konsole
- PyCharm Auswahl
- Requirements.txt

Bitte dafür kurz PyCharm neu starten

requirements.txt

Eine einfache Art, die Installation vorzugeben:

Doppelklick auf requirements.txt:



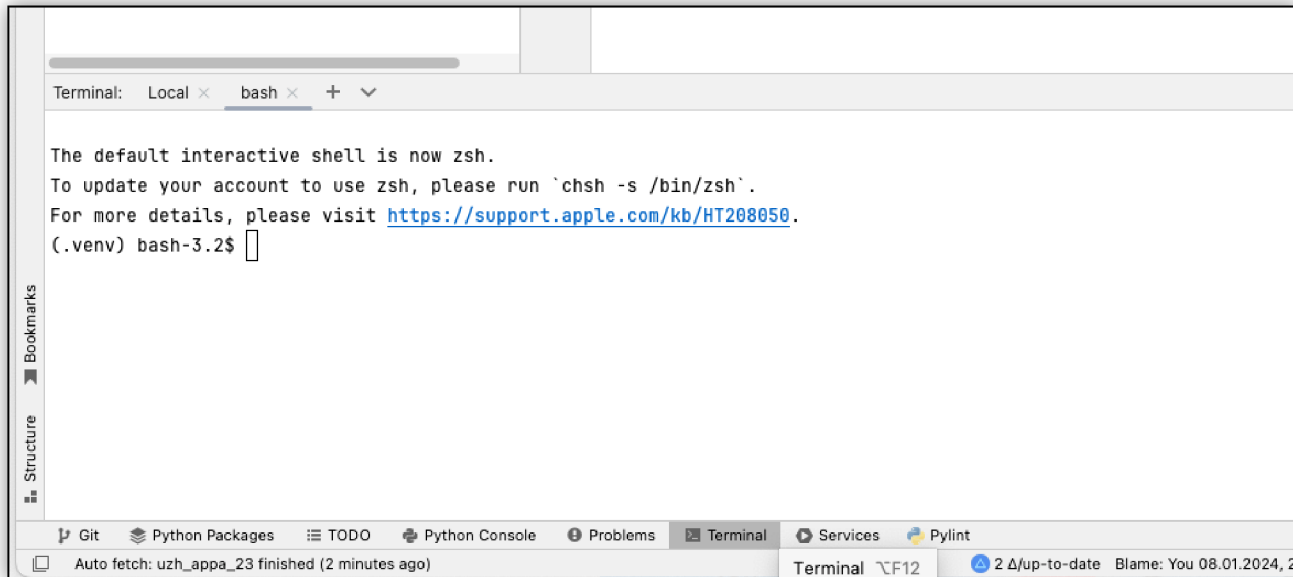
Oder



PyCharm Konsole

Eine Konsole wird mit PyCharm (und VSCode) mitgeliefert.

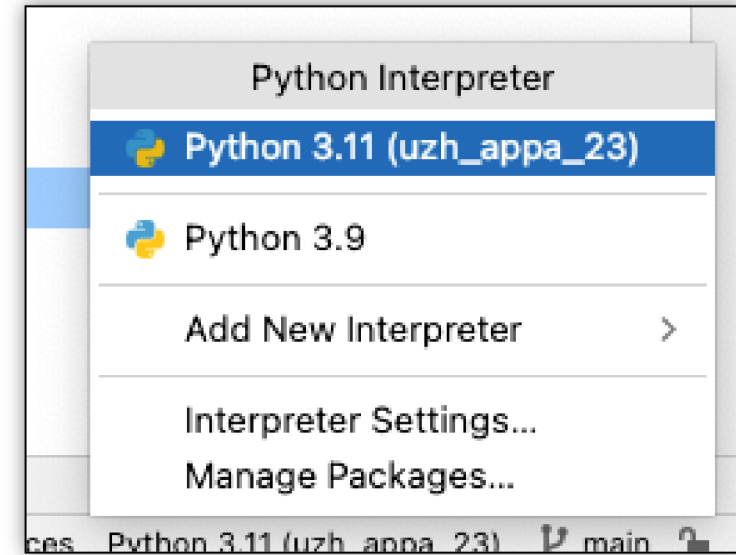
Mit dem Neustart wurde auch unser .venv erkannt (siehe (.venv)).



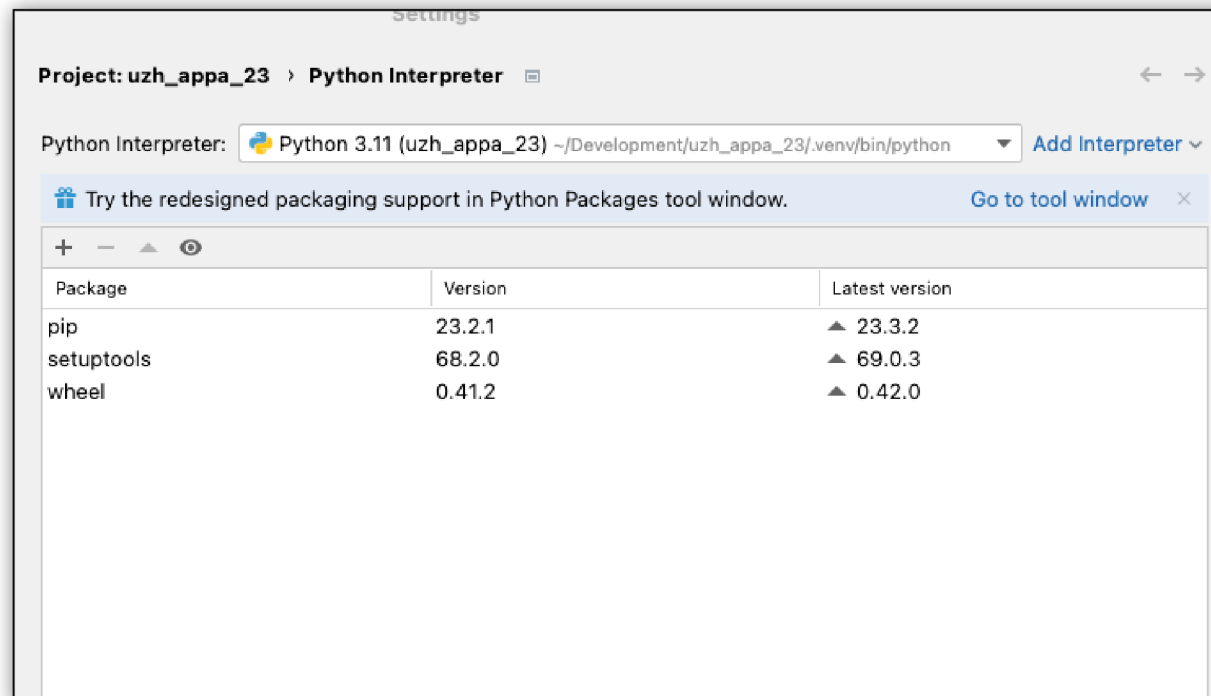
`pip install "paketname"`

PyCharm Auswahl

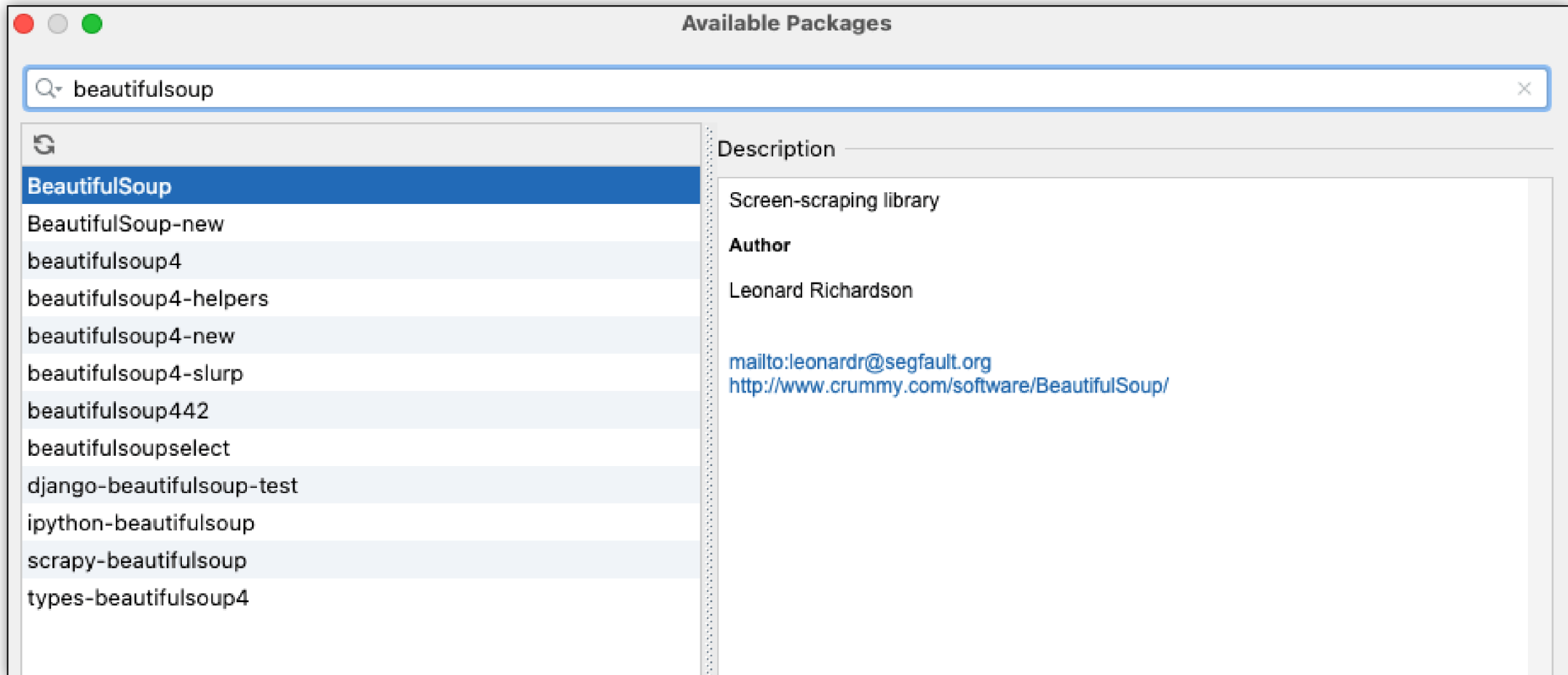
1) Klick auf Interpreter Settings



2) Plus klicken

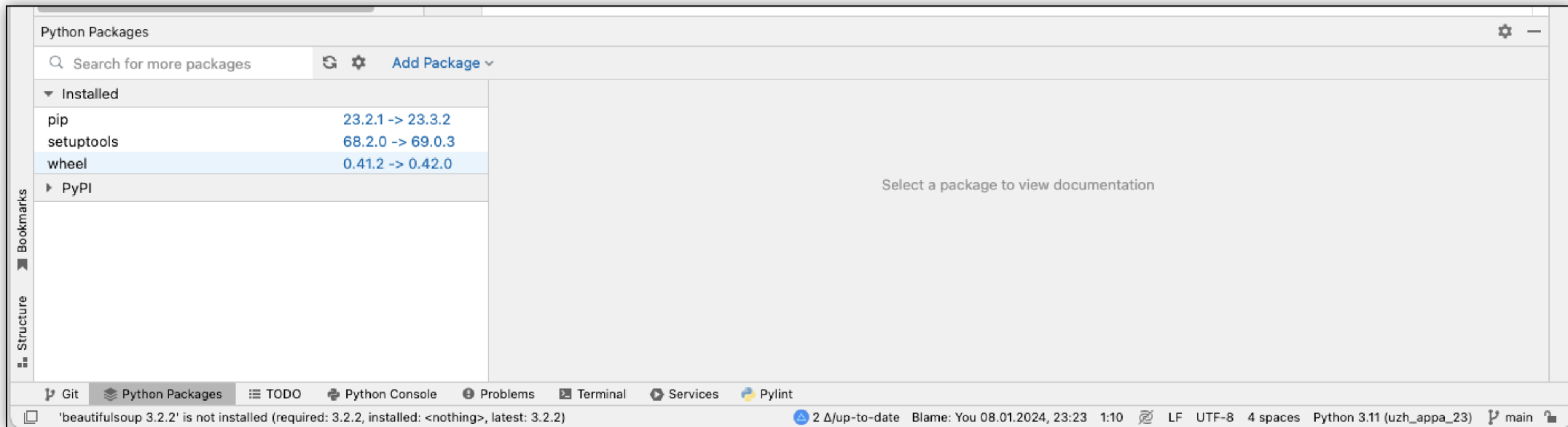


PyCharm Auswahl



PyCharm Auswahl (Modern)

Tool Window



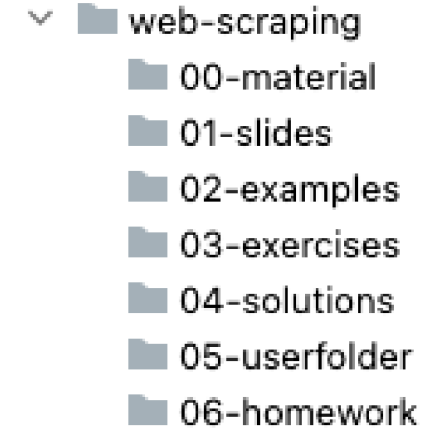
Übung - Paket installieren

1) Ein neue Datei in 05-userfolder erstellen

`test_sample.py`

2) Inhalt:

```
1 # content of test_sample.py
2 def inc(x):
3     return x + 1
4
5
6 def test_answer():
7     assert inc(3) == 5
```



A file explorer window showing a directory structure. The root directory is 'web-scraping', which is expanded. It contains several subdirectories: '00-material', '01-slides', '02-examples', '03-exercises', '04-solutions', '05-userfolder', and '06-homework'.

- web-scraping
 - 00-material
 - 01-slides
 - 02-examples
 - 03-exercises
 - 04-solutions
 - 05-userfolder
 - 06-homework

Übung - Paket installieren

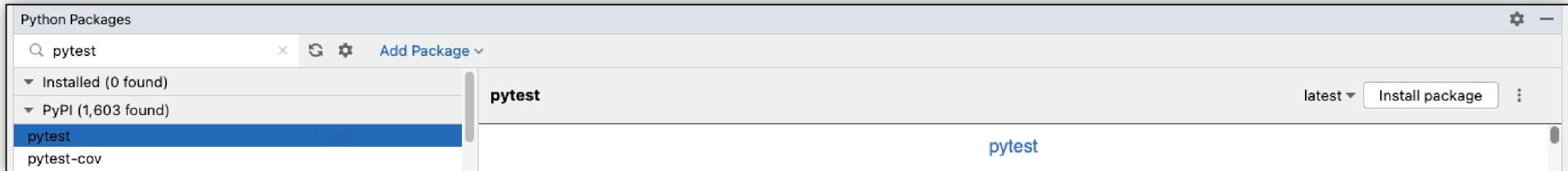
3) pytest installieren (egal auf welche Art)



4)

Übung - Paket installieren

3) pytest installieren (egal auf welche Art)

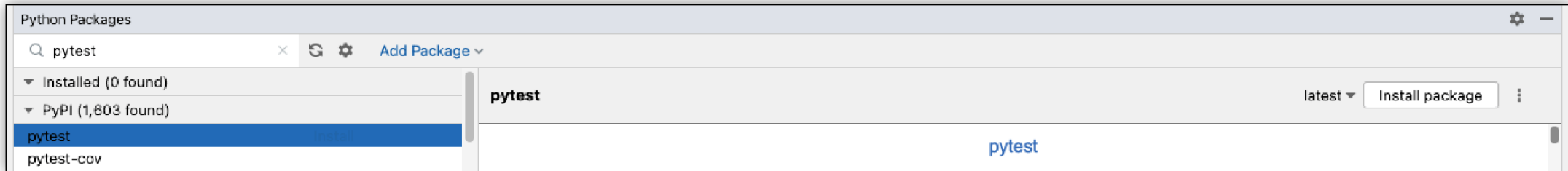


4)

```
1  # content of test_sample.py
2  usage new *
3  def inc(x):
4      return x + 1
5
6  new *
7  def test_answer():
8      assert inc(3) == 5
9
10 You, 3 minutes ago • Uncommitted changes
```


Übung - Paket installieren

3) pytest installieren (egal auf welche Art)



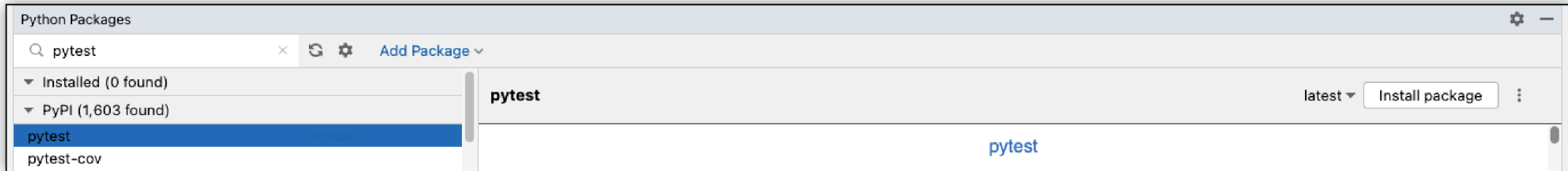
4)

```
1  # content of test_sample.py
2  usage new *
3  def inc(x):
4      return x + 1
5
6  new *
7  def test_answer():
8      assert inc(3) == 5
9
10 You, 3 minutes ago • Uncommitted changes
```



Übung - Paket installieren

3) pytest installieren (egal auf welche Art)



4)

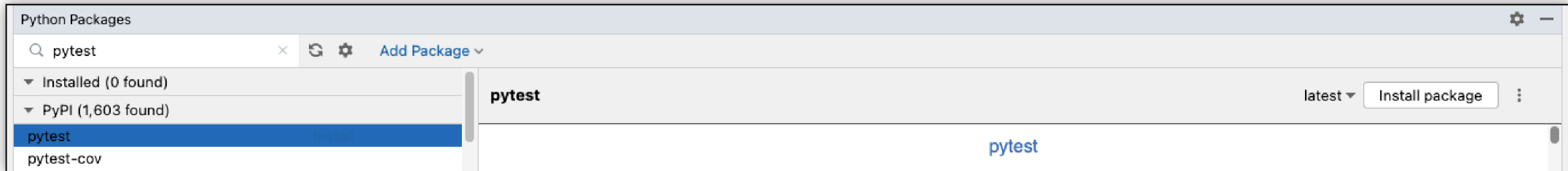
```
1 # content of test_sample.py
2 1 usage new *
3 def inc(x):
4     return x + 1
5
6 new *
7 def test_answer():
8     assert inc(3) == 5
9
10 You, 3 minutes ago • Uncommitted changes
```



```
1 # content of test_sample.py
2 1 usage new *
3 def inc(x):
4     return x + 1
5
6 new *
7 def test_answer():
8     assert inc(3) == 5
9
10 You, 3 minutes ago • Uncommitted changes
```

Übung - Paket installieren

3) pytest installieren (egal auf welche Art)



4)

```
1 # content of test_sample.py
2 1 usage new *
3 def inc(x):
4     return x + 1
5
6 new *
7 def test_answer():
8     assert inc(3) == 5
9
10 You, 3 minutes ago • Uncommitted changes
```



```
1 # content of test_sample.py
2 1 usage new *
3 def inc(x):
4     return x + 1
5
6 new *
7 def test_answer():
8     assert inc(3) == 5
9
10 You, 3 minutes ago • Uncommitted changes
```

Bereit durchzustarten?

