



# Computational Thinking, WS2023/24

## Praktische Übung 1: Wege der Python Programmierung

Prof. Dr.-Ing. Martin Hobelsberger

Dr. Benedikt Zönnchen

Prof. Dr.-Ing. Benedikt Dietrich

### Ziele des Praktikums

In dieser Übung richten Sie Ihr System ein, sodass Sie in der Programmiersprache *Python* mithilfe

- der Kommandozeile,
- einer Datei und
- eines *Jupyter-Notebooks*

programmieren können und so zukünftige Abgaben erledigen können.

### Aufgabe 1.1: Python installieren

Zunächst müssen wir sicherstellen, dass Sie *Python* auf Ihrem System installiert haben.

**Hinweise** zur Installation finden Sie beispielsweise hier.

- Überprüfen Sie, ob auf Ihrem System *Python* installiert ist.
- Lassen Sie sich die Version von *Python* auf Ihrer Kommandozeile ausgeben, falls *Python* installiert ist.
- Installieren Sie `python 3.11` durch die Installation von *Python*.

**Achtung:** Achten Sie beim Installieren darauf, dass Sie Python zur PATH-Umgebungsvariablen Ihres Betriebssystems hinzufügen, sodass Ihr System *Python* und *PIP* findet. Unter Windows ist hierfür gleich beim ersten Dialog des Installers ein Haken zu setzen.

- d) Lassen Sie sich die Version von *Python* auf Ihrer Kommandozeile ausgeben und überprüfen Sie diese.

```
python --version
```

- e) Prüfen Sie auf die selbe Art ob *PIP* richtig installiert wurde.

```
pip --version
```

### Aufgabe 1.2: Starten des Python-Interpreters

Starten Sie den sog. *Python*-Interpreter und berechnen Sie wie viele Sekunden innerhalb von 5 Tagen verstreichen. Spielen Sie mit dem Interpreter herum. Probieren Sie verschiedene Eingaben aus. Sie können den Interpreter mit den Tastenkürzeln `Strg + D` bzw. `ctrl + D` beenden. **Hinweise** finden Sie z.B. hier.

### Aufgabe 1.3: Python-Skript ausführen

- 1) Im *Moodlekurs* finden Sie eine Datei `script.py`. Speichern Sie sich die Datei in einem Ordner, welchen Sie auf Ihrem System finden.
- 2) Starten Sie Ihr Kommandozeilenprogramm (Konsole, Shell, Terminal) und bewegen Sie sich in den Ordner indem sich die Datei `script.py` nun befindet.
- 3) Führen Sie die Datei aus.

**Hinweise** finden Sie z. B. hier.

### Aufgabe 1.4: Python-Skript schreiben I

- 1) Öffnen Sie einen Texteditor Ihrer Wahl (z. B. TextEdit, Notepad, Notepad++)
- 2) Schreiben Sie folgenden Text hinein:

```
print("Hallo Welt!")
```

- 3) Speichern Sie die Datei unter dem Namen `hello.py` ab. (**Achten Sie darauf, dass Sie reinen Text und keine Formatierung abspeichern**)
- 4) Führen Sie die Datei, d.h. Ihr *Python*-Skript über die Kommandozeile aus.
- 5) Verändern Sie den Inhalt der Datei, Experimentieren Sie herum und führen Sie Ihr Skript immer wieder aus.

### Aufgabe 1.5: Python-Skript schreiben II

Öffnen Sie die zuvor heruntergeladene Datei `script.py` mit ihrem Texteditor. Nutzen Sie den *Python*-Code der Datei um ein Programm zu schreiben, was Ihnen eine Zahl  $n$  über die Kommandozeile einliest und die Anzahl der verstrichenen Sekunden innerhalb von  $n$  Tagen ausgibt.

### Aufgabe 1.6: Jupyter-Notebooks (lokal) installieren

Verwenden Sie den Paketmanager PIP um die *Jupyter-Notebook-Umgebung* zu installieren. Der Kommandozeilenbefehl lautet:

```
pip install jupyterlab
```

Sie müssen die Installation möglicherweise mit `y` bestätigen. **Hinweise** finden Sie z. B. hier.

### Aufgabe 1.7: Jupyter-Notebooks lokal starten

Starten Sie nun mit dem Kommandozeilenbefehl

```
jupyter lab
```

das sog. *Jupyter-Lab* in Ihrem *Browser*.

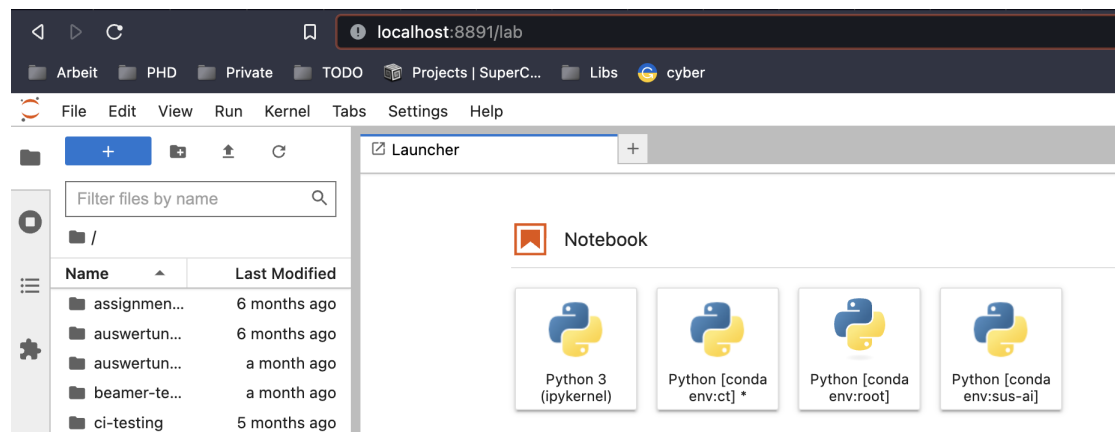


Abbildung 1: So sollte das etwa in Ihrem Browser aussehen.

Erstellen Sie ein neues *Notebook*.

### Aufgabe 1.8: Mit dem Notebook arbeiten

- Erstellen / starten Sie ein *Notebook* (falls noch nicht geschehen).
- Experimentieren Sie darin herum.
- Finden Sie heraus wie Sie eine sog. Zelle ausführen können.
- Finden Sie heraus wie Sie neue Zellen erstellen können.
- Finden Sie heraus wie Sie Text, der nicht ausgeführt werden kann, einfügen können.

**Hinweise** finden Sie z. B. hier.

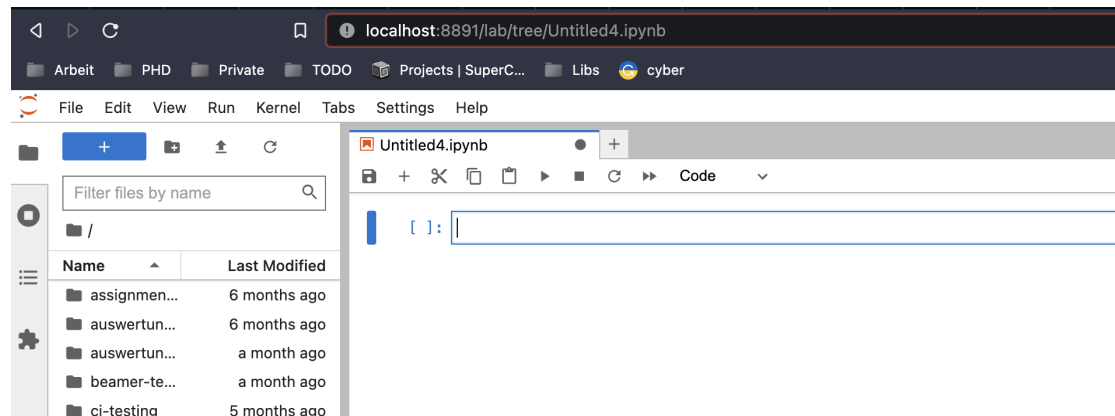


Abbildung 2: So sollte das etwa in Ihrem Browser aussehen.

### Aufgabe 1.9: Besuchen Sie unseren Datahub

Sie können Ihre *Notebooks* auch auf unserem Server <https://datahub.cs.hm.edu/> erstellen, verwalten, hochladen, herunterladen und ausführen. Gehen Sie auf diese Webseite und erstellen Sie dort probeweise ein *Notebook*.

### Aufgabe 1.10: Aufgabe auf den Datahub laden

Gehen Sie in den *Moodlekurs* der Vorlesung. Dort finden Sie einen Link, der den Download eines Aufgabeblasses (im Notebookformat) aktiviert. Klicken Sie auf diesen Link. Sie sollten zu unserem *Datahub* weitergeleitet werden und dort sollte sich ein neues, befülltes *Notebook* öffnen. Mit diesem Schritt haben Sie das Notebook in den Datahub importiert. Sie können das Notebook dort bearbeiten, speichern und auch herunterladen. Außerdem können Sie das Notebook mit dem *Visual Studio Code Server* öffnen.

### Aufgabe 1.11: Der Datahub und Visual Studio Code

Ein Nachteil des Jupyter Labs ist die umständliche Bedienung des Debuggers. Wir empfehlen Ihnen daher, die Notebooks mit Hilfe von *Visual Studio Code (VS Code)* zu bearbeiten. VS Code ist eine schlanke, aber sehr gut erweiterbare und vielseitige Entwicklungsumgebung. Sie können VS Code entweder lokal installieren oder Sie nutzen einen sog. *Visual Studio Code (VS Code) Server* der auf unserem *Datahub* läuft. Das bedeutet, Sie können auch mit einer abgespeckten Variante von VS Code auf unserem Server/*Datahub* arbeiten.

Um VS Code auf dem Server zu starten, öffnen Sie den Datahub (<https://datahub.cs.hm.edu/>) und wählen *File -> New Launcher* aus. Starten Sie VS Code, indem Sie auf den entsprechenden Eintrag im Launcher klicken. Wechseln Sie in VS Code in die Exploreransicht und öffnen Sie in das Notebook `01_arbeitsumgebung.ipynb`.

- a) Lösen Sie die Aufgabe des *Notebooks* auf dem *Datahub*.

- b) Führen Sie alle Zellen von oben nach unten aus.
- c) Dabei sollte eine neue *Zip*-Datei entstehen (Ihre Lösung)
- d) Geben Sie Ihre Lösung (die *Zip*-Datei) in Moodle ab.

### Aufgabe 1.12: Notebooks in Visual Studio Code (lokal, optional)

Sie können auch *VSC* lokal auf Ihrem Rechner benutzen. Dazu müssen Sie zunächst *VSC* selbst installieren (siehe <https://code.visualstudio.com/>). Haben Sie *VSC* erfolgreich installiert, benötigen Sie noch die folgenden Erweiterungen:

- Python (Identifier: `ms-python.python`)
- Jupyter (Identifier: `ms-toolsai.jupyter`)

Versuchen Sie ein *Notebook* in *VSC* auszuführen.

**Hinweis:** Wenn Sie die von uns bereitgestellten Notebooks auf Ihrem System ausführen wollen, müssen Sie ein *Python*-Paket namens *otter-grader* installieren. Installieren Sie es durch den Kommandozeilenbefehl

```
pip install otter-grader
```