

Vorbereitung: Python-Setup und Testlauf

Medien und Informatik verstehen

Richard Conrardy

1 Ziel

Vor der Präsenzsitzung richten Sie eine lauffähige Python-Umgebung ein und führen ein erstes Programm aus. So können wir die Präsenzzeit für das Programmieren nutzen und nicht für individuelle Installationsprobleme.

Erledigen Sie die Vorbereitung vor der Sitzung vom **11. November 2026**.

Dieser Aufbau basiert auf dem Material von Peter Giger¹.

2 Drei Teile der Umgebung

Sie benötigen:

1. **Python** – den Interpreter, der den Code ausführt;
2. **eine Entwicklungsumgebung (IDE)** – zum Schreiben und Ausführen des Codes;
3. **eine Python-Erweiterung bzw. einen ausgewählten Interpreter** – damit die IDE weiss, welches Python verwendet werden soll.

2.1 Empfohlener Weg: Positron

Installieren Sie **Positron**. Positron ist eine kostenlose Entwicklungsumgebung für Python und R. Falls auf dem Computer noch keine passende Python-Installation vorhanden ist, kann Positron Python beim Einrichten installieren.

Öffnen Sie danach in Positron einen eigenen Ordner für diesen Kurs, zum Beispiel `python-mi`. Über **File** → **Open Folder** (oder **Datei** → **Ordner öffnen**) wählen Sie den Ordner aus. Positron verwendet diesen Ordner als Arbeitsbereich; dort können Sie Python-Dateien speichern und später wiederfinden. Eine Einführung in Projekte und Ordner finden Sie im **offiziellen Python-Einstieg für Positron**.

Wählen Sie anschliessend einen Python-Interpreter aus. Der genaue Menüname kann je nach Version leicht abweichen; suchen Sie bei Bedarf nach **Interpreter** oder **Python interpreter**. Wenn Positron fragt, ob eine Python-Umgebung erstellt oder installiert werden soll, bestätigen Sie dies.

Falls Sie kein Desktopbetriebssystem besitzen oder keine Software installieren können, können Sie Python ersatzweise im Browser über **WebTigerPython** ausführen.

2.2 Alternative: Visual Studio Code

Installieren Sie **Visual Studio Code** und danach die **Microsoft-Python-Erweiterung**. Python selbst muss separat installiert sein. Eine aktuelle Anleitung finden Sie bei **Python in Visual Studio Code**.

Wählen Sie in VS Code über die Befehlspalette **Python: Select Interpreter** einen installierten Python-Interpreter aus.

3 Erster Testlauf

3.1 Datei anlegen

Erstellen Sie im geöffneten Kursordner eine neue Datei mit dem Namen `python_check.py`. Die Endung `.py` ist wichtig. Speichern Sie die Datei im geöffneten Ordner, nicht irgendwo im temporären Download- oder Desktop-Ordner.

¹Repository «Programmieren» von Peter Giger, Lizenz CC BY 4.0.

3.2 Code einfügen

```
name = "Python"  
zahl = 2 + 3  
  
print("Hallo", name)  
print("2 + 3 =", zahl)
```

3.3 Datei ausführen

Starten Sie die Datei mit der Ausführen-Schaltfläche Ihrer IDE. Falls eine Auswahl erscheint, wählen Sie den zuvor eingerichteten Python-Interpreter. Positron führt den Code anschliessend im unteren Bereich im **Terminal** bzw. in der **Konsole** aus. Dort sollte folgende Ausgabe erscheinen:

```
Hallo Python  
2 + 3 = 5
```

4 Wenn der Test nicht funktioniert

Gehen Sie diese Punkte der Reihe nach durch:

1. Ist die Datei wirklich als `python_check.py` gespeichert und nicht als `python_check.py.txt`?
2. Ist ein Python-Interpreter ausgewählt?
3. Haben Sie die Datei über die Python-Ausführen-Funktion der IDE gestartet?
4. Starten Sie die IDE neu und wählen Sie den Interpreter erneut.
5. Kopieren Sie die vollständige Fehlermeldung. Verändern Sie nicht nur einzelne Wörter davon.

Sie dürfen die Fehlermeldung mit einer KI-Hilfe (z.B. ChatGPT oder Copilot) analysieren lassen. Geben Sie dabei immer den vollständigen Fehler, den verwendeten Code und die Frage an, was Sie bereits versucht haben. Übernehmen Sie vorgeschlagene Änderungen nicht blind, sondern prüfen Sie, ob Sie sie verstehen.

Wenn es weiterhin nicht funktioniert, melden Sie sich **unbedingt vor der Lektion** bei mir und schicken Sie die vollständige Fehlermeldung sowie einen Screenshot oder eine kurze Beschreibung. Arbeiten Sie nicht mehrere Stunden allein an der Installation.