

Übung 1 – Python Crash-Course

Lernziele: In dieser Übung lernen wir, wie man:

- übliche Programmierungs-Anweisungen mit Python schreibt
- einfache Probleme mit Python löst.

Die Programmierungs-Aufgaben sind absichtlich ohne ausführliche Schritten beschrieben, damit Sie selbst die Möglichkeiten entdecken und überlegen müssen.

Hinweis: Vergessen Sie nicht, am Anfang der heutigen Folien sind 2 ausführliche Python-Tutorials vorgeschlagen.

Aufgabe 1 – Python installieren

Installieren Sie Python, die aktuellste Version 3.9.x.

Beginnen Sie hier: <https://www.python.org/downloads/>

Achtung bei Linux Ein paar extra Schritten müssen durchgeführt werden, bevor Sie Python-Quellcode-Package kompilieren:

- Behebung zum Problem mit SSL, folgen Sie die Anleitung hier:
(SSL wird bei pip, oder Package installieren, benötigt.)
<https://stackoverflow.com/questions/49094768/ssl-module-in-python-is-not-a-vailable>
Passen Sie die `SSL`-Variable in der `Setup`-Datei nach dem Pfad in Ihrem Computer an.
 - Behebung zum Problem mit sqlite3, folgen Sie die Anleitung hier:
<https://stackoverflow.com/questions/51012032/no-module-named-sqlite3-when-running-djangos-python3-manage-py-migrate>
-

Optionale Aufgabe 2 – PyCharm installieren

(Sie sind frei, eine IDE (oder gar keine) selber aussuchen. Ich empfehle PyCharm weil ich mit der vertraut bin und bei der Professional Version gibt es extra Funktionalität für Django.)

Installieren Sie PyCharm Professional Version, die aktuellste Version 2021.1. Mit Ihrer Beuth-Mail-Adresse bekommen Sie kostenlos eine Lizenz für die Professional Version.

Beginnen Sie hier: <https://www.jetbrains.com/pycharm/download/>

Aufgabe 3 – Python Grundlagen mittels Fragen lernen

Auf dieser Webseite: <https://www.w3schools.com/python/exercise.asp> ist eine Serie von Fragen mit enthüllbaren Lösungen.

Gehen Sie die Fragen durch und erraten Sie, bevor Sie die Lösungen enthüllen.

Aufgabe 4 – Modul implementieren und testen

Aufgabe 4a: Schreiben Sie ein Modul mit den folgenden Merkmalen:

- Modul-Name: `kreis_mathe`.
- Eine Funktion:
 - Funktions-Name: `flaeche_berechnen`.
 - Eingabe: Radius eines Kreises.
 - Rückgabe: Fläche nach gegebenem Radius.
- Eine Funktion:
 - Funktions-Name: `umfang_berechnen`.
 - Eingabe: Radius eines Kreises.
 - Rückgabe: Umfang nach gegebenem Radius.

Aufgabe 4b: Testen Sie das Modul, damit Sie beide Funktionen aufrufen und die Eingaben sowie die Rückgaben in der Console drucken.

Aufgabe 5 – Klasse implementieren und testen

Aufgabe 5a: Schreiben Sie eine Klasse – selbstverständlich auf Python – mit den folgenden Merkmalen:

- Klassen-Name: `Student`.
- Attribute:
 - `vorname`.
 - `nachname`.
 - `matrikelnummer`.
 - `geburtsdatum`.
 - Eine "Tabelle", die die Kurse und die entsprechenden Noten speichern.

Hinweis: Welcher Datentyp wäre zu welchem Attribut geeignet?

- Die `__repr__()`-Methode.
- Die `__str__()`-Methode.
- Eine Methode:

- Methoden-Name: `note_eintragen`.
- 1. Eingabe: `kurs`.
- 2. Eingabe: `note` für diesen `kurs`.
- Rückgabe: keine.
- Die Methode speichert die angegebenen Informationen.
- Eine Methode:
 - Methoden-Name: `noten_berichten`.
 - Eingabe: keine.
 - Rückgabe: keine.
 - Die Methode druckt in der Console alle Kurse und die entsprechenden Noten.
- Eine Methode:
 - Methoden-Name: `noten_schummeln`.
 - Eingabe: keine.
 - Die Methode betrachtet die Noten und alle Noten, die nicht 1,0 sind, werden zu 1,0 gemacht.
 - Rückgabe: Die Anzahl von Noten, die verändert sind.

Aufgabe 5b: Testen Sie die Klasse.

- Erzeugen Sie ein `Student`-Objekt.
- Mit dem Objekt tragen Sie ein paar Kurse mit unterschiedlichen Noten ein.
- Drucken Sie das Objekt per `__repr__()`- und `__str__()`-Methode in der Console.
- Schummeln Sie die Noten und berichten Sie die Anzahl von veränderten Noten in der Console.
- Berichten Sie die Noten nochmal in der Console.

Abgabe

Laden Sie im Moodle die folgenden Dateien hoch:

- Alle Quellcode-Dateien, die Sie geschrieben haben.
- Text-Dateien, die die Ergebnisse der Ausführungen Ihrer Quellcodes in der Console zeigt.