

“Hello World!”

Hausübung 0

October 10, 2018

## Einleitung

Im Rahmen der Übung dieser Lehrveranstaltung werden Sie mit der Programmiersprache Python arbeiten. Die erste Aufgabenstellung widmet sich der Installation des Interpreters und dem Einrichten einer Arbeitsumgebung.

Die klassische erste Aufgabe beim Lernen einer neuen Programmiersprache ist ein Programm zu schreiben, welches simple Textausgabe auf dem Bildschirm anzeigt. Daher eignet Sie sich bestens, um ein System auf korrekte Installation zu prüfen. Die Aufgabe der Studierenden ist es also, Python mithilfe des Anaconda Frameworks einzurichten und ein minimales Skript zu schreiben, welches die berühmten Worte “Hello World!” in die Kommandozeile ausgeben.

# 1 Tasks

## 1.1 Installation von Python

Bevor Sie zu anfangen können zu programmieren, muss eine Python Umgebung eingerichtet werden. Als Referenz für diese Lehrveranstaltung gilt die Version 3.7 (erlaubt ist alles ab Version 3.5). Um den Installationsprozess (vor allem unter Windows) zu erleichtern, empfiehlt sich, das Anaconda Framework<sup>1</sup> zu benutzen. Dieses liefert die meisten notwendigen Erweiterungen (packages) mit und kümmert sich um die korrekte Installation.

Ein kurzer Test, ob Python erfolgreich am System installiert ist, kann mit der Kommandozeile durchgeführt werden. Beim Aufruf von `python` sollte der Interpreter starten und einen Text ähnlich dem folgenden ausgeben:

```
>python
Python 3.7.0 (default, Jun 28 2018, 08:04:48) [MSC v.1912 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```



Es ist nicht vorgeschrieben, das Anaconda Framework zu verwenden, aber Aufgrund der Vielfalt an Installationsmöglichkeiten und Distributionen wird es keine garantierte Hilfestellung von Seiten der LV Leitung oder der Tutoren geben.

## 1.2 Ihr erstes Programm

Wie Eingangs erwähnt, ist es Tradition, als erste Übung ein “Hello World”-Programm zu schreiben. Die Studierenden Aufgabe ist es, ein Programm mit folgender Ausgabe zu schreiben:

```
Hello World!
```



Wenn dies auf Ihrem System ausführbar ist, kann man davon ausgehen, dass die Installation erfolgreich war. Zusätzlich kommen Sie erstmals mit der Grundfunktionalität von Python in berührung. Überlegen Sie sich den aller einfachsten, und saubersten Weg, wie Sie diese Aufgabenstellung lösen können. (*Hinweis:* Sie sollten nicht mehr als 2-3 Zeilen Code außerhalb des Kommentar-Headers haben)

# 2 Beschränkungen

- Fügen Sie keine Nutzlosen komponenten Ihrer Abgabe hinzu
- Sofern möglich, verwenden Sie eingebaute funktionen
- Importieren Sie keine extra packages (keine `import` Statement)
- Verwenden Sie keine Kommandozeilenparameter

---

<sup>1</sup><https://www.continuum.io/downloads>

### 3 Datei Header

All Ihre Quelldateien in Ihrer Abgabe müssen gleich zu Beginn einen Kommentar mit folgenden Informationen enthalten:

- Author: – Ihr Name
- MatNr: – Ihre Matrikelnummer
- Description: – Generelle Beschreibung der Datei
- Comments: – Kommentare, Erklärungen, usw

Bitte einfach den folgenden Code in Ihre Dateien kopieren und den Inhalt anpassen. Je nach PDF-Reader müssen Sie eventuell die Leerzeichen/Einrückungen per Hand anpassen.

**Beispielheader:**

```
#####  
# Author:      [FIRST NAME] [LAST NAME]  
# MatNr:       [MATR NR]  
# Description: [SHORTDESCRIPTION]  
# Comments:    [ANY RELEVANT COMMENTS.  
#              CAN BE MULTILINE]  
#####
```



### 4 Coding Standard

Für diese Lehrveranstaltung orientieren Sie sich am offiziellen PEP 8 Standard<sup>2</sup>. Dieser Beschreibt grundsätzliche Formalitäten im Bezug auf Ihren Code. Folgendes ist besonders zu Beachten:

**Sprache.** Code schreibt man in Englisch. Im internationalen Zeitalter ist es notwendig, dass auch jemand am anderen Ende der Welt verstehen kann, was Sie programmiert haben. Ihr gesamter Quellcode muss daher auf Englisch geschrieben sein. Dies betrifft sowohl die Kommentare also auch Variablenamen und Ähnliches.

**Leerzeichen statt Tabulatoren.** Python basiert auf Einrückungen, anstatt auf geschwungenen Klammern. Theoretisch gibt es die Möglichkeit, Leerzeichen (spaces) oder Tabulatoren (tabs) zu verwenden. PEP8 schreibt aber klar 4 Leerzeichen als Einrückungen vor. Die meisten Python Programmierumgebungen werden automatisch 4 Leerzeichen einfügen, wenn Sie auf die Tabulator-Taste drücken.

**Sprechende Namen.** Verwenden Sie kurze, aber sprechende Namen für Ihre Variablen, Funktionen, (und Ähnliches). Es muss eindeutig aus dem Namen hervorgehen, was die Aufgabe des Elements ist. Für simple Iterationen kann ein einfaches *i* ausreichend sein, aber dies kann schnell zu Chaos führen. Sollten Sie Variablen haben, die keine Aufgabe haben und nicht verwendet werden, schreibt PEP 8 vor, einen einfachen Unterstrich (  ) zu verwenden. Sollten Sie sich unsicher sein, dann beschreiben Sie ihre Variablen (und Namen) in einem Codekommentar.

**120 Zeichen Zeilenlänge.** PEP8 sieht Zeilenlängen von maximal 79 Zeichen vor. Ein anderes, etwas großzügigeres Limit, welches sich in der Szene etabliert hat, sieht eine Länge von 120 Zeichen vor. In dieser LV verwenden wir daher das erweiterte Limit. Bitte achten Sie darauf, dass keine Zeile in Ihrem Code diese Länge von 120 Zeichen überschreitet (gilt auch für Kommentare). *Hinweis:* Zeilenlänge inkludiert die Einrückungen mittels Leerzeichen!

---

<sup>2</sup><https://www.python.org/dev/peps/pep-0008/>

## 5 Automatisierte Tests

Ihr Programm wird mit dem folgenden Befehl ausgeführt:

```
>python assignment_0.py
```



Vergewissern Sie sich, dass Ihre Abgabe allen Beschränkungen, die in dieser Angabe erwähnt sind, konform ist. Zusätzlich wird jede Abgabe auch von einem Mitglied des Tutorenteams begutachtet. Ihre Tutoren führen auch die finale Bewertung (Punkte) durch.

## 6 Abgabe

### 6.1 Deadline

16. Oktober 2018 um 23:59:59.

Eine Spätabgabe ist nicht vorgesehen. Ausnahme bilden hier Notfälle. Sollte das Abgabesystem nicht online sein, verlängert sich die Deadline automatisch um 24 Stunden. Sollten Sie, aus diversen Gründen, nicht in der Lage sein, Ihre Abgabe hochzuladen, kontaktieren Sie Ihren Tutor *VOR* der Deadline. (*Hinweis:* Urlaub oder Ähnliches wird nicht als Grund akzeptiert!)

### 6.2 Hochladen der Abgaben

Assignments werden stets als Archive abgegeben. Erlaubt sind hier die Formate *.zip*, und *.tar.gz*. Zusätzlich zu ihren Quelldateien, soll Ihre Abgabe auf eine Datei namens *readme.txt* beinhalten. Das Vorhandensein der Datei ist Pflicht, ihr Inhalt aber optional. Sie soll folgenden Inhalt haben: (i) Die Zeit, die Sie benötigt haben, um die Aufgabenstellung zu absolvieren. (ii) Feedback, wo Sie Probleme hatten. Ihr Feedback ermöglicht es, verbreitete Probleme zu erkennen und die Vorlesung und Tutoriumseinheiten entsprechend anzupassen.

Abgaben erfolgen auf der Palme Website. Bitte prüfen Sie vor der Abgabe diese Kriterien:

- Datei- und Ordnerstruktur (siehe unten)
- Kommentarheader in jeder Quelldatei
- Coding Standard

### 6.3 Struktur der Abgabe

```
└─ assignment_0.zip (or assignment_0.tar.gz)
   └─ assignment_0.py
      └─ readme.txt
```