

# Programmierung mit Python für Programmierer

Kapitel 1 - Einführung und Datentypen

Autor: Dr. Christian Heckler

# Vorbemerkungen

- Verwendete Literatur: siehe Referenzen
- Verwendete Symbole:
  - **!**: Beispielprogramm
  - **i**: Weitere Erläuterungen im Kurs
  - **?**: Übung

# Vorraussetzungen und Ziele

- Voraussetzungen:
  - Erfahrungen in einer höheren Programmiersprache
  - Die folgenden Konzepte werden als bekannt vorausgesetzt
    - Bezeichner / Variablen
    - Datentypen: Zahlen, Zeichenketten (Strings), Kollektionen
    - Kontrollstrukturen: Verzweigungen, Schleifen
    - Prozeduren, Funktionen, Module
    - Objektorientierung (?)
- Ziel: Kompakte Einführung in die Grundlagen, so dass die Erarbeitung der Details und Module im Selbststudium möglich ist, z.B. über die genannten Bücher und die Dokumentation (<https://docs.python.org>)

# Kursinhalt

- Variablen und „Laufzeitmodell“
- Eingebaute Datentypen
- Kontrollstrukturen
- Funktionen, Module und Pakete
- Objektorientierung

# Eigenschaften von Python

- Interpretierte Skriptsprache
- Dynamisch getypt
- Verschiedene Programmierparadigmen (prozedural, OO, funktional, ...)
- Für alle gängigen Plattformen verfügbar (Windows, Unix, Mac)
- Insbesondere für Anfänger geeignet
- Umfangreiche Standardbibliothek
- Entwickelt von Guido van Rossum am CWI Amsterdam (Centrum Wiskunde & Informatica ) ab 1989
- Weite Verbreitung (s. diverse Programmiersprachenranglisten)

# Versionen von Python

- Es gibt verschiedene Versionen von Python
- Aktuell:
  - Python 2.x
  - Python 3.x
- Python 2 und Python 3 sind nicht (hundertprozentig) kompatibel
- Im Kurs wird Python 3 verwendet
- Der Interpreter für python 3 kann „python3“ (Linux, Apple) heißen.
- Die Version erhält man über den Aufruf

```
python -V bzw. python3 -V
```

# Python-Interpreter

- Der Python-Interpreter kann ein Python-Programm ausführen, das in einer Textdatei (mit der Endung .py) abgespeichert ist.
- Es gibt aber auch einen interaktiven Modus, in dem Python-Anweisungen ausgeführt werden können, ohne ein Programm zu schreiben.

# Python-Programme

- Erstellung mit beliebigem Texteditor
- Zeichencodierung UTF-8: wird von Python standardmäßig angenommen – alternativ erste Zeile im Quelltext mit der tatsächlichen Codierung

```
# -*- coding: utf-8 -*-
```

- Einheitliche Einrückung (Tabs oder Leerzeichen)
- Z.B.: Geany ([www.geany.org](http://www.geany.org)), thonny ([thonny.org](http://thonny.org)), Visual Studio Code
- Es gibt auch IDEs, z.B. PyCharme, Eclipse-Plugin, IDLE, ...
- Programmausführung
  - aus dem Editor / Entwicklungsumgebung (wenn unterstützt)
  - Doppelklick, falls Dateiendung „.py“ mit Python-Interpr. verk.
  - Kommandozeile: `python prg.py` bzw. `python3 prg.py`
  - Unix: Shebang: `#!/usr/bin/env python3`



# Interaktiver Modus

- Der Interpreter kann auch im interaktiven Modus aufgerufen werden:
  - Kommandozeile: `python` (bzw. `python3` unter Unix)
  - Über das Windows-Startmenu
  - Evtl. aus der Entwicklungsumgebung
- Der Interpreter meldet sich mit dem „Prompt“: `>>>`
- Jetzt können Anweisungen eingegeben werden, die mit der Return-Taste abgeschlossen werden, z.B.:

```
>>> print("Hallo")  
Hallo
```

# Referenzen

- [Ste] Ralph Steyer: Programmierung Grundlagen, Herdt-Verlag
- [Kle] Bernd Klein: Einführung in Python 3, Hanser-Verlag
- [Kof] Kofler: Python – Der Grundkurs, Rheinwerk Computing
- [EK] Johannes Ernesti, Peter Kaiser: Python 3 – Das umfassende Handbuch, Rheinwerk Computing
- [Mat] Eric Matthes: Python Crashkurs – Eine praktische, projektbasierte Programmier Einführung, dpunkt.verlag
- [Swe] Sweigart: Eigene Spiele programmieren: Python lernen