

# E

## Einleitung

### E.1 Was ist Python?

Python ist eine portable, interpretative, objektorientierte Programmiersprache. Ihre Entwicklung wurde 1989 von Guido van Rossum am Centrum voor Wiskunde en Informatica (CWI) in Amsterdam begonnen und wird nun durch die nichtkommerzielle Organisation »Python Software Foundation« (PSF, <http://www.python.org/psf>) koordiniert. Der Name soll an die britische Comedy-Gruppe Monty Python erinnern.

Ein Python-Programm – man bezeichnet es als Skript – ist ein Text, der von einem Interpreter ausgeführt werden kann. Weil Python-Skripte auf verschiedenen Systemplattformen (Unix, Windows, Mac OS) laufen können, bezeichnet man die Sprache als portabel. Die aktuelle Version von Python, auf die sich dieses Buch bezieht, ist Version 3.8. Sie kann von der Python-Homepage <http://www.python.org> heruntergeladen werden. Python ist kostenlos und kompatibel mit der GNU General Public License (GPL).

### E.2 Einige besondere Merkmale von Python

- ▶ Die Python-Syntax ermöglicht sehr kompakte Programmtexte.
- ▶ Das Layout des Quelltextes dient nicht allein der besseren Lesbarkeit, sondern hat eine Bedeutung. So markiert das Zeilenende das Ende einer Anweisung. Anweisungsblöcke (wie z.B. das Innere einer Schleife) werden durch Einrückung festgelegt. Zeilen des Programmtextes, die

um die gleiche Anzahl von Stellen eingerückt sind, gehören zum gleichen Anweisungsblock.

- ▶ Mit Python kann man objektorientiert, imperativ und funktional programmieren.
- ▶ Im Unterschied etwa zu Pascal oder Java muss den Variablen kein Datentyp explizit zugeordnet werden. Es gibt keine Variablendeklarationen. Der Datentyp ergibt sich aus dem Kontext. Wenn es erforderlich ist, finden automatische Typkonvertierungen statt.
- ▶ Mehrere Variablen können zu Tupeln zusammengefasst werden, die in einer einzigen Zuweisung verarbeitet werden können. Die Komponenten eines Tupels werden durch Kommata getrennt. Durch diesen Mechanismus können Hilfsvariablen eingespart werden. So werden mit einer einzigen Anweisung  $x, y = y, x$  die Inhalte der Variablen  $x$  und  $y$  vertauscht.
- ▶ In der Mathematik übliche Schreibweisen wurden in die Syntax übernommen. Ausdrücke wie  $a < b < c$  oder verkettete Zuweisungen  $a = b = c$  sind erlaubt.
- ▶ Python verwendet wenige, aber sehr mächtige Sprachkonstrukte. Auf alles Überflüssige wurde verzichtet. In dieser Hinsicht kann man Python als »minimalistisch« bezeichnen.
- ▶ Objekte besitzen einen Wahrheitswert. So haben alle Zahlen ungleich null und alle nicht leeren Zeichenketten den Wahrheitswert »wahr«.
- ▶ Langzahl-Arithmetik (Rechnen mit ganzen Zahlen beliebiger Länge) ist integriert.

### E.3 Python 2 und 3

Es ist wichtig zu wissen, dass es zwei unterschiedliche Sprachvarianten von Python gibt: Python 2 und Python 3. Im Jahre 2007 erschien mit Python 3 erstmals eine Version, die mit den Vorgänger-Versionen nicht kompatibel ist. Man entschied sich zu diesem Bruch, um einige grundlegende Designschwächen von Python zu beseitigen. Python wurde noch

konsistenter, als es bereits war. Wie bei Java können nun sämtliche Unicode-Zeichen für Bezeichner und Strings verwendet werden. Man unterscheidet nun zwischen Strings als Zeichenketten und Bytestrings als Oktettenfolgen. Es gibt keine `print`-Anweisung mehr, sondern eine Funktion `print()`.

Ältere Python-2-Programme können in der Regel nicht von einem Python-3-Interpreter ausgeführt werden. Die letzte Python 2-Version ist Python 2.7. Man kann es immer noch herunterladen und auf vielen Computern findet man noch Python 2. Aber seit 2020 wird Python 2 nicht mehr weiterentwickelt und gepflegt.

Dieses Buch verwendet durchgehend die Syntax von Python 3.

## E.4 Hinweise zum Lesen dieses Buches

### Typographie

- ▶ Python-Quelltext, Funktions- und Variablennamen, Operatoren, Grammatikregeln, Zahlen und mathematische Ausdrücke werden in einem Zeichenformat mit fester Breite gesetzt. Beispiele:

```
x = y + 1  
(x < y) and (len(liste) < 5)
```

- ▶ Bei der Darstellung des Formats eines Funktionsaufrufs sind die Argumente kursiv. Sie sind – im Unterschied zum Funktionsnamen – Metabezeichner, die nicht Buchstabe für Buchstabe so aufgeschrieben werden, wie es angegeben ist, sondern durch andere Bezeichner oder Literale ersetzt werden können. Beispiel:

```
range(zahl)
```

- ▶ Wichtige Passagen in Python-Listings, auf die im Text Bezug genommen wird, sind zuweilen fett gedruckt, um das Wiederfinden zu erleichtern.

### Beispiele

Die Beispiele mit Python-Quelltext in diesem Buch kann man in drei Gruppen einteilen:

- ▶ Auszüge aus einer interaktiven Session erkennt man an dem Python-Prompt `>>>`. Diese Beispiele kann man im Shell-Fenster einer Python-Entwicklungsumgebung (z.B. IDLE) ausprobieren. Hinter dem Prompt ist die Benutzereingabe. Zeilen ohne Prompt enthalten eine System-Antwort. Beispiel:

```
>>> x = 1
>>> y = 2
>>> print(x + y)
3
```

- ▶ Beispiele für eigenständige Python-Skripte oder Module mit Funktionsdefinitionen enthalten überhaupt kein Prompt. Sie werden in einem Editorfenster erstellt und dann durch Aufruf des Interpreters oder nach Import in einer interaktiven Session getestet. Beispiel:

```
# Quicksort
def qsort(L):
    if len(L) <= 1: return L
    else:
        return qsort( [ x for x in L[1:] if x < L[0]]\
            + [L[0]] \
            + qsort( [y for y in L[1:] if y >= L[0]] )
```

- ▶ Für Aufrufe in einem Konsolenfenster des Betriebssystems (z.B. Unix-Shell oder MS-DOS-Eingabeaufforderung) verwenden wir das Prompt `>`. Beispiel:

```
> python addiere.py 1 27
28
```

## Hinweise zum Aufbau der Kapitel

- ▶ Am Ende eines Unterkapitels mit einer Funktionsbeschreibung finden sich häufig Verweise auf andere Kapitel mit korrespondierendem Inhalt (»Siehe auch: ...«)
- ▶ Kapitel, in denen Module beschrieben werden, sind meist nach folgendem Schema aufgebaut:
  - ▶ Kurze Einleitung
  - ▶ Tabellarische Übersicht über die Objekte (Funktionen, Konstanten, Klassen, Methoden) des Moduls in alphabetischer Reihenfolge
  - ▶ Ausführliche Erklärung der wichtigsten Objekte mit Beispielen
  - ▶ Am Ende gelegentlich komplexere Anwendungsbeispiele
- ▶ Programmiertipps und wichtige Hinweise befinden sich in grauen Kästen.