Python, eine moderne Programmiersprache

1 Was ist Python?

Python ist eine moderne Programmiersprache, die sich bei vielen Entwicklerinnen und Entwicklern großer Beliebtheit erfreut. Sie wurde Anfang der 90er Jahre von Guido van Rossum, einem niederländischen Programmierer, entworfen und wird seitdem von einer großem Team Freiwilliger als freies Open-Source-Projekt gepflegt.

Python ist eine so genannte höhere Programmiersprache, die die Zeit der Programmiererin oder des Programmierers über die Zeit des Computers stellt. Gelegentlich ist Python-Code also etwas langsamer als zum Beispiel mühsam handoptimierter C-Code – dafür lässt es sich in Python viel schneller und angenehmer entwickeln.

Zu den Anwendungsbereichen von Python gehören unter anderen die Web-, System- und Spiele-Entwicklung. Außerdem wird Python für wissenschaftliche Zwecke eingesetzt – das ist der Aspekt, den wir beleuchten werden.

2 Installation von Python

Damit ein Computer Python-Programme ausführen kann, muss man zunächst einen *Python-Interpreter* installieren. Das ist die erste Aufgabe an euch!

Unter Ubuntu Linux und anderen Debian-Derivaten. Öffne eine Konsole und setz den Befehl `sudo apt-get install python-numpy python-scipy python-matplotlib` ab. Das war's schon.

Unter Mac OS X und Windows. Lade auf `http://continuum.io/downloads` das Komplettpaket Anaconda herunter und klicke dich durch die Installation. Wähle die Python-Version 2.7 und unter Windows die 32-Bit-Version (auch, wenn dein Computer ein 64-Bit-Prozessor haben sollte).

3 Erste Schritte

3.1 Hallo, Welt!

Das erste Programm, dass man schreibt, wenn man eine neue Programmiersprache lernt, ist *Hallo Welt!*: Ein Programm, dass eine kurze Meldung ausgibt und sich dann beendet. In Python sieht das so aus:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

print("Hallo Welt!")
```

Tippe das Programm ab (oder kopiere den Quellcode von XXX) und führe es aus. Wenn es nicht klappt, dann melde dich bei uns (XXX Forum?).

Die ersten beiden Zeilen finden sich in jedem Python-Programm. Die erste ist vor allem unter Linux und OS X wichtig; sie ist der Indikator für das Betriebssystem, dass es sich um ein Python-Programm handelt. Es ist guter Stil, sie auch unter Windows beizubehalten. Zeile 2 hat technische Gründe.[1]

Zeile 3 ist eine Leerzeile. Leerzeilen haben für Python selbst keine Bedeutung, können also nach Belieben hinzugefügt oder entfernt werden. Es ist aber guter Stil, einzelne Sinneinheiten durch Leerzeilen zu trennen, um den Code für den Menschen übersichtlicher zu gestalten. Es gibt ja auch einen Grund, wieso es im Deutschen und vielen anderen natürlichen Sprachen Absätze gibt.

Die eigentliche Arbeit wird durch Zeile 4 angestoßen. Dort wird die in Python vordefinierte Prozedur `print` mit dem *Argument* `¨Hallo, Welt!¨` aufgerufen. Die Schreibweise soll an die in der Mathematik übliche Notation für Funktionen erinnern – dort schreibt man zum Beispiel „$\sin(5)$“.

3.2 Die interaktive Python-Shell

- Einfache Rechnungen

3.3 Programmierfehler und wie man mit ihnen umgeht

Beim Programmieren macht man Fehler. Von denen gibt es vor allem zwei Arten: *Syntaxfehler* und *inhaltliche Fehler*. Ein Syntaxfehler tritt auf, wenn man sich nicht an die Schreibregeln von Python hält. Zum Beispiel wird der Code

```
print("Hallo, Welt!"
```

[1] Zeile 2 setzt fest, dass der Programmcode in der Zeichenkodierung UTF-8 (und nicht etwa in dem älteren Standard ISO-8859-1) interpretiert werden soll. Eine Zeichenkodierung gibt an, wie Umlaute und andere Zeichen, die über den Sprachschatz des amerikanischen ASCII-Standards aus den 70er Jahren (XXX) hinausgehen, als Bytes gespeichert werden sollen.

nicht funktionieren, da die schließende Klammer fehlt. Syntaxfehler werden vom Python-Interpreter direkt nach dem Start, noch bevor er mit dem Ausführen des Codes beginnt, erkannt und gemeldet.

Programmiersprachen wie Python sind in ihrer Syntax sehr streng. XXX Beispiel, XXX Warnung vor falscher Zeilenangabe im Fehlerbericht, XXX Schleichwerbung für statische Typsysteme

- inhaltliche Fehler (Beispiel: Variablenvertauschung)
- Umlaute
- Grundtechniken im Debugging

4 Plotten

5 Newton-Verfahren