

Programmieren in Python

1. Was ist Python?

Malte Helmert

Albert-Ludwigs-Universität Freiburg

KI-Praktikum, Sommersemester 2009

Kurz: Python ist eine objektorientierte Skriptsprache.

Ausführlicher: Python ist eine. . .

- ▶ objektorientierte,
- ▶ dynamisch getypte,
- ▶ interpretierte und
- ▶ interaktive
- ▶ High-Level-Programmiersprache.

Mehr zu diesen Eigenschaften später.

- ▶ Ursprünglich entwickelt von **Guido van Rossum** im Rahmen eines Forschungsprojekts am „Centrum voor Wiskunde en Informatica“ in Amsterdam.
- ▶ Entwickelt seit **1989**, erste öffentliche Version **1991**.
- ▶ Meilensteine: Versionen 1.0.0 (**1994**), 1.5 (**1998**), 2.0 (**2000**), 3.0 (**2008**)
- ▶ Mittlerweile wird Python als **Open-Source-Projekt** von der Allgemeinheit weiterentwickelt, wobei ein innerer Kern die meiste Arbeit übernimmt. Guido van Rossum hat als „BDFL“ (*benevolent dictator for life*, gütiger Diktator auf Lebenszeit) das letzte Wort.

Python ist *nicht* nach einem Reptil benannt, sondern nach **Monty Python's Flying Circus**, einer (hoffentlich!) bekannten englischen Komikertruppe aus den 1970ern.

Daher auch viele Namen von Tools rund um Python:

- ▶ IDLE
- ▶ Eric
- ▶ Bicycle Repair Man
- ▶ Grail

Wo andere Programmiersprachen die Variablen *foo* und *bar* verwenden, wählt man in Python gerne **spam** und **egg**.

- ▶ C, C++, Java
- ▶ Perl
- ▶ PHP
- ▶ LISP

Python hat gegenüber der C-Familie einen deutlich höheren Abstraktionsgrad („weiter weg von der Maschine“):

- ▶ Automatische Speicherverwaltung
- ▶ Unbeschränkte Ganzzahlarithmetik
- ▶ Eingebaute komplexe Datentypen: `list`, `dict`, `tuple`
- ▶ Funktionen höherer Ordnung: `map`, `filter`, `reduce`
- ▶ Alles ist ein Objekt
- ▶ Alles ist dynamisch: Metaklassen und Metaprogrammierung

Im Vergleich zu Sprachen aus der C-Familie sind Python-Programme:

- ▶ kürzer
- ▶ lesbarer
- ▶ portabler
- ▶ langsamer

Python vs. Perl

Gemeinsamkeiten:

- ▶ ursprüngliches Anwendungsgebiet: Unix-Scripting
- ▶ ähnlicher Abstraktionsgrad
- ▶ dynamisch getypt

Unterschiede:

Perl

There's more than one way to do it.

- ▶ viele Abkürzungen
- ▶ sehr kompakt
- ▶ schwach getypt: `2+"foo" = 2`
- ▶ im Wesentlichen prozedural
- ▶ auf Scripting zugeschnitten

Python

There should be one, and preferably only one, obvious way to do it.

- ▶ wenige Spezialfälle
- ▶ sehr lesbar
- ▶ stark getypt: `2+"foo" = Fehler`
- ▶ objekt-orientiert
- ▶ allgemein konzipiert

Gemeinsamkeiten:

- ▶ ähnlicher Abstraktionsgrad
- ▶ dynamisch getypt

Unterschiede:

PHP

Auf dynamische Webseiten zugeschnitten.

- ▶ viele automatische Dinge
- ▶ ein globaler Namensraum
- ▶ im Wesentlichen prozedural
- ▶ C-artige Syntax

Python

Als allgemeine Programmiersprache konzipiert.

- ▶ “Explicit is better than implicit.”
- ▶ Module und Namensräume
- ▶ objekt-orientiert
- ▶ keine C-artige Syntax

- ▶ Fundamental unterschiedliche Syntax:

LISP

```
(defun factorial (n)
  (if (<= n 1) 1
      (* n
         (factorial (- n 1)))
  ))
```

Python

```
def factorial(n):
    if n <= 1:
        return 1
    else:
        return n * factorial(n-1)
```

- ▶ Viele Gemeinsamkeiten im “Kern”:
 - *Python for Lisp Programmers*,
<http://www.norvig.com/python-lisp.html>

- ▶ Online-Dokumentation
- ▶ IDEs & Editoren
- ▶ Internet-Ressourcen
- ▶ Bücher

Einstiegspunkt: <http://docs.python.org/>

Besonders wichtig/interessant:

- ▶ am Anfang das **Tutorial**
(<http://docs.python.org/tutorial/index.html>)
- ▶ im Programmieralltag die **Library Reference**
(<http://docs.python.org/library/index.html>)

- ▶ **IDLE** ist die bei Python mitgelieferte IDE.
- ▶ **Boa Constructor** und **PythonWin** (nur Windows) sind weitere freie IDEs.
- ▶ **Black Adder**, **Komodo** und **Wing IDE** sind populäre kommerzielle Python-IDEs.
- ▶ Für **Visual Studio** (nur Windows) gibt es ein Python-Plugin.
- ▶ Für **Eclipse** ebenfalls (**Pydev**).

- ▶ **XEmacs** und **GNU Emacs** haben einen mitgelieferten Python-Modus, der automatisch verfügbar ist.
 - ▶ Für GNU Emacs gibt es sogar zwei.
 - ▶ Der aus meiner Sicht bessere ist der **nicht** vorinstallierte.
 - ▶ Unter Debian/Ubuntu: `sudo apt-get install python-mode`.
- ▶ **vim** und **gvim** unterstützen Python gut und können in Python programmiert werden.
- ▶ **jEdit** und **Leo** unterstützen Python sehr gut.

↔ Im Folgenden alle Beispiele unter Linux mit XEmacs.

Offizielle Website:

- ▶ <http://www.python.org/>
- ▶ interessant dort zum Beispiel: Dokumentation, Python FAQs, Python Wiki, PEPs, Python Package Index

Newsgroups:

- ▶ `comp.lang.python`
- ▶ `comp.lang.python.announce`

Mailingliste:

- ▶ `python-dev`: siehe <http://mail.python.org/mailman/listinfo/python-dev>
- ▶ Newsgroup-Interface über <http://www.gmane.org/>

- ▶ Alex Martelli: **Python in a Nutshell**
 - ▶ Alles, was man wissen muss. Für Fortgeschrittene.
 - ▶ Aktuelle Auflage (2006) behandelt Python 2.5.
- ▶ Alex Martelli, Anna Martelli Ravenscroft und David Ascher: **Python Cookbook**
 - ▶ Codebeispiele. Sehr nützlich.
 - ▶ Aktuelle Auflage (2005) behandelt Python 2.4.
- ▶ Mark Lutz und David Ascher: **Learning Python**
 - ▶ Guter Ruf, ich kenne es nicht. Für Einsteiger.
 - ▶ Aktuelle Auflage (2007) behandelt Python 2.6.
 - ▶ Deutsch als **Einführung in Python** (alte Auflage).
- ▶ ...viele andere (nennenswert vielleicht noch **Programming Python**)

Alle erwähnten Bücher sind im O'Reilly-Verlag erschienen.

Zwei kostenlose Online-Bücher zu Python:

- ▶ Mark Pilgrim: **Dive into Python**
 - ▶ <http://diveintopython.org/>
 - ▶ Für erfahrenere Programmierer.
 - ▶ Auch käuflich zu erwerben.
- ▶ Allen Downey, Jeff Elkner und Chris Meyers:
How to Think Like a Computer Scientist
 - ▶ <http://www.greenteapress.com/thinkpython/thinkCSpy/>
 - ▶ Für Programmieranfänger.
 - ▶ Spätere Auflagen käuflich zu erwerben.