Programmieren in Python

1. Was ist Python?

Robert Mattmüller

Albert-Ludwigs-Universität Freiburg

Handlungsplanungs-Praktikum
Wintersemester 2010/2011

Was ist Python?

Kurz: Python ist eine objektorientierte Skriptsprache.

Ausführlicher: Python ist eine ...

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

Mehr zu diesen Eigenschaften später.

Zur Geschichte Pythons

- 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.

Zum Namen

Python ist *nicht* nach einem Reptil benannt, sondern nach **Monty Python**, 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**.

Vergleich von Python mit anderen Programmiersprachen

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

Python vs. C, C++, Java

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`
- objektorientiert
- allgemein konzipiert

Python vs. PHP

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
- objektorientiert
- keine C-artige Syntax

Python vs. LISP

- 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`

Python-Ressourcen

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

Online-Dokumentation von Python

Einstiegspunkt: `http://docs.python.org/py3k/`

Besonders wichtig/interessant:

- am Anfang das **Tutorial** (`http://docs.python.org/py3k/tutorial/index.html`)
- im Programmieralltag die **Library Reference** (`http://docs.python.org/py3k/library/index.html`)

IDEs für Python

- **IDLE** ist die Standard-IDE für Python.
- **Eric** ist eine weitere beliebte freie IDE.
- **Komodo** und **Wing** sind populäre kommerzielle Python-IDEs.
- Für **Eclipse** gibt es ein Python-Plugin: **Pydev**.

Editoren für Python

- **XEmacs** und **GNU Emacs** haben einen mitgelieferten Python-Modus, der automatisch verfügbar ist.
  - Für GNU Emacs gibt es sogar zwei.
  - Der 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.

Offizielle Internet-Ressourcen zu Python

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`

Mailinglisten:

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

Bücher zu Python (aus Papier)

- 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: **Learning Python**
  - Guter Ruf. Für Einsteiger.
  - Aktuelle Auflage (2009) behandelt Python 2.6 und 3.0.
  - Deutsch als **Einführung in Python** (alte Auflage).
- . . . viele andere

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

Online-Bücher zu Python

Drei kostenlose Online-Bücher zu Python:

- Mark Pilgrim: **Dive Into Python 3**
  - `http://diveintopython3.org/`
  - Für erfahrenere Programmierer.
  - Auf dem Stand von 2010.
- Allen Downey, Jeffrey Elkner und Chris Meyers: **How to Think Like a Computer Scientist**
  - `http://www.greenteapress.com/thinkpython/thinkCSpy/`
  - Für Programmieranfänger.
  - Auf dem Stand von 2008.
- Peter Kaiser und Johannes Ernesti: **Python**
  - `http://openbook.galileocomputing.de/python/`
  - Für Programmieranfänger.
  - Auf deutsch.
  - Auf dem Stand von 2007.

Alle drei Bücher sind auch gedruckt auf toten Bäumen erhältlich.