

# Einführung in die Programmiersprache Python

Dipl.-Inf. Frank Hofmann

10. April 2007



# Zielsetzung

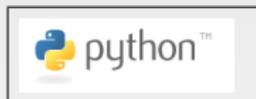
## Fragen, die sich jeder stellt

- Kann ich damit Programmieren lernen?
- Hilfe! Noch eine Programmiersprache?!
- Kann Python für mich nützlich sein?

# Übersicht

- 1 Erster Überblick
- 2 Sprachcharakter
- 3 Referenzen

# Python in Schlagworten



- höhere Programmiersprache
- entwickelt Anfang der 1990er Jahre von Guido van Rossum am Centrum voor Wiskunde en Informatica in Amsterdam (ursprünglich für das verteilte Betriebssystem Amoeba)
- Skript-Sprache  
Übersetzung des Python-Programmes in einen Zwischencode, der dann von einem Interpreter ausgeführt wird

# Herkunft und Bedeutung des Namens



Von links: Terry Jones,  
Graham Chapman, John  
Cleese, Eric Idle, Terry  
Gilliam, Michael Palin

- benannt nach der britischen Komikertruppe Monty Python
- kein Zusammenhang zu Schlangen

# Zielsetzung bei der Entwicklung (nach Guido van Rossum)

- eine einfache, intuitive Sprache, die Konkurrenten in Mächtigkeit in nichts nachsteht
- Open Source, so dass jeder bei der Entwicklung helfen kann
- Quelltext, der genauso einfach zu lesen ist wie reines Englisch
- für tägliche Aufgaben geeignet und kurze Entwicklungszeiten ermöglicht

# Umsetzung

- möglichst einfach und übersichtlich
  - wenige Schlüsselworte
  - reduzierte, grammatikalische Syntax
  - Optimierung auf Lesbarkeit

- Beispiel:

```
palette = ["rot", "gelb", "grün"]
for farbe in palette:
    print farbe
```

Ausgabe:

rot gelb grün

# Sprachcharakteristik

- leicht erlernbar
- lesbar, verständlich
- auch nach längerer Abstinenz
- geeignet für Teams
- geeignet für Einsteiger

# Projekte, die Python einsetzen (Auswahl)

- GIMP
- dia
- gqview
- OpenOffice
- Blender
- Maya
- Zope (teilweise)
- Google (teilweise)

# Plattform und Lizenz, Release

- Plattform:  
unabhängig, nutzbar für alle Plattformen
- Lizenz: Python Lizenz
- Release: 2.5 (September 2006)

# Programmierparadigmen

- imperativ  
Statements, Funktionen, Prozeduren ...
- objektorientiert  
Objekte, Klassen, Methoden ...
- funktional  
alles ist eine Funktion (mit Argumenten) ...

# Pythons Einflüsse

- Python wurde beeinflusst durch  
ABC, Algol 60, Modula-3, Icon, C, Perl, LISP, Smalltalk, Tcl
- Python beeinflusste  
Ruby, Boo

# Kerneigenschaften

- dynamische Speicherverwaltung
- keine explizite Typisierung
- Datentypen (Auswahl):
  - Integer
  - Fließkommazahlen
  - Strings
  - Listen
  - Tupel
  - Mengen
  - Arrays

# Strukturierung

- Strukturierung durch Einrückung
- keine Markierung durch Klammern und Schlüsselworte
- fördert Lesbarkeit
- Vergleich C/Python

## Fakultätsfunktion in C

```
int fakultaet(int x) {  
    if (x > 1)  
        return x * fakultaet(x-1);  
    else  
        return 1;  
}
```

## Fakultätsfunktion in Python

```
def fakultaet(x):  
    if x > 1:  
        return x * fakultaet(x - 1)  
    else:  
        return 1
```

# Literatur und Links

- Michael Weigend:  
Objektorientierte Programmierung mit Python  
mitp Verlag, 2005, ISBN 3-8266-1571-9
- David Mertz:  
Text Processing in Python, Addison-Wesley, 2003, ISBN  
0321-11254-7
- The Python Homepage  
<http://www.python.org>

# Veranstaltungshinweise

- **Python – eine Einführung**

Workshop

Freitag, den 13. April 2007, 17:30 bis 20:00 Uhr

- **Einführung in GTK/Python**

Vortrag

Dienstag, den 17. April 2007, 19:00

Workshop

Donnerstag, den 19. April 2007, 17:30 bis 20:00 Uhr

# The End

Danke für Ihre Aufmerksamkeit :-)

## **Kontakt:**

Dipl.-Inf. Frank Hofmann

Email <frank.hofmann@efho.de>

Hofmann EDV – Linux, Layout und Satz, Potsdam