

# Kurz-Einführung Python

Prof. Dr. Rüdiger Weis

Beuth Hochschule Berlin

Wintersemester 2009/2010

# Paul Graham

## Paul Graham

"The programmers you'll be able to hire to work on a Java project won't be as smart as the ones you could get to work on a project written in Python."

<http://www.paulgraham.com/gh.html>

# Paul Graham: The Python Paradox

## Paul Graham

"I didn't mean by this that Java programmers are dumb. I meant that Python programmers are smart."

<http://www.paulgraham.com/pypar.html>

# Why Python?

Eric Raymond

"I was generating working code nearly as fast as I could type."

"Why Python?"

<http://www.linuxjournal.com/node/3882/print>

# Google

## Peter Norvig, 'director of search quality' bei Google

"Python has been an important part of Google since the beginning, and remains so as the system grows and evolves. Today dozens of Google engineers use Python, and we're looking for more people with skills in this language."

# YouTube

Cuong Do, Software Architect, YouTube

"Python is fast enough for our site and allows us to produce maintainable features in record times, with a minimum of developers"

# Thawte Consulting, Mark Shuttleworth

## Mark Shuttleworth

"Python makes us extremely productive, and makes maintaining a large and rapidly evolving codebase relatively simple"

# Industrial Light & Magic

## Tommy Burnette, Senior Technical Director

"Python plays a key role in our production pipeline. Without it a project the size of Star Wars: Episode II would have been very difficult to pull off. From crowd rendering to batch processing to compositing, Python binds all things together "

# Industrial Light & Magic II

**Philip Peterson, Principal Engineer, Research & Development**

"Python is everywhere at ILM. It's used to extend the capabilities of our applications, as well as providing the glue between them. Every CG image we create has involved Python somewhere in the process"

# Beispiele Python Anwendungen

- Bittorrent
- Mojo Nation
- Miro
- Application-Server Zope
- Plone
- Django
- eduMagnet
- ...

# Literatur

## Web Quellen

- <http://www.python.org/>
- Think Like a Computer Scientist
  - Allen Downey, Jeff Elkner, Chris Meyers,
  - "How to Think Like a Computer Scientist: Learning with Python"
  - <http://www.greenteapress.com/thinkpython/>

# Python ist eine

- objektorientierte
- dynamische
- stark typisierter

Skript-Sprache.

# Entstehung

- 1990er Jahre von **Guido van Rossum** am Centrum voor Wiskunde en Informatica (CWI) Amsterdam
- Skriptsprache für das verteilte Betriebssystem **Amoeba** (Maindesigner: Andrew S. Tanenbaum) entwickelt.
- Benannt nach Monty Python.
- **Open Source**

# Veröffentlichung

- Guido van Rossum and Jelke de Boer, "Interactively Testing Remote Servers Using the Python Programming Language", CWI Quarterly, Volume 4, Issue 4 (December 1991), Amsterdam, pp 283-303.

# Systemprogrammierung

- Skriptsprache von Amoeba
- Xen
- GNU-Radio
- ...

# Entwurfsziele

- Einfach
- Übersichtlich
  - Blockbildung mittels Einrücken!
- Nutzt Erfahrungen mit ABC

# More than one religion

Python ermöglicht *sehr gut*

- objektorientierte
- strukturierte
- funktionale

Programmierung, erzwingt sie aber nicht.

# Schnelles Entwickeln

- Dynamische Typisierung
- Garbage Collection

# Mächtige Datenstrukturen

- Strings
- Listen
- Tupel
- Dictionaries

# Beispiel: Liste in Java

```
for(int i=0;i<array.length;i++)  
{  
    int x=array[i];  
    verarbeite(x);  
}
```

# Beispiel: Liste in Python

```
for x in list: verarbeite(x)
```

# Functionales Programmieren

- Lambda-Formen
- filter()
- map()
- reduce()
- List comprehensions

# Einsatzgebiete

- Rapid Development
- Eingebettete Skriptsprache

# Python Interpreter

```
$ python
Python 2.5.2 (r252:60911, Oct  5 2008, 19:24:49)
[GCC 4.3.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Tip: ipython

# Supertaschenrechner

- Beliebig lange Ganzzahlen
- Komplexe Zahlen
- ...

# Beliebig Lange Ganzzahlen

```
>>> 2**64
18446744073709551616L
>>> 2**128
340282366920938463463374607431768211456L
```

# Komplexe Zahlen

- Imaginäre Zahlen werden mit dem Suffix "j" oder "J" gekennzeichnet.
- Komplexe Zahlen werden als zwei Fließkommazahlen dargestellt.
- `complex(real, imag)` ergibt `real+imagJ`
- `z.real` und `z.imag` extrahieren Real- beziehungsweise Imaginär-Teil.

# Strenge Typprüfung

- Implizite Umwandlungen sind für numerische Typen.
- Keine implizite Umwandlung zwischen Zahlen und Zeichenketten (Unterschied zu Perl).

Prinzip der geringsten Überraschung.

# Hallo Welt

```
#include <iostream.h>

void main()
{
    cout << "Hello, world." << endl;
}
```

# Hallo Welt in Python

```
>>> print("Hallo Welt")  
Hallo Welt
```

# Shell Fenstern

```
#!/usr/bin/python
import Tkinter
fenster=Tkinter.Button(text="Hallo Welt",command='exit')
fenster.pack()
fenster.mainloop()
```

# Unser erstes GUI Fenster

```
$ chmod +x fenstern  
$ ./fenstern
```

# ©opyleft

## ©opyleft

- Erstellt mit Freier Software
- © Rüdiger Weis, Berlin 2009
- unter der GNU Free Documentation License.