Netzwerke – Übung SoSe 2020

Python 101

Benjamin.Troester@HTW-Berlin.de
PGP: ADE1 3997 3D5D B25D 3F8F 0A51 A03A 3A24 978D D673

Benjamin Tröster

Python

- Nicht die Schlange, sondern großartige britische Comedy

Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

Road-Map

Hochschule für Technik und Wirtschaft Berlin
University of Applied Sciences

Was ist Python

Python: Dynamische Programmiersprache, die unterschiedliche Programmierparadigma unterstützt:

- Prozedural – C, Fortran
- Objektorientiert – Smalltalk, Java
- Funktionla – LISP, Haskell

Also ist Python eine Multi-Paradigma Sprache, wie C++ oder Rust. Standard: Python Byte-Code wird im Python-Interpreter ausgeführt $\rightarrow$ Plattformunabhängiger Code

Warum Python?

Extrem vielfältige Programmiersprache

- Website development, data analysis, server maintenance, numerical analysis, ...
- Syntax is clear, easy to read and learn (almost pseudo code)
- Common language
- Intuitive object oriented programming
- Full modularity, hierarchical packages
- Comprehensive standard library for many tasks
- Big community
- Simply extendable via C/C++, wrapping of C/C++ libraries
- **Focus: Programming speed**

Python Zen

- 20 software principles that influence the design of Python:
  1. Beautiful is better than ugly.
  2. Explicit is better than implicit.
  3. Simple is better than complex.
  4. Complex is better than complicated.
  5. Flat is better than nested.
  6. Sparse is better than dense.
  7. Readability counts.
  8. Special cases aren't special enough to break the rules.
  9. Although practicality beats purity.
  10. Errors should never pass silently.
  11. Unless explicitly silenced.
  12. ...

Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

Hello, World!

```
1 #!/usr/bin/env python3
2 #This is a commentary
3 print("Hello world!")
```

```
1 $ python3 hello_world.py
2 Hello world!
3 $
```

```
4 $ chmod u+x hello_world.py
5 $ ./hello_world.py
6 Hello world!
7 $
```

Hello, World!

```
1  #!/usr/bin/env python3
2  name = input("'Whats your name?")
3  print("Hello", name)
```

```
8  $./hello_user.py'
9  Whats your name? Dieter
10 Hello Dieter
11 $
```

Strong and Dynamic Typing

Strong Typing:

- Object is of exactly one type! A string is always a string, an integer always an integer
- Counterexamples: PHP, JavaScript, C: char can be interpreted as short, void * can be everything

Dynamic Typing:

- No variable declaration
- Variable names can be assigned to different data types in the course of a program
- An object's attributes are checked only at run time
- Duck typing (an object is defined by its methods and attributes)

When I see a bird that walks like a duck and swims like a duck and quacks like

Example: Strong and Dynamic Typing

```
1  #!/usr/bin/env python3
2  number = 3
3  print(number, type(number))
4  print(number + 42)
5  number = "3"
6  print(number ,type(number))
7  print(number + 42)
```

```
12 3 <class 'int '>
13 45
14 3 <class 'str '>
15 Traceback(most recent call last ):
16 File "types.py" ,line 7, in <module>
17 print(number + 42)
18 TypeError: can only concatenate str (not "int") to str
```

REPL

The interpreter can be started in interactive mode:

```
$ python3
Python 3.7.2 ( default , May 23 2019 , 03:15:18)
[GCC 10.1.0] on freebsd
Type "help" , "copyright" , "credits" or "license" for
more information .
>>> print (" hello world")
hello world
>>> a = 3 + 4
>>> print (a)
7
>>> 3 + 4
7
>>>
```

	Python2	Python3
shebang	`#!/usr/bin/python`	`#!/usr/bin/python3`
IDLE	idle	idle3
print cmd (syntax)	print	print()
input cmd (syntax)	raw_input()	input()
unicode	u"..."	all strings
integer type	int/long	int (infinite)

Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

Numerical Data Types

- int: integer numbers (infinite)
- float: corresponds to double in C
- complex: complex numbers (j is the imaginary unit)

```
1 a = 1
2 c = 1.0
3 c = 1 e0
4 d = 1 + 0j
```

Hochschule für Technik und Wirtschaft Berlin
University of Applied Sciences

Operators on Numbers

- Basic arithmetics: + , - , * , /
  hint: Python 2 $\Rightarrow 1/2 = 0$
  Python 3 $\Rightarrow 1/2 = 0.5$
- Div and modulo operator: // , % , divmod(x, y)
- Absolute value: abs(x)
- Rounding: round(x)
- Conversion: int(x) , float(x) , complex(re [, im=0])
- Conjugate of a complex number: x.conjugate()
- Power: x ** y , pow(x, y)
- Result of a composition of different data types is of the "bigger" data type.