

Python

...für Anfänger

Unit 1

by

Dr. Günter Kolousek

Interpreter

- ▶ Starten

```
$ python
```

```
Python 3.4.3 (default, Mar 26 2015, 07:36:01)
```

```
[GCC 4.9.2 20150304 (prerelease)] on linux
```

```
Type "help", "copyright", "credits" or "license()"
```

```
>>>
```

- ▶ Beenden

```
>>> exit()
```

Datentypen - Zahlen

- ▶ Datentyp (engl. data type)
 - ▶ gibt Wertebereich an
 - ▶ legt Operationen fest
- ▶ Zahlen (engl. number)
 - ▶ ganze Zahlen (engl./lat. integer): 1, 3, 42,...
 - ▶ Gleitkommazahlen (engl. floating point number): 0.5, 3.1415926, 2.7182818,...

Ausdrücke

(engl. expression)

```
>>> 1 + 1
```

```
2
```

```
>>> _ + 1
```

```
3
```

```
>>> _ + 1
```

```
4
```

```
>>> type(4)
```

```
<class 'int'>
```

Ausdrücke - 2

```
>>> 1.5
```

```
1.5
```

```
>>> type(_)
```

```
<class 'float'>
```

```
>>> 1 + 1.5 * 3
```

```
5.5
```

```
>>> (1 + 1.5) * 3
```

```
7.5
```

Ausdrücke - 3

- Subtraktion, Division

```
>>> 12 / 3 - 1  
3.0
```

- Potenz: **

```
>>> 2 ** 3  
8
```

Fehler - Syntaktische Fehler

(engl. syntax error)

```
>>> How do you do?
      File "<stdin>", line 1
        How do you do?
            ^
SyntaxError: invalid syntax
>>>
```

→ Regeln der Grammatik (Struktur der Sprache) verletzt

Fehler - Semantische Fehler

```
>>> pi
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'pi' is not defined
>>>
```

→ Bedeutung falsch!

Fehler - Laufzeitfehler

(engl. runtime error)

```
>>> 1 / 0
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
ZeroDivisionError: division by zero
```

```
>>>
```

→ zur Laufzeit auf Grund "falscher" Eingabedaten

Fehler - Übersicht

(engl. error, failure)

- ▶ Syntaktische Fehler (Syntaxfehler)
- ▶ Semantische Fehler
- ▶ Laufzeitfehler
- ▶ logische Fehler
 - ▶ Fehler in der Logik des Programms (z.B. Endlosschleife)

→ Fehler sind unvermeidlich!

→ Umgehen lernen!!

Funktionen und Module

```
>>> sqrt(4)
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
NameError: name 'sqrt' is not defined
```

```
>>> from math import *
```

```
>>> sqrt(4)
```

```
2.0
```

```
>>> pi
```

```
3.141592653589793
```

Funktion

- ▶ Ähnlich wie in der Mathematik
- ▶ Funktion hat einen Namen → `sqrt`
- ▶ Funktion kann "aufgerufen" werden
 - ▶ Argumente innerhalb von runden Klammern → 4
 - ▶ Ergebnis wird als Rückgabewert zurückgeliefert → 2.0
 - ▶ Der Aufruf einer Funktion ist ein Ausdruck!

Modul

- ▶ Beinhaltet
 - ▶ Daten \rightarrow `pi`
 - ▶ Code \rightarrow `sqrt`
- ▶ in einer Datei gespeichert \rightarrow `math`
 - ▶ verschiedene Arten von Dateien möglich
- ▶ Inhalt des Moduls kann importiert werden
 \rightarrow `from ... import ...`

Datentypen - Strings

- String (auch Zeichenkette) ist eine Folge von Zeichen

```
>>> "Hello, World"
'Hello, World'
>>> "Hello, 'Maxi'"
"Hello, 'Maxi'"
>>> 'Hello, "Mini"'
'Hello, "Mini"'
>>> "Hello, " + '"Mini2"'
'Hello, "Mini2"'
```

print

```
>>> print("1 + 1")
1 + 1
>>> print(1 + 1)
2
>>> print("1 + 1 =", 1 + 1)
1 + 1 = 2
```

- ▶ vordefiniert, muss nicht importiert werden!
- ▶ liefert "nichts" zurück

print - 2

```
>>> print("Hello, World")
Hello, World
>>> print('Hello, "Mini"')
Hello, "Mini"
>>> print(1.5, "Orangen und", 2, 'Bananen')
1.5 Orangen und 2 Bananen
```


Exception

```
>>> 2 + " Bananen"
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
TypeError: unsupported operand type(s) for +: 'int'
```

- ▶ Typen werden überprüft!
- ▶ → Exception (dt. Ausnahme, Ausnahmesituation)

Minimales Python Programm

- ▶ Datei `hello.py` (mit Endung `.py`)
- ▶ Editor auf UTF-8 einstellen!
- ▶ Datei befüllen mit

```
print("Hello, World")
```
- ▶ Ausführen

```
$ python hello.py
Hello, World
```