Python
...für Anfänger

Unit 4

by

Dr. Günter Kolousek

21. Juli 2015

Boolsche Ausdrücke - 1

```
>>> 5.5 > 2
True
>>> 2 - 1 >= 1 and 1 < 1
False
>>> 1 < 1 or 42 % 2 == 0
True
>>> not 5 > 3
False
>>> True ^ True
False
```

Boolsche Ausdrücke - 2

```
>>> bool(1)
True
>>> bool(5)
True
>>> bool(-7.2)
True
>>> bool(0)
False
>>> bool(0.0)
False
>>> bool(”abc”)
True
>>> bool(””)
False
```

Boolsche Ausdrücke - 3

```
>>> int(True)
1
>>> int(False)
0
>>> float(True)
1.0
>>> str(True)
'True'
```

if - Anweisung - 1

```
x = int(input("x="))
y = int(input("y="))
if x == y:
    print("x == y")
elif x < y:
    print("x < y")
else:
    print("x > y")
```

if - Anweisung - 2

```
>>> if 1:
...     print(”Hi, how are you?”)
...
Hi, how are you?
>>> if ””:
...     print(”Never ever!”)
...
>>>
```

Funktion definieren - 1

```
>>> def print_name():  # Funktionsdefinition
...     print(”Maxi”)
...
>>> print_name()  # Funktionsaufruf
Maxi
>>> def print_name():
...     print(”Maxi”)
...     print(”Mustermann”)
...
>>> print_name()
Maxi
Mustermann
>>>
```

Funktion definieren - 2

```
>>> first_name = ”Maxi”
>>> last_name = ”Mustermann”
>>> def print_name():
...     print(first_name)
...     print(last_name)
...
>>> print_name()
Maxi
Mustermann
>>> print_name
<function print_name at 0xb6f366ec>
```

Funktion definieren - 3

```
>>> a = 3
>>> b = 4
>>> def sum():
...     res = a + b
...     print(res)
...
>>> sum()
7
>>> print(res)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'res' is not defined
```

Funktion definieren - 4

```
>>> res = 0
>>> sum()
7
>>> res
0
>>>
```

Funktion definieren - 5

```
>>> res = 0
>>> def acc():
...     res = res + x
...     return res
...
>>> x = 3
>>> acc()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 2, in acc
UnboundLocalError: local variable 'res' referenced
```

Funktion definieren - 6

```
>>> res = 0
>>> def acc():
...     global res  # pfui!
...     res = res + x
...
>>> x = 3
>>> acc()
>>> res
3
>>> x = 4
>>> acc()
>>> res
7
```

Short-circuit evaluation - 1

```
>>> def f(b):
...     print("inside f")
...     return b
...
>>> def g(b):
...     print("inside g")
...     return b
...
>>> f(True) and g(False)
inside f
inside g
False
>>> f(False) and g(True)
inside f
False
```

Short-circuit evaluation - 2

```
>>> f(False) or g(True)
inside f
inside g
True
>>> f(True) or g(False)
inside f
True
```

Boolsche vs. Bit-Operatoren

```
>>> f(False) & g(True)
inside f
inside g
False
>>> f(True) | g(False)
inside f
inside g
False
>>> f(True) ^ g(True)
inside f
inside g
False
```