

# Scriptsprachen

## Python Basics

Sascha Winter

Lehrstuhl fuer Bioinformatik  
Friedrich-Schiller-Universitaet Jena  
[sascha.winter@uni-jena.de](mailto:sascha.winter@uni-jena.de)

11.08.2014



# Python

- Erschien 1991, Guido van Rossum
- Nach Monty Python benannt
- Leicht erlernbare Syntax
- Gut lesbar
- Multi-Paradigmen Sprache
  - Objektorientiert, Imperativ, Funktional
- Python 2 vs. Python 3

# Python

- Interpretiert
- Automatische Speicherverwaltung
- Dynamisches Typensystem

# Python

```
if 2==2:
    print "TEST".strip("T") # Kommentar!
else:
    print "TEST".ssttrriipp("T")
```

# Python

```
python # interaktive  
python -c 'command' # wegwerf Code  
python FILE.py # fuehrt FILE aus  
python -i FILE.py # FILE, dann interaktive
```

# Blöcke

- Blocksyntax!

```
if 1 + 1 == 2 :  
    print "foo"  
    print "bar"
```

```
if 1 + 1 == 2 :  
    print "foo"; print "bar"
```

- Tab XOR Leerzeichen!
- Styleguide: Einrückung == 4 Leerzeichen

# Zahlen, Strings



# Zahlen

- Unendlich große ints

```
a, b = 5, 2
```

```
a + b, a - b, a * b, a / b, a % b
```

```
i = 0
```

```
i += 1
```

```
a, b = 5.5, 2.25
```

```
a + b, a - b, a * b, a / b
```

# Strings

'Single ' or "Double quotes"

```
print "A very long string with\  
a line break"
```

A very long string with a line break

```
print """Other very long string  
with line break"""
```

Other very long string  
with line break

# Strings

```
"Con" + " cat"
```

```
"Concat"
```

```
"Test" * 3
```

```
"TestTestTest"
```

# Strings

```
word = "Simple Test"  
word[3]  
"m"
```

```
word[:6]  
"Simple"
```

```
word[7:]  
"Test"
```

```
word[:i] + word[i:] == word
```

# Strings

```
word = "Simple Test"  
word[1:100]  
"Simple Test"
```

```
word[-1]  
"t"
```

```
word[-3:]  
"est"
```

# Strings

`s.endswith(suffix)` # Also with *start, end*

`s.startswith(prefix)` # Also with *start, end*

`s.find(substring)` # Also with *start, end*

# Strings

```
s.strip([chars]) # Also lstrip(), rstrip()
```

```
"testtest".strip(st) # "estte"
```

```
s.strip() # Remove whitespace
```

# Strings

```
s.split(string) # Also lsplit(), rsplit()  
# maxsplits
```

```
"1,2,,3".split(",") # ["1", "2", "", "3"]
```

```
"1 2    3".split() # ["1", "2", "3"]
```



# Strings

```
" , ".join(["1", "2", "3"]) # 1,2,3  
" , ".join("1,2,3".split(", ")) # 1,2,3  
# join is the reverse operation of split
```

# C-style Format Strings

```
"%3d, %3d, %3.3f" % (127, 8, 13.0236131)
" 127,   8,  13.024"
```

```
" Hello %s" % "World"
#=> " Hello World"
```

```
"%d + %d < 10" % (5, 4)
#=> "5 + 4 < 10"
```

# Logik

# Logik

<, >, ==, <=, >=, !=

a < b == c

**and, or, not**

FALSE: False, None, 0, 0.0, "", [], ...

# Logik

```
print 0 and "B"  
0
```

```
print "A" and False  
False
```

```
print "A" and "B"  
B
```

# Logik

```
a = someStringFunction() # Might be ""  
print a or "Default Value"
```

# Kontrollstrukturen

## IF

```
x = 4
if x < 2:
    print "x ist kleiner 2"
elif x == 2:
    print "x ist gleich 2"
else:
    print "x ist groesser 2"
print "Ende des IF"
```



# WHILE

```
a, b = 0, 1
while b < 10:
    print b
    a, b = b, a+b
```

```
1
1
2
3
5
8
```

# FOR

```
words = ["Mary", "had", "a", "little", "lamb"]  
for word in words:  
    print word
```

```
Mary  
had  
a  
little  
lamb
```

# FOR

```
for i in range(10):  
    print i
```

```
range(5, 10) #[5, 6, 7, 8, 9]
```

```
range(0, 10, 2) #[0, 2, 4, 6, 8]
```

# FOR

```
words = ["Mary", "had", "a", "little", "lamb"]  
for i in range(len(words)):  
    print i, words[i]
```

```
0 Mary  
1 had  
2 a  
3 little  
4 lamb
```

# Continue

```
for i in range(10):  
    if i % 2 == 0:  
        continue  
    print i
```

```
1  
3  
5  
7  
9
```

# Break, Else

```
for n in range(3,10):  
    for x in range(2, n):  
        if n % x == 0:  
            break  
    else:  
        print n
```

3  
5  
7

# Pass

```
for i in range(10):  
    if i % 2 == 0:  
        pass # TODO: Implement this later  
    else:  
        print i
```

0  
2  
4  
6  
8

- keyword **def** introduces a function definition
- first statement may be a **docstring**

### Defining and calling a function

```
def gcd(a,b):  
    '''Euclidean algorithm'''  
    while b != 0:  
        t, b, a = b, a%t, t  
    return a
```

```
gcd(42,23)
```