



Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python

Life Is Better Without Braces

Jochen Schulz

FH NORDAKADEMIE Elmshorn

Moderne Softwaretechnologien
26.9.2005



Gliederung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Einführung

Leckereien

Diskussion



Gliederung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Diskussion



Was ist Python?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python ist eine
▶ objektorientierte

Programmiersprache für Windows, Unix und Macintosh, OS/2,
Amiga. . .



Was ist Python?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python ist eine

- ▶ objektorientierte
- ▶ interaktive

Programmiersprache für Windows, Unix und Macintosh, OS/2, Amiga. . .



Was ist Python?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python ist eine

- ▶ objektorientierte
- ▶ interaktive
- ▶ interpretierte

Programmiersprache für Windows, Unix und Macintosh, OS/2, Amiga. . .



Und...

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python ist...

- ▶ umfangreich („batteries included“)

und macht Spaß!



Und...

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python ist...

- ▶ umfangreich („batteries included“)
- ▶ erweiterbar

und macht Spaß!



Und...

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python ist...

- ▶ umfangreich („batteries included“)
- ▶ erweiterbar
- ▶ dynamisch

und macht Spaß!



Und...

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python ist...

- ▶ umfangreich („batteries included“)
- ▶ erweiterbar
- ▶ dynamisch
- ▶ portabel

und macht Spaß!



Und...

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python ist...

- ▶ umfangreich („batteries included“)
- ▶ erweiterbar
- ▶ dynamisch
- ▶ portabel
- ▶ einfach

und macht Spaß!



Wer macht Python?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

▶ ab 1990: Guido van Rossum



Wer macht Python?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ ab 1990: Guido van Rossum
- ▶ seit Version 2.1: Python Software Foundation



Wer macht Python?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ ab 1990: Guido van Rossum
- ▶ seit Version 2.1: Python Software Foundation
- ▶ öffentliche Diskussion von *Python Enhancement Proposals* (PEPs)



Wer macht Python?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ ab 1990: Guido van Rossum
- ▶ seit Version 2.1: Python Software Foundation
- ▶ öffentliche Diskussion von *Python Enhancement Proposals* (PEPs)
- ▶ GvR „Benevolent Dictator for Life“ (BDFL)



Wer nutzt Python?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

▶ Google



Wer nutzt Python?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Google
- ▶ www.ZEIT.de (Zope, Plone)



Wer nutzt Python?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Google
- ▶ www.ZEIT.de (Zope, Plone)
- ▶ die NORDAKADEMIE (Mailman)



Wer nutzt Python?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Google
- ▶ www.ZEIT.de (Zope, Plone)
- ▶ die NORDAKADEMIE (Mailman)
- ▶ verfügbar bei vielen Webhostern



Wer nutzt Python?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Google
- ▶ www.ZEIT.de (Zope, Plone)
- ▶ die NORDAKADEMIE (Mailman)
- ▶ verfügbar bei vielen Webhostern
- ▶ MoinMoin (Wiki-Software)



Wer nutzt Python?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Google
- ▶ www.ZEIT.de (Zope, Plone)
- ▶ die NORDAKADEMIE (Mailman)
- ▶ verfügbar bei vielen Webhostern
- ▶ MoinMoin (Wiki-Software)
- ▶ Nokia (Developer Kit für Telefone der *Series 60* verfügbar)



Wer nutzt Python?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Google
- ▶ www.ZEIT.de (Zope, Plone)
- ▶ die NORDAKADEMIE (Mailman)
- ▶ verfügbar bei vielen Webhostern
- ▶ MoinMoin (Wiki-Software)
- ▶ Nokia (Developer Kit für Telefone der *Series 60* verfügbar)
- ▶ ...



Gliederung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Diskussion



Verzweigungen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
if a < b:  
    foo()
```




Verzweigungen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
if a < b:  
    foo()  
elif a > b:  
    bar()
```



Verzweigungen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
if a < b:  
    foo()  
elif a > b:  
    bar()  
else:  
    baz()
```



Verzweigungen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
if a < b:  
    foo()  
elif a > b:  
    bar()  
else:  
    baz()
```

▶ Kein case-Konstrukt vorhanden!



Schleifen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
while a < b:  
    foo()
```



Schleifen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
while a < b:  
    foo()
```

```
for elem in [-5, 23, 42]:  
    foo(elem)
```



Schleifen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
while a < b:  
    foo()
```

```
for elem in [-5, 23, 42]:  
    foo(elem)
```

- ▶ Keine *Repeat*-Schleife vorhanden!



Funktionen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
def function(param):  
    [...]  
    return something
```



Funktionen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
def function(param):  
    [...]  
    return something
```

- ▶ `return` ist optional, aber guter Stil. Soll kein Wert zurückgegeben werden, kann `return` auch allein stehen. In diesem Fall wird implizit `None` zurückgeliefert.



Funktionsparameter

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Ein Funktionsparameter kann als *optional* gekennzeichnet werden, indem ein Default-Wert angegeben wird:

```
def func(param1, param2=0):
```



Funktionsparameter

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Ein Funktionsparameter kann als *optional* gekennzeichnet werden, indem ein Default-Wert angegeben wird:

```
def func(param1, param2=0):
```

- ▶ Gültige Aufrufe von `func()`:

```
func(1, 4)
```



Funktionsparameter

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Ein Funktionsparameter kann als *optional* gekennzeichnet werden, indem ein Default-Wert angegeben wird:

```
def func(param1, param2=0):
```

- ▶ Gültige Aufrufe von `func()`:

```
func(1, 4)
func(1)    # der zweite Parameter
           # ist implizit 0
```



Funktionsparameter

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Ein Funktionsparameter kann als *optional* gekennzeichnet werden, indem ein Default-Wert angegeben wird:

```
def func(param1, param2=0):
```

- ▶ Gültige Aufrufe von `func()`:

```
func(1, 4)
func(1)    # der zweite Parameter
           # ist implizit 0
func(param2=4, param1=1)
```



Funktionsparameter

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Ein Funktionsparameter kann als *optional* gekennzeichnet werden, indem ein Default-Wert angegeben wird:

```
def func(param1, param2=0):
```

- ▶ Gültige Aufrufe von `func()`:

```
func(1, 4)
func(1)    # der zweite Parameter
           # ist implizit 0
func(param2=4, param1=1)
```

- ▶ Default-Parameter werden nur einmal ausgewertet!



Klassen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
class C(object):  
  
    classattr = 42
```



Klassen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
class C(object):  
  
    classattr = 42  
  
    def __init__(self, param):  
        self.attr = param
```



Klassen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
class C(object):  
  
    classattr = 42  
  
    def __init__(self, param):  
        self.attr = param  
  
    def method(self, param):  
        print param
```




Klassen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
class C(object):

    classattr = 42

    def __init__(self, param):
        self.attr = param

    def method(self, param):
        print param

    def __private_method(self, param):
        return
```



Objekte und Attributzugriff

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
obj = C(23)
obj.attr
```



Objekte und Attributzugriff

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
obj = C(23)
obj.attr
C.classattr
```



Objekte und Attributzugriff

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
obj = C(23)
obj.attr
C.classattr
obj.classattr
```



Objekte und Attributzugriff

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
obj = C(23)
obj.attr
C.classattr
obj.classattr
obj.method('hallo')
```



Objekte und Attributzugriff

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
obj = C(23)
obj.attr
C.classattr
obj.classattr
obj.method('hallo')
```

```
23
42
42
hallo
```



Klassen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Instanzvariablen werden generell in der `__init__`-Methode benannt – können aber jederzeit (auch außerhalb der Klasse!) zugewiesen werden.



Klassen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Instanzvariablen werden generell in der `__init__`-Methode benannt – können aber jederzeit (auch außerhalb der Klasse!) zugewiesen werden.
- ▶ Es gibt keine Klassenmethoden – stattdessen verwendet man Funktionen auf Modulebene



Klassen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Instanzvariablen werden generell in der `__init__`-Methode benannt – können aber jederzeit (auch außerhalb der Klasse!) zugewiesen werden.
- ▶ Es gibt keine Klassenmethoden – stattdessen verwendet man Funktionen auf Modulebene
- ▶ Klassenattribute können auch über Instanzen der Klasse angesprochen werden.



Klassen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Instanzvariablen werden generell in der `__init__`-Methode benannt – können aber jederzeit (auch außerhalb der Klasse!) zugewiesen werden.
- ▶ Es gibt keine Klassenmethoden – stattdessen verwendet man Funktionen auf Modulebene
- ▶ Klassenattribute können auch über Instanzen der Klasse angesprochen werden.
- ▶ Private Attribute werden durch zwei `_` zu Beginn des Attributnames gekennzeichnet. Zugriff wird nicht effektiv verhindert, sondern durch *name mangling* und Konvention *discouraged*.



Ausnahmen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

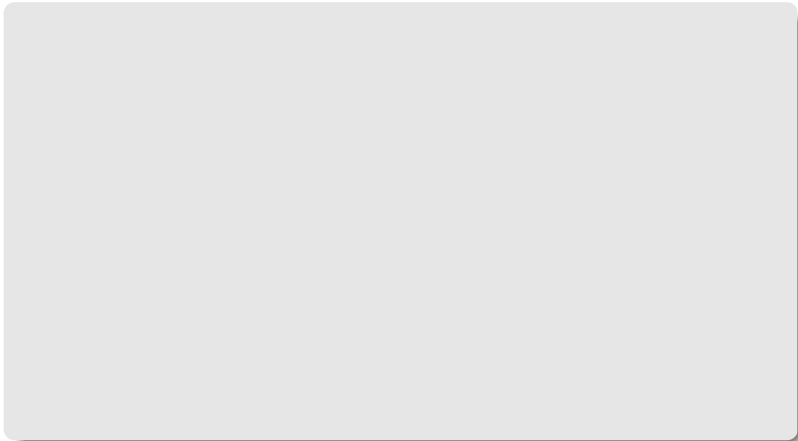
Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion





Ausnahmen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
try:  
    myfile = open('filename')
```



Ausnahmen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
try:
    myfile = open('filename')
except IOError, (no, msg):
    print 'I/O error(%s): %s' % (no, msg)
```



Ausnahmen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
try:
    myfile = open('filename')
except IOError, (no, msg):
    print 'I/O error(%s): %s' % (no, msg)
else:
    myfile.close()
```



Ausnahmen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
try:
    myfile = open('filename')
except IOError, (no, msg):
    print 'I/O error(%s): %s' % (no, msg)
else:
    myfile.close()

class MyException(Exception):
    pass
```



Ausnahmen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
try:
    myfile = open('filename')
except IOError, (no, msg):
    print 'I/O error(%s): %s' % (no, msg)
else:
    myfile.close()

class MyException(Exception):
    pass

raise MyException, 'Mein Fehler!'
```




Gliederung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Diskussion



None

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Repräsentiert „Nichts“, ähnlich `null` in Java



None

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Repräsentiert „Nichts“, ähnlich `null` in Java
- ▶ Ist ein *Singleton*, d.h. es gibt immer nur ein Objekt dieses Typs (ermöglicht Prüfung auf Objektidentität: `foo is None`)



Zahlen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ `bool`: `True` oder `False`. Es existiert jeweils immer nur ein Objekt beider Werte



Zahlen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ `bool`: True oder False. Es existiert jeweils immer nur ein Objekt beider Werte
- ▶ `int`: ganze Zahlen
Literal: 1 10 32391



Zahlen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ `bool`: `True` oder `False`. Es existiert jeweils immer nur ein Objekt beider Werte
- ▶ `int`: ganze Zahlen
Literal: `1 10 32391`
- ▶ `long`: große ganze Zahlen, beliebig lang
Literal: `53847874454325324523L`



Zahlen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ `bool`: `True` oder `False`. Es existiert jeweils immer nur ein Objekt beider Werte
- ▶ `int`: ganze Zahlen
Literal: `1 10 32391`
- ▶ `long`: große ganze Zahlen, beliebig lang
Literal: `53847874454325324523L`
- ▶ `float`: Fließkommazahlen
Literal: `0.5`



Zahlen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ `bool`: True oder False. Es existiert jeweils immer nur ein Objekt beider Werte
- ▶ `int`: ganze Zahlen
Literal: 1 10 32391
- ▶ `long`: große ganze Zahlen, beliebig lang
Literal: 53847874454325324523L
- ▶ `float`: Fließkommazahlen
Literal: 0.5
- ▶ `complex` : Fließkommazahl + Imaginärteil
Literal: (3.5+23.7J)



Sequenzen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ `tuple`: Listen fester Länge (immutable)
Literal: `(a, b, c)`



Sequenzen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ `tuple`: Listen fester Länge (immutable)
Literal: `(a, b, c)`
- ▶ `str` : Zeichenketten (intern Tupel, also auch immutable)
Literal: `'abc'` `“abc”` `‘‘‘abc’’’`



Sequenzen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ `tuple`: Listen fester Länge (immutable)
Literal: `(a, b, c)`
- ▶ `str` : Zeichenketten (intern Tupel, also auch immutable)
Literal: `'abc'` `''abc''` `''''''abc''''''`
- ▶ `list`: Listen variabler Länge beliebigen Inhalts.
Literal: `[a, b, c]`



Sequenzen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ `tuple`: Listen fester Länge (immutable)
Literal: `(a, b, c)`
- ▶ `str` : Zeichenketten (intern Tupel, also auch immutable)
Literal: `'abc'` `''abc''` `''''''abc''''''`
- ▶ `list`: Listen variabler Länge beliebigen Inhalts.
Literal: `[a, b, c]`
- ▶ Listenzugriff per Index, Zählung beginnt bei Null. Wird ein negativer Index angegeben, wird die Länge der Liste aufaddiert.



Sequenzen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ `tuple`: Listen fester Länge (immutable)
Literal: `(a, b, c)`
- ▶ `str` : Zeichenketten (intern Tupel, also auch immutable)
Literal: `'abc'` `''abc''` `''''''abc''''''`
- ▶ `list`: Listen variabler Länge beliebigen Inhalts.
Literal: `[a, b, c]`
- ▶ Listenzugriff per Index, Zählung beginnt bei Null. Wird ein negativer Index angegeben, wird die Länge der Liste aufaddiert.
- ▶ Zugriff auf Ausschnitte über *Slicing*.



Listenzugriff

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
>>> a = [1, 2, 3, 4]
```



Listenzugriff

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
>>> a = [1, 2, 3, 4]
```

```
>>> a[0]
```

```
1
```



Listenzugriff

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
>>> a = [1, 2, 3, 4]
```

```
>>> a[0]
```

```
1
```

```
>>> a[-1]
```

```
4
```




Listenzugriff

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
>>> a = [1, 2, 3, 4]
```

```
>>> a[0]
```

```
1
```

```
>>> a[-1]
```

```
4
```

```
>>> a[1:3]
```

```
[2, 3]
```



Listenzugriff

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
>>> a = [1, 2, 3, 4]
```

```
>>> a[0]
```

```
1
```

```
>>> a[-1]
```

```
4
```

```
>>> a[1:3]
```

```
[2, 3]
```

```
>>> a[2:]
```

```
[3, 4]
```



Listenzugriff

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
>>> a = [1, 2, 3, 4]
>>> a[0]
1
>>> a[-1]
4
>>> a[1:3]
[2, 3]
>>> a[2:]
[3, 4]
>>> a[:]          # flache Kopie der Liste
[1, 2, 3, 4]
```



Mappings

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

► Einziger Mapping-Type: dict



Mappings

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Einziger Mapping-Type: `dict`
- ▶ Dictionaries verhalten sich wie Sequenzen, allerdings kann statt über Indizes über (fast) beliebige Keys auf die Elemente zugegriffen werden.



Mappings

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Einziger Mapping-Type: `dict`
- ▶ Dictionaries verhalten sich wie Sequenzen, allerdings kann statt über Indizes über (fast) beliebige Keys auf die Elemente zugegriffen werden.
- ▶ Einschränkung: Keys müssen immutable sein.



Mappings

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Einziger Mapping-Type: `dict`
- ▶ Dictionaries verhalten sich wie Sequenzen, allerdings kann statt über Indizes über (fast) beliebige Keys auf die Elemente zugegriffen werden.
- ▶ Einschränkung: Keys müssen immutable sein.
- ▶ Die Reihenfolge der Elemente ist undefiniert – damit ist Slicing natürlich unmöglich.



Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
d = { 'i02b79' : Student('Meier'),  
      'i02b80' : Student('Petersen') }
```




Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
d = { 'i02b79' : Student('Meier'),  
      'i02b80' : Student('Petersen') }  
print d['i02b79']
```



Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
d = { 'i02b79' : Student('Meier'),  
      'i02b80' : Student('Petersen') }  
print d['i02b79']  
print d['abcdef']
```



Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
d = { 'i02b79' : Student('Meier'),  
      'i02b80' : Student('Petersen') }  
print d['i02b79']  
print d['abcdef']
```

```
Student Meier
```



Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
d = { 'i02b79' : Student('Meier'),  
      'i02b80' : Student('Petersen') }  
print d['i02b79']  
print d['abcdef']
```

```
Student Meier  
Traceback (most recent call last):  
File "<stdin>", line 1, in ?  
KeyError: 'abcdef'
```



Gliederung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Einführung

Leckereien

Diskussion



Gliederung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Einführung

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion



Die Funktion `filter()`

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

▶ gleiche Semantik wie in *Lisp*



Die Funktion `filter()`

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ gleiche Semantik wie in *Lisp*
- ▶ Rückgabewert sind alle Elemente einer Liste, für die eine gegebene Funktion `True` zurückliefert.



Die Funktion `filter()`

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ gleiche Semantik wie in *Lisp*
- ▶ Rückgabewert sind alle Elemente einer Liste, für die eine gegebene Funktion `True` zurückliefert.
- ▶ Aufruf: `filter(function, list)`



filter() Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
f = lambda x: x % 2 == 0
```



filter() Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
f = lambda x: x % 2 == 0
l = (34, 21, 432, 3, 4)
```



filter() Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
f = lambda x: x % 2 == 0
l = (34, 21, 432, 3, 4)

print filter(f, l)
```



filter() Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
f = lambda x: x % 2 == 0  
l = (34, 21, 432, 3, 4)
```

```
print filter(f, l)
```

```
(34, 432, 4)
```



Die Funktion map()

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

▶ gleiche Semantik wie in *Lisp*



Die Funktion map()

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ gleiche Semantik wie in *Lisp*
- ▶ Rückgabewert ist eine neue Liste mit dem Ergebnis einer gegebenen Funktion für jedes Element einer Liste.



Die Funktion `map()`

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ gleiche Semantik wie in *Lisp*
- ▶ Rückgabewert ist eine neue Liste mit dem Ergebnis einer gegebenen Funktion für jedes Element einer Liste.
- ▶ Aufruf: `map(function, list)`



map() Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
f = lambda x: x % 2 == 0
```



map() Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
f = lambda x: x % 2 == 0
l = (34, 21, 432, 3, 4)
```



map() Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
f = lambda x: x % 2 == 0
l = (34, 21, 432, 3, 4)
```

```
print map(f, l)
```



map() Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
f = lambda x: x % 2 == 0
l = (34, 21, 432, 3, 4)
```

```
print map(f, l)
```

```
(True, False, True, False, True)
```



Die Funktion reduce()

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ gleiche Semantik wie in *Lisp*



Die Funktion reduce()

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ gleiche Semantik wie in *Lisp*
- ▶ Reduziert eine Liste auf einen einzelnen Wert. Es wird eine gegebene Funktion von links nach rechts auf alle Elemente einer Liste angewandt.



Die Funktion reduce()

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ gleiche Semantik wie in *Lisp*
- ▶ Reduziert eine Liste auf einen einzelnen Wert. Es wird eine gegebene Funktion von links nach rechts auf alle Elemente einer Liste angewandt.
- ▶ Aufruf:
`reduce(function, list[, initial])`



reduce() Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
f = lambda x, y: x + y
```




reduce() Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
f = lambda x, y: x + y
l = (34, 21, 432, 3, 4)
```



reduce() Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
f = lambda x, y: x + y
l = (34, 21, 432, 3, 4)

print reduce(f, l)
```



reduce() Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
f = lambda x, y: x + y
```

```
l = (34, 21, 432, 3, 4)
```

```
print reduce(f, l)
```

```
print reduce(f, (), 5)
```



reduce() Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
f = lambda x, y: x + y
l = (34, 21, 432, 3, 4)
```

```
print reduce(f, l)
print reduce(f, (), 5)
```

```
((((34+21) + 432) + 3) + 4) = 494
5
```



Hinweise zur Benutzung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ `filter()`, `map()`, `reduce()` und `lambda` werden wahrscheinlich in Python 3.0 nicht mehr als built-in verfügbar sein (PEP 3000).



Hinweise zur Benutzung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ `filter()`, `map()`, `reduce()` und `lambda` werden wahrscheinlich in Python 3.0 nicht mehr als built-in verfügbar sein (PEP 3000).
- ▶ Statt komplizierter Lambda-Ausdrücke benutzt man besser Inline-Funktionen.



Hinweise zur Benutzung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ `filter()`, `map()`, `reduce()` und `lambda` werden wahrscheinlich in Python 3.0 nicht mehr als built-in verfügbar sein (PEP 3000).
- ▶ Statt komplizierter Lambda-Ausdrücke benutzt man besser Inline-Funktionen.
- ▶ Für den trivialen Fall `reduce(lambda x, y: x+y, list, 0)` kann man auch die eingebaute Funktion `sum(list)` benutzen.



List Comprehensions

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Ersetzen `filter()` und `map()` und vollständig



List Comprehensions

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Ersetzen `filter()` und `map()` und vollständig
- ▶ Haben ihren Ursprung in *Haskell*



List Comprehensions

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Ersetzen `filter()` und `map()` und vollständig
- ▶ Haben ihren Ursprung in *Haskell*
- ▶ Sind oft besser lesbar und flexibler als die obigen Methoden



List Comprehensions

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Ersetzen `filter()` und `map()` und vollständig
- ▶ Haben ihren Ursprung in *Haskell*
- ▶ Sind oft besser lesbar und flexibler als die obigen Methoden
- ▶ Form:

```
[expr for elem in list [condition]]
```



List Comprehensions: Beispiel I

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
l = (34, 21, 432, 3, 4)
```



List Comprehensions: Beispiel I

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
l = (34, 21, 432, 3, 4)
```

```
[x for x in l if x % 2 == 0]
```



List Comprehensions: Beispiel I

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
l = (34, 21, 432, 3, 4)
```

```
[x for x in l if x % 2 == 0]
```

```
(34, 432, 4)
```

- ▶ Entspricht genau der Anwendung von `filter()`



List Comprehensions: Beispiel II

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
l = (34, 21, 432, 3, 4)
```



List Comprehensions: Beispiel II

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
l = (34, 21, 432, 3, 4)
```

```
[x % 2 == 0 for x in l]
```




List Comprehensions: Beispiel II

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
l = (34, 21, 432, 3, 4)
```

```
[x % 2 == 0 for x in l]
```

```
(True, False, True, False, True)
```

- ▶ Entspricht genau der Anwendung von `map()`



Gliederung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Einführung

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion



Generator-Funktionen: Beispiel I

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
def reverse(seq):  
    for index in range(len(seq)-1, -1, -1):  
        yield seq[index]
```



Generator-Funktionen: Beispiel I

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
def reverse(seq):  
    for index in range(len(seq)-1, -1, -1):  
        yield seq[index]  
  
for char in reverse('Lotti'):  
    print char,
```



Generator-Funktionen: Beispiel I

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
def reverse(seq):  
    for index in range(len(seq)-1, -1, -1):  
        yield seq[index]
```

```
for char in reverse('Lotti'):  
    print char,
```

i t t o l



Generator-Funktionen: Beispiel II

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
def fibonacci():  
    a, b = 0, 1  
    while True:  
        yield a  
        a, b = b, a+b
```



Generator-Funktionen: Beispiel II

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
def fibonacci():
    a, b = 0, 1
    while True:
        yield a
        a, b = b, a+b

iterator = iter(fibonacci())
for i in range(0, 13):
    print iterator.next()
```



Generator-Funktionen: Beispiel II

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
def fibonacci():  
    a, b = 0, 1  
    while True:  
        yield a  
        a, b = b, a+b  
  
iterator = iter(fibonacci())  
for i in range(0, 13):  
    print iterator.next()
```

0 1 1 2 3 5 8 13 21 34 55 89 144



Generator-Funktionen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Generatoren sind zustandsbehaftete Funktionen.



Generator-Funktionen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Generatoren sind zustandsbehaftete Funktionen.
- ▶ Statt `return` geben sie Ihr Ergebnis mit `yield` zurück.



Generator-Funktionen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Generatoren sind zustandsbehaftete Funktionen.
- ▶ Statt `return` geben sie Ihr Ergebnis mit `yield` zurück.
- ▶ Wiederholte Aufrufe können unterschiedliche Ergebnisse liefern.



Generator-Funktionen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Generatoren sind zustandsbehaftete Funktionen.
- ▶ Statt `return` geben sie Ihr Ergebnis mit `yield` zurück.
- ▶ Wiederholte Aufrufe können unterschiedliche Ergebnisse liefern.
- ▶ Generatoren können beliebig viele Elemente zurückgeben.



Generator-Funktionen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Generatoren sind zustandsbehaftete Funktionen.
- ▶ Statt `return` geben sie Ihr Ergebnis mit `yield` zurück.
- ▶ Wiederholte Aufrufe können unterschiedliche Ergebnisse liefern.
- ▶ Generatoren können beliebig viele Elemente zurückgeben.
- ▶ Im Gegensatz zu List Comprehensions erzeugen sie nicht eine ganze Liste, sondern jedes Element einzeln (weniger Speicherbedarf!).



Generator-Expressions

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Eine Mischung aus List Comprehensions und
Generator-Funktionen:



Generator-Expressions

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Eine Mischung aus List Comprehensions und
Generator-Funktionen:

- ▶ Erzeugen ein Element zur Zeit



Generator-Expressions

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Eine Mischung aus List Comprehensions und Generator-Funktionen:

- ▶ Erzeugen ein Element zur Zeit
- ▶ Haben eine einfache Syntax zur Erzeugung:

```
(x**2 for x in (2, 6, 12, 32))
```




Generator-Expressions

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Eine Mischung aus List Comprehensions und Generator-Funktionen:

- ▶ Erzeugen ein Element zur Zeit
- ▶ Haben eine einfache Syntax zur Erzeugung:

```
(x**2 for x in (2, 6, 12, 32))
```

```
[4, 36, 144, 1024]
```



Gliederung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Einführung

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion



Simulation von built-in Types

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Das Verhalten von Objekten bei Anwendung der Standardsyntax kann angepasst werden:



Simulation von built-in Types

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Das Verhalten von Objekten bei Anwendung der Standardsyntax kann angepasst werden:

▶ Vergleiche:

< > <= >= == !=



Simulation von built-in Types

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Das Verhalten von Objekten bei Anwendung der Standardsyntax kann angepasst werden:

▶ Vergleiche:

< > <= >= == !=

▶ arithmetische Operationen:

+ - * / % and or ...



Simulation von built-in Types

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Das Verhalten von Objekten bei Anwendung der Standardsyntax kann angepasst werden:

- ▶ Vergleiche:
`< > <= >= == !=`
- ▶ arithmetische Operationen:
`+ - * / % and or ...`
- ▶ Zugriff per Index oder Key:
`obj[idx] del obj[idx]`



Simulation von built-in Types

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Das Verhalten von Objekten bei Anwendung der Standardsyntax kann angepasst werden:

- ▶ Vergleiche:

`< > <= >= == !=`

- ▶ arithmetische Operationen:

`+ - * / % and or ...`

- ▶ Zugriff per Index oder Key:

`obj[idx] del obj[idx]`

- ▶ Iteration:

`for elem in obj`



Vergleiche: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
class Rectangle(object):  
  
    def __init__(self, x, y):  
        self.x = x  
        self.y = y
```




Vergleiche: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
class Rectangle(object):

    def __init__(self, x, y):
        self.x = x
        self.y = y

    def area():
        return self.x * self.y
```



Vergleiche: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
class Rectangle(object):

    def __init__(self, x, y):
        self.x = x
        self.y = y

    def area():
        return self.x * self.y

    def __lt__(self, other):
        return self.area() < other.area()
```



Vergleiche: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
class Rectangle(object):

    def __init__(self, x, y):
        self.x = x
        self.y = y

    def area():
        return self.x * self.y

    def __lt__(self, other):
        return self.area() < other.area()

    def __gt__(self, other):
        return self.area() > other.area()
```



Vergleiche: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
a = Rectangle(2, 4)
b = Rectangle(4, 3)
print a < b
print a > b
```



Vergleiche: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
a = Rectangle(2, 4)
b = Rectangle(4, 3)
print a < b
print a > b
```

True



Vergleiche: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
a = Rectangle(2, 4)
b = Rectangle(4, 3)
print a < b
print a > b
```

```
True
False
```



Arithmetik: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
class Number(object):  
  
    def __init__(self, value):  
        self.value = value
```



Arithmetik: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
class Number(object):  
  
    def __init__(self, value):  
        self.value = value  
  
    def __add__(self, other):  
        return Number(self.value + other.value)
```




Arithmetik: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
class Number(object):  
  
    def __init__(self, value):  
        self.value = value  
  
    def __add__(self, other):  
        return Number(self.value + other.value)  
  
    def __sub__(self, other):  
        return Number(self.value - other.value)
```



Arithmetik: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
a = Number(5)
```

```
b = Number(7)
```



Arithmetik: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
a = Number(5)
```

```
b = Number(7)
```

```
c = a + b
```

```
d = a - b
```



Arithmetik: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
a = Number(5)
b = Number(7)
c = a + b
d = a - b
print type(c)
print c.value
print d.value
```



Arithmetik: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
a = Number(5)
b = Number(7)
c = a + b
d = a - b
print type(c)
print c.value
print d.value
```

```
<class '__main__.Number'>
```



Arithmetik: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
a = Number(5)
b = Number(7)
c = a + b
d = a - b
print type(c)
print c.value
print d.value
```

```
<class '__main__.Number'>
12
```



Arithmetik: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
a = Number(5)
b = Number(7)
c = a + b
d = a - b
print type(c)
print c.value
print d.value
```

```
<class '__main__.Number'>
12
-2
```



Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
class Zenturie(object):  
  
    def __init__(self):  
        self.studenten = {}
```




Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
class Zenturie(object):  
  
    def __init__(self):  
        self.studenten = {}  
  
    def __len__(self):  
        return len(self.studenten)
```



Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
class Zenturie(object):

    def __init__(self):
        self.studenten = {}

    def __len__(self):
        return len(self.studenten)

    def __iter__(self):
        for key in self.studenten.iterkeys():
            yield key
```



Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
def __getitem__(self, key):  
    return self.studenten[key]
```



Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
def __getitem__(self, key):  
    return self.studenten[key]  
  
def __setitem__(self, key, value):  
    self.studenten[key] = value
```



Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
def __getitem__(self, key):  
    return self.studenten[key]  
  
def __setitem__(self, key, value):  
    self.studenten[key] = value  
  
def __delitem__(self, key)  
    del self.studenten[key]
```



Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
myZenturie = Zenturie()
```



Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
myZenturie = Zenturie()
```

```
myZenturie['1862'] = 'Fuerst v. Bismarck'
```

```
myZenturie['1949'] = 'Konrad Adenauer'
```



Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
myZenturie = Zenturie()
```

```
myZenturie['1862'] = 'Fuerst v. Bismarck'
```

```
myZenturie['1949'] = 'Konrad Adenauer'
```

```
for student in myZenturie:  
    print myZenturie[student]
```




Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
myZenturie = Zenturie()
```

```
myZenturie['1862'] = 'Fuerst v. Bismarck'
```

```
myZenturie['1949'] = 'Konrad Adenauer'
```

```
for student in myZenturie:  
    print myZenturie[student]
```

```
Fuerst v. Bismarck  
Konrad Adenauer
```



Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
print len(myZenturie)
```



Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
print len(myZenturie)

del MyZenturie['1862']
```



Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
print len(myZenturie)

del MyZenturie['1862']

print len(myZenturie)
```



Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
print len(myZenturie)

del MyZenturie['1862']

print len(myZenturie)
```

2



Listenverhalten: Beispiel

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

```
print len(myZenturie)

del MyZenturie['1862']

print len(myZenturie)
```

2

1



Gliederung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Einführung

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion



Glade

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Demo



Gliederung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Einführung

Leckereien

Diskussion



Zusammenfassung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python hat Schwächen:

- ▶ keine Typsicherheit



Zusammenfassung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python hat Schwächen:

- ▶ keine Typsicherheit
- ▶ Performance



Zusammenfassung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python hat Schwächen:

- ▶ keine Typsicherheit
- ▶ Performance
- ▶ keine wirklich privaten Attribute



Zusammenfassung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python hat Schwächen:

- ▶ keine Typsicherheit
- ▶ Performance
- ▶ keine wirklich privaten Attribute
- ▶ keine Interfaces



Zusammenfassung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python hat Stärken:

- ▶ Keine Typsicherheit :)



Zusammenfassung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python hat Stärken:

- ▶ Keine Typsicherheit :)
- ▶ Schnell zu erlernen



Zusammenfassung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python hat Stärken:

- ▶ Keine Typsicherheit :)
- ▶ Schnell zu erlernen
- ▶ Hohe Produktivität



Zusammenfassung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python hat Stärken:

- ▶ Keine Typsicherheit :)
- ▶ Schnell zu erlernen
- ▶ Hohe Produktivität
- ▶ Sehr gute Lesbarkeit



Zusammenfassung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python hat Stärken:

- ▶ Keine Typsicherheit :)
- ▶ Schnell zu erlernen
- ▶ Hohe Produktivität
- ▶ Sehr gute Lesbarkeit
- ▶ Portabilität



Zusammenfassung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python hat Stärken:

- ▶ Keine Typsicherheit :)
- ▶ Schnell zu erlernen
- ▶ Hohe Produktivität
- ▶ Sehr gute Lesbarkeit
- ▶ Portabilität
- ▶ Dynamik



Zusammenfassung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python hat Stärken:

- ▶ Keine Typsicherheit :)
- ▶ Schnell zu erlernen
- ▶ Hohe Produktivität
- ▶ Sehr gute Lesbarkeit
- ▶ Portabilität
- ▶ Dynamik
- ▶ Offene Entwicklung, freie Dokumentation



Zusammenfassung

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Python hat Stärken:

- ▶ Keine Typsicherheit :)
- ▶ Schnell zu erlernen
- ▶ Hohe Produktivität
- ▶ Sehr gute Lesbarkeit
- ▶ Portabilität
- ▶ Dynamik
- ▶ Offene Entwicklung, freie Dokumentation
- ▶ Kostenfrei



Wofür kann ich Python nicht benutzen?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Anwendungen, bei denen Performance entscheidend ist



Wofür kann ich Python nicht benutzen?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Anwendungen, bei denen Performance entscheidend ist
- ▶ Anwendungen auf Embedded Devices



Wofür kann ich Python nicht benutzen?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Anwendungen, bei denen Performance entscheidend ist
- ▶ Anwendungen auf Embedded Devices
- ▶ Hardwarenahe Programmierung



Wofür kann ich Python benutzen?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Ersatz für Shellskripte



Wofür kann ich Python benutzen?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Ersatz für Shellskripte
- ▶ CGI-Programme



Wofür kann ich Python benutzen?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Ersatz für Shellskripte
- ▶ CGI-Programme
- ▶ Portable Anwendungen



Wofür kann ich Python benutzen?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Ersatz für Shellskripte
- ▶ CGI-Programme
- ▶ Portable Anwendungen
- ▶ Rapid Prototyping



Wofür kann ich Python benutzen?

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ Ersatz für Shellskripte
- ▶ CGI-Programme
- ▶ Portable Anwendungen
- ▶ Rapid Prototyping
- ▶ Anwendungen zwischen 10 und 10.000.000 Zeilen



Anlaufstellen

Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

- ▶ docs.python.org – offizielle Dokumentation, inkl. Tutorial, Sprachbeschreibung und Moduldokumentation
- ▶ DiveIntoPython.org – sehr gutes Buch für Python-Einsteiger mit Programmiererfahrung (auch gedruckt erhältlich)
- ▶ CafePy.com – enthält einige gute Artikel über Python-Internas



Python

Jochen Schulz

Einführung

Wer, Wo, Was?

Syntax

Typen

Leckereien

Funktionale Elemente

Generators

Operator Overloading

Glade

Diskussion

Fragen?