Scriptsprachen

Python Basics

Sascha Winter

Lehrstuhl fuer Bioinformatik
Friedrich-Schiller-Universitaet Jena
sascha.winter@uni-jena.de

12.08.2014

Errors and Exceptions

Errors and Exceptions

```
for arg in sys.argv[1:]:
    f = open(arg, 'r')
    print arg, 'has', len(f.readlines()), 'line
    f.close()

# Stops at first file it fails to open
```

Errors and Exceptions

```
for arg in sys.argv[1:]:
    try:
        f = open(arg, 'r')
    except IOError:
        print 'cannot open', arg
    else:
        print arg, 'has', len(f.readlines()), '
        f.close()
```

Errors and Exceptions

```
dict[key] # may raise KeyError

# Default value, kein KeyError
__missing__(key)

int(string) # may raise ValueError
```

Errors and Exceptions

```
except (RuntimeError, TypeError):
    pass

# Not except RuntimeError, TypeError: !
```

Standard Libraries

OS

```
import os

os.getcwd() # working dir

os.chdir('/var/log') # change dir

os.system('mkdir today') # bash commands

os.path # functions on pathnames
```

shutil

```
import shutil

shutil.copy(src, dst) # copy file
# if dst is dir, copy into dir

shutil.move(src, dst) # move file or dir
```

glob

```
import glob

glob.glob(pathname)
# Return list of pathnames that match pathname
# Bash style wildcards

# /usr/src/Python-1.5/Makefile

# ../../Tools/*/*.gif
```

Regular Expressions

```
import re
# Perl style regex
result = re.match(pattern, string)

re.split(pattern, string) # Split at regex
```

Regular Expressions

```
import re

# If regex is reused
matcher = re.compile(pattern, flags=0)
# flags: re.I (ignore case), ...

result = matcher.match(string)
# At the begining of string

otherResult = matcher.search(string)
# Anywhere in string
```

Regular Expressions

```
import re

matcher = re.compile("(\d+) (\w+)")

result = matcher.matches("1000 donuts")

if result:
    print result.group(0) # entire match
    print result.group(1) # first group
    print result.group(2) # second group
```

```
"1000 donuts"
"1000"
"donuts"
```

Regular Expressions

```
import re

matcher = re.compile("(\d+) (\w+)")

result = matcher.matches("1000 donuts")

if result:
    print result.start(0) # entire match
    print result.start(1) # first group
    print result.start(2) # second group
```

```
0
0
5
```

Classes

Classes

```
class MyClass:
    """A simple example class"""
    i = 12345
    def f(self):
        return 'hello world'

x = MyClass()

print x.i
12345

x.f()
hello world
```

Classes

```
class Complex:
    def __init__(self, real, imag):
        self.r = real
        self.i = imag

x = Complex(3.14, -1)
```

Inheritance

```
class MyList(list):
    listName = "My list"

x = MyList([1, 2, 3, 4])

print x
[1, 2, 3, 4]

print x.listName
My list
```

Multiple Inheritance

```
class DerivedClass(Base1, Base2):
    pass

# Old style:
# Search attribute in Base1 + base classes
# then in Base2 + base classes

# New style different
```