

NUMPY AND SCIPY PACKAGES

SEBASTIAN HAHN

1 WHAT ARE PACKAGES

1.1 PACKAGES AND MODULES

Module

.py file that contains functions/classes or variables

Package

Collection of python modules. Can be nested in folders

1.2 FOLDERS ON FILESYSTEM

- my_package
 - __init__.py
 - my_module.py

`__init__.py` tells python that the folder contains modules
code in `init.py` is executed during import!

<http://docs.python.org/2/tutorial/modules.html>

2 USING PACKAGES

- Packages have to be imported for the program to be able to use them

```
import math # import complete module
import very_long_package_name as vlp # shorten the name
from datetime import date # import only certain variables/classes/functions
```

- Avoid **from package import *** -> dirty namespace

Example - Open WinPython Command Prompt.exe or
Shell

3 IMPORTANT STANDARD LIBRARY PACKAGES

3.1 DATETIME MODULE

Representation of dates and times. [Documentation.](#)

```
from datetime import date
a = date(2015, 3, 1)
b = date.today()
print(a, b)
print(a.year, a.month, a.day) # attribute access
delta = b - a # difference are a datetime.timedelta object
print(delta)
print(type(delta))
```

```
(datetime.date(2015, 3, 1), datetime.date(2015, 4, 1))
(2015, 3, 1)
31 days, 0:00:00
<type 'datetime.timedelta'>
```

3.1.1 DATETIME

```
from datetime import date, datetime, timedelta, time
a = date(2008, 5, 1)
b = time(12, 30, 15)
print(datetime.combine(a, b))
print(datetime(2008, 5, 1, 12, 30, 16))
# parsing and formatting
dt = datetime.strptime("21/11/06 16:30", "%d/%m/%y %H:%M")
print(dt)
print(dt.strftime("%A, %d. %B %Y %I:%M%p"))
```

```
2008-05-01 12:30:15
2008-05-01 12:30:16
2006-11-21 16:30:00
Tuesday, 21. November 2006 04:30PM
```

Formatting codes

There is also a `calendar` module for e.g. leap year checking, getting days in a month.

3.2 MATH MODULE

Mathematical functions and constants.

```
import math
print(math.pi, math.e)
print(math.radians(180), math.degrees(2 * math.pi))
print(math.sin(math.pi / 4))
```

```
(3.141592653589793, 2.718281828459045)
(3.141592653589793, 360.0)
0.707106781187
```

[Docs.](#)

These is also cmath for complex numbers.