

Python Workshop

Martin Bellgardt

April 17, 2014

1 Syntax

Python-syntax orientiert sich an natürlicher Sprache. Anstatt Klammern wird Einrückung verwendet um Blöcke zu definieren.

```
import somePackage                # Ein Paket einbinden
from someOtherPackage import someFunction  # Nur eine Funktion
                                      # einbinden

def someFunction(arg1, arg2):      # eine Funktion definieren
    doStuff()                      # eine Funktion aufrufen

answer = 42                        # Integer
almostAnswer = 41.99              # Float
text = "This is a string"         # String

list = [1, 2, [3, 4], "Other stuff", 12.5]  # Liste
    # Eine Liste ist nicht auf einen Typen festgelegt

dictionary = {"Cheese" : 1, "Bacon" : 12.5}
    # Eine sehr nützliche Datenstruktur in Python

for i in list:                    # i wird ueber alle Listenelemente iteriert
    if i == 1 and answer > almostAnswer:  # If abfrage
        print "Viel Spass auf der KISS"   # Ausgabe
        print "Zahl" + str(12.5)          # str() macht strings
                                           # aus anderen Datentypen

file = open("someFile", "r")      # open some file for reading
for line in file:                 # For every line of the file
    print line                    # print the line
file.close()                     # close the file
```

2 Nützliche Pakete

Hier sind nur einige Beispiele für nützliche Pakete. Es gibt Python Pakete für so ziemlich alles. Im zweifelsfall einfach mal Google fragen.

- `sys` - Ermöglicht z.B. zugriff auf Programmargumente mit `sys.argv`
- `os` - Stellt funktionen bereit um mit Dateien und Pfaden zu arbeiten (löschen, verschieben, verlinken, etc.)
- `PIL (Python Image Library)` - Viele nützliche Funktionen zum laden, speichern und manipulieren von Bildern. (benutzen mit `import Image`)
- `matplotlib.pyplot` - Generiert Diagramme. Nützlich um Daten graphisch auszuwerten.
- `numpy, scipy` - Stellt fortgeschrittene mathematische funktionen und Strukturen bereit wie z.B. Matrizen, Vektoren etc.
- `pyffmpeg` - Wrapper für ffmpeg. Nützlich zum arbeiten mit Videos.
- `pyopencv` - Wrapper für OpenCV. Implementiert Algorithmen zur Bilderkennung.
- `urllib2` - Kann z.B. Dateien über HTTP herunterladen. Nützlich um z.B. auf web-APIs zuzugreifen.
- `antigravity` - Implementiert Levitation, sodass man nicht mehr auf überfüllte ASEAG Busse angewiesen ist.

3 Python Image Library

```
import Image
im = Image.open("image.png")           # Das Bild image.png wird geladen

r,g,b = im.getpixel((1,2))             # RGB werte fuer den Pixel an der stelle (1,2)
r,g,b,a = im.getpixel((1,2))           # oder RGBA fuer z.B. PNG bilder

width, height = im.size                 # breite und hoehe des bildes
```