

Inleiding tot Python

Brecht Baeten¹

¹KU Leuven, Technologie campus Diepenbeek,
e-mail: brecht.baeten@kuleuven.be

29 september 2016

Wat is Python?



- Progameertaal
- Z  r verscheiden toepassingsgebied (Wetenschappelijke berekeningen, Web toepassingen, GUIs, quick scripting)
- Zeer object geori  nteerd
- Packages voor wetenschappelijke toepassingen
- Gratis
- Verschillende GUI's beschikbaar, niet meegeleverd

Mac

- 2.7 standaard meegeleverd bij Mac OS X

Linux

- 2.7 standaard meegeleverd bij veel distributies
- 3.5 beschikbaar in de meeste package managers

```
apt-get install python3
```

Installatie

Windows

- WinPython <https://winpython.github.io/>

of

- Anaconda <https://www.continuum.io/downloads>
- Python(x,y) <http://python-xy.github.io/>

of

- Download een distributie (best 2.7 of 3.5) via <https://www.python.org/downloads/>
- Voeg Python toe aan het windows pad
- Download een degelijke text editor, bv. Notepad ++ <https://notepad-plus-plus.org/download/>
- Download een degelijke console, vb. ConEmu <http://sourceforge.net/projects/conemu/files/latest/download>

Spyder

The screenshot displays the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.5)". The top menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Tools, View, and Help. The toolbar contains icons for file operations and execution. The editor pane shows a file named "untitled0.py" with the following content:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Thu Sep 29 09:49:29 2016
4
5 @author: u0076268
6 """
7
8
```

The "File explorer" pane on the right shows a list of files and folders in the directory "t python/examples". The files are:

Name	Size	Type	Date
my module		File Folder	18/05
.gitignore	5 bytes	gitignore File	15/09
__init__.py	0 bytes	py File	16/09
csvreader.py	212 bytes	py File	16/09
data.csv	112 bytes	csv File	16/09
exercise.py	1 KB	py File	27/09
flow_control.py	106 bytes	py File	17/09
flow_control_advanced.py	382 bytes	py File	31/05
functions.py	334 bytes	py File	31/05
hello_world.py	21 bytes	py File	14/10

The "IPython console" pane at the bottom shows the IPython prompt and the following output:

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55)
[MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more
information.

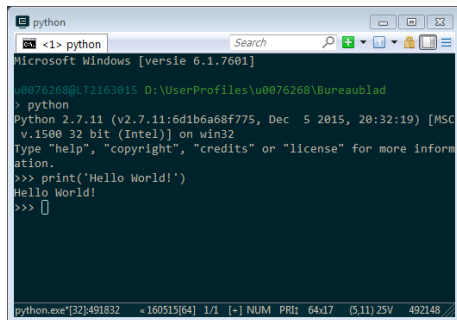
IPython 5.1.0 -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's
features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for
extra details.

In [1]: print('Hello World!')
```

The status bar at the bottom shows: Permissions: RW, End-of-lines: CRLF, Encoding: UTF-8, Line: 8, Column: 1, Memory: 75 %.

Interactieve console

- Open een console en typ "python"
- Commando's uitvoeren
- variabelen definiëren
- functies aanroepen
- Goed voor korte tests
- Sluiten met Ctrl + Z



```
python
C:\> python
Microsoft Windows [versie 6.1.7601]
u0076268@LT2163015 D:\UserProfiles\u0076268\Bureaublad
> python
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec 5 2015, 20:32:19) [MSC
v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more inform
ation.
>>> print('Hello World!')
Hello World!
>>> 
```

```
>>> print('Hello World!')
>>> x = 5
>>> x
>>> y = 4*x+x**2
>>> y
```

Scripts

- Opeenvolgende commando's opgeslagen in een .py bestand
 - Aanroepen vanuit het command prompt
 - Bestanden in de map waarin je python start zijn beschikbaar
 - Met "-i" als argument start een interactieve sessie na het uitvoeren van het script
-
- "Run" knop in Spyder

```
> python -i hello_world.py
```

Variablen

- Int
- Float
- List
- String
- Dictionary

```
>>> A = 1
>>> B = 3.572
>>> C = 'string'
>>> D = [1,2,3]
>>> D[0]
>>> D[-1]

>>> E = [1,'test',4,D]

>>> F = {'value':1, 3:'test', 'spam':'eggs'}
>>> F['spam']
```


Controle structuren

- for in :
- if : else:
- ...
- ":" en inspringen verplicht!

```
for i in range(10):  
    if i%2 == 0:  
        print('{} is even'.format(i))  
    else:  
        print('{} is oneven'.format(i))
```

Functies

- Groeperen van vaak gebruikte commando's
- Definieren voor aanroepen
- Documentatie, op te roepen via "**help**(digits2number)"

```
def digits2number(A,B,C,D):  
    """  
    returns a a number as if the arguments were  
        different digits in the number  
  
    Parameters:  
    A: float, hundreds  
    B: float, decades  
    C: float, units  
    D: float, tenths  
    """  
  
    return 100*A+10*B+C+0.1*D
```

Functies

- Functies kunnen in verschillende files (modules) worden gedefinieerd
- Functie gebruiken in een ander script kan met een "import" statement
- geïmporteerde modules hebben een eigen "namespace"

```
from . import functies
```

```
val = functies.digits2number(2,4,1,9)  
print(val)
```

of

```
from .functies import *
```

```
val = digits2number(2,4,1,9)  
print(val)
```

Packages

- Folders kunnen beschouwd worden als packages door een bestand "`__init__.py`" toe te voegen
- Code in "`__init__.py`" wordt eerst uitgevoerd (vb submodules importeren)

```
.  
|-- lib  
    |-- __init__.py  
    |-- functies.py
```

```
# __init__.py  
from . import functies
```

```
>>> import lib  
>>> val = lib.functies.digits2number(2,4,1,9)
```

Externe packages

- Numpy - Algebra
- Matplotlib - plots
- Scipy - Algemene wetenschappelijke functies

Installatie:

- Meegeleverd met WinPython

Of

- Voeg het "pip" pad ("`pythonfolder/Scripts`") toe aan het windows pad en installeer python packages via
`"pip install numpy"`

Of

- Distributie downloaden via de project website en installeren

```
>>> import numpy as np
>>> a = np.zeros(10)
```

Werken met numpy

- prealloceren met "np.zeros", "np.linspace",...
- "len()", "Array.shape"
- indexeren met "[]", ":", vanaf 0 tot -1

```
x = np.zeros(10)
for i in range(len(x)):
    x[i] = 5*i-2

z = np.linspace(4,8,20)

y = np.zeros( (10,4) )
for i in range(y.shape[0]):
    for j in range(y.shape[1]):
        y[i,j] = 4*i+3*j-2

y[4,1:]
```

Plotten

- "figure", "plot", "xlabel", "legend"

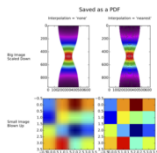
```
x = np.linspace(0, 2 * np.pi, 50)
s = np.sin(x)
c = scipy.integrate.cumtrapz(s, x)

plt.rc('text', usetex=True)
plt.rc('font', family='serif', size=8)
plt.rc('figure', autolayout=True)

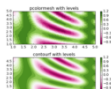
plt.figure(figsize=(10/2.54, 7/2.54))
plt.plot(x, s, '-s', label=r'sinus')
plt.plot(x[1:], c, label=r'$\int$ sinus$(x)$ d$x$')
plt.xlabel(r'$x$ (rad)')
plt.ylabel(r'$y$')
plt.legend(numpoints=1)

plt.savefig('sinus_cosinus.pdf')
plt.savefig('sinus_cosinus.png')
plt.show()
```

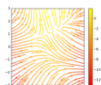
Matplotlib gallery (<http://matplotlib.org/gallery.html>)



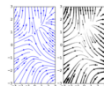
interpolation_none_vs_nearest



pcolormesh_levels



streamplot_demo_features



streamplot_demo_features



streamplot_demo_masking



streamplot_demo_start_points

Pie and polar charts



pie_demo_features



pie_demo_features



polar_bar_demo



polar_scatter_demo

Bestanden inlezen

- ascii: via file handler "with **open**('file.txt','r') as f:"
- Volledige file in één keer als string "data = f.read()"
- Lijn per lijn "for line in f:"
- "\t", "\n"
- Speciale packages voor speciale formaten (csv, openpyxl,...)

```
import csv

coord = []
with open('data.csv') as f:
    reader = csv.DictReader(f, delimiter='\t')
    for row in reader:
        coord.append( (row['x'],row['y'],row['y'],) )

for c in coord:
    print( c )
```

Een project structureren

- Gebruik functies
- Maak binnen functies gebruik van sub-functies indien nuttig
- Geef functies een betekenisvolle naam
- Groepeer functies die bij elkaar horen in een map en importeer deze
- Don't Repeat Yourself (DRY)
- Gebruik betekenisvolle namen voor variabelen
- Documenteer alles

Een project structureren

Voorbeeld folderstructuur:

```
mijnProject
|-- data
|   |-- mijndata.csv
|
|-- lib
|   |-- __init__.py
|   |
|   |-- data
|       |-- __init__.py
|       |-- data_inlezen.py
|       |-- data_naar_coordinaten.py
|       |
|   |-- plot
|       |-- __init__.py
|       |-- plot_coordinaten.m
|
|-- main.py
|-- readme
```

Een project structureren

Voorbeeld main.py:

```
# main.py
# dit script leest data in, vertaalt deze in coördinaten
# en maakt een plot

# functies importeren
import lib

# data inlezen en bewerken
data = lib.data.data_inlezen('data/mijndata.csv')
[x,y,z] = lib.data.data_naar_coördinaten(data)

# plotten
lib.plot.plot_coördinaten(x,y,z)
```

Oefening

Bepaal en teken het temperatuursverloop van een massa met warmtecapaciteit $C = 100 \times 10^3 \text{ J/K}$ en oppervlakte $A = 6 \text{ m}^2$ die van een initiële temperatuur van 80°C afkoelt in een omgeving van 20°C over een periode van 24h voor verschillende isolatie diktes. De massa is geïsoleerd met een materiaal met geleidingscoëfficiënt $k = 0.04 \text{ W/m}^2\text{K}$ en dikte d tussen 0 en 20 cm. De convectiecoëfficiënt h is $8 \text{ W/m}^2 \text{ K}$

Het temperatuursverloop kan gevonden worden door de geldende differentiaal vergelijking te discretiseren en numeriek te integreren:

$$T_{i+1} = T_i + \frac{UA}{C}(T_a - T_i)(t_{i+1} - t_i) \quad (1)$$

met:

$$U = \left(\frac{1}{h} + \frac{d}{k} \right)^{-1} \quad (2)$$

 2016 Brecht Baeten

Dit werk is gelicenseerd onder de licentie Creative Commons Naamsvermelding-GelijkDelen 4.0 Internationaal. Ga naar <http://creativecommons.org/licenses/by-sa/4.0/> om een kopie van de licentie te kunnen lezen.

De bron van dit document en alle voorbeelden zijn beschikbaar op <https://github.com/BrechtBa/inleiding-tot-python>