An introduction to Python

Thibaut Lunet, Aitor Pérez

March 19, 2018

How to get Python (+ useful packages ...)

We are going to use the **miniconda** installer, which is cross-platform and provides package management, together with the **spyder** IDE.

1. Go to https://conda.io/miniconda.html
   (or Google search : "miniconda download")
2. Depending on the operating system, download installer (Python 2.7)
3. Install Python and required packages
   - Mac OS X or Unix:
     3.1 Open a terminal
     3.2 Run "bash Miniconda[...].sh", and yes for all ...
     3.3 Open a new terminal, or run "source ~/.bashrc"
     3.4 Run "**conda install spyder numpy scipy matplotlib sympy**"
   - Windows:
     3.1 Double-click on the .exe file, and yes for all ...
     3.2 Open "conda prompt" terminal (installed with miniconda)
     3.3 Run "**conda install spyder numpy scipy matplotlib sympy**"

History

XXX slkdjlksj

Python vs. Others (Matlab, Fortran, C/C++, ...)

- License-free and open-source ($\neq$ Matlab)
- Huge users community, many (free) packages for many applications
- Extremely easy of use for non-I-love-programming people ($\neq$ Fortran, C/C++)
- Easy interface with other (more-efficient) programming languages $\Rightarrow$ computation can be accelerated using Fortran or C/C++ library ...
- Can scale to very large problems (parallel computing, ...)
- Structured and friendly ways for developing library ($\neq$ Matlab)

Python = many advantages, with very few drawbacks !

Functioning principles

XXX dsdskldj

Practical tools

Using python console

```
lunet@matlnx13: ~/Recherche/Enseignement/python-math/examples
lunet@matlnx13:examples$ python
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 18:10:19)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Running a script

```
lunet@matlnx13: ~/Recherche/Enseignement/python-math/examples
lunet@matlnx13:examples$ python helloWorld.py
1 + 1 = 2
   YAILLLLLE !

lunet@matlnx13:examples$
```

All-in-One solution $\Rightarrow$ **Spyder** !

Hello World!

All the examples and python files are available at:
https://gitlab.unige.ch/Thibaut.Lunet/python-math

A first easy step ...

1. Launch Spyder
   - Windows : double click on an icon somewhere ...
   - Mac OS X or Unix : run "spyder" in terminal
2. Discover a wonderful environment #woaaah
3. Go to lower right corner $\rightarrow$ IPython console
   - write "1+1"
   - press enter ...
4. Go to text editor (middle)
   - write "print('hello world')
   - save and run the file ...

Basic variables types and operations

Slide codes at: python-math/examples/codeexamples.py

```
# Integer
n = 1
m = 7 % 3  # m = 1

# Float: By default, double precision!
x = 0.5
y = x/7  # y = 0.07142857142857142

# Complex
z = 1+1j
w = z + x + n  # Automatic conversion, w = 2.5 + 1j

# String
s = 'salut'
t = 'toi'
r = s + t  # r = 'saluttoi'

# Boolean
p = True
q = (n != 1)*p + (n == 1)*(x < 10)*(y >= 0)  # q = True = 1
```

Lists

```python
# Lists
l = [1, 2, 5, 6]
# Access elements : l[0] = 1, l[2] = 5, l[-1] = 6
# Slice : l[1:3] = [2, 5]

# Nested list
nl = [['vive', 'la'], ['saucisse', 2], 'Toulouse']
# Access sublist element : nl[0] = ['vive', 'la']
# Access final element : nl[0][1] = 'la', nl[1][0] = 'saucisse'

# List comprehension
l1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
l2 = [3 * n + 1 for n in l1 if n % 2 == 0]
# l2 = [7, 13, 19, 25, 31]
```

Dictionary

```
# Dictionaries
d = {'first': 1,
     'second': 'two',
     'third': {'A': [3, 4, 5], 'B': True}}
# Access elements : d['first'] = 1, d['second'] = 'two'
#                   d['third'] = {'A': [3, 4, 5], 'B': True}
#                   d['third']['A'] = [3, 4, 5]
#                   d['third']['A'][1] = 4
#                   d['third']['B'] = True
```

Conditional structures

Tabs matter !

```
# If clause
if 1 == 2:
    print('Tocard')
elif 1 == 0:  # Not mandatory
    print('Toujours pas')
else:  # Not mandatory
    print("OK d'accord")

# For loop
for i in range(5):
    print('i = {}'.format(i))

# While loop
i = 0
while i < 10:
    print('TAIHOOO-' + str(i))
    i += 1
    if i == 5:
        break  # Allows to escape from the while loop
```

Function definition

```python
def funcA(a, b=1):
    return a + b


print(funcA(1))  # Print 2, default value (1) used for b
print(funcA(0.5, 2))  # Print 2.5, 2 is used for b
print(funcA(0.5, b=2))  # Equivalent way to set b=2
# funcA() -> ERROR : at least a must be given


# Arguments can be passed as dictionnary for multi-argument functions like :
def funcB(x, y, p1=None, p2=1, p3='o', p4=False):
    return '{}, {} -- p1={}, p2={}, p3={}, p4={}'.format(x, y, p1, p2, p3, p4)


# -- Arguments are only written once
kwargs = {'p1': 12, 'p2': 2, 'p3': 'i', 'p4': True}
print(funcB(1, 2, **kwargs))  # Print 1, 2 -- p1=12, p2=2, p3=i, p4=True
print(funcB('a', 'b', **kwargs))  # Print a, b -- p1=12, p2=2, p3=i, p4=True
```

File I/O

XXX sdjhj

Using Numpy to manipulate arrays

XXX TODO

Overview of Scipy functionalities

XXX TODO

Data visualization with Matplotlib

XXX TODO

Symbolic computation with Sympy

XXX TODO