

I. Introduction à Python



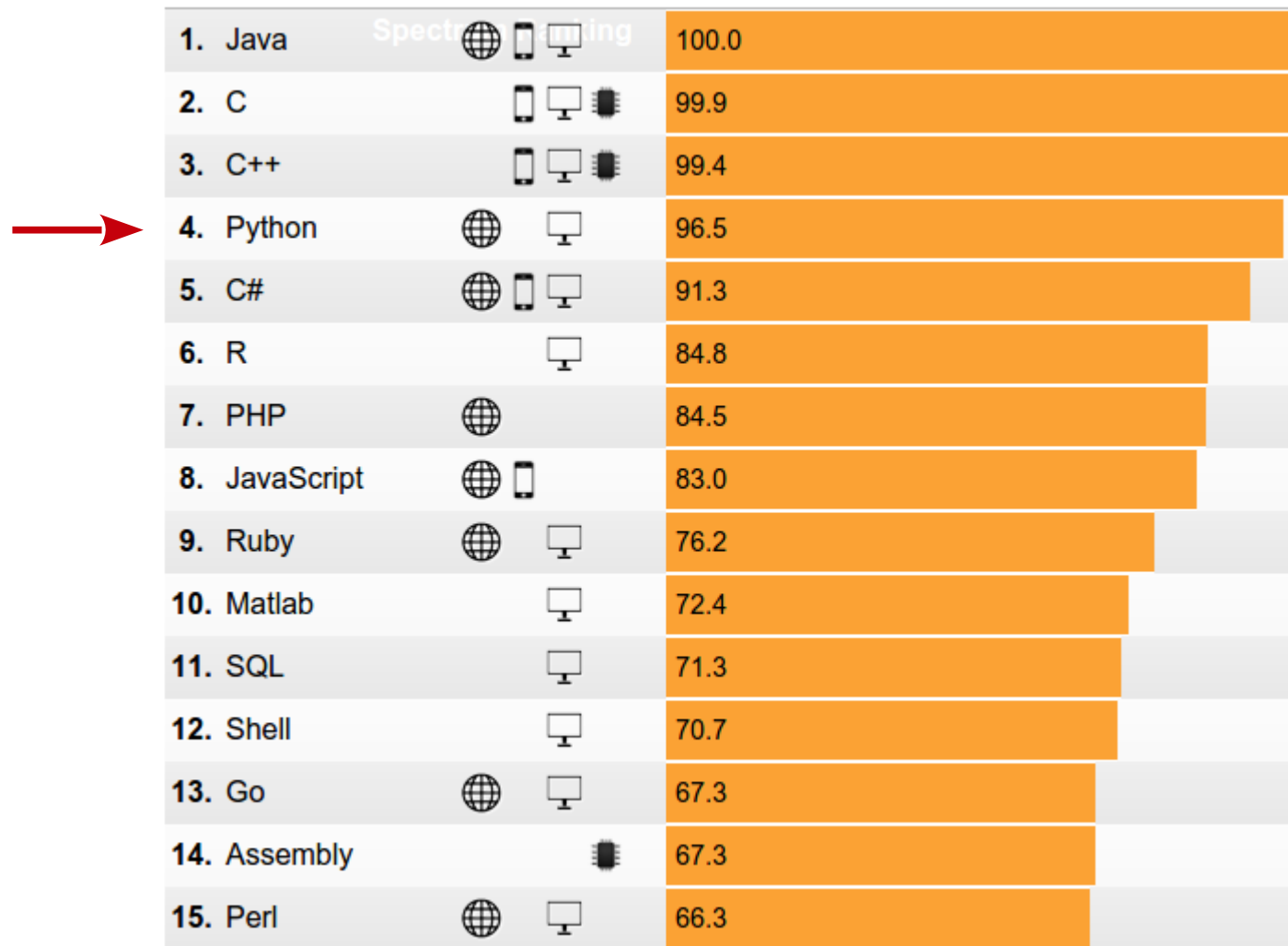
Michael Graupner
(michael.graupner@parisdescartes.fr)

Python

- langage moderne (depuis 1991) de programmation object
- langage interprété (pas de compilation nécessaire)
- l'accent est mis sur la lisibilité du code
- les concepts peuvent être exprimés en moins de lignes que c ou java
- vastes bibliothèques

Python : langage de programmation moderne orienté objet

Les meilleures langues de programmation 2015



[Source : IEEE Spectrum]

Python: syntaxe très claire, lisible → facile à apprendre

```
In [1]: # import modules
        from pylab import *

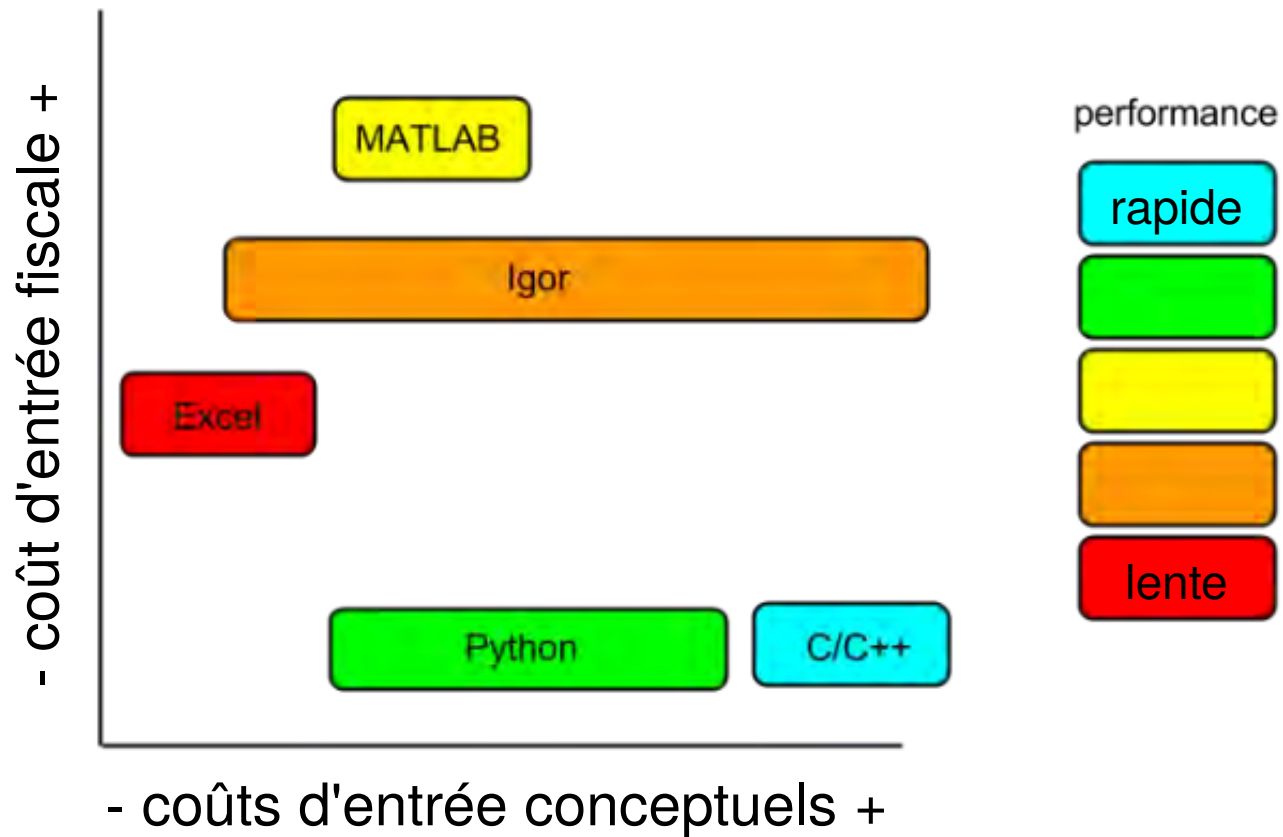
        # function declaration
        def update_values(x):
            return x+1

        x = 1
        if x>0:
            print 'Hello World!'
            x = update_values(x)

        print x
```

```
Hello World!
2
```

Python : gratuit et facile à apprendre



Python: bibliothèques standard et tierces étendues

- Extensions tierces pour pratiquement toutes les tâches

p.e. Reliures Python pour la boîte à outils GUI



Modules Python pour les neurosciences

- simulateurs et interfaces de simulateurs
- collecte et analyse des données
- partage, réutilisation, stockage de données et de modèles
- génération de stimulus
- recherche et optimisation des paramètres
- visualisation
- VLSI (very-large-scale integration - Intégration à très grande échelle)
interface matérielle

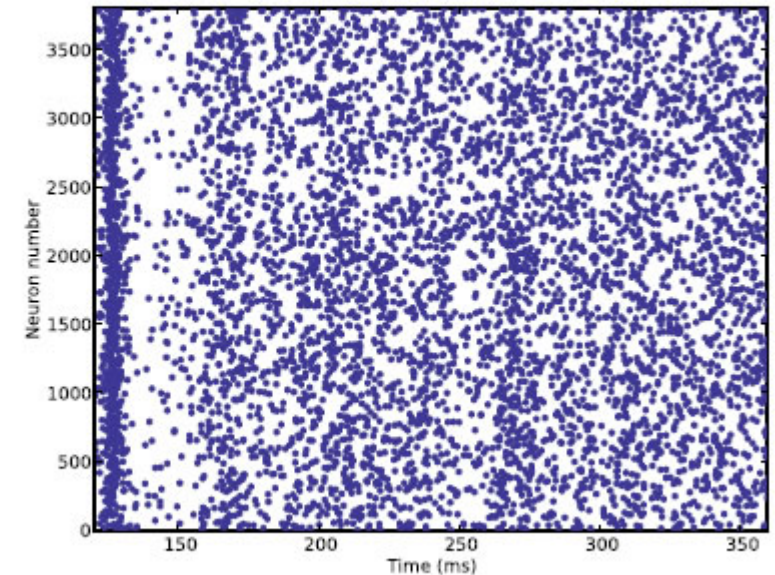
Python en Neurosciences : simulateurs et interfaces

p.e. Brian : le simulateur de réseau neuronal à spike

BRIAN

réseau récurrent, connecté de manière aléatoire

```
1 from brian import *
2 eqs = '''
3 dv/dt = (ge+gi-(v+49*mV))/(20*ms) : volt
4 dge/dt = -ge/(5*ms) : volt
5 dgi/dt = -gi/(10*ms) : volt
6 '''
7 P = NeuronGroup(4000, eqs, threshold=-50*mV, reset=-60*mV)
8 P.v = -60*mV+10*mV*rand(len(P))
9 Pe = P.subgroup(3200)
10 Pi = P.subgroup(800)
11 Ce = Connection(Pe, P, 'ge', weight=1.62*mV, sparseness=0.02)
12 Ci = Connection(Pi, P, 'gi', weight=-9*mV, sparseness=0.02)
13 M = SpikeMonitor(P)
14 run(1*second)
15 raster_plot(M)
16 show()
```

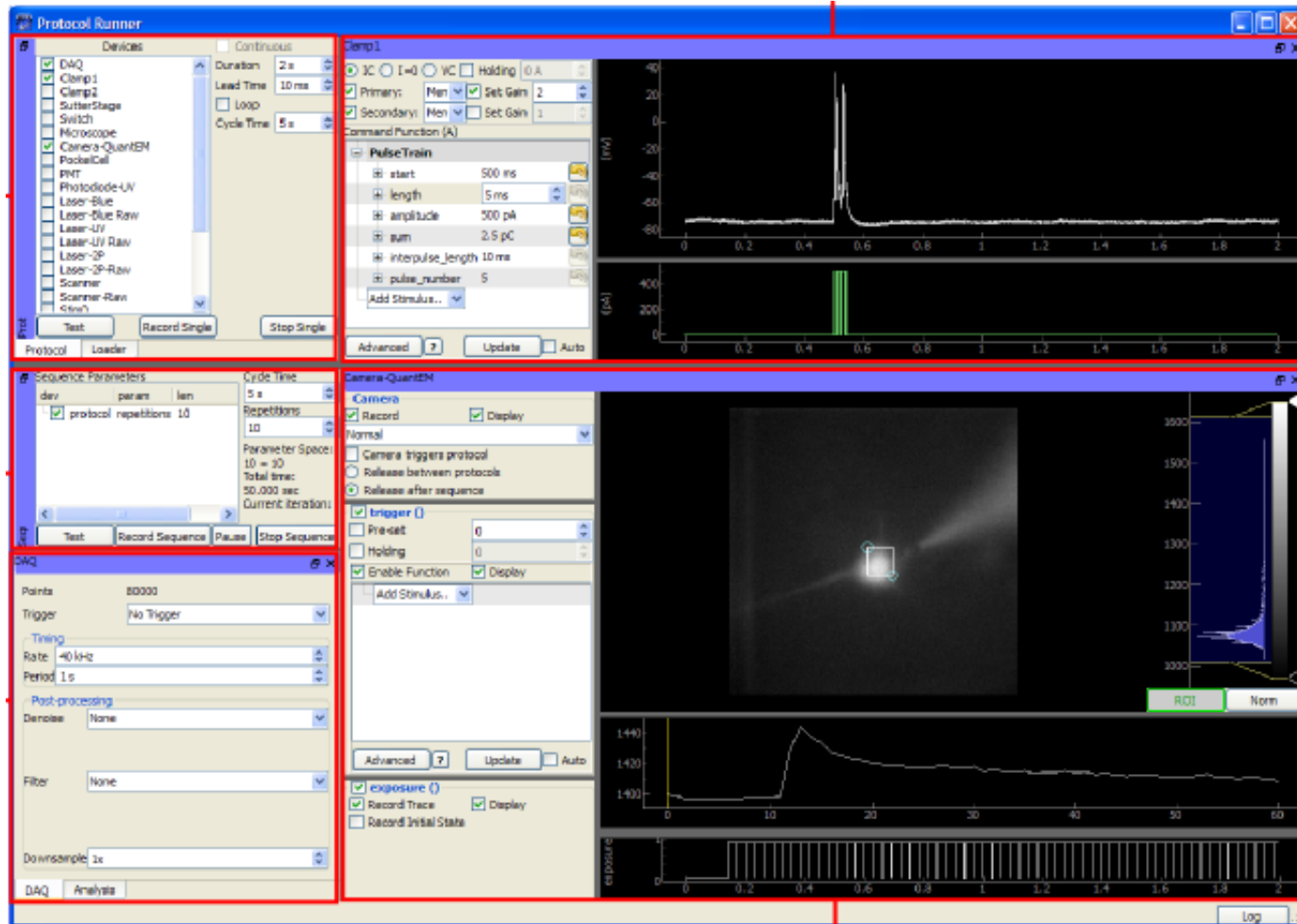


p.e. Python interface pour NEURON



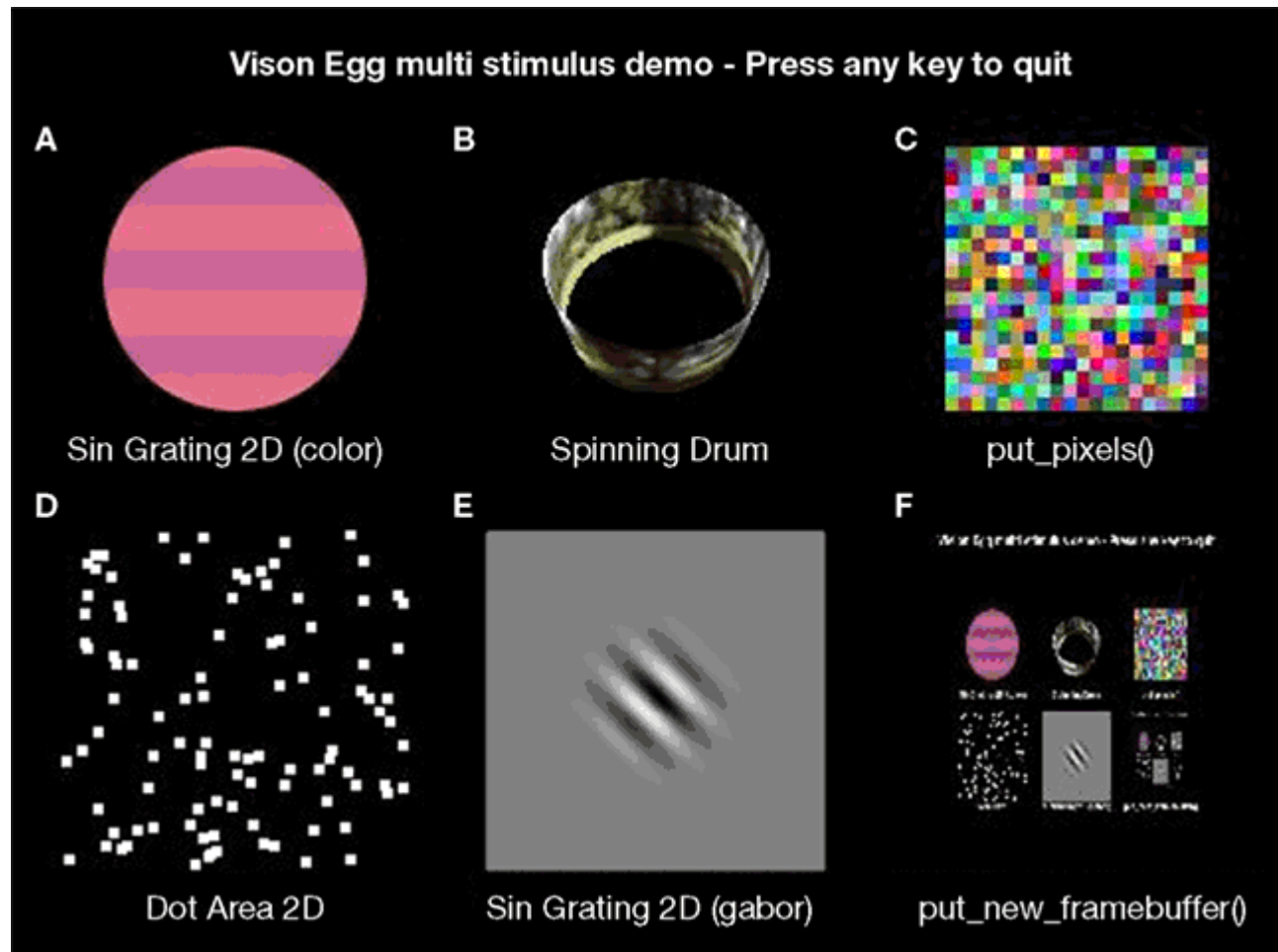
Py en Neuroscience : collecte et analyse des données

e.g. ACQ4



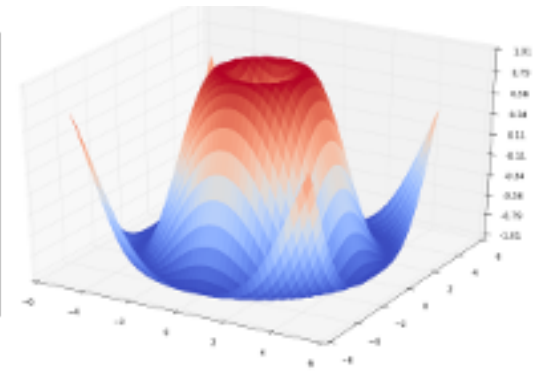
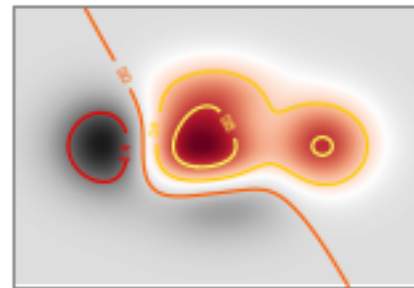
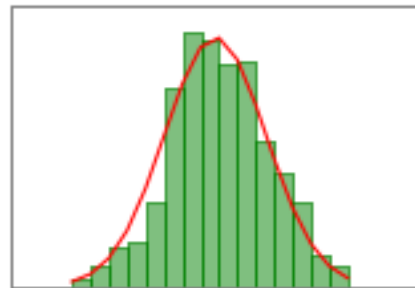
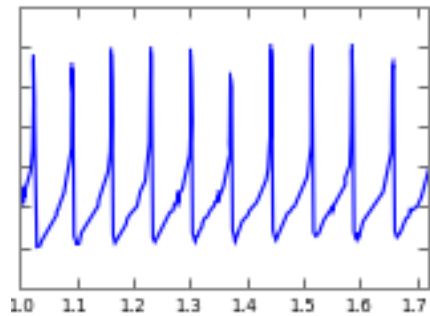
Python en Neurosciences : génération de stimulus

e.g. Vision EGG, ou PsychoPy



Python en Neuroscience : visualization

p.e. bibliothèque matplotlib



Commencer : installation de python

- Debian + Ubuntu Linux

```
apt-get install python-numpy python-scipy python-matplotlib \
ipython
```

- Windows, Mac OS X (distributions pour la gestion des paquets)

- Enthought Python : <https://www.enthought.com/>

- Anaconda de Continuum Analytics : <https://www.continuum.io/downloads>

- Python(x,y) <http://python-xy.github.io/>

- Mac OS X : Installer Fink, puis

```
fink install scipy-core-py25 scipy-py25 matplotlib-py25 ipython-py25
```

Commencer : interpréteur et IDEs

- **ipython**

- Interpréteur de ligne de commande (shell) interactive shell; introspection améliorée: surbrillance du code, auto-remplissage, etc.

- **Jython**

- Un autre interpréteur python écrit en java au lieu de c

- **IronPython**

- une implémentation python pour le framework .NET
- s'intègre bien avec d'autres langages .NET

- **Spyder** : Scientific PYthon Development EnviRonment

- **IPython Notebook**

- Interpréteur de ligne de commande dans le navigateur
- Combine l'exécution de code, le texte riche, les mathématiques, les tracés et le rich media

Spyder : capture d'écran



The screenshot shows the Spyder Python IDE interface. The main editor window displays a Python script named `Interpolation.py` with the following code:

```
1 """
2 Interpolation of an N-D curve
3 From the SciPy Cookbook
4 """
5
6 from numpy import arange, cos, linspace, pi, sin, random
7 from scipy.interpolate import splprep, splev
8
9 # make ascending spiral in 3-space
10 t=linspace(0,1.75*2*pi,100)
11
12 x = sin(t)
13 y = cos(t)
14 z = t
15
```

The Variable explorer panel on the right shows the following variables:

Name	Type	Size	Value
e	float	1	2.7182818284590451
pi	float	1	3.1415926535897931

The Object inspector panel shows the details for the selected object, which is an `array`:

array(...)
Function of `numpy.core.multiarray` module

`array(object, dtype=None, copy=True, order=None, subok=False, ndmin=0)`

Create an array.

Parameters

object: `array_like`
An array, any object exposing the array interface, an object whose `__array__` method returns an array, or any (nested) sequence.

dtype: `data-type, optional`
The desired data-type for the array. If not given, then the type will be determined as the minimum type required to hold

The Console panel shows the IPython 0.10.1 prompt and the following output:

```
IPython 0.10.1 -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object'. ?object also works, ?? prints more.

Welcome to pylab, a matplotlib-based Python environment.
For more information, type 'help(pylab)'.

In [1]:
```

The status bar at the bottom indicates: Permissions: RW | End-of-lines: LF | Encoding: UTF-8-GUESSED | Line: 7 | Column: 1

IPython [Jupyter] Notebook

IPython Dashboard x IPy spectrogram x

127.0.0.1:8888/a5222740-848b-4ac1-b212-d732c9f8f78b

IP[y]: Notebook spectrogram Last saved: Mar 07 11:14 PM

File Edit View Insert Cell Kernel Help

Markdown

Simple spectral analysis

An illustration of the [Discrete Fourier Transform](#)

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0, \dots, N-1$$

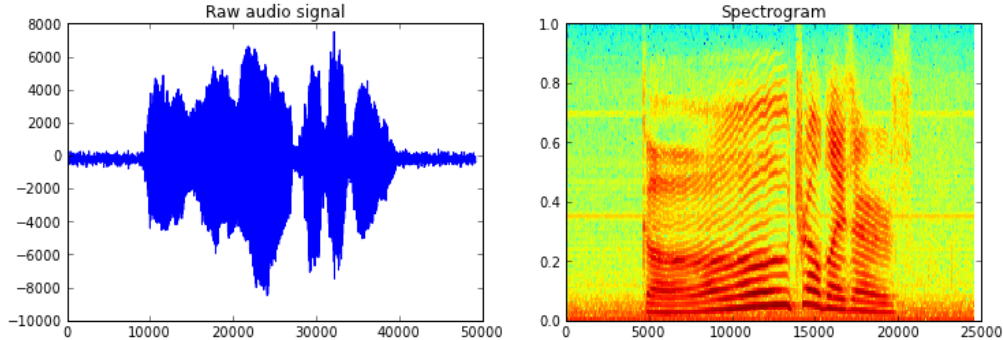
using windowing, to reveal the frequency content of a sound signal.

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin specgram routine:

```
In [2]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```



The figure displays two plots side-by-side. The left plot, titled "Raw audio signal", shows a blue waveform of an audio signal over time. The x-axis ranges from 0 to 50,000, and the y-axis ranges from -10,000 to 8,000. The right plot, titled "Spectrogram", shows the frequency content of the signal. The x-axis is time (0 to 25,000) and the y-axis is frequency (0.0 to 1.0). The spectrogram uses a color scale from blue (low intensity) to red (high intensity) to show the signal's frequency components over time.

Exécution de programmes python

- Les programmes Python peuvent être exécutés de manière interactive ou sous forme de scripts stockés dans un fichier
- L'interpréteur démarre en appelant **python** (ou **ipython**)

```
mgraupe@thinkpadx1:~> python
Python 2.7.10 (default, Oct 14 2015, 16:09:02)
Type "help", "copyright", "credits" or "license"
for more information.
>>> print 'Hello world!'
Hello world!
>>> x = 3
>>> print x+5
8
```

- Les scripts sont fournis comme arguments à l'interpréteur

```
mgraupe@thinkpadx1:~> python hello_world.py
Hello world!
```

- **python -i [script.py]** donne une invite interactive après l'exécution du script

Python scripts

- L'extension par défaut pour les fichiers python est **.py**
- Les scripts commencent avec l'interprète qu'ils doivent utiliser

```
#!/usr/bin/env python  
print 'Hello world!'
```

- En option, vous pouvez spécifier l'encodage du fichier dans la ligne 2

```
#!/usr/bin/env python  
# * coding: utf8 *  
print 'Total: 42 €'
```

- Les scripts doivent être exécutable : **chmod u+x <fichier>**
- Exécuter des scripts en tant que programmes autonomes

```
mgraupe@thinkpadx1:~> ./hello_world.py  
Hello world!
```

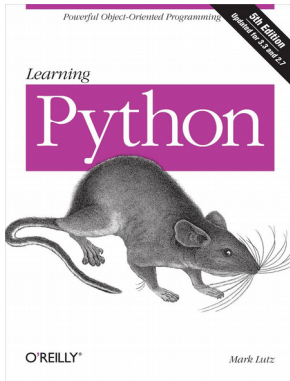
Ressources en ligne: généralités

- Index de documentation python :
<https://docs.python.org/2.7/>
- Référence bibliothèque Python :
<https://docs.python.org/2.7/library/>
- Plonger dans le python :
<http://www.diveintopython.net/>
- Activestate Python [livre de cuisine] :
<http://aspn.activestate.com/ASPN/Cookbook/Python>
- Le tutoriel python :
<https://docs.python.org/2/tutorial/index.html>
- Le tutoriel Numpy :
<http://www.time.mk/trajkovski/teaching/imi/2010-fall/NumPy/Tentative%20NumPy%20Tutorial%20-.html>
- Référence Scipy :
<http://docs.scipy.org/doc/scipy/reference/genindex.html>

Ressources en ligne: neurosciences

- Front Neuroinform 2015 – *Python in Neuroscience* :
<http://journal.frontiersin.org/article/10.3389/fninf.2015.00011/full>
- BCCN/FACETS Student Workshop - *Using Python for Computational*
<http://neuralensemble.org/cookbook/wiki/FacetsPythonCourse2008>
- BCCN cours - *Advanced Scientific Programming in Python* :
<https://python.g-node.org/wiki/schedule>
- Brian simulator :
<http://briansimulator.org/>

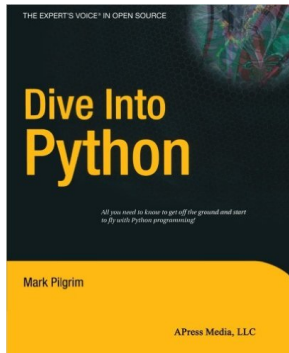
Livres



- Learning Python, 5th Edition

Mark Lutz

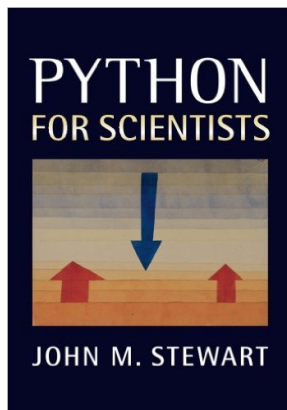
ISBN : 978-1-4493-5573-9



- Dive Into Python (3)

Mark Pilgrim

ISBN: 978-1590593561 (978-1430224150)



- Python for Scientists

John M. Stewart

ISBN: 978-1107686427