Neurosciences Computationnelles : TD1 Introduction à Python

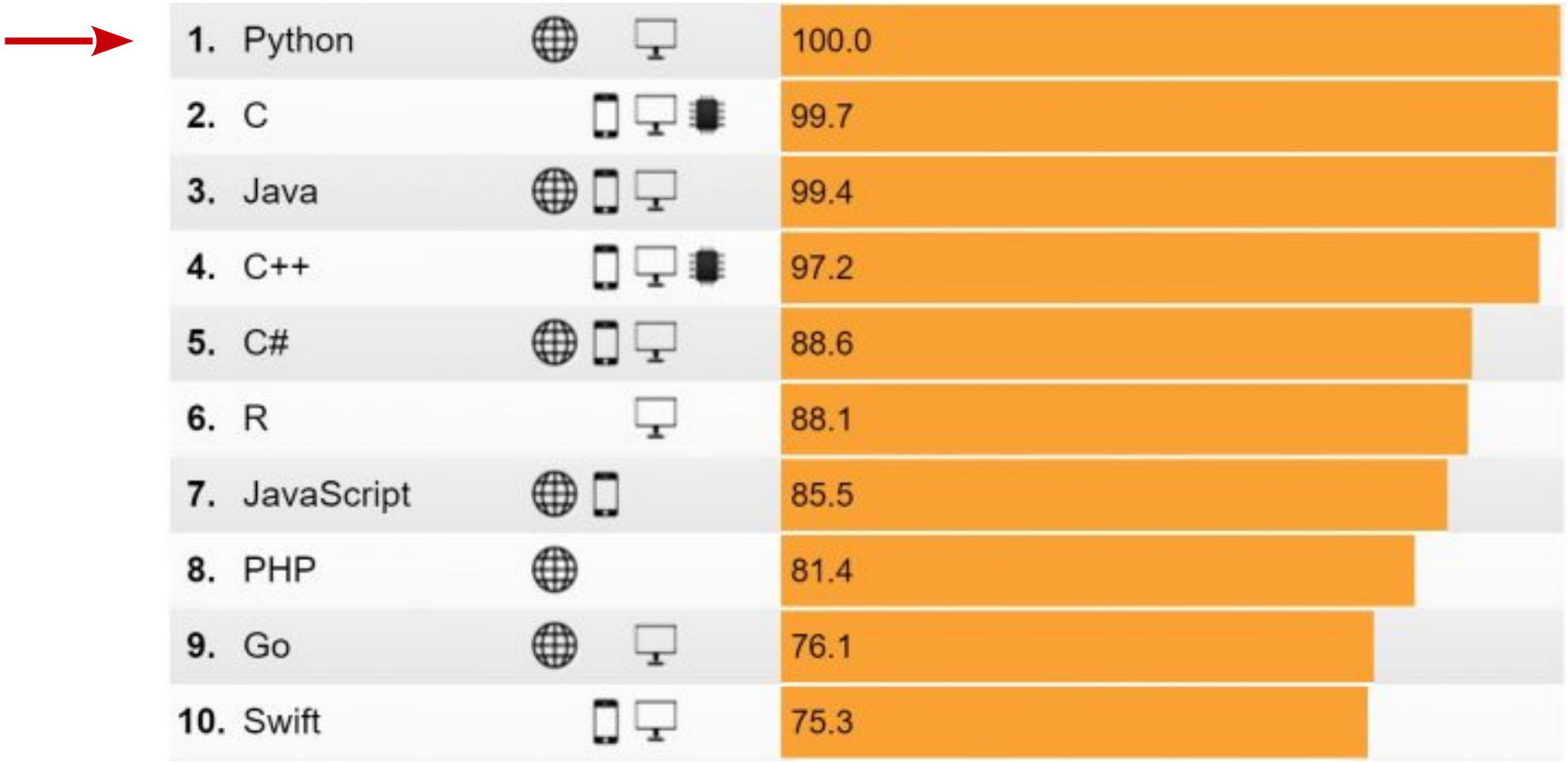
Michael Graupner
(michael.graupner@parisdescartes.fr)

Qu'est-ce que python ?

- language moderne (depuis 1991) de programmation
- 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/C++ ou Java
- vastes bibliothèques, fonctions disponibles
- visualisation facile

Python : langage de programmation moderne

Les langages de programmation les plus populaires en 2017



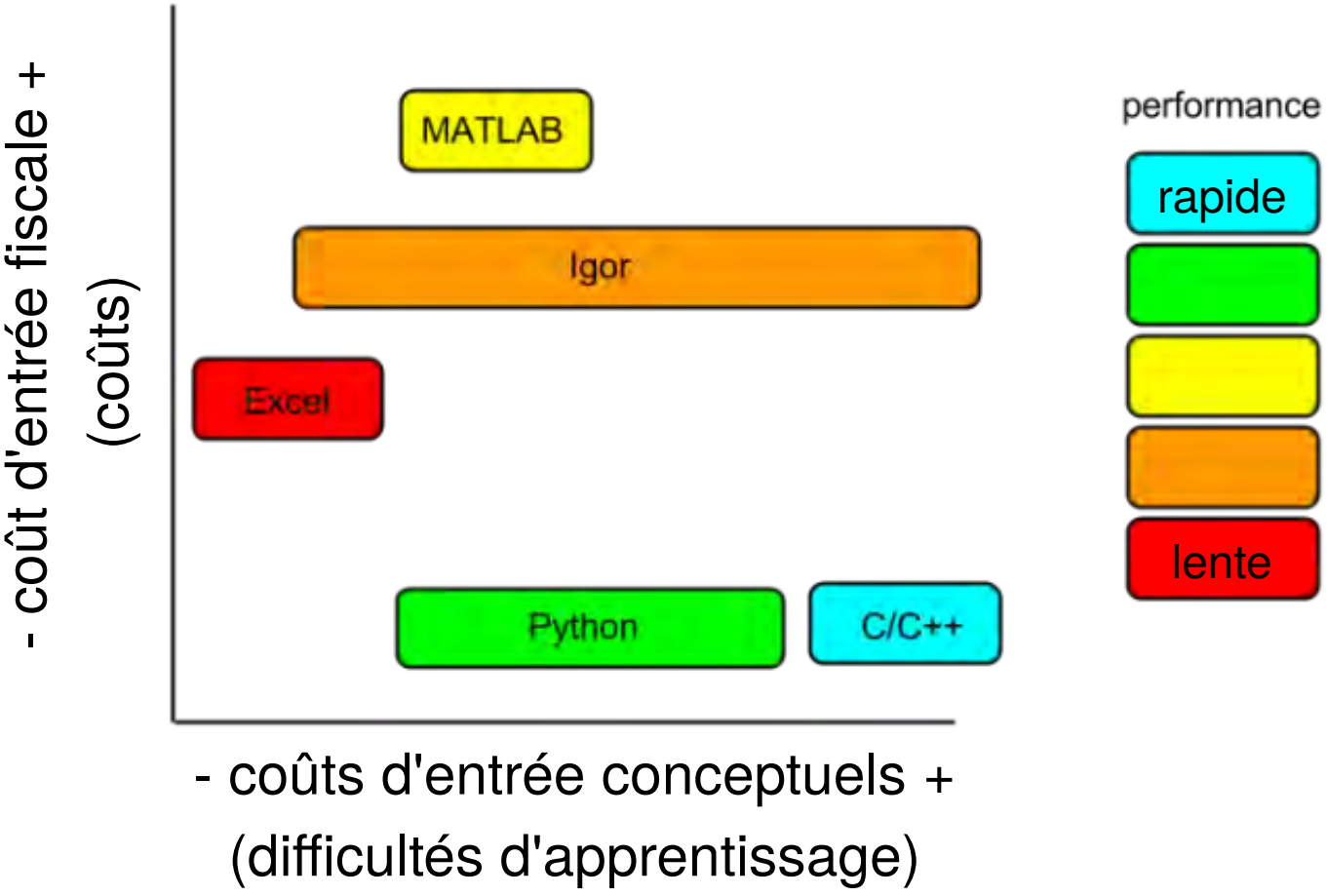
[Source : IEEE Spectrum]

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

- **wxPython :** Une boîte à outils GUI pour python.
- **SymPy :** peut faire l'évaluation algébrique, la différenciation, l'expansion, les nombres complexes, etc.
- **Pygame :** Une bibliothèque pour le développement de jeux 2D.
- **Twisted :** L'outil le plus important pour tout développement d'application réseau.
- **OpenCV :** Une bibliothèque conçue pour résoudre les problèmes de vision par ordinateur.

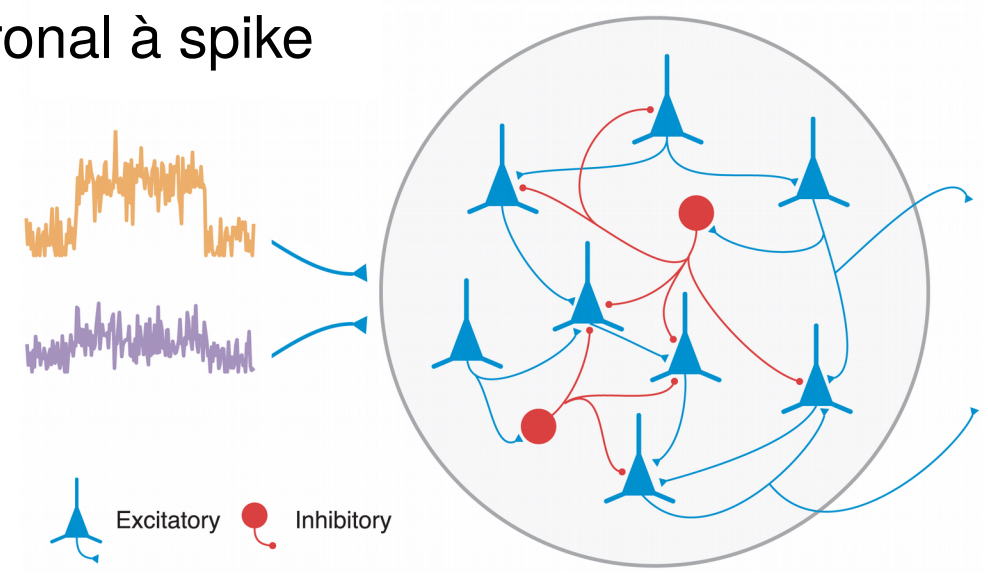
Python: bibliothèques standard et tierces étendues

Bibliothèques pour les neurosciences

- simulateurs des réseau neuronal 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

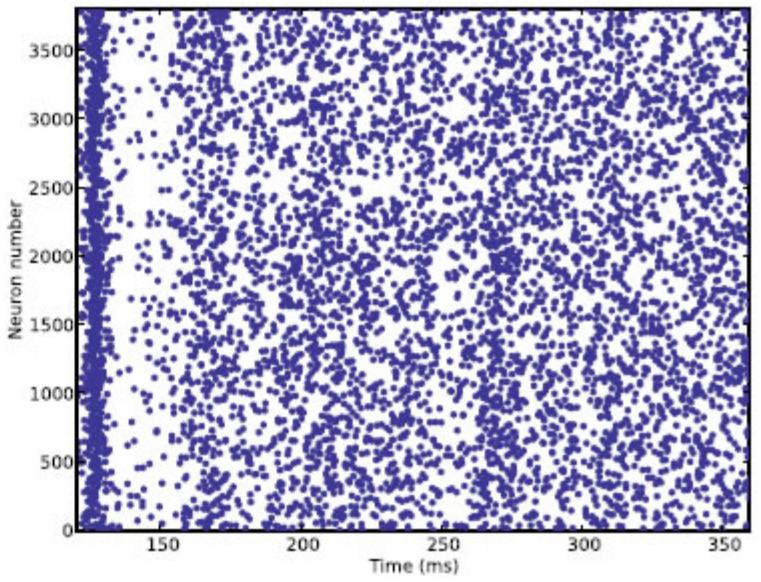
Python en Comp. Neurosci.: simlateur de réseau

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



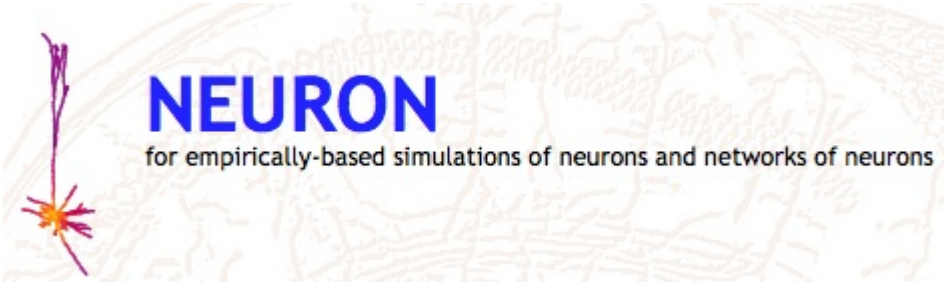
réseau récurrent, connecté de manière aléatoire

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

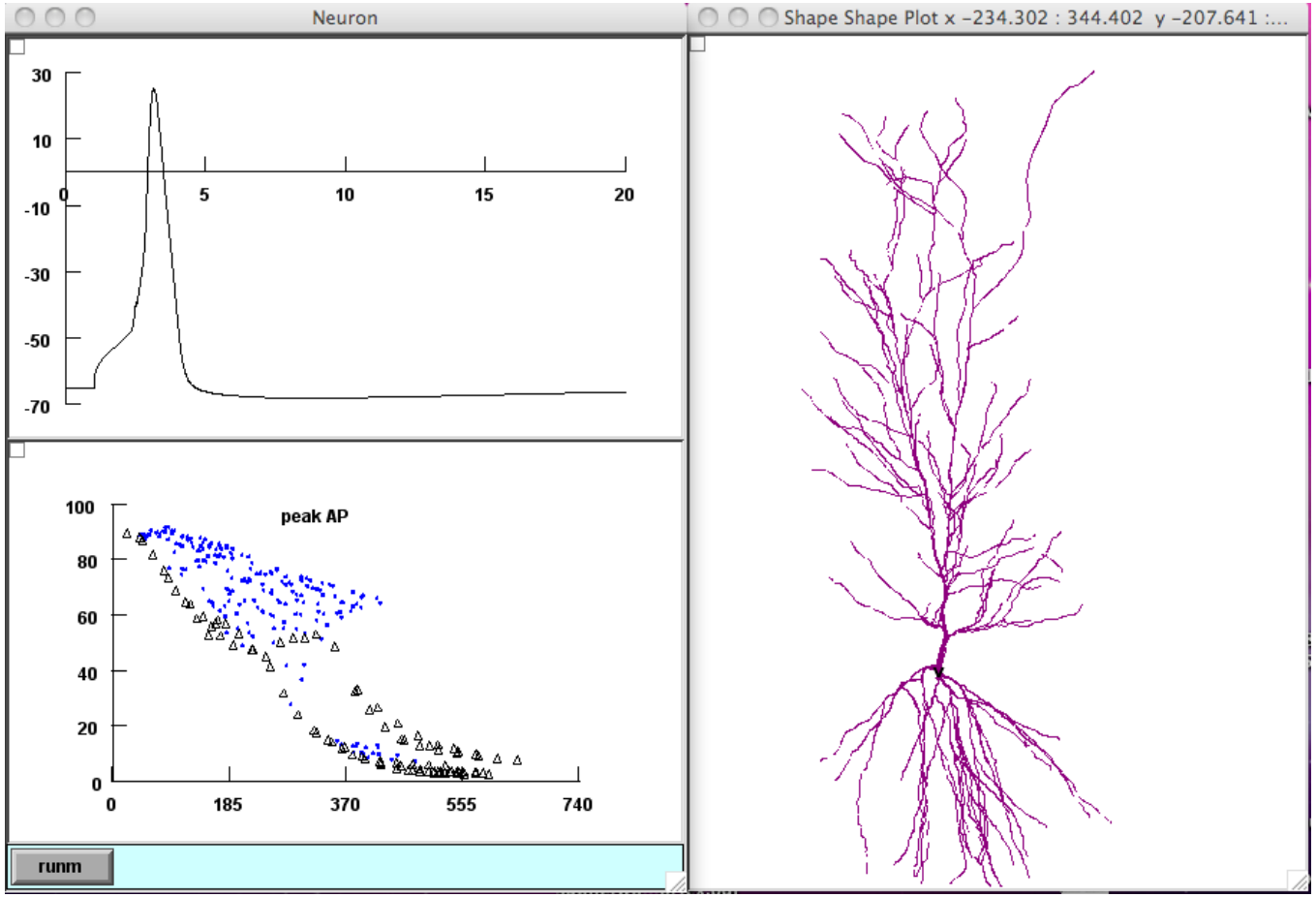


Python en Comp. Neuro.: simlateur des neurones

p.e. Python interface pour NEURON

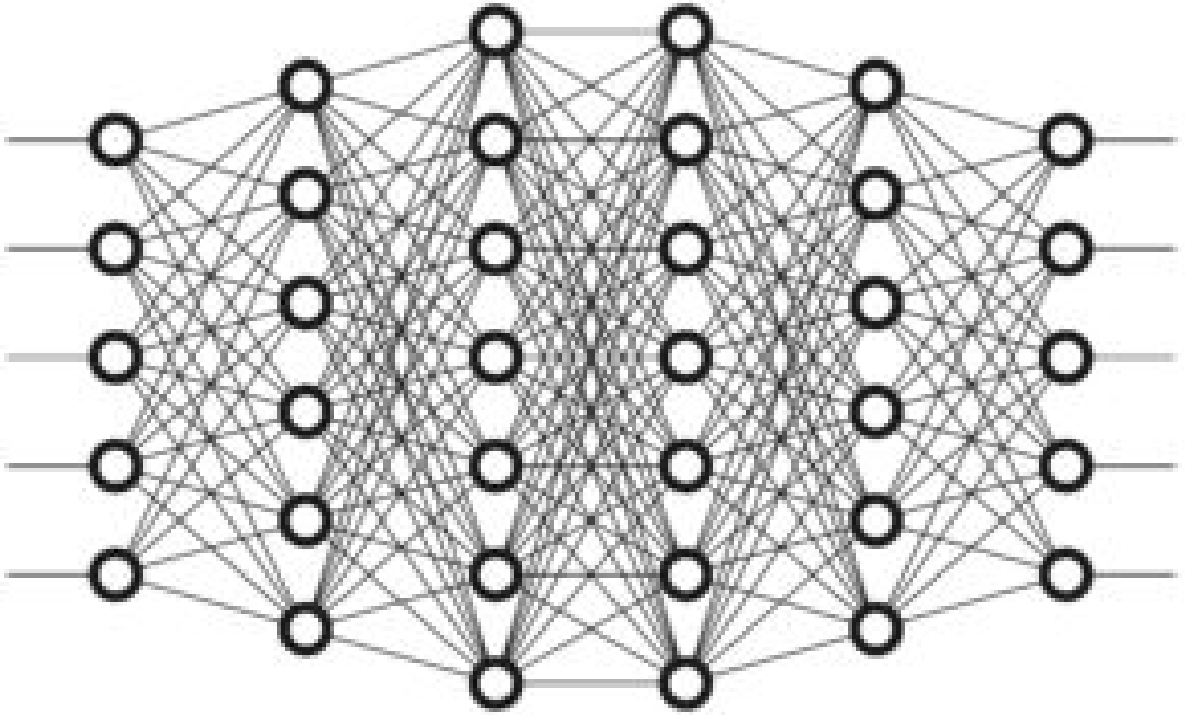


modèle compartimentaire simulant la propagation de la tension



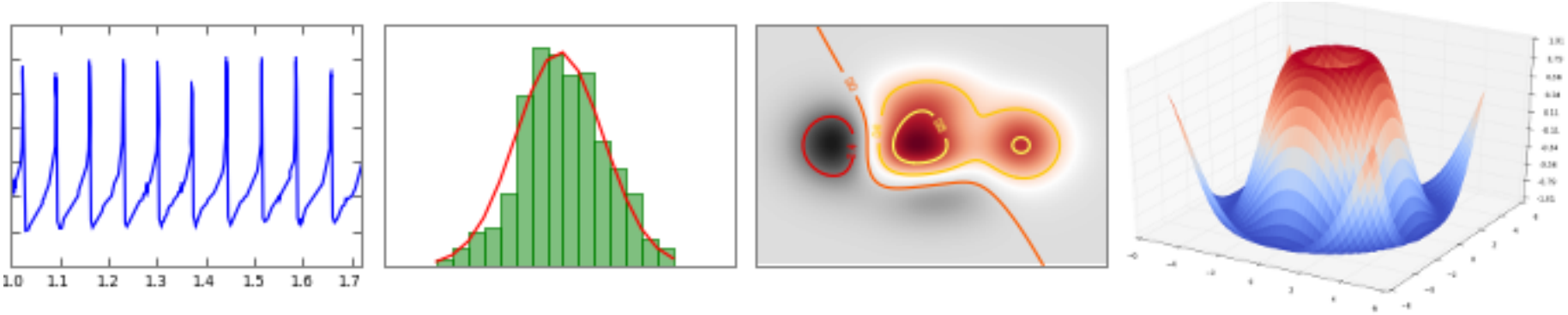
Python en Comp. Neuroscience : deep networks

p. e. TensorFlow - simuler des réseaux pour “deep learning”



Python en Comp. 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/bibliothèques)

  - Anaconda de Continuum Analaytics : https://www.continuum.io/downloads

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

  - 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**

  - Interprèteur de ligne de commande (shell) interactive shell; introspection améliorée: mise en évidence du code, auto-remplissage, etc.

- **Spyder** : Scientific PYthon Development EnviRonment

- **IPython/Jupyter Notebook**

  - Interprèteur de ligne de commande dans le navigateur

  - Combine l'exécution de code, le texte riche, les mathématiques, les tracés et le rich media

- **PyCharm :** environnement de développement professionnel

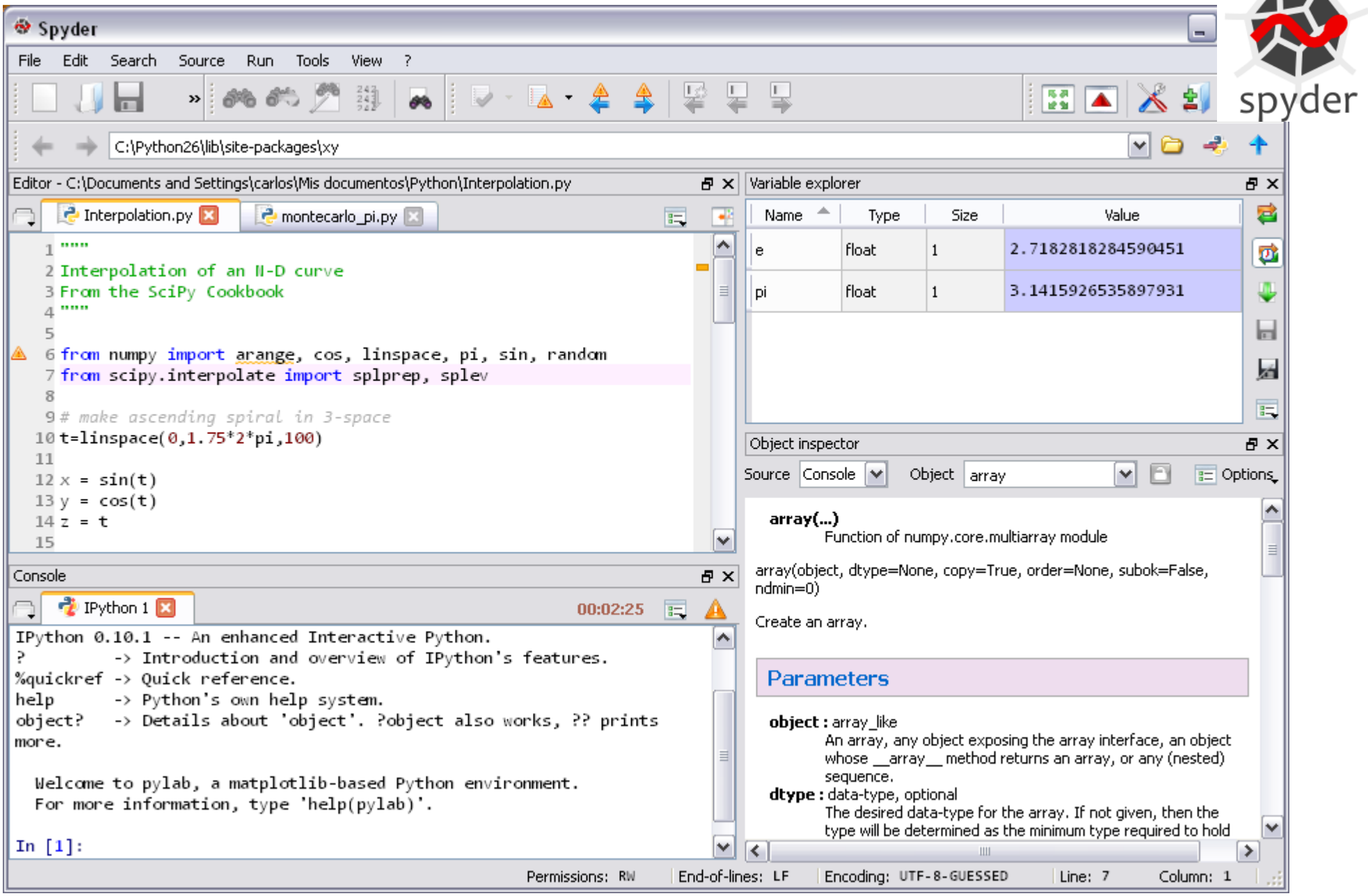
- **Jython**

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

- **IronPython**

  - une implémentation python pour le framework .NET

Spyder


Spyder
File Edit Search Source Run Tools View ?
C:\Python26\lib\site-packages\xy
Editor - C:\Documents and Settings\carlos\Mis documentos\Python\Interpolation.py
Interpolation.py
montecarlo_pi.py
"""
Interpolation of an N-D curve
From the SciPy Cookbook
"""
from numpy import arange, cos, linspace, pi, sin, random
from scipy.interpolate import splprep, splev
# make ascending spiral in 3-space
t=linspace(0,1.75*2*pi,100)
x = sin(t)
y = cos(t)
z = t
Variable explorer
Name Type Size Value
e float 1 2.7182818284590451
pi float 1 3.1415926535897931
Object inspector
Source Console Object array Options
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
Console
IPython 1
00:02:25
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]:
Permissions: RW
End-of-lines: LF
Encoding: UTF-8-GUESSED
Line: 7
Column: 1

IPython/Jupyter Notebook

IPy IPython Dashboard | IPy spectrogram

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

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

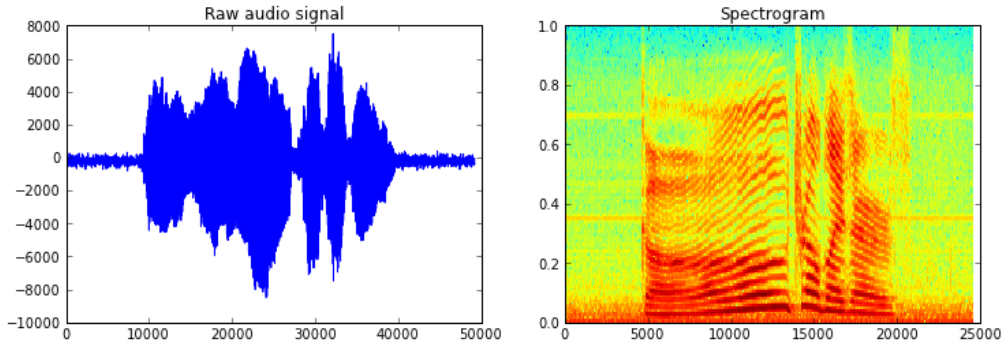
using windowing, to reveal the frequency content of a sound signal.

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

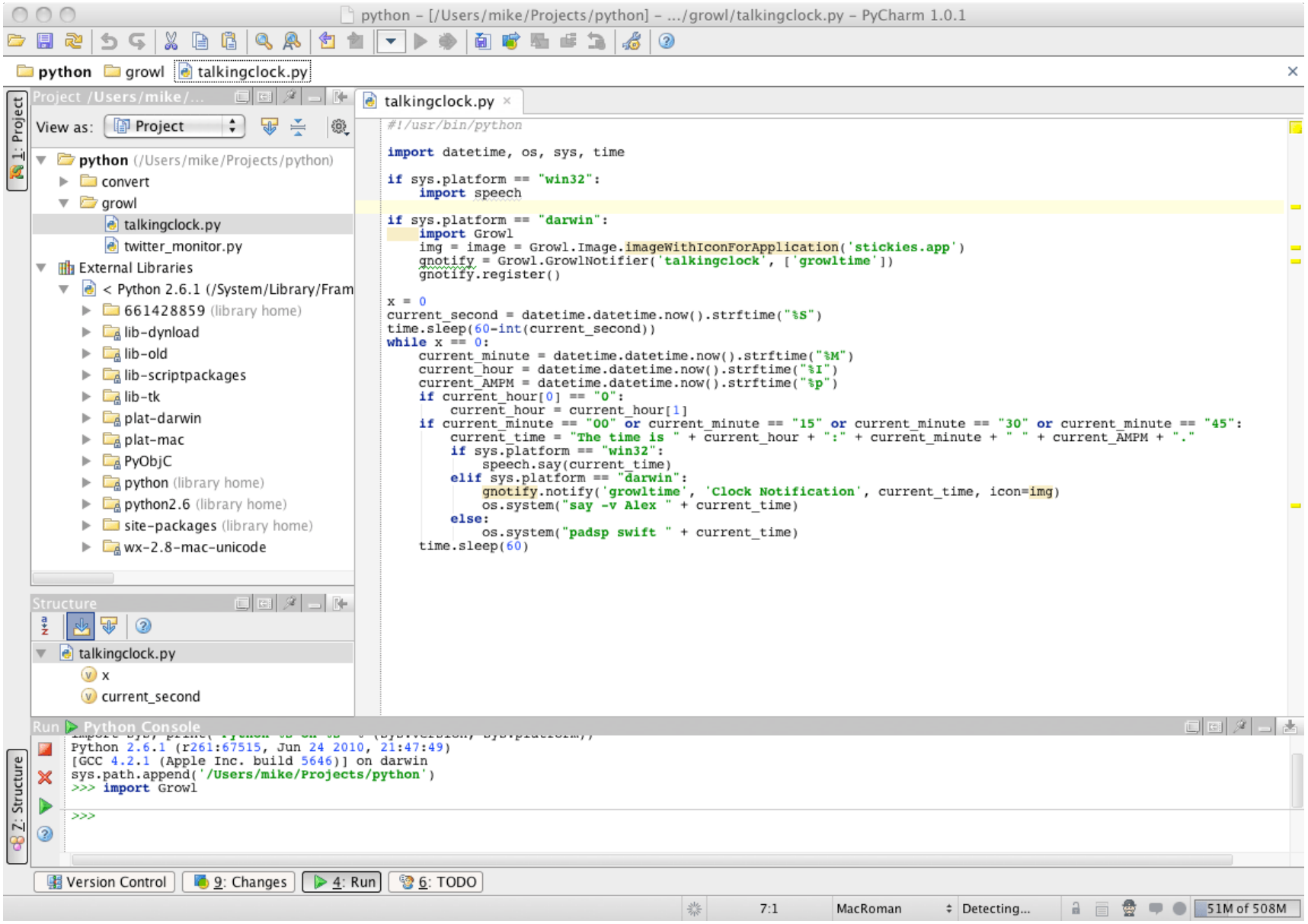
In [1]:
```
from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

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');
```



PyCharm


python – [/Users/mike/Projects/python] – .../growl/talkingclock.py – PyCharm 1.0.1
python
growl
talkingclock.py
Project /Users/mike/...
View as:
Project
python (/Users/mike/Projects/python)
convert
growl
talkingclock.py
twitter_monitor.py
External Libraries
< Python 2.6.1 (/System/Library/Fram
661428859 (library home)
lib-dynload
lib-old
lib-scriptpackages
lib-tk
plat-darwin
plat-mac
PyObjC
python (library home)
python2.6 (library home)
site-packages (library home)
wx-2.8-mac-unicode
Structure
talkingclock.py
x
current_second
talkingclock.py
#!/usr/bin/python
import datetime, os, sys, time
if sys.platform == "win32":
import speech
if sys.platform == "darwin":
import Growl
img = image = Growl.Image.imageWithIconForApplication('stickies.app')
gnotify = Growl.GrowlNotifier('talkingclock', ['growltime'])
gnotify.register()
x = 0
current_second = datetime.datetime.now().strftime("%S")
time.sleep(60-int(current_second))
while x == 0:
current_minute = datetime.datetime.now().strftime("%M")
current_hour = datetime.datetime.now().strftime("%I")
current_AMPM = datetime.datetime.now().strftime("%p")
if current_hour[0] == "0":
current_hour = current_hour[1]
if current_minute == "00" or current_minute == "15" or current_minute == "30" or current_minute == "45":
current_time = "The time is " + current_hour + ":" + current_minute + " " + current_AMPM + "."
if sys.platform == "win32":
speech.say(current_time)
elif sys.platform == "darwin":
gnotify.notify('growltime', 'Clock Notification', current_time, icon=img)
os.system("say -v Alex " + current_time)
else:
os.system("padsp swift " + current_time)
time.sleep(60)
Run
Python Console
Python 2.6.1 (r261:67515, Jun 24 2010, 21:47:49)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
sys.path.append('/Users/mike/Projects/python')
>>> import Growl
>>>
1: Project
7: Structure
Version Control
9: Changes
4: Run
6: TODO
7:1
MacRoman
Detecting...
51M of 508M

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

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`

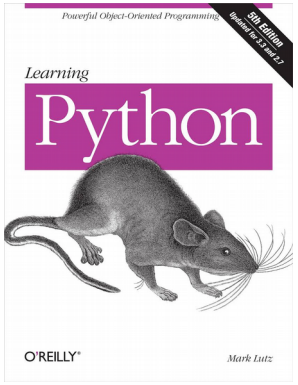
- Référence Scipy :

  `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/`

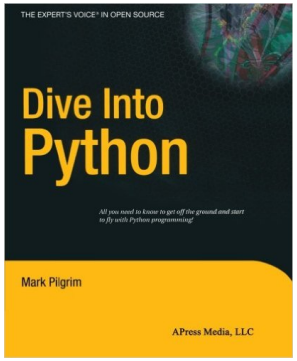
Livres



- Learning Python, 5th Edition

  Mark Lutz

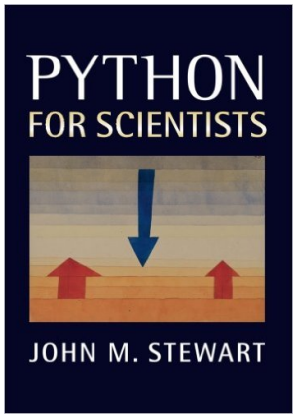
  ISBN : 978-1-4493-5573-9



- Dive Into Python (3)

  Mark Pilgrim

  ISBN: 978-1590593561 (978-1430224150)

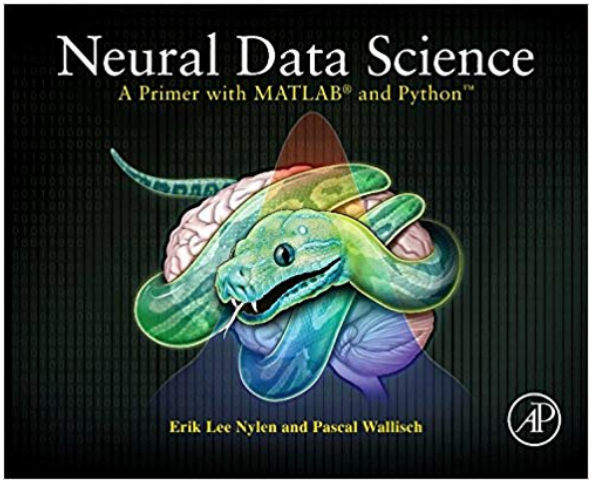


- Python for Scientists

  John M. Stewart

  ISBN: 978-1107686427

Livres



- Neural Data Science

  A primer with Matlab and Python

  Erik Lee Nylen (Author), Pascal Wallisch (Author)

  ISBN-10: 9780128040430