

Python, une introduction

O. Wilk

Calcul scientifique/Math/Cnam

Février 2011

le cnam

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Plusieurs parties :

- **Python, une introduction : des bases, du calcul scientifique (numpy, scipy, ...), de la visualisation.**
- Le module scipy plus en détails, couplage fortran-python (f2py) et le C ...
- Des applications : avec GetFem, ...

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Introduction

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Introduction

Un langage **libre** et **portable** sur n'importe quel système (Windows, Mac, Linux)

permettant de **relier des mondes et des langages différents** :

	<i>réseau</i>	
<i>calcul</i>	<i>Python</i>	<i>visualisation</i>
	<i>Java</i>	<i>VTK</i>
<i>C</i>		
	<i>Fortran</i>	<i>Cuda, OpenCL</i>
<i>entrée/sortie</i>	<i>IHM</i>	<i>GTK web</i>

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Introduction

- 1989 : Guido van Rossum développe Python,
- 1991 : Python est publié,
- 1994 : premières applications scientifiques.

Aujourd'hui :

Programming Language	Web Rate	Position Jan 2011	Position Jan 2006	Position Jan 1996
Java	18 %	1	1	5
C	16 %	2	2	1
C++	9 %	3	3	2
PHP	8 %	4	4	-
Python	6 %	5	8	22
C#	6 %	6	7	-
(Visual) Basic	6 %	7	5	3
Objective-C	3 %	8	44	-
Perl	3 %	9	6	6
Ruby	2 %	10	20	-
Lisp	1 %	13	14	13
Ada	1 %	20	17	12

Classement TIOBE / moteurs de recherche du web.

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

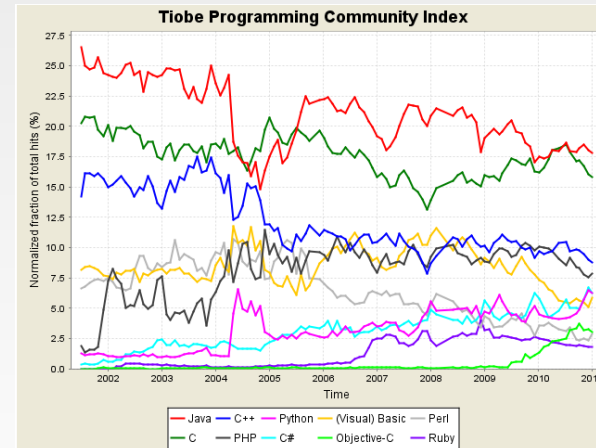
La visualisation

Matplotlib

Mayavi

Bibliographie

Introduction



(Seuls 3 langages progressent : Python, Objective-C : langage phare du développement pour Mac OS X, C# roi des langages .NET pour Microsoft)

Python élu langage de l'année 2010 par TIOBE.

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Introduction

Pourquoi Python s'impose-t-il ?

- Rien à déclarer, gestion automatique de la mémoire,
- orienté "objet", interprété et partiellement compilé (alternative à Matlab, Scilab, Octave, ... mais pas simplement),
- même si ce n'est pas le plus rapide, on peut l'interfacer avec d'autres langages plus performants : facile à interfacer avec le Fortran (f2py), le C (swig, boost), CUDA et OpenCL (pycuda, pyopencl), ...
- Des bibliothèques déjà très optimisées et un grand choix de bibliothèques,
- d'autres langages accessibles par python : Jython ⇒ Java,
- une plateforme très "communautaire".

Python permet d'intégrer facilement des codes de calcul existants (Fortran, C, ...), de les rendre accessibles (MacroCommandes, Gui), de traiter graphiquement leurs résultats et de poursuivre leurs développements ...

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Introduction

Un grand choix de bibliothèques et de logiciels ouverts, permettant de faire du :

web (Zope, Plone, Django, ...), bases de données (MySQL, Oracle, ...), réseaux (PyRO, ...), Gui (Gtk, Qt, WxWidgets, ...), graphique (gnuplot, matplotlib, VTK, MayaVi, ...), calcul scientifique (E.F. (FiPy, Getfem, ...), systèmes Dynamiques (SimPy), ...), mathématiques (Sage), statistiques (MDP), bioinformatique (BioPython, MMTK, ...), ...

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Des bases

Python,
une introduction
O. Wilk

Introduction

Des bases

Commentaires, chaînes
Types et opérations
Entrée - Sortie
Fonctions

Le calcul scientifique

Numpy
Scipy

La visualisation

Matplotlib
Mayavi

Bibliographie

Pour commencer (1/2)

Choisir son interface, son environnement de programmation :

- python,
- ipython (complétion automatique, ...), bpython, spyder,
- des environnement de dév. : idle, eclipse+PyDev.

Savoir chercher de l'information :

- sous le prompt Python : help(), help(nom d'une variable, d'une fonction ou d'un package),
- sur la toile : <http://docs.python.org>,
<http://www.python.org/doc/QuickRef.html>.

Python,
une introduction
O. Wilk

Introduction

Des bases

Commentaires, chaînes
Types et opérations
Entrée - Sortie
Fonctions

Le calcul scientifique

Numpy
Scipy

La visualisation

Matplotlib
Mayavi

Bibliographie

Pour commencer (2/2)

Respecter quelques règles :

- commencer en tout début de ligne (le décalage sert pour les test et les boucles),
- éviter les tabulations n'étant pas constituées d'espace d'un caractère.

Python,
une introduction
O. Wilk

Introduction

Des bases

Commentaires, chaînes
Types et opérations
Entrée - Sortie
Fonctions

Le calcul scientifique

Numpy
Scipy

La visualisation

Matplotlib
Mayavi

Bibliographie

Commentaires, chaînes de caractères (1/2)

```
CodePython.py
# -*- coding: utf8 -*-
# Mettre un commentaire dans le code python
print " Afficher un commentaire durant l'exécution."

# =====
# Une chaîne de caractères avec passage à la ligne

chaîne = """Le feu n'a plus de fumée
quand il est devenu flamme.
        Djalal al-Din Rumi"""

print chaîne
print " pour traiter une chaîne de caractères : "
print "      chaîne.find, chaîne.replace, chaîne.split, ..."

# =====
# Transformer un nombre en chaîne de caractères

a = 12345 ; b = -6789.01 ; c = 1.01e-6

print """ Ajouter une chaîne de caractères """+str(a)+"""
puis une autre """+str(b)+""" et la dernière """+str(c)+""". """

# =====
# Afficher en formatant, attention aux erreurs éventuelles

print " format s :      %s          %s          %s          " % (str(a), str(b), str(c))
print " format f : %12.2f %12.2f %12.2f          " % (a, b, c)
print " format d :      %6d          %6d          %6d          " % (a, b, c)
print " format e : %12.3e %12.3e %12.3e          " % (a, b, c)
```

Python,
une introduction
O. Wilk

Introduction

Des bases

Commentaires, chaînes
Types et opérations
Entrée - Sortie
Fonctions

Le calcul scientifique

Numpy
Scipy

La visualisation

Matplotlib
Mayavi

Bibliographie

Commentaires, chaînes de caractères (2/2)

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

```
[wilk@localhost ~]$ python CodePython.py
```

Afficher un commentaire durant l'exécution .

Le feu n'a plus de fumée
quand il est devenu flamme.

Djalal al-Din Rumi

pour traiter une chaîne de caractères :
chaîne.find, chaîne.replace, chaîne.split, ...

Ajouter une chaîne de caractères 12345 puis une autre -6789.01
et la dernière 1.01e-06.

```
format s :      12345      -6789.01      1.01e-06
format f :      12345.00   -6789.01      0.00
format d :      12345      -6789          0
format e :      1.234e+04   -6.789e+03  1.010e-06
```

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Types et opérations

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

```
CodePython.py
```

```
# -*- coding: utf8 -*-
print " Quelques operations "
print " 1.+2., 1.-2., 1.*2., 1./2., 2**2"
print " 1.+2., 1.-2., 1.*2., 1./2., 2**2"

print " Un complexe : "
a = complex(1.1,2.2)
print a.real, a.imag, type(a)

b = int(a.real) ; c = float(b) # Conversion ...

import math
print " Pi = %.60f" % (math.pi)
```

```
[wilk@localhost ~]$ python CodePython.py
```

```
Quelques operations
1.+2., 1.-2., 1.*2., 1./2., 2**2
3.0 -1.0 2.0 0.5 4

Un complexe :
1.1 2.2 <type 'complex'>

Pi = 3.1415926535897931159979634685441851615905761718750000000000000
```

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Tests et boucles (1/3)

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

```
CodePython.py
```

```
# -*- coding: utf8 -*-
# if

if ( 0 == 1 ) :
    # il faut indenter pour que Python sache que l'on est dans le test
    print " "
    print " On sort !"
    quit()
else :
    print " "
    print " tests : (0 < 1) ", (0 < 1), type((0 == 1))
    print " tests : (0 > 1) ", (0 > 1)
    print " tests : (0 == 1) and (1 == 1) ", (0 == 1) and (1 == 1)
    print " tests : (0 == 1) or (1 == 1) ", (0 == 1) or (1 == 1)
    # la fin de l'indentation marque la fin du test

print " On continue hors du test"
```

```
[wilk@localhost ~]$ python CodePython.py
```

```
tests : (0 < 1) True <type 'bool'>
tests : (0 > 1) False
tests : (0 == 1) and (1 == 1) False
tests : (0 == 1) or (1 == 1) True
```

On continue hors du test

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Tests et boucles (2/3)

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

```
CodePython.py
```

```
# -*- coding: utf8 -*-
# while

i = 0
while i < 3 :
    print i
    i = i + 1

while 1:
    try:
        x = float(raw_input(" Frapper au clavier un nombre
... (autre chose vous fera recommencer) : "))
        break
    except ValueError:
        print " Ce n'est pas correct ! Recommencez ..."

print x
```

```
[wilk@localhost ~]$ python CodePython.py
```

```
0
1
2
```

```
Frapper au clavier un nombre (autre chose vous fera recommencer) : azerty
Ce n'est pas correct ! Recommencez ...
Frapper au clavier un nombre (autre chose vous fera recommencer) : 123.1
```

```
123.1
```

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Tests et boucles (3/3)

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

CodePython.py

```
# -*- coding: utf8 -*-
# for

for i in xrange(0,5) :
    print i

print [j for j in xrange(0,5)]
```

[wilk@localhost ~]\$ python CodePython.py

```
0
1
2
3
4

[0, 1, 2, 3, 4]
```

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Entrée - Sortie (1/5)

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

CodePython.py

```
# -*- coding: utf8 -*-
# Récupérer des arguments en entrée

import sys # On importe un package
            # il va nous servir a récupérer des arguments d'entrée ...

Liste = sys.argv # On récupère les arguments d'entrée
              # python CeFichier.py arg1 arg2 ...

print " "
print " Les arguments sont dans la liste suivante : ", Liste
print [ Liste[j] for j in xrange(0,len(Liste)) ], type(Liste)
```

[wilk@localhost ~]\$ python CodePython.py 1 2

```
Les arguments sont dans la liste suivante : ['2a_In_Out.py', '1', '2']
['2a_In_Out.py', '1', '2'] <type 'list'>
```

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Entrée - Sortie (2/5)

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

CodePython.py

```
# -*- coding: utf8 -*-
# Récupérer des arguments en cours d'exécution

# Un nombre

x = input(" Frapper au clavier un nombre : ")

print x, type(x)

# Une chaîne de caractères

print " "
a = raw_input(" Frapper au clavier une chaîne de caractères (sans '') : ")
print a, a[0:3], a[1:3], type(a)
```

[wilk@localhost ~]\$ python CodePython.py 1 2

```
Fraper au clavier un nombre : 123.1
123.1 <type 'float'>
```

```
Fraper au clavier une chaîne de caractères (sans '') : azerty
azerty aze ze <type 'str'>
```

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Entrée - Sortie (3/5)

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

CodePython.py

```
# -*- coding: utf8 -*-

print " Ouverture du fichier en ecriture"

f = open('tmp.dat', 'w')

print " On écrit dans le fichier (avec des sauts de lignes)"

x = 1.1 ; y = 2

f.write( str(x) )
f.write( str(y) )
f.write( '\n' )
f.write( str(x) )
f.write( '\n' )
f.write( str(y) )

print " On ferme le fichier"

f.close()
```

[wilk@localhost ~]\$ cat tmp.dat

```
1.12
1.1
2
```

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Entrée - Sortie (4/5)

CodePython.py

```
# -*- coding: utf8 -*-
print " Différentes manières de lire "
print " V1"
f = open('tmp.dat', 'r')
print " ", f.read()
f.close()
print " V2"
f = open('tmp.dat', 'r')
print "1 ", f.readline()
print "2 ", f.readline()
f.close()
print " V3"
f = open('tmp.dat', 'r')
print " ", f.readlines()
f.close()
```

```
[wilk@localhost ~]$ python CodePython.py
```

```
Différentes manières de lire
V1
1.1 1.12
2
V2
1 1.12
2 1.1
V3
['1.12\n', '1.1\n', '2']
```

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Entrée - Sortie (5/5)

CodePython.py

```
# -*- coding: utf8 -*-
# Lire un fichier ascii contenant une matrice :
# voir aussi 'load', 'save', 'savez'

import numpy as np

a = np.loadtxt('mat.dat', delimiter = ',')

print type(a), a.shape, a.size
print " "
print a
```

```
[wilk@localhost ~]$ cat mat.dat
```

```
# commentaire
1, 2, 3
4, 5, 6
7, 8, 9
```

```
[wilk@localhost ~]$ python CodePython.py
```

```
<type 'numpy.ndarray'> (3, 3) 9
[[ 1.  2.  3.]
 [ 4.  5.  6.]
 [ 7.  8.  9.]]
```

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Fonctions (1/2)

CodePython.py

```
# -*- coding: Utf8 -*-
# Une fonction dans le corps du script ...
def Fibonacc1(n) :
    a, b = 0, 1
    for i in xrange(n) : a, b = b, a+b
    return b

print [Fibonacc1(i) for i in xrange(1,10)]

# importer une fonction qui n'est pas dans le même répertoire
import sys
sys.path.append("Bib_Perso") # On ajoute le le chemin du répertoire
import Fibonacc1_2

print [Fibonacc1_2.Fibonacc1_2(i) for i in xrange(1,10)]
```

```
[wilk@localhost ~]$ cat Bib_Perso/Fibonacc12.py
```

```
def Fibonacc1_2(n) :
    """Il est possible de documenter les fonctions
    etc ..."""
    a, b = 0, 1
    for i in xrange(n) : a, b = b, a+b
    return b
```

```
[wilk@localhost ~]$ python CodePython.py
```

```
[1, 2, 3, 5, 8, 13, 21, 34, 55]
[1, 2, 3, 5, 8, 13, 21, 34, 55]
```

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Fonctions (2/2)

CodePython.py

```
# -*- coding: Utf8 -*-
# Fonction à plusieurs variables in et out
def test(a, b) :

    c = a + b
    d = a * b

    return c, d

a = 2.0 ; b = 1.0

[c, d] = test(a, b)

print a, b, c, d

# exécuter des fonctions d'autres logiciels
# (hors python, sous shell unix/linux par ex)
import os

os.system('pwd')
```

```
[wilk@localhost ~]$ python CodePython.py
```

```
2.0 1.0 3.0 2.0

/home/wilk/Bureau/Python/Expose_LaTeX/Python_tests_LaTeX
```

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Python et le Calcul Scientifique

Python,
une introduction
O. Wilk

Introduction

Des bases

Commentaires, chaînes
Types et opérations
Entrée - Sortie
Fonctions

Le calcul scientifique

Numpy
Scipy

La visualisation

Matplotlib
Mayavi

Bibliographie

Numpy et Scipy

Python,
une introduction
O. Wilk

Introduction

Des bases

Commentaires, chaînes
Types et opérations
Entrée - Sortie
Fonctions

Le calcul scientifique

Numpy
Scipy

La visualisation

Matplotlib
Mayavi

Bibliographie

- Numpy : "the fundamental package needed for scientific computing with Python",
- Scipy : "a collection of mathematical algorithms and convenience functions built on the Numpy extension for Python",
- ...

Numpy

Python,
une introduction
O. Wilk

Introduction

Des bases

Commentaires, chaînes
Types et opérations
Entrée - Sortie
Fonctions

Le calcul scientifique

Numpy
Scipy

La visualisation

Matplotlib
Mayavi

Bibliographie

Numpy (1/5) : des tableaux.

Python,
une introduction
O. Wilk

Introduction

Des bases

Commentaires, chaînes
Types et opérations
Entrée - Sortie
Fonctions

Le calcul scientifique

Numpy
Scipy

La visualisation

Matplotlib
Mayavi

Bibliographie

```
[wilk@localhost ~]$ python
```

```
>>> import numpy as np
>>> a = np.array( [ (1.5,2,3), (4,5,6) ], dtype = 'float32' )
>>> print a
[[ 1.5  2.  3. ]
 [ 4.  5.  6. ]]

>>> a.ndim, a.shape, a.size, a.dtype
(2, (2, 3), 6, dtype('float32'))
>>> type(a)
<type 'numpy.ndarray'>

>>> b = np.zeros( (2,3), dtype = 'float32' )
>>> b
array([[ 0.,  0.,  0.],
       [ 0.,  0.,  0.]], dtype=float32)

>>> b = np.ones( (2,3), dtype = 'float32' )
>>> b
array([[ 1.,  1.,  1.],
       [ 1.,  1.,  1.]], dtype=float32)
```

Qq. types : int32, int64, float32, float64, complex64, complex128, ...

Numpy (2/5) : des opérations.

```
[wilk@localhost ~]$ python
```

```
>>> import numpy as np
>>> a = np.array( [ (1.5,2,3), (4,5,6) ], dtype = 'float32' )
>>> c = np.ones( (2,3), dtype = 'float32' )
>>> c
array([[ 1.,  1.,  1.],
       [ 1.,  1.,  1.]], dtype=float32)
>>> a*c                                # produit terme à terme
array([[ 1.5,  2. ,  3. ],
       [ 4. ,  5. ,  6. ]], dtype=float32)
>>> np.dot(a,c)                         # produit entre 2 matrices
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: matrices are not aligned
>>> np.transpose(a)
array([[ 1.5,  4. ],
       [ 2. ,  5. ],
       [ 3. ,  6. ]], dtype=float32)
>>> np.dot(np.transpose(a),c)
array([[ 5.5,  5.5,  5.5],
       [ 7. ,  7. ,  7. ],
       [ 9. ,  9. ,  9. ]], dtype=float32)
```

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Numpy (3/5) : des fonctions.

```
[wilk@localhost ~]$ python
```

```
>>> import numpy as np
>>> a = np.array( [ (1.5,2,3), (4,5,6) ], dtype='float32' )
>>> np.sum(a)                            # ou a.sum()
21.5
>>> np.min(a)                             # ou a.min()
1.5
>>> np.max(a)                             # ou a.max()
6.0
>>> np.mean(a)                            # ou a.mean()
3.5833333333333335
>>> np.median(a)                          # ou a.median()
3.5
>>> x = np.array([30,20,10])
>>> index = np.argsort(x)
>>> x[index]
array([10, 20, 30])
```

et aussi : np.exp, np.log, np.log10, np.sqrt, np.sin, np.cos, np.tan, np.arcsin, np.arccos, np.arctan, np.sign ...

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Numpy (4/5) : des affectations (attention).

```
[wilk@localhost ~]$ python
```

```
>>> import numpy as np
>>> x = np.array([30,20,10])
>>> y = x
>>> y is x
True
>>> y[1] = 22
>>> x
array([10, 22, 30])
>>> z = y.copy()
>>> z is y
False
>>> z[1] = 20
>>> y
array([10, 22, 30])
>>> z
array([10, 20, 30])
```

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Numpy (5/5) : des extractions.

```
[wilk@localhost ~]$ python
```

```
>>> import numpy as np
>>> a = np.array( [ (1.5,2,3,4), (4,5,6,7) ], dtype='float32' )
>>> a
array([[ 1.5,  2. ,  3. ,  4. ],
       [ 4. ,  5. ,  6. ,  7. ]], dtype=float32)
>>> a[:,1]
array([ 2. ,  5. ], dtype=float32)
>>> a[1,:]
array([ 4. ,  5. ,  6. ,  7. ], dtype=float32)
>>> a[1,0:3]
array([ 4. ,  5. ,  6. ], dtype=float32)
>>> a[1,1:3]
array([ 5. ,  6. ], dtype=float32)
```

et aussi : np.diag, np.linalg.det, np.linalg.eig, ...

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Scipy

Python,
une introduction
O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Scipy (1/5) : prolonge Numpy.

Python,
une introduction
O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

```
[wilk@localhost ~]$ python
```

```
>>> import numpy as np
>>> import scipy as sp

>>> np.sqrt(-1.), sp.sqrt(-1.)
(nan, 1j)

>>> np.log(-1.), sp.log(-1.)
(nan, 3.1415926535897931j)
```

...

Scipy (2/5) : des modules spécialisés.

Python,
une introduction
O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Sous module	Description
cluster	Clustering algorithms
constants	Physical and mathematical constants
fftpack	Fast Fourier Transform routines
integrate	Integration and ordinary differential equation solvers
interpolate	Interpolation and smoothing splines
io	Input and Output
linalg	Linear algebra
maxentropy	Maximum entropy methods
ndimage	N-dimensional image processing
odr	Orthogonal distance regression
optimize	Optimization and root-finding routines
signal	Signal processing
sparse	Sparse matrices and associated routines
spatial	Spatial data structures and algorithms
special	Special functions
stats	Statistical distributions and functions
weave	CC++ integration

Scipy (3/5) : ne pas oublier la doc.

Python,
une introduction
O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

```
[wilk@localhost ~]$ python
```

```
>>> import numpy as np
>>> import scipy as sp

>>> help()
```

Welcome to Python 2.7! This is the online help utility.

If this is your first time using Python, you should definitely check out the tutorial on the Internet at <http://docs.python.org/tutorial/>.

Enter the name of any module, keyword, or topic to get help on writing Python programs and using Python modules. To quit this help utility and return to the interpreter, just type "quit".

To get a list of available modules, keywords, or topics, type "modules", "keywords", or "topics". Each module also comes with a one-line summary of what it does; to list the modules whose summaries contain a given word such as "spam", type "modules spam".

```
help> scipy.linalg
```

...

Scipy (4/5) : ne pas oublier la doc.

Help on package scipy.linalg in scipy:

NAME
scipy.linalg

FILE
/usr/lib64/python2.7/site-packages/scipy/linalg/___init___py

DESCRIPTION
Linear algebra routines
=====

Linear Algebra Basics::

inv — Find the inverse of a square matrix
solve — Solve a linear system of equations
solve_banded — Solve a linear system of equations with a banded matrix
solveh_banded — Solve a linear system of equations with a Hermitian or symmetric banded matrix, returning the Cholesky decomposition as well
det — Find the determinant of a square matrix
norm — matrix and vector norm
lstsq — Solve linear least-squares problem
pinv — Pseudo-inverse (Moore-Penrose) using lstsq

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Scipy (5/5) : un exemple.

```
[wilk@localhost ~]$ python
```

```
>>> import numpy as np
>>> import scipy.sparse as sp_sp

>>> row = np.array([0,2,1,0,1,2])
>>> col = np.array([0,0,1,2,2,2])
>>> data = np.array([1,2,3,4,5,6])
>>> sp_sp.csc_matrix( (data,(row,col)), shape=(3,3) ).todense()
matrix([[1, 0, 4],
        [0, 3, 5],
        [2, 0, 6]])

>>> ptr_ind = np.array([0,2,3,6])
>>> ind_lig = np.array([0,2,1,0,1,2])
>>> data = np.array([1,2,3,4,5,6])
>>> sp_sp.csc_matrix( (data,ind_lig,ptr_ind), shape=(3,3) ).todense()
matrix([[1, 0, 4],
        [0, 3, 5],
        [2, 0, 6]])
```

... Il faut une séance complète pour présenter Scipy !

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Visualiser sous Python

- Matplotlib (courbes, images ... 2D),
- Mayavi (données sur maillages ... 3D (2D)),
- ...

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

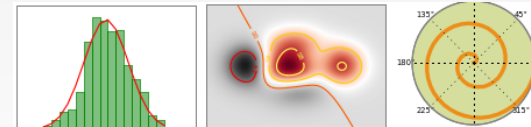
Matplotlib

Mayavi

Bibliographie

Matplotlib (1/4) : un point de vue.

- Visualisation 2D de qualité (png, pdf, ...),
- Matlab-like (pylab : pyplot + numpy),
- intégration de commandes LaTeX,
- doc. bien faite (<http://matplotlib.sourceforge.net/>).



Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Mayavi (2/3) : IHM (interactivité).

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

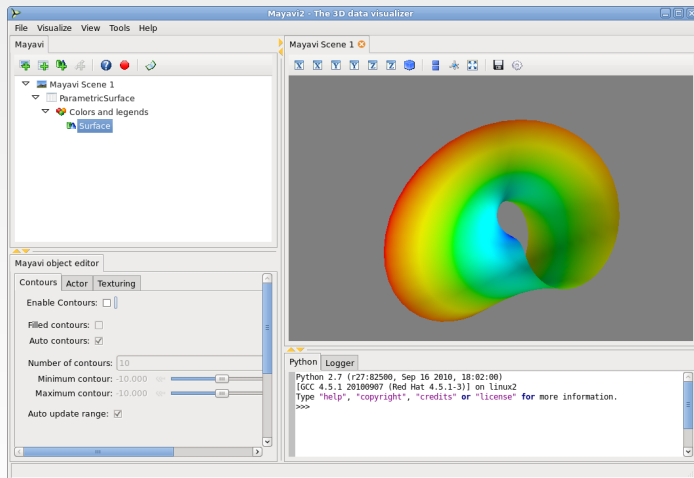
Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie



O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Mayavi (3/3) : une animation ds le corps du script.

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

```
[wilk@localhost ~]$ python
```

```
# -*- coding: utf8 -*-
import enthought.mayavi.mlab as mlab

x, y, z, tri, u = ...           # Coordonnées des sommets, connectivité
                               # des éléments triangles et champ associé.
Titre = ...                    # Chaînes de caractères donnant un titre
                               # à la figure

mlab.figure(size=(600,200))

ml = mlab.triangular_mesh(x, y, z, tri, scalars = u, colormap = 'Blues')

mlab.view( 0., 0., 1., [x0, y0, z0]) # Controle des paramètres "caméra"

mlab.colorbar(title = 'u', orientation = 'vertical')

mlab.title( Titre, size = 0.1, height = 0.9 )

mlab.savefig( 'image_0.png' )

mlab_tri = ml.mlab_source

for i in xrange(0,100):
    # Calcul de u
    u = ...
    # On change u dans la visualisation et on sauve une image
    mlab_tri.set( scalars = u ); mlab.savefig( 'image_'+str(i)'.png' )
```

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam

Bibliographie

Python,
une introduction

O. Wilk

Introduction

Des bases

Commentaires, chaînes

Types et opérations

Entrée - Sortie

Fonctions

Le calcul scientifique

Numpy

Scipy

La visualisation

Matplotlib

Mayavi

Bibliographie

Pierre Puiseux, Université de Pau et des Pays de l'Adour, Traduction française du tutorial écrit par Guido Van Rossum [http : // python.developpez.com / cours / TutoVanRossum /](http://python.developpez.com/cours/TutoVanRossum/).

Exposés journée Python à Autrans, 2010 : Sylvain Faure, Loïc Gouarin, Thierry Dumont, Pierre Raybaut

O. Wilk: Python, une introduction

Calcul scientifique/Math/Cnam