Introduction à Python

Intervenant : Jean-Christophe Gay

Mail : jean-christophe.gay@dauphine.fr

Bureau : B038

Plan du cours

Introduction

Jean-Christophe Gay :

- mail: jean-christophe.gay@dauphine.fr
- Bureau: B038 (de temps en temps et surtout le matin)

Expérience :

- 5 ans de développement en laboratoire
- 5 ans de développement pour l'Université Paris-Dauphine

Qu'est-ce que Python ?

Wikipédia :

> Python est un **langage de programmation** objet, multi-paradigme et multiplateformes. Il favorise la programmation **impérative structurée, fonctionnelle et orientée objet**. Il est doté d'un **typage dynamique fort**, d'une **gestion automatique de la mémoire** par ramasse-miettes et d'un système de gestion d'**exceptions** ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.
>
> Le langage Python est placé sous une **licence libre** proche de la licence BSD et fonctionne sur la plupart des plates-formes informatiques, des supercalculateurs aux ordinateurs centraux, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et aussi avec Java ou encore .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

Premiers pas en Python

Pourquoi Python ?

Version de Python ?

Installation de Python

Lancement de l'interpréteur

Une premiere opération

On quitte

Premiers pas en Python

Pourquoi Python ?

- Nous vivons dans un monde d'information
- Il y a beaucoup trop d'informations disponible à un moment donné
- Il faut un moyen de réduire l'information, particulièrement l'information financière

Premiers pas en Python

Pourquoi Python ?

- Python est gratuit ;
- Python est puissant, flexible et simple à apprendre ;
- Python est plus adapté au Big Data que R ;
- Python se compose de nombreux modules.

Premiers pas en Python

Pourquoi Python ?

Version de Python ?

- Il existe deux versions concurrentes de python : 2.7 et 3.6
- Les différences majeures sont :
  - print 'Hello World' fonctionne en version 2 mais pas en 3
  - les divisions entières sont différentes
  - plus sur http://sebastianraschka.com/Articles/2014_python_2_3_key_diff.html
- Pour ce cours les différences ne seront pas très importantes.

Premiers pas en python

Pourquoi Python ?

Version de Python ?

Installation de Python

Procédure d'installation :

- Se rendre sur le site de Python : www.python.org
- Section "Download" puis "Windows"
- Choisir l'installer qui convient

Premiers pas en python

Pourquoi Python ?

Version de Python ?

Installation de Python

Procédure d'installation :

- Se rendre sur le site de Python : www.continuum.io
- Section "Download" puis "Windows"
- Choisir l'installer qui convient

Premiers pas en python

Lancement de l'interpréteur

Dans un "terminal" lancer la commande python.

Cet interpréteur peut être utilisé pour lancer des commandes python et ainsi réaliser des calculs. Nous allons essayer rapidement.

Premiers pas en python

Lancement de l'interpréteur

Une premiere opération

Dans l'interpréteur :

```
>>> 1 + 1
2
>>> a = 1
>>> print a
1
>>> a + 2
3
```

Premiers pas en python

Lancement de l'interpréteur

Une premiere opération

On quitte

Dans l'interpréteur :

```
>>> quit
Use quit() or Ctrl-D (i.e. EOF) to exit
>>> quit()
```

Une première utilisation de Python

Si nous estimons qu'un retour de 100€ est attendu au bout d'un an avec une remise annuelle de 10%. La valeur actuelle d'une rentrée future d'argent est la suivante :

$$PV = \frac{FV}{(1+R)^n}$$

Dans cette équation PV représente la valeur présente et FV la valeur future, R est la remise et n le nombre de périodes.

Quelle est la valeur actuelle ?

```
>>> 100 / (1 + 0.1)
90.9090909090909090
>>>
```

Une première erreure

Que se passe-t-il pour la prochaine période ?

```
>>> 100 / (1+0.1)^2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  TypeError: unsupported operand type(s) for ^: 'float' and 'int'
>>>
```

En Python il faut utiliser '**' et non '^' pour les puissances.

```
>>> 100 / (1+0.1)**2
82.64462809917354
```

Attention à la case !

```
>>> x=2
>>> X
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'X' is not defined
>>>
```

Lorsque l'on nomme des variables ou des fonctions il faut faire attention à la case que l'on utilise. En effet Python est sensible à la case.

Type des variables

En Python les variables n'ont pas de type propre. Elles ont le type de la valeur qu'elles contiennent.

```
>>> x = 2
>>> x
2
>>> # x est un entier
>>> x="aa"
>>> x
'aa'
>>> # x est une chaîne de caractères
>>>
```

Trouver de l'aide

Facile : on en demande

```
>>> 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/2.7/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>
```

Trouver de l'aide

```
help> keywords

Here is a list of the Python keywords.  Enter any keyword to get more help.

and                 elif                if                  print
as                  else                import              raise
assert              except              in                  return
break               exec                is                  try
class               finally             lambda              while
continue            for                 not                 with
def                 from                or                  yield
del                 global              pass

help>
```

Trouver de l'aide en ligne

- Google + Stackoverflow
- Python Site
- Tutoriels divers…

Version de Python

- Avec un terminal : **python --version** python -V
- Avec Python

```
>>> import sys
>>> print sys.version
2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609]
```

Exercices

- Trouver l'aire d'un disque de diamètre 10 cm ;
- Trouver la longueur de la diagonale d'un carré de côté 1 ;
- Quelle est la circonférence d'un champs trapézoidal dont les côté font 127 mètres chacuns ?

Solutions

- Exercice 1

```
>>> aire=3.141592653*10**2
>>> aire
314.1592653
>>>
```

- Exercice 2

```
>>> diag=2**0.5
>>> diag
1.4142135623730951
```

- Exercice 3

```
>>> 127 + 127 + 127 + 127
508
```

Utilisation de Python en Quelques Lignes

Jouons avec les variables

Le module mathématique

Quelques fonctions utiles

Les tuples

Utilisation de Python en Quelques Lignes

Jouons avec les variables

- On assigne des valeurs aux variables :

```
>>> a = 1
>>> aa = 'Salut'
>>> b = 2
>>> a + b
3
```

- Afficher le contenu d'une variable :

```
>>> a
1
>>> aa
'Salut'
>>> print a
1
>>> print aa
Salut
```

Utilisation de Python en Quelques Lignes

Jouons avec les variables

- Quelques erreurs :

```
>>> aaa
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
NameError: name 'aaa' is not defined
```

```
>>> sqrt(1)
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
NameError: name 'sqrt' is not defined
```

Utilisation de Python en Quelques Lignes

Jouons avec les variables

- Du nommage des variables

```
>>> # Mauvais exemple
>>> x = 100
>>> y = 0.1
>>> z = x / ( 1 + y )
>>> print "Le résultat est", z
Le résultat est 90.9090909091
```

```
>>> # Bon exemple
>>> FV = 100
>>> R = 0.1
>>> PV = FV / ( 1 + R )
>>> print "Le résultat est", PV
Le résultat est 90.9090909091
```

Utilisation de Python en Quelques Lignes

Jouons avec les variables

Introspection

```
>>> dir
['__builtins__', '__doc__', '__name__', '__package__']
```

dir() permet d'accèder à l'ensemble des nom définis à un instant

```
>>> a = 1
>>> dir()
['__builtins__', '__doc__', '__name__', '__package__', 'a']
```

dir(a) permet d'avoir les fonctions applicable à a :

```
>>> type(a)
<type 'int'>
>>> dir(a)
['__abs__', '__add__', '__and__', '__cmp__',  ...
'denominator', 'imag', 'numerator', 'real']
```

Utilisation de Python en Quelques Lignes

Jouons avec les variables

Suppression d'une variable

```
>>> dir
<built-in function dir>
>>> dir()
['__builtins__', '__doc__', '__name__', '__package__']
>>> maVariable=1
>>> dir()
['__builtins__', '__doc__', '__name__', '__package__', 'maVar
>>> del(maVariable)
>>> dir()
['__builtins__', '__doc__', '__name__', '__package__']
```

Utilisation de Python en Quelques Lignes

Jouons avec les variables

Le module mathématique

- Les opérations de base

```
>>> # Addition
>>> 1 + 1
2
>>> # Soustraction
>>> 1 - 2
-1
>>> # Multiplication
>>> 2 * 2
4
>>> # Divisions
>>> 3 / 2
1
>>> 3 / 2.
1.5
>>> 3//2.
1.0
>>> int(2.5)
2
```

Utilisation de Python en Quelques Lignes

Jouons avec les variables

Le module mathématique

- Plus de mathématiques, le module

```
>>> dir()
['__builtins__', '__doc__', '__name__', '__package__']
>>> import math
>>> dir()
[... , 'math']
>>> dir(math)
['__doc__', '__name__', '__package__', 'acos', 'acosh',
... , 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'trunc']
>>> math.pow(2,2)
4.0
```

Utilisation de Python en Quelques Lignes

Jouons avec les variables

Le module mathématique

- Plus de mathématiques, le module

```
>>> help(pow)
Help on built-in function pow in module __builtin__:

pow(...)
    pow(x, y[, z]) -> number

    With two arguments, equivalent to x**y.  With three argume
    equivalent to (x**y) % z, but may be more efficient (e.g.

>>> pow(3,10,4)
1
>>> 3**10
59049
>>> 59049%4
1
```

Utilisation de Python en Quelques Lignes

Jouons avec les variables

Le module mathématique

- Choisir la précision

```
>>> 3/7.
0.42857142857142855
>>> payement=3/7.
>>> pay2=round(payement,4)
>>> pay2
0.4286
```

- Mais attention...

```
>>> payement*pow(10,6)
428571.4285714285
>>> pay2*pow(10,6)
428600.0
```

Utilisation de Python en Quelques Lignes

Jouons avec les variables

Le module mathématique

- e, Pi, log et exp

```
>>> math.e
2.718281828459045
>>> math.pi
3.141592653589793
>>> math.log(math.e)
1.0
>>> math.log(math.exp(2))
2.0
```

Utilisation de Python en Quelques Lignes

Jouons avec les variables

Le module mathématique

Une note sur import

- import math

```
>>> dir()
['__builtins__', '__doc__', '__name__', '__package__']
>>> import math
>>> dir()
[..., 'math']
```

- from math import

```
>>> dir()
['__builtins__', '__doc__', '__name__', '__package__']
>>> from math import pow
>>> dir()
[..., 'pow']
>>> from math import *
>>> dir()
[..., 'acos', 'acosh', 'asin', 'asinh', 'atan',
..., 'sqrt', 'tan', 'tanh', 'trunc']
```

Utilisation de Python en Quelques Lignes

- Affichage : print

```
>>> import math
>>> print 'Pi =', math.pi
>>> print math.pi
>>> print "Je crois que Pi vaut %f, enfin je crois, \
... ou alors c'est %f ?" % (math.pi,  math.e)
```

- Savoir de quel type est une variable

```
>>> x = 1
>>> type(x)
>>> x = float(x)
>>> type(x)
>>> if isInstance(x, (int, float)):
...     print "X est un nombre"
```

Utilisation de Python en Quelques Lignes

- Combinaison de chaînes de caractères

```
>>> a = 'Hello'
>>> b = 'World'
>>> a+b
>>> a+' '+b
>>> print "%s %s" % (a,b)
```

- Strip et upper

```
>>> x = '  hello '
>>> print x.upper()
>>> print x.capitalize()
>>> print x.strip()
>>> print x.strip().capitalize()
```

Utilisation de Python en Quelques Lignes

Les tuples

```
>>> t = ('JC', 33)
>>> print t
>>> len(t)
>>> t[0]
>>> type(t[1])
>>> type(t)
>>> print "My name is %s and my age is %d." % t
```

Exercices

- Comment trouver toutes les fonctions built-in ?
- Convertir 'Cet exercice est facile' en lettres majuscules.
- Nous avons 41 personnes dans la classe. Si nous devons faire des groupes de 3 pour des projets, combien de groupes et combien de personnes dans un groupe non entier ?
- Expliquer le résultat suivant :

```
>>> x=5.566
>>> round(x,2)
5.57
```

Solutions

- Exercice 4

```
>>> dir(__builtins__)
```

- Exercice 5

```
>>> s='Cet exercice est facile'.upper()
>>> print s
```

- Exercice 6

```
>>> print 'Avec %d élèves ont peut faire %d \
... groupes de %d personnes et il en restera %d.' % (41,41/3,3,41%3)
Avec 41 élèves ont peut faire 13 groupes de 3 personnes et il en restera 2.
```

Solutions

- Exercice 7 C'est le fonctionnement classique de l'arrondi en mathématique :

Si $x - int(x) < 0.5$
Alors $round(x) = int(x)$
Sinon $round(x) = int(x) + 1$

Programmation Fonctionnelle

- Une première fonction
- Notre propre module
- Un peu de mathématiques
- Utilisation de notre module

A suivre...

Bibliographie

- Python for Finance, Yuxin Yan, PACK, date ?
- Automate the boring stuff with Python
- Python for Finance, O'Reilly
- Mettre au moins un autre livre