



PRESENTATION DE PYTHON



I) Introduction

Python est un langage de programmation. C'est un logiciel libre et gratuit. Il a été développé au début des années 90. Il a connu plusieurs versions (la dernière version est Python 3.3.2 ... mais on utilisera aussi cette année Python 2.7.3 ou 2.7.5...)

La licence appartient actuellement à une association à but non lucratif : elle est en charge de valider ou non les évolutions proposées par des développeurs volontaires...

Outre le logiciel, Python se caractérise par une multitude de programmes librement accessibles par les utilisateurs... et ces programmes proviennent d'ailleurs des utilisateurs les plus performants. Ces programmes sont regroupés dans des "bibliothèques" (ou paquetages) qui sont soit inclus dans la librairie de base fournie avec le logiciel (comme l'interface graphique "Tkinter" ou le module "math" par exemple) soit à récupérer sur des sites plus spécialisés dans un domaine (comme les modules "Numpy" pour le calcul numérique, "Scipy" pour les applications scientifiques, "SymPy" pour le calcul formel, "Mathplotlib" pour le graphisme scientifique, "Pygame" pour des applications aux jeux.....)

II) Interpréteur Python

Une utilisation basique de Python est sous forme interactive sur une "console" Python dans laquelle les commandes sont exécutées aussitôt. On utilise alors Python comme une calculatrice évoluée.

Présentation de calculs et d'expressions effectués sur la console....

Les lignes de commande commencent par un "prompt" `>>>`. Les lignes situées juste en dessous correspondent aux réponses après exécution de l'instruction et de sa validation par la touche <Enter>.

En Python, les commentaires commencent par le symbole #. Sur une ligne, tout ce qui est écrit après ce # n'est pas exécuté lors d'une validation de la ligne.

Certaines opérations ou fonctions sont directement accessibles sous Python : +, *, -, **, /, //, %, int, print... D'autres sont dans une bibliothèque math que l'on charge avec la commande :

```
>>> from math import *
```

C'est le cas des fonctions usuelles : cos, sin, tan, tanh, cosh, sinh, atan, acos, asin, log, exp, sqrt

On peut travailler avec des chaînes de caractères, encadrées par " ou ', avec des nombres complexes... Python utilise la lettre j pour les nombres complexes. Certaines fonctionnalités sur les nombres complexes sont directement accessibles dans la bibliothèque de base : complex, abs, real, imag. D'autres sont accessibles dans la bibliothèque cmath : rect, phase, polar...



III) Environnement de développement Python

Une fois effectués ces calculs sur la console, on se rend vite compte qu'on ne peut pas sauvegarder notre travail. En fait, il faut utiliser un éditeur de texte (IDE, integrated development environment). Python en fournit un (IDLE basé sur Tkinter) mais d'autres sont disponibles (Spyder, IPython...). Ces programmes ouvrent deux fenêtres : une fenêtre "Python Shell" qui est une console améliorée et dont les commandes sont immédiatement exécutées, et une fenêtre de développement qui permet de taper les différentes commandes, les enregistrer, les débbuger, puis les exécuter dans la fenêtre Shell...

Une autre possibilité est d'installer des distributions complètes qui prennent en compte une fenêtre Python interactive, un éditeur de texte, une page d'aide, et qui intègrent des des bibliothèques spécifiques etc. Les plus courantes pour les Sciences sont Anaconda, WinPython, Python(x,y).

Cette année nous travaillerons avec Pyzo, qui fonctionne avec Python 3.4 et intègre les bibliothèques Numpy, Scipy, Matplotlib et Sympy

IV) Exercice

- 1) Calculer $1 - 1/3 - 1/3 - 1/3$. Qu'en concluez vous ?
- 2) Calculer la partie réelle, la partie imaginaire, le module du nombre complexe $3 + 5i$
- 3) A l'aide de la fonction print, faites afficher la phrase : J'ai dit "Non"
- 4) Affecter à la variable a la chaîne de caractères : "Demain".
Que vaut $a[1:3]$? $a[-1]$? $a[:3]$? $a[2:]$?