

Introduction à la programmation scientifique

Ahmed Ammar et Hassen Ghalila

Faculté des Sciences de Tunis, Université de Tunis El Manar

Contents

0.1	Langage Python	1
0.1.1	Avantages:	2
0.1.2	Inconvénients:	2
0.2	Installation d'un environnement Python scientifique	3
0.3	Documents et sites Web sur Python	3
0.4	Cours sur github:	3
0.5	Version de Python et bibliothèques utilisées	3

#Juste pour savoir la dernière fois que cela a été exécuté:

```
import time
print(time.ctime())
```

Mon May 8 02:05:43 2017

0.1 Langage Python

[Python] (<http://www.python.org/>) est un langage de programmation moderne de haut niveau, orienté objet et d'usage général.

Caractéristiques générales de Python:

- **** Langage simple:** ****** facile à lire et à apprendre avec une syntaxe minimaliste.
- **** Langage concis et expressif:** ****** moins de lignes de code, moins de bugs, plus facile à maintenir.

Détails techniques:

- **** Typé dynamiquement:** ****** Pas besoin de définir le type des variables, les arguments ou le type des fonctions.
- **** La gestion automatique de la mémoire:** ****** Aucune nécessité d'allouer explicitement et désallouer la mémoire pour les variables et les tableaux de données. Aucun bug de fuite de mémoire.
- **** Interprété:** ****** Pas besoin de compiler le code. L'interpréteur Python lit et exécute le code python directement.

0.1.1 Avantages:

- Le principal avantage est la facilité de programmation, qui minimise le temps nécessaire pour développer, déboguer et maintenir le code.
- Langage bien conçu qui encouragent les bonnes pratiques de programmation:
- Modulaire et orientée objet, permet l'encapsulation et la réutilisation de code. Il en résulte souvent un code plus transparent, maintenable et sans bug.
- Documentation intégré avec le code.
- De nombreuses bibliothèques standards, et de nombreux packages add-on.

0.1.2 Inconvénients:

- Puisque Python est un langage de programmation interprété et typé dynamiquement, l'exécution de code python peut être lent par rapport à des langages de programmation compilés à typage statique, tels que C et Fortran.
- Un peu trop décentralisé, avec différents environnements, bibliothèques, et documentation répartis à différents endroits. Cela peut le rendre difficile pour commencer.

0.2 Installation d'un environnement Python scientifique

[Anaconda CE](#). Anaconda Community Edition is free.

0.3 Documents et sites Web sur Python

- [Python](#). The official Python web site.
- [Python tutorials](#). The official Python tutorials.
- [Think Python](#). “How to Think Like a Computer Scientist” by Allen B. Downey (free book).
- [Python Course](#). This website contains a free and extensive online tutorial by Bernd Klein, well suited for self-learning. However, you can attend one of his Python courses in Paris, London, Toronto, Berlin, Frankfurt, Hamburg Munich or Lake Constance.

0.4 Cours sur github:

- “[Scientific Python Lectures](#)” by Robert Johansson
- “[Anatomy Of Matplotlib](#)” by Benjamin Root
- “[Python-lectures-Notebooks](#)” by Christophe Morisset

0.5 Version de Python et bibliothèques utilisées

```
print ("\t\tSystème utilisé")
import sys
print("Système :\t\t",sys.platform)
import platform
print(platform.platform())
print("Ordinateur:\t\t",platform.machine())
print("Version de Python:\t",sys.version)
import IPython
print("Version de IPython:\t",IPython.__version__)
import numpy
print("Version de numpy:\t",numpy.version.version)
```

```
import scipy
print("Version de scipy:\t",scipy.version.version)
import matplotlib
print("Version de matplotlib:\t",matplotlib.__version__)
```

Systeme utilisé

Systeme : linux

Linux-4.8.0-51-generic-x86_64-with-debian-stretch-sid

Ordinateur: x86_64

Version de Python: 3.6.0 |Anaconda custom (64-bit)| (default, Dec 23 2016, 12:22:05)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)]

Version de IPython: 5.1.0

Version de numpy: 1.11.3

Version de scipy: 0.18.1

Version de matplotlib: 2.0.0