

# Analyse numérique & Mathématiques appliqués

Phitem 2017/2018

Gilles Delaygue / Lucas Davaze

[gilles.delaygue@univ-grenoble-alpes.fr](mailto:gilles.delaygue@univ-grenoble-alpes.fr)  
[lucas.davaze@univ-grenoble-alpes.fr](mailto:lucas.davaze@univ-grenoble-alpes.fr)

# Langage de programmation

C'est un langage permettant d'écrire des programmes informatiques ...

On peut donc le lire/comprendre/traduire si on apprend à parler cette "langue" !

Programmation **≠** Algorithmie

# Langage de programmation

```
def add5(x):  
    return x+5  
  
def dotwrite(ast):  
    nodename = getNodeName()  
    label=symbol.sym_name.get(int(ast[0]),ast[0])  
    print '      %s [label="%s' % (nodename, label),  
    if isinstance(ast[1], str):  
        if ast[1].strip():  
            print '= %s";' % ast[1]  
        else:  
            print '"]'  
    else:  
        print '"]';'  
    children = []  
    for n, child in enumerate(ast[1:]):  
        children.append(dotwrite(child))  
    print ', ' %s -> {' % nodename  
    for n, child in enumerate(children):  
        print '%s' % name,
```

# Compilés

- Le code source (écrit par vous !) est traduit par un compilateur (ex : *traduction d'un livre Anglais en Français*)

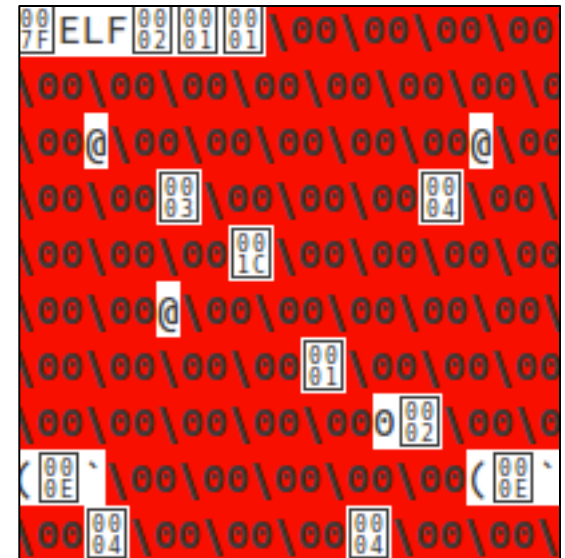
# Compilés

- Le code source (écrit par vous !) est traduit par un compilateur

```
#include <stdio.h>

int main()
{
    printf("hello world");
    return 0;
}
```

*Compilation*



# Compilés

- Le code source (écrit par vous !) est traduit par un compilateur (ex : *traduction d'un livre Anglais en Français*)
- C++ , Pascal, Fortran ...

# Compilés

+

- Le compilateur optimise le code source (+ efficace, + rapide)
- Une fois compilé, le programme peut-être utilisé sans le code source

-

- Pas nécessaire si vous faites de “petits” calculs

# Interprétés

- Le code source est “interprété” ligne par ligne (*ex : interpreteur radio*)
- Pour simplifier, l’interpréteur compile temporairement chaque ligne du code
- A chaque lancement du code, le code est “interprété”
- Python, Matlab, Scilab ...



# Interprétés

+

- + Facile à prendre en main qu'un langage compilé (correction du code ligne par ligne, le code s'exécute de "haut en bas")

-

- Beaucoup plus lent qu'un langage compilé

# Quel langage choisir ?

Aucun langage n'est véritablement le mieux ... !

Ils ont chacun leurs avantages/inconvénients et leur domaine d'utilisation du fait de leurs librairies

- R → Calculs scientifiques / statistiques
- Matlab/Scilab → Calculs scientifiques / matriciels
- Python → Calculs scientifiques / imagerie satellite
- Fortran → Calculs scientifiques / modèle physique complexes ...

Uniquement langage interprété :

- Matlab / Scilab
- Python

Avec un peu d'utilisation de tableur ... :

- OpenOffice Calc
- Excel

# Liens ...

- SCILAB :

- ✧ <https://www.scilab.org/fr/download/latest> (Pour télécharger la dernière version de Scilab)
- ✧ [http://www.scilab.org/en/content/download/846/7841/file/livret\\_maths\\_2013.pdf](http://www.scilab.org/en/content/download/846/7841/file/livret_maths_2013.pdf) (Aide officielle Scilab)
- ✧ <https://openclassrooms.com/courses/prise-en-main-de-scilab> (Cours OpenClassRoom donnant de bonnes bases)

- PYTHON :

- ✧ <https://www.anaconda.com/download/#windows> (Lien pour la distribution Anaconda Windows. Vous pouvez changer pour Mac / Linux)
- ✧ <https://openclassrooms.com/courses/apprenez-a-programmer-en-python> (La partie #1 offre de bonnes bases et montre des commandes que nous utiliserons)
- ✧ <http://pythontutor.com/visualize.html#mode=edit> (Site où vous pouvez lancer votre code pour voir ligne a ligne ce qu'il fait. Très instructif !!)

# Python, les bases ...

- Python est un langage interprété
- Plusieurs IDE (Integrated Development Environment) existent
  - Spyder (suite Anaconda)
  - PyCharm
  - Enthought Canopy
- Contrairement à Matlab, les libraires ne sont pas importées automatiquement

# Python, les bases ...

```
In [1]: sqrt(10)
```

```
Traceback (most recent call last):
```

```
File "<ipython-input-1-7695ba7a3161>", line 1, in <module>  
    sqrt(10)
```

```
NameError: name 'sqrt' is not defined
```

```
In [2]: import math
```

```
In [3]: math.sqrt(10)
```

```
Out[3]: 3.1622776601683795
```

# Python, les bases ...

- Différent de Matlab / Scilab que l'on télécharge avec toutes les librairies déjà importées
- Anaconda est livré avec + de 200 paquets
- Les + utilisés :
  - Numpy (gère les matrices + calculs scientifiques)
  - Math (contient toutes les fonctions mathématiques de base)
  - Matplotlib (permet de faire des graphes)
  - Os, Sys (permettent de gérer les fichiers de votre ordinateur)

# Python, les bases ...

- Si cela ne suffit pas ... (souvent la solution existe avec les 4 précédents ...)
- Possibilité d'en installer des nouveaux
- Si Anaconda,
  - “conda install “paquet”