

## Introduction language python

- 1) Comment utiliser python
    - a. Fichiers texte lancé à l'interpréteur
    - b. Lancer l'interpréteur
    - c. L'interpréteur interactif IPython
      - i. Les commandes magiques %whos, %run
    - d. Jupyter notebook
    - e. Un IDE (ex pycharm, spyder)
  - f. Quelques commandes utiles  
type(), dir(), id(), help()
- 2) Type de variables
    - a. types (voir poly d'introduction)
    - b. type <container>
      - i. list []
      - ii. tuple ()
      - iii. sets {}
      - iv. dictionnaires {}
      - v. commande append
    - c. Type numpy et la précision des nombres (**exo 1**)
      - i. Tableau 1D
      - ii. Tableau 2D

### Tableaux numpy

- d. Les indices de 0 à -1
- e. Pour les tableaux ordre de stockage /C et fortran (ou matlab)
- f. Le stockage interne (ordre / matlab), les types, vecteur et matrice
  - i. Quel intérêt ?
    1. Accélération d'une boucle
    2. Lien avec d'autres langages
  - g. Subtilité d'utilisation de sous partie de tableaux (slice etc)
  - h. Notion de calcul vectoriel / séquentiel ; calcul //
    - (**exo 4**)

0	1	2	3	4	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	---	---	----	----

```
arr.reshape((4, 3), order=?)
```

**C order (row major)**

0	1	2
3	4	5
6	7	8
9	10	11

order='C'

**Fortran order (column major)**

0	4	8
1	5	9
2	6	10
3	7	11

order='F'

- a. Mutable, immutable ??? (**exo 2**)
- b. Boucle for (**exo 3**)

3) Structure programme

- a. Scope ou espace de nommage ou spacename
  - i. Variable locale, globale
- b. Programme principal et sous programmes/fonction
  - i. Bibliothèque
- c. Sous python structure en module
  - i. Quels modules pour nous
  - ii. Comment les appeler
    - 1. From NN import \*
    - 2. Import NN
    - 3. Import NN as LeModuleQuiMinteresse

4) Notion de classe

- a. Histoire C, C++, java
- b. Membre et méthodes
- c. Constructeur, destructeur
  - i. Classe publique, privée, etc
  - ii. En python tout est classe
  - iii. Dir() pour le contenu d'une classe
  - iv. Comment nommer les classes
  - v. Héritage
  - vi. Overloading

5) Digression sur la mémoire langage/machine

- a. Deux classes (objets) peuvent pointer sur la même zone mémoire
- b. !!! Attention sous python, la copie d'un tableau vers un autre ne signifie pas forcément duplication, si on modifie l'un l'autre est modifié également !!!
- c. pas d'allocation, pas de libération de mémoire, le garbage collector

6) Entrée sortie fichiers

- a. Distinguer fichiers binaires et ascii, c'est quoi ?
- b. Fichier binaire, utilisation de numpy pour éviter les conversions
- c. Fichier ascii

7) Dessin

- a. Exercice

8) Créer une classe avec une constructeur, méthode et méthode de classe

- i. Méthode d'instance versus méthode de classe (**note 5**)  
@classmethod