

TP d'introduction à Python

Pour préparer votre séance de TP, taper les commandes suivantes dans un terminal

```
mkdir TP-Python
cd TP-Python
spyder
```

La commande `mkdir` sert à créer un répertoire ; la commande `cd` permet de se placer dans un répertoire.

Voici quelques commandes pour commencer à taper dans l'interpréteur (dans la `console`). Cherchez à comprendre le résultat renvoyé.

a) Affectation d'une variable.

```
>>> x=10
>>> y=5
```

b) Affectation d'un tableau.

```
>>> import numpy as np
>>> a=np.eye(5,5)
>>> print a
>>> x=np.arange(0,10,1)
>>> x
>>> y=np.arange(1,8,2)
>>> print(y)
>>> z=np.array([0, 1, 2, 5])
>>> print(z)
>>> v=np.array([[1,2],[2,3],[3,4]])
>>> print(v)
>>> type(v)
```

c) Opérations sur les vecteurs.

```
>>> import numpy as np
>>> x=np.arange(0,11,1)
>>> y=2*x
>>> np.size(x)
>>> np.shape(x)
>>> x[0]
>>> pow(x,2)
>>> x[1:]
>>> x[2:5]
>>> x[2:2:6]
>>> x=x[2:-1]
>>> y=y[1:-2]
>>> x+y
>>> np.vdot(x,y)
>>> np.dot(x,y)
```

d) Messages d'erreur : tapez l'une après l'autre les commandes suivantes, qui contiennent des erreurs, et lisez les messages d'erreur.

```

>>> toto
>>> import numpy as np
>>> x=np.arange(0,11,1)
>>> x(16)
>>> y=2*x
>>> x+y[2:]
>> size(x)

```

e) Boucle for : que font les instructions suivantes :

```

>>> for i in range(1,11):
...     print(i)
...
>>> a=0
>>> for i in range(1,11):
...     a=a+i
...     print(a)
...
>>> a=1
>>> for i in range(1,11,2):
...     a=a*i
...     print(a)
...
>>> for i in range(1,12,3):
...     print i
...
>>> u=[1,4,7,10]
>>> for i in u:
...     print(i)
...

```

f) Fonctions mathématiques : tapez les commandes suivantes et observez les résultats.

```

>>> import numpy as np
>>> import math
>>> pow(math.sin(2.5),2)+pow(math.cos(2.5),2)
>>> pow(np.sin(2.5),2)+pow(np.cos(2.5),2)
>>> math.exp(math.log(3))
>>> math.sqrt(-1)
>>> np.sqrt(-1)

```

A partir de maintenant vous allez écrire vos programmes dans des fichiers exécutables par Python. Pour cela, utilisez l'éditeur de Spyder et créez un fichier **exoNumero.py** pour chaque exercice.

Exercice 1 Que font les commandes (on aura préalablement importé **numpy** sous le nom **np**)

`np.ones((3,1)), np.ones((1,3)), np.ones(3), np.eye(3), np.eye(3,2)?`

Exercice 2 Définir de la façon la plus simple possible les vecteurs

$v_1 = (5, 9, 1, 3)$
 $v_2 = (2, 2, 2, \dots, 2)$ de taille 20
 $v_3 = (1, 2, 3, 4, \dots, 15, 16)$
 $v_4 = (28, 26, 24, 22, 20, 18, 16, 14, 12)$

On utilisera pour cela les fonctions **array**, **ones**, **arange** et **linspace** du module **numpy**.

Exercice 3 Définir une matrice et un vecteur numpy de la manière suivante

```
import numpy as np
A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
b = np.array([-3, -2, -1])
```

Question 1.

Quels sont les types des données de A et de b ?

Question 2.

Faire le produit matrice-vecteur et afficher le résultat.

Exercice 4 Définissez les matrices

$$A = \begin{bmatrix} 7 & 0 \\ -1 & 5 \\ -1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 4 \\ -4 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 7 \\ -3 \end{bmatrix}, \quad D = \begin{bmatrix} 8 & 2 \end{bmatrix}.$$

Calculez avec Python les produits AB , AC , CA , DC , DBC , $A^T A$ et AA^T en faisant appel à la fonction `dot` de `numpy`. La transposée d'une matrice se calcule avec la fonction `transpose`. Essayez de comprendre les éventuels messages d'erreur.

Exercice 5 On définit les vecteurs u_1, u_2, u_3 de $\mathbb{R}^5$ par

$$u_1 = \begin{bmatrix} 1 \\ -3 \\ 3 \\ 5 \\ 4 \end{bmatrix}, \quad u_2 = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 4 \\ 3 \end{bmatrix}, \quad u_3 = \begin{bmatrix} 2 \\ -5 \\ -1 \\ -6 \\ 1 \end{bmatrix}.$$

Construire la matrice A dont les colonnes sont formées des vecteurs $u_1, \dots, u_3$.

Exercice 6 Définir la fonction qui à x fait correspondre $\sin(x)$. L'appliquer au calcul du sinus de π puis au sinus du vecteur $(0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi)$.

Exercice 7 Un triangle quelconque peut être caractérisé par ses trois sommets $(x_1; y_1)$, $(x_2; y_2)$ et $(x_3; y_3)$. L'aire de ce triangle peut être calculé à l'aide de la formule suivante

$$aire = \frac{1}{2} | (x_1 - x_3)(y_2 - y_1) - (x_1 - x_2)(y_3 - y_1) |$$

Ecrire une fonction qui renvoie l'aire d'un triangle dont on donne les coordonnées des trois sommets en argument (par exemple `[ [0, 0], [0, 1], [1, 0] ]`).

Exercice 8 [Tracé de fonctions].

Importer le module `matplotlib.pyplot` et utiliser la fonction `plot` de ce module pour tracer les graphes des fonctions suivantes.

On rappelle que pour représenter le graphe d'une fonction f sur un intervalle $[a, b]$, il faut tout d'abord la «discrétiser»; c'est-à-dire définir une liste de points $(x_i, f(x_i))$ qui va permettre d'approcher le graphe de la fonction par une ligne brisée. Bien sur, plus on augmente le nombre de points, plus la ligne brisée est «proche» du graphe (en un certain sens). Plus concrètement, on commence par construire un vecteur X qui discrétise l'intervalle $[a, b]$, en considérant un ensemble de points équidistants dans $[a, b]$. Pour ce faire on se donne un entier N grand et on définit $X = (a, a + h, a + 2h, \dots, b - h, b)$ où $h = \frac{b-a}{N-1}$.

- i) Représenter graphiquement la fonction f définie par : $f(x) = \cos^2(x)$, $x \in [0, 5]$.
Pour le tracé, utiliser 6 points, puis 26 et enfin 301 points équirépartis dans le segment $[0, 5]$.
Préciser dans chaque cas le pas de la subdivision utilisée.
- ii) Représenter sur la même figure les fonctions :
 $f(x) = \cos^2(x)$, $g(x) = \cos(2x)$, $h(x) = \cos(x^2)$, $x \in [0, 5]$.
Penser à utiliser des symboles différents pour les différentes représentations graphiques et à mettre une légende.

Exercice 9 .

Soient f_1 , f_2 , f_3 et f_4 les fonctions définies par $f_1(x) = \sin(x)$, $f_2(x) = \sin(2x)$, $f_3(x) = \sin(3x)$, $f_4(x) = \sin(4x)$, $x \in \mathbb{R}$.

- i) Représenter sur la même figure les fonctions f_1, f_2, f_3 et f_4 sur l'intervalle $[0, 4\pi]$.
- ii) On pose $g(x) = \sum_{k=1}^4 \sin(kx)$, $x \in \mathbb{R}$. Dans la figure utilisée à la question 1, tracer la courbe représentative de g pour $x \in [0, 4\pi]$.