

TP1 : Introduction à Python

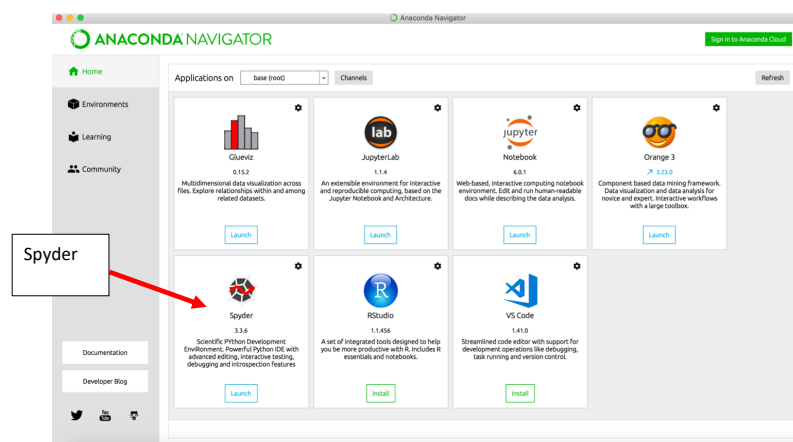
1 Lancement d'Anaconda

Anaconda est une distribution Python gratuite que vous pouvez aisément installer chez vous! anaconda). Pour télécharger Anaconda, rendez vous sur www.anaconda.com
Prenez la version qu'il vous faut :

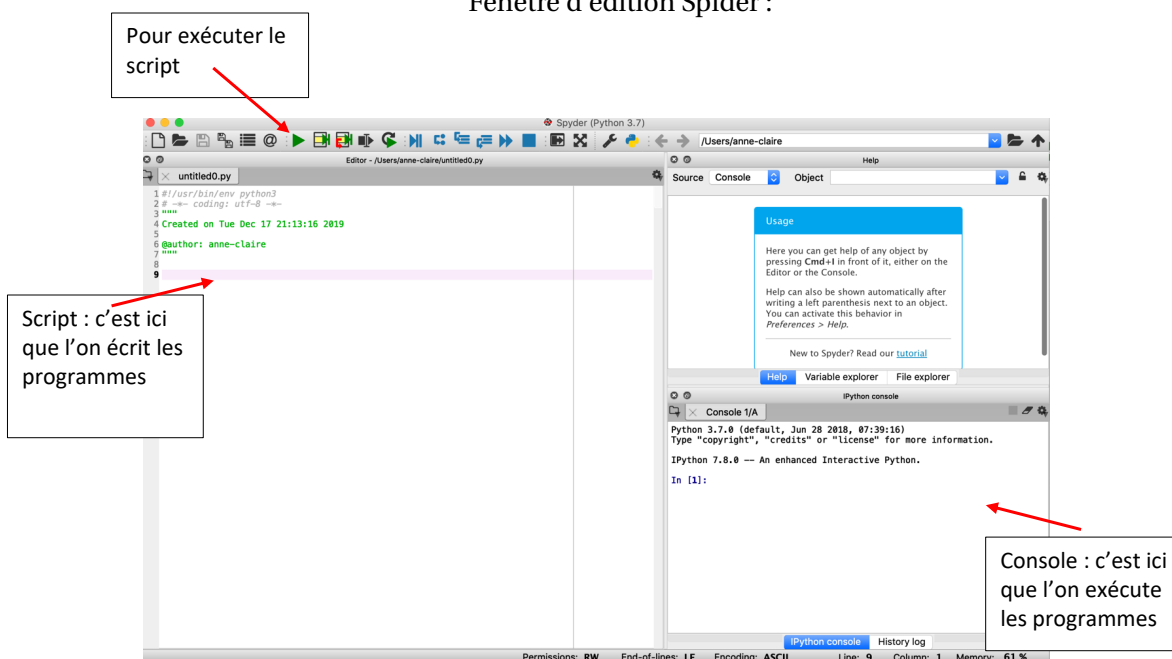
1. Choisissez le système d'exploitation cible (Windows, Mac, etc...)
2. Sélectionnez la version 3.11

À l'issue de l'installation, vous aurez Anaconda Navigator. Ce dernier est une interface graphique permettant de lancer Spyder qui est l' IDE (Integrated Development Environment) pour éditer du code Python que nous allons utiliser.

Navigateur Anaconda :



Fenêtre d'édition Spider :



C'est dans cet espace que nous allons travailler. Nous pourrions, entre autres :

1. Editer un script dans lequel seront écrits les programmes. Un script est enregistré avec l'extension .py (à gauche dans la fenêtre d'édition de Sider).
2. Visualiser les résultats après l'exécution des programmes (dans la console ou l'explorateur de variables).
3. Utiliser le débogueur.

Nous allons utiliser la bibliothèque PyLab pour effectuer de la programmation scientifique avec Python. Pour cela, il faut importer le package pylab au début de chaque script en tapant l'instruction suivante : **from pylab import ***

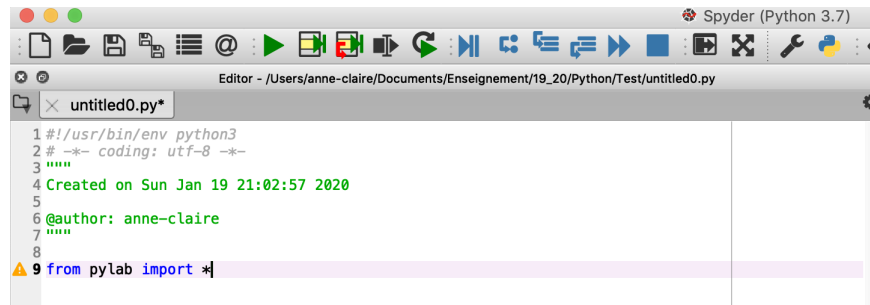


FIGURE 1 – On importe la bibliothèque Pylab au début de chaque programme

Python utilise l'indentation pour délimiter les blocs de codes (boucles, instruction conditionnée...). Cela contribue à la lisibilité et permet une meilleure lecture et compréhension du code. Cependant, il faut faire très attention à l'utilisation des espaces car une mauvaise indentation conduit à des erreurs dans le code.



Il faut veiller à la bonne indentation du programme!

2 Vecteurs et matrices

Indications sur la fonction *print* :

- `print('coucou')` ou `print('A[0]=')` permettent d'afficher, dans la console de spyder, les chaînes de caractères « coucou » et « A[0]= ».
- Pour afficher le contenu d'une variable ou le résultat d'une opération, il ne faut pas mettre « ' ».
- `print('A[0]= \n', A[0])` permet d'afficher « A[0]= », de sauter une ligne, puis d'afficher le résultat de `A[0]`.

Le cours : Extraire les coefficients d'un vecteur (avec l'enseignant)

1. Dans un script, saisir puis exécuter les 2 commandes suivantes :
`A = array([7, 2, 1, 3, 4, 6, 8, 1, 9])`
`print('A =', A)`
2. Exécutons maintenant les commandes suivantes dans la console puis analysons les résultats :
 - (a) `A[0]` renvoie
 - (b) `A[0 : 3]` renvoie
 - (c) `A[2 : 6]` renvoie
 - (d) `A[: 6]` renvoie
 - (e) `A[2 :]` renvoie
 - (f) `A[-1]` renvoie
 - (g) `A[1 : 8 : 2]` renvoie
 - (h) `A[:: -1]` renvoie

Exercice 1 : Extraire les coefficients d'un vecteur (à vous de jouer!)

1. Dans un script, saisir le code permettant de créer le vecteur $B = [3, -2, 11, 7, 9, 36, -8, 0, 5]$
2. Déterminer et exécuter les commandes qui permettent d'extraire les listes suivantes à partir du vecteur B :
 - (a) $\rightarrow [36]$
 - (b) $\rightarrow [11, 7, 9, 36, -8]$
 - (c) $\rightarrow [7, 9, 36, -8, 0, 5]$
 - (d) $\rightarrow [3, -2, 11, 7]$
 - (e) $\rightarrow [3, 11, 9, -8, 5]$
 - (f) $\rightarrow [-2, 7, 36, 0]$
 - (g) $\rightarrow [5, 0, -8, 36, 9, 7, 11]$

Le cours : Fonctions Python permettant de créer une liste (avec l'enseignant)

1. `arange(3,8)` renvoie
2. `arange(3,8,0.2)` renvoie
3. `linspace(3,8,5)` renvoie
4. `linspace(8,3,4)` renvoie

Exercice 2 : Fonctions Python permettant de créer une liste (à vous de jouer!)

1. Prévoir, sans utiliser l'ordinateur, ce que renvoient les commandes suivantes :

`arange(2,4,0.5)``arange(2,3,0.3)``linspace(2,7,6)``linspace(2,2,7)`
2. Déterminer et exécuter les codes permettant d'obtenir les listes suivantes (plusieurs réponses possibles) :
 - (a) $\rightarrow [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]$
 - (b) $\rightarrow [3, 3.25, 3.5, 3.75, 4, 4.25, 4.5]$
 - (c) $\rightarrow [7, 6.5, 6, 5.5, 5, 4.5, 4]$
 - (d) $\rightarrow [0, 0, 0, 0, 0, 0, 0, 0, 0]$

Le cours : Les matrices (avec l'enseignant)

Pour stocker les matrices $A = \begin{pmatrix} 1 & 1 & 2 \\ 0 & 2 & 9 \\ 1 & 3 & 7 \end{pmatrix}$ et $B = \begin{pmatrix} 3 & 1 & 2 \\ 0 & 1 & 4 \\ 1 & 2 & 3 \end{pmatrix}$ dans des variables en Python on écrit :

A = array([[1, 1, 2], [0, 2, 9], [1, 3, 7]]) et **B = array([[3, 1, 2], [0, 1, 4], [1, 2, 3]])**

On peut alors

1. Faire les opérations matricielles de base :

- La somme : $A + B$
- La différence : $A - B$
- le produit de A par 2 : $2 * A$
- Le produit matriciel : $\text{dot}(A, B)$
- La transposée : $\text{transpose}(A)$

Attention : $A * B$ ne renvoie pas le produit matriciel!

$A * B$ renvoie

2. Extraire des coefficients d'une matrice :

- (a) $A[2, 1]$ renvoie
- (b) $A[0, 1]$ renvoie
- (c) $A[0 : 2, 2]$ renvoie
- (d) $A[:, 1 :]$ renvoie

3. Modifier les coefficients d'une matrice avec, par exemple, les commandes $A[2, 1] = 17$ ou $A[2, :] = 3$.

Exercice 3 : Opérations sur les matrices

1. Dans une script, écrire le code permettant de créer les matrices suivantes :

$$A = \begin{pmatrix} 9 & -2 & 2 \\ 0 & 7 & 3 \\ 7 & 3 & -4 \end{pmatrix} \quad \text{et} \quad B = \begin{pmatrix} -3 & 4 & -2 \\ 0 & 1 & 3 \\ 11 & 0 & 5 \end{pmatrix}$$

2. Calculer

- (a) $3A + 2B =$
- (b) $A^2 =$
- (c) $A^3 =$
- (d) ${}^t(B \times A) =$
- (e) ${}^tA \times {}^tB =$

3. Donner les commandes qui permettent d'extraire les matrices suivantes de la matrice B?

- (a) $\rightarrow \begin{pmatrix} -3 & 4 \\ 0 & 1 \end{pmatrix}$
- (b) $\rightarrow \begin{pmatrix} 0 & 1 & 3 \end{pmatrix}$
- (c) $\rightarrow \begin{pmatrix} -2 \\ 3 \\ 5 \end{pmatrix}$
- (d) $\rightarrow \begin{pmatrix} -3 & 4 \\ 11 & 0 \end{pmatrix}$

Le cours : Opérations sur les matrices (suite)

On considère les matrices $A = \begin{pmatrix} 9 & -2 & 2 \\ 0 & 7 & 3 \\ 7 & 3 & -4 \end{pmatrix}$ et $b = \text{array}([[10, 11, 12]])$.

1. Il est possible de d'ajouter ou supprimer une ligne ou une colonne à une matrice :
 - (a) $\text{insert}(A, 2, b, 1)$ renvoie
 - (b) $\text{insert}(A, 2, b, 0)$ renvoie
 - (c) $B = \text{concatenate}((A, b))$ renvoie
 - (d) $A = \text{delete}(A, 1, 0)$ renvoie
 - (e) $D = \text{delete}(A, 1, 1)$ renvoie
2. Il est aussi possible de coller l'une au-dessus de l'autre deux matrices ayant le même nombre de colonnes, de récupérer la diagonale d'une matrice et de récupérer les dimensions d'une matrice :
 - (a) $[n, p] = \text{shape}(D)$ renvoie
 - (b) $\text{diag}(A)$ renvoie
 - (c) $\text{diag}([1, 2, 3])$ renvoie
3. Enfin, les commandes *ones* et *zeros* permettent de créer des matrices :
 - (a) $\text{zeros}([2, 5])$ renvoie
 - (b) $\text{ones}([3, 4])$ renvoie

Exercice 4 : Opérations sur les matrices (suite)

1. Dans un script, écrire le code permettant de créer les matrices suivantes :

$$A = \begin{pmatrix} 9 & -2 & 2 \\ 0 & 7 & 3 \\ 7 & 3 & -4 \end{pmatrix} \quad \text{et} \quad b = \begin{pmatrix} -3 & 4 & -2 \end{pmatrix}$$

2. Déterminer et exécuter les commandes permettant d'obtenir les matrices suivantes :

(a) $\rightarrow \begin{pmatrix} -2 & 2 \\ 7 & 3 \\ 3 & -4 \end{pmatrix}$	(d) $\rightarrow \begin{pmatrix} -3 & 4 & -2 \\ 9 & -2 & 2 \\ 0 & 7 & 3 \\ 7 & 3 & -4 \end{pmatrix}$
(b) $\rightarrow \begin{pmatrix} 9 & -2 & 2 \\ 0 & 7 & 3 \end{pmatrix}$	(e) $\rightarrow \begin{pmatrix} 9 & 7 & -4 \end{pmatrix}$
(c) $\rightarrow \begin{pmatrix} -3 & 9 & -2 & 2 \\ 4 & 0 & 7 & 3 \\ -2 & 7 & 3 & -4 \end{pmatrix}$	(f) $\rightarrow \begin{pmatrix} -3 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & -2 \end{pmatrix}$

Exercice 5

Avec les commandes vues lors de ce TP :

1. A l'aides des commandes *ones()* et *zeros()* (et *insert*, *delete*, *concatenate*), construire une matrice de taille 10×10 ne contenant que des 2 sur les neuf premières lignes et ne contenant que des zéros sur la dernière ligne.
2. Construire une matrice de taille 10×10 ne contenant que des 3 sur la diagonale et des 5 sur partout ailleurs.
3. Construire la matrice carrée A de taille 10 contenant des 1 sur la première ligne, la première colonne, la première colonne et la dernière colonne et avec des zéros partout ailleurs.

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

4. Construire la matrice carrée C de taille 10 contenant des 1 sur la première ligne et la dernière colonne, telle que la diagonale de A soit égale au vecteur [1 2 3 4 5 6 7 8 9 10] et avec des zéros partout ailleurs.

$$B = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 \end{pmatrix}$$

5. Construire une matrice de taille 10×10 contenant les nombres 1,2,3,4,5,6,7,8,9,10 sur la diagonale et des -2 sur partout ailleurs.