(Re)-découverte de Python

1 Introduction et démarrage

« Un algorithme est une suite finie de règles à appliquer dans un ordre déterminé à un nombre fini de données pour arriver, en un nombre fini d'étapes, à un certain résultat, et cela indépendamment des données. »

Par exemple une recette de cuisine ou une notice de montage d'un meuble en kit peuvent être considérés comme des algorithmes.
Un algorithme doit être :

- clair (compréhensible par tous)
- le plus général possible (pour répondre au plus de cas possibles)
- rapide (efficace).

Un **programme** est la traduction d'un algorithme en langage compréhensible par l'ordinateur.
Le langage au programme en prépa ECG est le langage **Python**.

Plusieurs possibilités pour installer Python. Celle que je vous recommande est la suivante :

▷ Installer la distribution *Anaconda* sur votre ordinateur. Elle est téléchargeable à l'adresse :
https ://www.anaconda.com/products/individual

▷ Python sera directement installé sur votre ordinateur et vous aurez directement accès à un IDE (environnement de travail) pour coder en Python. L'IDE que nous utiliserons en classe (disponible également sur les ordinateurs du lycée) est **Spyder**.

2 Découverte de la console et commandes de base

Lors du lancement de Spyder, une fenêtre s'ouvre.

▷ En bas à droite de cette fenêtre se trouve la console. Elle permet de faire des calculs ou de définir de petites fonctions... Dans la console, on voit `In [1]` ce qui signifie que Python attend une instruction.

▷ La fenêtre comprend à gauche un éditeur de texte dans lequel on peut écrire des scripts (ou programmes) un peu plus longs. On peut ainsi les enregistrer pour les utiliser lors d'une nouvelle session et/ou les exécuter.
Attention : Lors de l'exécution, les messages et affichages éventuels apparaissent dans la console.

2.1 Calculs sur les nombres réels

▷ Evaluer dans la console de manière successive, les instructions suivantes :

```
2+4      5-3      4*7      8/2      5**2
```

En déduire l'interprétation de ces symboles.

▷ Dans la console, vérifier que rien n'est inscrit devant l'invite de commande `In [..]`. Appuyer alors plusieurs fois sur la touche ↑ puis sur la touche ↓. A quoi servent ces touches ?

La console peut s'utiliser comme une calculatrice avec les **opérations usuelles** : addition (+), soustraction (-), multiplication ($\star$), division (/) et puissance ($\star\star$).

Exercice 2.1 Calculer avec Python :

$$A = 10 - 4\frac{\left(\frac{5}{9} - \frac{1}{3}\right)\left(3 - \frac{1}{2}\right)}{\frac{7}{9} - 3}; \qquad B = 2 \times \frac{3-5}{3 \times 4} + 2^3 \times \frac{\frac{7}{9}}{\frac{6}{4}}.$$

Attention

Les fonctions usuelles ainsi que les constantes usuelles π, e... ne sont pas directement accessibles. Il faut *importer une bibliothèque* (appelée parfois librairie) pour les utiliser.

2.2 Fonctions prédéfinies

Une des bibliothèques au programme est la bibliothèque `numpy`. Elle permet d'effectuer des calculs numériques avec Python et de manipuler des vecteurs (=des matrices lignes), ou plus généralement des matrices de toute taille.
Aujourd'hui, nous nous intéresserons à la partie « calculs numériques».
Comme pour toute bibliothèque, il faut commencer par l'importer pour pouvoir l'utiliser. On l'importe via la commande :

```
import numpy as np
```

On a ainsi accès à toutes les constantes et fonctions définies dans `numpy`. Pour les appeler, il faudra utiliser le préfixe `np.`
Par exemple, on peut utiliser π et e via les commandes `np.pi` et `np.e`
On a également accès aux fonctions usuelles, à savoir :

- `np.exp` pour l'exponentielle
- `np.log` pour le logarithme népérien
- `np.sin` pour le sinus
- `np.cos` pour le cosinus
- `np.tan` pour la tangente
- `np.sqrt` pour la racine carrée
- `np.abs` pour la valeur absolue
- `np.floor` pour la partie entière, etc

Comme ce sont des fonctions Python :

- Ne pas oublier les () au moment de les appliquer, par exemple : `np.exp(3)`
- le paramètre d'entrée n'a pas de type prédéfini. Il peut être un entier ou un réel mais également un vecteur ou une matrice. Dans ce cas, la fonction s'applique à tous les coefficients de la matrice.

▷ Traduire en Python l'expression mathématique $\mathrm{e}^{\left|\frac{-\sqrt{7}}{2}\right|}$.

Exercice 2.2 Calculer la valeur absolue et la partie entière des nombres suivants :

$$A = \frac{\sqrt{3}}{8} - \frac{2^6}{3} - \frac{5}{2}; \qquad B = \frac{131}{\sqrt{2}} + 10^3 - \frac{5}{49}; \qquad C = \frac{-5 - 2\sqrt{6}}{2}.$$

3 Les variables

Lorsqu'on programme en Python, il sera souvent utile de garder en mémoire certains résultats. Pour cela on utilise des **variables**.
Une **variable** est une case mémoire de l'ordinateur à laquelle on a donné un nom.
Pour **affecter** une valeur à une variable, on utilise le symbole d'affectation =. Lorsqu'une variable est créée, elle apparaît dans la partie en haut à droite de la fenêtre appelée **Variable explorer**. Il s'agira de rester vigilant car ces affectations sont globales et seront utilisées par Python dans toute la feuille de travail.

▷ Evaluer dans la console `a`. Noter le message d'erreur obtenu.

▷ Evaluer maintenant `a=5` puis de nouveau `a`. Comparer et expliquer ce résultat avec le précédent.

▷ Evaluer `5*a+1`. Noter précisément ce qui est affiché en retour.

▷ Evaluer `a=5*a+1`. Expliquer le fonctionnement de cette affectation et comparer l'affichage obtenu avec le précédent.

▷ Prédire, sans l'évaluer, le contenu de la variable `a` à la fin de la séquence suivante :
`a=1, a=np.sqrt(36*a+13), a=a/2, a=np.log(a+4), a=8`
Vérifier alors votre résultat en évaluant cette séquence. Expliquer le résultat obtenu.

▷ Evaluer la séquence `a=3, b=7, a==b, a=b, a==b`. Noter les résultats obtenus.

▷ A quoi correspondent les opérateurs = et == en Python ? On expliquera les résultats précédents.

▷ Prédire le résultat de la commande : `a==5*a+1`.

Exercice 3.1 On considère trois variables a, b et c et on effectue la suite des affectations ci-dessous :

```
In[1]   a = 1,   b = 2,   c = 3
In[2]   a = 2 * b + 5
In[3]   c = b,   b = a
In[4]   a = c - b + 2
```

Compléter le tableau de valeurs des variables a, b, c.

Variables	a	b	c
Affectation 1			
Affectation 2			
Affectation 3			
Affectation 4			

Exercice 3.2 Proposer une suite d'instructions qui permet d'échanger les valeurs de deux variables a et b.

4 Editeur de texte

4.1 Première utilisation

Taper directement dans la console a deux inconvénients : l'enregistrement n'est pas possible, et si plusieurs lignes d'instructions ont été tapées, les modifications ne sont pas aisées. Pour enchaîner plusieurs instructions, l'éditeur est l'outil approprié.

On tape dans l'éditeur comme dans n'importe quel traitement de texte. L'éditeur reconnaît le langage Python : il met en couleur les mots-clef, un décalage de début de ligne appelé *indentation* se fait automatiquement lorsqu'on commence une boucle ou un test, il ferme automatiquement les parenthèses ouvertes...

- ▷ Placez-vous dans la partie gauche de la fenêtre.
- ▷ Dans cette partie, taper une première ligne de commande `a=5`. Sur la ligne en dessous, taper la commande `b=8`.
- ▷ Dans le menu Fichier, choisir *Enregistrer sous* afin de sauvegarder ce document dans le dossier TP_1. On nommera ce fichier *monPremierProg*. Quel est le format de fichier proposé ? .....
- ▷ Pour exécuter votre programme, cliquer sur l'icône « Play » (triange vert). Cette icône permet à la fois d'enregistrer votre fichier et de l'exécuter.

> **A noter**
> Dès que vous modifier un fichier, pensez à la sauvegarder. Dans le cas contraire, vous serez à la merci d'un bug de l'ordinateur, d'une mauvaise manipulation ou même d'une simple panne de courant. Sauvegarder doit donc devenir un réflexe !

- ▷ Dans la console, évaluer `a` puis `b`. Quel est le contenu de ces variables ?

- ▷ Pour exécuter votre fichier, vous pouvez aussi : cliquer sur la touche F5 (ou sous Mac OS, cliquer sur fn-F5).

4.1.1 Affichage

Dans la console, un calcul est automatiquement affiché, et pour afficher une variable nommée a (par exemple), il suffit de taper dans la console :

```
In [..]    a
```

Mais dans l'éditeur, il faut utiliser la commande `print(...)`.
Par exemple, il faudra taper `print(3*4)` pour afficher le résultat de 3×4 ou `print(a)` pour afficher le contenu de la variable `a`.

4.1.2 Commentaires

Pour gagner en clarté, il est recommandé de commenter son script. A fortiori, si le script est utilisé dans une autre séance.
On peut par exemple expliquer ce que fait un programme ou bien ce que contient une variable...
Pour insérer des commentaires, il suffit de taper ♯ et ce qui suit ne sera pas lu par l'ordinateur (donc ni exécuté, ni affiché).