TP1 - Introduction

1. Généralités

1.1. **Mise en route.** Nous utiliserons le logiciel **Pyzo**, pour écrire des programmes en langage **Python**.
La fenêtre de Pyzo comporte deux parties principales :
— L'**éditeur**, qui est une sorte de bloc-notes pour écrire nos programme.
— Le **shell** : là quand on écrit une ligne, Python «répond».

Exercice 1. Déterminer par des essais quelle fenêtre est le shell.

On peut faire deux types d'actions dans le shell :
— **Évaluer** une **expression**, par exemple l'expression `5+3*6-1` : Python affiche le résultat.
— **Exécuter** une **instruction**, par exemple l'instruction `a=5+3` : Python obéit en silence.

L'**affectation** (**assignment** en anglais) est une instruction qui associe un nom et une valeur, par exemple lorsqu'on exécute l'affectation `a=5+3`, Python commence par calculer que 3+5 vaut 8, puis mémorise que le nom `a` désigne cette valeur. On peut alors l'utiliser dans des expressions comme `a*a-20`. Un nom ainsi relié à une valeur s'appelle une **variable**.
À retenir : l'affectation n'est pas symétrique ! `a=5+3` est correct mais `5+3=a` est incorrect.

Exercice 2. Si on écrivait une après l'autre dans le shell les lignes suivantes, que répondrait Python à chaque ligne ? Remplir la colonne «prédiction» sans toucher à l'ordinateur, puis faire la vérification.

Ligne à saisir	Prédiction	Vérification
`a=3`		
`3+4*2`		
`(5-4)/3`		
`b=4+a`		
`abs(-2)`		
`int(3/2)`		
`b`		
`a=b*b`		
`a=a+1`		
`a+=2`		
`b*=2`		
`a+b`		
`a**2`		
`a**b`		
`a//2`		
`a%2`		

Remarque : Si on a exécuté une mauvaise instruction, pour recommencer en oubliant toutes les variables qu'on a définies, on doit utiliser le menu «shell» de Pyzo et cliquer sur «Redémarrer»

Remarque : Si l'on veut réexécuter une instruction déjà exécutée, il est possible de la faire réapparaître sur la ligne courante du shell en utilisant les flèches du clavier.

Exercice 3. Comment peut-on consulter la valeur d'une variable, par exemple la variable `a`, en utilisant le shell ? .....

2. Listes

Une **liste** en Python est représentée entre crochets : `[ 3 , 0 , 8 , 12 ]`. Voici quelques opérations que l'on peut faire avec des listes :

- `a=[ 3 , 0 , 8 , 12 ]` : affectation. (Les espaces ne servent qu'à la lisibilité)
- `b=[ [9,8] , 2 , 23 ]` : Le premier élément de cette liste est une liste (`[9,8]`).
- `a+b` : **concaténation**, opération qui crée une nouvelle liste en mettant deux listes bout-à-bout.
- `b.append(5)` : ajout d'un élément à la fin de la liste `b`
- `a.pop()` : retrait du dernier élément
- `len(a)` : longueur d'une liste : le nombre d'éléments qu'elle possède.
- `a[0]` : cette expression désigne le premier élément de la liste `a`. Attention : la numérotation par défaut en python commence à 0 et non à 1.
- `a[i]` : cette expression désigne l'élément en position `i` dans la liste `a`.
- `a[2]=8` : remplace l'élément en position `2` dans la liste `a` par la valeur `8`

Exercice 4. Pour chacune des lignes suivantes qu'on écrira dans le shell l'une après l'autre :
- si c'est une expression, prédire la valeur qui sera affichée par Python en réponse,
- si c'est une instruction, prédire la nouvelle valeur de la variable impactée.

```
total=3 .....
total=10*total+7 .....
total=10*total+2 .....
G=[ 2, 23, [4,5] ] .....
L=G+[7,2,5] .....
L[0] .....
L[7-5] .....
L[ L[0] ] .....
L[2][0] .....
L.append( 6 ) .....
len(L) .....
L[len(L)-2] .....
L[len(L)-1] .....
H=[] .....
H.append( G[0]+G[1]) .....
H.append( G[2]+L[2]) .....
L[1+1]=4 .....
H[0]=L[1] .....
```

3. Calculs répétitifs

3.1. **Boucle for. Remarque :** Pour exécuter plusieurs instructions d'un seul coup sans attendre les réponses de Python, on peut les écrire dans l'éditeur, les sélectionner avec le curseur, et utiliser la fonction «Exécuter la sélection» dans le menu «Exécuter» de pyzo.

Regardons le programme suivant, que l'on écrira dans l'éditeur :

```
s=0
for coco in range(3,8):
```

```
  s+=coco
```

Les deux dernières lignes sont une **boucle** : elles sont équivalentes au programme répétitif ci-dessous, où la variable `coco` prendra successivement comme valeur tous les entiers naturels de l'intervalle $[3, 8[$ (8 exclu), et où l'instruction `s+=coco` (équivalente à `s=s+coco`) est répétée à chaque fois :

```
s=0
coco=3
s=s+coco
coco=4
s=s+coco
coco=5
s=s+coco
coco=6
s=s+coco
coco=7
s=s+coco
```

(1) Avant d'exécuter le programme, prédire quelle est la valeur de `s` à la fin : .....

(2) Si on remplace `range(3,8)` par `range(0,8)` dans le programme, quelle sera la valeur finale de `s` ? .....

Au moment d'écrire une boucle, ne pas oublier :

- la ligne qui introduit la boucle (avec le mot-clé `for`) crée une variable (ici nommée `coco`, mais son nom est libre), qui va prendre successivement plusieurs valeurs. C'est la **variable de boucle**.
- si on indique `in range(a,b)`, avec $a < b$, la variable de boucle ira de a à $b - 1$.
- l'instruction `range(n)` signifie `range(0,n)`,
- La première ligne se termine par **deux points**, qu'il ne faut pas oublier ! !
- la ou les lignes suivantes, que Python répétera pour chaque valeur de la variable de boucle, doivent être «**indentées**» : c'est-à-dire qu'elles commencent par des espaces, ce qui les **décale vers la droite**. C'est l'**indentation** qui permet à Python de savoir quelles lignes sont sujettes de la boucle `for`.

Exercice 5. En utilisant un programme inspiré du précédent, calculer la factorielle de 12, c'est à dire le produit $1 \times 2 \times 3 \ldots \times 12$.

Exercice 6. Expliquer ce que fait le programme ci-dessous. Commencer par écrire sur papier le programme répétitif équivalent.

```
L=[]
for k in range(0,10):
  L.append( 2*k+1 )
```

.....

Exercice 7. En utilisant un programme inspiré du précédent, construire la liste `LC` des 100 premiers carrés parfaits (les carrés des 100 premiers entiers) : 0, 1, 4, 9, 16, ... 9801 (le carré de n s'écrit en python `n**2`)

Exercice 8. Calculer la somme des 100 premiers carrés parfaits : $S = \sum_{k=0}^{99} k^2$ .....

Exercice 9. Écrire un programme qui calcule la somme des éléments d'une liste `L`, quels que soient son contenu et sa longueur. Le programme doit fonctionner pour n'importe quelle liste. On le testera avec la liste `LC`.

Exercice 10. Écrire un programme qui, étant donné un entier n, construit la liste des entiers `J=[n-1,n-2,...,2,1,0]` dans l'ordre inverse. Ainsi si `n=5` ce programme construit la liste `[4,3,2,1,0]`. Tester avec $n = 100$.

Exercice 11. Écrire un programme qui, étant donné une liste `L`, construit la liste des éléments de `L` dans l'ordre inverse. Ainsi si `L=[3,1,9,2]` ce programme construit la liste `I=[2,9,1,3]`. Tester avec `LC`.

Exercice 12. Écrire un programme qui, étant donné une liste `L`, construit la liste des variations entre un élement de `L` et le suivant. Ainsi si `L=[3,1,9,2]` ce programme construit la liste `V=[-2,8,-7]` (Remarquer que la liste à construire doit avoir un élément de moins que la liste de départ). Tester avec `LC`.

3.2. **Parcours d'une liste.** Dans une boucle `for`, l'indice peut aussi parcourir une liste.

Exercice 13. Devinez la valeur de S à la fin du programme suivant, puis testez :

```
L=[-1,3,8,4]
S=0
for k in L:
  S+=k
```

Reprendre l'exercice 9 en utilisant cette remarque.

Installation de Python et Pyzo

Pour pouvoir pratiquer Python sur votre ordinateur, suivez les etapes suivantes :

(1) Installation de Python (le «moteur» qui exécute les programmes)
Ceci est la procédure pour installer la distribution standard de Python (certains pourraient préférer la distribution Anaconda ou Miniconda)
   (a) Se rendre sur le site www.python.org
   (b) Cliquer sur l'onglet «Download»
   (c) Télécharger la dernière version : 3.10
   (d) Suivre l'assistant d'installation
   (e) Lors de l'installation, il vous faut choisir un emplacement pour mettre le programme Python : notez bien cet emplacement, il pourrait vous servir plus loin.

(2) Installation de Pyzo
   (a) Se rendre sur www.pyzo.org/start.html
   (b) Effectuer l'étape 1 de la procédure : installation de l'IDE Pyzo

(3) Lancement et paramétrage
   (a) Lancer le logiciel Pyzo
   (b) Si le shell fonctionne, c'est terminé : vous pouvez commencer à programmer.
   (c) Sinon, à la place du shell, vous trouvez un texte qui explique que Pyzo n'a pas détecté votre installation de Python. Il vous propose d'aller à la **configuration du shell** : cliquer sur ce lien
   (d) Dans la fenêtre qui s'ouvre, écrire dans le champs «exe» l'emplacement choisi lors de l'installation du programme Python, à l'étape 1e

(4) Installation du package matplotlib, qui nous servira bientôt à tracer des graphiques :
   (a) Assurez-vous d'avoir une bonne connexion internet : l'installation va télécharger beaucoup de données
   (b) Dans le shell, taper la commande `pip install matplotlib`.
   (c) Ensuite, normalement il suffit d'attendre que l'installation se termine.