

## TP

## Prise en main

## Savoirs et compétences :

- *Th. 0 : Connaître les bases de l'algorithmique et de la programmation (variable, types, structures, fonctions).*

## Consignes

- Commencez la séance en créant un dossier au nom du TP dans le répertoire dédié à l'informatique de votre compte.
- Après la séance, vous devez rédiger un compte-rendu de TP et l'envoyer au format électronique à votre enseignant.
- Essayer d'être le plus autonome possible.
- Ce TP est à faire en binôme, vous ne rendrez donc qu'un compte-rendu pour deux. Votre fichier portera un nom du type : `tp01_durif_pessoles.py`, où les noms de vos enseignants sont à remplacer par ceux des membres du binôme. Le nom de ce fichier ne devra comporter ni espace, ni accent, ni apostrophe, ni majuscule. Dans ce fichier, vous respecterez les consignes suivantes.
  - Écrivez d'abord en commentaires (ligne débutant par #), le titre du TP, les noms et prénoms des étudiants du groupe.
  - Commencez chaque question par son numéro écrit en commentaires.
  - Les questions demandant une réponse écrite seront rédigées en commentaires.
  - Les questions demandant une réponse sous forme de fonction ou de script respecteront pointilleusement les noms de variables et de fonctions demandés.

### Activité 1 – Environnement de développement intégré Python et prise en main élémentaire de Python

Lancer PYZO, Spyder ou IDLE. Un *interpréteur de commandes*, ou `shell`, s'affiche. Le symbole `>>>` signifie que Python attend vos instructions.

Sitôt une instruction tapée et validée (par la touche « Entrée »), le `shell` effectue le calcul demandé puis affiche un résultat, ou un message d'erreur. Il est extrêmement important de bien lire ces messages d'erreur, et de les comprendre !

**Question 1** Taper dans le shell les instructions suivantes. Que se passe-t-il ? Qu'est-ce que cela signifie ?

```
x = 42
y = 42.
type(x)
type(y)
x = x+y
x
type(x)
```

**Question 2** Décrire ce que font les opérations suivantes, après les avoir étudiées sur des exemples numériques.

+   -   \*   \*\*   /   //   %

**Question 3** Taper dans le shell les instructions suivantes. Que se passe-t-il ? Qu'est-ce que cela signifie ?

```
B = 42 > 41.
type(B)
```

**Question 4** Décrire ce que font les opérations suivantes, après les avoir étudiées sur des exemples numériques

==   !=   <   >   <=   >=

**Question 5** Taper dans le shell les instructions suivantes. Que se passe-t-il? Qu'est-ce que cela signifie?

3/0 > 5

**Question 6** Taper dans le shell les instructions suivantes. Que se passe-t-il? Qu'est-ce que cela signifie?

```
B = (42 > 41) or ( 3/0 > 5) .
type(B)
```

**Question 7** Décrire ce que font les opérations suivantes, après les avoir étudiées sur des exemples logiques.

or and not

**Question 8** Taper dans le shell les instructions suivantes. Que se passe-t-il? Qu'est-ce que cela signifie?

```
x = -3
abs(x)
help(abs)
```

**Question 9** Taper dans le shell les instructions suivantes. Que se passe-t-il? Qu'est-ce que cela signifie?

```
import math as m
import numpy as np
m.sin(m.pi)
np.sin(np.pi)
np.sin([0,np.pi])
m.sin([0,m.pi])
```

**Question 10** Taper dans le shell les instructions suivantes. Que se passe-t-il? Qu'est-ce que cela signifie?

```
L = [0,1,2,3,4,5,6]
type(L)
L[0]
L[6]
L[-1]
L[-2]
L[7]
L[1:4]
L[2:8]
L.append(7)
L
L = L.append(8)
L
```

Nous avons vu comment utiliser des fonctions et des bibliothèques. Nous pouvons bien entendu créer nos propres fonctions (et bibliothèques).

Dans PYZO ou IDLE, ouvrir un nouveau fichier (CTRL+N ou File / New file). L'enregistrer (CTRL + S ou File / Save) sous le nom TP02.py.

**Question 11** Taper dans cette fenêtre le script suivant.

```
def somme(n) :
    """Renvoie 0 + 1 + 2 + ... + n
    Precondition : n entier naturel"""
    return n*(n+1) // 2
```

Enregistrer puis appuyer sur la touche Ctrl+MAJ+E (sous PYZO) ou F5 (sous IDLE). Le *shell* doit s'afficher. Taper dans le *shell* les instructions suivantes.

```
somme(42)
somme(42.)
somme(-1515)
help(somme)
```

Que se passe-t-il? Qu'est-ce que cela signifie?

**Question 12** Comment peut-on utiliser la fonction écrite précédemment dans un autre script Python?

## Activité 2 – Types composés

**Question 13** Prévoir les résultats des expressions suivantes, puis le vérifier grâce à l'interpréteur interactif d'IDLE.

- |                                    |                                   |
|------------------------------------|-----------------------------------|
| a) <code>[1, 2, 3, "a"]</code>     | g) <code>[0, 0] + [0]</code>      |
| b) <code>123a</code>               | h) <code>len(["a", "b"])</code>   |
| c) <code>[]</code>                 | i) <code>len([])</code>           |
| d) <code>[] + []</code>            | j) <code>len([[]])</code>         |
| e) <code>[] + [] == []</code>      | k) <code>len([[[[]]])</code>      |
| f) <code>[1, 2] + [5, 7, 9]</code> | l) <code>len([0, 0] + [1])</code> |

**Question 14** Calculer cette suite d'expressions.

```
[i for i in range(10)]
[compt**2 for compt in range(7)]
[j+1 for j in range(-2, 8)]
```

Sur ce modèle, obtenir de manière synthétique :

- la liste des 20 premiers entiers naturels impairs;
- la liste de tous les multiples de 5 entre 100 et 200 (inclus);
- La liste de tous les cubes d'entiers naturels, inférieurs ou égaux à 1000.
- une liste contenant tous les termes entre -20 et 5 d'une progression arithmétique de raison 0,3 partant de -20.

## Activité 3 – Variables

**Question 15** Voici des affectations successives des variables *a* et *b*. Dresser un tableau donnant les valeurs de *a* et *b* à chaque étape.

```
>>> a = 1
>>> b = 5
>>> a = b-3
>>> b = 2*a
>>> a = a
>>> a = b
```

**Question 16** Écrire une séquence d'instructions qui échange les valeurs de deux variables *x* et *y*.

**Question 17** Écrire, sans variable supplémentaire, une suite d'affectation qui permute circulairement vers la gauche les valeurs des variables *x*, *y*, *z* : *x* prend la valeur de *y* qui prend celle de *z* qui prend celle de *x*.

**Question 18** Calculer, sans utiliser la fonction `sqrt` ni la division flottante `/`, les nombres suivants.

- |                    |                 |                           |                |
|--------------------|-----------------|---------------------------|----------------|
| a) $\frac{1}{7,9}$ | b) $\sqrt{6,2}$ | c) $\frac{1}{\sqrt{3,5}}$ | d) $2\sqrt{2}$ |
|--------------------|-----------------|---------------------------|----------------|

De base, on ne peut réaliser que des calculs élémentaires avec Python. Cependant, il est possible d'avoir accès à des possibilités de calcul plus avancées en utilisant une *bibliothèque*. Par exemple, la bibliothèque `math` permet d'avoir accès à de nombreux outils mathématiques. On peut donc taper

```
from math import sqrt, log, exp, sin, cos, tan, pi, e
```

pour avoir accès à toutes ces fonctions.

Calculer les nombres suivants (on n'hésitera pas à consulter l'aide en ligne).

a)  $e^2$   
b)  $\sqrt{13}$   
c)  $\cos\left(\frac{\pi}{5}\right)$

d)  $e^{\sqrt{5}}$   
e)  $\ln 2$   
f)  $\ln 10$

g)  $\log_2 10$   
h)  $\tan\left(\frac{\pi}{2}\right)$

#### Activité 4 – Structures de boucles

**Question 19** Un taupin se lance dans un marathon d'exercices, mais se fatigue vite. Il réalise le  $i^{\text{e}}$  exercice en  $\sqrt{i}$  minutes. Combien d'exercices arrive-t-il à faire en 4 heures ? Pour faciliter la correction, on écrira une fonction `nb_exos()` ne prenant pas d'argument et renvoyant le résultat demandé.

#### Activité 5 – Structures de boucles

**Question 20** Calculer  $2^9$  à l'aide d'une boucle itérative.

**Question 21** Écrire un algorithme affichant la table de multiplication de 9.

**Question 22** Calculer  $16!$  à l'aide d'une boucle itérative.

**Question 23** Calculer  $\sum_{k=0}^{15} \frac{1}{k!}$ .

**Question 24** Écrire une fonction calculant le nombre de chiffres d'un entier écrit en base 10.

**Question 25** On considère la suite  $u$  définie par  $\forall n \in \mathbb{N}^* \quad u_n = \sum_{k=1}^n \frac{1}{\sqrt{k}}$  Quel est la plus petite valeur  $n$  pour laquelle  $u_n \geq 1000$  ?

**Question 26** Écrire une fonction trouvant le plus petit nombre premier supérieur ou égal à un entier donné.

**Question 27** Écrire une fonction calculant le nombre de diviseurs d'un entier  $n$  donné.

**Question 28** Calculer  $p_5/q_5$  où  $p$  et  $q$  sont définies par :

$$p_0 = 1$$

$$q_0 = 1$$

$$\forall n \in \mathbb{N} \quad p_{n+1} = p_n^2 + 2q_n^2$$

$$\forall n \in \mathbb{N} \quad q_{n+1} = 2p_n q_n$$

#### Activité 6 – Simulation d'un prêt immobilier

Un banquier vous propose un prêt de 400 000 euros sur 40 ans «à 3% par an» — ce qui, dans le langage commercial des banquiers, veut dire 0,25% par mois — avec des mensualités de 1431,93 euros. Autrement dit, vous contractez une dette de 400 000 euros. Chaque mois, cette dette augmente de 0,25% puis est diminuée du montant de votre mensualité. À la fin des  $40 \times 12$  mensualités, il ne vous reste plus qu'à vous acquitter d'une toute petite dette, que vous rembourserez aussitôt.

**Question 29** Écrire une fonction `reste_a_payer(p, t, m, d)` renvoyant le montant de cette somme à rembourser immédiatement après le paiement de la dernière mensualité, où  $p$  est le montant total du prêt en euros (dans l'exemple, 400 000),  $t$  son taux mensuel (dans l'exemple,  $0,25 \times 10^{-2}$ ),  $m$  le montant d'une mensualité en euros (dans l'exemple, 1431,93) et  $d$  la durée en années (dans l'exemple, 40).

*Indice : dans le cas donné dans cet énoncé, vous devez trouver un montant restant d'un peu moins de 7,12 euros.*

**Question 30** Écrire une fonction `somme_totale_payee(p, t, m, d)` renvoyant la somme totale (mensualités plus le dernier paiement) que vous aurez payé au banquier.

**Question 31** Écrire une fonction `cout_total(p, t, m, d)` renvoyant le coût total du crédit, c'est-à-dire le total de ce que vous avez payé moins le montant du prêt.

Un banquier vous propose de vous prêter  $p$  euros, à un taux de  $12t\%$  par an — ce qui, dans le langage commercial des banquiers, veut dire  $t\%$  par mois — avec des mensualités de  $m$  euros. Autrement dit, vous contractez une dette de  $p$  euros. Chaque mois, cette dette augmente de  $t\%$  puis est diminuée du montant de votre mensualité. Lorsque votre dette, augmentée du taux, est inférieure à la mensualité, il suffit de régler le solde en une fois.

**Question 32** Écrire une fonction `duree_mensualite(p, t, m)` renvoyant le nombre de mensualités nécessaires au remboursement total du prêt.

**Question 33** Attention : que se passe-t-il si la mensualité est trop petite ?

Indice : dans le cas où le prêt est  $p = 4 \times 10^5$ , le taux est  $t = 0,25 \times 10^{-2}$  et la mensualité est  $m = 1431,93$ , on trouvera une durée de remboursement de 480 mois.

**Question 34** Écrire une fonction `tracer_mensualite(p, t, m)` permettant de tracer en fonction du numéro de la mensualité la dette restante (ou le capital restant dû) jusqu'à ce que le prêt soit remboursé. Cette fonction permettra également de tracer en fonction du numéro de la mensualité le montant de l'intérêt versé à la banque.