TRAVAUX PRATIQUES n° 1
Introduction à Pyzo et révisions sur le langage Python 3

Pour bien démarrer ce TP, commencez par lire les fiches méthode suivantes :

- ⋆ **I1** - Définir et calculer avec des variables
- ⋆ **I2** - Afficher une phrase dynamique
- ⋆ **I3** - Écrire un script

I. Introduction

1) Shell vs éditeur

Dans Pyzo, on retrouve deux parties principales où l'on peut écrire nos instructions :

- ⋆ le shell ou la console :

  C'est ici que se trouve le logiciel Python qui est chargé d'exécuter les instructions. Avantages et inconvénients du shell :

  - les instructions sont exécutées une à une c'est-à-dire qu'on peut écrire une instruction, taper `ENTREE` pour l'exécuter puis écrire une autre instruction ;
  - les instructions tapées ici ne sont pas enregistrées, une fois Pyzo fermé, elles sont perdues définitivement ;
  - si on s'est trompé dans l'instruction que l'on vient de taper, on ne peut la modifier que si on n'a pas encore validé, sinon il faut retaper l'instruction (on pensera à utiliser la touche $\boxed{\uparrow}$ pour gagner du temps).

  Le shell est donc pratique pour tester des instructions rapidement et de façon isolée du reste du programme mais ce n'est pas là qu'il faut écrire nos programmes.

- ⋆ l'éditeur :

  C'est un simple éditeur de texte où l'on peut écrire du code Python qui pourra être ensuite exécuté. Avantages et inconvénients de l'éditeur :

  - on peut écrire plusieurs instructions avant de les exécuter toutes d'un coup (mais toujours l'une après l'autre) ;
  - on peut enregistrer dans un fichier .py tout ce que l'on écrit dans l'éditeur ;
  - on peut ajouter des commentaires pour expliquer le programme ;
  - si on s'est trompé dans une instruction, on peut la modifier à tout moment.

  L'éditeur est donc la partie où l'on doit écrire nos programmes.

2) Structurer son TP

On prendra l'habitude de séparer les différentes parties d'un TP (exercices ou groupes d'exercices) en cellules : une cellule commence par un double-dièse ## suivi d'un titre. Par exemple :

```
## Exercice 1

print("Hello world!")

## Exercice 2
```

3) Exécuter le code écrit dans l'éditeur

- ⋆ `CTRL + E` : exécuter tout le fichier
- ⋆ `CTRL + ENTREE` : exécuter la cellule où se trouve le curseur
- ⋆ `ALT + ENTREE` : exécuter la sélection ou exécuter la ligne où se trouve le curseur si rien n'est sélectionné

4) Quelques rappels de syntaxes

- ⋆ Pour utiliser des fonctions mathématiques, il faut importer le module `math`. On importera uniquement les fonctions et constantes nécessaires :

```
from math import sqrt, exp, log, cos, pi
```

Bonne pratique

Cette instruction d'importation doit être tapée **dans l'éditeur** et **en début de fichier**. On peut par exemple créer une première cellule

```
## Modules

from math import sqrt, exp, log, cos, pi

## Exercice 1

print("Hello world!")
```

- ⋆ Pour définir ou modifier une variable :

```
nom_de_variable = valeur
```

Par exemple :

```
mon_age = 17              # Création de la variable mon_age (valeur = 17)
mon_age = mon_age + 1     # Modification de cette variable  (valeur = 18)
```

- ⋆ Pour afficher une phrase à l'écran :
  - **phrase simple**

    ```
    print("Hello world!")
    ```

  - **phrase dynamique**

    ```
    prenom = "Victor"
    nom = "Hugo"
    print("Bonjour", prenom, nom)                    # Méthode 1
    ```

```
print("Bonjour {} {} !".format(prenom, nom))     # Méthode 2
print(f"Bonjour {prenom} {nom} !")               # Méthode 3
```

⋆ Pour demander une information à l'utilisateur :

```
phrase = input("Entrez une phrase : ")
nombre_entier = int(input("Entrez un nombre entier : "))
nombre_reel = float(input("Entrez un nombre réel : "))
```

Attention :

- il faut toujours enregistrer la réponse dans une variable
- il faut toujours convertir la réponse en `int` ou `float` lorsque celle-ci est numérique

II. Exercices

1) Calculs

Exercice 1 : Calculer les quantités suivantes à l'aide de Python :

1) $\dfrac{(12+3)\times 14}{6\times 18}$
2) $1\times 2\times 3\times 4\times 5\times 6\times 7\times 8\times 9\times 10$
3) 2^{32}
4) $3\times 4^5\times \dfrac{13}{12-45}+5$

Exercice 2 : Calculer les quantités suivantes à l'aide de Python :

1) $\sqrt{5}$
2) e^5
3) $\ln 2$
4) $\cos\dfrac{\pi}{3}$

Exercice 3 :

1) Évaluer les expressions suivantes : `23 // 3` et `23 % 3`
2) Vérifier que si `q = 23 // 3` et `r = 23 % 3`, alors `23 = 3 * q + r`.
3) À quoi correspondent les valeurs `q` et `r` par rapport aux nombres `23` et `3` ? À quoi servent les opérations `//` et `%` ?

Exercice 4 (*Test de divisibilité*) : Pour savoir si un nombre entier `a` est un multiple d'un autre nombre entier `b`, on peut calculer `a % b` : si on trouve `0`, `a` est un multiple de `b`, sinon ce n'est pas un multiple de `b`.

Le nombre 123456789 est-il un multiple de 7 ? Si non, déterminer le multiple de 7 le plus proche de ce nombre.

2) Variables

Exercice 5 : Examiner la série de commandes ci-dessous. Prédire le résultat puis le confirmer en les exécutant.

```
x = 1
y = 2
x = x + y
y = x**y
z = y % x
print(x, y, z)
```

Exercice 6 : Examiner la série de commandes ci-dessous. Prédire le résultat puis le confirmer en les exécutant.

```
x, y = 1, 2
x, y = y, x + y
z = x ** y
print(x, y, z)
```

3) Premiers scripts

Un script est un petit programme composé d'une suite d'instructions. Il doit impérativement être écrit dans l'éditeur (dans une cellule) et non dans le shell ! En général, il commence par demander certaines informations à l'utilisateur puis affiche le résultat d'un calcul sous forme d'une phrase.

Exercice 7 : Exécuter le script suivant :

```
prenom = input("Entrez votre prénom : ")
nom = input("Entrez votre nom de famille : ")
age = int(input("Entrez votre âge : "))
taille = float(input("Entrez votre taille (en mètre) : "))
print(f"Bonjour {prenom} {nom}. Vous avez {age} ans et vous mesurez {taille}m.")
```

Répondre aux questions posées dans le shell. Bravo : vous avez exécuté votre premier script !

Exercice 8 : Écrire un script permettant de calculer le carré d'un nombre réel. Le script devra demander à l'utilisateur le nombre réel puis afficher le résultat à l'aide d'une phrase.

Exemple :

```
Entrez un nombre réel : 12.25
Le carré de ce nombre est 150.0625
```

Exercice 9 (*Échange de variables*) : Écrire un script qui demande à l'utilisateur d'entrer deux nombres `a` et `b`, qui les affiche puis échange leurs valeurs et les affiche encore.

Exemple :

```
Entrez la valeur du nombre a : 12
Entrez la valeur du nombre b : 35
a vaut 12 et b vaut 35.
Maintenant, a vaut 35 et b vaut 12.
```

Exercice 10 : Écrire un script qui demande à l'utilisateur son année de naissance et qui affiche l'âge qu'il aura atteint au cours de l'année 2026.

Exercice 11 (*Durées*) : *On considérera qu'une année vaut toujours* 365 *jours.*

1) Écrire un script qui demande à l'utilisateur un nombre d'années, un nombre de jours, un nombre d'heures, un nombre de minutes et un nombre de secondes et qui calcule et affiche l'équivalent de cette durée en secondes.
2) Écrire un script qui demande à l'utilisateur un nombre de secondes et qui calcule et affiche l'équivalent de cette durée en années, jours, heures, minutes et secondes (avec un nombre de jours inférieur à 365, un nombre d'heures inférieur à 24, un nombre de minutes et un nombre de secondes inférieurs à 60).

   On utilisera les opérations `//` *et* `%`*.*
3) Essayer de deviner combien de temps environ correspondent une durée d'un million de secondes et d'un milliard de secondes. Vérifier votre estimation grâce à votre programme.