

TP noté 3 – Python (10 points)

Parties 1, 2 et 3

Objectifs du TP : Apprendre à utiliser le langage python

Le TP est à faire **seul.e** !

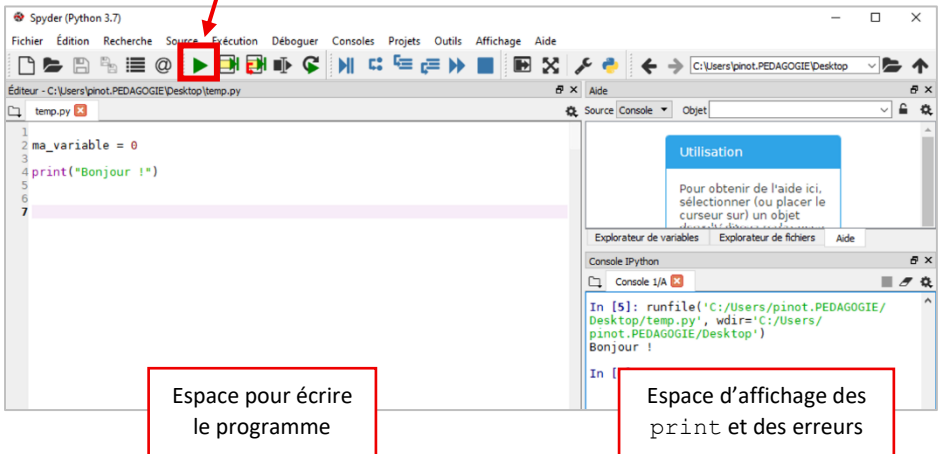
Consignes : Suivre les étapes du TP, faire les manipulations et répondre aux questions.

Introduction

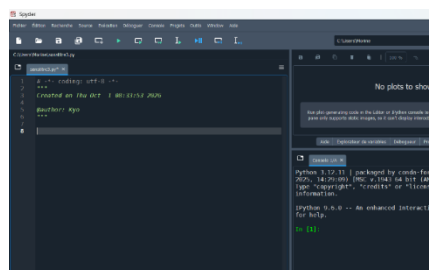
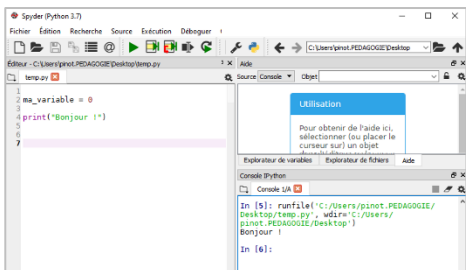
Pour programmer en Python, nous allons utiliser le logiciel **Spyder**.



Bouton pour exécuter
le programme



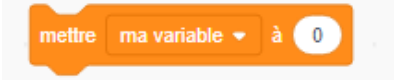
Le logiciel s'affiche par défaut en anglais et en noir.



Pour changer la langue et/ou la couleur : Appelez votre professeur.

Partie 1 : Les variables

On appelle **variable** les mots qui vont contenir une information.

	
	<pre>ma_variable = 0</pre>

- Dans l'exemple **Scratch** : `ma_variable` est le nom de la variable qui stocke l'entier 0.
- Dans l'exemple **Python** : `ma_variable` est le nom de la variable qui stocke l'entier 0.

/! En Python, il ne faut **pas** mettre d'espace dans le nom d'une variable (il faut mettre de tirets bas à la place)

Une variable contient une valeur qu'on appelle une donnée.

Exercice 1 :

- a. Ecrire le programme suivant dans Spyder (dans la partie gauche du logiciel) :

```
x = 3
y = 8
ma_variable = 0
salutation = "Bonjour"
```

- b. Enregistrer le programme dans vos Documents en nommant le fichier « premier_programme ».
- c. Lister les noms des variables (*attention, on veut les noms des variables, par leurs valeurs !*)
- d. Nous avons vu qu'il existait 4 principaux types de données (`int`, `float`, `str` et `bool`).

Pour chaque variable, indiquer le type de la donnée qu'elle contient.

- e. Exécuter le programme (cliquer sur le triangle vert en haut du logiciel).

Normalement, le texte `runfile ("...")` s'affiche dans la Console (à droite du logiciel).

`runfile ("...")` indique quel programme a été exécuté (ici `premier_programme`).

Si une erreur s'affiche sous `runfile ("...")` et que vous ne parvenez pas à trouver votre erreur : appelez votre professeur.

- f. Ajouter à la suite du programme les variables suivantes :

- La variable `mois` contient un entier (`int`) représentant ton mois de naissance ;
- La variable `identite` contient une chaîne de caractères (`str`) avec les initiales de ton prénom et de ton nom de famille ;
- La variable `a` contient le nombre à virgule (`float`) représentant le nombre 5,124 .

- g. Exécuter le programme.

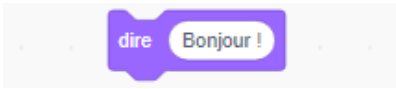
Si une erreur s'affiche et que vous ne parvenez pas à trouver votre erreur : appelez votre professeur.

- h. Noter les instructions (les lignes) que vous avez ajouté au programme à l'étape f. .

Partie 2 : Les affichages

L'**affichage** permet :

De montrer à l'utilisateur une phrase, un résultat, le contenu d'une variable, etc. avant, pendant ou après le programme.

	
Bloc :  Résultat : 	Programme : <pre>print("Bonjour !")</pre> Résultats : Bonjour !

/! En Python, si on veut afficher le contenu d'une variable, il ne faut **pas** mettre de guillemets.

Exercice 2 :

- a. Ajouter à la suite du programme de l'exercice 1 les instructions suivantes :

```
print("Bienvenue sur Spyder")
print(x)
print(y)
print(mois + x)
print(salutation)
print("identite")
```

- b. Exécuter le programme.

Sous `runfile("...")` vous devriez avoir l'affichage du contenu des `print`.

Si une erreur s'affiche, appelez votre professeur.

- c. Le dernier `print` affiche "identite".

Corriger le programme pour qu'il affiche le contenu de la variable `identite`.

Exécuter le programme pour voir le résultat de vos modifications.

- d. Que fallait-il faire à l'étape c. pour afficher le contenu de la variable `identite` ?

- e. Ajouter deux instructions à la suite du programme :

- Pour afficher le contenu de la variable `a`.
- Pour afficher le texte « Fin du programme ».

Exécuter le programme pour voir le résultat de vos modifications.

- f. Noter les instructions que vous avez ajouté au programme à l'étape e. .

Partie 3 : Les calculs

Exercice 3 :

- a. Dans Spyder, créer un nouveau fichier python.

Enregistrer-le dans vos Documents en nommant le fichier « exercice3 ».

- b. Ecrire le programme suivant dans ce nouveau fichier :

```
x = 10
y = 6
va = x * y
x = x + 1

print(x)
print(y)
print(va)
print(va / 2)
print(y - x)
```

- c. Exécuter le programme et observer l’affichage obtenu.
- d. Expliquer à quoi servent les symboles « + », « - », « * » et « / » dans le programme.
- e. Expliquer pourquoi `print(x)` affiche 11 alors que la première instruction du programme est `x = 10`.

Faire la suite du TP : Parties 4, 5 et 6