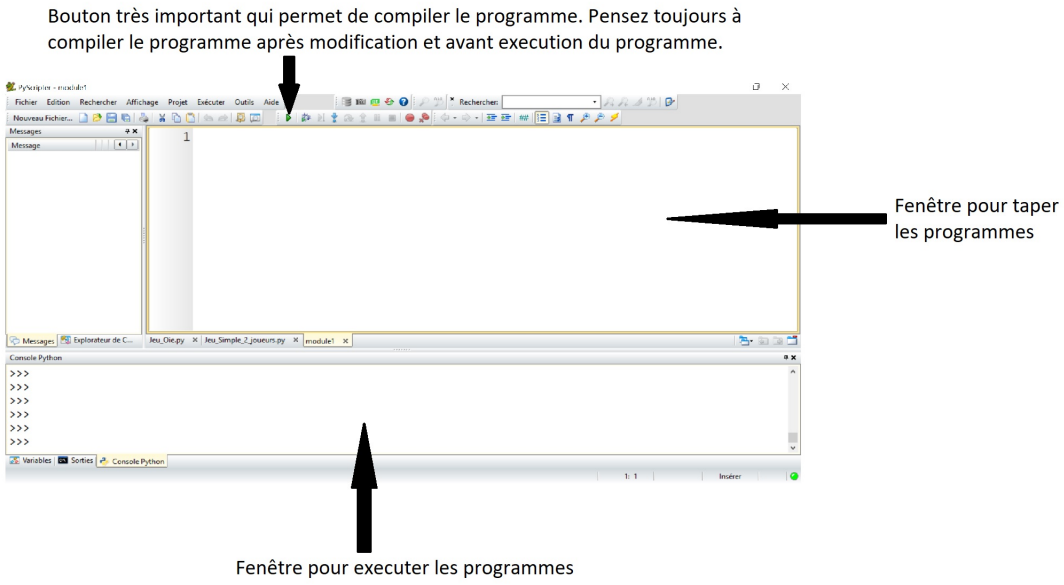
Initiation à Python : dynamique de populations

L'objectif est construire sur le logiciel Python *des programmes permettant de visualiser l'évolution de certaines populations.*

I Apprivoiser Python

Dans votre session, ouvrir le logiciel **EduPython** (sur le bureau, dans le dossier MATHS). La fenêtre ci-dessous s'ouvre alors :



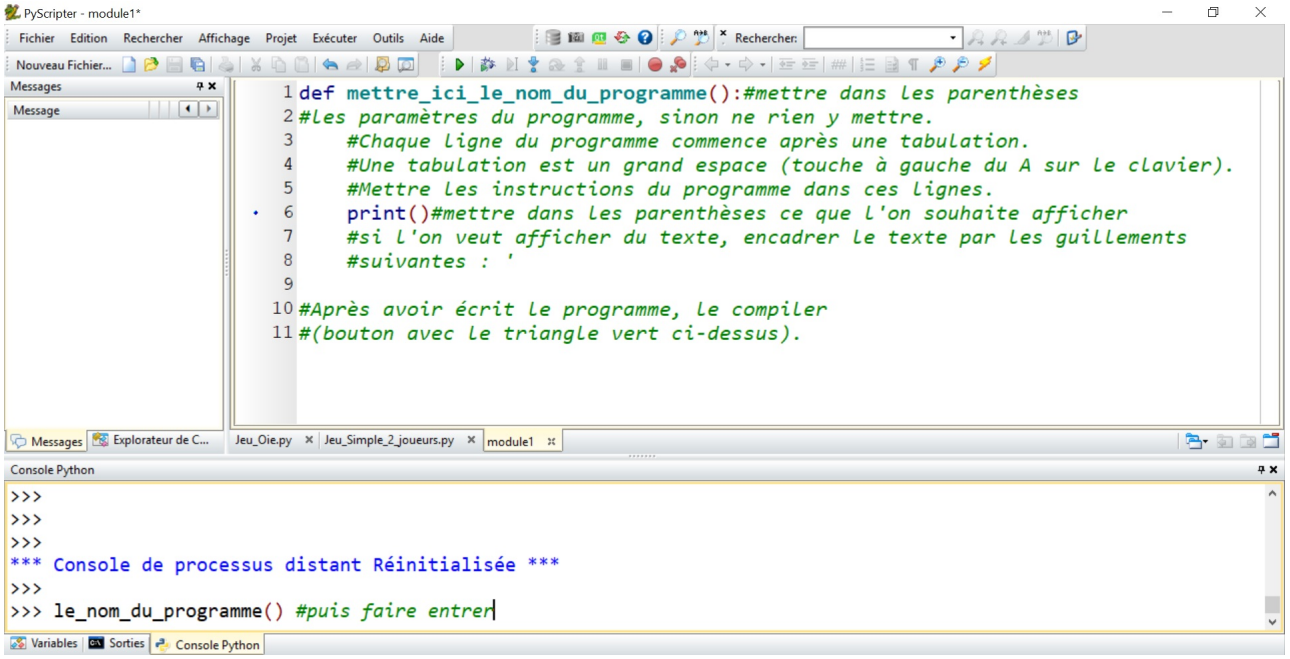
Exercice 1. L'interface des commandes d'exécutions.

La fenêtre du bas permet déjà de faire quelques opérations élémentaires.

1. *Du calcul :* taper devants les chevrons (le truc qui ressemble à $>>>$) une opération élémentaire comme $5+9$ puis faire entrer.
2. *Faire une affectation :* c'est-à-dire rentrer une valeur numérique (par exemple 5) dans une mémoire de l'ordinateur (par exemple n). Affecter 5 à la variable n.
3. **En une seule ligne**, ajouter 3 à la mémoire n et enregistrer cette valeur dans une nouvelle mémoire m.

Exercice 2. Le premier programme.

Dans la fenêtre du haut faire un premier programme affichant « Hello world ! » en vous aidant du squelette et des indications (après le signe #) suivants :



Exercice 3. Faire un nouveau programme qui demande un paramètre n, le multiplie par 5 et lui ajoute 3, puis retourne le nouveau nombre obtenu. La commande *return*, puis la variable considérée sans parenthèse autour, permet de retourner une valeur à la fin d'un programme.

II La population constante

Nous commençons notre initiation par l'étude d'une population la plus simple possible : rien ne se passe et la population reste constante au cours du temps. Deux notions seront abordées : la notion de liste puis de représentation graphique.

Exercice 4. Créer un nouveau programme qui

1. prend en paramètre la taille x de la population ET le nombre de générations n sur lequel on observe cette population,
2. créé une liste de n nombres x : à l'aide de la commande suivante :

```
u=[x]*n
```

3. retourne cette liste.

Exercice 5. Tester dans la fenêtre d'exécution la commande

```
m=list(range(n))
```

pour différentes valeurs de n. A quoi sert cette commande ?

On souhaite améliorer le programme de l'exercice 4. En première ligne de la fenêtre pour taper les programmes, rentrer la ligne de code suivante :

```
import matplotlib.pyplot as plt
```

A l'aide de cette bibliothèque, on va pouvoir utiliser la commande suivante :

```
plt.plot(x,y)
```

Cette commande relie des points dont les abscisses sont données par la liste x et les ordonnées par la liste y. Sauf que l'affichage n'est pas automatique. Pour que la fenêtre du graphique s'ouvre il faut taper dans la fenêtre d'exécution la commande suivante :

```
plt.show()
```

Exercice 6. Améliorer le programme de l'exercice 4 pour qu'il fasse tracer la population constante durant les n premières générations.

III La croissance linéaire

On suppose désormais que notre population augmente chaque année d'un même nombre p d'habitants. On suppose par exemple pour commencer que $p = 3$ et que la population initiale est de $u_0 = 0$ habitants.

Exercice 7.

1. Quel est le nombre d'habitants après un an ? deux ans ? trois ans ? cinq ans ?

.....

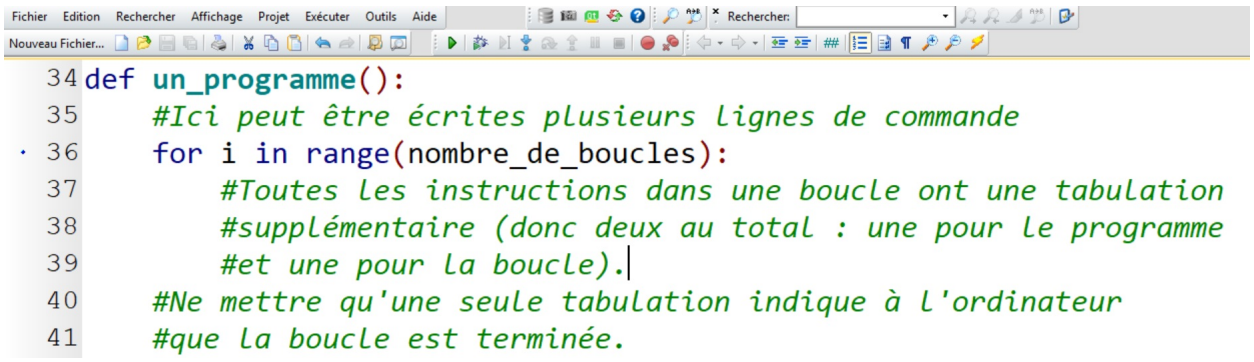
.....

Définition III.1

Une boucle *pour* permet de faire faire une même opération plusieurs fois à l'ordinateur sans avoir besoin de se répéter dans les instructions du programme.

Cela paraît au premier abord juste une économie d'écriture mais en réalité, une boucle pour est théoriquement plus importante que cela : il se peut que l'on ne connaisse pas à l'avance le nombre de fois où il faudra faire l'opération, lorsque ce nombre est un paramètre du programme.

Une boucle s'écrit de la façon suivante :



Exercice 8. Ecrire un programme qui

1. prend n le nombre d'années en paramètre

2. qui part d'une population $u = 0$
3. qui utilise une boucle *pour* pour augmenter la population u de 3 pendant n années.
4. retourne la valeur finale de la population.

On souhaite garder en mémoire toutes les valeurs de la population au cours du temps (comme lors de la population constante). La variable u ne sera donc plus un seul réel mais une liste. Pour comprendre l'utilisation d'une liste faire l'exercice suivant.

Exercice 9.

1. Dans la fenêtre d'exécution définir la liste suivante :

```
u=['carotte','poireau','patate']
```

   puis l'afficher.
2. Exécuter alors la commande suivante :

```
u.append('navet')
```

   puis l'afficher.
3. A quoi sert cette commande ?

   .....

   .....
4. Exécuter `u[1]`, puis `u[2]`, `u[3]` et `u[4]`. A quoi correspondent ces termes ?

   .....

   .....

Exercice 10. Ecrire un programme qui

1. prend n le nombre d'années en paramètre
2. crée une liste u ne contenant que 0 (la population initiale)
3. utilise une boucle pour ajouter chaque année la nouvelle population dans la liste.
4. retourne la liste.

Exercice 11. Modifier le programme précédent pour dessiner la graphe de la population (voir l'exercice 6).

Exercice 12. Modifier le programme précédent pour qu'il prenne également en paramètre p le nombre de nouveaux habitants par an et u_0 la population initiale. On fera également en sorte que le programme ne retourne que la dernière valeur de la population (et non toute la liste).

IV La croissance géométrique

On considère une population de cellules et on suppose qu'à chaque étape, chaque cellule se divise pour former deux cellules filles identiques à la cellule mère.

Si la population initiale est de $u_0 = 10$ combien a-t-on de cellule à la première génération ? A la deuxième ? A la troisième ?

.....

Exercice 13. Créer un programme qui prend en paramètre u_0 le nombre initial de cellules et n le nombre de générations que l'on observe puis qui trace la population au cours du temps et renvoie la population finale.

Exercice 14. Modifier le programme précédent pour qu'il prenne en paramètre p le nombre de nouvelles cellules filles créées par la cellule mère (p valait 2 dans l'exercice précédent).

V La croissance arithmético-géométrique

On suppose que l'année n, la population u_n subit la transformation affine suivante pour donner la population u_{n+1} :

$$u_{n+1} = au_n + b.$$

Exercice 15. Créer un programme qui prend en paramètre la population initiale u_0, les coefficients a et b et n le nombre de générations que l'on observe puis qui trace la population au cours du temps et renvoie la population finale.

Si $a > 1$ et $b > 0$, qu'observe-t-on ?

.....

.....

.....

Si $a > 1$ et $b < 0$, qu'observe-t-on ?

.....

.....

.....

Si $a \in [0; 1[$, qu'observe-t-on ? Calculer $\frac{b}{1-a}$. En quoi cette valeur est-elle importante ? Le vérifier en regardant plusieurs valeurs de b.

.....

.....

.....

.....

.....