

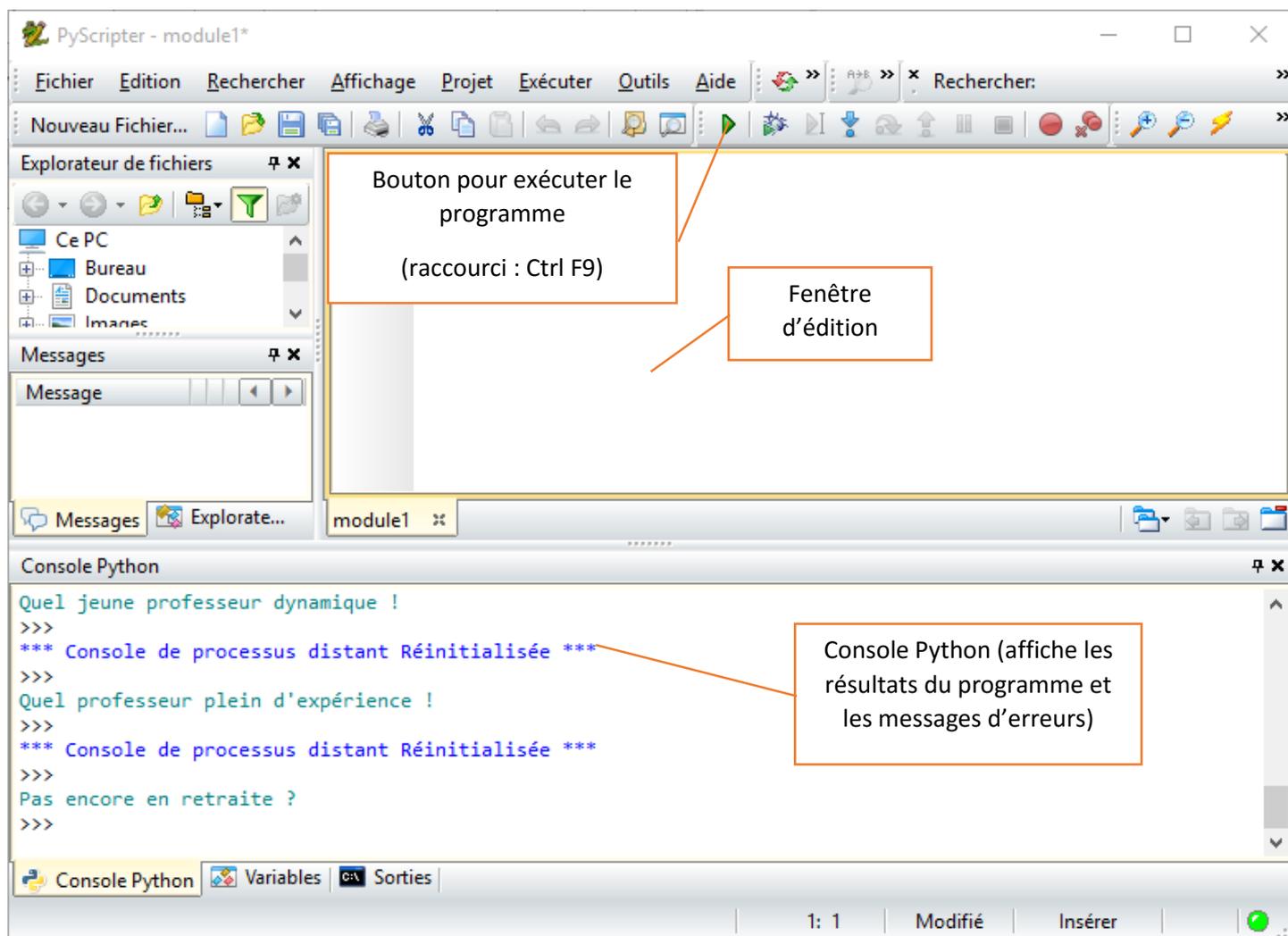
Python : les bases

Ce qu'il faut savoir pour la partie algorithmique du programme de mathématiques au lycée

1. EduPython

Pour démarrer rapidement, nous utiliserons EduPython : il s'agit d'une distribution de python dans laquelle plusieurs modules supplémentaires sont installés et prêts à l'emploi, notamment un module 'lycee' spécialement créé pour le lycée (!). EduPython est un logiciel libre et portable : il s'utilise sans installation et fonctionne même sur une clé USB.

EduPython est muni du logiciel PyScripter pour l'édition des programmes et leur exécution.



2. Entrées, sorties, affectation, calculs

Fonctionnalité	Syntaxe	Exemple
Affecter une valeur à une variable	=	a = 3 b = "toto" c = [1, 2, 3, "partez"]
Saisir un texte (par l'utilisateur)	input()	nom = input("Quel est votre nom ?")
Saisir un nombre (par l'utilisateur)	eval(input())	x = eval(input("Entrez la valeur de x : "))
Afficher le contenu d'une variable	print()	print(nom)
Affichage composé	print(...,...)	print("La valeur de x est :",x)
Opérations courantes sur les nombres	+, -, *, /	
Division euclidienne	// (quotient) et % (reste)	
Puissance	**	x = 2**3 (résultat : 8)

À savoir !

- Les variables représentant des nombres peuvent être de deux types distincts : 'int' pour des nombres entiers et 'float' pour des nombres décimaux. La commande 'eval()' reconnaît automatiquement le type de l'entrée de l'utilisateur. Pour certaines utilisation (notamment pour le compteur dans les boucles, voir ci-dessous), il est nécessaire d'utiliser une variable de type 'int'.
- Les autres fonctions mathématiques (comme la racine carrée) ne sont pas disponibles directement dans python. Pour les utiliser, il faut déjà les importer en ajoutant la ligne suivante au début du programme :

```
from math import *
```

Les fonctions disponibles sont alors : sqrt(), cos(), sin(), etc ...

- Les fonctions aléatoires sont définies dans le module 'random'. Pour les utiliser, il faut ajouter la ligne suivante en début de programme :

```
from random import *
```

Parmi toutes les fonctions disponibles, on a : randint(a,b) (entier aléatoire entre a et b inclus).

- Si on utilise EduPython, au lieu d'écrire les deux ligne précédentes, il suffit d'insérer l'unique ligne suivante en début de programme, et toutes les librairies utiles seront chargées :

```
from lycee import *
```

3. Les boucles

En python, il n'y a pas d'instruction du type début / fin de pour, de tant que, de si ...

Les blocs de commandes sont introduits par une ligne se terminant par deux points ':' et matérialisés par une indentation de 4 espaces. C'est une règle de syntaxe générale à respecter scrupuleusement.

Fonctionnalité	Syntaxe	Exemple
Boucle pour	<pre>for ... in range(...): </pre>	<pre>1 a = 0 2 for k in range(5): 3 print(k) 4 a = a+2 5 print(a)</pre>

Analyse de l'exemple

- les lignes 3 et 4 sont précédées d'une ligne 'for' terminée par deux points et sont indentées de 4 espaces à gauche. Ce sont les instructions des lignes 3 et 4 qui seront répétées 5 fois.
- La ligne 5 n'est pas indentée de 4 espaces à gauche, elle ne fait donc pas partie de la boucle for.
- Voici ci-contre le résultat de l'exécution de ce programme.
- Attention ! La variable 'k' qui sert de compteur dans la boucle 'for' n'a pas besoin d'être définie auparavant. C'est une variable de type 'int' et elle prend successivement les valeurs 0, 1, 2, 3, 4 (on a bien 5 valeurs différentes, mais en commençant à 0).
- D'une façon générale, la fonction 'range()' fonctionne comme suit :

range(n)	0, 1, 2, ..., n-1
range(a,b)	a, a+1, a+2, ..., b-1

```
>>>
0
1
2
3
4
10
```

range(a, b, p)	a, a+p, a+2p, ..., dernière valeur strictement inférieure à b
----------------	---

Fonctionnalité	Syntaxe	Exemple
Boucle tant que	while ... :	<pre> 1 k = 0 2 while k<=10: 3 k = k+1 4 print(k) 5 print("La boucle est bouclée") </pre>

Analyse de l'exemple

- On retrouve les mêmes règles d'indentation pour le bloc de commandes à répéter.
- La sortie de la boucle est conditionnée par le test d'une condition. La syntaxe pour les tests est la suivante :

Égalité	==
Différence	!=
Inférieur ou égal	<=
Supérieur ou égal	>=
Strictement inférieur	<
Strictement supérieur	>
Et logique	and
Ou logique	or

4. Instructions conditionnelles

On respecte toujours la structure avec les indentations et les deux points.

Les mots-clefs sont 'if', 'elif' et 'else'. 'elif' est la contraction de else if et permet d'éviter d'empiler plusieurs structures conditionnelles l'une dans l'autre.

L'exemple suivant doit être suffisamment explicite ...

```

1 age = eval(input("Quel est ton âge ?"))
2
3 if age < 23 :
4     print("Que fais-tu là ?")
5 elif age <= 40 :
6     print("Quel jeune professeur dynamique !")
7 elif age <= 62 :
8     print("Quel professeur plein d'expérience !")
9 else :
10    print("Pas encore en retraite ?")

```

5. Pour aller plus loin ...

Lire et travailler avec la documentation EduPython disponible à l'adresse suivante :

<http://download.tuxfamily.org/edupython/EduPython1.0.pdf>