

ALGORITHMIQUE

B – Programmation d'un algorithme en langage Python – les bibliothèques

Les algorithmes peuvent être programmés sur un ordinateur ou une calculatrice avec un langage de programmation adapté.

Programmer un algorithme, c'est le traduire dans un langage compréhensible par un logiciel donné.

1. La programmation en Python

Quand on ouvre l'interface d'une distribution Python, on découvre deux fenêtres : une fenêtre d'édition des programmes, l'**éditeur**, et une fenêtre où le programme s'exécute, la **console**.

Dans la console, on peut :

- faire des calculs, définir des variables et les intégrer dans des calculs ;
- écrire et exécuter un programme ;
- exécuter un programme saisi dans l'éditeur et demander les valeurs prises par les variables de ce programme.

```
# Créé par Michel, le
from lycee import *
x=3
y=5
z=x+y
x=z**3
y=y-z/4
```

Console Python

```
*** Python 3.4.5 [Continuum Analytics,
(default, Jul 5 2016, 14:56:50) [MSC
.1600 32 bit (Intel)] on win32. ***
>>> x
512
>>> y
3.0
```

- **La console** permet d'exécuter des instructions, des calculs, de lire la valeur de variables, etc. On reconnaît le mode console à l'invite de commande >>>.

```
>>> 5*6+7
37
>>> a=101
>>> a*a
10201
>>> type(a)
<class 'int'>
>>> |
```

- **L'éditeur** permet de taper le texte d'un programme. Suivant les options choisies, une coloration syntaxique met en valeur les mots-clés du langage. L'exécution du programme contenu dans l'éditeur affiche dans la console le résultat produit par l'exécution du programme (il se peut que rien ne s'affiche, selon les instructions du programme).

2. Les commandes de base

a) Après avoir ouvert Edupython, fermer les éventuels modules déjà ouverts car ceux-ci risquent de poser problème lors de la compilation.

Créer un nouveau programme : dans **Fichier**, choisir **Nouveau** puis **Nouveau Module Python** puis retourner dans **Fichier** et **Sauvegarder sous**. Faire une sauvegarde du programme sur votre clé USB dans le dossier approprié.

b) **Instruction d'affectation** : $c \leftarrow a$ s'écrit en code Python : $c = a$.

Un cas particulier important : l'incrémentation de 1

En algorithmique, il arrive souvent que la valeur d'une variable entière augmente d'une unité (cas des compteurs).

La formule à utiliser est la suivante : $x = x + 1$, $i = i + 1$, $compteur = compteur + 1$...

Exemple de programme:

```
x=5
x=2*x
x=x+1
```

Etat des variables

x
5
10
11

11 est la dernière valeur stockée par la variable x

c) Pour séparer deux instructions successives d'un programme, on peut soit aller à la ligne ; soit séparer ces deux instructions par un point-virgule (;).

d) Pour exécuter un programme, cliquer sur **Exécuter** (sur  ou sur F5 selon les versions).

Remarque : on peut écrire un commentaire dans un programme en le précédant du symbole #. Ce texte n'est pas pris en compte lors de l'exécution du programme.

e) Pour tester l'égalité de deux valeurs, on utilise l'opérateur `==`.

Exemple : on saisit dans la console $c = 3$, puis $c == 5$. On affecte à la variable c la valeur 3, puis on teste si celle-ci est égale à 5. C'est faux, donc le résultat affiché est : `False`.

Remarque : l'instruction `print(a)` permet d'afficher la valeur de la variable a dans un programme.

3 – Les bibliothèques

- ▶ Une bibliothèque est un ensemble de fonctions et de constantes (comme π) prêtes à être utilisées. Il existe une multitude de bibliothèques ayant chacune sa thématique propre.
- ▶ Pour en utiliser une, il faut la charger dans le programme. On dit alors qu'on importe la bibliothèque.

a) La bibliothèque pandas

La bibliothèque pandas est utilisée pour la structuration des données. Elle permet notamment d'exploiter des données sous format csv.

Pour importer la bibliothèque pandas, on écrit en début de programme : **`import pandas`**

b) La bibliothèque PIL

PIL ou pillow une bibliothèque python permettant la manipulation des images.

Pour importer la bibliothèque PIL, on écrit en début de programme : **`from PIL import Image`**

c) La bibliothèque math

- ▶ La bibliothèque math regroupe de nombreuses fonctions et constantes mathématiques.
- ▶ La fonction nommée `sqrt` permet de calculer la racine carrée d'un nombre, la constante nommée `pi` donne la valeur approchée de π .

Exemple : dans la ligne 1, on importe toutes les méthodes de la bibliothèque math.

Dans la ligne 2, on affecte $\sqrt{2}$ à la variable a .

Dans la ligne 3, on affecte une valeur approchée de π à la variable b .

Remarque : on peut également calculer les sinus, cosinus et tangente d'un angle avec les fonctions `sin`, `cos` et `tan`.

```
1 from math import *
2 a=sqrt(2)
3 b=pi
```

d) La bibliothèque pylab ou matplotlib

La bibliothèque pylab ou matplotlib regroupe des outils pour tracer et visualiser des données sous forme de graphiques.

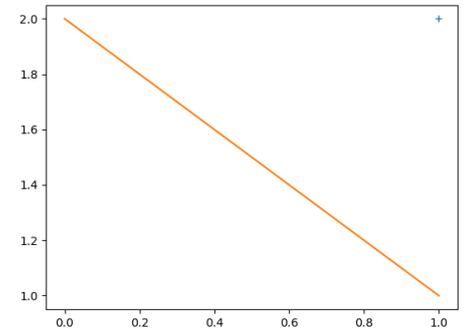
Exemple : dans la ligne 1, on importe la bibliothèque pylab.

Dans la ligne 2, on place sur le graphique le point de coordonnées (1 ;2). Le paramètre « + » permet d'obtenir une croix.

Dans la ligne 3, on trace le segment passant par les points de coordonnées (0 ;2) et (1 ;1). Notons qu'il faut mettre les abscisses entre crochets puis les ordonnées.

Dans la ligne 4, l'instruction show() permet d'afficher toutes les figures.

```
0 # Créé par ORDI, Le 28/6
1 from pylab import*
2 plot(1,2,"+")
3 plot([0,1],[2,1],'-')
4 show()
```



e)La bibliothèque random

lienmini.fr/10557-07

► La bibliothèque random permet de créer des nombres aléatoires. Elle contient en particulier la fonction random qui génère un nombre aléatoire entre 0 et 1 (1 exclu) et la fonction randint qui génère un entier aléatoire entre deux bornes (leurs valeurs incluses).

■ Exemple : dans la ligne 1, on importe la bibliothèque random.

Dans la ligne 2, on affecte un nombre aléatoire compris entre 0 et 1 (1 exclu) à la variable a.

Dans la ligne 3, on affecte un nombre aléatoire compris entre 1 et 6 (1 et 6 inclus) à la variable b.

```
1 from random import*
2 a=random()
3 b=randint(1,6)
```

f)La bibliothèque microbit

Cette bibliothèque permet de contrôler des cartes micro-bits.

Pour importer la bibliothèque microbits , on écrit en début de programme : **from microbit import ***

4.Commandes élémentaires de calcul

Commande	Donne
a+b	Somme de a et b
a-b	Différence entre a et b
a*b	Produit de a par b
a/b	Quotient de a par b sous forme de flottant
a//b	Quotient dans la division euclidienne de a par b
a%b	Reste dans la division euclidienne de a par b

Commande	Donne
x**n	x à la puissance n
abs(x)	La valeur absolue de x
int(x)	La partie entière de x
round(x,n)	La valeur arrondie de x à n décimales

5.Chaines de caractères et booléens

• Les chaînes de caractères se définissent à l'aide de guillemets doubles ou simples.

Par exemple : "papa" ou 'papa'.

Obtenir la longueur d'une chaîne de caractères C	len(C)
Extraire un caractère d'une chaîne de caractères C	C[k] renvoie le (k+1) ^e élément de C
Concaténer deux chaînes de caractères	"abcd" + "efg" donne "abcdefg"

• Le booléen Vrai est : **True**. Le booléen Faux est : **False**.

Exemple :

```
>>> a="vivelesmaths"
>>> len(a)
12
>>> a[4]
'l'
>>> a+a
'vivelesmathsvivelesmaths'
>>> b=(5<6)
>>> b
True
>>> |
```

Vidéo : lienmini.fr/10557-07 **QCM :** lienmini.fr/10557-08

Programmation sur TI : mathssa.fr/ti1 et mathssa.fr/ti2



CASIO

NUMWORKS

TEXAS INSTRUMENTS

Pour créer un programme

- 1 Sélectionner le menu Python à l'aide des flèches du curseur, puis valider avec **EXE**. Sélectionner **NEW**.
- 2 On complète au-dessous de « Nom du script » en donnant un nom au programme, puis on valide avec **EXE**. On se trouve dans l'éditeur, prêt à saisir les instructions.

- 1 Sélectionner l'icône Python à l'aide des flèches du curseur, puis valider avec **OK**.
- 2 Sélectionner **Ajouter un script** avec les flèches du curseur, puis **OK**.
- 3 Donner un nom au programme (suivi de .py déjà noté), puis deux fois sur **OK**. On se trouve alors dans l'éditeur.

- 1 Sélectionner le menu **apps** (touche **résol**), puis le sous-menu **Python** valider avec **entrée**.
- 2 Sélectionner **Nouv** (touche **zoom**).
- 3 Donner un nom au programme, suivi de **OK** (touche **graphe**). On se trouve alors dans l'éditeur.

Pour saisir un programme

On peut écrire les instructions lettre à lettre ou les chercher dans **CATALOG** (**SHIFT** **4**). Dans le catalogue, on obtient les instructions en tapant la première lettre de leur nom. On accède aux catégories par **CAT** :

2 : **Built-in** : def :return, for, if, while, and, or...
 3 : **math** : math import...
 4 : **random** : random import...
 On a la barre d'outils suivante :

FILE **RUN** **SYMBOL** **CHAR** **A↔a**

- 1 En sélectionnant **SYMBOL**, on a accès aux symboles suivants :
: ; # ' "
- 2 Le menu **CHAR** donne accès aux lettres, aux chiffres...
- 3 Par appui sur **▶**, on accède aux onglets suivants :
COMMAND **OPERAT** **JUMP** **SEARCH**

- 4 En sélectionnant **COMMAND**, on a accès à if, for et while.
 - 5 En sélectionnant **OPERAT**, on a accès aux fonctionnalités :
= ! > < %
- La touche **EXIT** permet de sortir d'un menu.

Utiliser les menus déroulants accessibles avec la touche **Toolbox**. On passe d'un menu à l'autre avec les flèches du curseur. Pour ouvrir le menu, on le sélectionne et on valide par **OK**.

- 1 **Le menu Boucles et tests** comprend quatre sous-menus : **For**, **If**, **While** et **Conditions** (comme $x < y$, $x == y$...).
- 2 **Le menu Modules** contient plusieurs modules (bibliothèques de fonctions) que l'on peut importer. Par exemple : **math**, **cmath** ou **random**. La liste des fonctions d'un module s'obtient en sélectionnant le module.
- 3 **Le menu Catalogue** comprend des fonctions comme **min**, **max**, **sin**...

- 4 **Le menu Fonctions** comprend les instructions **def fonction(x)** et **return**.
La touche **↶** permet de revenir au menu précédent.

Dans l'éditeur, on a accès à différents menus.

- 1 **Menu Fns** (touche **f(x)**), avec plusieurs onglets, parmi eux :
 - l'onglet **Ctl** permet de sélectionner toutes les instructions if, for, while...
 - l'onglet **Ops** comprend les tests ($x == y$, $x > y$...), les connecteurs or, and et les booléens True, False.
 - l'onglet **Fonc** donne accès à def fonction() et return.
 - l'onglet **Modul** pour les modules math et random.
- 2 **Menu a A #** (touche **fenêtre**) : lettres de l'alphabet, certains symboles (#, ", ...), des connecteurs (and, or) et des booléens (True, False).
- 3 **Menu Outils** (touche **zoom**) : il permet d'insérer une indentation, une ligne...

- 4 **Menu Script** (touche **graphe**) : il permet d'accéder à tous les programmes de la calculatrice.
On sort d'un menu avec la commande **Echap** (touche **f(x)**).

Pour exécuter un programme

Le programme étant saisi, on choisit **RUN**, puis on tape le nom de la fonction pour exécuter le programme dans la console.

On se place dans la console. Pour cela, on choisit **Console d'exécution**, puis **OK**.

Dans le gestionnaire de scripts, choisir à l'aide des touches de défilement le programme voulu, puis sélectionner le menu **Exéc**.

Onglet E/S :
lire ou
afficher une
valeur