

# ACT2 : NOTIONS DE BASE ET LANGAGE PYTHON

## I. NOTION D'ALGORITHME ET DE PROGRAMME

Un **algorithme** est une suite d'instructions : une recette de cuisine est un algorithme.

Il est composé :

- Des entrées (les ingrédients)
- Du traitement (les différentes étapes de la recette)
- De la sortie (un bon gâteau ou tout autre plat de votre choix ...)

Un algorithme peut s'écrire en langage naturel, ou en pseudo-code.

### **Exemple 1 :**

Langage naturel :

*a prend la valeur 3*

*b prend la valeur 5*

*c prend la valeur  $a \times b + 2 \times a$*

*Ecrire c*

pseudo-code

$a \leftarrow 3$

$b \leftarrow 5$

$c \leftarrow a * b + 2 * a$

*Afficher c*

*Dans cet exemple, identifier les entrées, le traitement et la sortie.*

*Compléter le tableau suivant, chaque ligne du tableau correspondant à une ligne de l'algorithme. Préciser la valeur de la sortie.*

étape	Valeur de a	Valeur de b	Valeur de c
1	3		
2			
3			

**Exemple 2 :** Reprendre les questions précédentes avec l'algorithme écrit en langage naturel suivant :

*a prend la valeur -1*

*b prend la valeur 2*

*a prend la valeur  $a^2$*

*Multiplier a par 5*

*b prend la valeur  $a \times b$*

*Afficher a*

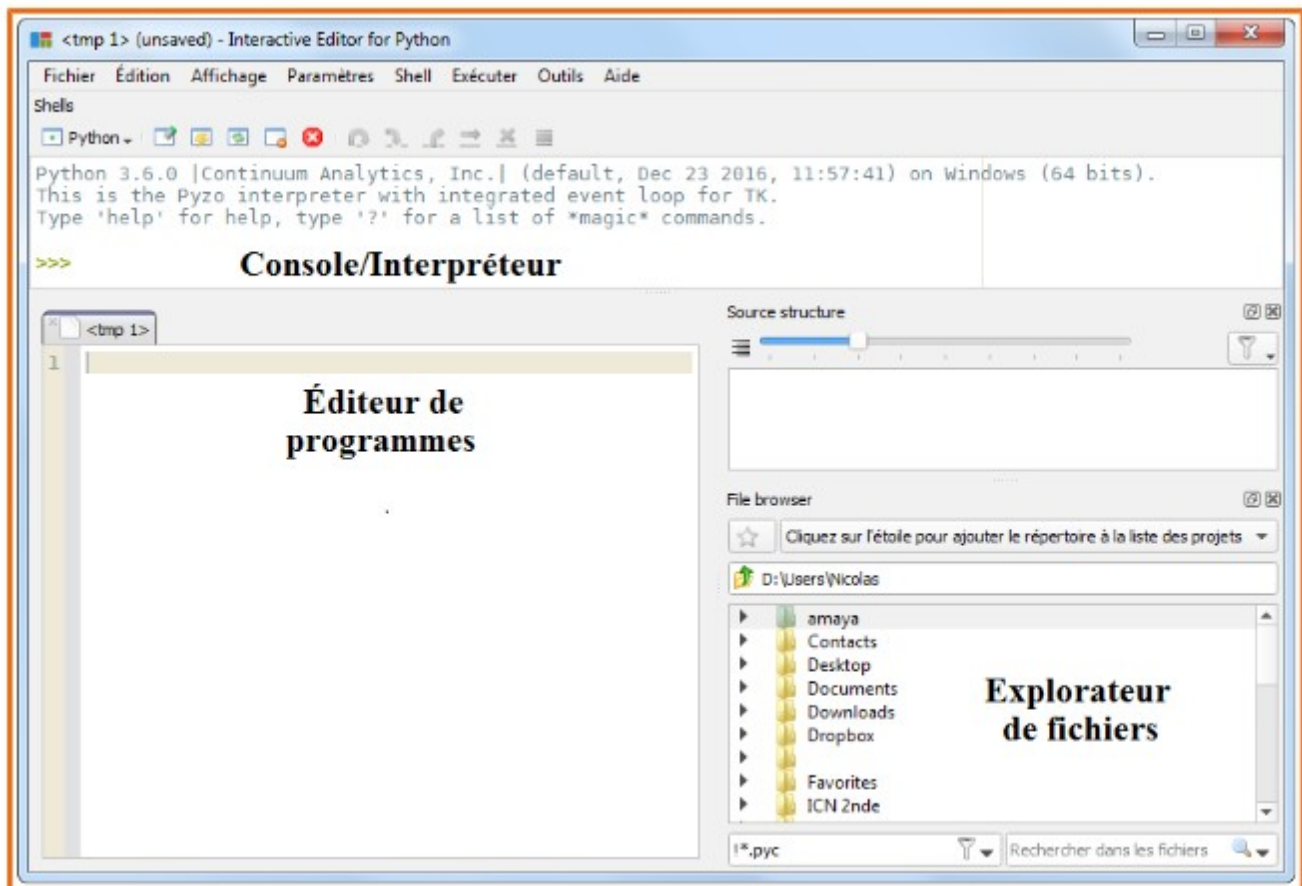
*Afficher b*

étape	Valeur de a	Valeur de b
1	-1	
2		
3		
4		
5		

Un **programme** est la traduction d'un algorithme dans un langage compréhensible par un ordinateur. Il est composé d'**instructions**, une instruction correspondant à une action.

## II. ENVIRONNEMENT PYTHON

Vous avez travaillé avec Scratch au collège. Au lycée, nous utiliserons le langage de programmation Python, et l'environnement de travail Edupython ou Pyzo.



Dans un premier temps, nous utiliserons uniquement la console.

**Exercice 1 :** Python est un langage interprété : lorsque l'on écrit une instruction dans la console, après les trois chevrons `>>>` et que l'on appuie sur la touche Entrée, cette instruction est immédiatement exécutée.

1°) Ecrire des calculs mathématiques simples dans la console faisant intervenir des nombres entiers, relatifs, réels, du type addition, soustraction, multiplication ...

2°) Faire de même avec des chaînes de caractères (du texte : « bonjour » par exemple, penser aux guillemets). Eh oui, on peut additionner sans souci des chaînes de caractères en Python (on parle de concaténation) :

```
>>> "hello"+"world"
'helloworld'
```

**Exercice 2 :** Compléter par le résultat du calcul écrit en Python :

$3+4*7$	$2**3$	$2*(3**2)$	$21/2$	$5/2*3$	$212*10*-2$

### III. LES VARIABLES

Pour écrire un programme, il faut enregistrer les données nécessaires au traitement dans des espaces mémoires.

Les **variables** désignent des emplacements de stockage. Elles sont repérées par des noms, et leur valeur peut évoluer au cours du temps.

Les variables peuvent être de différents types : entier (int), décimal (on parle de flottant : float), texte (on parle de chaîne de caractère : str).

En Python, on donne une valeur à une variable à l'aide de l'instruction `=`. Par exemple `a=2` signifie que la variable `a` contient le nombre entier 2. On parle **d'affectation de variable**.

#### ***Exemples :***

```
>>> a=2
>>> b=5
>>> c=a*b
>>> c
10
>>> article="le"
>>> nom="lycée"
>>> verbe="est"
>>> adjectif="chouette"
>>> phrase=article + nom + verbe + adjectif
>>> phrase
'lelycéeestchouette'
```

**Exercice 3 :** Quelles sont les variables utilisées dans l'exemple ci-contre ? Préciser la valeur de chacune d'elles. Change-t-elle lors de l'exécution du programme ?

```
>>> a=3
>>> b=5
>>> c=a*b+2*a
>>> print(c)
21
```

Même question avec le programme ci-contre.

Quelles sont les différences entre ces deux programmes ?

```
>>> a=3
>>> b=5
>>> a=a*b+2*a
>>> print(a)
21
```

**Exercice 4 :** Quelle est la valeur de  $x$  après exécution de la séquence d'instructions ci-contre ?

```
>>> x=4
>>> x=x+1
>>> x=x*x
```

**Exercice 5 :** Compléter le tableau en indiquant les valeurs successives de `a` et `b` au fur et à mesure de l'exécution de la séquence d'instructions donnée :

Instructions
<code>&gt;&gt;&gt; a=4</code>
<code>&gt;&gt;&gt; b=3</code>
<code>&gt;&gt;&gt; a=b</code>
<code>&gt;&gt;&gt; a=a+b</code>

<code>a</code>	<code>b</code>
4	

## IV. LES INSTRUCTIONS ELEMENTAIRES


Il existe deux types d'instructions élémentaires :

- Les opérations sur les variables : opérations mathématiques sur les nombres, longueur d'une chaîne de caractère ...
- Les instructions d'entrée et de sortie :
  - ✓ Une instruction d'entrée permet à un programme de lire des valeurs saisies au clavier par l'utilisateur
  - ✓ Une instruction de sortie affiche la (ou les) valeur(s) de variables à l'écran.

En Python, l'instruction d'entrée *input* permet d'affecter la valeur saisie dans une variable et l'instruction de sortie *print* permet d'afficher à l'écran la valeur de variables.

Attention : l'instruction *input* permet d'obtenir des chaînes de caractères. Si on veut saisir un entier, il faut écrire `int(input(...))`, si on veut saisir un décimal, il faut écrire `float(input(...))`. On peut mettre dans la parenthèse après le *input* un texte qui s'affichera lors de l'exécution.

### Exercice 6 :

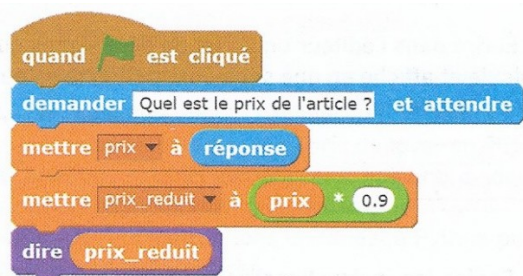
1. Recopier le programme suivant **dans la zone d'écriture du code**, puis l'exécuter en cliquant sur  dans la barre d'outils. Que fait ce programme ?

```
a=input("entrez un mot")
b=2*a
print (b)
```

2. Mêmes questions avec le programme :

```
a=float(input("entrez un nombre"))
b=2*a
print (b)
```

### Exercice 7 : Traduire en langage Python le programme suivant écrit avec Scratch.



**Exercice 8 :** La distance d'arrêt d'un véhicule lors d'un freinage d'urgence est la distance qu'il parcourt entre le moment où le conducteur voit l'obstacle et le moment où le véhicule est complètement arrêté. Une formule permettant de calculer cette distance d'arrêt est

$$d = 0,006 v^2 + \frac{5}{18} v, \text{ où } d \text{ est exprimée en mètre et } v \text{ en km.h}^{-1}.$$

1. Ecrire un script qui demande la vitesse du véhicule (nombre entier exprimé en  $\text{km.h}^{-1}$ ) et affiche sa distance d'arrêt.
2. Déterminer la distance d'arrêt d'un véhicule qui roule à  $50 \text{ km.h}^{-1}$ .