

## Automatisme 6 : programmation sur Python

### 1. Variables – affectation et commande

Définir une **variable**, c'est associer un nom à une valeur qu'on peut modifier au cours du temps. Le tableau d'état des variables est le tableau donnant l'état des variables à chaque étape de l'algorithme.

La **nature** de la variable porte le nom de **type**.

Pour affecter une valeur à une variable, on utilise le symbole « = »

Commande	Donne
$x**n$	$x$ à la puissance $n$
$a//b$	Quotient dans la division euclidienne de $a$ par $b$
$a\%b$	Reste dans la division euclidienne de $a$ par $b$

Nom du type	Mot clé en Python	Exemple
entier relatif (integer)	int	a=4
nombre décimal (flottant)	float	a=2.3
texte ou chaîne de caractères (string)	str	a="bonjour"
booléen	bool	<pre>&gt;&gt;&gt; a=(6&gt;5) &gt;&gt;&gt; a True &gt;&gt;&gt; type(a) &lt;class 'bool'&gt;</pre>
liste	list	<pre>&gt;&gt;&gt; a=[1,2,3] &gt;&gt;&gt; a [1, 2, 3] &gt;&gt;&gt; type(a) &lt;class 'list'&gt;</pre>

la virgule devient un point sur python

a contient le test (6>5).  
Sa valeur est Vrai

### Opération sur les listes :

- Initialisation : L=[...] ou L=[] (liste vide)
- Extraire un élément L[i]  
L[0] est le 1<sup>er</sup> élément
- Ajouter un élément :  
L=L+[.] ou L.append(...)
- Nombre d'élément d'une liste : len(L)

```
>>> L=[2,3,4,5,"m"]
>>> L[0]
2
>>> L[4]
'm'
>>> L=L+[7]
>>> L
[2, 3, 4, 5, 'm', 7]
>>> len(L)
6
>>> L.append(8)
>>> L
[2, 3, 4, 5, 'm', 7, 8]
```

**Application :** dressons le tableau d'état des variables pour le programme ci-dessous :

```
x=2;y=3
z=x+y+x*y
z=z**2
y=z/2
```

x	y	z
2	3	
2	3	11
2	3	121
2	60.5	121

En sortie de programme, x contient 2 , y contient 60,5 et z contient 121.

## 2.Les bibliothèques

► 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  $\pi$ .

■ **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  $\pi$  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
```

► 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)
```

## 3.Fonction :

► Les instructions contenues dans la fonction sont décalées vers la droite. Ce décalage, appelé **indentation**, peut être réalisé en créant 2 ou 4 espaces ou en utilisant la touche tabulation.

Pour utiliser une fonction , on doit faire « **appel** » à elle c'est-à-dire dans la console écrire le nom avec les valeurs des paramètres dans la parenthèses.

## 4.Structure conditionnelle :

► Une instruction conditionnelle est une instruction qui n'est exécutée que si une condition est réalisée. Autrement dit, si une condition est réalisée **alors** l'instruction est réalisée.

► La condition est suivie de deux points.

► Les instructions liées à la condition doivent être **indentées** (décalées vers la droite). On peut dire que c'est l'indentation qui remplace le mot « **alors** » qui n'existe pas en langage Python.

## Remarques :

- La structure « if .... else » sera utilisée lorsqu'il y a deux conditions possibles ; la structure « if ...elif...else » lorsqu'il y a plus de deux conditions possibles
- Pour tester une égalité , on écrit « == »

### Exemple :

Un club sportif fait fabriquer des tee-shirts au nom du club. Chaque tee-shirt est facturé 4 € mais ils sont facturés 3,50 € l'un si la commande est d'au moins 100 unités du produit.  
Soit  $n$  la variable égale au nombre de tee-shirts commandés et  $p$  le prix payé par le club.

Ecrire une fonction python « prix » permettant de déterminer le prix  $p$  en fonction du nombre  $n$  de tee-shirts vendus.

.....  
.....  
.....  
.....  
.....

## 5.Boucle bornée ou boucle pour :

Exemple en langage naturel:

**Pour** Marche d'escalier allant de 1 à 10  
Monter sur la marche suivante  
**Fin Pour**

<b>Pour</b> Variable allant de Valeur début à Valeur fin Instructions <b>Fin Pour</b>
---

On peut répéter les mêmes instructions pour un nombre de répétitions prédéfini par une **variable**. appelée **compteur**. Cette boucle est dite **bornée**.

Python	Remarques
<pre>for k in range(d, n + 1):     {instructions}</pre>	L'instruction <code>for k in range(d, n + 1)</code> fait parcourir à la variable $k$ tous les entiers de $d$ à $n$ . Lorsque $d = 0$ , on peut remplacer <code>range(d, n + 1)</code> par <code>range(n + 1)</code> .

Exemple 1: le programme ci-dessous permet de simuler 100 lancers d'un dé équilibré à 6 faces.

Compléter le programme ci-dessous.

```
• from ..... import *  
• for i in range(...):  
•     a=randint(...)  
•     print(a, end=" - ")  
# end="-" permet d'avoir  
# les résultats en ligne séparés par un tiret
```

Compléter le programme ci-dessous permettant de calculer la somme  $S = 0 + 1 + 2 + \dots + 100$

.....  
.....  
.....  
.....  
.....

## **6. Boucle non bornée ou boucle tant que :**

Lorsque l'on ne connaît pas à l'avance le nombre de répétitions, on doit faire appel à une boucle non bornée ou boucle tant que.

Exemple en langage naturel:

**Tant que** *Le verre n'est pas plein*  
*Verser de l'eau*  
**Fin Tant que**

<b>Tant que</b> <i>Condition est vraie</i> <i>Instructions</i> <b>Fin Tant que</b>
--

On peut répéter les mêmes instructions tant qu'une condition reste vérifiée.

Exemple en langage naturel:

**Tant que** *Le verre n'est pas plein*  
*Verser de l'eau*  
**Fin Tant que**

<b>Tant que</b> <i>Condition est vraie</i> <i>Instructions</i> <b>Fin Tant que</b>
--

On peut répéter les mêmes instructions tant qu'une condition reste vérifiée.

## **Programmation en Python :**

```
while condition:  
    instruction1  
    instruction2
```

Exemple : on dispose d'un capital de 10 000 € que l'on place à un taux d'intérêt annuel de 5%

Le programme ci-dessous permet d'afficher le nombre d'années nécessaire pour que la somme placée ait au moins doublé c'est-à-dire soit supérieure ou égale à 20 000 euros.

Compléter ce programme.

```
S=10000  
compteur=0  
  
while S<.....:  
    S=S*.....  
    compteur=.....  
print(compteur)
```