

Introduction à Python et l'algorithmique

2. Les principales commandes en Python

1. Les variables

Un algorithme (ou un programme informatique), agit sur des nombres, des textes, ... Ces différents éléments sont stockés dans des variables. On peut se représenter une variable comme une boîte portant une étiquette ("le nom de la variable") à l'intérieur de laquelle on peut placer un contenu.

En informatique, les variables sont des emplacements réservés dans la mémoire de l'ordinateur auxquels on attribue une étiquette.

Définition:

Déclarer une variable c'est indiquer le nom et le type (nombre texte, tableau,...) d'une variable que l'on utilisera dans l'algorithme.

Remarque:

En général, on déclare des variables se fait au début de l'algorithme avant la première instruction.

Définition:

Affecter une variable, c'est attribuer une valeur à cette variable. Si la variable contenait déjà une valeur, cette ancienne valeur est effacée.

Exemple:

Ecrivons l'algorithme suivant en Python:

- Affecter 10 à x
- Multiplier ce nombre par 3
- Multiplier ce dernier par 0,5

```
In [1]: x = 10  
print("x vaut : ", x)
```

```
x vaut : 10
```

```
In [2]: x = x * 3
        print("x vaut : ", x)
```

x vaut : 30

```
In [3]: x = x * 0.5
        print("x vaut : ", x)
```

x vaut : 15.0

On pourrait plus simplement calculer en 1 fois

```
In [4]: x = 10
        x = x * 3 * 0.5
        print("x vaut : ", x)
```

x vaut : 15.0

Les différents types de variables

Dans un algorithme ou un programme, les variables considérées ont des types qui définissent la nature des valeurs qu'elles peuvent prendre. Les 3 types principaux considérés en classe de Seconde sont:

- les entiers quand les valeurs possibles sont des nombres entiers. On les note *integer* ou *int*
- les flottants (notés *float*) quand les valeurs possibles de la variable sont des nombres réels
- les chaînes de caractères (dits *string* ou *str*) quand les valeurs possibles de la variable sont des mots ou des phrases

Exemple(s):

On peut vérifier le type de variable à l'aide de la commande `type()`

```
In [5]: x = 10
        print("x est une variable de type: ", type(x))
```

x est une variable de type: <class 'int'>

```
In [6]: y = "hello"
        print("y est une variable de type: ", type(y))
```

y est une variable de type: <class 'str'>

```
In [7]: z = 13 / 15
        print("z est une variable de type: ", type(z))
```

z est une variable de type: <class 'float'>

2. Instructions conditionnelles

Définition:

Dans un algorithme, on est parfois amené à exécuter une ou plusieurs instructions uniquement si une certaine condition est vérifiée, c'est ce que l'on appelle des instructions conditionnelles. Si la condition n'est pas vérifiée, on peut soit exécuter un autre bloc d'instructions, soit ne rien faire. Dans les deux cas, on exécute ensuite la suite de l'algorithme. Si {condition} Alors {Action} Sinon {Action2}

Exemple:

In [8]:

```
age = 16
if (age < 18):
    print("Vous êtes mineur")
else:
    print("Vous êtes majeur")
```

Vous êtes mineur

3. Les boucles

Il existe deux types de boucles:

- les **boucles bornées** (ou boucle "For") pour exécuter un nombre déterminé de fois un même bloc d'instructions. Ces boucles sont munies d'une variable compteur pour vérifier le nombre d'instructions
- les **boucles non bornées** (ou boucle "While") pour répéter un même bloc d'instructions tant qu'une certaine condition est vérifiée.

Exemple d'une boucle "For"

In [9]:

```
x = 0
for i in range(0, 5):
    x = x + 1

print("nouvelle valeur de x: ", x)
```

nouvelle valeur de x: 5

Dans cet exemple, on ajoute 1 à la valeur x à chaque tour de boucle.

Exemple d'une boucle "While"

In [10]:

```
y = 0
while (y < 10):
    y = y + 1

print("Nouvelle valeur de y: ", y)
```

Nouvelle valeur de y: 10

Exercice 1

Ecrire en Python l'algorithme suivant:

- Affecter à la variable `a` la valeur 12

- Multiplier `a` par 5
- Pour `i` allant de 0 à 8:
 - Soustraire `i` à `a`
- Afficher la nouvelle valeur de `a`

Correction:

In [11]:

```
a = 12
a = a * 5
for i in range(0, 8):
    a = a - i

print("valeur de a = ", a)
```

valeur de a = 32

Exercice 2

Ecrire en Python l'algorithme suivant:

- Affecter à la variable `estpair` la valeur 96
- Tant que `estpair` est divisible par 4
 - Diviser `estpair` par 4
- Afficher la nouvelle valeur de `estpair`

Astuce: en Python, on peut connaître le reste d'une division euclidienne on utilise le symbole `%`. Par exemple `6 % 3 = 0`

Correction:

In [12]:

```
estpair = 96
while (estpair % 4 == 0):
    estpair = estpair / 4

print("nouvelle valeur de estpair = ", estpair)
```

nouvelle valeur de estpair = 6.0

4. Les fonctions

Pour plus de lisibilité, ou plus simplement pour éviter de ré-écrire de multiples fois les mêmes instructions, il peut être utile de définir une **fonction** i.e. un bloc d'instructions qui ne sera exécuté que s'il est appelé.

Une fonction possède généralement des paramètres et retourne une valeur de retour (mais pas systématiquement)

Exemple:

In [14]:

```
def calcul_imc (masse, taille):  
    imc = masse / (taille* taille)  
    return imc  
  
imc_jean = calcul_imc (60, 1.6)  
print("L'IMC de Jean est : ", imc_jean)  
  
masse_henri = 85  
taille_henri = 2.05  
imc_henri = calcul_imc(85, 2.05)  
print("L'IMC de Henri est : ", imc_henri)
```

```
L'IMC de Jean est : 23.437499999999996  
L'IMC de Henri est : 20.226055919095778
```

5. Les modules Pythons utiles

Un module Python est une sorte de bibliothèque contenant un grand nombre de fonctions prêtes à l'emploi. Nous utiliserons dans cette année:

- **numpy** : pour les calculs de maths
- **matplotlib.pyplot** : pour dessiner des graphiques
- **random**: pour générer des nombres aléatoires

Pour importer un module Python, il suffit d'écrire au début du programme `import <nom_module>` puis, pour utiliser une fonction incluse dans un module, d'appeler la fonction de la manière suivante: `<nom_module>.<nom_fonction>`

Exemple:

Générer un nombre entier aléatoire entre 1 et 13

Correction:

In [16]:

```
import random  
print("voici un nombre aléatoire: ", random.randint(1, 13))
```

```
voici un nombre aléatoire: 5
```