

PYTHON 1 Les variables

Choix du nom

Il est inutile de déclarer les variables avant de les utiliser. Leur affectation décidera de leur type. Le nom de la variable doit toujours commencer par une lettre. Les majuscules sont prises en compte.

Certains mots sont interdits, car réservés à Python : *and - def - del - elif - else - for - from - global - if - import - return - while ...*

Remarque : Pour utiliser des accents dans les chaînes de caractères et dans les commentaires, placer la ligne #-*- coding: utf-8 -*- en début de fichier.

Types de variables

- Le type **int** : entier. Il n'est limité en taille que par la mémoire de la machine. Pour une 32-bits, ils vont de -2^{31} à $2^{31}-1$. Les entiers sont en base 10 par défaut, mais on peut aussi utiliser les bases 2 et 16.
- Le type **long** : Nombre entier aussi grand que l'on veut car sa taille n'est limitée que par la mémoire allouée par l'ordinateur à l'interpréteur Python.
- Le type **float** : nombres à virgule flottante : noté avec un point et non une virgule. Peut être donné en notation puissance de 10 : 2.718 ; 3^5 (=30000)
- le type **str** : chaîne . c'est un type de données non modifiables qui représente une séquence de caractères Unicode
- le type **bool** : booléen. Elle n'a que deux valeurs possibles : False, True.
- le type **list** : liste d'objets ou de variables, dans des crochets, séparés par une virgule et indicé à partir de 0.
- le type **tuple**, : liste de longueur fixe
- le type **dict** : dictionnaire,
- le type **complex** : nombre complexe que l'on note a + bj

Affectation

On **affecte une valeur à une variable** en utilisant le signe '='
Dans une affectation, le membre de gauche **reçoit** le membre de droite.
Python, permet **d'affecter plusieurs variables à la fois** ce qui est très utile.

Exemples : `a=123`

`liste = [1, 2, 3]`

`a, b, c = 1, 2, 3`

`a, b = b, a`

a reçoit 123

liste reçoit [1,2,3]

a reçoit 1, b reçoit 2 et c reçoit 3

intervertit les valeurs de a et b

PYTHON 2 :Les entrées/sorties

Les entrées

La fonction **input()** interrompt le programme, affiche une éventuelle question et attend que l'utilisateur entre une donnée et la valide par entrée. Elle renvoie systématiquement une chaîne de caractères. Donc si c'est un nombre que l'on veut manipuler, il faudra utiliser la fonction `int()` ou `float()` s'il s'agit d'un décimal pour convertir la chaîne en nombre.

Exemples :

`x = float(input(« x = ? »))`

Affiche `x = ?`, transtype le nombre rentré en float et l'affecte à la variable x

`nb = int(input(« Entrez un nombre entier : »))`

Affiche `Entrez un nombre entier :` l'affecte à nb avec le type entier

Les sorties

L'instruction **print ()** permet d'afficher des sorties à l'écran :

Exemple :

`a=3`

`b = 5`

`print("Le produit de", a, "par", b, "vaut :", a * b)`

`>>> Le produit de 3 par 5 vaut : 15`

Supprimer le retour à la ligne :

L'instruction `print (... , end='')` supprime le retour à la ligne dans une suite d'instruction (boucle, conditionnel...)