

INTRODUCTION A LA PROGRAMMATION PYTHON

MatheX – Licence CC BY-NC-SA 4.0 - <https://www.mathexien.com>

#2. Variable et Type de donnée

Objectifs:

- Introduire le concept de Variable
- Programmer des instructions d'affectation
- Différencier les types de données de base

Notion de variables

On a souvent besoin dans un programme de conserver une donnée en mémoire en vue d'un traitement ultérieur, on utilise pour cela une **variable** qui est représentée par un nom:

```
1 | a = 3
```

Cette instruction est une instruction d'**affectation**: on affecte à la variable nommée **a** la valeur **3**:

```
1 | a = 3
2 | print( a )
```

```
1 | 3
```

Pour faire simple, on peut voir une variable comme une boîte qui a un nom et qui contient une donnée.

Pour être plus précis, à une variable est associée une adresse mémoire et cette adresse mémoire contient la donnée.

On peut affecter à une variable directement une donnée mais aussi le contenu d'une autre variable, le résultat d'une expression arithmétique, la combinaison d'expressions, ... :

```
1 | a = 3
2 | a = a + 1    # on met dans a: (l'ancienne valeur de a) + 1
3 | b = a
4 | b = b * 2    # on met dans b: (l'ancienne valeur de b) * 2
5 | c = a ** b
```

Mission 2.1.

Déterminez la valeur des variables du programme ci-dessus:

1. sans exécuter le programme
2. en ajoutant des `print()` à la fin du programme et en l'exécutant

Mission 2.2.

Ecrire un programme qui demande à l'utilisateur son nom, le récupère, puis lui demande son prénom, et enfin affiche un message de bienvenue contenant son nom et son prénom.

Type de données

Le **type** d'une donnée correspond à la nature de la donnée, nous en avons déjà vu certains en #1, voici la liste des types de base:

Type	Nature	Exemple	Opération
int (integer)	nombre entier relatif	2 1000000 -3 2020	+ (addition) * (multiplication) / (division décimale) ** (puissance) // (quotient division euclidienne) % (reste division euclidienne)
float (floating point)	nombre décimal relatif	2.3 -1.7245	+ (addition) * (multiplication) / (division décimale)
str (string)	chaîne de caractères (caractère, texte)	"Hello" "2020"	+ (concaténation) * (concaténation multiple)
bool (boolean)	booléen	True False	and (et logique) or (ou logique) not (non logique)

Pour compléter, on verra plus tard dans la série un dernier type: les listes.

Le type d'une variable est le type de la donnée affectée à la variable.

Il est très important de maîtriser la typologie car les opérations et d'autres manipulations ne sont pas les mêmes selon le type de la donnée:

```
1 # Initialisation des variables (première affectation)
2 nombre_a = 3
3 nombre_b = 5
4 string_a = 'texte a'
5 string_b = 'texte b'
6 boolean_a = True
7 boolean_b = False
8
9 # opérations correctes
10 nombre_a + nombre_b # addition de deux nombres (résultat: 8)
11 string_a + string_b # concaténation de deux textes (résultat: 'texte atexte b')
12 boolean_a and boolean_b # et logique entre deux booléens (résultat: False)
13 boolean_a or boolean_b # ou logique entre deux booléens (résultat: True)
14 not boolean_a # non logique d'un booléen (résultat: False)
15
16 # opérations incorrectes: l'exécution entraîne une erreur (ou le résultat est inconsistant)
17 nombre_a + string_a # + entre un nombre et une chaîne de caractère
18 nombre_a and nombre_b # et logique entre deux nombres
19 not string_a # non logique d'un texte
```

On peut connaître le type d'une variable avec la fonction: `type(nom_variable)`

On peut changer le type d'une variable (lorsque c'est possible) avec une opération de conversion (casting):

```
1 # initialisation des variables
2 a = 3
3 print( type( a ) ) # l'instruction affichera int
4 b = '2.3'
5 print( type( b ) ) # l'instruction affichera str
6
7 # casting
8 c = str( a )      # la fonction str() convertit le nombre en string
9 print( type( c ) ) # l'instruction affichera str
10
11 d = int( c )     # la fonction int() convertit le texte en nombre entier (si c'est possible !)
12 print( type( d ) ) # l'instruction affichera int
13
14 e = float( b )   # la fonction float() convertit le texte en nombre décimal (si c'est possible !)
15 print( type( e ) ) # l'instruction affichera float
```

Mission 2.3.

On reprend la mission 2.2. mais en plus:

- on demande à l'utilisateur son année de naissance
- on lui précisera son âge (en fin d'année) dans le message de bienvenue

NB: la fonction `input()` renvoie toujours une string (et pas un nombre)

Vidéo

https://youtu.be/krt_dZwXsjk