

# Python et micro:bit - Cours

– Septembre 2022

## 2 Corpus élémentaire de Python

### 2.1 Affectation

#### Définition :

Affectation L'affectation de variable, représenté en python par le symbole =, est l'opération qui va associé une valeur (pas nécessairement numérique) à un nom de variable.

Pour suivre l'affectation de variable, on peut compléter un tableau d'état

```
1 a = 23
2 a = a + 1
3 b = 10
4 a = b
```

Instruction	a	b
ligne 1	23	
ligne 2	24	
ligne 3	24	10
ligne 4	10	10

### 2.2 Types de bases

Vous serez amené à utiliser essentiellement 4 types de base :

Nom	Les entiers signés	Les flottants	Les strings	Les booléens
Type	int	float	str	bool
exemple	2, -2, 0	2.2, -3.14	"2", "zaer"	True, False
Opérateurs	+ - * // % **	+ - * / **	+ *	or and not

Pour connaître le type d'une variable, on peut utiliser la fonction `type`

Exemple :

```
1 >>> a = "aer"
2 >>> type(a)
3 str
```

### 2.3 Conditions

Pour s'adapter le comportement d'un programme à différentes situations, on pourra utiliser les mots clés `if/elif/else`

Exemple :

```
1 age = str(input{"Que1 age as tu?"})
2
3 if age < 2: # Condition 1
4     print("Tu es un bébé") # block 1
5 elif age < 18: # Condition 2
6     print("Tu n'es pas majeur") # block 2
7     print("Tu peux pas passer le permis") # block2
8 else:
9     print("Tu peux passer le permis") # block3
10
11 print("J'ai plus rien à dire")
```

Remarques :

- Seul le mot clé `if` est obligatoire.
- Il peut y avoir autant de `elif` que vous souhaitez
- À la première condition validée, uniquement le block indenté en dessous sera exécuté.
- Les conditions sont de type booléen (vrai ou faux). Attention, Python donne parfois des valeurs booléennes à d'autres types.

### Principaux opérateurs sur les `int` ou `float`

```
x == y ...  
x != y ...  
x < y ...  
x > y ...  
x <= y ...  
x >= y ...
```

### Principaux opérateur sur les `bool`

```
E and F ...  
E or F ...  
not E ...
```

### Principaux opérateur `str`

```
c in S ...
```

#### À faire au crayon à papier

Décrire la comparaison effectuée par chaque opérateur

### Exemples

```
1 >>> 3 < 5  
2 ...  
3 >>> (3 < 5) and (5 == 2*2+1)  
4 ...  
5 >>> not (5 <= 5)  
6 ...  
7 >>> "a" in "uiopup"  
8 ...
```

#### À faire au crayon à papier

Compléter l'exemple précédent avec True ou False