

# Quelques instructions générales du langage python

## Les variables

### Pour stocker une valeur dans une variable

```
nom_variable = valeur
```

Remarque, la valeur peut-être un nombre ou un texte.

```
texte_1 = "Bonjour"
```

```
nombre_1 = 45
```

### Pour utiliser une variable

```
nombre_1 = 8
```

```
nombre_2 = 15
```

```
nombre_3 = nombre_1 + nombre_2 # nombre_3 contient maintenant la valeur 23
```

Remarque : on peut augmenter la valeur d'une variable de 1 de la façon suivante :

```
compteur = compteur + 1
```

## Les conditions

L'instruction `if` permet d'exécuter un ensemble d'instructions uniquement si une condition est vérifiée et un autre ensemble d'instruction si la condition n'est pas vérifiée.

### Exemple avec `if seul`

```
nombre = 0
if nombre < 5:
    nombre = nombre + 1
    display.show(nombre)
```

### Exemple avec `if et else`

```
nombre = 0
if nombre < 5:
    nombre = nombre + 1
else:
    display.show(Image.HAPPY)
```

## Les boucles

Pour répéter un ensemble d'instructions un nombre donné de fois :

```
for i in range(20):
    # instructions à répéter
```

Pour répéter un ensemble d'instruction indéfiniment

```
while True:
    # instructions à répéter
```

## Obtenir un nombre aléatoire

Il faut ajouter la ligne suivante en début de programme :

```
from random import *
```

La fonction `randint` permet d'obtenir un nombre aléatoire. Elle s'utilise de deux façons :

```
nb_aleat_1 = randint(6) # permet d'obtenir un nombre aléatoire entre 0 à 6
nb_aleat_2 = randint(1, 6) # permet d'obtenir un nombre aléatoire de 1 à 6
```

# Quelques instructions spécifiques à la carte micro:bit

## Première ligne de code à mettre tout le temps

Cette ligne de code permet l'utilisation des instructions spécifiques à la carte micro:bit

```
| from microbit import *
```

## Utiliser les leds

### Afficher une image préexistante

```
| display.show(Image.HAPPY)
```

### Eteindre toutes les led

```
| display.clear()
```

### Afficher un texte ou un nombre

```
| display.show("Bonjour")
```

### Allumer ou éteindre 1 pixel

Pour allumer le pixel de coordonnées x (entre 0 et 4) et y (entre 0 et 4) avec une intensité égale à val (valeur entre 0 et 9) :

```
| display.set_pixel(x, y, val)
```

## Gestion du temps

### Mettre le programme en pause

```
| sleep(500) # la valeur est en millisecondes
```

### Exécuter des instructions en boucle

```
| while True:
|     display.show(Image.HAPPY)
|     sleep(500)
|     display.clear()
|     sleep(500)
```

## Utilisation des boutons

Le code `button_a.was_pressed()` et `button_a.is_pressed()` peuvent être utilisés dans les conditions.

```
| from microbit import *
| while True:
|     if button_a.is_pressed():
|         display.show(Image.HAPPY)
|     else:
|         display.show(Image.SAD)
```

## Utilisation du capteur de lumière

Le code `display.read_light_level()` permet d'obtenir la luminosité captée par les leds (entier entre 0 et 255).

```
| while True:
|     if display.read_light_level() > 80:
|         display.show(Image.HAPPY)
|     else:
|         display.show(Image.SAD)
```