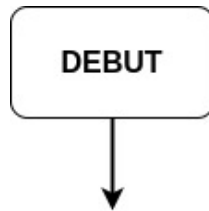


cheatsheet

March 14, 2023

1 Cheat sheet : Formalisme algorigrammes - language Python

1.1 Début de programme ou de fonction

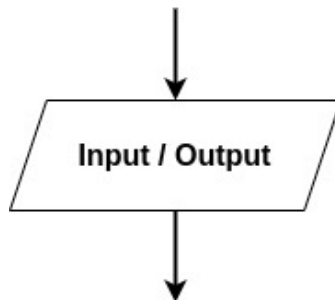


Début de **programme** : c'est la première ligne (numéro 0)

Début de **fonction** :

```
def maFonction(argument1, argument2):
```

1.2 Input/Output



Entrée:

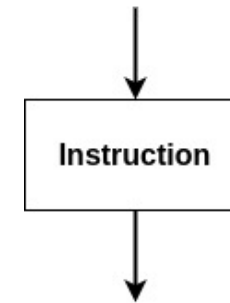
```
maVariable = input("Entrez une valeur : ")
```

le **type** de `maVariable` est `str` (une chaîne de caractères)

Sortie:

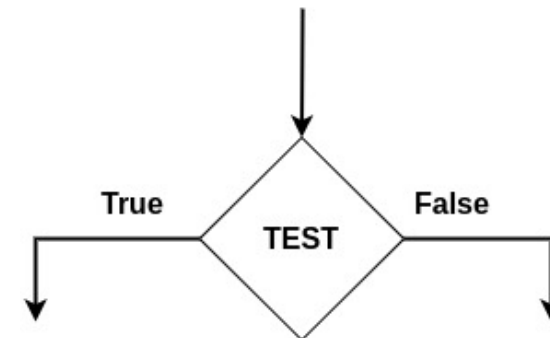
```
print("Hello world")
```

1.3 Instruction



Une instruction est une ligne du programme ou de la fonction qui n'est pas un **commentaire** (commençant par le signe #) ni une ligne blanche

1.4 Test conditionnel



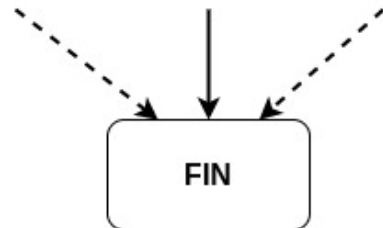
Le test conditionnel s'utilise avec la structure `if` (obligatoire), `elif` (facultatif) et `else` (facultatif).

Exemple : la variable `x` est-elle plus grande que 4 ?

```
if x > 4 :  
    print("x est plus grand que 4")
```

```
else:
    print("x est plus petit ou égal à 4")
```

1.5 Fin de programme ou de fonction



Fin de **programme** : c'est la dernière ligne du programme

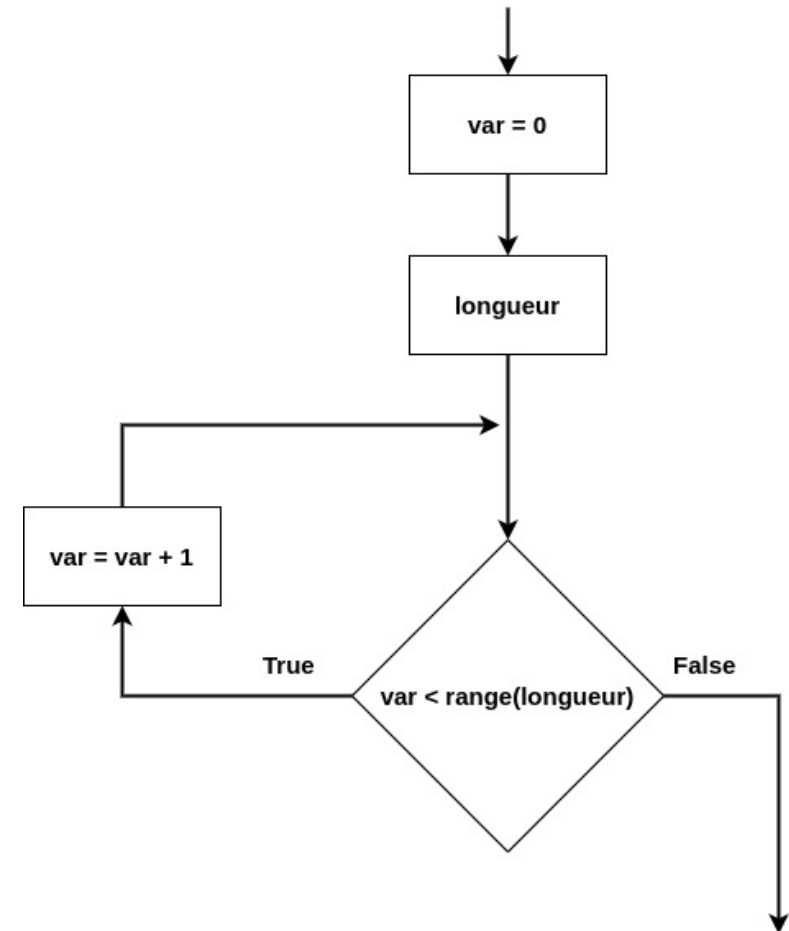
Fin de **fonction** : c'est lorsque l'indentation (les 4 caractères ESPACE ou TABULATION) revient le plus à gauche

Exemple :

```
def maFonction(argument1, argument2):
    instruction1
    instruction2
# La fin de la fonction est atteinte ici, le programme principal continue
instruction3
```

1.6 Boucles

La boucle s'exprime sous la forme d'un test conditionnel sur la valeur de la variable *incrément* :



En Python on utilise la fonction `range(borneInf, borneSup, increment)` (où `borneInf` facultatif est la borne inférieure à partir de laquelle on compte, `borneSup` obligatoire est la borne supérieure **non comprise** jusqu'où on compte et `increment` facultatif le pas de comptage).

1.6.1 Exemple 1

On veut compter de 0 à 6 (non compris)

```
[ ]: for var in range(6):  
    print(var)
```

```
0  
1  
2  
3  
4  
5
```

1.6.2 Exemple 2

On veut compter de 4 à 10 (non compris)

```
[ ]: for var in range(4,10):  
    print(var)
```

```
4  
5  
6  
7  
8  
9
```

1.6.3 Exemple 3

On veut compter de 7 à 17 (compris) avec un pas de 2, donc seulement les nombres impairs :

```
[ ]: for var in range(7,18,2):  
    print(var)
```

```
7  
9  
11  
13  
15  
17
```

```
[ ]:
```