



# INSAlgo - Cours 1

...

Sujet du cours : Python, les bases de la base, etc.

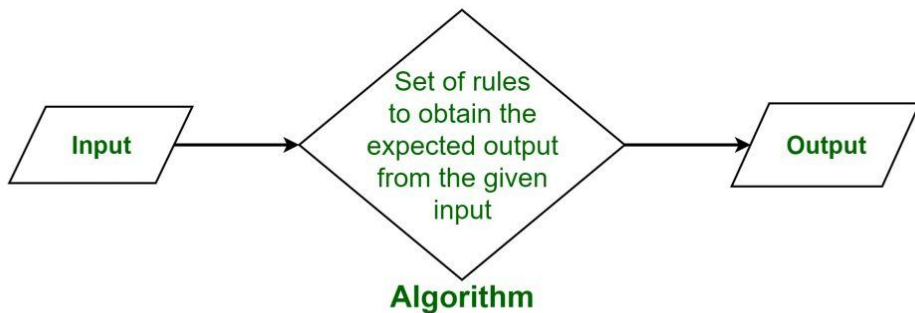
- commencez les exos si vous maîtrisez déjà -

Cours en Python car la syntaxe est simple et compréhensible pour les débutants ! ٩(●~●)۶

# Un algorithme, c'est quoi ?



## What is Algorithm?



## Exemple:

Problème: Faire la somme de deux nombres

Input: 2 et 3

**Algo: 2+3**

Output: 6

# Les variables (la base de la base) : **int**, **float**, **bool**, **str**



```
a = 3.3 # On met la valeur 3 dans la variable appelée a
b = int(3.3) # ici on aura b = 3
```

```
pi = 3.14 # nombre à virgule (float)
prenom = "Sebasbas" # chaîne de caractères (String)
on_a_faim = True # booléen (vrai ou faux)
```

```
a = 1
b = 2
laSomme = a + b
```

# Conversions



On peut convertir entre plusieurs types facilement. Par exemple :

```
int('123')  
float('456.0')  
float(123)           # => 123.0  
int('0xf3c56', 16)   # 998486  
int('1011010', 2)    # 90  
str([1,2,3,4])        # => '[1,2,3,4]'  
list('abcdef')        # => ['a','b','c','d','e','f']
```

# Input / Output



```
a = input ()
```

```
[blablabla algo]  
answer = a +4 - 8 * 4864864645
```

```
print(answer)
```

# Plus d'exemples de input



INPUT:

4

a

b

c

d

```
n = int(input())
```

```
l = []
```

```
for i in range(n):
```

```
    l.append(input())
```

INPUT:

4

a b c d

```
n = int(input())
```

```
l = input().split(" ")
```

```
l = ['a','b','c','d']
```

# Quelques opérations élémentaires (voir CheatSheet)



```
a = 2
b = 5

c = a + b # c = 7
d = b - a # d = 3
e = a * b # e = 10
f = b / a # f = 2.5
g = b // a # g = 2 (le quotient de b/a)
h = b % a # h = 1 (le reste de b/a)
b += a # équivalent à b = b + a (marche aussi avec -=)
```

# Les conditions



Les conditions “if” se font comme ceci:

```
if (s == 'hacked' or and t>1):  
    hackStatus = True  
    print('hack successful')  
else:  
    hackStatus = false  
print('done')
```

```
if (a%2 == 0):  
    print('a est un nombre pair')  
else:  
    print('a est un impairrrrr')
```

Attention : L'indentation est extrêmement importante en python. Il n'y a pas d'accolades comme dans la plupart des langages, donc l'interpréteur détermine les blocs en fonction de leur indentation. Il faut aussi avoir une consistance au sein d'un fichier (ne pas mélanger les espaces et les tabulations). Il est recommandé d'utiliser 4 espaces ou des tabs



# Les conditions



On peut aussi chaîner les tests logiques :

```
if (5 < timeValue <= 10 and not kill) or (hackedStatus==true):  
    print('hack successful')
```

Attention, on écrit bien **and**, **or** ou **not** et pas &&, || ou !

# Les boucles

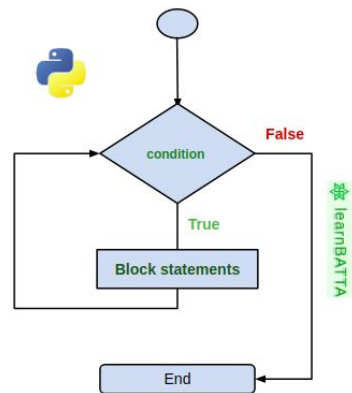


On reste dans la boucle tant que la condition n'est pas remplie et continue les opérations:

```
while x < 100:  
    x += 1  
    if y == 2:  
        break
```

```
for i in range(0,10):  
  
    print i * 2
```

*# for (int i = 0;  
i < 10; i++)  
en C++/Java*



working with while loop in python

# Les listes



```
ListVide = []  
myList = [2,3,1,2,5,6,1,4,1,2]  
print(myList) # >>> {2,3,1,2,5,6,1,4,1,2}
```

```
myList.append(3)  
print(myList) # >>> {2,3,1,2,5,6,1,4,1,2,3}
```

```
myList[1] = 2  
print(myList) # >>> {2,2,1,2,5,6,1,4,1,2,3}
```

```
print(len(myList)) # >>> 11
```

# Pour parcourir une liste!

La base (tab est une liste btw) :

```
for i in range( len(tab) ):  
    print( tab[i] )
```

OU

```
for val in tab:  
    print(val)
```

# Les fonctions: (peut être pas nécessaire pour les 1ers exos mais au cas où!)



Exemple de déclaration et utilisation de fonction :

```
def hello(myArg) :  
    return 'Hello ' + myArg + ' !'  
  
print(hello('World'))
```

Comme vous commencez à le remarquer, il n'y a pas vraiment de type en python. On ne déclare donc pas de type d'arguments ou de retour (c'est possible mais rarement utilisé, et ça ne sert à rien en algo).

# Passons au exos ?

>> Go faire une démonstration de la plateforme



## A retenir:

```
input() - split() - print()
if (...) : else :
while(...):
for (i in range(N))
tab.append() ...
```

Remarque : ce cours est très minimal juste pour découvrir les éléments nécessaires pour jouer avec des algorithmes!

Si vous êtes **chauds**, il y a beaucoup d'autres choses sur Python à découvrir! (dict, sets, library, split, pop...)

# Installation de Python 3 (si on a le temps?)



## Windows :

Installez Anaconda, qui inclus Python et plein de bibliothèques pratiques :

- <https://www.anaconda.com/download/>

Pour l'éditeur, nous vous conseillons PyCharm :

- <https://www.jetbrains.com/toolbox/app/>
- Nos mail @insa nous permettent de télécharger la version Pro, même si la version Community est suffisante

## Linux :

Bien qu'Anaconda existe aussi pour Linux, la présence de gestionnaires de paquets baisse son intérêt pour notre utilisation. Comme Python est déjà installé sur la plupart des distributions, vous n'avez rien à faire !

Pour l'éditeur, nous conseillons PyCharm :

- <https://www.jetbrains.com/toolbox/app/>