

Python : Les boucles

I) La fonction range

Définition :

La fonction **range** est la création d'une liste où l'on donne le **premier nombre inclus, dernier nombre non inclus, ainsi que le pas** écrits de la façon suivante : **range (premier nombre inclus, dernier nombre non inclus, le pas).**

Exemples :

```
a=range(2,10,3)
```

Le pas est 3 donc on va de 3 en 3 en commençant par le premier nombre 2 en s'arrêtant à 10 non compris.

```
a = range(2,10,3)
```

a=[2 ,5,8] on s'arrête à 8 car 8+3= 11 on dépasse 10

```
b= range(0,10,2)
```

Le pas est 2, on va de 2 en 2 en partant du premier nombre 0 et on s'arrête à 10 non compris :

```
b=range(0,10,2)
```

```
b=[0,2,4,6,8]
```

On peut écrire aussi **range (premier nombre, deuxième nombre non inclus)** dans ce cas le premier nombre est le premier de la liste inclus, le dernier est le dernier de la liste non inclus, le pas n'étant pas précisé il sera automatiquement de 1.

Exemple :

```
c= range (3,6) alors c= [3, 4, 5] le pas sera de 1 automatiquement
```

```
d= range(7,12) alors d=[7,8,9,10,11]
```

On peut écrire aussi **range (nombre)**, ce nombre est alors le dernier nombre non inclus, le premier étant automatiquement 0 et le pas est 1.

```
e=range (6) alors e= [0,1,2,3,4,5], s'il n'y a qu'un seul nombre alors par défaut le premier nombre est , la pas est et on va jusqu'à non compris.
```

```
f=range(9) alors f=[0,1,2,3,4,5,6,7,8]
```

Cette fonction est beaucoup utilisée dans les boucles un des chapitres suivants.

Exemples :

a=range(4) , il n'y a qu'un seul nombre donc c'est forcément le dernier nombre non inclus le pas est 1 et le premier nombre 0 donc **a = [0,1,2,3]**

b = range(6,9) , il y a deux nombres donc le pas est 1 le premier nombre 6 et on s'arrête à 9 non inclus donc **b = [6,7,8]**

c = range(6,12,3) dans ce cas le premier nombre est 6 le dernier non inclus est 12 et le pas est 3 donc **c=[6,9]** **12 ne peut être pris.**

II) Les boucles

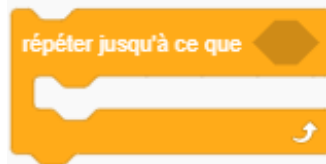
Au collège, lorsque l'on veut répéter une ou plusieurs(s) instruction(s) un certain nombre de fois, ou indéfiniment on utilisait les blocs :



Il suffit de compléter le nombre de fois où la boucle est répétée.



Le même bloc d'instruction(s) est répété indéfiniment.



Le même bloc d'instruction(s) est répété jusqu'à ce qu'une condition soit réalisée.

1) La boucle « Pour »

La boucle « pour » c'est-à-dire « for » en Python est une boucle bornée.

En algorithmie : on écrit

en Python on écrit :

Pour i allant de 0 à N :

for i in range(N) :

Séquence d'instructions

Séquence d'instructions

Fin de Pour

La variable i prend les valeurs allant de 0 à N-1

Attention , lorsque l'on écrit l'algorithme la fin de la boucle est notée par « Fin du pour ».

En Python à la suite de l'instruction : « for i in range(N) : » on passe à la ligne avec une indentation qui se termine dès la séquence d'instructions terminée. La fin de l'indentation marque la fin de la boucle.

Exemple :

```
def somme () :  
    S= 0  
    for i in range(1,6) :  
        S = S+i  
    Return S
```

Etape 0 : S = 0

Etape 1 : S = 0+1 = 1

Etape 2 : S = 1+2 = 3

Etape 2 : S = 3+3 = 6

Etape 3 : S = 6+4 = 10

Etape 4 : S = 10 + 5 = 15

**i va prendre successivement les valeurs
1 ; 2 ; 3 ; 4 et 5**

2) La boucle « Tant que ... »

La boucle « Tant que » c'est-à-dire « **while** » en Python est une boucle non bornée. Tant que la condition est vraie, la séquence d'instructions est exécutée.

En algorithmie : on écrit

tant que la condition est vraie :

Séquence d'instructions

Fin de Tant que

en Python on écrit :

while condition :

Séquence d'instructions

Attention , lorsque l'on écrit l'algorithme la fin de la boucle est notée par « Fin du tant que ».

En Python à la suite de l'instruction : « while suivi de la condition : » on passe à la ligne avec une indentation qui se termine dès la séquence d'instructions terminée. La fin de l'indentation marque la fin de la boucle.

Exemple :

```
>>> n=0
>>> i=0
>>> while n<10:
...     i=i+1
...     n=n+3*i
...
>>> print(n)
18
```

Etape 0, $n=0$ et $i=0$

Etape 1 : $i=0+1 = 1$ et $n=0+3\times 1=3$

Etape 2 : $i=1+1=2$ et $n = 3+3\times 2 = 9$ n reste inférieur à 10 donc la boucle continue

Etape 3 : $i =2+1 = 3$ et $n=9+3\times 3=18$

n est plus grand que 10, la boucle s'arrête, la dernière valeur de n est donc 18.