



## Récurtivité

### Définition :

On appelle récursive toute fonction ou procédure qui s'appelle elle-même.

Examinons deux fonctions récursives

```
1 def fonction(n):
2     if n>0:
3         print(n)
4
5         fonction(n-1)
6
7 def fonction_bis(n):
8     if n>0:
9
10        fonction_bis(n-1)
11    print(n)
12
```

Quels sont les affichages produits par ces deux fonctions avec  $n = 5$  ?

Testez ces codes dans une console.

Pour *fonction* : 5 s'affiche

On appelle fonction(4) -> 4 s'affiche  
On appelle fonction(3) : -> 3 s'affiche  
On appelle fonction(2) : -> 2 s'affiche  
On appelle fonction(1) : -> 1 s'affiche.

Pour *fonction\_bis* :

On appelle fonction(4)  
On appelle fonction\_bis(3)  
On appelle fonction\_bis(2)  
On appelle fonction\_bis(1).

A ce stade, aucune fonction\_bis n'est exécutée. On a, stocké en mémoire, une pile de fonctions à exécuter.

				fonction_bis(1)
			fonction_bis(2)	fonction_bis(2)
		fonction_bis(3)	fonction_bis(3)	fonction_bis(3)
	fonction_bis(4)	fonction_bis(4)	fonction_bis(4)	fonction_bis(4)
fonction_bis(5)	fonction_bis(5)	fonction_bis(5)	fonction_bis(5)	fonction_bis(5)

Arrivé à n= 1, le if n'est plus exécuté et l'on sort de la condition et arrive sur print(n), 0 est affiché . A ce stade, c'est fonction\_bis(1) qui est exécutée (sur le dessus de la pile) .Et donc 1 s'affiche, puis 2,3,4,5.

```
In [13]: fonction_bis(5)
0
1
2
3
4
5
```

Terminaison d'une fonction récursive

Le problème que l'on peut rencontrer avec des fonctions qui s'appellent elles même st que l'exécution du programme ne se termine jamais .

```
1 def fonction():
2     print ('ça finit quand ?')
3     fonction()
4
```

```
RecursionError: maximum recursion depth exceeded while calling a Python object
```

**Il est nécessaire lorsque le travaille avec la récursivité de prévoir un cas de base qui permette de sortir du programme.**

Exemple : Factoriel(n).

Rappel : On appelle factoriel(n) la fonction qui à tout entier n positif associe le produit des n premiers entiers strictement positifs et 1 si n =0

$$\begin{cases} fact(n) = \prod_{i=1}^n i = n * (n - 1) \dots 2 * 1 \\ fact(n) = 1 \text{ si } n = 0 \end{cases}$$

Factoriel n se note n ! en mathématiques.

Exemple 4 ! = 4\*3\*2\*1 = 24

Voici un algorithme traitant du factoriel.

```
##### ALGORITHME factoriel n #####  
On demande en entrée un entier n positif ou nul
```

```
c ← 1  
fact ← 1  
tant que c < n  
    c ← c + 1  
    fact ← fact × c  
fin tant que
```

**Ecrire un programme récursif qui prenne en paramètre un entier positif n et renvoie n!**