Chapitre 2

La récursivité

2.1 Les piles d'exécution

NSI Tle - JB Duthoit

Une poupée russe, c'est une poupée avec une poupée russe à l'intérieur !

2.1.1 Analyse d'un programme

Analyser et tester ce programme

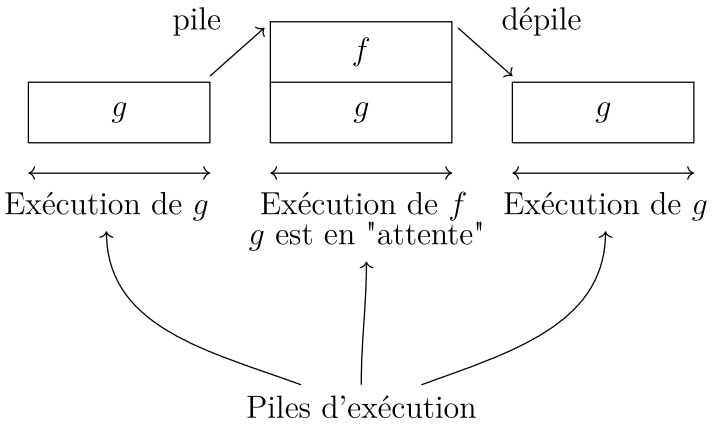
```
def f():
    for i in range(1,6):
        print(f"{i} affiché par la fonction f")

def g():
    for i in range(1,6):
        print(f"{i} affiché par la fonction g")
        if i == 3:
            f()
g()
```

Dans l'exemple ci-dessus, nous avons une fonction `g` qui appelle une autre fonction `f`. La principale chose à retenir de cet exemple est que l'exécution de `g` est interrompue pendant l'exécution de `f`.
Une fois l'exécution de `f` terminée, l'exécution de `g` reprendra là où elle avait été interrompue.

☛ Analyse ce programme dans Python Tutor.

2.1.2 Les piles d'exécution

Pour gérer ces fonctions qui appellent d'autres fonctions, le système utilise une ***"pile d'exécution"***.
Une pile d'exécution permet d'enregistrer des informations sur les fonctions en cours d'exécution dans un programme.
On parle de ***pile***, car les exécutions successives "s'empilent" les unes sur les autres. Si nous nous intéressons à la pile d'exécution du programme étudié ci-dessus, nous obtenons le schéma suivant :



Lorsque l'on fait fonctionner un programme récursif, il utilise forcément des piles d'exécution.