

# I.S.N. Fiche n°1 – Récursivité

Le contenu de cette fiche est inspiré du livre suivant : Dowek et coauteurs, Informatique et sciences du numérique, page 80 du livre, c.-à-d. page 93 du fichier. Lien ici, 22.78 Mio

## 1 Le principe de la récursivité

### 1.1 Une fonction qui appelle une autre fonction

Tout d'abord, on rappelle qu'une fonction peut appeler une autre fonction, comme dans l'exemple ci-dessous :

```
def afficherLigne():
    print("-----")

def afficherTitre():
    afficherLigne()
    print("  Mon titre")
    afficherLigne()
```

Ce qui donne dans la console :

```
>>> afficherTitre()
-----
  Mon titre
-----
```

### 1.2 Une fonction qui s'appelle elle-même

En fait, il est également possible d'appeler une fonction depuis le corps de la fonction elle-même. Ce principe s'appelle la récursivité. Voici un exemple :

```
def calculerPuissanceDeDeux(exposant):
    if exposant == 0:
        return 1
    else:
        return ( 2 * calculerPuissanceDeDeux(exposant - 1) )
```

Ce qui donne dans la console :

```
>>> calculerPuissanceDeDeux(5)
32
>>> calculerPuissanceDeDeux(0)
1
>>> calculerPuissanceDeDeux(1)
2
```

Il faut toujours faire attention à la *terminaison* de notre algorithme, autrement dit, que la fonction ne s'appelle pas indéfiniment. Pour cela :

- nous avons prévu un *cas de base*, ici `exposant == 0`, dans lequel la fonction ne s'appelle pas elle-même.
- chaque appel de fonction se fait avec une valeur de l'argument qui est un entier diminué de 1 à chaque fois. Donc à la fin, le dernier appel se fait avec un argument de 0, ce qui est le *cas de base*.

### 1.3 Exercice à faire : la fonction factorielle

Pour tout entier naturel non nul  $n$ , on pose  $n! = 1 \times 2 \times \dots \times n$ . On pose aussi  $0! = 1$ . On dit que  $n!$  est la factorielle de  $n$ .

Écrire une fonction récursive qui calcule la factorielle d'un nombre entier naturel  $n$ .

## 1.4 Des exemples (prolongement)

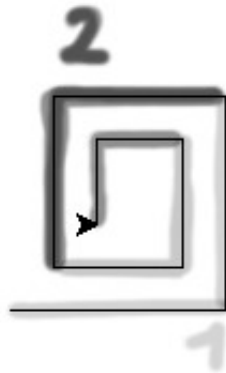
- la suite de Fibonacci (Voir livre de Dowek p. 84 consigne + « Aller plus loin »),
- coefficients binomiaux, triangle de Pascal,
- fonction récursive de la conjecture de Syracuse (conjecture de Collatz),
- tours de Hanoï.

## 2 Des images récursives

À l'aide de la tortue python, on peut construire des images récursives.

### 2.1 Exercice à faire : une spirale

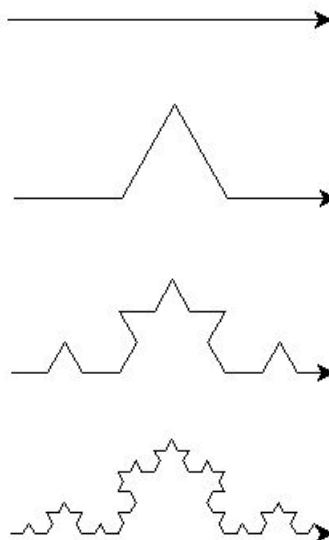
On peut construire une spirale comme ci-dessous. Chaque étape est constituée de deux segments de même longueur. A l'étape suivante, les segments ont une longueur plus courte de 10 pixels par exemple. Écrire une fonction récursive qui permet de construire une telle spirale.



### 2.2 Un exemple : la courbe de Von Koch

Au départ, on considère un segment.

A chaque étape, au lieu de tracer le segment, on trace 4 segments de longueurs trois fois plus courte, les deux segments au centre étant les côtés d'un triangle équilatéral.



### 2.3 Autres exemples (prolongement)

On pourra aussi chercher sur le Web :

- flocon de Von Koch,
- arbre fractal,
- arbre de pythagore,
- triangle de Sierpinski, tapis de Sierpinski,
- courbe de Hilbert, courbe du dragon, L-système.