

# TP flocon de von Koch

Christophe Viroulaud

Terminale - NSI

**Lang 06**

Le flocon de von Koch est l'une des premières courbes fractales à avoir été décrites (bien avant l'invention du terme *fractal*). Elle a été inventée en 1904 par le mathématicien suédois Helge von Koch.

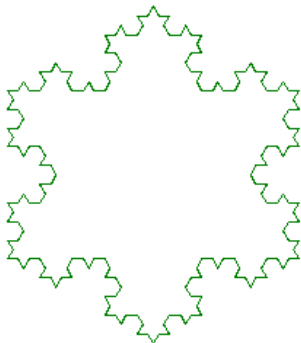


FIGURE 1 – flocon de von Koch

Bibliothèque  
turtle : rappels

Flocon de von  
Koch

Principe  
Construction récursive  
Construction du flocon

L'objectif de cette séance est de créer un programme récursif permettant de dessiner un flocon de von Koch

Bibliothèque  
**turtle** : rappels

Flocon de von  
Koch

Principe  
Construction récursive  
Construction du flocon

1. Bibliothèque **turtle** : rappels

2. Flocon de von Koch

La bibliothèque `turtle` permet de dessiner des figures géométriques simplement. La documentation se trouve ici :

[https:  
//docs.python.org/fr/3.8/library/turtle.html](https://docs.python.org/fr/3.8/library/turtle.html)

**Activité 1 :** Créer un fichier `spirale.py`, importer la bibliothèque `turtle` et l'initialiser avec les paramètres suivants :

```
1 colormode(255)
2 speed(5)
3 hideturtle()
```

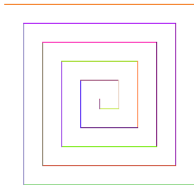


FIGURE 2 – Figure à obtenir à la fin de l'activité

Pour dessiner la figure, le programme respecte l'algorithme :

- ▶ choisir une couleur,
- ▶ tracer un trait de dimension `dim`,
- ▶ tourner de  $90^\circ$ ,
- ▶ répéter l'opération en diminuant la dimension de 10.

## Activité 2 :

1. Écrire la fonction `creer_couleur()`  $\rightarrow$  tuple qui renvoie un tuple `(r, g, b)` de trois entiers aléatoires compris entre 0 et 255.
2. Écrire la fonction récursive `dessiner(dim: int)`  $\rightarrow$  `None` qui respecte l'algorithme.
3. Exécuter la fonction avec une dimension de 200.

# Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.



```
1 def creer_couleur() -> tuple:
2     """
3     renvoie une couleur (r, g, b)
4     """
5     return (randint(0, 255) for _ in range(3))
```

Code 1 – Choix de la couleur

```
1 def dessiner(dim: int) -> None:
2     """
3     dessine une spirale carrée
4     """
5     if dim >= 0:
6         pencolor(creer_couleur())
7         forward(dim)
8         right(90)
9         dessiner(dim-10)
10
11 dessiner(200)
```

Code 2 – Dessiner

Bibliothèque  
turtle : rappels

Flocon de von  
Koch

Principe  
Construction récursive  
Construction du flocon

## 1. Bibliothèque turtle : rappels

## 2. Flocon de von Koch

### 2.1 Principe

### 2.2 Construction récursive

### 2.3 Construction du flocon

# Principe

On peut créer le flocon à partir d'un segment de droite, en le modifiant récursivement selon la précision désirée :

- ▶ Diviser le segment en trois segments de longueurs égales.
- ▶ Tourner de  $60^\circ$ .
- ▶ Recommencer pour chaque segment obtenu.

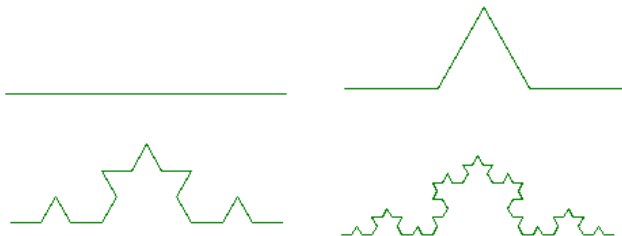


FIGURE 3 – Étapes de construction

Bibliothèque  
turtle : rappels

Flocon de von  
Koch

Principe

**Construction récursive**

Construction du flocon

## 1. Bibliothèque turtle : rappels

## 2. Flocon de von Koch

### 2.1 Principe

### 2.2 Construction récursive

### 2.3 Construction du flocon

## Activité 3 :

1. Observer la figure 3 pour déterminer les étapes récursives.
2. Quelle est l'étape limite (qui stoppe les appels récursif) ?

# Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Chaque trait est partagé en 3 morceaux. En prenant en compte les rotations on obtient 4 traits. Au lieu de tracer un trait, l'étape récursive consiste à partager à nouveau ce trait. L'étape finale consiste à tracer effectivement un trait. Elle est atteinte quand on arrive à la précision désirée.



**Activité 4 :** Écrire la fonction

`courbe_koch(precision: int, mesure: int) →`  
None qui construit un côté du flocon (figure 3).

# Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

```
1 def courbe_koch(precision: int, mesure: int) ->  
    None:  
2     if precision == 0:  
3         forward(mesure)  
4     else:  
5         courbe_koch(precision-1, mesure//3)  
6         left(60)  
7         courbe_koch(precision-1, mesure//3)  
8         right(120)  
9         courbe_koch(precision-1, mesure//3)  
10        left(60)  
11        courbe_koch(precision-1, mesure//3)
```

Code 3 – Tracer une branche

Bibliothèque  
turtle : rappels

Flocon de von  
Koch

Principe  
Construction récursive  
Construction du flocon

## 1. Bibliothèque turtle : rappels

## 2. Flocon de von Koch

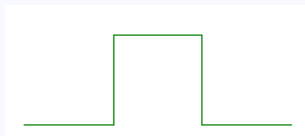
### 2.1 Principe

### 2.2 Construction récursive

### 2.3 Construction du flocon

## Activité 5 :

1. Écrire la fonction `flocon(precision: int, mesure: int) → None` qui utilise la fonction `courbe_koch` pour tracer le flocon.
2. Tester la fonction pour plusieurs précisions.  
Attention le temps de tracé peut être très long pour des valeurs supérieures à 5.
3. **Pour les plus avancés :**
  - ▶ Tracer chaque trait d'une couleur différente.
  - ▶ Construire un flocon avec la courbe suivante :



# Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

```
1 def flocon(precision: int, mesure: int) -> None:
2     for _ in range(3):
3         courbe_koch(precision, mesure)
4         right(120)
5
6
7 flocon(3, 500)
```

Code 4 – Tracé du flocon

```
1 def courbe_koch_carre(precision: int, mesure: int
  ) -> None:
2     if precision == 0:
3         forward(mesure)
4     else:
5         courbe_koch_carre(precision-1, mesure//3)
6         left(90)
7         courbe_koch_carre(precision-1, mesure//3)
8         right(90)
9         courbe_koch_carre(precision-1, mesure//3)
10        right(90)
11        courbe_koch_carre(precision-1, mesure//3)
12        left(90)
13        courbe_koch_carre(precision-1, mesure//3)
```

Code 5 – von Koch quadratique



# Correction

```
1 def flocon_carre(precision: int, mesure: int) ->  
    None:  
2     for _ in range(4):  
3         courbe_koch_carre(precision, mesure)  
4         right(90)
```

