

## Programmation fonctionnelle

informatique / licence 2

*Les exercices qui suivent sont traités en langage Scheme. On utilisera l'interface DrRacket. On changera cette fois le langage utilisé en sélectionnant «Racket» et en vérifiant que l'indication de langage `#lang racket` apparaît bien au début du fichier.*

Tout le travail de la séance repose sur l'utilisation des «tortues» décrites dans le documentation du langage : [https://docs.racket-lang.org/turtles/Traditional\\_Turtles.html](https://docs.racket-lang.org/turtles/Traditional_Turtles.html). Le script doit commencer par importer le module de la façon suivante :

```
(require graphics/turtles)
```

On se servira ensuite essentiellement des commandes :

```
(turtles) ; initialisation de la fenêtre de dessin
(draw x)  ; avance d'une distance x en laissant une trace
(move x)  ; avance d'une distance x sans laisser de trace
(turn x)  ; tourne la tortue d'un angle de x degrés
```

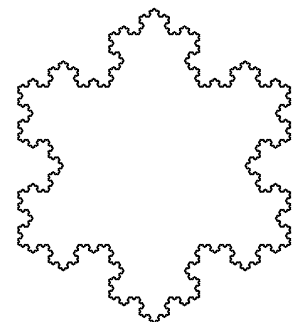
*Tracer les figures décrites ci-dessous en procédant par substitution : une fonction récursive trace un motif simple lors de la condition d'arrêt, tandis que dans le cas général elle s'appelle plusieurs fois sur une taille plus petite.*

1. En remplaçant de façon répétée (et récursive) le tracé d'un segment par le motif ci-dessous à gauche (composé de quatre segments trois fois plus petits), on obtient la courbe de Koch ci-dessous à droite. Le niveau de récursivité peut être choisi arbitrairement.

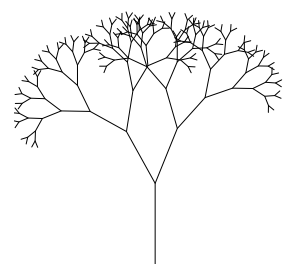


Écrire une fonction en Scheme traçant la courbe de Koch avec un niveau de récursivité arbitraire (le courbe ci-dessus à gauche correspond un niveau de 1 tandis que la version de droite correspond à un niveau un peu plus élevé).

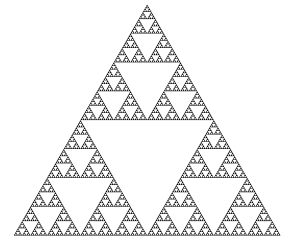
Il est ensuite possible de relier trois motifs de ce genre pour former le fameux flocon de neige. Bien observer la figure ci-contre et repérer où se situent les trois segments. Écrire ensuite une fonction de tracé du flocon de neige.



2. On peut également tracer un arbre en traçant un tronc d'où partent deux branches — chaque nouvelle branche étant à son tour assimilée à un arbre plus petit (les angles et la taille du «tronc» constituant la nouvelle branche devront être choisis de façon à obtenir un rendu esthétique). Une part d'aléatoire peut ensuite être introduite dans le choix des angles et des longueurs, pour gagner en réalisme, en utilisant (`random`) — qui renvoie un flottant entre 0 et 1 — ou (`random x`) — qui renvoie un entier entre 0 et  $x-1$ .



**3.** Le triangle de Sierpiński se trace en substituant trois triangles judicieusement positionnés à la place d'un unique triangle (regarder la figure ci-contre pour bien repérer les trois grands triangles). Le processus se répète de façon récursive. Un niveau arbitraire de récursivité pourra être spécifié par l'utilisateur afin d'augmenter le niveau de détail dans le rendu.



**4.** La courbe du dragon s'obtient elle aussi par substitution récursive d'un motif simple (seulement deux segments) à chaque segment tracé, mais la substitution part alternativement vers la droite et vers la gauche. Voici les premières étapes du tracé :

