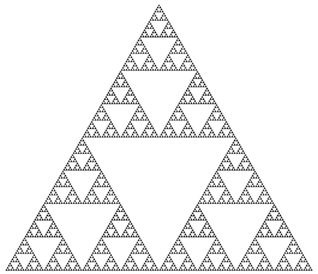
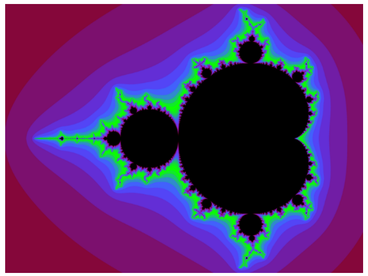
フラクタル図形

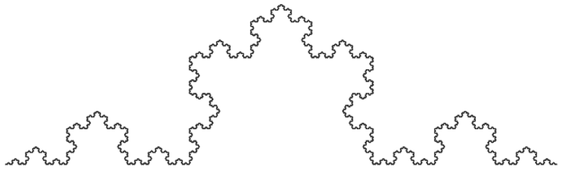
- 図の一部分に、図形全体と相似になっている部分を含むような図形をフラクタル図形と呼ぶ



シェルピンスキーの
ギャスケット



マンデルブロ集合

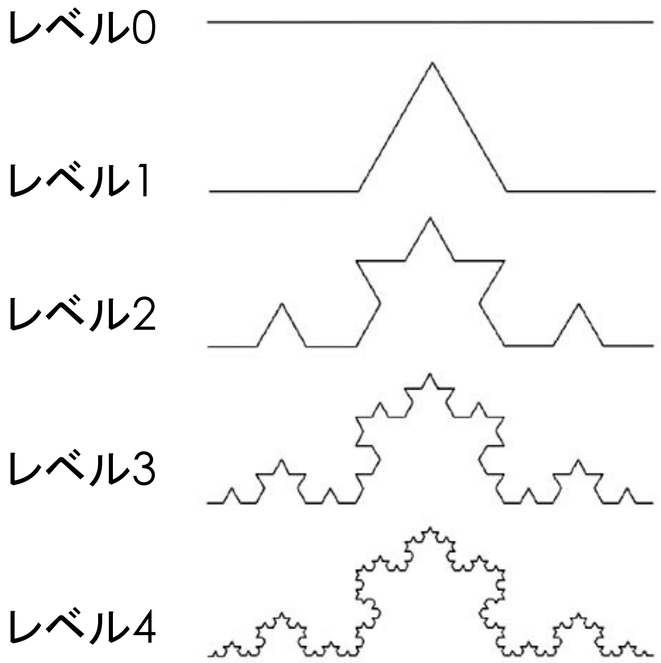


コッホ曲線

- ここでは描きやすさの観点からコッホ曲線に着目する

コッホ曲線について

- レベル0は直線を表す
- レベル1は、直線を3等分し、
  中央を底辺のない
  正三角形にする
  - 60°左回転し、
    3等分した長さだけ進み、
    120°右回転する



- レベル2は、レベル1の処理を、レベル1で得られた4つの直線について繰り返す
- 以下、レベル3,レベル4も同様に繰り返す

（発展）課題4

- コッホ曲線を描く以下の関数を定義しよう
  - koch(length, level)
    - length: コッホ曲線を描く長さを指定する
    - level: コッホ曲線のレベルを指定する

- 再帰呼び出しをうまく使うと便利です
  - 再帰呼び出し：ある関数の中で再びその関数自身を呼び出す
  - 構造のイメージをつかめない場合は、次ページに書くヒントを参考にしてみてください
  - また完成したら、プログラムの処理をステップバイステップで説明してみてください

ヒント

Turtle Graphics Playground

```
1  # Sample Program
2
3  import turtle
4
5  def koch(length, level):
6      if level == 0:
7          turtle.forward(length)
8      else:
9          new_length = length/3
10         koch(
11         turtle.left(
12         koch(
13         turtle.right(
14         koch(
15         turtle.left(
16         koch(
17
18
19 koch(200, 3)
```

levelが0のときには直線を描きます

辺の長さを3等分した値を求めます

亀の向きを工夫しながら
パラメータを工夫して、
自分自身を呼び出します

Run

（発展）課題５

- コッホ雪片
  - 発展課題４で作ったコッホの曲線を利用して、三角形を作ってください
  - レベルを様々に変えて試してください
  - レベルを増やした図形は雪の結晶体のように見えることから「コッホ雪片」と呼ばれています