

# 視覺化程式設計-碎形繪圖專題

## 模組化-遞迴函式 Fractals

### 學習活動 4 : Drawing Fractals by programming

#### 一、實作以下程式，繪製 Koch Curve 第 0~3 步驟圖形

```
1. from turtle import *
2. def f(t, length, depth):
3.     if depth == 0:
4.         t.forward(length)
5.         return
6.     else:
7.         f(t, length/3, depth-1)
8.         t.left(60)
9.         f(t, length/3, depth-1)
10.        t.right(120)
11.        f(t, length/3, depth-1)
12.        t.left(60)
13.        f(t, length/3, depth-1)
14. canvas = Screen()           # Set up the window and its attributes
15. koch = Turtle()             # create koch
16. f(koch, 200, _____)    # Call the function to draw the koch Curve
17. canvas.exitonclick()
```

第 0 步驟





第 1 步驟

第 2 步驟





第 3 步驟

## 二、觀察 Koch Curve

第零步驟	第一步驟	第二步驟	第三步驟
			

```
def f(t, length, depth):
```

起始元： <pre>    if depth == 0:         t.forward(length)</pre>	第零步驟 
生成元： <pre>    else:         f(t, length/3, depth-1)         t.left(60)         f(t, length/3, depth-1)         t.right(120)         f(t, length/3, depth-1)         t.left(60)         f(t, length/3, depth-1)</pre>	第一步驟 

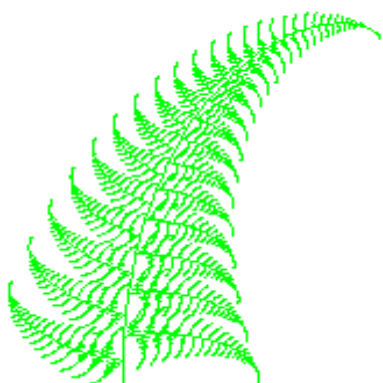
討論：程式呼叫 `f(Koch, 150, 2)` 時，請畫下其 `f` 函式呼叫的歷程

### 三、實作以下程式，繪製蕨葉(Fern)。試修改生長規則另生成不同蕨葉

```
from turtle import *

def fern(t, size):
    if size < 5 :
        return
    t.forward(size/25)
    t.left(80)
    fern(t, size*.3)
    t.right(82)
    t.forward(size/25)
    t.right(80)
    fern(t, size*.3)
    t.left(78)
    fern(t, size*.9)
    t.left(2)
    t.back(size/25)
    t.left(2)
    t.back(size/25)

canvas = Screen()          # Set up the window and its attributes
mytree = Turtle()         # create mytree
mytree.left(90)
fern(mytree, 305)         # Call the function to draw the fern
canvas.exitonclick()
```



修改生長規則另生成不同蕨葉：

#### 四、實作以下程式，繪製一棵樹(tree)。試修改生長規則另生成不同形態的樹

```
1. from turtle import *
2.
3. def tree(t, order, len):
4.     if order == 1 :
5.         t.forward(len)
6.         t.left(30)
7.         t.forward(len)
8.         t.backward(len)
9.         t.right(30)
10.        t.right(30)
11.        t.forward(len)
12.        t.backward(len)
13.        t.left(30)
14.        t.backward(len)
15.        return
16.    else :
17.        t.forward(len)
18.        t.left(30)
19.        tree(t, order-1, len*.75)
20.        t.right(30)
21.        t.right(45)
22.        tree(t, order-1, len*.75)
23.        t.left(45)
24.        t.backward(len)
25.
26. canvas = Screen()           # Set up the window and its attributes
27. mt= Turtle()               # create mytree
28. mt.left(90)
29. mt.hideturtle()            # hide the turtle
30. tree(mt, 9, 50)            # Call the function to draw the tree
31. canvas.exitonclick()
```

[1] 執行結果：



[2] 試修改生長規則另生成不同形態的樹

tree 12 100

change "LT" "RT"

change LT angle and RT angle

change \_\_\_\_\_

## 五、建構碎形圖規則，以程式繪製碎形圖

### [1] 碎形繪製規則

起始元	生成元

### [2] 碎形函式

起始元	生成元

### [3] 依碎形繪製規則疊代建構至第 4 代碎形

第零步驟	第一步驟	第二步驟	第三步驟

### [4] 碎形程式碼