Drawing Projects

On Hour of Code / Scratch / Python turtle

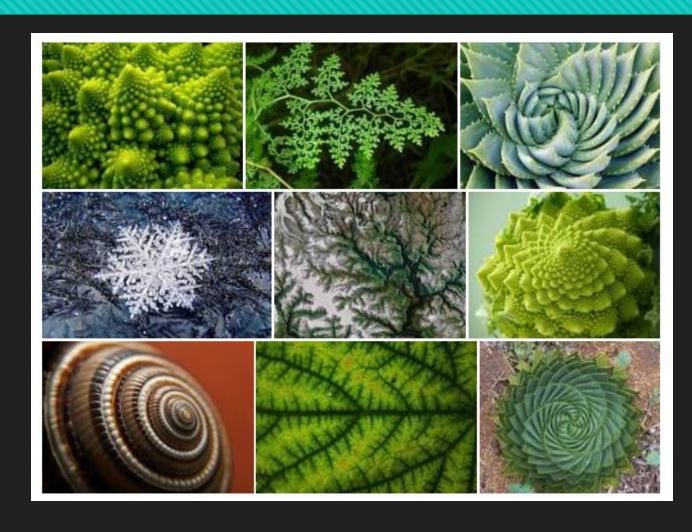
課程實施:8-10年級學生

課程時數:6-8小時

學習單元:模組化程式設計、遞迴函式

碎形~尋找大自然的密碼 Fractals – code in nature

大自然界存在的 神秘圖形規律



專題簡述

- ○本專題引導學生觀察多種碎形圖形的產生規則(rules),學習尋找圖形樣式 (finding patterns)。透過對碎形(fractals)的自我相似性與建構步驟的認識, 鼓勵學生自行創造圖形規則,設計屬於自己的碎形。
- 專題重點為重複結構與與函式的學習。活動中透過重複樣式的辨識與碎形 圖形的建構,讓學生體驗分解問題、樣式辨識、模型化與抽象化等運算思 維歷程,處理複雜而龐大的任務。
- Concepts covered:數學、幾何、碎形、藝術、設計、自我相似性、程式設計、迴圈、疊代、函式、遞迴函式、運算思維

<第一堂> Fractals & patterns

學習活動 1:尋找大自然的密碼 code in nature

Amazing fractals found in nature



Photo: <u>Jitze</u>/Flickr



Photo: Rum Bucolic Ape/Flickr



Photo: Aidan M. Grey/Flickr

Amazing fractals found in nature







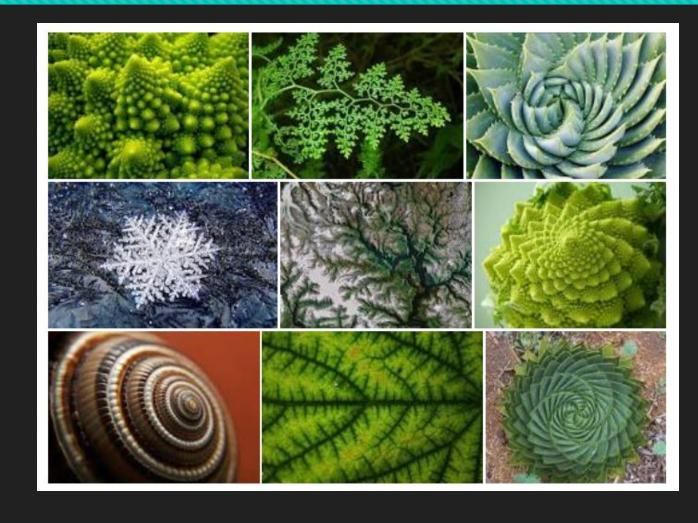
Photo: Wikimedia Commons

Photo: Wikimedia Commons

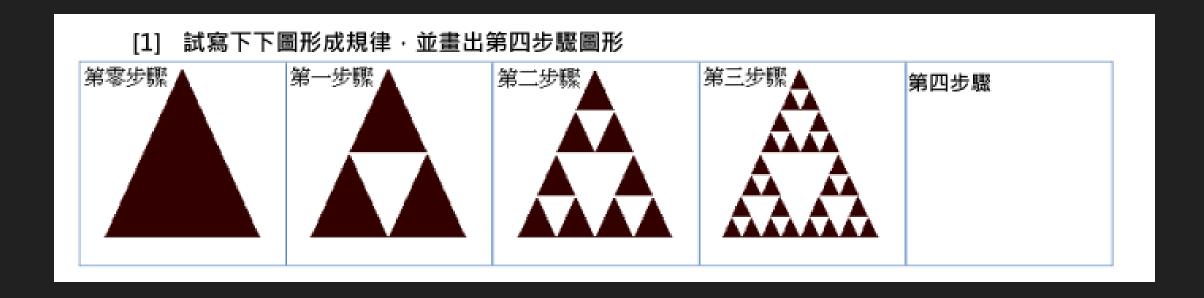
Photo: Burroblando/Flickr

分組活動:尋找大自然界存在的神秘圖形規律

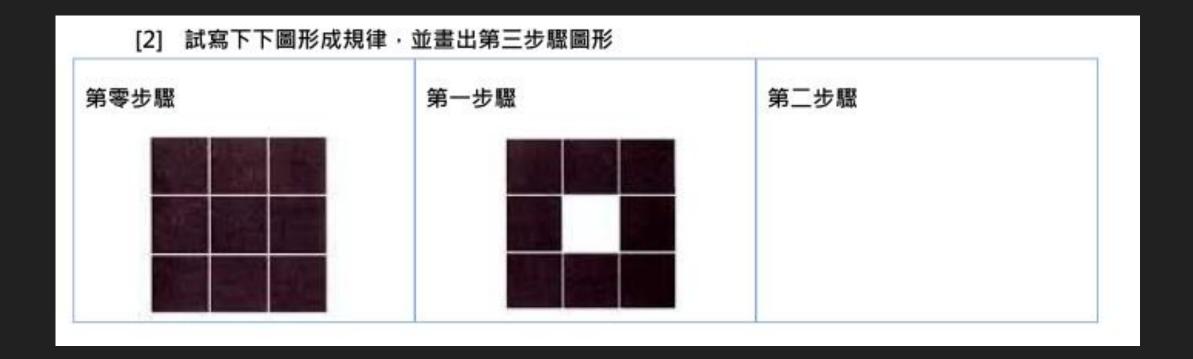
- ○請討論以上圖形的特徵與規律 (至少寫3個)。
- 請整理寫下各組分享的圖形特 徵與規律



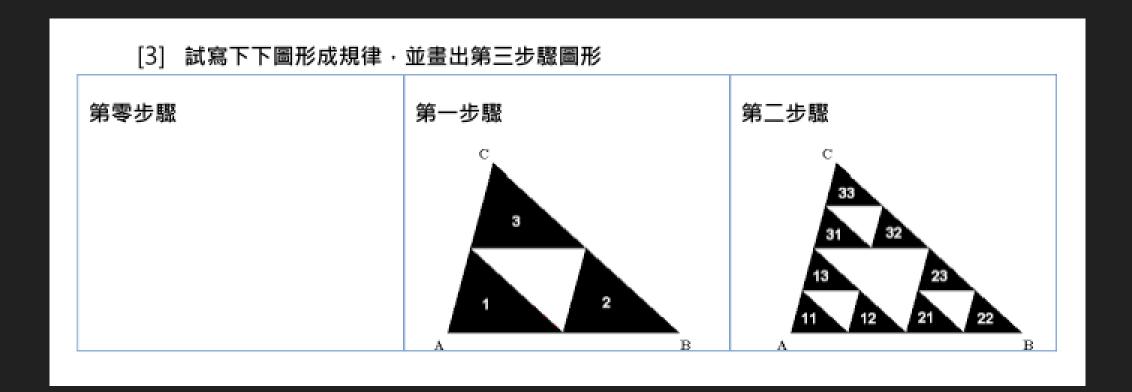
分組活動: 討論碎形的樣式(patterns)與形成規律(modeling)



分組活動: 討論碎形的樣式(patterns)與形成規律(modeling)



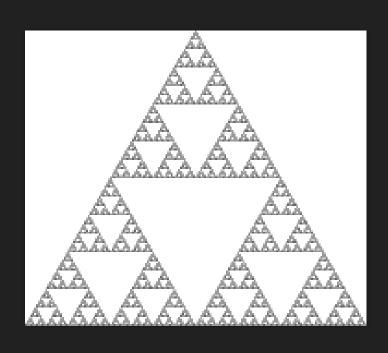
分組活動: 討論碎形的樣式(patterns)與形成規律(modeling)



Fractals 碎形幾何

What is fractal?

○ 碎形 (Fractal)被定義為「一個零碎的幾何形狀,可以分成數個部分,且每一部分都 (至少近似地)是整體縮小後的形狀」,即具有自我相似的性質。





碎形 (Fractal)

- 數學家**曼德布洛特(Benoit** Mandelbrot)在七〇年代所發展出的幾 何概念
- 提供了藝術設計、科學理論、自然景象三 方面的新關聯與新詮釋。
- 碎形的深刻內涵促成了不同領域的結合。



Fig 1. Benoit Mandelbrot

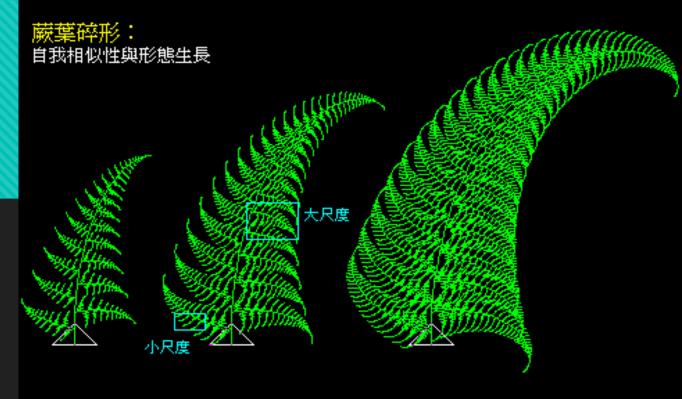
碎形- 自我相似性 self-similarity

- 碎形用來研究邊線曲折、結構細膩、圖案複雜,但是卻具有「自我相似性」 (self-similarity)的物件形態。
- 自然界隨處可見的雪**花晶體、樹幹枝椏、血管分支、海岸沿線、雲彩的邊** 緣以及**天空的閃電**——這些都是碎形
- ○特徵:都是結構上一層又一層相類似的重複。
- 觀察一棵蒼茂大樹與它主幹上的樹枝、樹枝上的枝杈、葉脈的紋路
 - ○它們的形狀非常雷同,僅僅有大小與方位上的差異。
 - ○這種關係在幾何學上稱為自我相似性,是碎形研究的核心。

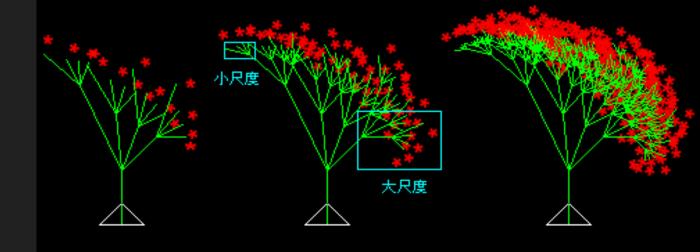
蕨葉碎形、樹枝碎形

小藍框是大藍框的標準縮小版 小藍框是由更多、更小的相似 細節所構成,

把一整體縮小之後再嵌入自身



樹枝碎形: 自我相似性與形態生長



<第二堂>Draw patterns

學習活動 2:建構自己的碎形規則與碎形圖

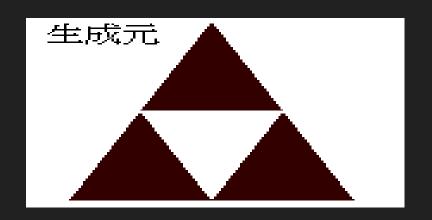
一、起始元與生成元疊代法

- ○在繪製碎形的方法中,「**起始元與生成元疊代法**(Generator Iteration Method)」是最為直觀與最容易操作的
- ○「完全自我相似」(Strict Self-Similarity)碎形
- ○「完全自我相似」(Strict Self-Similarity)碎形疊代法必須指定 起始元(Initiator)與生成元(Generator)。

起始元與生成元

- ○**起始元**:是碎形一開始的圖形,起始元是由單一的或幾個自我相似的幾何單元(例如線段、三角形或矩形......等等)所組成。
- **○生成元**:是起始元中的每一個自我相似的幾何單元下一次疊代的圖形。

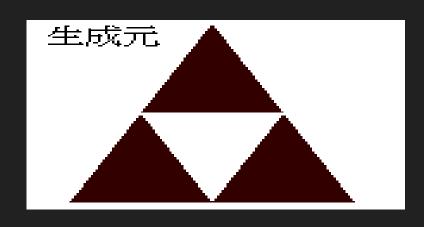




謝爾賓斯基(Sierpinski)三角形

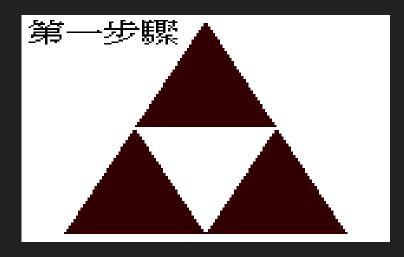
- ○以 Sierpinski Gasket 為例
- ○從下面的步驟會發現,顯然碎形的第零步驟就是起始元,而第一步驟則是以生成元來取代 代起始元中所有的幾何單元,而第二步驟便 是以生成元來取代第一步驟中所有的幾何單 元,接下來的步驟,即是以相同的方法重複 疊代下去。

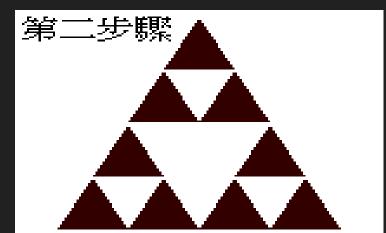


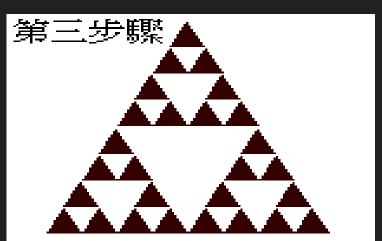


謝爾賓斯基三角形

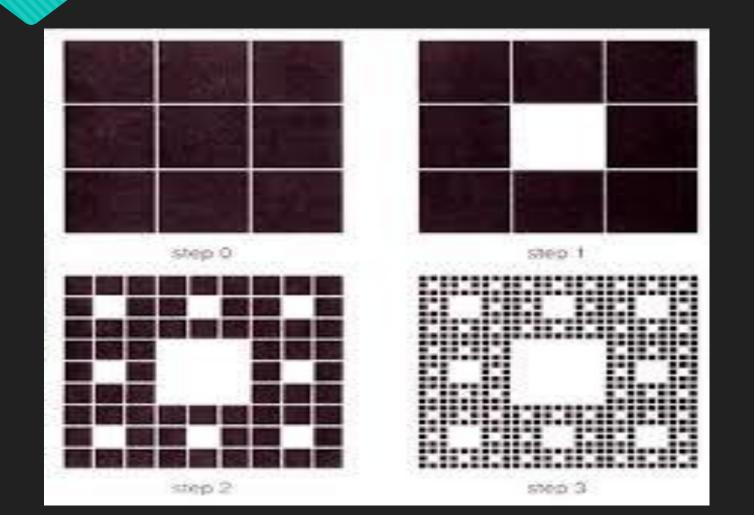




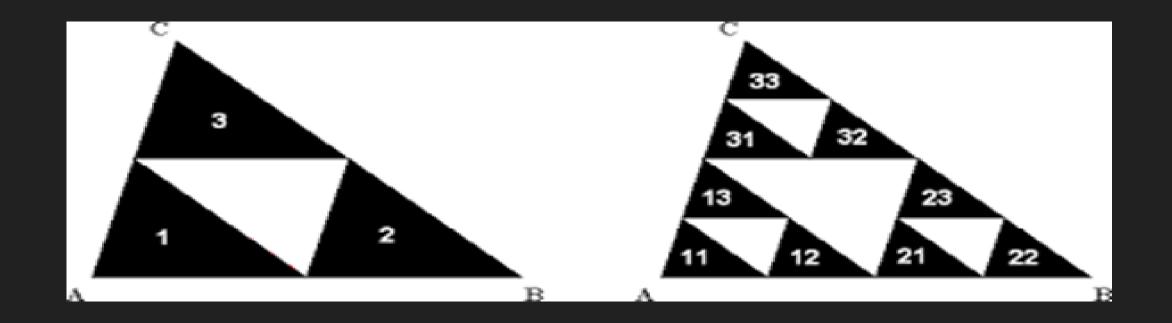




其他例子-1



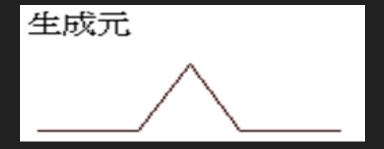
其他例子-2



Koch Curve (Snowflake Curve)

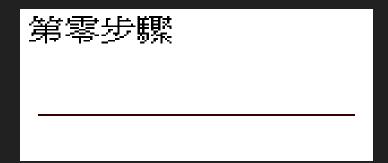
- 起始元:由單一的線段幾何單元所組成的
- 生成元:生成元本身就定義了碎形的繪製規則與碎形的特徵。將在每一個 步驟裡取代這些線段單元。

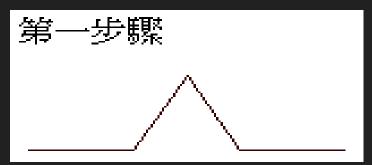
起始元



Koch Curve (Snowflake Curve)

Koch Curve 每一個步驟的疊代過程:









分組任務:建構碎形

| 1、設計碎形繪製規則 | | | | | |
|------------|-----|--|--|--|--|
| 起始元 | 生成元 | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

2、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 |
|------|------|------|------|
| | | | |
| | | | |
| | | | |
| | | | |

<第三堂> Drawing Patterns by programming

學習活動 3: Geometric drawing by programming

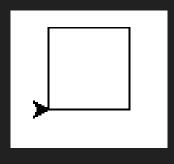
藝術家(Artist): 運用已知程式積木完成繪圖任務

| 步骤 | 任務説明 | 完成作品 | 程式積木 |
|----|--|----------|---|
| 1. | 階段 10 的第 1 關 歡迎光簡藝術家!首先·讓我們 試著用 "向右旋轉" 及 "向前移 動" 積木・畫一個簡單的正方形 吧。每週週長為 100 像素。 | R | 程式場本 内部移動で 200で 物業 (所容別で 90で 度 |
| 2. | 階段 10 的第 2 開- 用 120 度 · 60 度及 150 做素當 做邊長 · 畫出遍賴鏡石吧 | • | 程が後代で100 総元 (南田林 190 平 田 (南石林 190 平 田 |
| 3. | 階段 10 的第 3 開 你能用溫長 100 像素及旋轉 60 度·來畫出鑄個六邊形碼? | R | 程が後末 南部移動で100 商業 同名制で190で 財 |
| 4. | 階段 10 的第 4 關 現在你有了一個新的程式積木 可以玩職・這個程式積木是"超 圈"、它能將一組命令重複執 行。如果將這些指令重覆執行 6 次、會發生什麼事情呢? | <u>R</u> | 京芸様木 南前移動 ▼ 100 |

| 步程 | 任務説明 | 完成作品 | 程式積木 |
|----|--|-------|--|
| 5. | 階段 10 的第 5 關 你需要順禮循環多少次·才能將 速頭鑽石畫成一朵花呢? | * | (ACTION ACTION A |
| 6. | 階段 10 的第 6 關 請注意我們是如何隱化讓石代 裝:藉由將它變成一個迴團。你 可以將整個序列使用迴圖重複 執行 12 次來書出一朵花。 證 示:在鎦石迴團執行後你需要旋 轉 30 度,這樣你才不會畫在相 同的鑽石上。 | | THE COLUMN TWO IS NOT |
| 7_ | 階段 10 的第 7 關 循環這個畫出六塊形的程式編 碼六來來完成時個圖案。每一次 你需要轉多少角度?提示:把圖 的度數去除以循環的次數。 | • | TOTAL STREET |
| 8. | 階段 10 的第 8 關 一個商式是另一種程式工具來 幫助你避免重複打相同的程式 描碼。遊假商式是用來畫花朵, 所以你可以在任何你想要的時候用它來畫花朵。用這個商式和 新的"鼠"積本來畫這些花朵。提 示: 谑無花朵彼此態格 150 做素 | SE SE | COMPANY TO SERVICE STATE OF THE PROPERTY OF TH |

Draw shapes in Python Turtle Graphics

```
from turtle import *
canvas = Screen()
sarah = Turtle()
sarah.forward(50)
sarah.left(90)
sarah.forward(50)
sarah.left(90)
sarah.forward(50)
sarah.left(90)
sarah.forward(50)
sarah.left(90)
canvas.exitonclick()
```



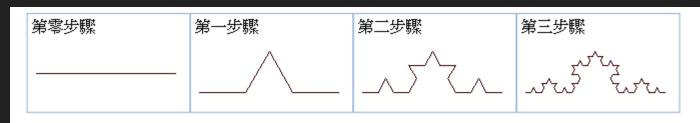
```
from turtle import *
canvas = Screen()
sarah = Turtle()
for i in range(4):
#repeat four times
    sarah.forward(50)
    sarah.left(90)
canvas.exitonclick()
```

<第四堂>Drawing Fractals by programming

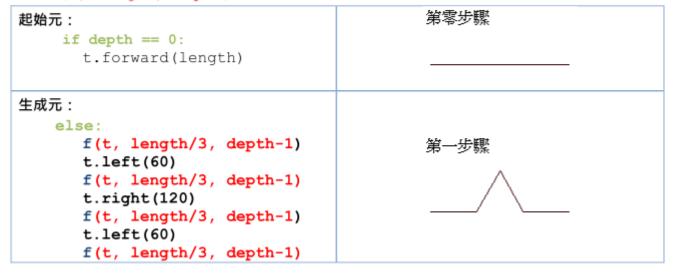
模組化-遞迴函式Fractals

程式實作與觀察-繪製Koch Curve

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



def f(t, length, depth):



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

程式實作與觀察

- ○繪製蕨葉(Fern),試修改生長規則另生成不同蕨葉
- ○繪製碎形樹,試修改生長規則另生成不同形態的樹





建構碎形圖規則,以程式繪製碎形圖

| 起始元 | 生成元 |
|-----|-----|
| 起始元 | 生成元 |
| | |
| | |

碎形繪製規則

| [3] 依幹形繪製規則豐代建構至第4代幹形 | | | | |
|-----------------------|------|------|------|--|
| 第零步驟 | 第一步驟 | 第二步驟 | 第三步驟 | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

[4] 碎形程式碼