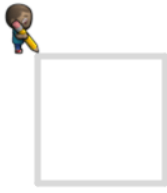
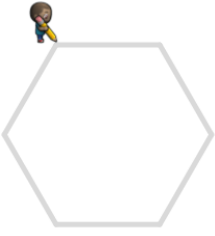
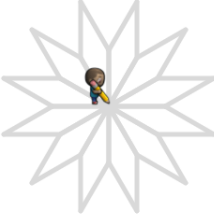






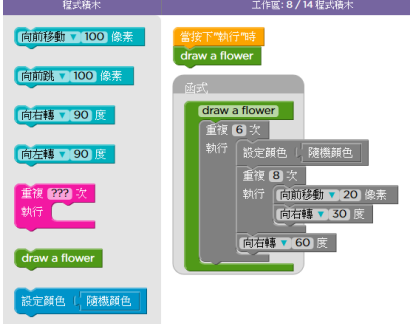


視覺化程式設計-碎形繪圖專題

學習活動 3 : Geometric drawing by programming

一、藝術家(Artist) : 運用已知程式積木完成繪圖任務

步驟	任務說明	完成作品	程式積木
1.	<p>階段 10 的第 1 關</p> <p>歡迎光臨藝術家！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。每邊邊長為 100 像素。</p>		<p>程式積木</p> <ul style="list-style-type: none"> 向前移動 100 像素 向右轉 90 度 向左轉 90 度
2.	<p>階段 10 的第 2 關-</p> <p>用 120 度、60 度及 150 像素當做邊長，畫出這顆鑽石吧</p>		<p>程式積木</p> <ul style="list-style-type: none"> 向前移動 100 像素 向右轉 90 度 向左轉 90 度
3.	<p>階段 10 的第 3 關</p> <p>你能用邊長 100 像素及旋轉 60 度，來畫出這個六邊形嗎？</p>		<p>程式積木</p> <ul style="list-style-type: none"> 向前移動 100 像素 向右轉 90 度 向左轉 90 度
4.	<p>階段 10 的第 4 關</p> <p>現在你有了一個新的程式積木可以玩囉，這個程式積木是"迴圈"，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？</p>		<p>程式積木</p> <ul style="list-style-type: none"> 向前移動 100 像素 向右轉 90 度 向右轉 90 度 重複 6 次 執行

步驟	任務說明	完成作品	程式積木
5.	<p>階段 10 的第 5 關</p> <p>你需要重複循環多少次，才能將這顆鑽石畫成一朵花呢？</p>		
6.	<p>階段 10 的第 6 關</p> <p>請注意我們是如何簡化鑽石代碼：藉由將它變成一個迴圈。你可以將整個序列使用迴圈重複執行 12 次來畫出一朵花。提示：在鑽石迴圈執行後你需要旋轉 30 度，這樣你才不會畫在相同的鑽石上。</p>		
7.	<p>階段 10 的第 7 關</p> <p>循環這個畫出六邊形的程式編碼六次來完成這個圖案。每一次你需要轉多少角度？提示：把圓的度數去除以循環的次數。</p>		
8.	<p>階段 10 的第 8 關</p> <p>一個函式是另一種程式工具來幫助你避免重複打相同的程式編碼。這個函式是用來畫花朵，所以你可以在任何你想要的時候用它來畫花朵。用這個函式和新的"跳"積木來畫這些花朵。提示：這些花朵彼此間格 150 像素</p>		

步驟	任務說明	完成作品	程式積木
9.	<p>階段 10 的第 9 關</p> <p>這裡有一個我們之前編寫用來畫六邊形的函式叫做"畫一個形狀"。執行程式看看他會做什麼並去試著修改函式看會發生什麼事。你可以讓他畫一個樣式是由正方形或三角形或是八邊形組成的嗎?</p>		
10.	<p>階段 10 的第 10 關</p> <p>現在去試著做任何你喜歡的圖案! 你有一個空的函式用來做任何你想做的事!</p>		

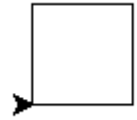
二、Draw shapes in Python Turtle Graphics

[1] 繪製邊長 50 的正方形 (Square1.py)

```
from turtle import *
canvas = Screen()
sarah = Turtle()

sarah.forward(50) # make sarah draw a square
sarah.left(90)
sarah.forward(50)
sarah.left(90)
sarah.forward(50)
sarah.left(90)
sarah.forward(50)
sarah.left(90)

canvas.exitonclick()
```



[2] 繪製邊長 50 的正方形 (Square2.py)

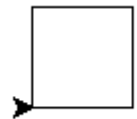
觀察上述程式碼，完成下列問題

- 重複的段落是：
- 重複次數是：_____

```
from turtle import *
canvas = Screen()
sarah = Turtle()

for i in range(4): #repeat four times
    sarah.forward(50)
    sarah.left(90)

canvas.exitonclick()
```



[3] 繪製邊長 150 的三角形 (Triangle.py)

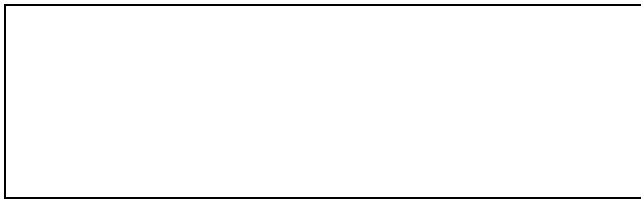
```
from turtle import *
canvas = Screen()
sarah = Turtle()

for i in range(__):
    sarah.forward(__)
    sarah.left(__)

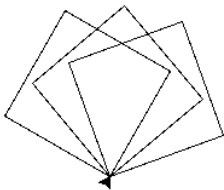
canvas.exitonclick()
```

[4] 繪製長 150 寬 100 的長方形(Rectangle.py)

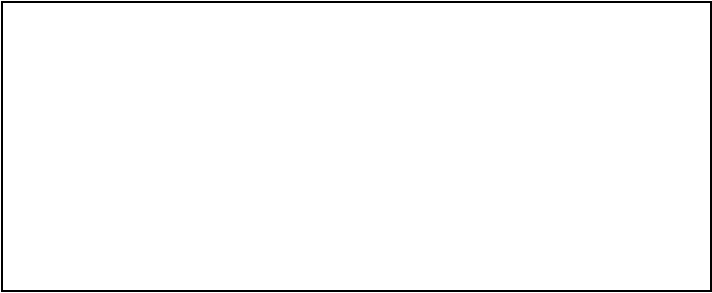
```
from turtle import *
canvas = Screen()
sarah = Turtle()

for i in range(__):
    
    canvas.exitonclick()
```

[5] 討論：繪製邊長 150 · 20 度角的迴旋正方形



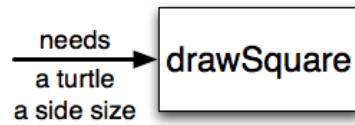
```
from turtle import *
canvas = Screen()
sarah = Turtle()



canvas.exitonclick()
```

三、To make procedure in Python to enter command quicker.

```
def name ( parameters ) :  
    statements1  
    statements2  
    statements3
```



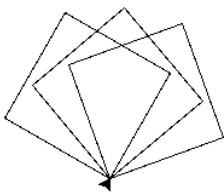
[1] 繪製邊長 150 的正方形 (Square1.py)

```
from turtle import *
```

```
def drawSquare(t, size):  
    for i in range(4):  
        t.forward(size)  
        t.left(90)
```

```
canvas = Screen()  
sarah = Turtle()  
drawSquare(alex, 150)  
canvas.exitonclick()
```

[2] 繪製邊長 150, 20 度角的迴旋正方形 (TiltedSquare.py)



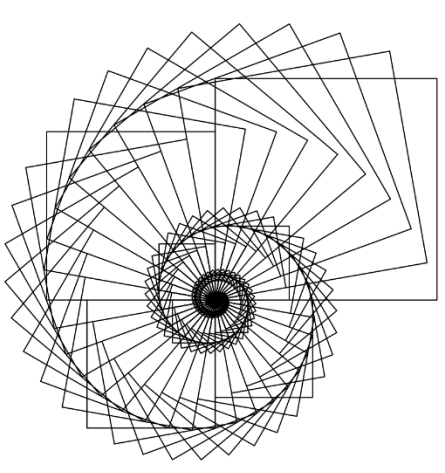
```
from turtle import *
```

```
def drawSquare(t, size):  
    for i in range(4):  
        t.forward(size)  
        t.left(90)
```

```
canvas = Screen()  
sarah = Turtle()
```

```
canvas.exitonclick()
```

[3] 繪製邊長 1-50 的迴旋正方形, 20 度角的迴旋正方形 (SpiralSquare.py)



```
from turtle import *  
def drawSquare(t, size):  
    for i in range(4):  
        t.forward(size)  
        t.left(90)  
  
canvas = Screen()  
sarah = Turtle()  
  
canvas.exitonclick()
```

[4] 繪製下列幾何圖形(任選 3 個)

