

第2章 基礎程式設計(1)

- 認識演算法與程式語言
- Scratch 程式設計 - 基礎篇
- Scratch 程式設計 - 計算篇
- Scratch 程式設計 - 繪圖篇



翰林出版

目錄



基礎程式設計

1. 認識演算法與程式語言

2. Scratch程式設計-基礎

3. Scratch程式設計-計算

4. Scratch程式設計-繪圖

Scratch-繪圖篇



畫正
方形

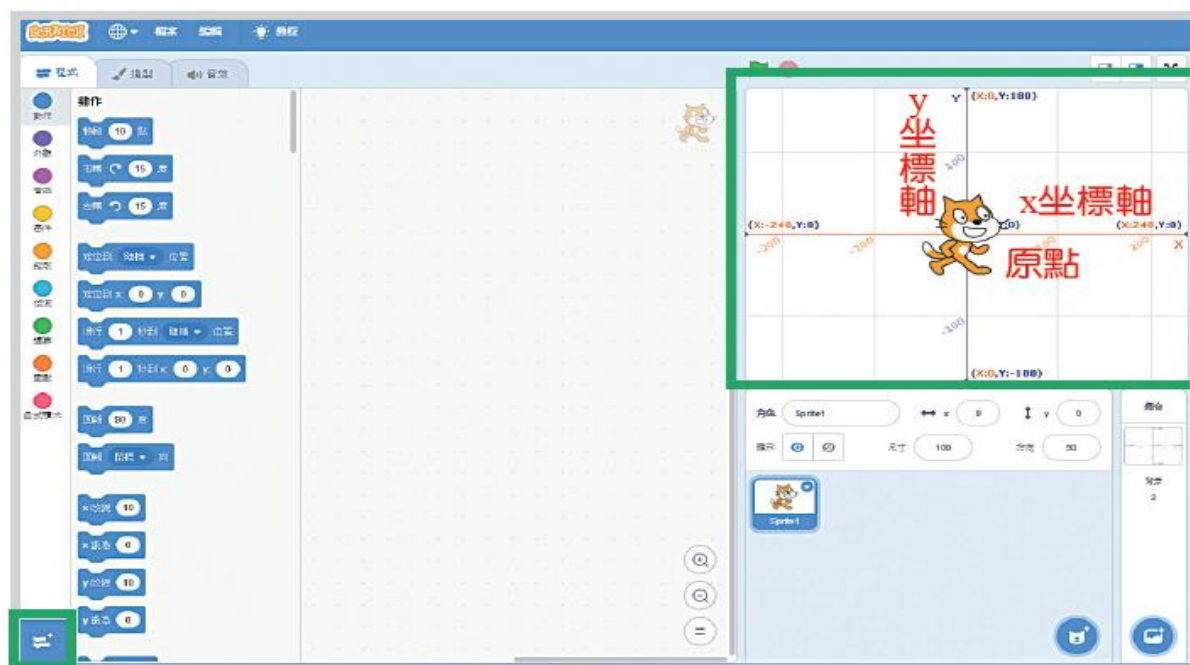
擴散
方形

旋轉
方形

繪圖功能



舞臺畫面寬480點，高360點。畫面正中央是坐標系統的原點 (0,0)。

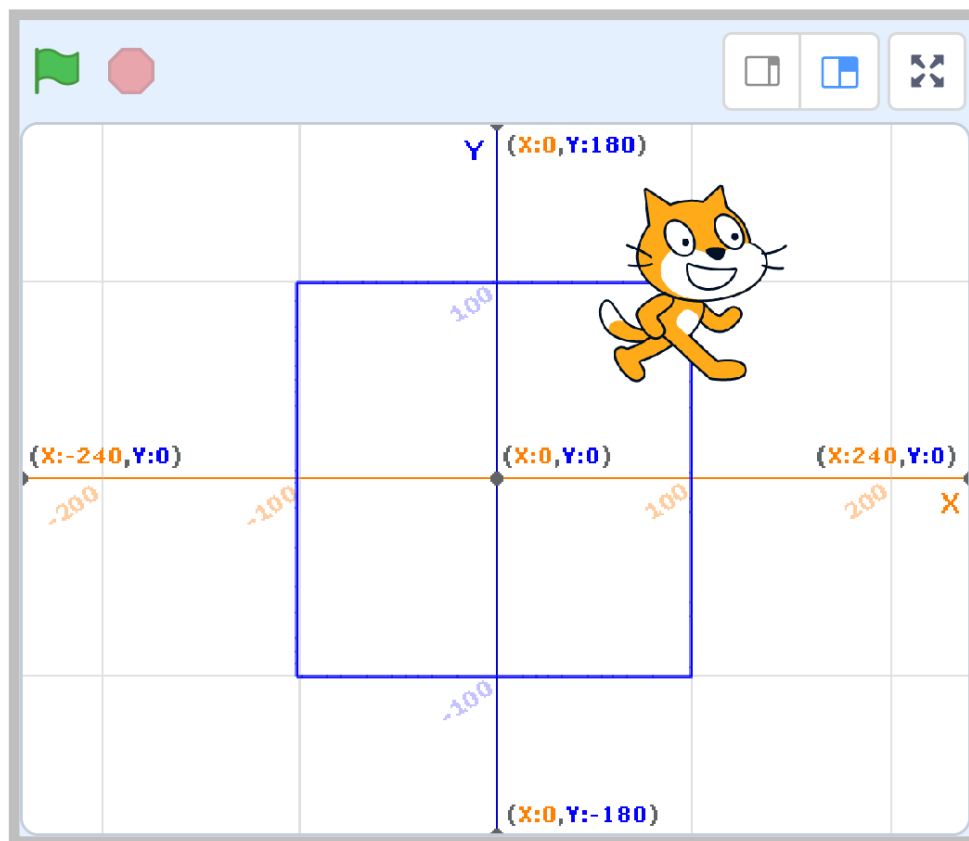


舞臺區標示出坐標軸與原點

繪圖功能



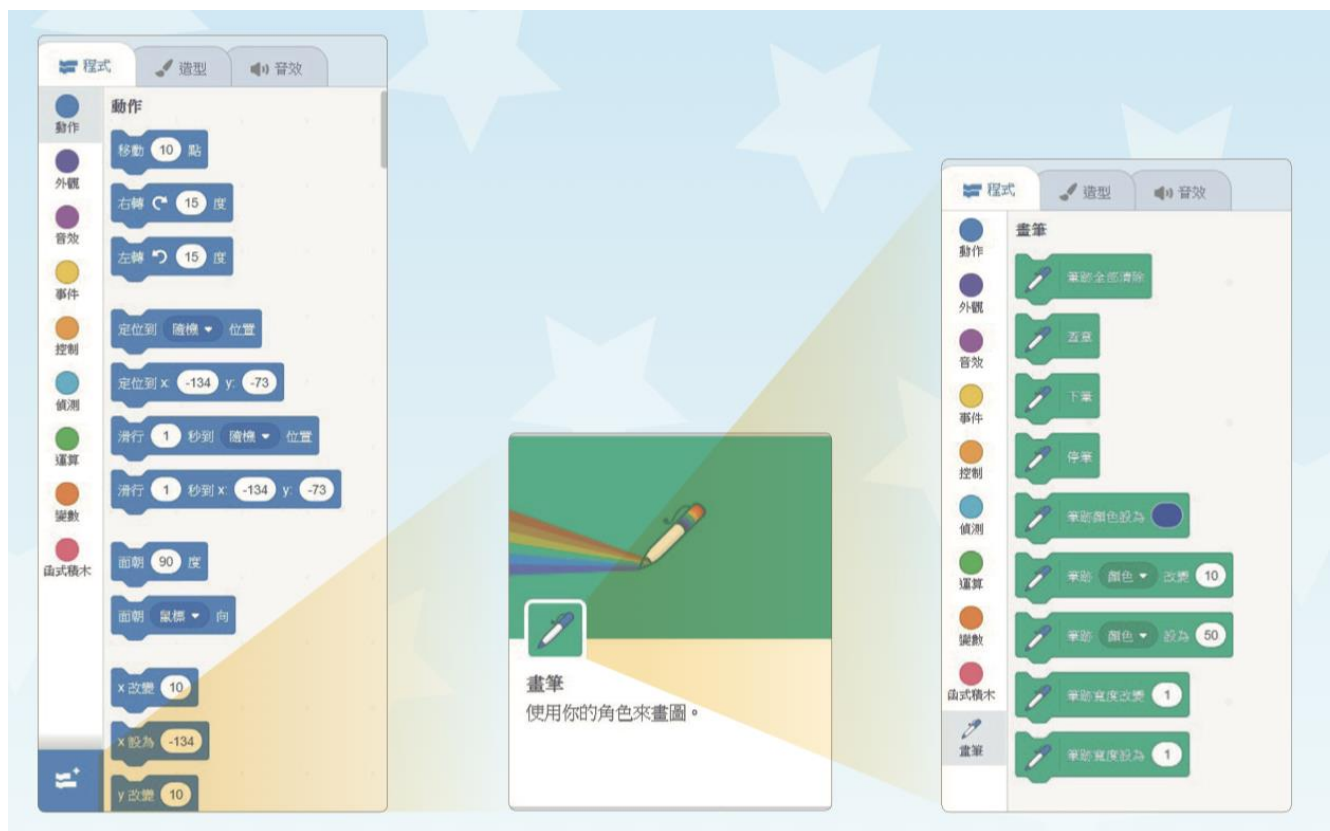
坐標系統的原點是(0,0) ， 原點的X軸
往右是正數，
往左是負數；
原點的y軸
往上是正數，
往下是負數。



繪圖功能



使用畫筆前需要先從**擴展功能**中新增相關積木



腳本區下方可以添加擴展功能

繪圖功能



從擴展功能增加後，畫筆的相關積木會呈現在「程式」的面板內。

The image shows two parts of the Scratch interface. On the left is the '選擇擴充功能' (Select Extensions) window, which lists various extensions. The '畫筆' (Drawing) extension is highlighted with a red box. On the right is the '程式' (Code) editor, which shows a list of drawing blocks on the left sidebar and a stack of drawing blocks in the main workspace. The drawing blocks include: '筆跡全部清除' (Erase all), '蓋章' (Stamp), '下筆' (Begin drawing), '停筆' (End drawing), '筆跡顏色設為' (Set drawing color to), '筆跡 顏色 改變 10' (Change drawing color by 10), '筆跡 顏色 設為 50' (Set drawing color to 50), '筆跡寬度改變 1' (Change drawing width by 1), and '筆跡寬度設為 1' (Set drawing width to 1).

繪圖功能

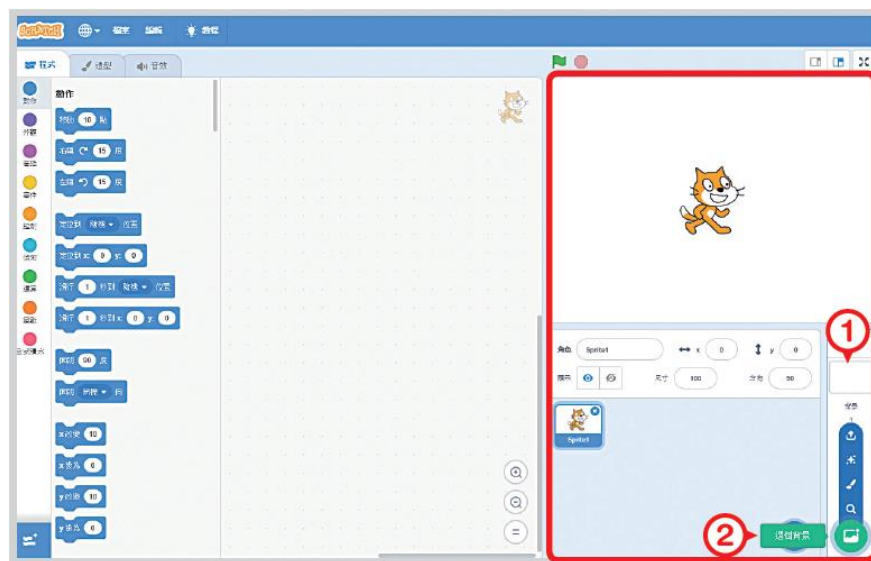


準備工作：在舞台匯入「坐標」為背景，
以便在範例中觀察小貓移動的變化

舞臺設計

步驟1: 開新檔案，匯入舞臺背景

1. 點選舞臺
2. 點選選個背景



繪圖功能

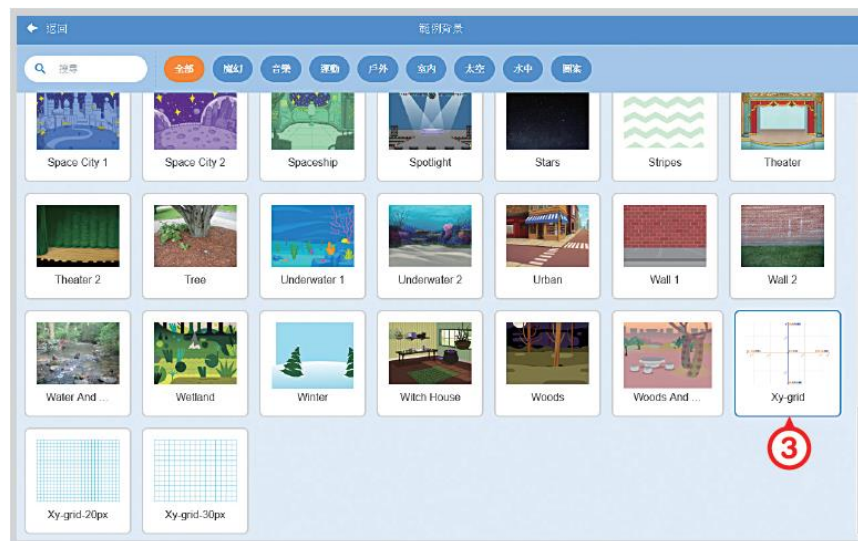


準備工作：在舞台匯入「坐標」為背景，
以便在範例中觀察小貓移動的變化

舞臺設計

步驟2: 選擇舞臺樣式

3. 從背景圖庫中，
選擇 **xy-grid**
圖片



畫正方形



接下來利用三種不同的方法來畫正方形，思考看看它們的優缺點！

坐標
積木

透過X、Y軸的數值設定，畫出一個正四方形。

方向
積木

利用方向與移動指令，來產生正四方形。

迴圈
計次

使用計次式迴圈讓小貓畫出一個正方形。

範例1-畫正方形



利用坐標積木讓小貓畫出一個正方形

問題解析

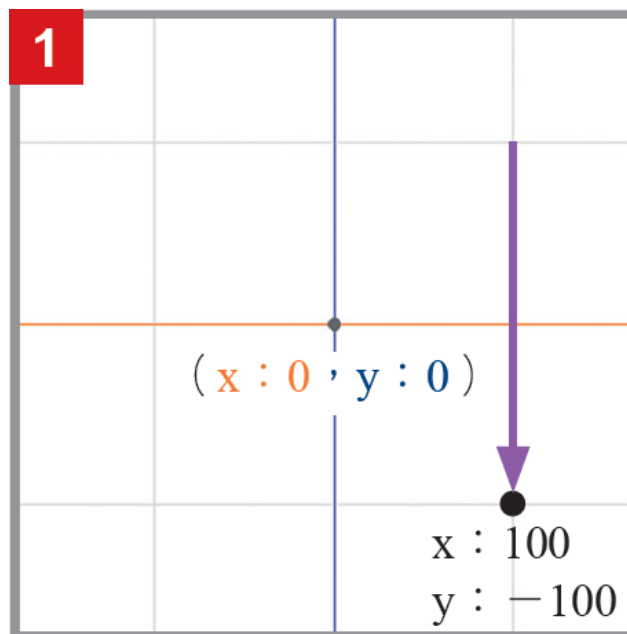
問題解析	問題實作
(A) 如何設定角色的初始位置？	輸入角色一開始出現的位置 
(B) 如何控制角色滑行至指定位置？	輸入滑行時間與終點位置的坐標 

用坐標積木畫正方形



利用坐標積木讓小貓畫出一個正方形

步驟：設定繪圖位置的坐標



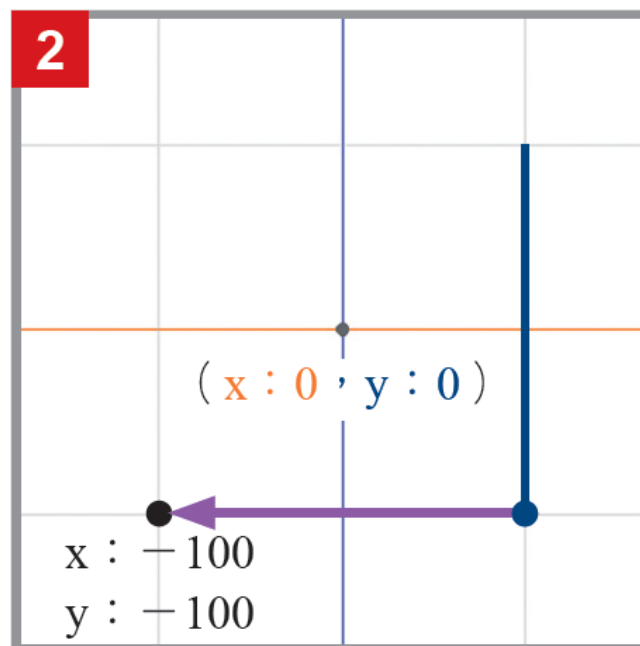
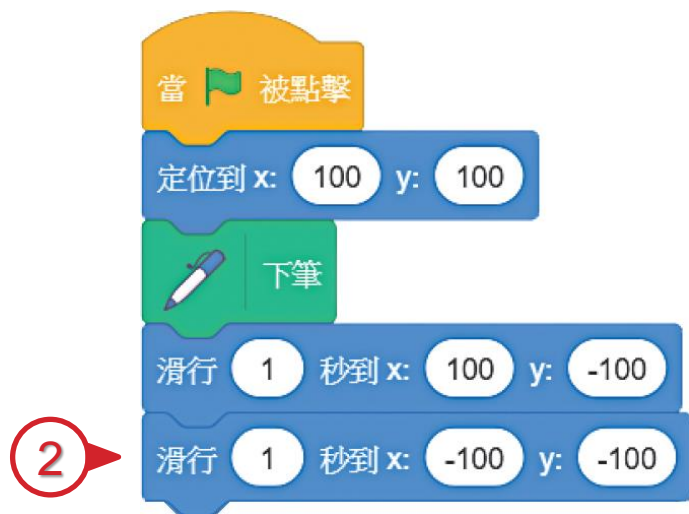
繪圖步驟拆解

用坐標積木畫正方形



利用坐標積木讓小貓畫出一個正方形

步驟：設定繪圖位置的坐標



繪圖步驟拆解

用坐標積木畫正方形



利用坐標積木讓小貓畫出一個正方形

步驟：設定繪圖位置的坐標

當 被點擊

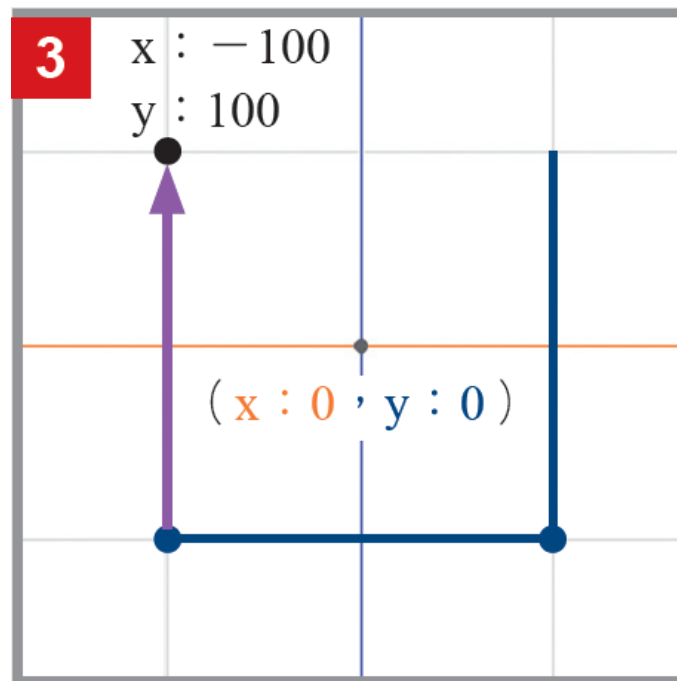
定位到 x: 100 y: 100

下筆

滑行 1 秒到 x: 100 y: -100

滑行 1 秒到 x: -100 y: -100

3 滑行 1 秒到 x: -100 y: 100



繪圖步驟拆解

用坐標積木畫正方形



利用坐標積木讓小貓畫出一個正方形

步驟：設定繪圖位置的坐標



當 旗 被點擊

定位到 x: 100 y: 100

下筆

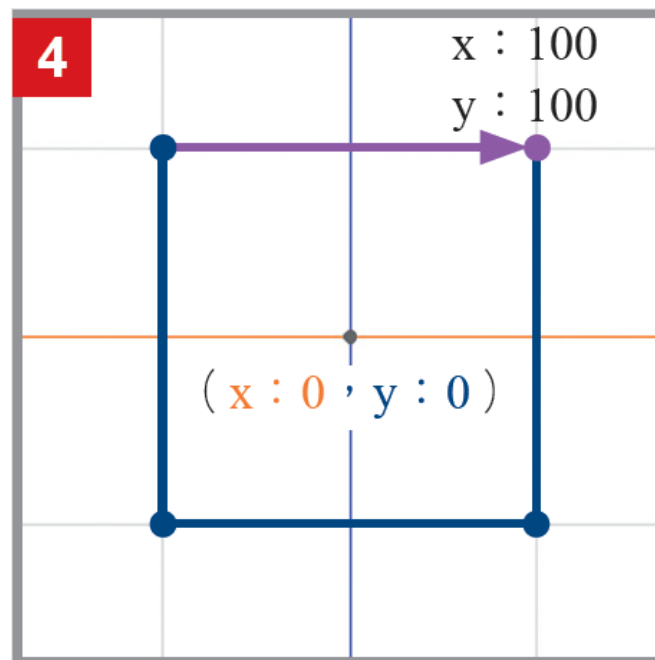
滑行 1 秒到 x: 100 y: -100

滑行 1 秒到 x: -100 y: -100

滑行 1 秒到 x: -100 y: 100

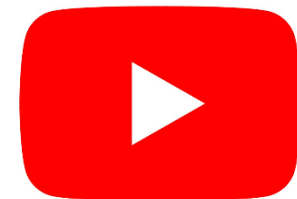
4 滑行 1 秒到 x: 100 y: 100

停筆



繪圖步驟拆解

用坐標積木畫正方形



利用坐標積木讓小貓畫出一個正方形

完整程式



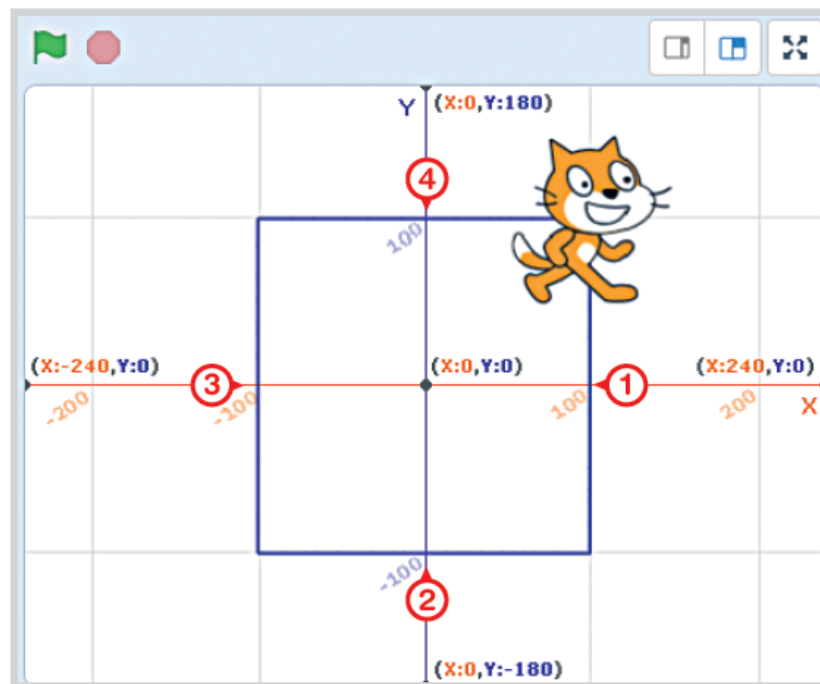
當被點擊

定位到 x: 100 y: 100

下筆

- 1 滑行 1 秒到 x: 100 y: -100
- 2 滑行 1 秒到 x: -100 y: -100
- 3 滑行 1 秒到 x: -100 y: 100
- 4 滑行 1 秒到 x: 100 y: 100

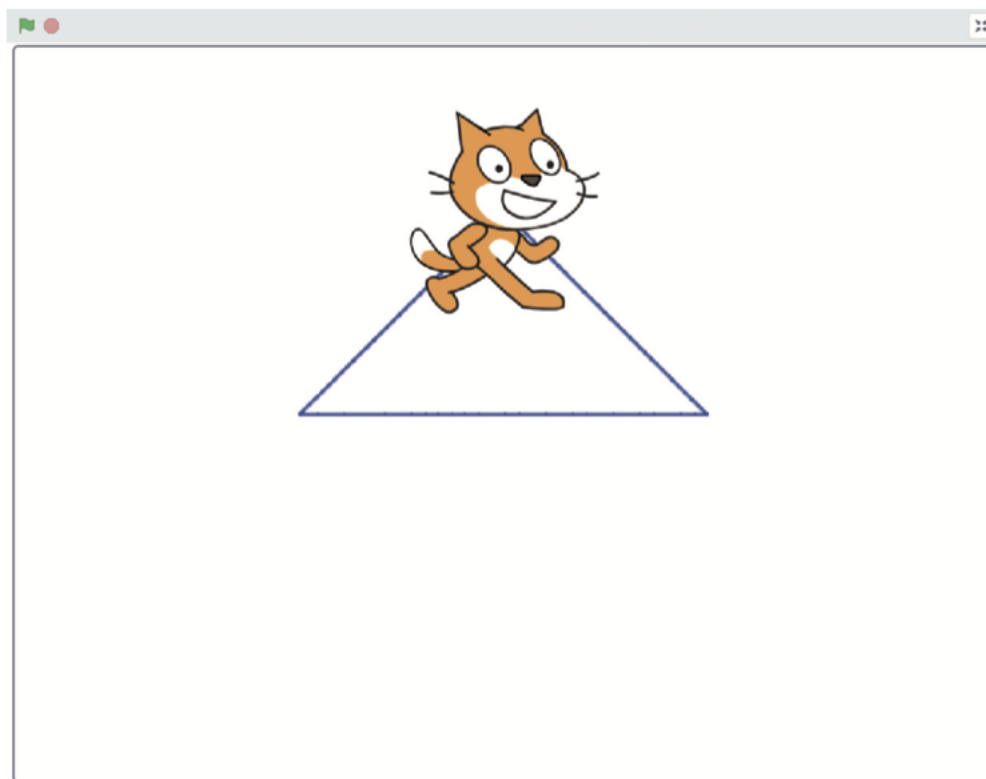
停筆



課堂練習



利用**坐標**積木讓小貓畫出一個三角形





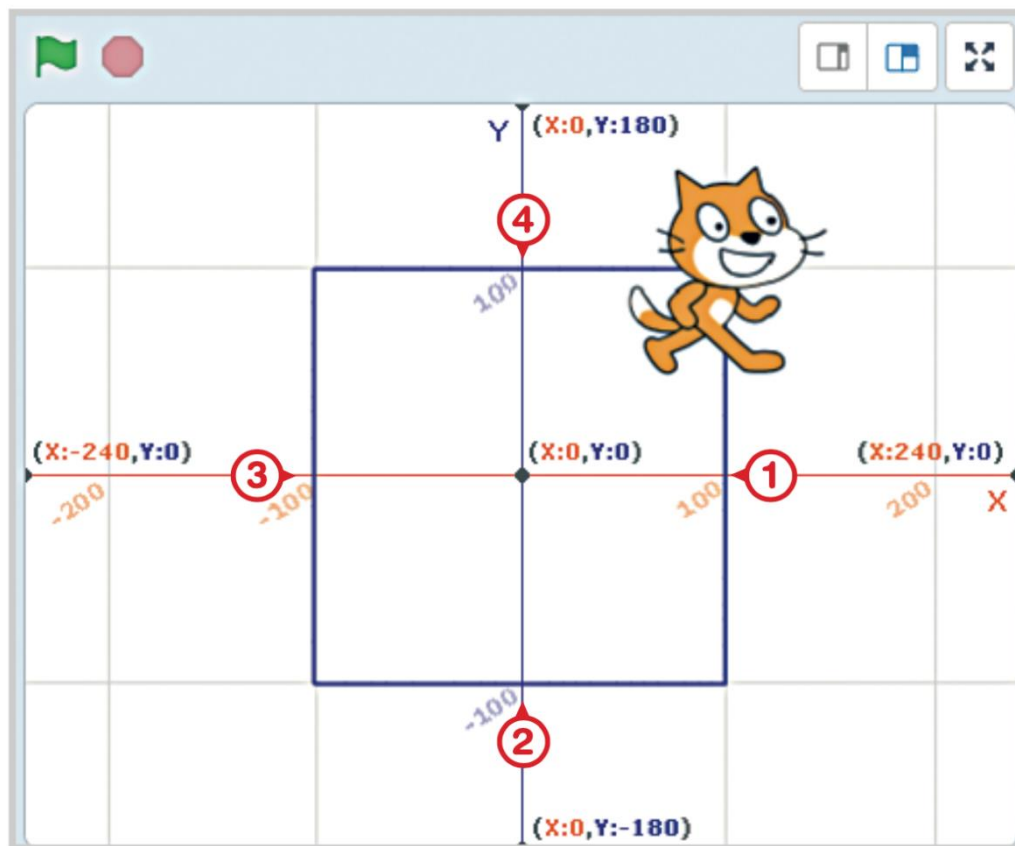
利用坐標積木讓小貓畫出一個三角形
撰寫程式

```
當 旗 被 點 擊
  定位到 x: 0 y: 100
  下 筆
  滑 行 1 秒 到 x: 100 y: 0
  滑 行 1 秒 到 x: -100 y: 0
  滑 行 1 秒 到 x: 0 y: 100
  停 筆
```

用方向積木畫正方形



利用方向積木讓小貓畫出一個正方形



用方向積木畫正方形



問題解析

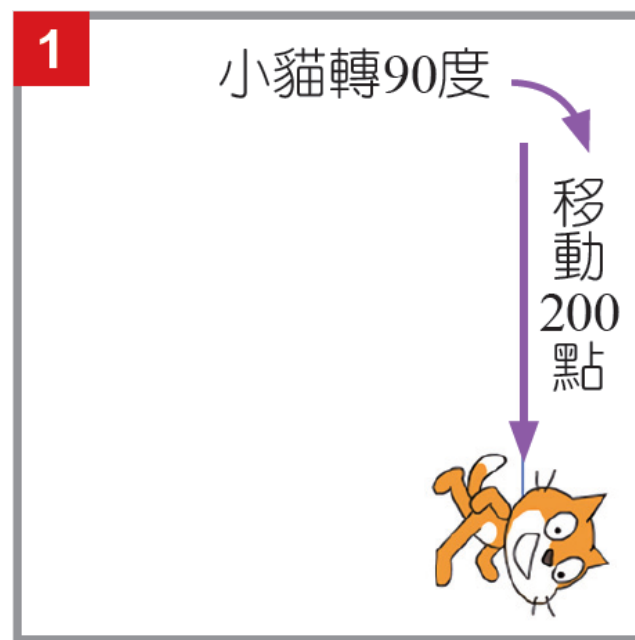
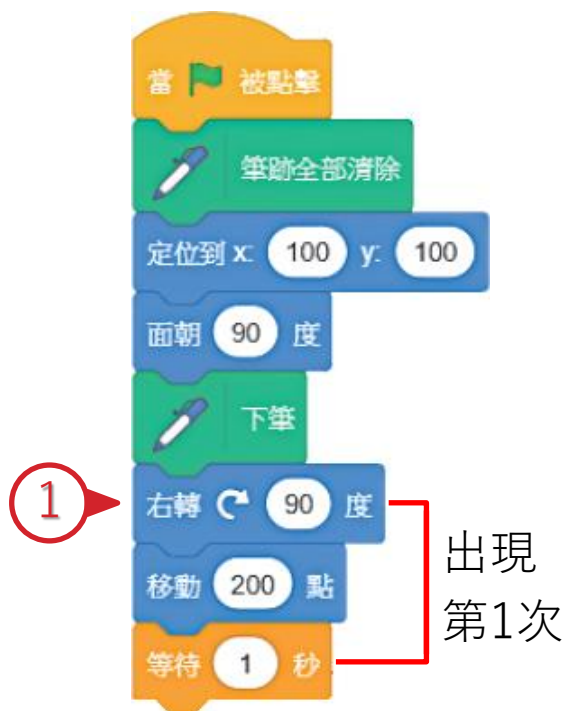
問題解析	問題實作
(A)如何設定角色的初始方位？	輸入角色一開始所朝向的方位。  
(B) 如何控制角色的轉向？	輸入旋轉角度，向不同方向旋轉。    
(C)如何控制角色移動的距離？	輸入要往前移動幾個點，代表移動幾個像素。 

用方向積木畫正方形



利用方向積木讓小貓畫出一個正方形

步驟:設定前進方向與移動距離



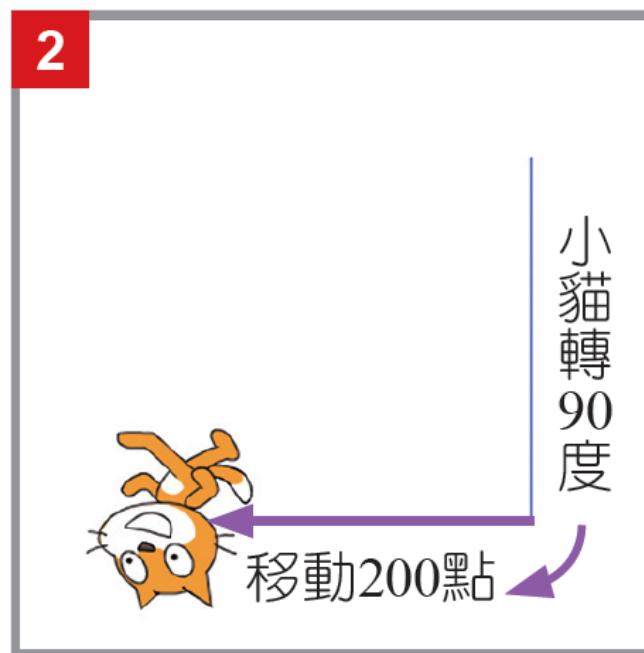
繪圖步驟拆解

用方向積木畫正方形



利用方向積木讓小貓畫出一個正方形

步驟:設定前進方向與移動距離



繪圖步驟拆解

用方向積木畫正方形

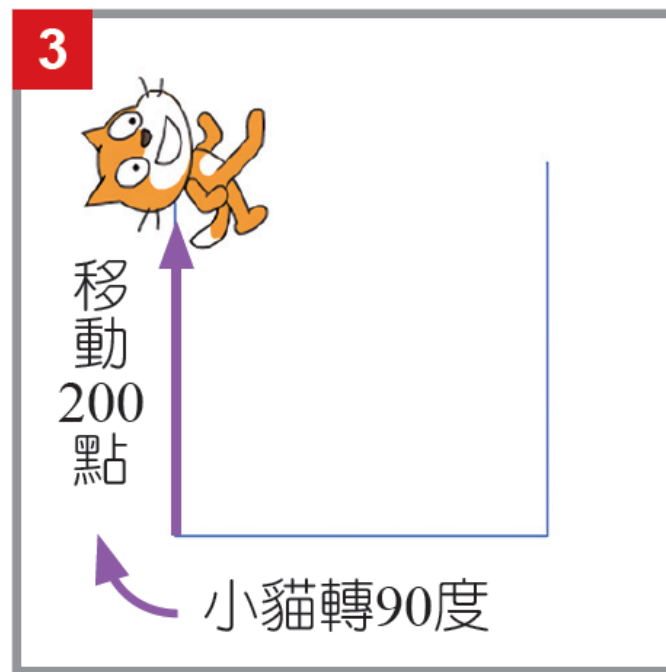


利用方向積木讓小貓畫出一個正方形

步驟:設定前進方向與移動距離



出現
第3次



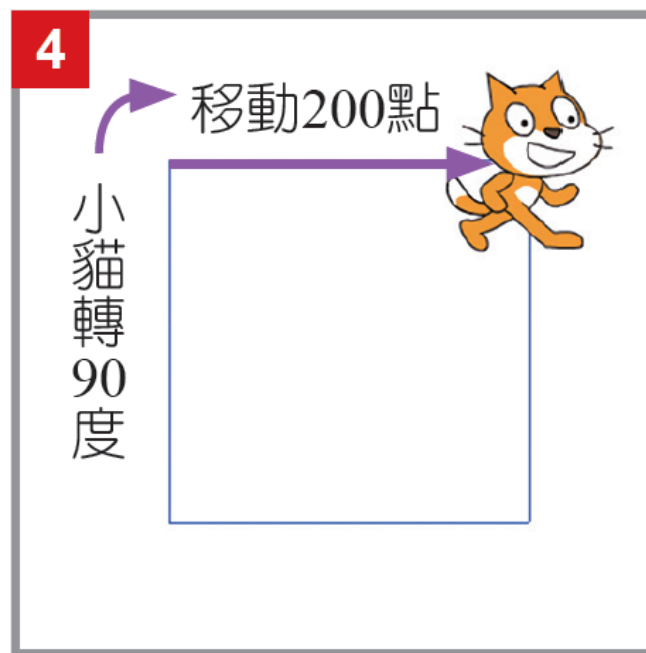
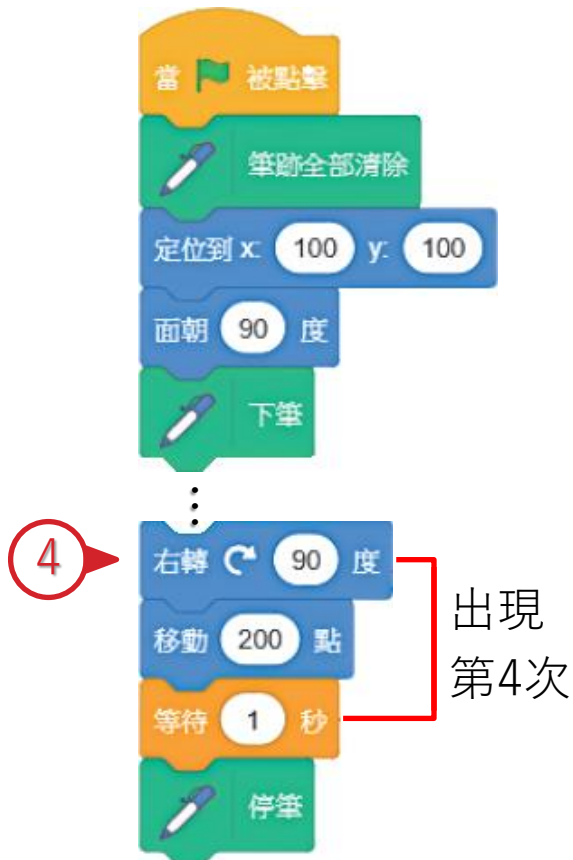
繪圖步驟拆解

用方向積木畫正方形



利用方向積木讓小貓畫出一個正方形

步驟:設定前進方向與移動距離

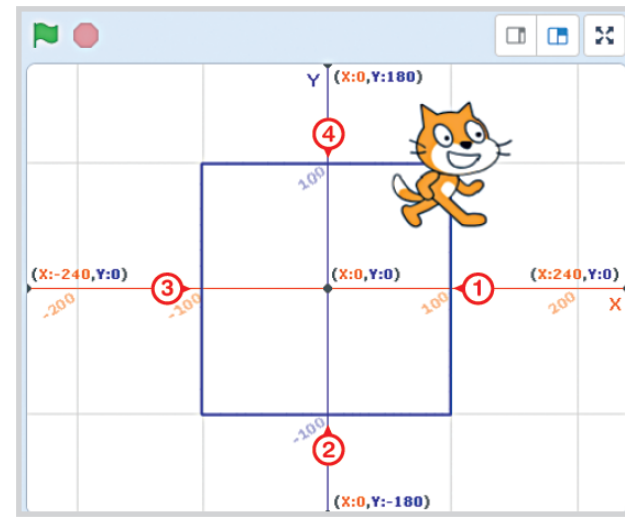
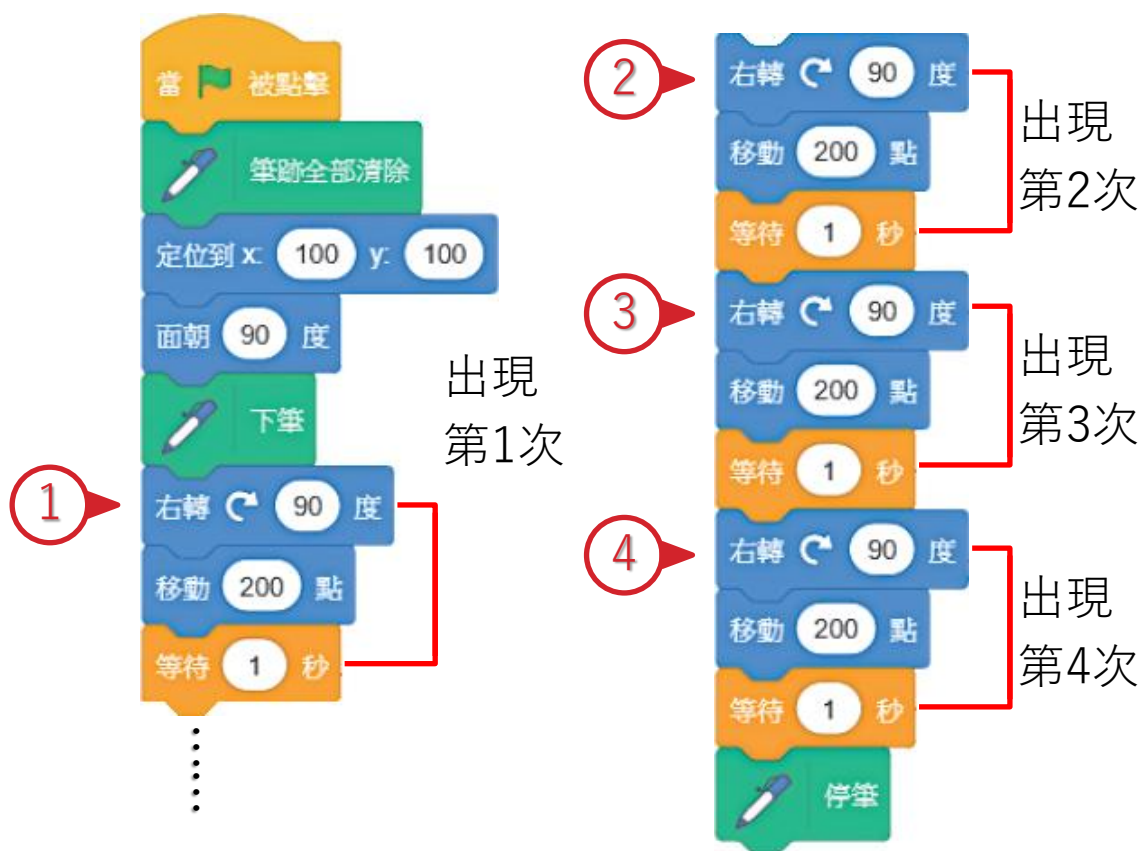


繪圖步驟拆解

用方向積木畫正方形



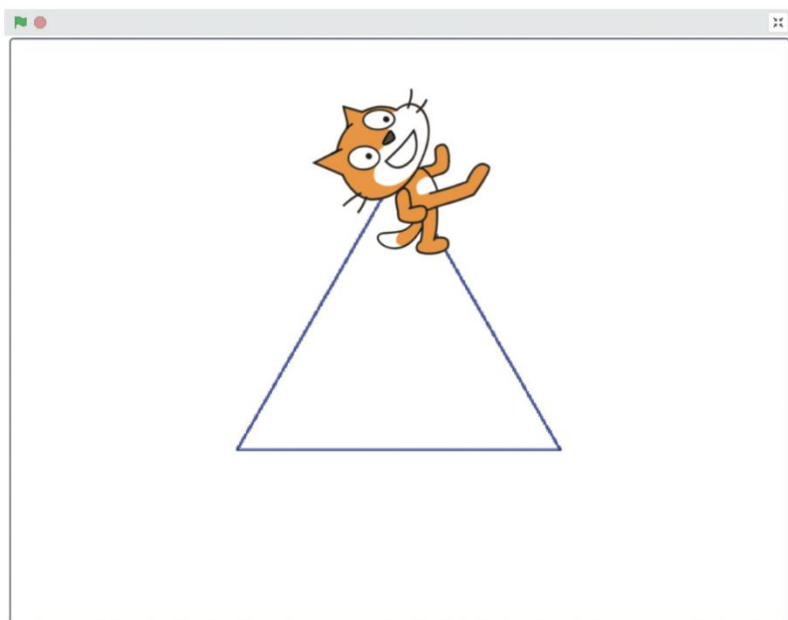
利用方向積木讓小貓畫出一個正方形



課堂練習



利用**方向積木**讓小貓畫出一個三方形

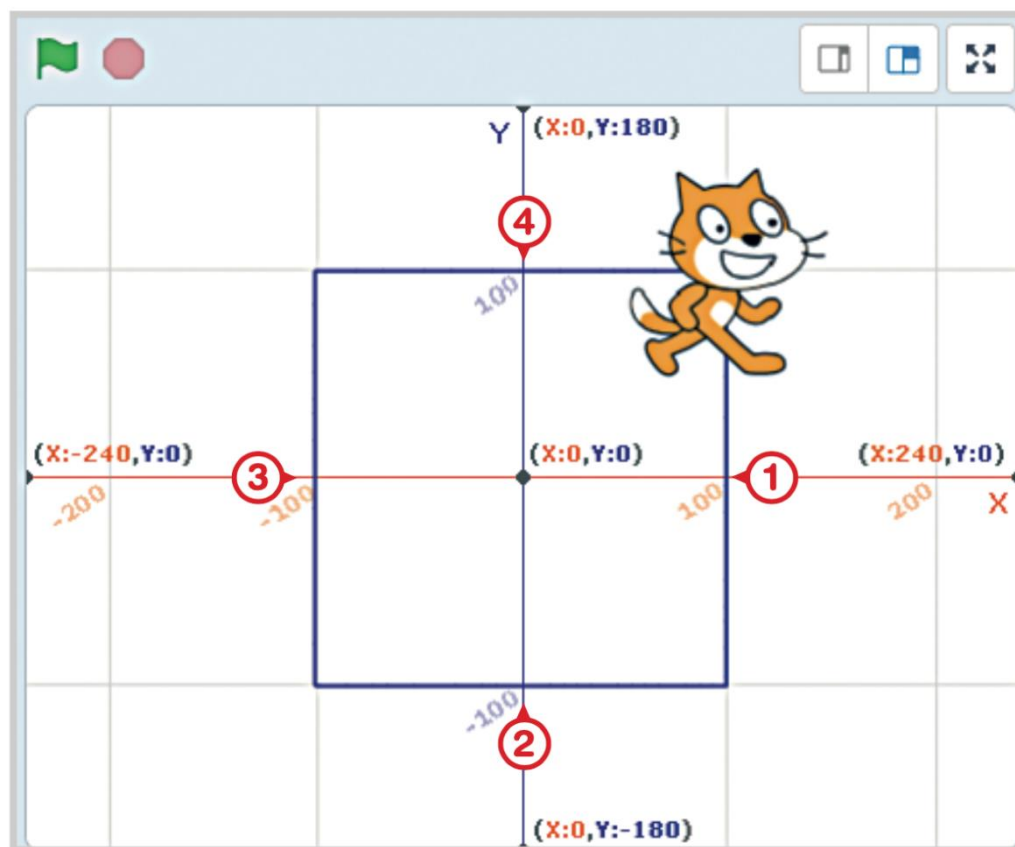


```
當 被點擊  
筆跡全部清除  
定位到 x: 0 y: 100  
面朝 30 度  
下筆  
右轉 120 度  
移動 200 點  
等待 1 秒  
右轉 120 度  
移動 200 點  
等待 1 秒  
右轉 120 度  
移動 200 點  
等待 1 秒  
停筆
```

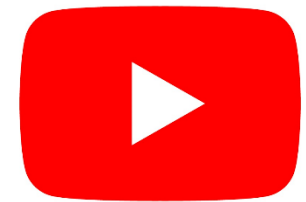
用計次式迴圈畫正方形



利用計次式迴圈讓小貓畫出一個正方形

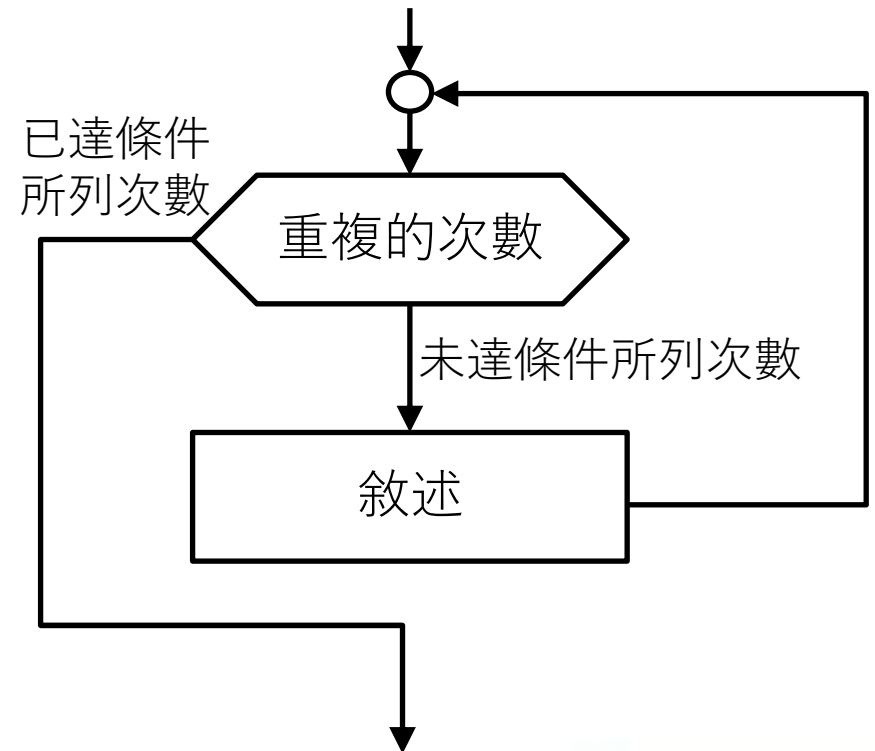
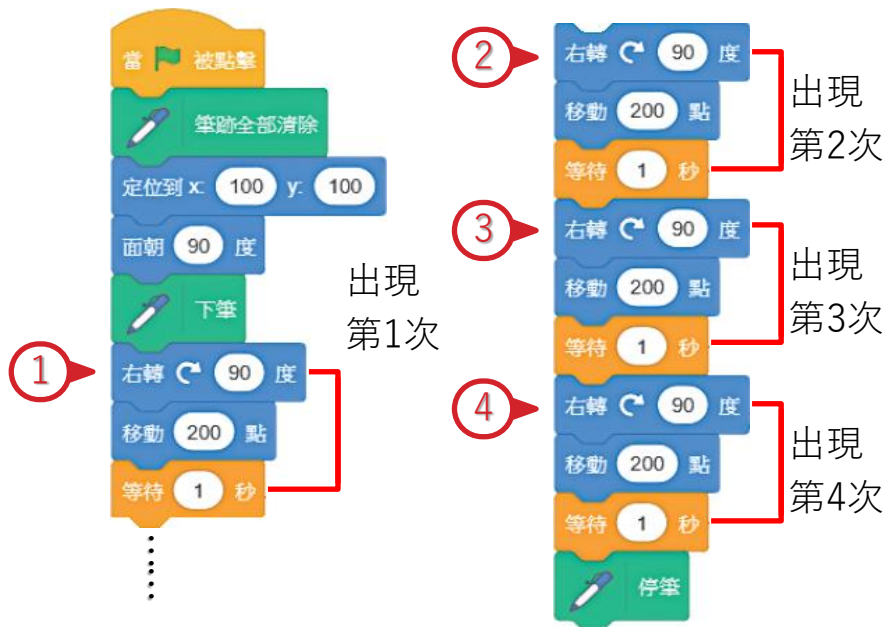


用計次式迴圈畫正方形



利用計次式迴圈讓小貓畫出一個正方形

步驟:設定計次式迴圈，取代重複的部份。



用計次式迴圈畫正方形



問題分析

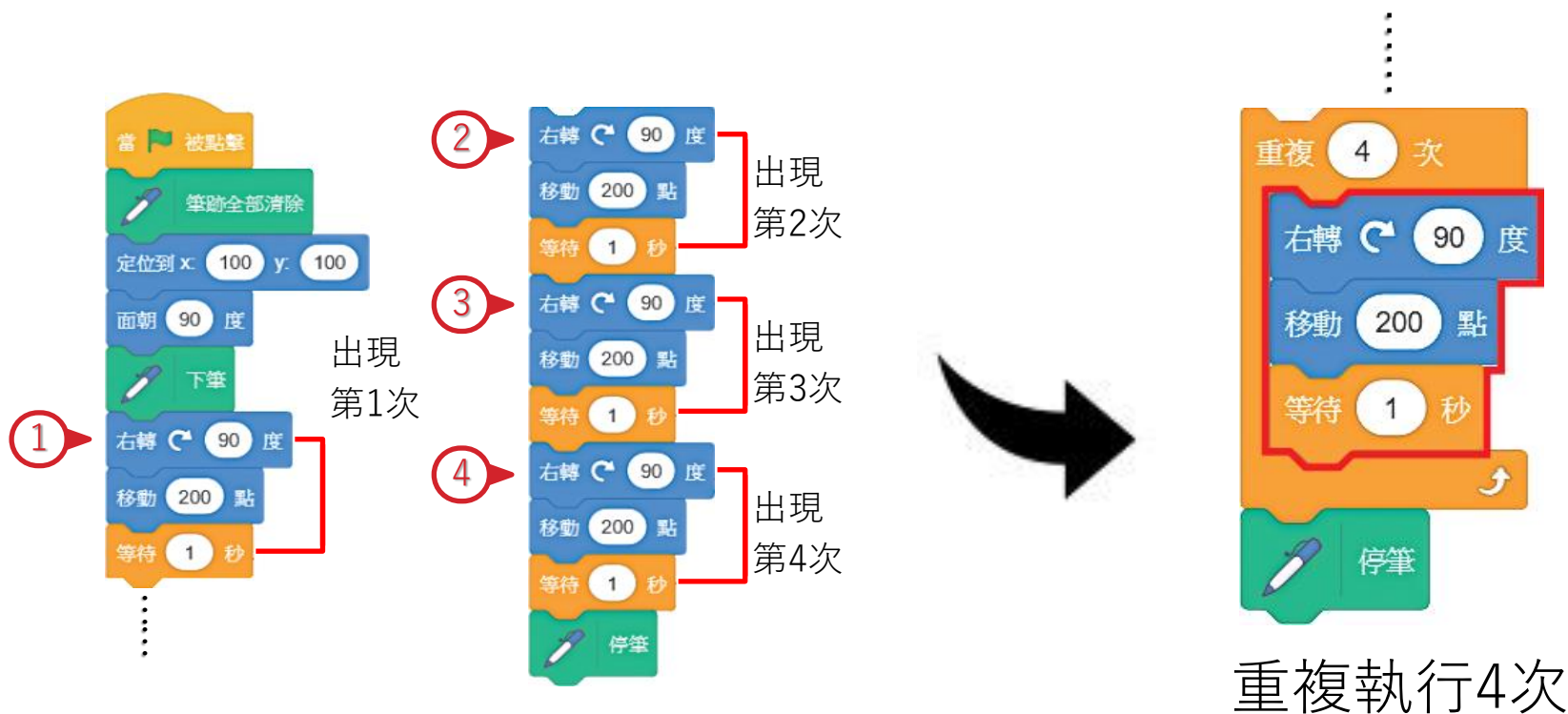
問題解析	問題實作
(A) 如何設定計次式迴圈？	用此積木取代重複執行程式碼  <p>輸入重複次數</p> <p>嵌入重複執行程式碼</p>
B) 如何控制角色的轉向？	輸入旋轉角度，向不同方向旋轉 
(C) 如何控制角色移動的距離？	用此積木，輸入要往前移動幾個點，代表移動幾個像素 

用計次式迴圈畫正方形



利用計次式迴圈讓小貓畫出一個正方形

步驟:設定計次式迴圈, 取代重複的部份。



用計次式迴圈畫正方形

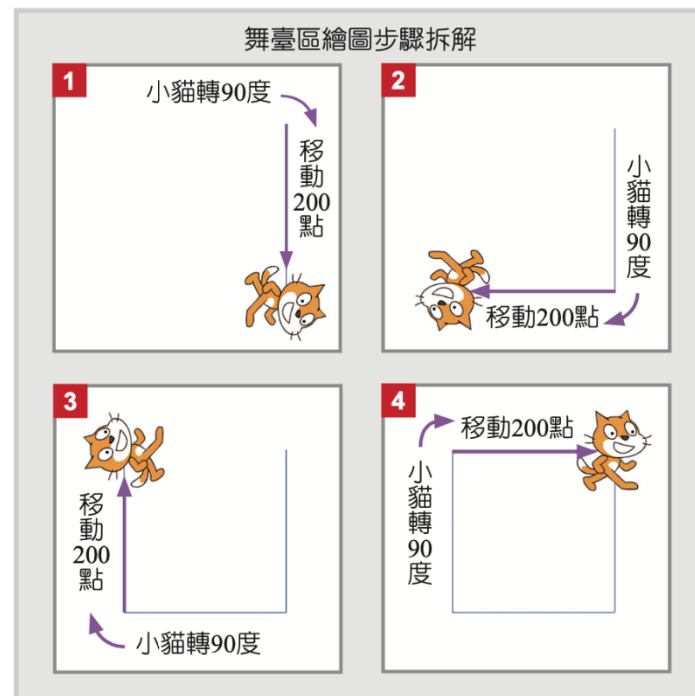


利用計次式迴圈讓小貓畫出一個正方形

步驟:設定計次式迴圈，取代重複的部份。



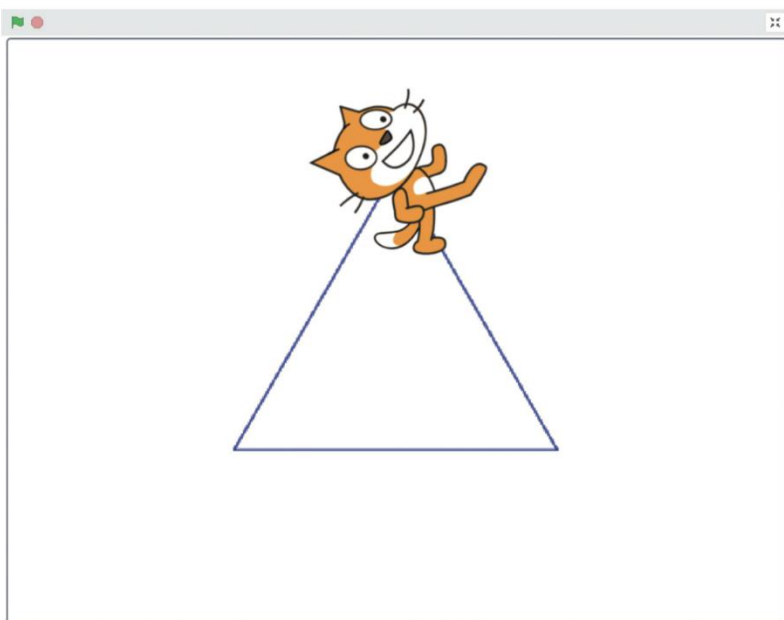
重複執行
4次，如右
圖的①～④



課堂練習



利用計次式迴圈讓小貓畫出一個三方形



Scratch-繪圖篇



畫正
方形

擴散
方形

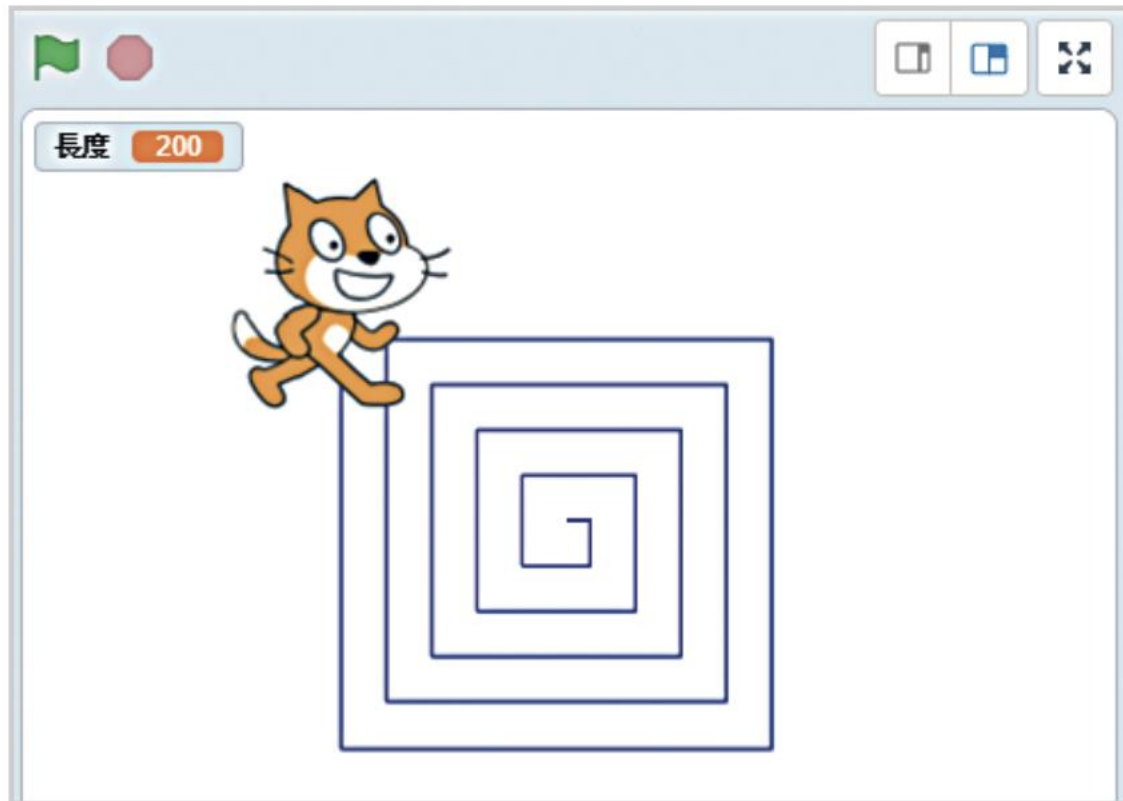
旋轉
方形



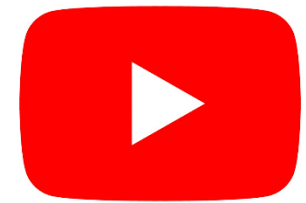
範例-畫擴散方形



畫擴散方形，每次移動的距離都增加，並轉 90 度。想一想，有沒有比較快的方法



用循序結構畫擴散方形

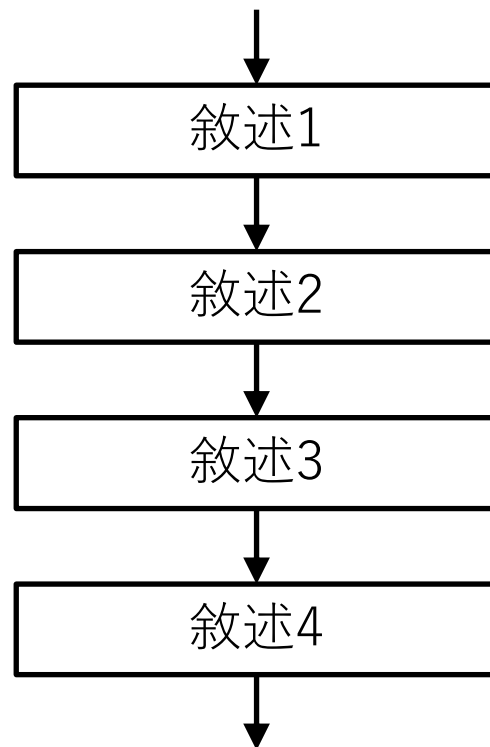


利用循序結構畫出一個擴散的方形

循序結構－流程圖

影片

[Scratch 繪圖篇
- 畫擴散方形](#)



用循序結構畫擴散方形



問題分析

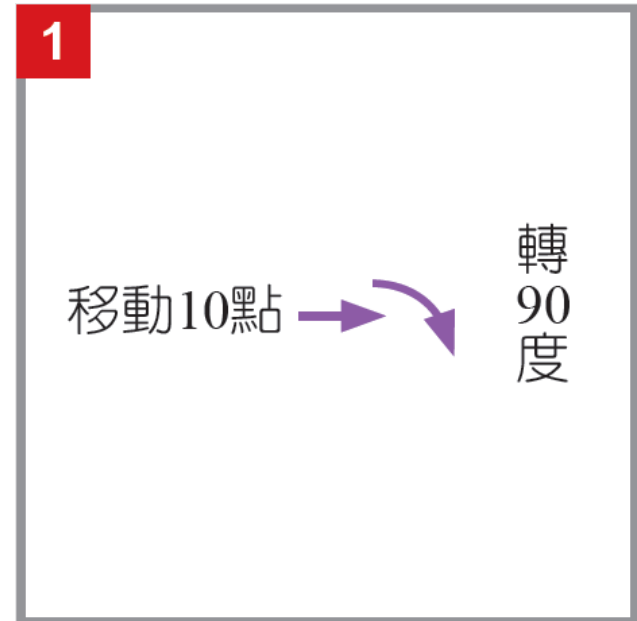
問題解析	問題實作
(A) 如何控制角色移動的距離？	輸入往前移動幾個點，逐漸增加移動的距離。  A blue Scratch 'Move' block with the text '移動 10 點'. The number '10' is circled in red.
(B) 如何控制角色的轉向？	輸入旋轉角度，向不同方向旋轉。  A blue Scratch 'Turn' block with the text '右轉 90 度' and a right-turn arrow icon. The number '90' is circled in red. To the right of the block is a small orange cat sprite.

用循序結構畫擴散方形



利用循序結構畫出一個擴散的方形

步驟:設定前進方向與移動距離



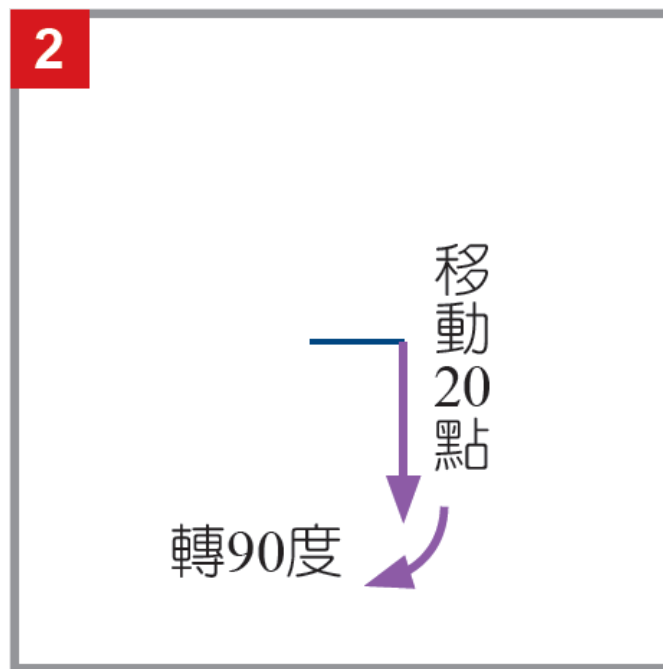
繪圖步驟拆解

用循序結構畫擴散方形



利用循序結構畫出一個擴散的方形

步驟:設定前進方向與移動距離



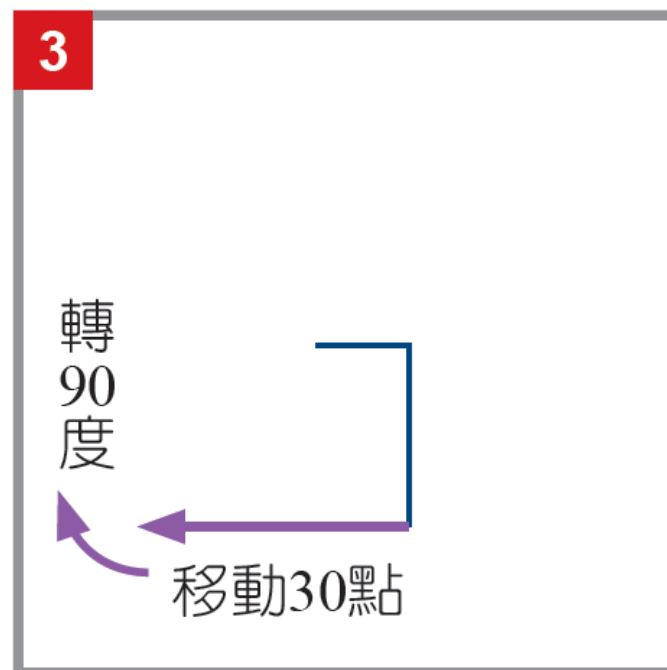
繪圖步驟拆解

用循序結構畫擴散方形



利用循序結構畫出一個擴散的方形

步驟:設定前進方向與移動距離



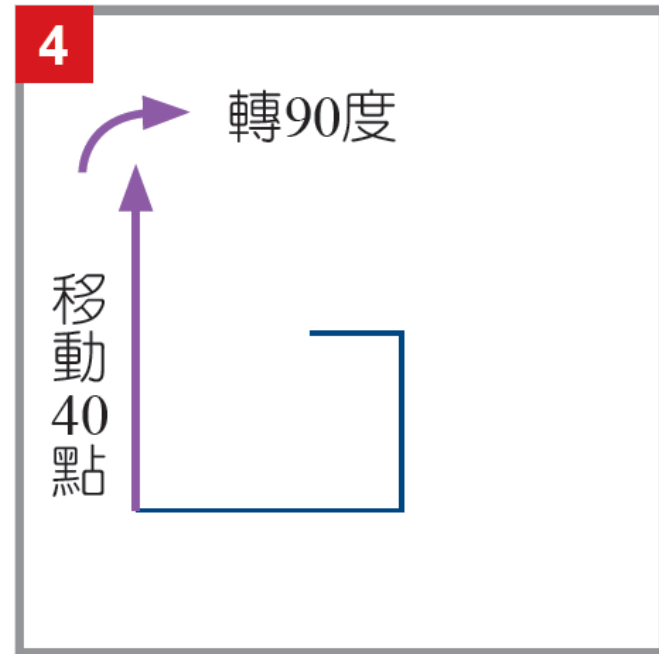
繪圖步驟拆解

用循序結構畫擴散方形



利用循序結構畫出一個擴散的方形

步驟:設定前進方向與移動距離



繪圖步驟拆解

用循序結構畫擴散方形



利用循序結構畫出一個擴散的方形

步驟:設定前進方向與移動距離

5

當 被點擊

筆跡全部清除

定位到 x: 0 y: 0

面朝 90 度

下筆

...

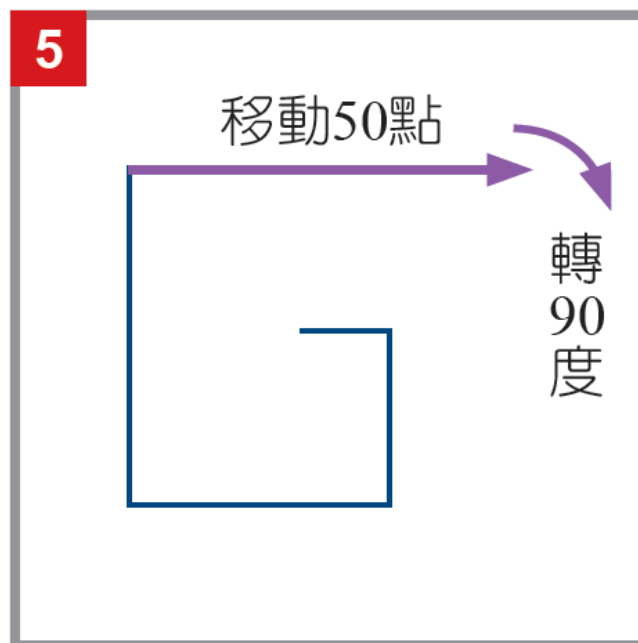
移動 50 點

右轉 90 度

等待 1 秒

停筆

出現第5次
移動50點



繪圖步驟拆解

用循序結構畫擴散方形



利用循序結構畫出一個擴散的方形

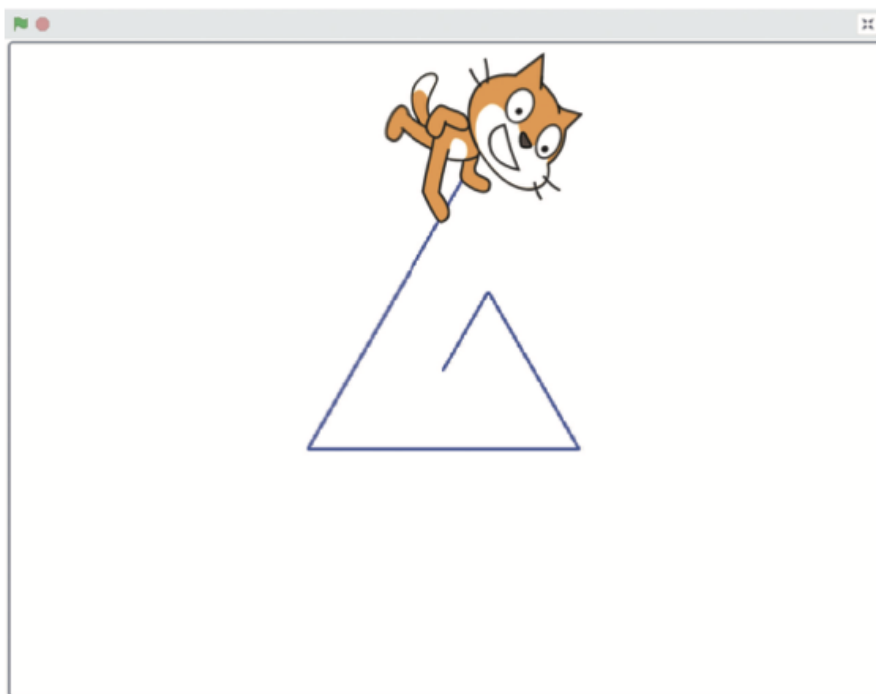
這麼多重複的積木怎麼利用變數簡化呢？



課堂練習



利用**循序結構**畫出一個擴散的三角形



```
當 被點擊  
筆跡全部清除  
定位到 x: 0 y: 0  
面朝 30 度  
下筆  
移動 50 點  
右轉 120 度  
等待 1 秒  
移動 100 點  
右轉 120 度  
等待 1 秒  
移動 150 點  
右轉 120 度  
等待 1 秒  
移動 200 點  
右轉 120 度  
等待 1 秒  
停筆
```


計次式迴圈與變數的應用



利用計次式迴圈與變數畫出一個擴散的方形

步驟1: 設定新變數

1. 點選變數
2. 按建立一個變數



計次式迴圈與變數的應用



利用計次式迴圈與變數畫出一個擴散的方形

步驟2:選擇名稱與適用範圍

3. 跳出新的變數視窗

4. 輸入變數名稱:長度

5. 選擇適用範圍:所有角色

6. 按確定

The screenshot shows a dialog box titled "新的變數" (New Variable) with a close button (X) in the top right corner. The dialog contains the following elements:

- ③ A blue header bar with the title "新的變數" and a close button.
- "新變數的名稱" (New Variable Name) label above a text input field.
- ④ The text input field containing the character "長度" (Length).
- ⑤ Two radio button options: "適用於所有角色" (Apply to all roles) which is selected, and "僅適用當前角色" (Apply only to current role).
- ⑥ Two buttons at the bottom: "取消" (Cancel) and "確定" (Confirm).

計次式迴圈與變數的應用



利用計次式迴圈與變數畫出一個擴散的方形

步驟3: 產生變數積木群組

7. 產生5個變數積木



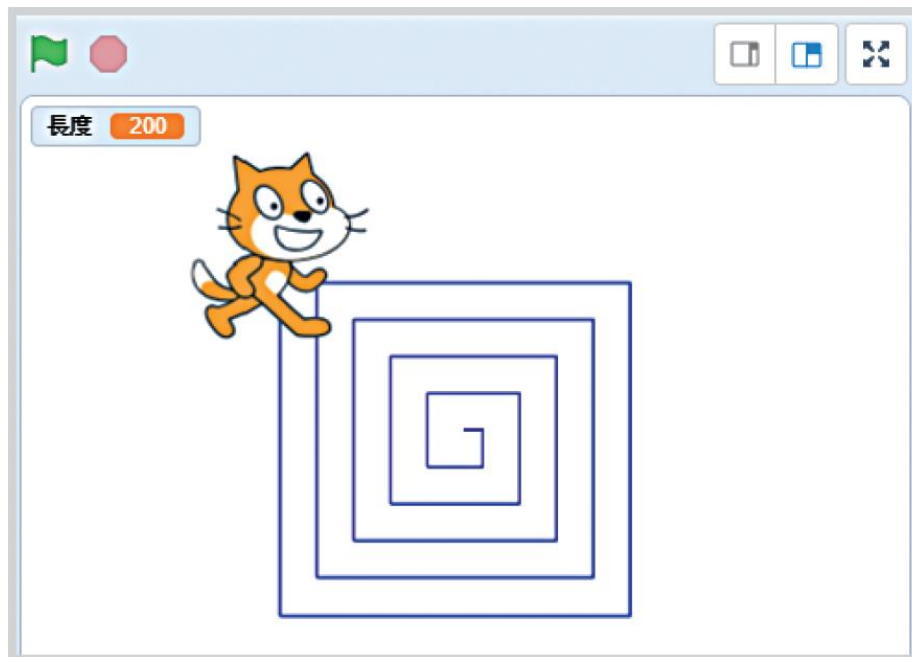
計次式迴圈與變數的應用



利用計次式迴圈與變數畫出一個擴散的方形

步驟4: 迴圈積木撰寫程式




加入變數積木，
再利用迴圈積木
取代上一個範例
中重複的部分。



計次式迴圈與變數的應用



問題分析

問題解析	問題實作
(A) 如何設定變數的初始值？	輸入變數一開始的數值 
(B) 如何改變變數的數值？	輸入每次執行到此積木時，要重新儲存到變數的數 
(C) 如何改變每次移動的距離？	用變數取代固定數值，改變每次移動的距離。 

計次式迴圈與變數的應用



利用計次式迴圈與變數畫出一個擴散的方形

步驟4: 迴圈積木撰寫程式



一開始，變數中
沒有任何數值



長度

設定變數初始值



長度

= 0 , 儲存後數值為



長度

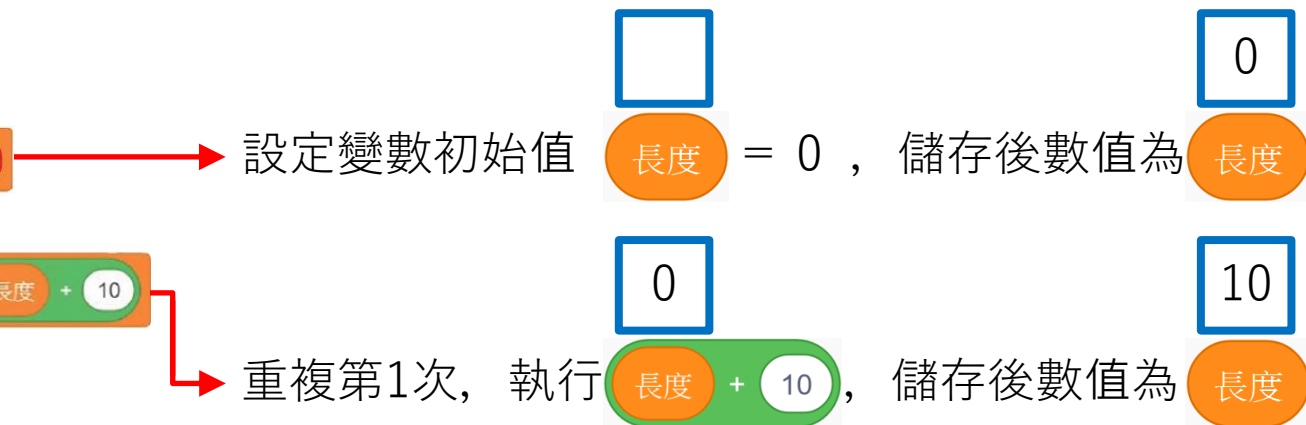
計次式迴圈與變數的應用



利用計次式迴圈與變數畫出一個擴散的方形

步驟4: 迴圈積木撰寫程式

```
當被點擊  
筆跡全部清除  
定位到 x: 0 y: 0  
面朝 90 度  
下筆  
變數 長度 設為 0  
重複 20 次  
  變數 長度 設為 長度 + 10  
  移動 長度 點  
  右轉 90 度  
  等待 0.5 秒  
停筆
```



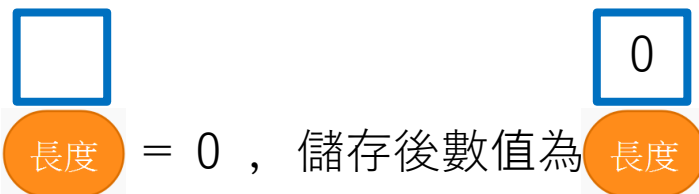
計次式迴圈與變數的應用



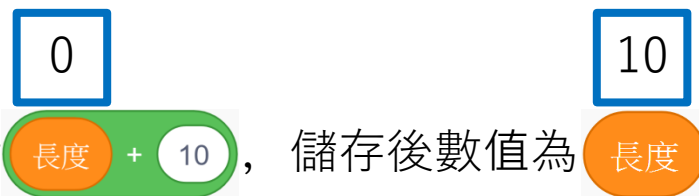
利用計次式迴圈與變數畫出一個擴散的方形

步驟4: 迴圈積木撰寫程式

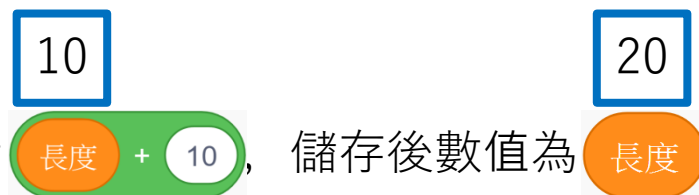
設定變數初始值



重複第1次, 執行



重複第2次, 執行



計次式迴圈與變數的應用



利用計次式迴圈與變數畫出一個擴散的方形

步驟4: 迴圈積木撰寫程式



設定變數初始值



長度

= 0 , 儲存後數值為



0

長度

。

重複第3次, 執行



20

長度

+ 10

, 儲存後數值為



30

長度

。

計次式迴圈與變數的應用

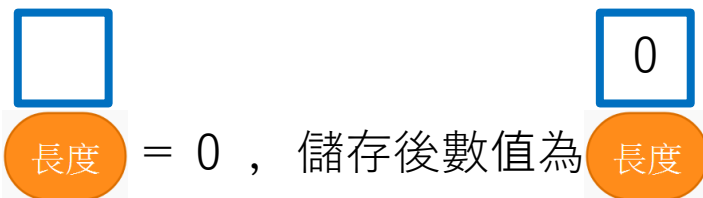


利用計次式迴圈與變數畫出一個擴散的方形

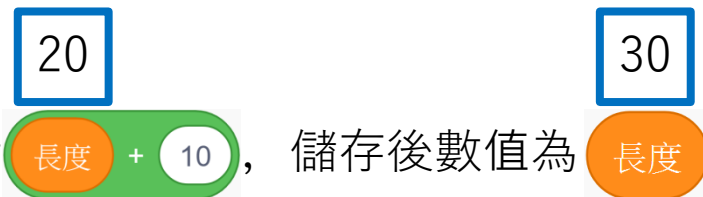
步驟4: 迴圈積木撰寫程式



設定變數初始值



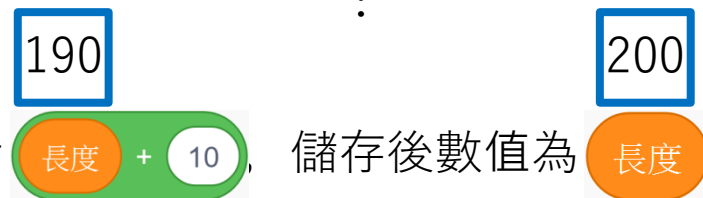
重複第3次, 執行



⋮ 不斷重複計算

⋮ 隨時指定新數值

重複第20次, 執行

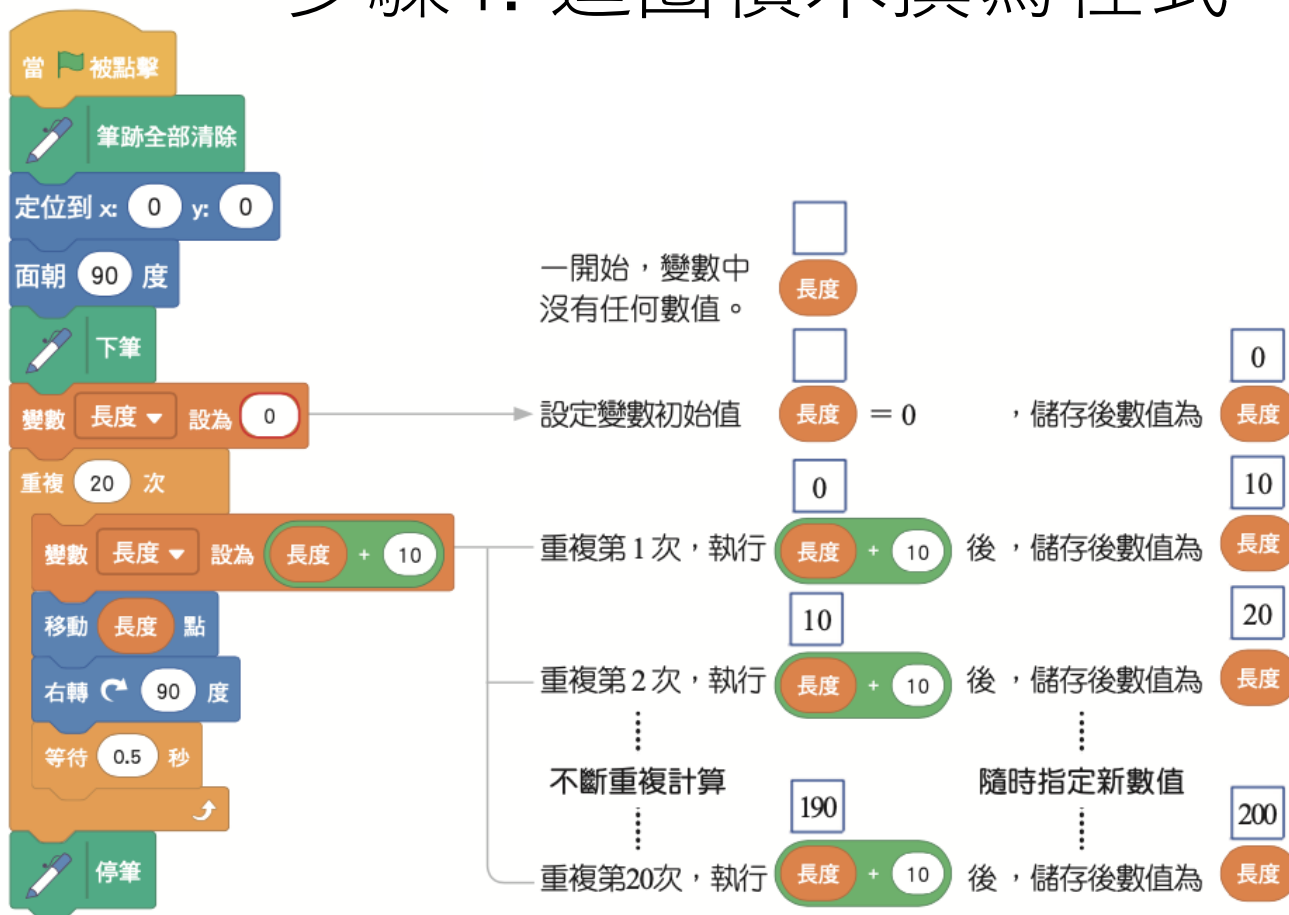


計次式迴圈與變數的應用



利用計次式迴圈與變數畫出一個擴散的方形

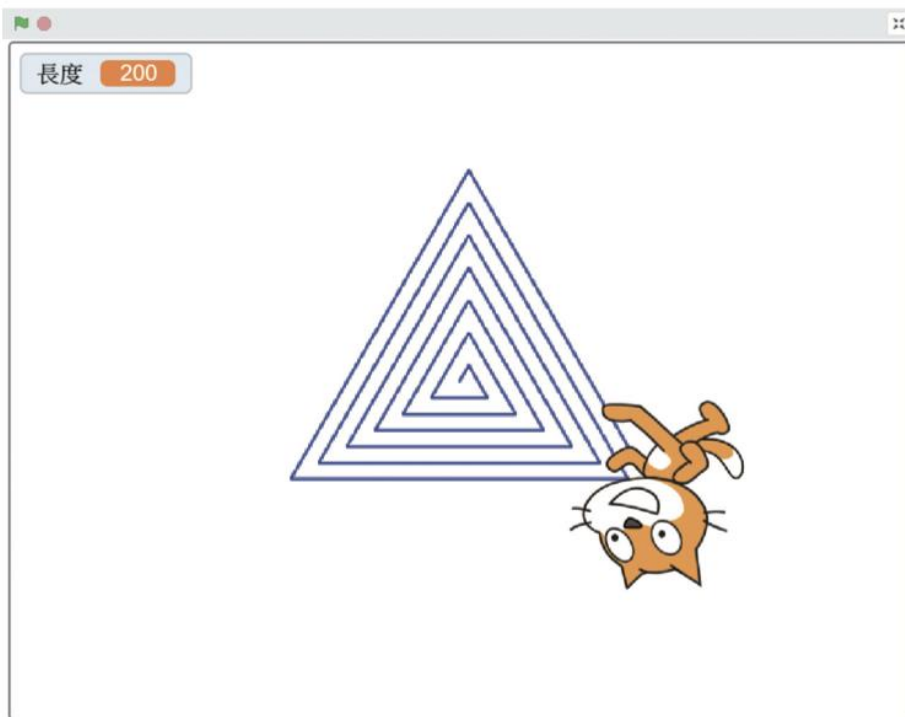
步驟4: 迴圈積木撰寫程式



課堂練習



用計次式迴圈與變數畫出一個擴散的三角形



```
當 被點擊
筆跡全部清除
定位到 x: 0 y: 0
面朝 30 度
下筆
變數 長度 設為 0
重複 20 次
  變數 長度 設為 長度 + 10
  移動 長度 點
  右轉 120 度
  等待 0.5 秒
停筆
```



Scratch-繪圖篇



畫正
方形

擴散
方形

旋轉
方形

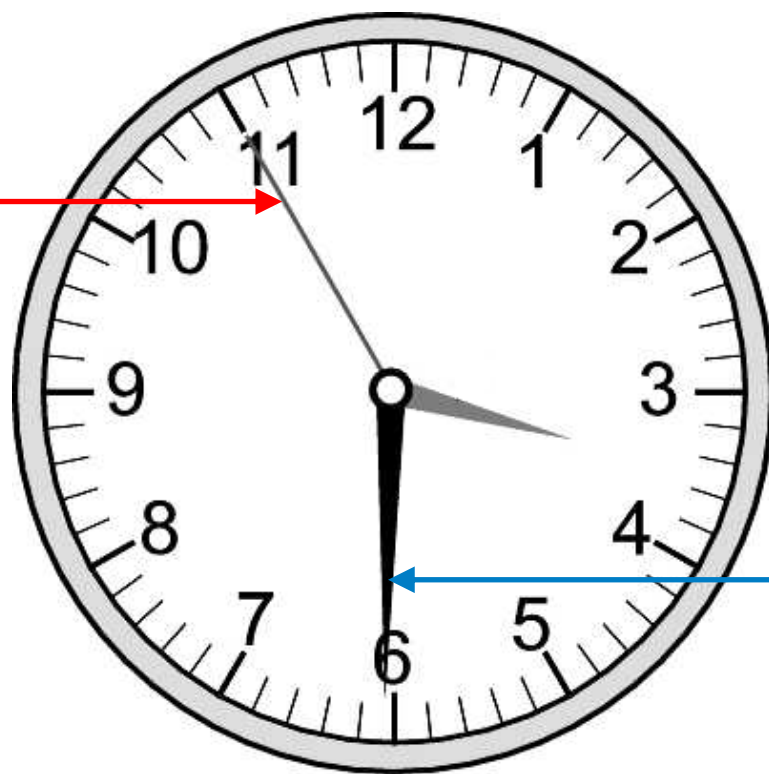


巢狀結構畫旋轉的正方形



利用巢狀結構畫出12個旋轉的正方形

巢狀結構是結構裡面還有結構。原理就像
時鐘的秒針和分針



秒針轉一圈，
分針才會加1，
秒針等同於
內層迴圈

分針要等秒針轉一
圈後，才會加1，分
針等同於外層迴圈

巢狀結構畫旋轉的正方形



利用巢狀結構畫出12個旋轉的正方形

想想看

如果每次都先畫一個正方形後，再旋轉 30 度，要如何用迴圈進行繪圖呢？

巢狀結構畫旋轉的正方形



利用巢狀結構畫出12個旋轉的正方形

影片

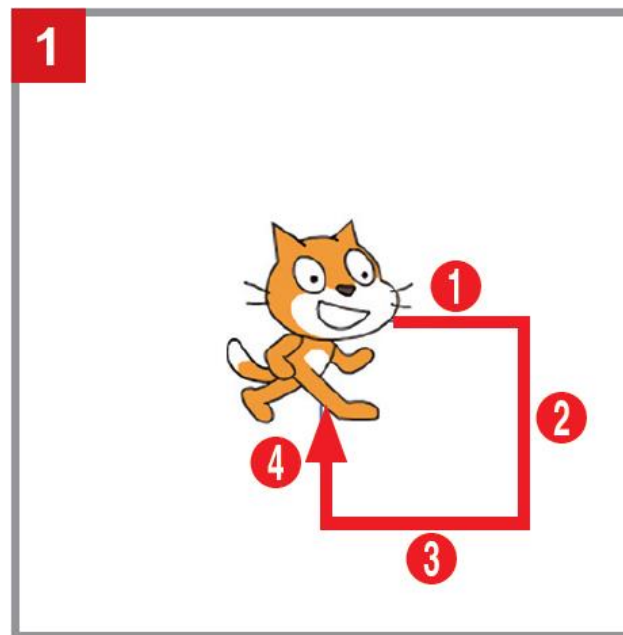
[Scratch繪圖篇 - 畫旋轉正方形](#)



內層迴圈重複執行4次，對照右圖

① ~ ④。

畫邊長 100 點正方形

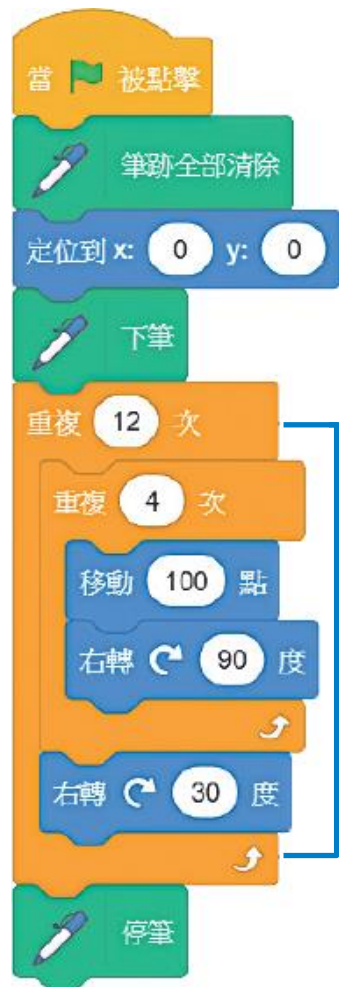


繪圖步驟拆解

巢狀結構畫旋轉的正方形

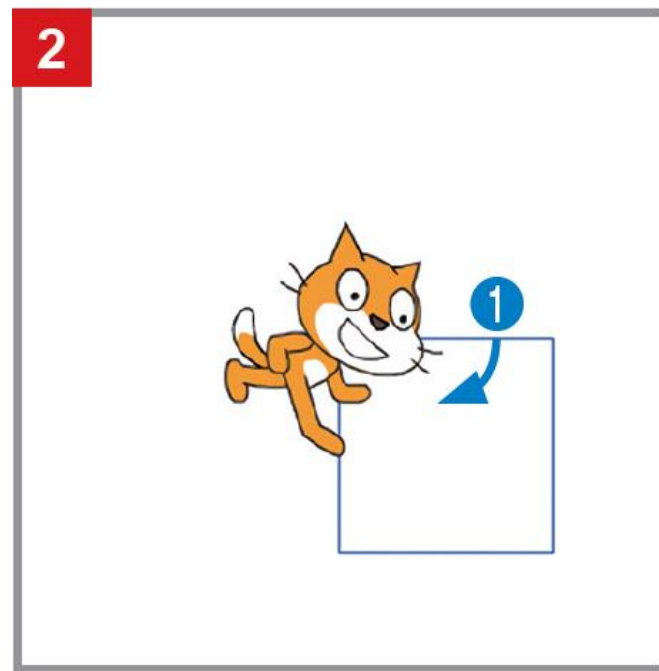


利用**巢狀結構**畫出12個旋轉的正方形



外層迴圈重複1次，內層迴圈就會重複執行4次，對照右圖的①。

小貓轉30度



繪圖步驟拆解

範例-畫旋轉的正方形



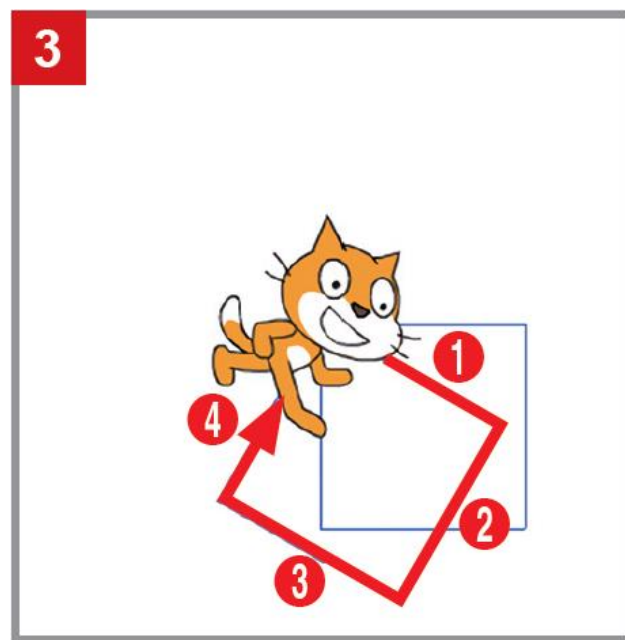
利用**巢狀結構**畫出12個旋轉的正方形。



內層迴圈重複執行4次，對照右圖

① ~ ④。

畫邊長 100 點正方形

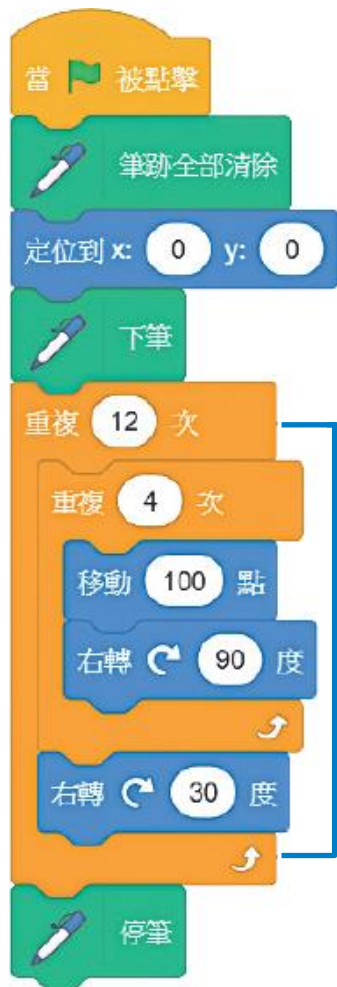


繪圖步驟拆解

範例-畫旋轉的正方形

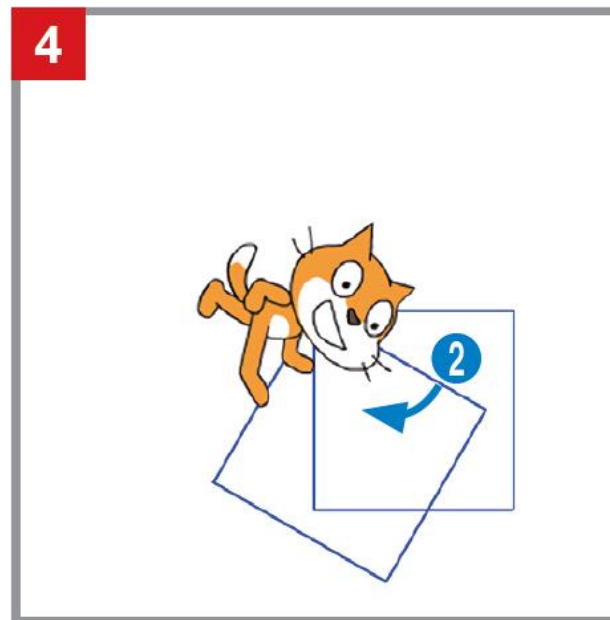


利用**巢狀結構**畫出12個旋轉的正方形。



外層迴圈重複第2次，內層迴圈就會重複執行4次，對照右圖的②。

小貓轉30度



繪圖步驟拆解

範例-畫旋轉的正方形

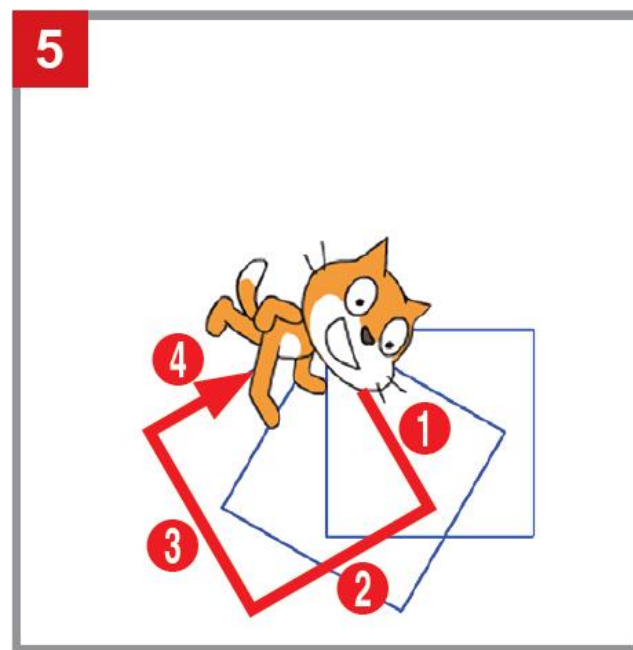


利用**巢狀結構**畫出12個旋轉的正方形。



內層迴圈重複執行4次，對照右圖
① ~ ④。

畫邊長 100 點正方形



繪圖步驟拆解

範例-畫旋轉的正方形

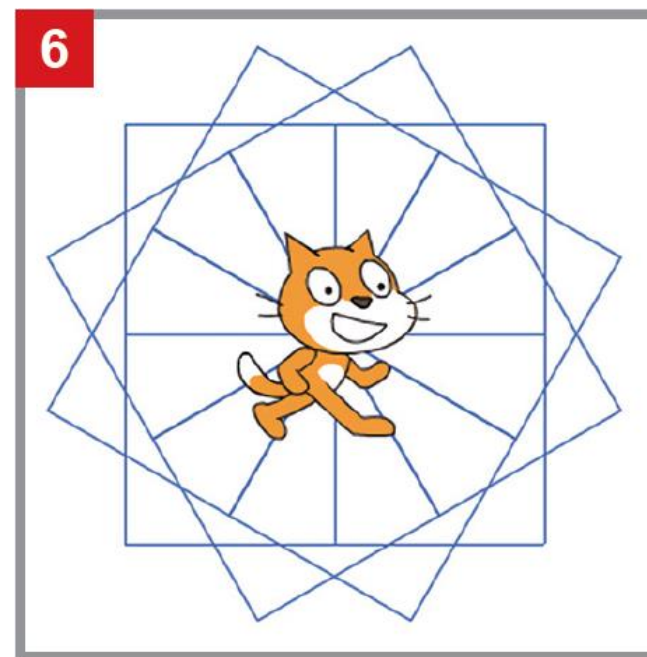


利用**巢狀結構**畫出12個旋轉的正方形。



外層迴圈每重複1次，內層迴圈就會重複執行4次，直到完成圖形，對照右圖。

最後完成圖形

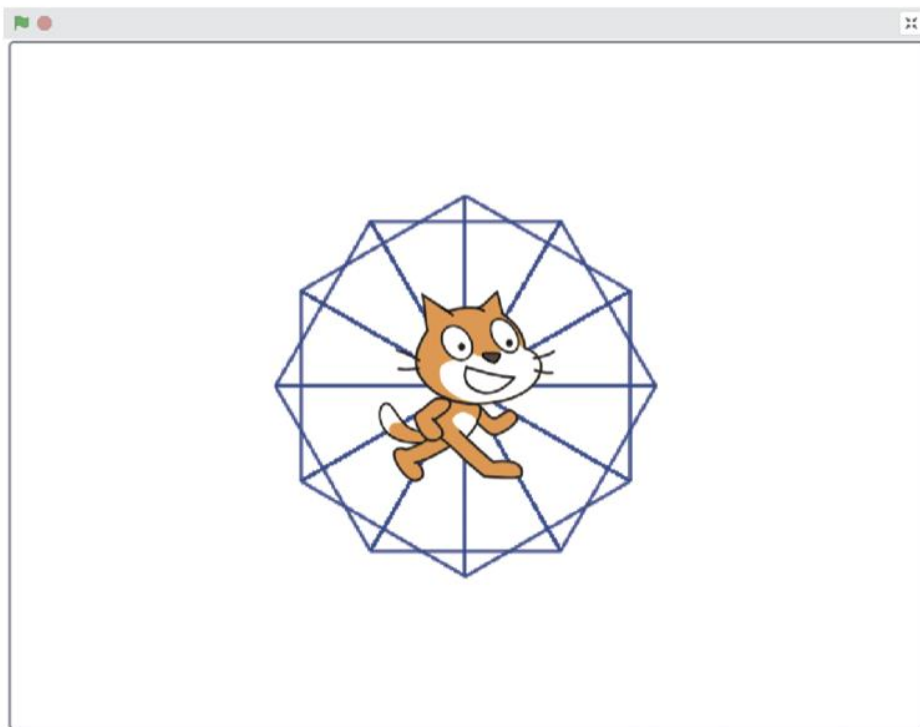


繪圖步驟拆解

課堂練習



利用**巢狀結構**畫出 12 個旋轉的正三角形



```
當 被點擊  
筆跡全部清除  
定位到 x: 0 y: 0  
下筆  
重複 12 次  
  重複 3 次  
    移動 100 點  
    右轉 120 度  
  右轉 30 度  
停筆
```


重點回顧



1. 演算法是一種**解決問題的方法**，
程式語言是**實踐演算法的工具**。
2. 每個人的思考模式不同，解決問題的方法也不同，所設計的演算法也會不同，但最重要的是執行時能產生**正確**的結果。
3. 雖然不同程式語言的語法不一樣，但**基本邏輯**類似。



你還想知道什麼？

