

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



畫正
方形

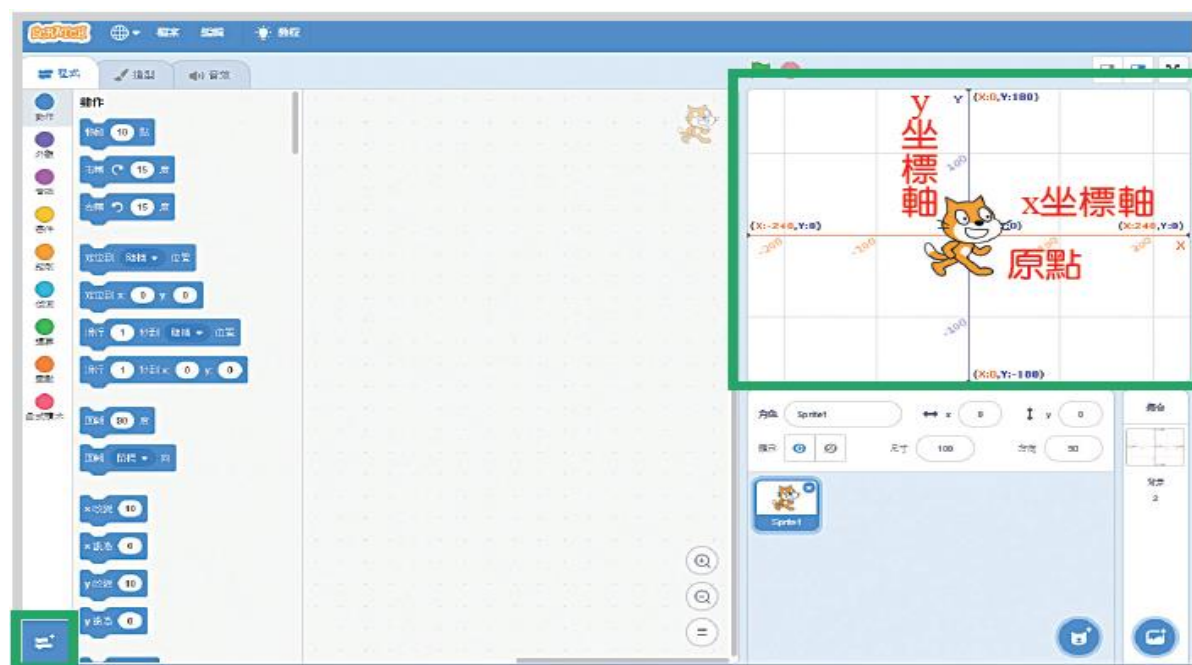
擴散
方形

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

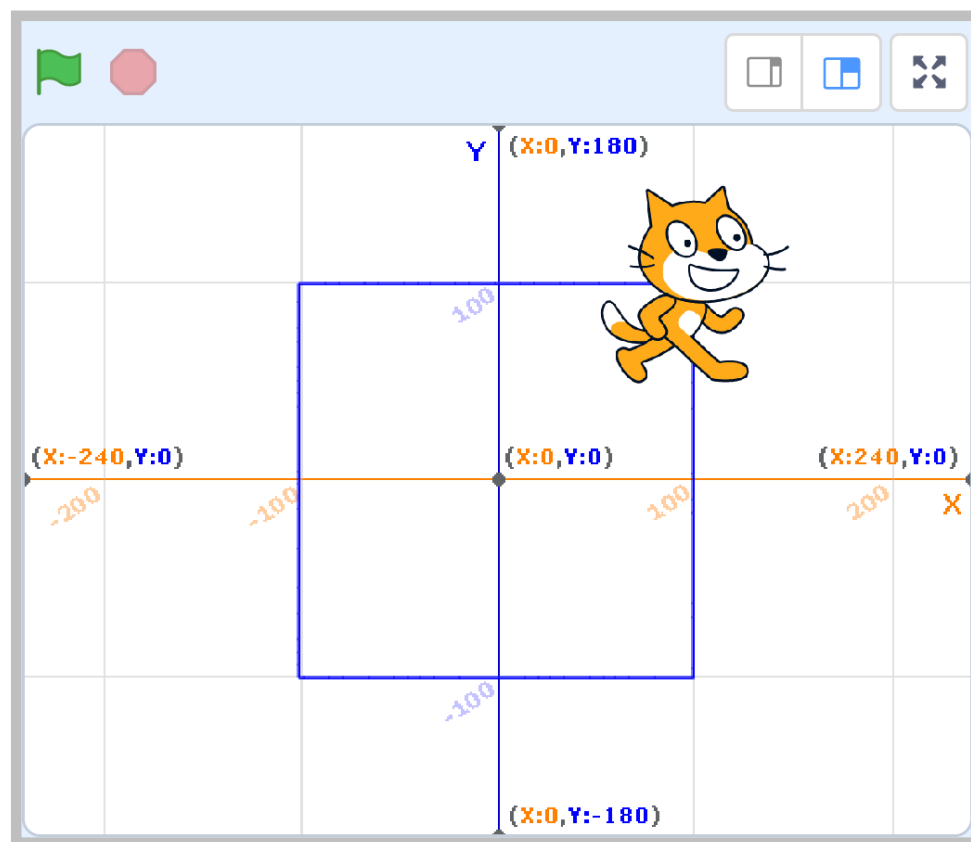
往右是正數，

往左是負數；

原點的y軸

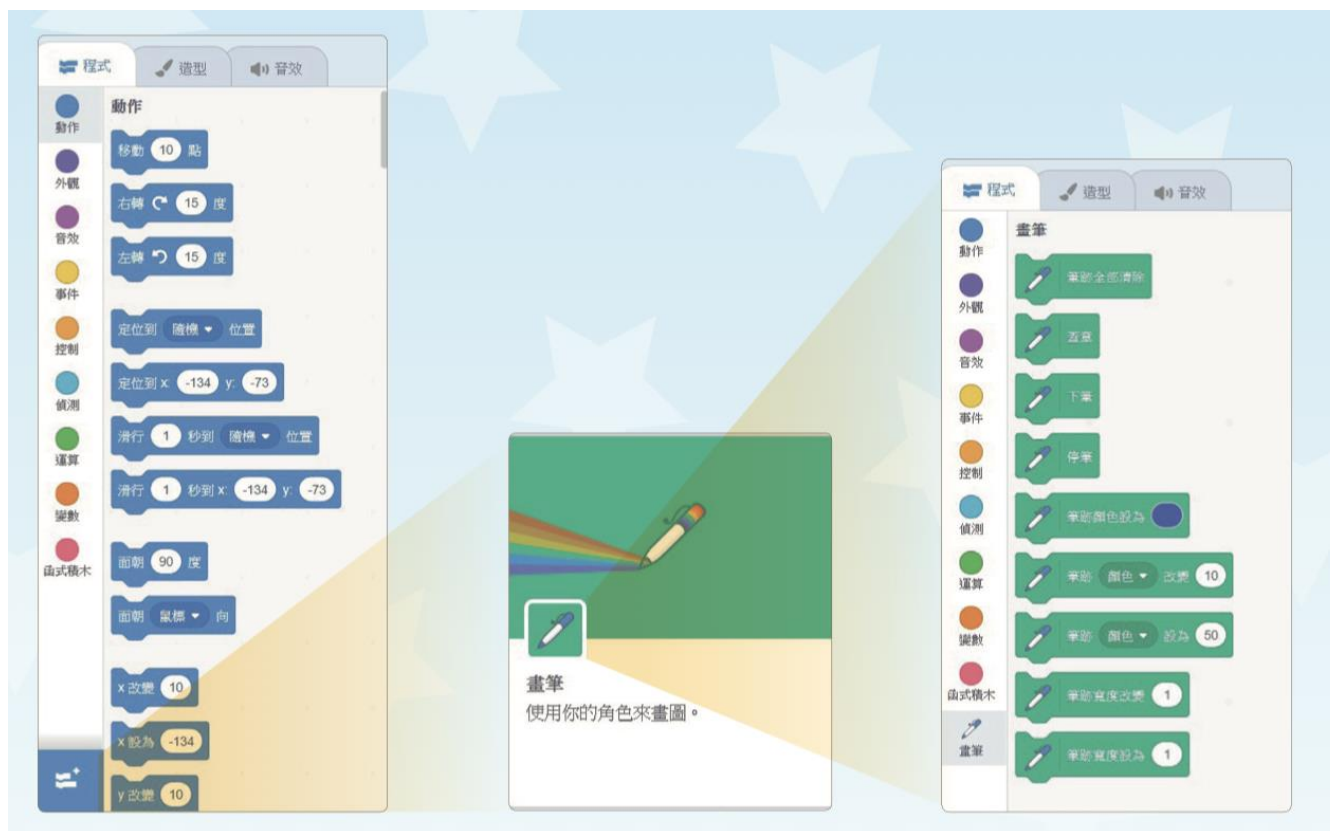
往上是正數，

往下是負數。





使用畫筆前需要先從**擴展功能**中新增相關積木

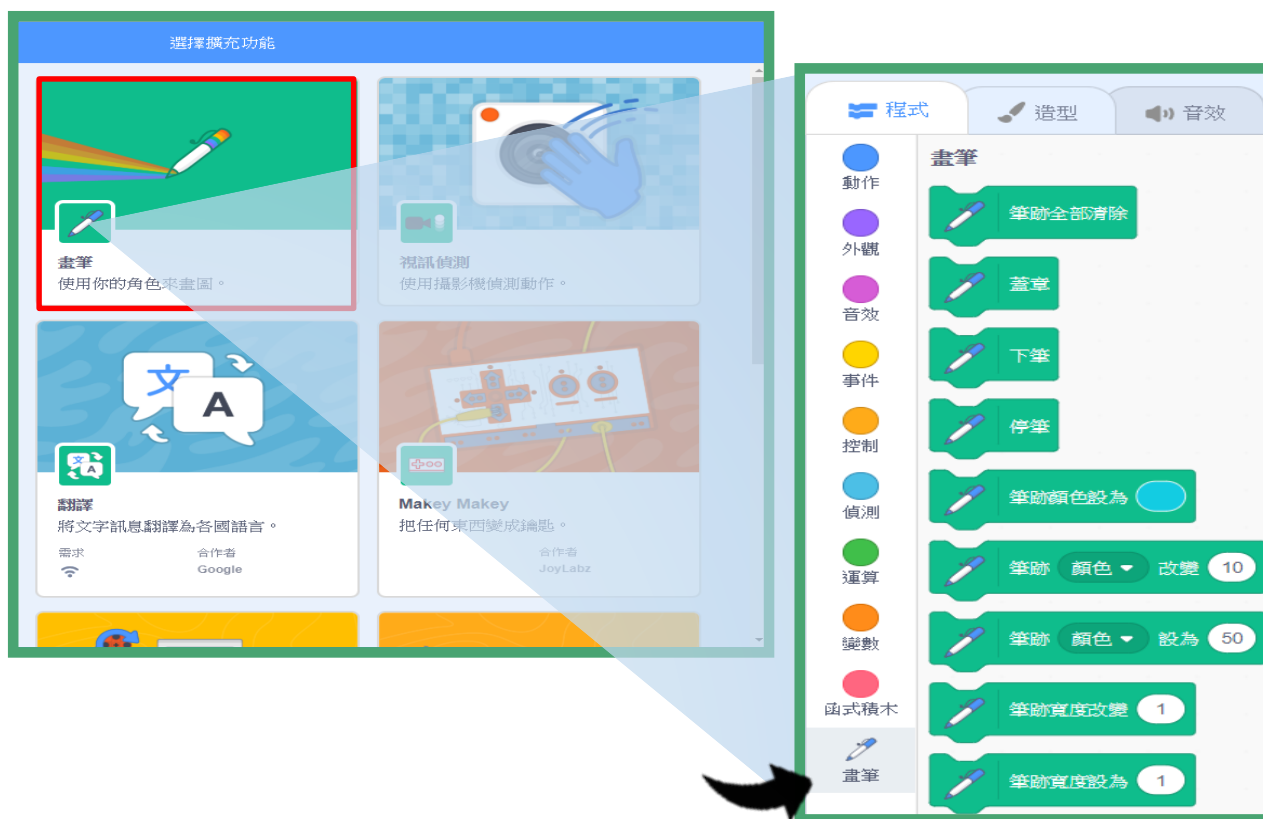


腳本區下方可以添加擴展功能



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。





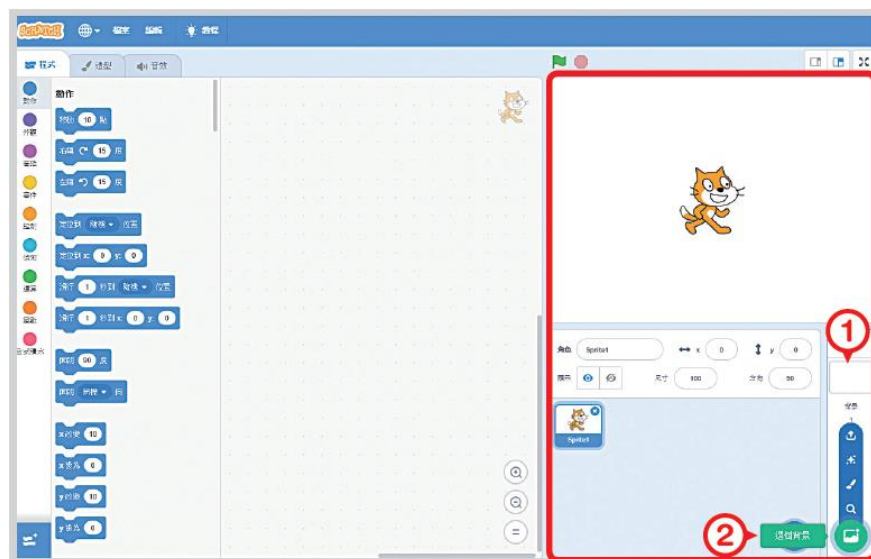
繪圖功能

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**





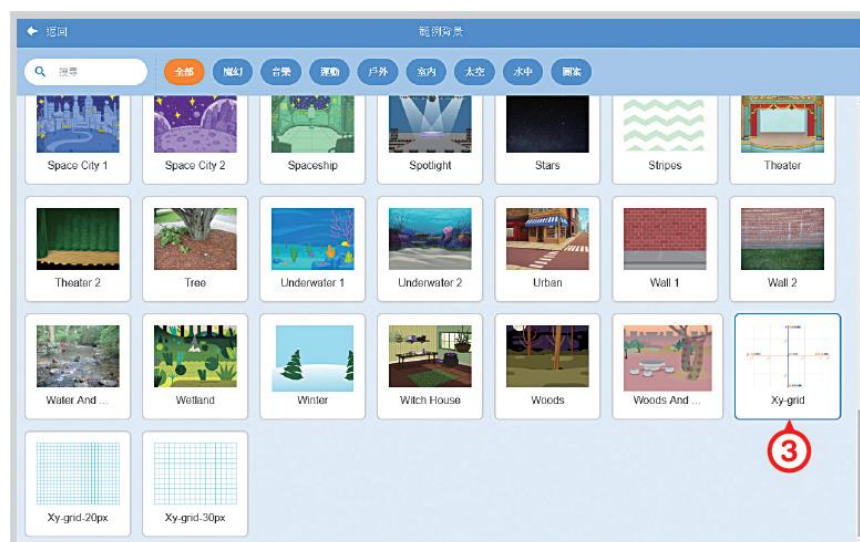
繪圖功能

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。

用坐標積木畫正方形

203

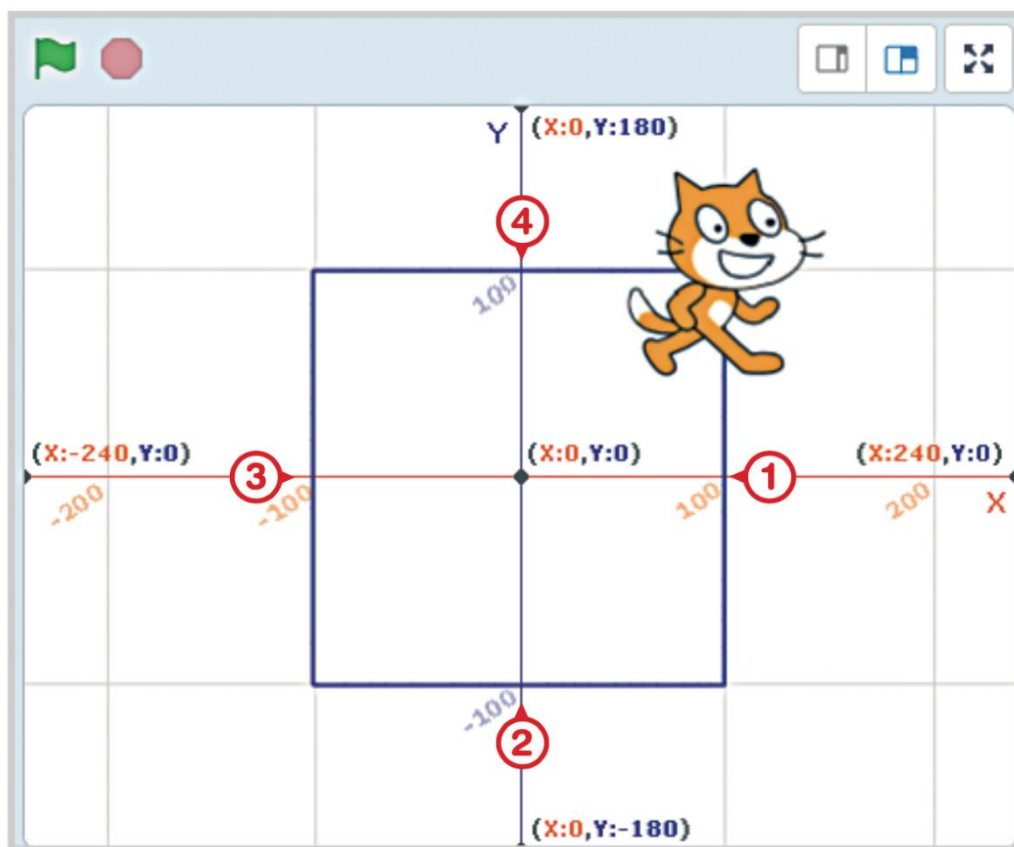


利用**坐標積木**讓小貓畫出一個正方形



影片

[Scratch繪圖篇
- 畫正方形](#)





範例1-畫正方形

利用坐標積木讓小貓畫出一個正方形

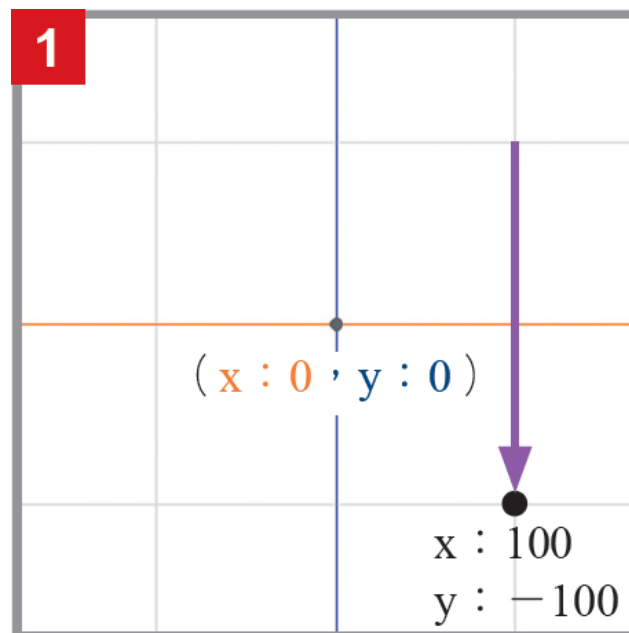
問題解析

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 



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

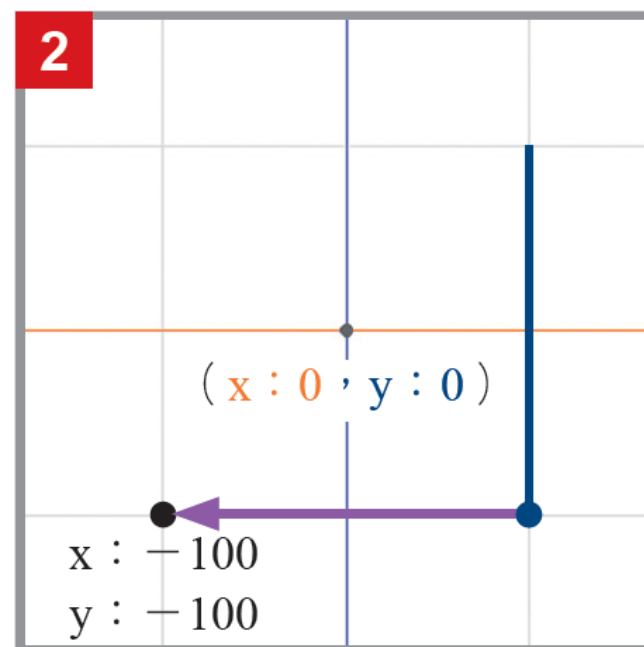
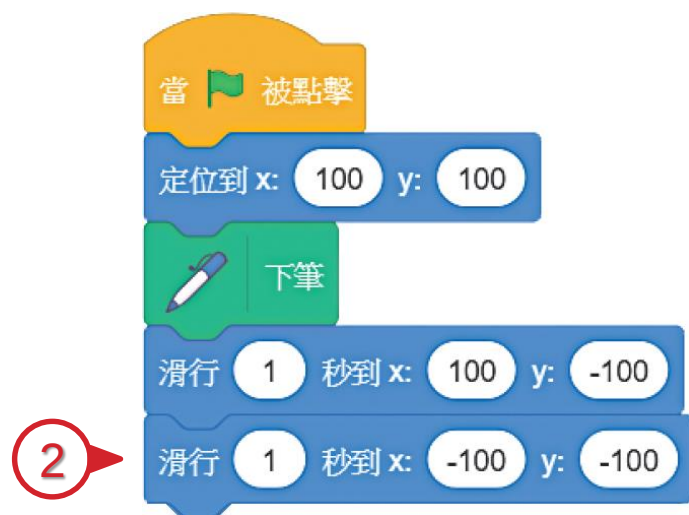


繪圖步驟拆解



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

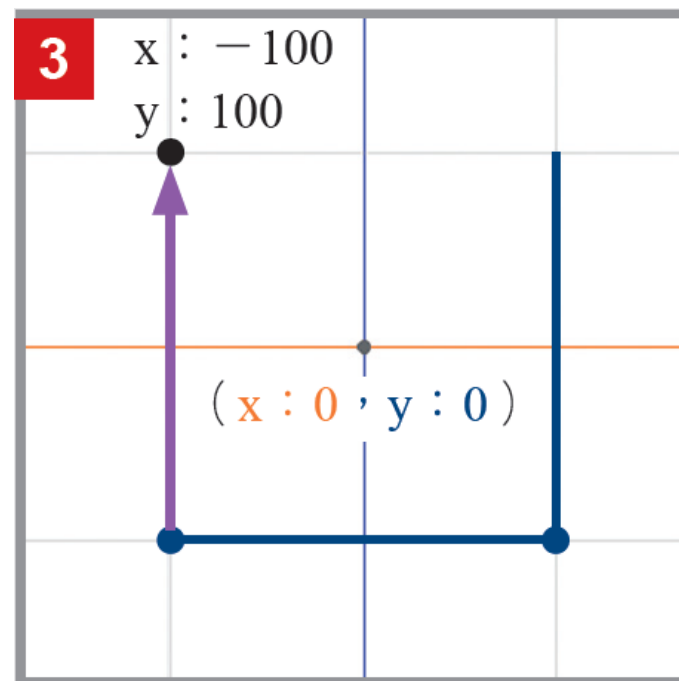
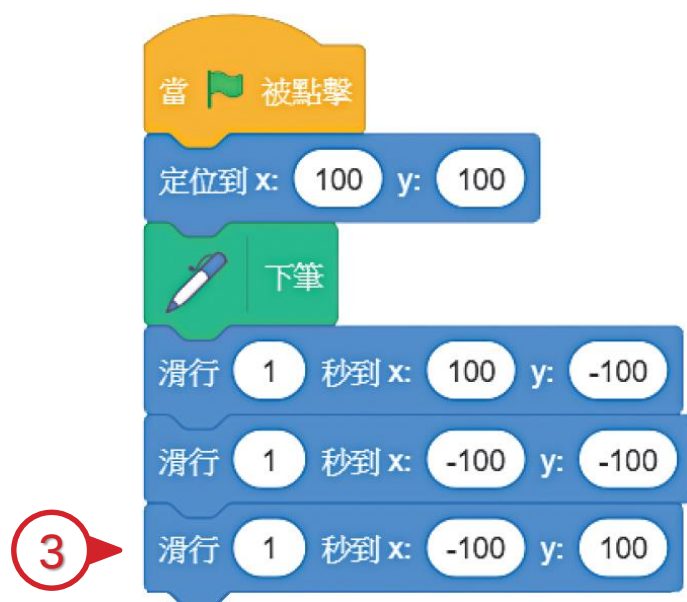


繪圖步驟拆解



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

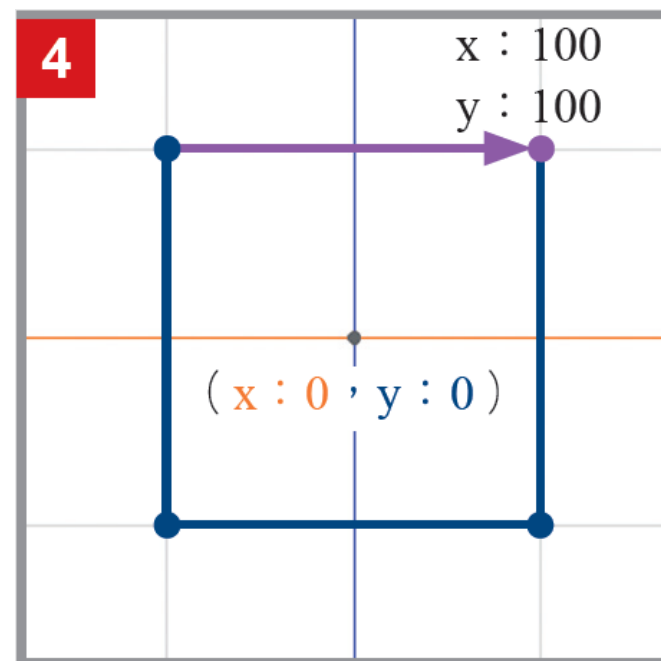


繪圖步驟拆解



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

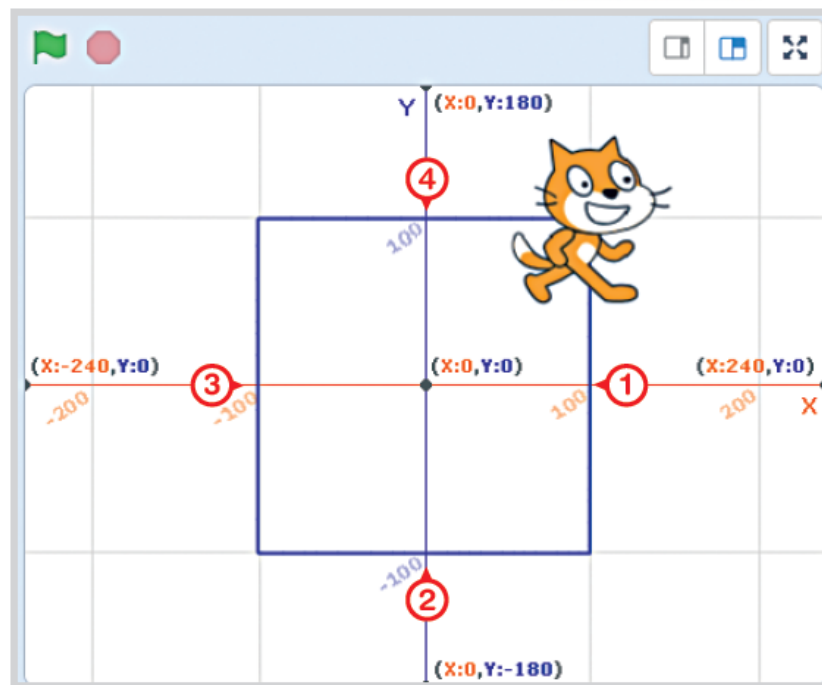


繪圖步驟拆解



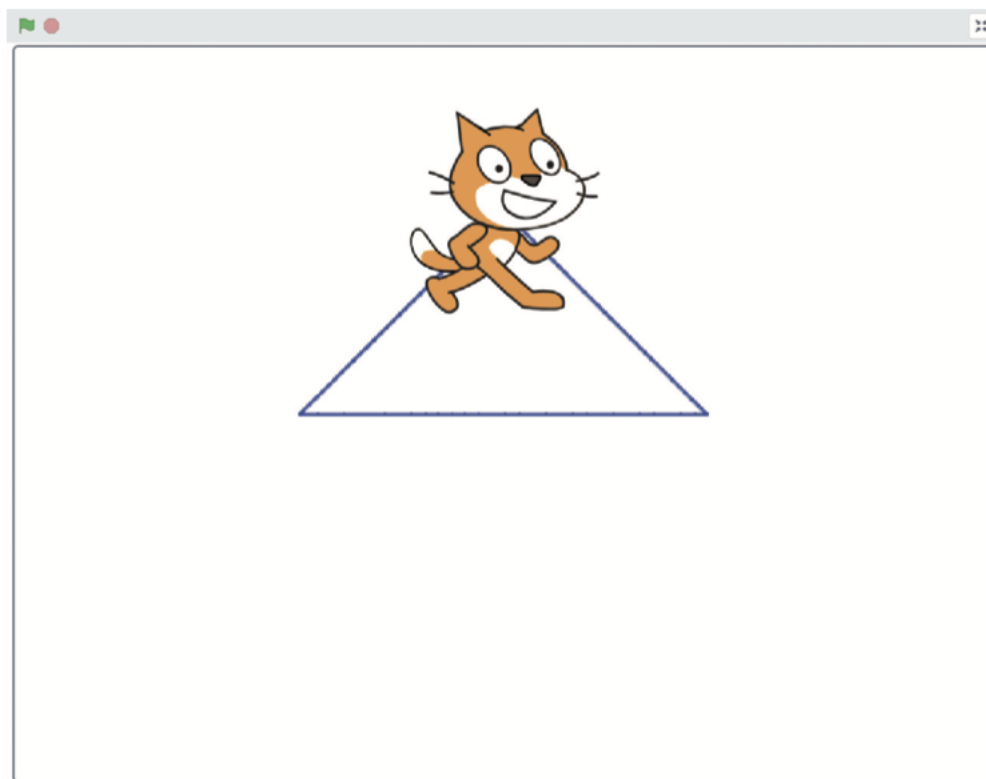
利用坐標積木讓小貓畫出一個正方形

完整程式





利用**坐標**積木讓小貓畫出一個三角形





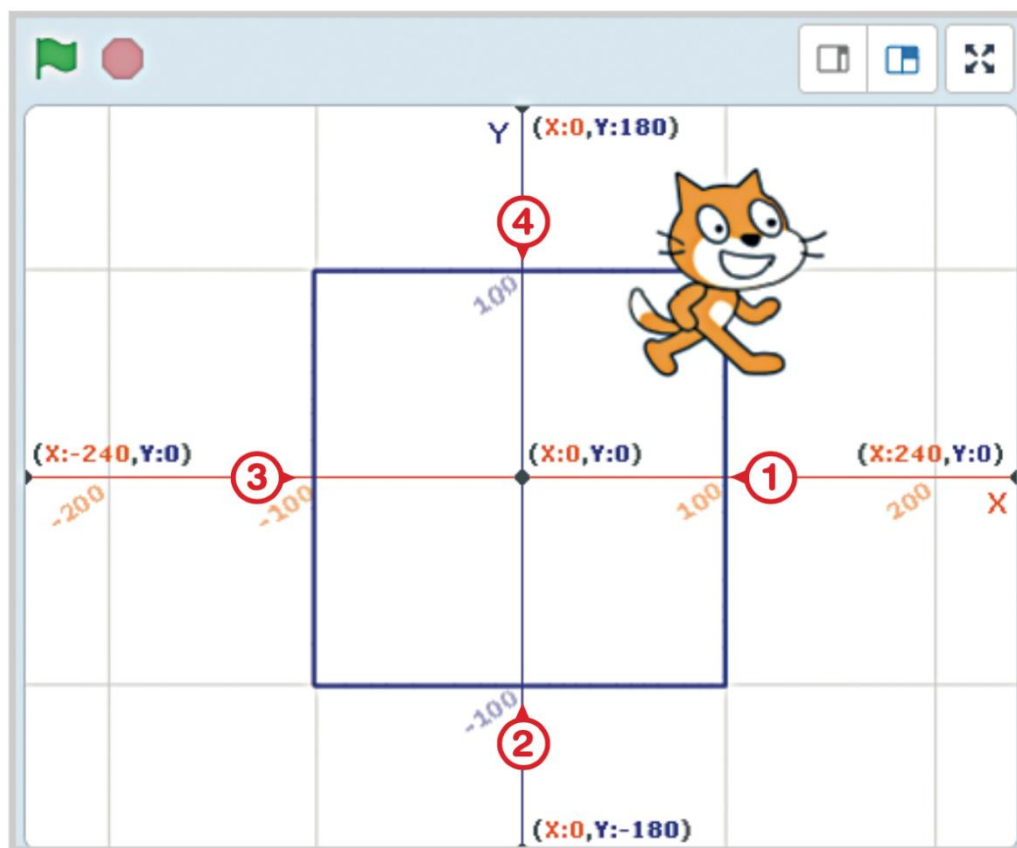
利用坐標積木讓小貓畫出一個三角形 撰寫程式





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





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。 

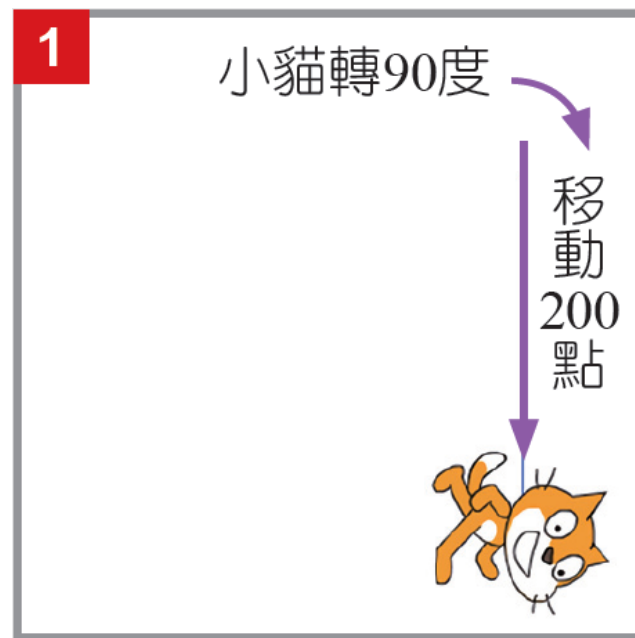
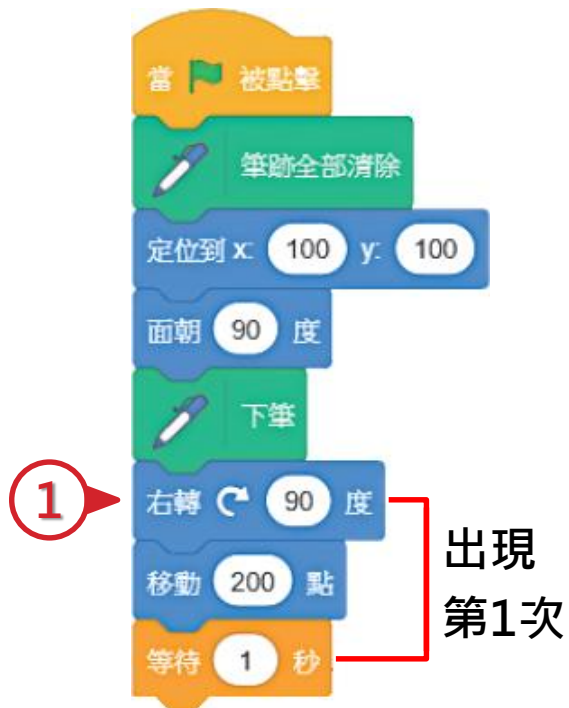
用方向積木畫正方形

204



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

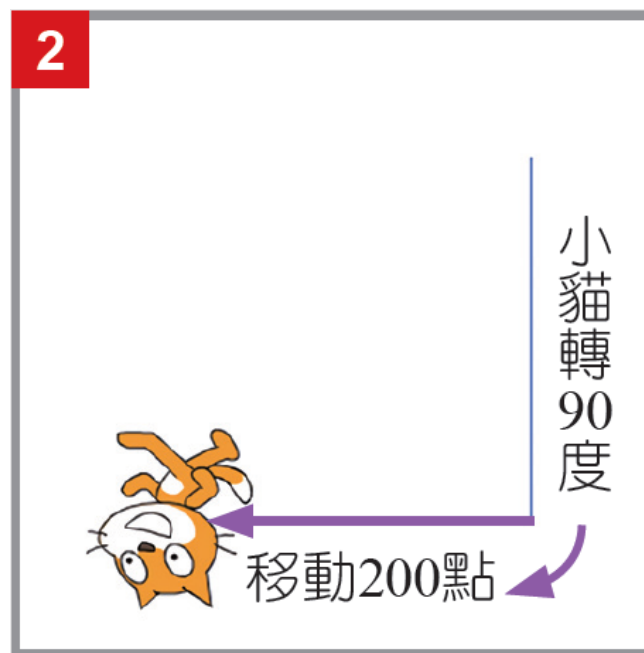
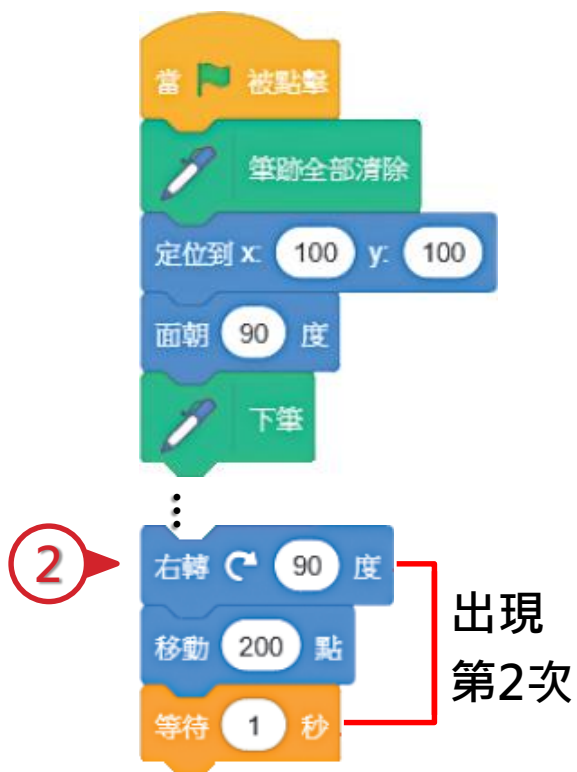
用方向積木畫正方形

204



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

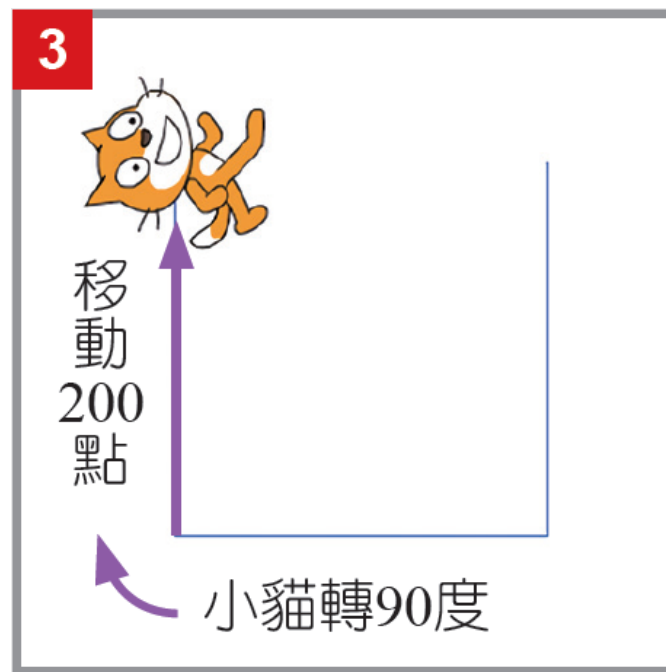
用方向積木畫正方形

204



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



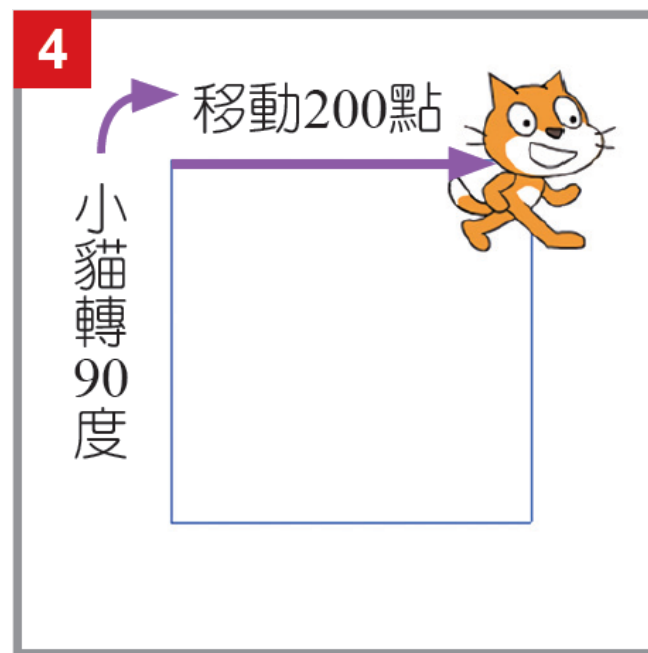
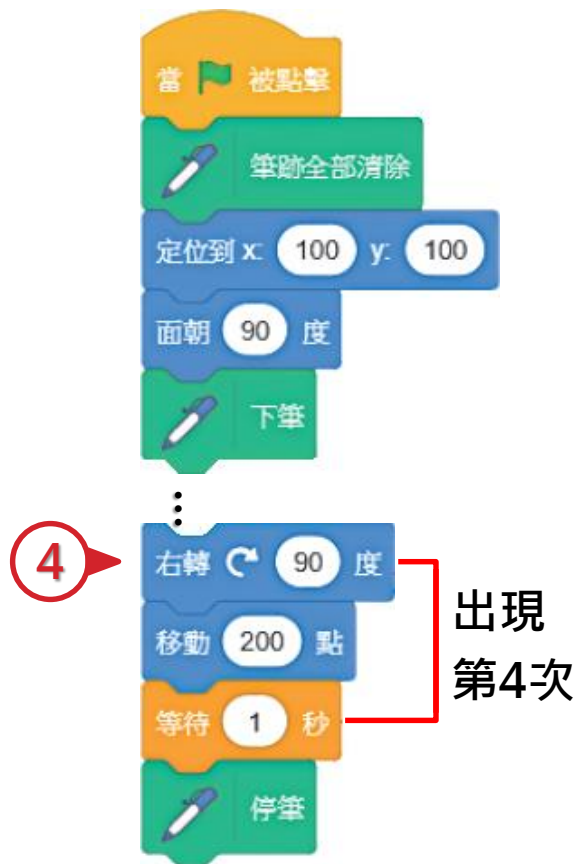
繪圖步驟拆解



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



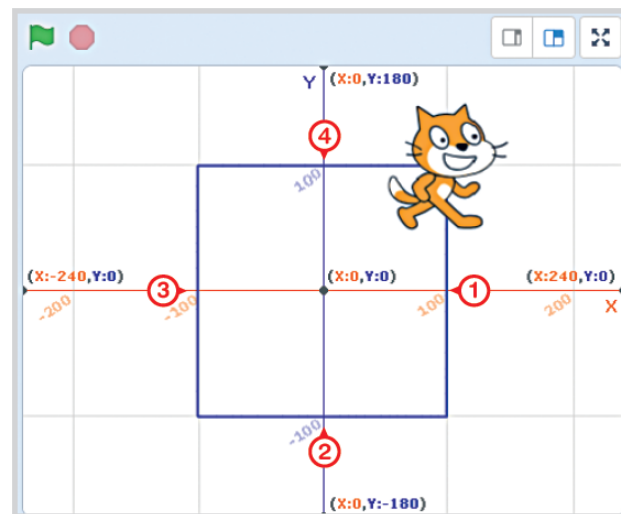
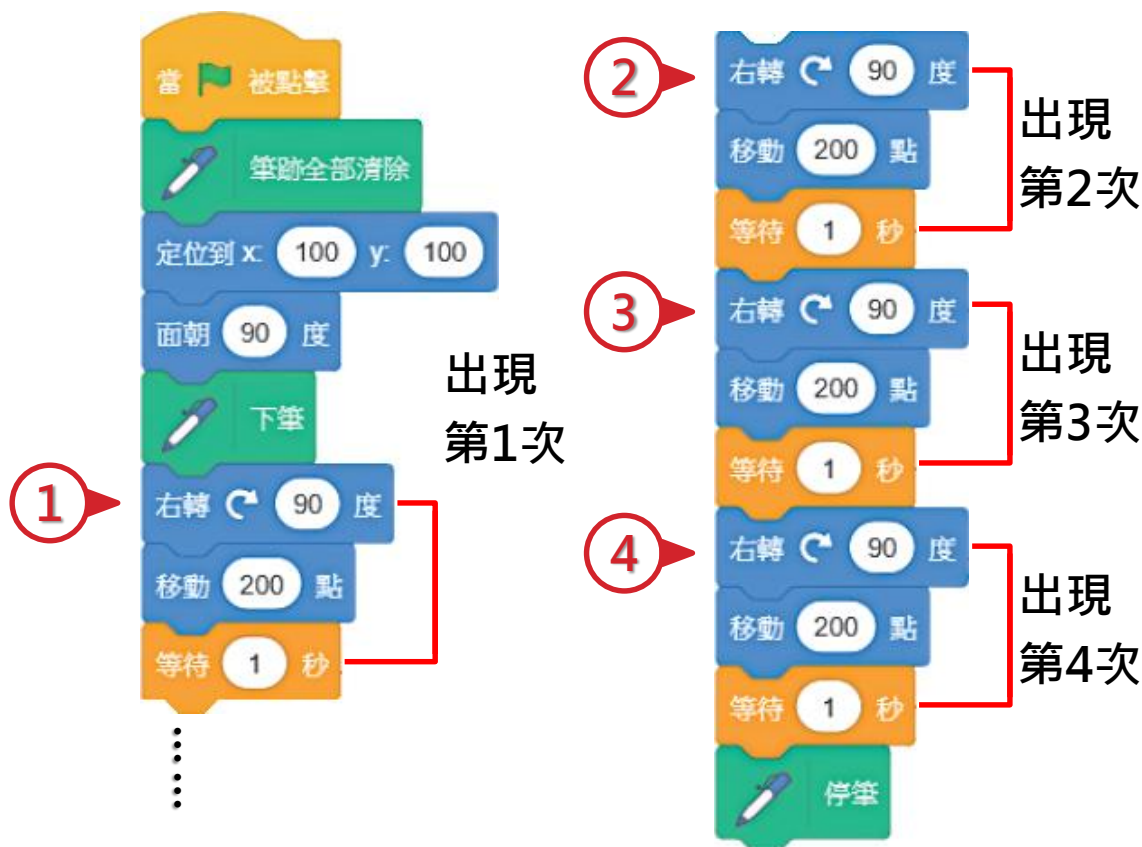
繪圖步驟拆解

用方向積木畫正方形

204

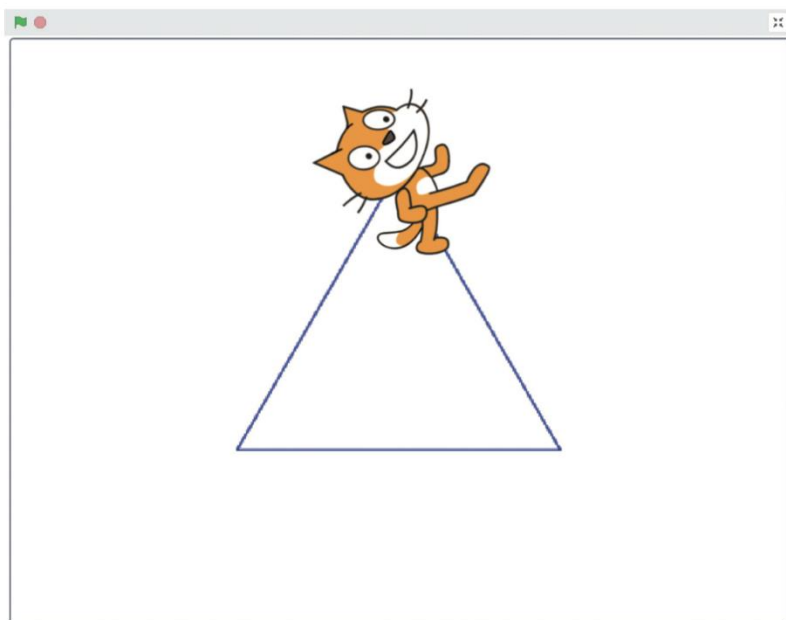


利用方向積木讓小貓畫出一個正方形





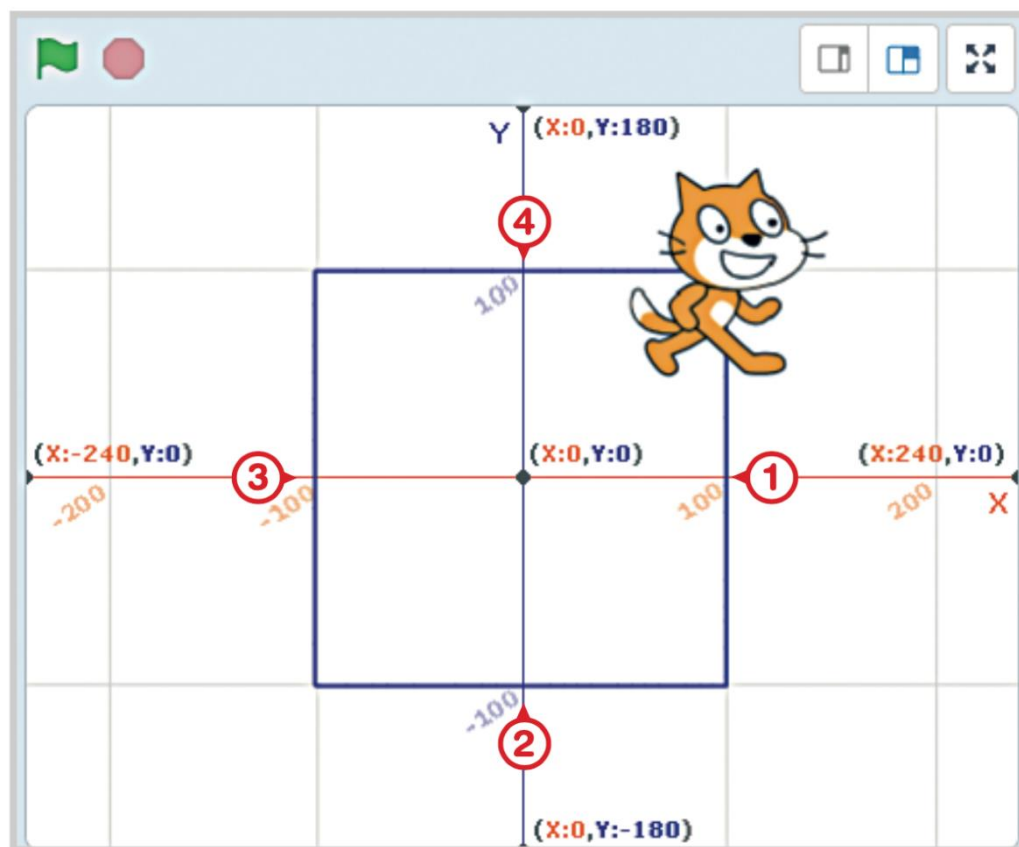
利用方向積木讓小貓畫出一個三角形





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

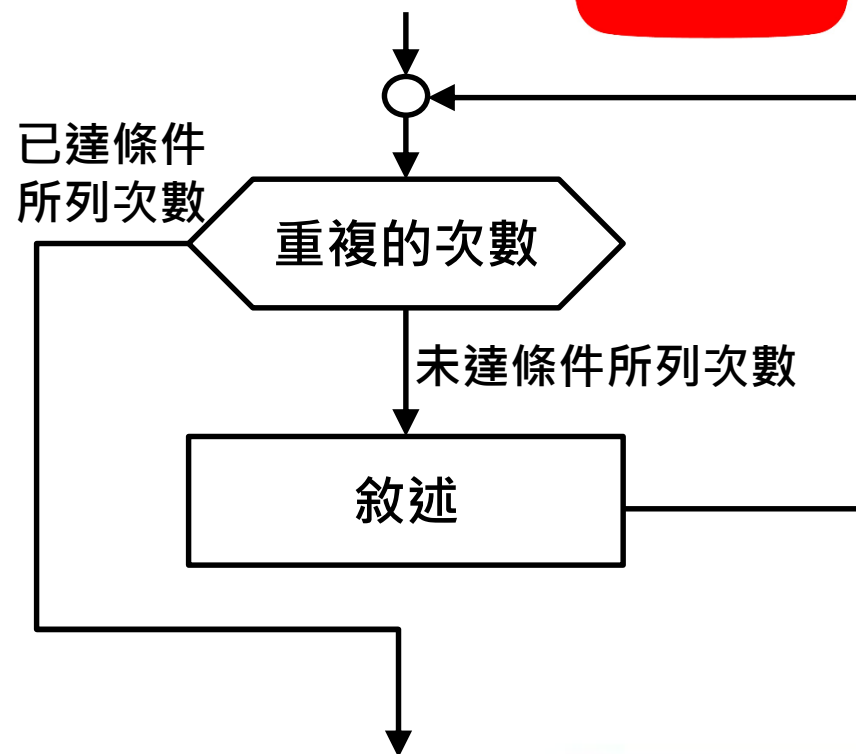
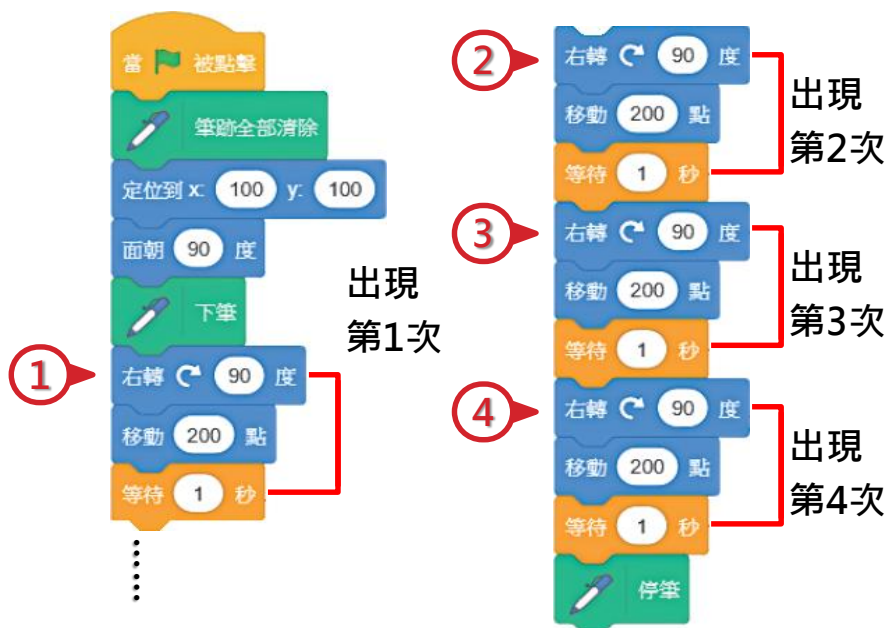




用計次式迴圈畫正方形

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

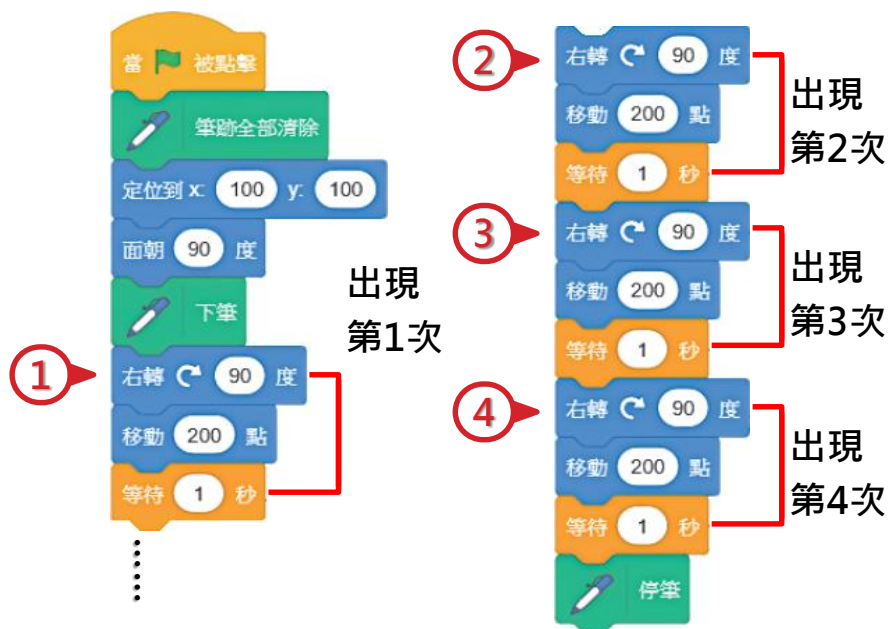
問題解析	問題實作
(A) 如何設定計次式迴圈？	用此積木取代重複執行程式碼
B) 如何控制角色的轉向？	輸入旋轉角度，向不同方向旋轉
(C) 如何控制角色移動的距離？	用此積木，輸入要往前移動幾個點，代表移動幾個像素



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。



重複執行4次



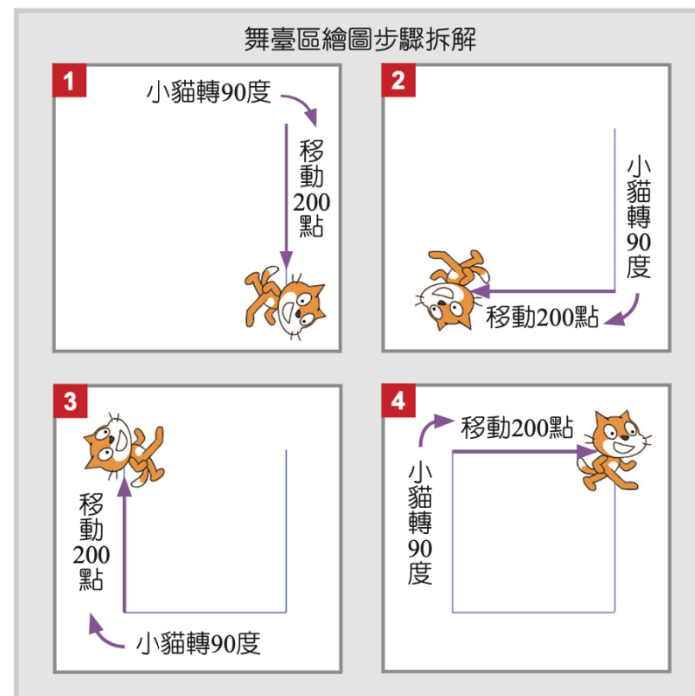
用計次式迴圈畫正方形

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。

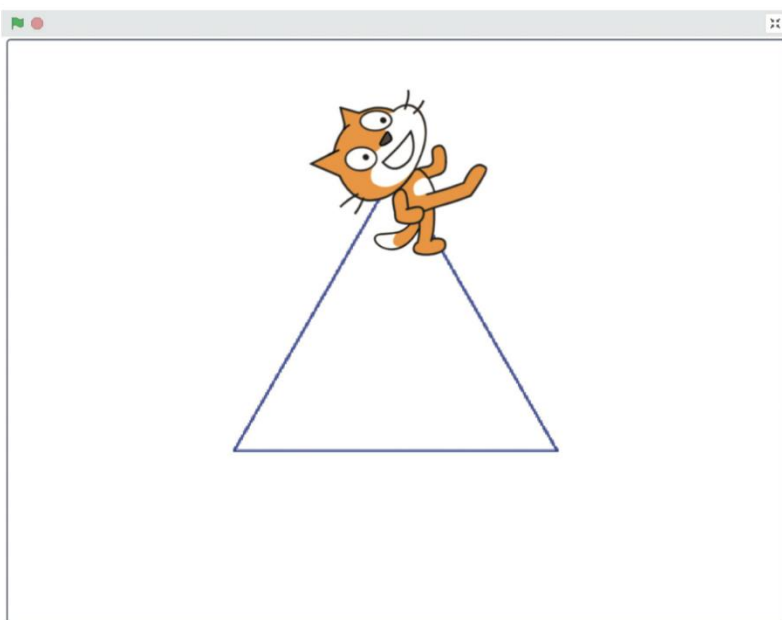


重複執行
4次，如右
圖的①~④





利用計次式迴圈讓小貓畫出一個三角形





畫正
方形

擴散
方形

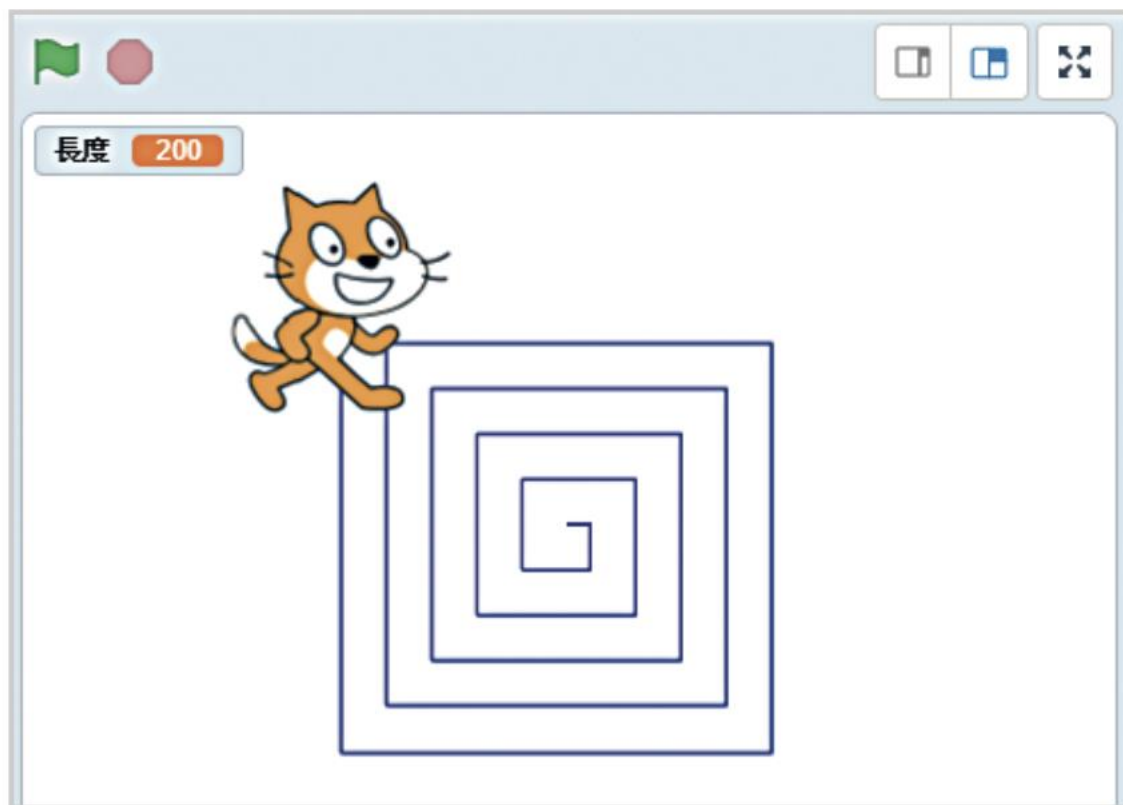
旋轉
方形





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

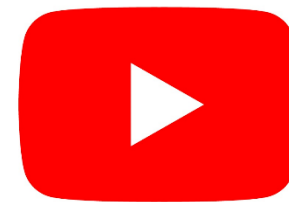
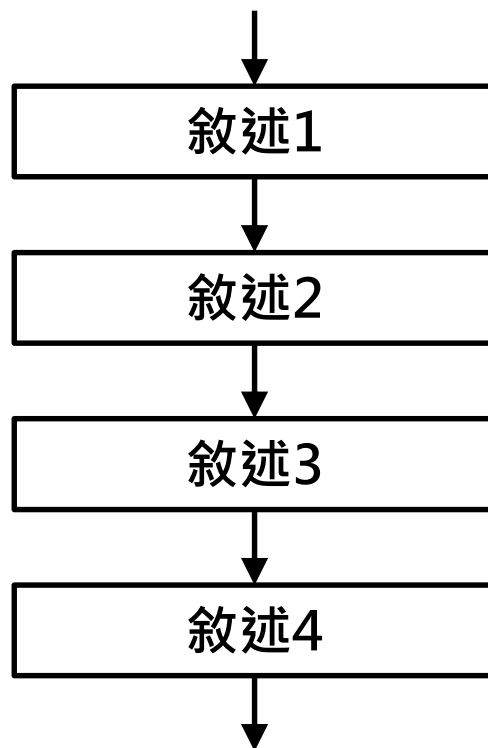




用循序結構畫擴散方形

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。 
(B) 如何控制角色的轉向？	輸入旋轉角度，向不同方向旋轉。 

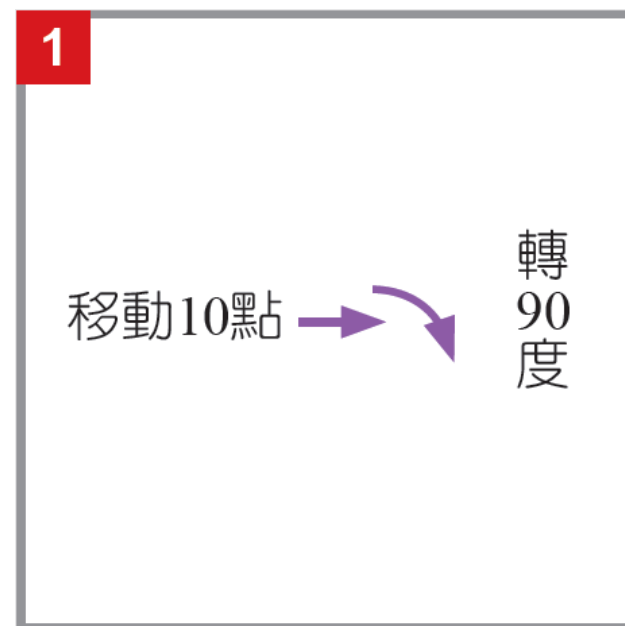
用循序結構畫擴散方形

206



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

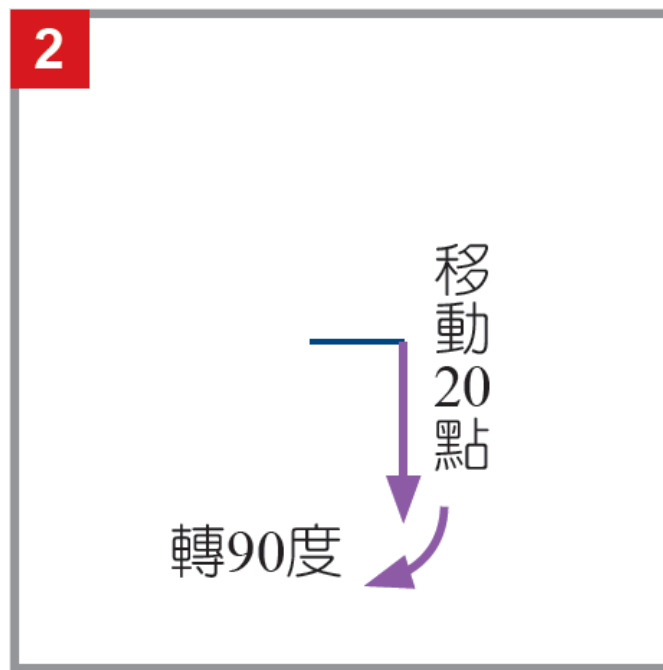


繪圖步驟拆解



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



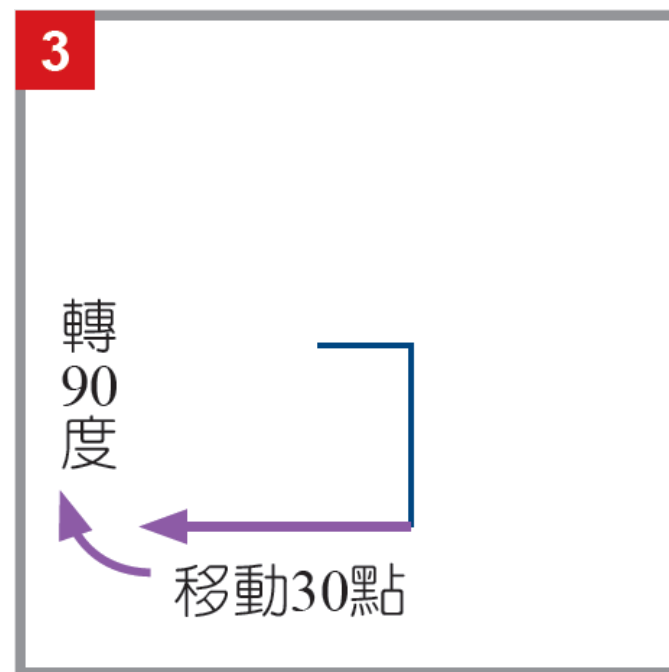
繪圖步驟拆解



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



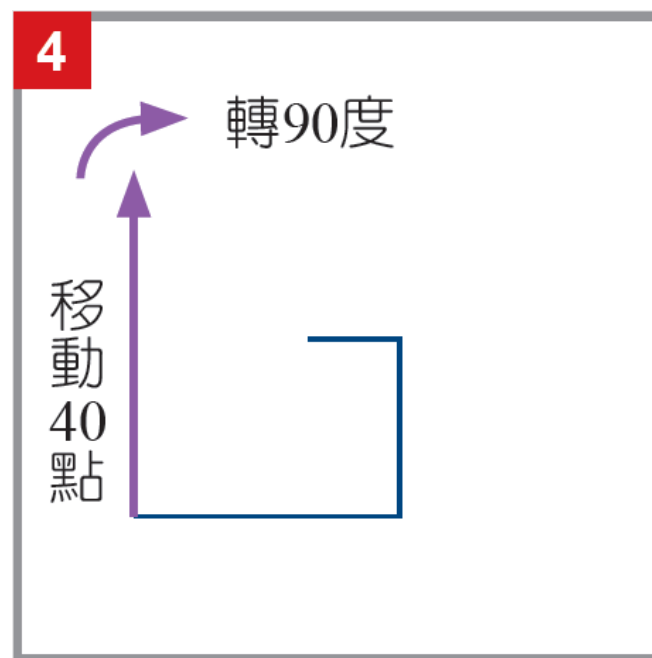
繪圖步驟拆解



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

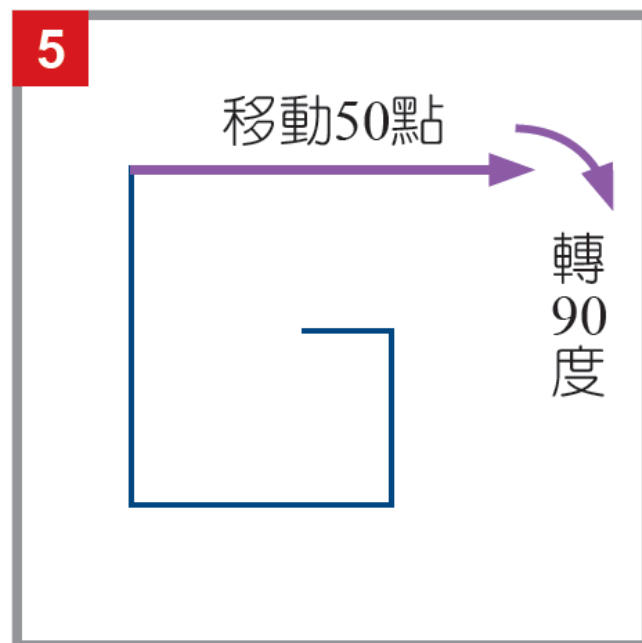
用循序結構畫擴散方形

206



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

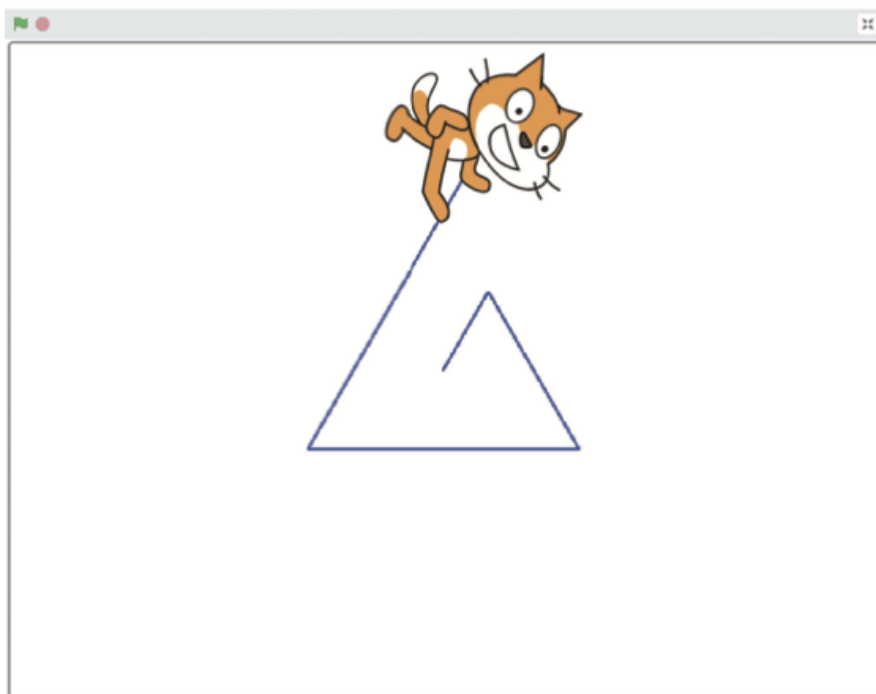
這麼多重複的積木怎麼利用變數簡化呢？





課堂練習

利用循序結構畫出一個擴散的三角形



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:

- Annotation 3: Points to the dialog title bar.
- Text: '新變數的名稱' (New Variable Name).
- Annotation 4: Points to the text input field containing '長度' (Length).
- Annotation 5: Points to the radio button selection area.
- Text: '● 適用於所有角色' (Selected) and '○ 僅適用當前角色' (Unselected).
- Buttons: '取消' (Cancel) and '確定' (Confirm).
- Annotation 6: Points to the '確定' (Confirm) button.



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



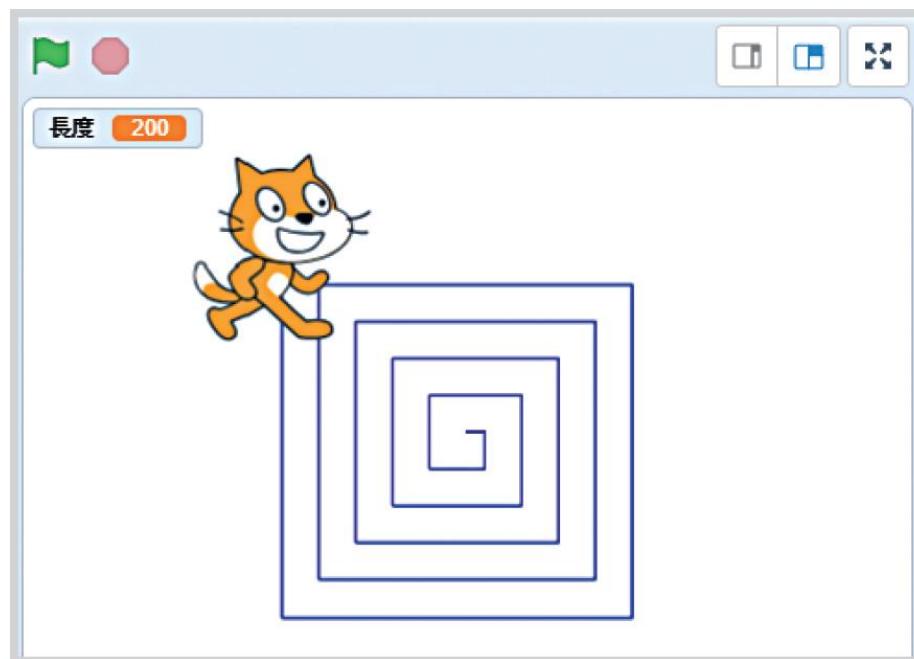


計次式迴圈與變數的應用

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，
再利用計次式迴圈積木
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



長度

設定變數初始值



0



= 0，儲存後數值為

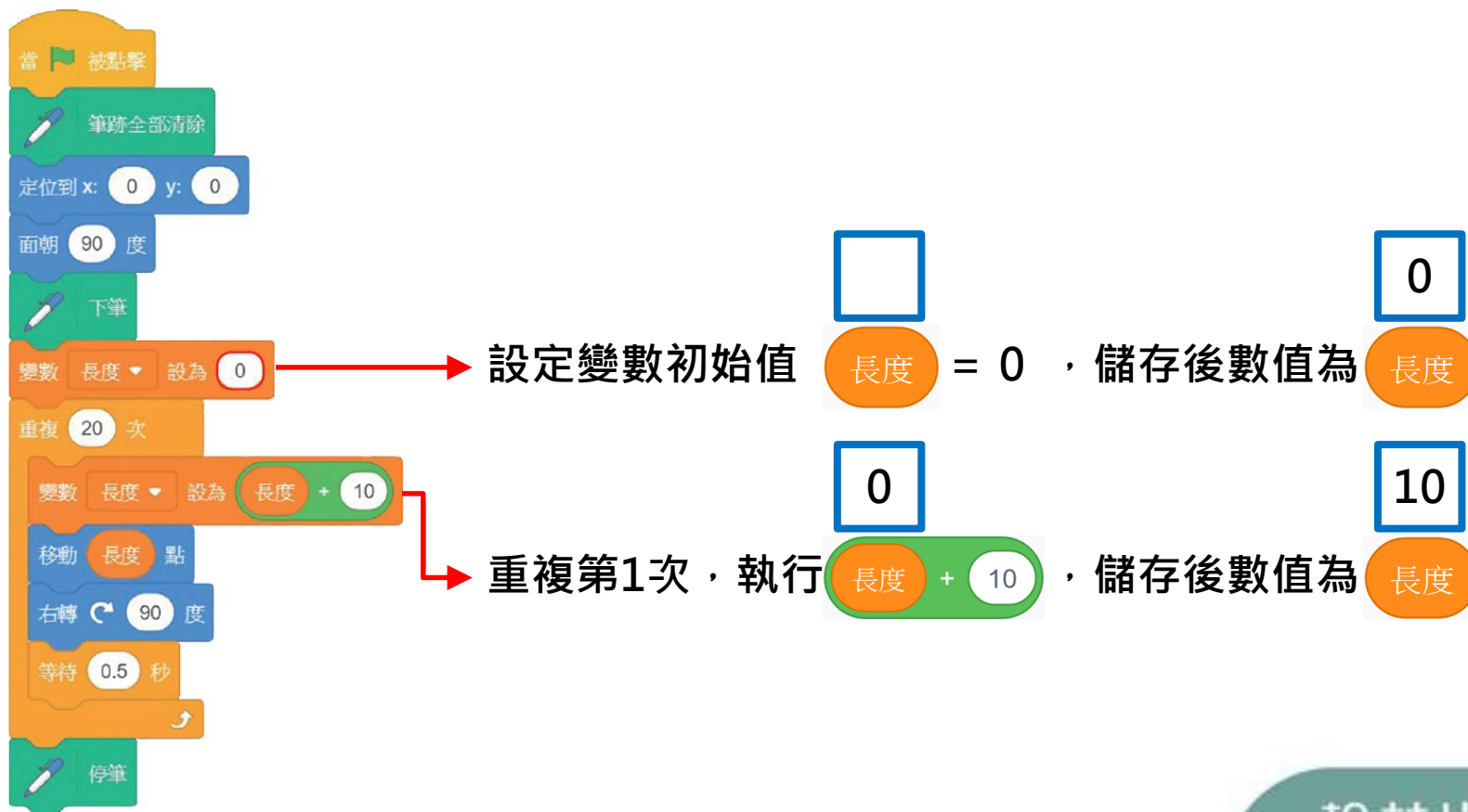
長度



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





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

長度

0

重複第1次，執行



長度

+ 10 , 儲存後數值為

長度

10

重複第2次，執行



長度

+ 10 , 儲存後數值為

長度

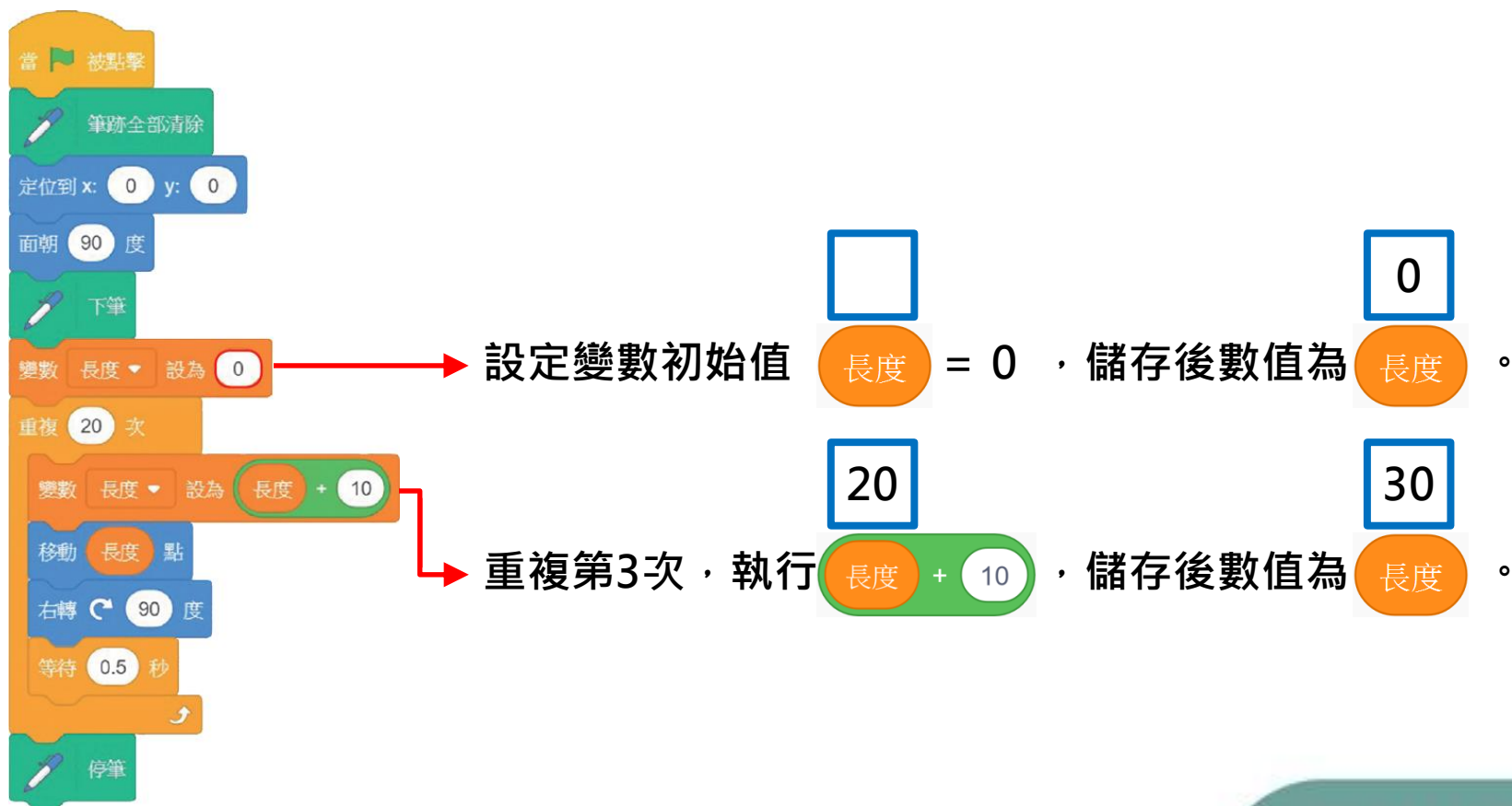
20



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

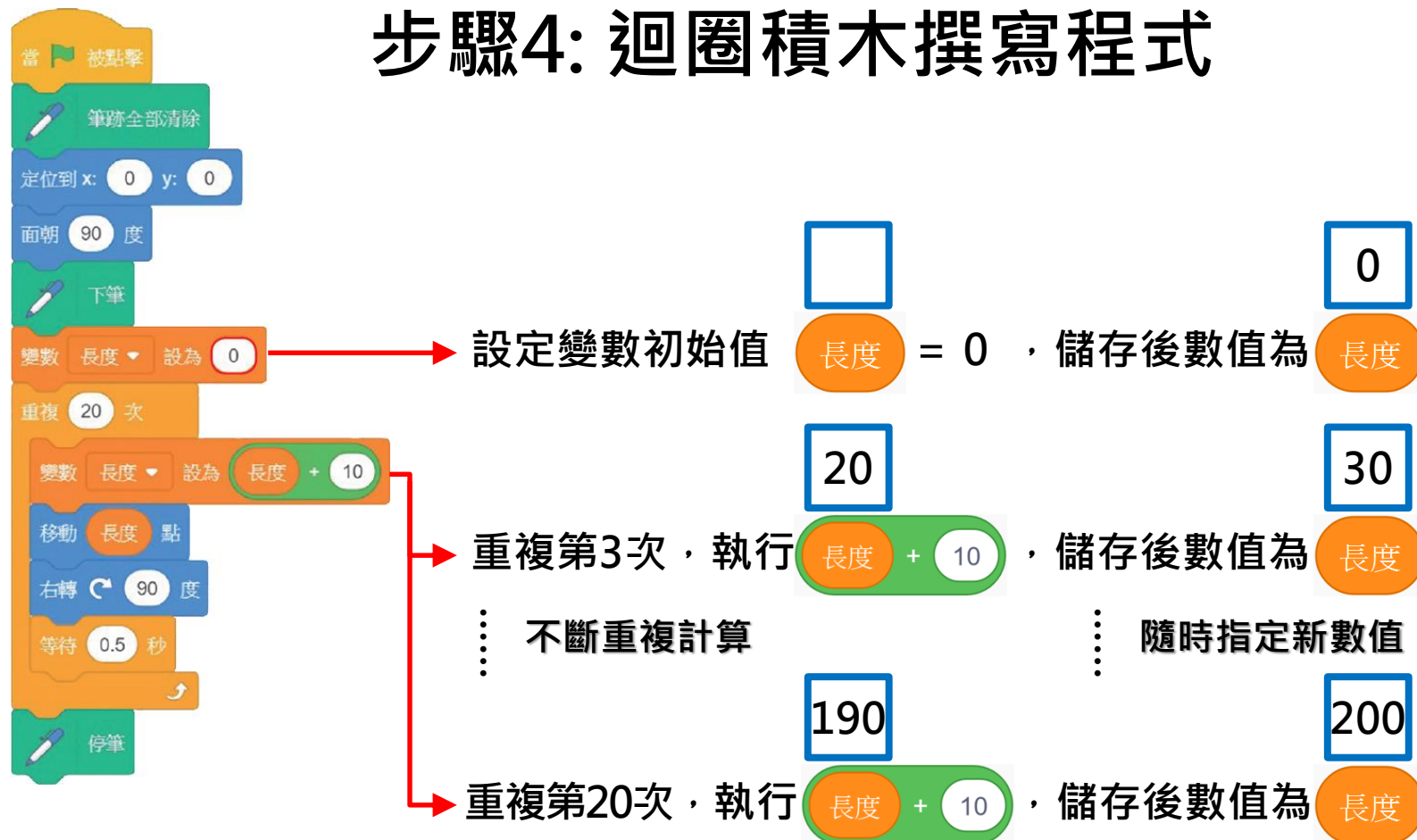




計次式迴圈與變數的應用

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

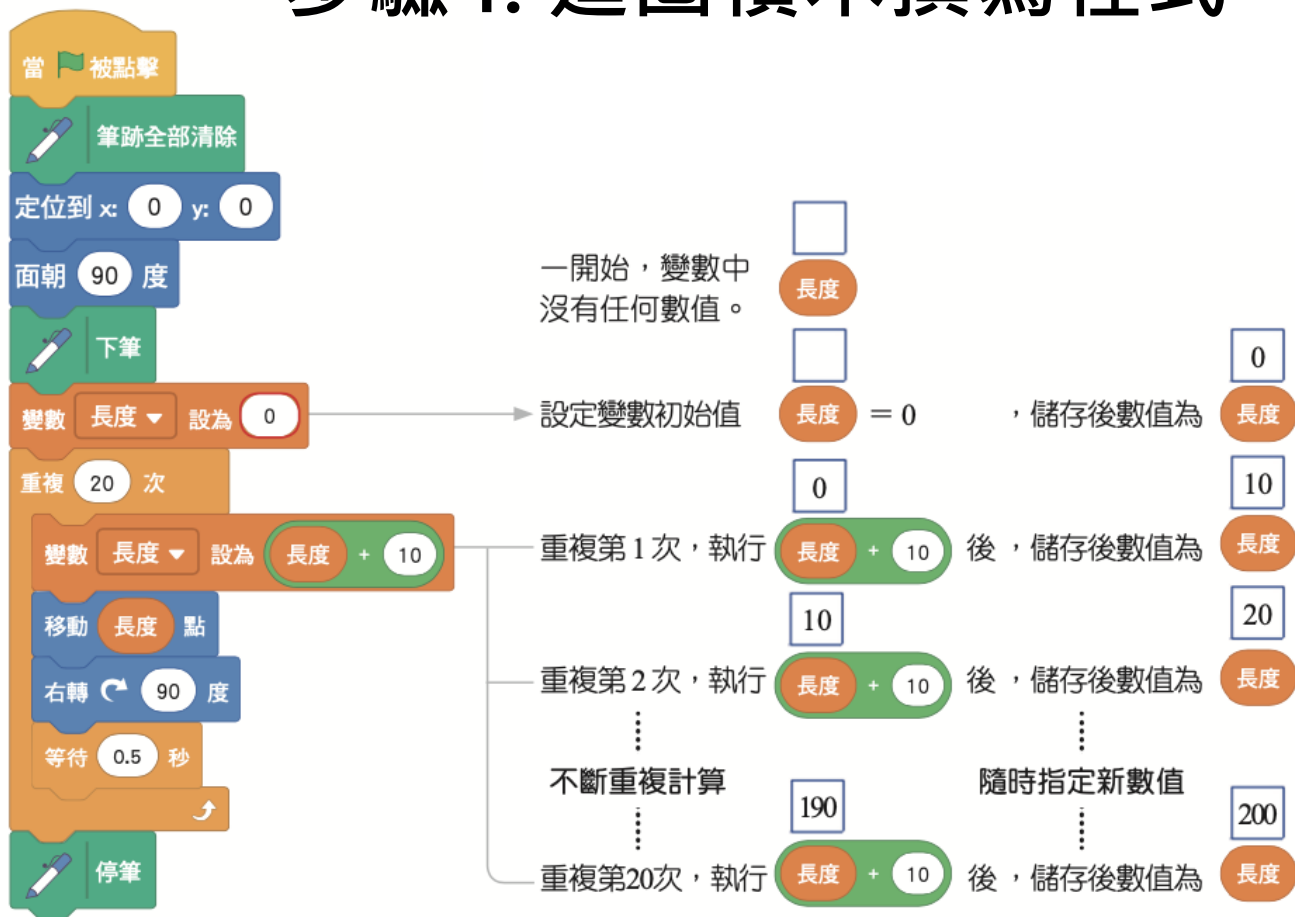




計次式迴圈與變數的應用

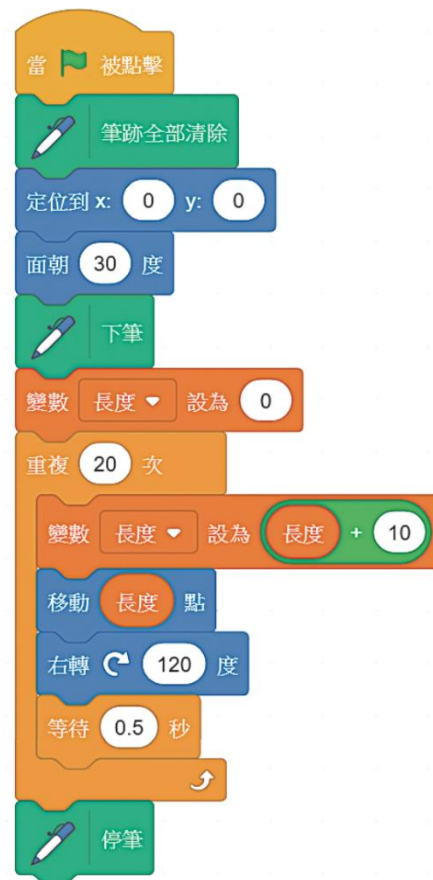
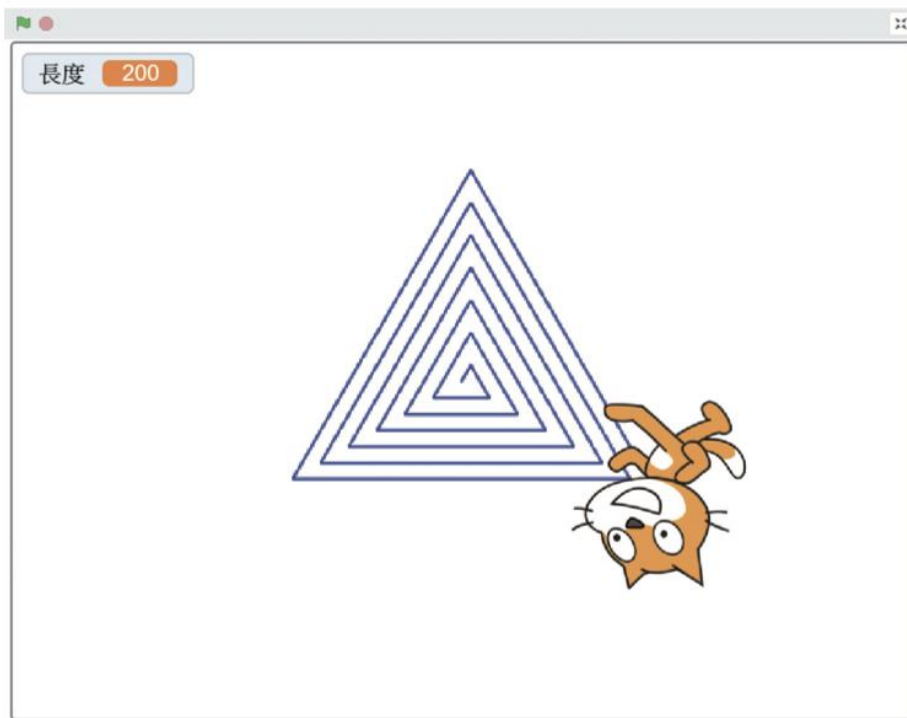
利用計次式迴圈與變數畫出一個擴散的方形

步驟4: 迴圈積木撰寫程式





用計次式迴圈與變數畫出一個擴散的三角形





畫正
方形

擴散
方形

旋轉
方形

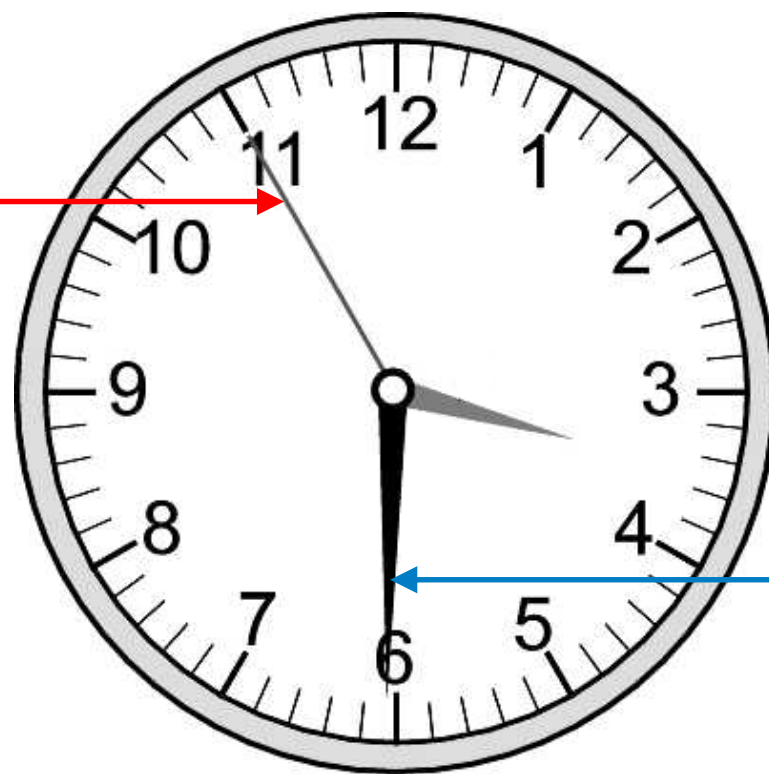




巢狀結構畫旋轉的正方形

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

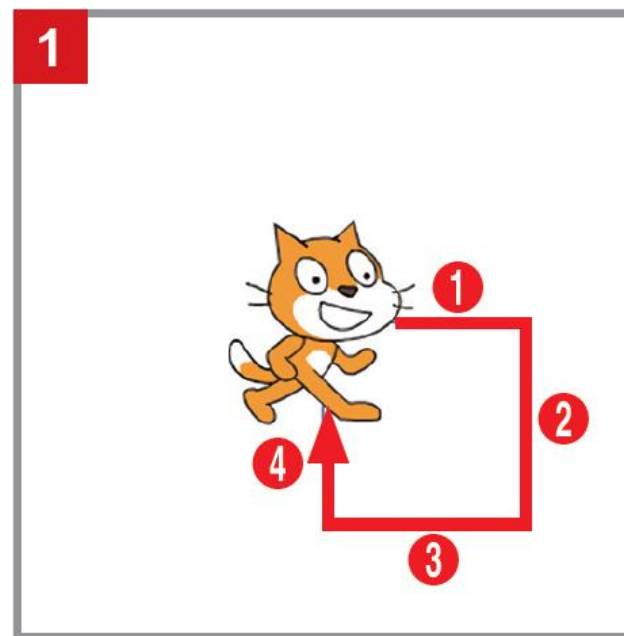
[Scratch繪圖篇 - 畫旋轉正方形](#)

畫邊長 100 點正方形



內層迴圈重複執行
4次，對照右圖

① ~ ④。

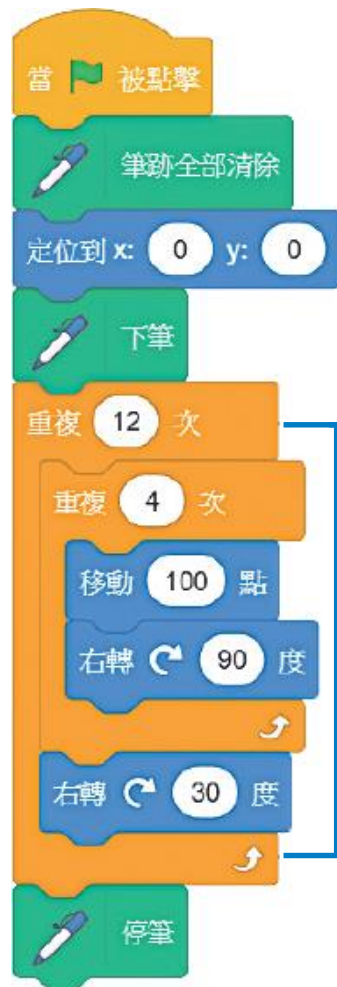


繪圖步驟拆解



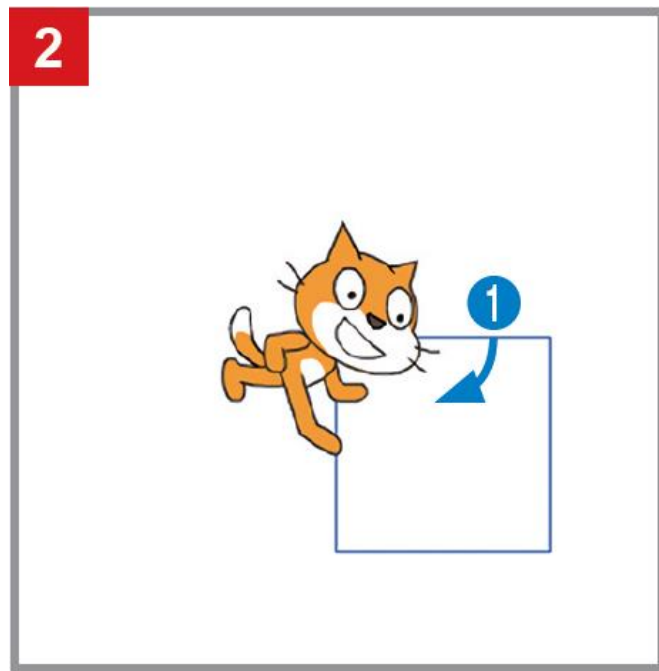
巢狀結構畫旋轉的正方形

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



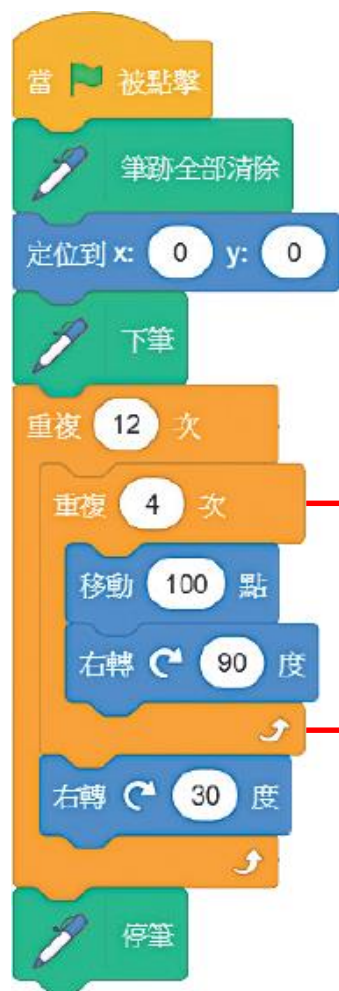
繪圖步驟拆解

範例-畫旋轉的正方形

210



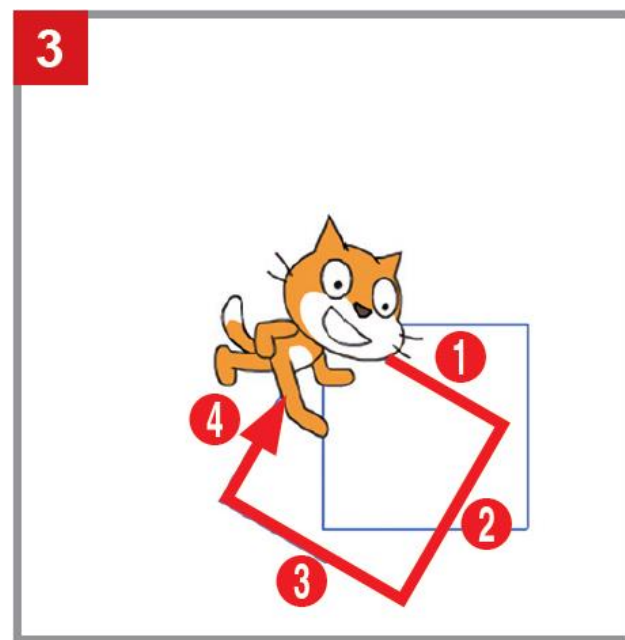
利用**巢狀結構**畫出12個旋轉的正方形。



內層迴圈重複執行4次，對照右圖

① ~ ④。

畫邊長 100 點正方形



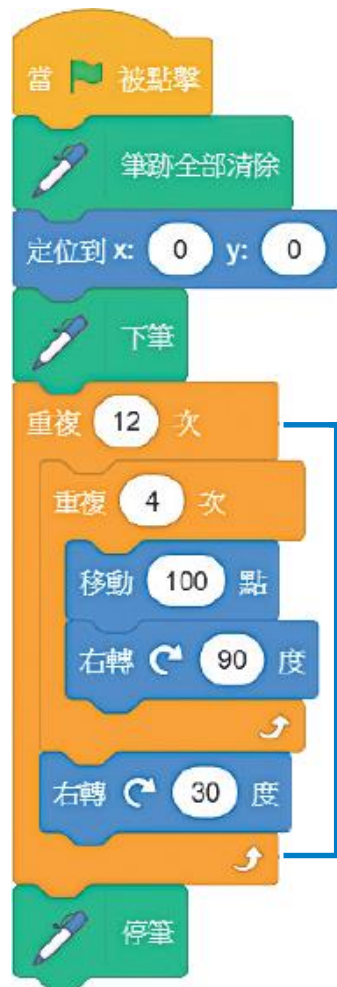
繪圖步驟拆解

範例-畫旋轉的正方形

210

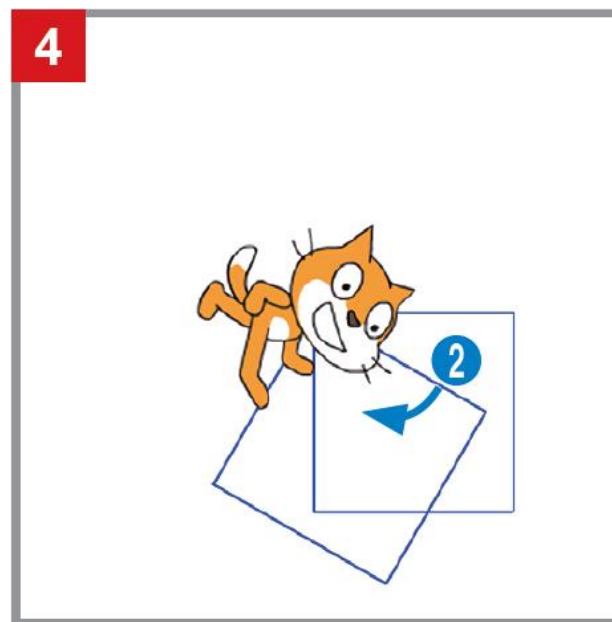


利用**巢狀結構**畫出12個旋轉的正方形。



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



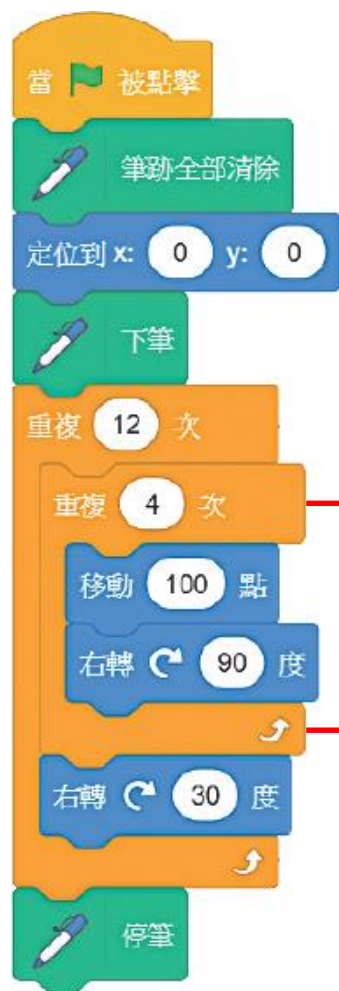
繪圖步驟拆解

範例-畫旋轉的正方形

210



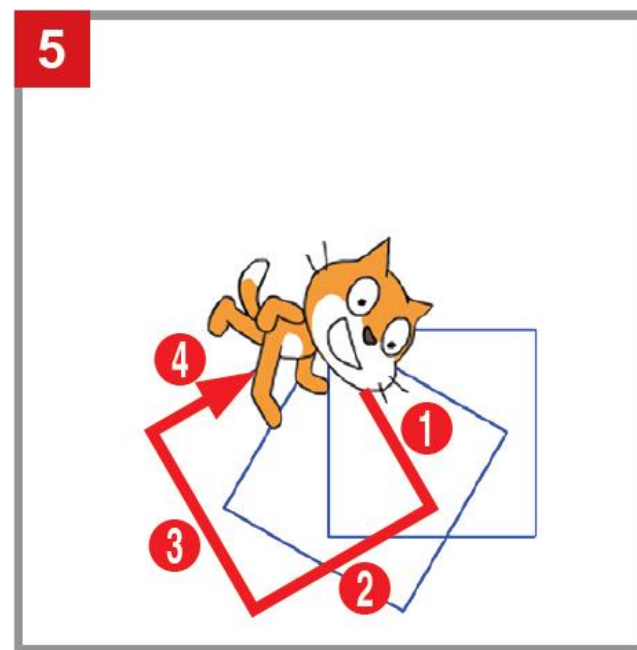
利用**巢狀結構**畫出12個旋轉的正方形。



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



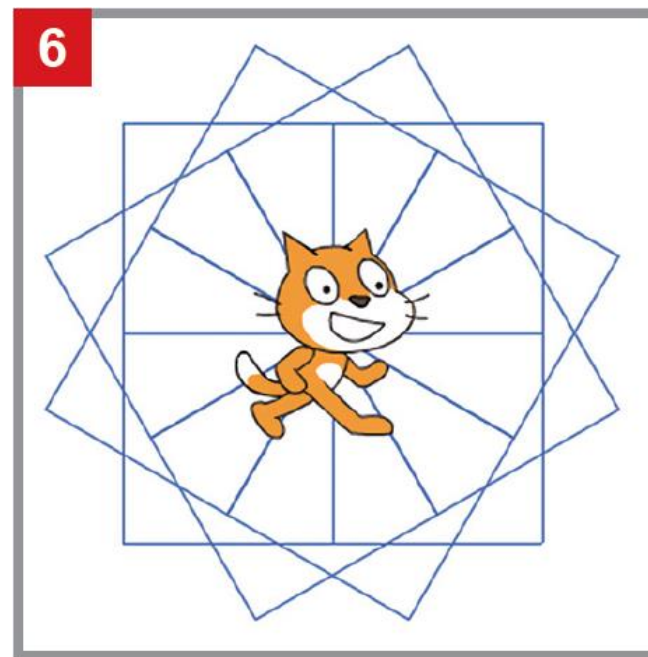
210

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。

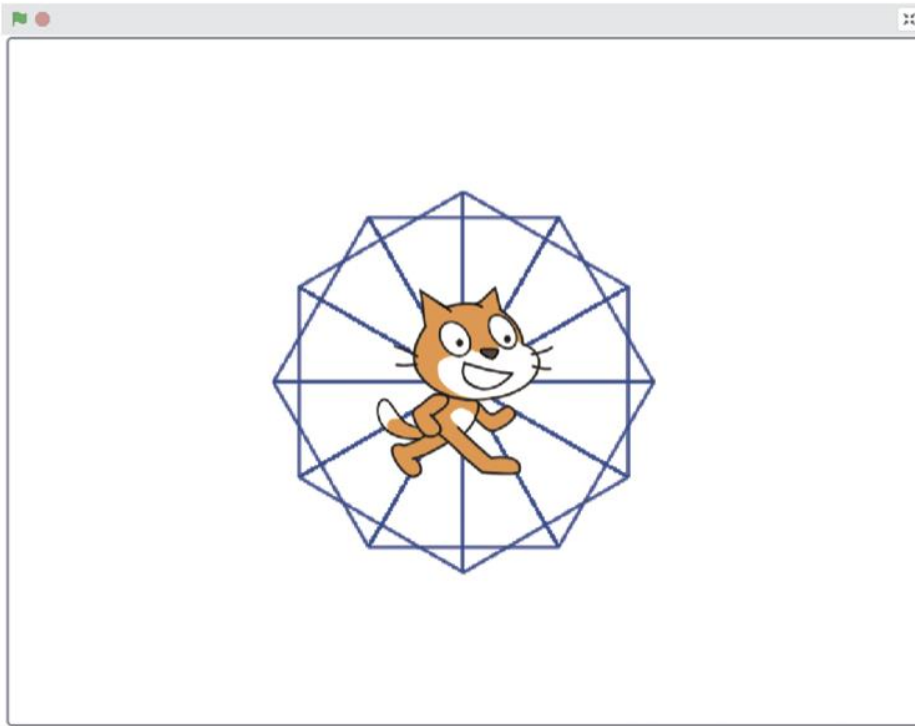
最後完成圖形



繪圖步驟拆解



利用巢狀結構畫出 12 個旋轉的正三角形





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。

線上派卷



第 2 章已經全部學習完畢，
點擊速測派按鈕進行第 2 章的題目練習！

 速測派

CH2 基礎程式設計 (1)



你還想知道什麼？

