

4-1 模組化的概念

4-2 認識模組化程式設計

4-3 模組化程式設計的應用

第 4 章



※本教材使用 Scratch 3.0 版本，因 Scratch 會不定期更新，介面可能與本教材略有不同。

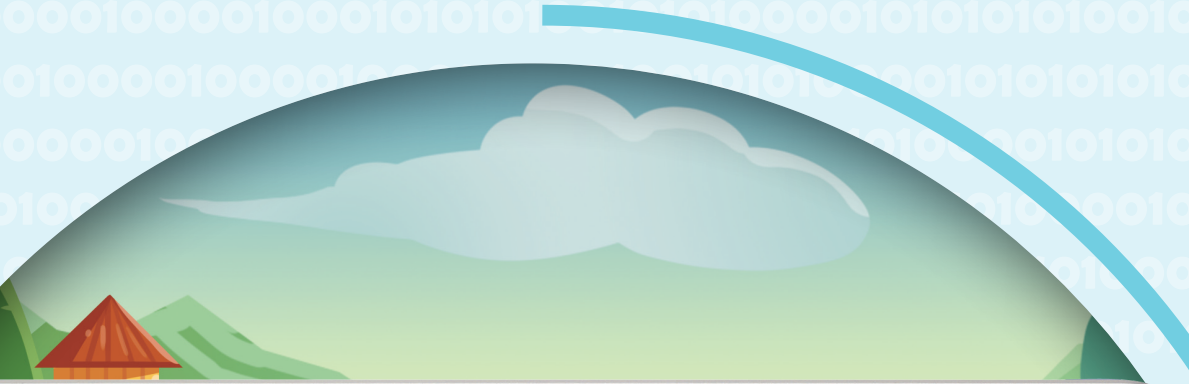
進階程式設計 (2)

上學期學習完進階程式設計 (1) 後，同學們有沒有覺得很多問題都可以用程式來解決？接下來要再介紹模組化程式設計，這也是另一種常常使用在撰寫程式的方法，相信你學完之後，會覺得可以利用程式來解決更多的問題。

本章首先介紹模組化的概念，接著開始進行模組化程式設計，藉著改寫七年級學習過的兩個畫正方形的範例，了解程式模組化後的便利性。最後以小鳥吃蟲的遊戲做總結，讓同學了解模組化程式設計的應用。

相信你已經迫不及待想了解模組化程式設計的便利性，現在就讓我們開始進程式設計吧！





Scratch 程式碼：

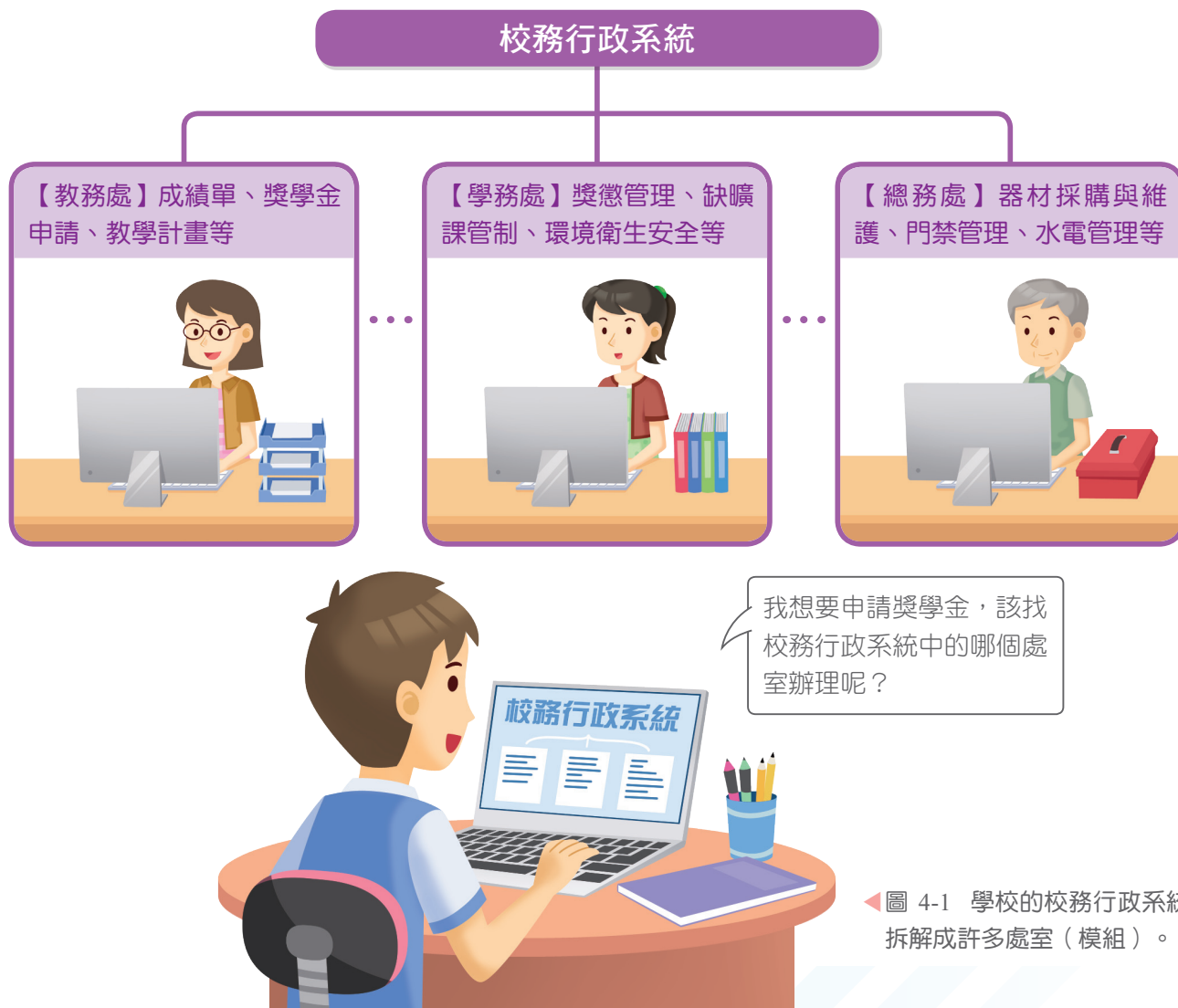
- 當被點擊
- 滑鼠鍵被按下?
- 重複無限次
 - 造型換成 小鳥彎腰
- 如果 那麼 否則
 - 定位到 鼠標
- 造型換成 小鳥站立



4-1 模組化的概念

在設計規模較大的程式時，常會運用模組化的概念；也就是把常用或重複使用的程式碼獨立出來成為模組，讓其他程式也可以輕鬆的利用模組化的程式碼。在撰寫程式解決問題的過程中，若是能夠善用模組化的概念，將有助於把原有的問題拆解成較小的問題，然後分別去解決，也方便程式的維護與修改。

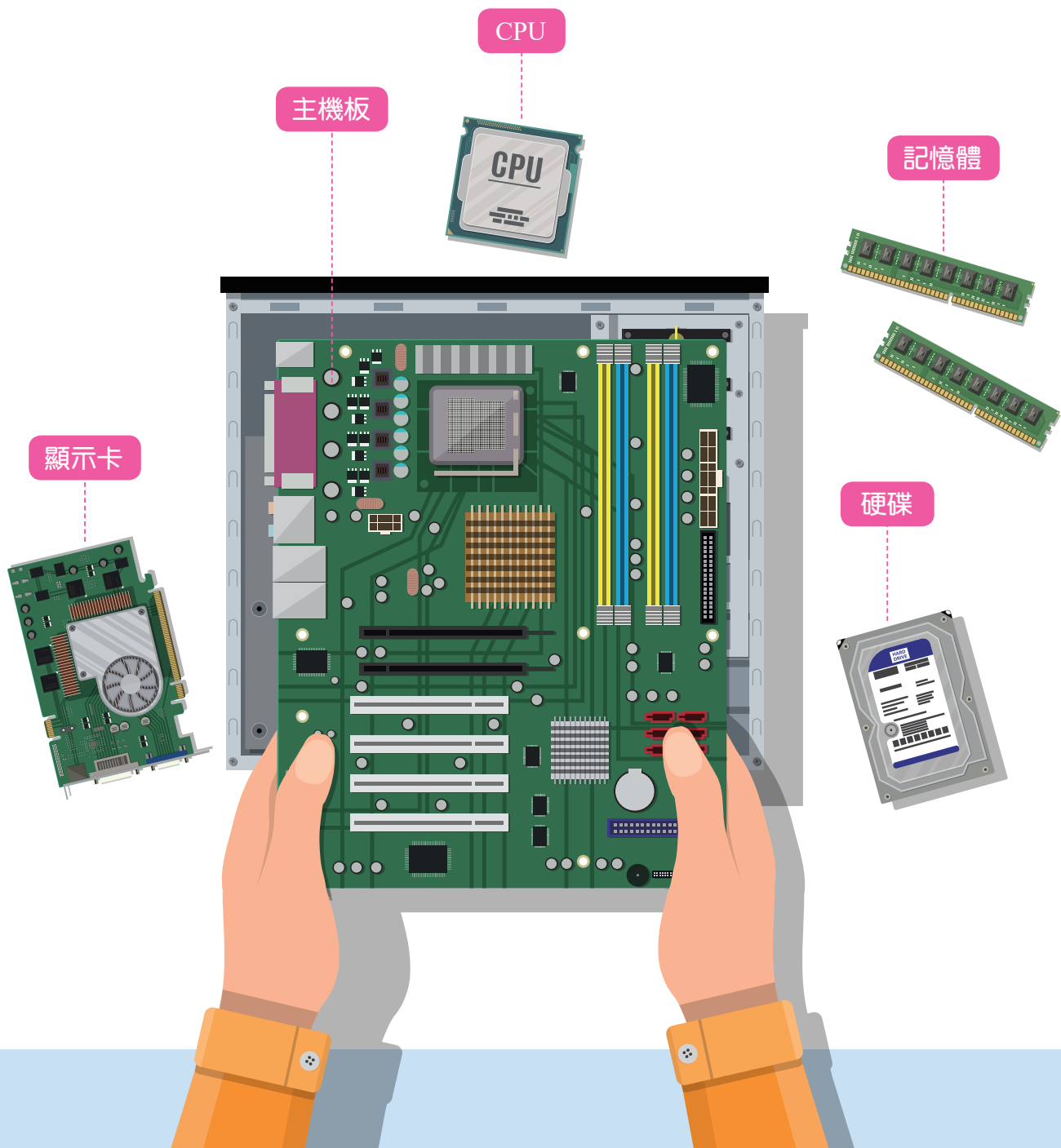
在學校的校務行政系統中，就經常用到模組化的概念。例如：各處室（如教務、學務、總務等）業務系統設計，都可各自設計成可以拆解的模組，各自發展完成後，再整合成為一個完整的校務行政系統（圖 4-1）。



◀ 圖 4-1 學校的校務行政系統可拆解成許多處室（模組）。

電腦主機也是運用模組化的設計，例如：中央處理器（CPU）、記憶體、硬碟、顯示卡、主機板等都可視為獨立的模組（圖 4-2）。

▼圖 4-2 電腦主機包含許多模組，可分別進行模組的更換與升級。



4-2 認識模組化程式設計

4-2-1 Scratch 中的模組化

在程式設計方面，如圖 4-3 我們想要畫出一個正三角形與一個正方形，程式會較為冗長，若想要修改或除錯某一個圖形時，也不容易直覺的找出該圖形所屬的積木。



The image shows a Scratch script for drawing a triangle and a square. The script consists of the following blocks:

- 當 被點擊
- 筆跡全部清除
- 定位到 x: -150 y: 0
- 下筆
- 重複 3 次
 - 移動 50 點
 - 左轉 120 度
- 停筆
- 移動 80 點
- 下筆
- 重複 4 次
 - 移動 50 點
 - 左轉 90 度
- 停筆
- 移動 80 點

The central canvas shows a blue triangle and a blue square, with the Scratch cat character next to them.

終於完成程式了！但我想
要修改正三角形的邊長，
要修改哪個積木呢？

讓我們來幫忙吧！如果
將程式模組化，維護或
修改積木會更方便。



▲圖 4-3 撰寫畫出正三角形與正方形的程式（以 Scratch 為例）。


透過撰寫副程式（sub-program）來達到模組化的成效，模組化也有助於增進程式碼的可讀性，例如：將圖 4-3 程式碼模組化，有人負責撰寫主程式，其他人則撰寫畫正三角形、畫正方形的副程式（圖 4-4），而在主程式中也可以更專注於描述任務本身，而不是完成任務的細節。



The image shows a Scratch workspace with three columns of code blocks. The first column, labeled '主程式撰寫' (Main Program Writing), contains a '當被點擊' (When clicked) event block, followed by '筆跡全部清除' (Erase all), '定位到 x: -150 y: 0' (Go to x: -150 y: 0), '畫正三角形' (Draw a triangle), '移動 80 點' (Move 80 points), '畫正方形' (Draw a square), and '移動 80 點' (Move 80 points). The second column, labeled '阿顯提供' (Provided by A Xian), contains a '定義 畫正三角形' (Define draw triangle) block, '下筆' (Pen down), a '重複 3 次' (Repeat 3 times) loop containing '移動 50 點' (Move 50 points) and '左轉 120 度' (Turn left 120 degrees), and '停筆' (Pen up). The third column, labeled '星兒提供' (Provided by Xing Er), contains a '定義 畫正方形' (Define draw square) block, '下筆' (Pen down), a '重複 4 次' (Repeat 4 times) loop containing '移動 50 點' (Move 50 points) and '左轉 90 度' (Turn left 90 degrees), and '停筆' (Pen up).

運用副程式對程式碼進行模組化設計，透過模組化，把功能獨立寫成一個一個的模組後，再進行組裝，優點如下：

1. 讓程式碼容易閱讀、維護與修改。
2. 有助於多人合作開發大程式。



▲圖 4-4 利用模組化撰寫畫出正三角形與正方形的程式（以 Scratch 為例）。

透過模組化設計，可以把問題拆解成獨立的小模組，例如：將畫正三角形、畫正方形這兩項獨立的任務分別寫成兩個不同的副程式，等到需要用到時，再由主程式去呼叫。而在 Scratch 中，使用者可自訂**函式積木**來撰寫副程式，例如：想要畫一個或多個正方形時，可以將畫正方形寫成一個副程式，當呼叫一次副程式時，就可以畫出一個正方形。

小知識

副程式

副程式是集合一些常用且重複撰寫的程式碼，其特性為可簡化主程式、使程式更容易閱讀、維護程式也更方便、且可節省程式所占用的記憶體，以及節省重複撰寫程式的時間。



定義副程式

建立函式積木後，程式面板中會自動出現此積木，接著在下方撰寫副程式的任務內容。



呼叫副程式

在撰寫主程式時，可以在函式積木類別中使用此積木，以執行副程式的任務內容。

命名函式積木時，建議簡潔易懂即可。

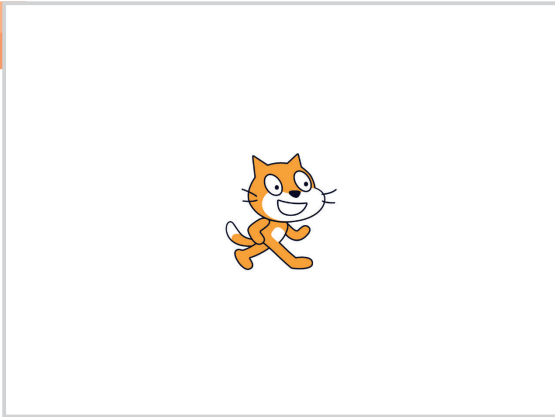


在七年級的繪圖篇中，利用巢狀結構畫出六個平行排列的正方形，接下來，讓我們學習怎麼將程式碼模組化。

範例：畫平行排列的正方形

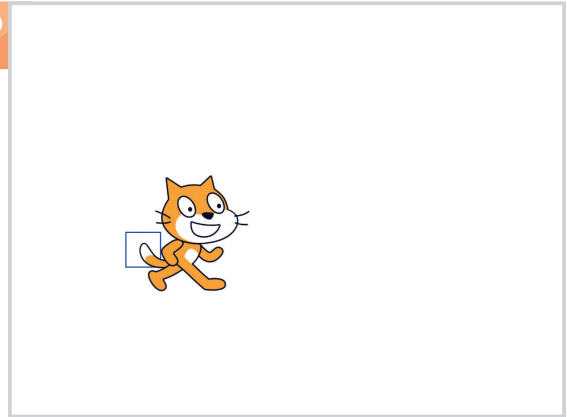
按下綠旗後，使小貓向右依序畫出六個平行排列的正方形。請執行《畫平行排列的正方形》的程式，想一想這個範例的程式是如何運作？

1



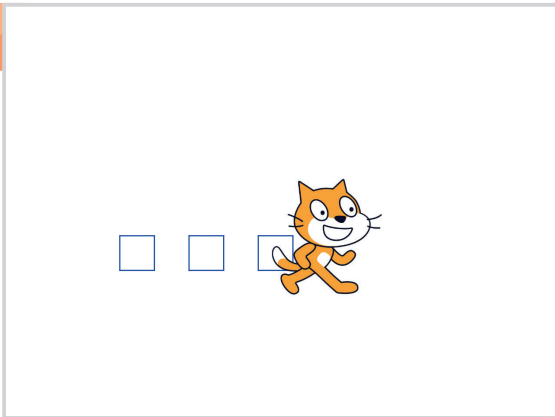
範例執行前

2



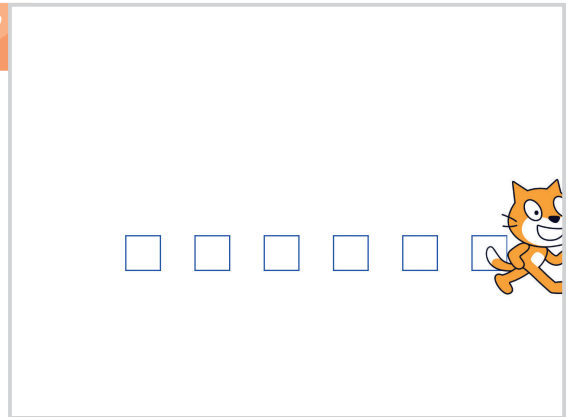
範例執行後，小貓從定位的位置先畫出一個正方形。

3



接著向右走動固定的距離，再依序畫出正方形。

4

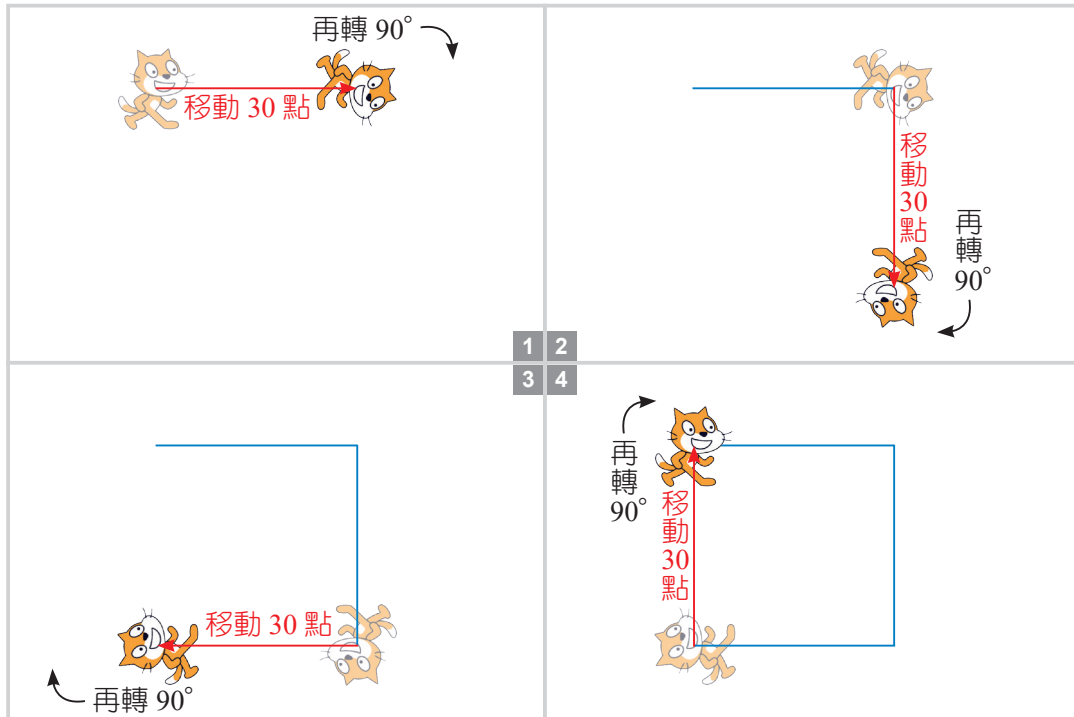


小貓向右依序畫完六個平行排列的正方形。

問題分析

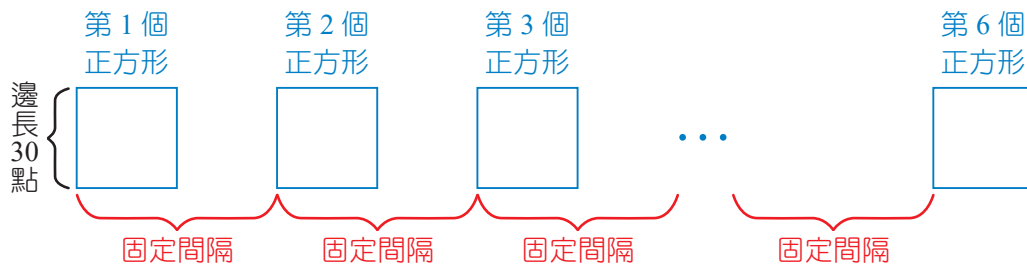
我們可以將這個程式範例拆解幾個部分如下：

① 如何畫出正方形？



② 如何畫出六個間隔相同的正方形？

1. 執行時，如何設定起始的定位位置，並考量畫完六個間隔相同的正方形時，不會超出畫面？



③ 如何利用副程式將程式碼模組化？

④ 如何設定副程式？

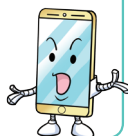
1. 執行時，如何使用副程式，畫出一個正方形？

⑤ 如何呼叫副程式？

1. 執行時，如何利用副程式，向右畫出六個平行排列的正方形？

解題步驟

讓我們透過 、，複習七年級所學過巢狀結構的 Scratch 程式吧！





問題拆解

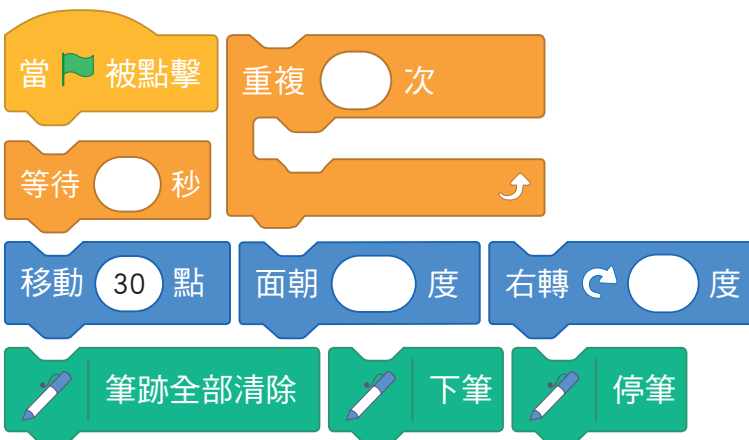
如何畫出正方形？

1

步驟 1

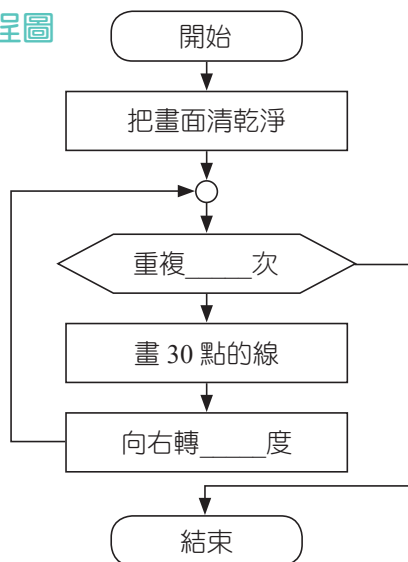
請依右方提示的積木進行組裝，完成畫正方形的程式。

1. 還記得在七年級是如何添加**畫筆**類別的積木嗎？
2. 如果將  改用 ，要怎麼組裝積木呢？



小知識

流程圖



接下來要學習進階程式設計。從現在開始提示的積木數量可能不完全，請同學依程式邏輯判斷需要使用的積木。



問題
拆解


如何畫出六個間隔相同的正方形？

2




步驟
2

請結合上個步驟，並依右方提示的積木進行組裝，設計可以讓小貓向右畫出六個平行排列的正方形。


請先將 ^{步驟 1} 的積木組裝完成，再與下方的積木進行組裝。

當  被點擊

移動 點 定位到 x: y:

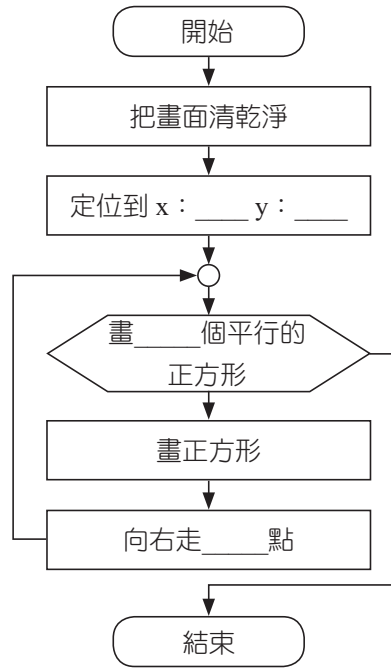
 筆跡全部清除  下筆  停筆

重複 次



小知識

流程圖



問題
拆解

如何利用副程式將程式碼模組化？

3

請同學想想看，哪個部分是一直重複執行的？

可以將重複執行的程式設計成一個獨立的模組。



接下來，讓我們學習如何利用副程式來重新撰寫此範例吧！

請同學想想看，哪個類別的積木可以用來撰寫副程式呢？



問題
拆解

如何設定副程式？

4

步驟
3新增畫正方形的函
式積木。① 點選小貓角色，再點選
函式積木類別。② 按下**建立一個積木**。③ 命名為**畫正方形**。④ 按下**確定**鍵。

步驟 4

請依右方提示的積木進行組裝，完成畫正方形的副程式。

組裝完成的程式，與 **步驟 2** 的程式有什麼地方相同？有什麼地方不同？

定義 畫正方形

當角色被點擊

右轉 90 度

移動 30 點

面朝 90 度

筆跡全部清除

下筆

停筆

重複 4 次

問題拆解

5

如何呼叫副程式？

步驟 5

請依右方提示的積木進行組裝，讓小貓向右畫出六個平行排列的正方形。

畫正方形

當 被點擊

定位到 x: y:

移動 30 點

筆跡全部清除

下筆

停筆

重複 6 次



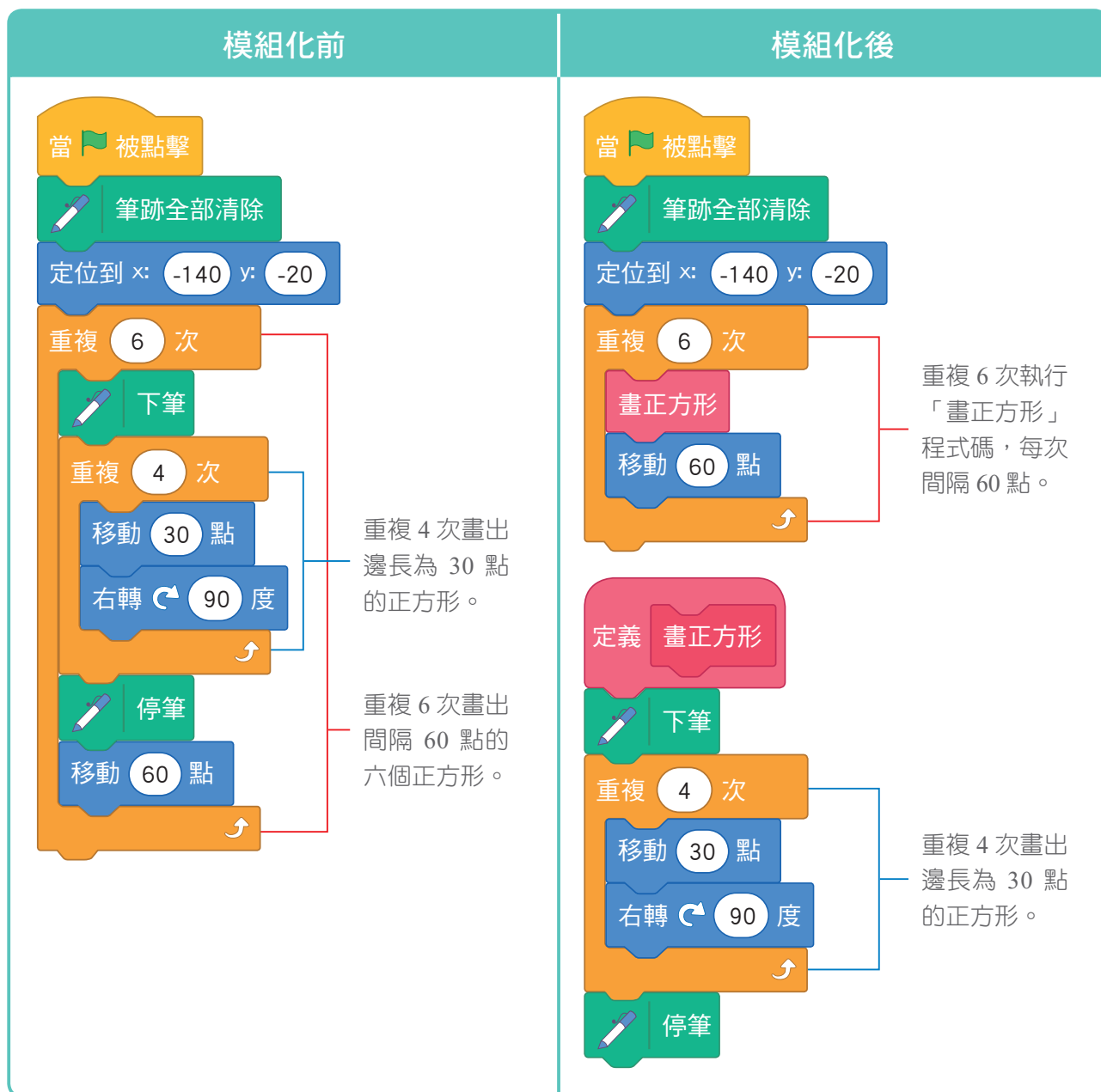
練習

想想看，如果要向上依序畫出六個平行排列的正方形，要如何進行？

在此單元中，我學到的有：

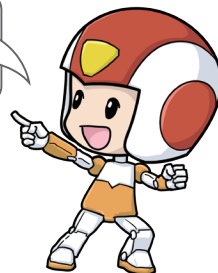
1. 將整個大任務切割成幾個小任務。
2. 利用函式來完成整個程式的某項小任務。

將「畫正方形」寫成副程式後，可清楚看出此程式何時需要畫正方形，因此較容易閱讀，模組化前後的程式碼如下所示。



想想看，其他重複執行的程式，可以設為副程式嗎？

說說看，你比較喜歡哪一種方法，為什麼？



4-2-2 Scratch 中的模組化參數

副程式可以為我們完成某些特定任務（如畫正方形），而當任務須隨著情況調整時（如畫正方形、畫正三角形或畫正五邊形等），為了讓副程式更具彈性，在定義副程式時，也可以同時為這個副程式定義參數（parameter）。

參數是我們想要為此副程式彈性調整的變數，這些變數可以是數字、字串或布林值（boolean value）等不同資料型態。在 Scratch 中則是使用函式積木的**添加輸入方塊**來描述參數，例如：想要畫多邊形時，可以設定參數為邊數，當呼叫副程式並指定參數值為 3，就可以畫出正三角形。

小知識

布林值

電腦儲存不同型態的資料會使用不同的記憶體大小，其中有一類稱為布林值，它只有兩種狀態，true（真）和 false（假），所以電腦用最少的記憶體大小來儲存此類變數。



定義

畫多邊形

邊數

定義副程式

建立函式積木後，程式面板中會自動出現此積木，接著在下方撰寫副程式結合參數的任務內容。

參數

定義副程式時，可直接拖曳拉出嵌入至其他積木內；呼叫副程式時，則傳入指定的參數值。

畫多邊形

呼叫副程式

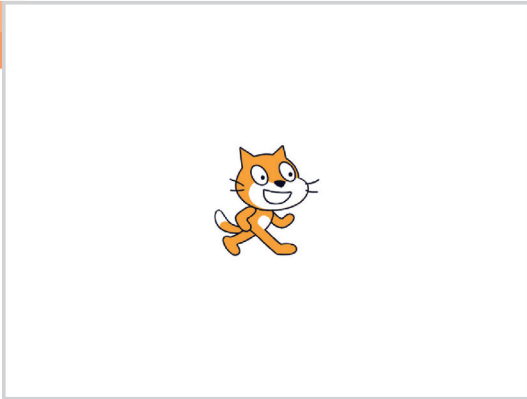
在撰寫主程式時，可以在函式積木類別使用此積木，並在空白處輸入參數值，以執行副程式的任務內容。

在七年級的繪圖篇中，利用巢狀結構畫出逐漸擴大的正方形，接下來，讓我們進一步學習怎麼利用參數將程式碼模組化。

範例：畫逐漸擴大的正方形

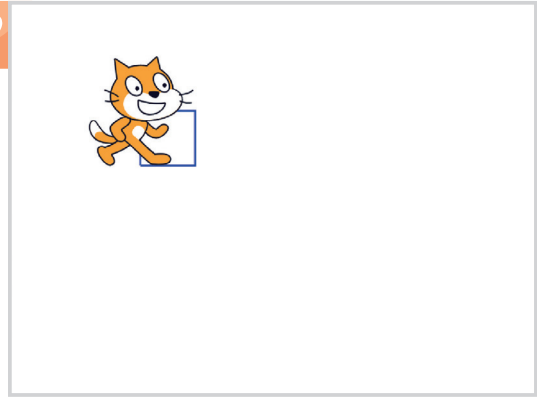
按下綠旗後，使小貓向右依序畫出四個逐漸擴大的正方形。請執行《畫逐漸擴大的正方形》的程式，想一想這個範例的程式是如何運作？

1



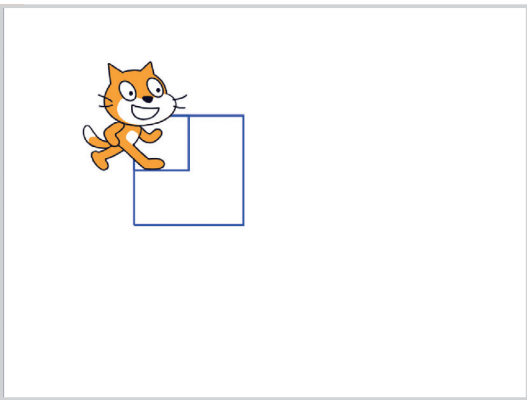
範例執行前

2



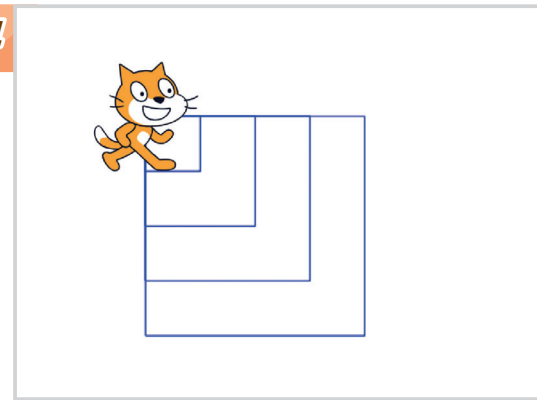
範例執行後，小貓從定位的位置先畫出一個正方形。

3



接著再向右依序畫出不同邊長的正方形（邊長增加固定的長度）。

4



小貓向右依序畫完四個逐漸擴大的正方形。

問題分析

我們可以將這個程式範例拆解幾個部分如下：

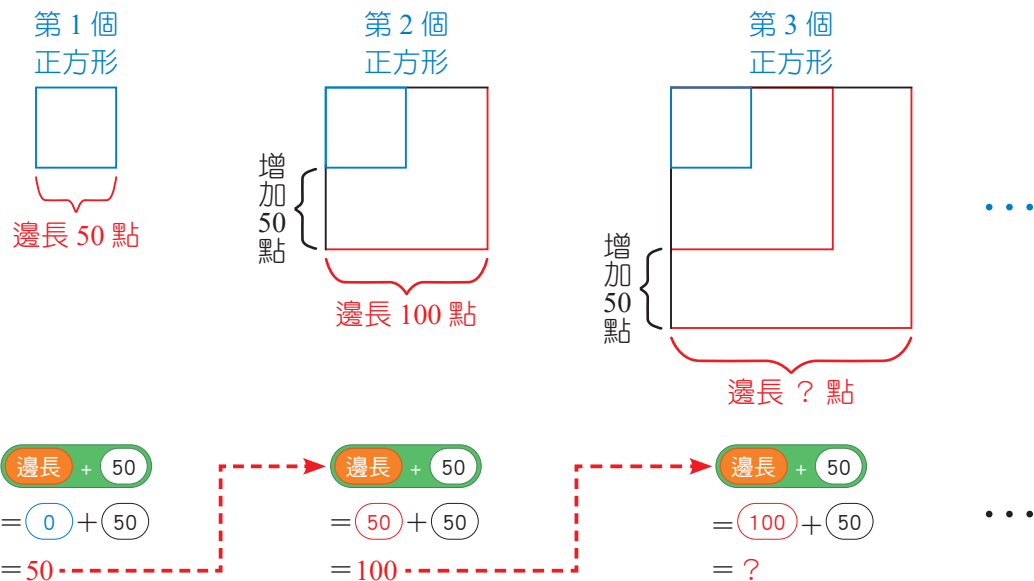
① 如何畫出正方形？

② 如何畫出四個不同大小的正方形？

1. 執行時，如何設定起始的定位位置，並考量畫完四個不同大小的正方形時，不會超出畫面？

利用變數，設定四個正方形的邊長：

變數 邊長 ▾ 設為 邊長 + 50 ◯



③ 如何利用副程式將程式碼模組化？

④ 如何設定副程式？

1. 執行時，如何使用副程式，畫出四個不同大小的正方形？

⑤ 如何呼叫副程式？

1. 執行時，如何利用副程式，向右畫出四個逐漸擴大的正方形？

⑥ 如何利用副程式的參數將程式碼模組化？

⑦ 如何設定副程式的參數？

1. 執行時，如何使用副程式的參數，畫出四個不同大小的正方形？

⑧ 如何呼叫副程式的參數？

1. 執行時，如何利用副程式的參數，向右畫出四個逐漸擴大的正方形？

解題步驟



讓我們透過 **問題拆解 1**、**問題拆解 2**，複習七年級所學過巢狀結構的 Scratch 程式吧！

這次來挑戰一次將程式組裝完成吧！讓小貓向右畫出四個逐漸擴大的正方形。



問題拆解

如何畫出正方形？

1

步驟 1

請依右方提示的積木進行組裝，完成畫正方形的程式。

利用邊長的變數，讓小貓移動時，畫出正方形的邊長與旋轉角度，完成一個正方形。

當  被點擊

筆跡全部清除



下筆



停筆

移動  點右轉  度重複  次

變數

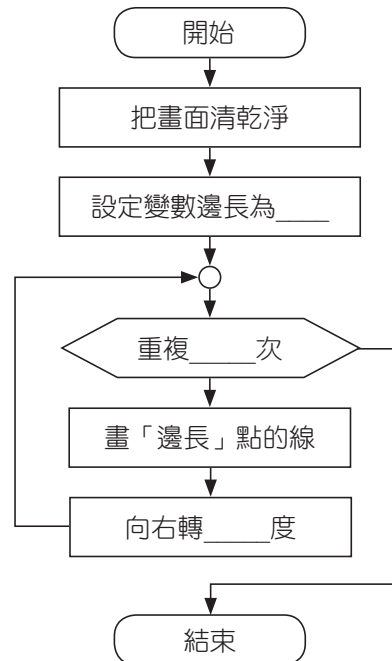
邊長 ▾

設為 

邊長

小知識

流程圖



問題拆解

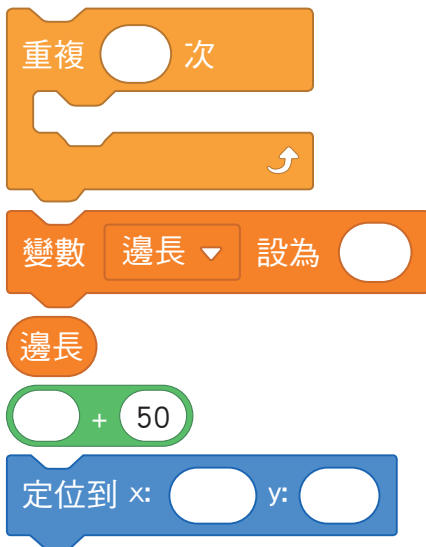
2

如何畫出四個不同大小的正方形？

步驟 2

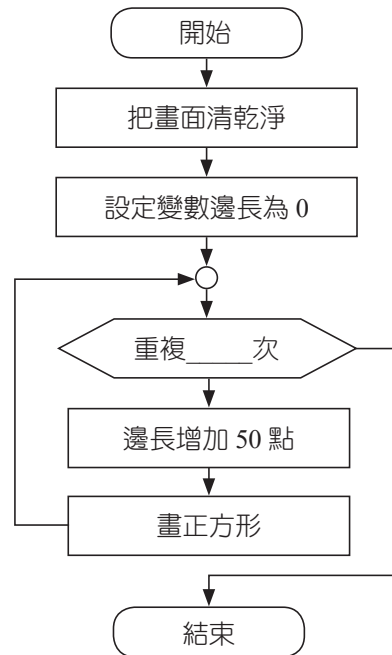
請結合上個步驟，並依右方提示的積木進行組裝，設計可以讓小貓向右畫出四個逐漸擴大的正方形。

1. 請先將 **步驟 1** 的積木組裝完成，再與右方的積木進行組裝。
2. 利用邊長的變數，讓小貓每畫完一個正方形，邊長則固定增加 50 點，完成四個逐漸擴大的正方形。



小知識

流程圖



問題拆解

3

如何利用副程式將程式碼模組化？

請同學想想看，哪個部分是一直重複執行的？



接下來，讓我們學習如何利用副程式來重新撰寫此範例吧！

上個範例畫出六個正方形是**相同大小**，但這次四個正方形是**不同大小**，請同學想想看，如何設計副程式呢？



問題
拆解

如何設定副程式？

4

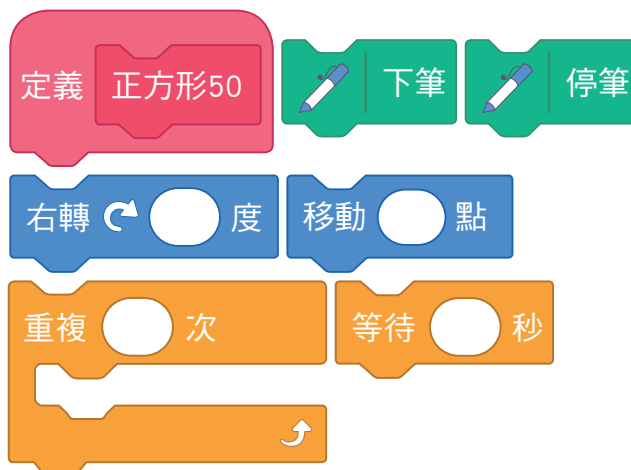
步驟
3新增畫正方形的函
式積木。① 點選小貓角色，再點選
函式積木類別。② 按下**建立一個積木**。③ 命名為**正方形 50**。④ 按下**確定**鍵。

5 請同學想想看，如何新增其他三個函式積木，並依序命名為正方形 100、正方形 150、正方形 200？



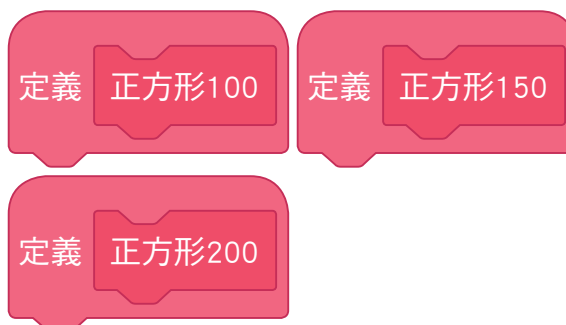
步驟 4 請依右方提示的積木進行組裝，完成畫正方形 50 的副程式。

請同學想想看，如何撰寫畫出邊長為 50 點正方形的副程式？



步驟 5 請依據上個步驟，完成畫正方形 100、畫正方形 150、畫正方形 200 的副程式。

請同學想想看，若複製 步驟 4 組裝完成的副程式進行調整，右方的積木要如何修改？

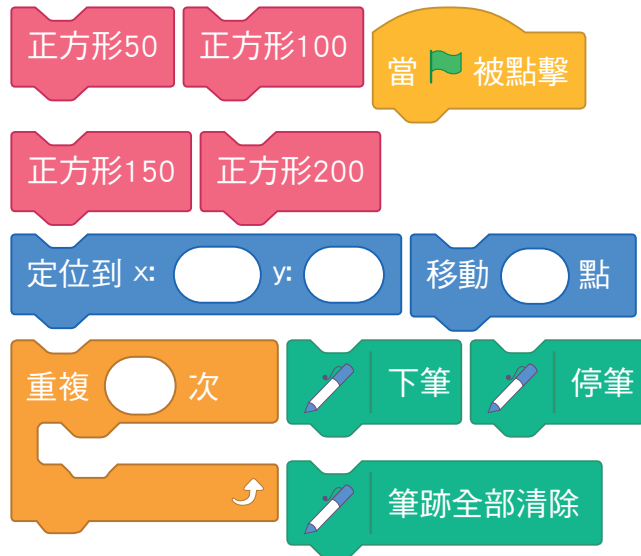


問題拆解 5 如何呼叫副程式？

5

步驟 6

請依右方提示的積木進行組裝，讓小貓向右畫出四個逐漸擴大的正方形。



問題拆解 6 如何利用副程式的參數將程式碼模組化？

6

請同學想想看，哪個部分可以使用副程式的參數，讓程式更簡化？

運用副程式的參數可以將四個副程式簡化成一個副程式，讓我們學習如何利用副程式的參數來重新撰寫此範例吧！



問題拆解 7 如何設定副程式的參數？

7

步驟 7

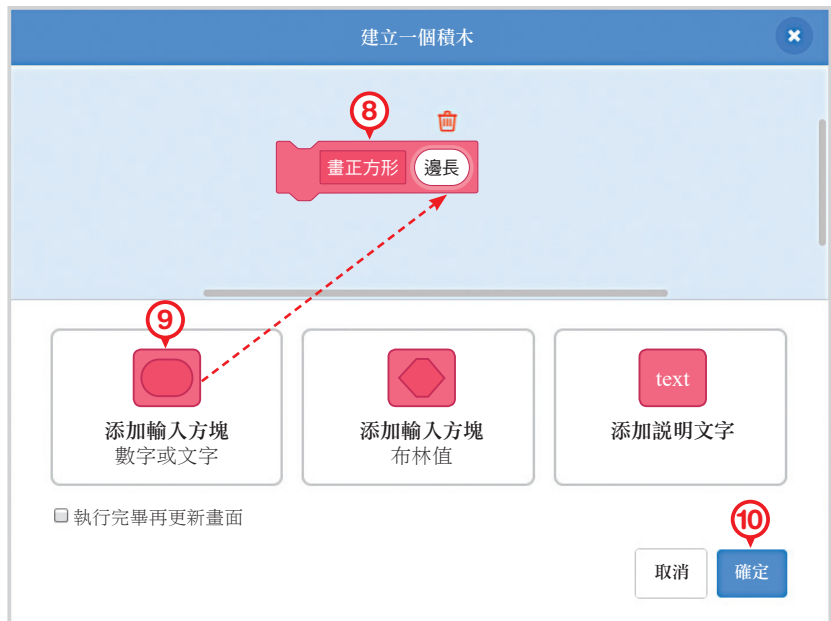
新增畫正方形的函式積木。

6 點選小貓角色，再點選函式積木類別

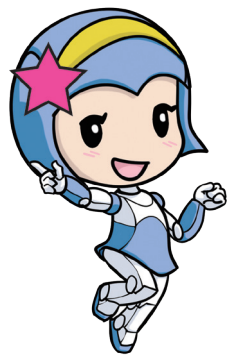
7 按下 建立一個積木。



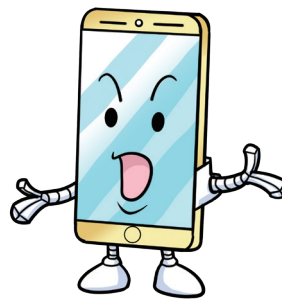
- 8 命名為畫正方形。
- 9 點選**添加輸入方塊**（**數字**或**文字**），並輸入**邊長**。
- 10 按下**確定**鍵。



建立畫正方形函式的方法，在上一個範例有學過。



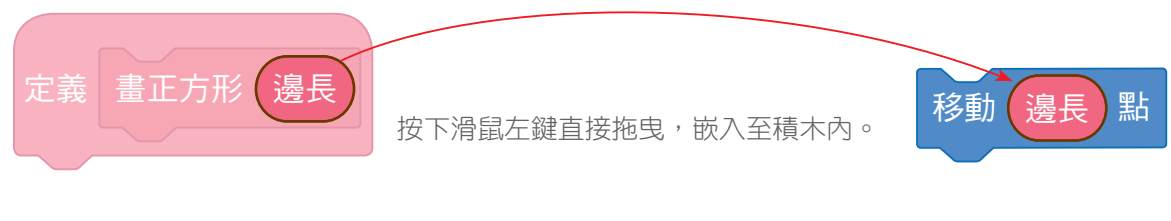
對阿！在撰寫比較複雜的程式時，會常常使用函式來簡化，所以一定要學會喔！



小提示

如何使用此積木？

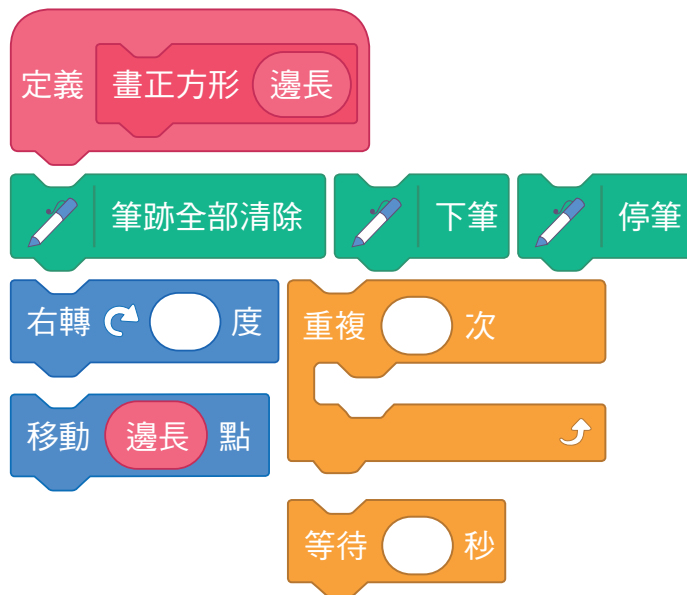
在撰寫副程式的參數時，可加以利用**添加輸入方塊**，來存入資料給副程式使用。



步驟 8

請依右方提示的積木進行組裝，完成畫不同大小正方形的副程式。

1. 如何讓程式執行的速度慢一些？
2. 此處為何不用「移動 點」，而是使用「移動 邊長 點」？
3. 組裝完成的副程式，與 [步驟 4](#)、[步驟 5](#) 的副程式有何不同呢？



問題拆解

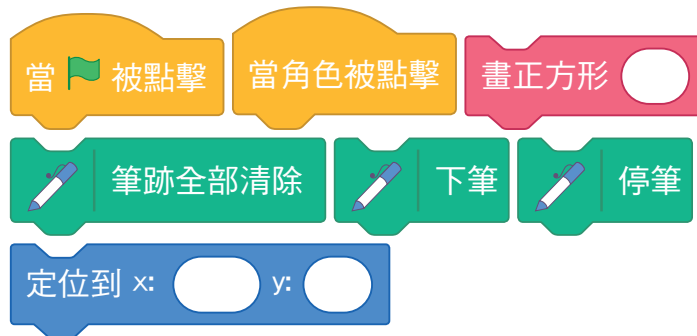
8

如何呼叫副程式的參數？

步驟 9

請依右方提示的積木進行組裝，讓小貓向右畫出四個逐漸擴大的正方形。

1. 請同學想想看，「畫正方形 」代表什麼意思？
2. 四個正方形的邊長要設為多少？



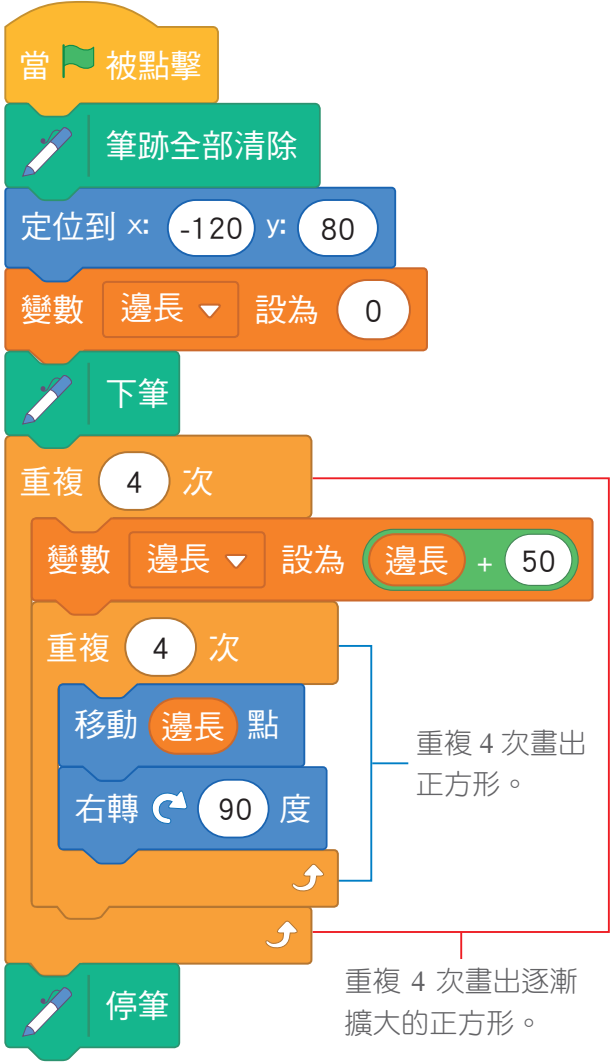
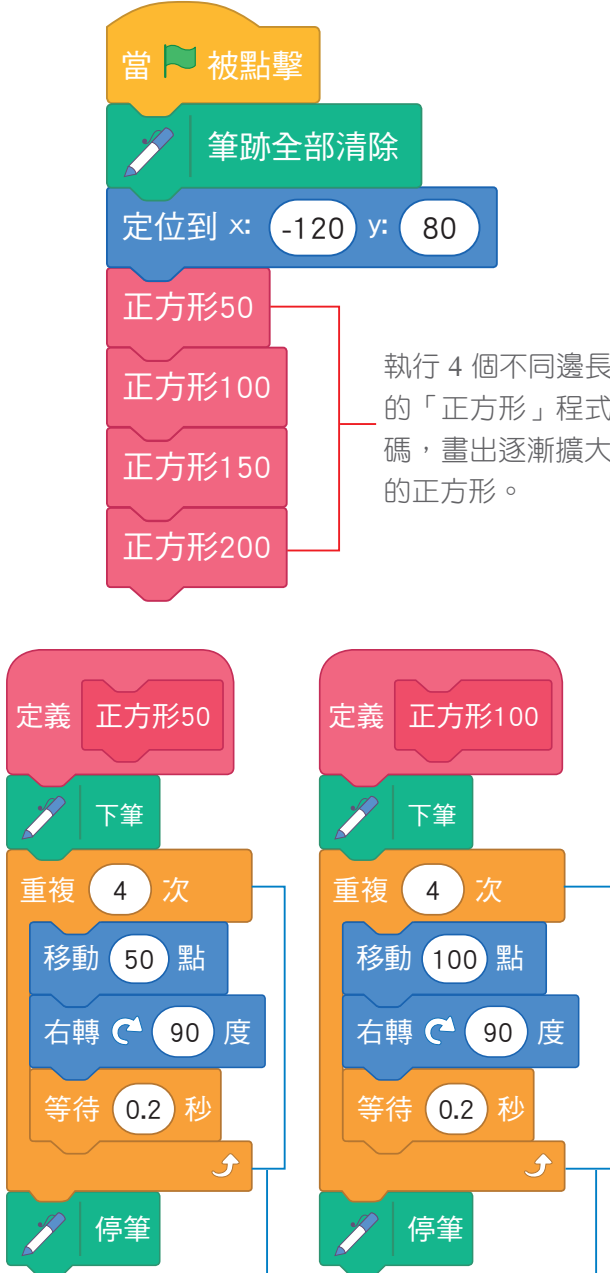
練習

想想看，如果小貓向左畫出四個逐漸擴大的正方形，要如何進行？

在此單元中，我學到的有：

1. 建立添加輸入方塊，以便傳資料進入函式。
2. 利用添加輸入方塊的函式來完成整個程式的某項小任務。

在這個範例中，我們使用不同的方法來完成《畫逐漸擴大的正方形》，說說看，這三種方式的優點與缺點為何？你比較喜歡哪一種方法？

模組化前	利用【副程式】
 <p>當 旗幟被點擊</p> <p>筆跡全部清除</p> <p>定位到 x: -120 y: 80</p> <p>變數 邊長 設為 0</p> <p>下筆</p> <p>重複 4 次</p> <p>變數 邊長 設為 邊長 + 50</p> <p>重複 4 次</p> <p>移動 邊長 點</p> <p>右轉 90 度</p> <p>停筆</p> <p>重複 4 次畫出逐漸擴大的正方形。</p> <p>重複 4 次畫出正方形。</p>	 <p>當 旗幟被點擊</p> <p>筆跡全部清除</p> <p>定位到 x: -120 y: 80</p> <p>正方形50</p> <p>正方形100</p> <p>正方形150</p> <p>正方形200</p> <p>執行 4 個不同邊長的「正方形」程式碼，畫出逐漸擴大的正方形。</p> <p>定義 正方形50</p> <p>下筆</p> <p>重複 4 次</p> <p>移動 50 點</p> <p>右轉 90 度</p> <p>等待 0.2 秒</p> <p>停筆</p> <p>重複 4 次畫出邊長為 50 點的正方形。</p> <p>定義 正方形100</p> <p>下筆</p> <p>重複 4 次</p> <p>移動 100 點</p> <p>右轉 90 度</p> <p>等待 0.2 秒</p> <p>停筆</p> <p>重複 4 次畫出邊長為 100 點的正方形。</p>



進行模組化後

利用【副程式的參數】進行模組化後

```

    定義 正方形150
    下筆
    重複 4 次
        移動 150 點
        右轉 90 度
        等待 0.2 秒
    停筆
    
```

重複 4 次畫出邊長為 150 點的正方形。

```

    定義 正方形200
    下筆
    重複 4 次
        移動 200 點
        右轉 90 度
        等待 0.2 秒
    停筆
    
```

重複 4 次畫出邊長為 200 點的正方形。

```

    當 被點擊
    筆跡全部清除
    定位到 x: -120 y: 80
    畫正方形 50
    畫正方形 100
    畫正方形 150
    畫正方形 200
    
```

執行 4 個不同參數的「畫正方形」程式碼，畫出逐漸擴大的正方形。

```

    定義 畫正方形 邊長
    下筆
    重複 4 次
        移動 邊長 點
        右轉 90 度
        等待 0.2 秒
    停筆
    
```

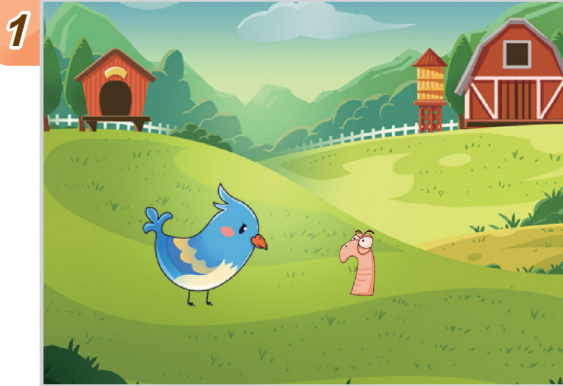
重複 4 次畫出邊長為指定參數值的正方形。

4-3 模組化程式設計的應用

運用副程式將七年級繪圖篇兩個畫正方形範例的程式碼，進行模組化設計後，程式碼是不是比較容易閱讀？接下來，讓我們用下面的範例來處理複雜一點的問題。

範例：小鳥吃蟲

在畫面中，以一個草原為背景，有一隻小鳥和一隻蟲，按下綠旗後，蟲會先隱藏並在隨機位置上產生十隻蟲，用滑鼠移動小鳥，在蟲上方按下滑鼠可以吃掉蟲，吃掉一隻蟲後，就會再產生一隻蟲在隨機的位置上。請執行《小鳥吃蟲》的程式，想一想這個範例的程式是如何運作？



範例執行前，畫面上有一隻小鳥和一隻蟲。



範例執行後，蟲會先隱藏並在隨機位置產生十隻蟲，按下滑鼠後，小鳥變換彎腰造型吃蟲。



蟲被吃掉後，小鳥換回站立造型，再於隨機位置產生一隻蟲。



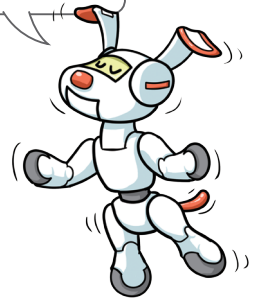
再按下滑鼠後，小鳥變換彎腰造型吃另一隻蟲。

問題分析

我們可以將這個程式範例拆解幾個部分如下：

- ① 如何建立背景？
- ② 如何建立蟲與小鳥角色？
- ③ 如何處理蟲角色分身與動畫，並利用副程式將程式模組化？
 1. 執行時，如何使用副程式，產生一隻在隨機位置出現的蟲？
 2. 執行時，如何利用副程式，產生十隻蟲？
 3. 執行時，如何讓蟲被小鳥吃掉後就消失，接著在隨機位置出現？

程式執行時會一直需要產生蟲的動作，所以可以包裝成一個獨立的模組，方便主程式重複呼叫。



按下滑鼠且鳥嘴碰到蟲



蟲消失



蟲在隨機位置出現



- ④ 如何處理小鳥角色動畫？

1. 執行時，如何讓小鳥隨著滑鼠移動？
2. 執行時，當滑鼠按下時，如何讓小鳥變換造型？

未按下滑鼠時



造型：小鳥站立



按下滑鼠時



造型：小鳥彎腰



解題步驟


問題
拆解

如何建立背景？

1

步驟
1

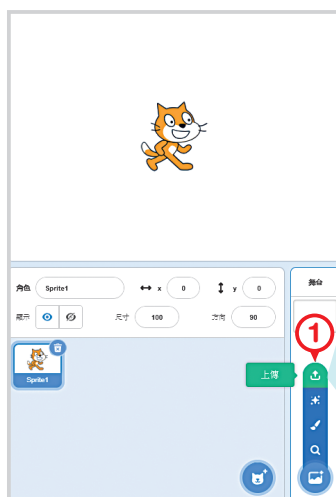
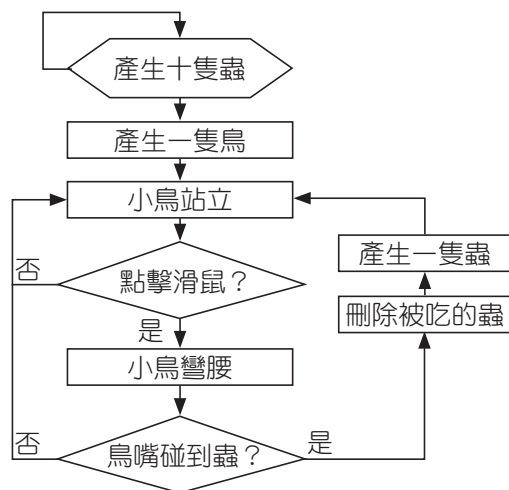
開新檔案，匯入舞臺背景。

1 在角色區中，點選下方**選個背景**按鍵列的 。

2 從電腦中，匯入**草原**背景。

小知識

流程圖



問題
拆解


如何建立蟲與小鳥角色？

2

步驟
2

新增蟲角色。


3 刪除小貓角色。

4 在角色區中，點選下方**選個角色**按鍵列的 。

5 從電腦中，匯入**蟲**角色。




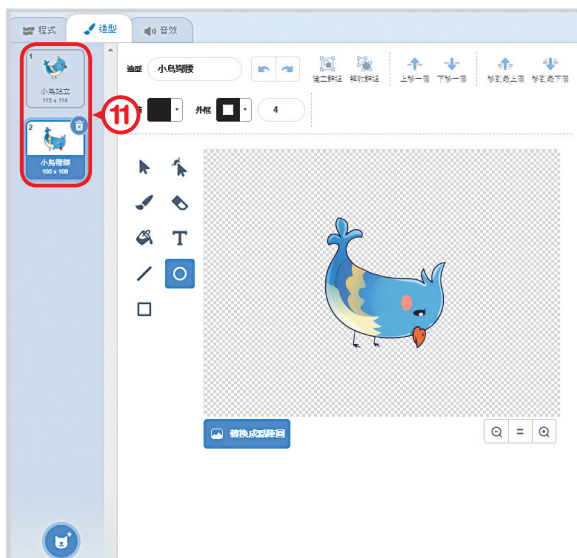
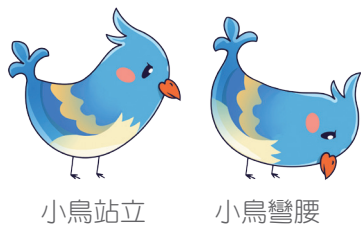
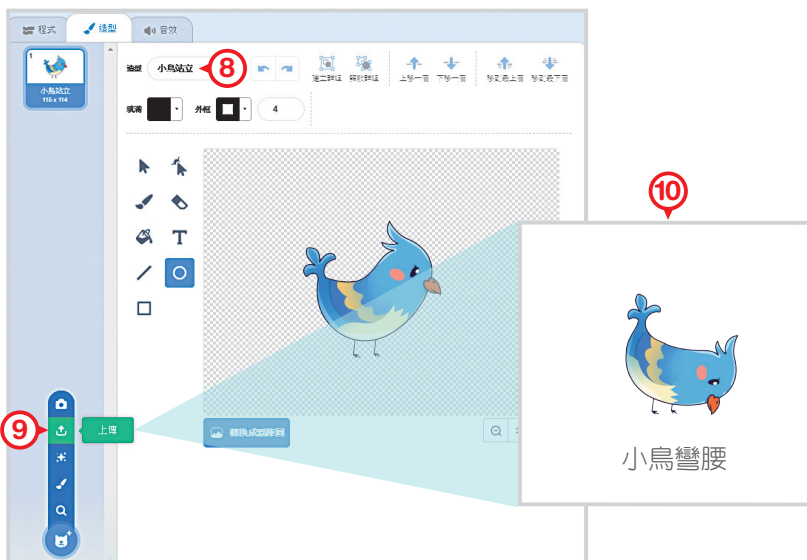
步驟 3 新增小鳥角色。

- ⑥ 在角色區中，點選下方**選個角色**按鍵列的 。
- ⑦ 從電腦中，匯入**小鳥**角色。



步驟 4 在小鳥角色中，匯入彎腰時的造型。

- ⑧ 將小鳥造型命名為**小鳥站立**。
- ⑨ 在造型面板中，點選下方**選個造型**按鍵列的 。
- ⑩ 從電腦中，匯入**小鳥彎腰**造型。
- ⑪ 小鳥角色有兩種造型。



問題
拆解

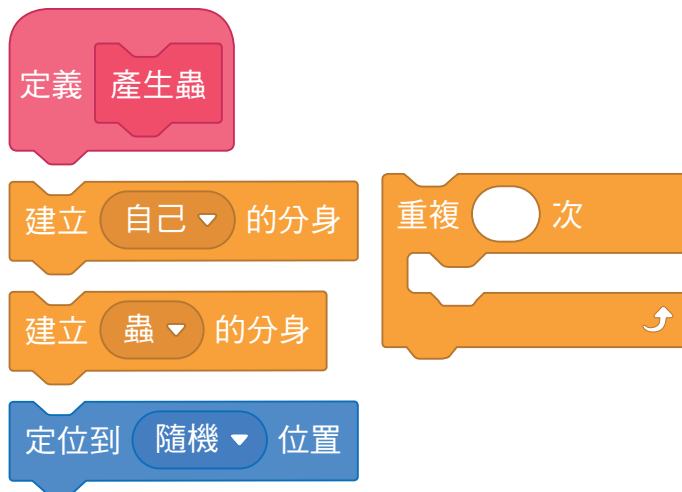
3

如何處理蟲角色分身與動畫，並
利用副程式將程式模組化？

步驟
5

請依右方提示的積木進行組
裝，設定產生蟲的副程式。

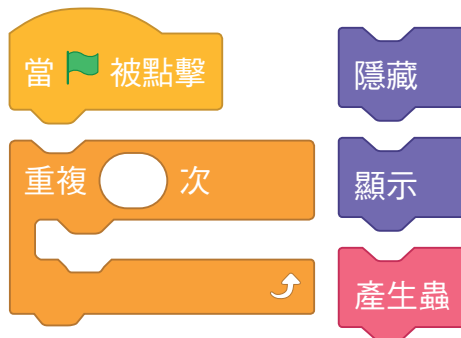
使蟲產生分身並隨機出現。



步驟
6

請依右方提示的積木進行組
裝，設定蟲出現的程式。

利用產生蟲的副程式，讓十隻蟲在
隨機位置出現。





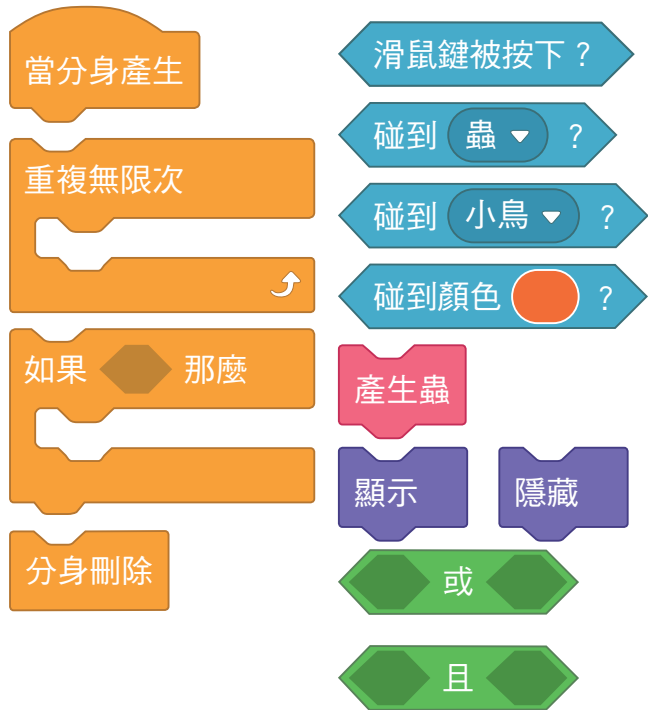
我要變出十隻分身，
製造出蟲蟲大軍！

在上學期的分身篇中，
本尊要隱藏，在這裡是
否也需要隱藏呢？

步驟 7

請依右方提示的積木進行組裝，完成蟲的動畫。

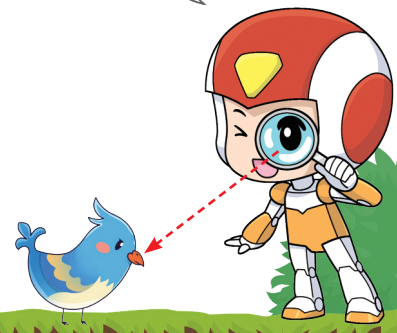
1. 還記得在上學期《螞蟻搬乳酪》的範例中，程式是如何判斷螞蟻找到乳酪的？
2. 請同學想想看，程式要如何判斷小鳥吃到蟲？
3. 說說看， 中 是指什麼的顏色？
4. 當小鳥吃到蟲，且蟲消失，要使用什麼積木？
5. 當蟲被吃掉後，要如何再產生一隻蟲？



因為要呈現小鳥吃蟲的動畫，而我們希望是小鳥嘴巴碰到蟲才會有反應。

但是小鳥嘴巴不是一個角色，要如何進行？

這時候，可以利用小鳥嘴巴的顏色來當判定條件！



請依右方提示的積木進行組裝，完成小鳥吃蟲的動畫。

1. 請同學想想看，如何讓小鳥隨著滑鼠游標移動？
2. 當按下滑鼠時，表示小鳥欲吃蟲，變換為小鳥彎腰的造型；未按下滑鼠時，則維持小鳥站立的造型。

當  被點擊

重複無限次 

如果  那麼

否則

空白  鍵被按下？

滑鼠鍵被按下？

造型換成 小鳥站立 

造型換成 小鳥彎腰 

定位到 隨機  位置

定位到 鼠標  位置




我不使用副程式，還是可以完成蟲角色的程式耶！

但是想變更蟲角色的設定，只要調整產生蟲的副程式，就可以更直覺、更方便的維護與修改程式！



在此單元中，我學到的有：

1. 利用分身，隨機產生角色。
2. 使用  來進行事件驅動程式。

重點回顧

● 模組化的概念

在撰寫程式解決問題的過程中，若是能夠善用模組化的概念，可將原有的問題拆解成較小的問題，然後分別去解決。

● 副程式的概念

在程式設計方面，可以透過撰寫副程式來達到模組化的成效，將功能獨立的部分，分別寫成一個一個的模組後，再進行組裝，例如：將獨立的小任務先分別寫成不同的副程式，等到需要用到時，再撰寫主程式去呼叫。這樣不僅可讓程式碼較容易閱讀、維護與修改，也有助於多人合作開發大程式。

● 定義副程式

在定義副程式時，為了讓副程式更具彈性，也可以同時為這個副程式定義「參數」。參數是我們想要為此副程式彈性調整的變數，這些變數可以是數字、字串或布林值等不同資料型態，在 Scratch 中則是使用函式積木的**添加輸入方塊**來描述參數，因此在呼叫此類副程式時，必須要傳入相對應的參數值。

● 模組化程式設計

透過本章將程式進行模組化，並使用七年級繪圖篇兩個畫正方形的範例進行實作，經由程式碼進行模組化前後的比對，更能清楚了解程式模組化後，程式碼的呈現更容易閱讀。

