



四、本教材資訊科技篇的特色如下：

- (一) 章節結構嚴謹，能充分反映國中資訊科技課程綱要的學習重點。
- (二) 重視運算思維歷程，培養學生能面對問題、提出問題、分析問題、並解決問題的能力。
- (三) 圖文並茂讓課本內容生動活潑，能吸引學生注意並引起學習興趣。
- (四) 簡明扼要，以保留資訊科技發展快速之內容的適時性。
- (五) 強調實/操作，除可避免傳統記誦教學外，也可減輕學生學習的負擔，同時可提昇學生的學習動機，維持學習的興趣。
- (六) 具有彈性，所有硬體及軟體未明確指定設備或環境，此種考慮也是為了增加適時性，選擇教學當時最適當的教學及學習環境。
- (七) 強調合作與共創學習，以培養學生團隊合作的態度與精神。
- (八) 視學習成果，主要考慮維持學生的學習動機與學習成就。
- (九) 配合學生之認知能力發展，個人的生活及學習經驗，使學生在課堂中能愉快且自然地學習資訊科技之基本能力及技能。

五、本教材期望能夠讓教師在授課時，能夠有豐富的參考資源，依據自己的教學需求，選用適切的活動或內容以進行教學。此外，學生也能夠把本教材當作是一本最棒的工具書，當遇到日常生活中的相關問題時，可以隨時拿出來查詢，以協助他們解決問題。

六、本教材如有未盡周詳之處，敬請各校教師隨時提供意見，以供修訂時之參考，不勝感激。



阿顯



星兒



拜特



波波



吉嘎

努力、勇於挑戰。

溫柔、善解人意。

熱心、大家的資訊站。

穩重、憨厚又樂於助人。

聽話、喜歡跟著星兒。



目次

**壹**

生活 科技篇

**關卡
4**

動力與運輸

挑戰 1 運輸科技系統**挑戰 2** 運輸系統的形式**挑戰 3** 運輸載具與動力運用

- 闖關任務**
1. 動力保養大挑戰
 2. 太陽能發電動力車

**關卡
5**

製作電動液壓 動力機械手臂

貳

資訊 科技篇

第 4 章

進階程式設計(2) 120

- | | | |
|------------|------------|-----|
| 4-1 | 模組化的概念 | 122 |
| 4-2 | 認識模組化程式設計 | 126 |
| 4-3 | 模組化程式設計的應用 | 142 |

第 5 章

媒體與資訊 科技相關社 會議題

148

- | | | |
|------------|---------|-----|
| 5-1 | 媒體與資訊科技 | 150 |
| 5-2 | 資訊失序 | 152 |
| 5-3 | 言論自由 | 157 |
| 5-4 | 網路霸凌 | 159 |
| 5-5 | 網路成癮 | 164 |



自己動手操作
與觀察，培養
「做」的能力。



查詢相關資料或
報告分享，學習
使「用」科技產
品。



用腦力激盪設計
與批判科技，培
養「想」的能
力。



課程內容的延伸
補充，增強知識
的深度與廣度。



依條件進行闖關，
透過做、用、想解
決問題，並分享成
品。



運輸科技對社會與環境的影響

關卡
6

挑戰 1 運輸對社會的影響

挑戰 2 運輸對環境的影響

闖關任務 1. 求職博覽會

2. 科技達人追追追

第6章 基本演算法的介紹

166

6-1	演算法概念與原理	168
6-2	排序的原理與範例	170
6-3	搜尋的原理與範例	191

附錄

生科圖片來源	
資訊圖片來源	228
資訊中英文對照	229
第4章 程式參考解答	230
第5章 資訊失序補充案例	235
第6章 程式參考解答	236
第6章 補充演算法	244
附件1 方格紙	
附件2 三角格紙	
附件3 液壓手臂	



課程內容的延伸補充，增強知識的深度與廣度。



針對陌生、易產生迷思的概念，進行補充及說明。



有系統的歸納重要觀念、運算、程式…等。



活用課文中提到的概念去延伸學習。



將範例問題做步驟式提問，引導學生思考。

第 4 章

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行程式設計吧！

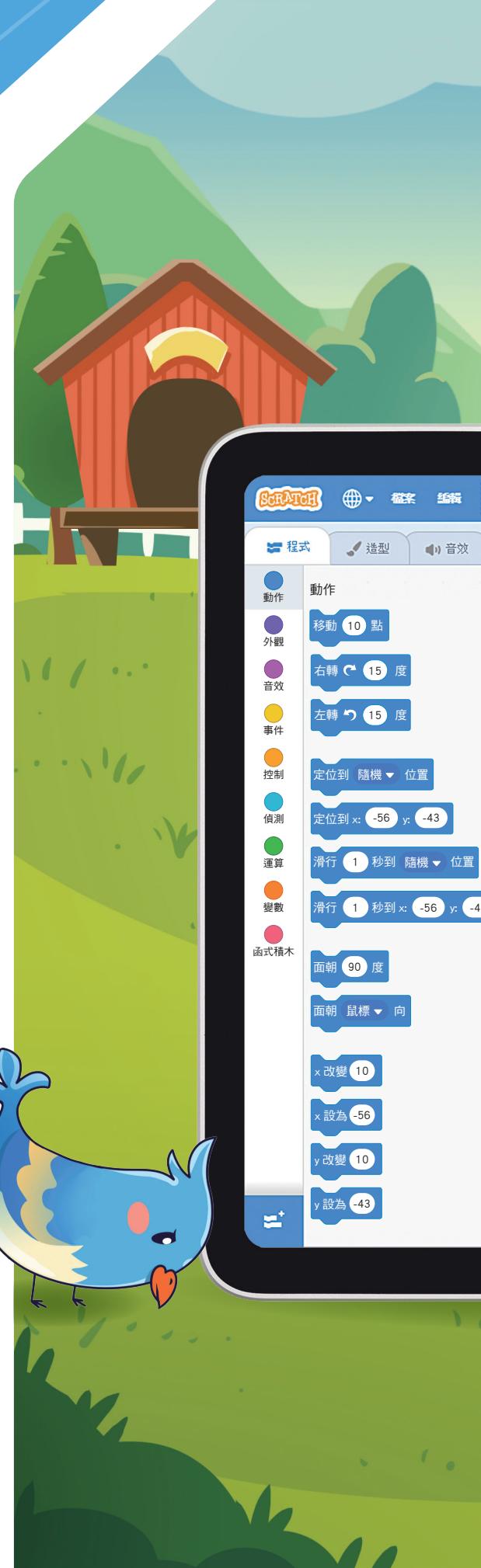


※本教材使用 Scratch 3.0 版本，因 Scratch 會不定期更新，介面可能與本教材略有不同。

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

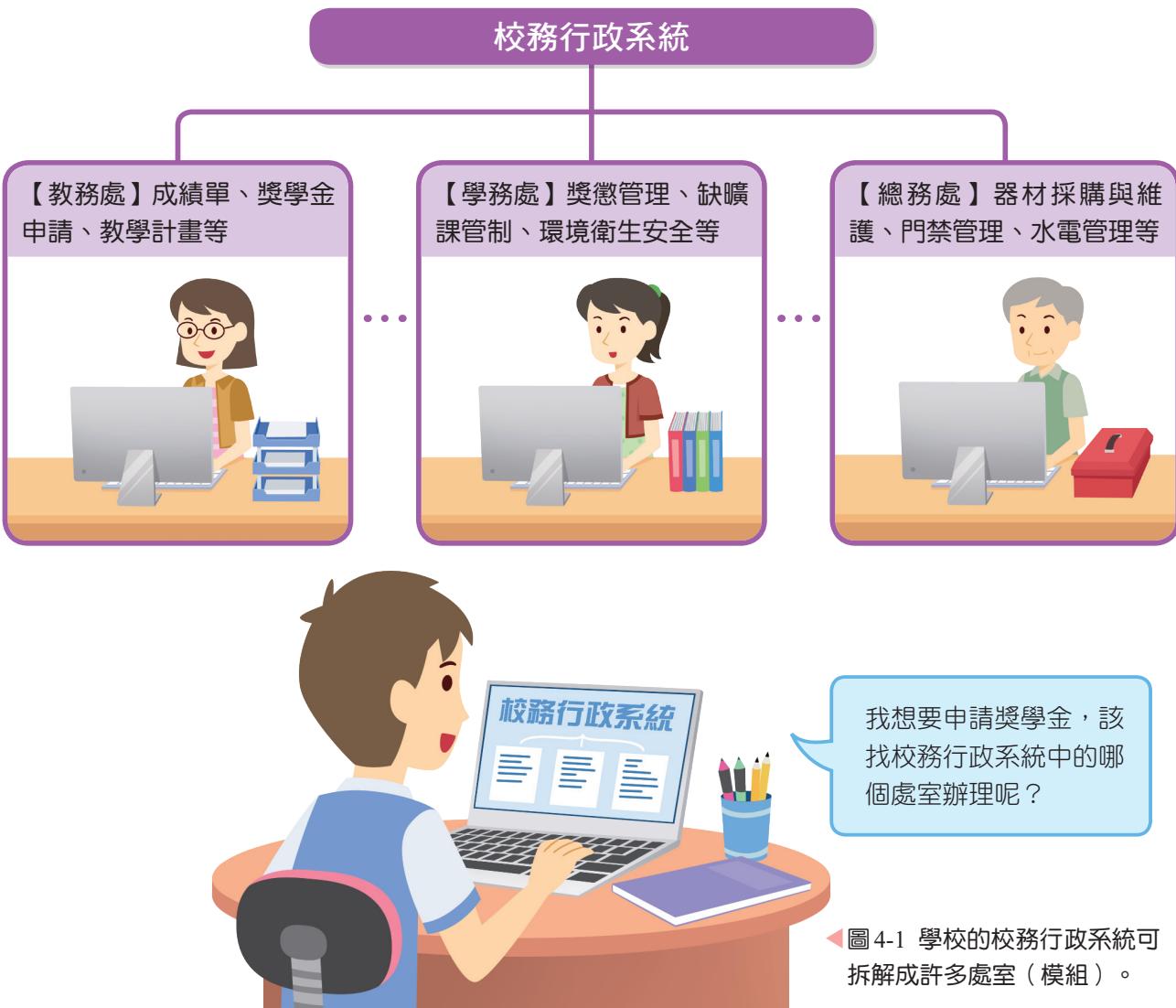




4-1 模組化的概念

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。

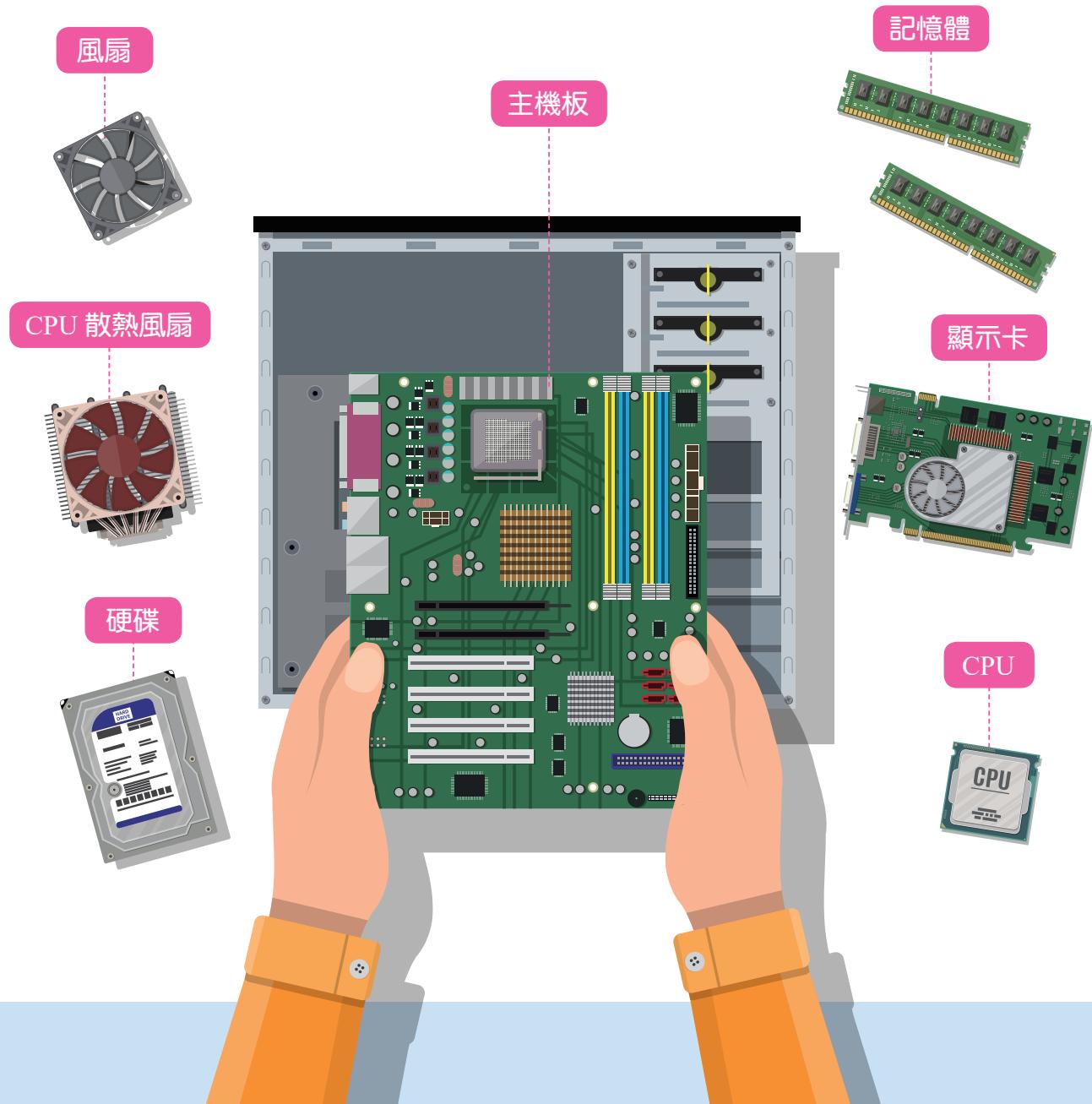
在學校的校務行政系統中，就經常用到模組化的概念。例如：各處室（如教務、學務、總務等）業務系統設計，都可各自設計成可以拆解的模組，各自發展完成後，再整合成為一個完整的校務行政系統（圖 4-1）。



◆ 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，可分別進行更換與升級。



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。例如：將畫三角形、畫正方形、畫星星等三項獨立的任務分別寫成三個不同的副程式，等到需要用到時，再由主程式去呼叫。

小知識

副程式

副程式（sub-program）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、且可節省程式所佔用的記憶體，以及節省重複撰寫程式的時間。



▲圖 4-3 運用副程式使撰寫程式更清楚明瞭（以 Scratch 為例）。

在軟體開發的實務中，經常會有多人合作撰寫程式的情形，透過模組化設計，就可以將問題拆解成獨立的小模組，然後將工作指派給不同的人完成。此外，模組化也有助於增進程式碼的可讀性。例如：在 Scratch 中，使用者可以用自訂函式積木來撰寫副程式，我們將畫三角形、畫正方形、畫星星分別寫成三個獨立的副程式，因此，在主程式中，便可以更專注於描述任務本身，而不是完成任務的細節（圖 4-3）。

The screenshot shows the Scratch interface with a script editor on the left and a stage with a cat sprite on the right. The script consists of the following blocks:

- 当 [绿旗] 被点击
- 笔迹全部清除
- 定位到 x: -150 y: 0
- 画三角形
- 移动 (80) 点
- 画正方形
- 移动 (80) 点
- 画星星
- 移动 (80) 点

On the right side of the stage, there is a speech bubble containing the text: "看我的厲害！利用副程式，就可以輕鬆執行畫出三角形、正方形和星星的複雜程式了！" Below the stage is a cartoon smartphone character with arms and legs, standing next to a chalkboard.

在此單元中，我學到的有：

1. 撰寫副程式。
2. 利用若干個副程式，整合成一個完整的繪圖程式。

4-2 認識模組化程式設計

運用副程式，我們可以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，不僅可以讓程式碼容易閱讀、維護與修改，也有助於多人合作開發大程式。

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以及逐漸擴大的正方形，接下來，讓我們來學習怎麼將程式碼模組化。

4-2-1 畫平行排列的正方形



範 例

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



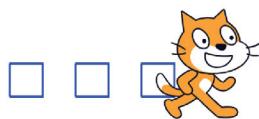
2



範例執行前

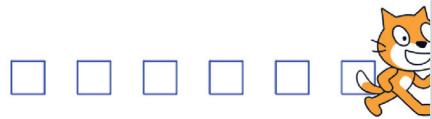
範例執行後，小貓從定位的位置先畫出一個正方形。

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？
 - 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



還記得七年級學過巢狀結構的 Scratch 程式嗎？

讓我們透過 ①、② 來複習吧！

問題
拆解

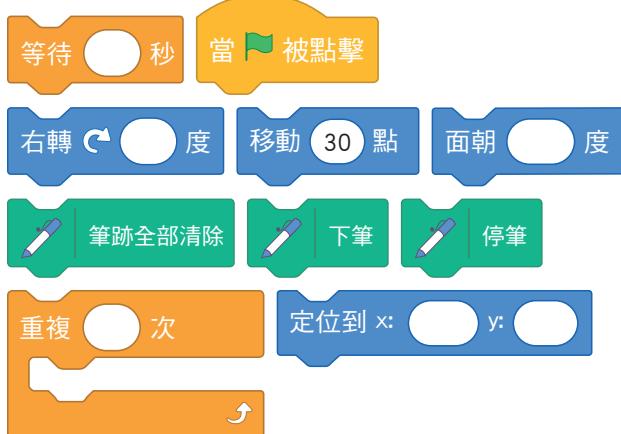
1

如何畫出正方形？

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. 說說看，正方形的每一個內角是幾度？
3. **移動 30 點** 是代表什麼意思？
4. 如果將 **右轉 15 度** 改成 **左轉 15 度**，
執行的結果有何不同？
5. 如何讓程式執行的速度慢一些？
6. 如果將 **右轉 15 度** 改用 **面朝 15 度**，
要怎麼組裝積木呢？



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

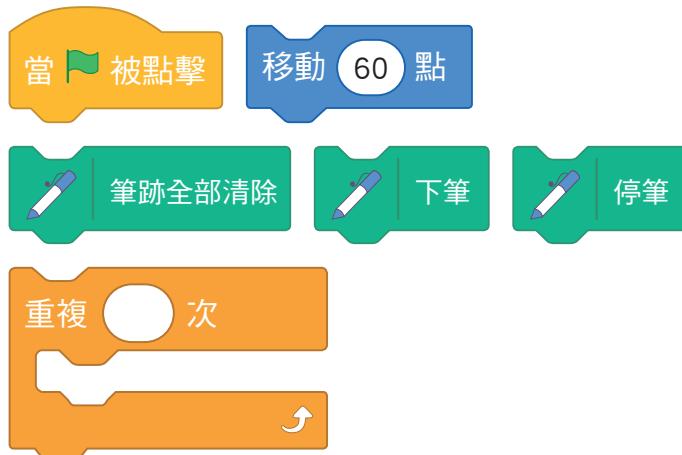
2

如何畫出六個間隔相同
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，再與右方的積木進行組裝。

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？

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**。



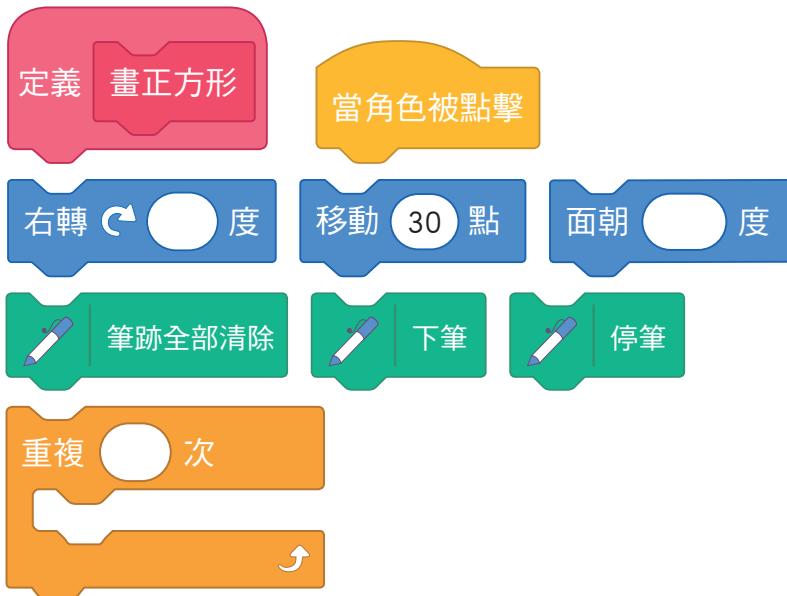
- ③ 命名為**畫正方形**。
- ④ 按下**確定**鍵。



**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。

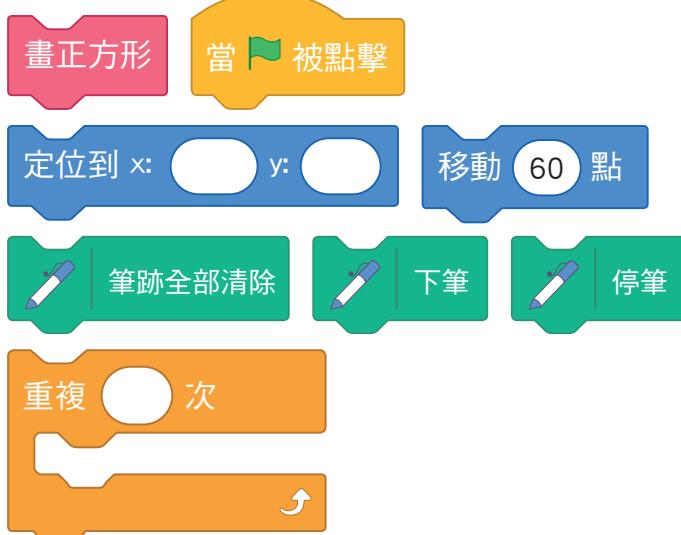
組裝完成的程式，與**步驟2**的程式有什麼地方相同？有什麼地方不同？

**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。

**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：

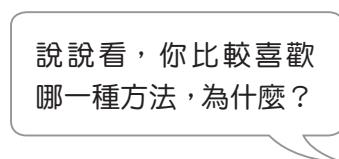
1. 將整個大任務切割成幾個小任務。
2. 利用函式來完成整個程式的某項小任務。

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。

模組化前	模組化後
<pre> when green flag clicked clear pen go to x: -140 y: -20 repeat (6) pen down repeat (4) move (30) steps turn (90) degrees end move (60) steps end end </pre> <p>重複 4 次畫出 邊長為 30 點 的正方形。</p> <p>重複 6 次畫出 間隔 60 點的 六個正方形。</p>	<pre> when green flag clicked clear pen go to x: -140 y: -20 repeat (6) draw square move (60) steps end end </pre> <p>重複 6 次執行 「畫正方形」 程式碼，每次 間隔 60 點。</p>
	<pre> define draw square pen down repeat (4) move (30) steps turn (90) degrees end pen up end </pre> <p>重複 4 次畫出 邊長為 30 點 的正方形。</p>



想想看，其他重複執
行的程式，可以設為
副程式嗎？



 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），所以電腦只用最少的記憶體大小來儲存此類變數。

4-2-2 畫逐漸擴大的正方形

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）等不同型態。在 Scratch 中則是使用**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。

 範例

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

2

範例執行前

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 如何畫出四個不同大小的正方形？

1. 如何畫出正方形？

② 如何利用副程式將程式碼模組化？

③ 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？

④ 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？

⑤ 如何利用副程式的參數將程式碼模組化？

⑥ 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？

⑦ 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



還記得七年級學過巢狀結構的 Scratch 程式嗎？

讓我們透過 問題
拆解 來複習吧！

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。



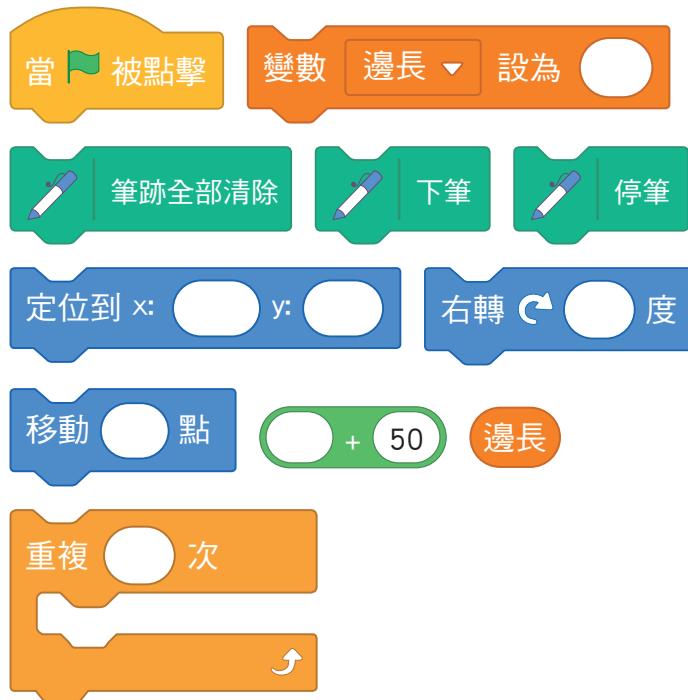
問題
拆解

1 如何畫出四個不同大小的正方形？

步驟
1

請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. 請同學想想看，如何利用變數畫出正方形？
2. 請同學想想看，設定邊長的變數，目的為何？

問題
拆解

2 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？

3

步驟
2

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。

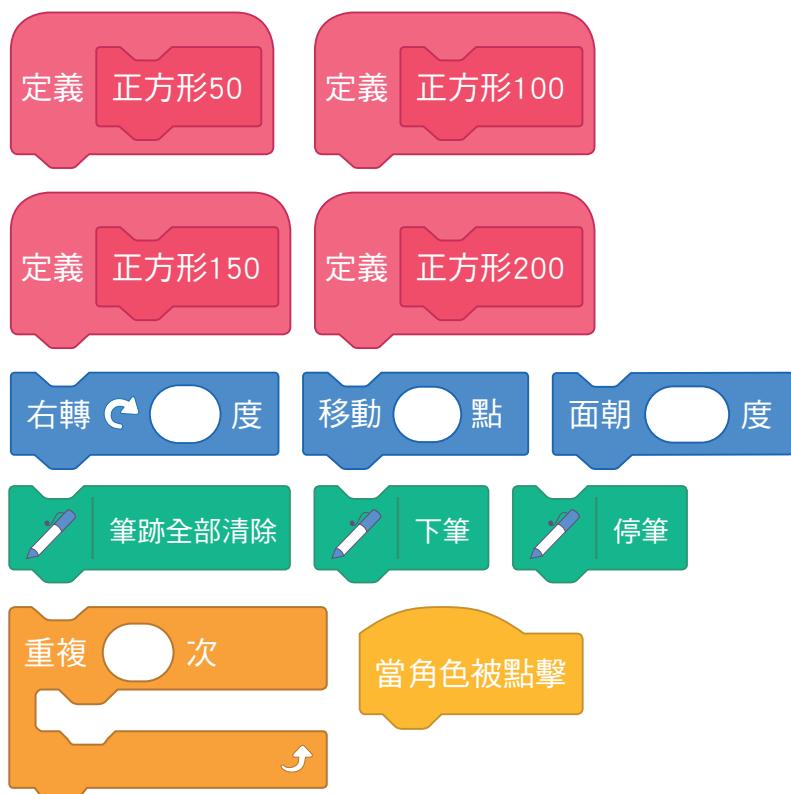
- ⑤ 請同學想想看，如何新增其他三個函式積木，並依序命名為正方形 100、正方形 150、正方形 200？



步驟
3

- 請依右方提示的積木進行組裝，完成四個畫正方形的副程式。

請同學想想看，為什麼要設定四個畫正方形的副程式？



問題
拆解

4

如何呼叫副程式？

步驟
4

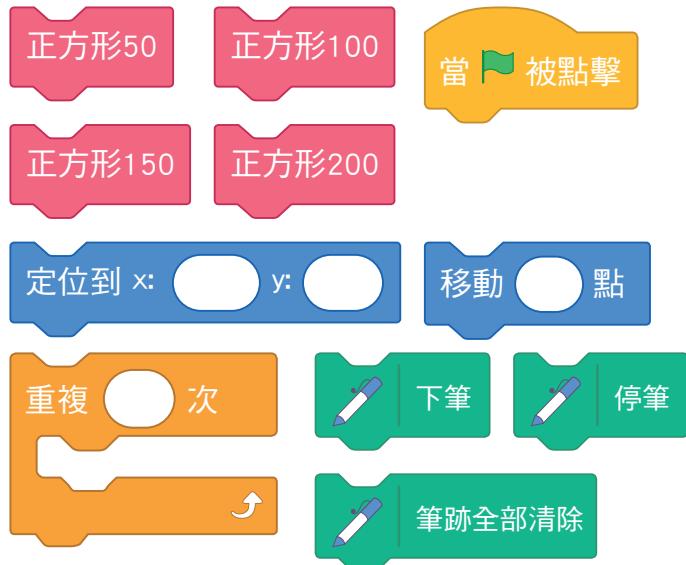
請依右方提示的積木進行組裝，讓小貓向右畫出四個逐漸擴大的正方形。

問題
拆解

5

如何利用副程式的參數將程式碼模組化？

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

6

如何設定副程式的參數？

步驟
5

新增畫正方形的函式積木。

6

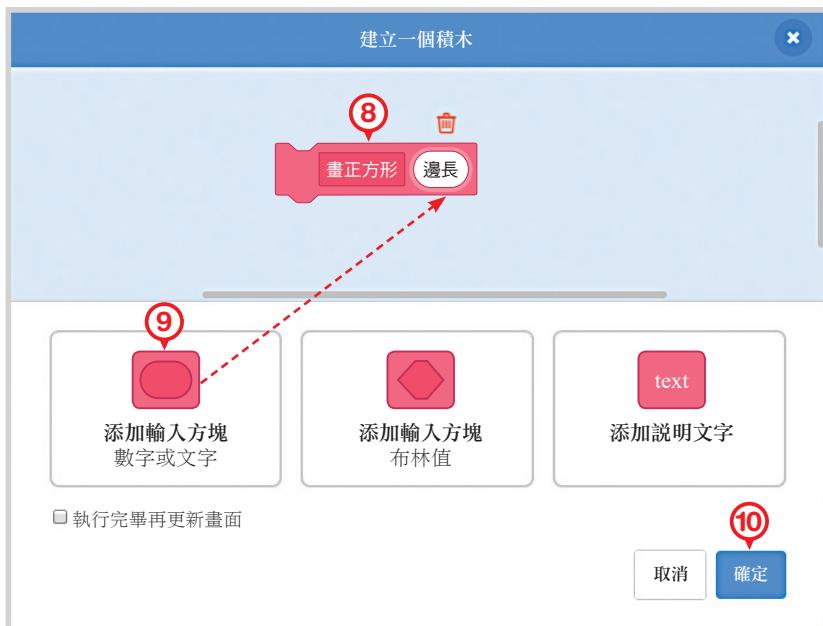
點選小貓角色，再點選**函式積木**類別。

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！



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。

定義 **畫正方形** **邊長**

移動 **邊長** **點**

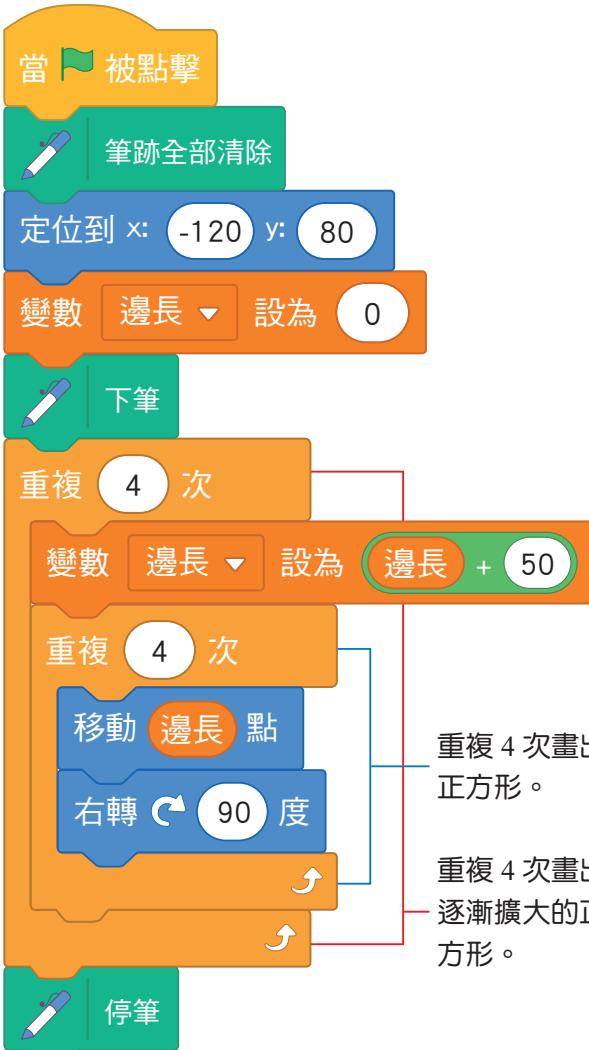
按下滑鼠左鍵直接拖曳，嵌入至積木內。

步驟
6

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？

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】
 <pre> 当 [绿旗] 被点击 笔迹全部清除 定位到 x: -120 y: 80 变数 [边长 v] 设为 0 下笔 重复 (4) 次 变数 [边长 v] 设为 [边长 + (50)] 移动 [边长] 步 右转 (90) 度 停笔 </pre> <p>重複 4 次畫出正方形。</p> <p>重複 4 次畫出逐漸擴大的正方形。</p>	 <pre> 当 [绿旗] 被点击 笔迹全部清除 定位到 x: -120 y: 80 正方形 (50) 正方形 (100) 正方形 (150) 正方形 (200) </pre>
	 <pre> 定义 [正方形50] 下笔 重复 (4) 次 移动 (50) 步 右转 (90) 度 等待 (0.2) 秒 停笔 </pre>  <pre> 定义 [正方形100] 下笔 重复 (4) 次 移动 (100) 步 右转 (90) 度 等待 (0.2) 秒 停笔 </pre>



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
```

```
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
```

利用【副程式的參數】進行模組化後

```
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
```

```
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
```

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。

範例一 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？

1



範例執行前，畫面上有一隻小鳥和一隻蟲。

2



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。

3



蟲被吃掉後，小鳥換回站立造型，再於隨機位置產生一隻蟲。

4



再按下滑鼠後，小鳥變換彎腰造型吃另一隻蟲。

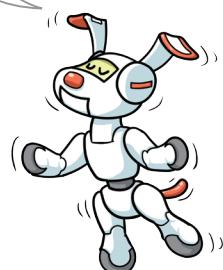


問題分析

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 如何將產生蟲寫成一個獨立的模組？
- ④ 如何用滑鼠控制小鳥吃蟲，並變換造型？

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。



解題步驟

問題拆解

1

如何建立背景？

步驟 1

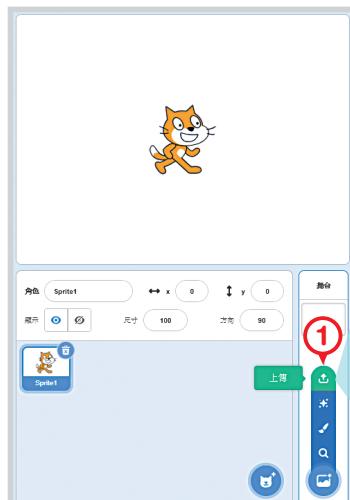
開新檔案，匯入舞臺背景。

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。



2



草原

問題拆解

2

如何建立蟲與小鳥角色？

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。



5



蟲

**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。

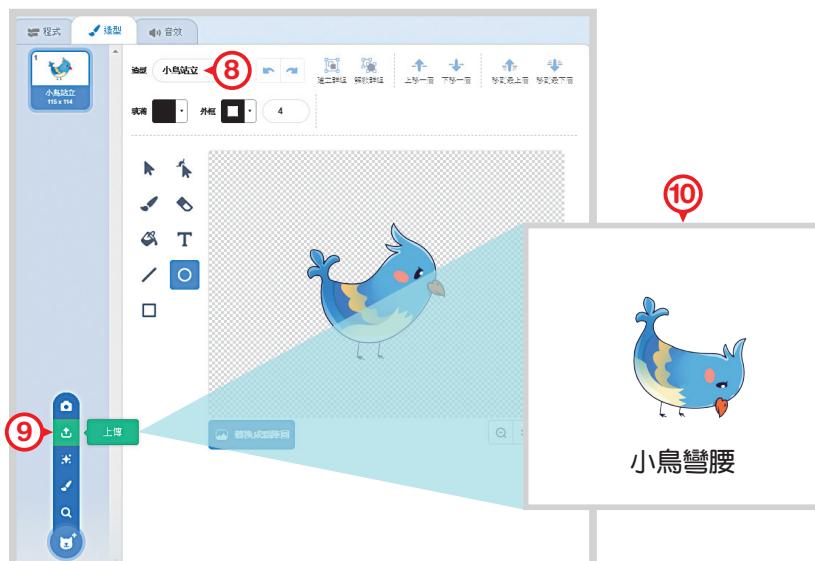


**步驟
4** 在小鳥角色中，匯入彎腰時的造型。

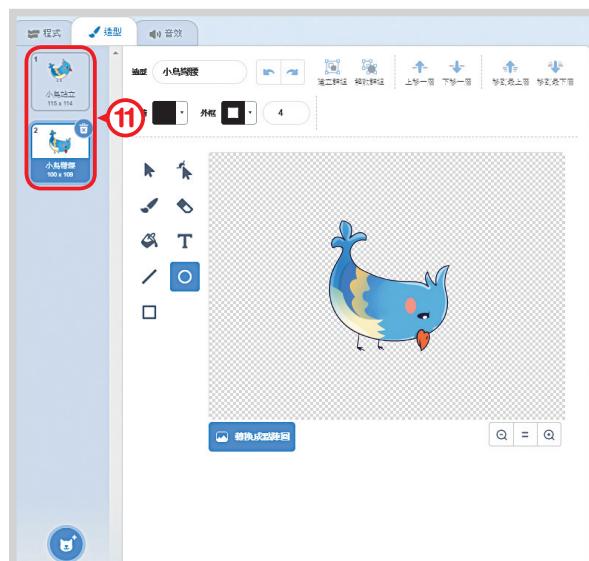
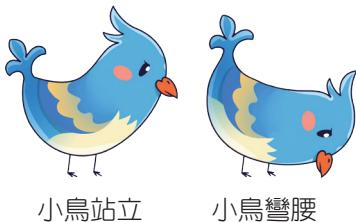
8 將小鳥造型命名為小鳥站立。

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

如何將產生蟲寫成一個獨立的模組？

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。

定義 產生蟲

建立 自己 ▾ 的分身

重複 次

建立 蟲 ▾ 的分身

定位到 隨機 ▾ 位置

步驟
6

請依右方提示的積木進行組裝，完成蟲的動畫。

- 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？
- 請同學想想看，程式要如何判斷小鳥吃到蟲？
- 說說看，碰到顏色 ? 中 ? 是指什麼的顏色？
- 當小鳥吃到蟲，且蟲消失，要使用什麼積木？
- 當蟲被吃掉後，要如何再產生一隻蟲？

當分身產生

滑鼠鍵被按下？

重複無限次

碰到 蟲 ▾ ?

如果 那麼

碰到 小鳥 ▾ ?

分身刪除

碰到顏色 ?

產生蟲 顯示 隱藏

或

且

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，那要如何進行？

這時候，可以利用小鳥嘴巴的顏色來當判定條件！



步驟
7

請依右方提示的積木進行組裝，設定蟲出現的程式。

為什麼不直接建立自己的分身，而是執行產生蟲的副程式？



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？

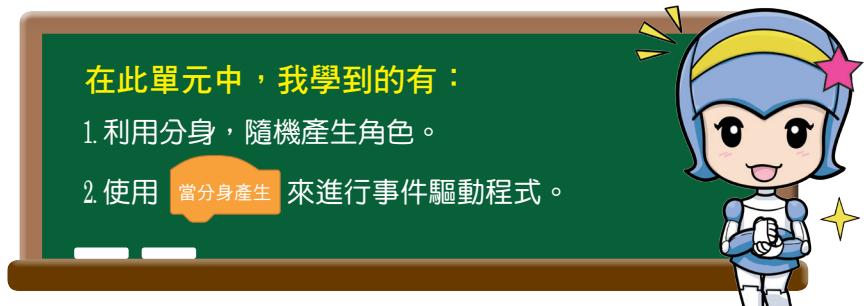
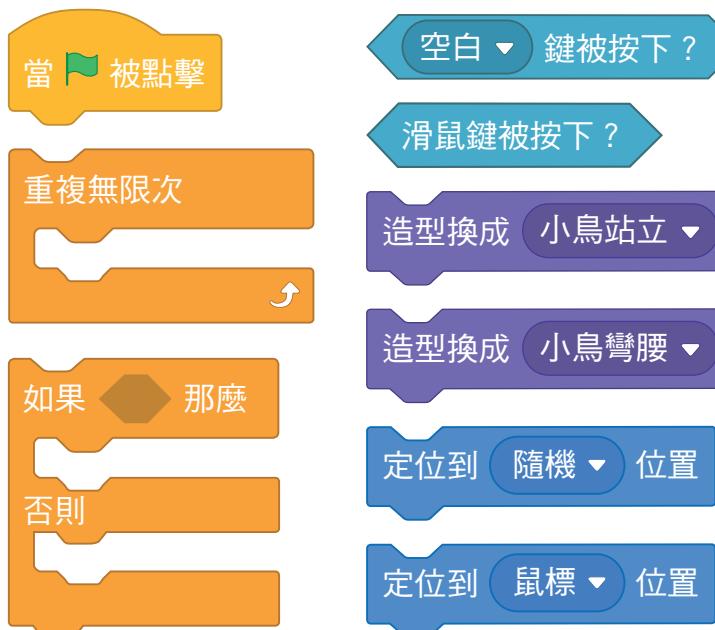
問題
拆解
4

如何用滑鼠控制小鳥吃蟲，並變換造型？

步驟
8

請依右方提示的積木進行組裝，完成小鳥吃蟲的動畫。

請同學想想看，如何讓小鳥隨著滑鼠移動？





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，有助於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型態，在 Scratch 中則是使用**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。

將七年級繪圖篇兩個畫正方形的範例程式進行模組化後，你覺得程式模組化的前後有什麼差異呢？



第 5 章

媒體與資訊科技相關 社會議題

本章首先說明媒體與資訊科技的關係，在各行業日漸資訊化之後，媒體與資訊科技也相輔相成；資訊科技增強了傳統媒體的播放／傳輸功能，使用者的角色也擁有產出／播放資訊的主動權。因為資訊化傳播的量與速度遠高於傳統媒體，人人具有表達的主動權，以言論為表達媒介的人類社會，也衍生多元且複雜的社會議題。

接著討論三個與言論有關的議題，也就是資訊失序、言論自由、以及網路霸凌。而有些使用者過度沉迷網路，以致有成癮傾向的現象也在本章一併討論。

資訊科技使用者在上述議題所表現的行為，往往介於正當與逾越之間，要取得平衡點則有賴個人的媒體／資訊素養了！

- 5-1 媒體與資訊科技**
- 5-2 資訊失序**
- 5-3 言論自由**
- 5-4 網路霸凌**
- 5-5 網路成癮**





5-1 媒體與資訊科技

媒體是指承載、傳輸和控制資訊的材料及工具；資訊科技泛指用於管理與處理資訊所採用各種技術的總稱。為了善用媒體，媒體界提出使用者要具備媒體素養（media literacy），又稱媒體識讀或媒介識讀，也就是指一般使用者能夠理解及詮釋所傳達的資訊，並參與媒體資訊的產出。麥克盧爾（McClure）於 1994 年表示資訊科技能以多元的方式呈現文字、圖形、聲音、多媒體等媒介，因此提出資訊素養（information literacy）的概念。**資訊素養**是指個人知道何時需要資訊，並能有效找尋、評估和正確利用資訊的能力。資訊素養有助於媒體素養的提升，也可強化大眾媒體的運用，兩者相輔相成。

媒體包括平面媒體（如報紙、雜誌）及電子媒體（如廣播、電視）等大眾媒體。近年來因網路科技的快速發展，又產生了新的傳播媒體，例如：影音分享平臺（如 YouTube）、直播（live broadcast）等（圖 5-1）。此種發展對媒體傳播資訊的方式起了很大的變化，過去民眾都是被動的接受傳播媒體發布的資訊；而現在人人可以發布資訊，擁有散布資訊的主動權。也因為如此，資訊的產生日趨多元且複雜。

◆圖 5-1 直播示意圖。



在網路未出現前，傳統媒體與資訊科技似乎涇渭分明，網路問世後，兩者的界線愈來愈模糊，而社群媒體、影音分享平臺興起後，傳統媒體與資訊科技幾乎合而為一，目前資訊幾乎都藉由網路傳播，其傳遞訊息的速度遠超過傳統媒體，其影響也超乎我們的想像。網際網路的蓬勃發展，透過搜尋引擎，就能很快找到資訊。現今資訊供應量快速增加，但卻未必正確，也不一定能被正確利用，或是對使用者有益。資訊通常需經過分析與組織後，才能產生正面的效果，這就要仰賴個人的媒體或資訊素養來處理資訊。做為一位資訊社會的數位公民，不論是資訊的接收者或是散布者，具備充分的媒體或資訊素養是資訊社會不可或缺的能力。

在充滿媒體資訊的生活及工作環境中，有關資訊失序（information disorder）、言論自由、網路霸凌（cyberbullying）、網路成癮（internet addiction）等（圖 5-2），都頗受社會各界關注，以下將分別討論這些議題。



▲圖 5-2 網路霸凌與網路成癮示意圖。

5-2 資訊失序

目前網路科技十分便捷，資訊的交換與傳播速度遠高於傳統社會，不肖分子就利用此種利器，擾亂社會秩序，甚至危害國家安全。由於不實資訊流傳頻繁，因此有平面媒體刊登專欄「謠言終結站」，藉以澄清資訊。德國哥德堡大學（University of Gothenburg）有大型政治學跨國調查計劃（Varieties of Democracy，簡稱 V-Dem），曾報導各國遭受外國假資訊攻擊的程度，臺灣名列全球第一。也有工商服務顧問公司（Ipsos）針對 25 國逾 2 萬 5000 名網路使用者所做的年度調查，顯示有百分之八十六的受訪者曾被假新聞（fake news）愚弄。

目前假新聞與不實資訊氾濫，大家都知道其危害的嚴重性，但要查證並非易事，也是非常棘手的問題，以致政府立法或修法，以重刑收遏阻之效，例如：災害防救法中，納入禁止散播假新聞；再如動物傳染病防治條例中，禁止故意傳播有關動物傳染病之謠言或不實訊息等。

何謂不實資訊？中文裡常用「假新聞」或「不實訊息」，但在定義上並沒有明顯的區分。根據歐洲理事會對「資訊失序」2017 年的報告，提出以下三種不同的類型：



① 錯誤資訊 (mis-information)

指虛假資訊，雖與事實不符，但沒有惡意，也沒有造成傷害。



② 不實資訊 (dis-information)

指與事實不符，且有惡意，為了傷害特定對象或達到某種目的。



③ 惡意資訊 (mal-information)

符合事實但有惡意，通常為了攻擊特定對象，刻意大肆渲染，造成對方傷害。

為了檢視資訊失序，可以依資訊虛假的情況和造成傷害的程度來描述三種資訊之間的差異，但要辨識資訊的真假，是件不容易的事。現行「社會秩序維護法」第 63 條第 1 項第 5 款規定「散布謠言，足以影響公共之安寧者」，也會受罰，雖對假訊息有所規範，目前檢警單位對假訊息定義，主要包括惡意、虛假、具危害性，但仍過於抽象、欠明確。

由於查證不易，非營利組織《世界經濟論壇》表示，上網發表誤導性資訊幾乎不會遭到譴責，反而獲得更多關注，以致讓更多人上網發表誤導性資訊，此種趨勢令人憂心（自由時報，2019-12-6）。為了避免被不實資訊操弄，商業網路大戶如臺灣 Google，包括其所屬的 YouTube 所有平臺，在選舉活動期間，則暫停競選廣告。Facebook 則要求刊登社會議題、選舉或政府廣告，皆須揭露出席者資訊（蘋果日報，2019-11-14）。

新聞快報 // 錯誤資訊 – 「吃棉花減重」網路謠言事件

哇！原來棉花是植物纖維，當主食吃可以減肥…

最近好像變胖了。要開始減肥了！

吃了就不餓了！

一年之後

肚子好痛啊！以後不敢亂信謠言了！

不合常理的網路謠言，例如：有少女誤信網路謠言，以為吃棉花可以減肥，以致身體力行達一年，幾乎吃掉一條棉被，甚至圍巾的流蘇也吞掉，結果致病。

自由時報，2019-11-21

新聞
快報

不實資訊 – 「衛生紙之亂」事件

根據查證為不實謠言，
該超市恐觸法…哇！衛生紙要漲價了，
要再囤多一點！

18:32 衛生紙調漲 人民瘋搶缺貨



2018年2月23日，知名量販業者為了促銷自家衛生紙，促進衛生紙特賣銷售檔期的業績，以電子郵件與即時通訊軟體向媒體發送「衛生紙確定大漲30%，賣場業績急飆五倍」的訊息，指稱衛生紙近期將會大漲，且指稱「調漲時間點最快落在三月中旬，最慢四月前必漲」，並以廣泛性的指稱：「叫得出名字」的品牌衛生紙會調漲，讓收到訊息者自行推論認為大多數衛生紙品牌都會調漲。

一般無法判斷新聞真假的消費者，也為了購買衛生紙而疲於奔命。社會各界均為此假新聞付出重大的社會成本。因此，公平交易委員會在2018年3月21日以該公司「以誤導消費者之衛生紙漲價訊息，進行不實促銷，為足以影響交易秩序之欺罔行為，違反公平交易法第二十五條規定」，裁罰新臺幣三百五十萬元罰鍰。

中國時報，2018-12-27



生活案例 // 惡意資訊 – 「網路貼文抹黑」情境模擬事件

陳大華以投資股市及期貨致富，也頗有名氣。經常有團體邀他演講投資理財之道。他常說，賺錢不是罪過，但要取之有道。後來他參加民意代表選舉，有人（他的競爭對手）在網路上貼文，說他政商勾結，炒作股票，為了賺錢，無所不用其極，意圖使其不當選。經過查訪，陳大華投資致富，與人為善是事實，但其行為並無不法。他的競爭對手卻懷疑其政商勾結，非法炒作，以此在網路上散播，意圖造成對陳大華的傷害。



查證假新聞與假資訊的相關資源

行政院有即時新聞澄清專區 (<https://www.ey.gov.tw/Page/5519E969E8931E4E>)。

民間也有「真的假的」 (<https://cofacts.g0v.tw/>、Line ID : @ cofacts) 與「MyGoPen (麥擋騙)」 (<https://www.mygopen.com/>、Line ID : @ mygopen) 的闢謠網站，且可利用 Line 查證假資訊服務。

我們平時就要留意，遇到危言聳聽或不合常理的資訊，如能依照防範假新聞或不實消息的原則：**三不二要**（不輕信、不散播、不製造、要查證、要澄清，圖 5-3），或許可以減少一些干擾。

如前所述，現在人人可以發布訊息，擁有散布資訊的主動權，做為一個有**媒體素養 / 資訊素養**的數位公民，更應該注意資訊的散布，要力求正確，共同維護社會的安寧、國家的安全。



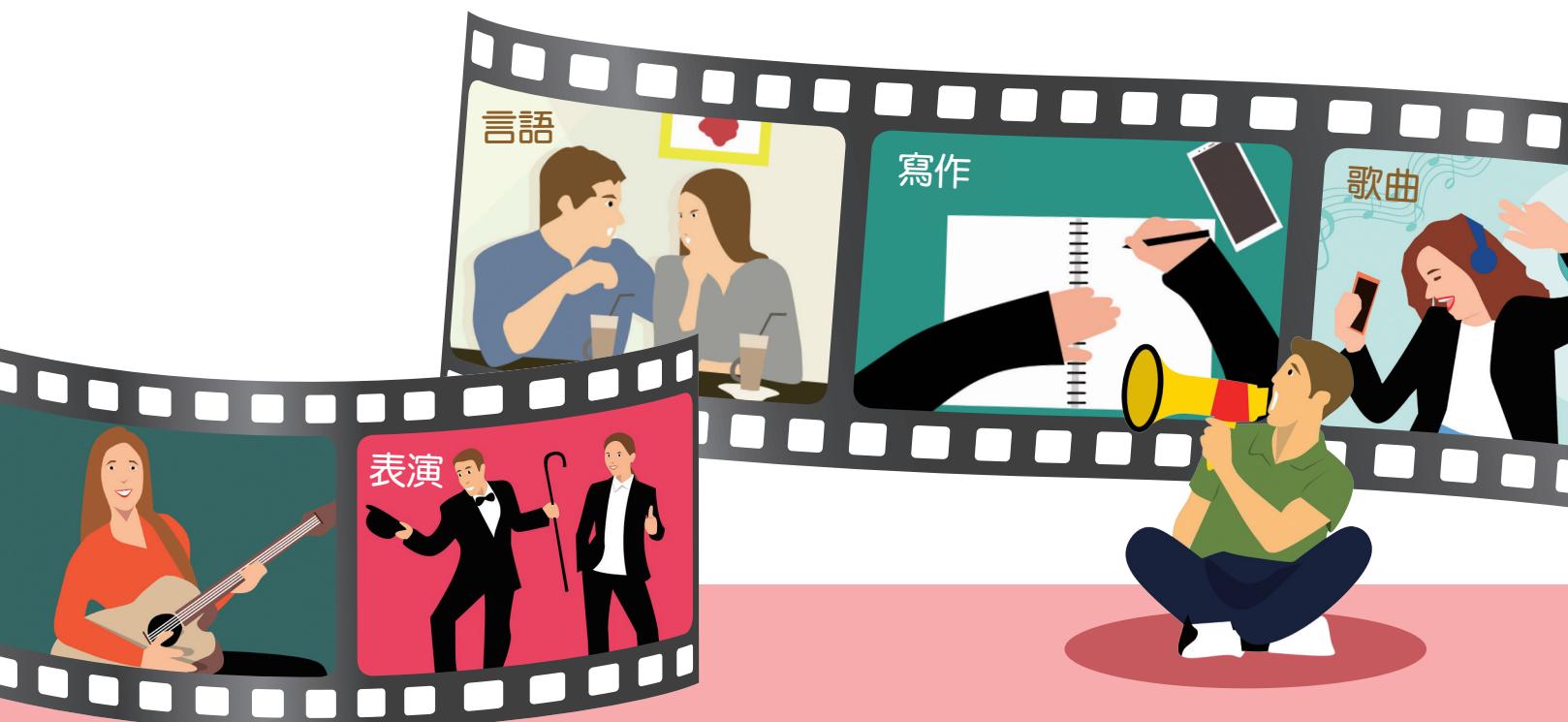
▲圖 5-3 防範不實資訊的三不二要原則示意圖。

5-3 言論自由

言論是指透過具有表達及溝通功能的方式，表現個人的意見或思想，藉以做為說服他人的行為。表達及溝通的方式不限於言語，其他如寫作、歌曲、表演等，也是表達思想的方式（圖 5-4）。言論自由則是指人民有表達意見或思想等的權利，也有收受資訊的權利，例如：獲取知識、近用資訊的權利等。

我國憲法賦予人民言論自由的權利，人民可以自由的發表言論以及聽取他人陳述意見。但為了避免濫用言論自由，憲法中也規範人民的言論自由必須在「防止妨礙他人自由、避免緊急危難、維持社會秩序及增進公共利益」等情況下，才受到保障，否則可以用法律來限制。

言論自由雖然是人民的基本權利，但為了顧及他人名譽、隱私及公共利益的保護，法律可以對言論自由做合理的限制。我國政府在戒嚴時期規範言論自由的方式，主要採取「事前限制」；目前我國與其他民主國家大都使用「事後懲罰」，也就是指對人民發表的言論，事前不予干涉，但事後若是發現其言論觸犯法律，或是損害他人利益時，行為人要負起相關法律責任。



▲圖 5-4 表達與溝通方式包含言語、寫作、歌曲和表演。

現實社會中法律對言論自由的保障和規範，同樣適用於網路社會。在網路世界裡，任何人都可以上網，同樣也是達到一種法律上所稱的「公然」狀態。網際網路是一個開放環境，符合公然的要件，因此，若有人在網路上發表言論，致使他人的名譽受損，即是犯了公然侮辱罪。同樣的，意圖散布、指責或傳送足以毀損他人名譽之事者，則犯了誹謗罪。網路社會是一個比現實社會更為公開的環境，網路使用者不要把言論自由無限上綱，以免觸犯法律，不可不慎！（圖 5-5）



小知識

公然

只要是行為人在表達言論的當時，為「不特定人或多數人得以共見共聞之狀況」。多數人包括特定之多數人在內；只有少數的特定人在場，就不能算是公然。例如：夫妻關起門來互相罵對方，如果只有少數的特定人在場，就不算是公然。如果場地是在人來人往的公共場所，雙方口無遮攔地侮辱他方的惡言惡語，都是不特定的多數人所共見共聞，就構成公然侮辱罪。

公然侮辱罪 vs. 誹謗罪

- 公然侮辱罪：指犯罪行為人在不特定人或多數人得以共見共聞的公開狀態，侮辱、謾罵特定人或可得推知之人，使他人名譽受損。名譽受損之人不僅可以請求民事損害賠償，行為人更可能觸犯刑事責任。
- 誹謗罪：指犯罪行為人意圖以言詞散布於眾或散布文字、圖畫，而指摘或傳述足以毀損他人名譽的具體事實，而且指摘或傳述不以公開狀態為必要，即使是私下傳述也在誹謗罪處罰之列。

▼圖 5-5 不觸犯法律的言論，才享有言論自由。



5-4 網路霸凌

霸凌主要是強勢者長期且重複的透過肢體、言語以及其他方式，對弱勢者進行欺負或騷擾的行為，在各種團體都可能發生；如果發生在校園就稱為校園霸凌（school bullying）。

教育部為維護校園安寧，在修正公布的教育基本法第八條，規定校園霸凌防制準則，且定義校園霸凌是指相同或不同學校的學生與學生間，在校園內、外所發生之個人或集體持續以言語、文字、圖畫、符號、肢體動作或其他方式，直接或間接對他人進行貶抑、排擠、欺負、騷擾或戲弄等行為，使他人處於具有敵意或不友善的校園學習環境，或難以抗拒，產生精神上、生理上或財產上的損害，或影響正常學習活動的進行（圖 5-6）。

小知識

教育部防制校園霸凌專區
提供校園霸凌投訴專線與相關資源等。
網址：<https://csrc.edu.tw/bully/>







何謂校園霸凌？

行為：個人或集體持續以言語、文字、圖畫、符號、肢體動作或其他方式，直接或間接對他人為脅迫、指摘、欺負、騷擾或戲弄等行為。

結果：使他人處於具有敵意或不友善的校園學習環境或使他人難以抗拒、產生精神上、生理上或財產上的損害或影響他人正常學習活動之進行。

霸凌類型可分為：肢體霸凌、言語霸凌、網絡霸凌、情緒霸凌、反覆霸凌。

貼文內容	可能觸犯法律
發表或散播批評、謠諑、不實的言論	刑法謠言罪 民法侵權行為
發送警告、恐嚇的言論	刑法恐嚇危言安全罪
上傳或散播不雅、破壞他人名譽的圖片／影片	兒童及少年福利與權益保障法 刑法謠言罪或妨害名譽罪等
將淫穢行為錄影、上傳網路供人觀看	刑法瀕露罪

▶ 圖 5-6 教育部防制校園霸凌相關海報。

（資料來源：教育部防制校園霸凌專區網站）

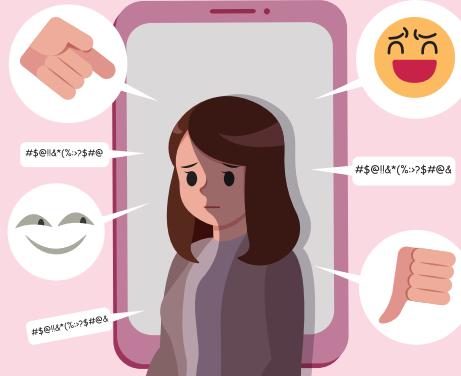
在網路盛行的時代，實體校園環境除了透過肢體傷害的霸凌方式外，其他霸凌方式幾乎都可以在網路上進行，因此，以下霸凌行為的描述主要以網路霸凌為主。

5-4-1 常見的網路霸凌行為

1 文字嘲弄

通常用有歧視、嘲笑、甚至惡毒的文字，透過網路去批評或騷擾，造成受凌者極大的精神壓力（圖 5-7）。

► 圖 5-7 文字嘲弄示意圖。



2 圖像騷擾

在網路上，公開散布受凌者私密照，或經過變造的不實剪接照片及不雅的圖片等。圖像的騷擾也經常加上嘲弄的文字，讓受凌者的名聲及生活受到汙衊與干擾（圖 5-8）。

► 圖 5-8 圖像騷擾示意圖。



3 訊息恐嚇

以直接傳送電子郵件或手機簡訊等方式加以恐嚇，使受凌者防不勝防，疲於應付，造成嚴重精神傷害（圖 5-9）。

► 圖 5-9 訊息恐嚇示意圖。



4 社交孤立

霸凌者透過電子郵件、手機簡訊或社群網站等，大量發送不實且負面的訊息，使受凌者人際關係惡化、受到誤解等精神傷害，讓受凌者社交關係受到同儕排斥，造成孤立（圖 5-10）。

▶ 圖 5-10 社交孤立示意圖。



以上這些霸凌行為，對於正處於身心發展階段的青少年會有極大的傷害。遭受霸凌的青少年有可能對人際關係產生不滿、自我排斥、甚至出現身體及精神上的健康警訊。我國各級學校都頗重視這個問題，提倡友善校園，明令禁止霸凌，以維護校園安寧（圖 5-11）。



▶ 圖 5-11 友善校園禁止網路霸凌行為。



5-4-2 如何面對網路霸凌

校園中會有潛藏的霸凌者，人人都可能是受凌的對象，因此，如何應對網路霸凌，需有正確且完整的觀念。

- 1** 受凌者要勇於求助，不要沉默的以為自己就可以處理好。
- 2** 若收到類似霸凌的訊息，不管是在課堂中，或是來自網路，都應該勇於告訴師長及父母。
- 3** 若是不斷收到同學的電子郵件或手機簡訊的恐嚇威脅，立即封鎖其電子郵件或手機號碼。
- 4** 若是不明人士任意張貼不實言論在公開的網路空間，造成旁人誤解，應立即告知師長及父母。
- 5** 留下畫面截圖以保留證據。
- 6** 向學校投訴信箱、縣市反霸凌投訴專線投訴。

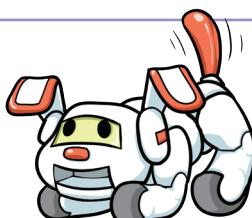


霸凌事件存在於校園內外，若不及時遏止，對受凌者、旁觀者，甚至霸凌者的身心發展影響鉅大。



教育部防制校園霸凌專線

遇到霸凌事件，可撥打 24 小時免付費專線進行投訴。
電話：0800-200-885（耳鈴鈴、幫幫我）





5-4-3 網路霸凌的法律問題

網路屬於公開的空間，雖然文字嘲弄或是訊息恐嚇的網路霸凌不一定會構成違法，但是對霸凌者而言，若不加以防範、矯治其行為與態度，最後極有可能惡化成觸法行為。

常見的網路霸凌行為與法律責任：

行為	法律責任
發表或散播批評、誹謗、不實的言論（包含舉辦或參與惡意票選活動）	刑法公然侮辱罪、誹謗罪、民法侵權行為
在公開的網路空間發表警告、恐嚇的言論	刑法恐嚇危害安全罪
上傳或散播不雅、破壞他人名譽的照片／影片（包含移花接木的不實照片）	刑法公然侮辱罪、誹謗罪、妨害風化罪、妨害秘密罪、散布或販賣猥亵物品及製造持有罪
上傳攻擊行為的影片	刑法傷害罪
公布他人個人資料（包含人肉搜索後公開他人個人資料）	個人資料保護法、刑法妨害秘密罪
盜用他人帳號（以便冒名進行以上行為）	刑法無故入侵電腦罪

(資料來源：[教育部全民資安素養網站 https://isafe.moe.edu.tw/article/2015](https://isafe.moe.edu.tw/article/2015))



小知識

網路霸凌的法律諮詢管道

若有相關法律疑問，也可以尋找財團法人法律扶助基金會洽詢，並提供電話法律諮詢、現場法律諮詢與視訊法律諮詢。

諮詢方式說明網址：<https://www.laf.org.tw/index.php?action=apply&cid=183>

教育部防制校園霸凌專區

列出校園霸凌防制準則修正條文。

網址：<https://bully.moe.edu.tw/details/141>

5-5 網路成癮

由於網路資訊發達，根據調查，臺灣民眾上網率已達百分之八十以上，其中 18 至 30 歲的民眾上網率已達到百分之百。此資料顯示，網路儼然已成為我們日常生活的一部分。對於使用網路的時間多久才算合理，並無一定標準，也因年齡層或職業類別而有差異，但應以不影響身心發展及健康為原則；過度或不當使用，對生理及心理都會產生負面的影響。

在資訊時代，網路資源相當豐富，如果很少使用或不使用資訊科技產品，可能因此喪失很多使用網路資源的機會；但若過度使用，恐怕會沉迷其中（圖 5-12）。心理學家或精神科醫師指出，平均每週上網超過 40 小時者，會有較高網路沉迷的風險。若有沉迷傾向，對身心恐造成負面影響，例如：睡眠不足、精神不振、注意力不專注、容易分心等，這些症狀都會影響到心理的健康，也會對課業產生不利的影響。



▶ 圖 5-12 過度使用網路，容易造成網路成癮。

小知識

憂鬱與網路使用現況調查

2019 年，董氏基金會以大臺北地區國、高中職生為對象進行調查，結果發現：36.5% 受訪者假日上網總時數超過 6 小時，平日也有 18.3%。此外，有 21.1% 受訪者假日每次上網平均持續超過 4 小時，42.7% 平均每隔 30 分鐘就想上網。此調查也發現，上網時數愈久，憂鬱程度也愈高。

網路使用評量表

為了面對網路沉迷的問題，可至臺北市政府衛生局社區心理衛生中心網頁（<https://mental-health.gov.taipei/>）的心理評量專區自行填寫「網路使用評量表」，依此量表加總指數，將網路沉迷分為三等級，如下所示：

- (1) 正常：上網行為正常。(2) 預警：過度使用網路。(3) 危險：病態使用習慣。



重點回顧

在各行業資訊化後，資訊科技也逐漸取代了傳統媒體的功能。以言論為表達媒介的人類社會，也衍生很多社會議題。本章主要討論三種常見的議題，以及網路成癮。

防範資訊失序

現在資訊科技發達，釋放資訊容易，來源也多，真假辨識不易，查證也難，以致假資訊扭曲了民眾知的權利。個人應該要有防範的意識，也就是三不二要原則：不輕信、不散播、不製造、要查證、要澄清。

避免言論自由濫用

網路環境被認為是公開的場合，合乎法律上公然的條件。當有人在網路散布侮辱他人的言論，使他人的名譽受損，恐就觸犯公然侮辱罪。同樣的，指責或傳送足以毀損他人名譽之言論，且散布於網路，則恐犯了誹謗罪，言論自由不能無限上綱！

防阻網路霸凌

因網路的方便，霸凌更容易發生。霸凌者常有反社會人格，其心態是蓄意的、敵意的、惡意的，其行為是重複的、持續的對受凌者攻擊，造成受凌人心靈受創，也會造成課業成就低落、人際孤立，甚至有可能迫使受凌人產生報復，轉而霸凌他人。當遭受霸凌時，要勇敢說出來，並向師長及家長報告，或向學校投訴，防阻霸凌者得寸進尺。

避免網路成癮

對於使用網路的時間多久才算合理，並無一定標準，應以不影響身心發展及健康為原則。平均每週上網超過 40 小時者，就有網路沉迷的風險。



第 6 章

基本演算法的介紹

何謂演算法？簡單的說，演算法就是一種解決問題的方法。在生活周遭的很多問題，隨時隨地可以用電腦程式語言實作演算法，幫我們解決問題。

七年級時，已學過基礎程式設計，也大致了解演算法的基本概念。加上八年級學過的進階程式設計，我們可以更進一步來學習，電腦如何透過程式語言來完成常見的排序與搜尋問題。

本章首先介紹演算法的基本原理及表示方式，接著用實例推導排序與搜尋問題，爾後再進入Scratch中來實作演算法，讓同學了解電腦解決問題的方式。希望這些方法能讓同學在日後遇到資料的排序或搜尋時，都可以運用程式設計實作演算法來解決問題。

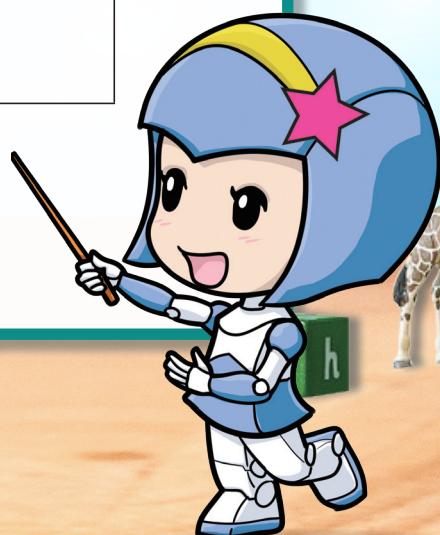
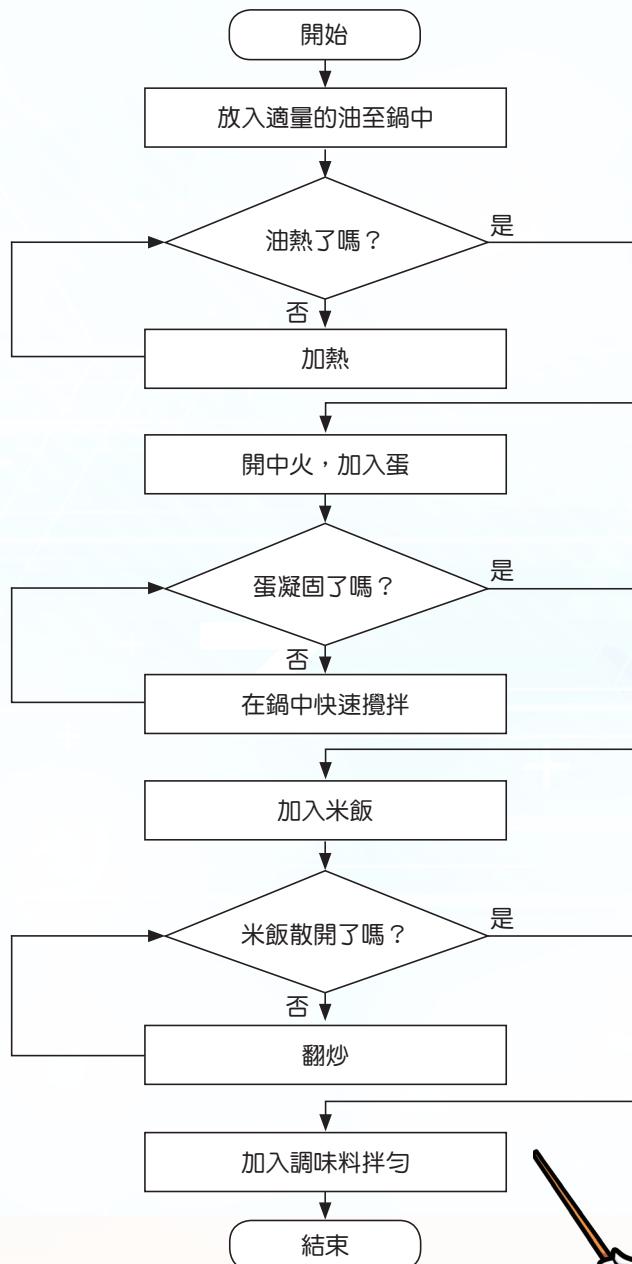
哇，演算法的功用這麼厲害，就讓我們趕快來認識它吧！

6-1 演算法概念與原理

6-2 排序的原理與範例

6-3 搜尋的原理與範例





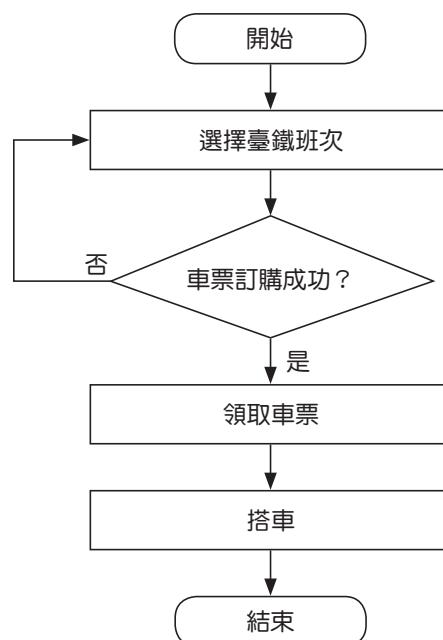
6-1 演算法概念與原理

什麼是演算法（algorithm）？簡單的說就是解決問題的方法。在日常生活中，演算法時時刻刻在幫我們解決問題。例如：在七年級第二章曾提到蛋炒飯食譜，即可視為是一種演算法，幫我們把製作蛋炒飯這個問題抽象化為可操作的步驟。在資訊科技領域中，演算法是一個可以交由電腦進行計算的具體步驟，它是一組有限運算規則的集合，包含問題精確的輸入、處理、輸出等。

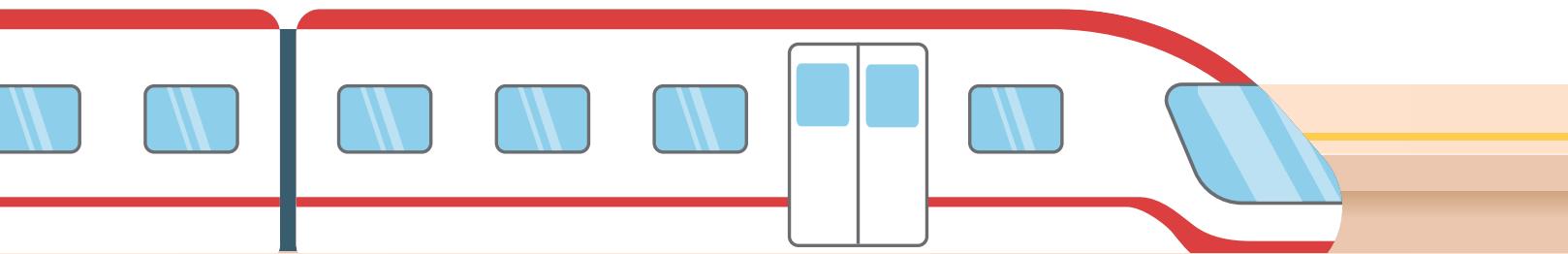
演算法可以利用文字敘述（圖 6-1）、流程圖（圖 6-2）或其他方式表示。像是日常生活中的食譜即是一種用文字敘述表示的演算法。



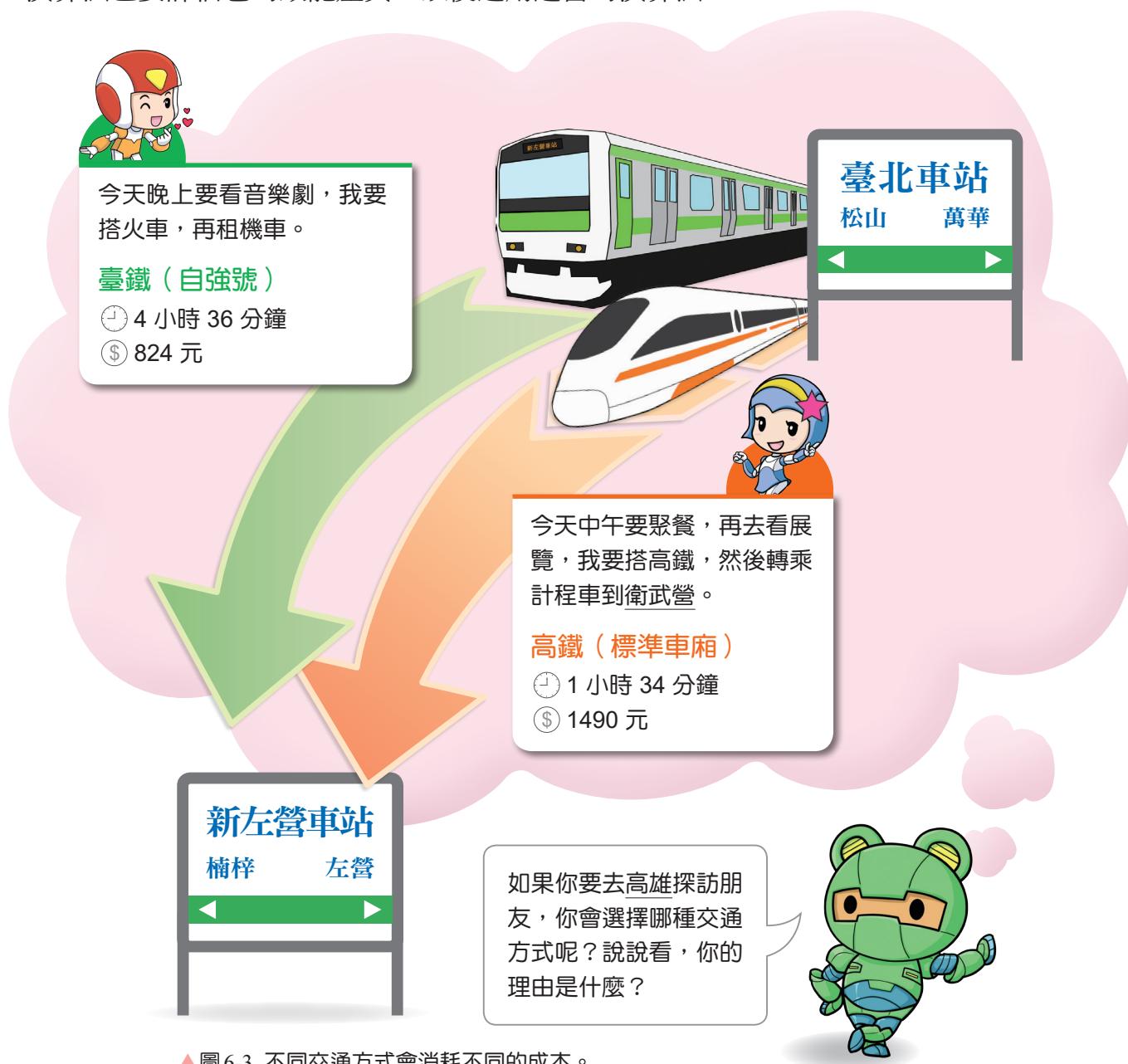
▲圖 6-1 使用文字敘述表示的演算法。



▲圖 6-2 使用流程圖表示的演算法。



在日常生活中，為了到達某個目的地，例如：要到達高雄衛武營國家藝術文化中心，從臺北車站出發到達新左營車站，阿顯要看晚上時段的音樂劇，所以選擇搭乘臺鐵，再轉乘接駁車或租機車，以節省車資；星兒中午要先與朋友聚餐，再到附近看展覽，最後到衛武營觀賞音樂劇，可以選擇搭乘高鐵，再轉乘計程車，以節省時間。不同方式會付出不同的成本（如時間、金錢），進而影響效能，都是必須考量的因素（圖6-3）。在資訊科技領域中，不同的演算法都必須能夠精確的解決問題外，執行不同的演算法也要評估它的效能差異，以便選用適當的演算法。

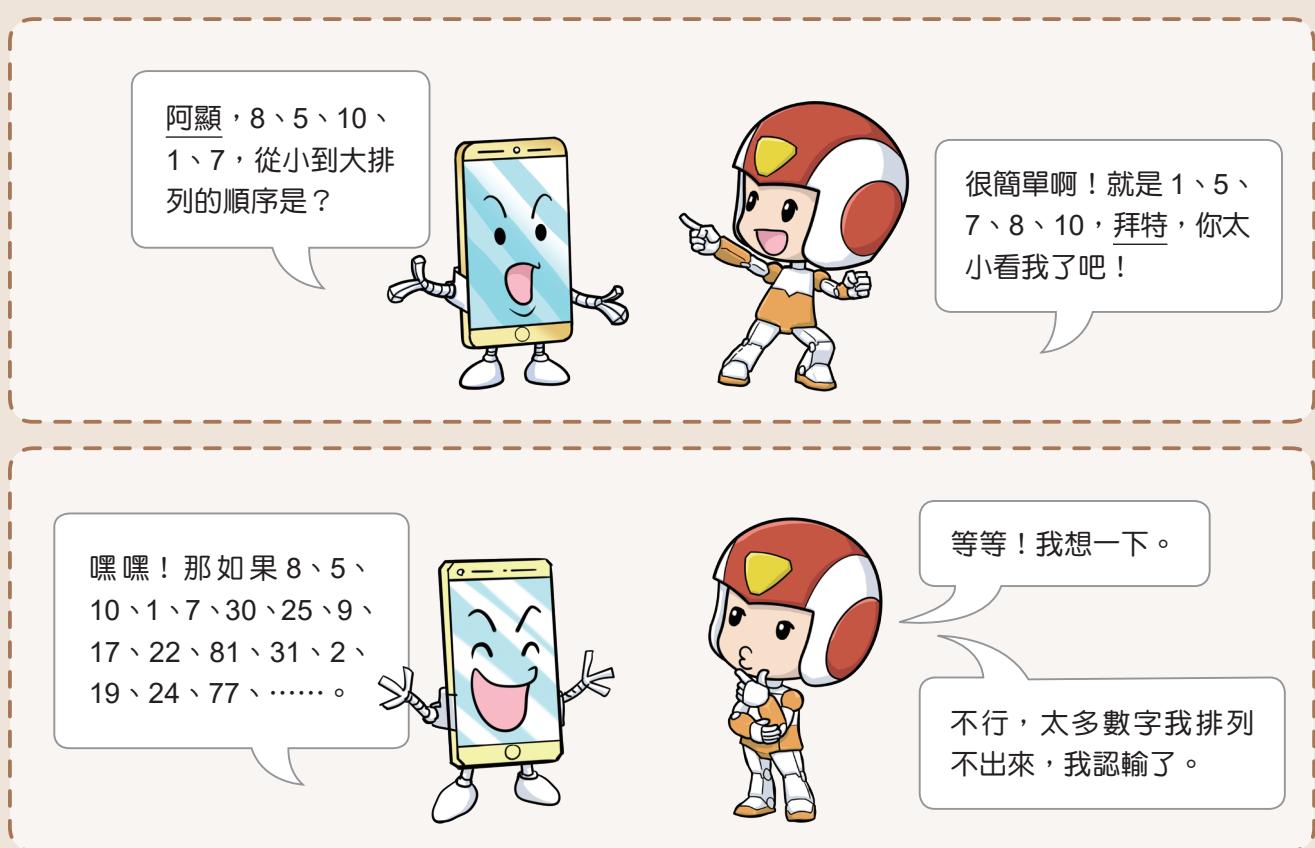


▲圖6-3 不同交通方式會消耗不同的成本。

6-2 排序的原理與範例

將資料依順序排列，是我們經常需要處理的問題，例如：日常生活中，在班級中依照同學的身高安排前後的座位，或是計算各式各樣的排行榜等。我們在七年級第三章資料處理單元，已經學過使用試算表工具來將銷售統計進行排序，然而，試算表內部是如何處理排序這個問題呢？本單元將介紹選擇排序法與插入排序法這兩種不同的演算法，為你揭開排序問題的神秘面紗。

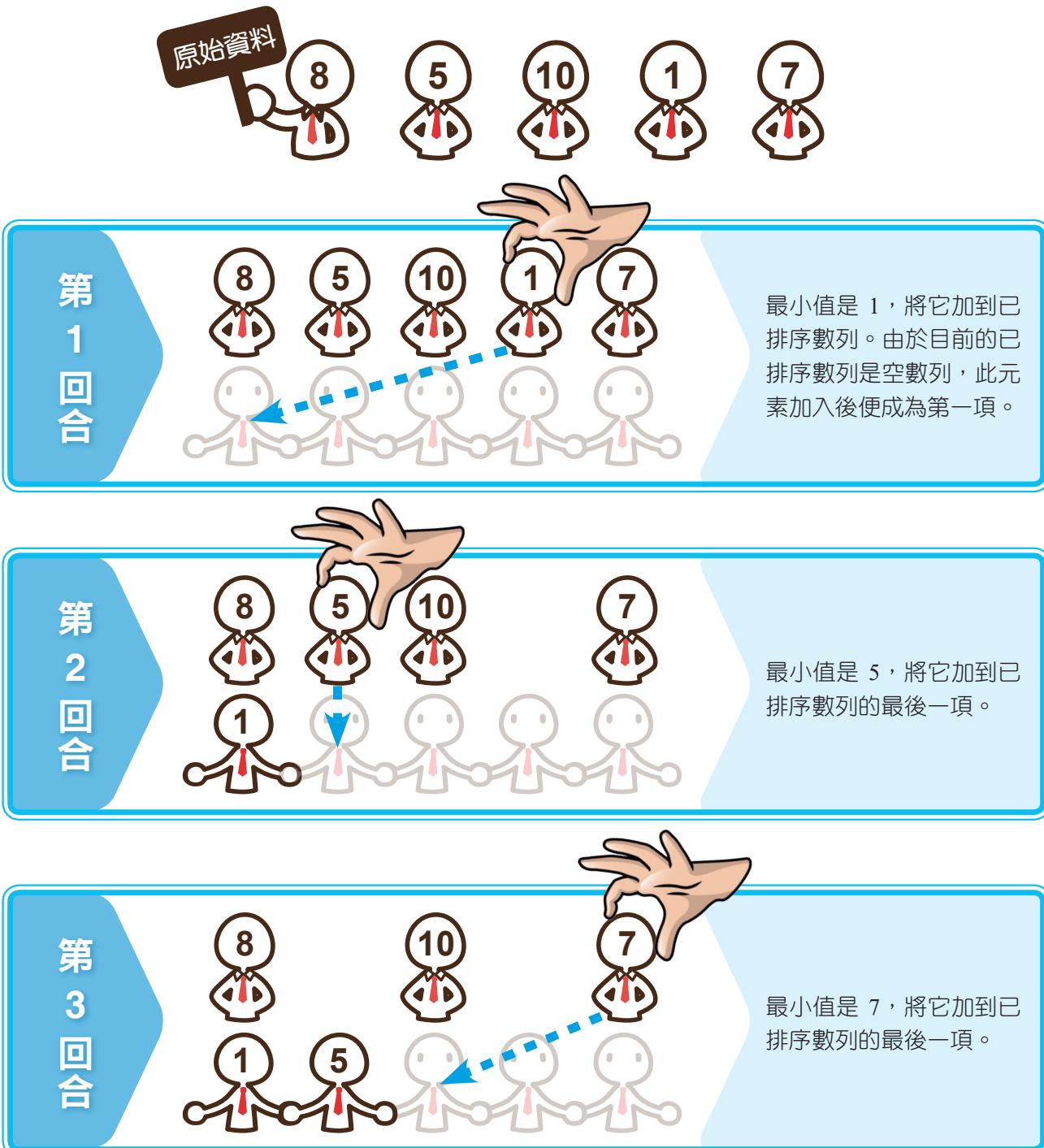
首先，我們將排序問題簡化成數字的排序。假設原始資料有：8、5、10、1、7，共 5 個隨機數字，經從小到大排序後資料變成：1、5、7、8、10。或許資料只有 5 個數字，所以你能夠很容易的就把這 5 個數字從小到大排列，但是如果資料量暴增到 100 筆，你就得要從解決問題的過程中找出規則（圖 6-4），將其發展成一套演算法，而這套演算法適用於各種大小的資料量。



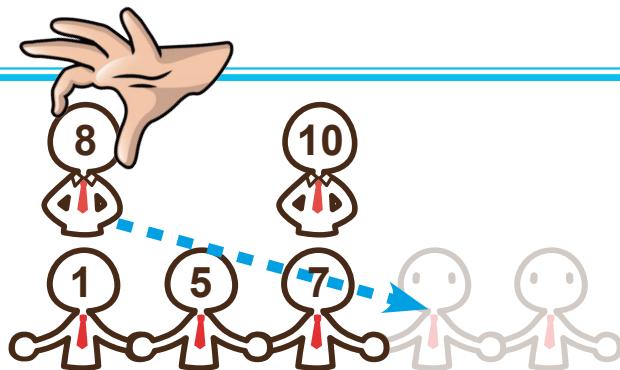
▲圖 6-4 排序不同的資料量。

6-2-1 選擇排序法

選擇排序法（selection sort algorithm）的概念是反覆從未排序的原始資料中取出最小的元素，加到已排序數列的最後一項，待所有原始資料中的元素都取出後，已排序的資料就是我們要的結果，舉例如下：

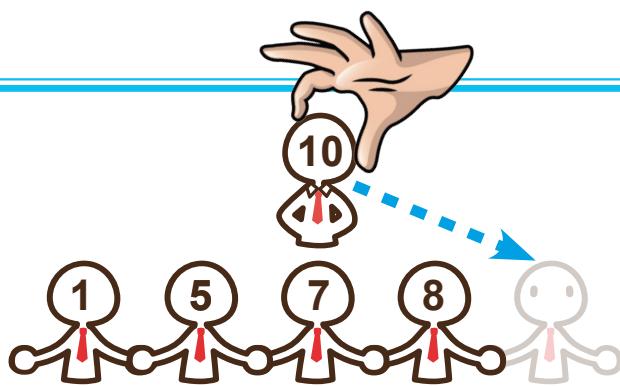


第 4 回合



最小值是 8，將它加到已排序數列的最後一項。

第 5 回合



最小值是 10，將它加到已排序數列的最後一項。

已排序
資料



我們可以從以上的流程，歸納出實作的步驟：

1. 先從未排序的原始資料中找到第一個最小的元素，將它加到已排序數列的第一項。
2. 接著從未排序的原始資料中找到最小的元素。
3. 將此元素加到已排序數列的最後一項。
4. 重複第 2、3 點的步驟，直到原始資料全部處理完成。





範 例一選擇排序法

點擊綠旗後，小貓執行下列動作：

1. 先從數字 1 ~ 10 中隨機取出 5 個數字，並將這 5 個數字放入一個原始資料裡。
2. 從原始資料中逐次選出一個最小值，並說出所選取的數字。
3. 將此數字放入一個已排序資料裡。
4. 最後完成大小順序的排列後，小貓會說出這 5 個數字的大小順序。

請執行《選擇排序法》的程式，想一想這個範例的程式是如何運作？

1

原始資料	已排序資料
1 8	(empty)
2 5	
3 10	
4 1	
5 7	
+ 長度 5 =	+ 長度 0 =

2

原始資料	已排序資料
1 8	(empty)
2 5	
3 10	
4 1	
5 7	
+ 長度 5 =	+ 長度 0 =

目前從原始資料中找到的最
小值是1

範例執行後，原始資料呈現隨機添加的 5 個數字。

點選小貓後，第 1 回合取出原始資料的最小值後，並說出：「目前從原始資料中找到的最小值是…」，然後放入已排序資料裡。

3

原始資料	已排序資料
1 8	1 1
2 10	2 5
3 7	
+ 長度 3 =	+ 長度 2 =

目前從原始資料中找到的最
小值是7

4

原始資料	已排序資料
(empty)	1 1
	2 5
	3 7
	4 8
	5 10
+ 長度 0 =	+ 長度 5 =

這5個數字由小排到大的順
序是1 5 7 8 10

接著第 2~5 回合都取出原始資料的最小值，並說出：「目前從原始資料中找到的最小值是…」，陸續放入已排序資料裡。

最後全部執行完畢後，小貓說出：「這 5 個數字由小排到大的順序是…」。



問題分析

我們可以將這個程式範例拆解幾個部分如下：

- ① 如何從數字 1 ~ 10 中隨機取出 5 個數字，並將這 5 個數字放入未排序數列裡？
 1. 如何表示未排序數列？
 2. 如何從數字 1~10 中隨機取出 5 個數字，並放入未排序數列裡？
- ② 如何從未排序數列中找到最小的數字？
 1. 如何找出最小值？
- ③ 如何將此數字加到已排序數列的最後一項？
 1. 如何表示已排序數列？
 2. 此數字指的是什麼？
 3. 如何將原始資料中的最小值添加到已排序數列的最後一項？
- ④ 重複執行上述的動作，直到未排序數列全部處理完成。

解題步驟

問題
拆解

如何從數字 1 ~ 10 中隨機取出 5 個數字，並將這 5 個數字放入未排序數列裡？

① 想一想，這句話包含了什麼重要的訊息？

1. 未排序數列：



在 Scratch 中如何記錄數列呢？



使用 **清單**，我們將這個清單命名為原始資料。

2. 隨機取出：



如何從數字 1~10 中隨機取出 5 個數字，並將這 5 個數字放入未排序數列裡？



使用 **迴圈**，從數字 1~10 中隨機取出 5 個數字，並添加到原始資料清單中。



如果隨機取出的數字重複時，該怎麼辦？



1. 此處隨機選取數字時，有可能會出現重複的數字，但不影響結果。
2. 如果要設定隨機選取的數字不能重複的話，此時程式會變得比較複雜。
3. 我們先從簡單的（假設數字不重複）進行介紹，如果同學有興趣想了解，也可以動手試試看。

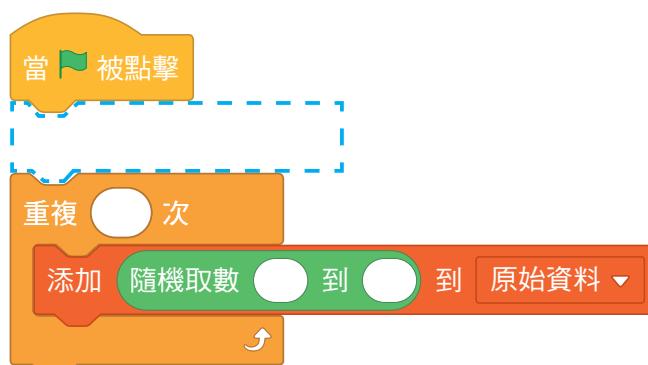
步驟 1 設定清單。

- ① 新增清單，命名為原始資料。



步驟 2 請同學想想看，並回答下面的問題及完成右方的積木。

1. 虛線框的積木是什麼？
2. 迴圈要執行幾次？



問題
拆解

如何從未排序數列中找到最小的數字？

(2)

想一想，這句話包含了什麼重要的訊息？

找到最小的數字：



你知道在 Scratch 中，怎麼找出數列中的最小值？



1. 需要取一個數字與最小值進行比較，所以先設定**資料位置**與**最小值位置**兩個變數。
2. 先假設第一個數字是最小值，所以設定**資料位置**、**最小值位置**的變數皆為 1。
3. 接著取第二個數字，所以**資料位置**設為 2，再跟**最小值**進行比較：
 - (1) 如果第一個數字（**最小值**）比較小，就繼續再跟下一個數字比較。
 - (2) 如果第二個數字比較小，就將第二個數字設為**最小值**，再跟下一個數字比較。
4. 直到與所有的數字都完成比較。
5. 因為進行找出最小值的流程一直重複，所以可以將找出最小值的方法設為副程式。

步驟
3

設定函式積木與變數。

(2)

新增**函式積木**，命名為**找出最小值位置**。



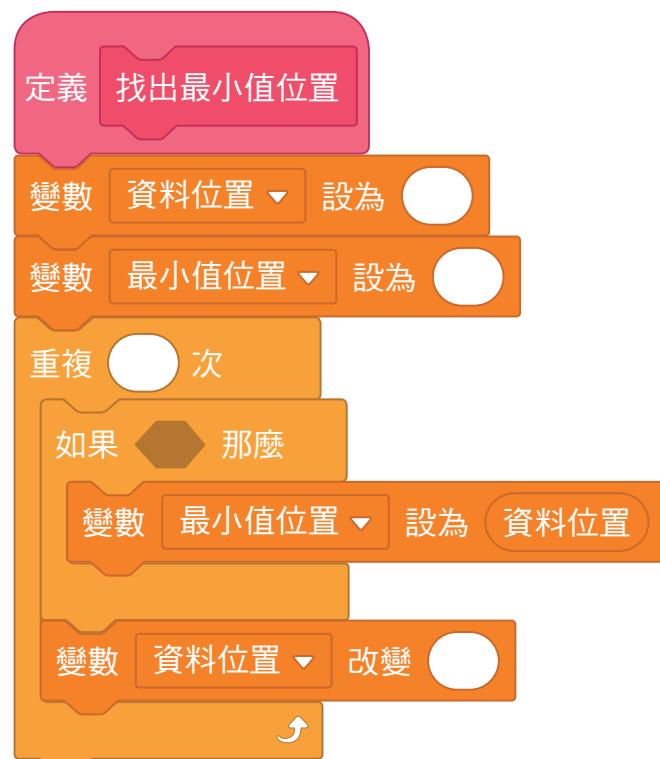
- ③ 新增變數，分別命名為最小值位置與資料位置。



步驟
4

請同學想想看，並回答下面的問題及完成右方的積木。

1. 為什麼要設定找出最小值位置的函式積木？
2. 資料位置與最小值位置的變數一開始要設為多少？
3. 迴圈要執行幾次？
4. 判斷的條件要怎麼設定？
5. 資料位置的變數要改變多少？



問題
拆解

如何將此數字加到已排序數列的最後一項？

(3)

想一想，這句話包含了什麼重要的訊息？

1. 已排序數列：



在 Scratch 中如何記錄數列呢？

使用 **清單**，我們將這個清單命名為**已排序資料**。

2. 此數字：



此數字指的是什麼呢？

就是上一個步驟找出來**原始資料清單**中的最小值。

3. 加到已排序數列的最後一項：

如何將**未排序數列**中的最小值添加到**已排序數列**的最後一項？

1. 把每一次從**原始資料清單**中找到的最小值，添加到**已排序資料清單**中。
2. 因為最小值已加到**已排序資料清單**中，所以把剛剛從**原始資料清單**中找到的最小值刪除。
3. 第二次開始找到的最小值會比上一次找到的大，所以每次添加到**已排序資料清單**時，要添加到最後一項。

步驟 5 設定清單。

- ④ 新增清單，命名為已排序資料。



步驟 6 請同學想想看，並回答下面的問題及完成右方的積木。

1. 虛線框的積木是什麼？
2. 要將原始資料清單的第幾項添加到已排序資料清單？
3. 要刪除原始資料清單的第幾項？



問題拆解

重複執行上述的動作，直到未排序數列全部處理完成。

- ④ 想一想，這句話包含了什麼重要的訊息？

1. 重複執行上述的動作：



如何重複執行上述的動作？

我們已經將找出最小值位置包裝成一個副程式，只要呼叫它就可以完成找到原始資料清單中的最小值位置，再把它與步驟 6 的程式拼接起來。



2. 直到未排序數列全部處理完成：



直到未排序數列全部處理完成的意思是什麼？

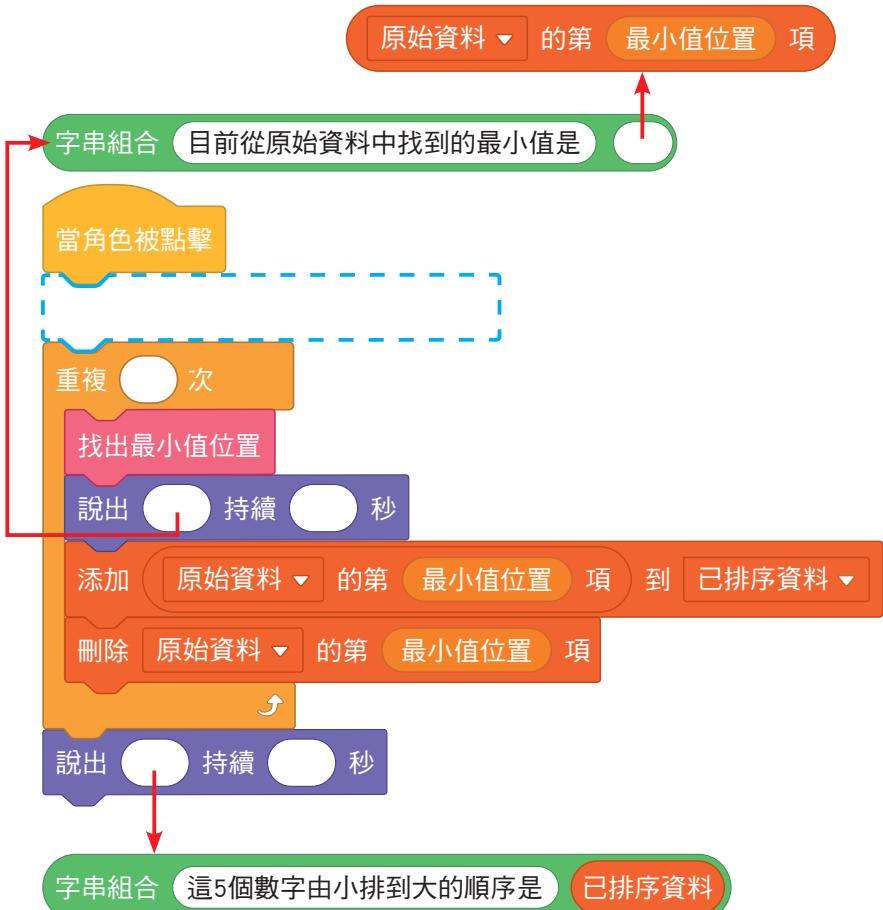


使用迴圈，將原始資料清單中的每個數字都執行一次。

步驟
7

請同學想想看，並回答下面的問題及完成右方的積木。

1. 是否可以把 **當角色被點擊** 換成 **當  被點擊** ？
2. 說說看，虛線的積木是什麼？
3. 復圈要執行幾次？
4. 當何時候要執行找出最小值位置的副程式？
5. 為什麼要讓小貓一直說出：「目前從原始資料中找到的最小值是…」？
6. 最後如何讓小貓說出執行結果：「這 5 個數字由小排到大的順序是…」？

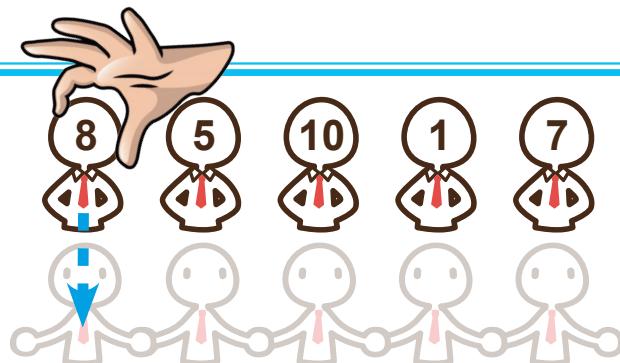


6-2-2 插入排序法

插入排序法（insertion sort algorithm）的概念是逐一從未排序的原始資料中取出元素，再從已排序數列由前往後找到適當的位置插入，如果遇到大於自己的元素就插入此元素之前；否則插入在已排序數列的最後一項，舉例如下：

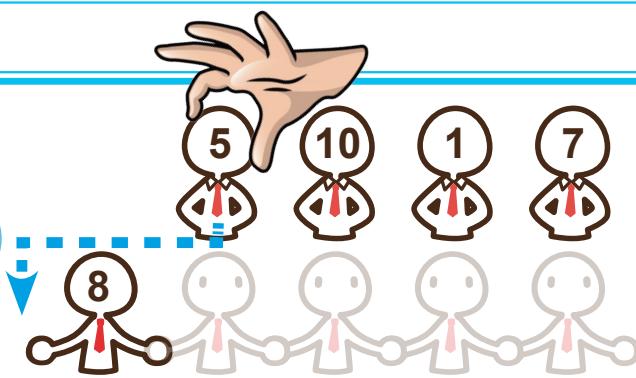


第1回合



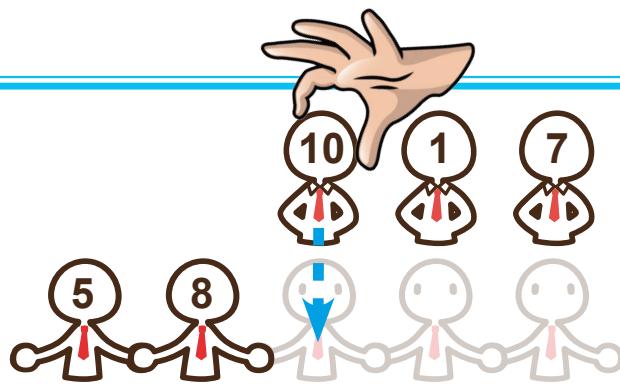
取出第 1 個元素，
將它加到已排序數列。
由於目前的
已排序數列是空數列，
此元素加入後
便成為第一項。

第2回合



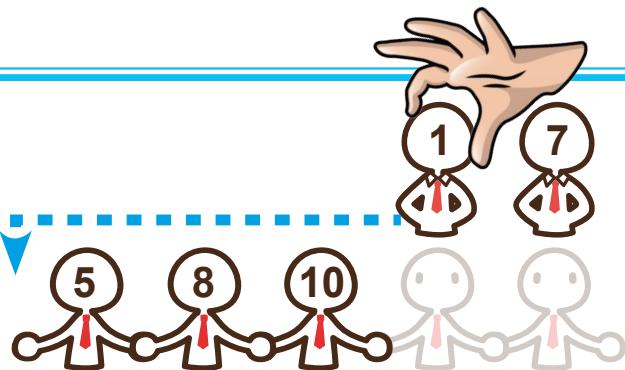
取出第 2 個元素，按
照由小至大的順序，
將它插入已排序數列
適當的位置中。

第3回合



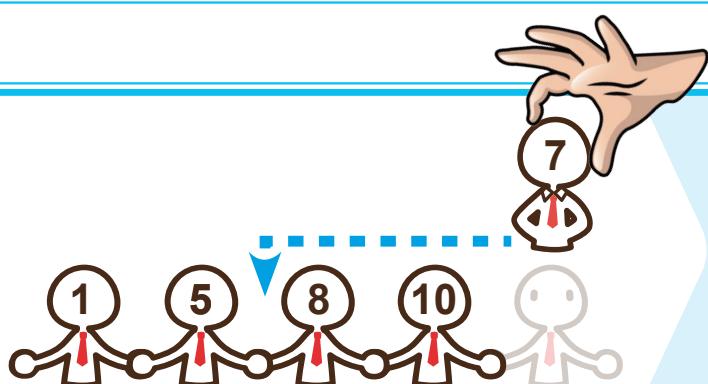
取出第 3 個元素，按
照由小至大的順序，
將它插入已排序數列
適當的位置中。

第4回合



取出第 4 個元素，按照由小至大的順序，將它插入已排序數列適當的位置中。

第5回合



取出第 5 個元素，按照由小至大的順序，將它插入已排序數列適當的位置中。

已排序
資料



我們可以從以上的流程，歸納出實作的步驟：

1. 先從未排序的原始資料中，取出第 1 個元素加到已排序數列中的第一項。
2. 接著從未排序的原始資料中逐一取出元素。
3. 由前往後和已排序數列元素比較，遇到大於自己的元素就插入此元素之前；否則插入在已排序數列的最後一項。
4. 重複第 2、3 點的步驟，直到原始資料全部處理完成。





範 例—插入排序法

點擊綠旗後，小貓執行下列動作：

1. 先從數字 1 ~ 10 中隨機取出 5 個數字，並將這 5 個數字放入一個原始資料裡。
2. 從原始資料中逐次選出第一個數字，並說出所選取的數字。
3. 將取出的數字放入一個已排序資料裡。
4. 最後完成大小順序的排列後，小貓會說出這 5 個數字的大小順序。

請執行《插入排序法》的程式，想一想這個範例的程式是如何運作？

1

原始資料	已排序資料 (empty)
1 8	
2 5	
3 10	
4 1	
5 7	
+ 長度 5 =	+ 長度 0 =

範例執行後，原始資料呈現隨機添加的 5 個數字。

2

原始資料	已排序資料 (empty)
1 8	
2 5	
3 10	
4 1	
5 7	
+ 長度 5 =	+ 長度 0 =

目前從原始資料取出的數字是 8

點選小貓後，第 1 回合挑選原始資料的第一個數字，並說出：「目前從原始資料取出的數字是…」，然後放入已排序資料裡。

3

原始資料	已排序資料
1 10	1 5
2 1	2 8
3 7	
+ 長度 3 =	+ 長度 2 =

目前從原始資料取出的數字是 10

接著第 2~5 回合都從原始資料取出第一個數字，並說出：「目前從原始資料取出的數字是…」，依照由小到大陸續放入已排序資料裡。

4

原始資料 (empty)	已排序資料
	1 1
	2 5
	3 7
	4 8
	5 10
+ 長度 0 =	+ 長度 5 =

這 5 個數字由小排到大的順序是 1 5 7 8 10

最後全部執行完畢後，小貓說出：「這 5 個數字由小排到大的順序是…」。



問題分析

我們可以將這個程式範例拆解幾個部分如下：

① 如何從數字 1 ~ 10 中隨機取出 5 個數字，並將這 5 個數字放入未排序數列裡？

1. 如何表示未排序數列？

2. 如何從數字 1~10 中隨機取出 5 個數字，並放入未排序數列裡？

② 如何從未排序數列中逐一取出數字？

1. 如何確保不會從未排序數列中重複取出已經排序過的元素？

③ 如何由前往後和已排序數列的數字進行比較？遇到大於自己的數字就插入此數字之前；若都沒有則插入在後面。

1. 如何表示已排序數列？

2. 如何由前往後進行比較？

3. 如何將取出的數字放入已排序數列裡？

④ 重複執行上述的動作，直到未排序數列全部處理完成。

解題步驟

問題
拆解

如何從數字 1 ~ 10 中隨機取出 5 個數字，並將這 5 個數字放入未排序數列裡？

1 想一想，這句話包含了什麼重要的訊息？

1. 未排序數列：



在 Scratch 中如何記錄數列呢？

使用 **清單**，我們將這個清單命名為原始資料。



2. 隨機取出：



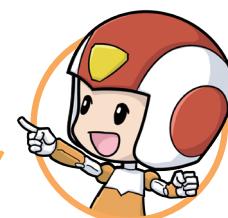
如何從數字 1~10 中隨機取出 5 個數字，並將這 5 個數字放入未排序數列裡？



使用迴圈，從數字 1~10 中隨機取出 5 個數字，並添加到原始資料清單中。



這個方式好像在前面有學過？



是啊！這個概念很重要，讓我們再練習一次。

步驟 1

設定清單。

- ① 新增清單，命名為原始資料。



**步驟
2**

請依右方提示的積木進行組裝，完成從數字 1～10 中隨機取出 5 個數字，並添加到原始資料清單中的程式。

**問題
拆解**

如何從未排序數列中逐一取出數字？

2

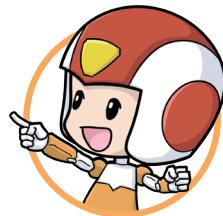
想一想，這句話包含了什麼重要的訊息？

逐一取出數字：



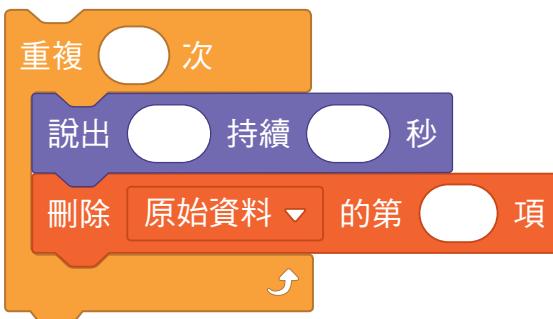
如何確保不會從未排序數列中重複取出已經排序過的元素？

取出後，將原始資料清單中的第一個數字刪除，這樣每次選取第一個數字時，都是未取過的元素。

**步驟
3**

請同學想想看，並回答下面的問題及完成右方的積木。

1. 這個迴圈要執行幾次？
2. 如何讓小貓說出：「目前從原始資料取出的數字是…」？
3. 要刪除原始資料清單的第幾項？



**問題
拆解**

如何由前往後和已排序數列的數字進行比較？遇到大於自己的數字就插入此數字之前；

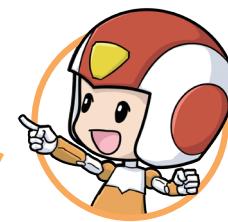
(3) 若都沒有則插入在後面。

想一想，這句話包含了什麼重要的訊息？

1. 已排序數列：



在 Scratch 中如何記錄數列呢？

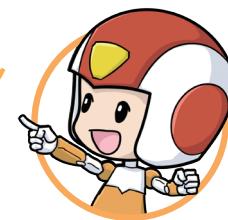


使用**清單**，我們將這個清單命名為**已排序資料**。

2. 由前往後進行比較：



如何記錄比較的資料位置？



使用**變數**，由前往後比較的目的是要找出插入資料的位置，所以我們把這個變數命名為**插入位置**。

3. 遇到大於自己的數字就插入此數字之前；若都沒有則插入在後面：



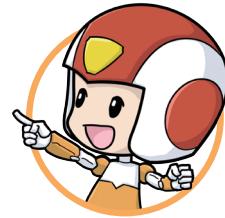
如何把資料由前往後比較，直到遇到已排序資料清單中大於自己的數字？



使用**條件式迴圈**，尋找已排序資料清單中要插入的位置，遇到大於自己的數字就跳出迴圈。



如何判斷已排序資料清單中都沒有比自己大的數字？



利用已排序資料清單的長度，如果插入位置比它大，就跳出迴圈。



想想看，這個迴圈的判斷條件該如何設定？



只要有一個條件成立就可以，所以…

步驟 4

設定清單與變數。

- ② 新增清單，命名為已排序資料。



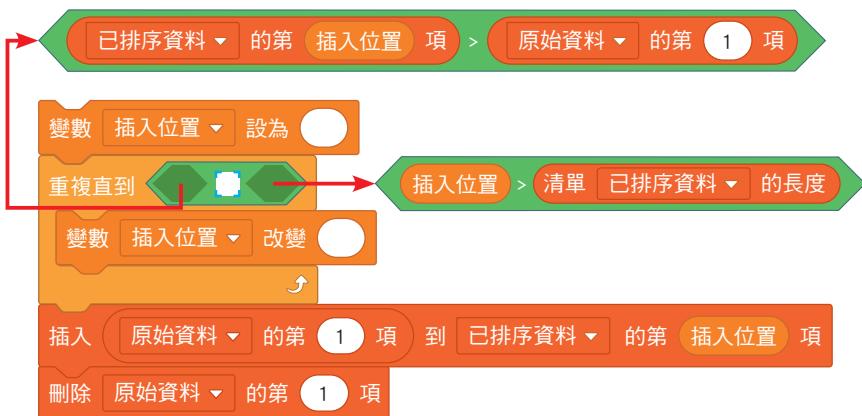
- ③ 新增變數，命名為插入位置。



步驟
5

請同學想想看，並回答下面的問題及完成右方的積木。

1. 插入位置的變數一開始要設為多少？
2. 迴圈的判斷條件應使用「或」還是「且」才正確？
3. 迴圈內插入位置的變數要改變多少？



問題
拆解

4

重複執行上述的動作，直到未排序數列全部處理完成。

想一想，這句話包含了什麼重要的訊息？

1. 重複執行上述的動作：



如何重複執行上述的動作？



將步驟 5 的條件式迴圈及相關指令，插入步驟 3 的計次式迴圈裡。

2. 直到未排序數列全部處理完成：



直到未排序數列全部處理完成的意思是什麼？

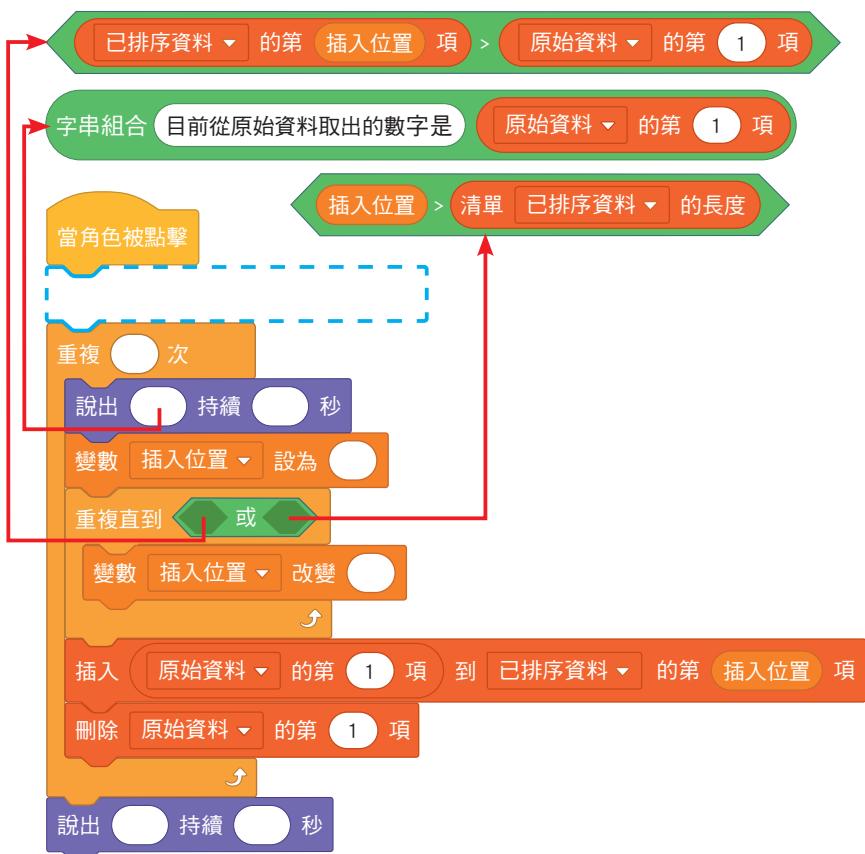


使用迴圈，將原始資料清單中的每個數字都執行一次。

**步驟
6**

請同學想想看，並回答下面的問題及完成右方的積木。

1. 是否可以把 **當角色被點擊** 換成 **當  被點擊** ？
2. 說說看，虛線的積木是什麼？
3. 繞圈要執行幾次？
4. 為什麼要讓小貓一直說出：「目前從原始資料取出的數字是…」？
5. 最後如何讓小貓說出執行結果：「這 5 個數字由小排到大的順序是…」？

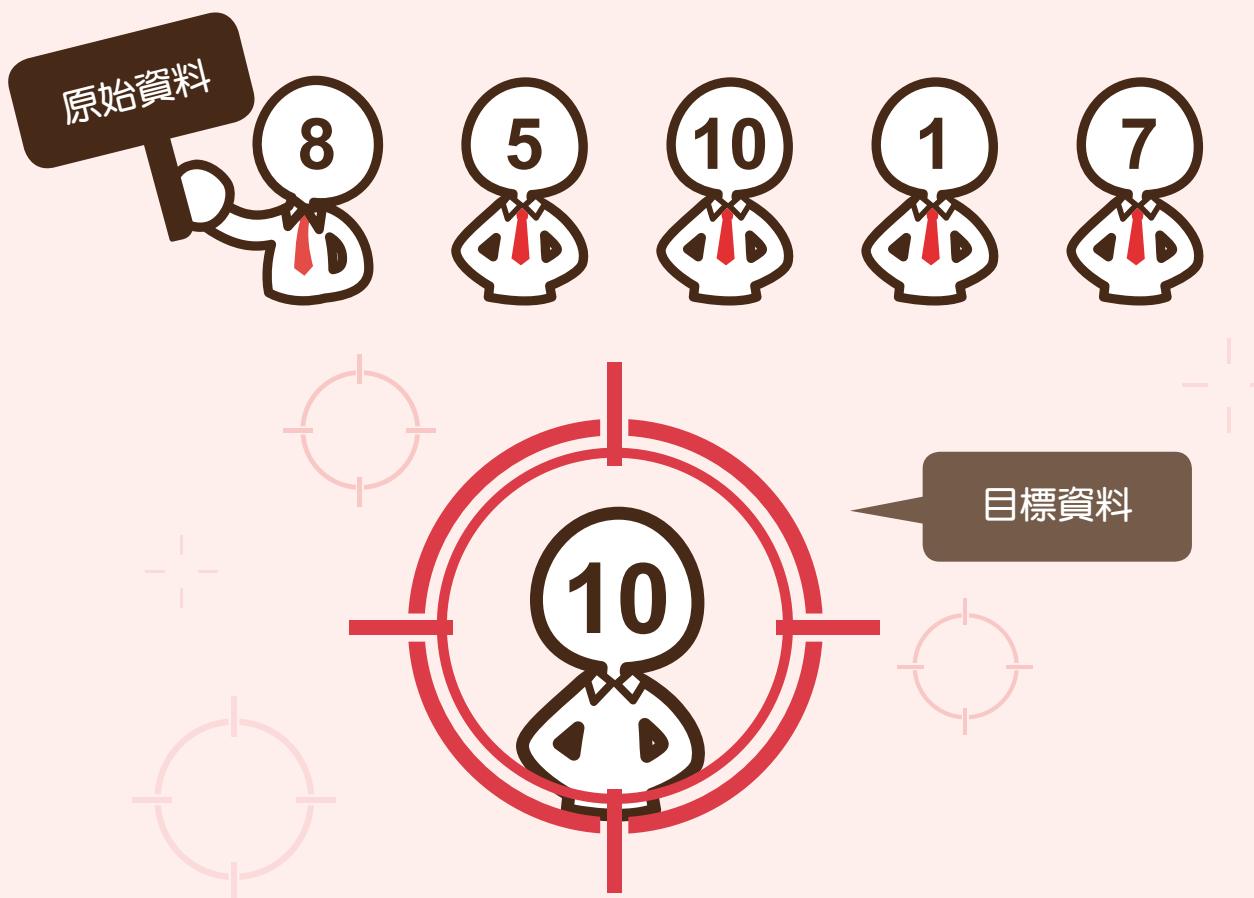


6-3 搜尋的原理與範例

首先，我們將搜尋問題簡化為數字的搜尋。假設原始資料有：8、5、10、1、7，共 5 個數字，搜尋的目標資料是 10，因為原始資料只有 5 個數字，所以你能夠一眼就看到 10 在原始資料的第 3 筆，但是如果資料量暴增到 100 個，你就得要從解決問題的過程中找出規則，將其發展成一套搜尋演算法，而這套演算法適用於各種大小的資料量與各種不同的目標資料。

6-3-1 循序搜尋法

循序搜尋法（sequential search algorithm）是用來達成搜尋特定資料。它是從第一個元素開始取出，依序逐個與「目標資料」相互比較，直到找到所要的元素或所有資料均尋找完為止，舉例如下：

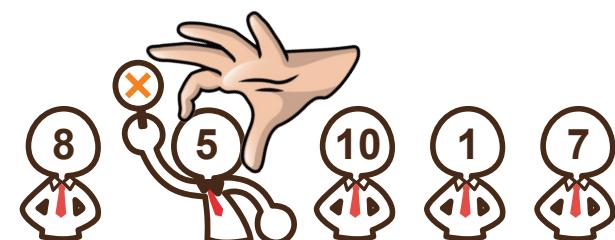


第 1 回合



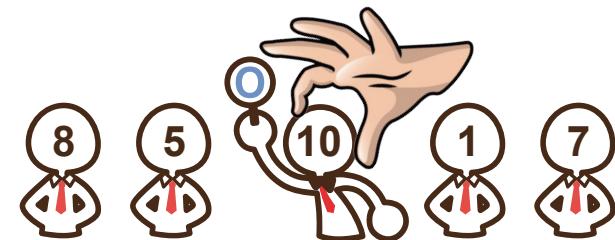
取出第 1 個元素 8，
不是目標資料 10。

第 2 回合



取出第 2 個元素 5，
不是目標資料 10。

第 3 回合

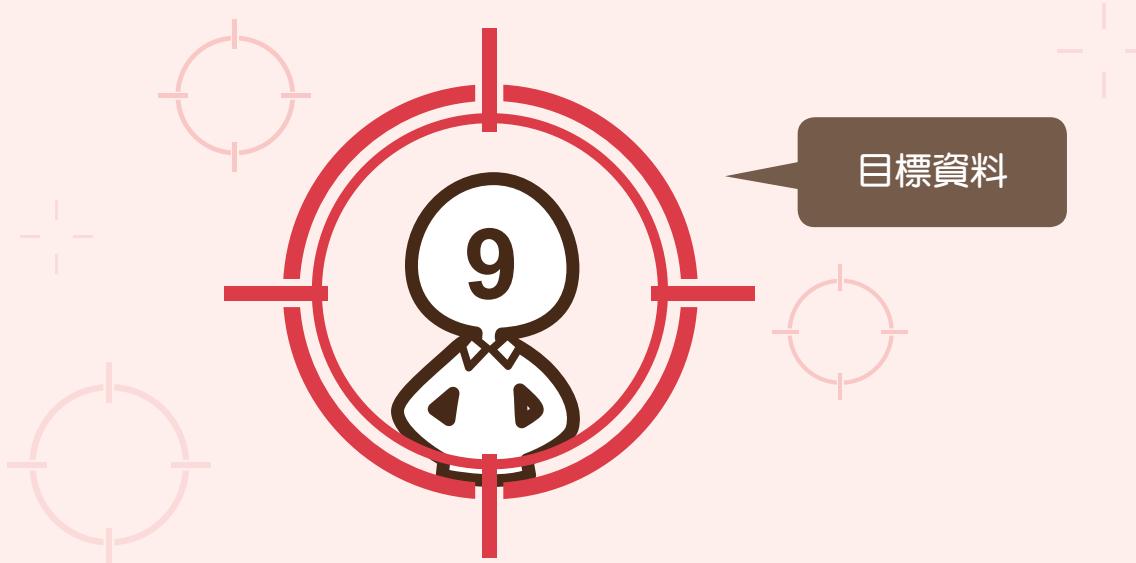


取出第 3 個元素 10，
找到目標資料。

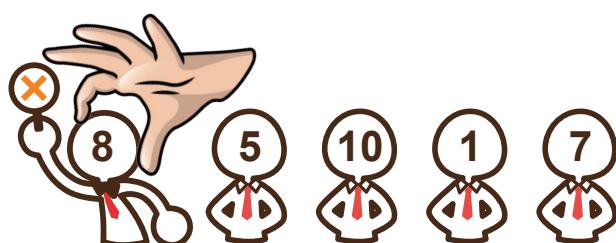
如果目標資料不在原始資料中，
流程會怎麼樣呢？

接下來，讓我來說明吧！



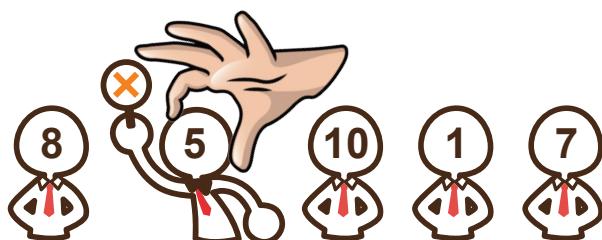


第
1
回
合



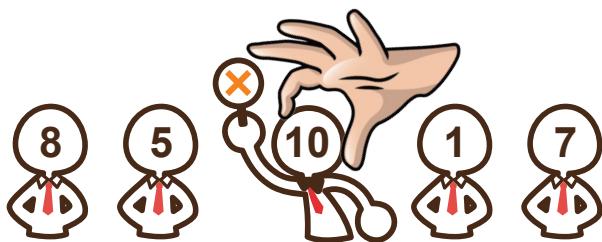
取出第 1 個元素 8，
不是目標資料 9。

第
2
回
合



取出第 2 個元素 5，
不是目標資料 9。

第 3 回合



取出第 3 個元素 10，
不是目標資料 9。

第 4 回合



取出第 4 個元素 1，
不是目標資料 9。

第 5 回合



取出第 5 個元素 7，
不是目標資料 9，搜
尋結果無此數字。

我們可以從以上的流程，歸納出實作的步驟：

1. 從未排序的原始資料中逐一取出元素。
2. 將取出元素與目標資料加以比較。
3. 重複第 1、2 點的步驟，直到找到目標資料或原始資料所有元素均比較完為止。





範例一循序搜尋法

點擊綠旗後，小貓執行下列動作：

1. 先從數字 1 ~ 100 中隨機取出 50 個數字，並將這 50 個數字放入一個原始資料裡。
 2. 請使用者任意輸入一個數字做為目標資料。
 3. 從原始資料中依序選出一個數字與目標資料進行比對。
 4. 如果目標資料與原始資料的數字相同，則說出：「找到了，位於第幾個數字」。
 5. 如果從原始資料裡都找不到與目標資料的數字相同，則說出：「沒有符合的數字」。

請執行《循序拋棄法》的程式，想一想這個範例的程式是如何運作？

1 原始資料

1	46
2	100
3	63
4	84
5	49
6	63
7	57

+ 長度 50 =

位置 0

請輸入要搜尋的數字
(1-100) :



2 原始資料

1	46
2	100
3	63
4	84
5	49
6	63
7	67

+ 長度 50 =

位置 2

目前比對的數字是100



範例執行後，原始資料呈現隨機添加的 50 個數字，並讓小貓詢問後，輸入欲搜尋的數字。

每回合依序從原始資料裡取出一個數字與目標資料進行比對，並說出：「目前比對的數字是……」。

3 原始資料

1	46
2	100
3	63
4	84
5	49
6	63
7	67
+ 長度 50	=

位置 5

找到了，位於第5個數字



4 原始資料

1	51
2	66
3	39
4	85
5	48
6	40
7	64

+ 長度 50 =

位置 51

沒有符合的數字



若在原始資料裡找到搜尋的數字，小貓說出：「找到了，位於第幾個數字」。

若在原始資料裡都沒有找到目標資料的數字，則小貓說出：「沒有符合的數字」。



問題分析

我們可以將這個程式範例拆解幾個部分如下：

① 如何從數字 1 ~ 100 中隨機取出 50 個數字，並將這 50 個數字放入一個數列裡？

1. 如何表示此數列？

2. 如何從數字 1 ~ 100 中隨機取出 50 個數字，並放入此數列裡？

② 如何從原始資料中逐一取出數字？

1. 如何逐一取出數字？

2. 如何設定停止搜尋的條件？

③ 如何將取出的數字與目標資料進行比對？

1. 如何取得目標資料？

2. 如何進行比對？

④ 重複執行上述的動作，直到從原始資料中找到與目標資料相同的數字為止；或原始資料所有數字均與目標資料比較完為止。

⑤ 如何說出比對的結果？

1. 當找到原始資料的數字與目標資料相同時，要怎麼表示？

2. 如果從原始資料都找不到與目標資料相同的數字，要怎麼表示？

解題步驟

問題
拆解

如何從數字 1 ~ 100 中隨機取出 50 個數字，並將這 50 個數字放入一個數列裡？

① 想一想，這句話包含了什麼重要的訊息？

1. 一個數列：



在 Scratch 中如何記錄數列呢？

使用 **清單**，我們將這個清單命名為原始資料。



2. 隨機取出：



如何從數字 1~100 中隨機取出 50 個數字，並將這 50 個數字放入一個數列裡？



在前面的範例裡，我們已經學過如何從數字 1~10 中隨機取出 5 個數字，並添加到原始資料清單中，現在也可以使用同樣的方法來完成。

步驟 1 設定清單。

- ① 新增清單，命名為原始資料。



步驟 2 請依右方提示的積木進行組裝，完成從數字 1~100 中隨機取出 50 個數字，並添加到原始資料清單中的程式。



問題
拆解

如何從原始資料中逐一取出數字？

② 想一想，這句話包含了什麼重要的訊息？

逐一取出數字：



如何從第一個數字循序取出到最後一個數字？



使用迴圈，逐一取出原始資料清單中的數字，等到全部取出後，就要停止搜尋。

步驟
3

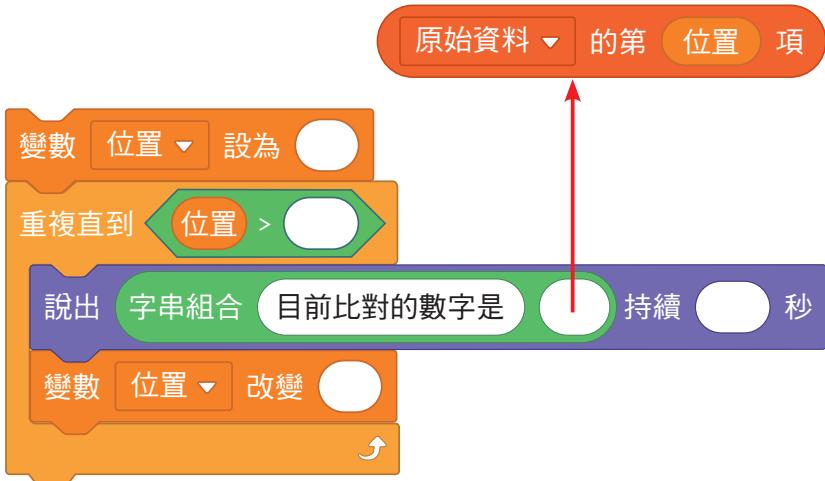
設定變數。

② 新增變數，命名為位置。

步驟
4

請同學想想看，並回答下面的問題及完成右方的積木。

1. 位置的變數一開始要設為多少？
2. 判斷條件  要填入什麼？
3. 復圈內位置的變數每次要改變多少？
4. 說說看，為什麼要放入說出積木？



問題
拆解

如何將取出的數字與目標資料進行比對？

(3) 想一想，這句話包含了什麼重要的訊息？

1. 目標資料：



如何取得目標資料？

我們可以透過**詢問**積木，請使用者輸入一個數字做為搜尋的目標資料。



2. 進行比對：



如何將目標資料與原始資料清單中的數字進行比對？

目標資料與原始資料清單中的每個數字進行比對，當目標資料與原始資料清單中的某個數字相等時，則停止比對。

步驟
5

設定詢問積木與目標資

料。

1. 請同學想想看，詢問積木在什麼類別裡？
2. 如何請使用者從數字 1～100 任意輸入一個數字？
3. 如何將使用者輸入的數字做為搜尋的目標資料？
4. 如何設定停止比對的條件？



並等待



= 原始資料 ▾ 的第 位置 項

問題
拆解

重複執行上述的動作，直到從原始資料中找到與目標資料相同的數字為止；或原始資料

4 所有數字均與目標資料比較完為止。

想一想，這句話包含了什麼重要的訊息？

直到從原始資料中找到與目標資料相同的數字為止；或原始資料所有數字均與目標資料比較完為止：



有兩個情況，如何判斷任一情況發生時都要跳出條件式迴圈？

1. 使用邏輯運算，處理 **步驟 4**、**步驟 5** 的判斷條件。

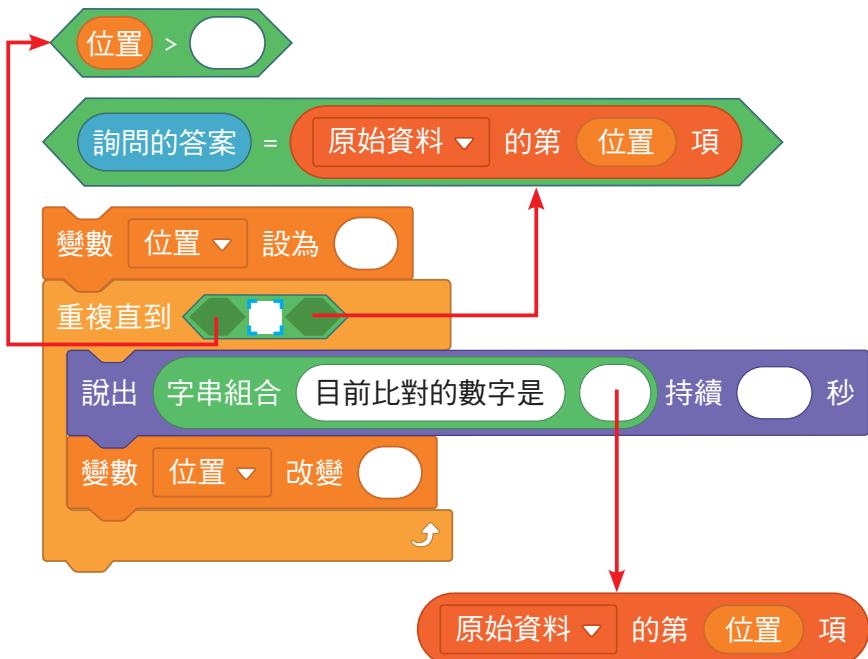
2. 整合 **步驟 4**、**步驟 5** 的程式碼。

步驟
6

請同學想想看，並回答

下面的問題及完成右方的積木。

1. 說說看，詢問積木要放在什麼位置？
2. 說說看，迴圈的邏輯運算，要使用「或」還是「且」？



問題
拆解

如何說出比對的結果？

5

想一想，這句話包含了什麼重要的訊息？

說出比對的結果：



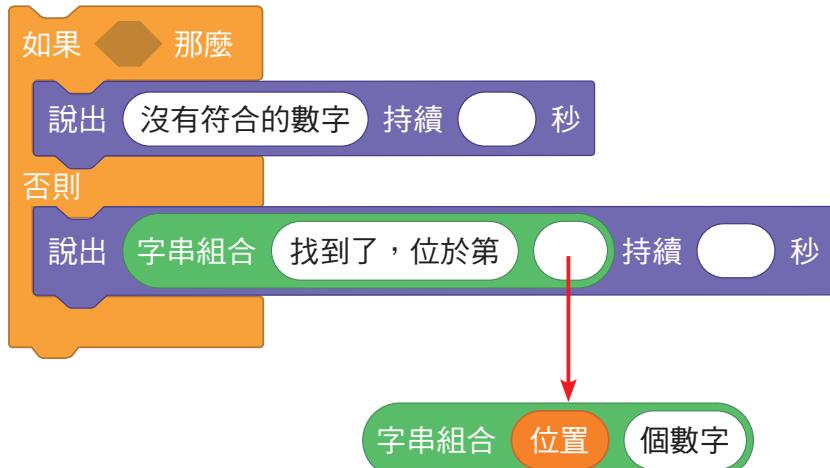
1. 當找到**原始資料清單**中的數字與**目標資料**相同時，要怎麼表示？
2. 如果從**原始資料清單**中都找不到與**目標資料**相同的數字，要怎麼表示？

使用**選擇結構**，以位置的變數來判斷，並讓小貓說出結果。

步驟
7

請同學想想看，並回答下面的問題及完成右方的積木。

1. 說說看，選擇結構的判斷條件是什麼？
2. 說說看，如果選擇結構裡上下兩層的積木對換，判斷條件要改成什麼？



6-3-2 二元搜尋法

二元搜尋法 (binary search algorithm) 是指對於已排序資料進行折半搜尋，如果欲搜尋數字比中間值大，那左半部比中間值小的數字就不用再比較，待搜尋資料量馬上少了一半。

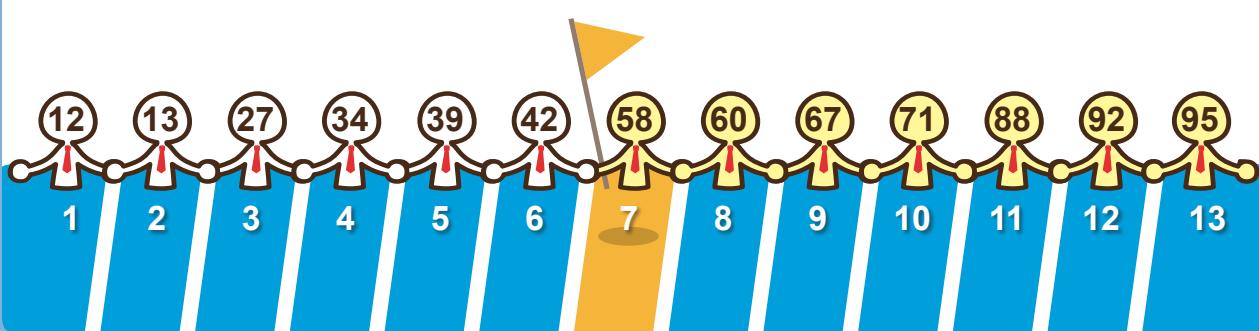
假設已排序的資料如下，在這 13 個數字中要搜尋數字 67 是否存在，若使用二元搜尋法，只需要找 4 次就可以找到，效率極高，舉例如下：



第1回合

開始位置：1
結束位置：13
二分位置： $(1+13)\div 2=7$

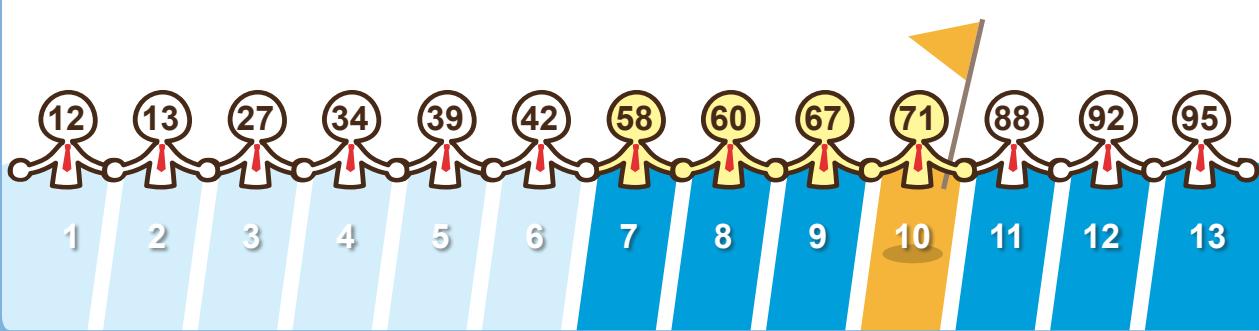
第7個元素：58
58 < 67 取右半部



第2回合

開始位置：7
結束位置：13
二分位置： $(7+13)\div 2=10$

第10個元素：71
71 > 67 取左半部

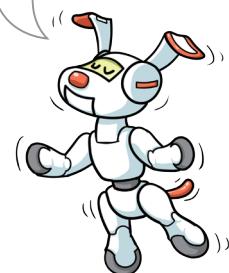


二分位置

將資料的數值由小至大排序後，將資料切分成2等分，切分的位置（二分位置）即為整個數列的中間位置。

想想看，如果二分位置非整數時，該取哪個位置呢？

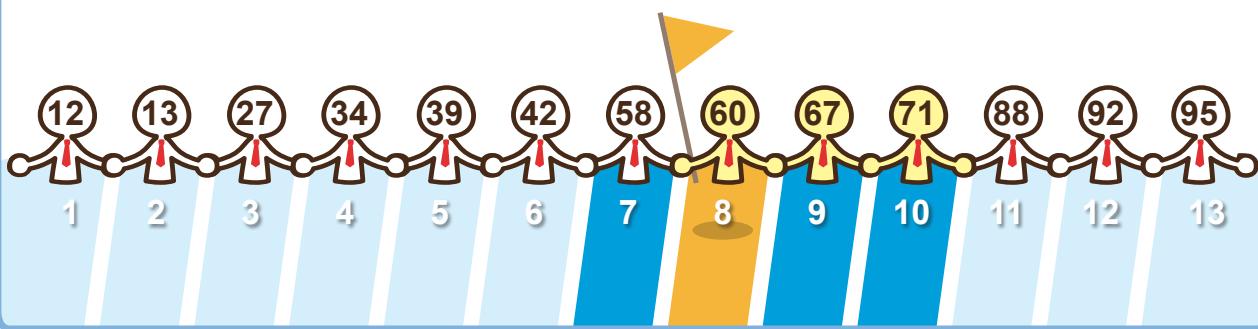
可以取整數部分，當成二分位置。



第 3 回合

開始位置 : 7
 結束位置 : 10
 二分位置 : $(7 + 10) \div 2 = 8.5$
 (取 8.5 的整數部分 8)

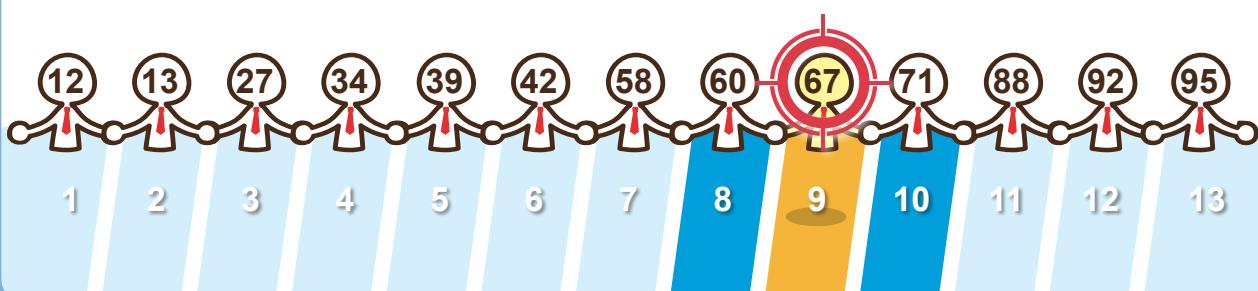
第 8 個元素 : 60
 $60 < 67$ 取右半部



第 4 回合

開始位置 : 8
 結束位置 : 10
 二分位置 : $(8 + 10) \div 2 = 9$

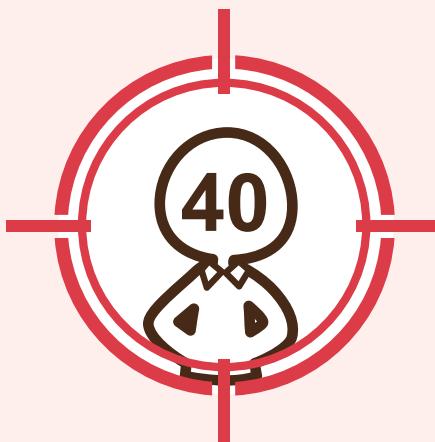
第 9 個元素 : 67
 $67 = 67$ 找到目標資料



嘆！那找不到目標資料時，流程是如何進行呢？

讓我們繼續看下去吧！





第1回合

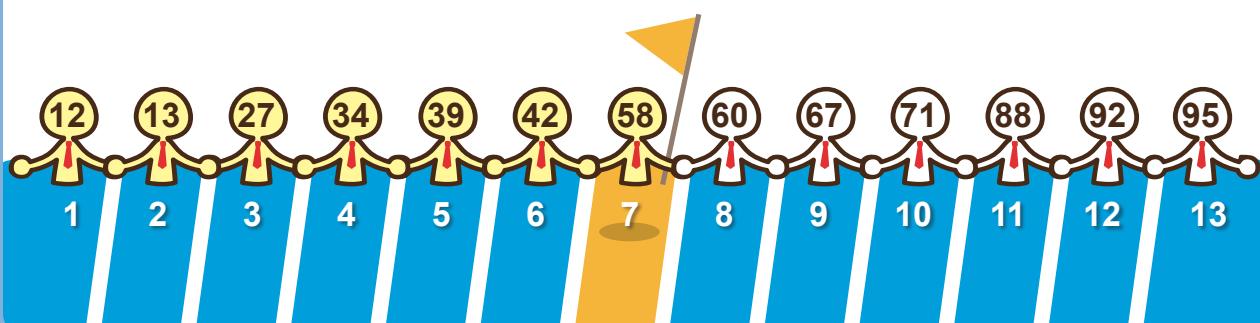
開始位置：1

結束位置：13

二分位置： $(1 + 13) \div 2 = 7$

第7個元素：58

$58 > 40$ 取左半部



第2回合

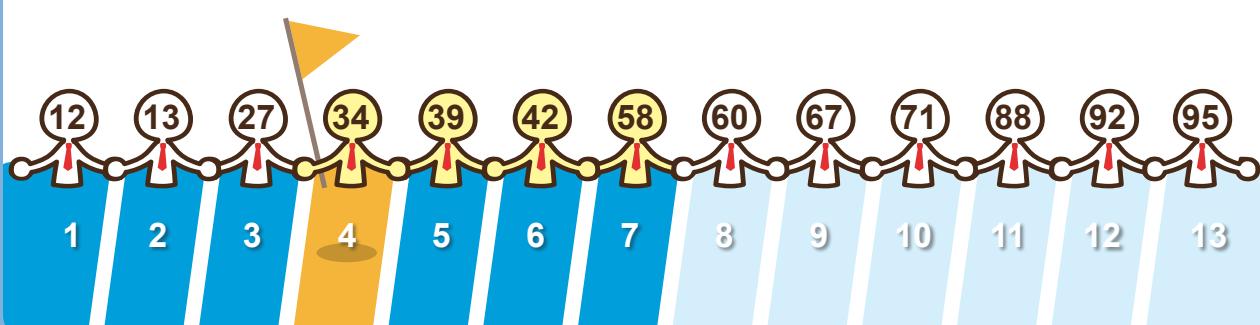
開始位置：1

結束位置：7

二分位置： $(1 + 7) \div 2 = 4$

第4個元素：34

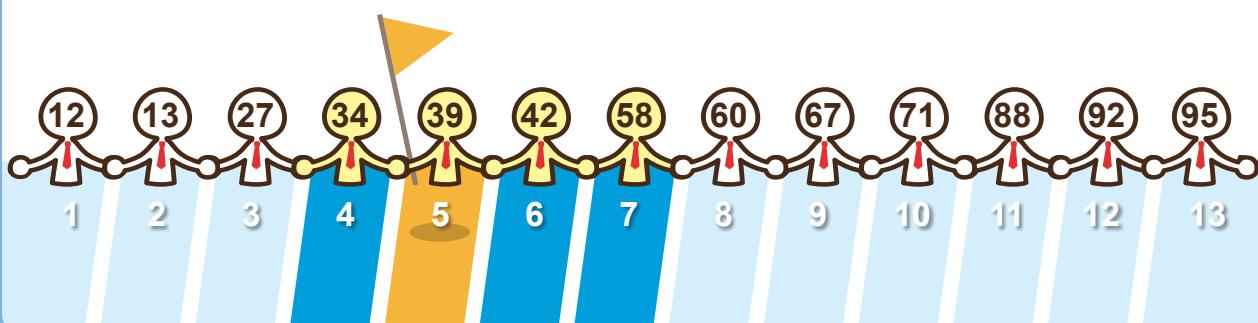
$34 < 40$ 取右半部



第 3 回合

開始位置：4
結束位置：7
二分位置： $(4+7)\div 2=5.5$
(取 5.5 的整數部分 5)

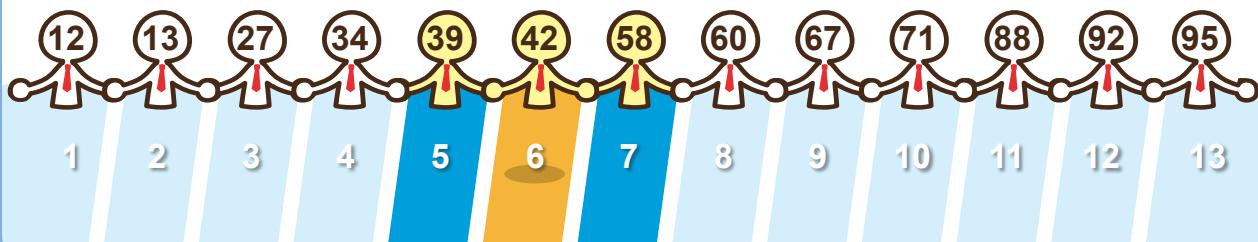
第 5 個元素：39
 $39 < 40$ 取右半部



第 4 回合

開始位置：5
結束位置：7
二分位置： $(5+7)\div 2=6$

第 6 個元素：42
 $42 > 40$ ，二分位置 = 結束位置 - 1，搜尋結果無此數字。



我們可以從以上的流程，歸納出實作的步驟：

1. 從已排序原始資料的二分位置，取出元素。
2. 將取出元素與目標資料加以比較。若目標資料大於該元素，則排除左半部數列；目標資料小於該元素，則排除右半部數列。
3. 重複第 1、2 點的步驟，直到找到目標資料或原始資料所有元素均比較完為止。



在進行二元搜尋法時，原始資料裡的數字要由小到大排列好，首先我們先從已經由小到大排列的資料進行二元搜尋法。



範例—二元搜尋法 1 (原始資料已排序)

點擊綠旗後，小貓執行下列動作：

1. 建立一個已排序的原始資料。
2. 請使用者任意輸入一個數字做為目標資料。
3. 將原始資料 2 等分，得到二分位置所代表的數字與目標資料進行比對。
4. 如果二分位置所代表的數字與目標資料相同，則說出：「找到了，位於第幾個數字」。
5. 如果從原始資料裡都找不到與目標資料的數字相同，則說出：「沒有符合的數字」。

請執行《二元搜尋法 1》的程式，想一想這個範例的程式是如何運作？

1

```

    list [ ]
    var [Start position v] is [0]
    var [End position v] is [0]
    var [Binary position v] is [0]
  
```

原始資料
(empty)

開始位置 0
結束位置 0
二分位置 0

+ 長度 0 =

範例執行前，原始資料裡內容是空的。

2

```

    list [5, 7, 12, 14, 15, 17, 19]
    var [Start position v] is [1]
    var [End position v] is [50]
    var [Binary position v] is [0]
  
```

原始資料

開始位置 1
結束位置 50
二分位置 0

請輸入要搜尋的數字
(1-100) :

範例執行後，匯入已排序資料，接著點選小貓後，詢問欲搜尋的數字並輸入數字。

3

```

    list [5, 7, 12, 14, 15, 17, 18]
    var [Start position v] is [8]
    var [End position v] is [10]
    var [Binary position v] is [9]
  
```

原始資料

開始位置 8
結束位置 10
二分位置 9

找到了，位於第9個數字

若在原始資料裡找到搜尋的數字，小貓說出：「找到了，位於第幾個數字」。

4

```

    list [55, 56, 57, 59, 60, 62, 73]
    var [Start position v] is [34]
    var [End position v] is [35]
    var [Binary position v] is [34]
  
```

原始資料

開始位置 34
結束位置 35
二分位置 34

沒有符合的數字

若搜尋的數字未在原始資料裡，則小貓說出：「沒有符合的數字」。



問題分析

我們可以將這個程式範例拆解幾個部分如下：

① 如何將現有的已排序數列放入原始資料裡？

1. 如何使用現有的已排序數列？
2. 如何表示原始資料？

② 如何從已排序數列的範圍中找出位於中間位置的數字？

1. 如何定義數列的範圍？
2. 如何找出中間位置的數字？
3. 當計算出來的中間位置不是整數時，要如何處理？
4. 在 Scratch 中，如何表示取整數？

③ 如何將二分位置所表示的數字與目標資料進行比對？

1. 如何取得目標資料？
2. 如何進行比對？

④ 重複執行上述的動作，如果還有資料則進行比對，最後說出比對的結果。

1. 如何判斷還有沒有資料？
2. 要怎麼說出比對的結果？

解題步驟

問題
拆解

如何將現有的已排序數列放入原始資料裡？

1 想一想，這句話包含了什麼重要的訊息？

1. 現有的已排序數列：

小提示

匯入現有數列的其他方式

1. **添加** **到** **原始資料** ：依序新增設

定的數值至原始資料清單中。

2. **插入** **到** **原始資料** 的第 **項** ：

新增設定的數值至原始資料清單中指定的項次。



如何使用現有的已排序數列？

可從電腦中，匯入已排序數列的文字檔。



2. 原始資料：



在 Scratch 中如何表示原始資料呢？



使用**清單**，我們將這個清單命名為**原始資料**。

**步驟
1**

設定清單。

- 新增**清單**，命名為**原始資料**。



**步驟
2**

匯入資料。

- 在**原始資料**清單上，按右鍵選擇**匯入**。
- 從電腦中，匯入**已排序的原始資料**。



問題
拆解

如何從已排序數列的範圍中找出位於中間位置的數字？

(2) 想一想，這句話包含了什麼重要的訊息？

1. 已排序數列的範圍：



如何定義數列的範圍，以便找出數字個數？



使用開始位置和結束位置的變數，就可以定義數列的範圍。

2. 找出位於中間位置的數字：



如何找出位於中間位置的數字？



1. 把開始位置與結束位置相加除以 2，就是**中間位置**。

2. 因為是 2 等分，所以我們就把中間位置用變數來命名為**二分位置**。



如果把開始位置與結束位置相加除以 2 的結果是小數時，要怎麼處理？



這時就只要取**整數**部分即可。



在 Scratch 中，如何表示取整數？



取整數的意思就是不要個位數以下的數，也就是小數點後的數都捨去，在數學上就是無條件捨去的概念，所以…

步驟
3

設定變數。

- ④ 新增變數，分別命名為開始位置、結束位置與二分位置。

步驟
4

請同學想想看，並回答下面的問題及完成右方的積木。

1. 開始位置的變數一開始要設為多少？
2. 結束位置的變數一開始要設為多少？
3. 二分位置的變數一開始要設為多少？
4. 無條件捨去的積木在什麼類別裡？
5. 此處的 怎麼設定？



問題
拆解

如何將二分位置所表示的數字與目標資料進行比對？

(3)

想一想，這句話包含了什麼重要的訊息？

1. 目標資料：



如何取得目標資料？



我們可以透過**詢問**積木，請使用者輸入一個數字做為搜尋的目標資料。

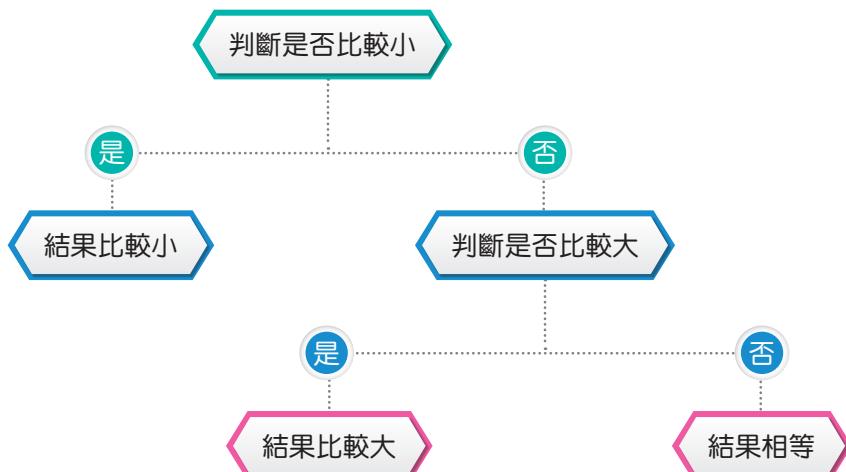
2. 進行比對：



比對的結果只會是較小、較大或相等這三種情況中的一種，如何用程式表達這種三選一的情況？



使用兩個**選擇結構**，第一個選擇結構先判斷是否比較小，如果不成立；再用第二個選擇結構判斷是否比較大，如果也不成立，就是相等。

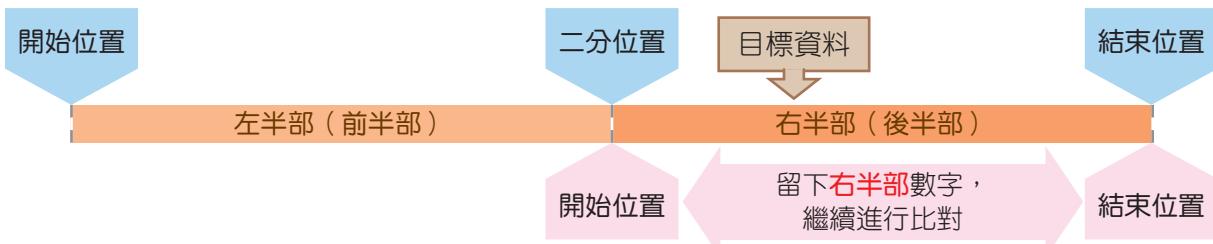




如果二分位置所表示的數字比較小，要留下哪些數字繼續進行比對？



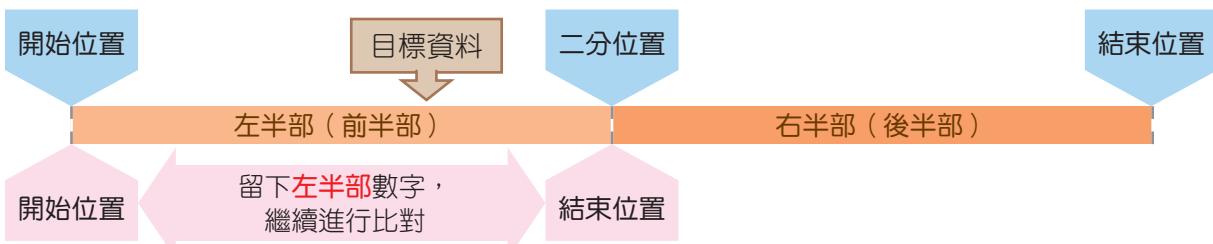
1. 因為右半部（後半部）數字比較大，即目標數字在**右半部**，故留下右半部數字。
2. 此時設定開始位置 = 二分位置，再繼續將右半部數字 2 等分後，重新取二分位置與目標資料進行比對。



如果二分位置所表示的數字比較大，要留下哪些數字繼續進行比對？



1. 因為左半部（前半部）數字比較小，即目標數字在**左半部**，故留下左半部數字。
2. 此時設定結束位置 = 二分位置，再繼續將左半部數字 2 等分後，重新取二分位置與目標資料進行比對





如果二分位置所表示的數字與目標資料相等時，程式要如何設定？



此時表示已經找到了，就用二分位置所表示的數字 = 目標資料，做為判斷條件。



也有可能發生目標資料不在待搜尋數列中，要如何處理這種找不到的情況？



此時表示找不到了，就用二分位置 = 結束位置 - 1，做為判斷條件。



為什麼找不到目標資料的判斷條件是這樣設定呢？



因為開始比較後，設定開始位置 = 二分位置或結束位置 = 二分位置，所以開始位置與結束位置會越來越接近，執行到最後，開始位置、二分位置與結束位置都會相差 1，這個情況就是找不到目標資料。

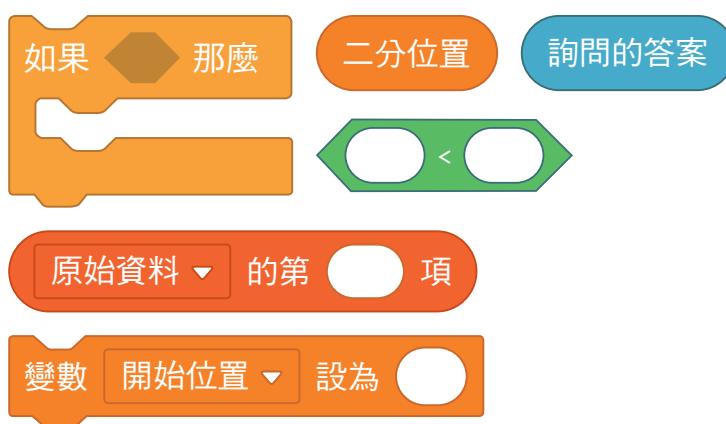
**步驟
5**

設定詢問積木與目標資料。

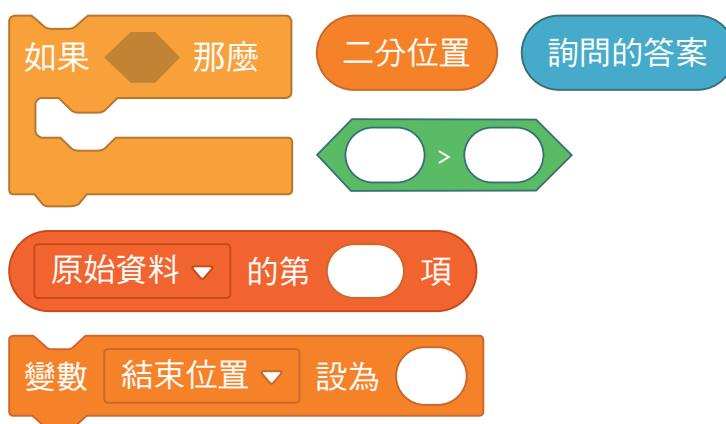
1. 請同學想想看，詢問積木在什麼類別裡？
2. 如何請使用者從數字 1~100 任意輸入一個數字？
3. 如何將使用者輸入的數字做為搜尋的目標資料？
4. 請依右方提示的積木進行組裝，完成詢問積木與目標資料的設定。

**步驟
6**

請依右方提示的積木進行組裝，完成二分位置所代表的數字小於目標資料的程式。

**步驟
7**

請依右方提示的積木進行組裝，完成二分位置所代表的數字大於目標資料的程式。



問題
拆解

重複執行上述的動作，如果還有資料則進行比對，最後說出比對的結果。

4

想一想，這句話包含了什麼重要的訊息？

1. 還有資料：



如何判斷還有沒有資料？

1. 如果二分位置所表示的數字 = 目標資料，表示已經找到目標資料。
2. 如果二分位置 = 結束位置 - 1，表示該找的都找過了，最後並沒有找到目標資料。



2. 說出比對的結果：



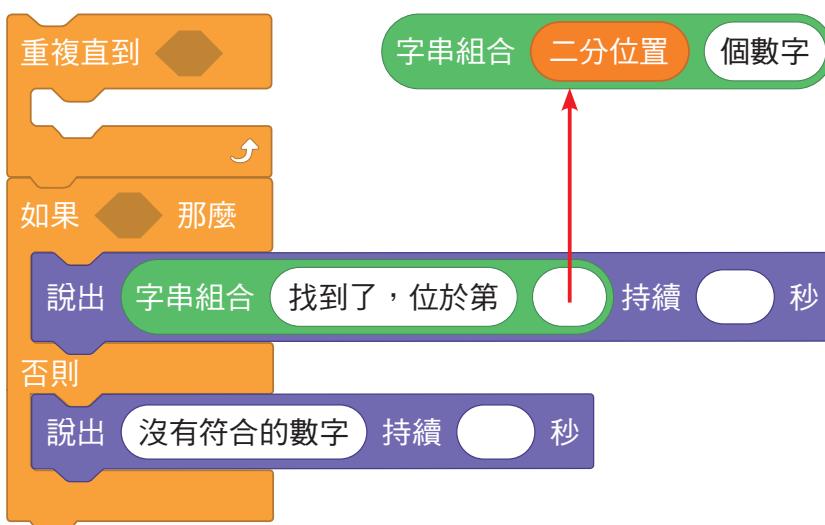
要使用哪一種重複結構？

因為我們不知道要重複幾次，所以不能用計次式迴圈。而且結束搜尋是在某些特定狀況下達成，所以使用**條件式迴圈**。

步驟
8

請同學想想看，並回答下面的問題及完成右方的積木。

1. 說說看，條件式迴圈積木要如何設定停止條件？
2. 說說看，選擇結構的判斷條件是什麼？
3. 說說看，如果選擇結構裡上下兩層的積木對換，判斷條件要改成什麼？



二元搜尋法必須應用在資料已排序的情況下，所以遇到資料未排序的情況，就要先使用排序演算法將資料排序。



範例一 二元搜尋法 2 (原始資料未排序)

點擊綠旗後，小貓執行下列動作：

1. 先從數字 1 ~ 100 中隨機取出 50 個數字，並將這 50 個數字放入一個原始資料裡。
2. 使用選擇排序法，將這 50 個數字由小到大排列。
3. 請使用者任意輸入一個數字做為目標資料。
4. 使用二元搜尋法進行目標資料的搜尋。
5. 如果二分位置所代表的數字與目標資料相同，則說出：「找到了，位於第幾個數字」。
6. 如果從原始資料裡都找不到與目標資料的數字相同，則說出：「沒有符合的數字」。

請執行《二元搜尋法 2》的程式，想一想這個範例的程式是如何運作？

1

2

範例執行後，原始資料呈現隨機添加的 50 個數字，並由小到大排列後，儲存到已排序資料裡。

3

若在已排序資料裡找到搜尋的數字，小貓說出：「找到了，位於第幾個數字」。

4

若搜尋的數字未在已排序資料裡，則小貓說出：「沒有符合的數字」。



問題分析

我們可以將這個程式範例拆解幾個部分如下：

- ① 如何從數字 1 ~ 100 中隨機取出 50 個數字，放入未排序數列裡，並將此數列裡的數字由小到大排列，儲存在已排序數列中？
 1. 如何表示未排序數列？
 2. 如何從數字 1 ~ 100 中隨機取出 50 個數字，並放入未排序數列裡？
 3. 如何將原始資料的數字由小到大排列，並儲存在已排序資料裡？
- ② 如何從已排序數列的範圍中找出位於中間位置的數字？
 1. 如何定義數列的範圍？
 2. 如何找出中間位置的數字？
 3. 當計算出來的中間位置不是整數時，要如何處理？
 4. 在 Scratch 中，如何表示取整數？
- ③ 如何將二分位置所表示的數字與目標資料進行比對？
 1. 如何取得目標資料？
 2. 如何進行比對？
- ④ 重複執行上述的動作，如果還有資料則進行比對，最後說出比對的結果。
 1. 如何判斷還有沒有資料？
 2. 要怎麼說出比對的結果？

解題步驟

問題
拆解

如何從數字 1 ~ 100 中隨機取出 50 個數字，放入未排序數列裡，並將此數列裡的數字

1 由小到大排列，儲存在已排序數列中？

想一想，這句話包含了什麼重要的訊息？

1. 未排序數列和已排序數列：



在 Scratch 中如何記錄數列呢？

使用 **清單**，我們新增兩個清單命名為原始資料
和已排序資料。



2. 隨機取出：



如何從數字 1~100 中隨機取出 50 個數字，並將這 50 個數字放入未排序數列裡？



還記得前面的範例是如何進行的嗎？同學們如果忘記了，趕快溫習一下喔！

3. 由小到大排列：



如何將原始資料清單中的數字由小到大排序，儲存在已排序資料清單？



套用我們之前學過的排序演算法，例如：選擇排序法。

步驟 1

設定清單。

- 新增清單，分別命名為原始資料與已排序資料。

The image shows the Scratch script editor interface. On the left, the script palette is open with the '變數' (Variables) category selected. It contains four scripts: '建立一個變數' (Create a variable), 'my variable' set to 0; '變數 my variable 改變 [1]' (Change variable my variable by 1); '變數 my variable 顯示' (Show variable my variable); and '變數 my variable 隱藏' (Hide variable my variable). Below these are '建立一個清單' (Create a list) and '建立一個積木' (Create a block).

Two dialog boxes are overlaid on the screen:

- 新的清單 (New List) dialog:** Shows a text input field with '1 新清單的名稱' (Name of new list) containing '原始資料' (Original Data). A radio button group below it has '適用於所有角色' (For all sprites) selected. Buttons at the bottom are '取消' (Cancel) and '確定' (OK).
- 新的清單 (New List) dialog:** Shows a text input field with '1 新清單的名稱' (Name of new list) containing '已排序資料' (Sorted Data). A radio button group below it has '適用於所有角色' (For all sprites) selected. Buttons at the bottom are '取消' (Cancel) and '確定' (OK).

步驟
2

設定函式積木。

- ② 新增函式積木，分別命名為找出最小值位置與選擇排序法。



步驟
3

請依右方提示的積木進行組裝，完成從數字 1 ~ 100 中隨機取出 50 個數字，並添加到原始資料清單中的程式。

需不需要刪除已排序資料的積木呢？

當 被點擊

隨機取數 到

重複 次

刪除 的所有項目

刪除 的所有項目

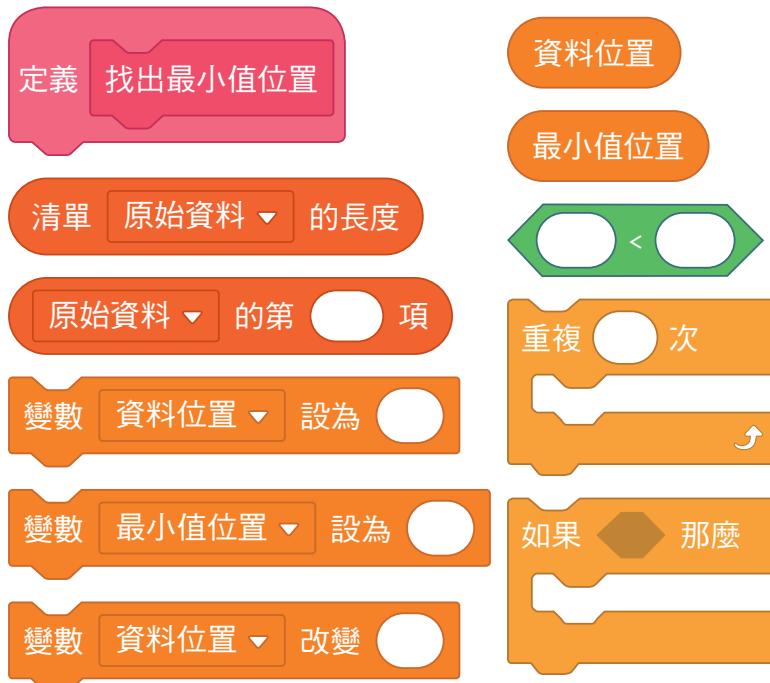
添加 到



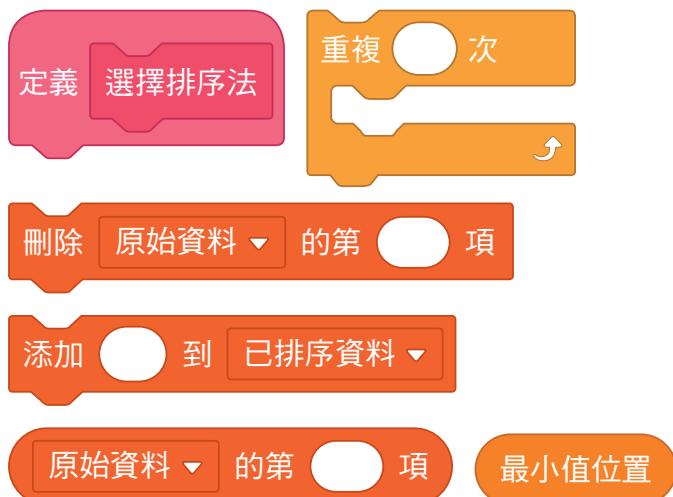
同學們，如果忘記了，趕快溫習一下喔。

**步驟
4**

請依右方提示的積木進行組裝，完成找出最小值位置的副程式。

**步驟
5**

請依右方提示的積木進行組裝，完成選擇排序法的副程式。



想想看，要把 [選擇排序法] 與
[找出最小值位置] 插在 **步驟 3 ~ 5**
的哪個積木下方？



想想看，如果使用插入排序法進行數字的大小比較，程式要如何撰寫呢？

問題
拆解

2

步驟
6

如何從已排序數列的範圍中找出位於中間位置的數字？

設定變數。

- ③ 新增**變數**，分別命名為開始位置、結束位置與二分位置。

步驟
7

請同學想想看，並回答下面的問題及完成右方的積木。

1. 開始位置的變數一開始要設為多少？
2. 結束位置的變數一開始要設為多少？
3. 二分位置的變數一開始要設為多少？
4. 無條件捨去的積木在什麼類別裡？
5. 此處的 怎麼設定？



問題
拆解

如何將二分位置所表示的數字與目標資料進行比對？

3

想一想，如何取得目標資料，並判斷比對的結果為較小、較大或相等這三種的情況？

步驟
8

設定詢問積木與目標資料。

1. 請同學想想看，詢問積木在什麼類別裡？
2. 如何請使用者從數字 1~100 任意輸入一個數字？
3. 如何將使用者輸入的數字做為搜尋的目標資料？
4. 請依右方提示的積木進行組裝，完成詢問積木與目標資料的設定。

詢問的答案

二元搜尋法

當角色被點擊

詢問 並等待

請先完成二元搜尋法的副程式。

問題
拆解

重複執行上述的動作，如果還有資料則進行比對，最後說出比對的結果。

4

想一想，如何使用重複結構來比對完所有資料，並說出結果？

步驟
9

請依右方提示的積木進行組裝，完成所有資料的比對後，並說出比對結果的程式。

重複直到 如果 那麼字串組合

否則

說出 持續 秒

二分位置



還記得怎麼設定迴圈的停止條件與選擇結構的判斷條件嗎？趕快複習《二元搜尋法 1》吧！



重點回顧

演算法的概念

演算法就是解決問題的方法。在資訊科技領域，演算法是一個交由電腦進行計算的具體步驟，它是一組有限運算規則的集合，包含問題的輸入、處理、輸出等。

演算法的表示方法

演算法可以利用文字敘述、流程圖或其他方式表示。

選擇排序法

概念是反覆從未排序的資料中取出最小的元素，加入到另一個資料的最後一項，當所有元素都取出後，它的結果就是已排序的資料。

插入排序法

概念是逐一取出未排序資料中的元素，再從另一個已排序資料中，由前往後找到適當的位置插入，就完成資料的排序。

循序搜尋法

用來搜尋特定資料的方法。從第一個資料開始取出，依序逐一與「目標資料」相互比較，直到找到所要的元素或所有資料均尋找完為止。

二元搜尋法

對於已排序資料進行折半搜尋，如果要搜尋的元素比中間值大，那另一半比中間值小的元素就不用再比較，待搜尋的資料馬上減少一半，反覆進行此步驟，就可以快速找到「目標資料」。

資訊圖片來源

第4章 進階程式設計(2)

頁碼	圖號內容	提供來源
120-121	章首頁示意圖	dreamstime 提供 編輯部製作
122	圖 4-1 校務行政系統的處室模組示意圖	編輯部製作
123	圖 4-2 電腦主機包含許多模組示意圖	depositphotos 提供
124-125	圖 4-3 副程式的運用示意圖	編輯部製作
145	程式設定說明背景	pixabay 提供

第5章 媒體與資訊科技相關社會議題

頁碼	圖號內容	提供來源
148-149	章首頁示意圖	dreamstime 提供
150	圖 5-1 直播示意圖	Adobe Stock 提供
151	圖 5-2 網路霸凌與網路成癮示意圖	Adobe Stock 提供
153	錯誤資訊事件示意圖	編輯部製作
154	不實資訊事件示意圖	編輯部製作
155	惡意資訊事件示意圖	編輯部製作
156	圖 5-3 防範不實資訊的三不二要原則示意圖	編輯部製作
157	圖 5-4 表達與溝通方式示意圖	pixabay 提供 編輯部製作
158	圖 5-5 言論自由示意圖	depositphotos 提供
159	圖 5-6 教育部防制校園霸凌相關海報	教育部防制校園霸凌專區
160	圖 5-7 文字嘲弄示意圖	Adobe Stock 提供
160	圖 5-8 圖像騷擾示意圖	depositphotos 提供
160	圖 5-9 訊息恐嚇示意圖	depositphotos 提供
161	圖 5-10 社交孤立示意圖	depositphotos 提供
161	圖 5-11 禁止網路霸凌示意圖	depositphotos 提供
162	面對網路霸凌的正確觀念	編輯部製作
164	圖 5-12 造成網路成癮示意圖	編輯部製作

第6章 基本演算法的介紹

頁碼	圖號內容	提供來源
166-167	章首頁示意圖	depositphotos 提供 Adobe Stock 提供 pixabay 提供 編輯部製作
168	圖 6-1 使用文字敘述表示的演算法	編輯部製作
168	圖 6-2 使用流程圖表示的演算法	編輯部製作
169	圖 6-3 不同交通方式會消耗不同的成本	pixabay 提供 編輯部製作
170	圖 6-4 排序不同的資料量	編輯部製作
171-172	選擇排序法示意圖	depositphotos 提供 編輯部製作
181-182	插入排序法示意圖	depositphotos 提供 編輯部製作
191-194	循序搜尋法示意圖	depositphotos 提供 編輯部製作
202-206	二元搜尋法示意圖	depositphotos 提供 編輯部製作

資訊中英文對照

第4章 進階程式設計(2)

頁碼	中文	英文
123	中央處理器	CPU (Central Processing Unit)
124	副程式	sub-program
132	參數	parameter
132	布林值	boolean value
132	真	true
132	假	false

第6章 基本演算法的介紹

頁碼	中文	英文
168	演算法	algorithm
171	選擇排序法	selection sort algorithm
181	插入排序法	insertion sort algorithm
191	循序搜尋法	sequential search algorithm
202	二元搜尋法	binary search algorithm

第5章 媒體與資訊科技相關社會議題

頁碼	中文	英文
150	媒體素養	media literacy
150	麥克盧爾	Charles Rosemary McClure
150	資訊素養	information literacy
150	直播	live broadcast
151	資訊失序	information disorder
151	網路霸凌	cyberbullying
151	網路成癮	internet addiction
152	哥德堡大學	University of Gothenburg
152	政治學跨國調查計劃	V-Dem (Varieties of Democracy)
152	易普索市場研究股份有限公司	Ipsos
152	假新聞	fake news
152	錯誤資訊	mis-information
152	不實資訊	dis-information
152	惡意資訊	mal-information
159	校園霸凌	school bullying

利用英文名稱，可以搜尋到更多相關資料！



●第 4 章 程式參考解答●



《畫平行排列的正方形》參考解答

對應課本第 126 頁



```

當旗子被點擊
  筆跡全部清除
  定位到 x: -140 y: -20
  重複 6 次
    畫正方形
    移動 60 點
  結束重複

```

```

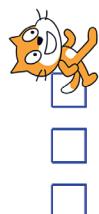
定義 畫正方形
  下筆
  重複 4 次
    移動 30 點
    右轉 90 度
  停筆

```



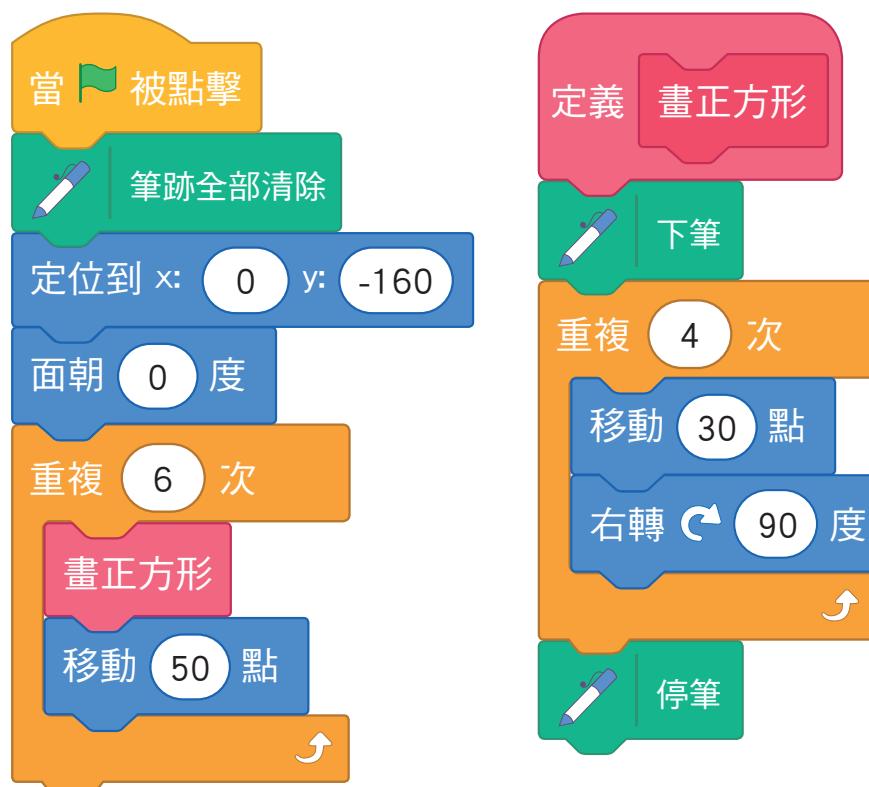
《練習》參考解答

對應課本第 130 頁



程式執行後，小貓從定位位置向上走動固定
的距離，並依序畫出正方形。

小貓向上依序畫完六個平行排列的正方形。





《畫逐漸擴大的正方形》參考解答

對應課本第 132 頁



```
當旗子被點擊
  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
```

```
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
```

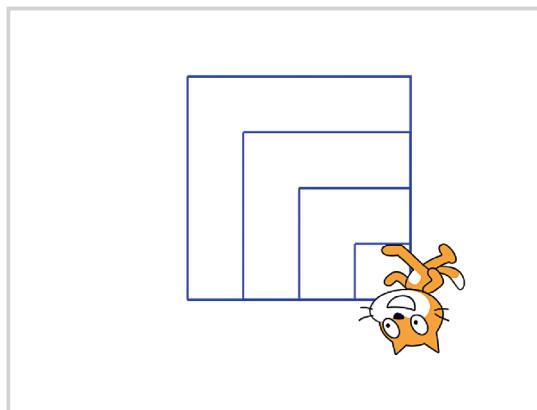


《練習》參考解答

對應課本第 139 頁



程式執行後，小貓移動至定位的位置，向左
畫出第一個正方形。



小貓向左畫完四個逐漸擴大的正方形。

```

當旗幟被點擊
  筆跡全部清除
  定位到 x: 120 y: -80
  面朝 -90 度
  畫正方形 50
  畫正方形 100
  畫正方形 150
  畫正方形 200
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

```



《小鳥吃蟲》參考解答

對應課本第 142 頁



```

當 [被點擊]
    顯示
    重複 (10 次)
        產生蟲
    隱藏

```

```

定義 [產生蟲]
    定位到 [隨機] 位置
    建立 [自己] 的分身

```

```

當 [被點擊]
    重複無限次
        定位到 [鼠標] 位置
        如果 [滑鼠鍵被按下?]
            則 [造型換成 小鳥彎腰]
        否則
            [造型換成 小鳥站立]

```

```

當 [分身產生]
    重複無限次
        如果 [碰到顏色 [紅色] ? 且 [滑鼠鍵被按下?]
            則 [產生蟲]
            [分身刪除]
        ]

```

第 5 章 資訊失序補充案例

對應課本第 152 頁

案例 1

謠言終結站－法務部要將所有非法來臺者遞解出境？

【網傳內容】：法務部於 12 月 9 日通報各執法機關，鑑於近月偷渡前來臺灣人數急增，法務部決定將所有非法進入臺灣人士分批遞解出境，並經嚴格檢查政治難民申請。

【查證結果】：法務部澄清，「本部非入出境移民及遞解偷渡犯出境業務主管機關」，不能也不會發布上述訊息；該訊息冒用法務部中英文名義及部徽，已交由調查局調查不實資訊來源。

自由時報，2019-12-10

案例 2

散布動物傳染病謠言入法－動物傳染病防治條例

「動物傳染病防治條例部分條文修正案」，未來故意散播有關動物傳染病之謠言或不實訊息，足生損害於公眾或他人者，將可處五萬元以上、一百萬元以下罰鍰。

聯合報，2019-12-03

案例 3

散布災害謠言入法－災害防救法條例

立法院會三讀修正通過，《災害防救法》第 41 條條文明定，若明知災害不實訊息，卻通報消防或警察單位者，可處新台幣 30 萬以上、50 萬元以下罰金。

三讀條文也明定，若散播有關災害的謠言或不實訊息，足生損害於公眾或他人者，處 3 年以下有期徒刑、拘役或 100 萬元以下罰金；若致人於死者，處無期徒刑或 7 年以上有期徒刑、致重傷者，處 3 年以上 10 年以下有期徒刑。

風傳媒，2019-05-07

第 6 章 程式參考解答

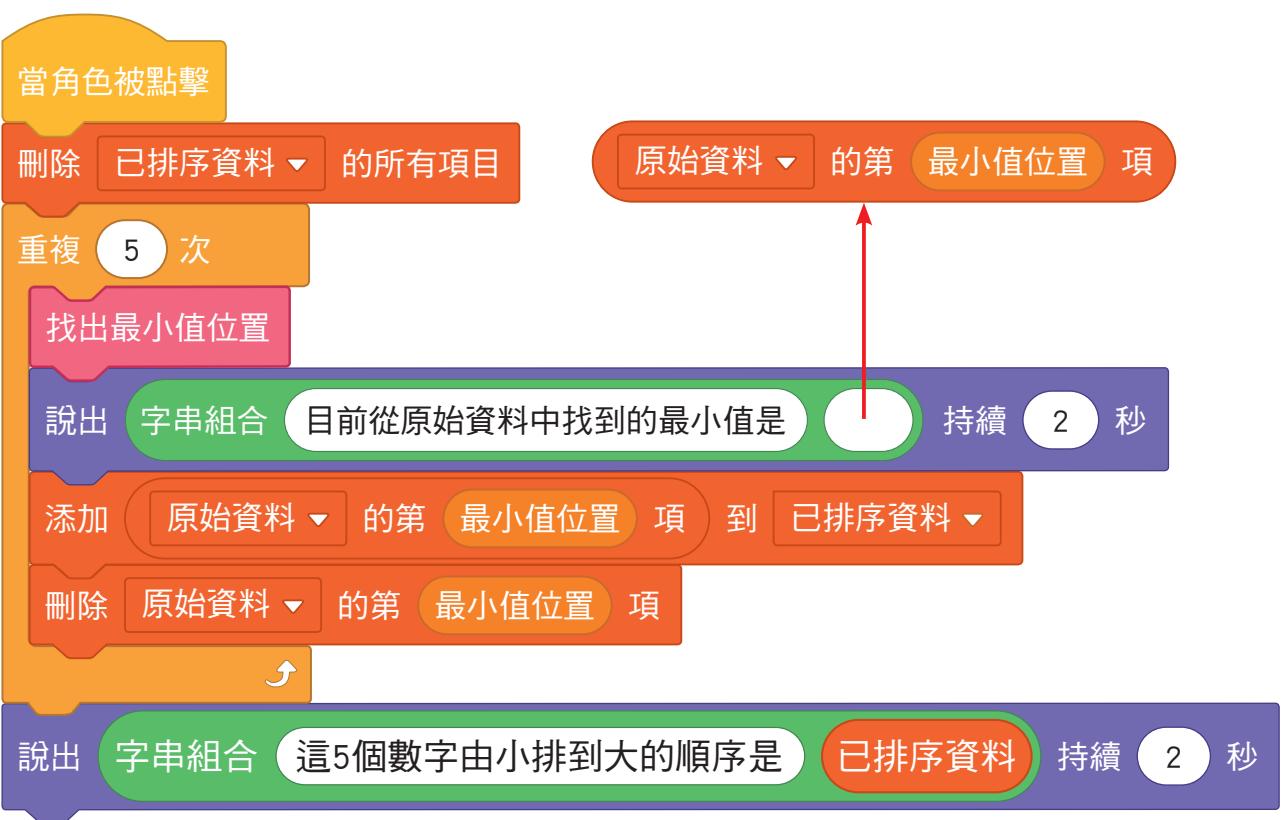


《選擇排序法》參考解答

對應課本第 173 頁

```
當 [被點擊] 時
    刪除 [原始資料 v] 的所有項目
    重複 (5) 次
        添加 (隨機取數 (1) 到 (10) 到 [原始資料 v])
```

```
定義 [找出最小值位置]
    變數 [資料位置] 設為 (1)
    變數 [最小值位置] 設為 (1)
    重複 ((清單 [原始資料 v] 的長度) 次
        如果 ((原始資料 v) 的第 (資料位置) 項 < (原始資料 v) 的第 (最小值位置) 項) 那麼
            變數 [最小值位置] 設為 (資料位置)
        變數 [資料位置] 改變 (-1)
```





《插入排序法》參考解答

對應課本第 183 頁

```

當 指定旗幟 被點擊
    刪除 原始資料 的所有項目
    重複 (5) 次
        添加 隨機取數 (1) 到 (10) 到 原始資料
    結束重複

```

```

當 角色被點擊
    刪除 已排序資料 的所有項目
    重複 (5) 次
        說出 字串組合 (目前從原始資料取出的數字是) (原始資料 的第 (1) 項) 持續 (2) 秒
        變數 插入位置 設為 (1)
        重複直到 (或)
            變數 插入位置 改變 (1)
            插入 (原始資料 的第 (1) 項) 到 已排序資料 的第 (插入位置) 項
            刪除 (原始資料 的第 (1) 項)
        結束重複
        說出 字串組合 (這5個數字由小排到大的順序是) (已排序資料) 持續 (2) 秒

```



《循序搜尋法》參考解答

對應課本第 195 頁

```

當 按鈕 被點擊
    刪除 原始資料 的所有項目
    重複 (50 次)
        添加 [隨機取數 (1 到 100) 到 原始資料]
    詢問 [請輸入要搜尋的數字 (1-100) :] 並等待
    變數 位置 設為 1
    重複直到 [位置 > 50 或 詢問的答案 = 原始資料 的第 位置 項]
        說出 [字串組合 目前比對的數字是 原始資料 的第 位置 項] 持續 0.5 秒
        變數 位置 改變 (1)
    如果 [位置 > 50 那麼]
        說出 [沒有符合的數字] 持續 2 秒
    否則
        說出 [字串組合 找到了，位於第 位置 個數字] 持續 2 秒

```



《二元搜尋法 1》參考解答

對應課本第 207 頁

```

當 被點擊
    變數 開始位置 設為 1
    變數 結束位置 設為 50
    變數 二分位置 設為 0
    詢問 請輸入要搜尋的數字 (1-100) : 並等待
    重複直到 詢問的答案 = 原始資料 的第 二分位置 項 或 二分位置 = 結束位置 - 1
        變數 二分位置 設為 無條件捨去 數值 開始位置 + 結束位置 / 2
        如果 原始資料 的第 二分位置 項 < 詢問的答案 那麼
            變數 開始位置 設為 二分位置
        否則
            如果 原始資料 的第 二分位置 項 > 詢問的答案 那麼
                變數 結束位置 設為 二分位置
            否則
                如果 詢問的答案 = 原始資料 的第 二分位置 項 那麼
                    說出 字串組合 找到了，位於第 字串組合 二分位置 個數字 持續 2 秒
                否則
                    說出 沒有符合的數字 持續 2 秒
    結束重複

```



《二元搜尋法 2》參考解答

對應課本第 217 頁

```

當 🏁 被點擊
    刪除 原始資料 ▾ 的所有項目
    刪除 已排序資料 ▾ 的所有項目
    重複 (50 次)
        添加 隨機取數 (1 到 100) 到 原始資料 ▾
    選擇排序法

```

```

定義 選擇排序法
    重複 (50 次)
        找出最小值位置
        添加 原始資料 ▾ 的第 最小值位置 項 到 已排序資料 ▾
        刪除 原始資料 ▾ 的第 最小值位置 項

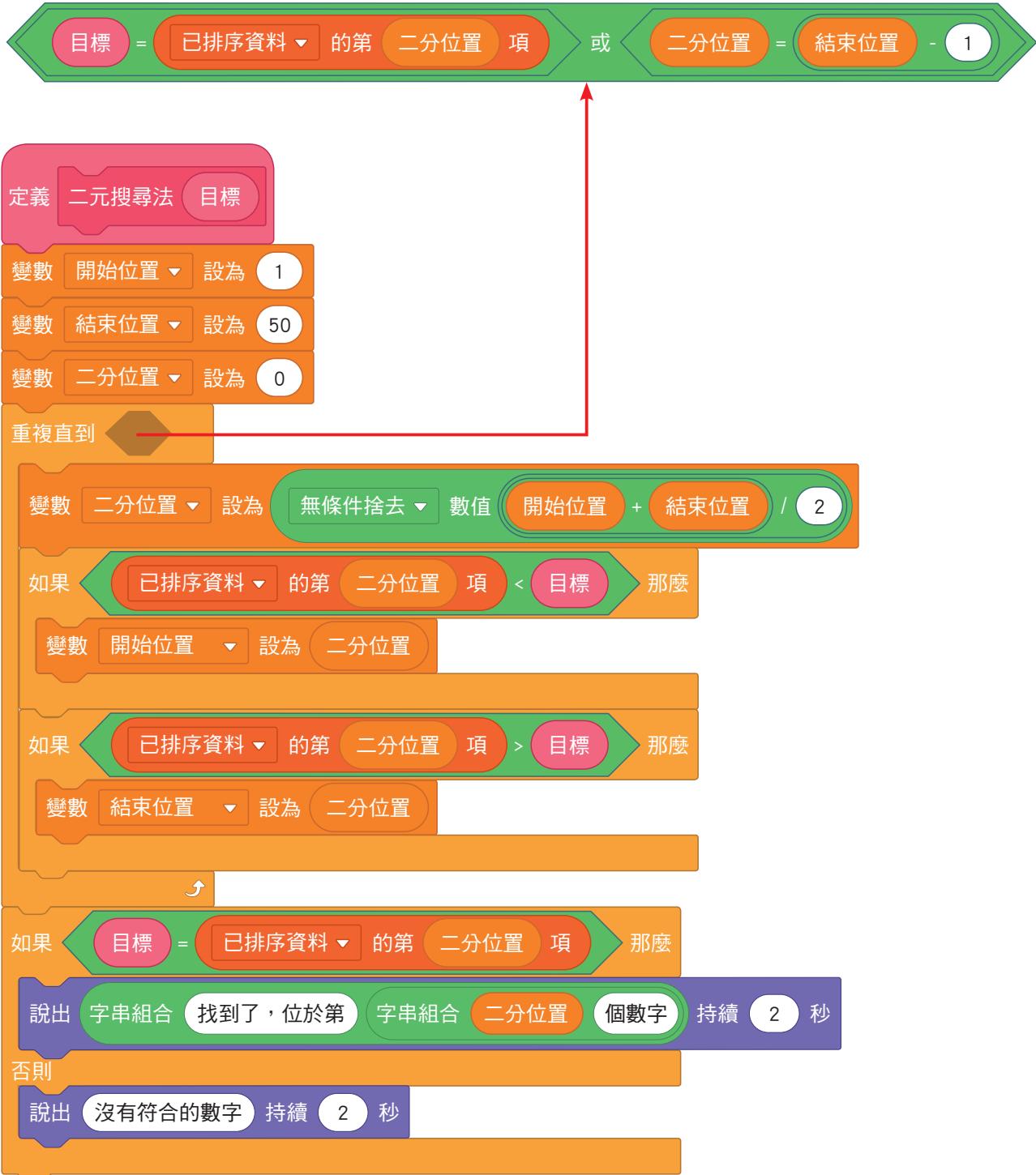
```

定義 找出最小值位置

變數 資料位置 ▼ 設為 1
變數 最小值位置 ▼ 設為 1
重複 清單 原始資料 ▼ 的長度 次
 如果 原始資料 ▼ 的第 資料位置 項 < 原始資料 ▼ 的第 最小值位置 項 那麼
 變數 最小值位置 ▼ 設為 資料位置
 變數 資料位置 ▼ 改變 1

當角色被點擊

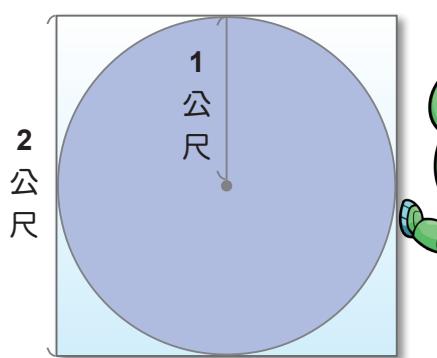
詢問 請輸入要搜尋的數字 (1-100) : 並等待
二元搜尋法 詢問的答案



●第6章 補充演算法—蒙地卡羅法●

還記得小學六年級數學教過的圓周率嗎？雖然這個問題早就已經有人解決了，但是如果能夠發明不同的方法來解決問題也是一種創新。在 1940 年代中期電腦發明了之後，電腦科學家就發現可以利用電腦產生的亂數來解決某些計算的問題，其中一個就是蒙地卡羅法。這個演算法的應用很廣，它也是近代 Google 公司發展的 AlphaGo 電腦圍棋背後的演算法之一。以下將介紹如何利用蒙地卡羅法來求取圓周率的近似值。

假設有一個半徑 1 公尺的圓形，在它的外圍畫一個正方形，這個正方形的邊長會是 2 公尺。如果我們在正方形中鋪滿豆子，有些豆子會在圓外，有些豆子會在圓內。而我們計算在圓內的豆子數，與全部在正方形中的所有豆子數，可以得到一個比例，這個比例會近似於圓形的面積與正方形的面積比。



還記得面積公式嗎？
圓形面積 = 半徑 × 半徑 × 圓周率
正方形面積 = 邊長 × 邊長

接下來，讓我們來計算兩個圖形的面積比吧！

面積

半徑 1 公尺的圓形面積 = $1 \times 1 \times \pi$ = 圓周率

邊長 2 公尺的正方形面積 = $2 \times 2 = 4$

面積比

正方形面積 : 圓形面積 = 4 : 圓周率

這個比例應該會近似於全部豆子數與圓形內豆子數的比例，如果我們可以數出全部豆子數與圓內豆子數，我們就可以算出圓周率的近似值。而且豆子愈小顆，樣本數愈多，算出來的圓周率近似值就會更逼近圓周率的理論值。這樣一顆一顆數豆子是很辛苦的，我們可以運用電腦的亂數來模擬。

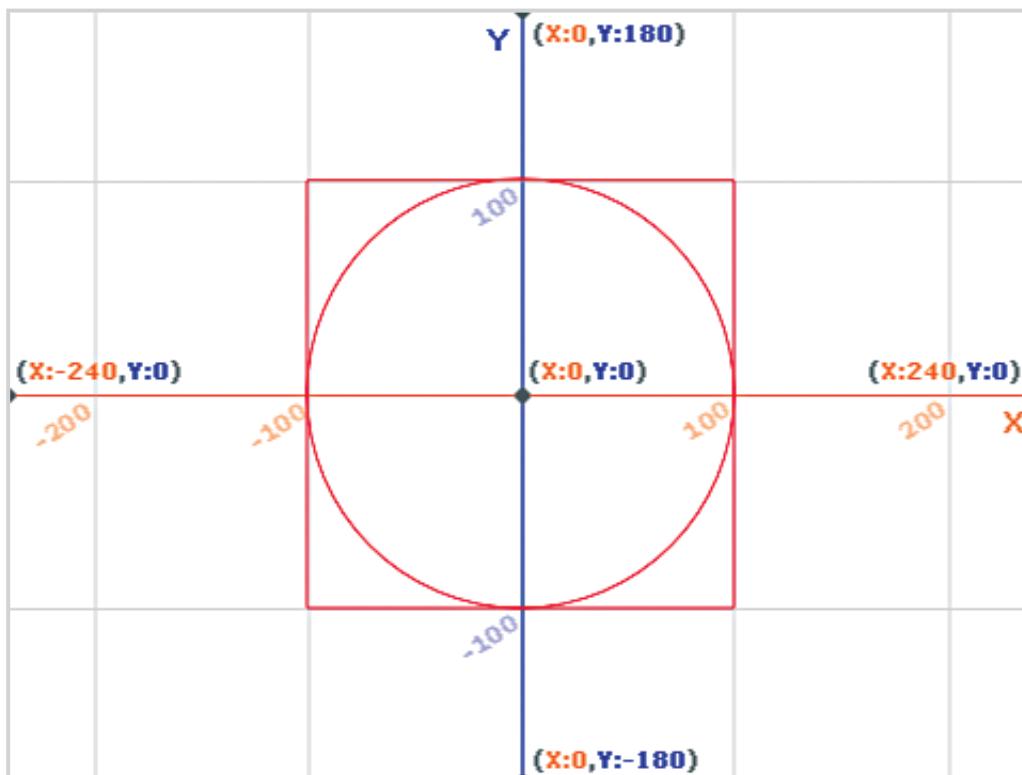


我們將圓形與正方形放到直角坐標平面上，來討論問題，而圓形的直徑與正方形的邊長需相同，並將圓形放至正方形內。假設圓的半徑是 100 單位，正方形的邊長是 200 單位，圓心的位置在原點 $(0, 0)$ 。接著，我們隨機產生 1 個點（即代表豆子），坐標位置是 (x, y) ，為了確保這個點會落在正方形之內，所以 x 坐標的範圍是 $-100 < x < 100$ ，而 y 的範圍是 $-100 < y < 100$ 。

我們如何判斷這個點有沒有落在圓內呢？

1. 如果這個點和圓心的距離小於圓的半徑，就會落在圓內。
2. 如果這個點和圓心的距離等於半徑，就會落在圓上。
3. 如果這個點和圓心的距離大於半徑，就會落在圓外。

根據畢氏定理找出直角坐標上某個點與原點的距離公式是 $x^2 + y^2 = \text{距離}^2$ ，從這個公式我們可以進一步推導出如果 $x^2 + y^2 < \text{半徑}^2$ ，就可以判斷這個點的坐標 (x, y) 是否落在圓內。

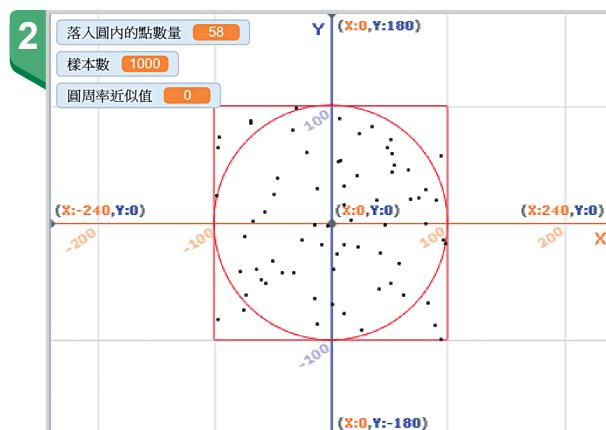


範例一蒙地卡羅法

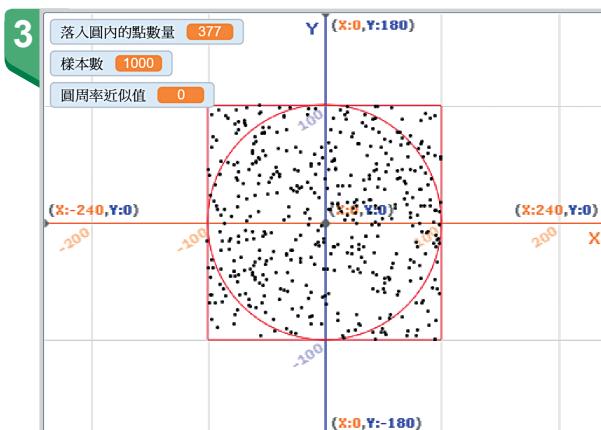
在畫面中，直角坐標上有一個半徑為 100 單位的圓形，放在邊長為 200 單位的正方形內當作背景，按下綠旗後，隨機在正方形內產生共 1000 個小點，並顯示變數包含落入圓內的點數量、樣本數和圓周率近似值。請執行《蒙地卡羅法》的程式，想一想這個範例的程式是如何運作？



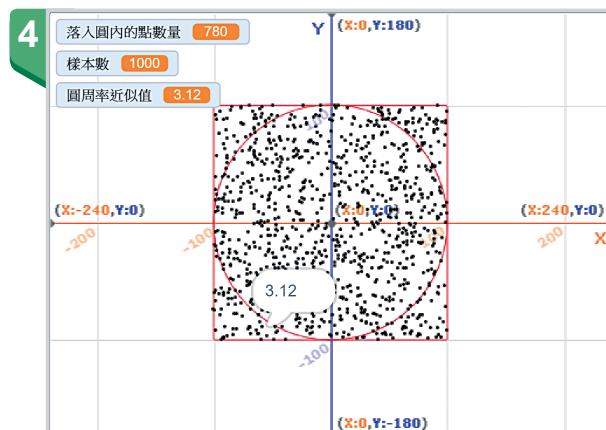
範例執行前



範例執行後，隨機在正方形中出現小點，並計算落入圓內的點數量。



隨機在正方形中出現小點，直到出現 1000 個小點為止。



接著計算並說出圓周率近似值。

本範例參考程式如下：

