第6章

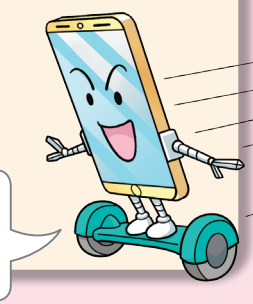
基本演算法的介紹

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。

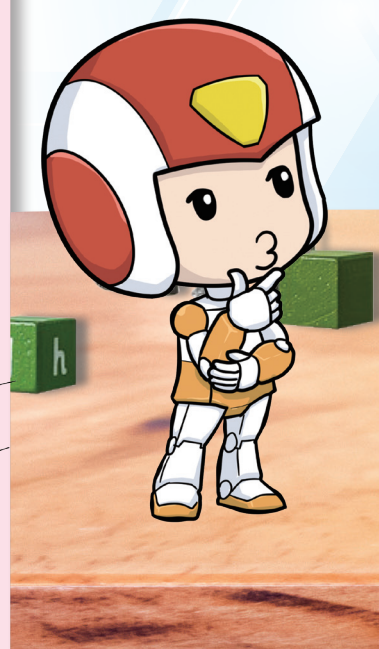
七年級時，已學過基礎程式設計，也大致了解演算法的基本概念。加上八年級學過的進階程式設計，我們可以更進一步來學習，電腦如何透過程式語言來完成常見的排序與搜尋問題。

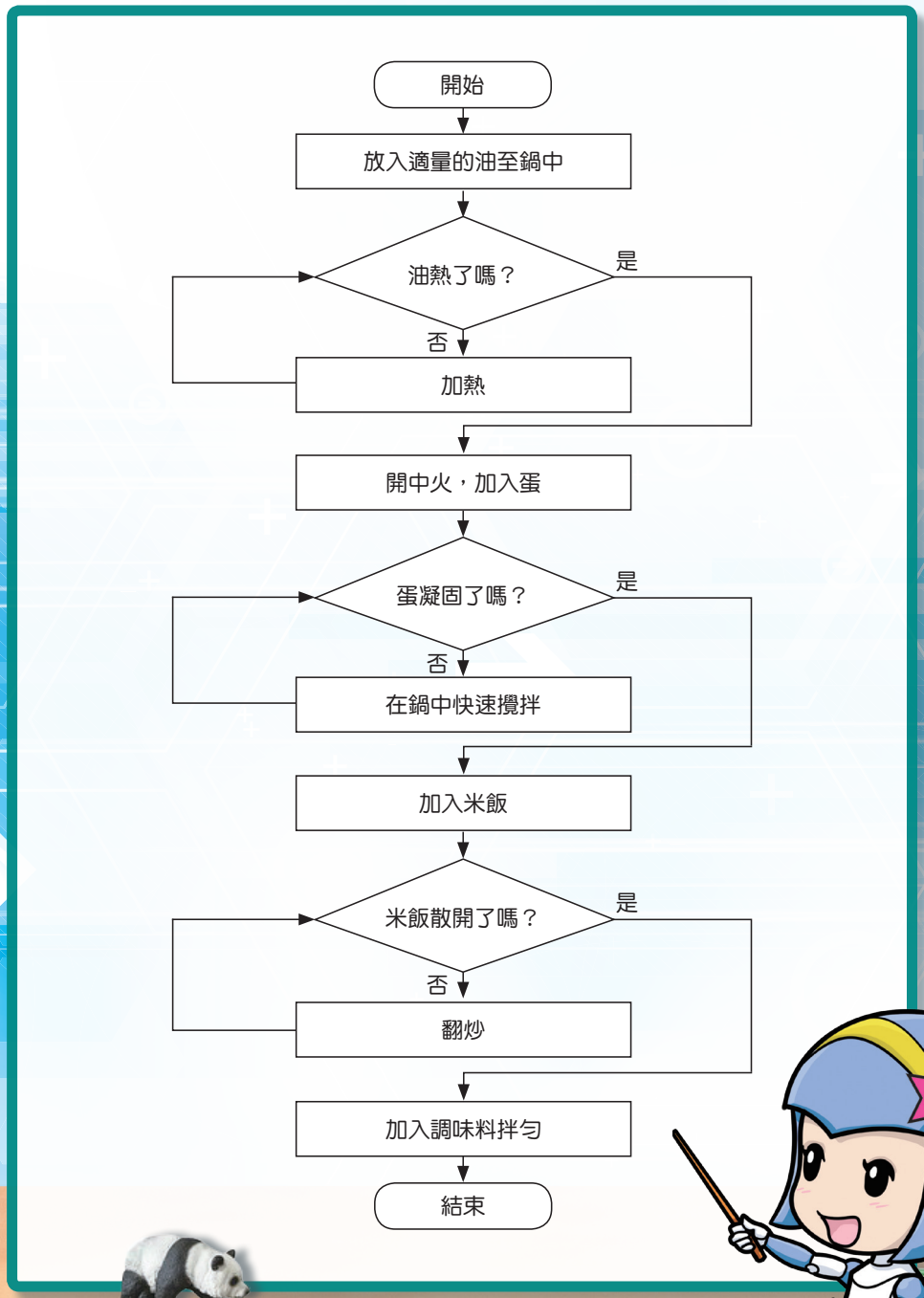
本章首先介紹演算法的基本原理及表示方式，接著用實例推導排序與搜尋問題，爾後再進入 Scratch 中來實作演算法，讓同學了解電腦解決問題的方式。希望這些方法能讓同學在日後遇到資料的排序或搜尋時，都可以運用程式設計實作演算法來解決問題。

哇，演算法的功用這麼厲害，就讓我們趕快來認識它吧！



補充資源手冊提供程式參考解答，輔助學習更快上手！另外也有補充範例，快來挑戰吧！



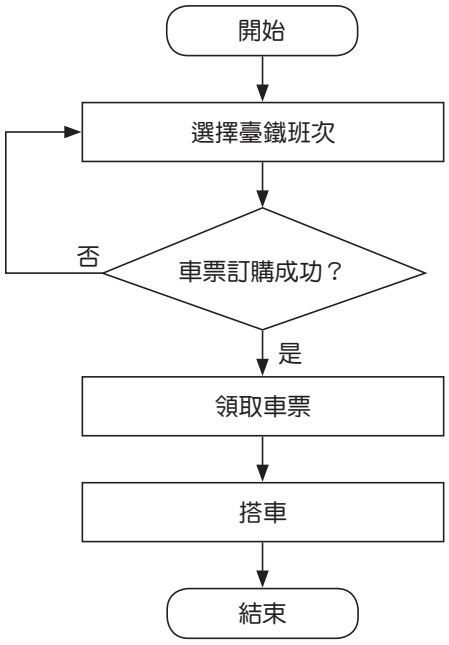

開始
放入適量的油至鍋中
油熱了嗎？
是
否
加熱
開中火，加入蛋
蛋凝固了嗎？
是
否
在鍋中快速攪拌
加入米飯
米飯散開了嗎？
是
否
翻炒
加入調味料拌勻
結束

6-1 演算法概念與原理

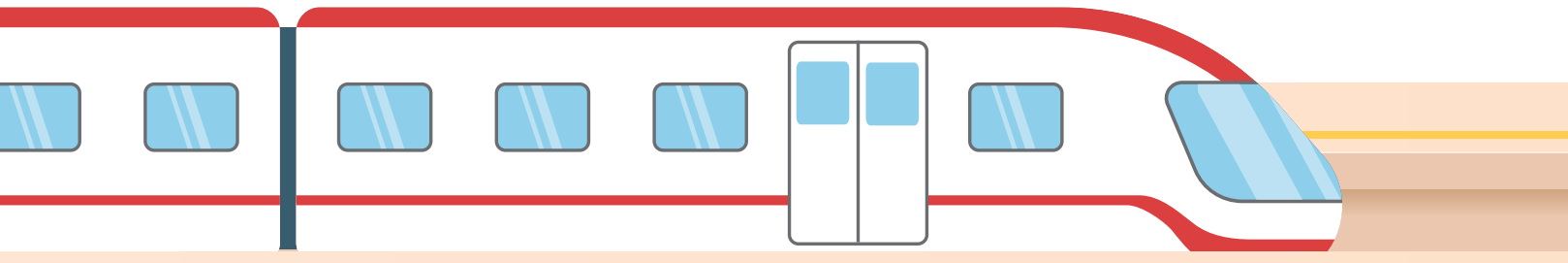
什麼是演算法（algorithm）？簡單的說就是解決問題的方法。在日常生活中，演算法時時刻刻在幫我們解決問題。例如：在七年級第 2 章曾提到蛋炒飯食譜，即可視為是一種演算法，幫我們把製作蛋炒飯這個程序化為可操作的步驟。在資訊科技領域中，演算法是一個可以交由電腦進行計算的具體步驟，它是一組有限運算規則的集合，包含問題精確的輸入、處理、輸出等。

演算法可以利用文字敘述、流程圖或其他方式表示。像是日常生活中搭火車即是一種用文字敘述表示的演算法（圖 6-1），也可使用流程圖（圖 6-2）來呈現。

▲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

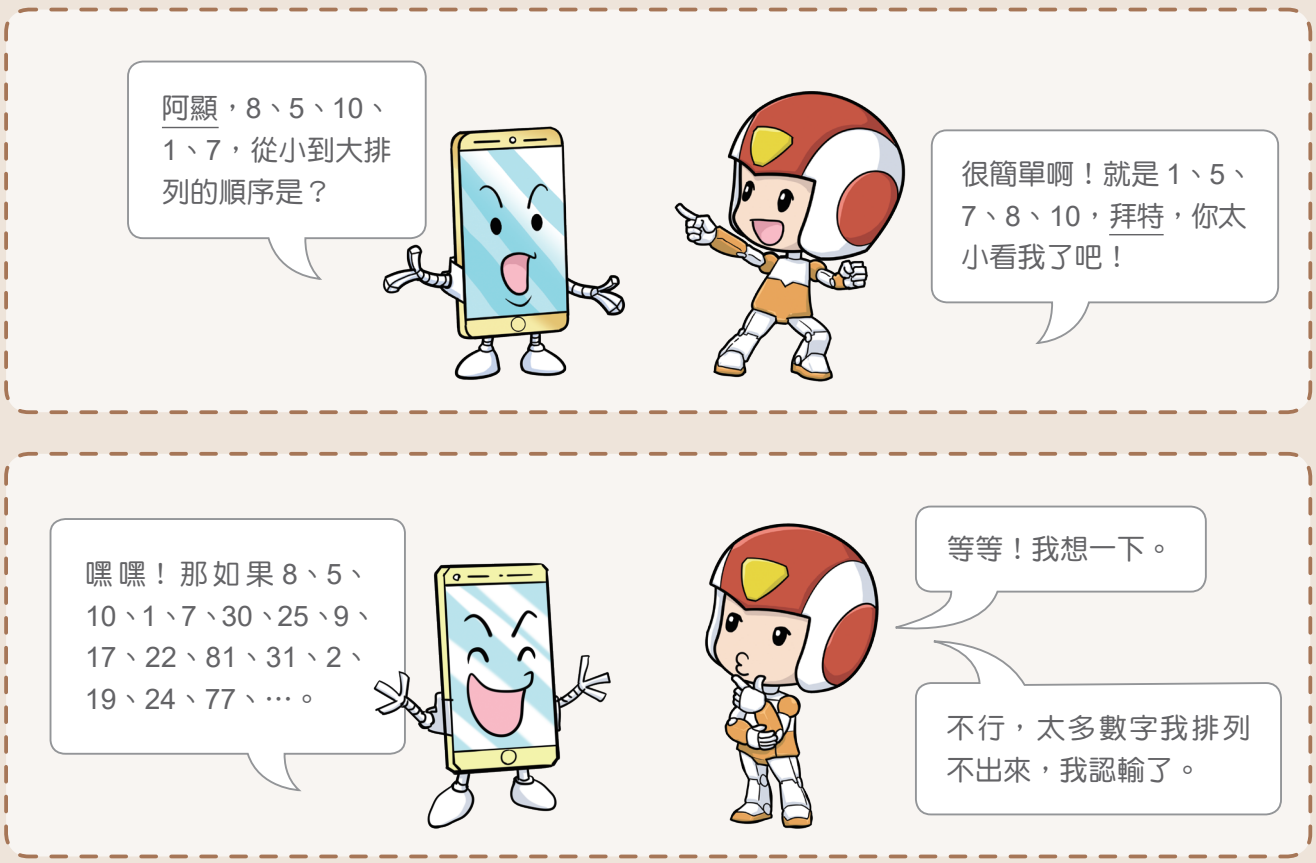
將資料依順序排列是我們經常需要處理的問題，例如：日常生活中，老師在班級裡依照同學的身高安排前後的座位，或是計算各式各樣的排行榜等。接下來，讓我們藉由生活案例來了解，日常中會如何使用到排序演算法的概念。

生活案例

華森從來沒玩過撲克牌，所以他拜班上的麗娜為師。而撲克牌基本功就是要先學會整理牌，因此麗娜教了兩種理牌方式，將撲克牌由小排到大，於是華森便開啟他的撲克牌練習之路，先找出適合自己的理牌方式，再學習各種撲克牌遊戲。

我們在七年級第 3 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？本節將進一步介紹選擇排序法（selection sort）與插入排序法（insertion sort）這兩種不同的演算法，為你揭開排序問題的神秘面紗，並可透過附件 4、5 的操作練習，加深排序演算法的概念。

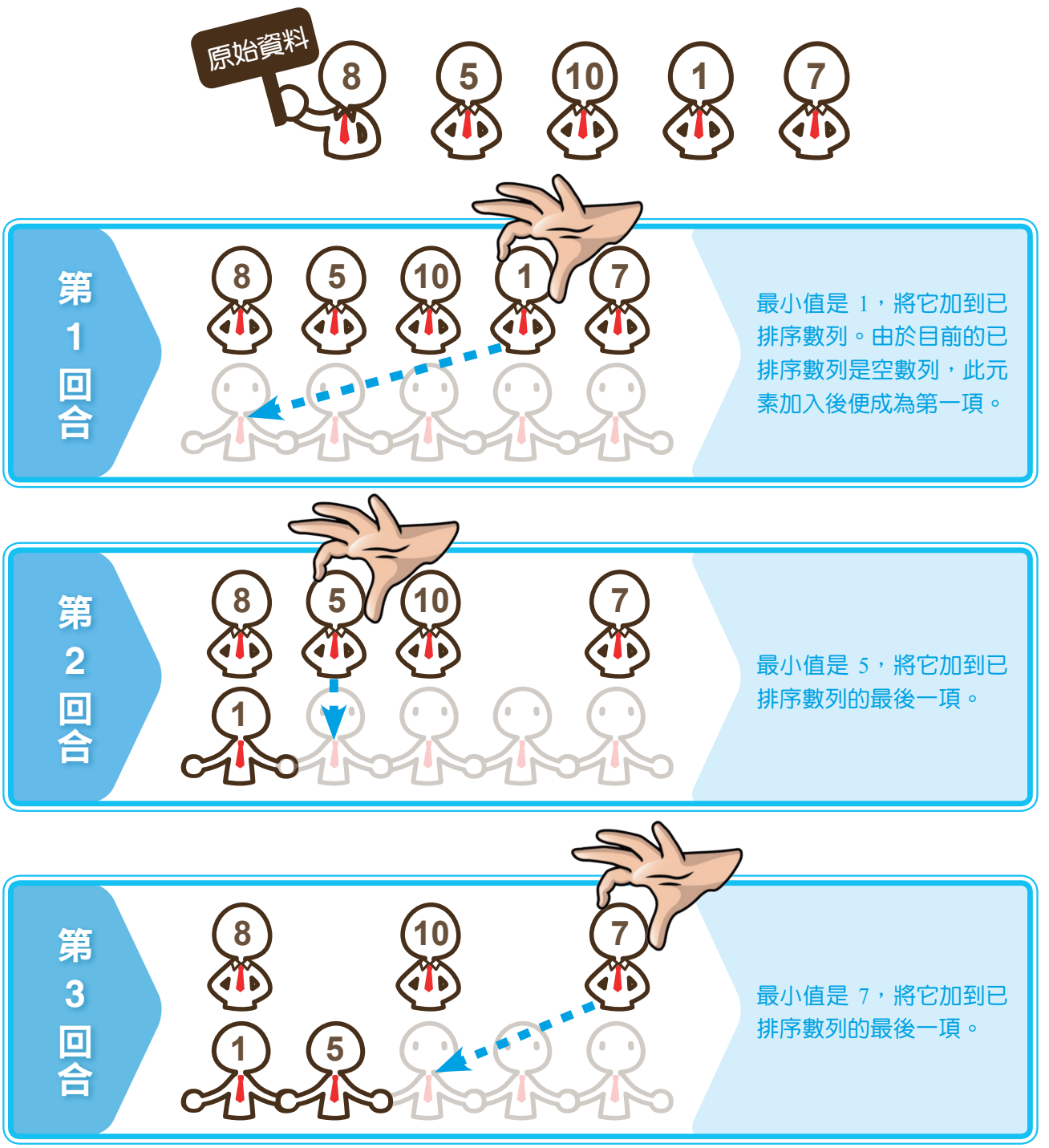
首先，我們將排序問題簡化成數字的排序。假設原始資料有：8、5、10、1、7，共 5 個隨機數字，經由從小到大排序後資料變成：1、5、7、8、10。因為資料只有 5 個數字，所以你能夠很容易就把這 5 個數字從小到大排列，但是如果資料量暴增到 100 筆，你就得要從解決問題的過程中找出規則（圖 6-4），將其發展成一套演算法，而這套演算法適用於各種大小的資料量。

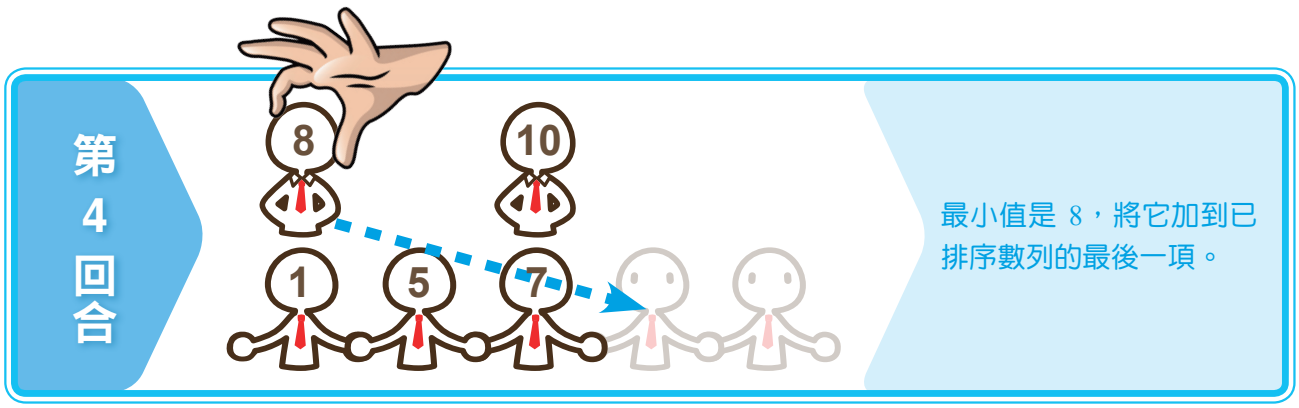


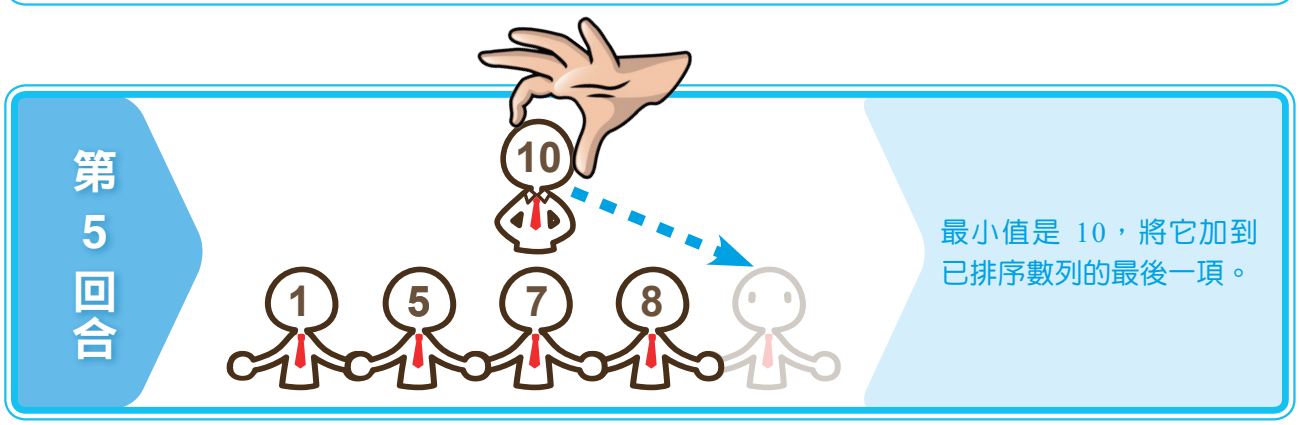
▲圖 6-4 排序不同的資料量。

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：





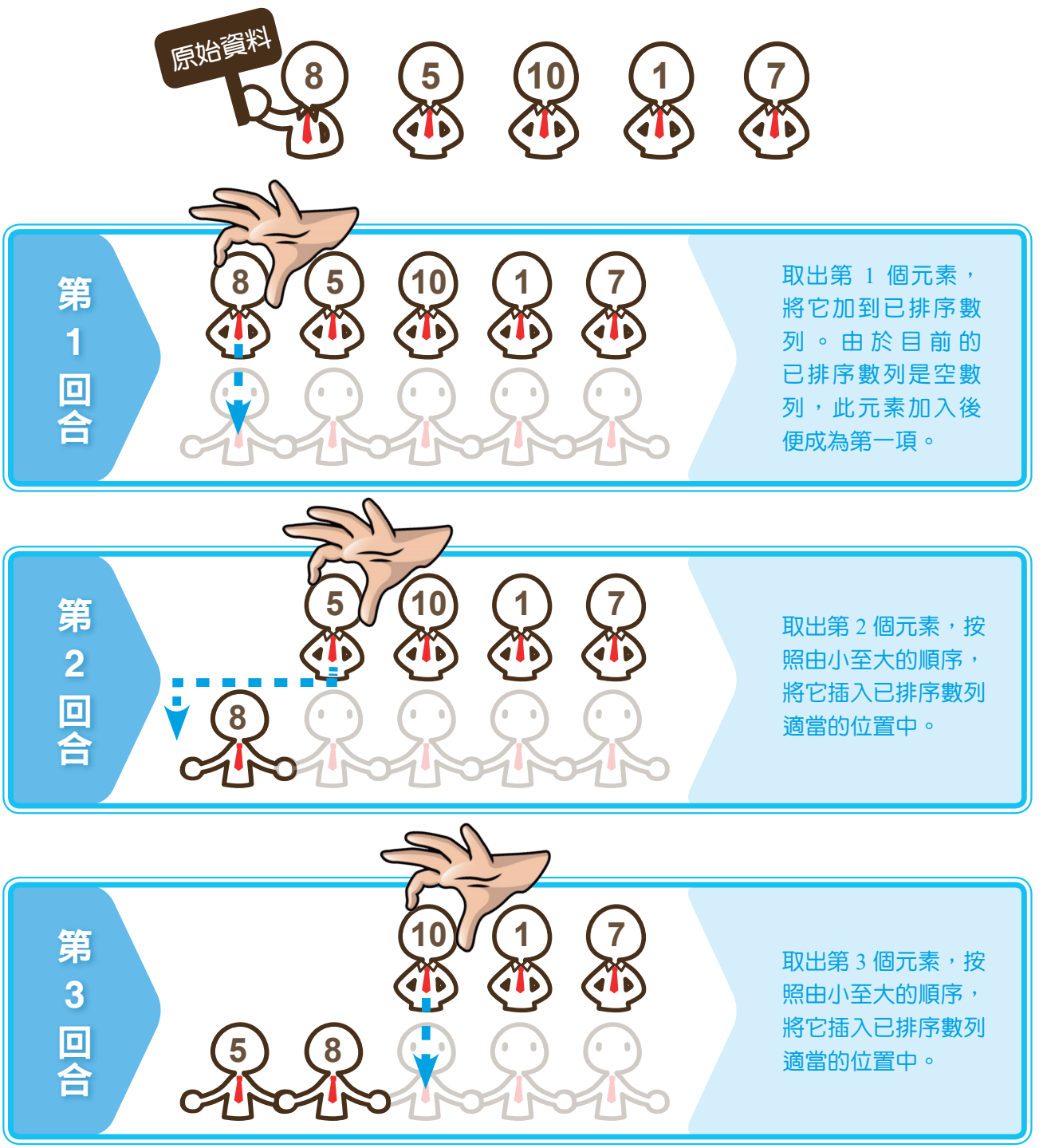


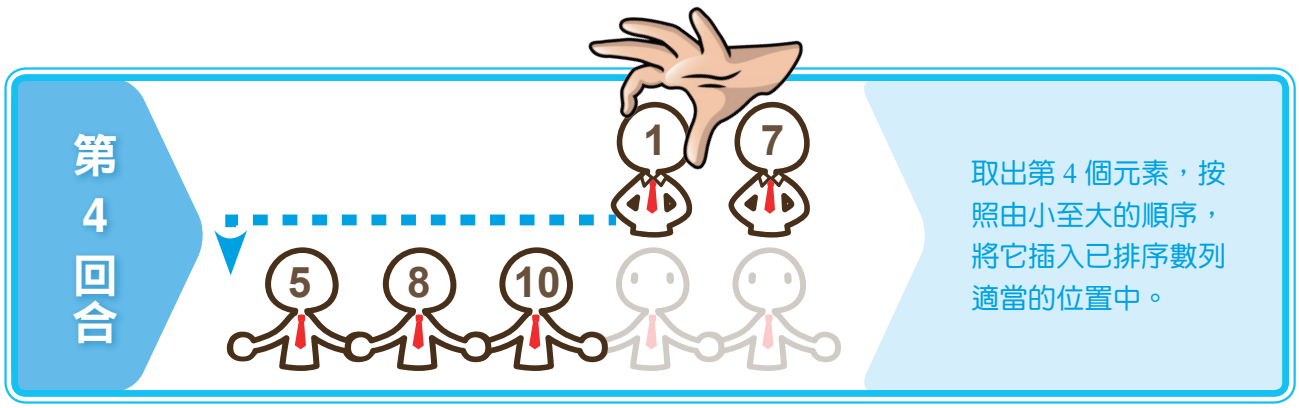
我們可以從以上的流程，歸納出實作的步驟

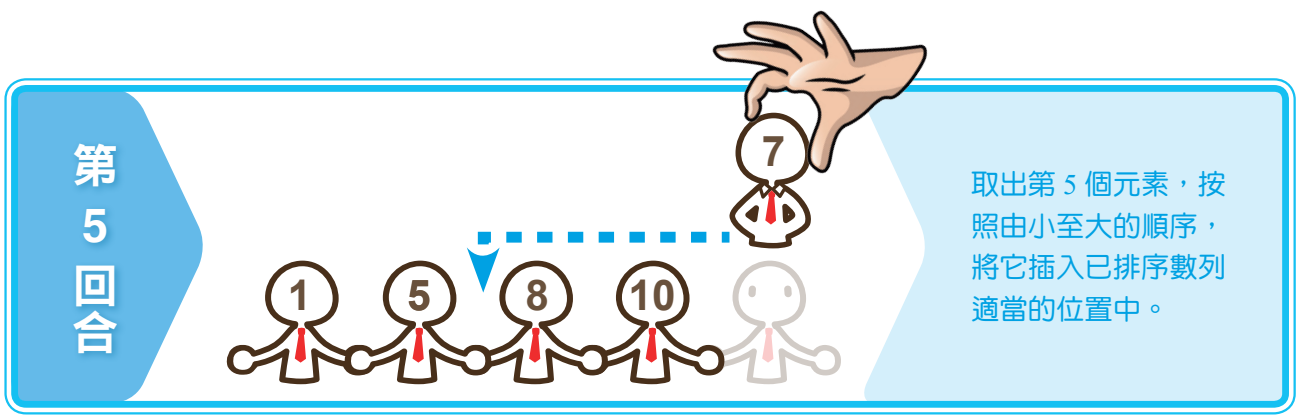
1. 先從未排序的原始資料中找到第一個最小的元素，將它加到已排序數列的第一項。
2. 接著從未排序的原始資料中找到最小的元素。
3. 將此元素加到已排序數列的最後一項。
4. 重複第 2、3 點的步驟，直到原始資料全部處理完成。

6-2-2 插入排序法

插入排序法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：

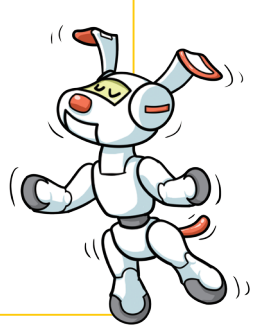






我們可以從以上的流程，歸納出實作的步驟

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。



6-2-3 排序法程式設計的應用

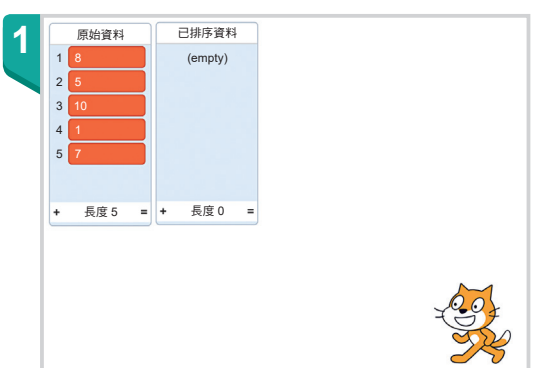
認識選擇排序法與插入排序法的原理後，在 Scratch 中是如何執行的呢？我們以選擇排序法做為範例，了解程式運作的邏輯與過程。

範例：選擇排序法

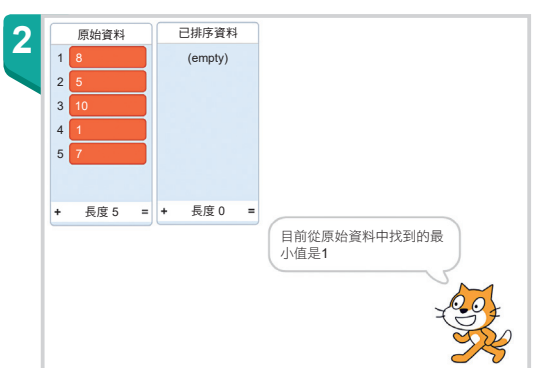
點擊綠旗後，小貓執行下列動作：

1. **建立從數字 1～10 中隨機取出 5 個數字的純文字檔，並將此檔案匯入原始資料裡。**
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. **最後完成由小到大的排列後，小貓會說出這 5 個數字的排列順序。**

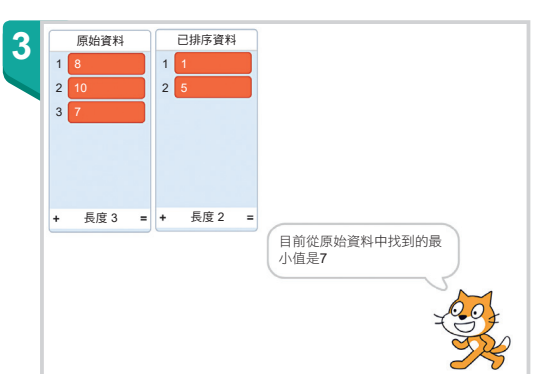
請執行《選擇排序法》的程式，想一想這個範例的程式是如何運作？



範例執行後，**原始資料**呈現匯入的未排序數列。



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。



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 如何將現有的未排序數列匯入原始資料裡？

1. 如何使用現有的未排序數列？
2. 如何表示原始資料？

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？

假設原始資料如下表，預設最小值為**第 1 項**，將原始資料的每一項數字依序與最小值進行比對，比對完即可找出最小值。

原始資料	
項次	數字
1	8
2	5
3	10
4	1
5	7

第一次

原始資料**第 1 項**　最小值為**第 1 項**

8 ＝ 8

預設最小值位置為**第 1 項**，數字為 **8**。

第二次

原始資料**第 2 項**　最小值為**第 1 項**

5 ＜ 8

最小值位置由**第 1 項**改為**第 2 項**，數字為 **5**。

第三次

原始資料**第 3 項**　最小值為**第 2 項**

10 ＞ 5

最小值位置維持為**第 2 項**，數字為 **5**。

第四次

原始資料**第 4 項**　最小值為**第 2 項**

1 ＜ 5

最小值位置由**第 2 項**改為**第 4 項**，數字為 **1**。

第五次

原始資料**第 5 項**　最小值為**第 4 項**

7 ＞ 1

最小值位置維持為**第 4 項**，數字為 **1**。

➡ 比對完原始資料的所有數字，最小值位置為**第 4 項**，數字為 **1**。

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？

找出每一個回合的最小值及其位置後，添加至已排序資料清單並刪除該數字。

第一回合

最小值位置為 **4**，數字為 **1**。

原始資料	
項次	數字
1	8
2	5
3	10
4	1
5	7

已排序資料	
項次	數字

刪除原始資料清單**第 4 項**

原始資料	
項次	數字
1	8
2	5
3	10
4	7

已排序資料	
項次	數字
1	1

第二回合

最小值位置為 **2**，數字為 **5**。

原始資料	
項次	數字
1	8
2	5
3	10
4	7

已排序資料	
項次	數字
1	1

刪除原始資料清單**第 2 項**

原始資料	
項次	數字
1	8
2	10
3	7

已排序資料	
項次	數字
1	1
2	5

⋮

第五回合

最小值位置為 **1**，數字為 **10**。

原始資料	
項次	數字
1	10

已排序資料	
項次	數字
1	1
2	5
3	7
4	8

刪除原始資料清單**第 1 項**

原始資料	
項次	數字

已排序資料	
項次	數字
1	1
2	5
3	7
4	8
5	10

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 1 **如何將現有的未排序數列匯入原始資料裡？**

想一想，這句話包含了什麼重要的訊息？

1. **現有的未排序數列：**

如何使用現有的未排序數列？

可從電腦中，匯入未排序數列的文字檔。

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**。

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。

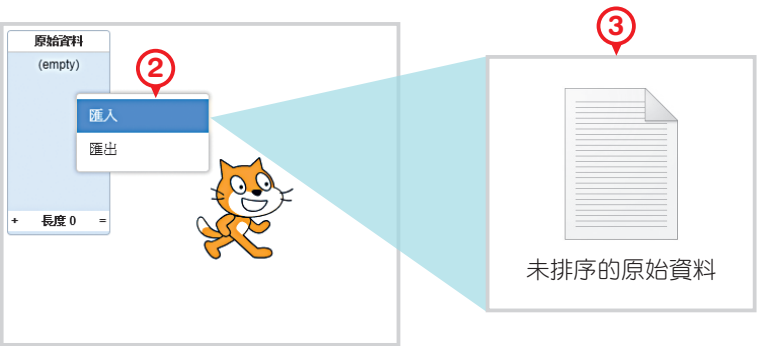
步驟 1 **設定清單。**

① 新增**清單**，命名為**原始資料**。

步驟 2 匯入資料。

② 在**原始資料**清單上，按右鍵選擇**匯入**。

③ 從電腦中，匯入**未排序的原始資料**。



問題拆解 2 **如何從未排序數列中找到最小的數字？**

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，依序比較所有數字後找出最小值。
3. 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。

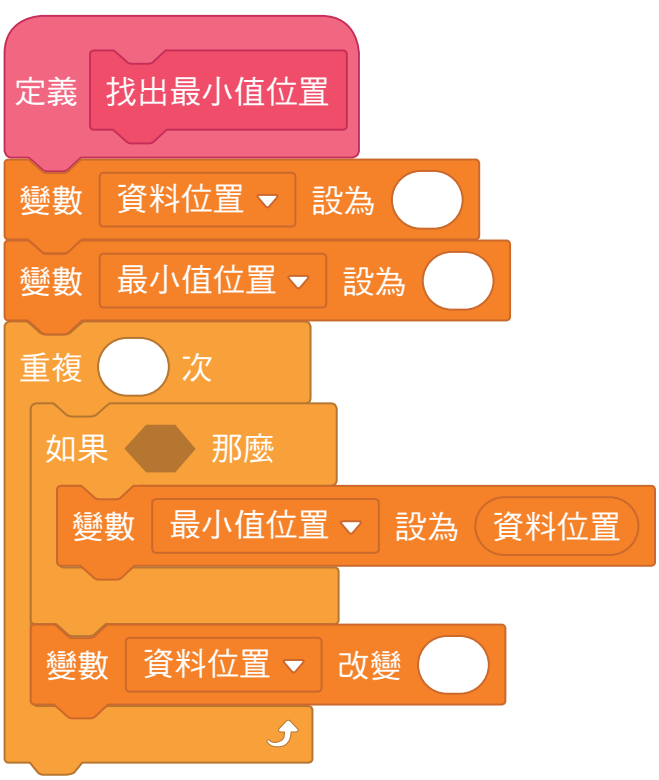
步驟 3 設定函式積木與變數。

④ 新增**函式積木**，命名為**找出最小值位置**。

⑤ 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 3

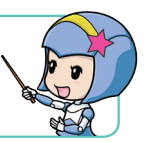
如何將此數字加到已排序數列的最後一項？

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？

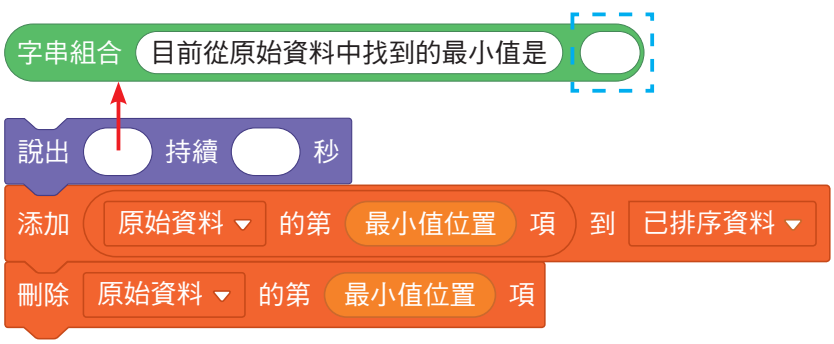
1. 把每一次從**原始資料**清單中找到的最小值， 添加到**已排序資料**清單中。
2. 因為最小值已加到**已排序資料**清單中，所以把剛剛從**原始資料**清單中找到的最小值刪除。
3. 第二次開始找到的最小值會比上一次找到的大，所以每次添加到**已排序資料**清單時，要添加到最後一項。

步驟 5 設定清單。

⑥ 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 4 重複執行上述的動作，直到未排序數列全部處理完成。

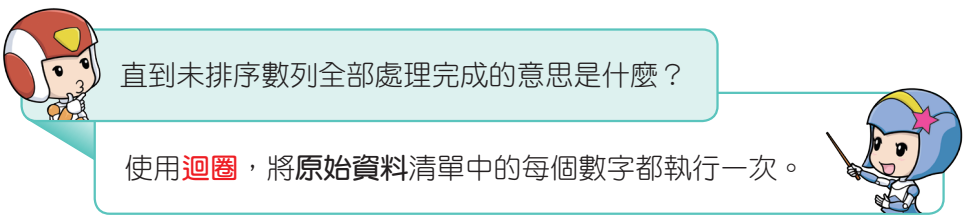
想一想，這句話包含了什麼重要的訊息？

1. **重複執行上述的動作：**

如何重複執行上述的動作？

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。

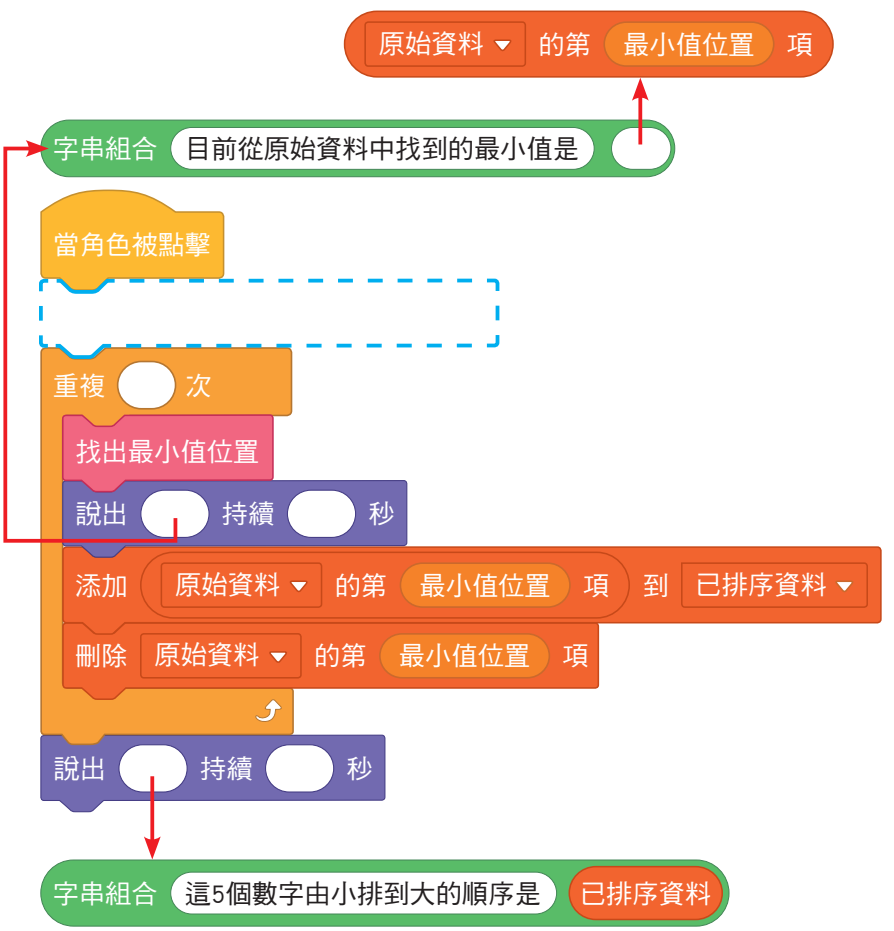
2. 直到未排序數列全部處理完成：



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，虛線框的積木是什麼？
3. 迴圈要執行幾次？
4. 什麼時候要執行**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…」？



6-3 搜尋的原理與範例

在大量資料中找尋設定的目標資料是很常見的情況，例如：日常生活中，在大賣場中找尋想買的用品，或是在排隊人龍中找尋朋友等。接下來，讓我們藉由生活案例來了解，日常中會如何使用到搜尋演算法的概念。

生活案例

志傑老師班上有 32 位同學，每年都會舉辦交換禮物，今年想讓學生自己選禮物，所以第一個選擇的人占有很大的優勢。因此老師請班長和學藝股長規畫如何選出第一位同學，他們各想了一個有趣的方式，最後再交由老師決定交換禮物的規則。

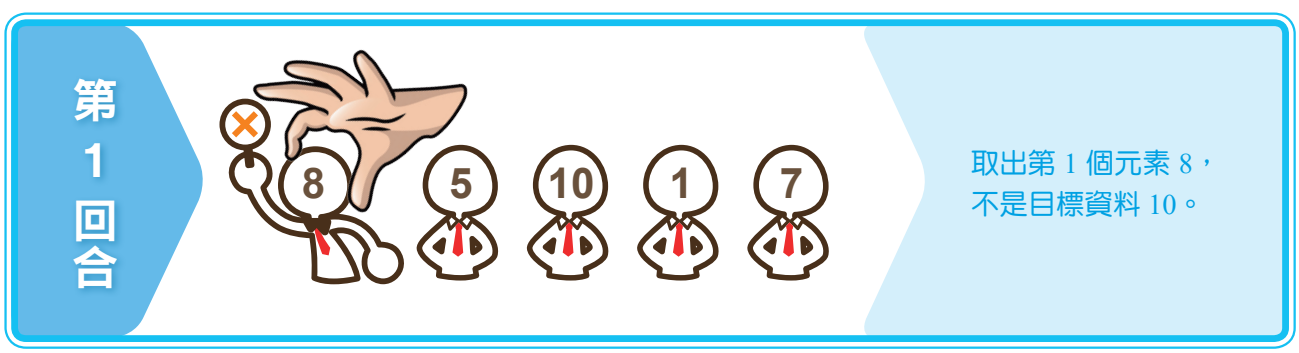
依照不同的資料，可以使用的搜尋方式也不同，又是如何辨別找到目標資料，以及找不到目標資料的情況呢？本節將進一步介紹循序搜尋法（sequential search）與二元搜尋法（binary search）這兩種不同的演算法，為你揭開搜尋問題的神祕面紗，並可透過附件 4、5 的操作練習，加深搜尋演算法的概念。

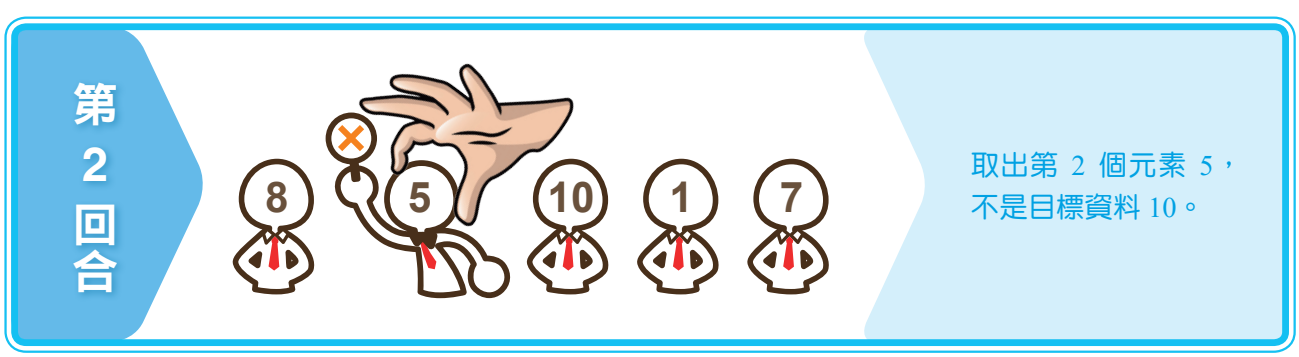
首先，我們將搜尋問題簡化為數字的搜尋。假設原始資料有：8、5、10、1、7，共 5 個數字，搜尋的目標資料是 10，因為原始資料只有 5 個數字，所以你能夠一眼就看到 10 在原始資料的第 3 筆，但是如果資料量暴增到 100 個，你就得要從解決問題的過程中找出規則（圖 6-5），將其發展成一套搜尋演算法，而這套演算法適用於各種大小的資料量與各種不同的目標資料。

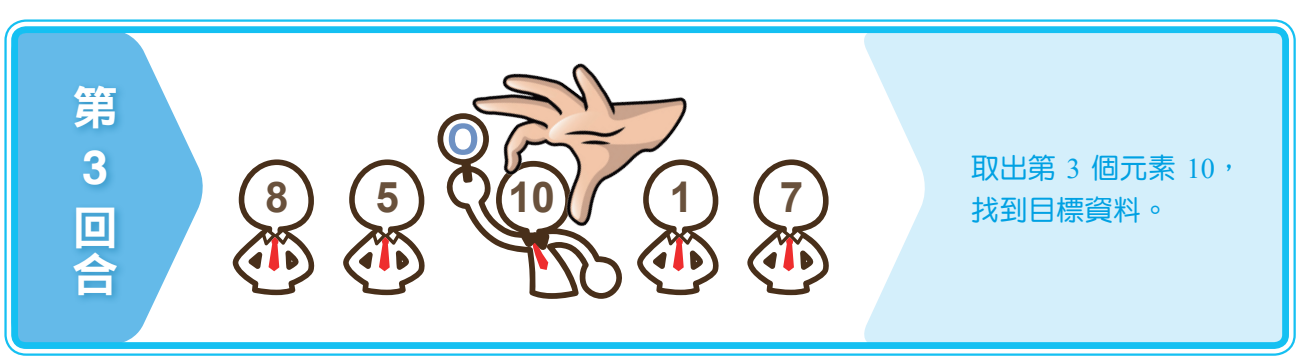
▲圖 6-5 在不同的資料量搜尋目標資料。

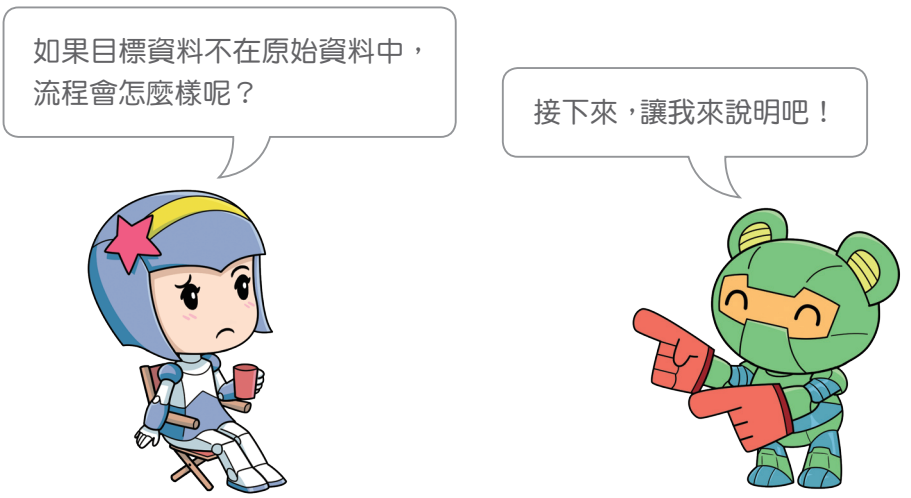
6-3-1 循序搜尋法

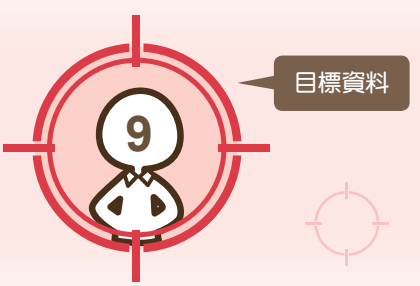
循序搜尋法是用來達成搜尋特定資料，它是從第一個元素開始取出，依序逐個與「目標資料」相互比較，直到找到所要的元素或所有資料均尋找完為止，舉例如下：

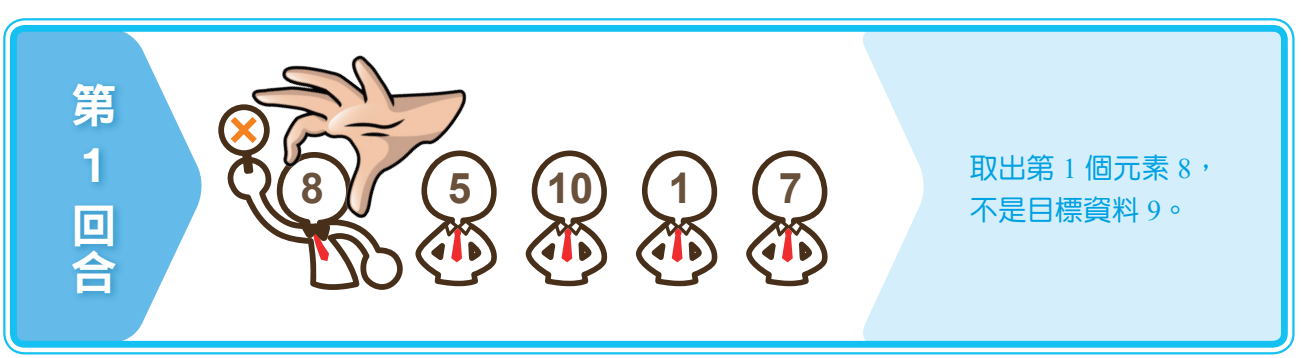


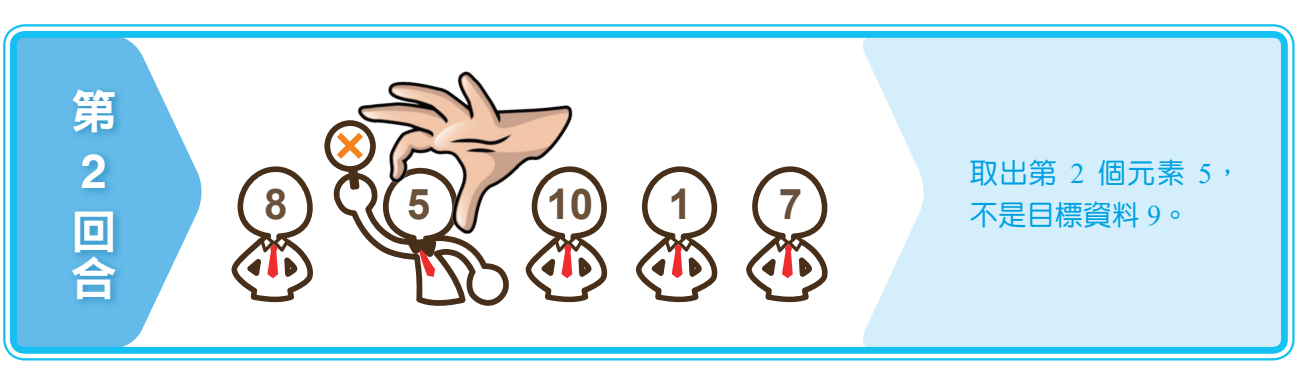




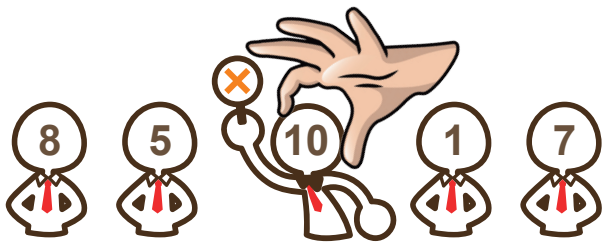

如果目標資料不在原始資料中，流程會怎麼樣呢？
接下來，讓我來說明吧！


9
目標資料


第 1 回合
8
5
10
1
7
取出第 1 個元素 8，不是目標資料 9。

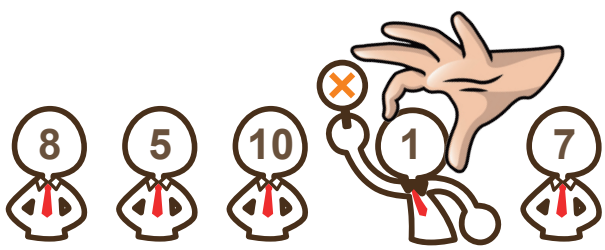

第 2 回合
8
5
10
1
7
取出第 2 個元素 5，不是目標資料 9。

第 3 回合



取出第 3 個元素 10，不是目標資料 9。

第 4 回合



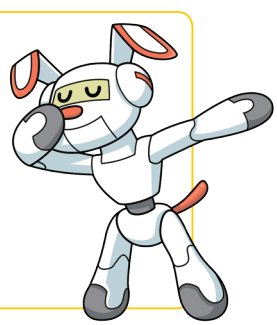
取出第 4 個元素 1，不是目標資料 9。

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

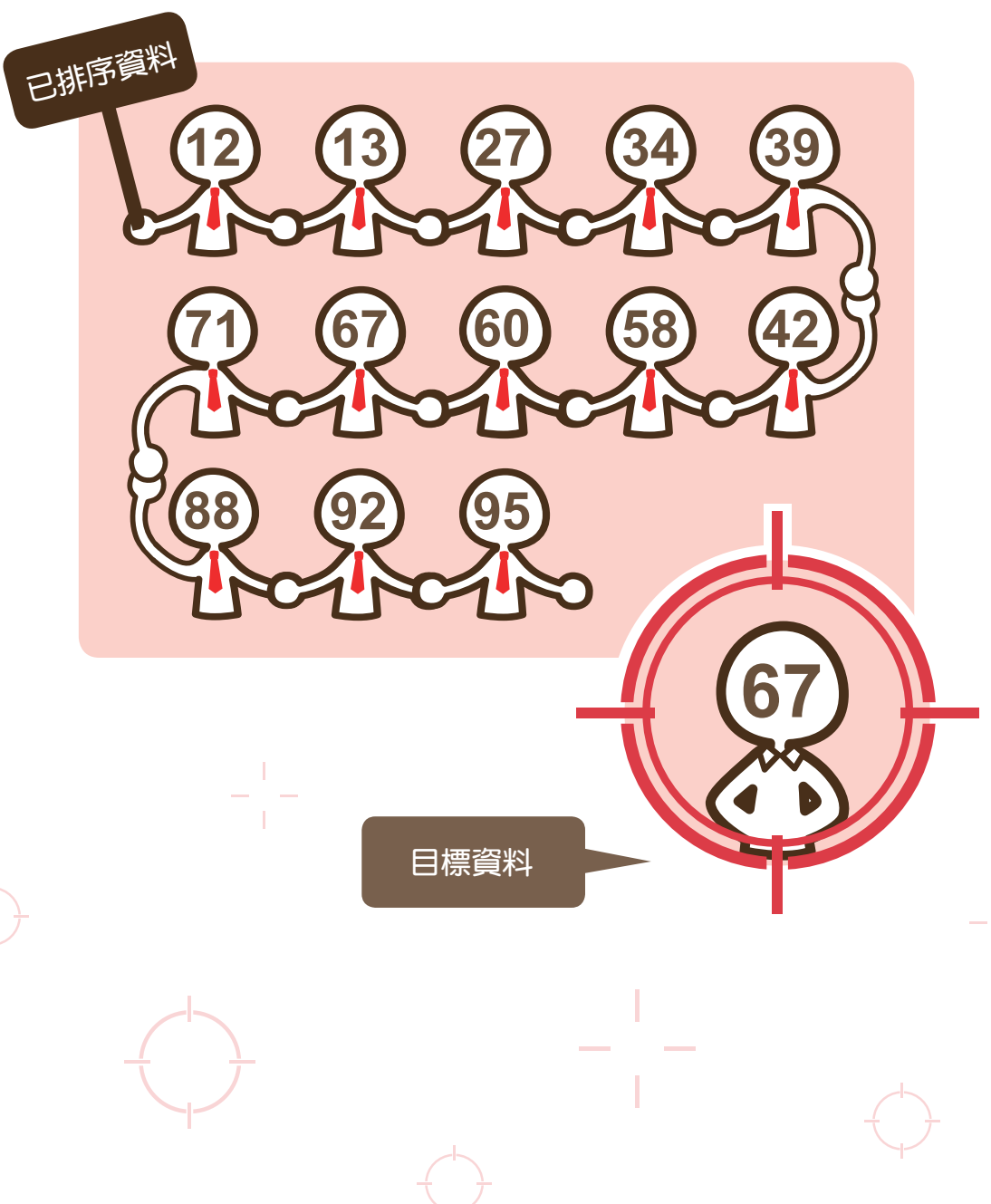
1. 從未排序的原始資料中逐一取出元素。
2. 將取出元素與目標資料加以比較。
3. 重複第 1、2 點的步驟，直到找到目標資料或原始資料所有元素均比較完為止。



6-3-2 二元搜尋法

二元搜尋法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。

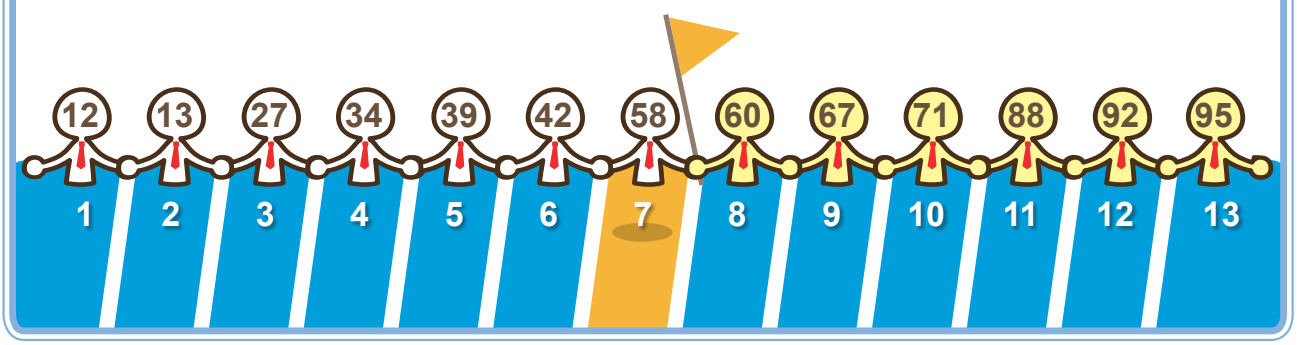
假設已排序的資料如下，在這 13 個數字中要搜尋數字 67 是否存在，若使用二元搜尋法，只需要找 4 次就可以找到，效率極高，舉例如下：



第 1 回合

開始位置：1
結束位置：13
二分位置：(1＋13)÷2＝7

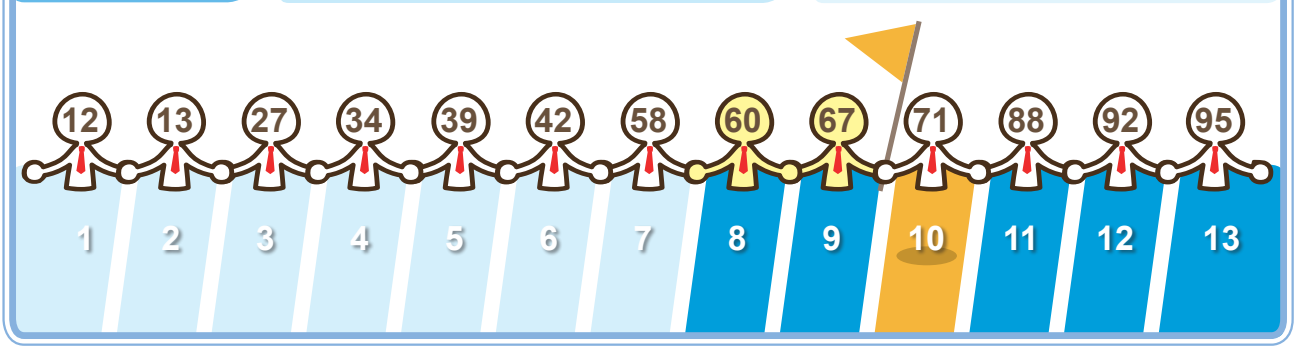
第 7 個元素：58
58＜67 取後（右）半部



第 2 回合

開始位置：8
結束位置：13
二分位置：(8＋13)÷2＝10.5
（取 10.5 的整數部分 10）

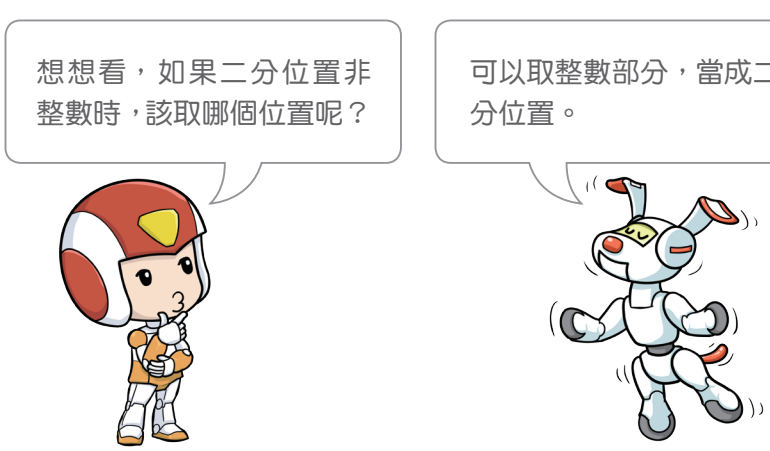
第 10 個元素：71
71＞67 取前（左）半部



小提示

二分位置

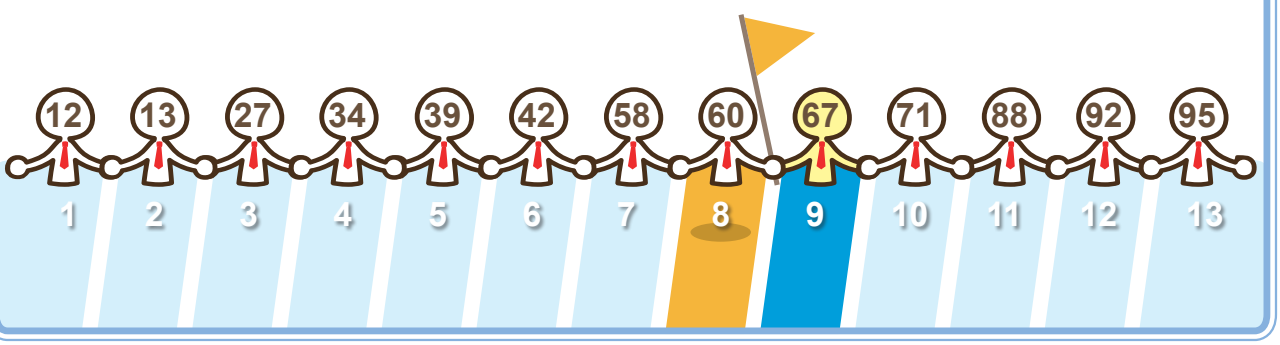
將資料的數值由小至大排序後，將資料切分成 2 等分，切分的位置（二分位置）即為整個數列的中間位置。



第 3 回合

開始位置：8
結束位置：9
二分位置：(8+9)÷2=8.5
（取 8.5 的整數部分 8）

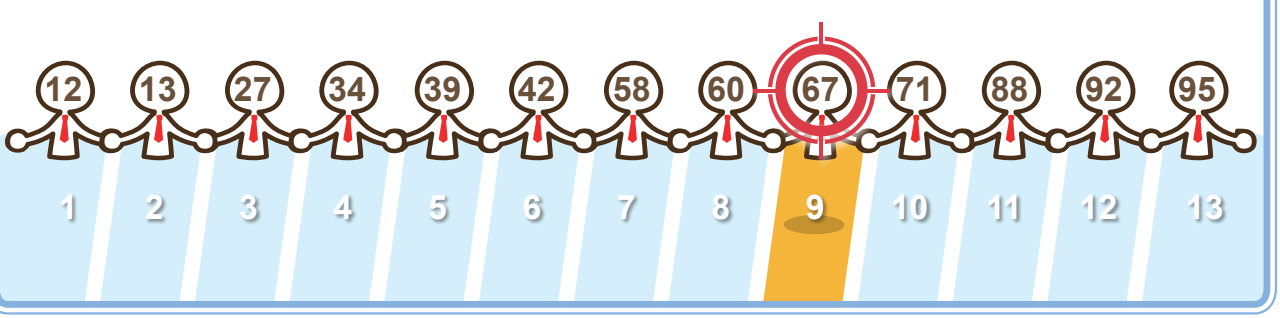
第 8 個元素：60
60＜67 取後（右）半部



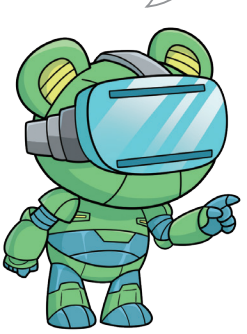
第 4 回合

開始位置：9
結束位置：9
二分位置：(9+9)÷2=9

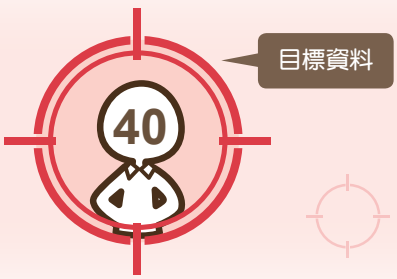
第 9 個元素：67
67＝67 找到目標資料



咦！那找不到目標資料時，流程是如何進行呢？



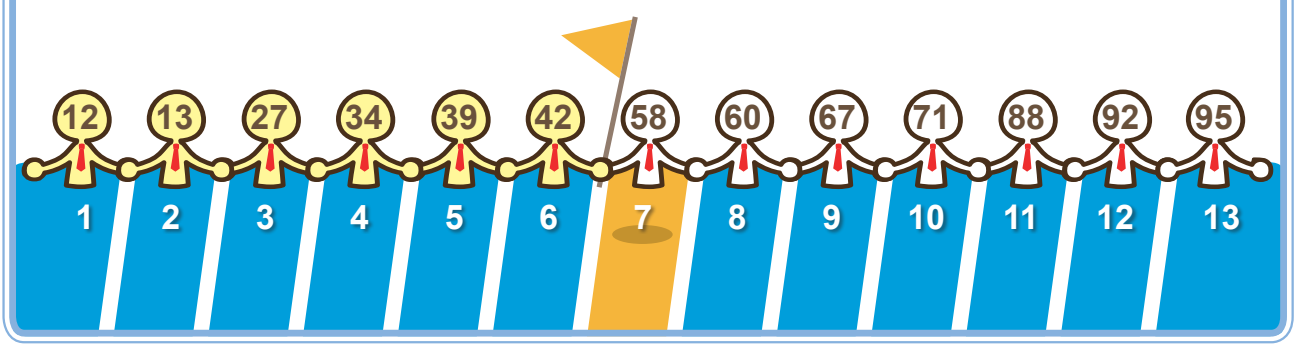
讓我們繼續看下去吧！



第 1 回合

開始位置：1
結束位置：13
二分位置：(1＋13)÷2＝7

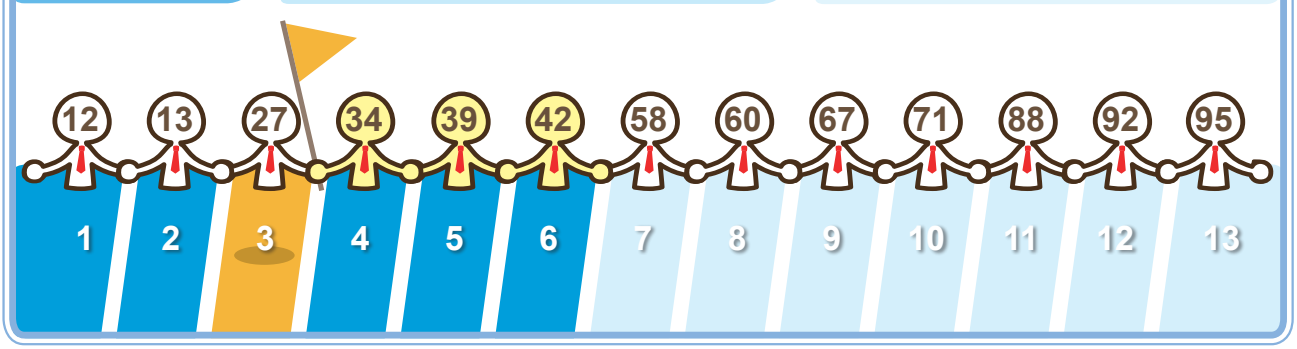
第 7 個元素：58
58＞40 取前（左）半部

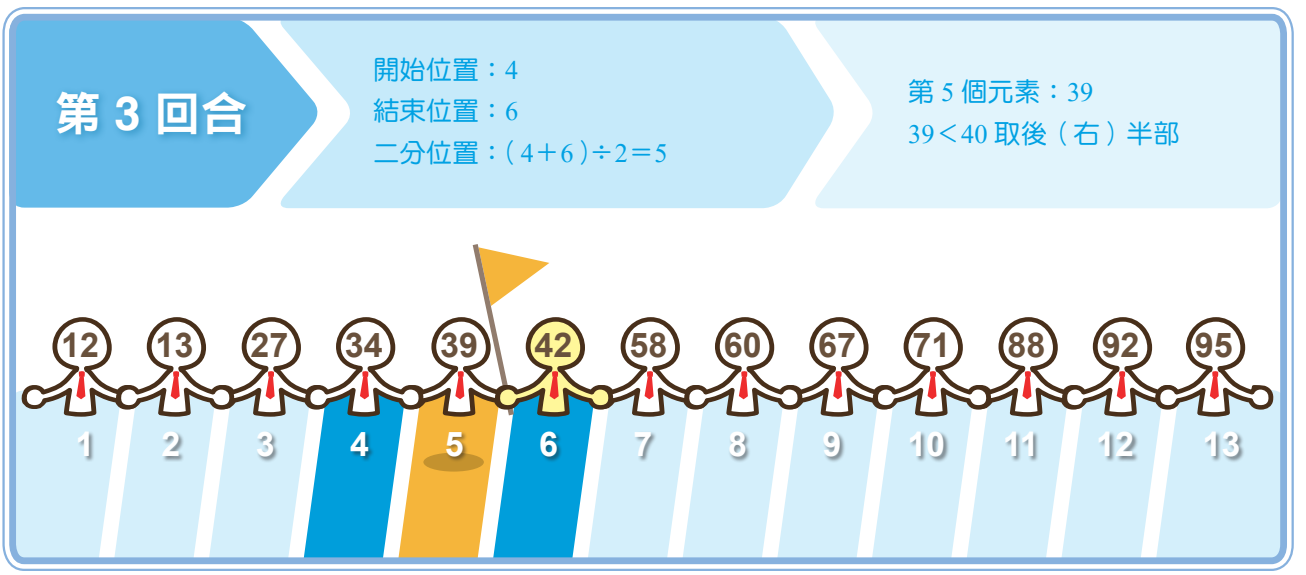


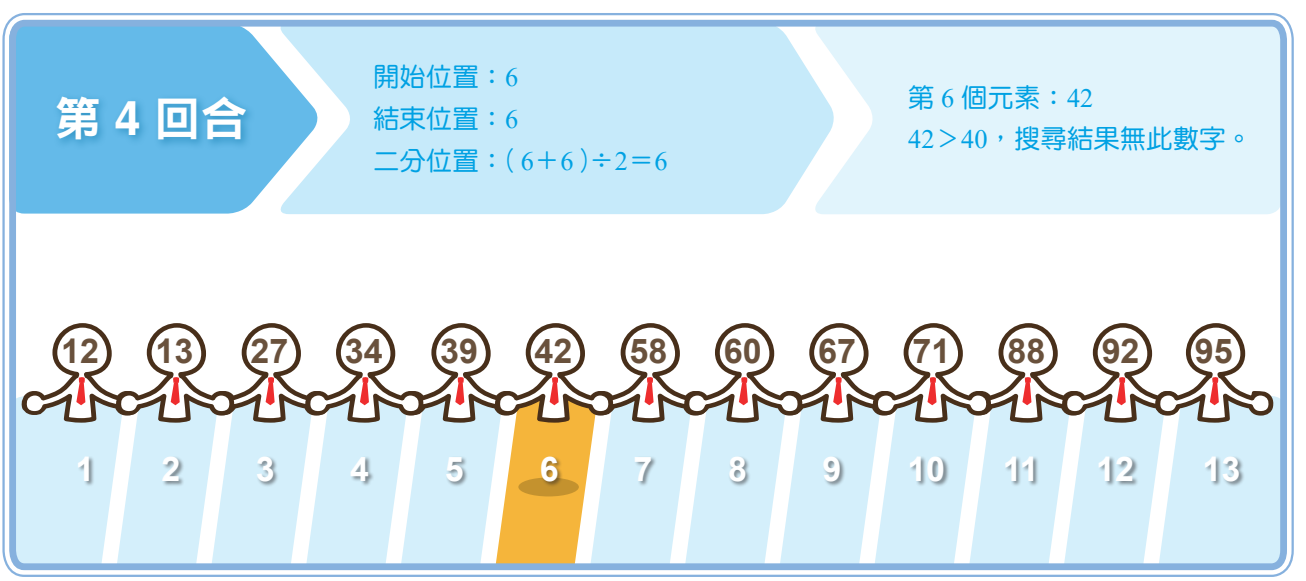
第 2 回合

開始位置：1
結束位置：6
二分位置：(1＋6)÷2＝3.5
（取 3.5 的整數部分 3）

第 3 個元素：27
27＜40 取後（右）半部







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，則取後（右）半部數列；目標資料小於該元素，則取前（左）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。

6-3-3 搜尋法程式設計的應用

認識循序搜尋法與二元搜尋法的原理後，在 Scratch 中是如何進行搜尋及判斷找到資料與否？我們以循序搜尋法做為範例，了解程式運作的邏輯與過程。

範例：循序搜尋法

點擊綠旗後，小貓執行下列動作：

1. **建立從數字 1～100 中隨機取出 50 個數字的純文字檔，並將此檔案匯入原始資料裡。**
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《循序搜尋法》的程式，想一想這個範例的程式是如何運作？

1

範例執行後，**原始資料**呈現匯入的未排序數列，並讓小貓詢問後，輸入欲搜尋的數字。

2

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

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

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 如何將現有的未排序數列匯入原始資料裡？

1. 如何使用現有的未排序數列？
2. 如何表示原始資料？

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？

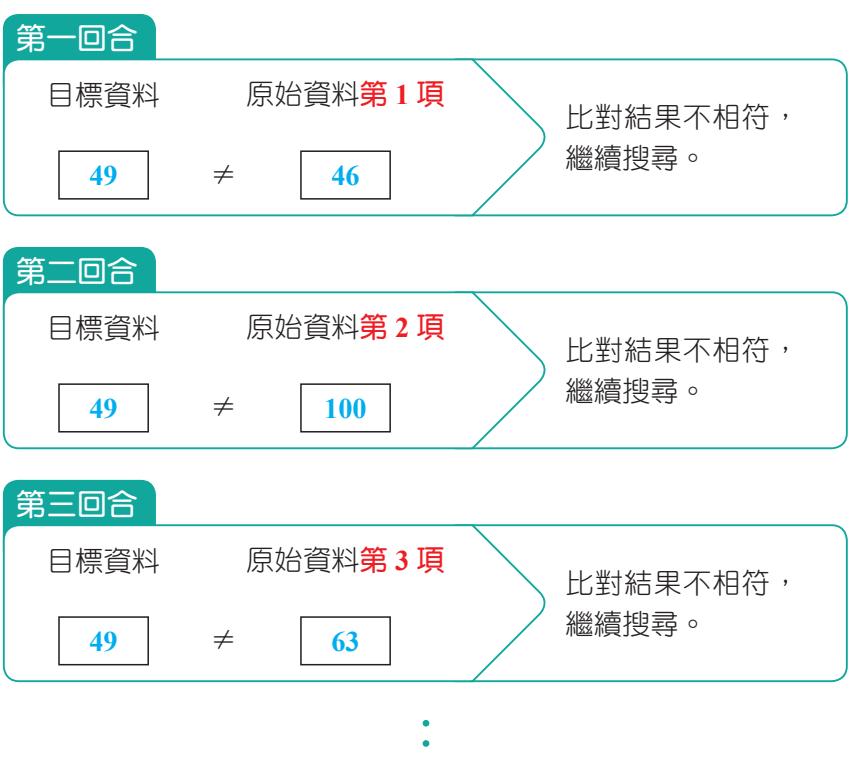
③ 如何將取出的數字與目標資料進行比對？

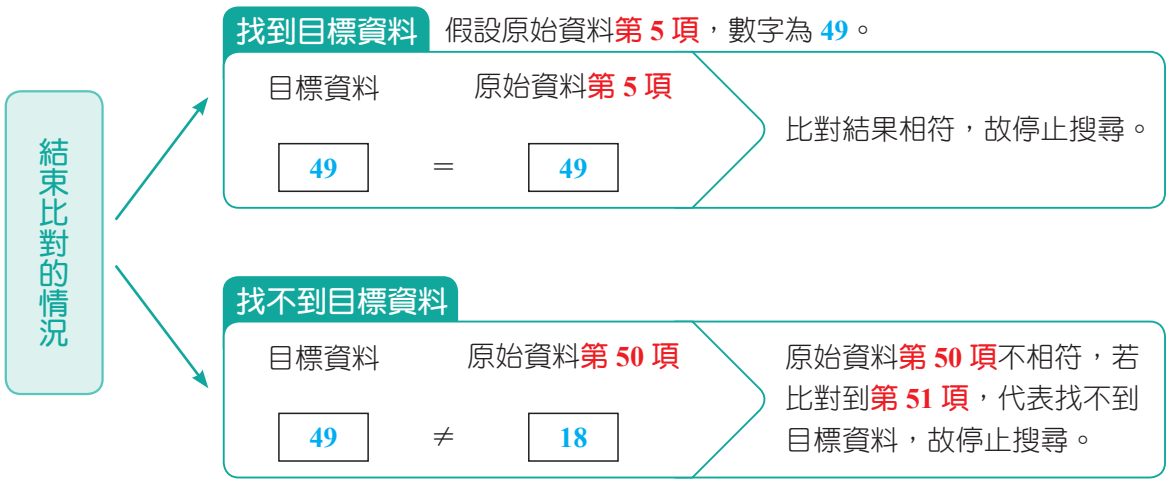
1. 如何取得目標資料？
2. 如何進行比對？

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。

假設原始資料如下表，欲搜尋的目標資料為 **49**，並依序與原始資料的每一項數字進行比對。

原始資料	
項次	數字
1	46
2	100
3	63
⋮	⋮
50	18

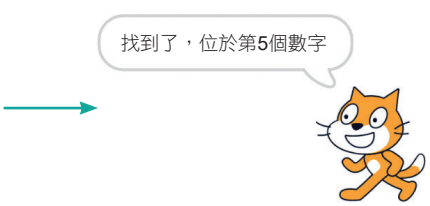




⑤ 如何說出比對的結果？

1. 當找到原始資料的數字與目標資料相同時，要怎麼表示？

比對過程	角色
目標資料＝原始資料內數字 （假設目標資料＝原始資料第 5 項數字）	找到了，位於第5個數字



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？

比對過程	角色
目標資料≠原始資料內數字	沒有符合的數字

解題步驟

問題拆解 1

如何將現有的未排序數列匯入原始資料裡？

想一想，這句話包含了什麼重要的訊息？

1. 現有的未排序數列：

如何使用現有的未排序數列？

因為數列共有 50 個數字，可從電腦中，匯入未排序數列的文字檔較為方便。

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**。

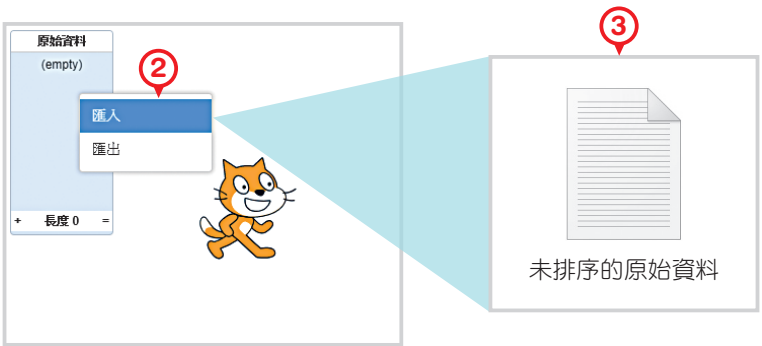
步驟 1 **設定清單。**

① 新增清單，命名為**原始資料**。

步驟 2 匯入資料。

② 在**原始資料**清單上，按右鍵選擇**匯入**。

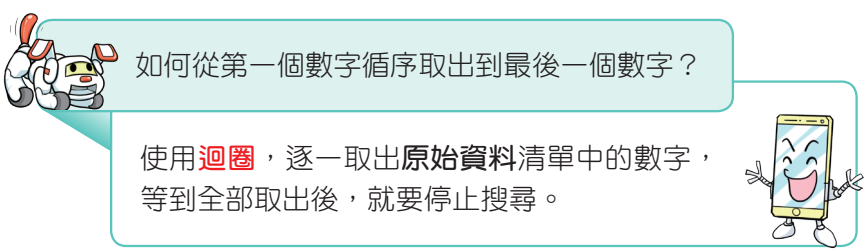
③ 從電腦中，匯入**未排序的原始資料**。



問題拆解 ② 如何從原始資料中逐一取出數字？

想一想，這句話包含了什麼重要的訊息？

逐一取出數字：



步驟 3 設定變數。

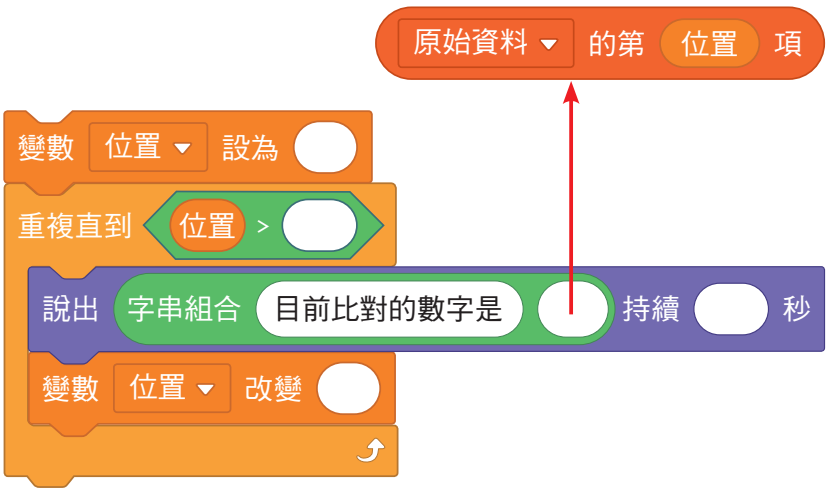
④ 新增**變數**，命名為**位置**。

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. 迴圈內**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 3

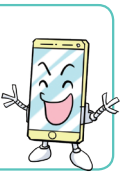
如何將取出的數字與目標資料進行比對？

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？

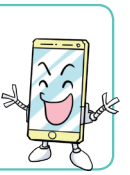
我們可以透過**詢問**積木，請使用者輸入一個數字做為搜尋的目標資料。



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？

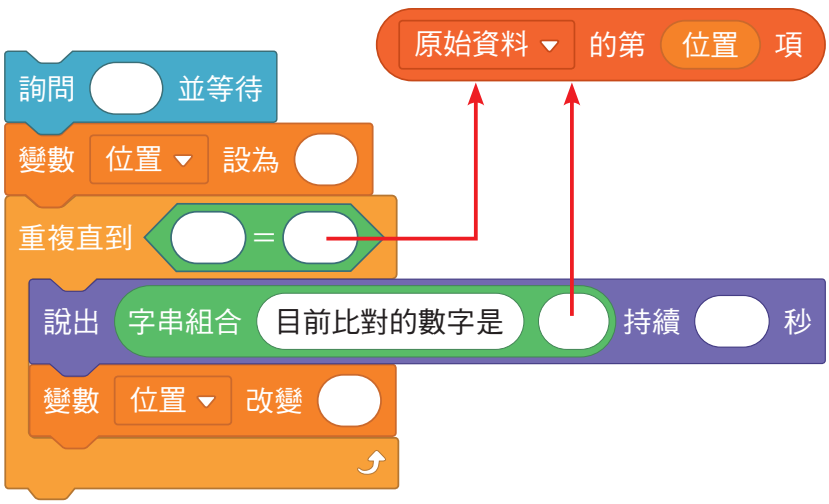
目標資料與**原始資料**清單中的每個數字進行比對，當目標資料與**原始資料**清單中的某個數字相等時，則停止比對。



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. 如何設定判斷條件，讓使用者輸入的數字做為搜尋的目標，並在找到目標資料時，即停止比對？



問題拆解 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。

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：

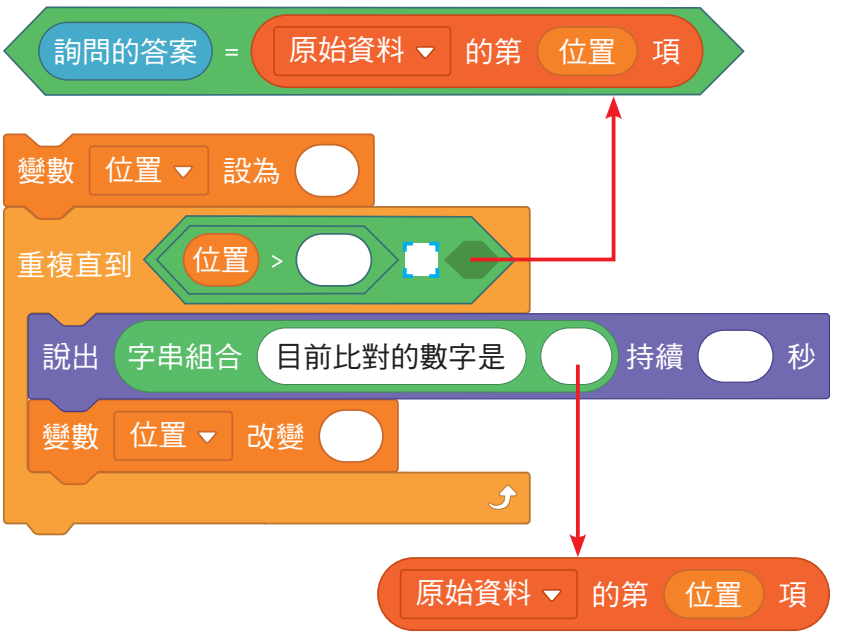
有兩個情況，如何判斷任一情況發生時都要跳出條件式迴圈？

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 5

如何說出比對的結果？

想一想，這句話包含了什麼重要的訊息？

說出比對的結果：

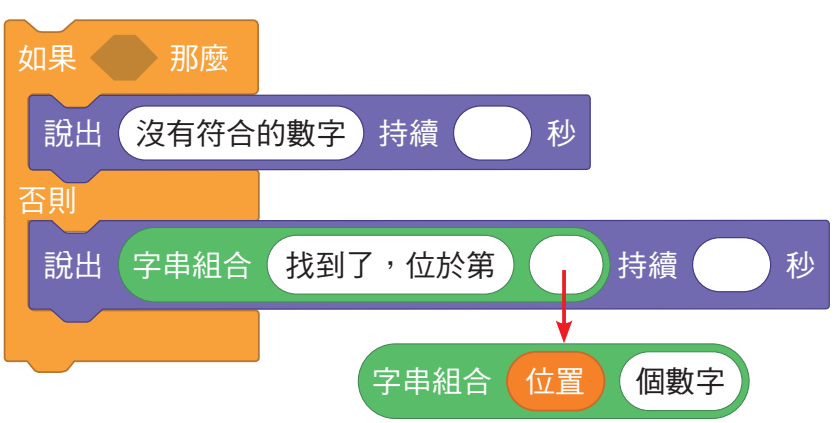
1. 當找到**原始資料**清單中的數字與目標資料相同時，要怎麼表示？
2. 如果從**原始資料**清單中都找不到與目標資料相同的數字，要怎麼表示？

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？



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」。