

3-2 陣列程式—簡易點餐機

- 任務說明
 - 概念加油站：認識「角色分身」
 - 概念加油站：利用「分身」產生多個角色
 - 概念加油站：計算總和
 - 逐步解析1
 - 逐步解析2
 - 逐步解析3
 - 逐步解析4

任務說明

1.設計點餐機介面

2.利用 、 按鈕增減餐點數量。

(1) 餐點數量即時顯示於清單中。

(2) 即時呈現「總金額」。

3.按下 ，會用語音說出消費總金額。

4.按下 ，會清空點餐資料與總金額。



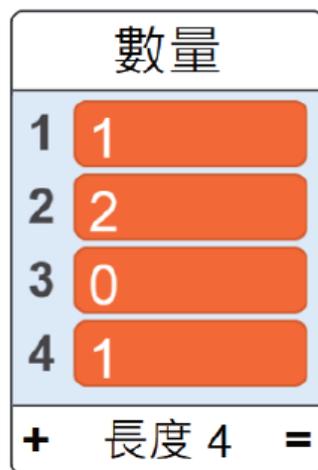
▶1 設定點餐介面

利用「分身」設定
餐點圖、**+**、**-**按
鈕的位置。



▶2 點餐功能

利用 **+**、**-** 按鈕，
增減餐點數量，並記
錄於清單中。



▶3 餐點明細、總金額

點餐時顯示 餐點明細、總金額。



▶4 結帳、清空

按下 結帳 會利用語音說出消費總金額；
按下 清空 則會清除點餐資料，以重新點餐。



概念加油站

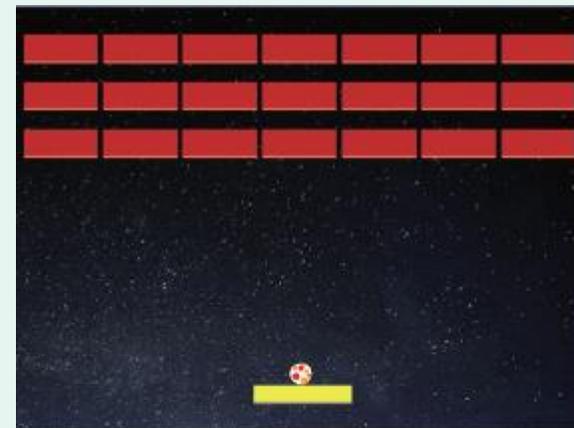
認識「角色分身」

概念1 認識「角色分身」

Scratch 「分身」功能可做到：

1. 快速複製：

設定好一個角色（本尊）的程式，快速產生具有相同程式的「分身」。



2. 獨立控制：

每個分身各自獨立，不會互相影響。

例如：一個分身被移除，不影響其他分身。



分身程式積木功能說明

Scratch積木	說明
	<p>可以建立自己的分身，也可以建立其他角色的分身。</p>  

▪ 分身程式積木功能說明

Scratch積木	說明
	撰寫在「本尊」中，當分身產生之後，分身會執行下面的程式。
	撰寫在「本尊」中，可以將畫面上的分身刪除，而原角色不受影響。



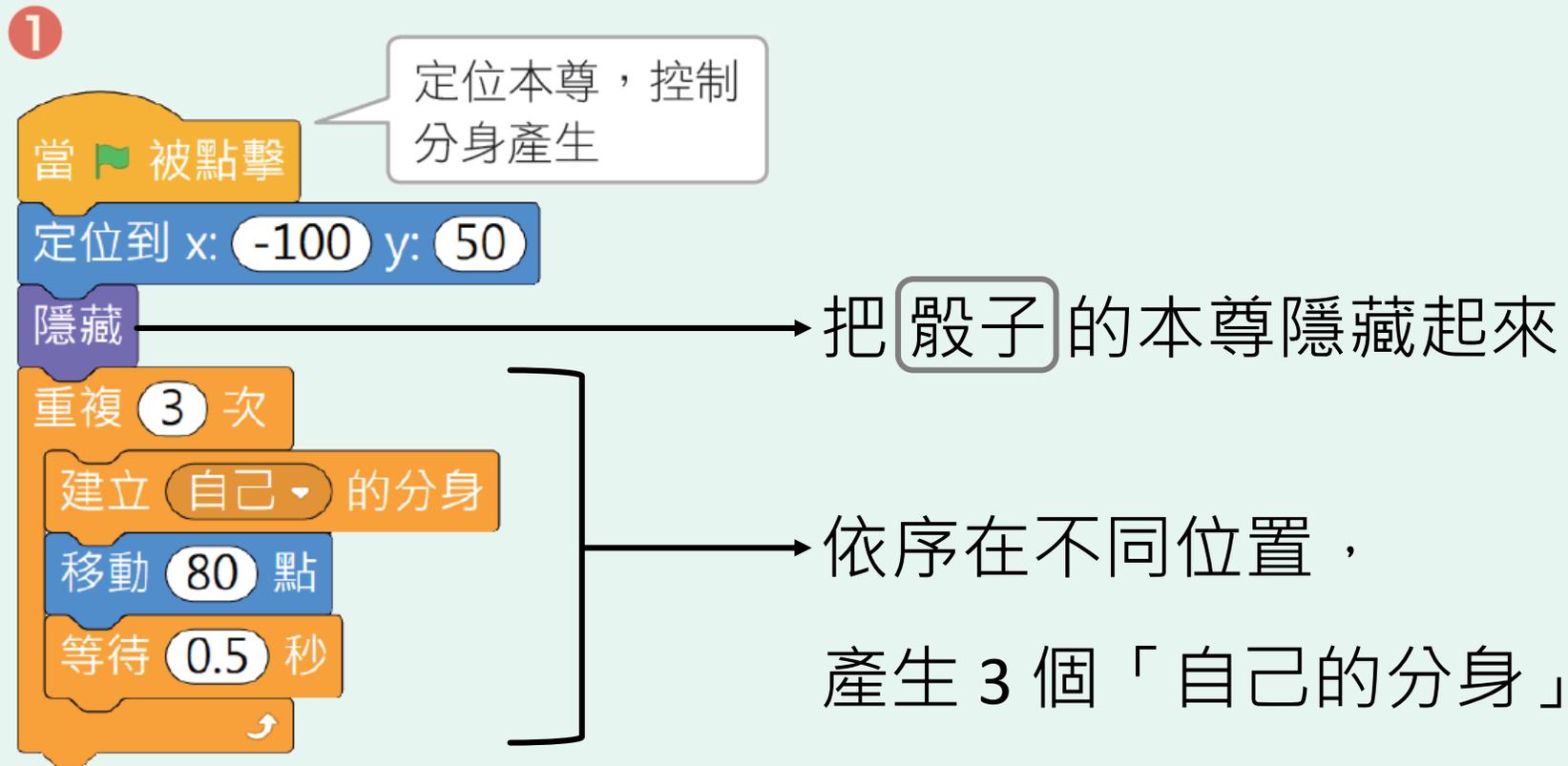
概念加油站

利用「分身」產生多個角色

概念2 建立角色自己的分身

以檔案**擲骰子.sb3**為例：

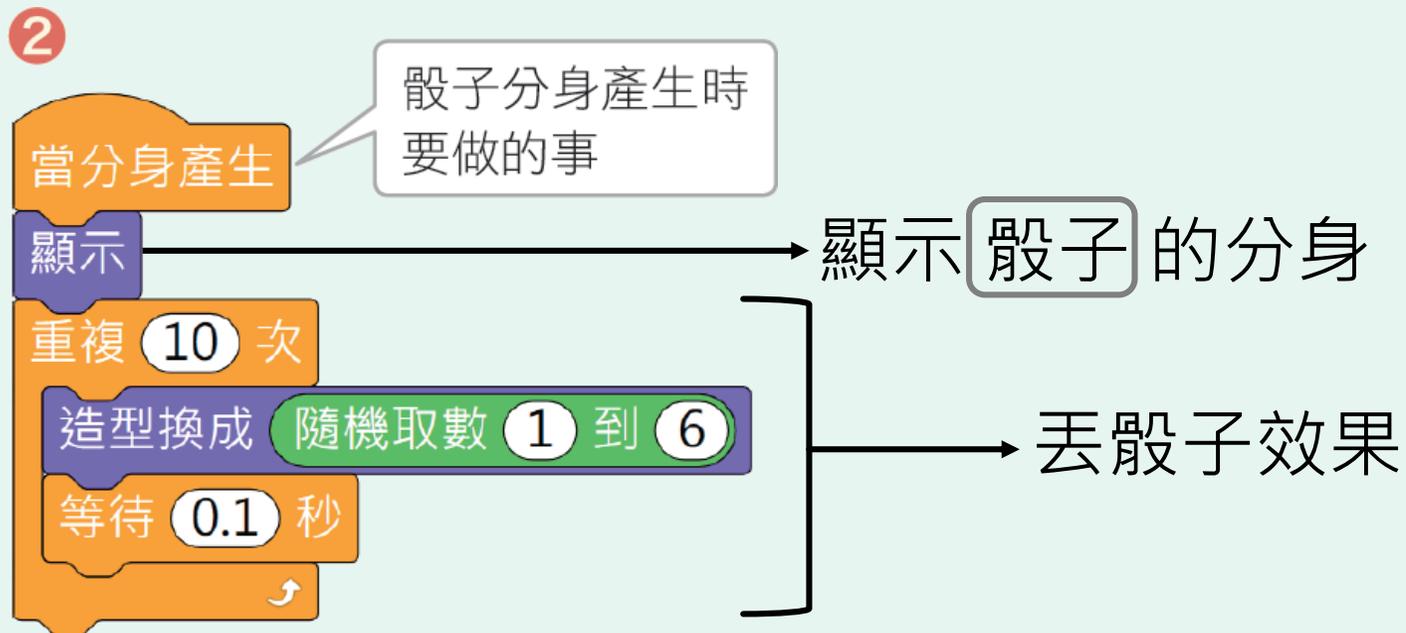
- 1 點擊綠旗後，會在畫面中產生 3 顆骰子。



概念2 建立角色自己的分身

以檔案**擲骰子.sb3**為例：

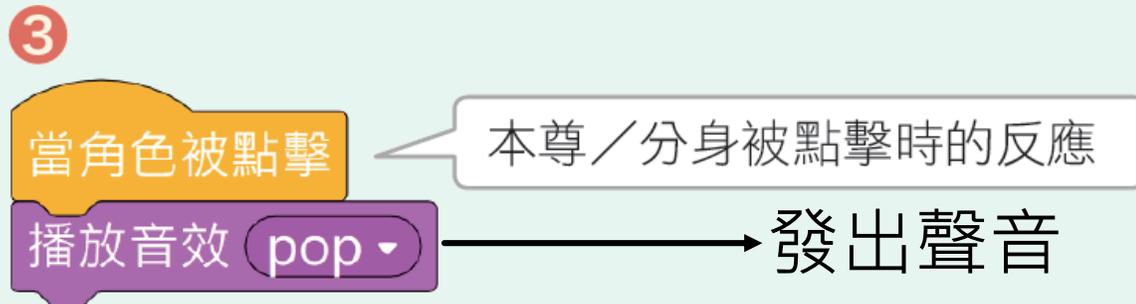
② 每個骰子出現時，隨機切換點數，呈現丟骰子效果。



☞ 概念2 建立角色自己的分身

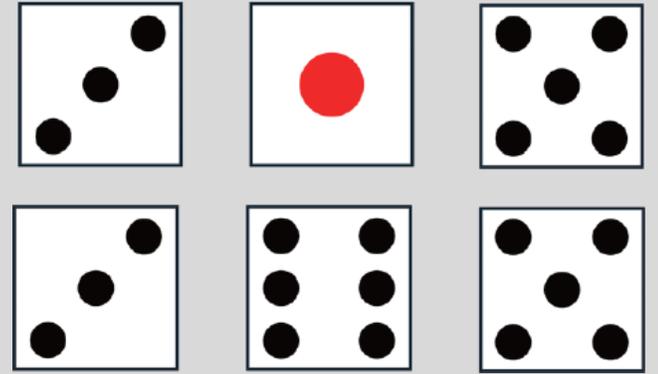
以檔案擲骰子.sb3 為例：

③ 用滑鼠點擊骰子時，骰子發出聲音。



請試著修改 **擲骰子.sb3** 的程式：

1. 使畫面呈現擲出 6 顆骰子（如右圖）。



這邊要放解答程式及說明
像是右邊的呈現方式
圖我是抓之前的範例
再請編輯補

```
當 旗幟 被點擊
  變數 時間 設為 10
  重複無限次
    等待 1 秒
    變數 時間 改變 -1
    如果 時間 = 0 那麼
      停止 全部
```

停止所有角色、舞臺的全部程式。

請試著修改 **擲骰子.sb3** 的程式：

2. 當骰子被點擊時，該骰子要發出聲音，然後消失，而其他骰子不動。

這邊要放解答程式及說明
像是右邊的呈現方式
圖我是抓之前的範例
再請編輯補

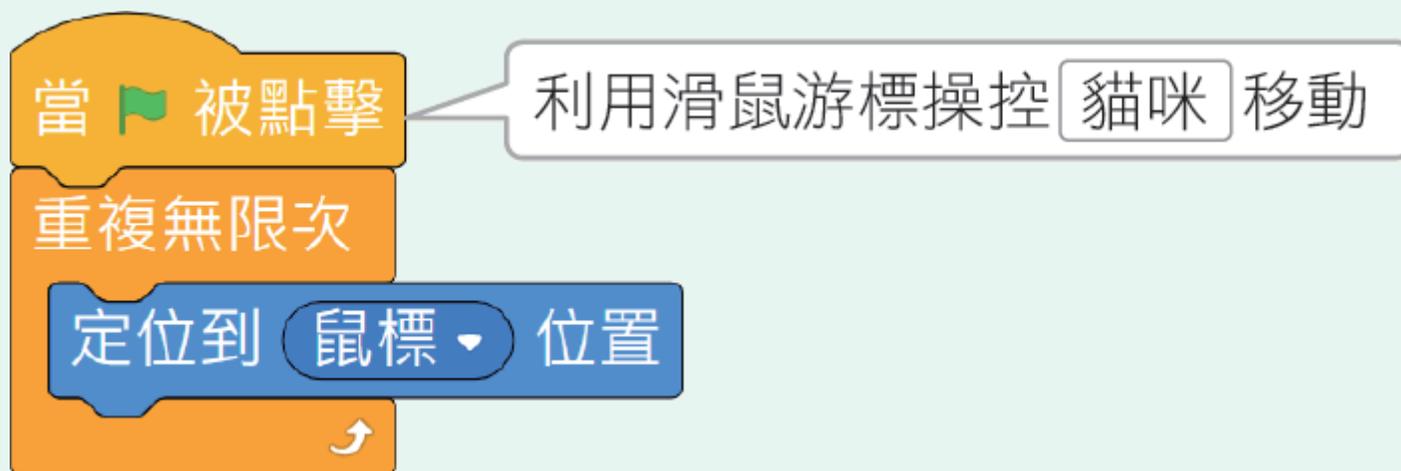


停止所有角色、舞臺的全部程式。

概念 2 建立其他角色的分身

以檔案 `飛貓子彈.sb3` 為例：

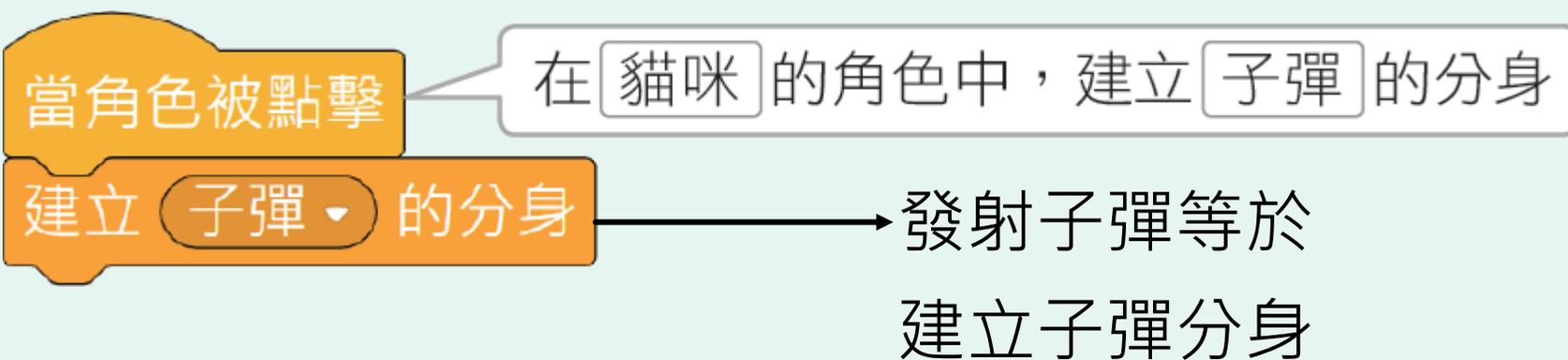
1. 點擊綠旗執行程式，貓咪會隨著滑鼠游標移動。



概念2 建立其他角色的分身

以檔案 `飛貓子彈.sb3` 為例：

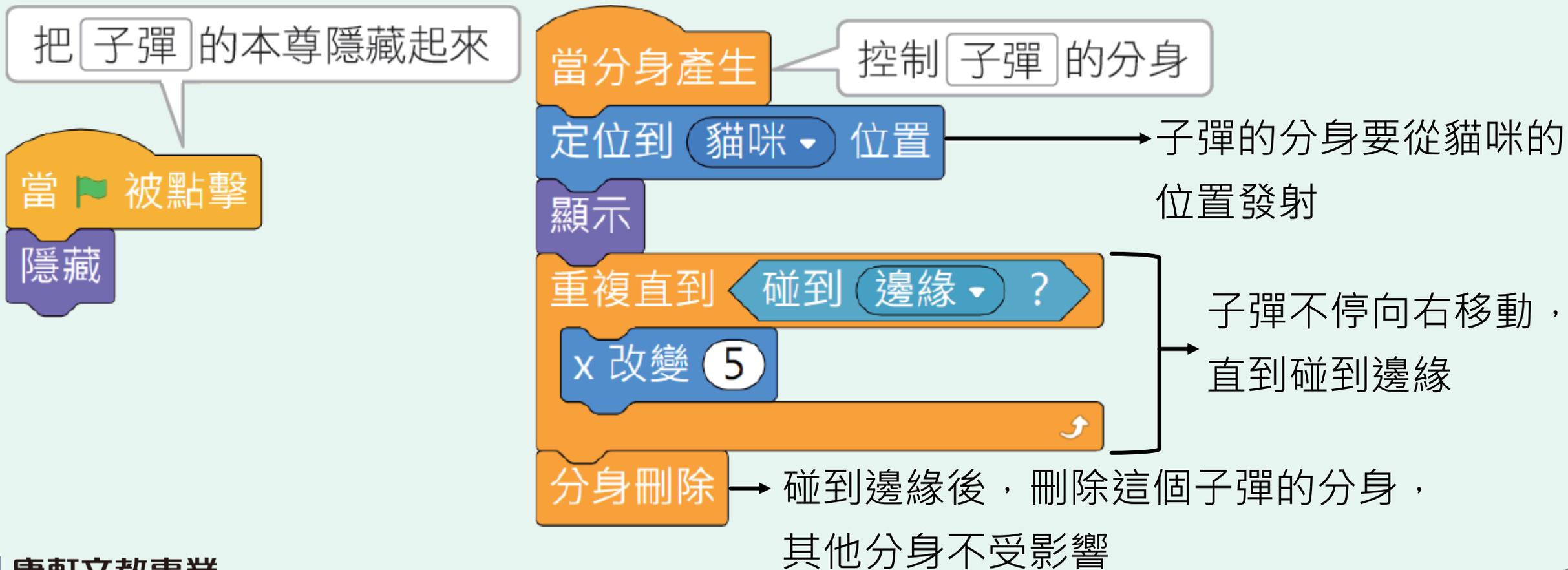
2. 點擊滑鼠（相當於點擊貓咪）發射子彈，連續點擊可連續發射。



概念2 建立其他角色的分身

以檔案 **飛貓子彈.sb3** 為例：

3. 子彈不斷向右移動，碰到舞臺邊緣時消失。





概念加油站

計算總和

概念3 計算總和

- 計算「1 加到 3 的總和」

變數 sum 設為 0

變數 n 設為 0

重複 3 次

變數 n 改變 1

加總公式

變數 sum 設為 $sum + n$ ($sum = sum + n$)



計算總和的程式如左，變數的計算過程說明如下：



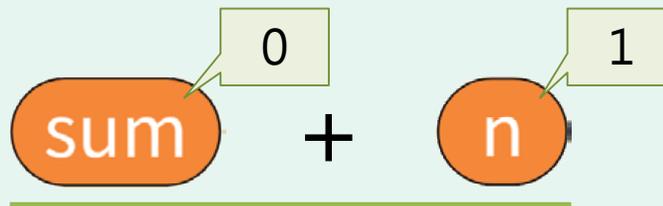
- 變數 `sum` 設為 `sum + n` (`sum = sum + n`)
- 「`sum = sum + n`」不是一個數學式，因為當 $n \neq 0$ ，這個公式就不可能成立。
- 在文字式程式語言中，「`=`」是代表「把右邊的值存入左邊的變數中」，也就是變數積木上所寫「設為」的概念。

概念3 計算總和

- 計算「1 加到 3 的總和」

第 1 輪

步驟①

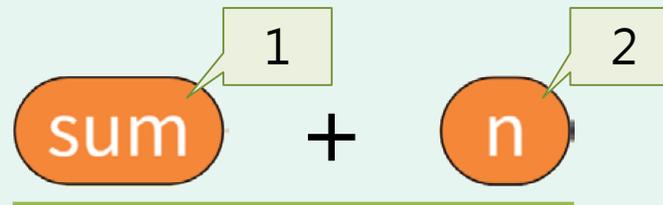


步驟②

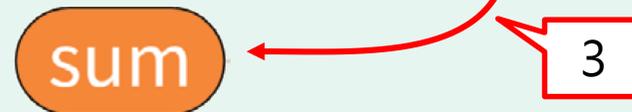


第 2 輪

步驟①

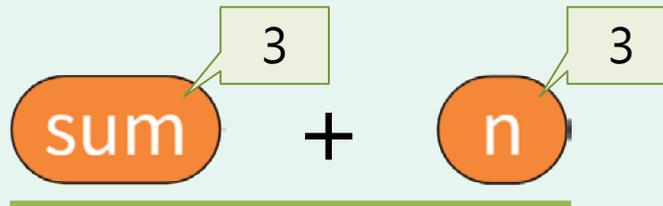


步驟②



第 3 輪

步驟①



步驟②



請開啟新檔案撰寫程式，讓小貓咪說出：

1. 「 $1 + 2 + 3 + \dots + 99 + 100$ 」的值為何？

這邊要放解答是多少
以及放程式及說明
像是右邊的呈現方式
圖我是抓之前的範例
再請編輯補



停止所有角色、
舞臺的全部程式。

請開啟新檔案撰寫程式，讓小貓咪說出：

2. 「 $2 + 4 + 6 + \dots + 98 + 100$ 」的值為何？

這邊要放解答是多少
以及放程式及說明
像是右邊的呈現方式
圖我是抓之前的範例
再請編輯補

```
當 旗幟被點擊  
變數 時間 設為 10  
重複無限次  
  等待 1 秒  
  變數 時間 改變 -1  
  如果 時間 = 0 那麼  
    停止 全部
```

停止所有角色、
舞臺的全部程式。

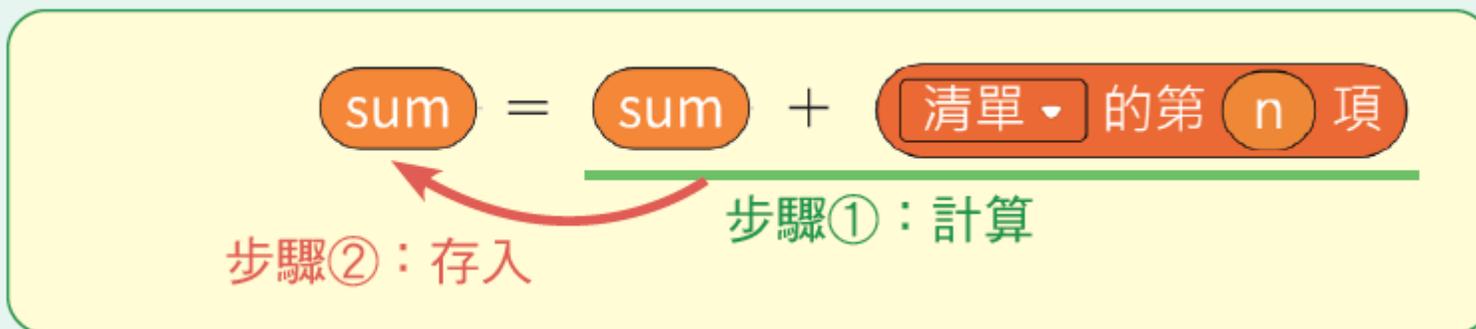
概念3 計算總和

- 計算「清單資料的總和」

變數 `sum` 設為 `sum` + 清單 的第 `n` 項

($sum = sum + \text{清單}[n]$)

計算總和的程式如右，變數的計算過程說明如下：



概念3 計算總和

清單	
1	20
2	25
3	50
+ 長度 3 =	

- 計算「清單資料的總和」

第 1 輪

步驟①



步驟②



第 2 輪

步驟①



步驟②



第 3 輪

步驟①



步驟②



附圖為小宇零用錢收支情形，請開啟 **收支表.sb3**

，撰寫計算程式，讓小貓咪說出計算結果：

1. 小宇零用錢整體結餘為多少？（所有數值加總）

收支表	
1	200
2	-50
3	-30
4	150
5	-60
+ 長度 5	=

這邊要放解答是多少
以及放程式及說明
像是右邊的呈現方式
圖我是抓之前的範例
再請編輯補

```
當 被點擊
變數 時間 設為 10
重複無限次
  等待 1 秒
  變數 時間 改變 -1
  如果 時間 = 0 那麼
    停止 全部
```

停止所有角色、
舞臺的全部程式。

附圖為小宇零用錢收支情形，請開啟 **收支表.sb3**

，撰寫計算程式，讓小貓咪說出計算結果：

2. 小宇本週共支出多少錢？（所有負值加總）

收支表	
1	200
2	-50
3	-30
4	150
5	-60
+ 長度 5 =	

這邊要放解答是多少
以及放程式及說明
像是右邊的呈現方式
圖我是抓之前的範例
再請編輯補

```
當 被點擊
變數 時間 設為 10
重複無限次
  等待 1 秒
  變數 時間 改變 -1
  如果 時間 = 0 那麼
    停止 全部
```

停止所有角色、舞臺的全部程式。

概念3 計算總和

- 先進行清單運算，再計算總和

變數 `sum` 設為 0

變數 `n` 設為 0

重複 3 次

變數 `n` 改變 1

加總公式

變數 `sum` 設為 `sum` + `單價` 的第 `n` 項 * `數量` 的第 `n` 項

$$\text{sum} = \text{sum} + (\text{單價} [n] \times \text{數量} [n])$$

計算清單總和的程式如左，變數的計算過程如下：

`sum` = `sum` + `單價` 的第 `n` 項 × `數量` 的第 `n` 項

步驟①：計算

步驟②：存入

概念3 計算總和

單價		數量	
1	20	1	2
2	25	2	6
3	50	3	1
+ 長度 3 =		+ 長度 3 =	

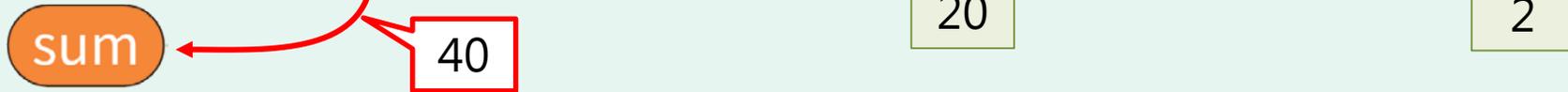
- 先進行清單運算，再計算總和

第 1 輪

步驟①

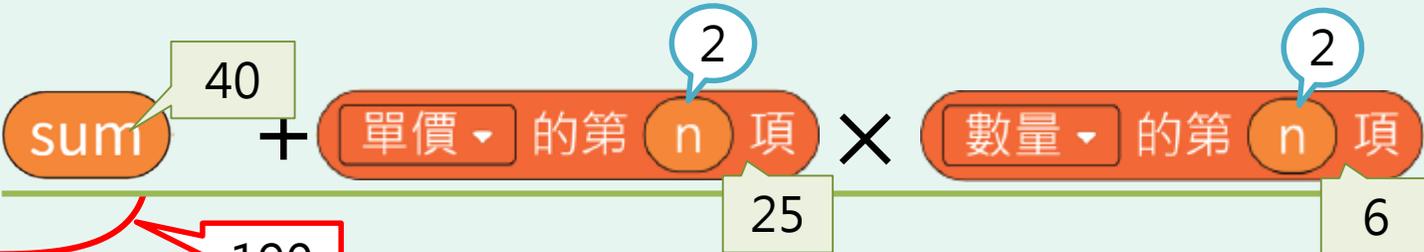


步驟②

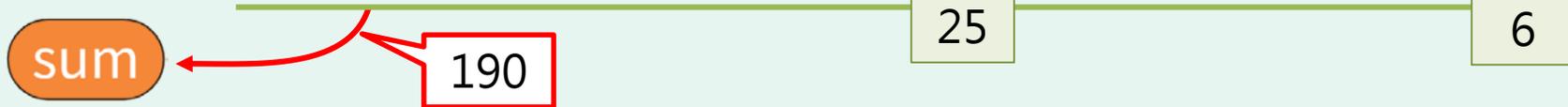


第 2 輪

步驟①

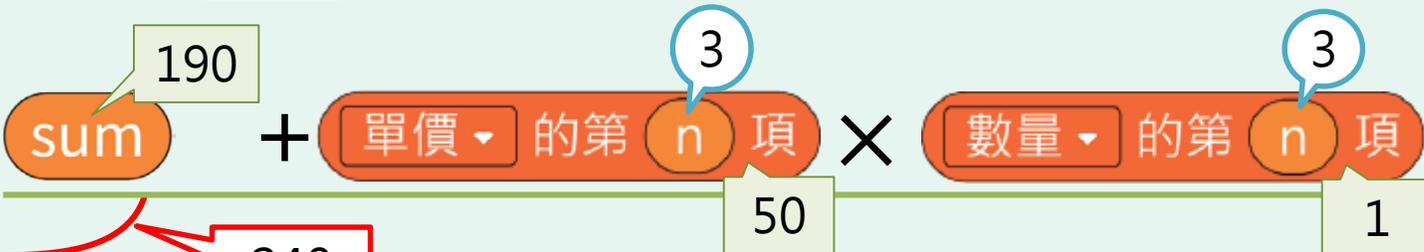


步驟②

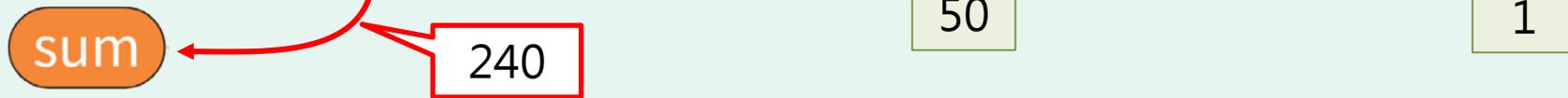


第 3 輪

步驟①



步驟②



逐步解析 1

設定點餐介面

題目解析流程

▶1 設定點餐介面

利用「分身」設定
餐點圖、**+**、**-**按
鈕的位置。



▶2 點餐功能

利用 **+**、**-** 按鈕，
增減餐點數量，並記
錄於清單中。

數量	
1	1
2	2
3	0
4	1
+ 長度 4 =	

▶3 餐點明細、總金額

點餐時顯示 **餐點明細**、**總金額**。



▶4 結帳、清空

按下 **結帳** 會利用語音說出消費總金額；
按下 **清空** 則會清除點餐資料，以重新點餐。

檔案中已包含具有 4 種造型

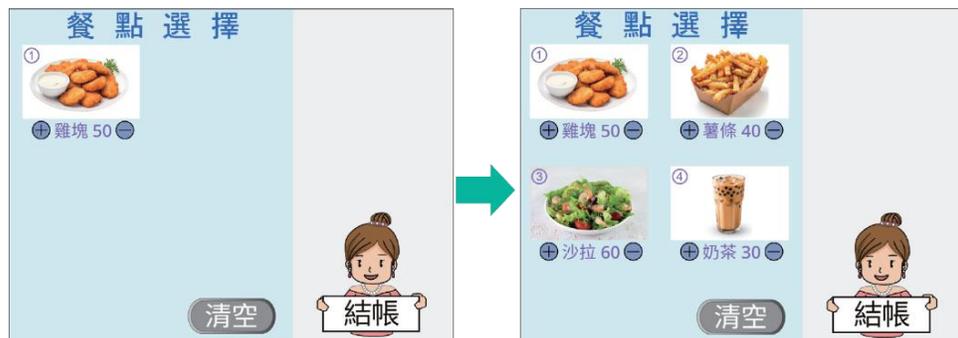
的 **餐點圖**，以及 **+**、**-**，

我們要利用「分身」功能，

將各角色安排成點餐介面。

2

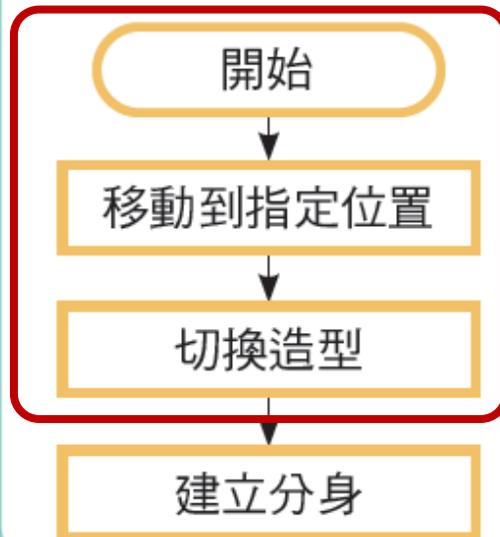
3



流程圖

【餐點圖】

1



【+、-】



Q1 設定初始坐標與造型

Q2 產生分身

Q3 設定顯示 / 隱藏狀態

Q1

設定初始
坐標與造型

程式開始時，設定本尊的
初始狀態。

當 被點擊

定位到 x: y:

造型換成 雞塊

- ✓ 雞塊
- 薯條
- 沙拉
- 奶茶

Q2

產生分身

重複利用分身功能產生分身，
並設定造型與位置。

重複 次

造型換成下一個

程式區塊

x 改變

建立 的分身

Q3

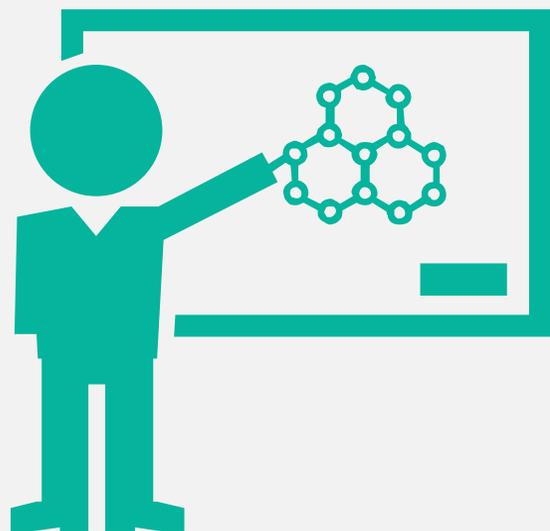
設定顯示 /
隱藏狀態

顯示分身、隱藏本尊。

當分身產生

顯示

隱藏



請看老師 解題示範

逐步解析 2

點餐功能

▶1 設定點餐介面

利用「分身」設定
餐點圖、**+**、**-**按
鈕的位置。



▶2 點餐功能

利用 **+**、**-** 按鈕，
增減餐點數量，並記
錄於清單中。

數量	
1	1
2	2
3	0
4	1
+ 長度 4 =	

▶3 餐點明細、總金額

點餐時顯示 **餐點明細**、**總金額**。



▶4 結帳、清空

按下 **結帳** 會利用語音說出消費總金額；
按下 **清空** 則會清除點餐資料，以重新點餐。

利用 $+$ 、 $-$ 進行點餐，並記錄在數量清單中。

1

1. 點擊時要發出提示音。

2. 增減數量：

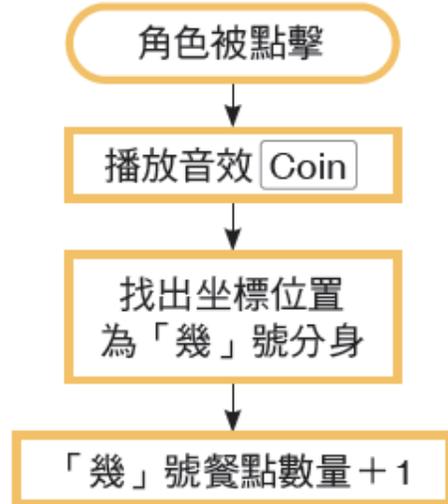
2

依據點擊的餐點編號，在清單數量中對應的資料位置 $+1$ 或 -1 。

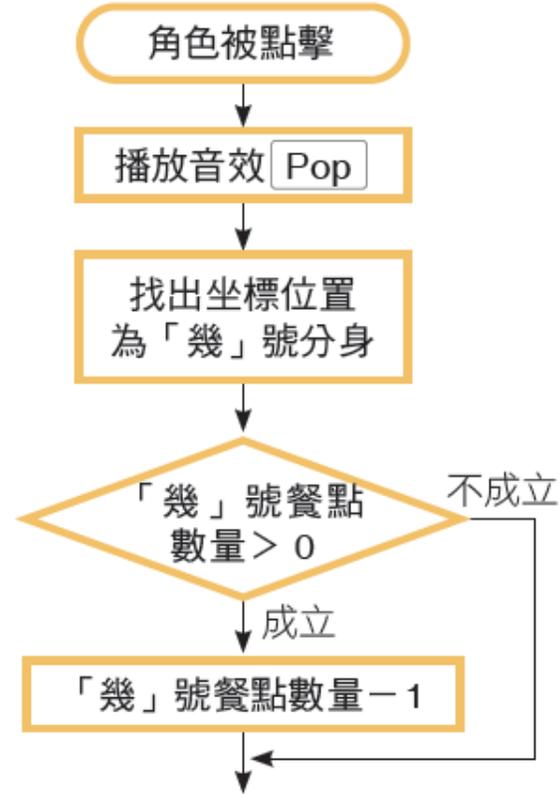
(※餐點的數量不能是負值。)

流程圖

[+]



[-]



Q1 顯示數量清單

Q2 增減餐點數量

Q1

顯示
數量清單

顯示清單，
以檢視結果。

<input checked="" type="checkbox"/>	變數	<input type="text" value="建立一個清單"/>
<input type="checkbox"/>	名稱	
<input type="checkbox"/>	價格	
<input checked="" type="checkbox"/>	數量	
<input type="checkbox"/>	餐點明細	

Q2

增減
餐點數量

判斷點擊的是哪份餐點的 + 或 - ，
並將結果存入對應清單中。
※餐點的數量不能是負值。

當角色被點擊

播放音效

如果 條件 那麼

程式區塊1

程式區塊2

變數 設為

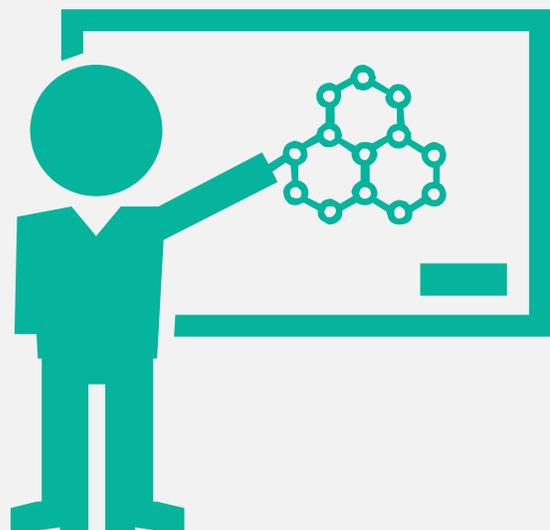
x 座標 y 座標

= 且

替換 的第 項為

的第 項 變數

+ - >



請看老師 解題示範

逐步解析 3

餐點明細、總金額

題目解析流程

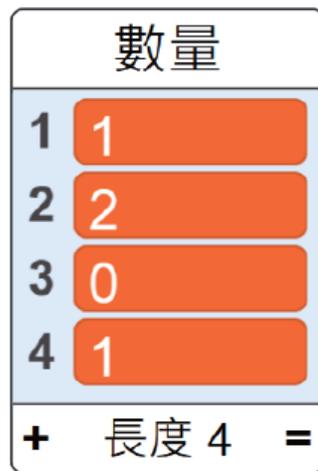
▶1 設定點餐介面

利用「分身」設定
餐點圖、**+**、**-**按
鈕的位置。



▶2 點餐功能

利用 **+**、**-** 按鈕，
增減餐點數量，並記
錄於清單中。



▶3 餐點明細、總金額

點餐時顯示 **餐點明細**、**總金額**。

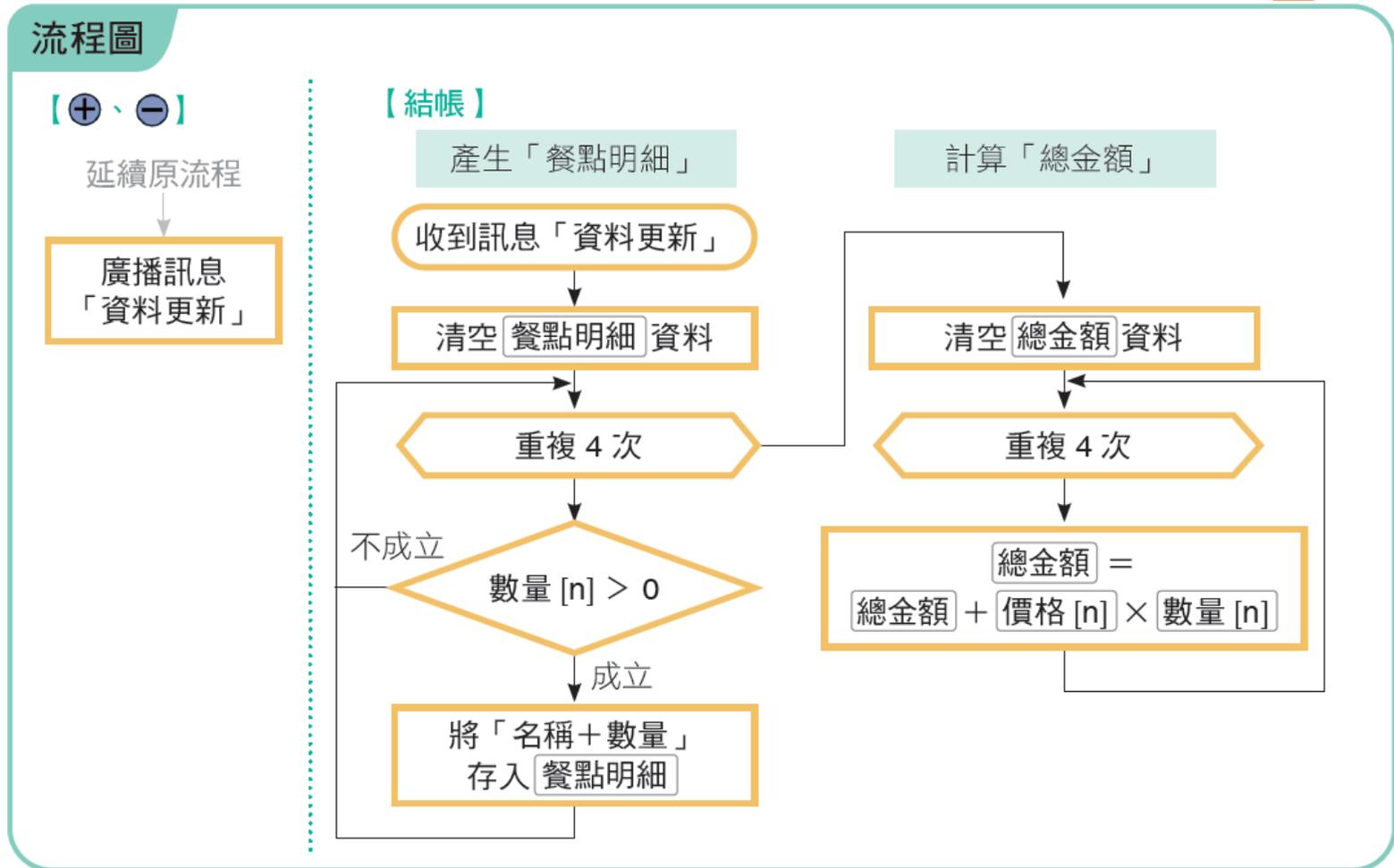


▶4 結帳、清空

按下 **結帳** 會利用語音說出消費總金額；
按下 **清空** 則會清除點餐資料，以重新點餐。

1. 每次點擊 \oplus 、 \ominus 時，將「名稱 + 數量」呈現在餐點明細清單中。
2. 每次點擊 \oplus 、 \ominus 時，計算總金額，呈現在變數總金額中。

1



Q1
點餐結果呈現時機

Q2
即時呈現點餐結果

Q3
計算總金額

Q1

點餐結果
呈現時機

利用廣播功能，
通知點餐明細更新。

廣播訊息

Q2

即時呈現
點餐結果

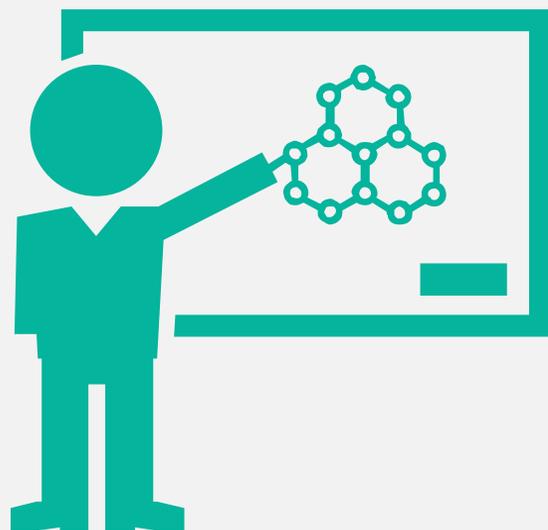
Q3

計算
總金額

收到廣播訊息，更新呈現在畫面上的數據。

The code blocks are as follows:

- Q2:**
 - 當收到訊息 []
 - 刪除 [] 的所有項目
 - 變數 [] 設為 []
 - 變數 []
 - 變數 [] 改變 []
 - 重複 [] 次 [] 程式區塊
- Q3:**
 - 如果 [] 條件 [] 那麼 [] 程式區塊1 []
 - [] 程式區塊2 []
 - [] > []
 - 添加 [] 到 []
 - 字串組合 [] []
 - [] 的 第 [] 項 []
 - [] + [] [] * []



請看老師 解題示範

逐步解析 4

結帳、清空

▶1 設定點餐介面

利用「分身」設定
餐點圖、**+**、**-**按
鈕的位置。



▶2 點餐功能

利用 **+**、**-** 按鈕，
增減餐點數量，並記
錄於清單中。

數量	
1	1
2	2
3	0
4	1
+ 長度 4 =	

▶3 餐點明細、總金額

點餐時顯示 **餐點明細**、**總金額**。



▶4 結帳、清空

按下 **結帳** 會利用語音說出消費總金額；
按下 **清空** 則會清除點餐資料，以重新點餐。

點完餐後，可進行結帳，或是清空點餐資料，重新點餐。

1. 點擊 **結帳**：

- (1) 按鈕先縮小再復原 (互動效果)。
- (2) 利用語音說出「總金額○○元」。

2. 點擊 **清空**：

- (1) 按鈕先縮小再復原 (互動效果)。
- (2) 刪除點餐的資料、總金額歸零。



Q1
按鈕結帳功能設定

Q2
按鈕清空功能設定

流程圖

【結帳】



【清空】



Q1

按鈕結帳
功能設定

唸出總金額是多少。

當角色被點擊

尺寸設為 %

等待 秒

唸出

字串組合

變數

Q2

按鈕清空
功能設定

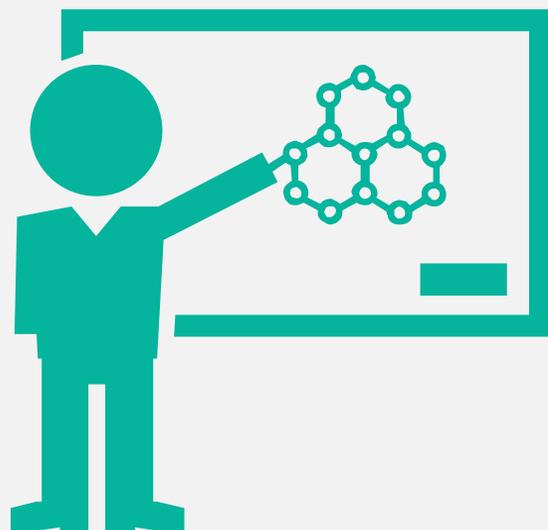
清空所有資料。

當角色被點擊

尺寸設為 %

等待 秒

函式



請看老師 解題示範

▪ 全域變數 (Scratch 預設值)

- 程式裡所有角色都能同時存取、修改。
- 概念理解：
共用一個「存錢筒」，不管什麼情況下，大家讀取到的值都是一樣的。

建立一個變數



新的變數

新變數的名稱

適用於所有角色 僅適用當前角色

取消 確定

▪ 區域變數

- 僅限於擁有該變數的角色能存取、修改。
- 各角色（分身）都可以有相同名稱的變數。
- 概念理解：
每個角色（分身）都有一個「存錢筒」，互不影響，也不能被其他角色讀取。

建立一個變數

新的變數

新變數的名稱

適用於所有角色 僅適用當前角色

取消 確定

▪ 區域變數的運用（以⊕為例）

1. 在⊕的本尊裡，設定「區域變數清單位置」。

新的變數

新變數的名稱

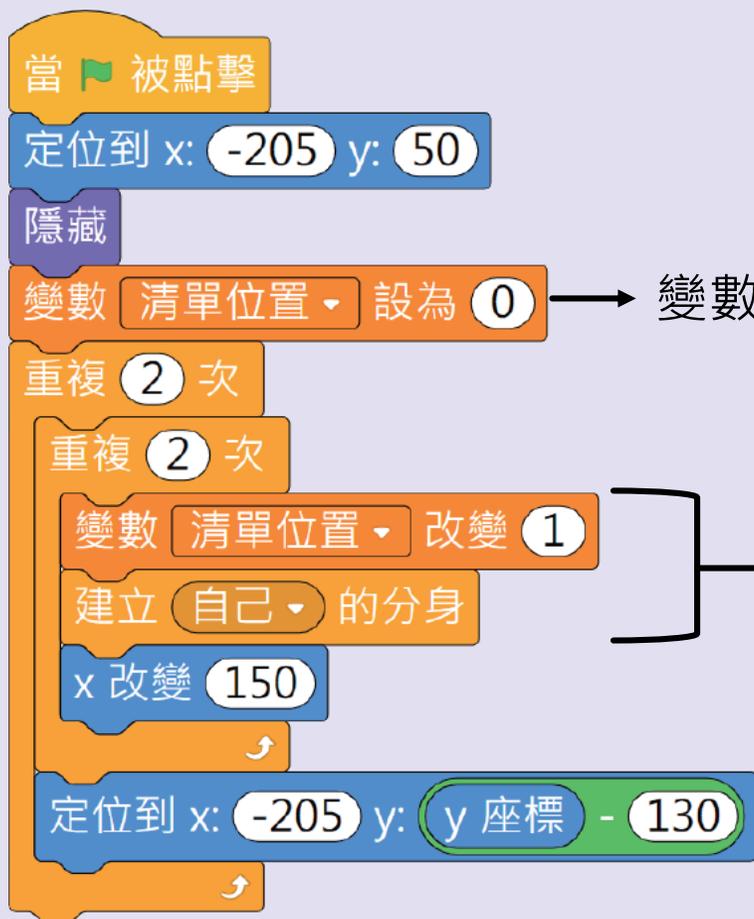
清單位置

適用於所有角色 僅適用當前角色

取消 確定

區域變數的運用 (以 \oplus 為例)

2. 在 \oplus 每次產生「分身」前，讓「清單位置 + 1」。



變數起始值為 0

每次產生分身前，變數值 + 1，即：

第 1 個分身，拿到「清單位置 = 1」的區域變數

第 2 個分身，拿到「清單位置 = 2」的區域變數

第 3 個分身，拿到「清單位置 = 3」的區域變數

第 4 個分身，拿到「清單位置 = 4」的區域變數

因為是區域變數，所以各分身的值互不影響。

區域變數的運用 (以 + 為例)

3. 當角色 (分身) 被點擊時，利用區域變數「清單位置」的值來控制點餐結果，省去利用位置判斷的繁瑣。



第 1 個分身被點擊時，數量[清單位置] 的值 + 1

→ 區域變數 清單位置 = 1 數量 [1]

第 2 個分身被點擊時，數量[清單位置] 的值 + 1

→ 區域變數 清單位置 = 2 數量 [2]

第 3 個分身被點擊時，數量[清單位置] 的值 + 1

→ 區域變數 清單位置 = 3 數量 [3]

第 4 個分身被點擊時，數量[清單位置] 的值 + 1

→ 區域變數 清單位置 = 4 數量 [4]

紅利點數算算看

為促進買氣，商店舉辦優惠活動，平日（星期一~五）消費可獲得 10% 紅利點數，假日（星期六、日）消費可獲得 20% 紅利點數。



例 ①若在星期一消費 100 元

→可獲得紅利點數

$$100 \times 0.1 = 10 \text{ 點}$$

②若在星期六消費 100 元

→可獲得紅利點數

$$100 \times 0.2 = 20 \text{ 點}$$



小橙將這星期在商店中的消費金額記錄在清單中，請以檔案 `3-2 小試身手.sb3` 來撰寫程式，協助小橙計算共可獲得多少紅利點數。

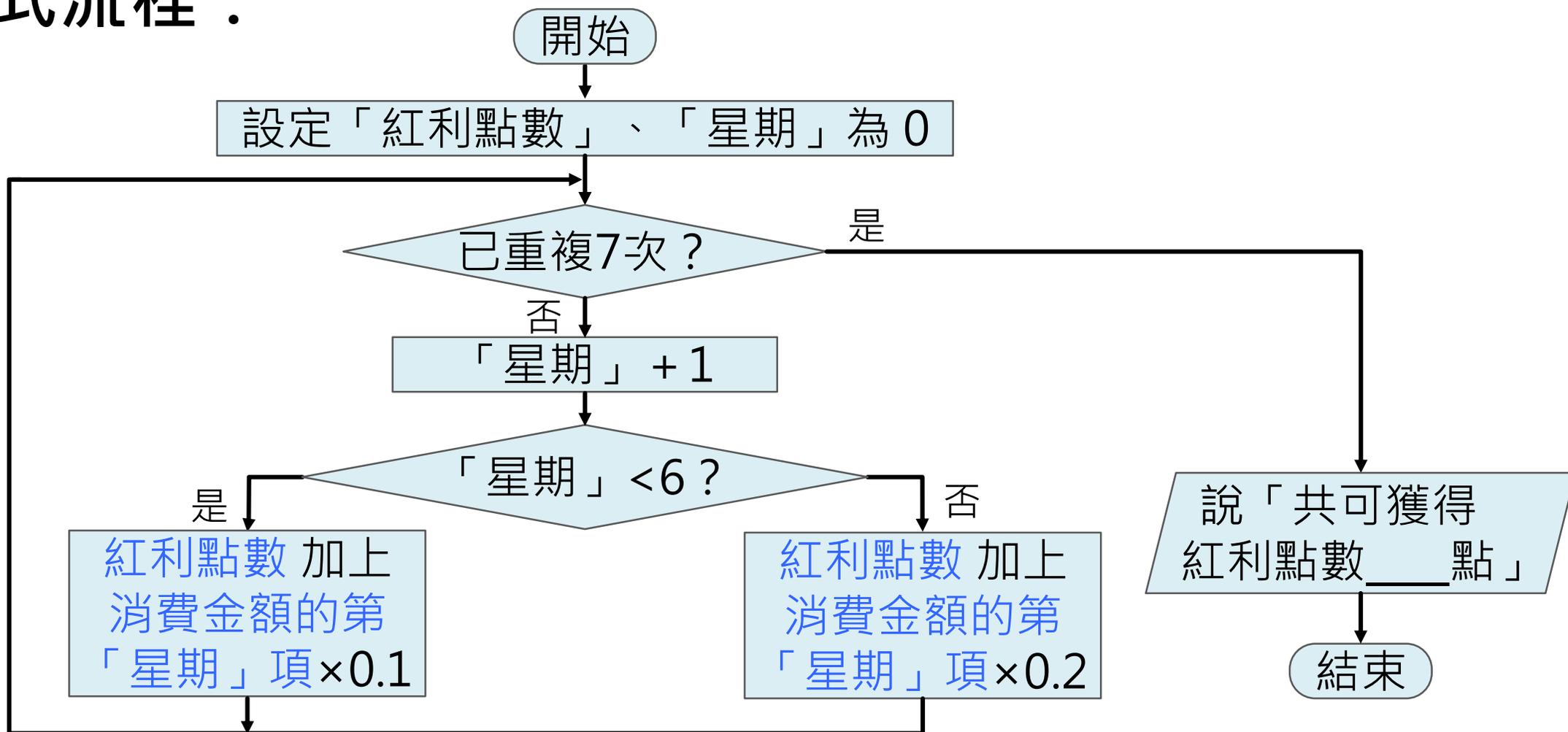
星期	一	二	三	四	五	六	日
金額	50	0	40	30	0	70	10

註 程式中已建立以上資料，你也可以先用試算表軟體算算看，檢驗自己寫的程式是否正確。

1. 讀取清單中的資料。
2. 計算並說出「共可獲得紅利點數_____點」。



程式流程：



解題關鍵：

① 設定變數：



② 重複執行：



③ 條件判斷：



④ 讀取清單：



⑤ 計算：



⑥ 說出：



小試身手 抽抽樂

為了公平起見，老師上課時常會以抽籤的方式，請同學回答問題或是進行分組。現在，讓我們試著協助老師，完成一個抽籤程式吧！



註 在角色「老師」的身上，已含有建立學生名單的程式，你也可以自己改成班上同學的名單。

1. 建立 **抽1人**、**抽2人**、**抽3人** 及 **抽n人** 4 個按鈕，其中 **抽n人** 可以讓使用者自行輸入人數。
2. 依指定的人數，抽出不重複的姓名並存入清單中，每抽出一個人，老師說出「幸運兒： 」。



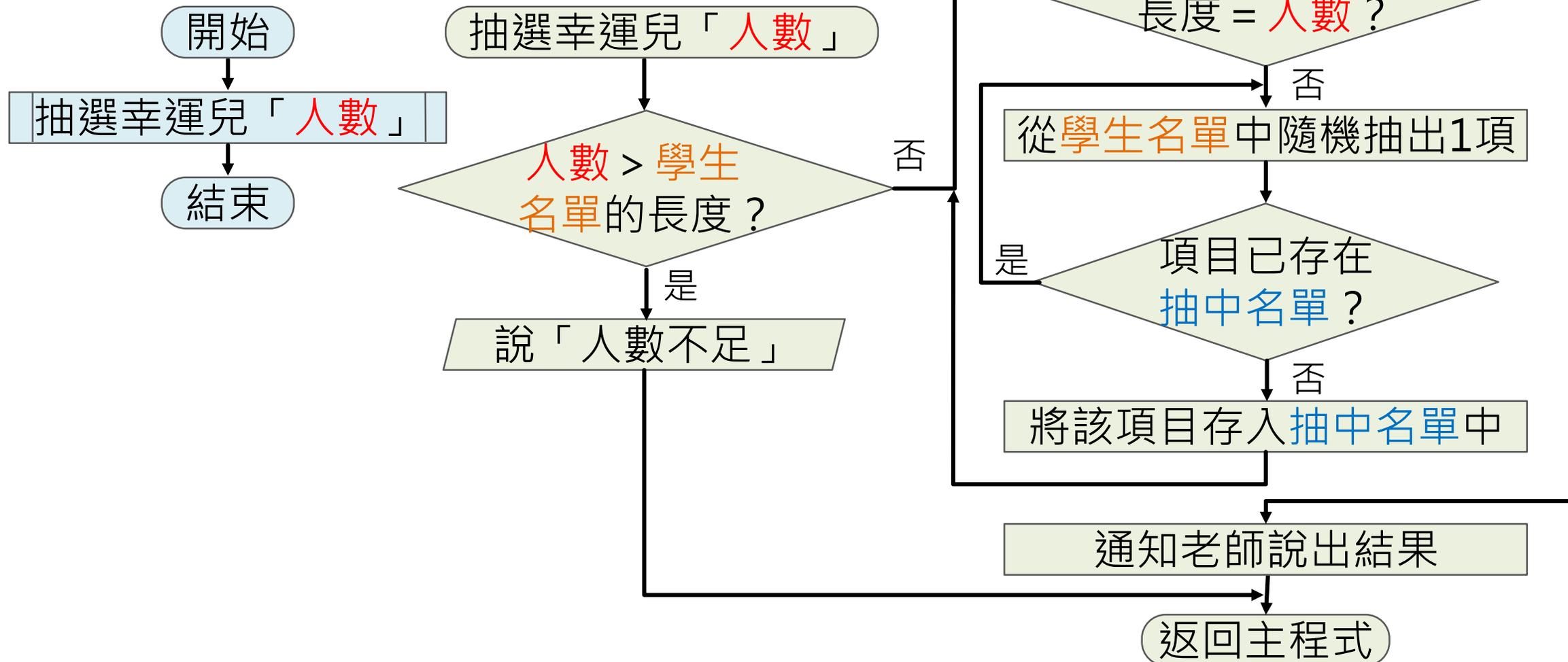
3. 按下 **抽n人** 後：

(1) 請使用者輸入要抽取的人數

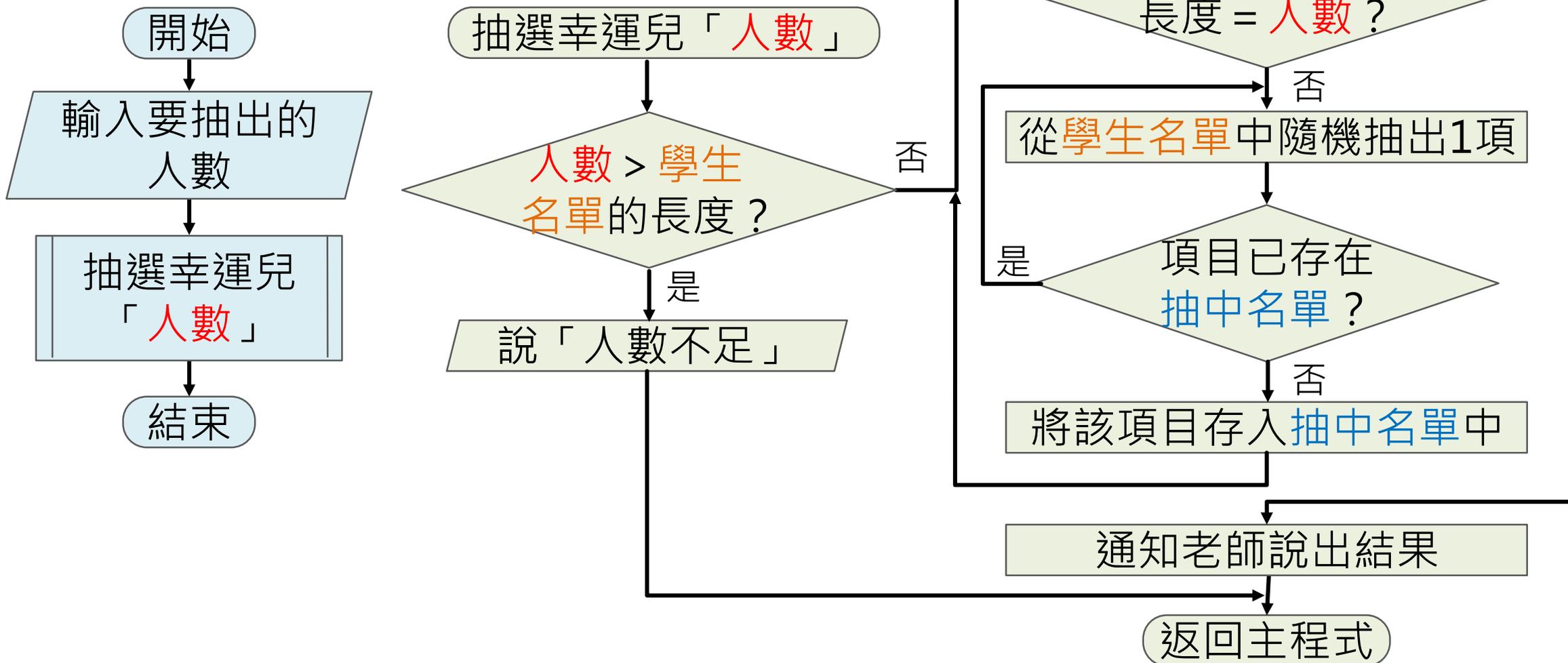
(2) 如果要抽出的人數大於學生名單的長度，代表學生人數不足無法抽籤，說出「人數不足」。



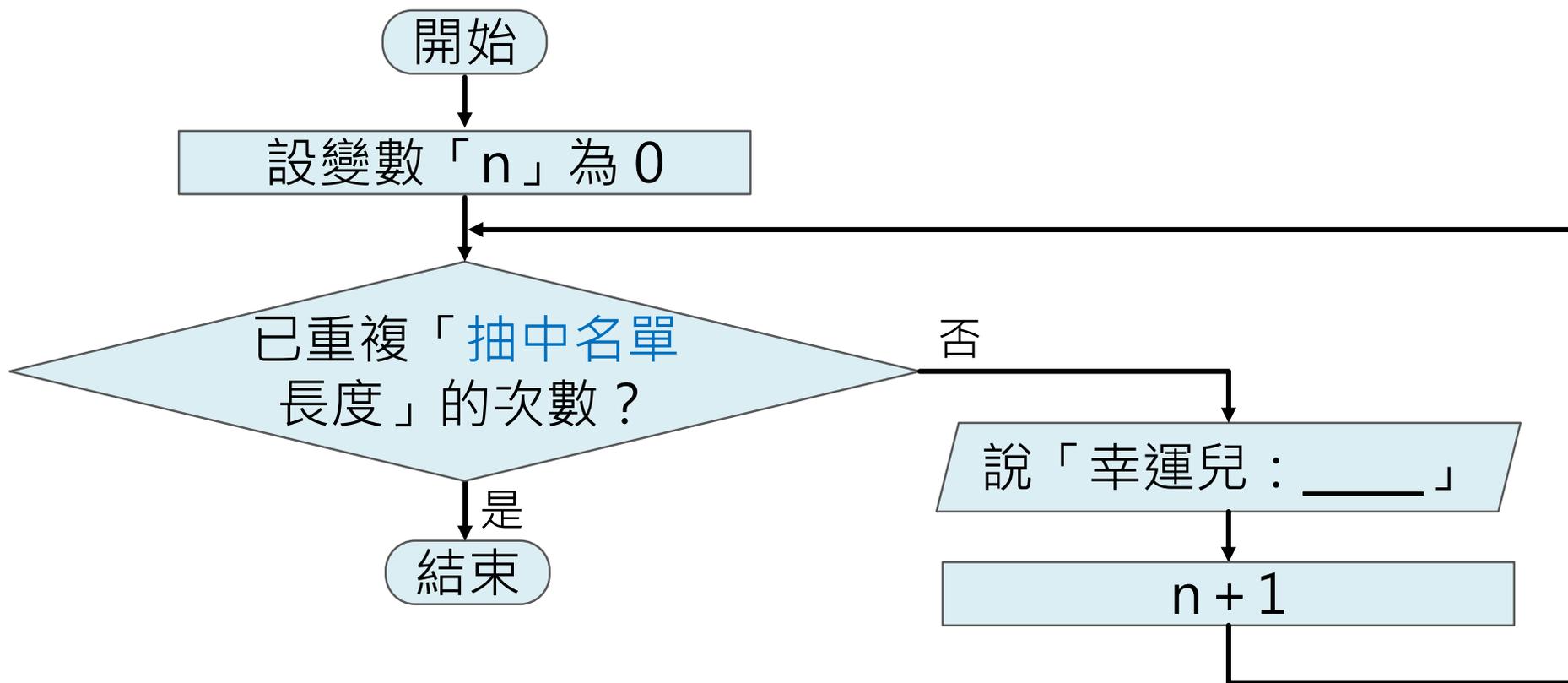
程式流程：角色 **抽1人**、**抽2人**、**抽3人**



程式流程：角色 抽n人



程式流程：角色 老師



解題關鍵：

① 設定變數：



② 加入 / 讀取清單：



③ 條件判斷：



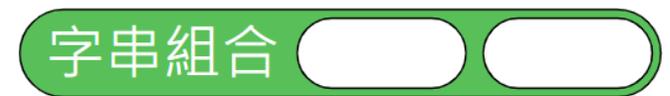
④ 重複執行：



⑤ 函式：



⑥ 說出：



3 · 2 陣列程式—簡易點餐機

結束