

# 4

## 程式應用專題 —— 幸運彩球

### 本章概念

- ▶ 4-1 選號與開獎 P.68  
重複執行程式，直到條件成立  
判斷資料是否重複
- ▶ 4-2 彩球號碼 P.85  
利用造型編號呈現彩球、  
角色分身的使用時機與方法





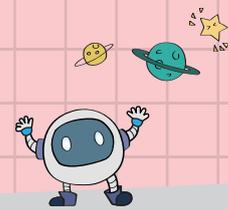
每到過年時，公益彩券的加碼活動總是引起「樂透」風潮，除了增添過節氣氛，而且玩遊戲還能做公益，真是好玩又有趣。

讓我們來小試身手，模仿樂透彩的玩法，自己設計一個「幸運彩球」的遊戲，和家人、朋友同樂，一起來試試自己的運氣吧！



# 4-1

# 選號與開獎



## 任務說明

我們要透過自製的幸運彩球遊戲，讓親朋好友們都來試試手氣。本小節程式主要分為「使用者選號」與「電腦開獎」兩部分。

請播放範例影片 [選號與開獎.mp4](#)，觀察程式的執行情形，一起用 Scratch 來完成任務吧！



### 【程式摘要】



1. 按下 **選號鈕**：

- (1) 利用提問讓玩家進行選號。
- (2) 共要選出 4 個介於 1 ~ 20 之間、不重複的數字。

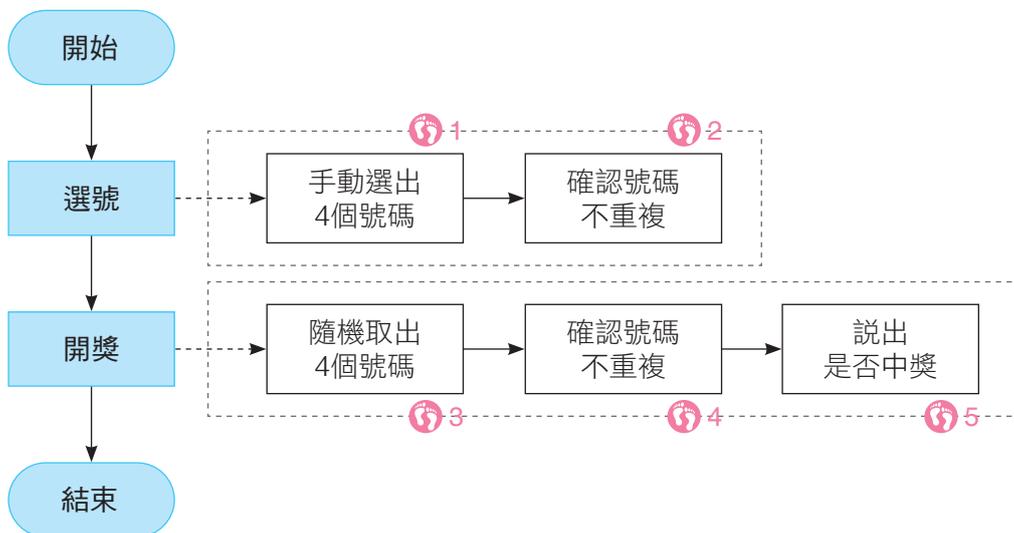


2. 按下 **開獎鈕**：

- (1) 讓程式自動選出 4 個介於 1 ~ 20 之間、不重複的開獎號碼。
- (2) 判斷玩家中了幾個號碼。

### 題目解析流程

搭配【逐步解析】說明



在這一節中，我們將學習利用「重複結構」中的「重複直到」來重複執行程式，並判斷資料是否重複（表 1-4-1）。



## 學習目標

1. 重複執行程式，直到條件成立。
2. 判斷資料是否重複。

表 1-4-1 首次使用的積木

積木類型	外觀	功能
變數類		偵測清單中是否包含特定資料。
控制類		不斷的執行程式區塊 1，直到條件式成立才會停止。
運算類		從指定的範圍中，任意選取一個數字。



## 重複直到

我們已學過「重複無限次」、「重複指定次數」，而在某些情況下，我們不知道程式應該重複幾次，但有明確的中止條件，就需要使用「重複直到」的結構。例如：登入帳號密碼時，如果輸入錯誤就必須重新輸入，不確定會重複幾次，但必須執行到正確為止。

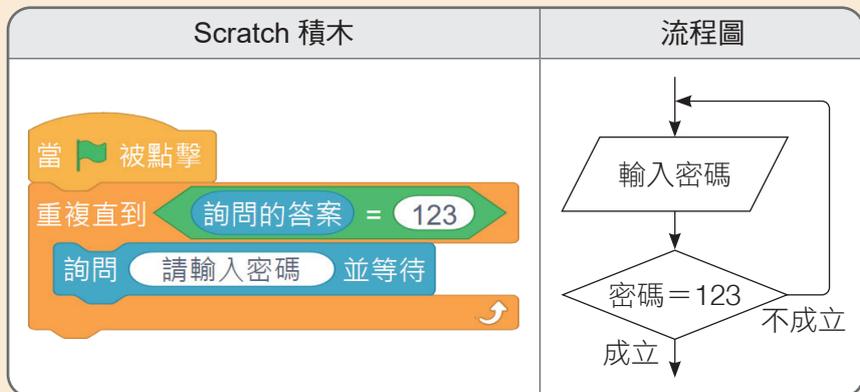
## ● 功能說明

重複結構中的「重複直到」會不斷地執行程式區塊 1，直到條件成立才停止。



## ● 應用介紹

以密碼 123 為例，如果輸入錯誤，就必須重新輸入，直到正確為止。



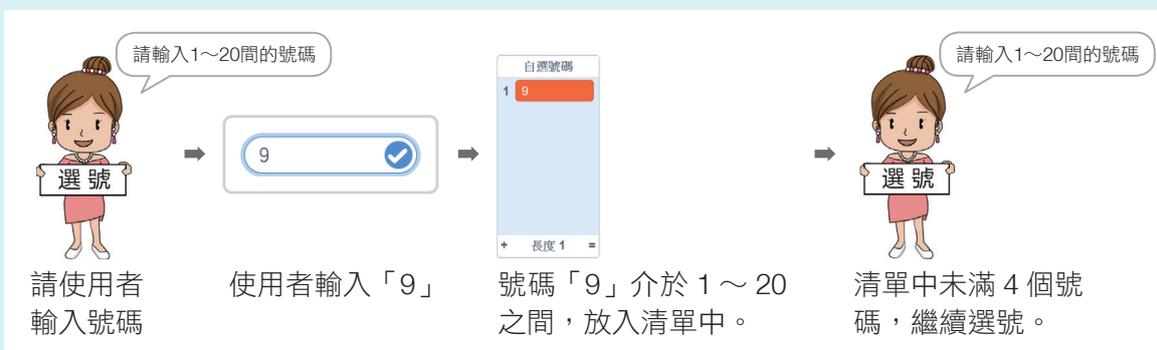


現在，請以 **選號鈕** 為角色撰寫程式：

1. 當角色被點擊時，請使用者輸入號碼。
2. 輸入的號碼必須介於 1 ~ 20 之間：
  - (1) 符合：將號碼存入清單 **自選號碼** 中。
  - (2) 不符合：說出「號碼錯誤」。
3. 共要選出 4 個號碼。



### 狀況 1：輸入的號碼介於 1 ~ 20 之間



### 狀況 2：輸入的號碼不在 1 ~ 20 之間



## 問題思考

Q1

點擊角色  
觸發程式

Q2

設定詢問  
輸入號碼

Q3

對號碼進行  
條件判斷

Q4

共要選出  
4 個號碼

↑ 想一想，如何判斷  
已選出 4 個號碼？

## 解題分析

Q1

點擊角色  
觸發程式

點擊角色開始執行程式：


 當角色被點擊

Q2

設定詢問  
輸入號碼

設定詢問：

利用  詢問 並等待 提問，並等待輸入號碼。

Q3

對號碼進行  
條件判斷

(1) 條件判斷：

① 使用  如果 那麼 進行條件判斷。

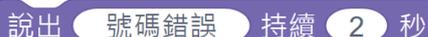
否則

② 判斷「輸入的答案是否介於 1 ~ 20 之間」：

條件式的寫法有很多種，但要留意應使用「且」還是「或」，  
以及執行的動作是放在「成立」還是「不成立」。

情況	條件式 (填寫：且/或)	若成立 (勾選)	若不成立 (勾選)
a	答案 $\geq$ 1 _____ 答案 $\leq$ 20	<input type="checkbox"/> 儲存號碼 <input type="checkbox"/> 提示錯誤	<input type="checkbox"/> 儲存號碼 <input type="checkbox"/> 提示錯誤
b	答案 $>$ 0 _____ 答案 $<$ 21	<input type="checkbox"/> 儲存號碼 <input type="checkbox"/> 提示錯誤	<input type="checkbox"/> 儲存號碼 <input type="checkbox"/> 提示錯誤
c	答案 $<$ 1 _____ 答案 $>$ 20	<input type="checkbox"/> 儲存號碼 <input type="checkbox"/> 提示錯誤	<input type="checkbox"/> 儲存號碼 <input type="checkbox"/> 提示錯誤

(2) 若號碼不在 1 ~ 20 之間：說出「號碼錯誤」。


 說出 號碼錯誤 持續 2 秒

(3) 若號碼在 1 ~ 20 之間：將答案儲存起來。

- ① 共要儲存 4 筆資料 → 建立清單來存放。
- ② 將答案存入清單中：

添加 詢問的答案 到 自選號碼 ▾

註 請將清單安排於舞臺左側，方便檢視清單內容。



不停重複執行程式，直到選出 4 個號碼：

(1) 重複結構：重複直到

(2) 條件式：清單中有 4 筆資料。

清單 自選號碼 ▾ 的長度 = 4

Q4

共要選出  
4 個號碼



## 手腦並用

若使用下圖的程式，有時無法順利選出 4 個號碼，為什麼？



## 【參考程式】



## 手腦並用

試試看，如果使用者輸入的號碼是介於 1 ~ 20 之間的小數，例如：3.14、5.5、9.8 等，會發生什麼事呢？想一想，可以如何修正程式？





在【逐步解析 1】的選號程式中，玩家可能會選出重複的號碼，例如：9、12、9、17。現在我們要修改程式，增加「所選的號碼不能和已選出的號碼重複」這個條件：

1. 號碼不重複：存入清單中。
2. 號碼重複：說「號碼重複」，並重新選號。

### 狀況 1：號碼不重複



### 狀況 2：號碼重複



## 問題思考



判斷號碼  
是否重複

## 解題分析



判斷號碼  
是否重複

判斷「新輸入的號碼是否重複」：

(1) 將「輸入的答案」和「清單中現存的項目」進行比對。

➡ 清單 自選號碼 ▾ 包含 詢問的答案 ?

(2) 號碼重複：說出 號碼重複 持續 2 秒

(3) 號碼不重複：添加 詢問的答案 到 自選號碼 ▾

## 【參考程式】（\*代表本次新增或修改的程式）





在【逐步解析 2】中完成了選號程式，現在到了開獎的時刻啦！請以 **開獎鈕** 為角色撰寫程式：

1. 當 **開獎鈕** 被點擊時開始程式。
2. 如果玩家還沒選出 4 個自選號碼，說「選號未完成」，並結束程式。
3. 若自選號碼已完成，程式隨機選出 4 個介於 1 ~ 20 之間的數字。
4. 為了增加開獎的樂趣，程式每秒開出一個號碼，讓號碼逐一出現。



問題思考

Q1

點擊角色  
觸發程式

Q2

判斷是否  
選號完畢

Q3

選出 4 個  
隨機號碼

Q4

每秒開出  
一個號碼

註 本程式與【逐步解析 1】程式雷同，可以參考【逐步解析 1】的程式加以修改。

解題分析

Q1

點擊角色  
觸發程式

點擊角色開始執行程式：

當角色被點擊

Q2

判斷是否  
選號完畢

條件判斷：

(1) 條件式：**自選號碼** 未滿 4 個。

➡ **清單** **自選號碼** 的長度 < 4

(2) 若未選完自選號碼：說「選號未完成」。

➡ 說出 **選號未完成** 持續 2 秒

(3) 若已選完自選號碼：隨機選出開獎號碼。

## Q3

選出 4 個  
隨機號碼(1) 隨機取出 1 ~ 20 之間的號碼：利用 **運算類** 的 **隨機取數**。→ **隨機取數 1 到 20**

(2) 儲存中獎號碼：

① 建立清單：**開獎號碼**。**註** 請將清單安排於畫面右側，方便檢視清單內容。

② 將取出的數字存入清單中：

**添加 隨機取數 1 到 20 到 開獎號碼**

(3) 共需選出 4 個號碼：

**重複直到 清單 開獎號碼 的長度 = 4**

## Q4

每秒開出  
一個號碼

開出一個號碼之後，等待 1 秒鐘。

→ **等待 1 秒**

## 【參考程式】

```

01 當角色被點擊 Q1
02 刪除 開獎號碼 的所有項目 -- 初始設定
03 如果 清單 自選號碼 的長度 < 4 那麼 Q2
04 說出 選號未完成 持續 2 秒
    否則
05 重複直到 清單 開獎號碼 的長度 = 4
06 添加 隨機取數 1 到 20 到 開獎號碼
07 等待 1 秒 Q4

```



## 手腦並用

在【逐步解析 3】中，雖然開出的號碼都介於 1 ~ 20 之間，但可能會開出重複的數字，例如：5、7、7、11。為了避免號碼重複，我們使用與【逐步解析 2】一樣的判斷方法撰寫程式，如下圖第 06、07 行。程式執行後，還是可能會有重複的號碼被存入清單「開獎號碼」中，想一想，為什麼呢？



## 逐步解析 4 避免開獎號碼重複

接續【逐步解析 3】，現在我們要增加條件「隨機開出的號碼不能重複，才可以存入清單「開獎號碼」中」。



### 問題思考

#### Q1

暫存  
隨機取數

#### Q2

存入不重  
複的號碼

↑ 想一想，為什麼要暫存隨機取數？

## 解題分析

## Q1

暫存  
隨機取數

(1) 概念說明：

**問題** 在手腦並用中，第一次隨機取數（06行）的號碼，與第二次（07行）的取號可能不同。**方法** 可利用變數暫存資料，確保存入的資料與判斷的資料一致。(2) 程式說明：設定變數 **取號**。

判斷新取出的號碼是否重複：

(1) 選擇結構：

**註** 只有號碼不重複時才執行「存入清單」的指令，因此使用單向選擇結構。

(2) 條件式：① 判斷號碼重複：



② 判斷號碼不重複：增加邏輯運算積木

(3) 若號碼不重複：將隨機取出的號碼存入清單 **開獎號碼** 中。

## Q2

存入不重複的號碼

## 【參考程式】（\*代表本次新增或修改的程式）





## 逐步解析

### 5

## 統計對獎結果

接續【逐步解析 4】的程式，選出開獎號碼之後，要說出對獎的結果：

1. 統計對獎結果，說「中了\_\_\_\_\_個號碼」。
2. 若完全沒中，則說「下次再加油！」。



### 問題思考

#### Q1

判斷是否  
對中號碼

#### Q2

說出  
對獎結果

### 解題分析

#### Q1

判斷是否  
對中號碼

每次隨機取出開獎號碼後，判斷「自選號碼」是否中獎。

清單 自選號碼 ▾ 包含 取號 ?

#### Q2

說出  
對獎結果

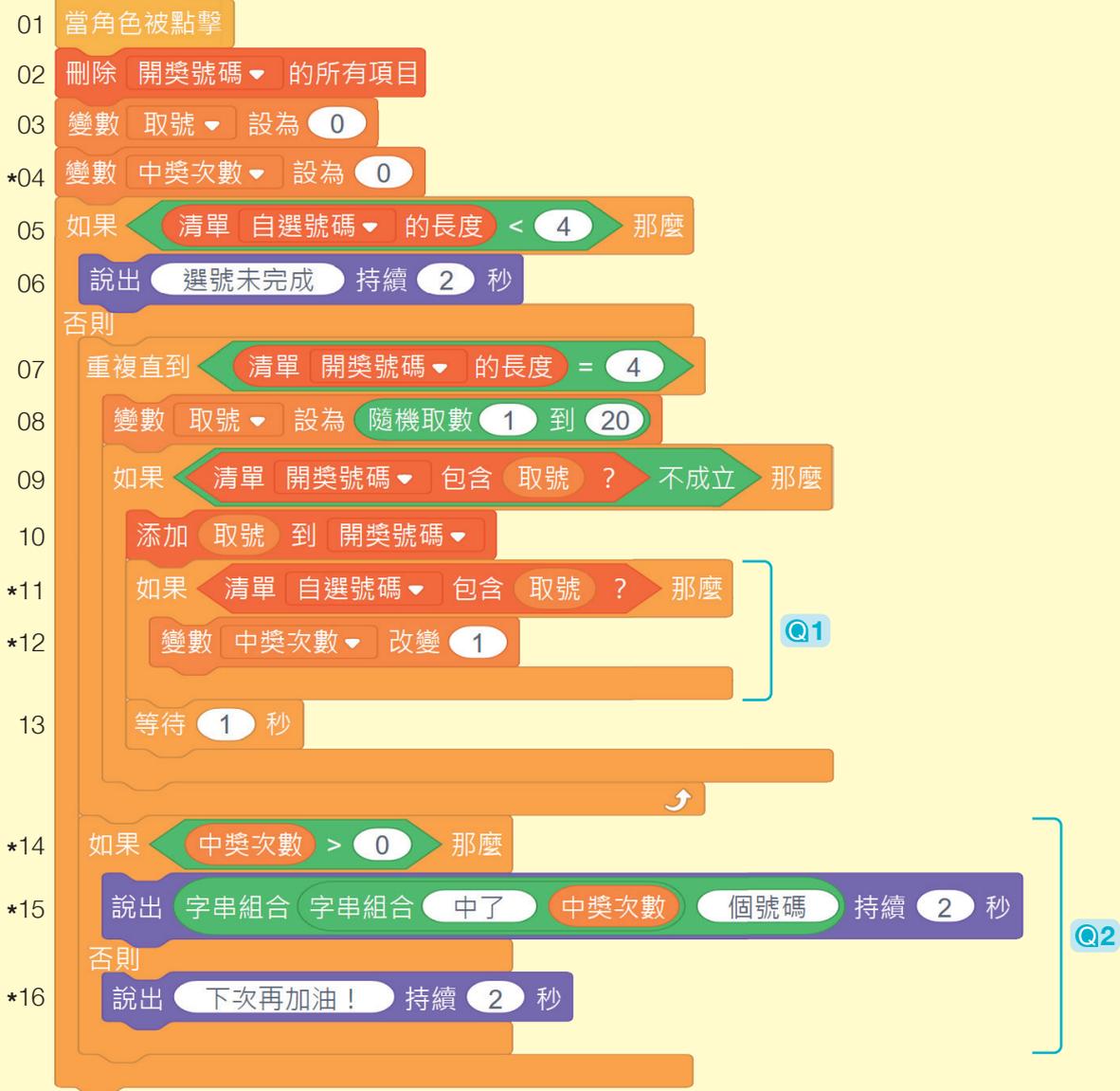
(1) 統計對中幾個號碼：

- ① 設定變數 `中獎次數`，用來統計對中的次數。
- ② 每對中一個號碼，變數的值就「+1」。

變數 中獎次數 ▾ 改變 1

- (2) 若對中的次數為 0，說「下次再加油！」。
- (3) 若有對中號碼，說「中了\_\_\_\_\_個號碼」。

## 【參考程式】（\*代表本次新增或修改的程式）





●問題

在不同的情境下，要選的號碼範圍與數量可能不同，例如大樂透是在 1 ~ 49 號中選出 6 個號碼，而今彩 539 則是在 1 ~ 39 號中選出 5 個號碼。此時若需要更改範圍，就必須逐行檢查程式、一筆一筆地更正。

```

01 當角色被點擊
02 刪除 自選號碼 ▾ 的所有項目
03 重複直到 清單 自選號碼 ▾ 的長度 = 4
04 詢問 請輸入 1~20 間的號碼 並等待
05 如果 詢問的答案 < 1 或 詢問的答案 > 20 那麼
06 說出 號碼錯誤 持續 2 秒
否則
07 如果 清單 自選號碼 ▾ 包含 詢問的答案 ? 那麼
08 說出 號碼重複 持續 2 秒
否則
09 添加 詢問的答案 到 自選號碼 ▾

```

●解決方案

我們可以利用函式將程式通則化，日後要更改選號數或範圍時，只要修改函式的參數即可。

※ 定義函式

```

01 定義 選號 選號數 最小值 最大值
02 刪除 自選號碼 ▾ 的所有項目
03 重複直到 清單 自選號碼 ▾ 的長度 = 選號數
04 詢問 字串組合 請輸入 字串組合 最小值 字串組合 ~ 字串組合 最大值 間的號碼 並等待
05 如果 詢問的答案 < 最小值 或 詢問的答案 > 最大值 那麼
06 說出 號碼錯誤 持續 2 秒
否則
07 如果 清單 自選號碼 ▾ 包含 詢問的答案 ? 那麼
08 說出 號碼重複 持續 2 秒
否則
09 添加 詢問的答案 到 自選號碼 ▾

```

※ 呼叫函式

```

01 當角色被點擊
02 選號 4 1 20 -- 參數 4、1、20，依序代表選號數、最小值、最大值。

```

在 04 行程式中，詢問的內容也包含選號範圍，因此改用字串組合的方式，修改詢問內容，以符合選號範圍。

### ● 設定方法

參數可以將數字或文字從主程式傳遞到函式中，在自定函式時，可依需求增減所需的參數：

- 1 點擊「添加輸入方塊數字或文字」。
- 2 修改參數名稱。
- 3 重複步驟 1、2，直到新增完畢。
- 4 點擊「確定」。



註 也可在函式中「添加說明文字」，作為呼叫函式時的輔助說明。

→ 選號 選號數 最小值 最大值



### 手腦並用

開獎的程式目前固定開出 4 個介於 1 ~ 20 之間的號碼，試試看，將開獎的程式也設定為函式，方便日後進行修改。



## 4-1 小試身手

# 抽抽樂

↓ 檔案 4-1 小試身手 下載方式請見 P.3

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！

請以檔案 `4-1 小試身手.sb3` 來撰寫程式。

**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」。

# 4-2 彩球號碼

## 任務說明

在 4-1 節中，我們利用清單來儲存號碼，完成了選號與開獎程式。本節中，我們要利用彩球來表現選號的結果，讓程式的呈現更加生動有趣。

請播放範例影片 `彩球號碼.mp4`，觀察程式的執行情形，一起用 Scratch 來完成任務吧！

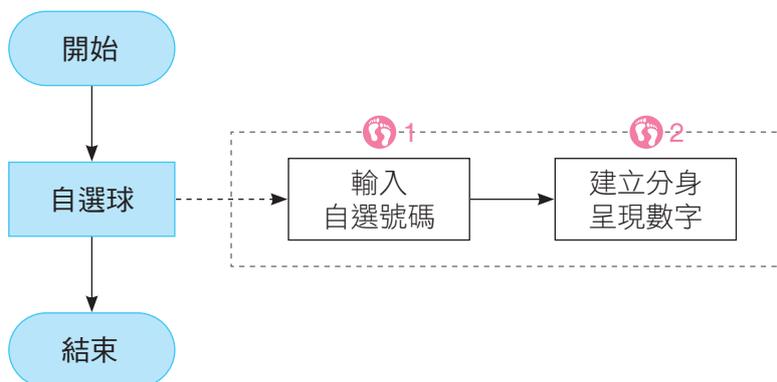
### 【程式摘要】

1. 利用角色 `自選球` 呈現開獎過程。
2. 隨著選號的過程，逐一顯示選出的自選球。



### 題目解析流程

搭配【逐步解析】說明



在這一節中，我們將學習利用「切換造型」的方式來顯示選號的號碼；再進一步利用「角色分身」的功能，以彩球呈現選號的號碼（表 1-4-2）。



### 學習目標

1. 利用造型編號呈現彩球。
2. 角色分身的使用時機與方法。

表 1-4-2 首次使用的積木

積木類型	外觀	功能
控制類		建立分身後，開始執行下方的指令積木。
		建立角色的分身。
		刪除角色的分身。



### 逐步解析

#### 利用彩球造型呈現號碼

使用檔案 4-2-1.sb3

在正式修改程式前，我們先以檔案 4-2-1.sb3 來試試看，如何讓彩球隨著選號而呈現出編號。

在角色「彩球」中含有 21 種造型，分別標示「數字 1~20」及「？」。請撰寫程式：

1. 當「彩球」被點擊時，請使用者輸入 1~20 間的號碼。
2. 使用者輸入號碼後，「彩球」呈現出該號碼。



造型名稱	1	2	3	...	19	20	21
造型外觀				...			

註「？」為彩球的初始造型。

## 問題思考

Q1

被點擊時請使用  
者輸入號碼

Q2

依據號碼  
顯示造型

## 解題分析

Q1

被點擊時  
請使用者  
輸入號碼

(1) 點擊角色開始執程式：當角色被點擊

(2) 請使用者輸入號碼：詢問 並等待

Q2

依據號碼  
顯示造型

(1) 要呈現的號碼：來自 詢問的答案

(2) 選擇呈現的造型：造型換成

註 由於本程式中，角色的造型編號和呈現的彩球號碼是對應的，所以可以直接利用號碼來切換造型。

## 【參考程式】

- 01 當角色被點擊
- 02 詢問 請輸入1~20間的號碼 並等待
- 03 造型換成 詢問的答案

Q1

Q2

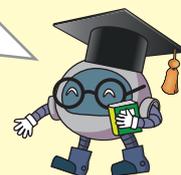
本程式利用輸入的答案來切換造型，若資料存在清單中，也可以利用讀取清單的方法來切換造型。

以彩球呈現清單中的第一個數字為例：

造型換成 自選號碼 的第 1 項

9

自選號碼	
1	9
2	8
3	4
4	7
+ 長度 4 =	



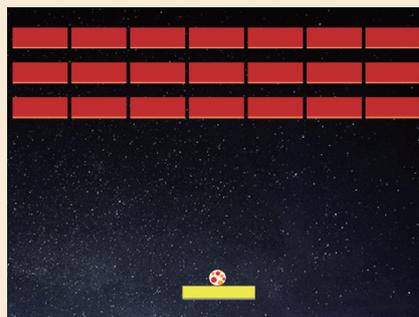


## 角色分身

有些電玩遊戲中會重複出現許多相同的角色，像是「打磚塊遊戲」中有許多磚塊，而「飛機射擊遊戲」中有可以連續發射的子彈。

以打磚塊遊戲為例，每一個磚塊的程式都相同，但排放的位置不同。雖然我們可以先撰寫好一個磚塊的程式，再複製產生其他磚塊，但每一個磚塊都要逐一修改定位，將是一件非常麻煩和耗時的工作。

在 Scratch 中，我們可以利用「分身」的功能來複製出其他分身。分身會繼承原角色的所有程式、屬性，而且角色與分身是各自獨立的，彼此的任何改變不會互相影響。例如打磚塊遊戲中，我們利用一個磚塊產生許多磚塊的分身，當其中一個分身被球擊中而消失（隱藏），不影響原角色或其他分身的顯示狀態。



### ●分身程式積木的使用

在 Scratch 的 **控制類** 積木中，有三個角色分身程式積木：

1. **建立 自己 的分身**：可以建立自己的分身，也可以建立其他角色的分身（如右圖）。

**註** 若在 **小貓咪** 的程式中建立 **蘋果** 的分身，此時 **小貓咪** 的程式無法控制 **蘋果** 分身的行為，只能在 **蘋果** 角色的程式裡控制及刪除。

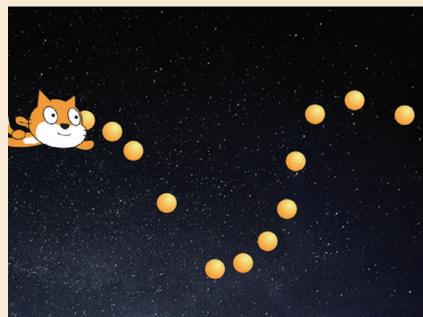
2. **當分身產生**：撰寫在角色中，當分身產生之後，分身會執行下面的程式。
3. **分身刪除**：可以將畫面上的分身刪除，而原角色不受影響。

建立 自己 的分身

✓ 自己  
小貓咪  
蘋果

## ●應用範例

以檔案「飛貓子彈.sb3」為例，點擊「綠旗」開始執行程式後，「貓咪」會隨著滑鼠游標移動；當我們點擊滑鼠（相當於點擊「貓咪」）就可以發射「子彈」，連續點擊可連續發射，「子彈」會不斷向右移動，在碰到舞臺邊緣時才消失。其程式說明如下：



## ●貓咪

- 01 當「綠旗被點擊」時，利用滑鼠游標操控「貓咪」移動。
- 02 重複無限次。
- 03 定位到「鼠標」位置。

- 01 當角色被點擊時，在「貓咪」的角色中，建立「子彈」的分身。
- 02 建立「子彈」的分身。

## ●子彈

- 01 當「綠旗被點擊」時，把「子彈」的本尊隱藏起來。
- 02 隱藏。

- 01 當分身產生時，控制「子彈」的分身。
  - 02 定位到「貓咪」位置。
  - 03 顯示。
  - 04 重複直到「碰到邊緣？」。
  - 05 x 改變 5。
  - 06 分身刪除 -- 碰到邊緣後，刪除這個「子彈」的分身，其他分身不受影響。
- 子彈的分身要從「貓咪」的位置發射
- 子彈不停向右移動



## 逐步解析 2

### 利用彩球分身呈現自選號碼

現在，我們要接續 4-1 節【逐步解析 2】的程式，修改角色 **自選球** 與 **選號鈕** 的程式，當使用者選號時，逐一利用「分身」的功能，讓彩球出現在畫面上：

1. 點擊 **綠旗** 開始程式，或點擊 **選號鈕** 時，清除畫面上的彩球。

想一想，**自選球** 要怎麼知道 **選號鈕** 被點擊了？



2. 自選號碼存入清單的同時，**自選球** 的分身出現於畫面上「自選號碼區」。  
(每選一個號碼，就出現一個彩球)

### 問題思考

#### Q1

設定彩球  
初始狀態

#### Q2

產生分身

### 解題分析

#### Q1

設定彩球  
初始狀態

(1) 回復初始狀態的時機：

① 程式開始時、選號開始時：當 **綠旗** 被點擊、當收到訊息 **選號** ▾

② 角色 **選號鈕** 被點擊時：當角色被點擊

→ **選號鈕** 要發出通知：廣播訊息 **選號** ▾

(2) **自選球** 的初始狀態：

- ① 定位到正確的位置。
- ② 隱藏。
- ③ 刪除已經建立的分身。

#### Q2

產生分身

(1) 產生分身的時機：

① 自選號碼正確且不重複，放入清單時。

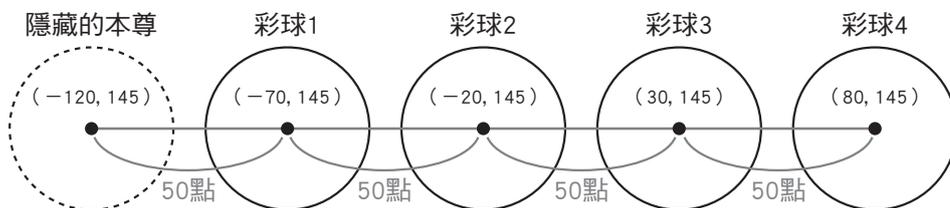
→ 程式要寫在 **選號鈕** 身上。

② 方法：建立 **自選球** ▾ 的分身

(2) [自選球] 的分身產生時：

① 定位到正確位置。

- a. 起始坐標：(-120, 145)
- b. 隨著開出的號碼個數調整位置，每顆彩球相距 50 點。



c. 分身排列位置計算：

$-120 + (50 \times \text{第幾個彩球})$  → 想一想，還有什麼算法？

② 呈現正確的號碼。

→ 想一想，如何知道現在是「第幾個彩球」？

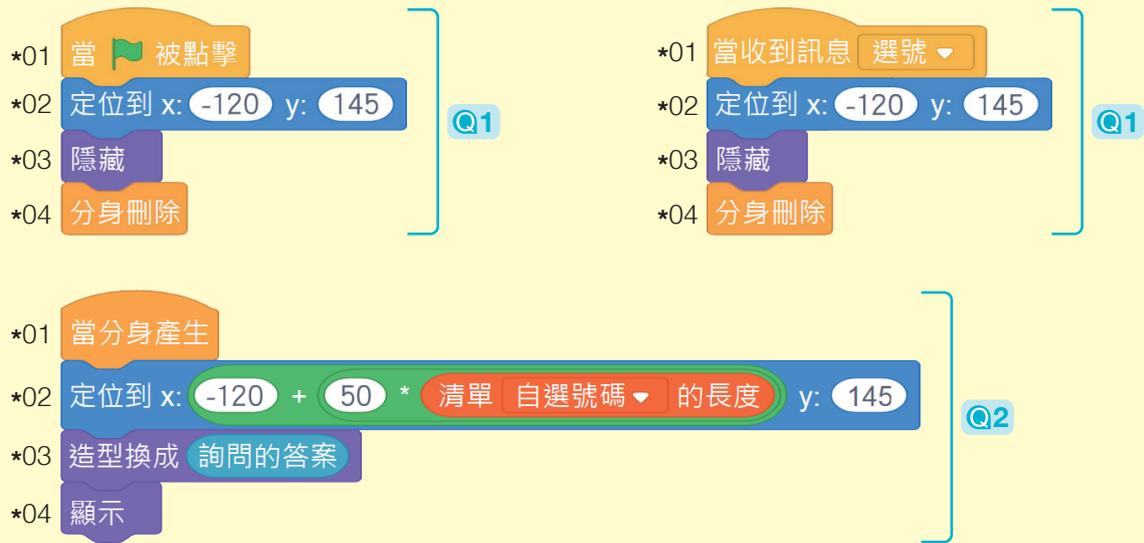
③ 狀態由隱藏變為顯示。

### 【參考程式】 (\* 代表本次新增或修改的程式)

#### ● 選號鈕



## ●自選球



## 手腦並用

1. 開獎球的程式邏輯與自選球大致相同，請自行完成開獎球的程式。
2. 在確認自選球、開獎球的呈現均正確後，可將清單自選號碼、開獎號碼設為隱藏，讓遊戲畫面更加清爽！



4-2  
小試身手

## 幸運拉霸

↓ 檔案 4-2 小試身手 下載方式請見 P.3

本節中，我們學會了如何根據清單的資料來決定角色的造型，現在，我們就要利用這個技巧來製作另一個經典遊戲——幸運拉霸。

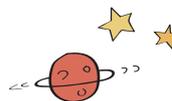
請以 4-2小試身手.sb3 撰寫程式，讓大家一起來碰碰運氣。



1. 程式中已設定 5 個必要角色：`水果 1 ~ 3`、`Start 鈕`、`字幕`。
2. 按下 `Start 鈕` 後，`水果 1 ~ 3` 開始變換造型。
3. 水果共有 5 種造型，變換一段時間後，會隨機停在某個造型。
4. `字幕`：
  - (1) 開獎過程中，顯示「努力開獎中～」。
  - (2) 若出現 3 個相同的水果，顯示「恭喜中獎！」。
  - (2) 若未中獎，則顯示「下次再加油」。



## 第4章 學習重點



### 4-1 選號與開獎

#### 任務要求

- 使用檔案 4-1-1.sb3 撰寫程式。
- 按下 **選號鈕** 後，玩家要在 1 ~ 20 號中，選出 4 個不重複的號碼。
  - 若超出範圍，說「號碼錯誤」，並繼續選號。
  - 若號碼重複，說「號碼重複」，並繼續選號。
- 按下 **開獎鈕** 後，程式會在 1 ~ 20 號中，隨機取出 4 個不重複的開獎號碼，並判斷玩家是否中獎。



#### 問題思考

- 如何判斷使用者輸入的號碼「在範圍內且不重複」？
- 如何儲存合格的號碼？
- 如何重複執行程式，直到選完所有號碼？
- 如何隨機取出不重複的開獎號碼？
- 如何判斷玩家是否中獎？

#### 解題分析

Q1. (1) 要判斷「是否在範圍內」與「是否重複」，會使用到兩個「選擇結構」。

(2) 條件判斷：

① 號碼不在 1 ~ 20 的範圍內，表示號碼錯誤，需重選。



② 號碼與清單中的重複，需重選。



Q2. (1) 要儲存多個號碼，因類型相同，可儲存在 1 個清單中。

(2) 將資料存入清單：**添加**  **到** **自選號碼**

Q3. 重複執行直到條件成立：**重複直到** **清單 自選號碼 的長度 = 4**



Q4. 取出開獎號碼：

- (1) 隨機取號：運算類的 隨機取數 到
- (2) 設定取號範圍：1 ~ 20
- (3) 設定變數暫存取號：取號

Q5. 判斷是否中獎：清單 自選號碼 包含 取號 ？

## 4-2 彩球號碼

### 任務要求

1. 接續 4-1 節程式。
2. 利用角色 自選球 建立分身：
 

隨著選號的過程，逐一顯示該號碼的彩球。

### 問題思考

- Q1. 如何建立角色分身？
- Q2. 如何定位分身的坐標？
- Q3. 如何讓彩球依號碼呈現不同的數字？



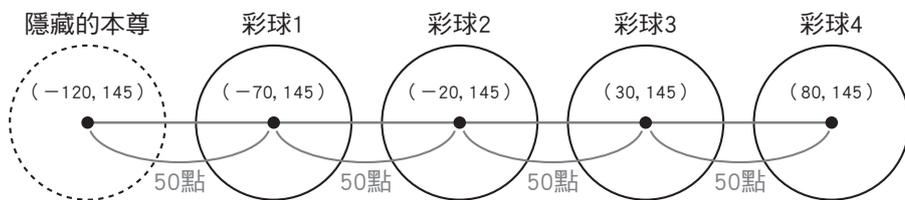
### 解題分析

Q1. 建立分身：

- (1) 建立時機：使用者輸入的號碼存入清單 自選號碼 時。
- (2) 建立方法：建立 自選球 的分身

Q2. 定位分身的坐標：

- (1) 定位方法：定位到 x: y:
- (2) 隨著選出的號碼個數調整位置，每顆彩球相距 50 點。



- (3) 分身排列位置計算： $-120 + (50 \times \text{第幾個彩球})$

Q3. 依號碼呈現數字：

彩球的造型編號與數字對應，可利用 造型換成 來呈現。

# 電子試算表始祖 VisiCalc

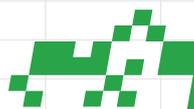
在本章中，我們利用清單（陣列）來儲存號碼，並進行資料的比對，而我們平常進行資料處理使用的「試算表」，也是以陣列的方式呈現。Scratch 的清單只有「序號」這一個索引值，是一維陣列；試算表有「欄」、「列」兩個索引值，屬於二維陣列。

## 試算表的誕生

世界上第一個試算表軟體「VisiCalc」誕生於西元 1979 年，由丹·布瑞克林（Dan Bricklin，西元 1951 年～）與包伯·法蘭克斯頓（Bob Frankston，西元 1949 年～）共同開發。

丹·布瑞克林在就讀哈佛商學院時，每週都必須進行商業案例分析，過程中有大量的資料要整理成表格，並依公式逐一計算。如果當中有一個數據錯了，就必須擦掉好幾個相關的欄位，再全部重新計算。

丹·布瑞克林在一次次的修改過程中開始構思，如果可以利用電腦自動處理表格中的計算，就能節省大量的時間。於是邀請程式設計師好友包伯·法蘭克斯頓共同開發出 VisiCalc 這個可以模擬表格計算的試算表軟體。



丹·布瑞克林。

№	Наименование OC	Амортиз. на месяц		Оборот за месяц на рубль		Выпуск на рубль		Усред. % /год	Сумма на 1 рубль
		кол.	сумма	кол.	сумма	кол.	сумма		
1	Затраты 2-й этажности на рубль	-	-	1	20000	-	-	0,1	200 <sup>0</sup>
2	ремонт	-	-	2	18000	1	9000	0,42	756 <sup>0</sup>
3	ремонт	-	-	-	30000	-	90000	-	956 <sup>0</sup>
4	Амортизация на рубль	-	-	1	20000	-	-	1,2	2400 <sup>0</sup>
5	ремонт	-	-	1	5000	-	-	0,69	600 <sup>0</sup>
6	ремонт	-	-	1	8000	-	-	0,46	61 <sup>0</sup>
7	ремонт	-	-	-	58000	-	-	-	661 <sup>0</sup>
8	Бюджет	-	-	-	48000	-	90000	-	6480 <sup>0</sup>

在電子試算表出現之前，必須仰賴紙筆計算。

ITEM	NO.	UNIT	COST
MUCK RAKE	4	1	5
BUZZ CUT	1	1	156
FOE TONER	25	4	124
EYE SNUFF	2	40000	80000
SUBTOTAL			13155.50
9.75% TAX			1282.66
TOTAL			14438.16

VisiCalc 試算表畫面。VisiCalc 為「看得見 (visible) 的計算 (calculate)」之意。

在早期，商程式都以大型計算機的應用為主，例如庫存系統、薪資系統等，不但只有大型企業才有這樣的設備，就連操作上也較為困難，一般民眾根本沒有機會使用。

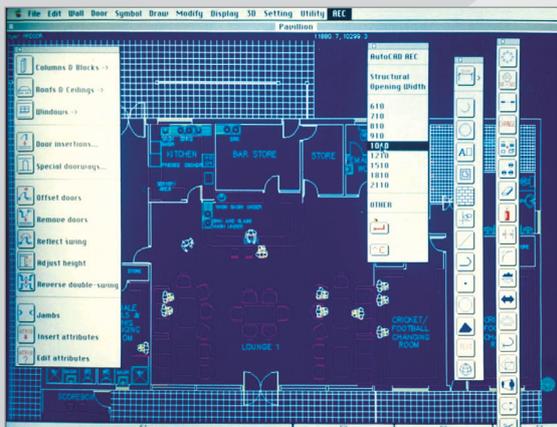
VisiCalc 則是在「蘋果電腦 Apple II」上發行，是個人電腦上的第一個試算表軟體，簡便的操作方式讓一般人也能順利操作，有許多使用者為了 VisiCalc 而購買 Apple II，直接帶動蘋果電腦的購買風潮，因此有了「殺手級應用」的稱號。



▲ 蘋果電腦 Apple II。

## 殺手級應用 (killer application)

殺手級應用指的是「利用創新與科技帶動社會迅速改變的革命性服務」，這樣的服務通常能創造龐大的市場利益。西元 2012 年，美國知名科技雜誌《eWeek》曾評選電腦軟體領域的殺手級應用，入選的除了本文提到的 VisiCalc，還有我們常用的文書處理軟體 Microsoft Word、電腦輔助設計軟體 AutoCAD、音樂軟體 iTunes、網路瀏覽器 Internet Explorer 等。



▲ AutoCAD 擺脫紙筆繪製的設計流程，改變了機械、建築、工業設計、室內設計等行業的作業方式。



▲ iTunes 改變違法下載音樂的亂象，建立合法付費的線上數位音樂經營模式，也改變了唱片市場的生態。