

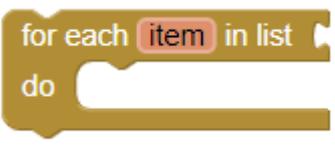
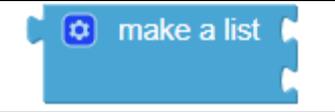
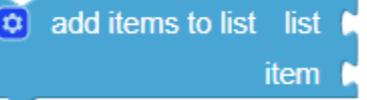
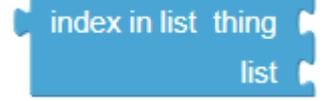
# Unit 5 List 清單介紹

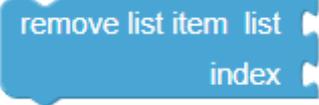
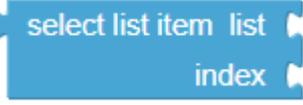
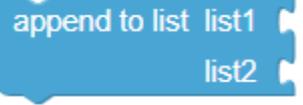
作者：蕭志翔

## 本章介紹

本章主要介紹 List 指令及其概念。List 就是一般程式中的陣列是一種資料結構，用來儲存一群相同資料型態的元素，而每一個元素都佔有自己的記憶體位置，通常是連續的，每個元素也都有一個編號能夠對應他們，稱為索引 (index)。簡單來說陣列就像是一個分類好的櫃子，例如你設定他是放衣物的，而你將襪子放在第一格，衣服第二格、褲子第三格，你就能透過這些位子找到你要的東西。陣列的另一個功用就是為資料留一個儲存空間，當你不需要某些資料時就能將其清空，就不會讓用不到的資料留在你的程式中占空間並拖慢執行速度，而 List 的編號 (index) 是從 1 開始，一般程式的陣列則是從 0 開始。

## 認識本範例使用的拼圖和其基本定義

Control 指令區	
	For 迴圈的延伸，在程式中宣告變數清單，依清單的內容元素數量，決定迴圈執行次數。
List 指令區	
	產生一個空的清單。
	產生一個清單並可以新增內容元素進去，如果沒新增任何內容元素，則為一個空的清單。
	可以將內容元素加入指定清單後面。
	用來回傳清單的長度。
	用來指定項目在清單中的編號。
	將指定的元素內容放到清單中的指定編號。

	隨機從清單中選一個項目。
	從清單中刪除指定的編號。
	取得清單中的指定編號內容。
	將兩個清單組合成一個新的清單。

### For each item 迴圈介紹

For each item 迴圈是基於 For 迴圈所做的延伸，以在程式裡宣告變數清單，並根據此清單內的元素數量決定執行次數的迴圈。



For each item 迴圈會連結一個選單 ( list )，並透過選單中的元素內容 ( item ) 來決定迴圈執行次數。假設今天我的 list 裡有 7 筆資料，分別是星期一到星期日，那麼迴圈在執行時就會執行七次。

特別要注意的地方是 For each item 迴圈在新增時，就會自動生成一個**區域變數 item**，此區域變數主要是用來讓使用者取得選單中的元素內容，並在程式的設定下顯示出來。如果對全域變數及區域變數有不了解的地方，可以透過前面的章節來認識全域變數及區域變數的不同。



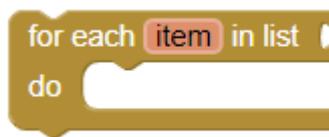
自動生成的區域變數 item

## For each item 迴圈方塊設定

For each item 迴圈方塊是在內建方塊(Blocks)項目點選流程控制(Control)，選取 For each item 拼圖。



我們可以透過點選深色的 item 格子，進行區域變數名稱的更改。



更改名稱前



更改名稱後

## Create empty list / make a list 清單介紹

Create empty list 主要是新增一個空白的清單，在這空白的清單中，並沒有任何的元素內容 ( item )。



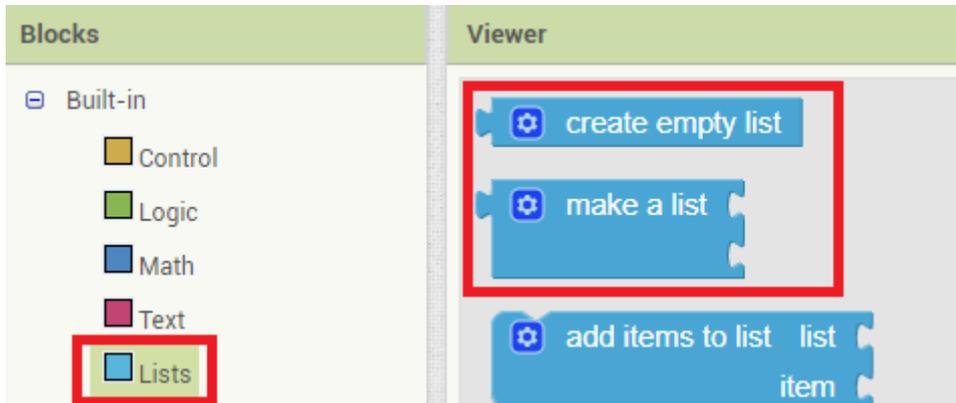
而 make a list 主要是新增一個可以自己指定元素內容 ( item ) 的選單，如果沒有新增元素內容 ( item )，則是一個空白的清單。



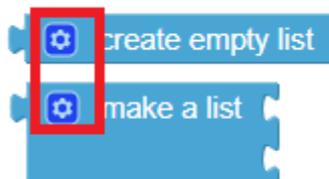
其實 Create empty list 與 make a list 並沒有太多的差別，如果在 Create empty list 上新增元素內容 ( item )，那麼他就會自動變成 make a list。若將 make a list 的元素內容 ( item ) 全部刪除，則會變成 Create empty list 空白清單，以上將在下面的拼圖設定做示範。

## Create empty list / make a list 清單方塊設定

Create empty list 與 make a list 清單方塊兩個都是在內建方塊(Blocks)項目中，點選清單(Lists)，選取 **Create empty list** 與 **make a list** 拼圖。



我們可以透過 **Create empty list** 與 **make a list** 方塊上的齒輪，進行元素內容 ( item ) 的新增和移除，當我們在 **Create empty list** 新增元素內容 ( item ) 後，它會變成 **make a list**。



方塊上的齒輪位置



將 item 新增至 list 中



新增完成後，變成 **make a list**

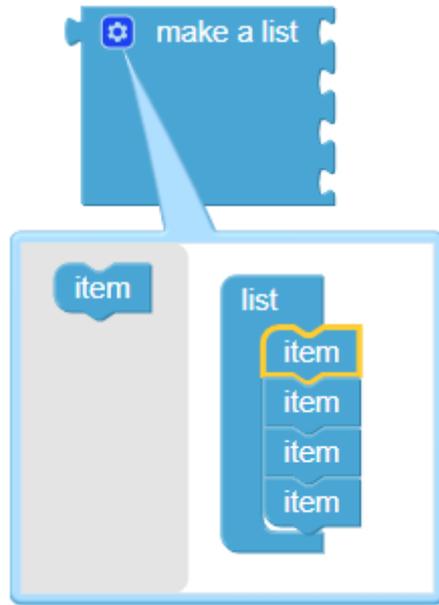
如果我們將 **make a list** 的元素內容 ( item ) 移除後，則會變成 **Create empty list** 空白清單，而我們也能透過新增更多元素內容 ( item )，來增加格子。



將元素內容 ( item ) 從 list 拉走移除



list 空白後變成 **Create empty list** 空白清單

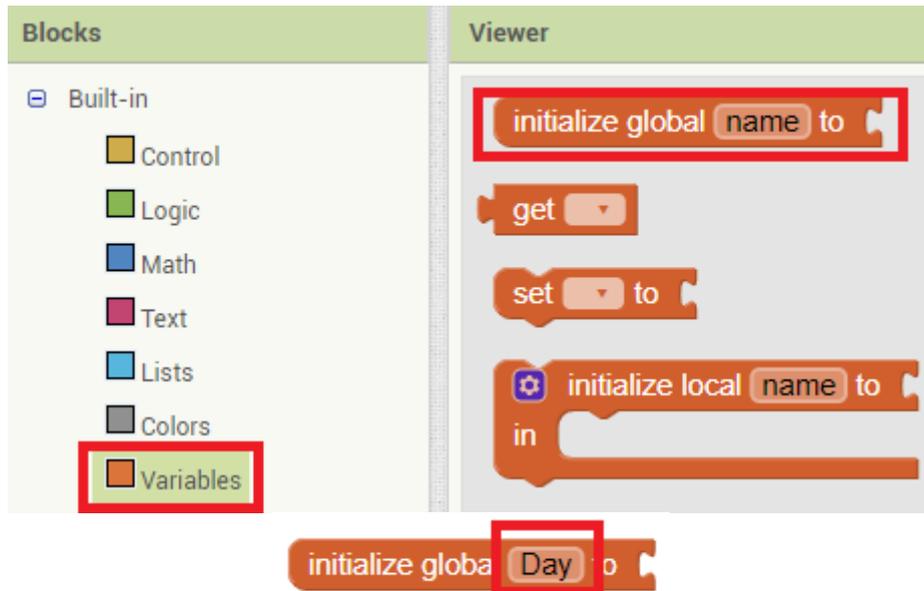


新增元素內容 ( item ) 增加格子

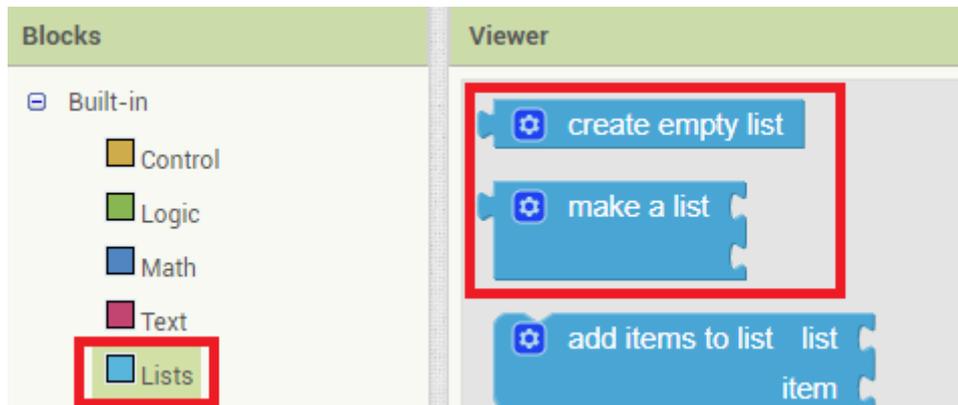
## 範例說明

請跟著範例做出一個 List 清單，用來存放星期一到星期五，並用 For each item 來依序列出來，為方便測試我們需要新增 Button 及 Label 來觸發與顯示。

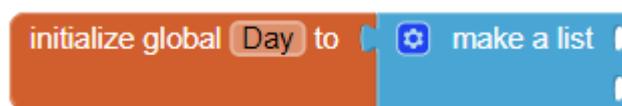
先從內建方塊(Blocks)項目中的變數(Variables)新增一個全域變數，並取名為 Day。



接下來找到 Lists 新增一個 List 清單，不管是 **Create empty list** 或 **make a list** 任何一個都可以。



將全域變數 Day 與清單組合在一起，再新增星期一到星期五到 List 清單中。

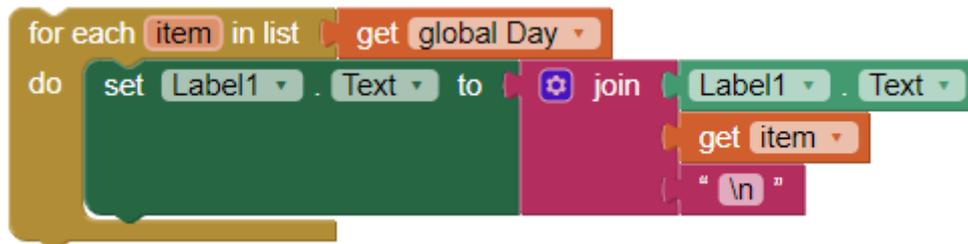




接下來從流程控制(Control)中，新增一個 **For each item** 拼圖。



設定 **For each item** 取得全域變數 Day，並在 Label 上顯示，而顯示的內容需要先顯示 Label.Text，這樣才可以每次執行迴圈時，顯示出當次迴圈的內容。接下來要注意我們要顯示的是區域變數 item，而非全域變數 Day，最後記得在文字內容後面加上\n 換行。



最後我們將此方塊拼圖與 Button.Click 做結合並觀看測試結果吧。





測試開始畫面



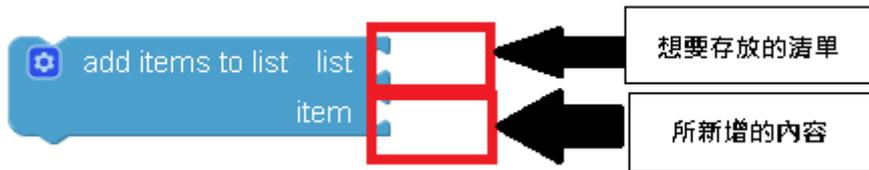
測試結果畫面

## 活動

試著不使用 For each item，直接用 Label 顯示全域變數 Day 看看測試結果會有什麼不一樣的變化。

## Add items to list / length of list 介紹

**Add items to list** 用於新增元素內容 ( item ) 到 list 選單中，讓使用者可以透過此指令，將輸入的東西設定為元素內容 ( item ) 並儲存於選單中。

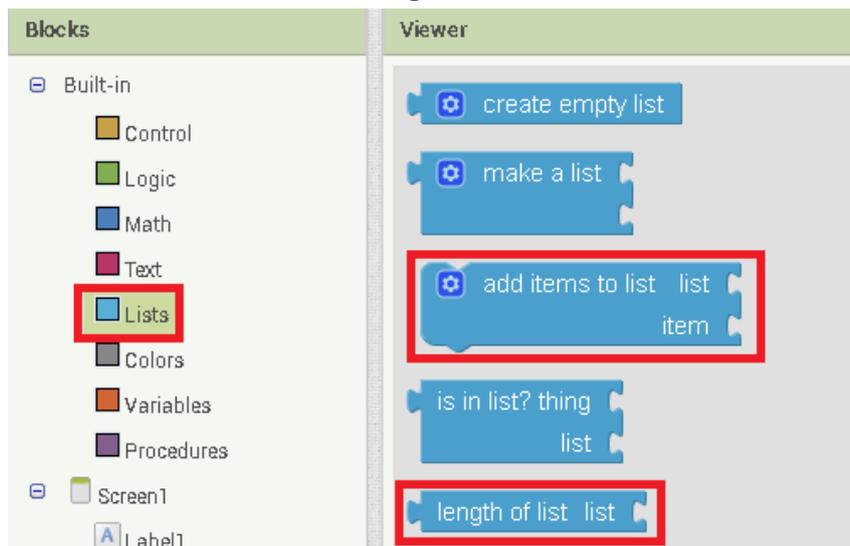


**length of list** 則是用來回傳 list 清單的長度，也就是清單中所儲存的元素內容 ( item ) 數量。範例一為儲存星期一到星期五的清單，而此清單的長度就為 5。

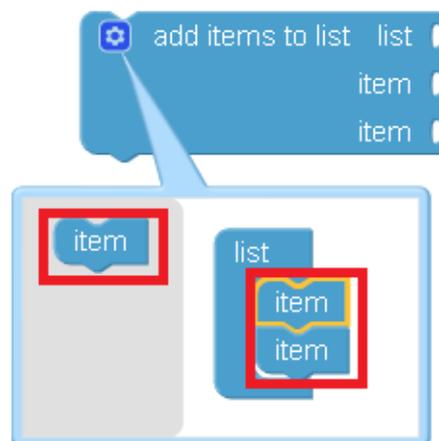


## **Add items to list / length of list 方塊設定**

**Add items to list** 與 **length of list** 兩個方塊都在內建方塊(Blocks)項目中，點選清單(Lists)，選取 **add items to list** 與 **length of list** 拼圖。



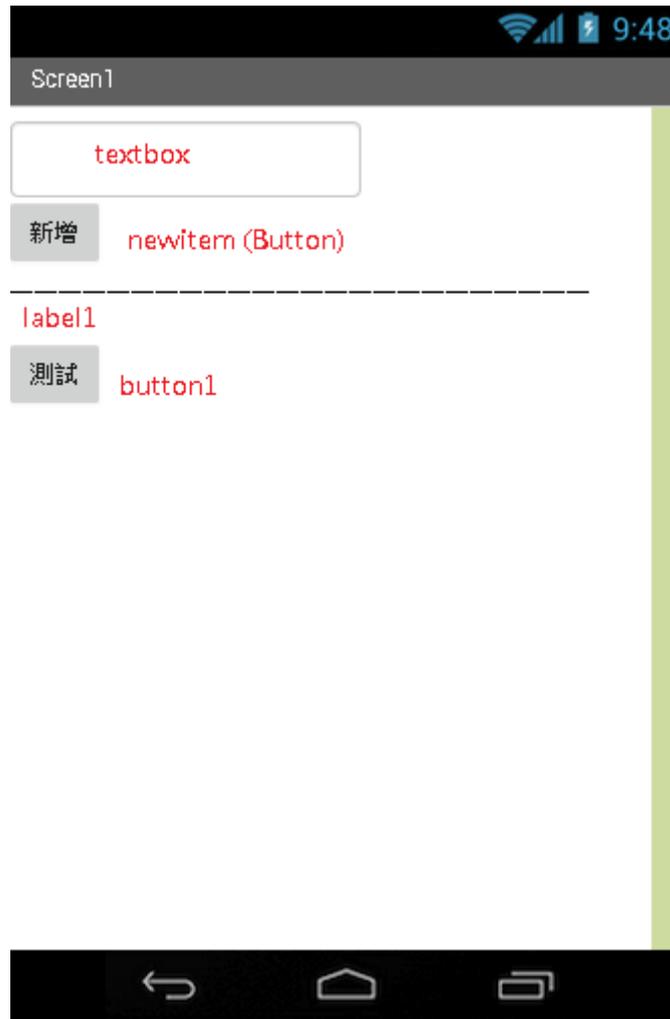
而 **add items to list** 可以透過點選上方齒輪，將 item 拖移至 list 清單中，這樣就可以一次新增多個元素內容 ( item )。



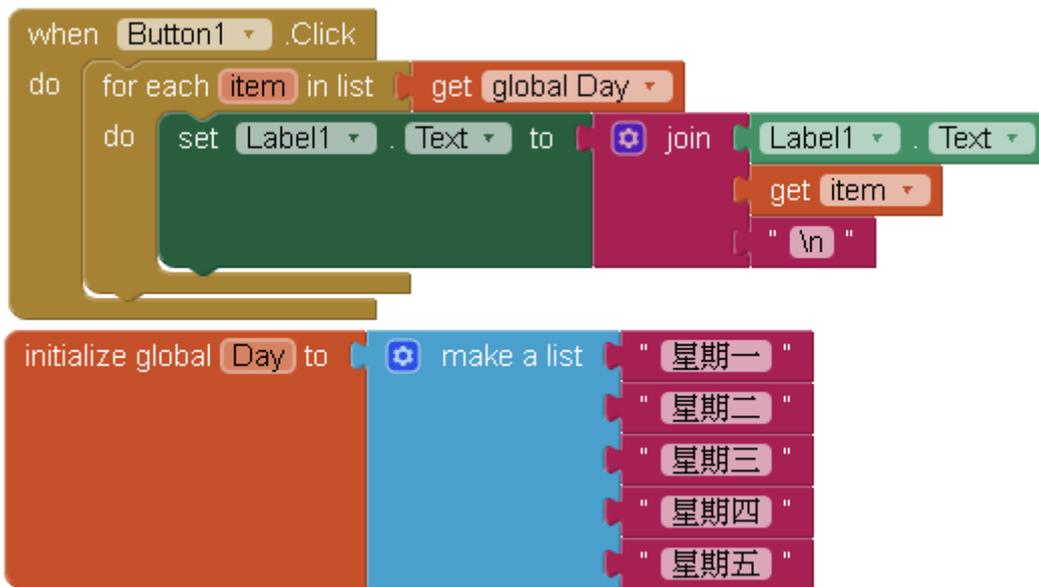
## 範例說明

請依照上一個範例進行修改，透過 TextBox 及 button 將星期六及星期日兩個元素內容 ( item ) 新增至 Day 清單中，並設置新的 label 用來顯示清單長度，透過測試按鈕觀察新增元素內容 ( item ) 前後的長度及清單內容差異。

版面配置依個人喜好編排即可。



延續範例一的程式內容，進行方塊拼圖的新增。

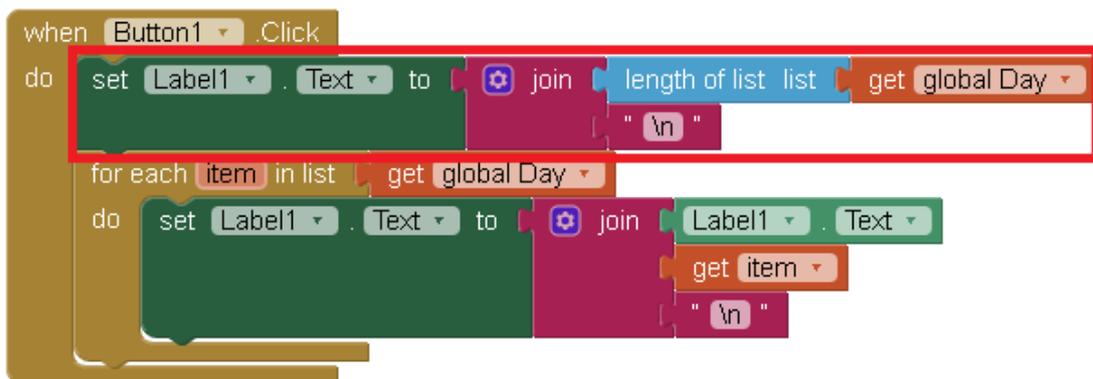


範例一程式方塊拼圖

設定按下新增按鈕 (newitem) 後，就會將 Textbox 中的內容新增至 Day 清單。首先從清單(Lists)中，新增 **add items to list** 與新增按鈕 (newitem) 進行組合，並設定清單為取得全域變數 Day，而新增的元素內容 (item) 為 **textbox.Text**。



之後設定 Label.Text 為顯示清單長度 **length of list**，將 Label.Text 與 **length of list** 組合起來，並設定為取得全域變數 Day 的長度，最後與 **button.Click** 組合。



這樣程式方面就完成了，在新增元素內容 ( item ) 前，先按下測試鈕，觀察新增前的顯示內容及清單長度。完成後，再新增星期六及星期日，並觀察新的顯示內容及其清單長度。

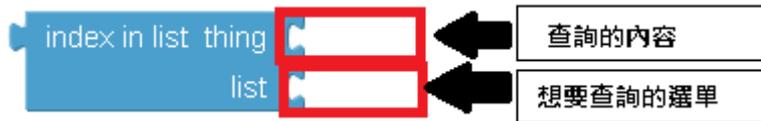


## 活動

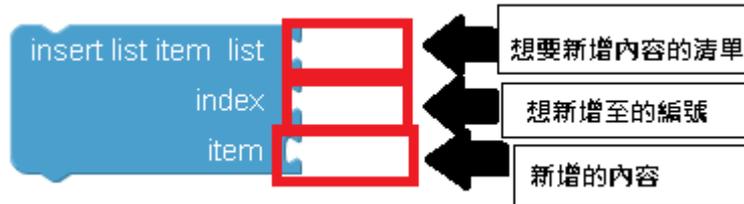
試著新增一個空白清單，並慢慢新增星期一到星期日，將 **length of list** 放在 **For each item** 中，觀察每筆元素內容增加後的清單長度。

## Index in list / insert list item / pick random item 介紹

**index in list** 就主要是用來調查指定項目在清單中的編號，如果找不到指定的項目，則會顯示編號 0。



**insert list item** 則用來將指定的元素內容新增至清單中的指定編號當中，例如將星期日新增至 Day 清單中的編號 3，這時原本在編號 3 的星期三則會被星期日取代，這時的顯示內容則會是，星期一、星期二、星期日、星期四....。



**pick random item** 會隨機從清單中選取一個元素內容 ( item )。



## **Index in list / insert list item / pick random item 方塊設定**

**index in list**、**insert list item**、**pick random item** 這三個方塊都在內建方塊(Blocks)項目中，點選清單(Lists)新增。



如果清單只有到編號 3，在 **insert list item** 設定元素內容到清單編號 4 時，**insert list item** 則會直接新增清單編號到 4。

## 範例說明

跟著範例做出可以幫助使用者選擇要吃什麼的 App，使用 textbox 讓使用者可以輸入編號與食物名稱，並新增一個查詢按鈕及新增按鈕，讓使用者可以查詢輸入的食物所在的編號位置，以及新增食物名稱至清單中，最後透過隨機選取，從清單中隨機選取一項食物給使用者。

版面配置一樣依個人喜好編排即可。



程式方面，我們先新增一個空白的清單並存放到全域變數 food 中。

initialize global food to create empty list

設定新增食物按鈕(newfood)與 **insert list item** 方塊組合，並分別在編號( index ) 與元素內容( item ) 後面接上 newnumber.Text 及 newfoodname.Text 兩個 Textbox 的輸入內容，而清單方面則接上全域變數 food 。

```
when newfood .Click
do
  insert list item list
  index newnumber .Text
  item newfoodname .Text
  list get global food
```

接下來設定查詢按鈕( lookingfor )，將 set label1.Text 與 index in list 結合，讓 label 的文字可以顯示所查詢到的編號，thing 後面接上 textbox 的 foodname.text 讓使用者輸入食物名稱後能查詢到此食物的編號位置，而清單方面則一樣設定為全域變數 food 。

```
when lookingfor .Click
do
  set Label1 .Text to
  index in list thing
  list foodname .Text
  list get global food
```

最後設定吃什麼按鈕( eat )，一樣設定 set label1.Text 將它與 **pick random item** 結合。清單一樣為全域變數 food，這樣當使用者按下按鈕時，App 會隨機幫使用者選取 food 清單中的一樣食物並顯示出來。

```
when eat .Click
do
  set Label1 .Text to
  pick a random item list
  list get global food
```

程式方塊都設定好後，依序新增牛肉麵、炒飯、肉燥飯，並實際測試看看新增、查詢、隨機產生三項功能。

Screen1

Hint for TextBox1

查詢

輸入編號 Hint for TextBox4

輸入食物名稱 Hint for TextBox3

新增

吃什麼

Text for Label1

Screen1

Hint for TextBox1

查詢

輸入編號 3

輸入食物名稱 肉燥飯

新增

吃什麼

Text for Label1

• 𠃊 □ ←

初始畫面

Screen1

牛肉麵

查詢

輸入編號 3

輸入食物名稱 肉燥飯

新增

吃什麼

1

• 𠃊 □ ←

分別新增三項食物

Screen1

牛肉麵

查詢

輸入編號 3

輸入食物名稱 肉燥飯

新增

吃什麼

肉燥飯

• 𠃊 □ ←

查詢牛肉麵顯示編號 1

• 𠃊 □ ←

按下吃什麼隨機產生

## 活動

請新增五筆食物內容至範例中測試，最後在依序設定新的食物，並用編號 1~5 將原本編號的內容蓋掉，看看測試結果是否不同。

## Remove list item / select list item / append to list 介紹

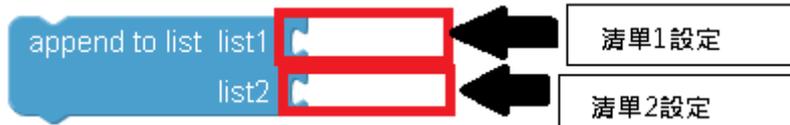
**remove list item** 主要是用來清除清單中指定編號內容，要注意的是如果使用 **remove list item** 將編號 1 的星期一清除，這時編號 2 的星期二則會變成編號 1，而後面的編號也則會往前推移。



**select list item** 的功能為能從清單中，將指定編號的內容顯示出來。

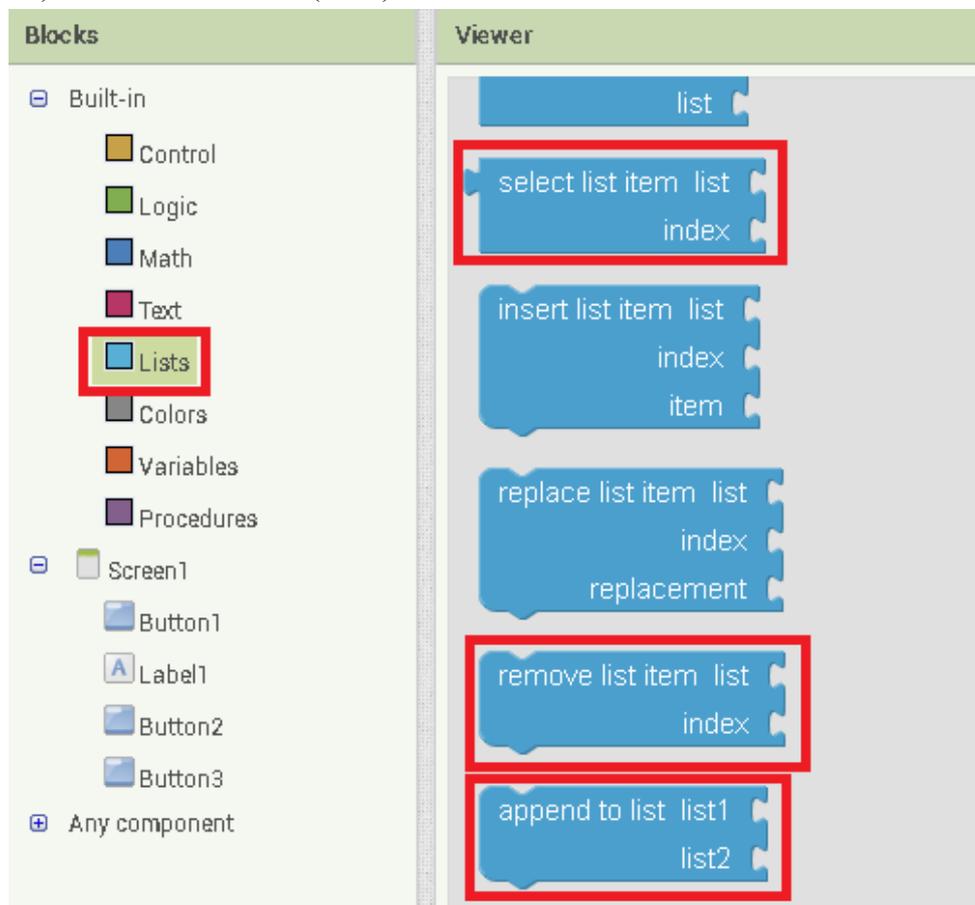


**append to list** 主要用來將兩個不同的清單合成為一個新的清單。



## **Remove list item / select list item / append to list 方塊設定**

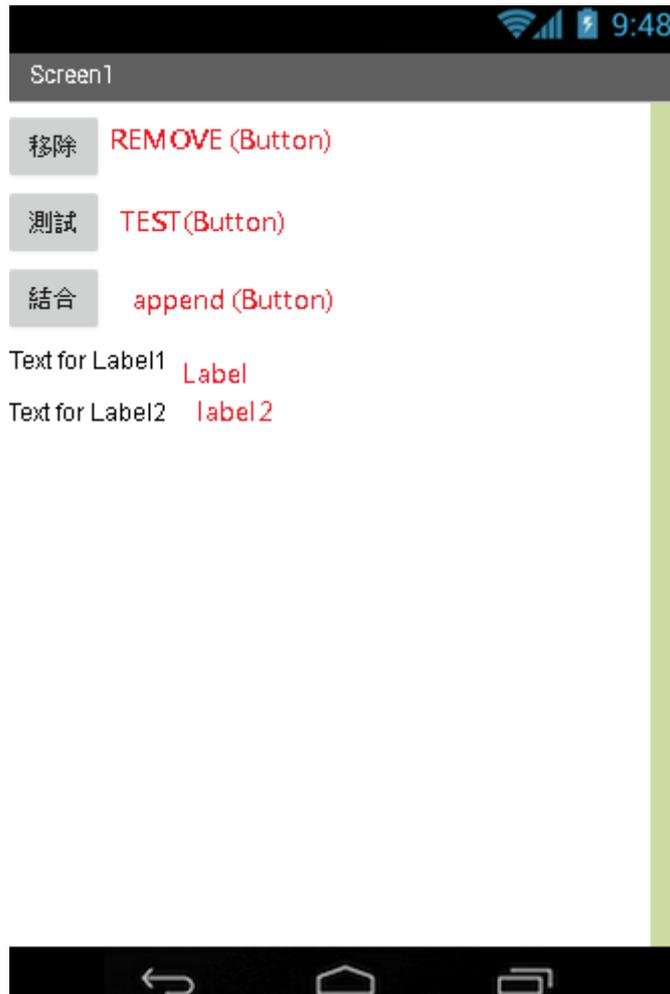
**remove list item**、**select list item**、**append to list** 這三個方塊一樣都在內建方塊 (Blocks) 項目中，點選清單(Lists)新增。



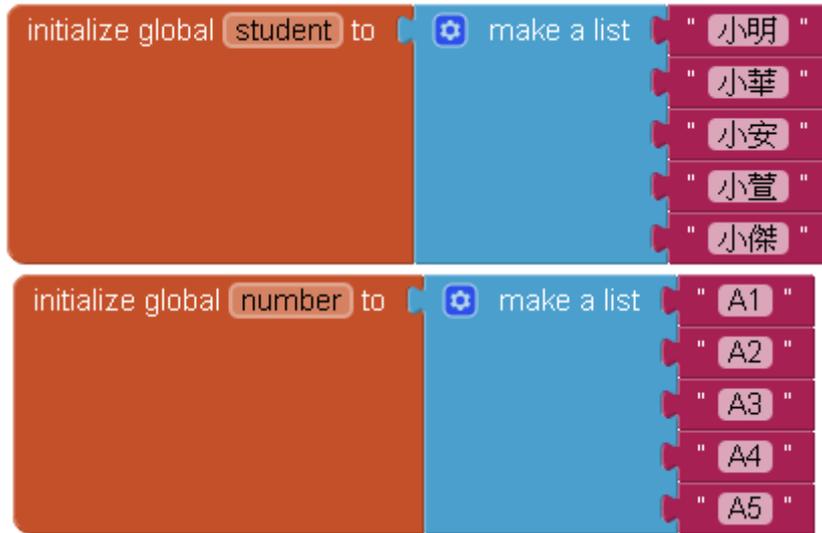
## 範例說明

請新增兩個清單，一個存放學生名稱，另一個用來存放學號，並個別填上五筆元素內容 (itm) 至兩個清單中，接著分別作出測試、移除、結合三個按鈕，並設定測試及移除的清單內容編號為 1。先進行一次測試觀察結果，並在按完移除按鈕後，按一次測試觀察結果，最後再按下結合，並將結果用 **For each item** 迴圈展示出來。

版面配置依個人喜好編排即可。



首先我們需要新增兩個清單，一個名為 `student` 用來存放學生姓名，另一個叫做 `number` 用來存放學號，接著個別輸入五筆元素內容 ( `item` ) 進去。



接下來進行移除按鈕 ( REMOVE ) 的功能設定。將 REMOVE.Click 與 `remove list item` 結合起來，而 `student` 跟 `number` 兩個清單都需要清除編號 1，所以需要組合兩個 `remove list item`，最後再分別接上對應的清單 `list`，以及編號 1，這樣移除按鈕就完成了。



這裡要注意的是，在編號 ( `index` ) 後面要接上的是 `math`，而不是字串 `text`。

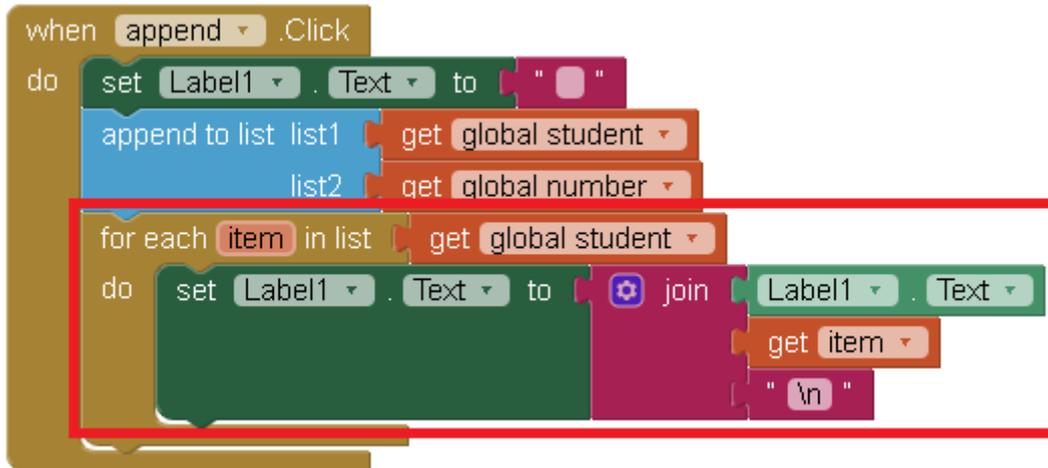
再來進行測試按鈕 ( TEST ) 的設定，設定 TEST.Click 與 `set label.Text` 組合，並在後面接上 `select list item`，與 `remove list item` 一樣，我們需要將兩個清單都做測試，所以需要進行兩次的設定。而清單一樣為設定 `student` 跟 `number` 兩個清單，編號 ( `index` ) 也一樣設置為 1。



最後設定結合按鈕 ( append )，將 append.Click 與 **append to list** 結合，並在選單 list 1、2，分別組合上 student 跟 number 兩個清單。



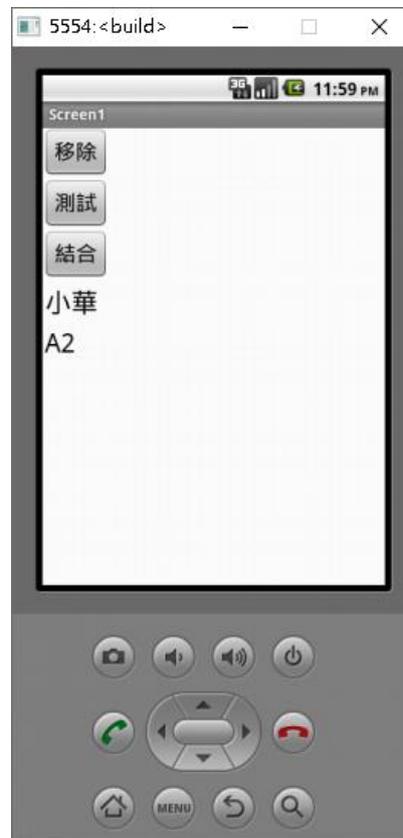
接著用 **For each item 迴圈** 與 set label1.Text 結合，並將結果依序顯示出來。



最後觀察測試結果。



移除前測試



移除後測試



移除後結合兩個清單合成一個

## 活動

請用 **For each item** 迴圈依序顯示 Student 及 number 兩個清單，並用 **remove list item** 移除編號 2，最後再次用 **For each item** 迴圈依序顯示觀察一次。

## 練習題

1. 自行新增一個陣列，並新增一個按鈕，按下可顯示陣列所有內容。
2. 接上題，新增 2 個按鈕，分別可以新增資料進陣列、刪除陣列資料。
3. 接上題，新增 textbox 讓使用者輸入數字，並且可讀取陣列內第幾個元素，顯示至畫面中。
4. 選單練習，新增一陣列加入至選單，讓使用者可從選單選取陣列內資料。
5. 寫一樂透開獎程式，產生所有開獎號碼(例如 1~49)，所有數字加入自 num 陣列內，使用者輸入開獎個數，每次從 num 陣列抽取一個數字加入 lottery 陣列內，並顯示 lottery 陣列內所有元素。