MIT APP INVENTOR
專題習作

L3AIFP
學習指引

在專題習作中，你會使用以往學過的知識製作一個有關健康習慣主題的程式！把從「香港旅遊指南」、「雙按鈕遊戲」或「 你猜我畫 」中學會製作的元素進行改變，或直接加入並製作自己的遊戲！

在這裡開始

運用你的《健康習慣程式設計工作紙》及「待辦清單」，開始製作你的程式。

- ❑ 在每個教節開始時，製作一個清單寫下你在該教節的任務。
- ❑ 按照「待辦清單」上的任務進行編程。
- ❑ 在教節完結前，記錄已完成的任務。
- ❑ 遇上困難時，記錄遇到的情況，並構思如何解決。

你將會跟你的伙伴一起編程，製作程式。

雙人編程的守則

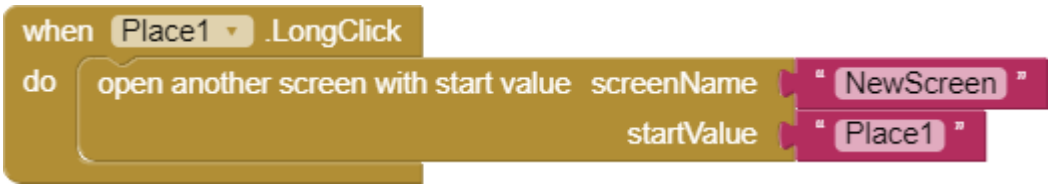
應該：尊重他人
應該：與伙伴談論編程的問題
應該：向伙伴解釋你在做甚麼
應該：預計下一步需要做甚麼和提出建議
應該：時常與伙伴交換工作或角色
不應該：對伙伴頤指氣使
不應該：佔據滑鼠或鍵盤

元件

切換屏幕

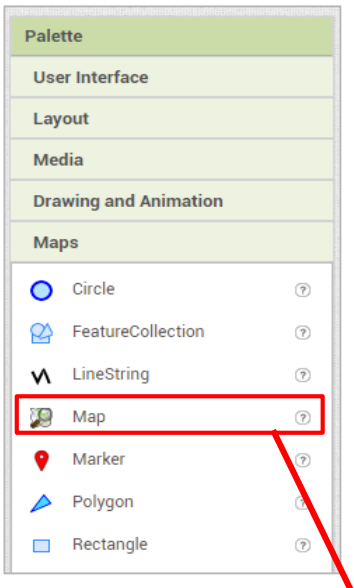
如果要從一個屏幕切換到另一個屏幕，你可用 **open another screen** 指令方塊，並將屏幕名稱更改為文字方塊。在下列例子中，事件**Button1.Click** 觸發了開啟 **Screen1**。

你也可以透過 **open another screen with start value** 指令方塊，在屏幕之間傳遞信息。在下列例子中，當你長按 **Place1** 按鈕，便會打開另一屏幕 **NewScreen**，同時傳送初始值 “Place1”。

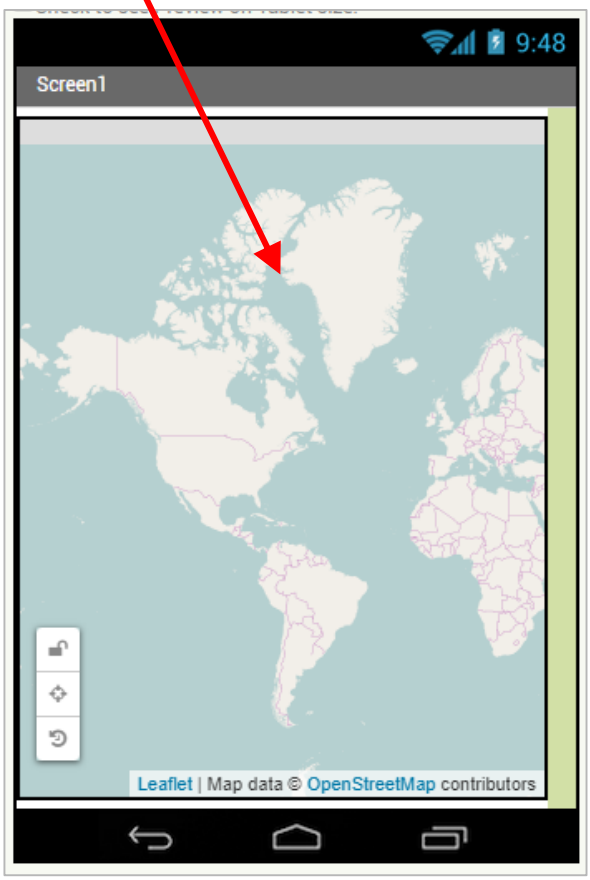


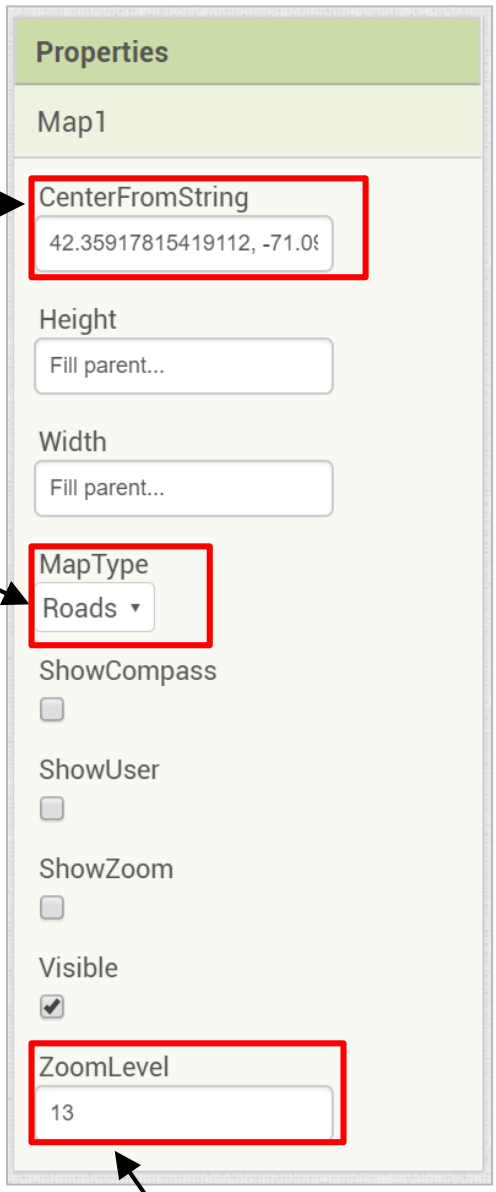
元件

地圖



- ❑ 透過 **CenterFromString** 自設緯度和經度，使地圖置中 。
- ❑ 透過 **MapType** 查看道路、空中、及地勢的情況。



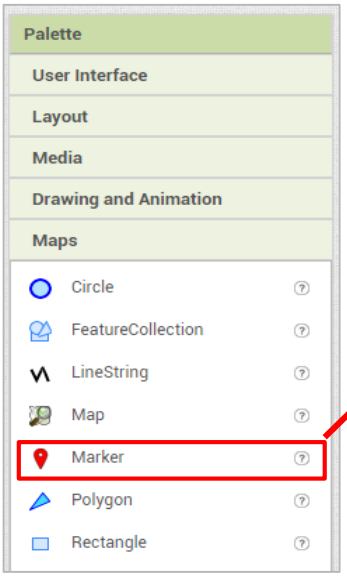


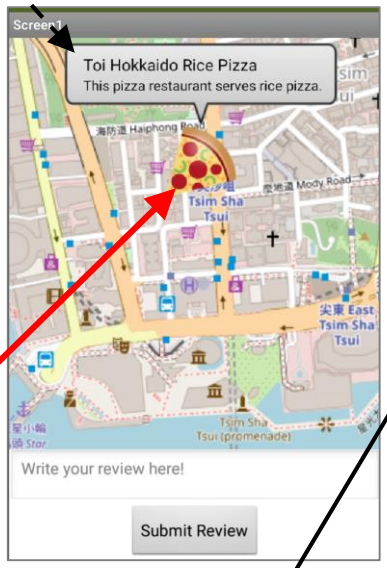
- ❑ 透過 **ZoomLevel** 放大縮小， 數值由 1-18 。

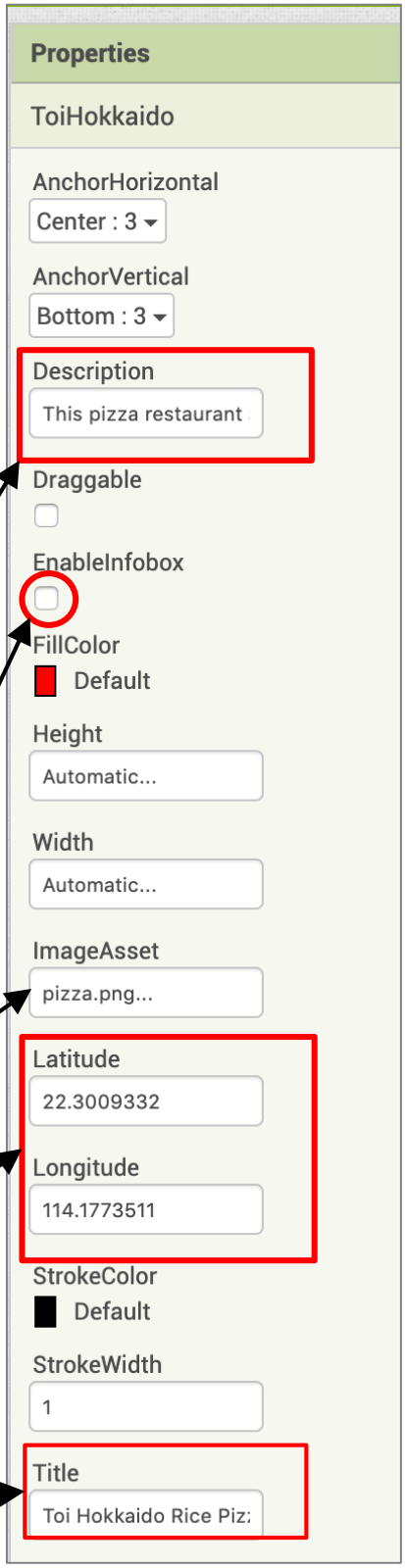
元件

標記

資訊框







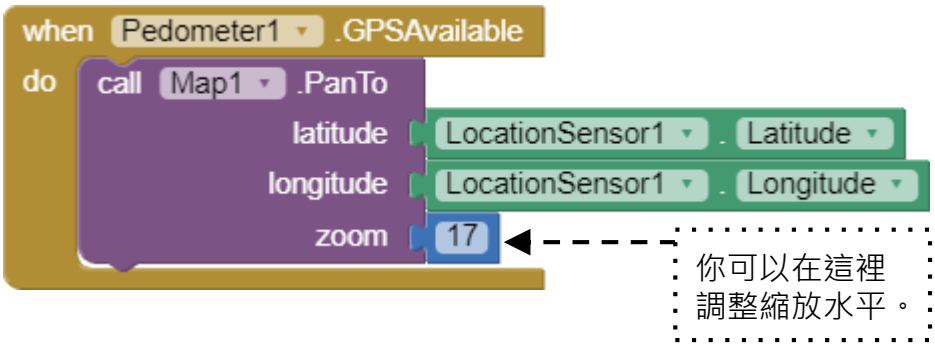
- ❑ 透過 **Description**，你可以寫下更詳盡的文字描述；當點擊標記時，讓資訊框顯示該些內容。
- ❑ 剔選 **EnableInfobox** ，當用家點擊時，他們能閱讀該地點的資訊。
- ❑ 透過**ImageAsset** ，你可以將標記從圖釘改為圖像！
- ❑ **緯度 (Latitude)** 和**經度 (Longitude)** 是標記在地圖上的確切地理位置。
- ❑ **Title** 是資訊框的標題。

健康習慣程式

元件

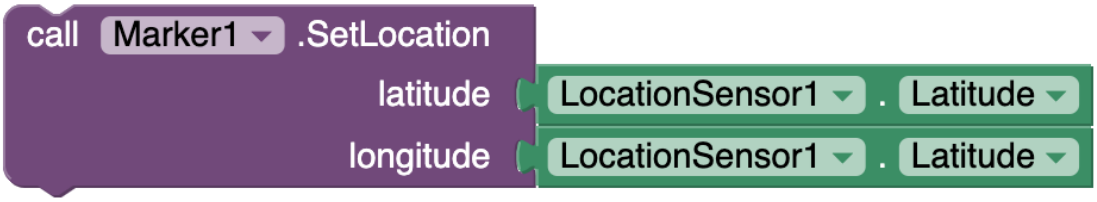
地圖

Map.PanTo 將地圖放置於指定的**緯度**和**經度**，並放大至指定的縮放程度。
該移動是動態的。



標記

Marker.SetLocation 移動標記至地圖上指定的緯度和經度。
相比分開設定緯度和經度，這個方法更有效率。

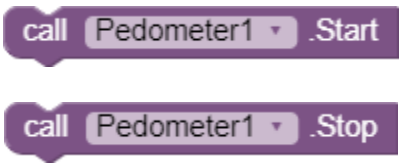


健康習慣程式

元件

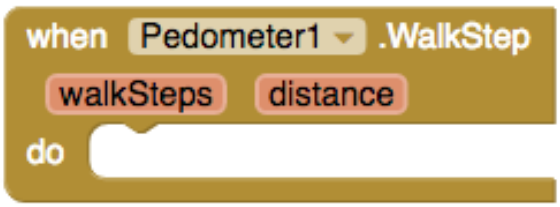
計步器 – 計算步數

使用 **Pedometer.Start** 及**Pedometer.Stop** 指令方塊，啟動或停止計步器。

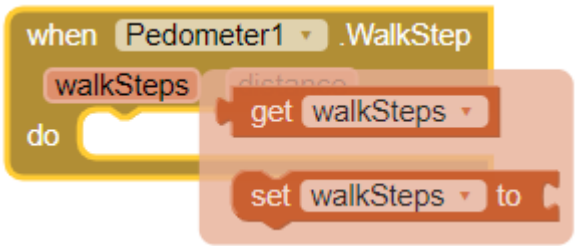


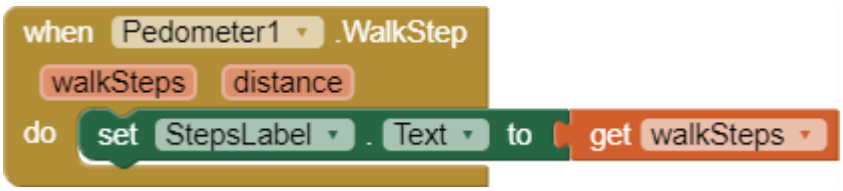
Pedometer1.WalkStep 是一個觸發計步器感應步伐的**事件**。

- **walkSteps** 是總步數。
- **distance** 是總距離。



將計步器計算的總步數展示在 StepsLabel.Text。

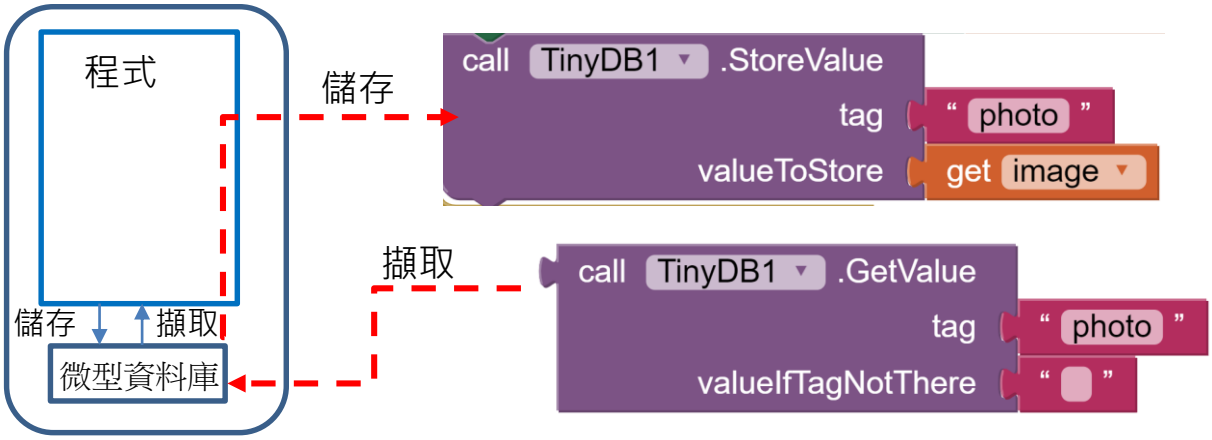




元件

微型資料庫

微型資料庫 (Tiny Database, TinyDB) ，是能夠持續儲存數據的元件。
因此，儲存的數據會保留在你的移動裝置上，以供你再次運行程式時使用。



微型資料庫有兩個主要的指令方塊：**StoreValue** 及 **GetValue** 。

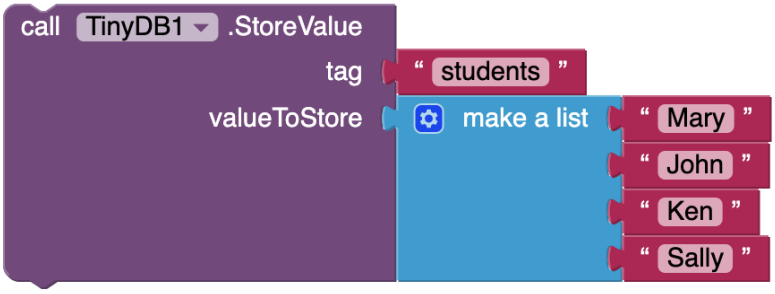
- **tag** （標籤）像一個變數的名稱，而**valueToStore** 就是一個儲存在標籤裡的數值。

- **GetValue** 透過 **tag** 從之前儲存的數據中獲取數值。如果沒有儲存的數值，則會傳回 **valueIfTagNotThere**。確認**valueIfTagNotThere** 要與之前儲的存數據類型匹配（數字 、文字 、清單）。

元件

微型資料庫

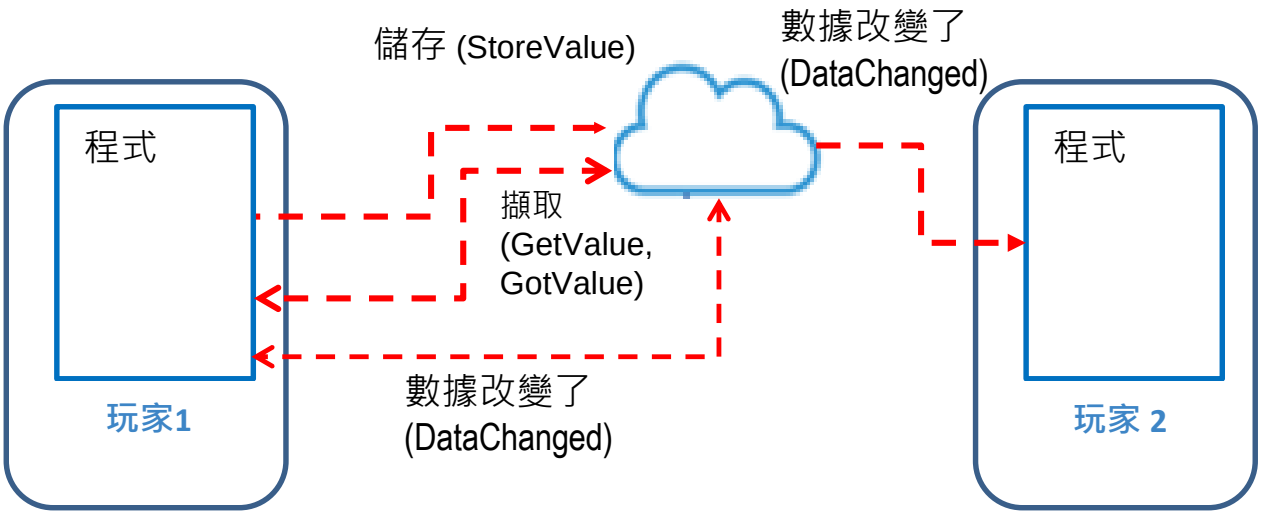
如果要儲存多個類似的項目，可以使用清單。



元件

CloudDB

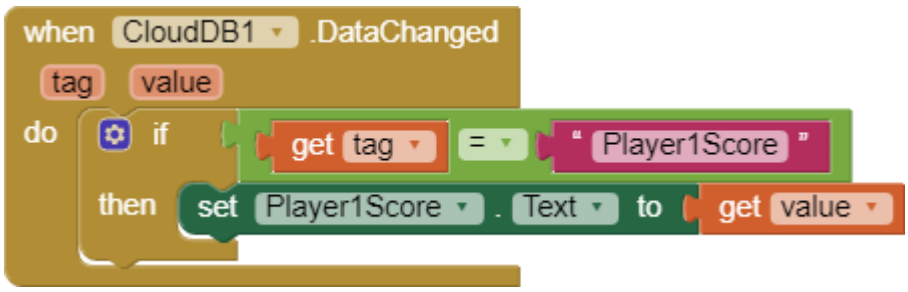
CloudDB 和 **TinyDB** 的相似之處在於，他們都會儲存數據。
然而， TinyDB 將數據儲存在裝置裡，而 **CloudDB** 將數據儲存在雲端，因此可以在多部裝置中分享數據 。



在CloudDB，你可以透過 tag 儲存數值，就像TinyDB 。

在任何時候，當任何一部裝置在CloudDB儲存了一個數值，就會產生 **CloudDB.DataChanged 事件**至*所有*連結了的裝置。

例如，如玩家1 的 Player1Score 有任何更新，*所有*玩家都會透過 **DataChanged** 獲得更新後的數值。

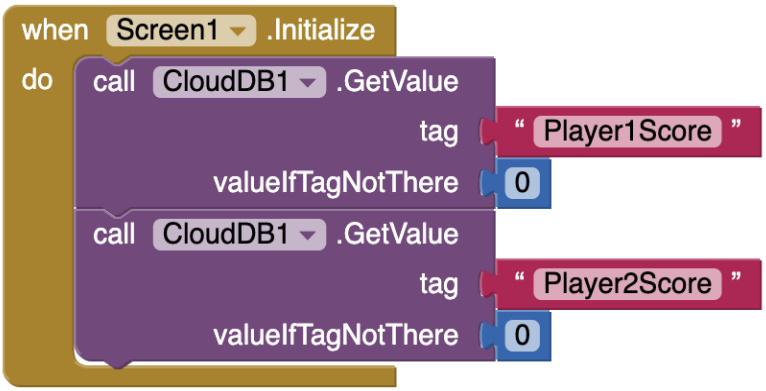


元件

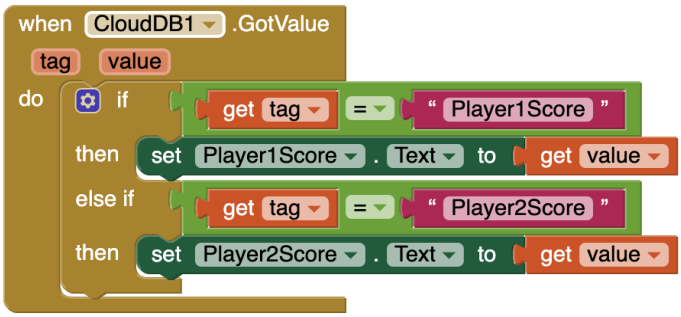
CloudDB

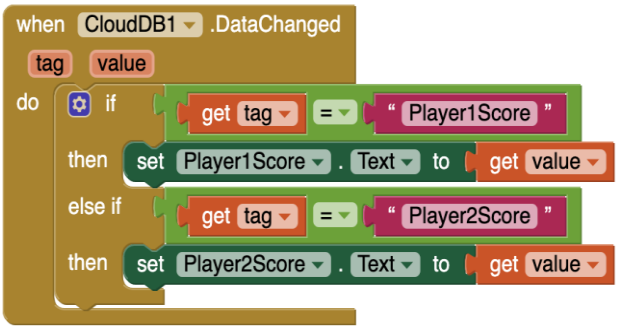
當 **DataChanged** 更新所有玩家的數據，明確地向 CloudDB 詢問 tag 的數值很有幫助。例如當你剛開始執行一個程式， 在CloudDB 取得最新的資訊是很重要的。

GetValue 透過 **tag** 從之前儲存的數據中獲取數值。如果沒有儲存的數值，就會傳回 **valueIfTagNotThere**。



當 CloudDB 執行 **GetValue** 的要求，便會觸發 **GotValue** 事件。
配合 **GotValue** 及 **DataChanged** 事件指令方塊是一個好習慣，用以檢查程式裡任何可能的 tag。



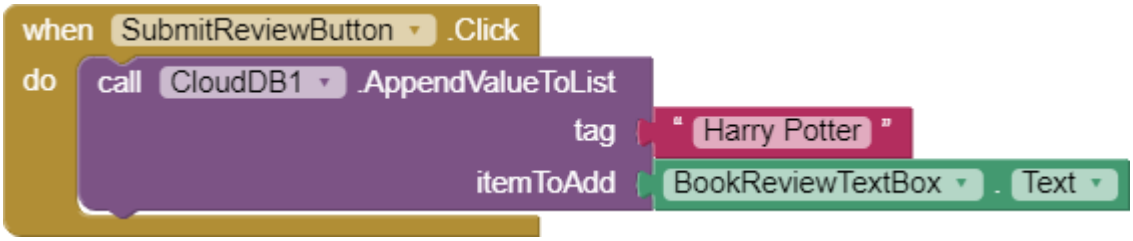


元件

CloudDB

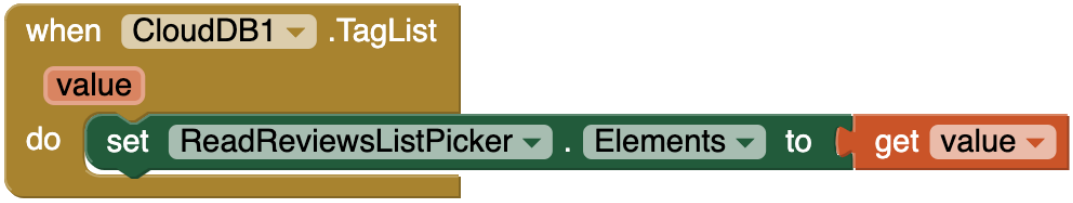
AppendValueToList 將數值加到儲存在Cloud DB中的清單結尾。

在此例子中，點擊 **SubmitReviewButton** 時，它觸發 **CloudDB.AppendValueToList**，將 **BookReviewTextBox** 的數值，以 “Harry Potter” 的 tag 加至清單結尾。



GetTagList 獲取程式裏所有 tag 的清單。CloudDB 傳回 **TagList** 事件中的已知 tag 的清單。

在這裡例子中，當用戶觸摸 ListPicker （清單選擇器）時，將呼叫 **GetTagList**。當 CloudDB 傳回標籤清單時，ListPicker 的元素將被設置為標籤清單（在本例子中為書籍清單）。



元件

清單選擇器

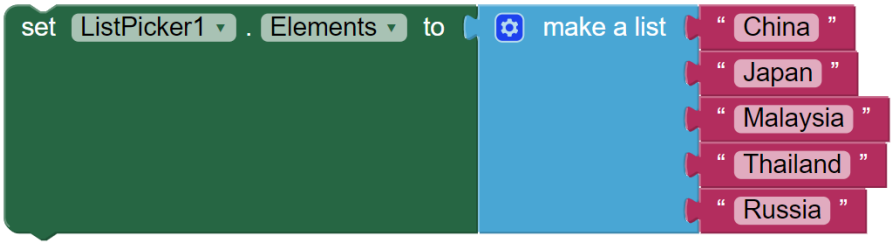
清單選擇器就像一個下拉式功能表，用家可以選擇清單中的項目。

要設置清單選擇器中的元素，可以使用三種不同的方法。

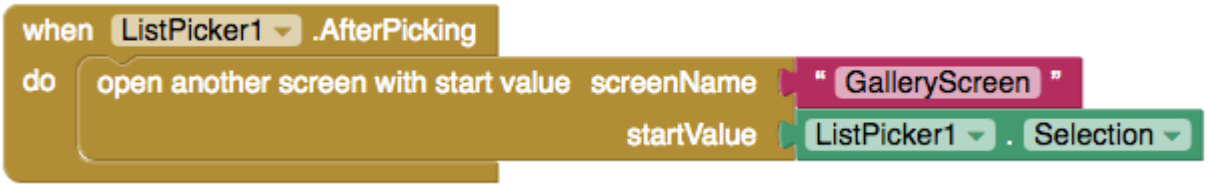
- ❑ 在 Designer，設定 **ElementsFromString** 屬性。

- ❑ 在Blocks Editor ，設定**ListPicker1.ElementsFromString**。

- ❑ 或者，你可以選擇清單而非字串，以設定**ListPicker1.Elements**。



當用家在清單選擇器 *select*（選擇）項目，你可以利用 Selection（選中項）執行行動，例如打開另一屏幕並傳送初始值。



元件

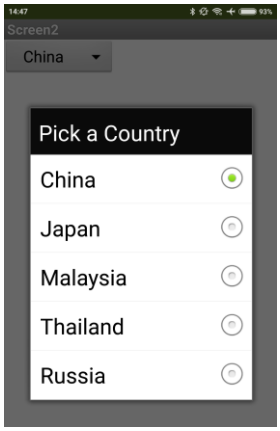
清單顯示器及下拉式選單

清單顯示器 (ListView) 和下拉式選單 (Spinner)就像清單選擇器，他們只有少許差別。

清單顯示器

下拉式選單

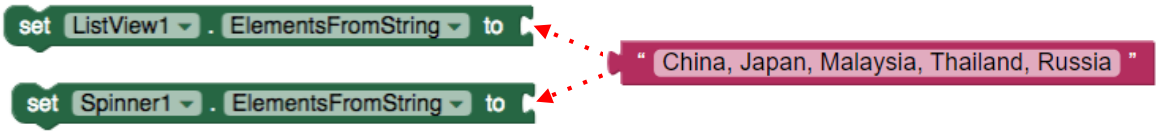
點擊獲得
彈出視窗



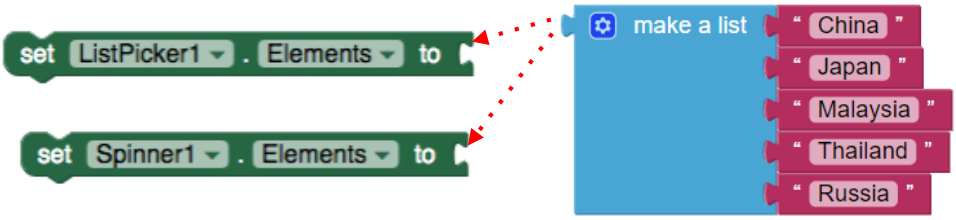
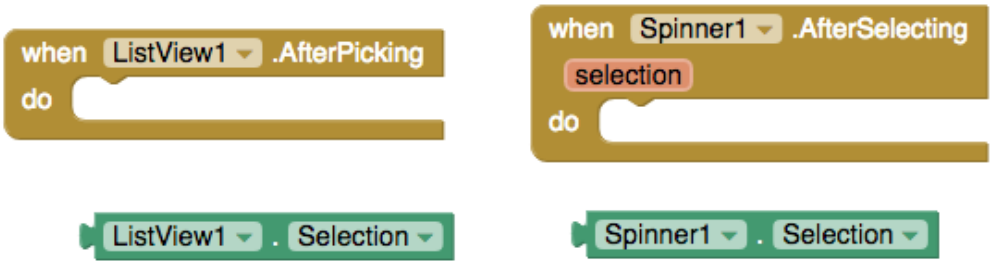
你可以設定 ElementsFromString 及 Elements，就像清單選擇器。

- ❑ 在畫面編排中，設定**ElementsFromString**屬性。

- ❑ 在程式設計中，設定**ElementsFromString**。



- ❑ 或者， 你可以以清單模式設定**Elements**。



清單顯示器和下拉式選單同樣有 **AfterPicking** 及 **AfterSelected** 事件指定方塊，你可以使用 Selection 執行。