**Lab4 Activity**

1. 本節目的：

* 理解什麼是Activity，並產生一個Activity
* 如何透過Intent切換Activity
* 理解透過Bundle攜帶資料
* 理解透過onActivityResult()方法返回資料

1. 觀念說明

Android應用程式元件包含Activity、Service、BroadcastReceiver、Content Providers這類的Java程式，而Activity 是其中最重要的應用程式元件，負責提供應用程式顯示畫面上的相關工作，大部分的APP所顯示的畫面都是寫在Activity之上，不論是列表、圖片或是地圖的畫面，都是基於Activity來呈現。

如以下的APP畫面，就是基於Activity實現出來的。

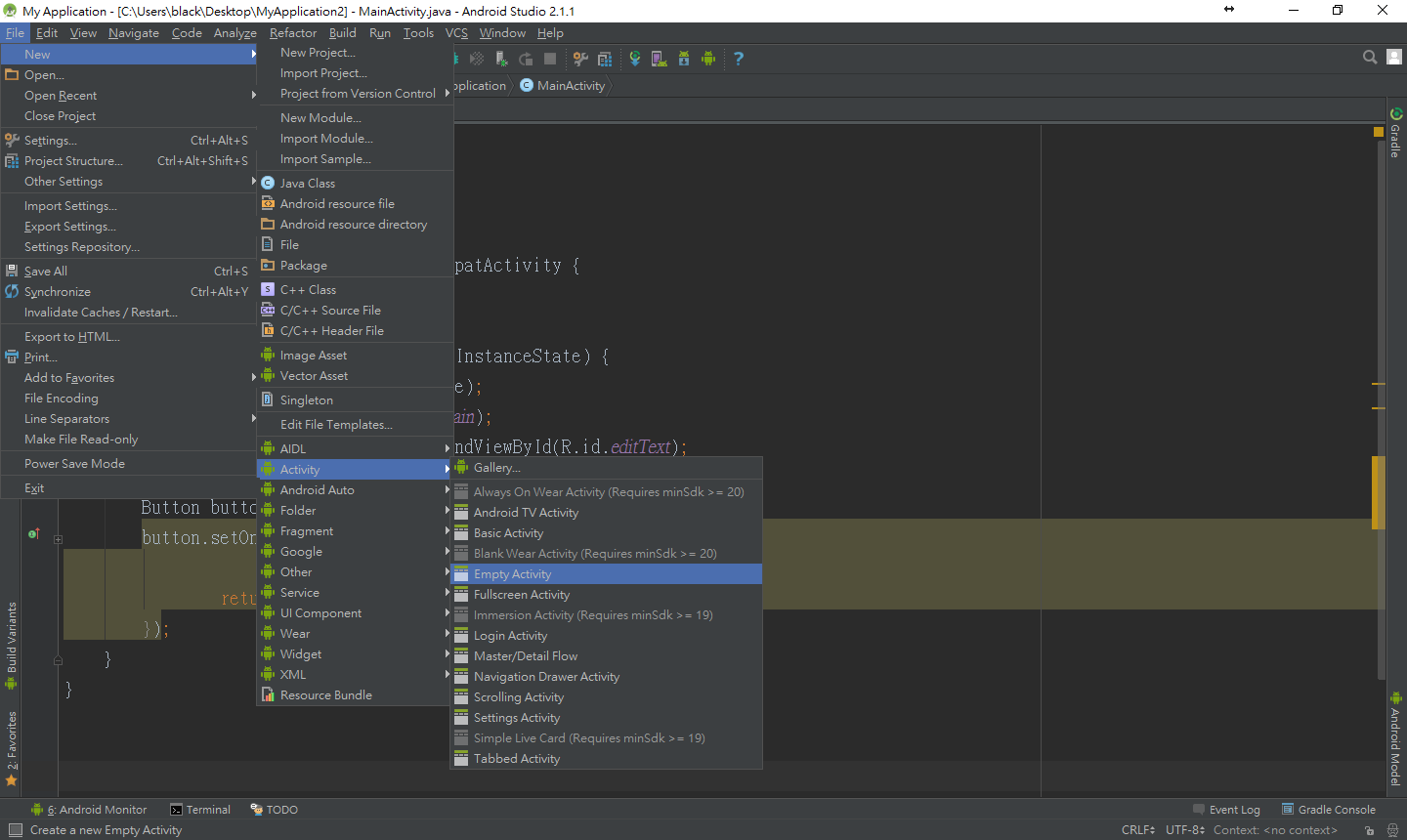
  

要在螢幕上顯示畫面需要透過**畫面配置元件(\*.xml)**與產生控制的**應用程式元件(\*.java)**。前面我們學到畫面配置元件即為Layout，用於決定了每個元件的擺放位置，而 Activity賦予這個畫面配置能與使用者互動的功能。

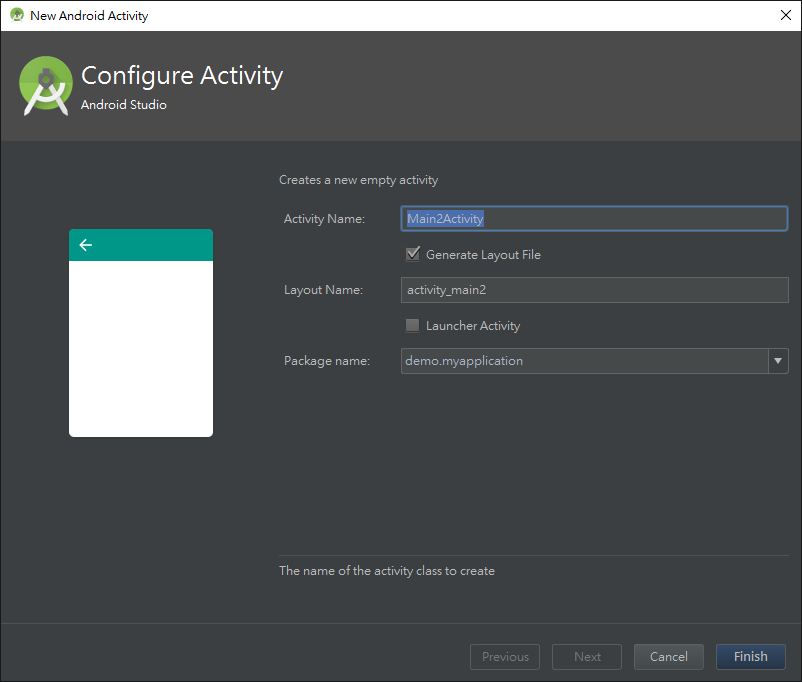
我們能透過Activity來顯示出某些資訊(圖片、文字或是地圖)給使用者，或是將使用者的操作傳送給程式來做控制(監聽器)，因此Activity扮演著Android使用者介面的角色。

1. **產生Activity**

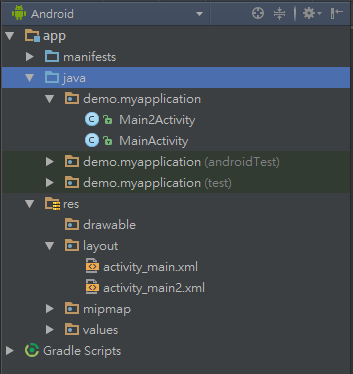
要產生出一個新的Activity，首先先選擇File/New/Activity/Empty Activity來產生出空白的Activity。



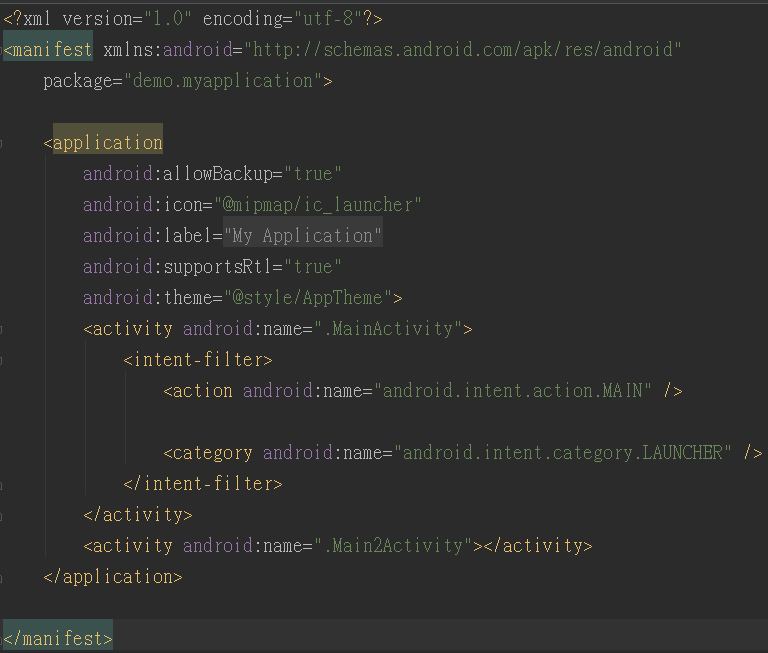
選擇後可於下面的視窗中修改Activity的名稱與對應Layout的名稱，如果只有更改Activity的名稱，Android會自動幫你修改Layout的名稱。



按下finish後，可以於左邊目錄中看到系統幫你產生出Main2Activity.java以及activity\_main2.xml。



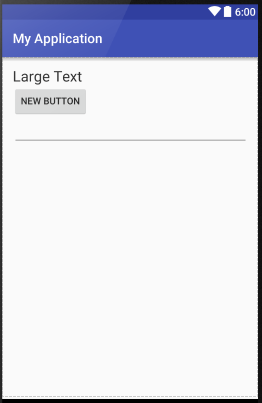
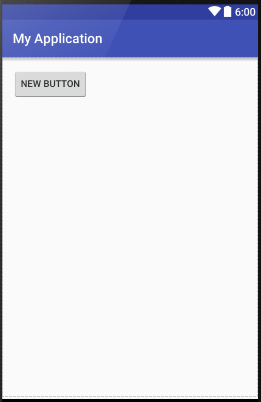
而AndroidManifest.xml也會自動增加Main2Activity的資訊。



1. **使用Intent切換Activity**

Android 應用程式元件(Activity、Service、BroadcastReceiver、Content Providers)之間的切換會需要透過「Intent」。Intent是可用來向另一個應用程式元件(Activity、Service、BroadcastReceiver、Content Providers)**要求動作**的傳遞物件。最基本的Intent用途是來啟動其他的應用程式元件。如果啟動的對象是Activity，則可以在畫面上顯示一個新的Activity，我們也可以說這是Activity間的切換動作。

Intent字義上是指「意圖」，以Activity切換的目的上來解釋我們可以口語描述成「A元件意圖啟動B元件」。下面我們以MainActivity切換至Main2Activity為例，MainActivity就表示A元件，Main2Activity就表示B元件，兩者透過意圖傳遞把顯示畫面由MainActivity改為Main2Activity。

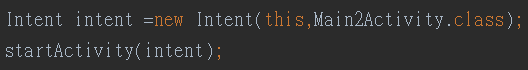
Intent

意圖啟動

Main2Activity

MainActivity

實際編寫的程式碼如下：

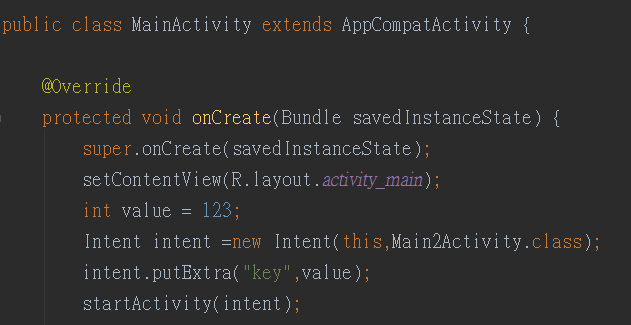


Intent有兩個參數，第一個參數我們要描述由哪個元件發起這個意圖，如從MainActivity發起則要填入MainActivity或是this(表示MainActivity本身)，第二個參數則要描述要接受意圖(被啟動)的是哪個元件，對象如果為Main2Activity則要描述成Main2Activity**.class**。

而要將這個Intent的發出，我們需要用到startActivity()方法來送出Intent，Main2Activity獲得通知後便會被啟動，並覆蓋在MainActivity之上。

1. **傳遞資料**

Intent除了可以做到基本的切換之外，Intent也可夾帶一些資料到接收意圖方去，例如某個使用者在MainActivity填寫了一個表單，並希望在Main2Activity看到結果。這時我們就要使用Intent傳遞資料的方法，以下則是最簡單的傳遞資料語法，主要是描述(1)將MainActivity透過Intent切換到Main2Activity，(2) 透過Intent傳送123的資料，從MainActivity到Main2Activity：

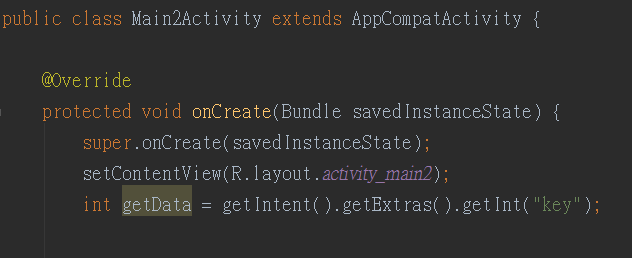


夾帶資料

想傳遞的資料

intent.putExtra()可以想像是把想傳遞的資料 (value)貼上一個標籤(key)，接收的對象可以透過標籤去得到他要的資料。

接收到intent而被喚起的Main2Activity如果要取得傳過來的資料，可以用以下語法：

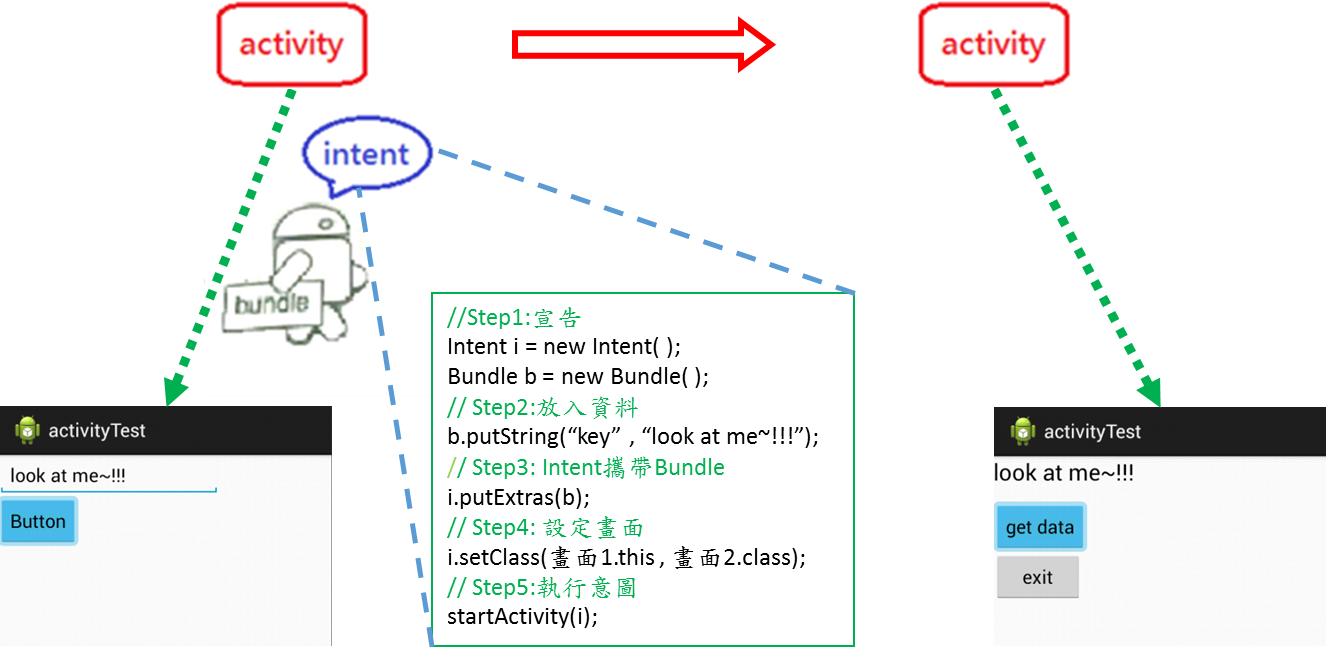


得到資料的對象

以key找到對應的資料並讀取

getIntent()可以取得從MainActivity傳過來的Intent，而getExtras()則可以取得底下夾帶的資料，我們可以使用getInt(key)方法去找到你傳遞的資料，他的返回值就是MainActivity夾帶的資料。此外，由於傳遞的資料是int型態，因此使用getInt()，如果是float型態則要用getFloat()，以此類推。

然而intent.putExtra()每一次傳遞資料我們都必須要知道傳遞的資料的型態是甚麼，而且資料只能單獨的傳入、單獨的讀出，這樣資料的完整性並不高，有時候我們希望某些資料能被視為整體一次傳遞，例如一份菜單資料，我們不要每一項餐點資料都單獨傳過去，而是能以訂單為單位傳送。因此我們就會需要用到打包成包裹的概念，而在android中這就是Bundle。



Bundle可以一次打包不同的資料型別，例如int或String，打包時需要依據型態透過，putInt()或putString()來儲存資料。

舉個例子，我們希望從MainActivity中夾帶一筆整數資料以及一筆字串資料到Main2Activity去，我們編寫的程式如下：

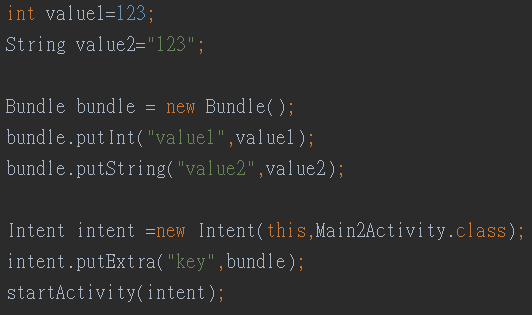
夾帶Bundle

想傳遞的多筆資料

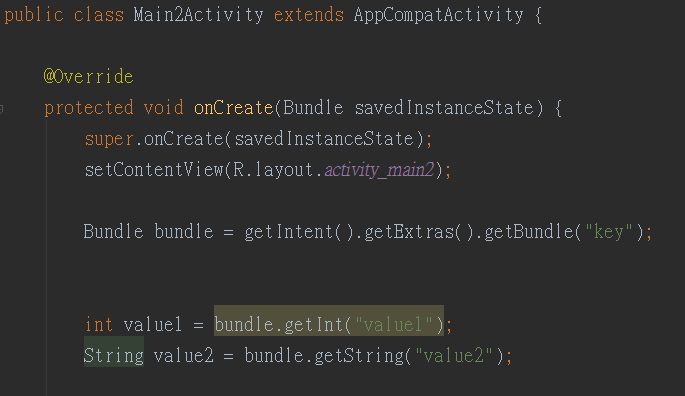
存入一筆字串

存入一筆整數

把資料打包成Bundle



而新的Activity只需要取出Bundle就可以還原資料。



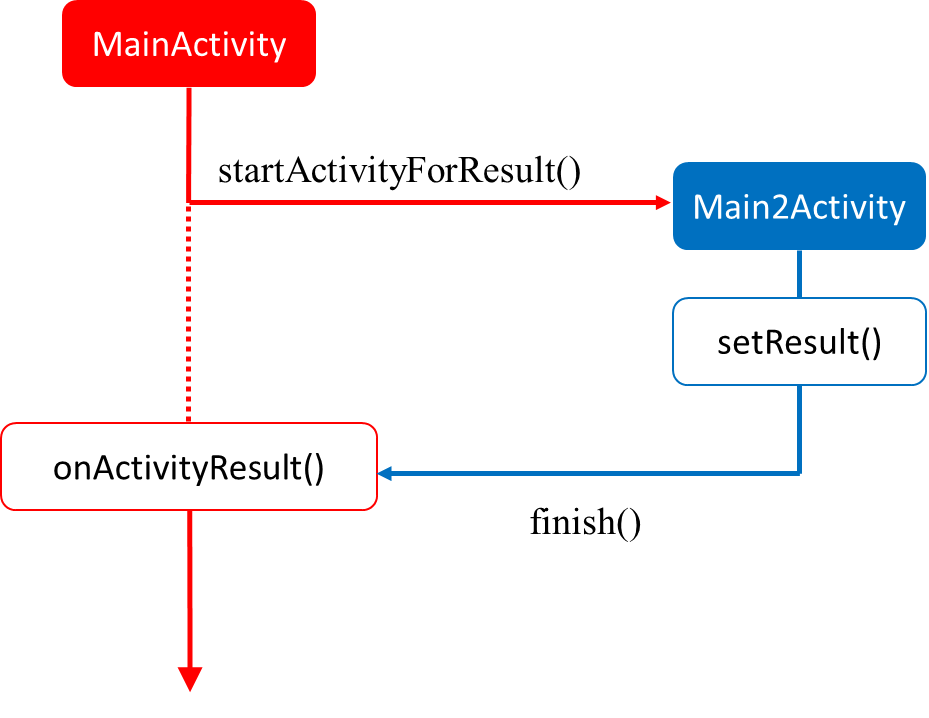
以key找到對應的Bundle並讀取

打開Bundle

取出資料

1. **返回資料**

透過Intent方法啟動的Activity，除了之前介紹的startActivity()方法之外，某些情況我們希望新的Activity在接收到資料後，能再夾帶資料返回到前一個Activity，實現兩個Activity資料往來的目的，這時我們就會使用到startActivityForResult()，實現的步驟流程如下：



(4)

(3)

(2)

(1)

1. MainActivity使用startActivityForResult()方法，啟動Main2Activity。
2. Main2Activity使用setResult()方法，儲存要返回的資料。
3. Main2Activity使用finish()方法結束Main2Activity，並返回MainActivity。
4. MainActivity使用onActivityResult()方法，取得返回資料。

此例中，我們要從MainActivity傳送夾帶一筆整數資料以及一筆字串資料到Main2Activity去，並且再接收Main2Activity回傳的資料。依據上述四個步驟編寫後的程式如下：

4)使用onActivityResult()方法，取得返回資料。

1)使用startActivityForResult()方法，啟動Main2Activity

啟動Main2Activity

取得返回資料

確認Main2Activity執行的狀態

驗證發出對象

標記發出意圖的對象是誰



* requestCode的目的就是在於我們向新Activity提出要求並得到回覆時，用來判斷發出需求者(某個功能或頁面)為誰，以及要如何應對。requestCode扮演著一種需求者的編號，讓原Activity可以根據這編號來判斷發出需求的對象。
* onActivityResult()會等待新的Activity返回結果，我們可以根據傳過去的requestCode去驗證發出對象，之後透過resultCode確認在Main2Activity中執行的情況如何。



結束Activity

儲存要返回的資料

3)使用finish()方法結束Main2Activity，並返回MainActivity。

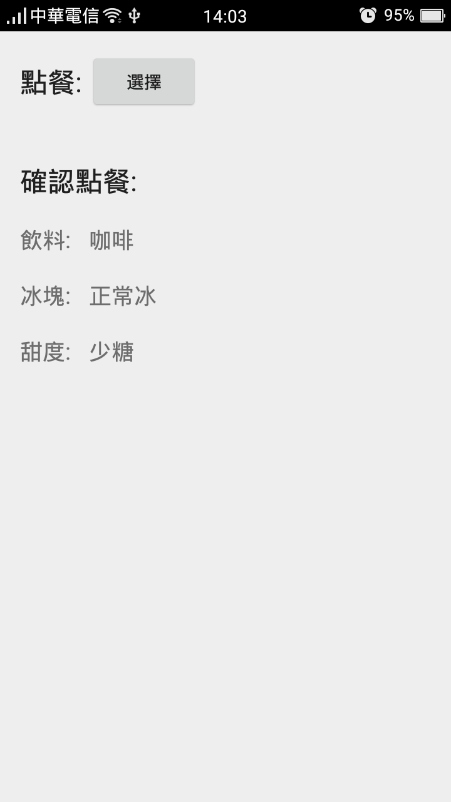
2)使用setResult()方法。

標記狀態

* resultCode可以用於回報執行結果給MainActivity，例如成功時我們可以考慮用resultCode=0來表示，失敗時resultCode=-1，用這方法來告知原Activity該如何處理。

1. 設計重點:

* 本次範例實作一個點餐系統，透過設計兩個不同的Activity，分別帶有不同的布局來完成點餐作業
* Activity aty1按下選擇按鈕後會切換到Activity aty2
* Activity aty2中可以輸入飲料與選擇甜度、冰塊，設定完點餐資訊後再回傳aty1點餐資訊
* Activity aty2接收點餐資訊資訊後可以顯示訂單資訊

resultCode=101

requestCode=1

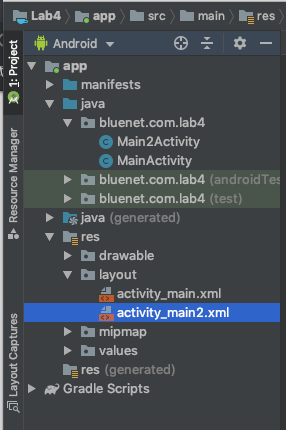
aty2

aty1

1. 設計步驟:

Step1

新建專案，建立MainActivity，以及 Main2Activity

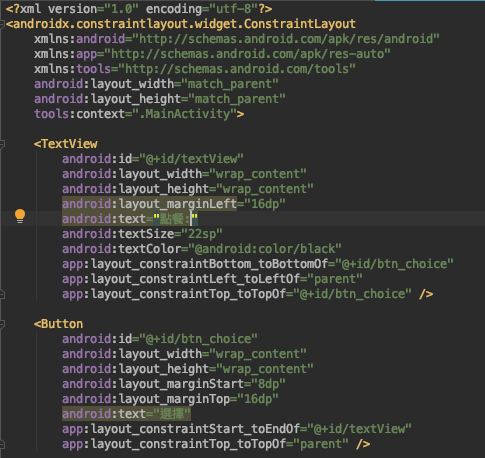


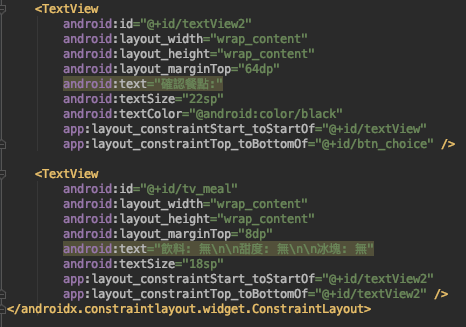
Step2

繪製MainActivity layout



對應的xml如下：





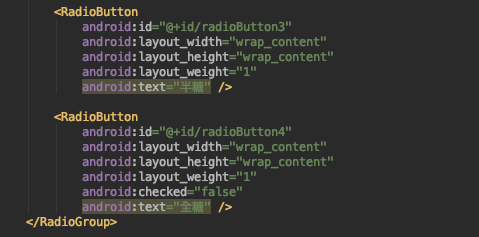
Step3

繪製Main2Activity layout



對應的xml如下：

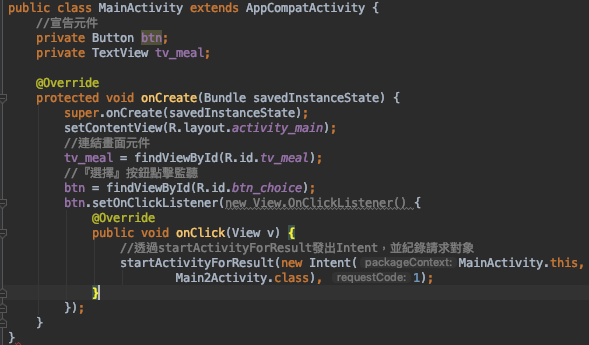






Step4

1. 撰寫MainActivity程式，按下按鈕後切換至Main2Activity



Button點擊事件

連接Button元件

透過Intent切換至aty2並傳遞requestCode=1來標記發出者

連接TextView元件

初始化Activity

連接main1.xml畫面



1. 建立onActivityResult()接收返回資料後，將data的內容讀出以TextView作顯示



取得甜度資料

取得飲料資料

取得冰塊資料

確認Main2Activity執行的狀態

讀取Bundle資料

透過TextView.setText()顯示資料

驗證發出對象



Step5

撰寫aty2程式

1. 用RadioGroup監聽RadioButton的按下事件，並改更改成的字串當作甜度資訊



連接RadioGroup畫面元件

按下radioButton4，紀錄全糖字串

按下radioButton3，紀錄半糖字串

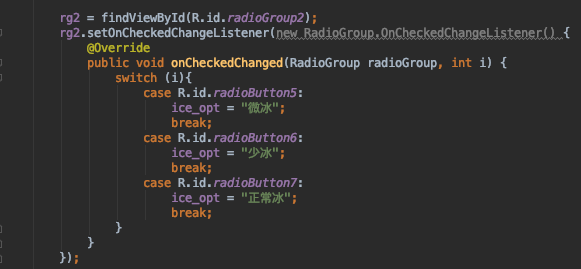
按下radioButton2，紀錄少糖字串

連接main2.xml畫面

初始化Activity

按下radioButton1，紀錄無糖字串

1. 用RadioGroup監聽RadioButton的按下事件，並改更改成的字串當作甜度資訊



按下radioButton7，紀錄正常冰字串

按下radioButton6，紀錄少冰字串

按下radioButton5，紀錄微冰字串

1. 用Button監聽按下事件，飲料、甜度、冰塊的資訊回傳到前一頁面



用101標記執行狀態並記錄Intent

把甜度與冰塊資訊放入Bundle

連接EditText元件

取得EditText字串內容

Button點擊事件

連接Button元件

