**Lab5: 提示訊息元件**

1. 本節目的：

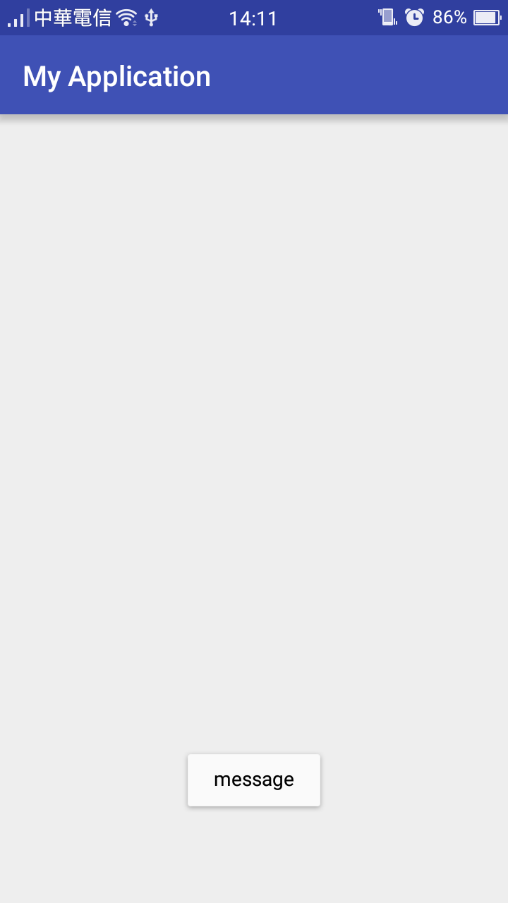
* 學習利用Toast的方法顯示文字訊息
* 學習利用客製化Toast的方法同時顯示文字和圖片訊息
* 透過AlertDialog顯示提示訊息
* 透過AlertDialog顯示陣列資料

1. 觀念說明：

我們很常在應用程式中遇到按下某個按鈕或是畫面時，系統會彈出訊息或是對話框於畫面上，本章節會教導如何實現這幾種常用的提示訊息。

1. **Toast-快顯訊息**
   1. **介紹**

Toast是一種很快速的即時訊息，常用在通知使用者各種立即的資訊上，顯示後幾秒內就會消失。Toast主要可以應用在反應某個操作下的回饋，例如告知使用者某些設定上的成功與否。也很常被作為debug手段。



出現幾秒後自動消失

* 1. **用法**

Toast最簡單的使用方法是透過Toast的靜態函式makeText來產生文字內容，方法如下：

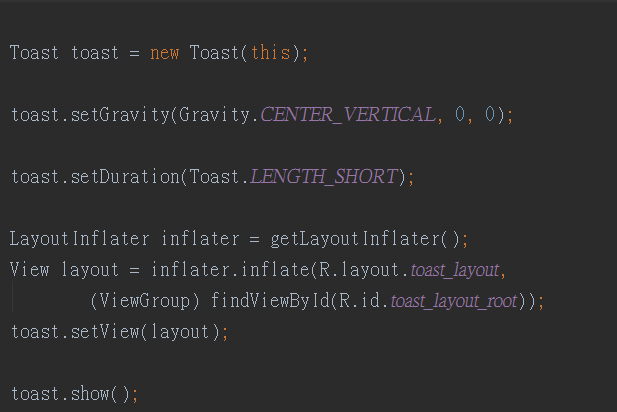


makeText的第一個的參數要傳入調用Toast的對象，通常下應要填入本身的Activity實體(this)。第二個的參數為傳入字串，作為輸出畫面的內容。第三個的參數為持續時間，LENGTH\_SHORT持續時間較短，LENGTH\_LONG持續時間較長。

makeText產生後的結果會是個Toast的實體，就可以使用show()的方法將Toast資訊顯示到螢幕上。

* 1. **客製化Toast**

除了用makeText簡單而快速的產生Toast之外，也可做到位置改變或是自訂顯示的內容。這邊我們就需要先了解幾個Toast提供方法，實現的程式碼如下：



Step2：Toast在畫面中顯示位置

Step5：顯示畫面

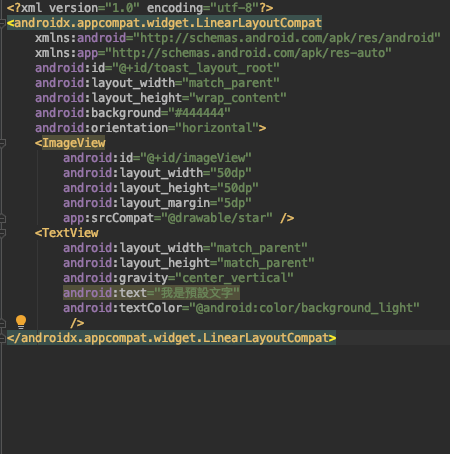
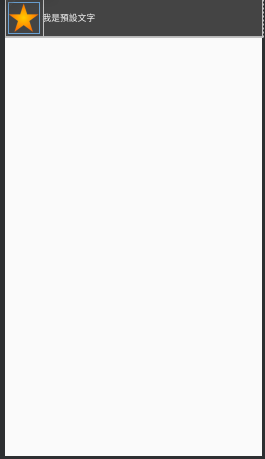
Step4：放入自定義的畫面

Step3：Toast在畫面中顯示的持續時間

Step1：初始化Toast

1. 我們需要自行創建出Toast實體，與makeText時雷同要傳入使用對象。
2. setGravity()方法可以指定我們的Toast位置。第一個的參數要傳入Toast要貼齊的方向。第二與第三個的參數則是傳入要與貼齊方向的長與寬間距。
3. setDuration()為持續時間，用法與makeText的第三個參數一樣。
4. Toast不只是能顯示文字，不過當我們希望呈現出更複雜的畫面，例如有圖片或文字，或兩者並存顯示的Toast時，就需要自行設計Xml畫面。我們可以與Activity設計顯示元件方式一樣，透過layout(xml)設計畫面，然後將完成的layout(xml)透過setView()方法來放入Toast中呈現。

* 如下為toast\_layout.xml即為要讓Toast顯示的畫面設計。

* 而程式中需要使用getLayoutInflater().inflate(R.layout.toast\_layout,(ViewGroup)findViewById.toast\_layout\_root)這方法來得到toast\_layout的layout。之後便可將該layout傳入setView()以設定顯示畫面。

View layout = getLayoutInflater().inflate(R.layout.*toast\_layout*

,(ViewGroup)findViewById(R.id.*toast\_layout\_root*));

toast.setView(layout);

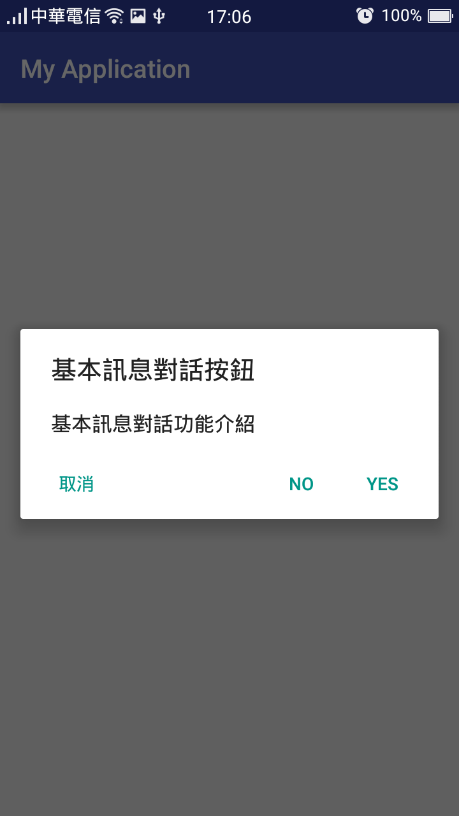
1. 透過show()的方法，將Toast做顯示。



1. **AlertDialog-對話方塊**
   1. **介紹**

當我們想要彈出訊息，並且希望使用者能與其互動，這時我們會使用AlertDialog。

AlertDialog對話方塊像Windows的彈跳視窗，他功能非常強大，不只是可以放上文字，還可以放上任何元件。



中立內容：setNeutralButton()

負面按鈕：setNegativeButton() setNegativeButton()

正面按鈕：setPositiveButton()

文字內容:setMessage()

標題: setTitle()

* 1. **用法**

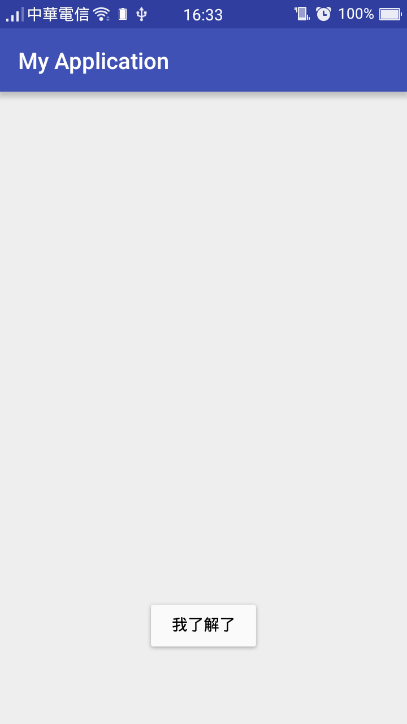
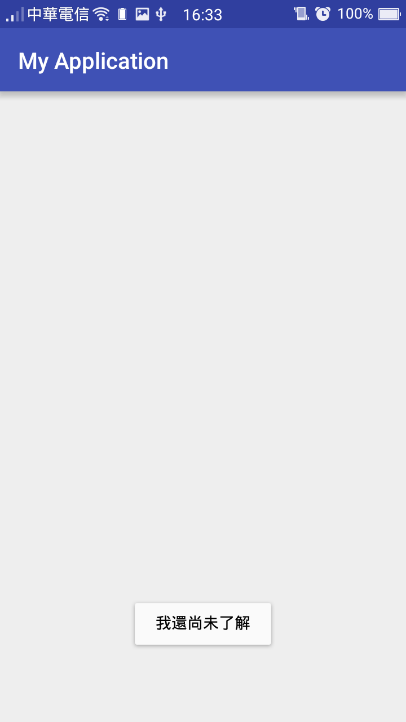
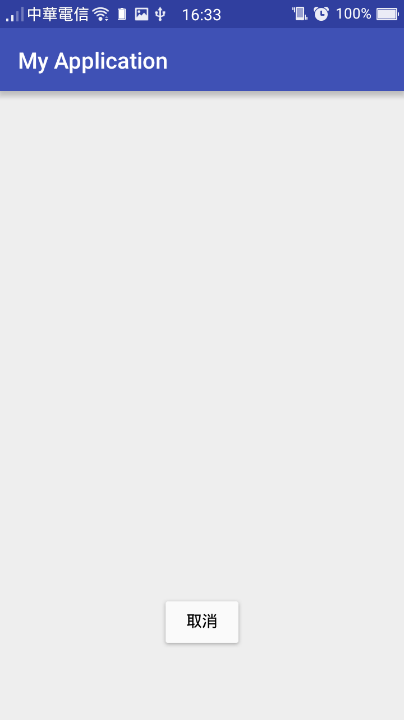
與Toast相比，AlertDialog的功能複雜很多，因此我們先從基本的提供的功能理解起：

* setTitle()：對話方塊的標題
* setMessage()：對話方塊的文字內容
* setItems()：對話方塊的列表內容
* setPositiveButton()：在對話方塊中加入正面的按鈕
* setNegativeButton()：在對話方塊中加入負面的按鈕
* setNeutralButton()：在對話方塊中加入中立的按鈕
* show()：顯示對話方塊

在產生的AlertDialog實體中，對話方塊會依據裝置的不同會有不同的顯示面板，以下是幾個實作的案例：

* **含確定、拒絕與取消按鈕的對話框**







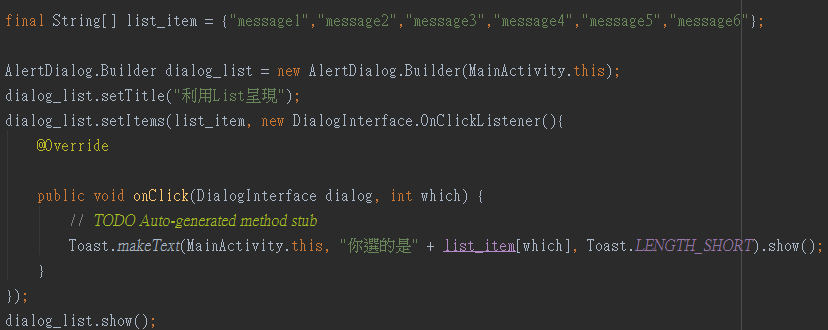
setPositiveButton()、setNegativeButton()、setNeutralButton()主要影響按鈕位置，實際使用時的可不依照定義去使用。其中，兩個參數中第一個是按鈕名稱，第二個則要傳入DialogInterface類別下的監聽器(OnClickListener)來做事件處理。

* **含列表的對話框**



標題: setTitle()

列表項目: setItems ()



建立AlertDialog物件

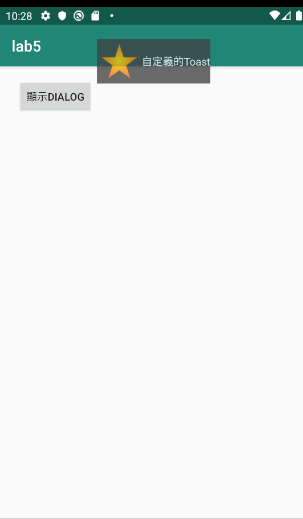
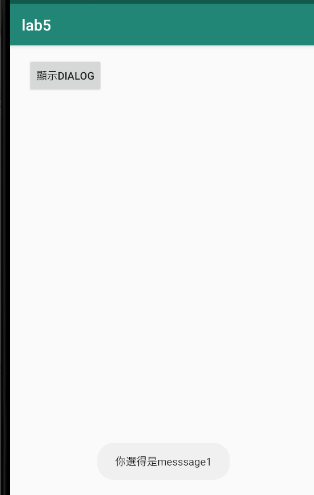
建立要顯示在的列表上的字串

依照被點擊的項目顯示字串

列表使用setItems()來顯示列表項目，第一個參數需要傳入一個字串陣列，第二個則要傳入DialogInterface類別下的監聽器(OnClickListener)來做事件處理， onClick事件處理的第二個參數會回傳點擊的項目編號(依照陣列的順序)。

1. 設計重點：

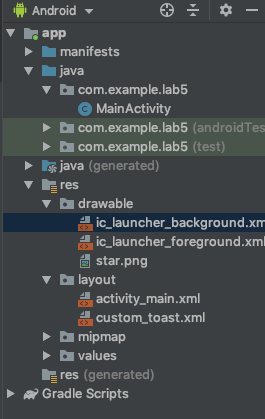
* 設計一個AlertDialog，並根據下方按鈕選擇顯示不同的功能
* 點擊”顯示LIST”按鈕，顯示AlertDialog的List
* 點擊”自定義TOAST”按鈕，顯示客製化的Toast
* 點擊”取消”按鈕，顯示原生Toast

**** ** **

1. 設計步驟:

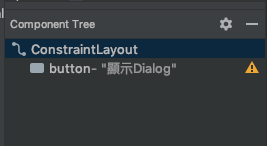
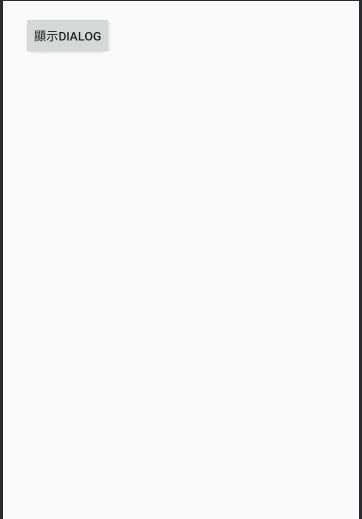
Step1

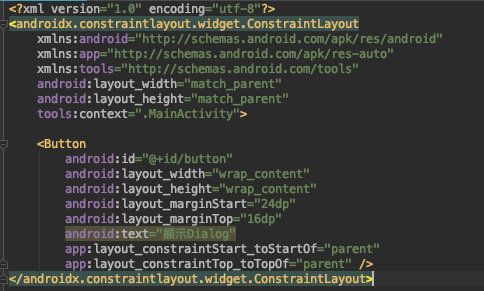
建立專案，並將附件的star.png放於drawable底下

****

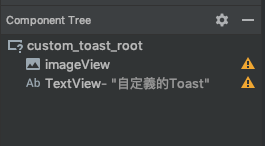
Step2

1. 繪製activity\_main.xml

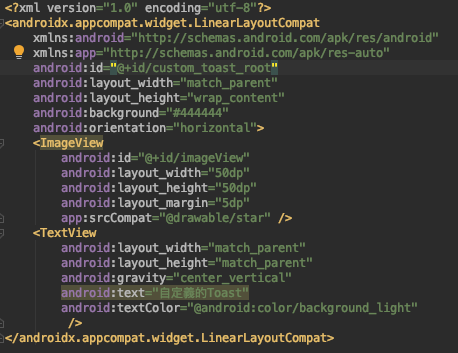


對應的xml如下：

1. 繪製custom\_toast.xml，來顯示客製化的toast樣式

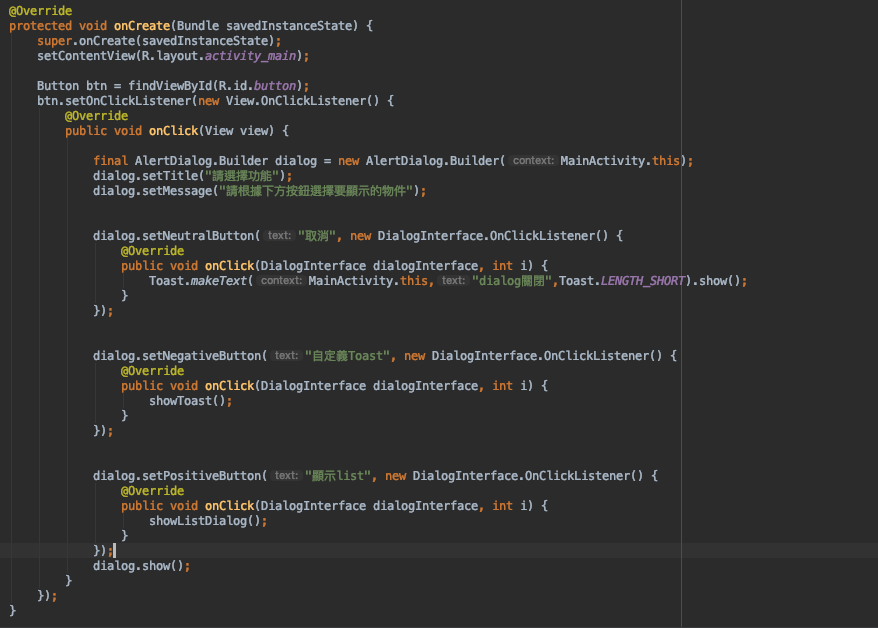
對應的xml如下：

****

Step3

編寫MainActivity的程式

1. 編寫含確定、拒絕與取消按鈕的對話框。其中NeutralButton執行Toast.makeText()方法，NegativeButton執行副程式showToast()，PositiveButton執行副程式showListDialog()。



Button點擊事件

建立AlertDialog物件

執行副程式來顯示含列表的對話框

執行副程式來顯示客製化Toast

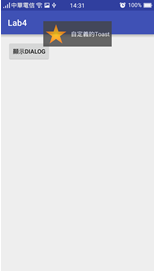
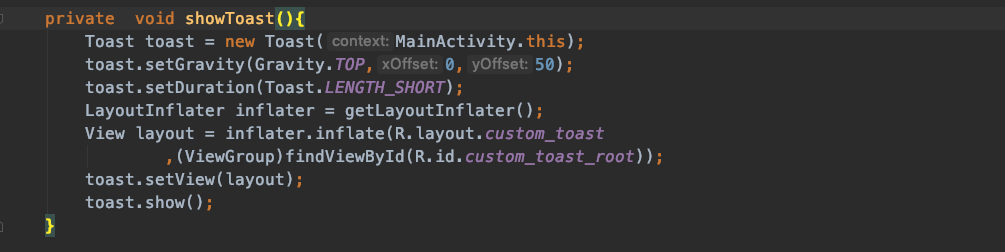
使用makeText顯示訊息

連接Button元件

連接main1.xml畫面

初始化Activity

1. showToast()中編寫顯示客製化Toast

****

Step4：取得自定義的畫面

Step2：Toast在畫面中顯示位置

Step1：初始化Toast

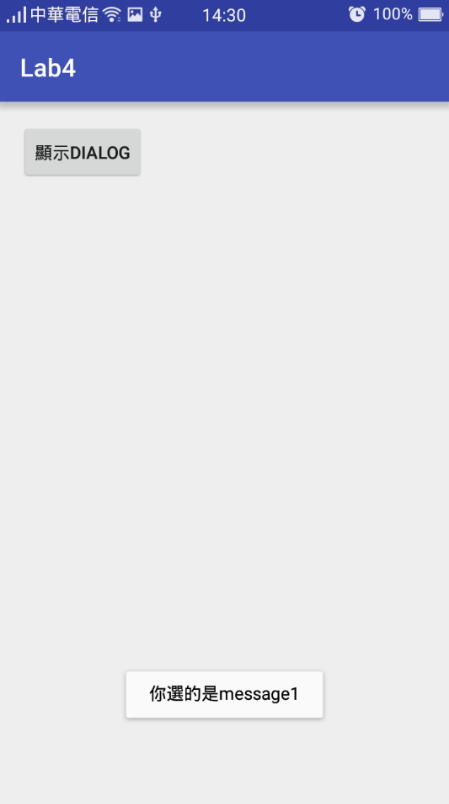
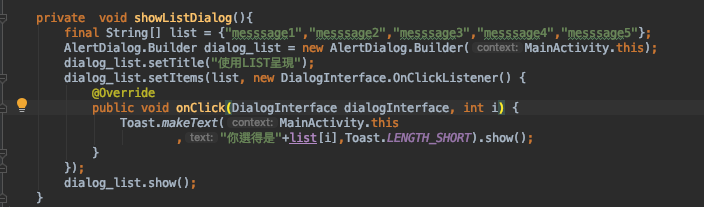
Step3：Toast在畫面中顯示的持續時間

Step5：放入自定義的畫面(custom\_toast.xml)

Step6：顯示畫面

1. showListDialog ()中編寫含列表的對話框

建立要顯示在的列表上的字串

****

建立AlertDialog物件

依照被點擊的項目用Toast顯示字串