Lab6
提示訊息元件

本節目的：

- 利用 Toast 的方法顯示文字訊息。
- 客製化 Toast 顯示文字與圖片訊息。
- 透過 AlertDialog 顯示提示訊息與陣列資料。

6.1 顯示訊息：

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。

6.1.1 Toast-快顯訊息

Toast 是一種快速的即時訊息，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，如圖 6-1 所示。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。

圖 6-1 Toast 出現幾秒後自動消失

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：

```
Toast.makeText(this, "文字訊息", Toast.LENGTH_SHORT).show()
```

makeText 的第一個參數要傳入呼叫 Toast 的對象，通常下應要填入本身的 Activity 實體（this）。第二參數傳入字串，作為輸出畫面的內容。第三參數為持續時間，「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，就可以利用 show()將訊息顯示到螢幕上。

■ 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：

```
//Step1：初始化 Toast
val toast = Toast(this)
//Step2：Toast 在畫面中顯示位置
toast.setGravity(Gravity.TOP, 0, 50)
//Step3：Toast 在畫面中顯示的持續時間
toast.duration = Toast.LENGTH_SHORT
//Step4：放入自定義的畫面
toast.view = layoutInflater.inflate(R.layout.toast_custom,
                                    null)
//Step5：顯示畫面
toast.show()
```

Step1 我們需要自行創建出 Toast 實體，與 makeText 時雷同要傳入使用對象。

Step2 setGravity()方法可以指定我們的 Toast 位置。第一個參數要傳入 Toast 要貼齊的方向。第二與第三個參數則是傳入要與貼齊方向的長與寬間距。

Step3 duration 為持續時間，用法與 makeText 的第三參數一樣。

Step4 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）指定為 view 來放入 Toast 中呈現。

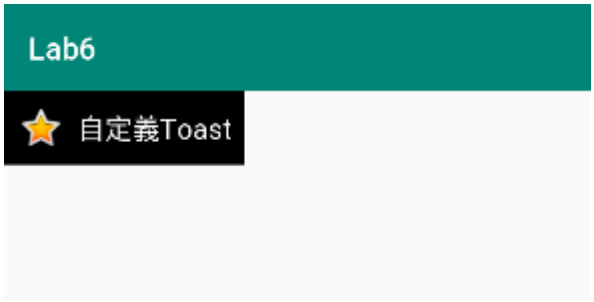


圖 6-2 自定義 Toast layout

Step5 透過 show()方法，將 Toast 做顯示如圖 6-3 所示。

圖 6-3 顯示自定義 Toast

6.1.2 AlertDialog-對話方塊

當我們想要彈出一個訊息，並且希望使用者能與其互動，這時我們會使用 AlertDialog，AlertDialog 對話方塊很像 Windows 上的彈跳視窗，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，如圖 6-4 所示。

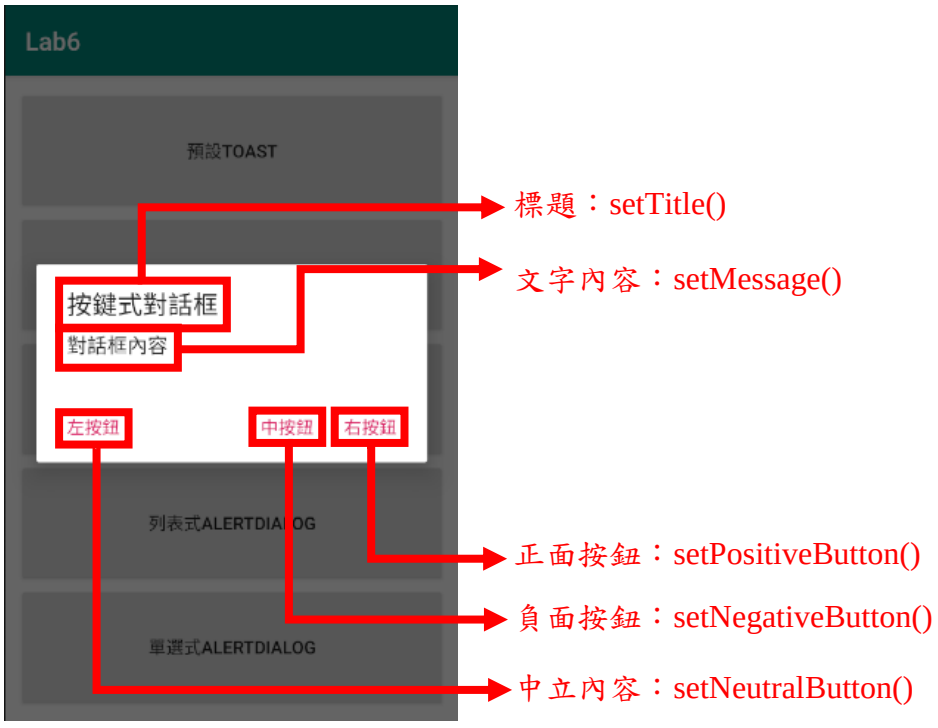


圖 6-4 按鍵式對話框

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()：在對話方塊加入的列表內容。
- setSingleChoiceItems()：在對話方塊加入單選列表。
- 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

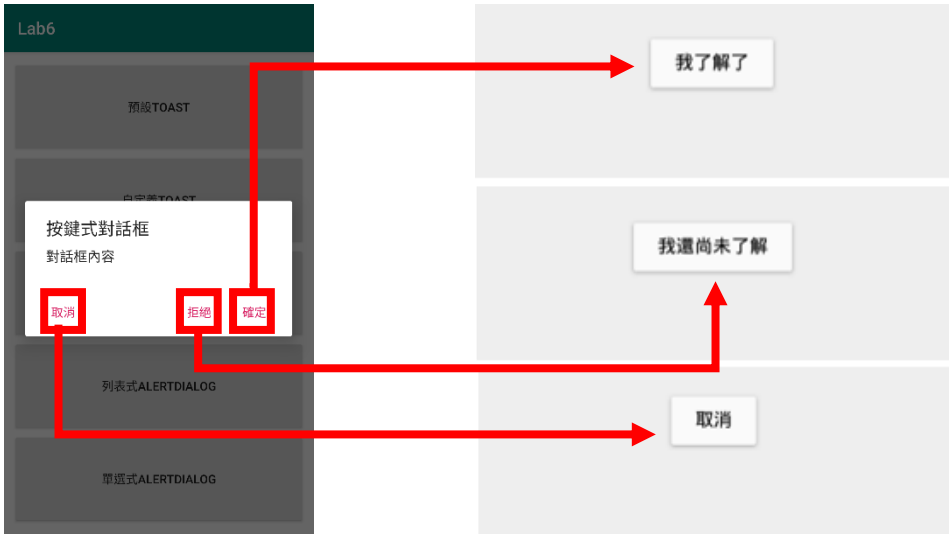


圖 6-5 按下對話框按鈕（左）與顯示對應文字（右）

```
AlertDialog.Builder(this)
    .setTitle("按鍵式對話框")
    .setMessage("對話框內容")
    .setNeutralButton("取消") { dialog, which ->
        //顯示取消按鈕被點擊
        Toast.makeText(this,"取消", Toast.LENGTH_SHORT).show()
    }
    .setNegativeButton("拒絕") { dialog, which ->
```

```
        //顯示拒絕按鈕被點擊
        Toast.makeText(this,"拒絕", Toast.LENGTH_SHORT).show()
    }
    .setPositiveButton("確定") { dialog, which ->
        //顯示確定按鈕被點擊
        Toast.makeText(this,"確定", Toast.LENGTH_SHORT).show()
    }.show()
```

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

圖 6-6 列表式對話框

```
//建立要顯示在的列表上的字串
val list_item = arrayOf("對話框選項 1", "對話框選項 2", "對話框選項 3", "對話框選項 4", "對話框選項 5")
//建立 AlertDialog 物件
AlertDialog.Builder(this)
    .setTitle("列表式對話框")
    .setItems(list_item) { dialogInterface, i ->
        //顯示被點擊的項目
        Toast.makeText(this,"你選的是" + list_item[i],
```

```
                                        Toast.LENGTH_SHORT).show()
    }.show()
```

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 事件處理的第二參數會回傳點擊的項目編號（依照陣列的順序）。

■ 單選式的對話框

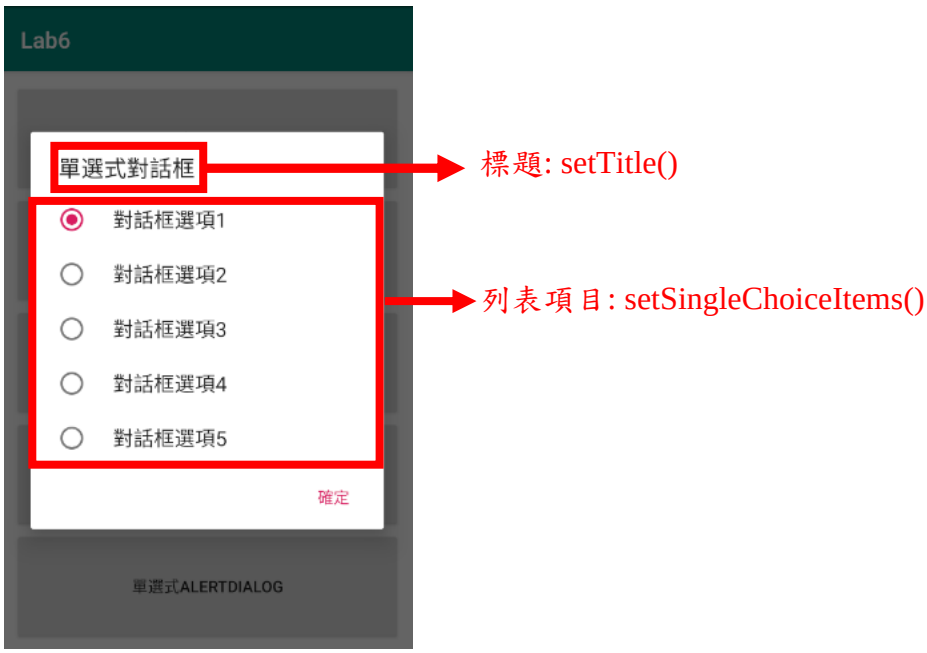


圖 6-7 單選式對話框

```
val list_item = arrayOf("對話框選項 1", "對話框選項 2", "對話框選項 3", "對話框選項 4", "對話框選項 5")
var position = 0
//建立 AlertDialog 物件
AlertDialog.Builder(this)
    .setTitle("單選式對話框")
    .setSingleChoiceItems(list_item, 0) { dialogInterface, i ->
        //記錄被按下的位置
        position = i
    }
    .setPositiveButton("確定") { dialog, which ->
        //顯示被點擊的項目
        Toast.makeText(this,"你選的是" + list_item[position],
```

```
                                        Toast.LENGTH_SHORT).show()
    }.show()
```

列表使用 setSingleChoiceItems()來顯示單選列表項目，第一個參數需要傳入一個字串陣列，第二參數是預設按下的選項，第三則要傳入 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）。

6.2 提示訊息演練

- 圖 6-8 設計一個 APP，並根據按鈕選擇顯示不同的提示訊息。
- 點擊「Toast」按鈕，顯示預設 Toast 或客製化 Toast。
- 點擊「AlertDialog」按鈕，顯示按鈕式、列表式或單選式 AlertDialog。

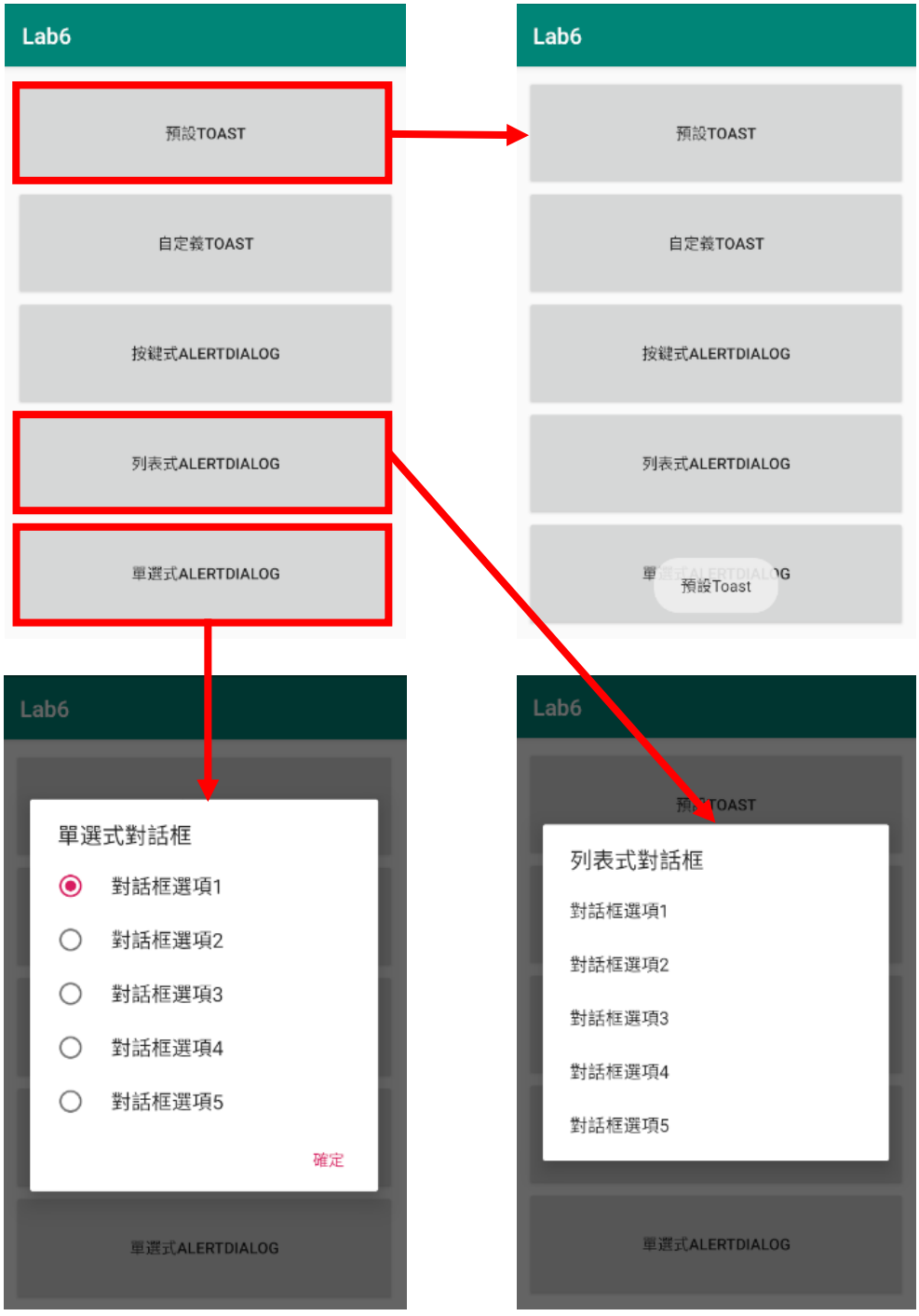


圖 6-8 APP 預覽畫面（左一）、顯示預設 Toast（右一）、
顯示單選式對話框（左二）、顯示列表式對話框（右二）

6.2.1 畫面布局與客製化 Toast

Step1 建立新專案，以及圖 6-9 對應的 class 與 xml。

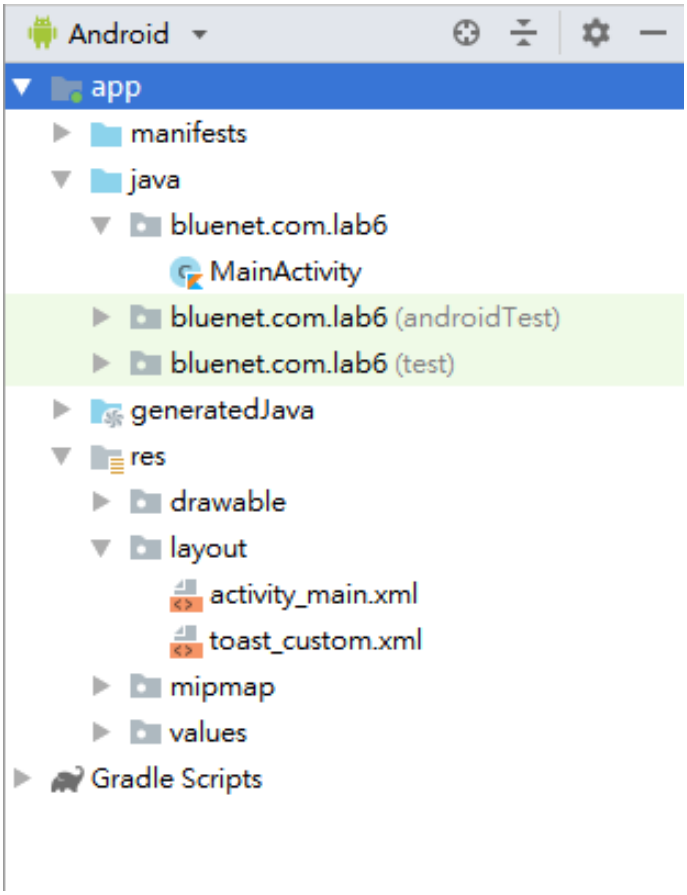


圖 6-9 APP 專案架構

Step2 繪製 activity_main.xml，如圖 6-10 所示。

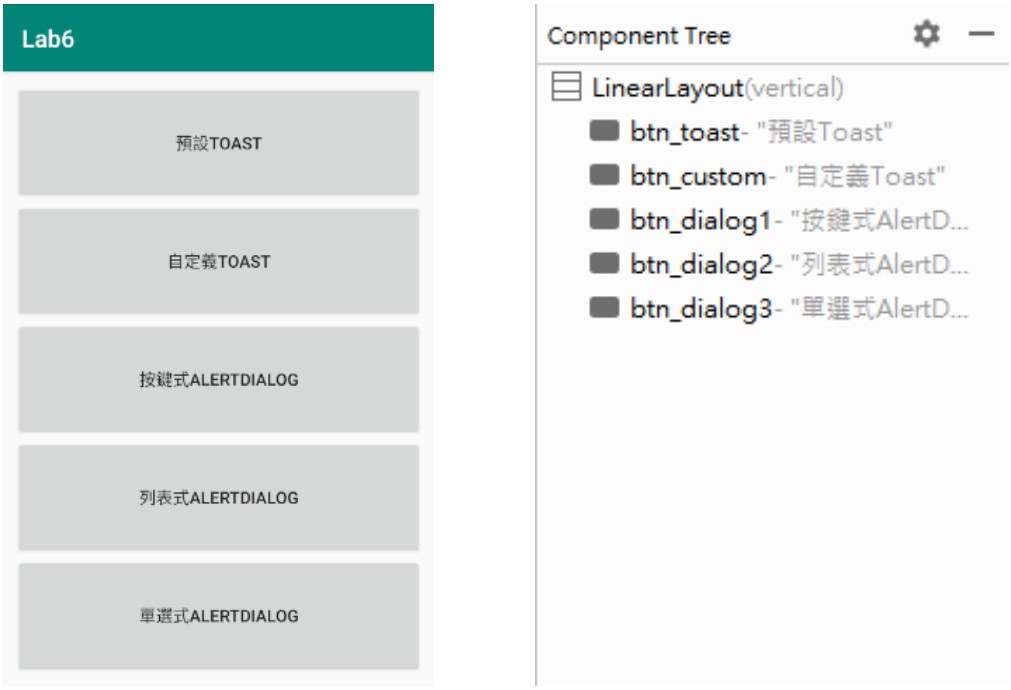


圖 6-10 APP 預覽畫面（左）與布局元件樹（右）

對應的 xml 如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:tools="http://schemas.android.com/tools"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:padding="10dp"
            tools:context=".MainActivity">

    <Button
            android:id="@+id/btn_toast"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="預設 Toast" />

    <Button
```

```
            android:id="@+id/btn_custom"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="自定義 Toast" />

    <Button
            android:id="@+id/btn_dialog1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="按鍵式 AlertDialog" />

    <Button
            android:id="@+id/btn_dialog2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="列表式 AlertDialog" />

    <Button
            android:id="@+id/btn_dialog3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="單選式 AlertDialog" />
</LinearLayout>
```

Step3 繪製 custom_toast.xml，如圖 6-11 所示，顯示客製化的 toast 樣式。

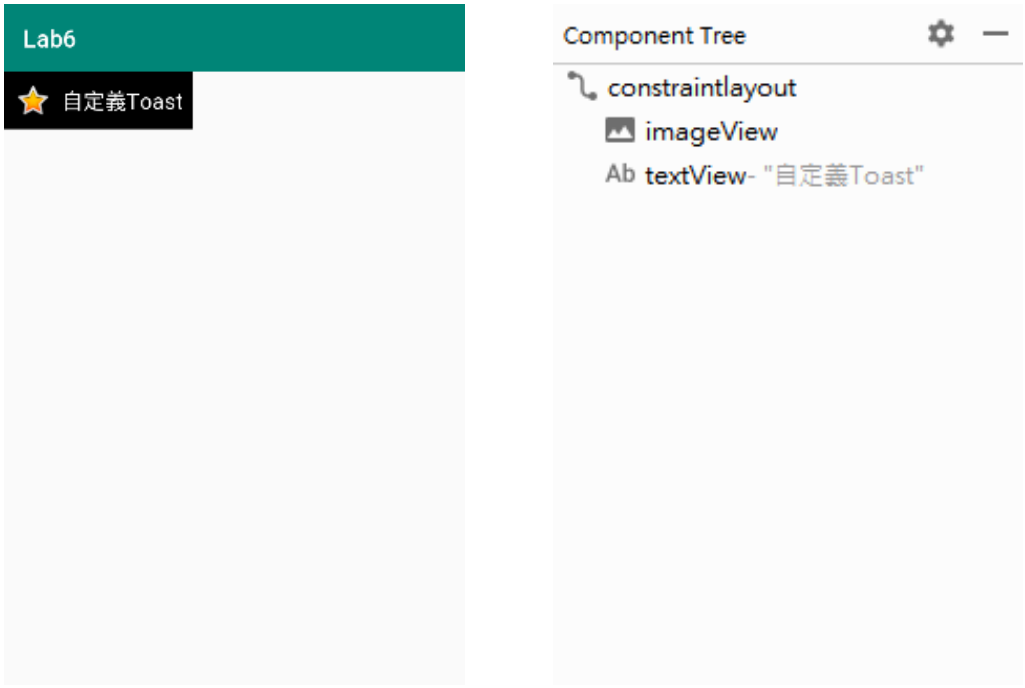


圖 6-11 客製化 Toast 預覽畫面與 Toast 布局元件樹

對應的 xml 如下：

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@android:color/black"
    android:id="@+id/constraintlayout">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginBottom="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
```

```
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@android:drawable/btn_star_big_on" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:text="自定義 Toast"
        android:textSize="18sp"
        android:textColor="@android:color/white"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/imageView"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

6.2.2 加入對話框監聽事件

Step1 編寫 MainActivity 的程式，加入一般 Toast 與按鈕式對話框，按下選擇鈕後執行對應的 Toast.makeText()方法。

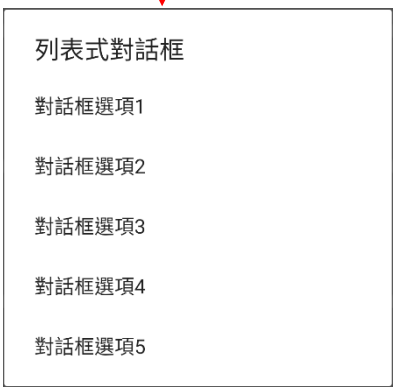
```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        //建立要顯示在的列表上的字串
        val list_item = arrayOf("對話框選項 1", "對話框選項 2", "
對話框選項 3", "對話框選項 4", "對話框選項 5")
        //Button 點擊事件
        btn_toast.setOnClickListener {
        //使用 Toast 顯示訊息
            Toast.makeText(this,"預設 Toast",
                                    Toast.LENGTH_SHORT).show()
        }

        btn_dialog1.setOnClickListener {
            //建立 AlertDialog 物件
            AlertDialog.Builder(this)
                .setTitle("按鍵式對話框")
                .setMessage("對話框內容")
                .setNeutralButton("左按鈕") { dialog, which ->
                    Toast.makeText(this,"左按鈕",
                                    Toast.LENGTH_SHORT).show()
                }
                .setNegativeButton("中按鈕") { dialog, which ->
                    Toast.makeText(this,"中按鈕",
                                    Toast.LENGTH_SHORT).show()
                }
                .setPositiveButton("右按鈕") { dialog, which ->
                    Toast.makeText(this,"右按鈕",
                                    Toast.LENGTH_SHORT).show()
                }.show()
        }
    }
}
```

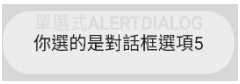
Step2 撰寫客製化 Toast，使用 setItems()加入清單列表。

```
btn_custom.setOnClickListener {
    //宣告 Toast
    val toast = Toast(this)
    //Toast 在畫面中顯示位置
    //toast.setGravity(Gravity.TOP, 0, 50)
    //Toast 在畫面中顯示的持續時間
    toast.duration = Toast.LENGTH_SHORT
    //放入自定義的畫面(custom_toast.xml)
    toast.view =
            layoutInflater.inflate(R.layout.toast_custom,null)
    //顯示畫面
    toast.show()
}
```

Step3 撰寫列表式對話框，使用 setItems()加入清單列表。

```
btn_dialog2.setOnClickListener {
    //建立 AlertDialog 物件
    AlertDialog.Builder(this)
        .setTitle("列表式對話框")
        .setItems(list_item) { dialogInterface, i ->
            Toast.makeText(this,"你選的是" + list_item[i],
                                    Toast.LENGTH_SHORT).show()
        }.show()
}
```





Step4 撰寫單選式對話框，使用 setSingleChoiceItems()加入單選列表。

```
btn_dialog3.setOnClickListener {
    var position = 0
    //建立 AlertDialog 物件
    AlertDialog.Builder(this)
        .setTitle("單選式對話框")
        .setSingleChoiceItems(list_item, 0) {
           dialogInterface, i ->
           position = i
        }
        .setPositiveButton("確定") { dialog, which ->
            Toast.makeText(this,"你選的是" +
                 list_item[position], Toast.LENGTH_SHORT).show()
        }.show()
}
```