ANDROID

UI 設計

Android 專案目錄架構

- Android 專案建立後會自動產生3個主要目錄
- **src**：Java程式檔案
- **res**：資源(文字、圖形、聲音檔案等)、與UI設定有關的layout檔
  - 此目錄內檔案名稱只能為小寫字母、數字、_、.
- **gen**：R.java 根據res目錄內容自動產生
  - 不要去修改R.java

- Android 中所有的資源檔案（圖片、XML等）命名都必須使用英文小寫，檔名中間只允許加上底線符號。其他的檔名都會造成無法正常產生R.java 檔，讓你的程式無法編譯

res/

- drawable 目錄提供圖形相關資源，支援 PNG、JPG、GIF 檔案
  - 依照螢幕解析度高低而分成 drawable-hdpi (高解析度)、drawable-mdpi (中解析度)、drawable-ldpi (低解析度)
  - 建議相同圖形依照不同解析度製作成三份，分別放在對應的目錄下，當行動裝置會自動判斷最適當的解析度去呈現
- layout 目錄存放與 UI設計有關的 layout 檔案 (預設為 main.xml)
- values目錄存放 UI所需用到的文字 (預設為 strings.xml)

XML

- Android 為了單純化介面修改方式，將介面描述部份的程式碼，抽取到程式外部的 XML 描述文件中
- XML (Extensible Markup Language) 是一種標記描述語言
- Android 文件網站上各種可用介面元件列表
- http://developer.android.com/guide/tutorials/views/index.html

版面配置 LAYOUT

Hello World! 預設程式介面講解

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3    android:orientation="vertical"
4    android:layout_width="fill_parent"
5    android:layout_height="fill_parent"
6    >
7     <TextView
8        android:layout_width="fill_parent"
9        android:layout_height="wrap_content"
10       android:text="@string/hello_world"
11       />
12 </LinearLayout>
```

左側 res →layout→activity_main.xml 會開啟GUI畫面

第 1 行

「main.xml」文件裡，第一行是每個 XML 描述檔固定的開頭內容，用來指示這個文字檔案是以 XML格式描述的

第 2, 6 與 12 行

- "線性版面配置"(LinearLayout)標籤，使用了兩個「LinearLayout」標籤，來表示一個介面元件的區塊 <LinearLayout></LinearLayout>
- 後頭的標籤前加上一個「/」符號來表示結束標籤
- 包含在「LinearLayout」標籤中，所有元件的配置方式，是將一個接一個元件由上而下排下來
- 注意標籤需要兩兩對稱。一個標籤<linearlayout>在一串敘述的前頭，另一個標籤</linearlayout>在敘述的末尾

第 3-5 行

- 為「LinearLayout」標籤的「屬性」，在 layout 目錄中的標籤，大多數的屬性前都有一個「android:」前綴
- 介面元件都有許多共同的屬性， Android 介面元件的寬度、長度設定屬性分別叫做「android:layout_width」、「android:layout_height」

–設定為「fill_parent」參數值，就是"填滿整個上層元件"

- 介面元件彼此間也會有一些不同的屬性，例如「LinearLayout」標籤的「android:orientation」(版面走向) 屬性

–「vertical」(垂直)屬性值，表示這個介面的版面配置方式是從上而下垂直地排列其內含的介面元件

第 7 和 11 行

- TextView (文字檢視) 作用是顯示文字到螢幕上

第 8-10 行

- TextView 介面元件中包含的屬性
- 「android: layout_width」的「fill_parent」參數值表示寬度填滿整個上層介面元件(即 LinearLayout 介面元件)
- 「android:layout_height」則是用上一個新的參數值「wrap_content」(包住內容)，即隨著文字欄位行數的不同而改變這個介面元件的高度
- 最後的「android:text」屬性則是 TextView 介面元件的主要屬性，亦即文字欄位中顯示的文字內容

layout的基礎類別

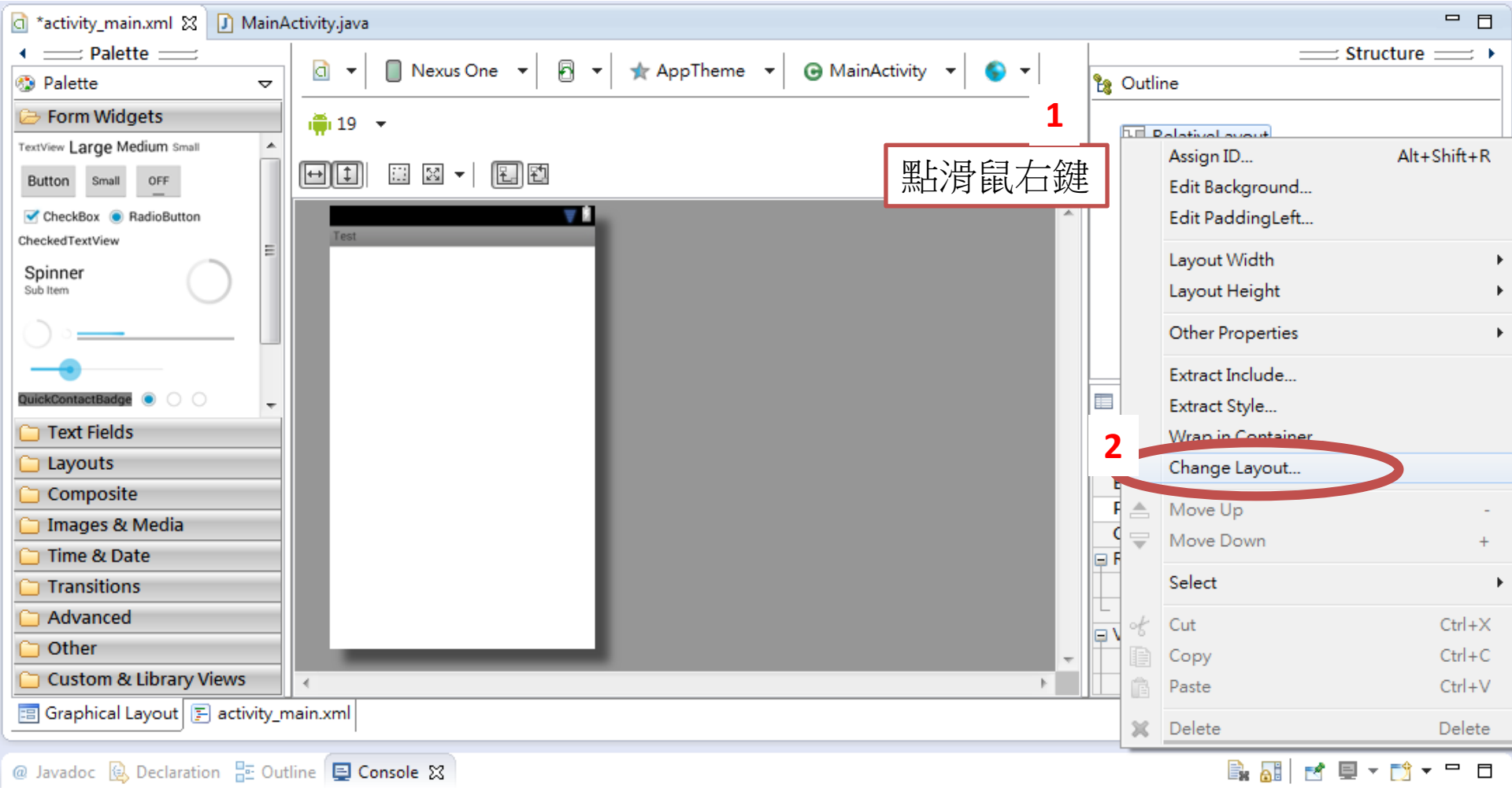
- 各種佈局配置(layout)元件的基礎類別，常見的有
- LinearLayout(線性版面配置)
- FrameLayout(框架版面配置)
- TableLayout(表格版面配置)
- AbsoluteLayout(絕對位置版面配置)
- RelativeLayout(相對位置版面配置)

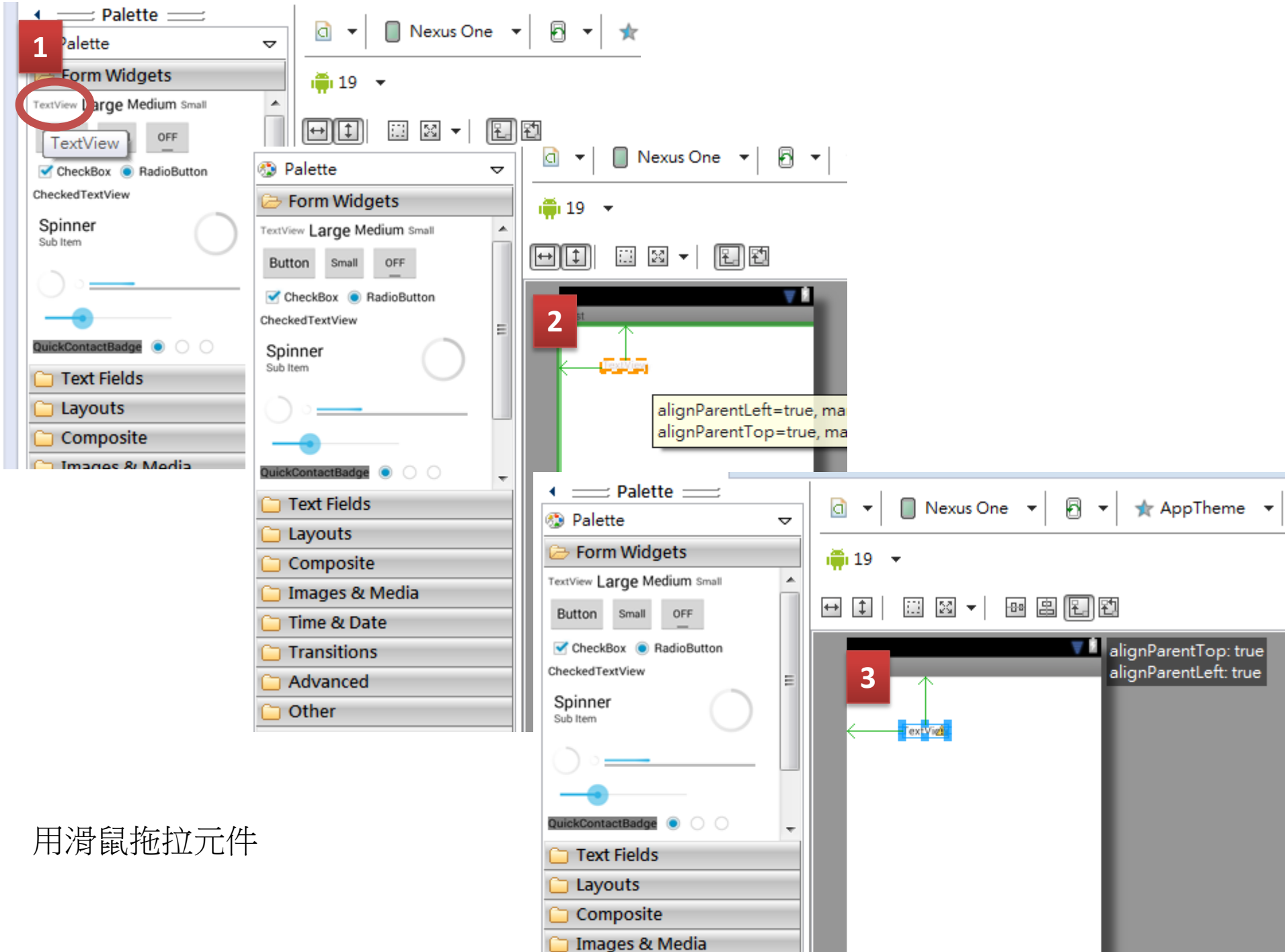
介面元件

視圖(View)

- 原先已預設了一個Linear Layout介面元件以及一個”Hello,World”字樣的textView介面元件→將這兩個元件全部刪除
- 從左側Palette拉4個元件放到畫面
  - TextView(用來提示填入資料)
  - EditText(文字輸入欄位用來填入資料)
  - Button(一個按鈕來做程式運算)
  - TextView(顯示結果)
- 選擇物件，在 Properties(屬性設定窗)裡找到 Text 設定它的Value

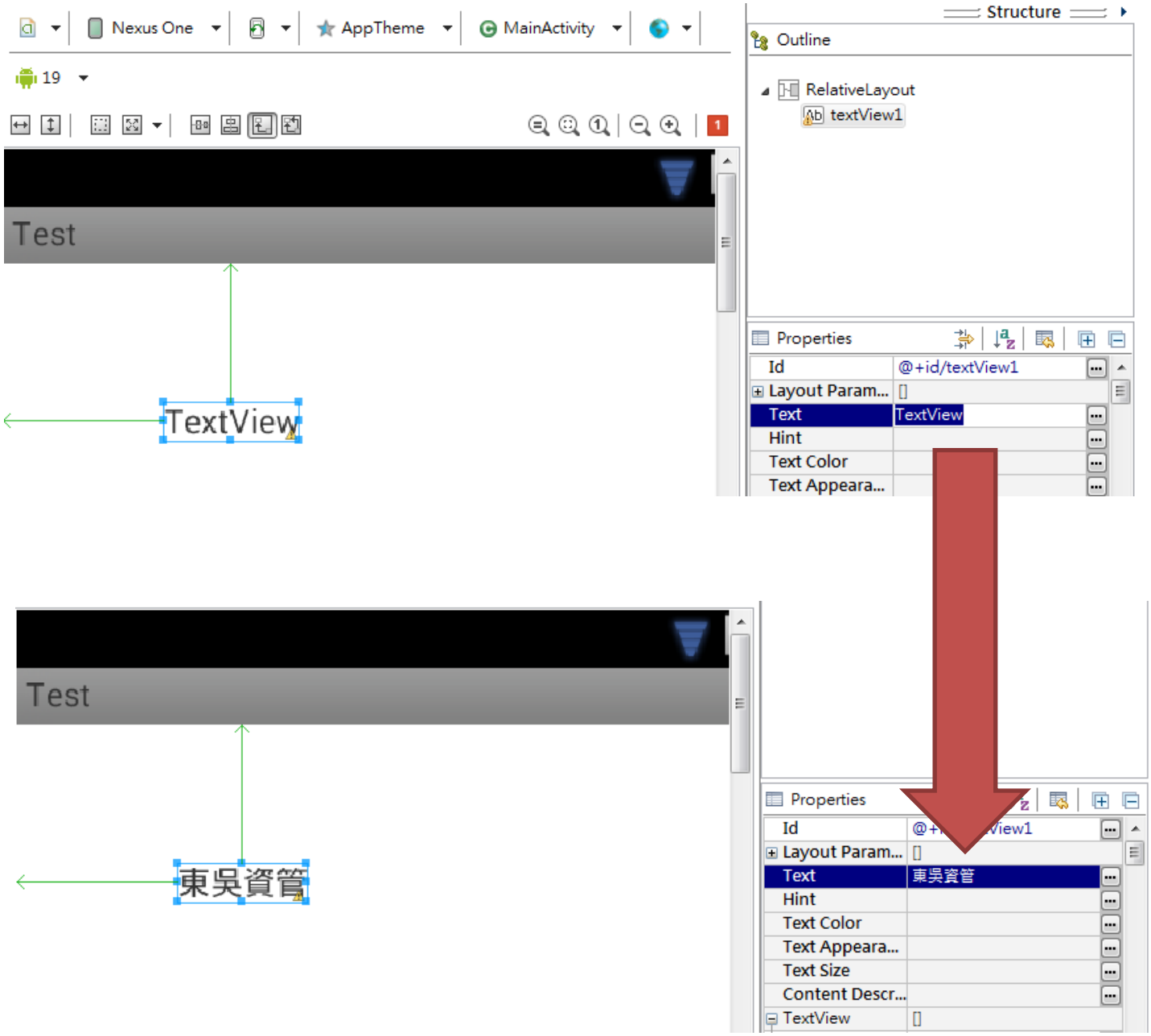
更改Layout


*activity_main.xml
MainActivity.java
Palette
Form Widgets
Text Fields
Layouts
Composite
Images & Media
Time & Date
Transitions
Advanced
Other
Custom & Library Views
Nexus One
AppTheme
MainActivity
19
1
點滑鼠右鍵
Test
Structure
Outline
Assign ID... Alt+Shift+R
Edit Background...
Edit PaddingLeft...
Layout Width
Layout Height
Other Properties
Extract Include...
Extract Style...
Wrap in Container
2
Change Layout...
Move Up
Move Down
Select
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Graphical Layout
activity_main.xml
Javadoc
Declaration
Outline
Console


1
Palette
Form Widgets
TextView
Large Medium Small
TextView
OFF
CheckBox
RadioButton
CheckedTextView
Spinner
Sub Item
QuickContactBadge
Text Fields
Layouts
Composite
Nexus One
19
2
alignParentLeft=true, ma
alignParentTop=true, ma
Button
Small
OFF
Images & Media
Time & Date
Transitions
Advanced
Other
AppTheme
3
alignParentTop: true
alignParentLeft: true

用滑鼠拖拉元件

選擇物件，在 Properties(屬性設定窗)裡找到Text 設定它的Value



元件全部放好長這樣

開始設計

- 在頁面標籤下選擇「main.xml」標籤，來檢視程式碼
- 定義一個姓名輸入欄位，就會用到 EditText，與 TextView 介面元件
  - 「android:text」屬性是指定 EditText介面元件預設顯示的文字

```
1  <TextView
2     android:layout_width=  “fill_parent”
3     android:layout_height=  “wrap_content”
4     android:text="姓名："
5     />
6  <EditText android:id="@+id/name"
7     android:layout_width="fill_parent"
8     android:layout_height="wrap_content"
10     android:text=""
11     />
```

- Button (按鈕)介面元件
  - 「android:text」屬性用來顯示按鈕上的文字

```
<Button android:id="@+id/submit"
        android:layout_width= " fill_parent"
        android:layout_height= " wrap_content "
        android:text="確定"
        />
```

- 從文件中挑出我們需要的 TextView(文字檢視)、EditText(編輯文字)、Button(按鈕) 三種介面元件，照前面的設計擺進 LinearLayout (線性版面配置)元件中。完整的「main.xml」介面描述檔如下：

```
<?xml version="1.0" encoding="utf-8"?>
   <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
      android:orientation="vertical"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      >
      <TextView
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
         android:text="姓名： "
         />
      <EditText android:id="@+id/name"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
         android:text=""
         />
     <Button android:id="@+id/submit"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
         android:text="確定"
         />
      <TextView android:id="@+id/result"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
         android:text=""
         />
   </LinearLayout>
```

- 可以啟動模擬器檢視執行結果。或是在頁面標籤下選擇「Layout」標籤，來預覽頁面配置

按下"確定" 按鈕後，應用程式完全沒反應!

- 我們還要
  - 處理從介面輸入取得姓名
  - 將輸入資料加上運算
  - 將結果輸出到螢幕上

存取識別符號

- 對於那些將在程式中被參考(reference)到的介面元件（如按鈕 Button、文字輸入欄位 EditText），我們需要先在 XML描述檔中，定義該介面元件的「android:id」識別符號屬性
- 在程式碼中我們要操作這個介面元件（取出欄位中輸入的資料、取得按下按鈕事件...）時，就能透過「android:id」識別符號來調用這個介面元件

```
<EditText android:id="@+id/name"
  />
```

- 「name」是這個介面元件的 android:id

將字串抽離 XML

- 開啟 res/values/strings.xml
- 原始的內容為

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">Test</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>

</resources>
```

  - 「app_name」，用來表示應用程式名稱
- 將剛才XML檔中的文字敘述抽取出來，整理進 strings.xml 檔案
- 完整的 strings.xml 檔案如下：

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">Test</string>
    <string name="action_settings">Settings</string>
    <string name="test_name">姓名</string>
    <string name="btn">確定</string>
    <string name="test_result">歡迎訊息</string>

</resources>
```

- 存取 string 類型的格式，來取代 main.xml 檔案中原本寫死的文字敘述

```
<?xml version="1.0" encoding="utf-8"?>
  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/text_name" />

    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        />

    <Button
        android:id="@+id/submit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/btn" />

    <TextView
        android:id="@+id/result"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/text_result" />
  </LinearLayout>
```

主要程式邏輯

預設的內容

- 打開「src/ csim.scu.test 」目錄下的「 MainActivity.java」檔案

```
package csim.scu.test;

import android.os.Bundle;
import android.app.Activity; //在所有的 Android 應用程式中都會用到這兩個 Package
import android.view.Menu;

public class MainActivity extends Activity {  //開始程式的主體
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); //「super」關鍵字代表「Bmi」類別的上層類別(Activity)，執行 Activity 類別中 onCreate 方法的內容
        setContentView(R.layout.activity_main);
    }
}
```

- 「onCreate」方法傳入了一個名為「savedInstanceState」的「Bundle」型別參數
  - 只要知道「Bundle」的內容與手機平台的記憶體管理有關
  - 當 Android 應用程式啟動、換到背景等待、關閉時，都會用到「savedInstanceState」這個實體來處理記憶體相關的事
- 「@Override」語句的作用是告訴程式我們要覆寫這個「onCreate」方法。當我們打開程式時，程式不再使用從「bmi」類別中繼承來的「onCreate」方法，而是使用我們在程式中自訂的行為
- 「onCreate」 方法則是每個 Activity 類別初始化時都會去呼叫的方法。我們想做的事，是保持原本「onCreate」 方法預設的動作，然後在其中加入我們想要的內容
- 原本繼承自 「Activity」類別的「onCreate」方法，其原本內容都被覆載掉了。因此想將原本的「onCreate」方法內容保留，並在其中加入我們的內容的話，就要使用「super」語句。當程式運行到我們覆寫的「onCreate」 方法時，透過「super.onCreate(savedInstanceState);」語句，會先將原本「Activity」類別中的「onCreate」方法執行一次，然後再執行我們覆寫的「onCreate」方法裡面其他的程式內容

導入其他用到的模組、取得介面元件、對按鈕設定動作

專案 exercise： TextView(文字檢視)、EditText(編輯文字)、Button(按鈕) 介面元件

http://ycajd.weebly.com/android251632723125033299923124324335382833033 2.html

範例程式

- **Intentbundle**：切換頁面顯示
- **helloscu**：**圖形按鈕 + 圖片**
- **animals**：**簡易物件導向範例**