

Android

程式設計入門

Jimmy Chang
jimmy@airbridge.com.tw

大綱

- Eclipse開發環境安裝設定
- 第一個App & 目錄說明
- Android Layout探討
- Activity探討
- UI使用注意事項
- 介面元件介紹實例
- 專案開發實例

開發環境安裝

- Step1 安裝Java Software Developer Kit (JDK)
 - Java開發工具包
- Step2 安裝Eclipse (IDE)
 - 開發環境
- Step3 安裝Eclipse外掛Android Developer Tools(ADT)
 - 方便進行編輯、儲存、測試、除錯
- Step4 安裝Android軟體開發工具包(SDK)
 - 提供完整Android API
- Step5 安裝模擬器 Android Virtual Device (AVD)












下載Java SDK

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Java SE Development Kit 8u25

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux x86	135.24 MB	 jdk-8u25-linux-i586.rpm
Linux x86	154.88 MB	 jdk-8u25-linux-i586.tar.gz
Linux x64	135.6 MB	 jdk-8u25-linux-x64.rpm
Linux x64	153.42 MB	 jdk-8u25-linux-x64.tar.gz
Mac OS X x64	209.13 MB	 jdk-8u25-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	137.01 MB	 jdk-8u25-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	97.14 MB	 jdk-8u25-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	137.11 MB	 jdk-8u25-solaris-x64.tar.Z
Solaris x64	94.24 MB	 jdk-8u25-solaris-x64.tar.gz
Windows x86	157.26 MB	 jdk-8u25-windows-i586.exe
Windows x64	169.62 MB	 jdk-8u25-windows-x64.exe

下載Android SDK

<http://developer.android.com/sdk/index.html>

The screenshot shows the Android Developer website with the 'Develop' tab selected. A 'Download' dropdown menu is open, highlighting 'Installing the SDK' and 'Adding SDK Packages'. The main content area features the Android Studio logo and the text 'The official Android IDE'. A list of links includes 'Android Studio IDE', 'Android SDK tools', 'Android 5.0 (Lollipop) Platform', and 'Android 5.0 emulator system image with Google APIs'. A green button labeled 'Download Android Studio for Mac' is visible. Below the button are links for 'System Requirements', 'Other Download Options', 'Migrating to Android Studio', and 'Take a Survey'. The background image shows a laptop displaying the Android Studio IDE with a code editor and a virtual device emulator.

Developers | Design | **Develop** | Distribute

Training | API Guides | Reference | **Tools** | Google Services | Samples

Download

- Installing the SDK
- Adding SDK Packages

Android Studio

- Workflow
- Tools Help
- Build System
- Support Library
- Revisions
- NDK
- ADK
- Eclipse with ADT

Android Studio

The official Android IDE

- Android Studio IDE
- Android SDK tools
- Android 5.0 (Lollipop) Platform
- Android 5.0 emulator system image with Google APIs

Download Android Studio for Mac

- System Requirements
- Other Download Options
- Migrating to Android Studio
- Take a Survey



Developers ▾

Design

Develop

Distribute

Training

API Guides

Reference

Tools

Google Services

Samples

Download ^

Installing the SDK

Adding SDK Packages

Android Studio ▾

Workflow ▾

Tools Help ▾

Build System ▾

Support Library ▾

Revisions ▾

Installing the Android SDK

If you haven't already, [download the Android SDK bundle for Android Studio](#) or the [stand-alone SDK Tools](#).

Then, select which SDK bundle you want to install:

Android Studio

Stand-alone SDK Tools

Installing the Stand-alone SDK Tools

The stand-alone SDK Tools package does not include a complete Android development environment. It includes only the core SDK tools, which you can access from a command line or with a plugin for your favorite IDE (if available).

If you didn't download the SDK tools, go [download the SDK now](#), or switch to the [Android Studio install](#) instructions.

To get started on Mac OSX:

[\[Show instructions for all platforms \]](#)

Unpack the ZIP file you've downloaded. By default, it's unpacked into a directory named `android-sdk-mac_x86`. Move it to an appropriate location on your machine, such as a "Development" directory in your home directory.

Make a note of the name and location of the SDK directory on your system—you will need to refer to the SDK directory later when using the SDK tools from the command line.

The Android SDK tools are now ready to begin developing apps, but there are still a couple packages you should add to make your Android SDK complete.

[CONTINUE: ADDING SDK PACKAGES >](#)

Other Download Options

SDK Tools Only

If you prefer to use a different IDE or run the tools from the command line or with build scripts, you can instead download the stand-alone Android SDK Tools. These packages provide the basic SDK tools for app development, without an IDE. Also see the [SDK tools release notes](#).


Platform	Package	Size	SHA-1 Checksum
Windows	installer_r24.0.2-windows.exe (Recommended)	91428280 bytes	edac14e1541e97d68821fa3a709b4ea8c659e676
	android-sdk_r24.0.2-windows.zip	139473113 bytes	51269c8336f936fc9b9538f9b9ca236b78fb4e4b
Mac OS X	android-sdk_r24.0.2-macosx.zip	87262823 bytes	3ab5e0ab0db5e7c45de9da7ff525dee6cfa97455
Linux	android-sdk_r24.0.2-linux.tgz	140097024 bytes	b6fd75e8b06b0028c2427e6da7d8a09d8f956a86

SDK Tool安裝後記得資料夾，
之後設定需要

下載Eclipse


<http://www.eclipse.org/downloads/index.php>

Eclipse Luna (4.4.1) Release for Windows



Eclipse IDE for Java Developers, 154 MB
Downloaded 3,555,547 Times

The essential tools for any Java developer, including a Java IDE, a CVS client, Git client, XML Editor, Mylyn, Maven integration...




Windows 32 Bit
Windows 64 Bit

Windows


Eclipse Luna (4.4.1) Release for Mac OS X (Cocoa)

Mac OS X users please note: Eclipse requires Mac OS X 10.5 (Leopard) or greater.



Eclipse IDE for Java Developers, 153 MB
Downloaded 3,555,547 Times

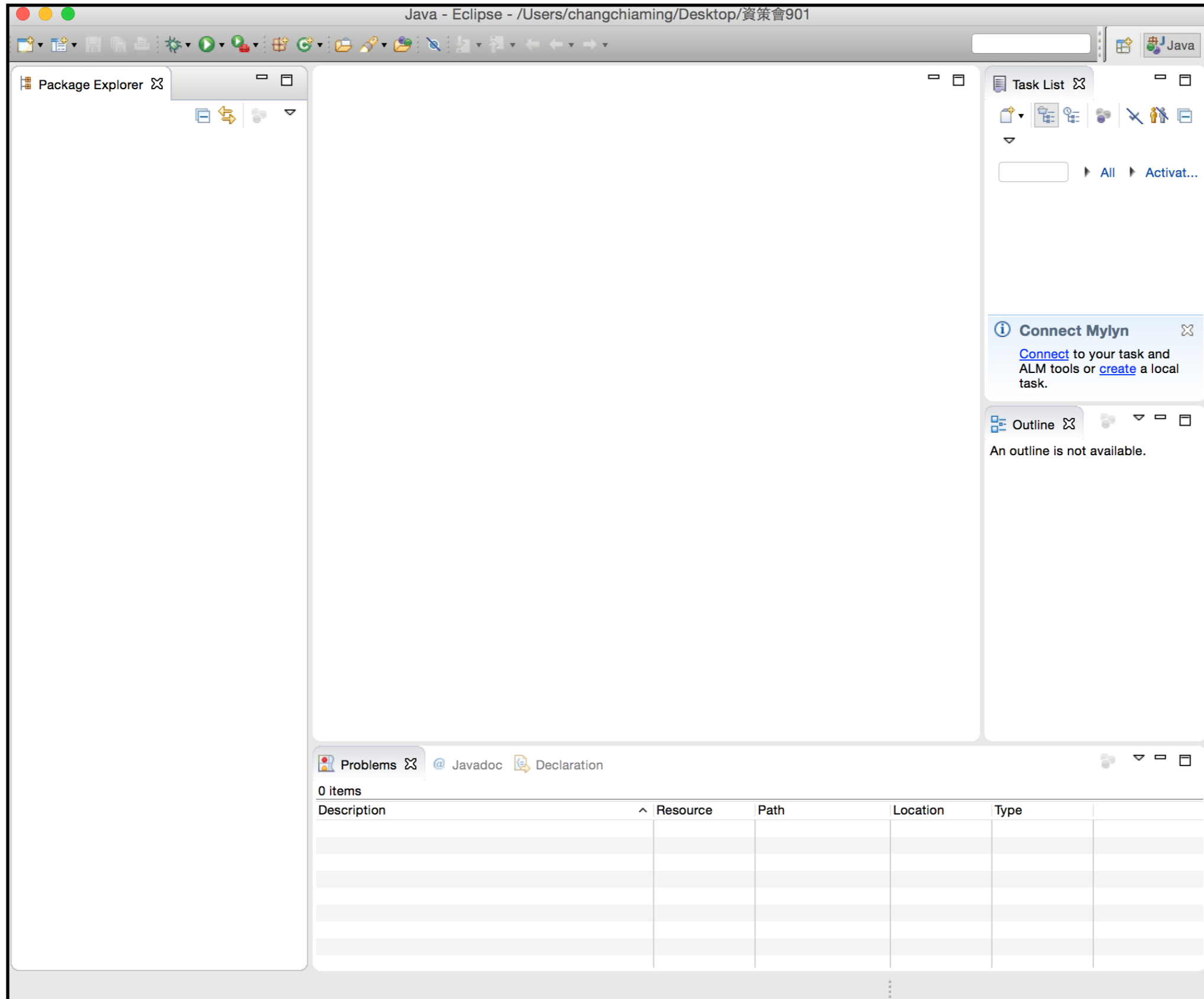
The essential tools for any Java developer, including a Java IDE, a CVS client, Git client, XML Editor, Mylyn, Maven integration...



Mac OS X 32 Bit
Mac OS X 64 Bit

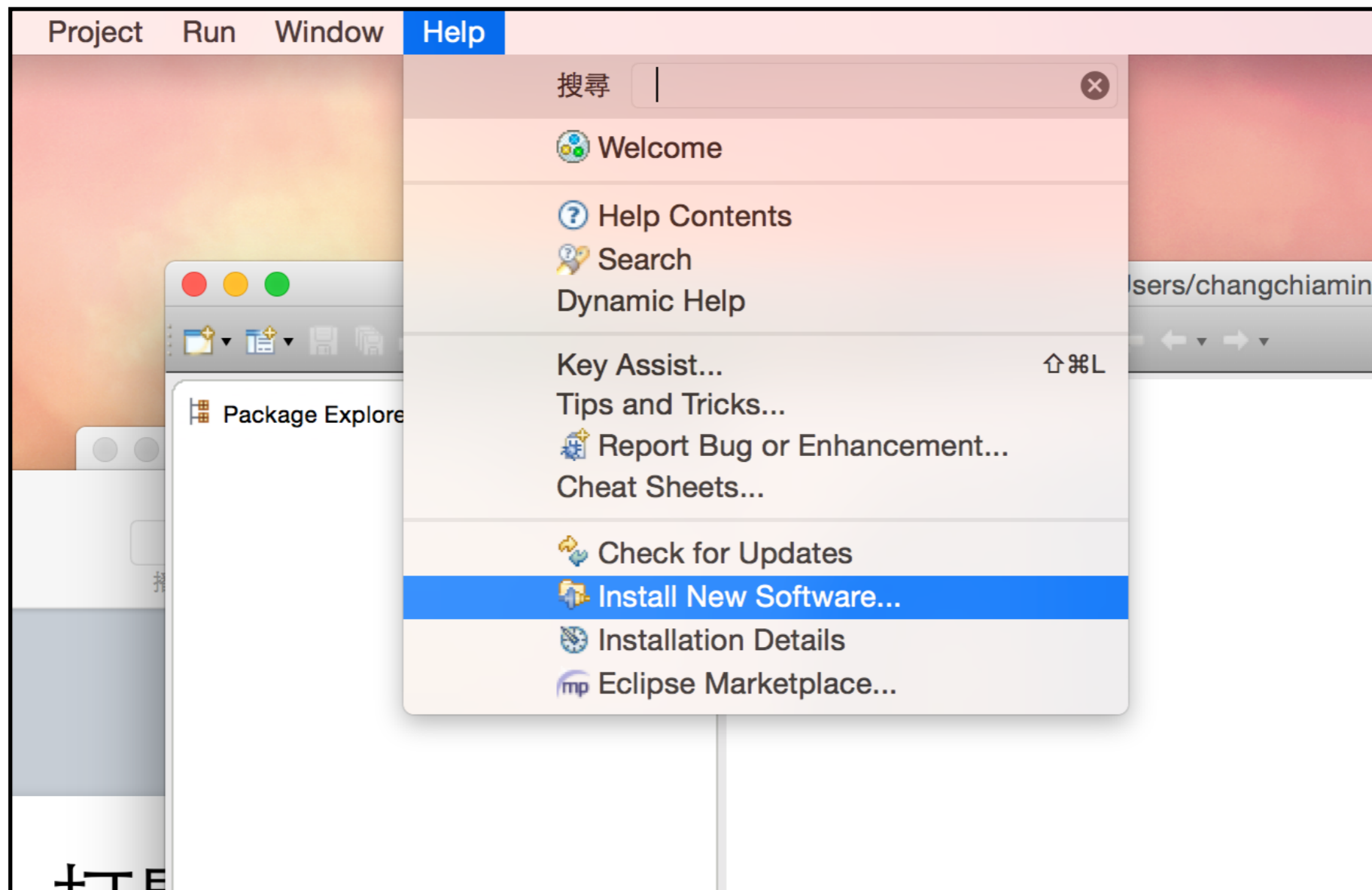
Mac

打開Eclipse，設定workspace

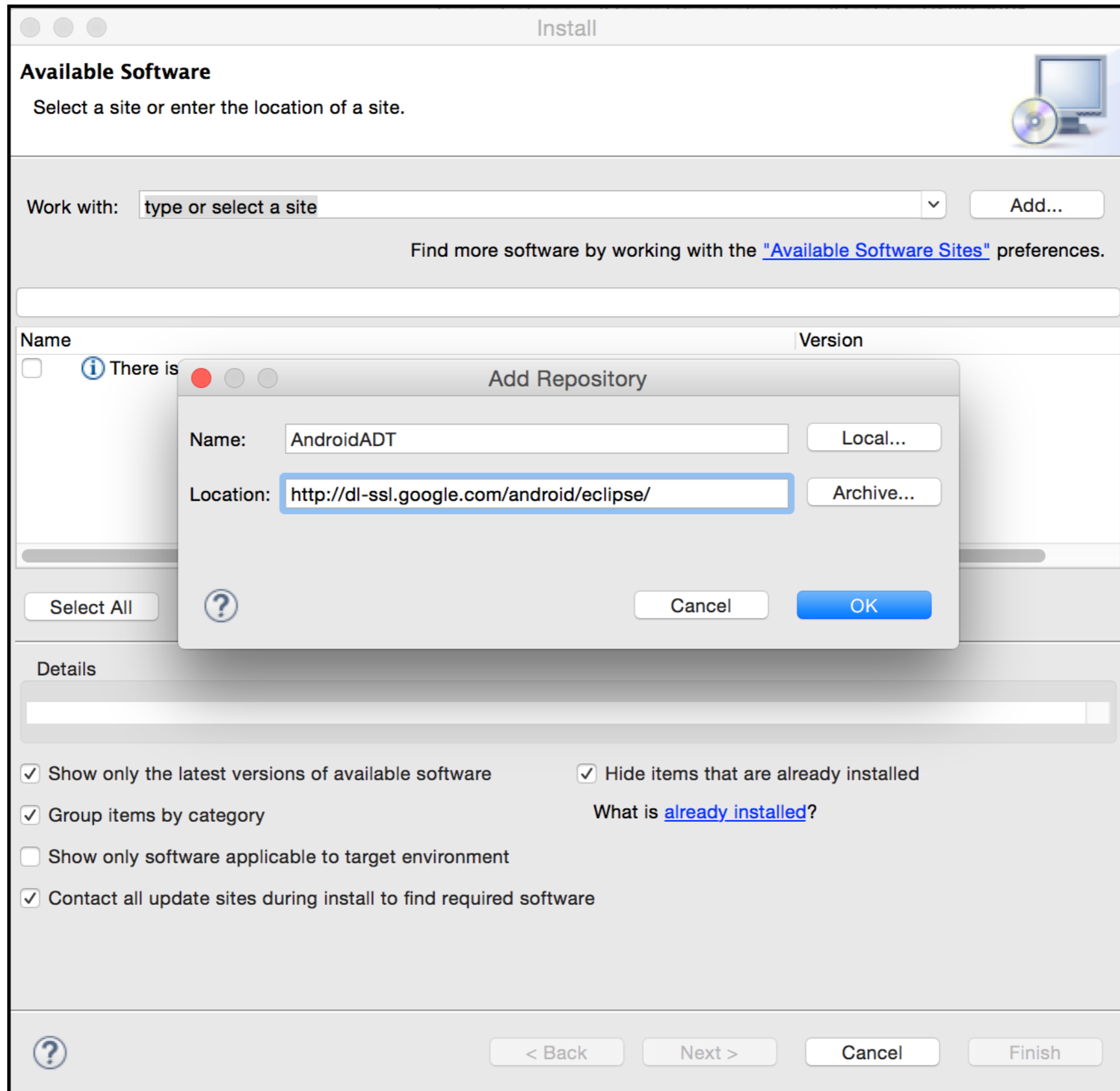


打開Eclipse，設定workspace，安裝ADT

- Help > Install New Software




http://dl-ssl.google.com/android/eclipse/



Work with: AndroidADT - <http://dl-ssl.google.com/android>

Find more soft

Name

 Developer Tools

Licenses:

- ▶ Apache License
- ▶ Eclipse Foundation Software User Agreement
- ▶ Eclipse Public License - v 1.0
- ▶ Note: jcommon-1.0.12.jar is under the BSD license rather than th
- ▶ Note: kxml2-2.3.0.jar is under the BSD license rather than the EF

License text:

Apache License

Version 2.0, January 2004
<http://www.apache.org/licenses/>

**TERMS AND CONDITIONS FOR USE,
REPRODUCTION, AND DISTRIBUTION**

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

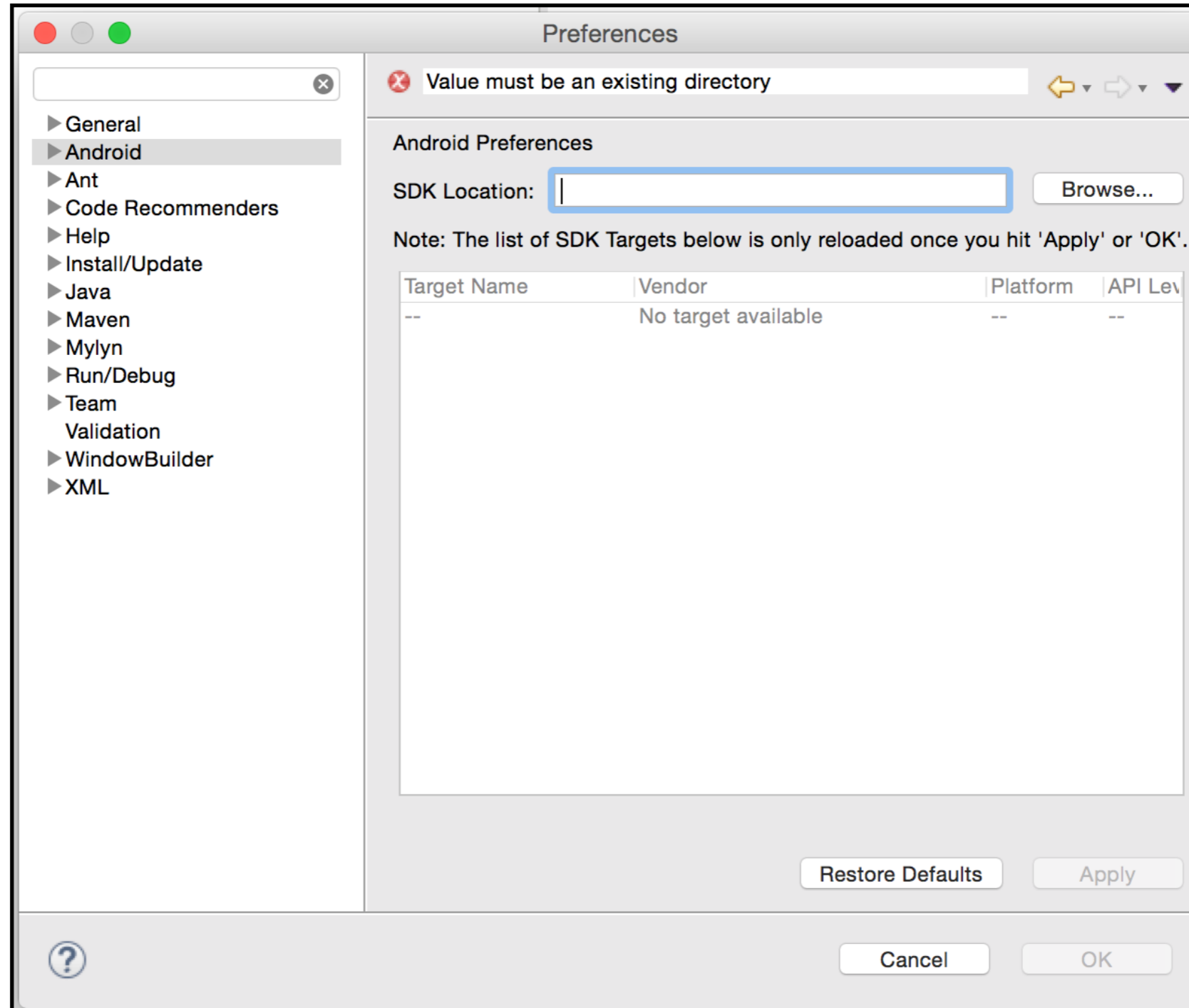
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%)

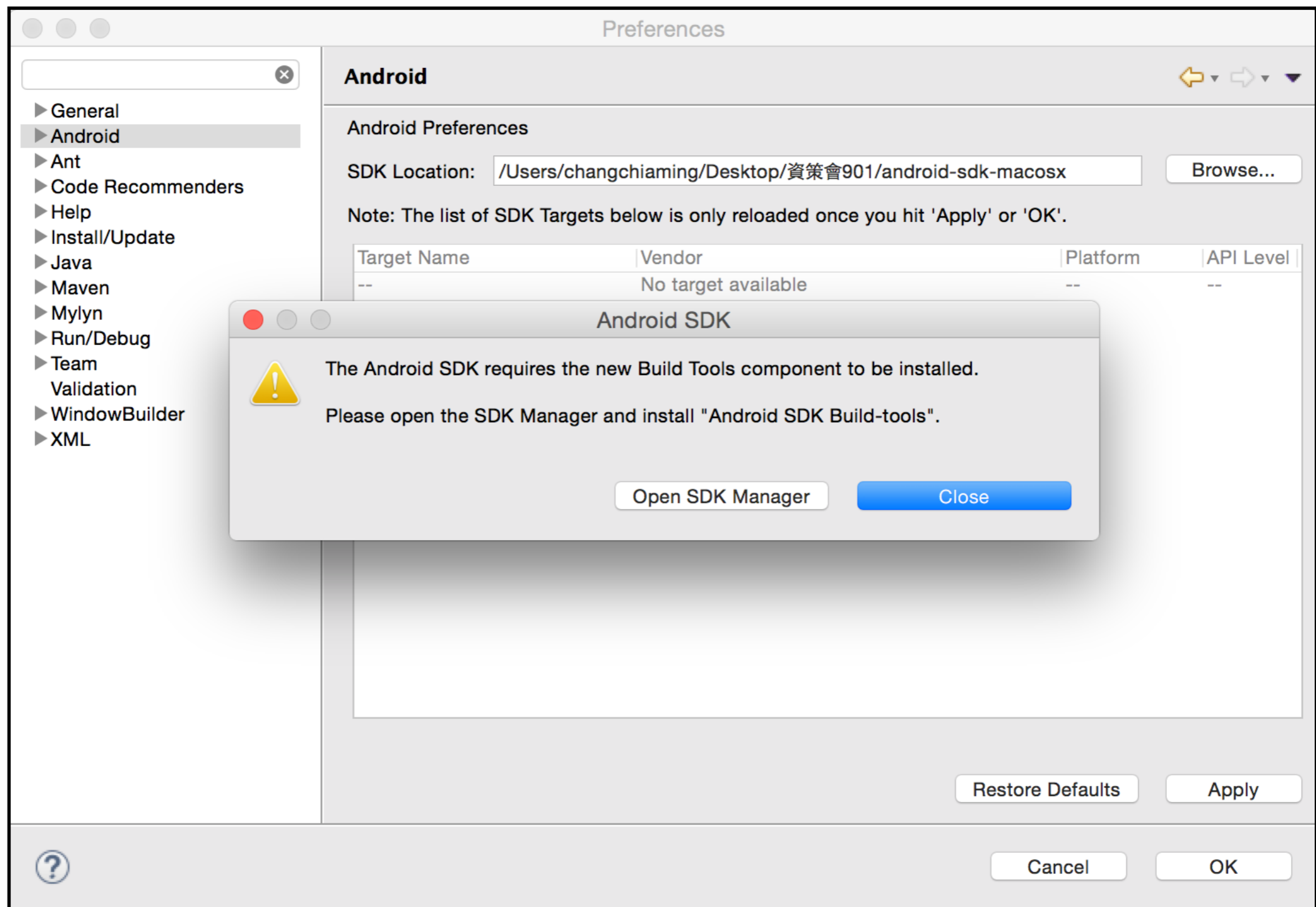
- I accept the terms of the license agreements
- I do not accept the terms of the license agreements

設定Eclipse的Android SDK

- Preference > Android > SDK Location



若有miss的狀況> Open SDK Manager



安裝Android 4.0即可

Android SDK Manager

SDK Path: /Users/changchiaming/Desktop/資策會901/android-sdk-macosx/

Packages

Name	API	Rev.	Status
<input type="checkbox"/> Google APIs Intel x86 Atom System Image	21	3	<input type="checkbox"/> Not installed
<input type="checkbox"/> Google APIs Intel x86 Atom System Image	21	3	<input type="checkbox"/> Not installed
<input type="checkbox"/> Google APIs	21	1	<input type="checkbox"/> Not installed
<input type="checkbox"/> Sources for Android SDK	21	1	<input type="checkbox"/> Not installed
<input type="checkbox"/> ▶ Android 4.4W.2 (API 20)			
<input type="checkbox"/> ▶ Android 4.4.2 (API 19)			
<input type="checkbox"/> ▶ Android 4.3.1 (API 18)			
<input type="checkbox"/> ▶ Android 4.2.2 (API 17)			
<input type="checkbox"/> ▶ Android 4.1.2 (API 16)			
<input type="checkbox"/> ▶ Android 4.0.3 (API 15)			
<input type="checkbox"/> ▼ Android 4.0 (API 14)			
<input checked="" type="checkbox"/> SDK Platform	14	4	<input type="checkbox"/> Not installed
<input type="checkbox"/> Samples for SDK	14	2	<input type="checkbox"/> Not installed
<input checked="" type="checkbox"/> ARM EABI v7a System Image	14	2	<input type="checkbox"/> Not installed
<input checked="" type="checkbox"/> Google APIs	14	2	<input type="checkbox"/> Not installed
<input checked="" type="checkbox"/> Sources for Android SDK	14	1	<input type="checkbox"/> Not installed
<input type="checkbox"/> ▶ Android 3.2 (API 13)			

Show: Updates/New Installed Select [New](#) or [Updates](#)

Obsolete [Deselect All](#)

Install 8 packages...

Delete 1 package...

Done loading packages.

Android ADT Bundle快速安裝

The screenshot shows the Android SDK website in a browser window. The page title is "Android SDK | Android Developers". The URL is "developer.android.com/intl/zh-tw/sdk/index.html". The page has a navigation bar with "Developers" and tabs for "設計", "開發", and "發佈". Below the navigation bar are links for "訓練課程", "API 指南", "參考資源", "相關工具", "Google Services", and "Samples".

The main content area is titled "Get the Android SDK". It contains the following text:

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

If you're a new Android developer, we recommend you download the ADT Bundle to quickly start developing apps. It includes the essential Android SDK components and a version of the Eclipse IDE with built-in **ADT (Android Developer Tools)** to streamline your Android app development.

With a single download, the ADT Bundle includes everything you need to begin developing apps:

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform
- The latest Android system image for the emulator

Android Studio Early Access Preview

A new Android development environment called Android Studio, based on IntelliJ IDEA, is now available as an **early access preview**. For more information, see [Getting Started with Android Studio](#).

If you prefer to use an existing version of Eclipse or another IDE, you can instead take a more customized approach to installing the Android SDK. See the following instructions:

- [USE AN EXISTING IDE](#)
- [SYSTEM REQUIREMENTS](#)

On the right side of the page, there is a 3D illustration of the Android robot surrounded by blue blocks. Below it is a blue button with the text "Download the SDK" and "ADT Bundle for Mac". This button is highlighted with a red dashed circle.

Get the Android SDK

Before installing the Android SDK, you must agree to the following terms and conditions.

Terms and Conditions

This is the Android Software Development Kit License Agreement

1. Introduction

1.1 The Android Software Development Kit (referred to in this License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of this License Agreement. This License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.

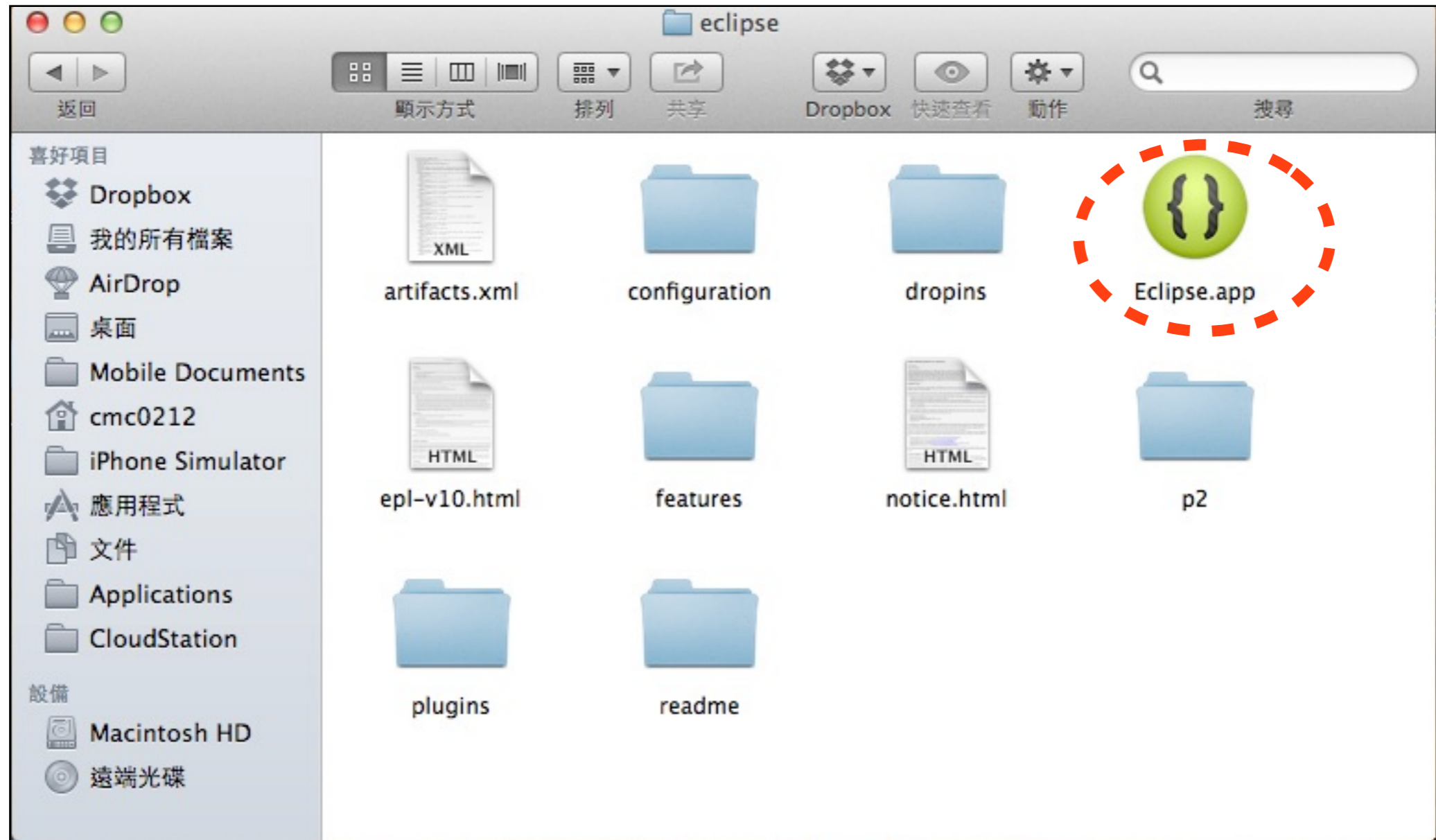
1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: <http://source.android.com/>, as updated from time to time.

1.3 "Google" means Google Inc., a Delaware corporation with principal place of business at 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States.

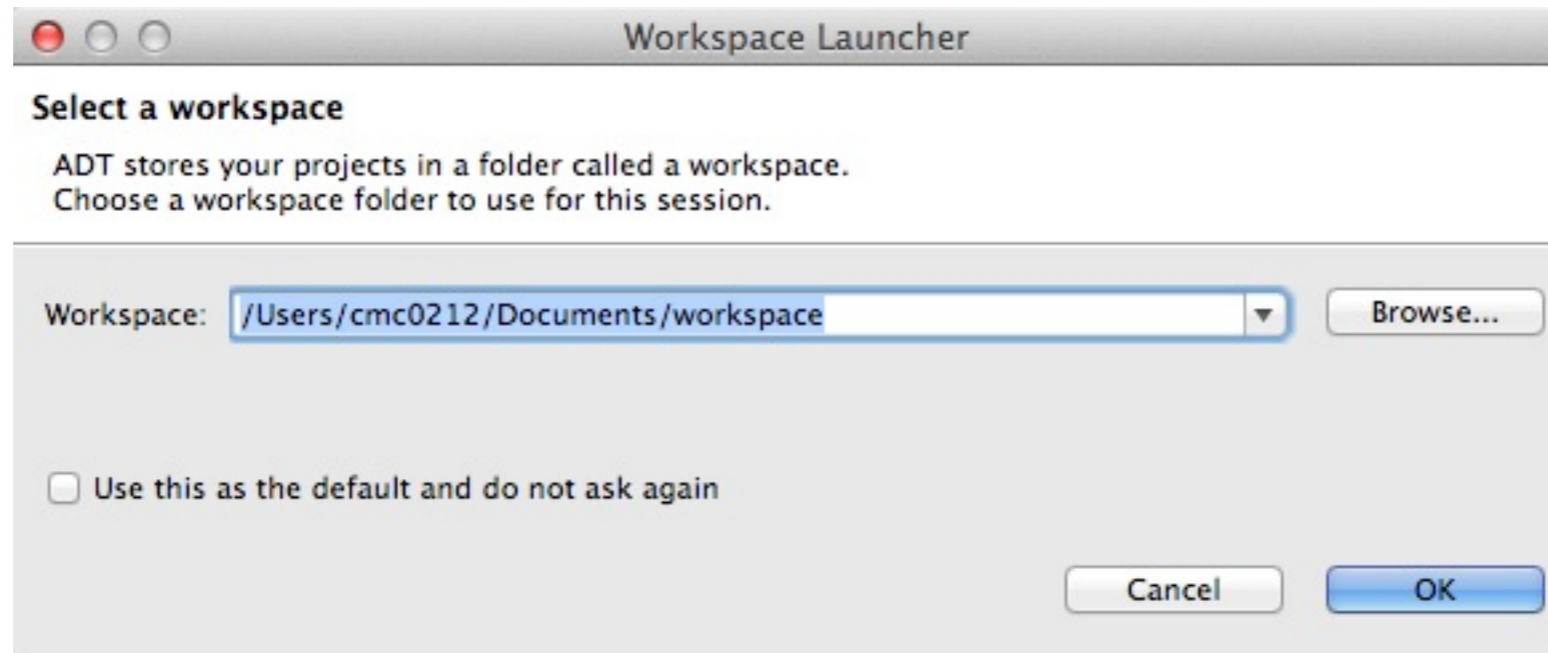
I have read and agree with the above terms and conditions

[Download the SDK ADT Bundle for Mac](#)

執行Eclipse



設定工作區



Eclipse環境

程式編輯區

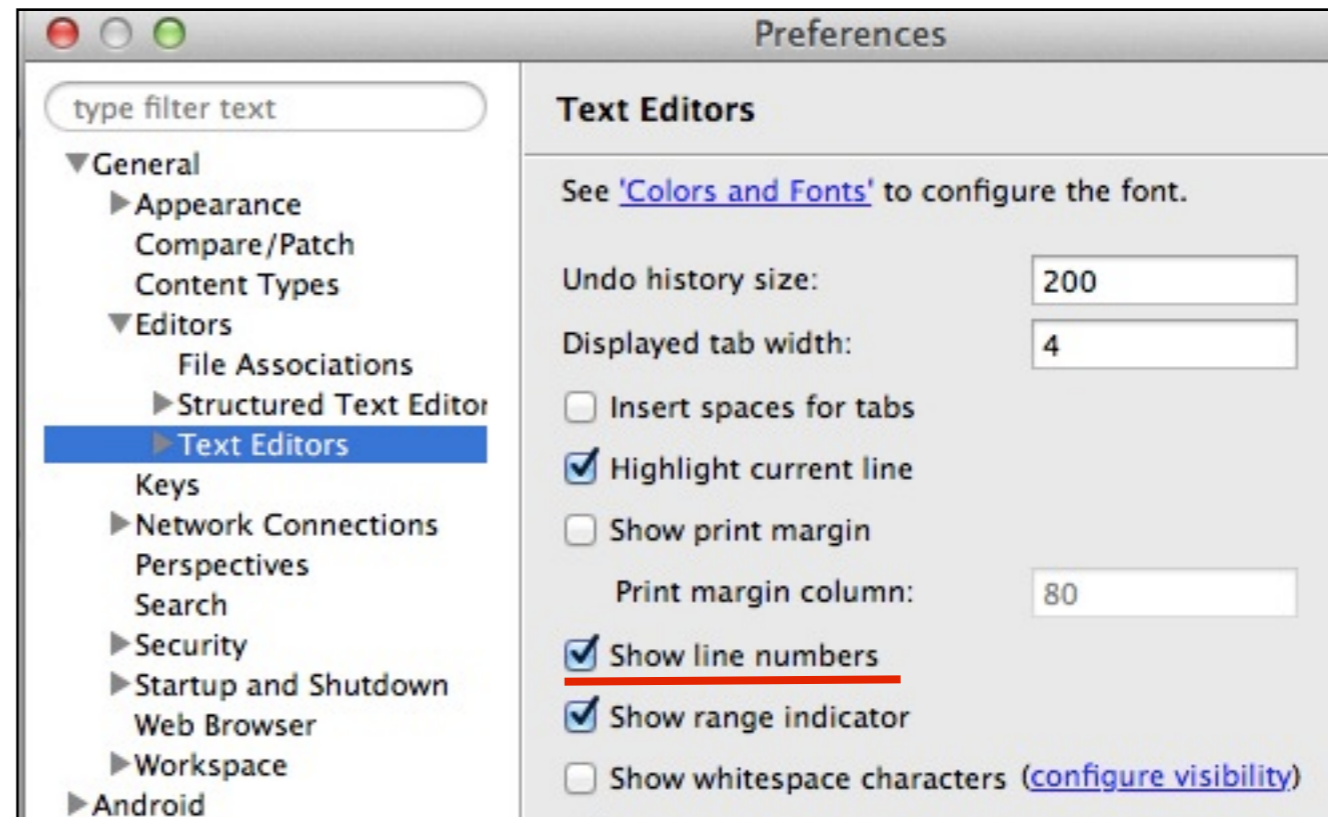
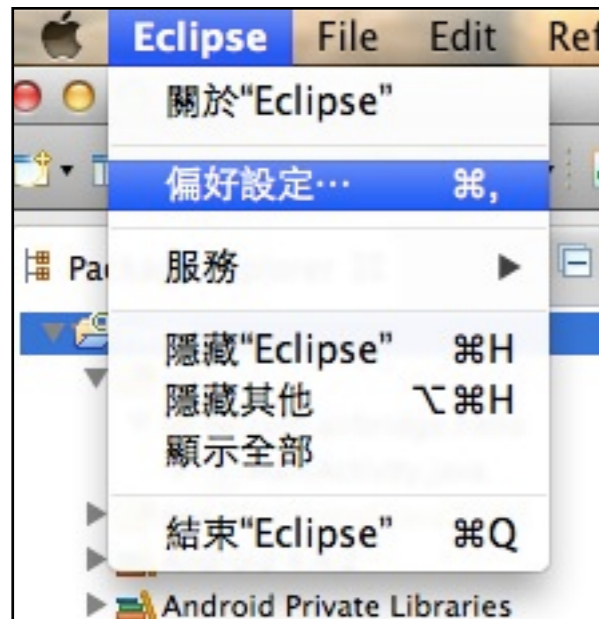
檔案瀏覽

程式架構輔助視窗

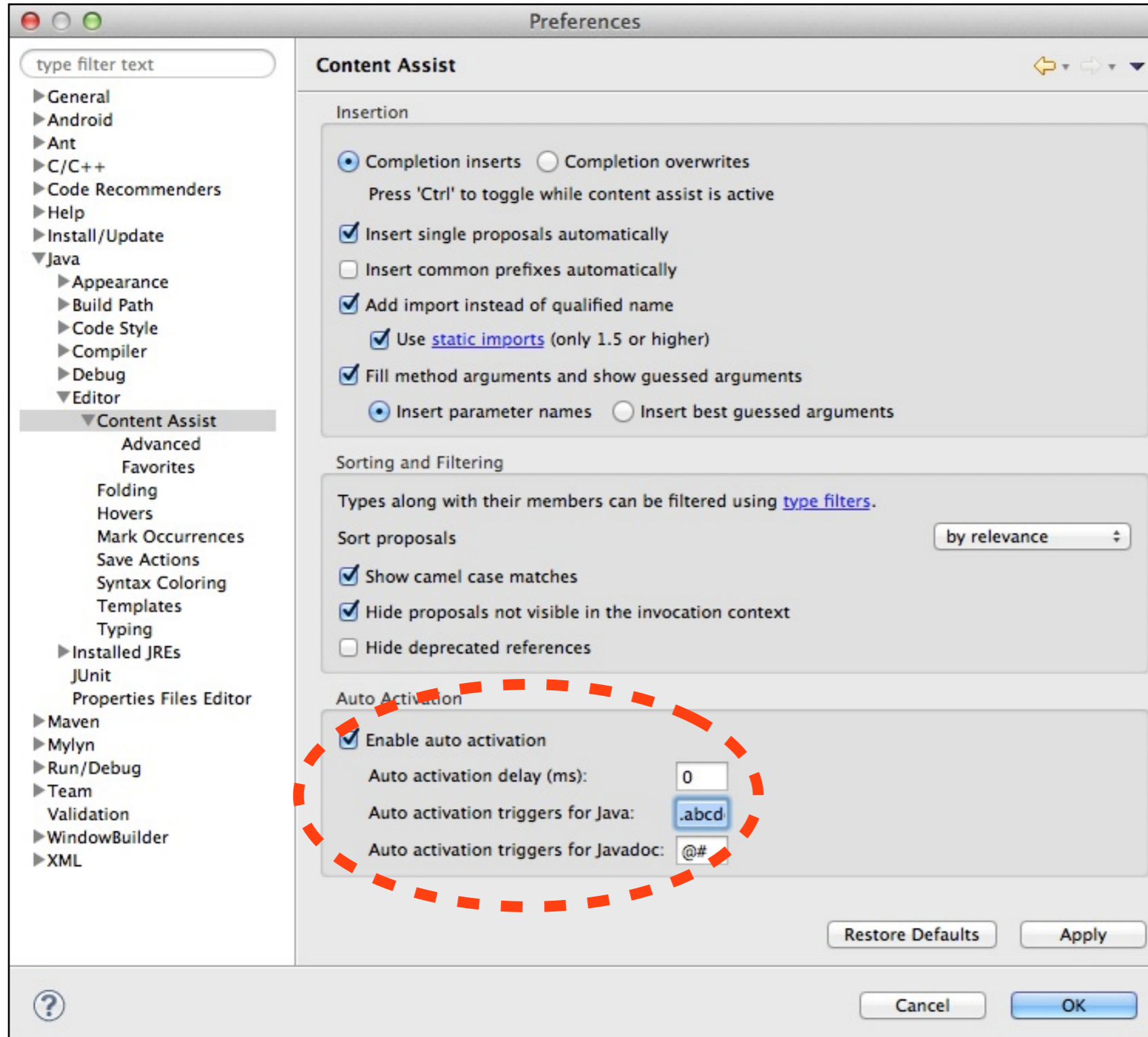
其他輔助視窗



Tip: 顯示行號

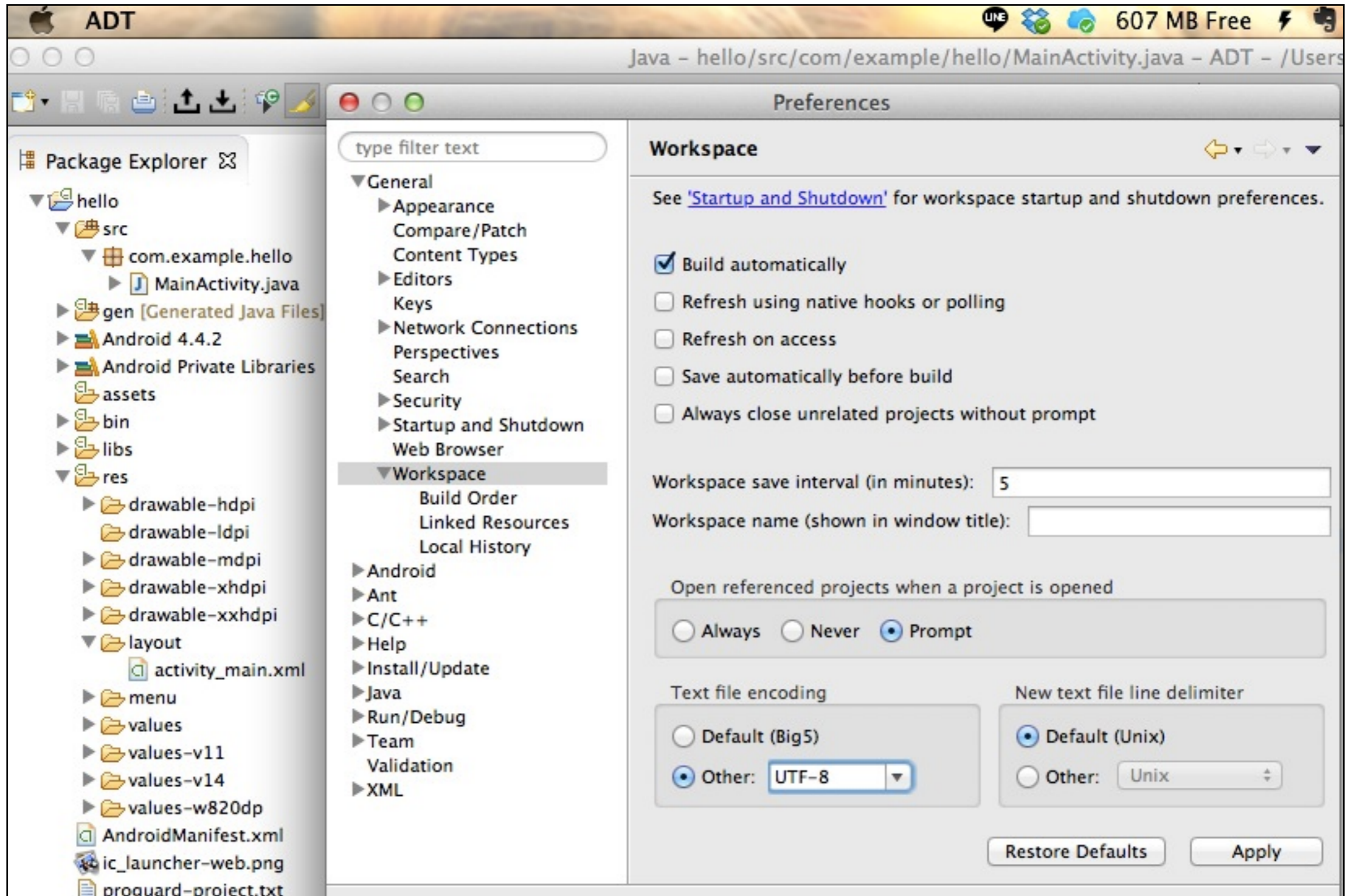


自動完成



or Alt+/

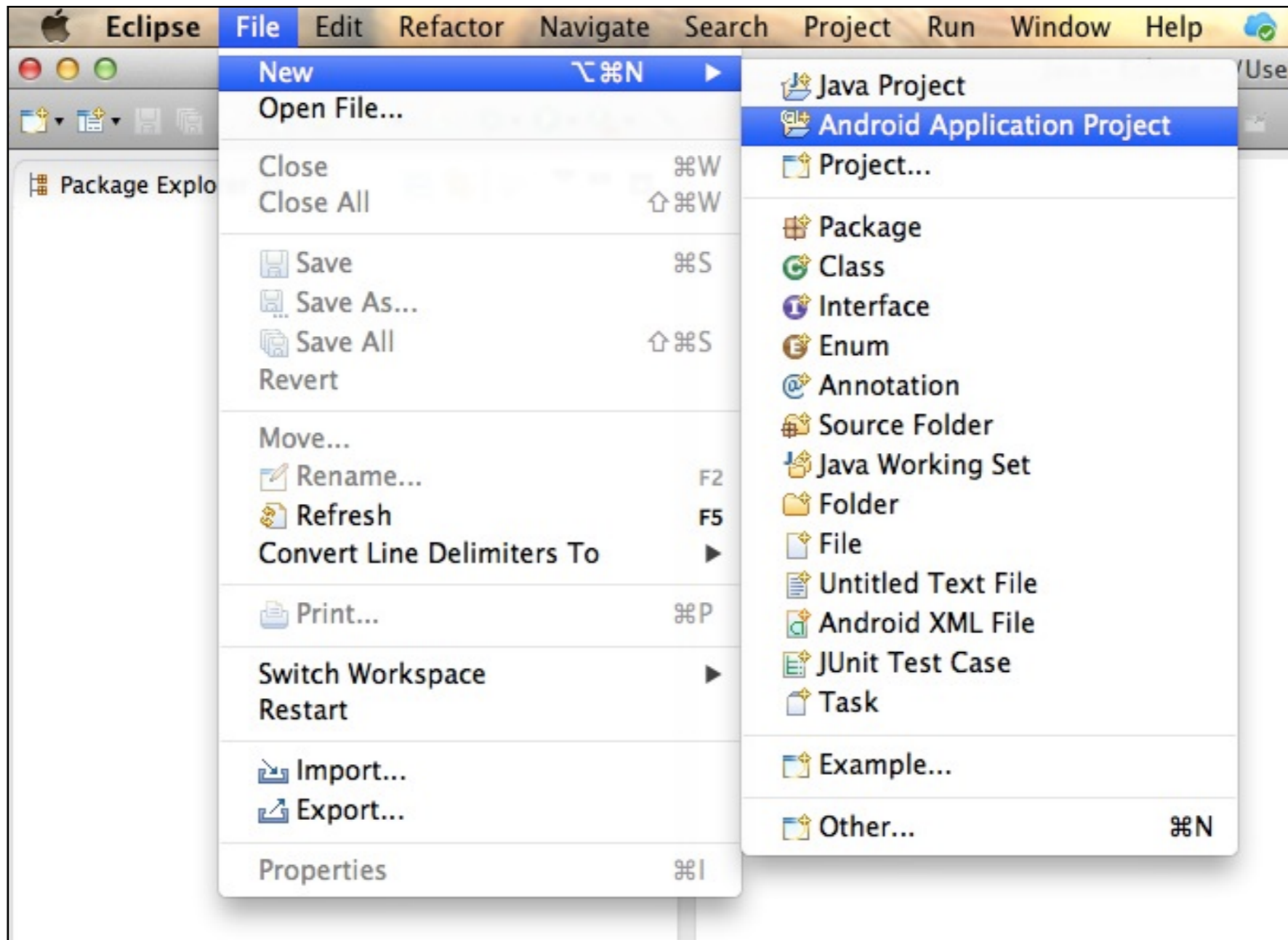
設定編碼為UTF-8



第一個App & 目錄說明

建立第一個專案

- File > New > Android Application Project



若是找不到 New > Android Project

Window > Customize Perspective
> Menu Visibility > File > New
> Android Project

New Android Application

Enter an application name

Application Name:

Project Name:

Package Name:

Minimum Required SDK:

Target SDK:

Compile With:

Theme:

The application name is shown in the Play Store, as well as in the Manage Application list in Settings.

< Back Next > Cancel Finish

應用程式名稱
(iOS: Bundle Display)

專案名稱

軟體名稱
(iOS:Bundle identifier)

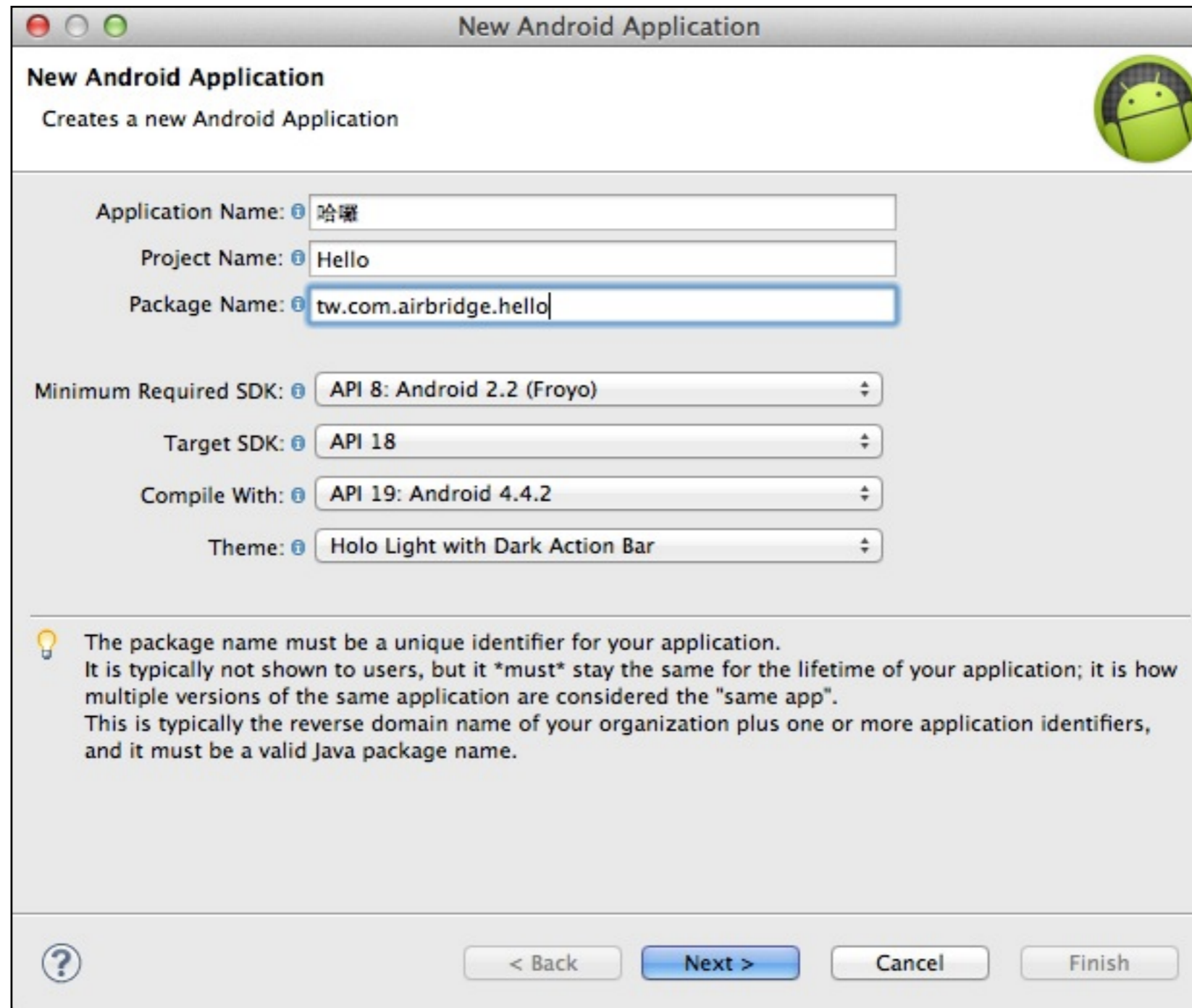
最低需求版本

目標版本

編譯版本

版面

填入專案名稱



New Android Application

Creates a new Android Application

Application Name: 哈囉

Project Name: Hello


Package Name: tw.com.airbridge.hello


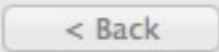
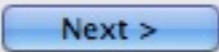
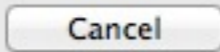
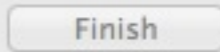
Minimum Required SDK: API 8: Android 2.2 (Froyo)

Target SDK: API 18

Compile With: API 19: Android 4.4.2

Theme: Holo Light with Dark Action Bar

 The package name must be a unique identifier for your application. It is typically not shown to users, but it *must* stay the same for the lifetime of your application; it is how multiple versions of the same application are considered the "same app". This is typically the reverse domain name of your organization plus one or more application identifiers, and it must be a valid Java package name.

專案建立結果

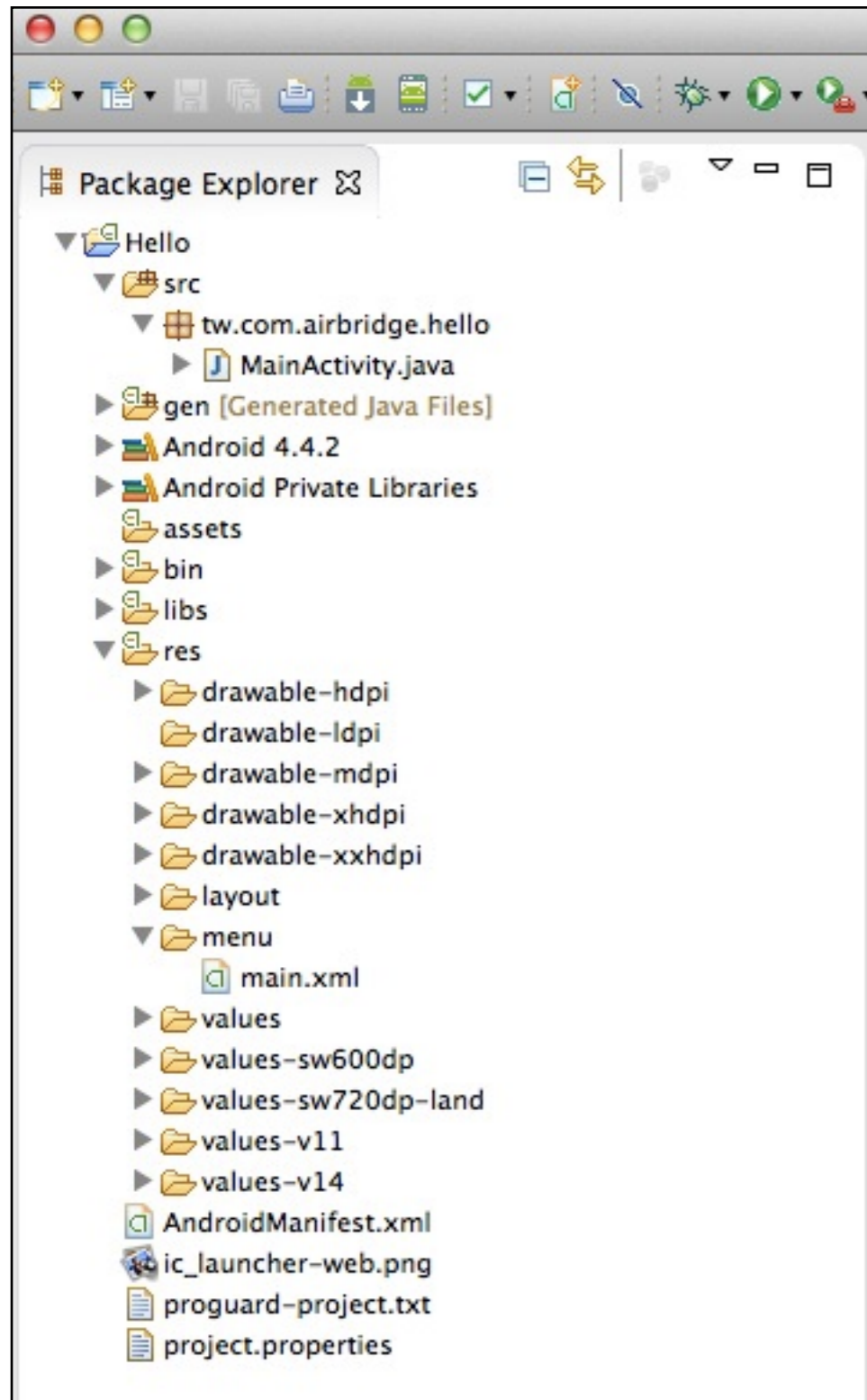


專案檔案目錄

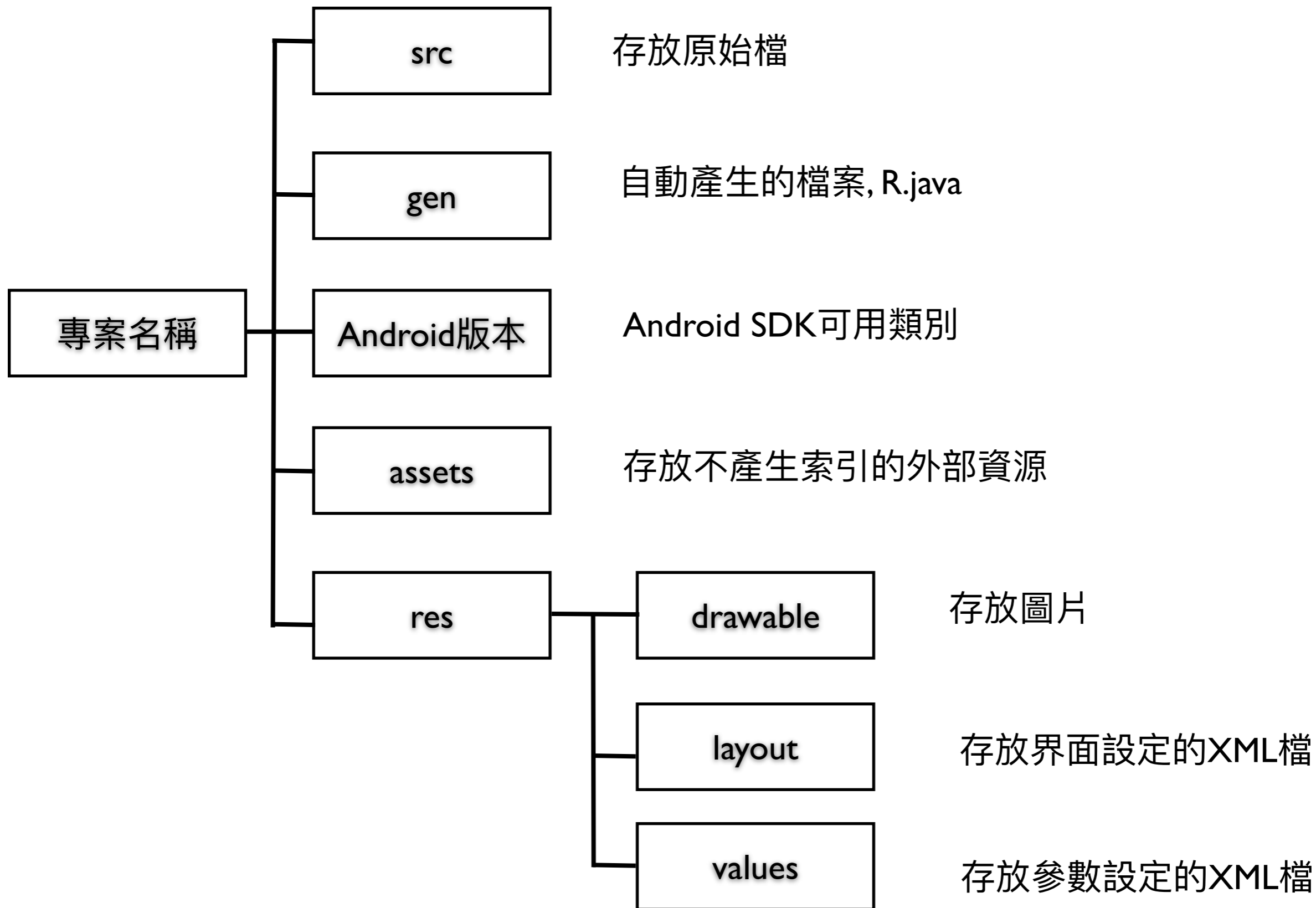
程式編輯區

其他輔助視窗

檔案結構說明



- src: 存放原始碼
- gen: 自動產生的檔案
- res: 相關資源檔案
 - drawable: 影像圖型檔案
 - layout: 版面配置檔案
 - values: 使用資料值



其他產生檔案

- AndroidManifest.xml檔
 - 紀錄應用程式的套件名稱、版本資訊、權限設定等屬性
- proguard.cfg檔
 - 防止Android應用程式被反組譯
- project.properties檔
 - 紀錄Android SDK版本資訊
- ic_launcher-web.png
 - 內定的icon檔

程式說明

```
activity_main.xml MainActivity.java
1 package tw.com.airbridge.hello;
2
3 import android.os.Bundle;
4
5 public class MainActivity extends Activity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11    }
12
13    @Override
14    public boolean onCreateOptionsMenu(Menu menu) {
15        // Inflate the menu; this adds items to the action bar if it is present.
16        getMenuInflater().inflate(R.menu.main, menu);
17        return true;
18    }
19
20 }
21
22
23
```

android程式進入點 onCreate(..)

- 1 `package tw.com.airbridge.hello;` //目前的class是在哪個package裡面, package有點像是資料庫
- 2 `import android.os.Bundle;` //將需要的class import進來

3 public class MainActivity extends Activity

補充說明: Java宣告一個class的語法

```
存取權限 class 類別名稱 extends 父類別  
{  
    變數;  
    方法;  
}
```

4 Activity: 手機的基本畫面

5

@Override: 在android中，用來重寫父類別函式時使用

補充說明: @Override 重寫

- 重寫為物件導向概念中多型的表現，用來客制化我們自己的需求

```
6 protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
}
```

補充說明:Java的方法宣告語法

存取權限 回傳值 方法名稱 (參數)

{

..

.

}

6

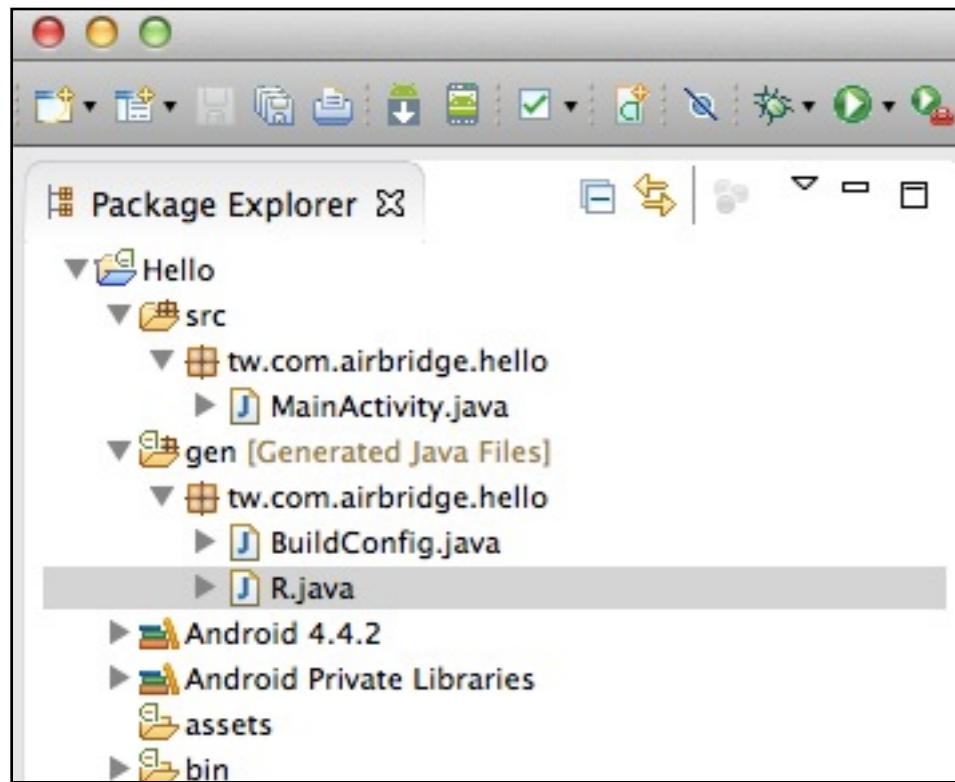
```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
}
```

android程式進入點

初始化

指定顯示的layout為哪個檔案

Bundle類別： Intent之間，需傳遞的資料或參數可以藉由Bundle來傳遞

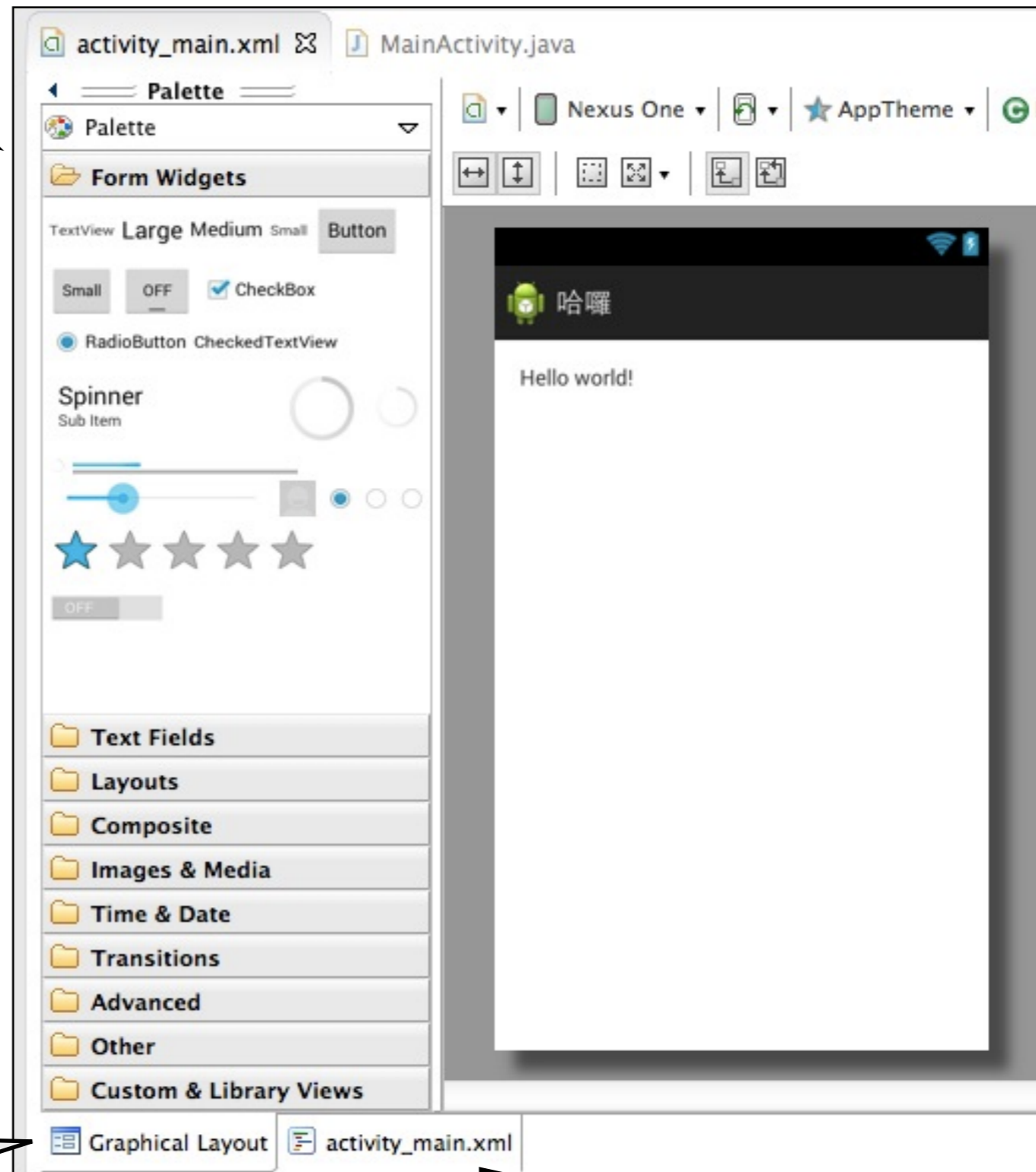
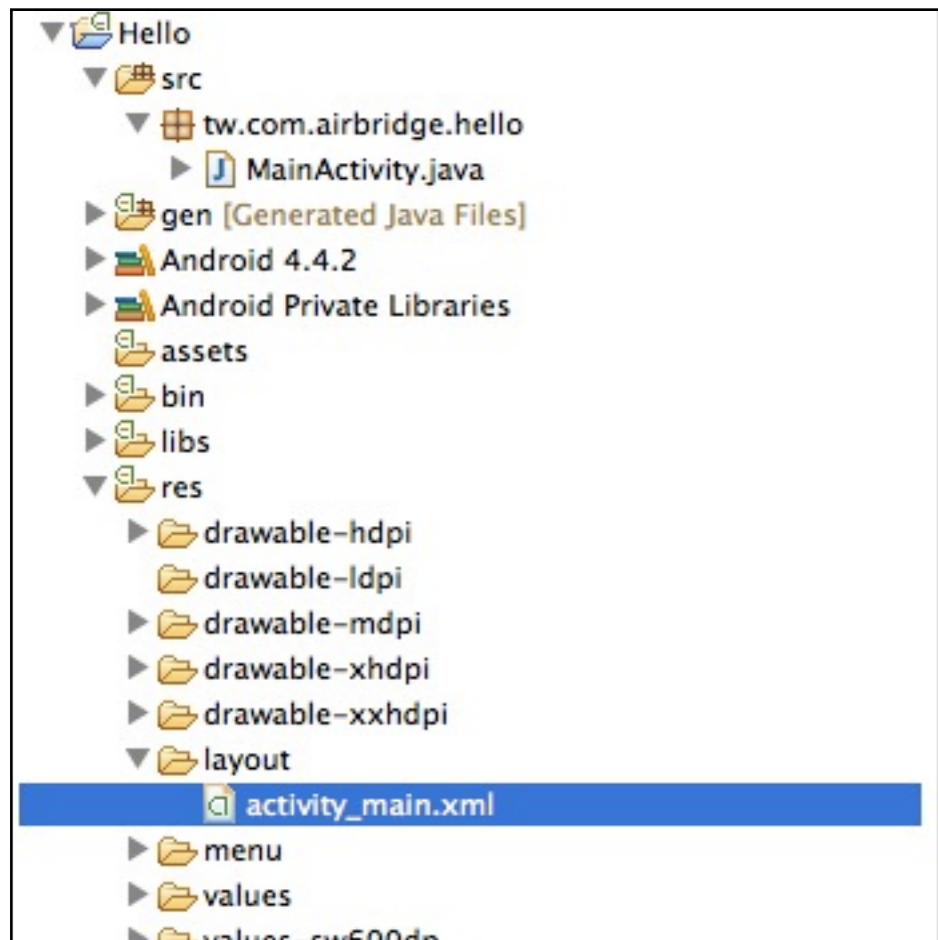


```
1+ /* AUTO-GENERATED FILE. DO NOT MODIFY.
7
8 package tw.com.airbridge.hello;
9
10 public final class R {
11     public static final class attr {
12     }
13     public static final class dimen {
14         /** Default screen margins, per the Android Design guidelines.
15
16         Customize dimensions originally defined in res/values/dimens.xml (such as
17         screen margins) for sw720dp devices (e.g. 10" tablets) in landscape here.
18
19         */
20         public static final int activity_horizontal_margin=0x7f040000;
21         public static final int activity_vertical_margin=0x7f040001;
22     }
23     public static final class drawable {
24         public static final int ic_launcher=0x7f020000;
25     }
26     public static final class id {
27         public static final int action_settings=0x7f080000;
28     }
29     public static final class layout {
30         public static final int activity_main=0x7f030000;
31     }
32     public static final class menu {
33         public static final int main=0x7f070000;
34     }
35     public static final class string {
36         public static final int action_settings=0x7f050001;
37         public static final int app_name=0x7f050000;
38         public static final int hello_world=0x7f050002;
39     }
40     public static final class style {
41         /**
```

- R類別: 由系統自動產生的，用於統和所有的資源，包括圖檔、layout、字串...

觀看Layout檔 activity_main.xml

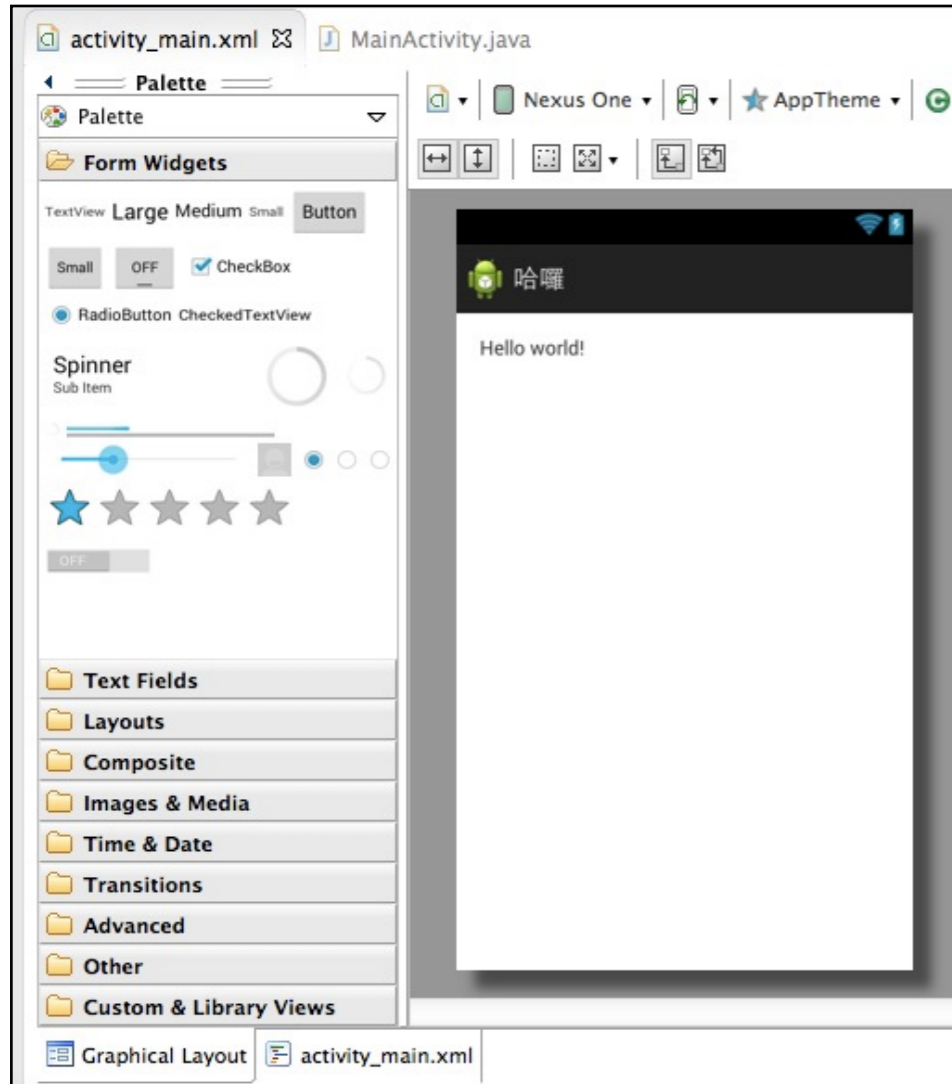
可以加入的元件



圖像模式

文字模式

Layout檔 activity_main.xml的文字模式

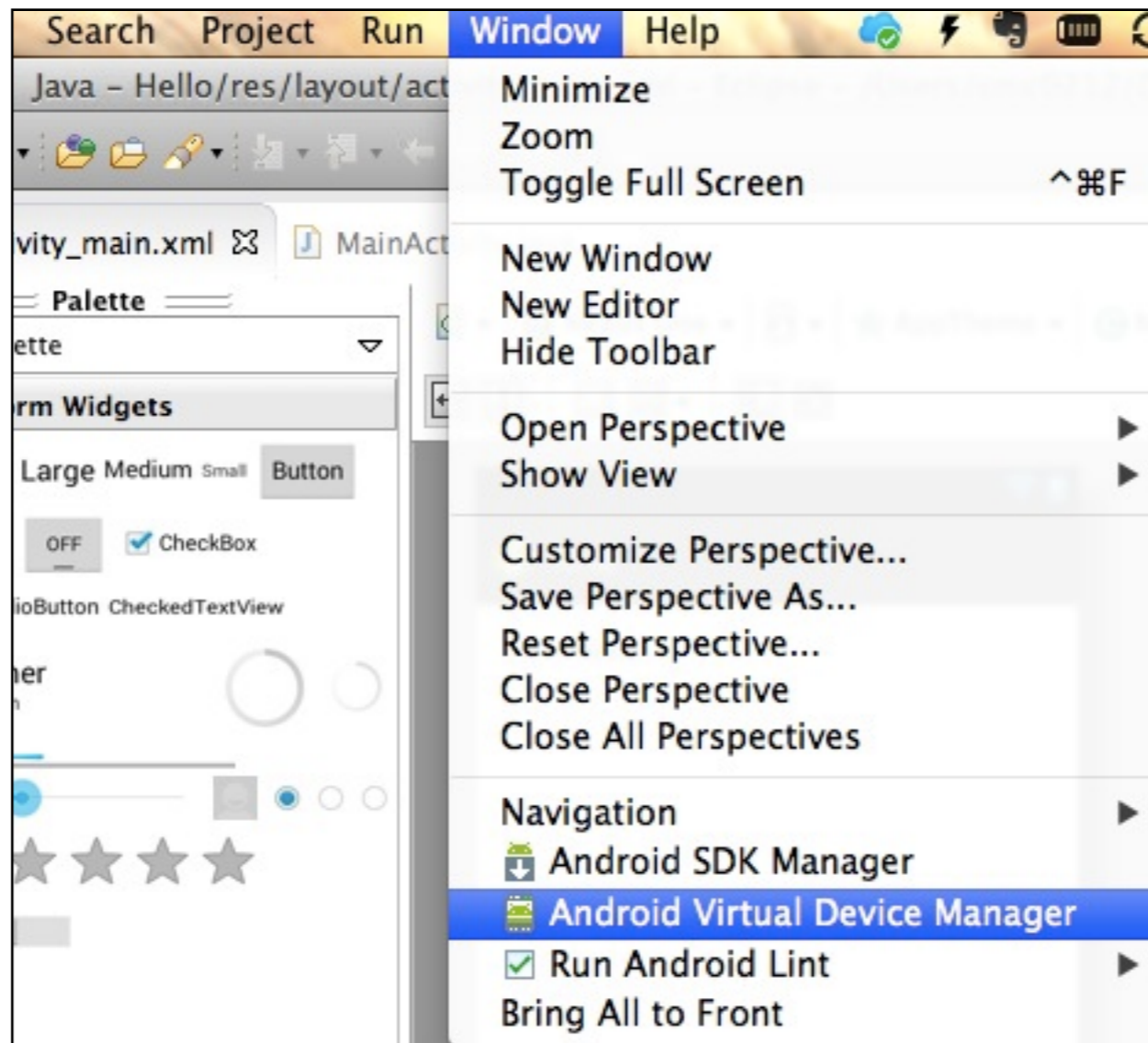


```
activity_main.xml MainActivity.java
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:paddingBottom="@dimen/activity_vertical_margin"
6   android:paddingLeft="@dimen/activity_horizontal_margin"
7   android:paddingRight="@dimen/activity_horizontal_margin"
8   android:paddingTop="@dimen/activity_vertical_margin"
9   tools:context=".MainActivity" >
10
11   <TextView
12     android:layout_width="wrap_content"
13     android:layout_height="wrap_content"
14     android:text="@string/hello_world" />
15
16 </RelativeLayout>
17
```

This screenshot shows the text editor view of the activity_main.xml file in Android Studio. The code defines a RelativeLayout with a single TextView child. The RelativeLayout has a width and height of 'match_parent' and padding defined by dimensions. The TextView has a width and height of 'wrap_content' and its text is set to '@string/hello_world'. The bottom of the screen shows the 'Graphical Layout' tab and the filename 'activity_main.xml'.

使用Android的模擬器

- Window > Android Virtual Device Manager > New



設定你的Android裝置

使用電腦實體鍵盤

模擬Android按鈕

AVD Name: AVD4.0

Device: 5.1" WVGA (480 × 800: mdpi)

Target: Android 4.0 – API Level 14

CPU/ABI: ARM (armeabi-v7a)

Keyboard: Hardware keyboard present

Skin: Display a skin with hardware controls

Front Camera: None

Back Camera: None

Memory Options: RAM: 2048 VM Heap: 64

Internal Storage: 200 MiB

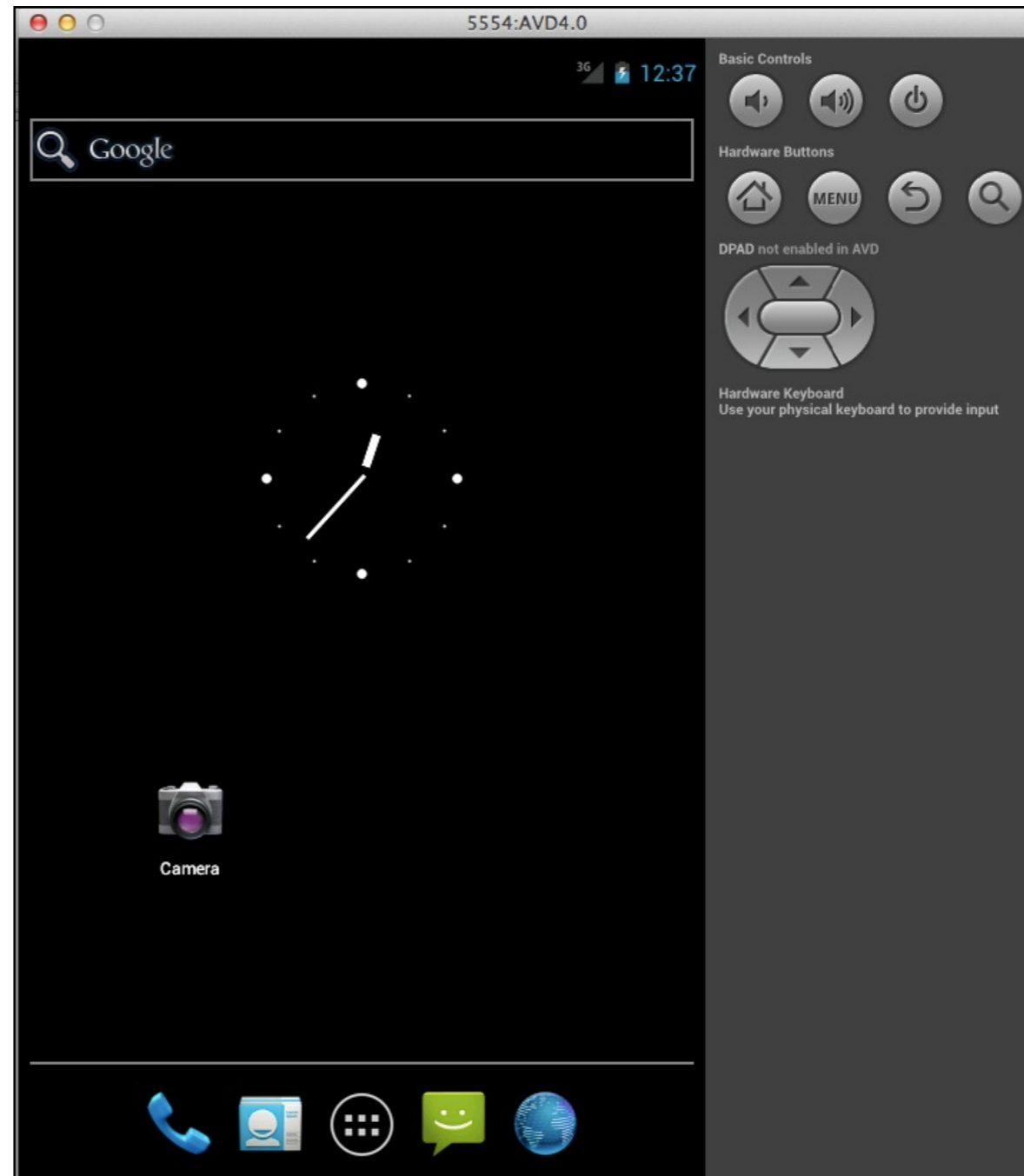
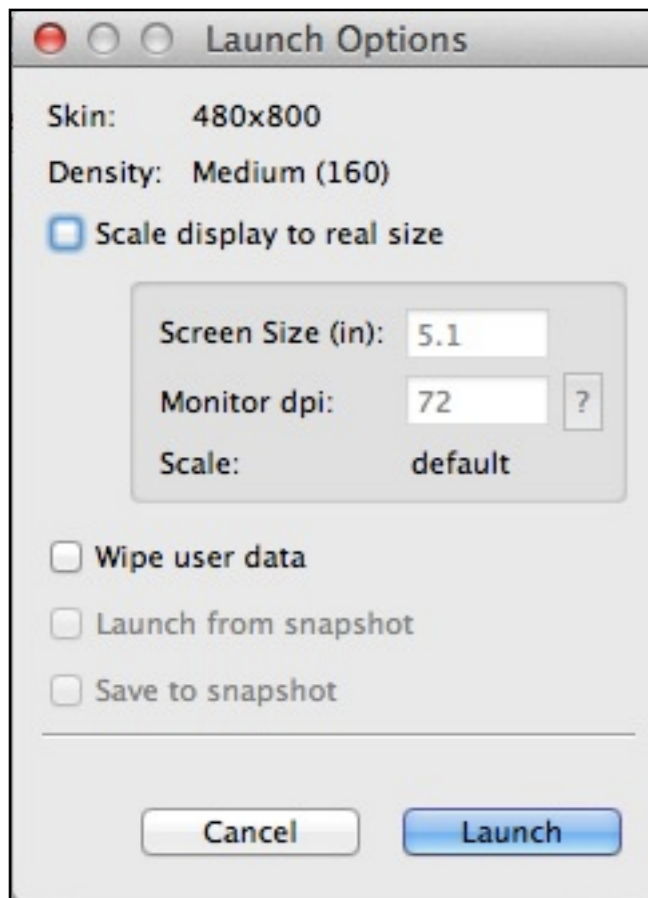
SD Card: Size: 1024 MiB File: Browse...

Emulation Options: Snapshot Use Host GPU

Override the existing AVD with the same name

Cancel OK

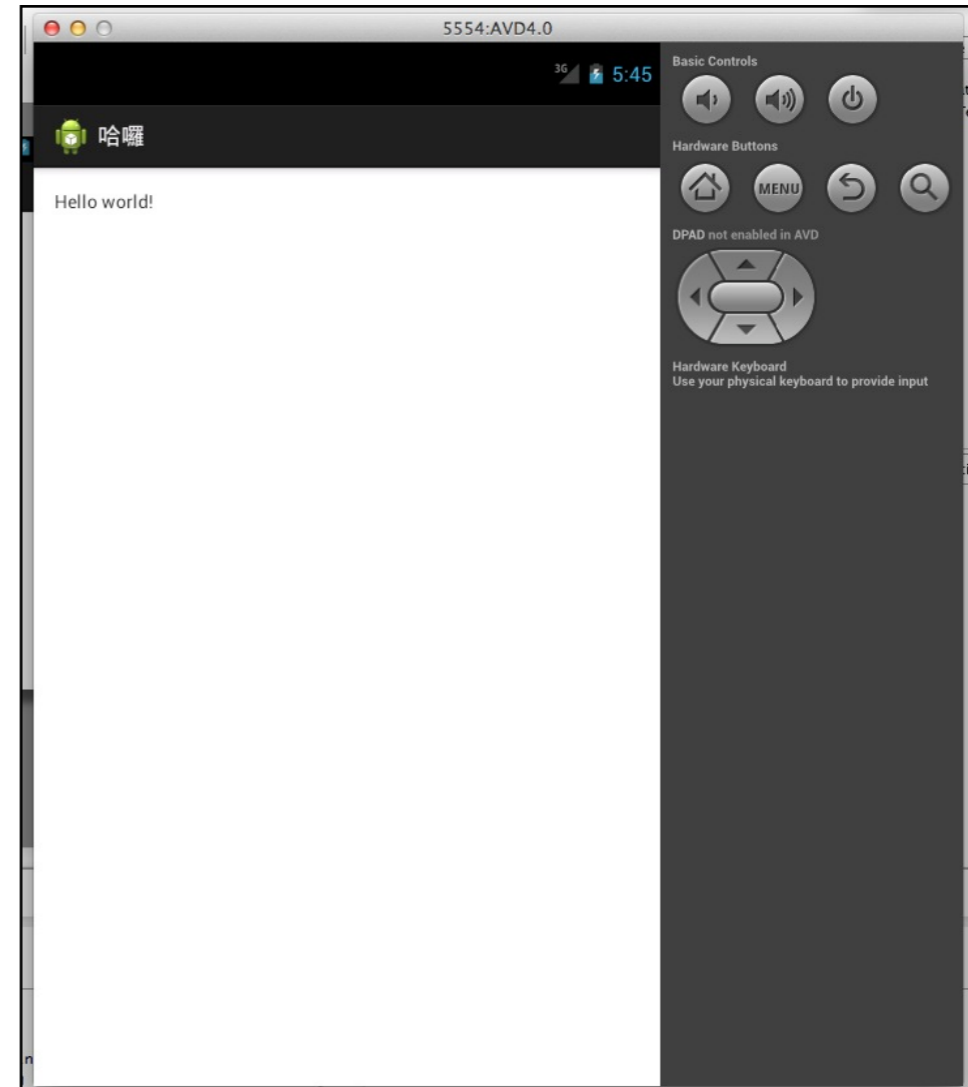
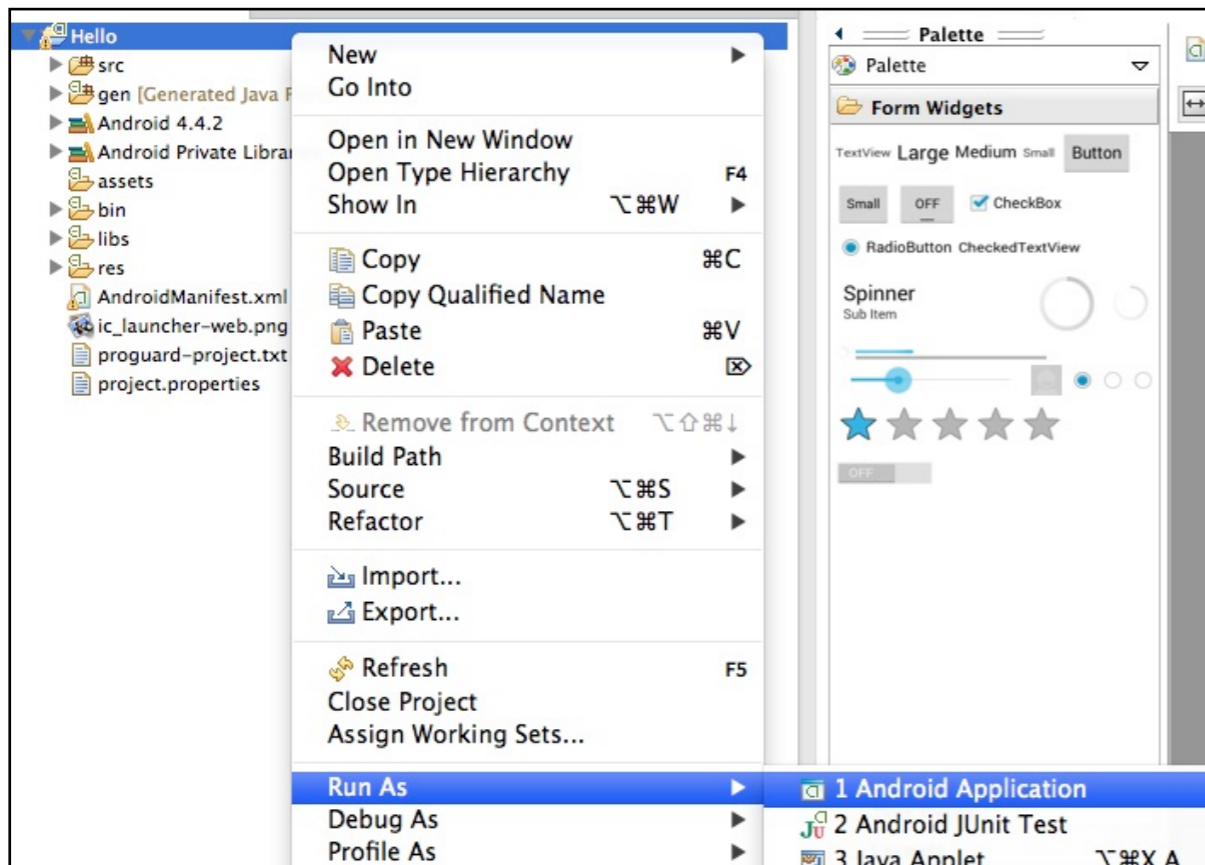
- Start > Launch



(我們的Android模擬裝置)

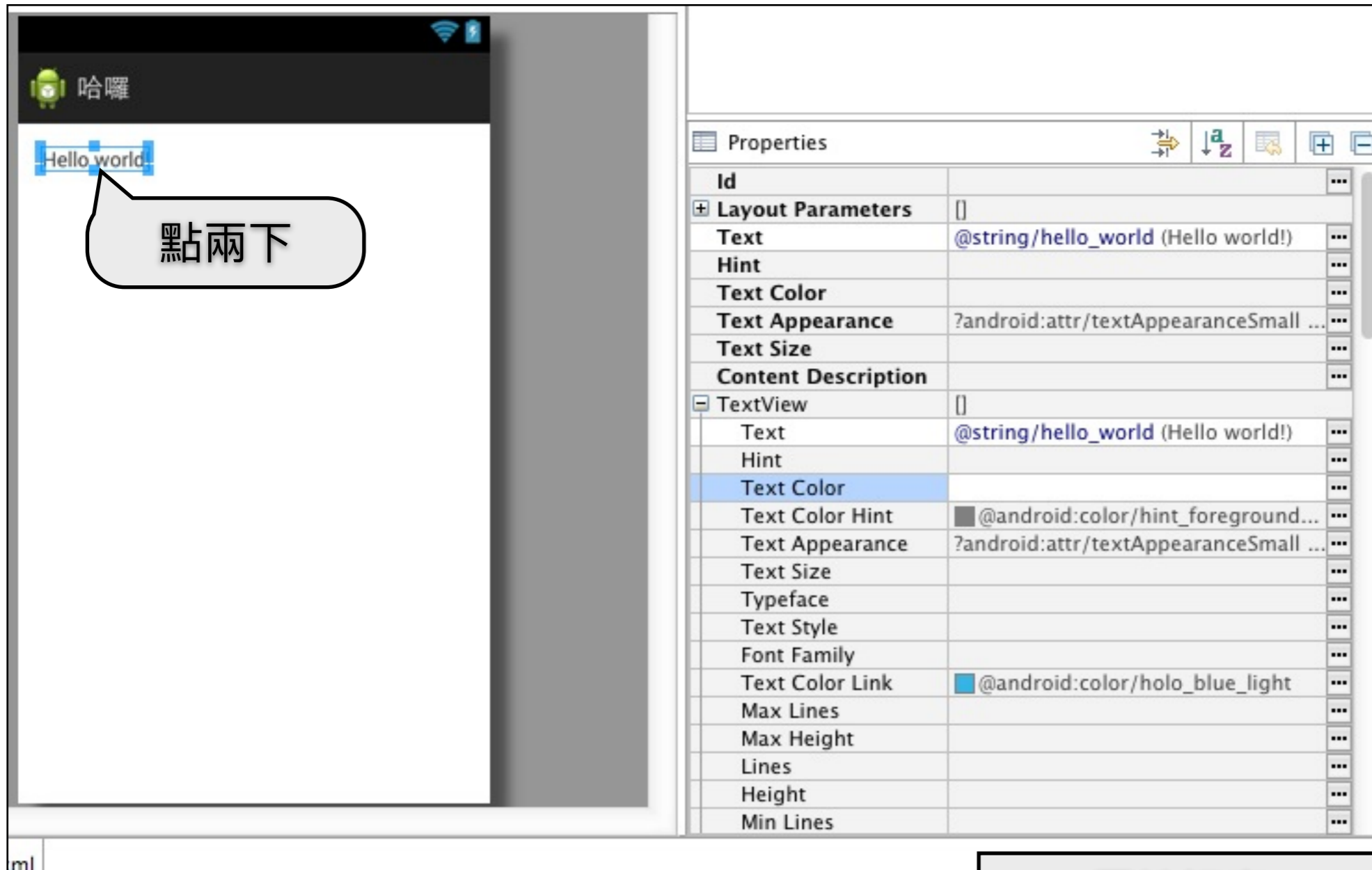
執行專案Hello

- 專案上按右鍵 > Run As > Android Application



(執行結果)

更改一下Hello World



圖示模式

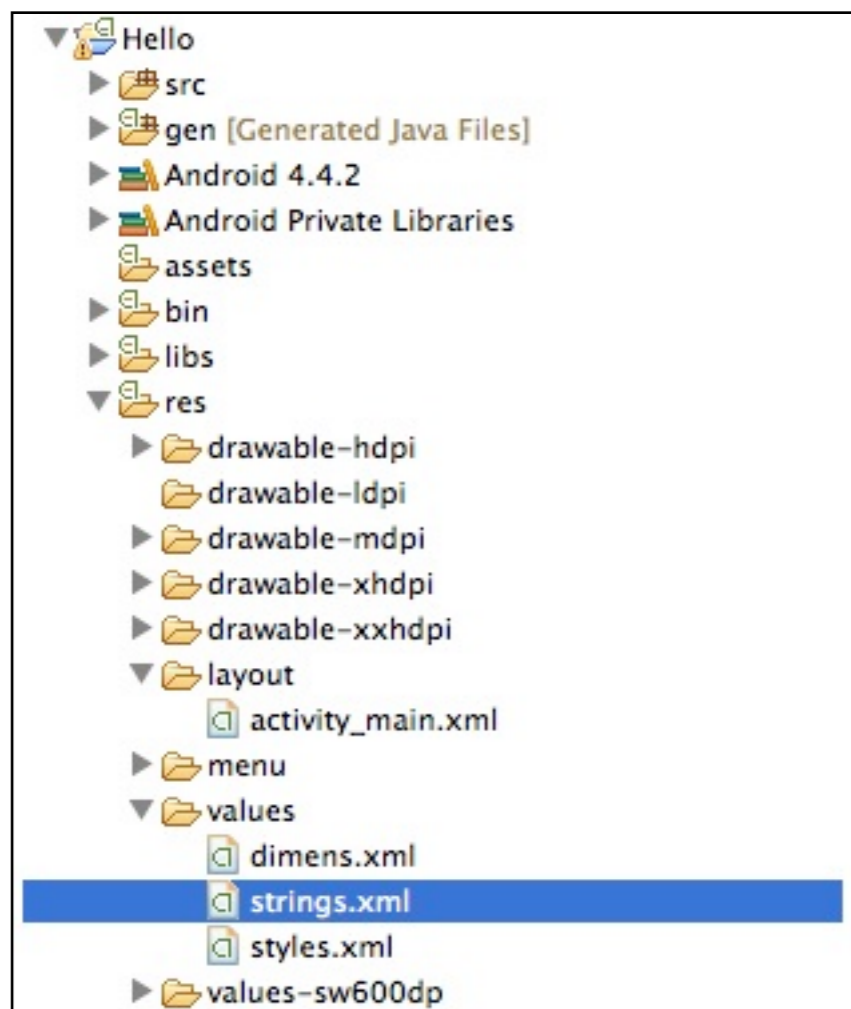
屬性視窗

進入文字編輯模式

```
activity_main.xml MainActivity.java
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:paddingBottom="@dimen/activity_vertical_margin"
6   android:paddingLeft="@dimen/activity_horizontal_margin"
7   android:paddingRight="@dimen/activity_horizontal_margin"
8   android:paddingTop="@dimen/activity_vertical_margin"
9   tools:context=".MainActivity" >
10
11   <TextView
12     android:layout_width="wrap_content"
13     android:layout_height="wrap_content"
14     android:text="@string/hello_world" />
15
16 </RelativeLayout>
17
```

尋找string資源裡的hello_world這個變數對應到的字串

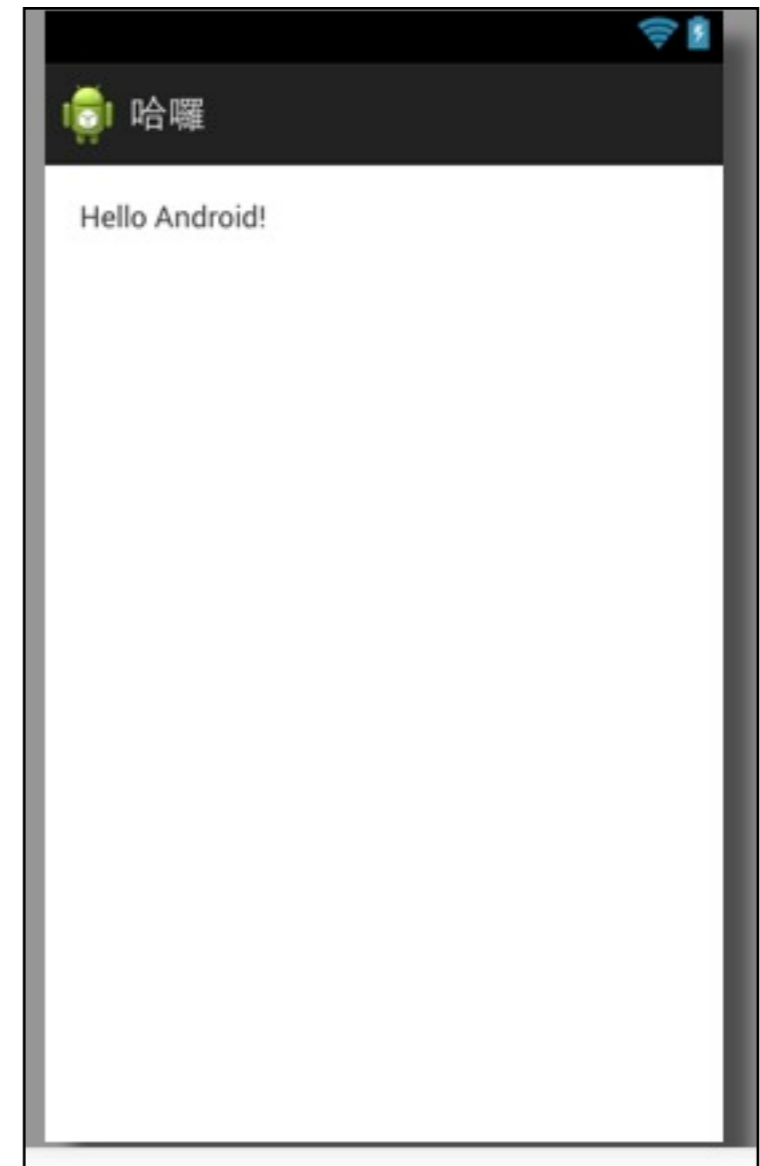
觀看string資源 res > values > strings.xml



```
activity_main.xml MainActivity.java strings.xml ✕
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3
4     <string name="app_name">哈囉</string>
5     <string name="action_settings">Settings</string>
6     <string name="hello_world">Hello world!</string>
7     <string name="hello_android">Hello Android!</string>
8 </resources>
9
```

成功修改成Hello Android

```
activity_main.xml MainActivity.java strings.xml
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:paddingBottom="@dimen/activity_vertical_margin"
6   android:paddingLeft="@dimen/activity_horizontal_margin"
7   android:paddingRight="@dimen/activity_horizontal_margin"
8   android:paddingTop="@dimen/activity_vertical_margin"
9   tools:context=".MainActivity" >
10
11   <TextView
12     android:layout_width="wrap_content"
13     android:layout_height="wrap_content"
14     android:text="@string/hello_android" />
15
16 </RelativeLayout>
17
```



android:layout_width="wrap_content" ??

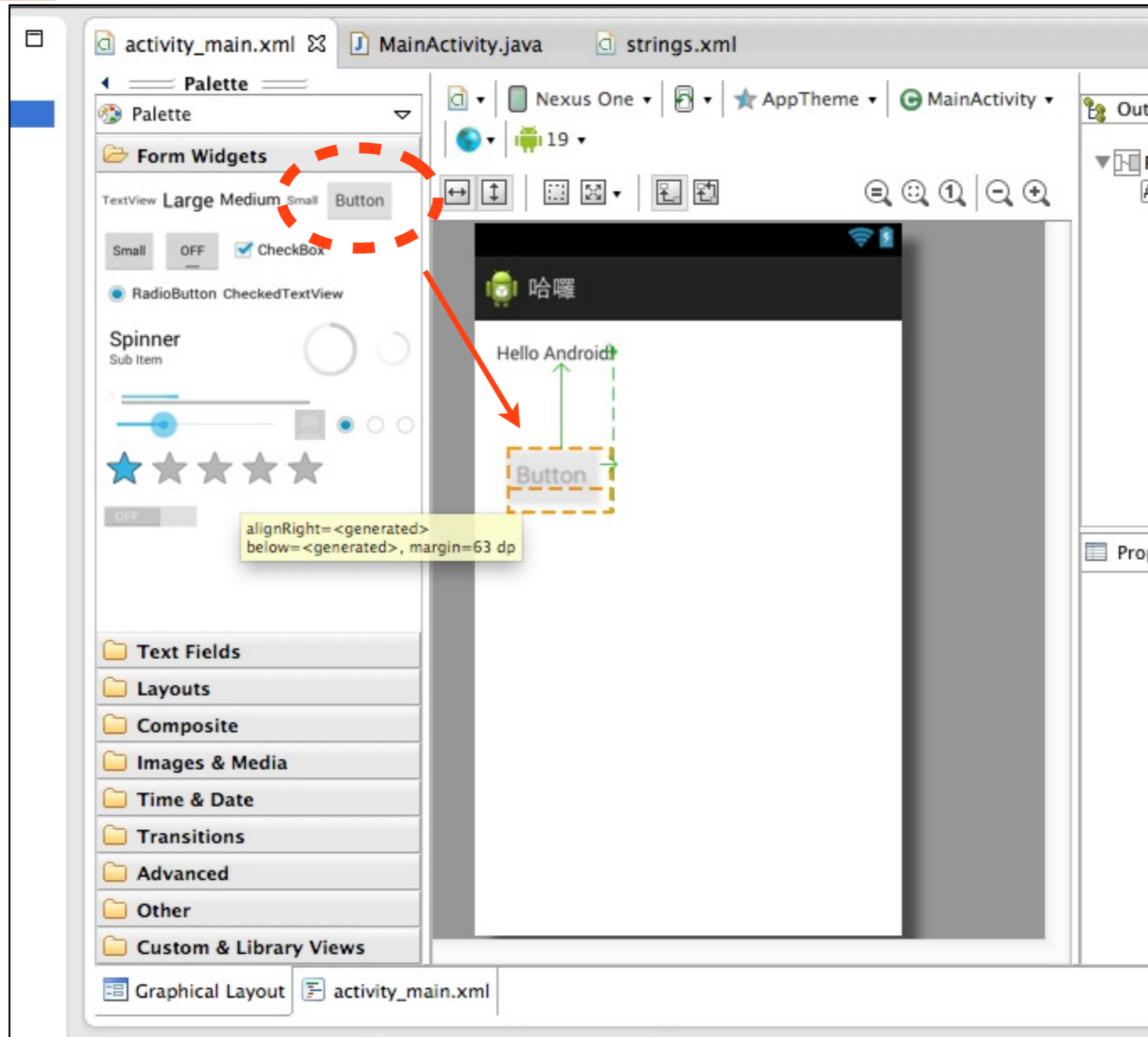
```
activity_main.xml MainActivity.java strings.xml
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:paddingBottom="@dimen/activity_vertical_margin"
6   android:paddingLeft="@dimen/activity_horizontal_margin"
7   android:paddingRight="@dimen/activity_horizontal_margin"
8   android:paddingTop="@dimen/activity_vertical_margin"
9   tools:context=".MainActivity" >
10
11   <TextView
12     android:layout_width="wrap_content"
13     android:layout_height="wrap_content"
14     android:text="@string/hello_android" />
15
16 </RelativeLayout>
17
```

android的width or height常用的長度描述

- `wrap_content`: 指的是長度隨內容而變
- `match_parent`: 指的是長度和父View一樣
- `fill_parent` 和 `match_parent` 一樣
(Android 2.2以前的用法)

加入一個按鈕試試

iOS: UIButton
Android: Button



改變元件?必須先知道元件





在Android中，元件可以透過唯一的id來取得，再指定到相對的參考變數即可。

用`findViewById(id)`即可取得元件對應id的元件

Android如何反應動作?



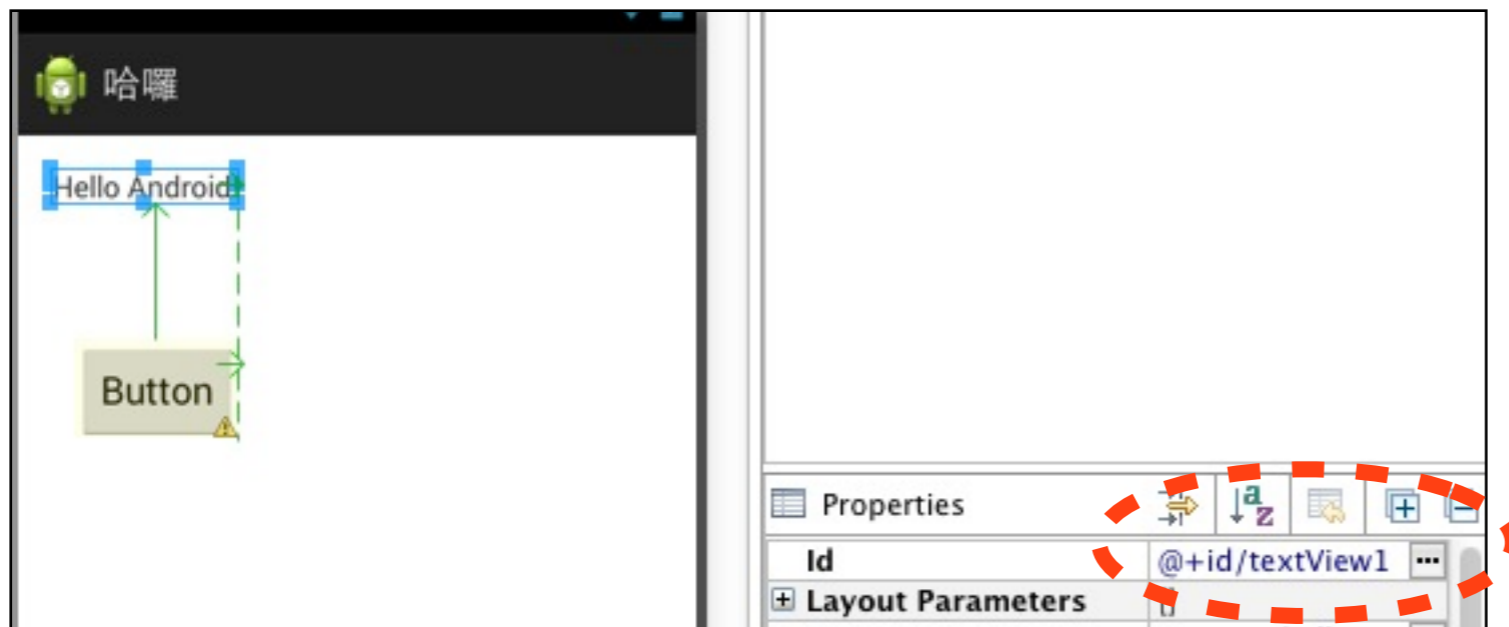


在Android中，元件的動作有其各別提供的觸發方法 (Listener)，元件透過id取得後，再指定其方法即可

按按鈕來變更Hello文字

辨識元件的第一步，指定id

TextView設定id為textview1



(找到元件的id屬性)

取得對應id元件

```
package tw.com.airbridge.hello;

import android.os.Bundle;
import android.app.Activity;

//1. 要使用Button及TextView要先import其對應的class
import android.widget.*;

public class MainActivity extends Activity {

    private Button myButton;
    private TextView myTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity\_main);

        //2. 在取得contentView之後，可以透過findViewById取得元件，記得要轉型
        myTextView=(TextView)findViewById(R.id.textView1);
        myButton=(Button)findViewById(R.id.button1);

    }
}
```

```
package tw.com.airbridge.hello;

import android.os.Bundle;
import android.app.Activity;

//1. 要使用Button及TextView要先import其對應的class
import android.widget.*;

public class MainActivity extends Activity {

    private Button myButton;
    private TextView myTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //2. 在取得contentView之後，可以透過findViewById取得元件，記得要轉型
        myTextView=(TextView)findViewById(R.id.textView1);
        myButton=(Button)findViewById(R.id.button1);

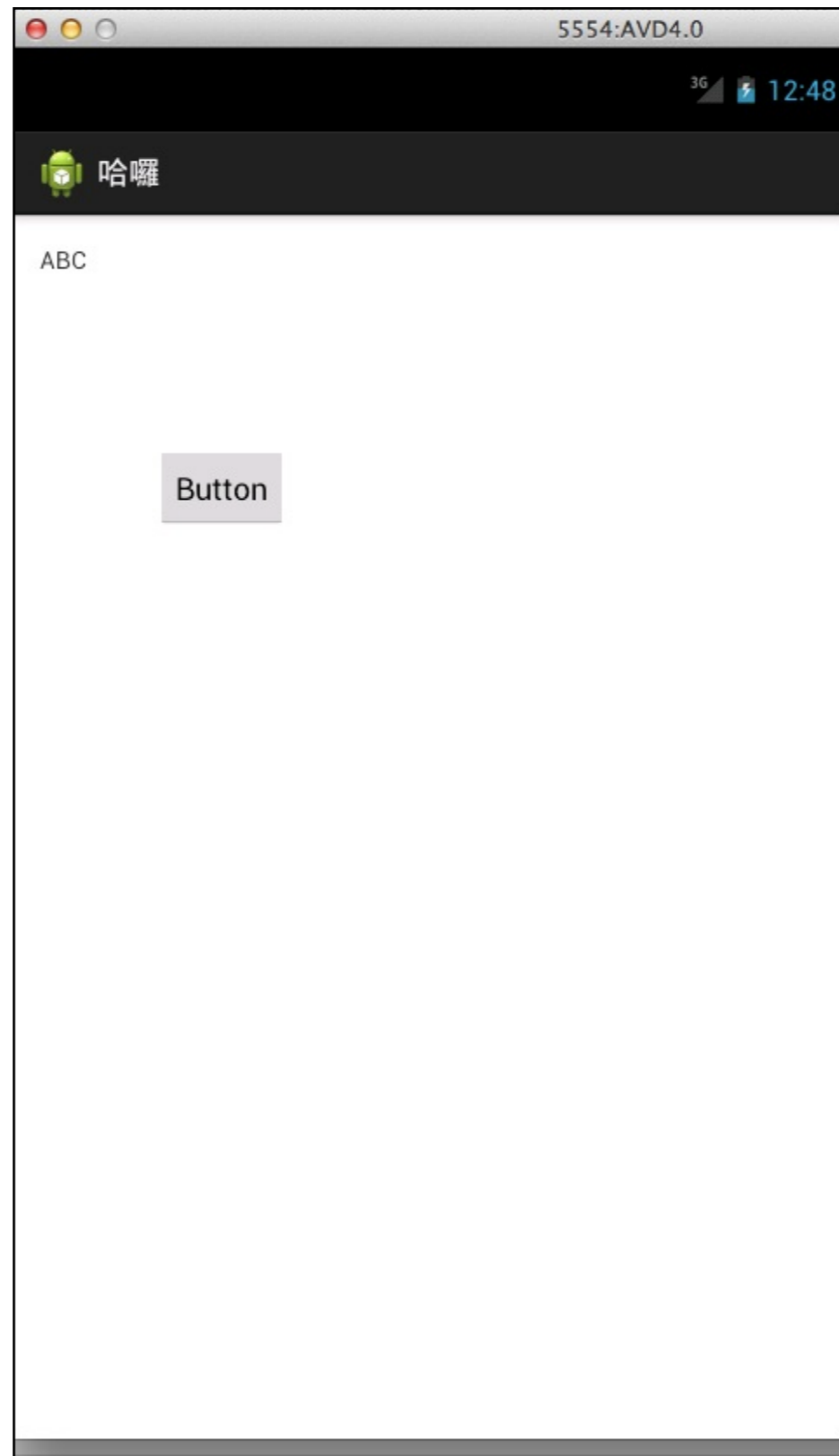
        //3. 指定Button要執行的Listener
        myButton.setOnClickListener(myListener);
    }

    //4. 指定Button在View被點擊時的動作
    private Button.OnClickListener myListener = new Button.OnClickListener(){

        @Override
        public void onClick(View v) {
            myTextView.setText("ABC");
        }

    };
}
```

執行結果



Layout 探討

RelativeLayout

使用時機

- 常為內建layout
- 依照相對關係排列
- 可攜度較差
- 盡量不要使用

MainActivity.java activity_main.xml

Palette

Form Widgets

TextView Large Medium Small Button

Small OFF CheckBox

RadioButton CheckedTextView

Spinner
Sub Item

OFF

Text Fields

Layouts

Composite

Images & Media

Time & Date

Transitions

Advanced

Custom & Library Views

Graphical Layout activity_main.xml

Nexus S

AppTheme {activityClass}

layout1

Hello world!

TextView

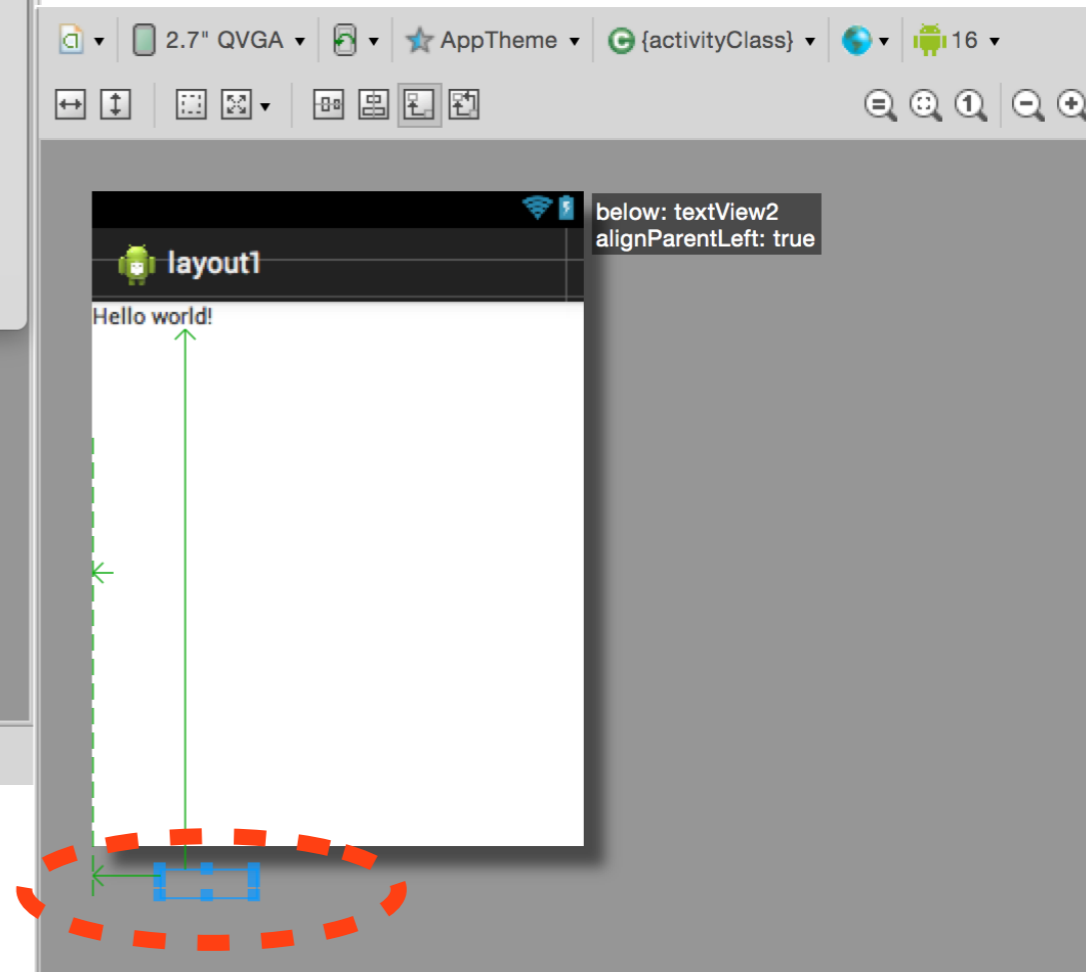
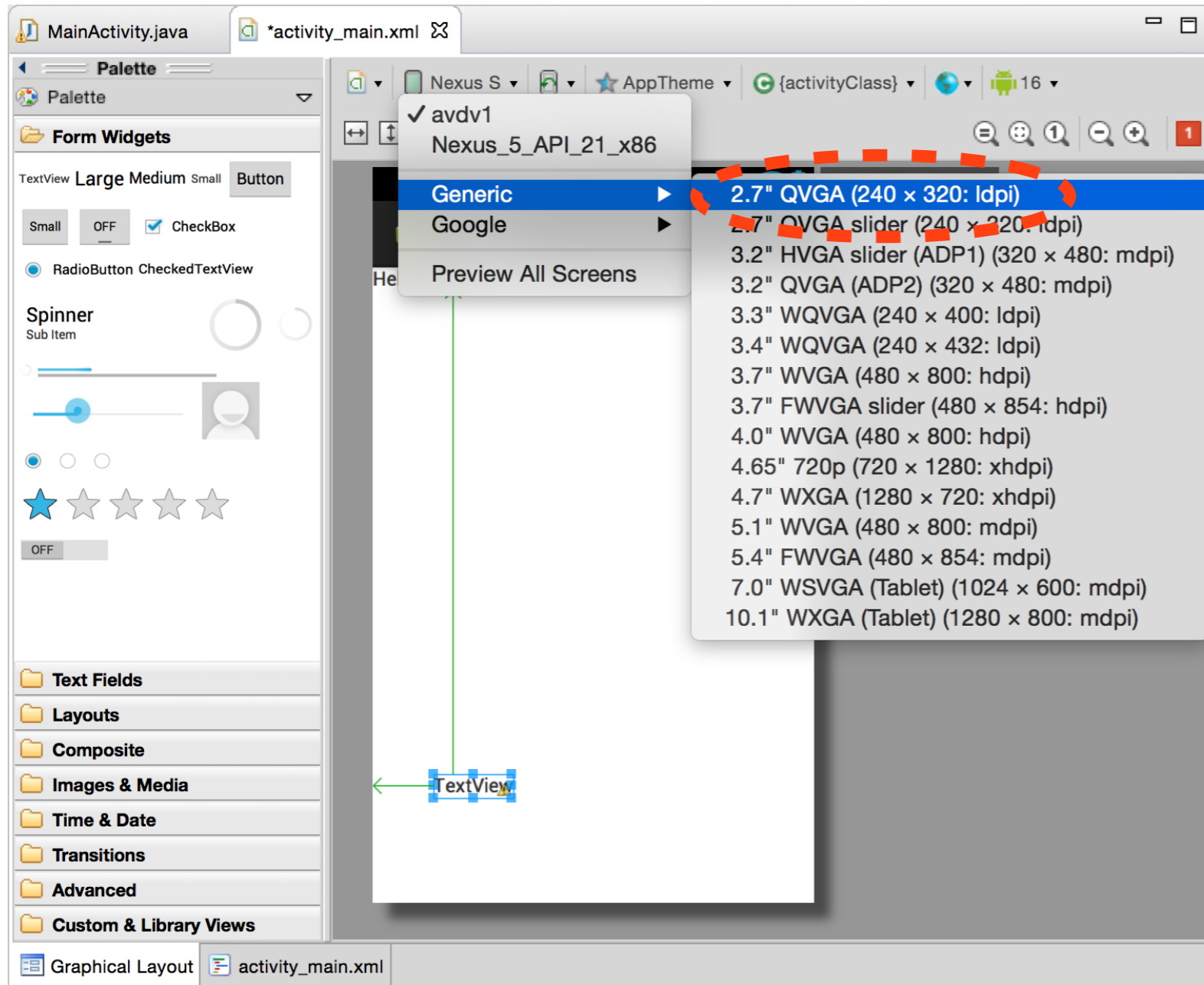
MainActivity.java *activity_main.xml

```

1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   tools:context="${relativePackage}.${activityClass}" >
6
7   <TextView
8     android:id="@+id/textView2"
9     android:layout_width="wrap_content"
10    android:layout_height="wrap_content"
11    android:text="@string/hello_world" />
12
13   <TextView
14     android:id="@+id/textView1"
15     android:layout_width="wrap_content"
16     android:layout_height="wrap_content"
17     android:layout_alignParentLeft="true"
18     android:layout_below="@+id/textView2"
19     android:layout_marginLeft="44dp"
20     android:layout_marginTop="350dp"
21     android:text="TextView" />
22
23 </RelativeLayout>

```

切換裝置後



LinearLayout

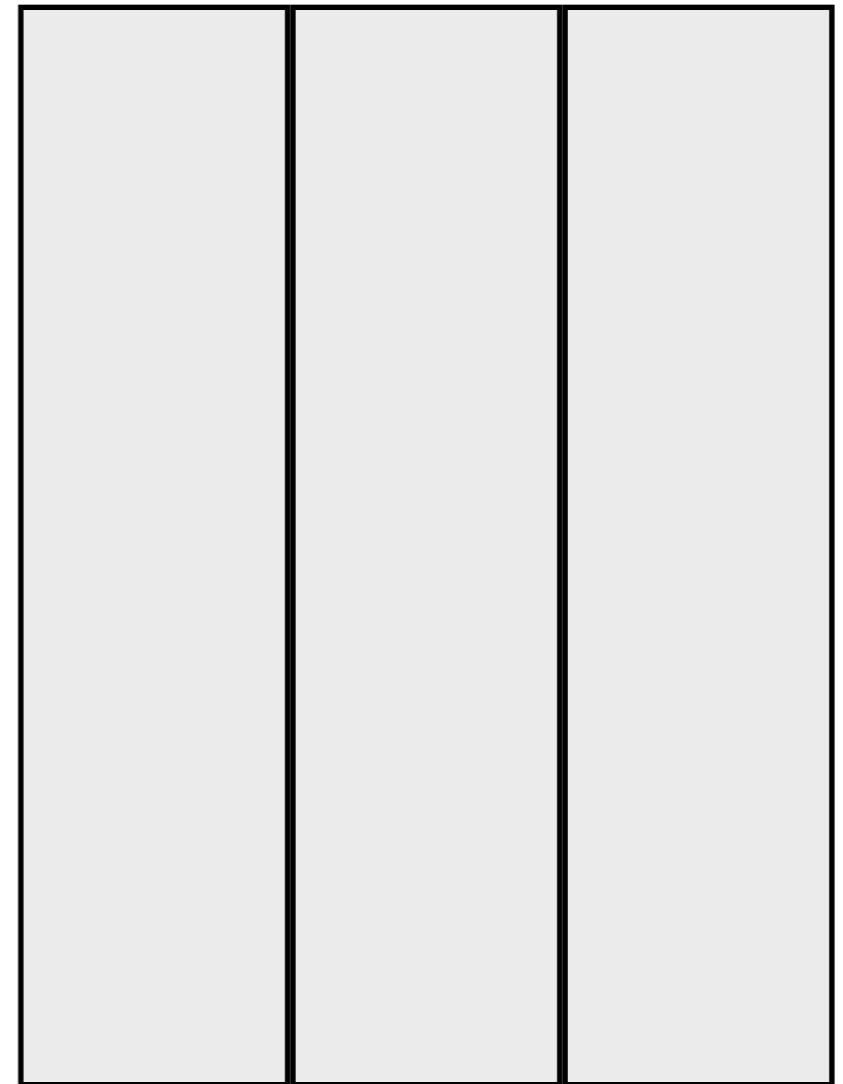
使用時機

- 最常使用layout
- 可依照順序排列
- 可依照比例設定長度
- 缺點:無法重疊顯示

LinearLayout



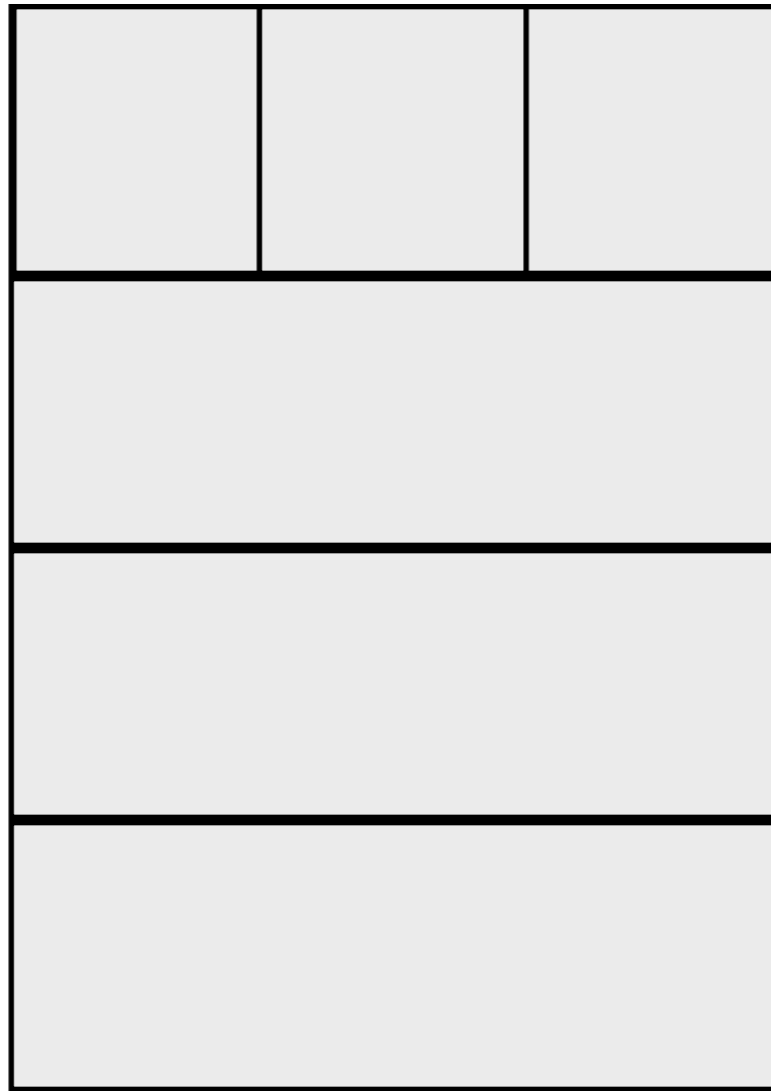
Vertical



Horizontal

所有元件按照順序排列

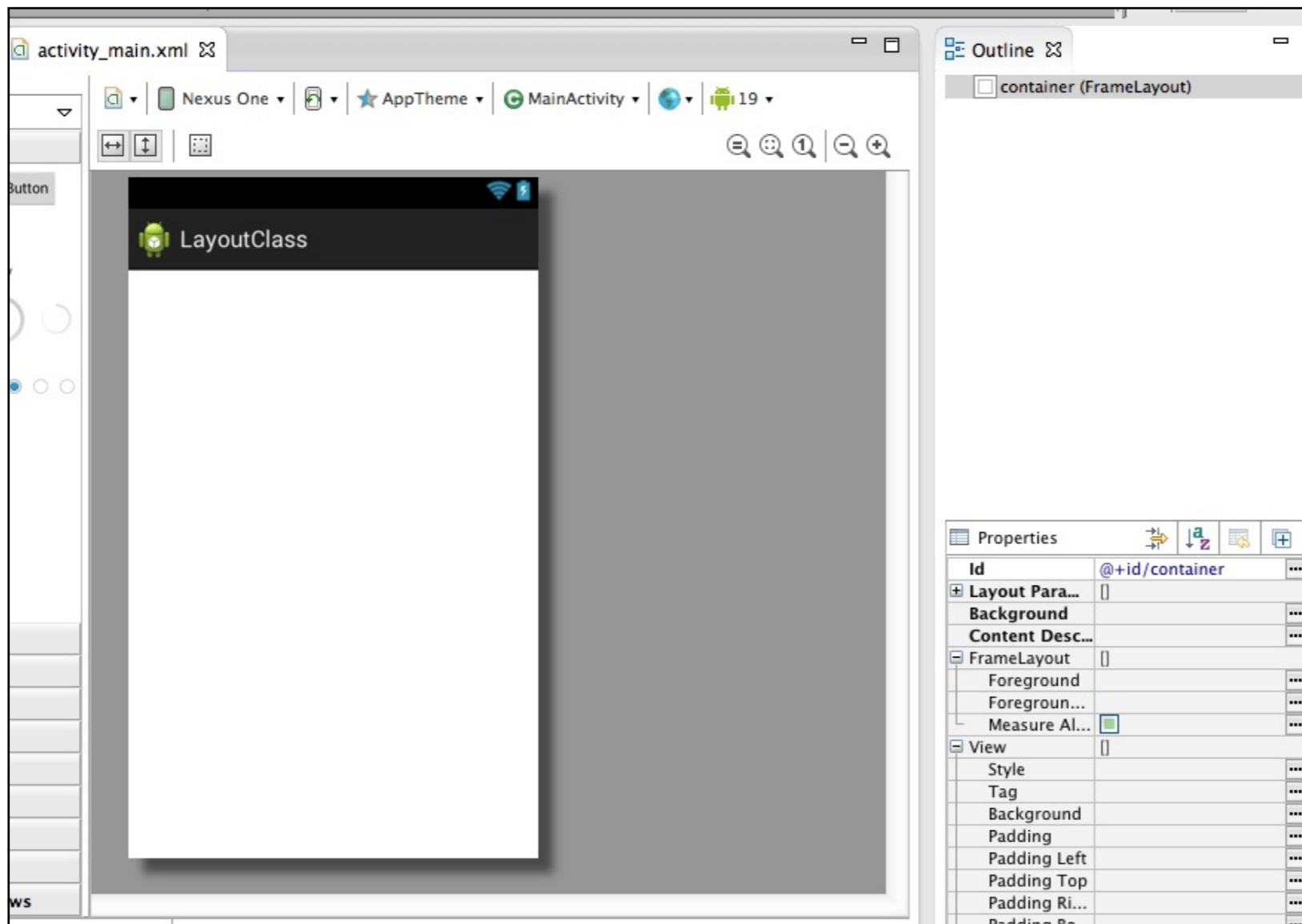
巢狀的LinearLayout



垂直之中有水平

Example: LayoutClass

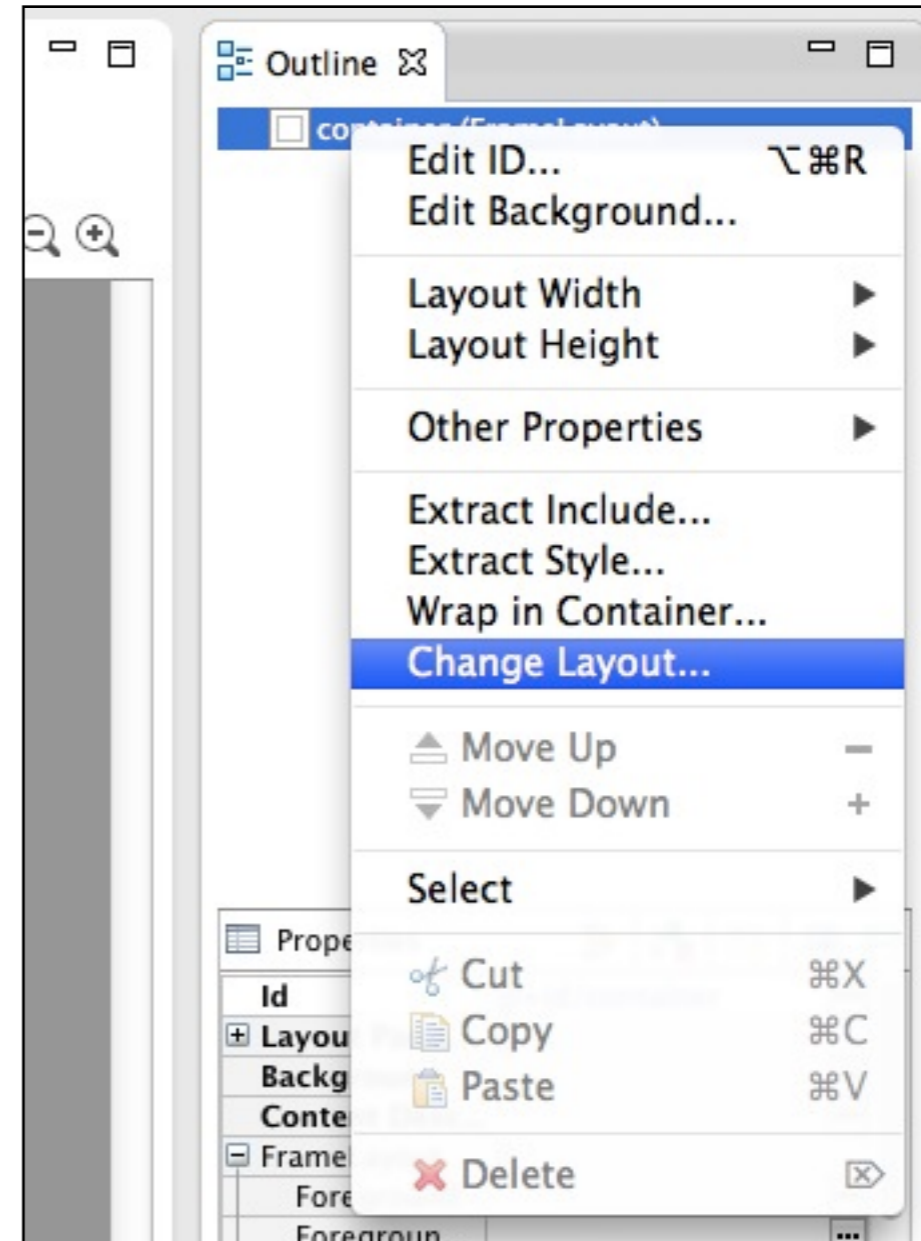
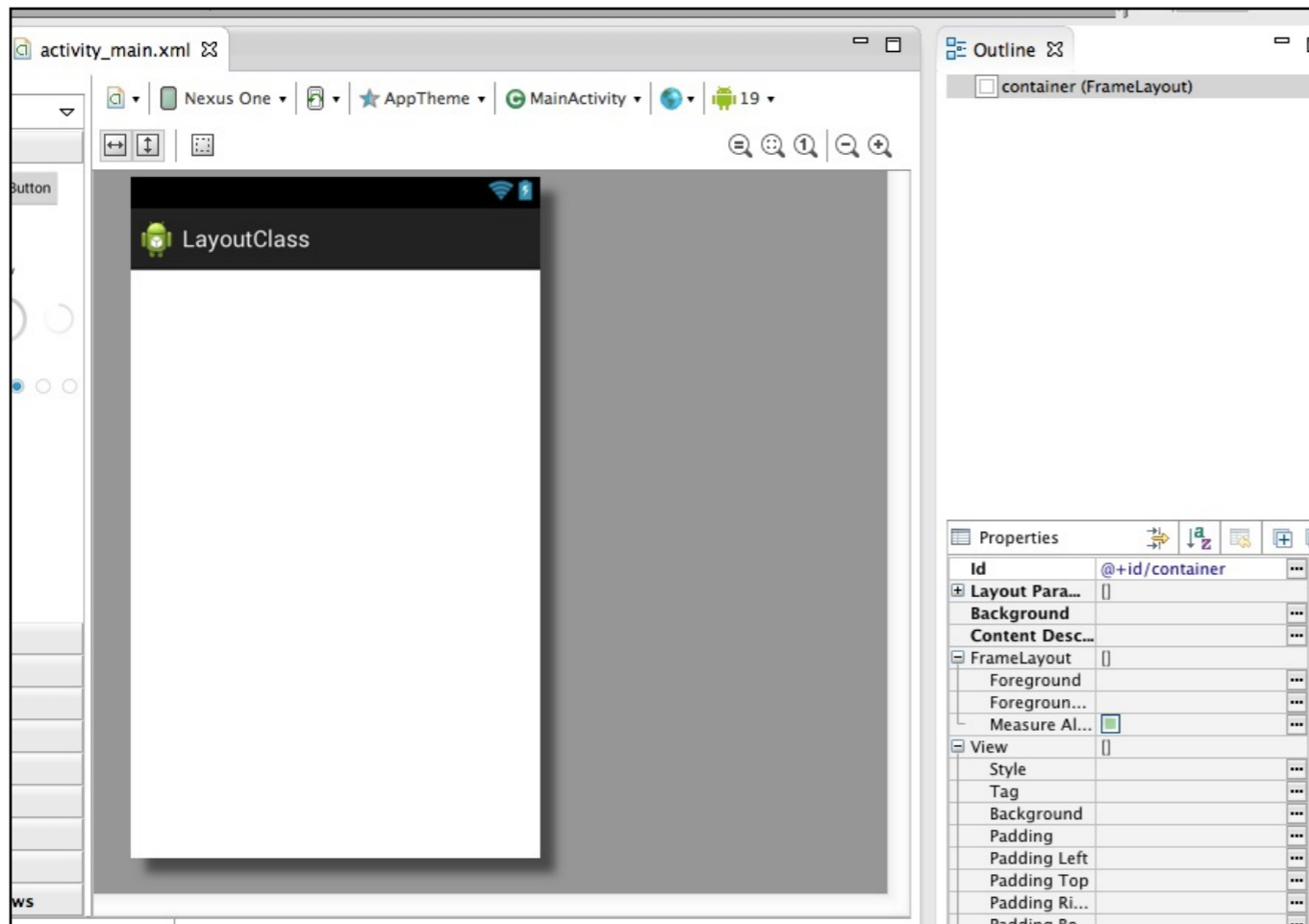
- 建立一專案名為LayoutClass



最後結果

Example: LayoutClass

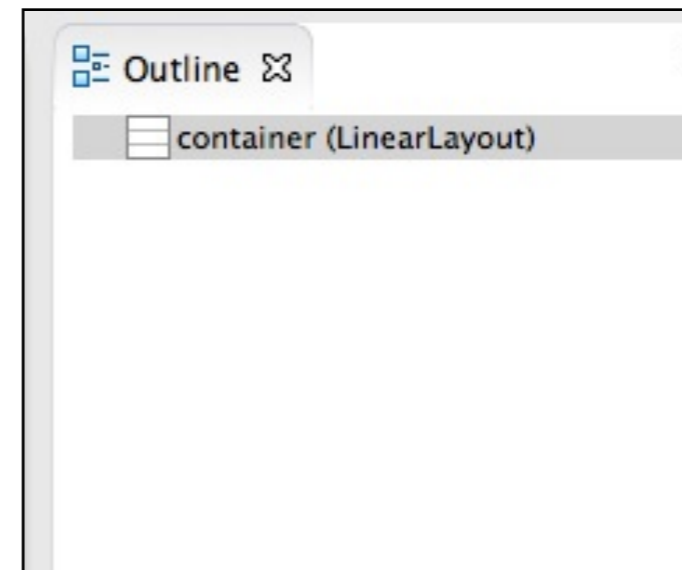
內建為FrameLayout: 常用於畫面
切換，是一個畫面的容器，裡面
的View會重疊，依照顯示時間不
同出現View



變更為Vertical



變更前

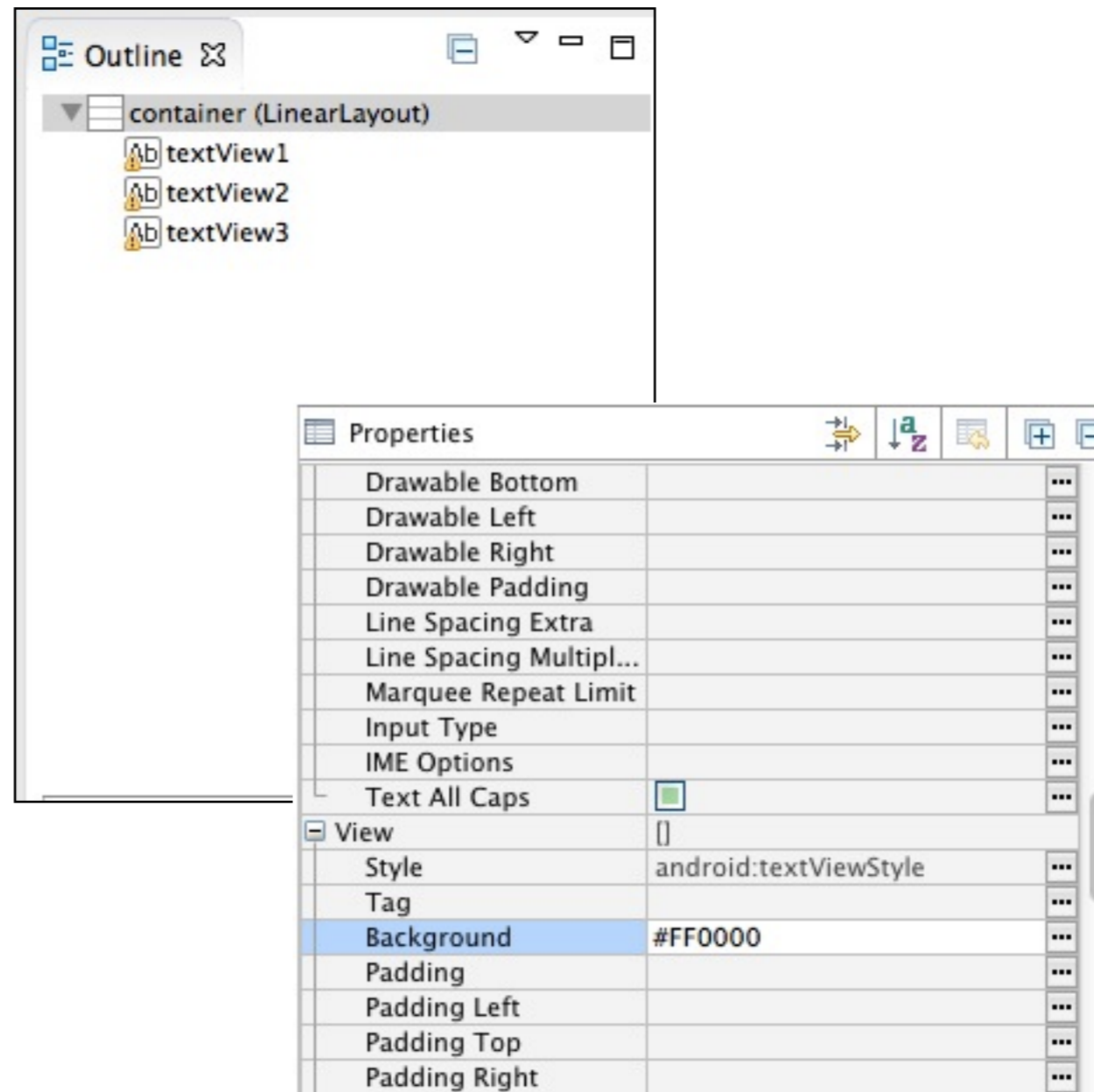


變更後，元件可依序由
上而下排列

加入TextView1 ~ TextView3，並且變更

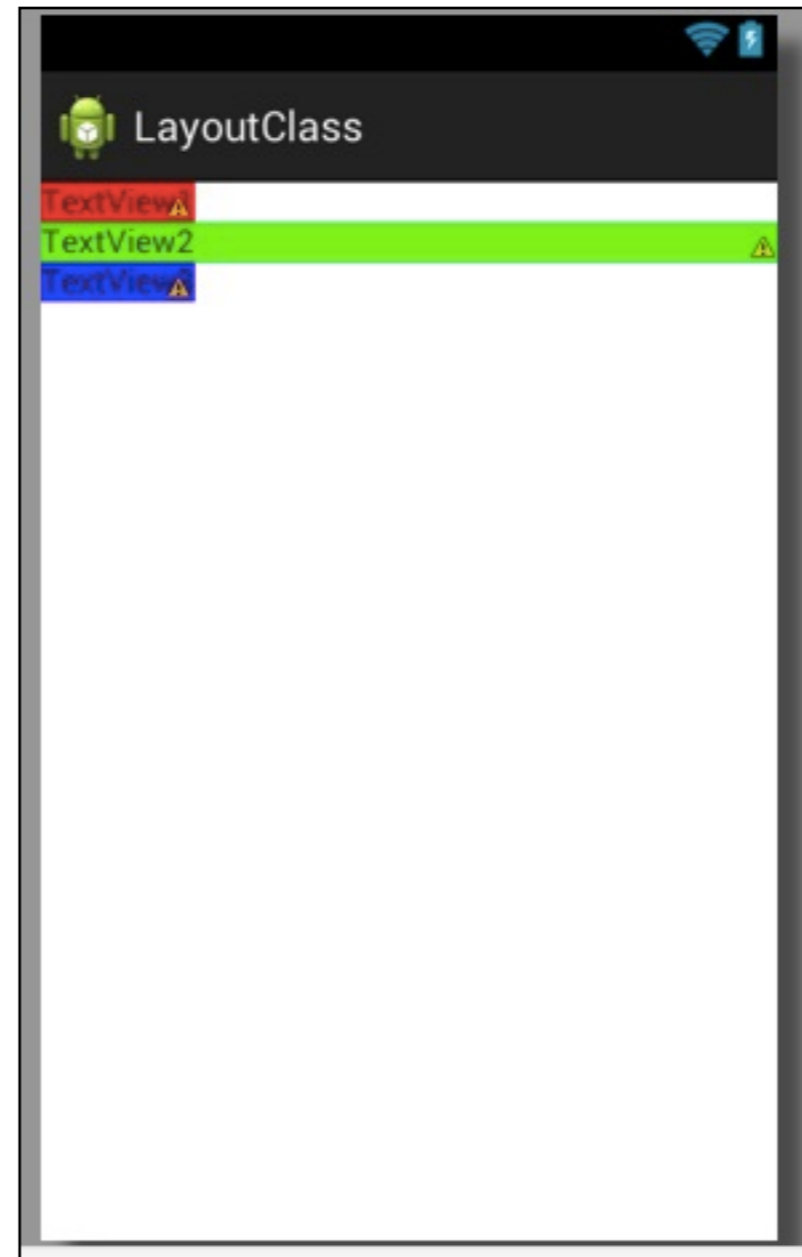
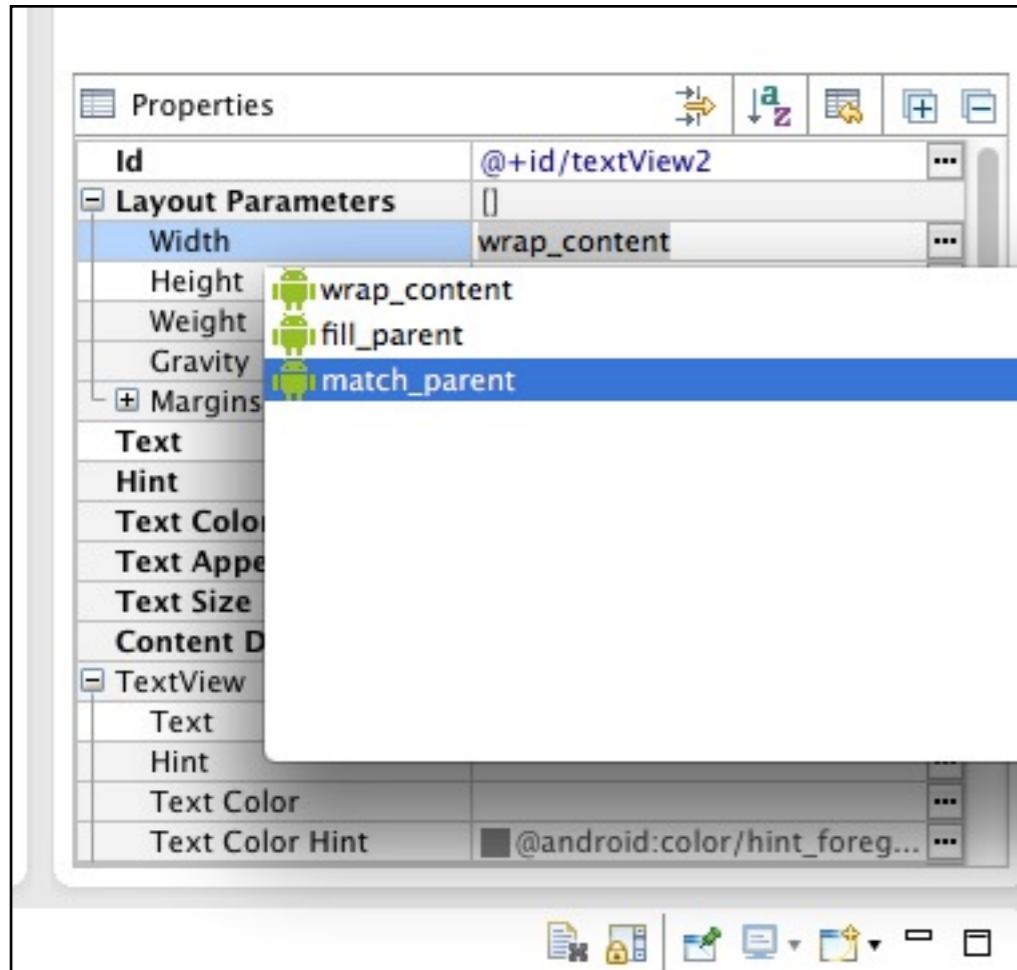


Layout結果



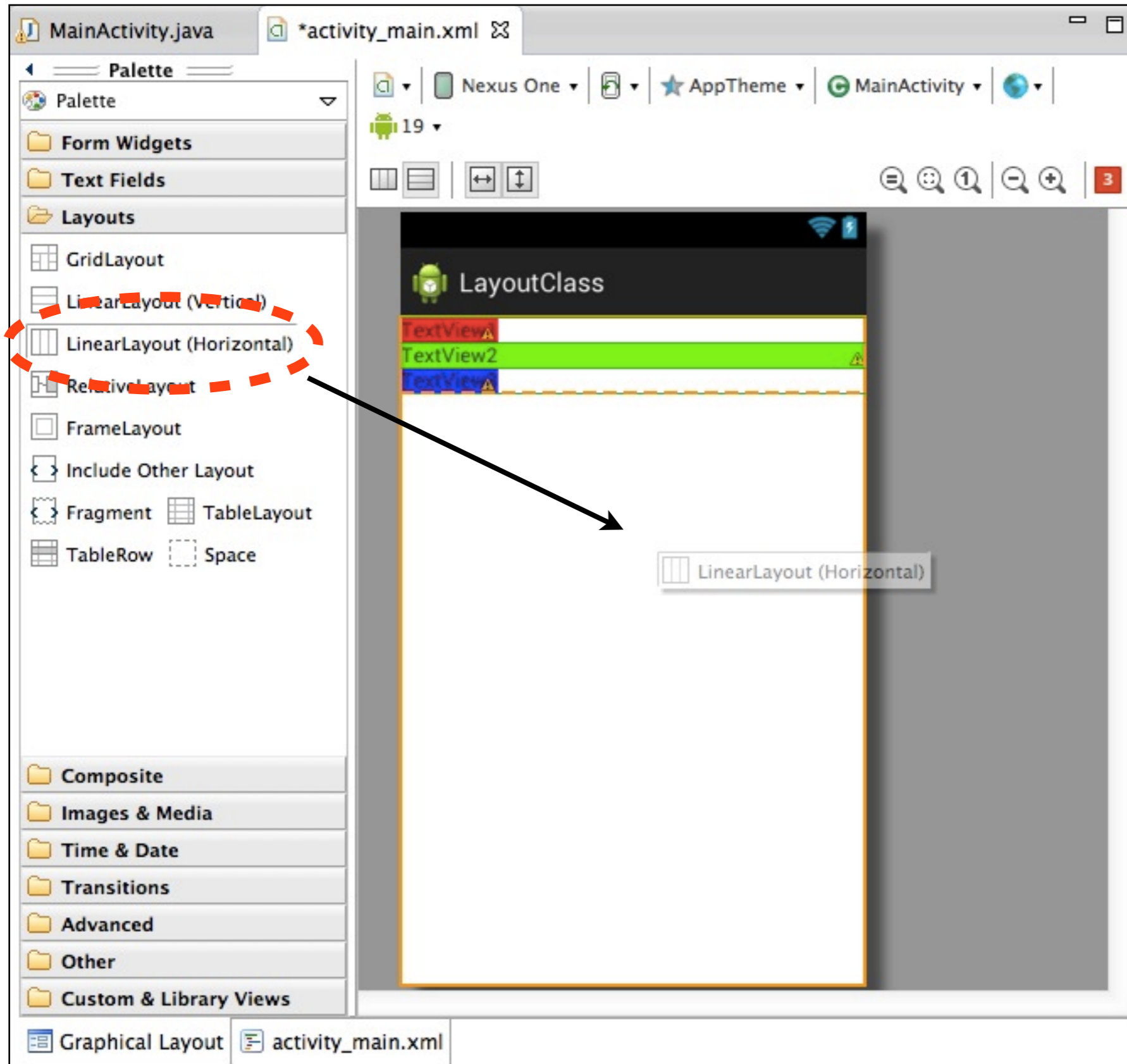
從屬性欄位的View>Background變

變更TextView2的寬度為match_parent



Layout結果，最寬的範圍和
parent View一樣

加入一個Horizontal的Layout



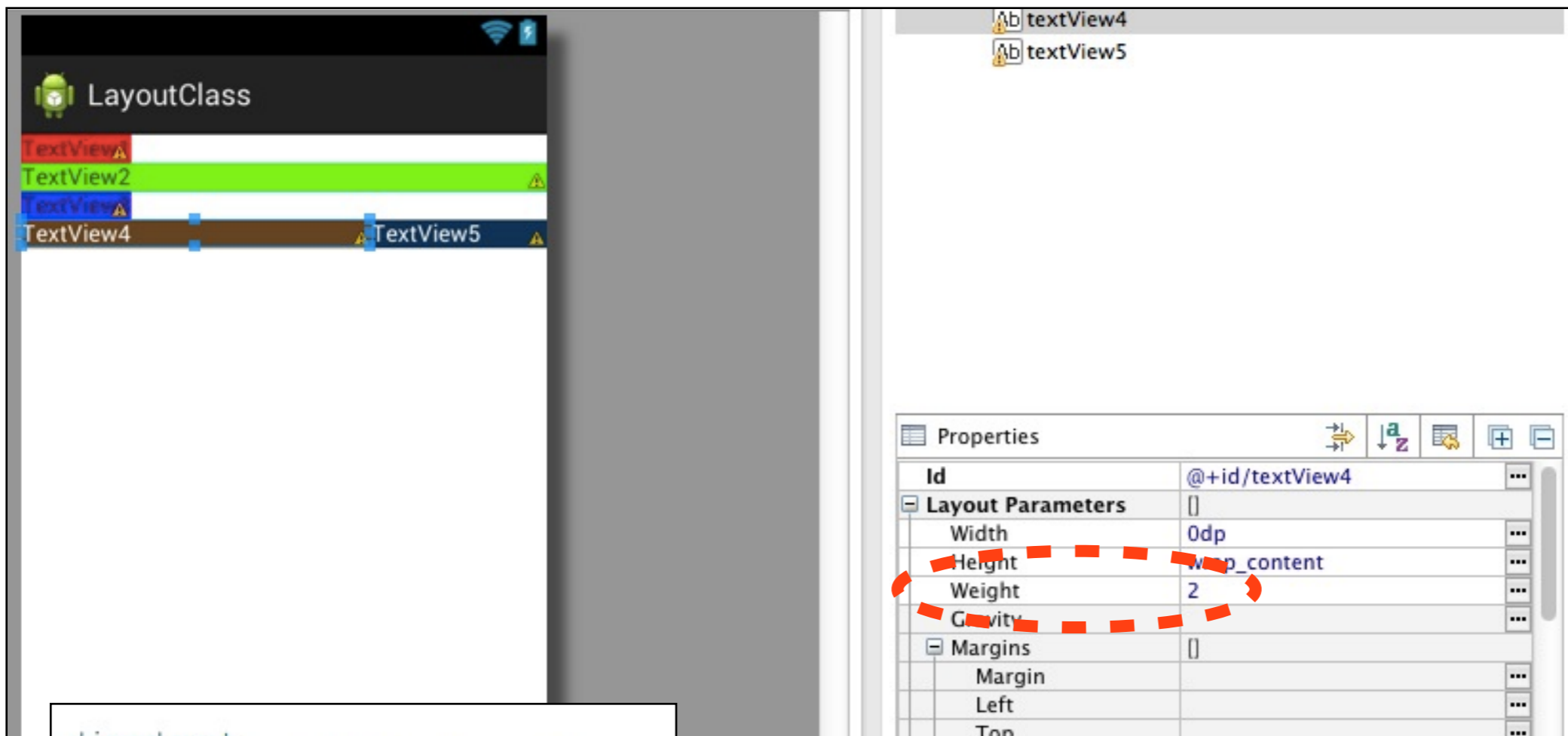
加入二個TextView到Horizontal



Android的Layout差異性解決方案

- `wrap_content`、`match_parent`、`fill_parent` 是符合Android眾多不同Layout的一種方式
- `weight`(權重比例)是另一種等比例縮放的解決方式

變更TextView4的weight為2，TextView5的Weight為1



```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content" >

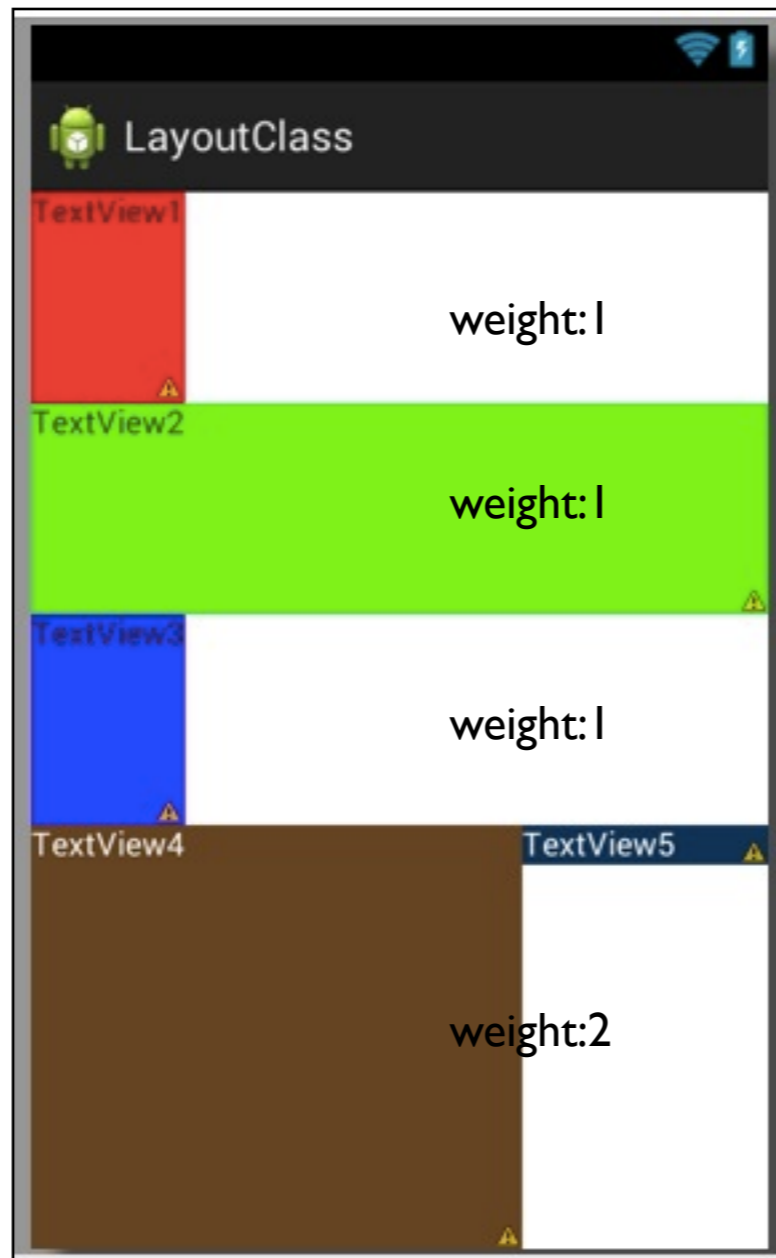
  <TextView
    android:id="@+id/textView4"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:background="#654321"
    android:text="TextView4"
    android:textColor="#FFFFFF" />

  <TextView
    android:id="@+id/textView5"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:background="#123456"
    android:text="TextView5"
    android:textColor="#FFFFFF" />

</LinearLayout>
```

注意!!
當要依照比例改變的是寬度的時候，

Exercise: 請完成如下Layout

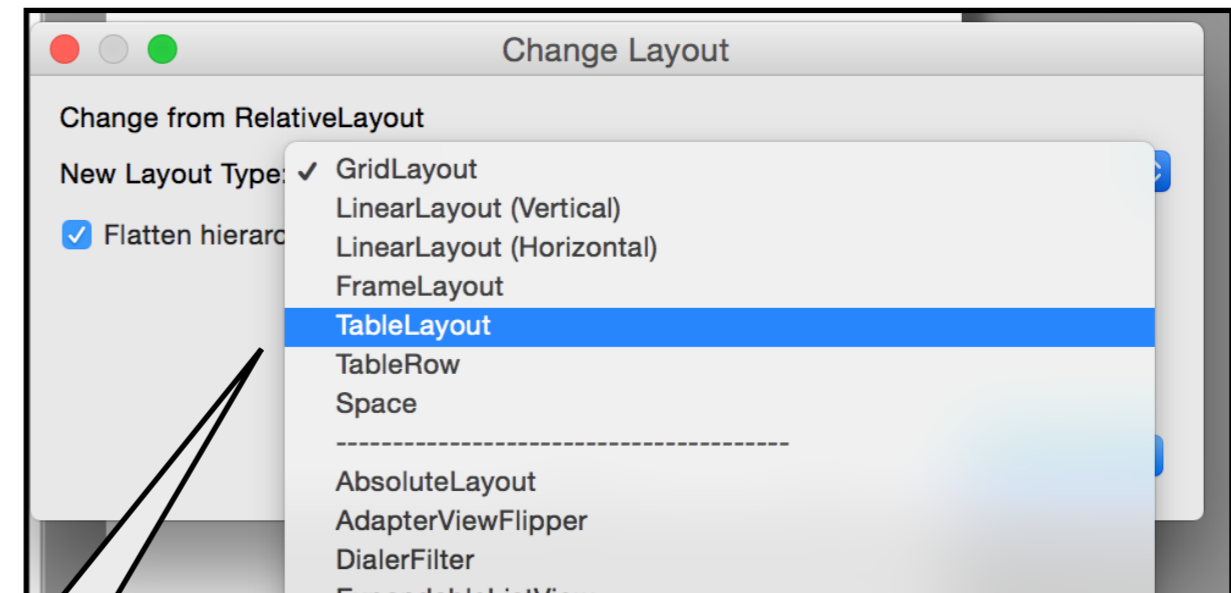
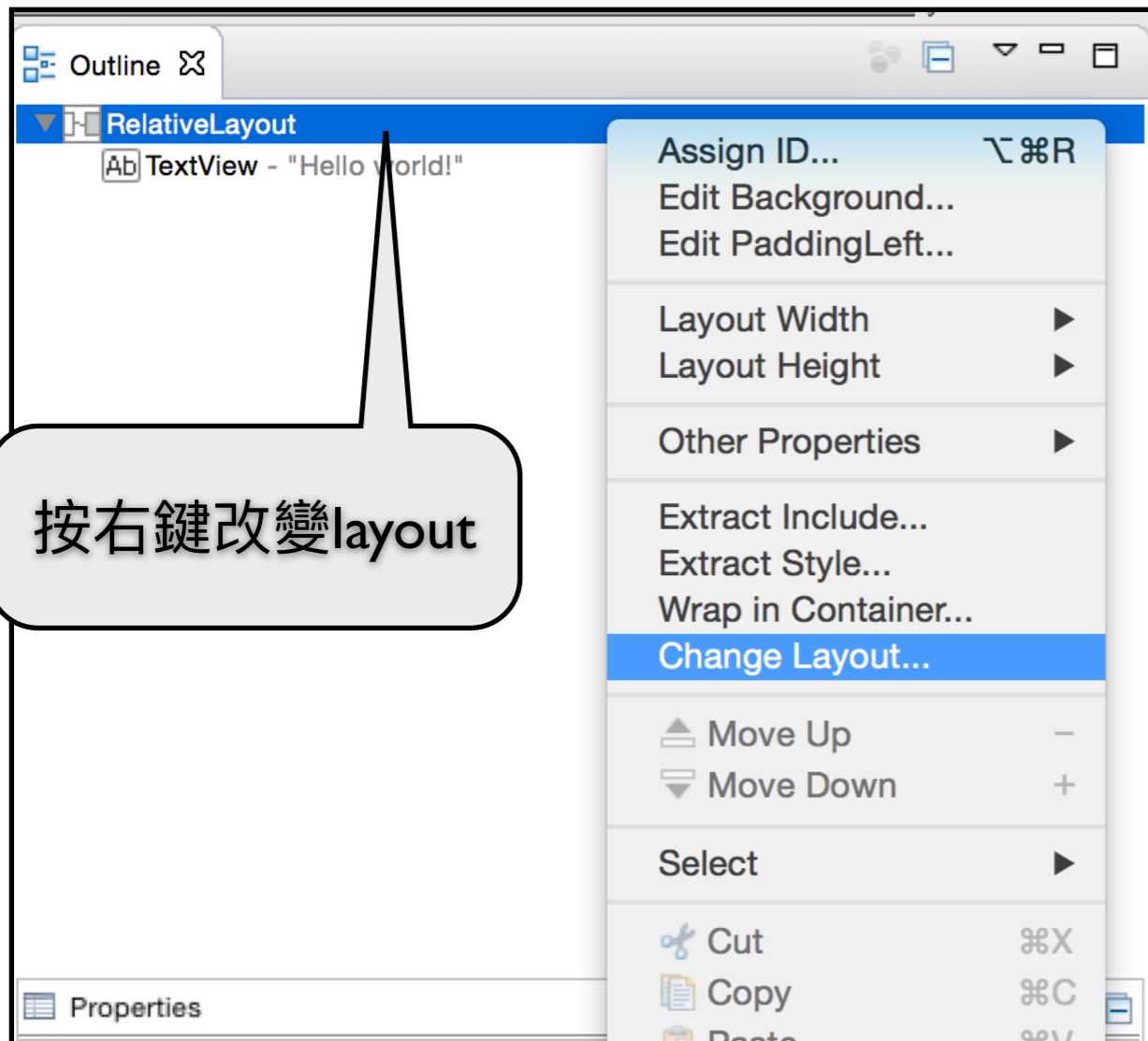


TableLayout

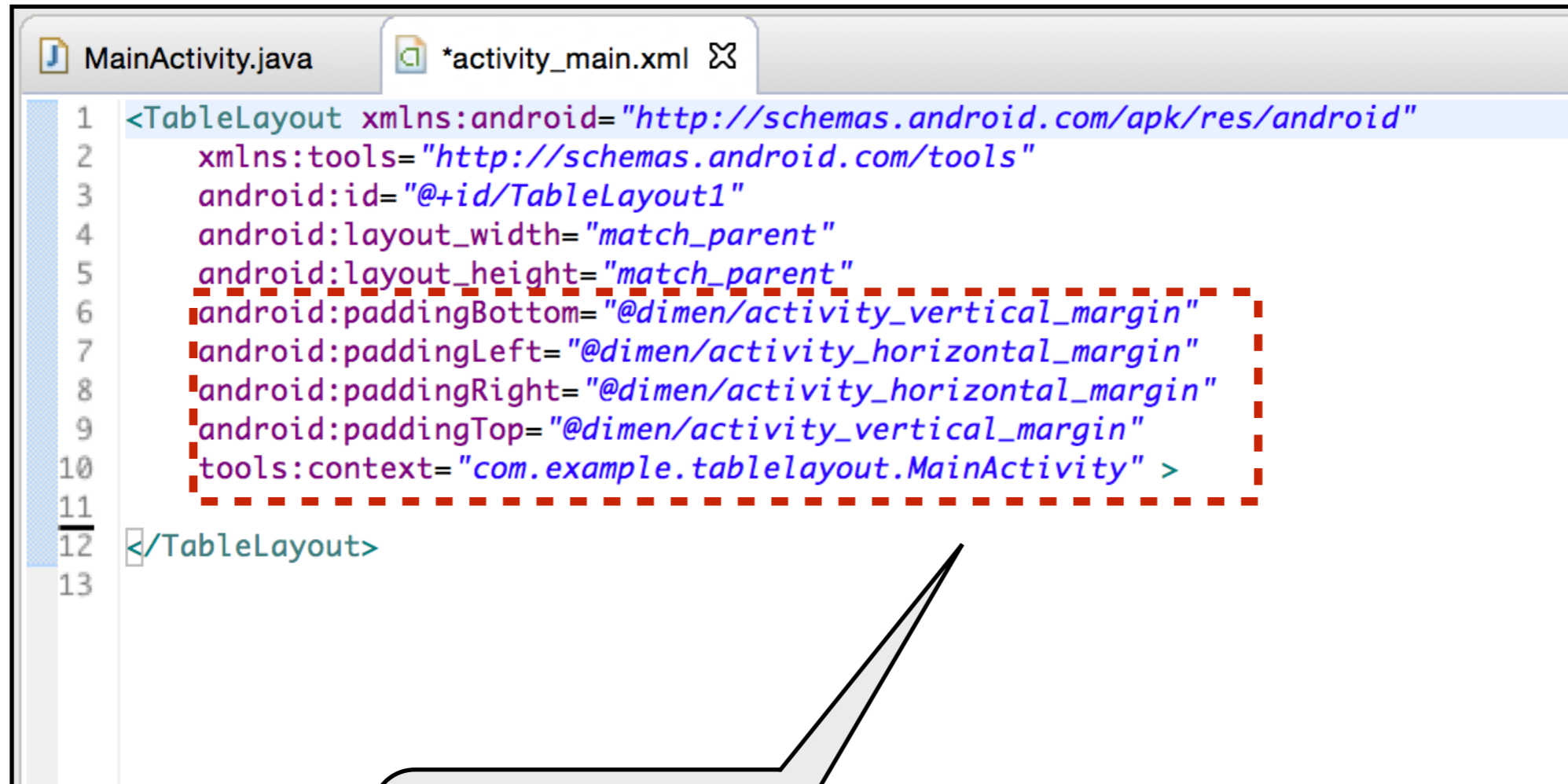
使用時機

- 對表格型態的layout很方便

建立專案，改變Layout為TableLayout



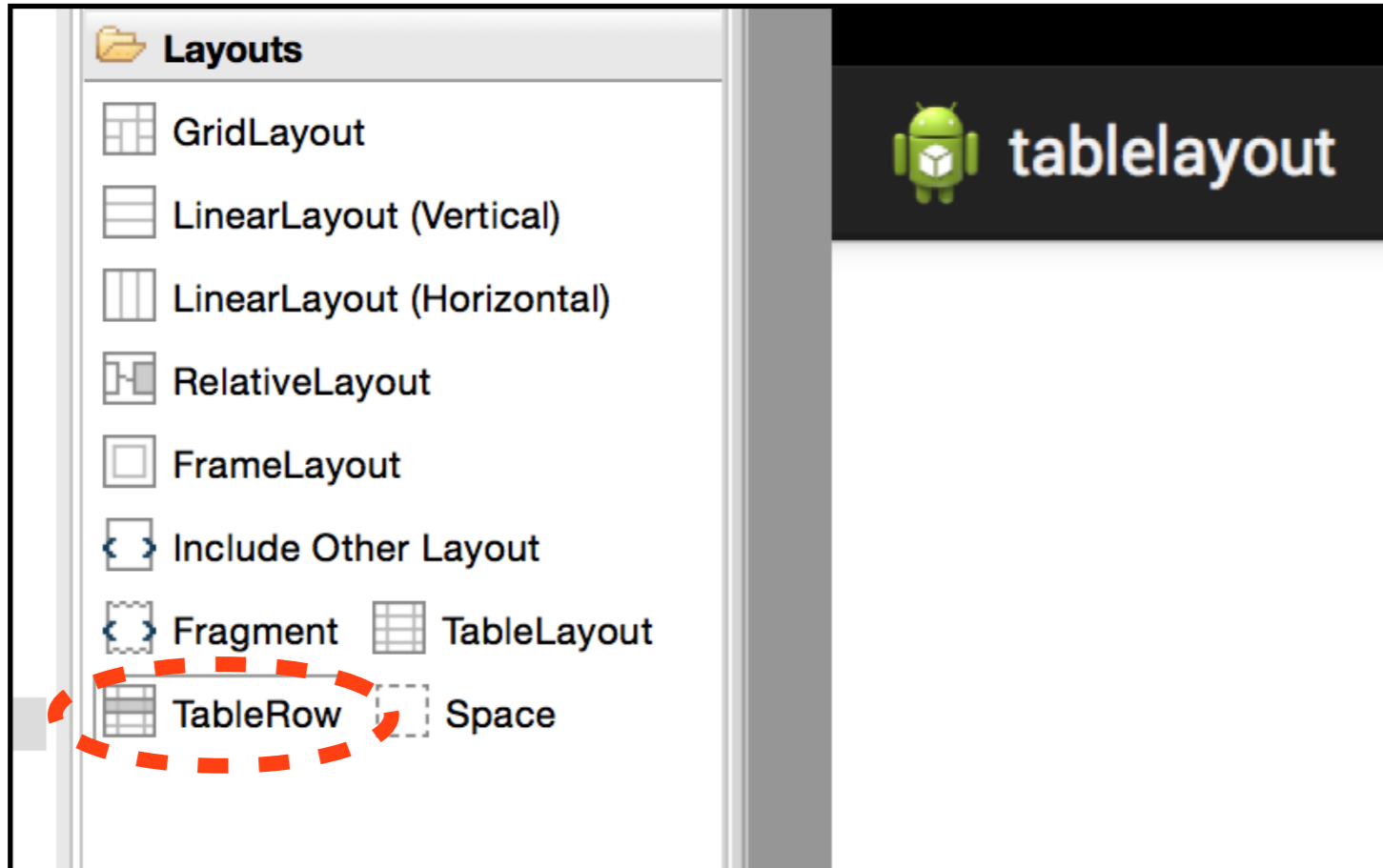
改變layout時產生的多餘設定，造成layout有邊界



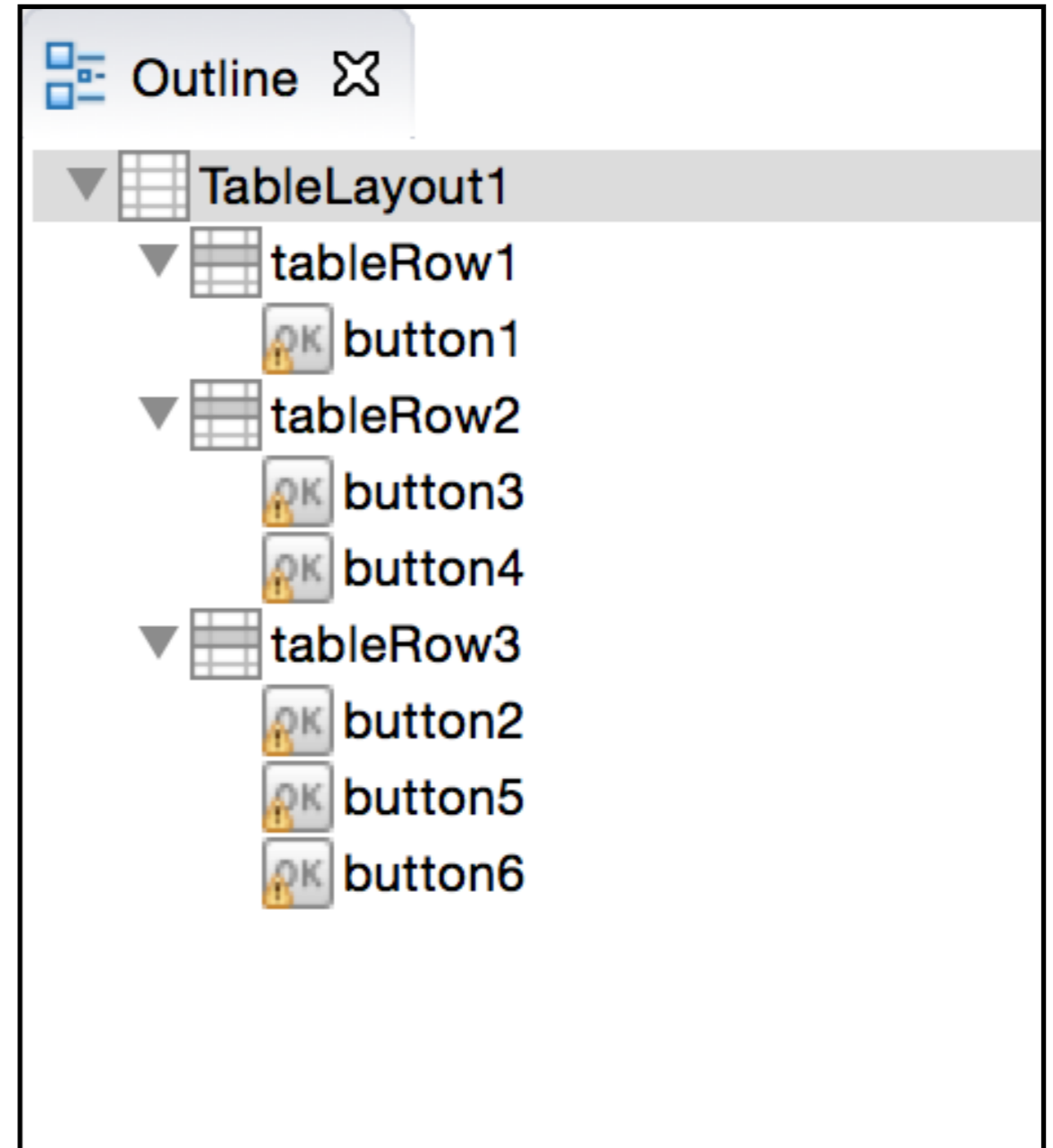
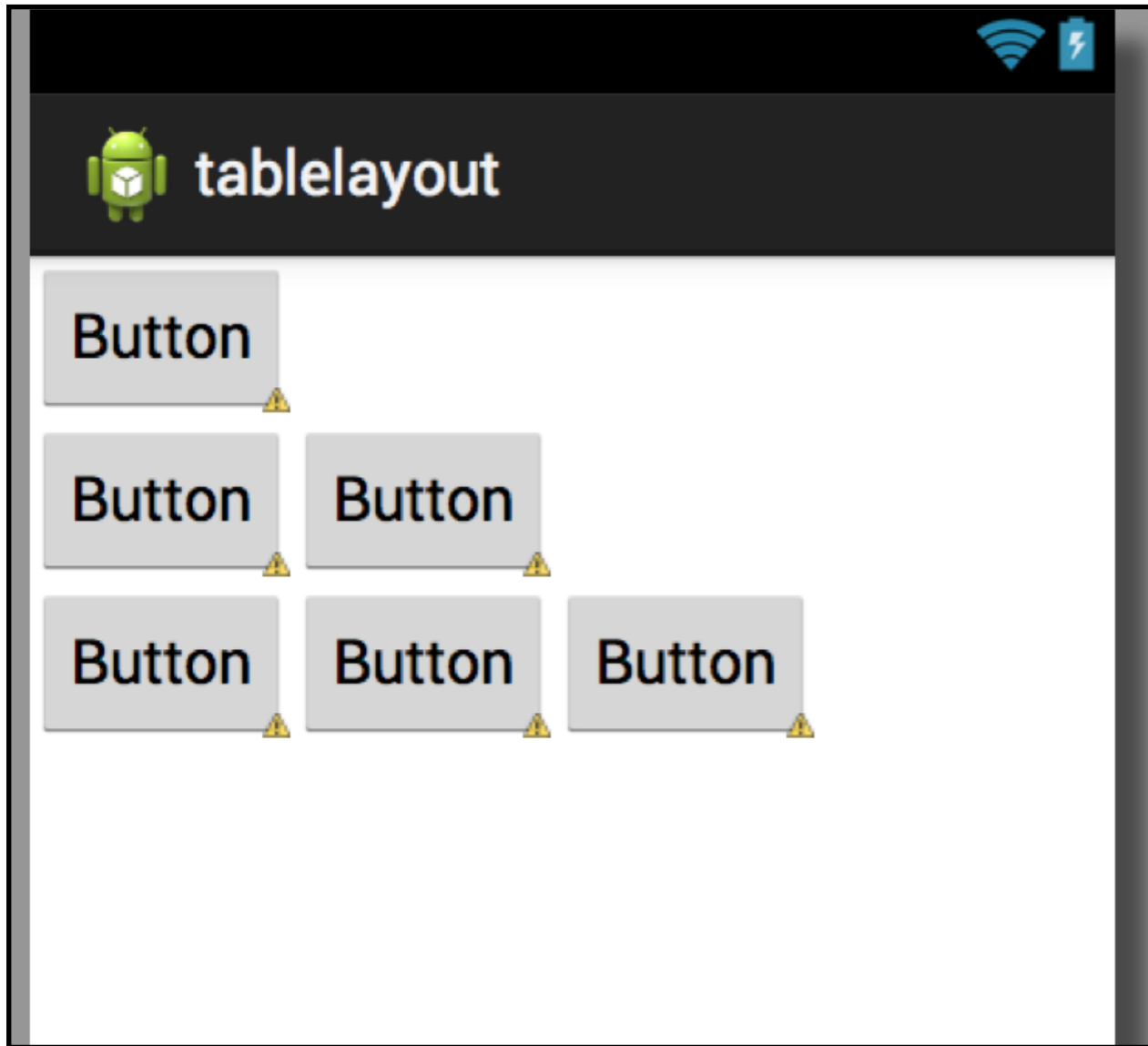
```
1 <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:id="@+id/TableLayout1"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:paddingBottom="@dimen/activity_vertical_margin"
7   android:paddingLeft="@dimen/activity_horizontal_margin"
8   android:paddingRight="@dimen/activity_horizontal_margin"
9   android:paddingTop="@dimen/activity_vertical_margin"
10  tools:context="com.example.tablelayout.MainActivity" >
11
12 </TableLayout>
13
```

Delete!

TableLayout的元素TableRow (其內的元素才可視為表格元素)

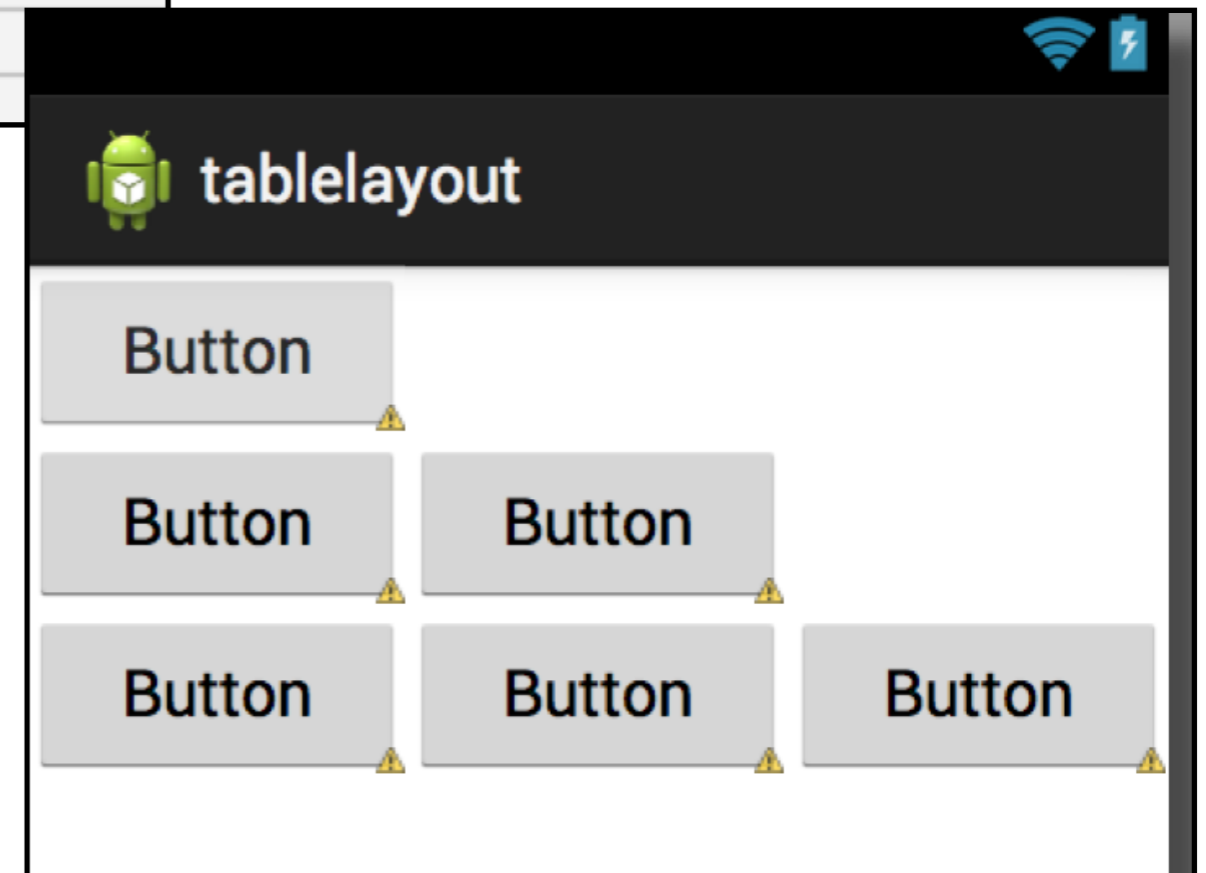


加入如下的排列

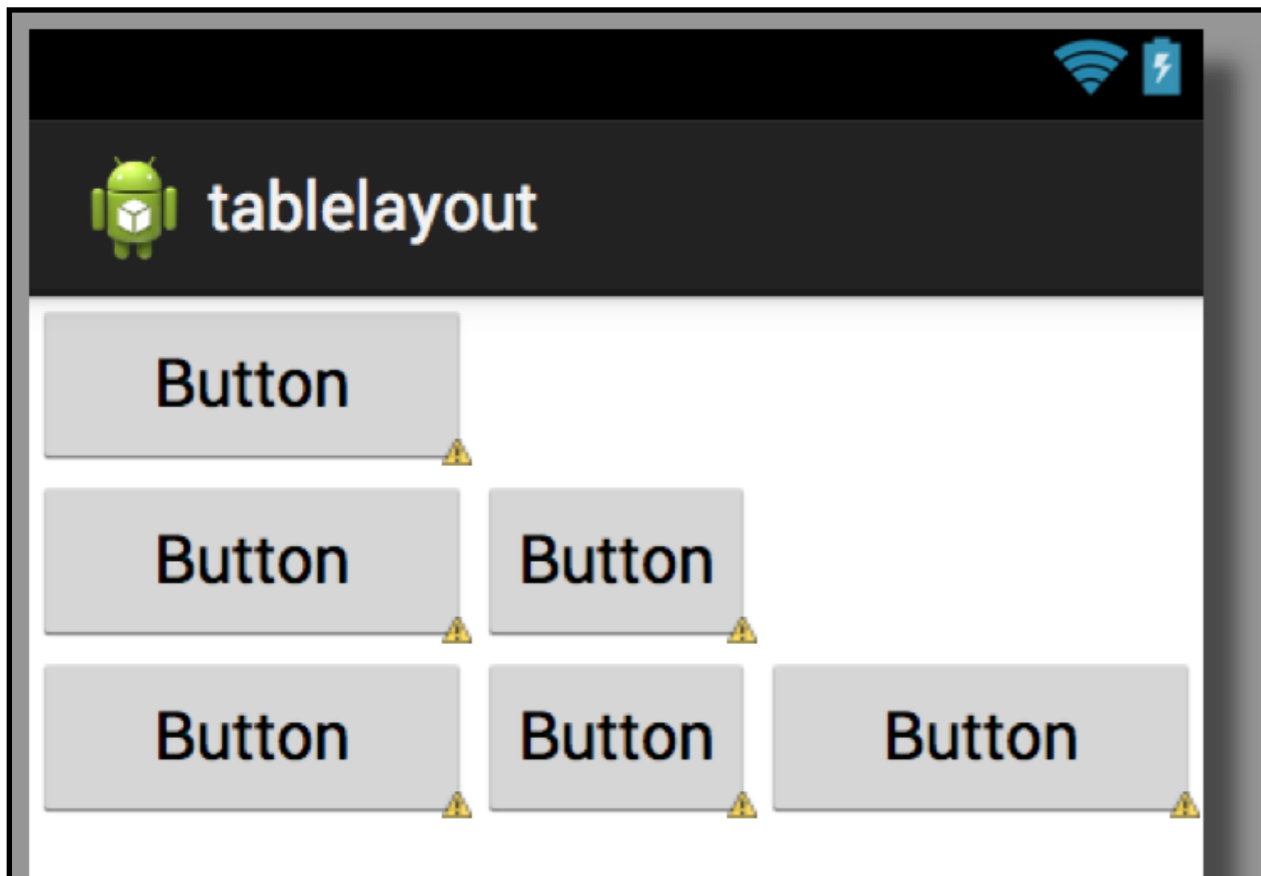


設定TableLayout的可延展屬性Stretch Columns="*"，平均分配欄位寬度

Properties	
Layout Parameters	
Gravity	
Width	match_parent
Height	match_parent
+ Margins	
Stretch Columns	*
Shrink Columns	
Orientation	
Content Description	
Gravity	

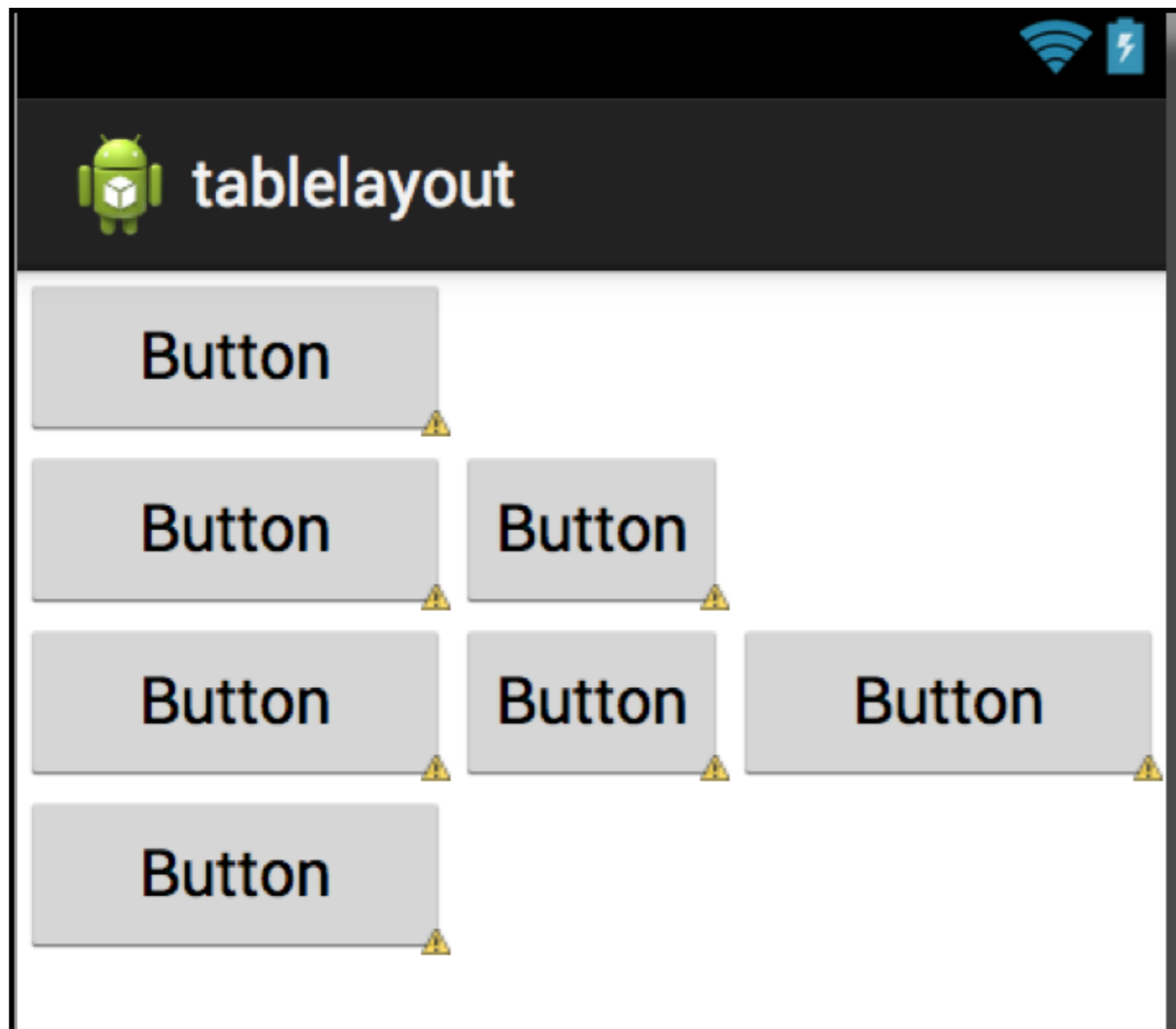


Stretch Columns="0,2"的結果



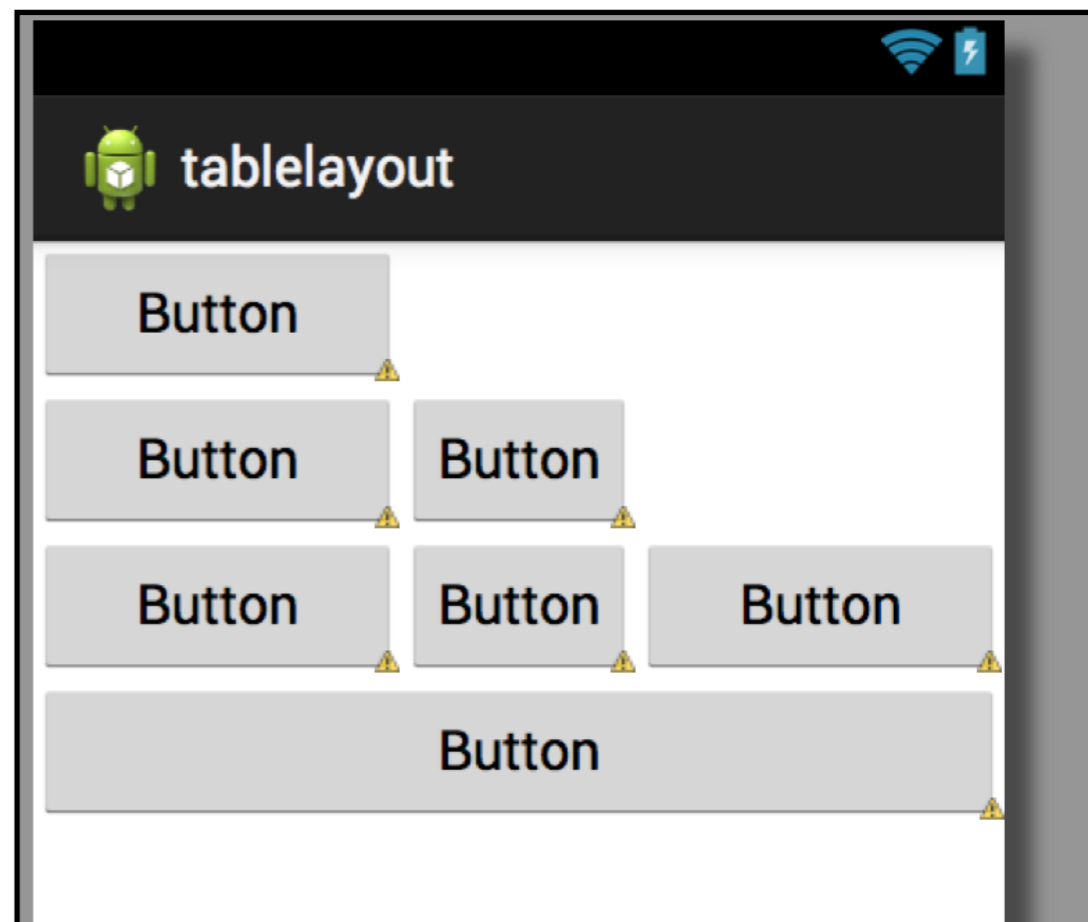
Properties	
Id	@+id/TableLayout1
Layout Parameters	[]
Gravity	
Width	match_parent
Height	match_parent
Margins	[]
Stretch Columns	0,2
Shrink Columns	
Orientation	
Content Description	
Gravity	

多加一個TableRow，並加入一個Button



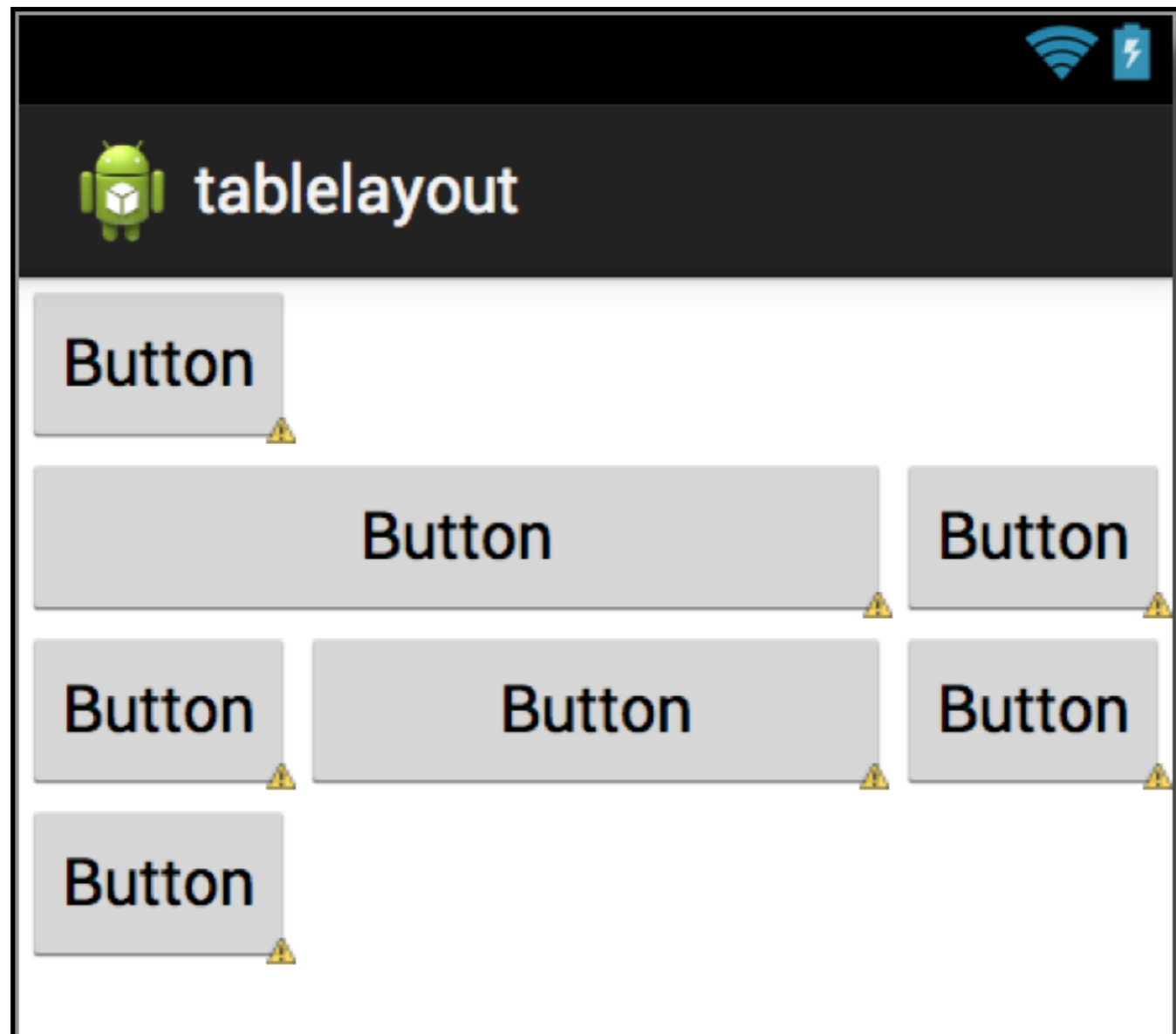
設定Button的layout_span="3"

```
</TableRow>
<TableRow
  android:id="@+id/tableRow4"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content" >
  <Button
    android:id="@+id/button7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_span="3"
    android:text="Button" />
</TableRow>
```



(顯示結果)

Exercise: 請完成以下Layout



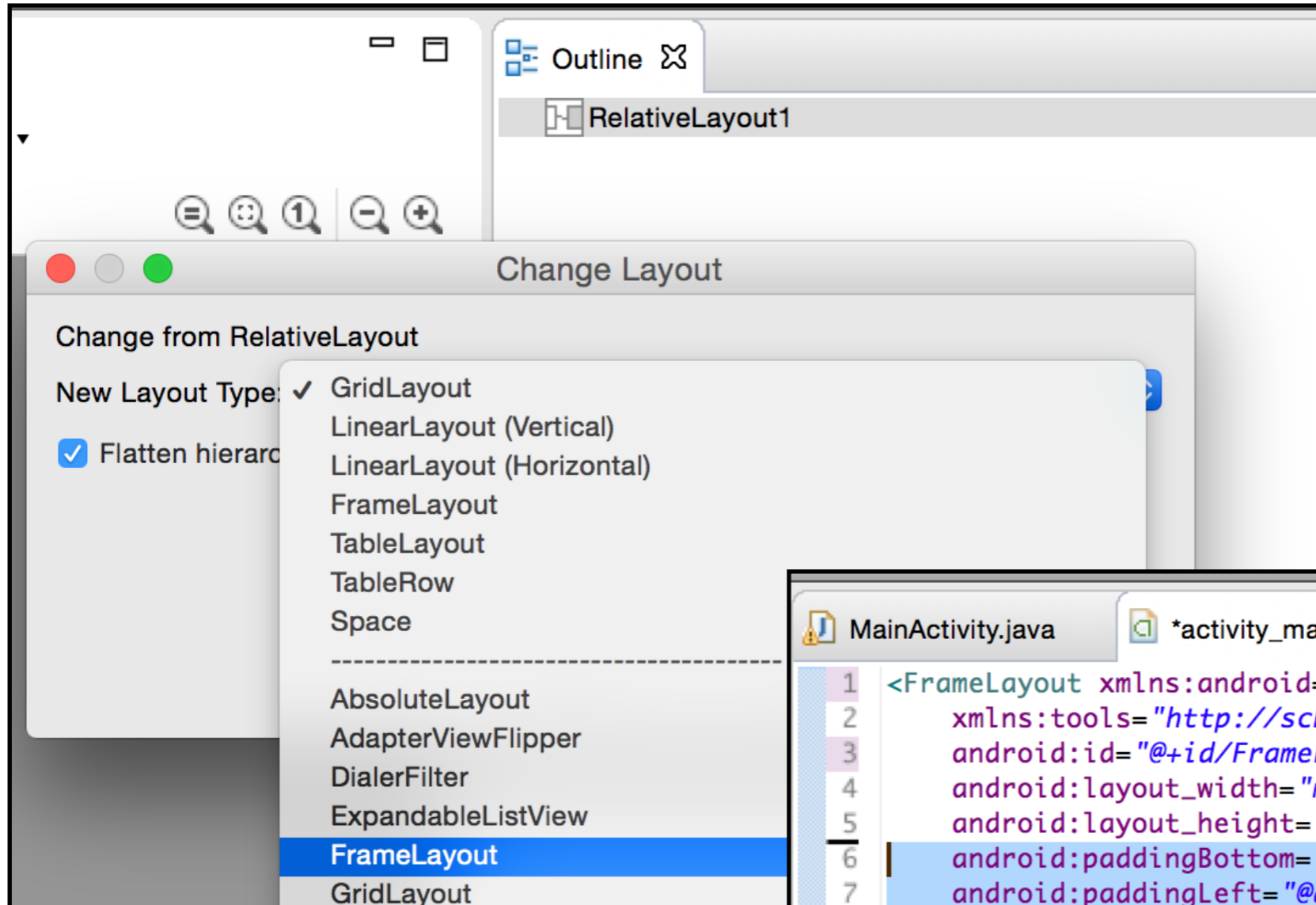
FrameLayout

使用時機

- 畫面需要重疊顯示時（例如使用說明）
- 不同時間出現不同內容（部分隱藏）

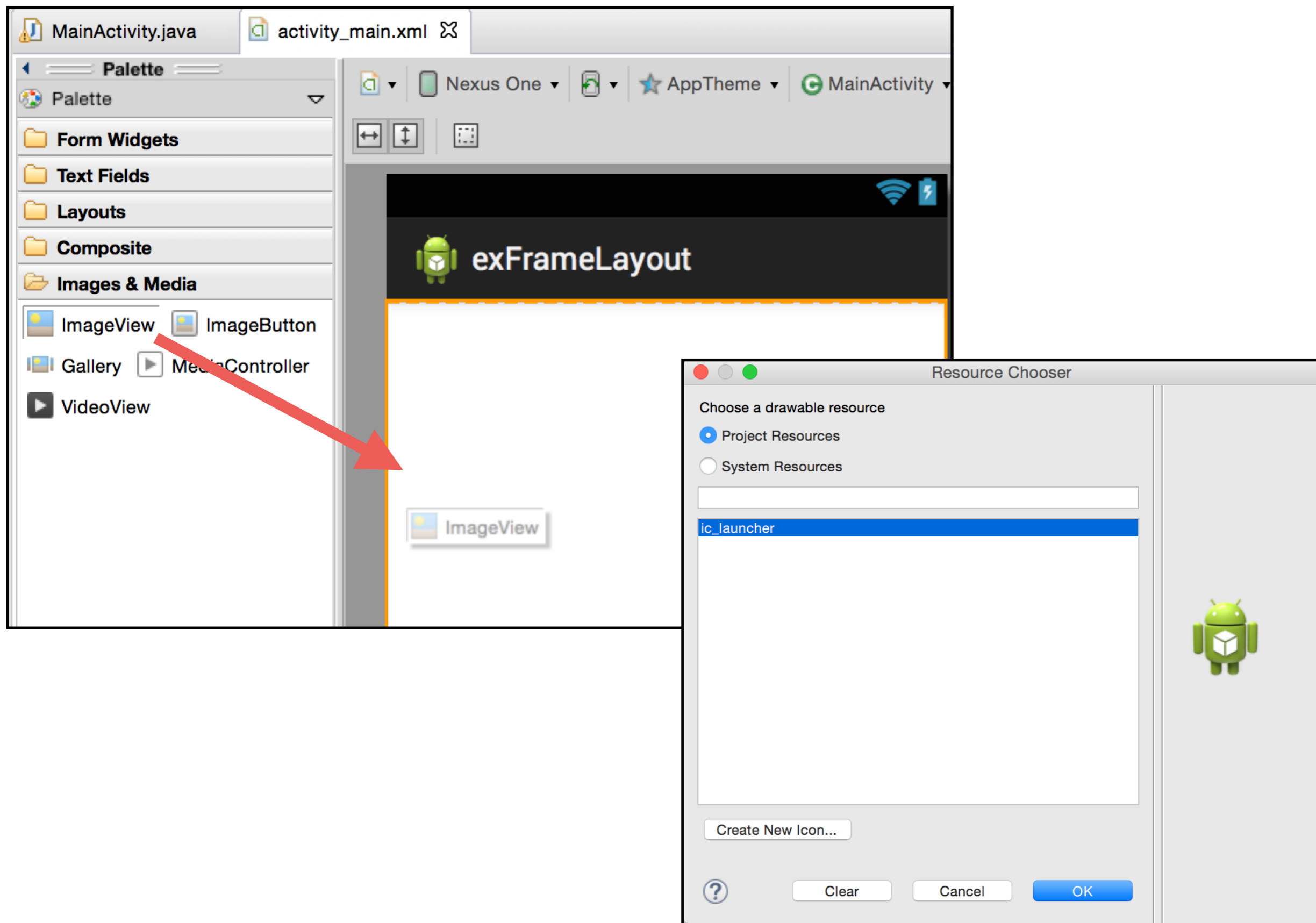
新增專案exFrameLayout

- RelativeLayout > FrameLayout > 刪除邊界設定



```
MainActivity.java  *activity_main.xml
1 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:id="@+id/FrameLayout1"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:paddingBottom="@dimen/activity_vertical_margin"
7   android:paddingLeft="@dimen/activity_horizontal_margin"
8   android:paddingRight="@dimen/activity_horizontal_margin"
9   android:paddingTop="@dimen/activity_vertical_margin"
10  tools:context="com.example.exframelayout.MainActivity" >
11
12 </FrameLayout>
13
```

新增ImageView > 選擇ic_launcher圖片

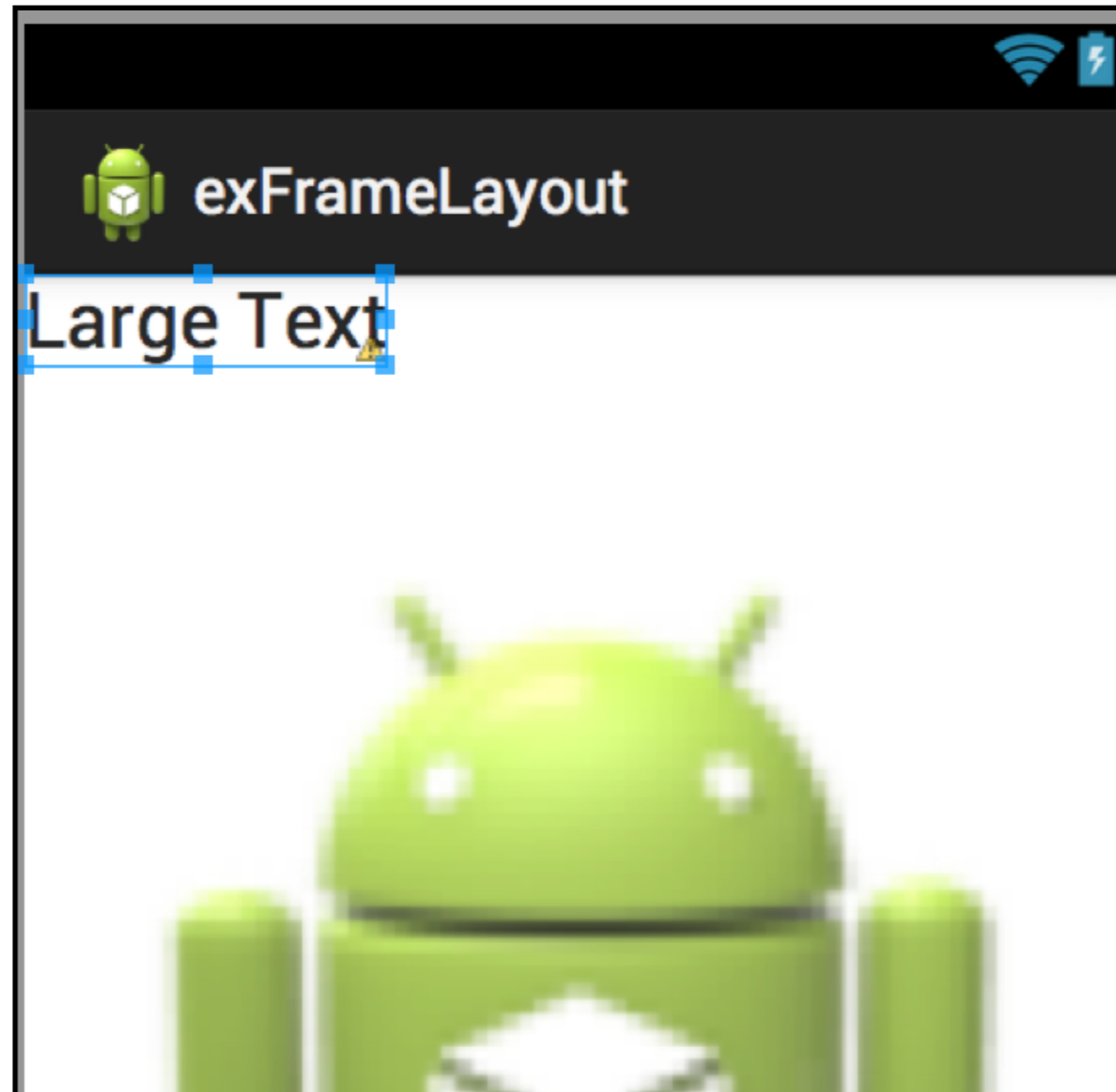


Width=fill_parent, Height=fill_parent



Properties	
Id	@+id/imageView1
Layout Parameters	
Gravity	
Width	fill_parent
Height	fill_parent
...	...

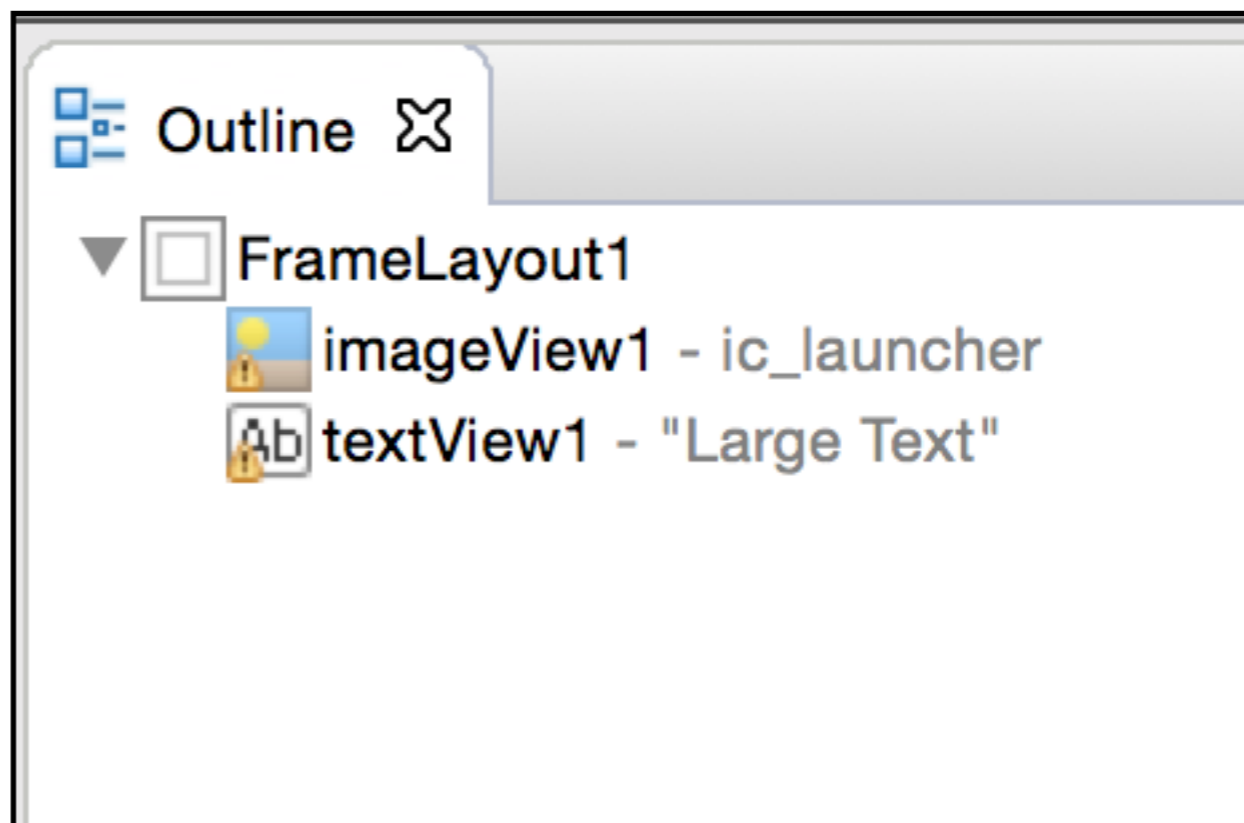
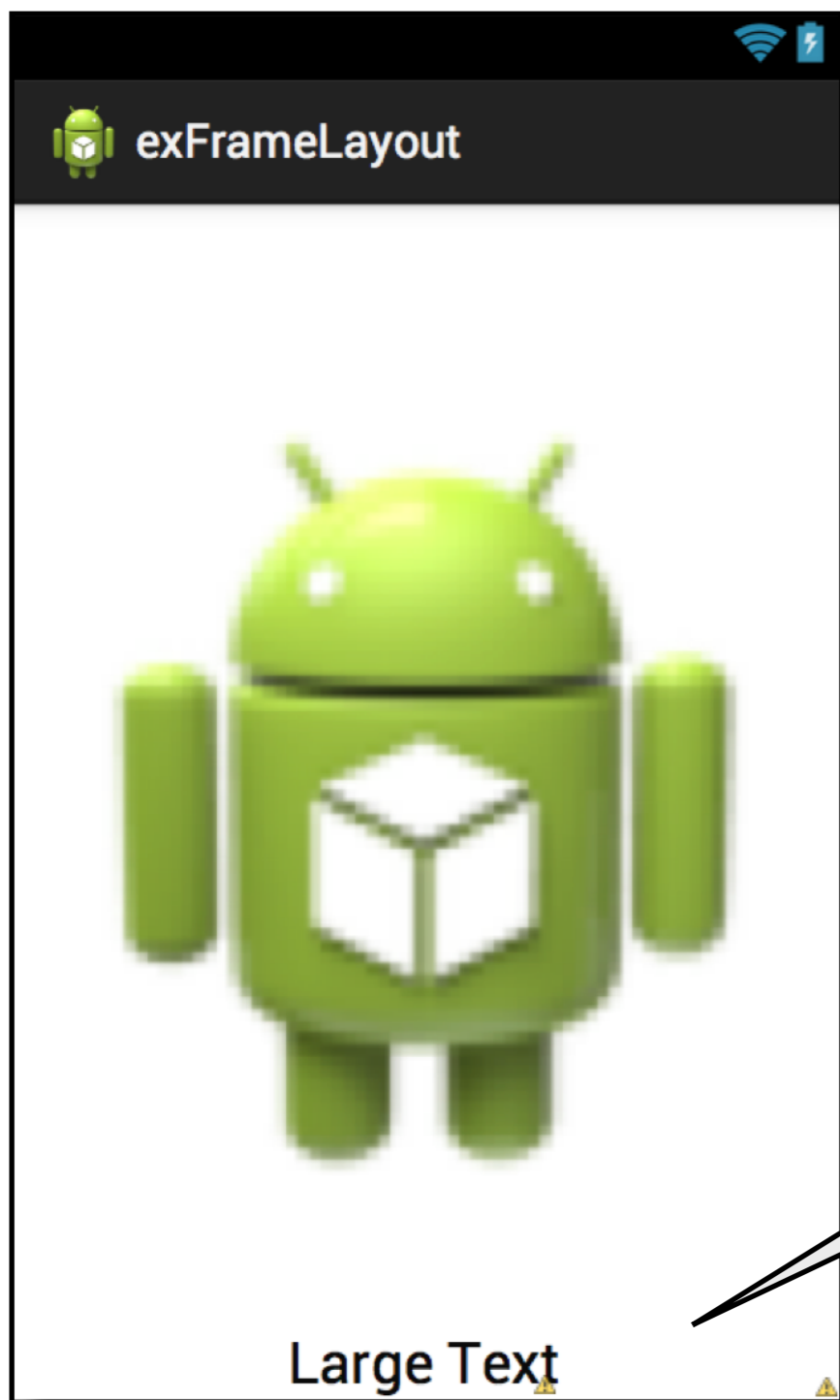
加入文字，會顯示在左上方



設定TextView的Layout Parameters > Gravity > bottom|center

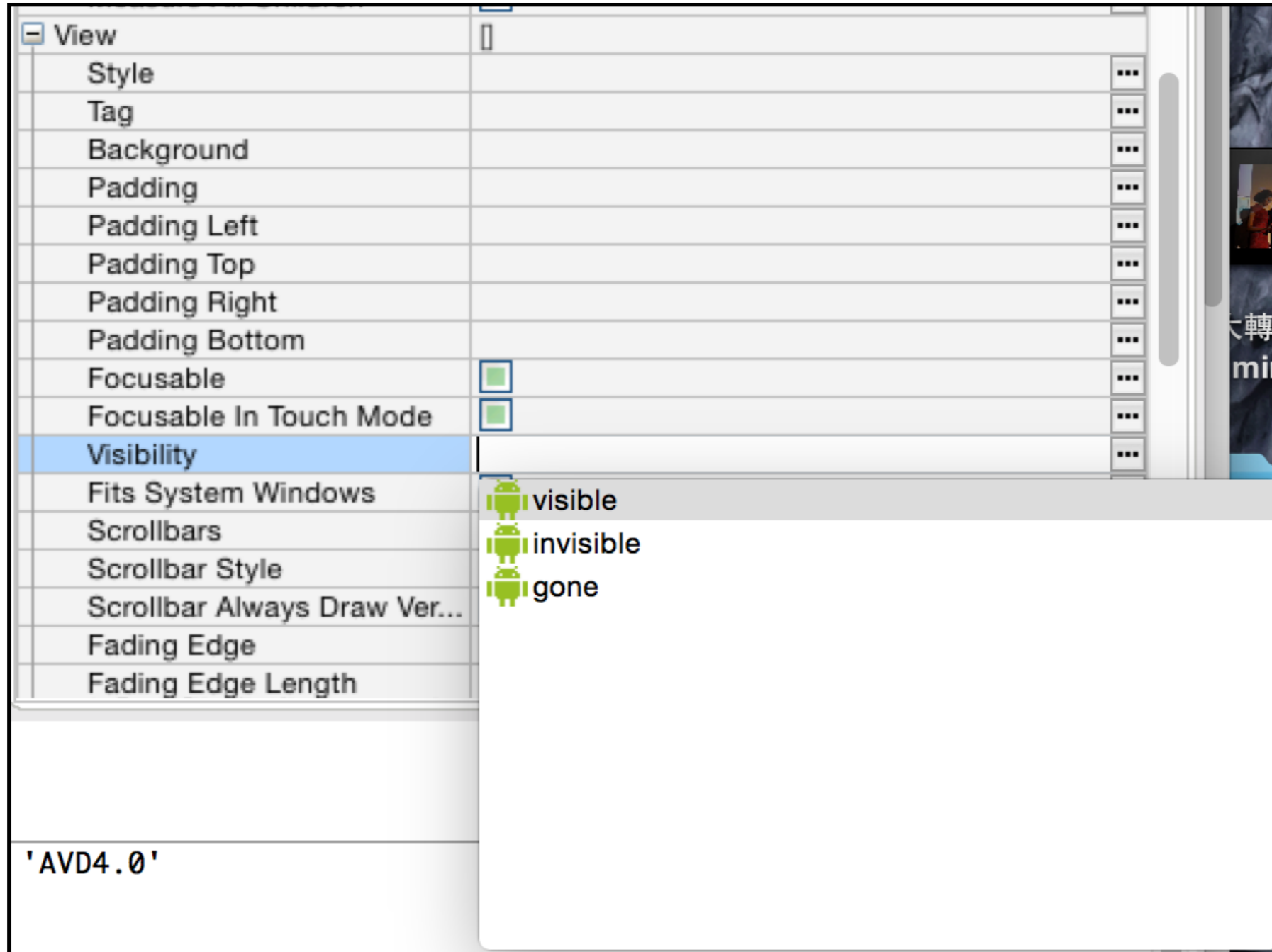


顯示結果



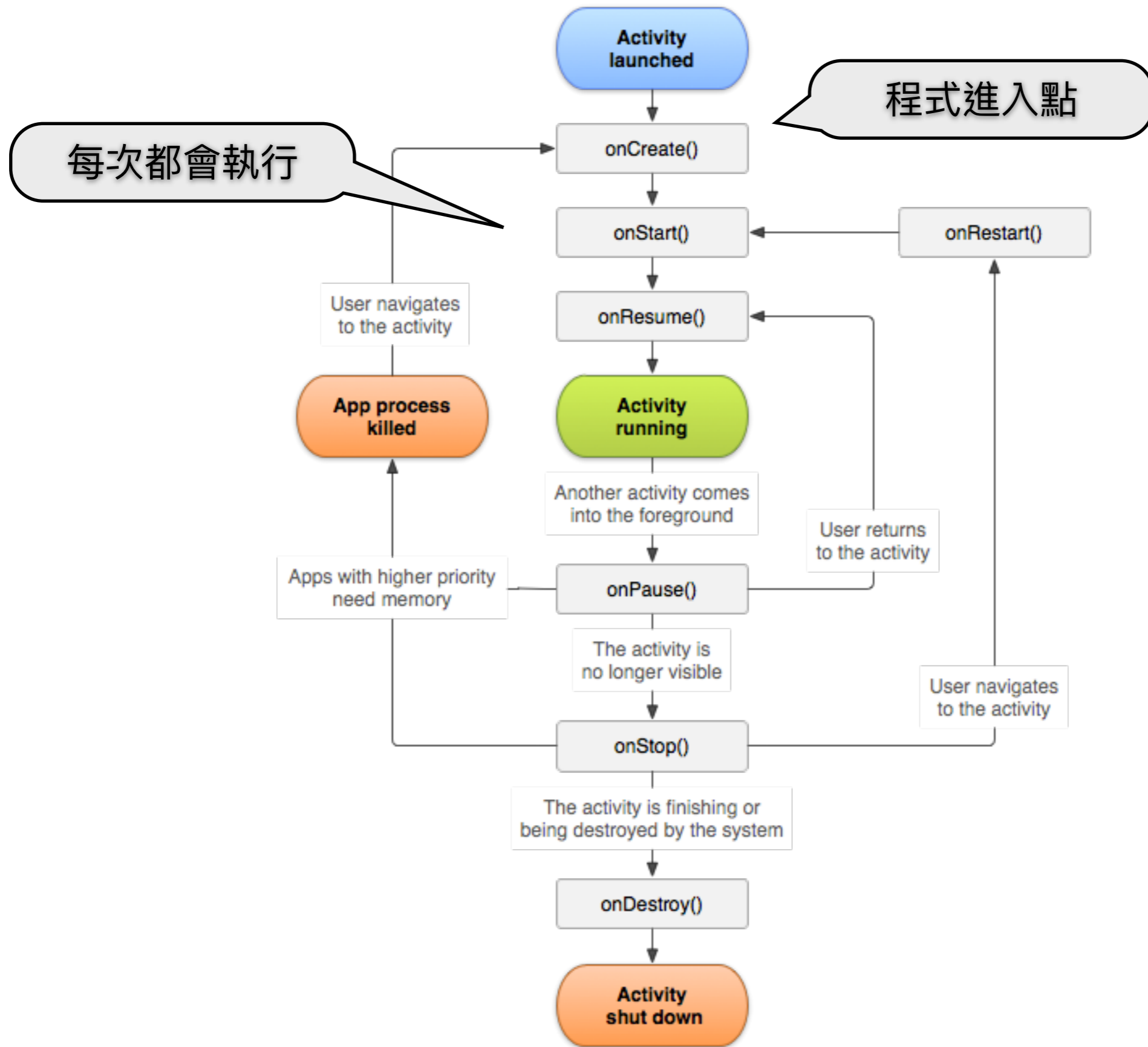
文字疊在圖片上面

View屬性有Visibility，可設定隱藏或顯示

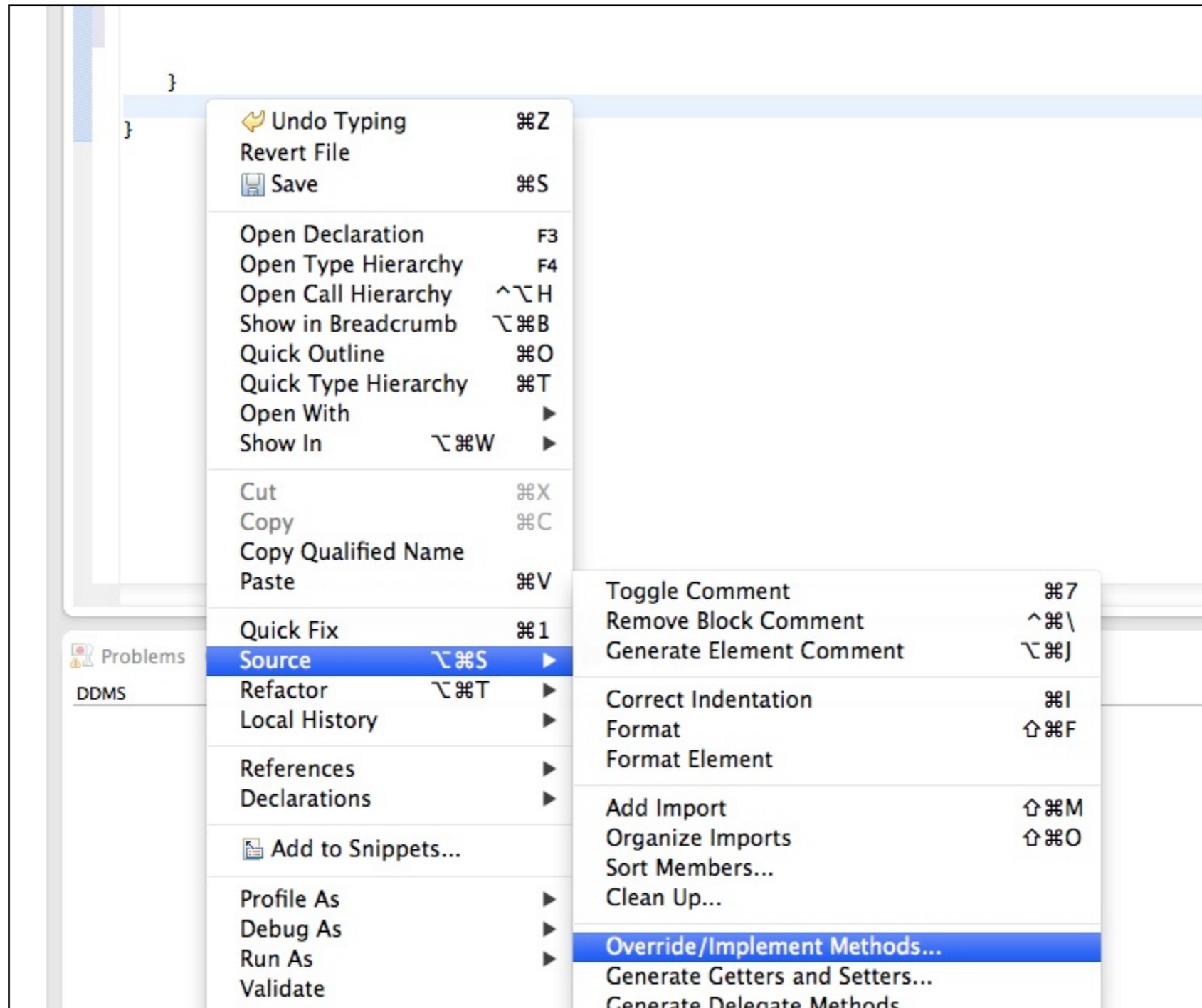


Activity 探討

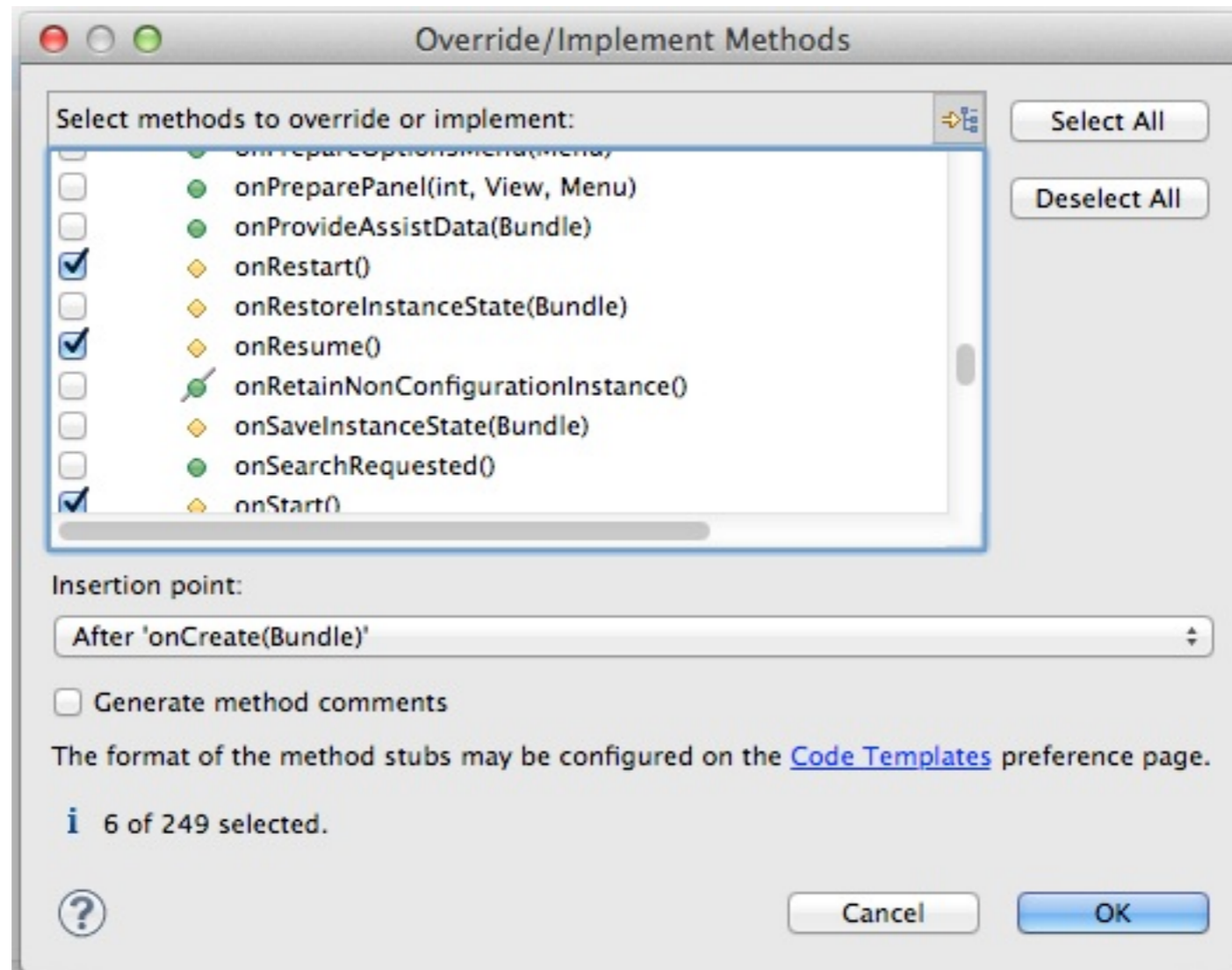
- Activity 的生命週期
- Intent
- Activity 的生命週期實例
- Activity 之間變數的傳遞(透過Intent)



在各工作階段接入Log觀察



onCreate 、 onStart 、 onResume 、 onPause 、 onStop 、
onDestroy



(透過界面直接列出想要的function)

```
*MainActivity.java  fragment_main.xml
@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();
}

@Override
protected void onPause() {
    // TODO Auto-generated method stub
    super.onPause();
}

@Override
protected void onRestart() {
    // TODO Auto-generated method stub
    super.onRestart();
}

@Override
protected void onResume() {
    // TODO Auto-generated method stub
    super.onResume();
}

@Override
protected void onStart() {
    // TODO Auto-generated method stub
    super.onStart();
}

@Override
protected void onStop() {
    // TODO Auto-generated method stub
    super.onStop();
}
}
```

```
@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();
    Log.i("test", "onDestroy");
}

@Override
protected void onPause() {
    // TODO Auto-generated method stub
    super.onPause();
    Log.i("test", "onPause");
}

@Override
protected void onRestart() {
    // TODO Auto-generated method stub
    super.onRestart();
    Log.i("test", "onRestart");
}

@Override
protected void onResume() {
    // TODO Auto-generated method stub
    super.onResume();
    Log.i("test", "onResume");
}
```

Log.i(tag,msg);

(加入tag="test"以及要顯示的訊息)

顯示自訂Tag的訊息

The screenshot shows the 'Logcat Message Filter Settings' dialog box on the left. The 'Filter Name' is 'testActivity'. Under 'by Log Tag', the value 'test' is entered. The 'by Log Level' is set to 'verbose'. A 'Cancel' button is at the bottom right. On the right, the LogCat view shows a 'Saved Filters' list with 'testActivity' selected. The main LogCat area has a search bar and a table with columns 'Lev', 'Time', 'PID', and 'TID'.

The screenshot shows the LogCat view with the filter 'testActivity' applied. The 'Saved Filters' list on the left shows 'testActivity (6)' and 'com.example.activitytest (Session Fil...'. The main LogCat area has a search bar and a table with columns 'Lev', 'Time', 'PID', 'TID', 'Application', 'Tag', and 'Text'. The table contains six rows of log messages, all with level 'I' and tag 'test'.

Lev	Time	PID	TID	Application	Tag	Text
I	06-06 14:21:49.626	5026	5026	com.example.activ...	test	onRestart
I	06-06 14:21:49.626	5026	5026	com.example.activ...	test	onStart
I	06-06 14:21:49.641	5026	5026	com.example.activ...	test	onResume
I	06-06 14:21:55.996	5026	5026	com.example.activ...	test	onPause
I	06-06 14:21:56.356	5026	5026	com.example.activ...	test	onStop
I	06-06 14:21:56.356	5026	5026	com.example.activ...	test	onDestroy

(Activity從開始到結束)

Exercise

- 練習追蹤Activity的生命週期

由Activity到Activity



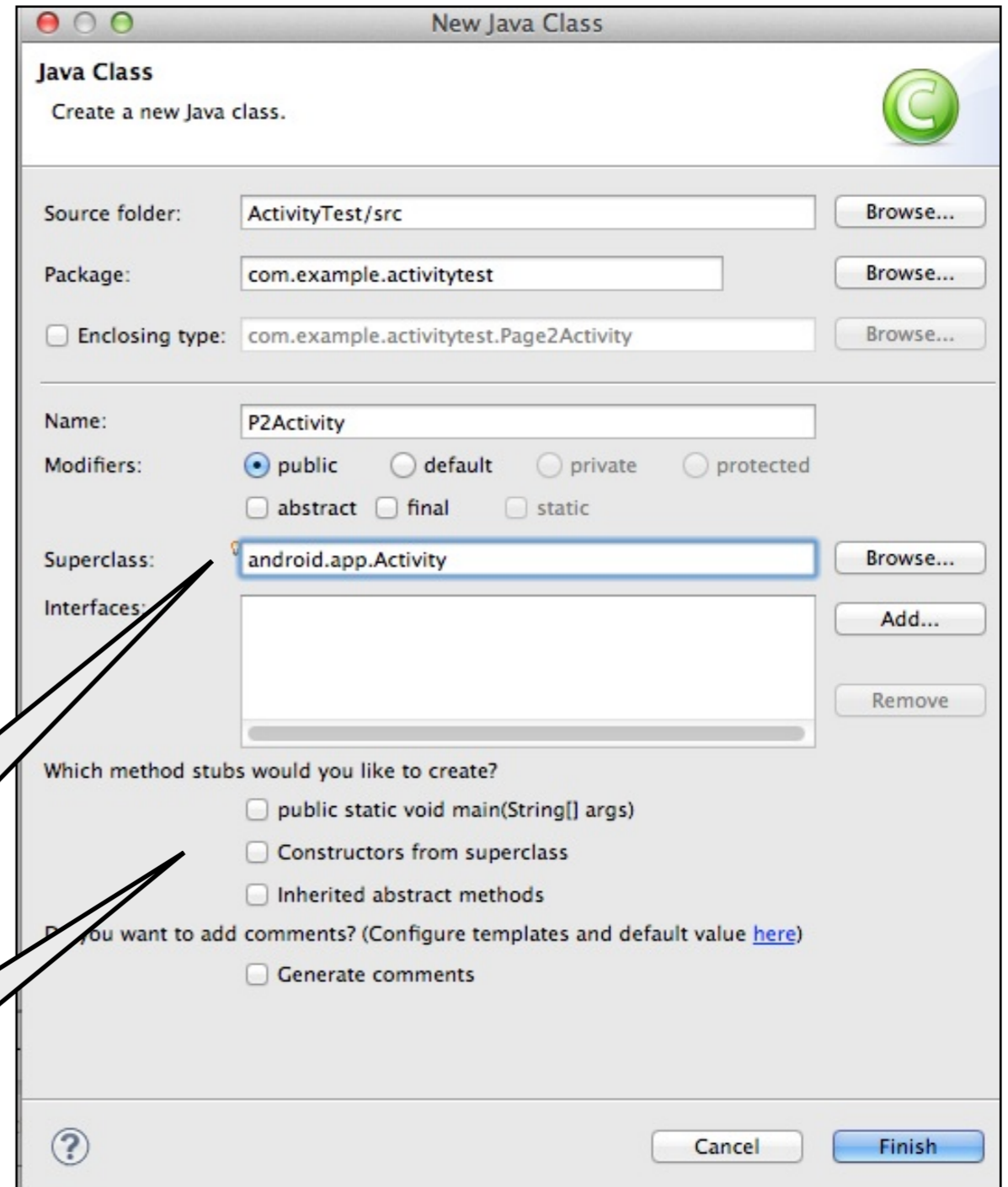
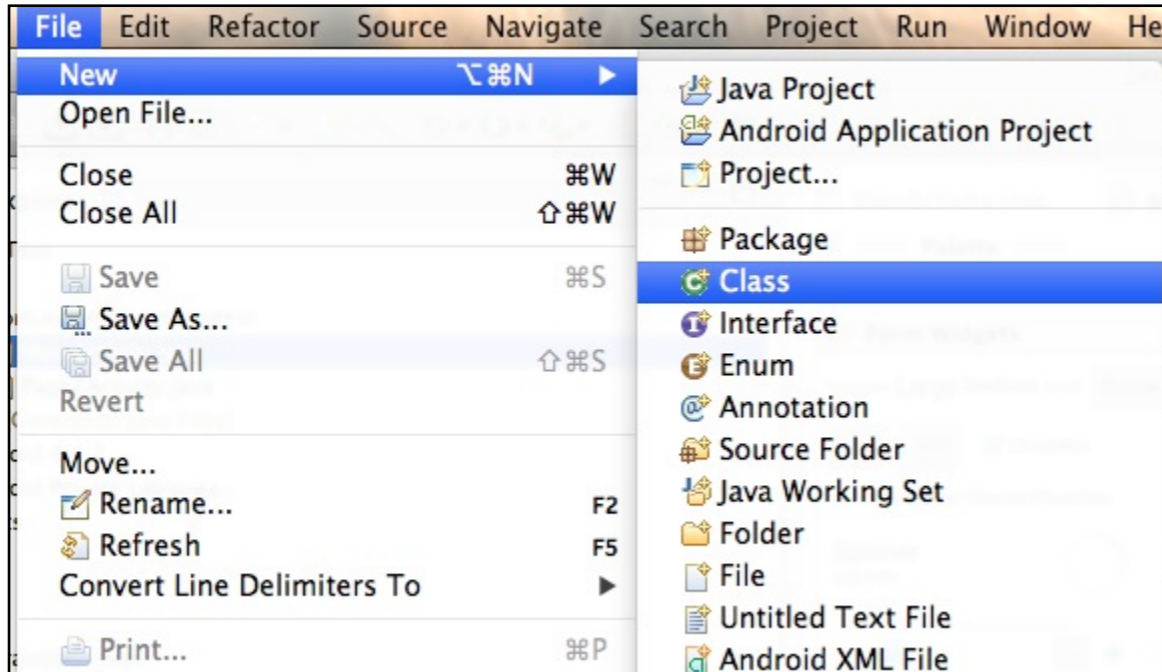
id為button1

A screenshot of the Android Studio Properties panel for a button widget. The panel shows various properties and their values. A callout bubble points to the 'Id' property.

Properties	
Id	@+id/button1
Layout Parameters	{}
Style	android:buttonStyle
Text	Next Activity
Hint	
Content Description	
TextView	{}
Text	Next Activity
Hint	
Text Color	@android:color/primary_text_holo_light
Text Color Hint	@android:color/hint_foreground_holo_li...
Text Appearance	?android:attr/textAppearanceMediumInverse
Text Size	
Typeface	
Text Style	
Font Family	
Text Color Link	@android:color/holo_blue_light
Max Lines	

建立另一個Activity P2Activity.java

- File > New > Class



繼承自Activity

都不要勾選

在MainActivity加入Button相關的動作

```
private Button nextBtn;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    Log.i("test", "onCreate");  
    nextBtn=(Button)findViewById(R.id.button1);  
    nextBtn.setOnClickListener(btnListener);  
  
}  
OnClickListener btnListener= new OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        Intent nextIntent=new Intent(getApplicationContext(), P2Activity.class);  
        startActivity(nextIntent);  
    }  
};
```

執行

Context

要被執行的class

AndroidManifest.xml加入新的Activity才可識別

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.activitytest"
    android:versionCode="1"
    android:versionName="1.0" >

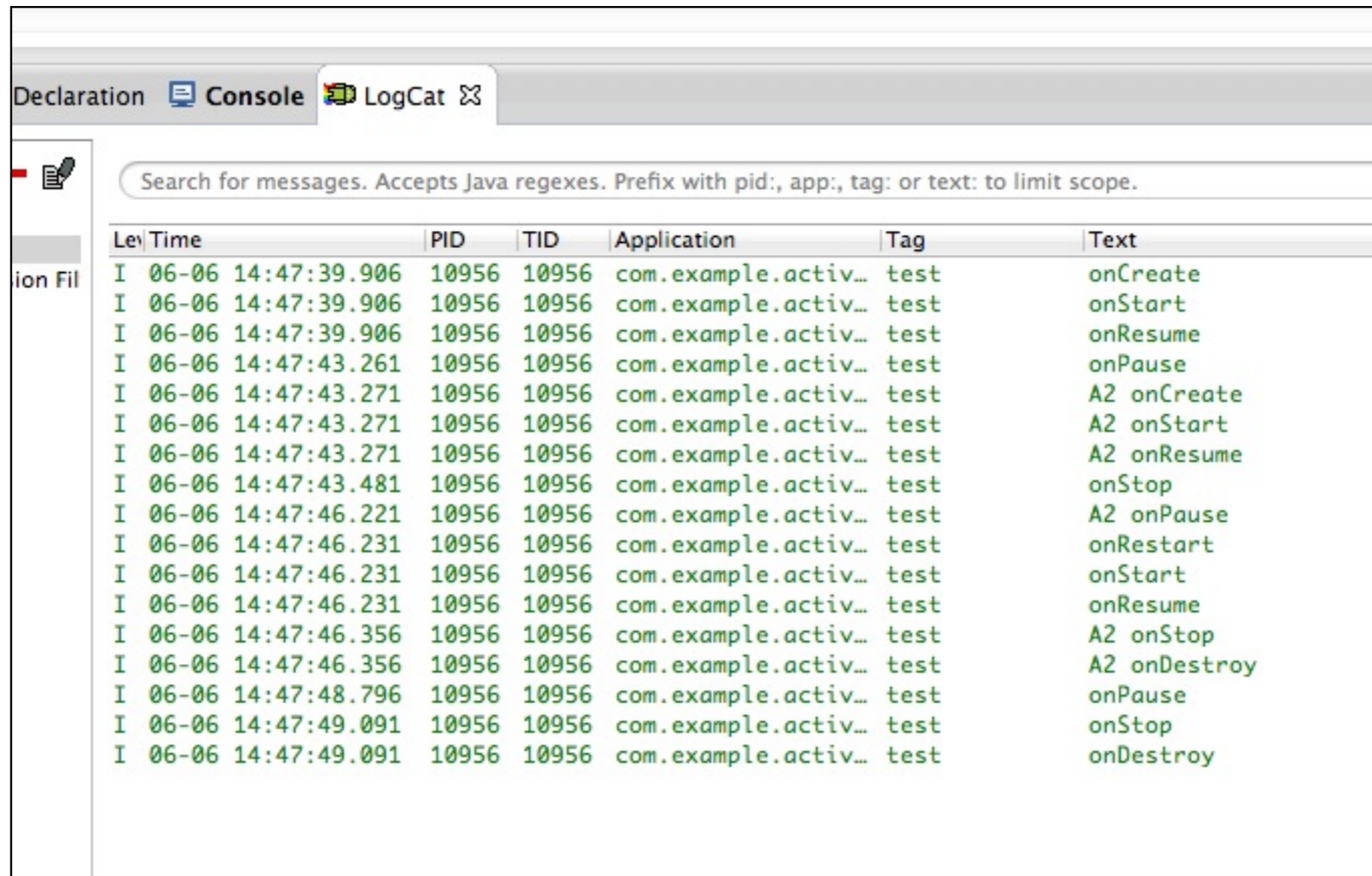
    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="19" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.activitytest.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <activity android:name="com.example.activitytest.P2Activity"> </activity>
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        </application>

</manifest>
```

執行結果



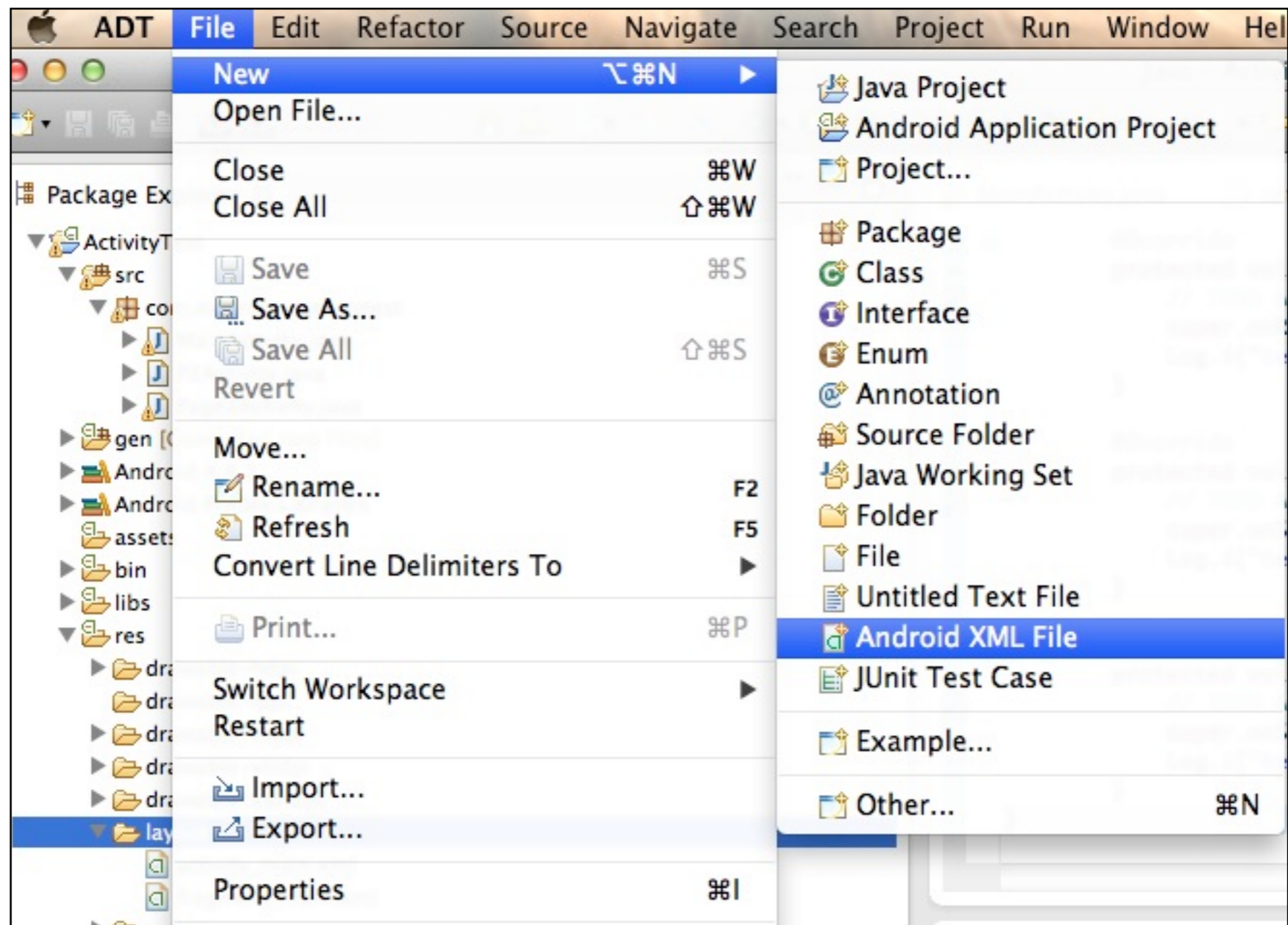
The screenshot shows the LogCat window in an IDE. The window has tabs for 'Declaration', 'Console', and 'LogCat'. Below the tabs is a search bar with the text: 'Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope.' The main area displays a list of log messages in a table format. The messages are all level 'I' (Info) and are from the application 'com.example.activ...'. The messages show the lifecycle of two activities: 'test' and 'A2'. The 'test' activity goes through onCreate, onStart, onResume, onPause, onStop, and onDestroy. The 'A2' activity goes through onCreate, onStart, onResume, onPause, and onDestroy.

Level	Time	PID	TID	Application	Tag	Text
I	06-06 14:47:39.906	10956	10956	com.example.activ...	test	onCreate
I	06-06 14:47:39.906	10956	10956	com.example.activ...	test	onStart
I	06-06 14:47:39.906	10956	10956	com.example.activ...	test	onResume
I	06-06 14:47:43.261	10956	10956	com.example.activ...	test	onPause
I	06-06 14:47:43.271	10956	10956	com.example.activ...	test	A2 onCreate
I	06-06 14:47:43.271	10956	10956	com.example.activ...	test	A2 onStart
I	06-06 14:47:43.271	10956	10956	com.example.activ...	test	A2 onResume
I	06-06 14:47:43.481	10956	10956	com.example.activ...	test	onStop
I	06-06 14:47:46.221	10956	10956	com.example.activ...	test	A2 onPause
I	06-06 14:47:46.231	10956	10956	com.example.activ...	test	onRestart
I	06-06 14:47:46.231	10956	10956	com.example.activ...	test	onStart
I	06-06 14:47:46.231	10956	10956	com.example.activ...	test	onResume
I	06-06 14:47:46.356	10956	10956	com.example.activ...	test	A2 onStop
I	06-06 14:47:46.356	10956	10956	com.example.activ...	test	A2 onDestroy
I	06-06 14:47:48.796	10956	10956	com.example.activ...	test	onPause
I	06-06 14:47:49.091	10956	10956	com.example.activ...	test	onStop
I	06-06 14:47:49.091	10956	10956	com.example.activ...	test	onDestroy

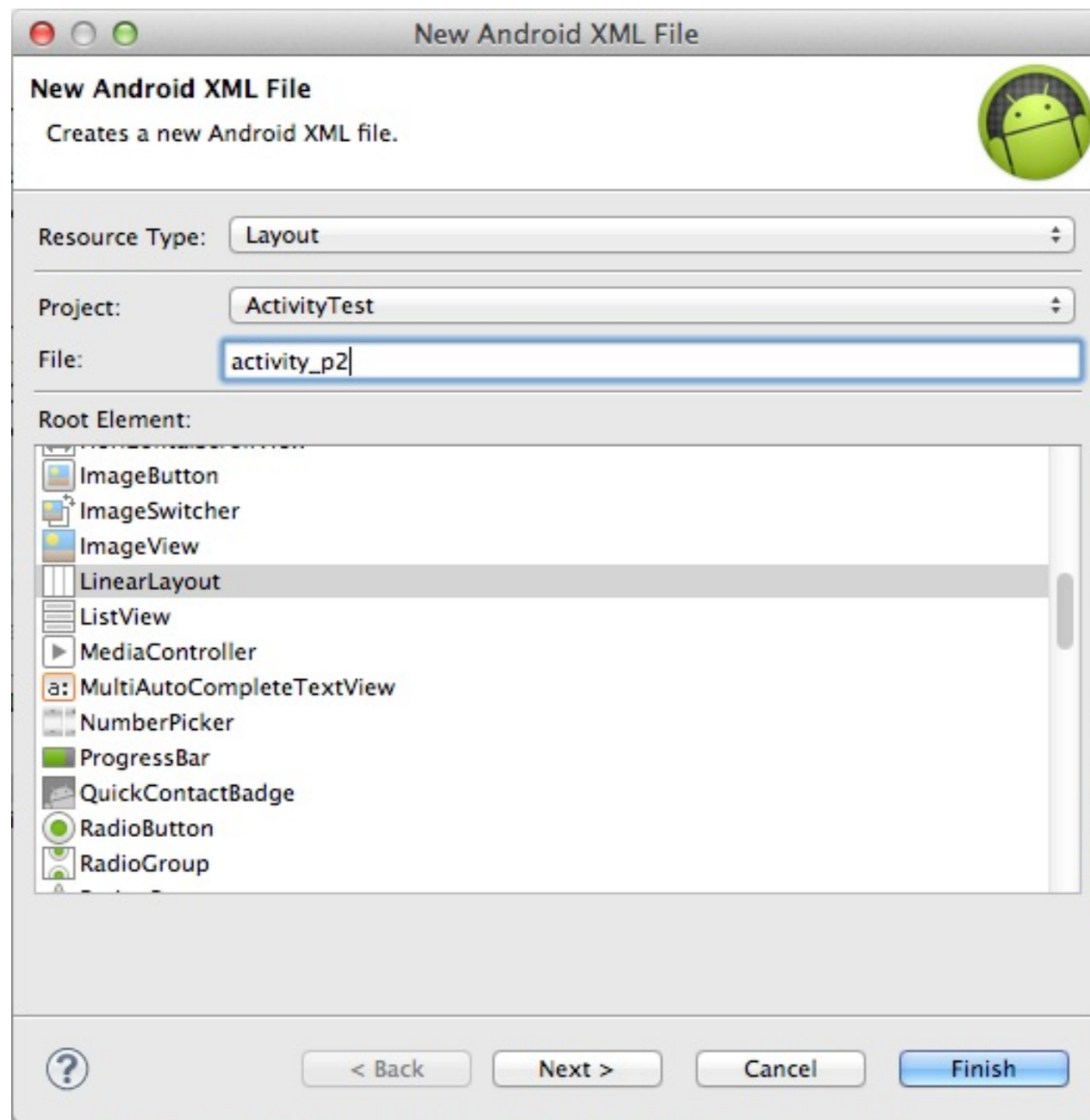
Exercise

- 練習追蹤多個Activity生命週期

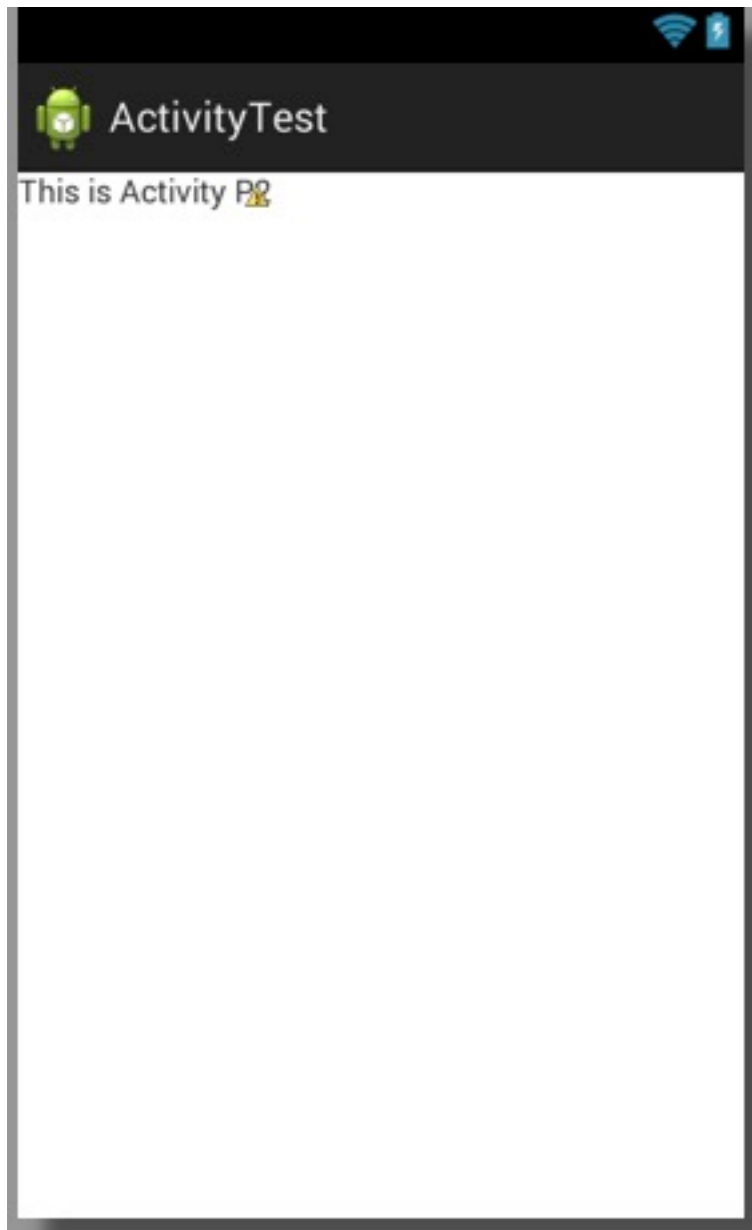
為P2Activity加入對應的顯示畫面(View)



選擇LinearLayout，指定檔名為activity_p2



為activity_p2加入一個TextView



```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    Log.i("test", "int layoutResID");
    setContentView(R.layout.);
}

@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();
    Log.i("test", "A2 onDest");
}

@Override
protected void onPause() {
    // TODO Auto-generated method stub
    super.onPause();
    Log.i("test", "A2 onPause");
}
```

activity_main : int - R.layout
activity_p2 : int - R.layout
fragment_main : int - R.layout
class : Class<com.example.activitytest.R.layout>
this

Press "\/" to show Template Proposals

在P2Activity.java
onCreate裡加入layout名稱為
R.layout.activity_p2

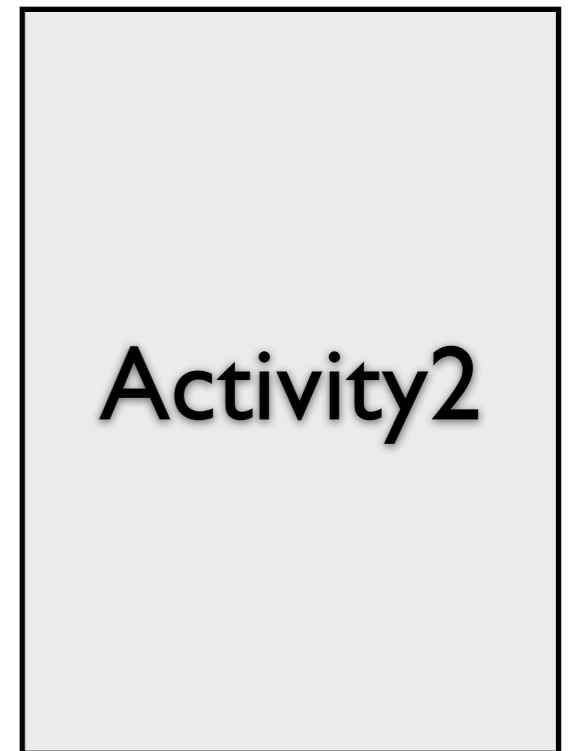


(執行結果)

Activity之間變數的傳遞

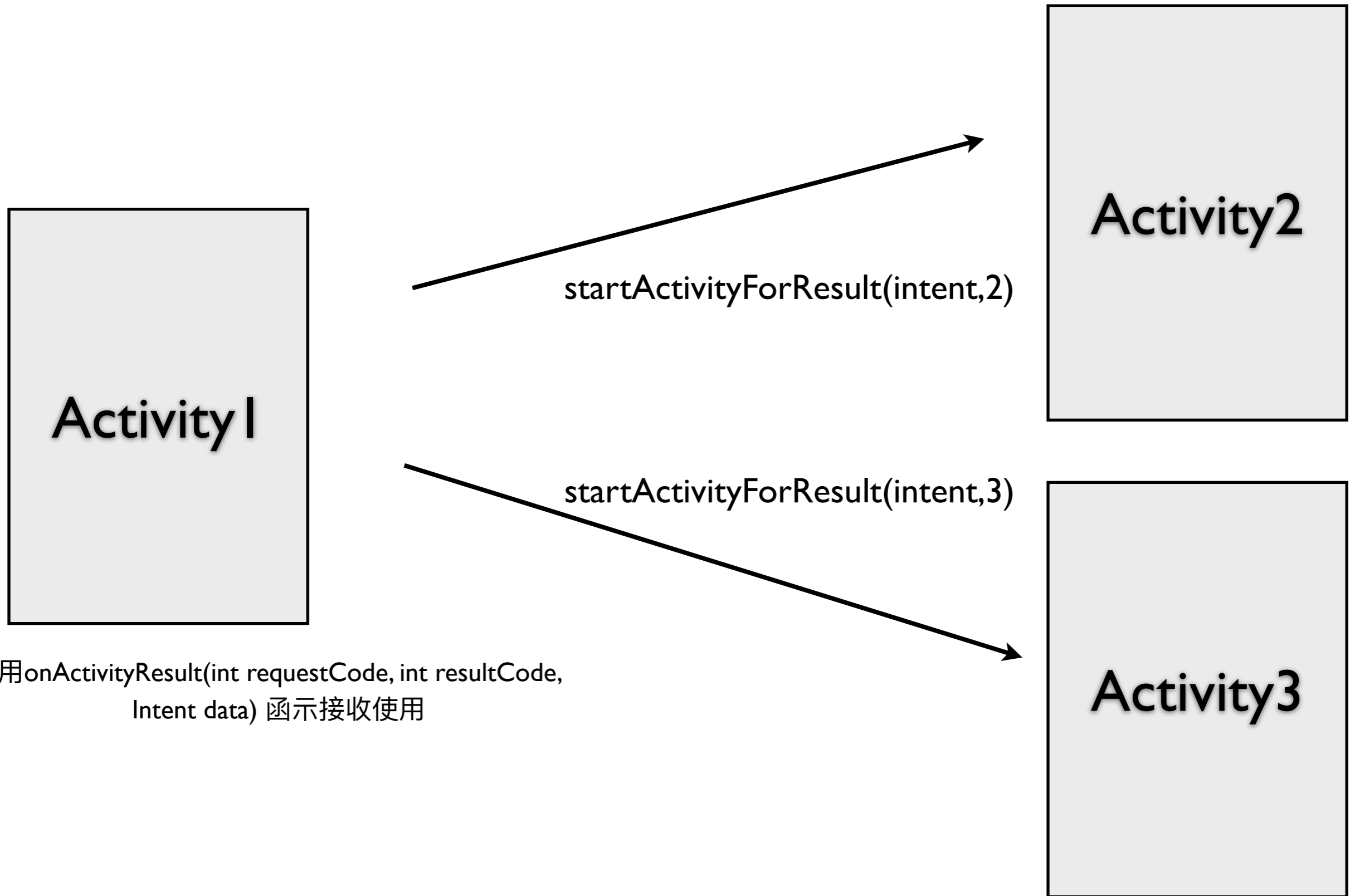


```
putExtra("key",value)  
startActivity(intent)
```



```
Intent it=getIntent()  
XXX var=it.getXXXExtra("key")
```

Activity之間變數的傳遞

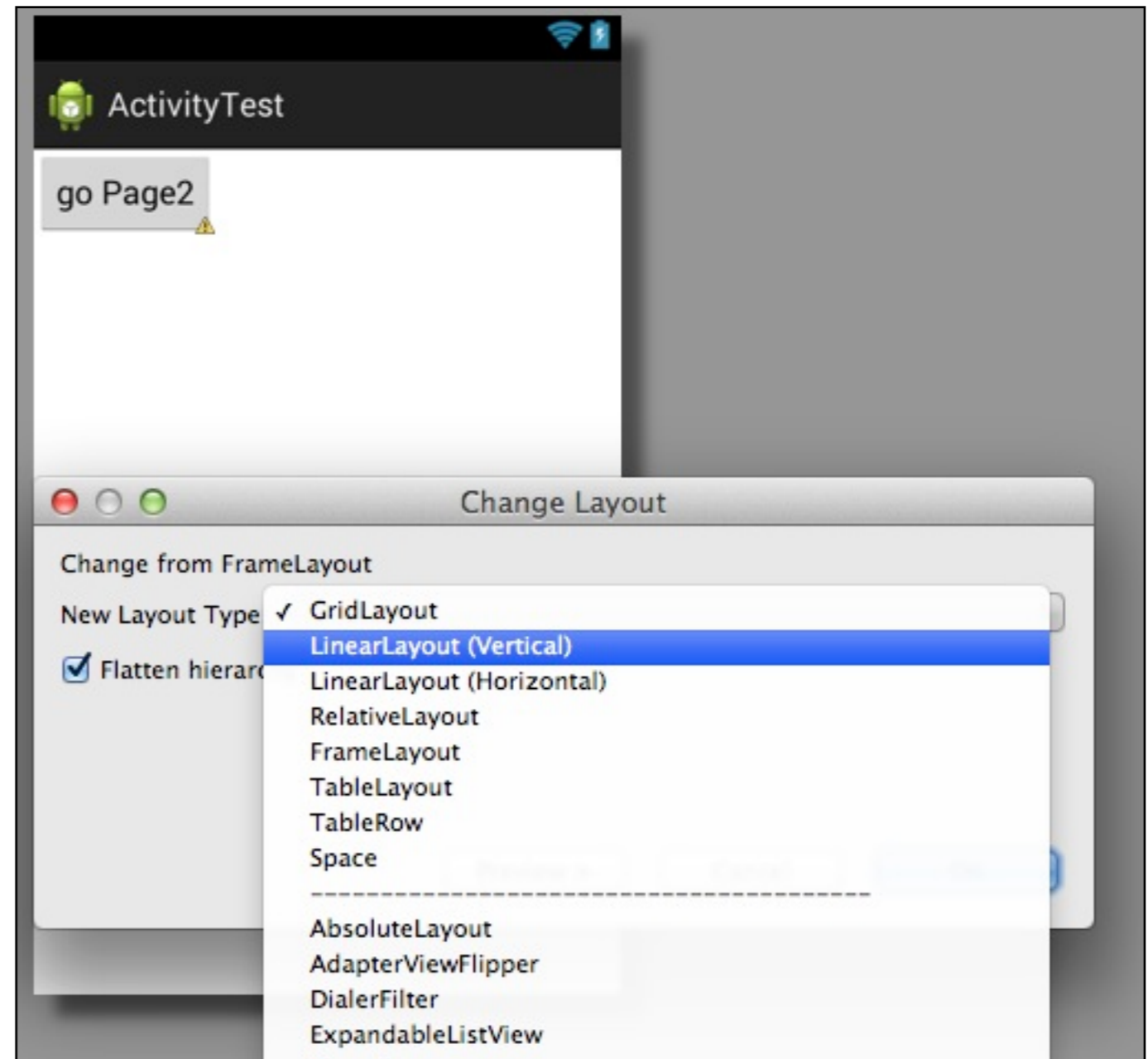
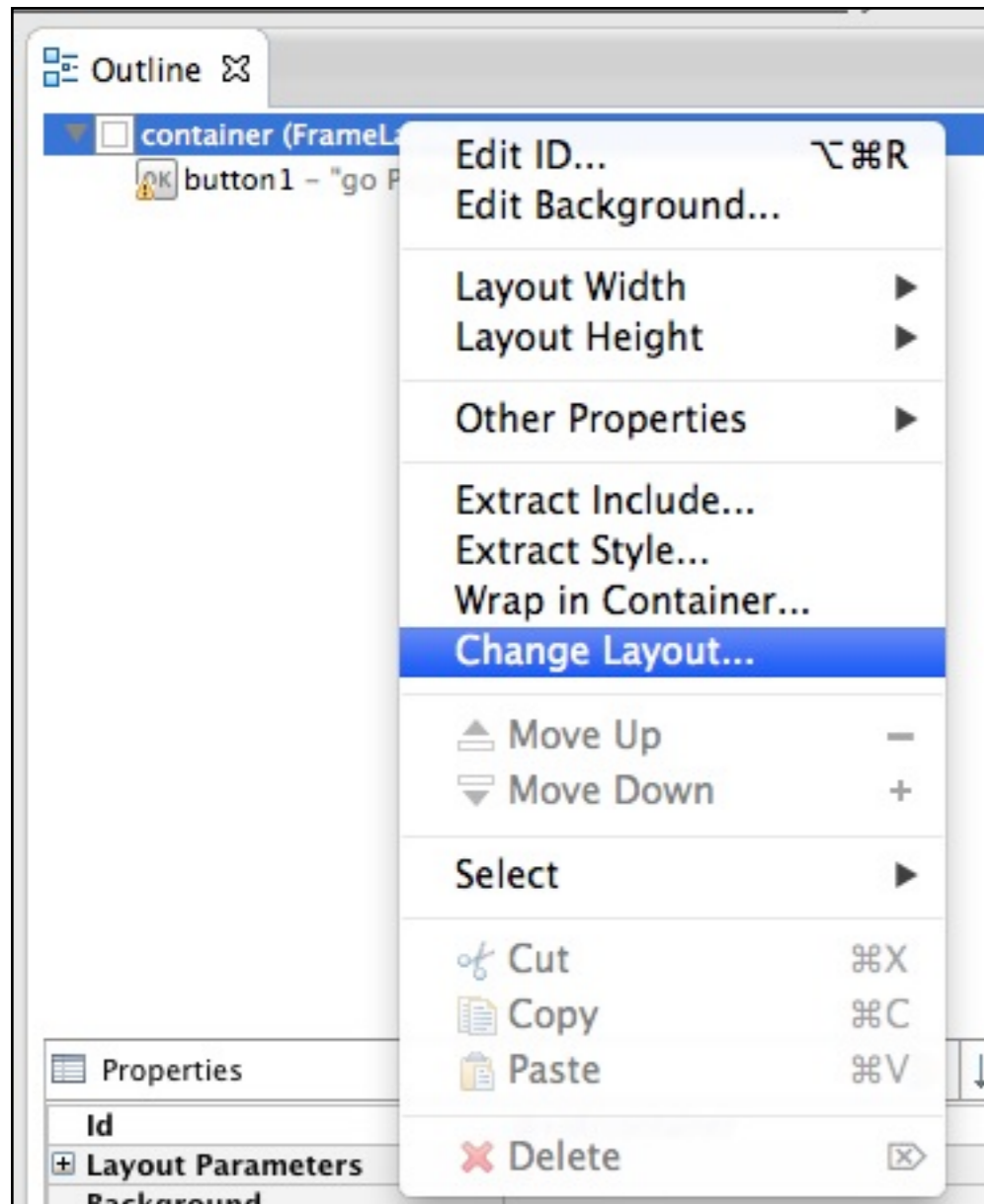


用 `onActivityResult(int requestCode, int resultCode, Intent data)` 函示接收使用

用 `void finish()` 函示設定回傳內容

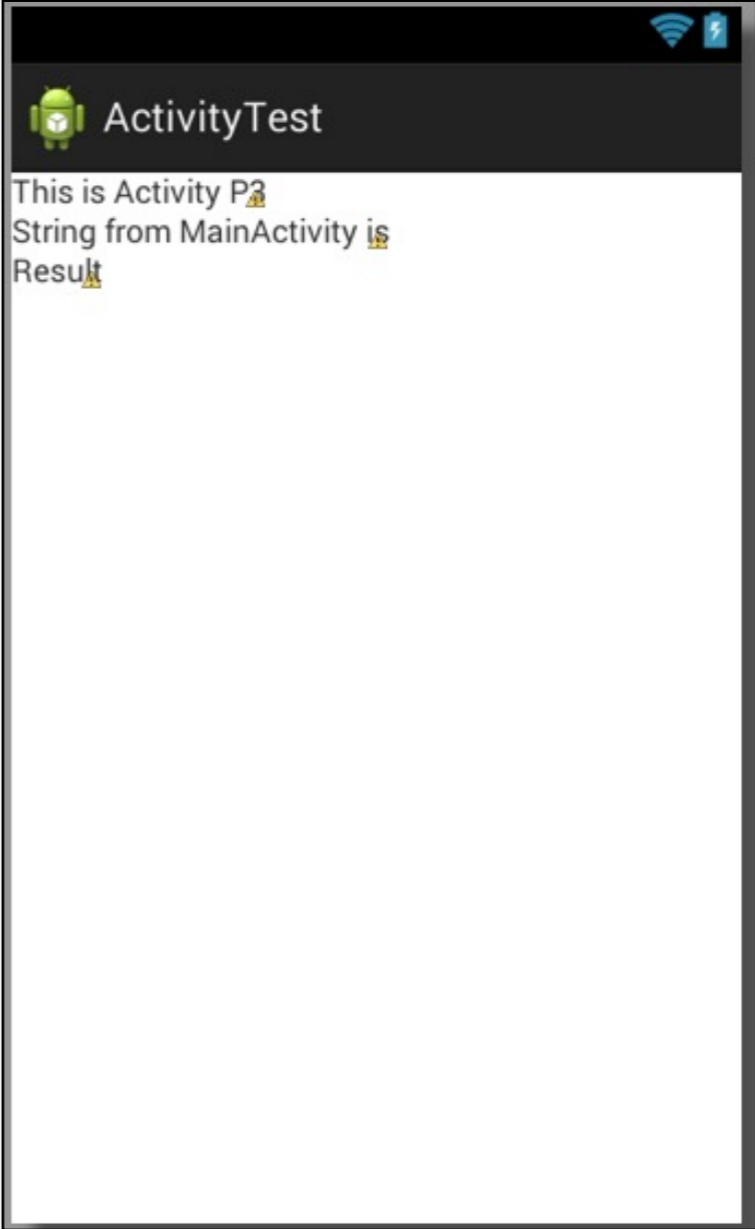
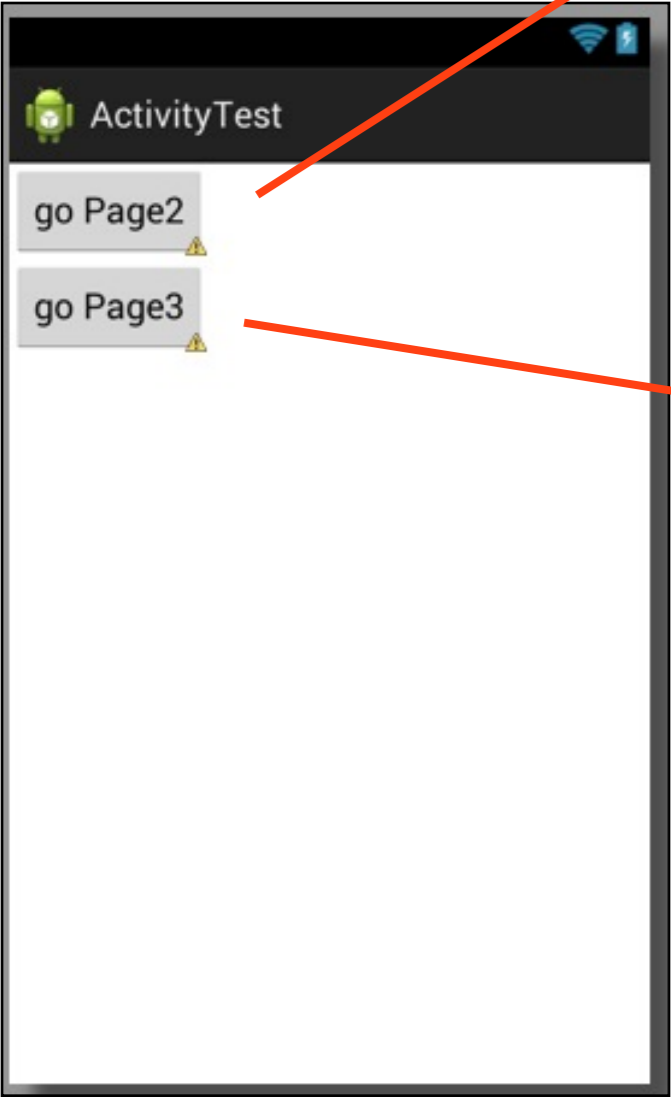
練習Activity之間的變數傳遞

- 首先更改Layout為LinearLayout



練習Activity之間的變數傳遞

- MainActivity加入go Page2及go Page3按鈕
- 新增P3Activity.java以及Layout:
activity_p3.xml
- activity_p3.xml加入識別文字”This is Activity P3”，並加入顯示文字用於顯示由MainActivity傳來的變數



Exercise

- 練習Activity變數傳遞與呼叫另一個Activity

MainActivity.java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Log.i("test", "onCreate");
    Btn2P2=(Button)findViewById(R.id.button1);
    Btn2P2.setOnClickListener(btnListener);

    Btn2P3=(Button)findViewById(R.id.button2);
    Btn2P3.setOnClickListener(btnListener);
}
```

注意!同一個Listener

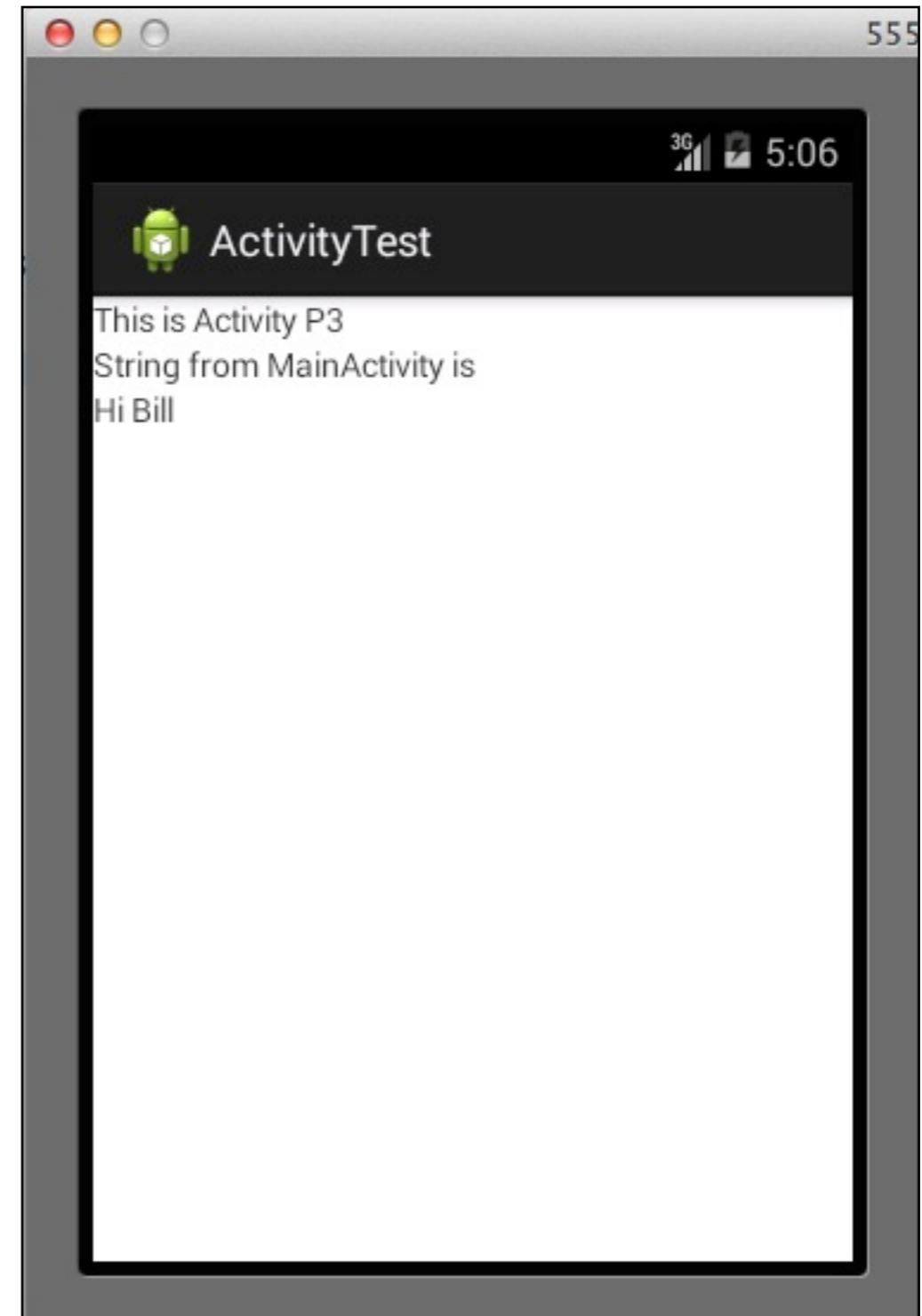
```
OnClickListener btnListener= new OnClickListener() {

    @Override
    public void onClick(View v) {
        if(v.getId()==R.id.button1)
        {
            Intent P2Intent=new Intent(getBaseContext(), P2Activity.class);
            P2Intent.putExtra("name", "Hi Jimmy");
            startActivityForResult(P2Intent, 2);
            startActivity(P2Intent);
        }else if(v.getId()==R.id.button2)
        {
            Intent P3Intent=new Intent(getBaseContext(),P3Activity.class);
            P3Intent.putExtra("name", "Hi Bill");
            startActivityForResult(P3Intent, 3);
        }
    }
};
```

記得在AndroidManifest.xml加入此Activity

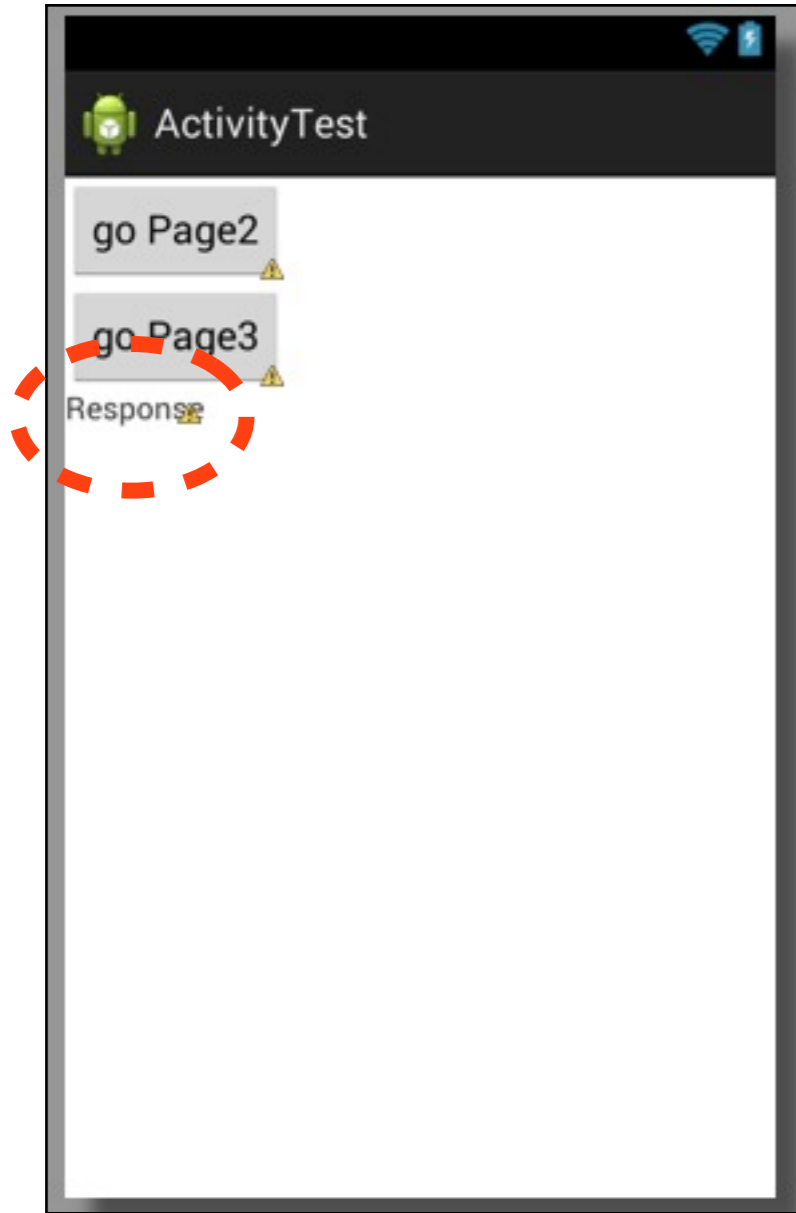
P3Activity.java

```
public class P3Activity extends Activity {  
  
    private TextView resulTextView;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        // TODO Auto-generated method stub  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_p3);  
        resulTextView=(TextView)findViewById(R.id.textView3);  
  
        Intent getIntent=getIntent();  
        String nameString=getIntent.getStringExtra("name");  
        resulTextView.setText(nameString);  
    }  
}
```



由P3Activity回傳資料

activity_main.xml



Properties	
Id	@+id/textView1
Layout Parameters	{}
Text	Response
Hint	
Text Color	
Text Appearance	?android:attr/textAppearanceSmall (@andr...
Text Size	
Content Description	
TextView	{} Text: Response Hint: Text Color: Text Color Hint: @android:color/hint_foreground_holo_li... Text Appearance: ?android:attr/textAppearanceSmall (@andr... Text Size: Typeface: Text Style: Font Family: Text Color Link: @android:color/holo_blue_light Max Lines: Max Height: Lines:

加入回傳顯示用字串

P3Activity.java 加入回傳字串

```
@Override
public void finish() {
    // TODO Auto-generated method stub
    Intent responseIntent=new Intent();

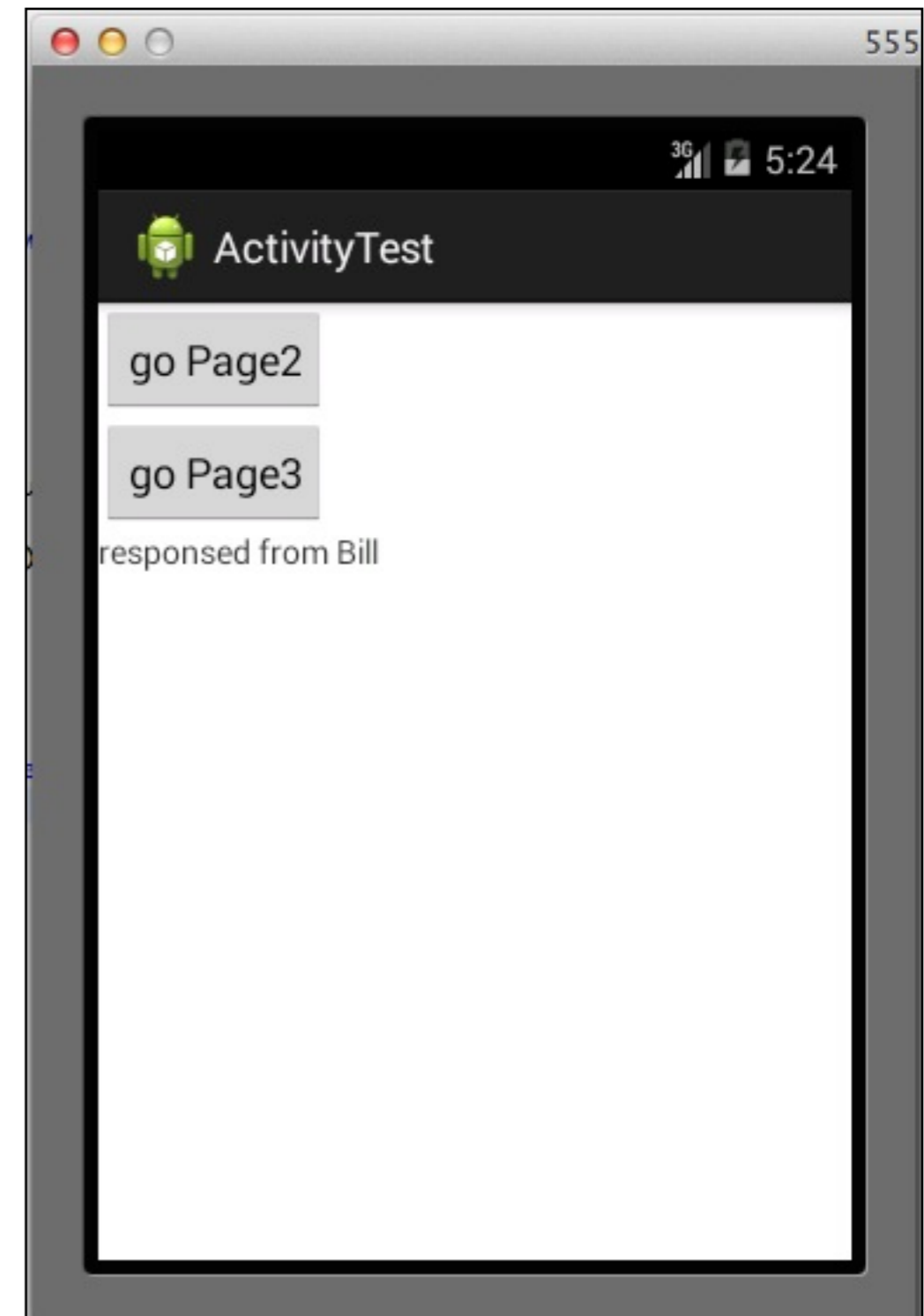
    responseIntent.putExtra("response", "responded from Bill");
    setResult(RESULT_OK, responseIntent);
    super.finish();
}
```

MainActivity.java 加入回傳判斷

```
private TextView responseTextView;
..
responseTextView=(TextView)findViewById(R.id.textView1);
..

protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // TODO Auto-generated method stub
    super.onActivityResult(requestCode, resultCode, data);

    if(requestCode==3)
    {
        if(resultCode==RESULT_OK)
        {
            String response=data.getStringExtra("response");
            responseTextView.setText(response);
        }
    }
}
```



(執行結果)

Exercise

- 練習由另一個Activity回傳資料

你已學會

- Android如何獲取View上的元件
- 如何設定動作
- Activity的生命週期
- Activity切換到另一個Activity
- Activity參數傳遞

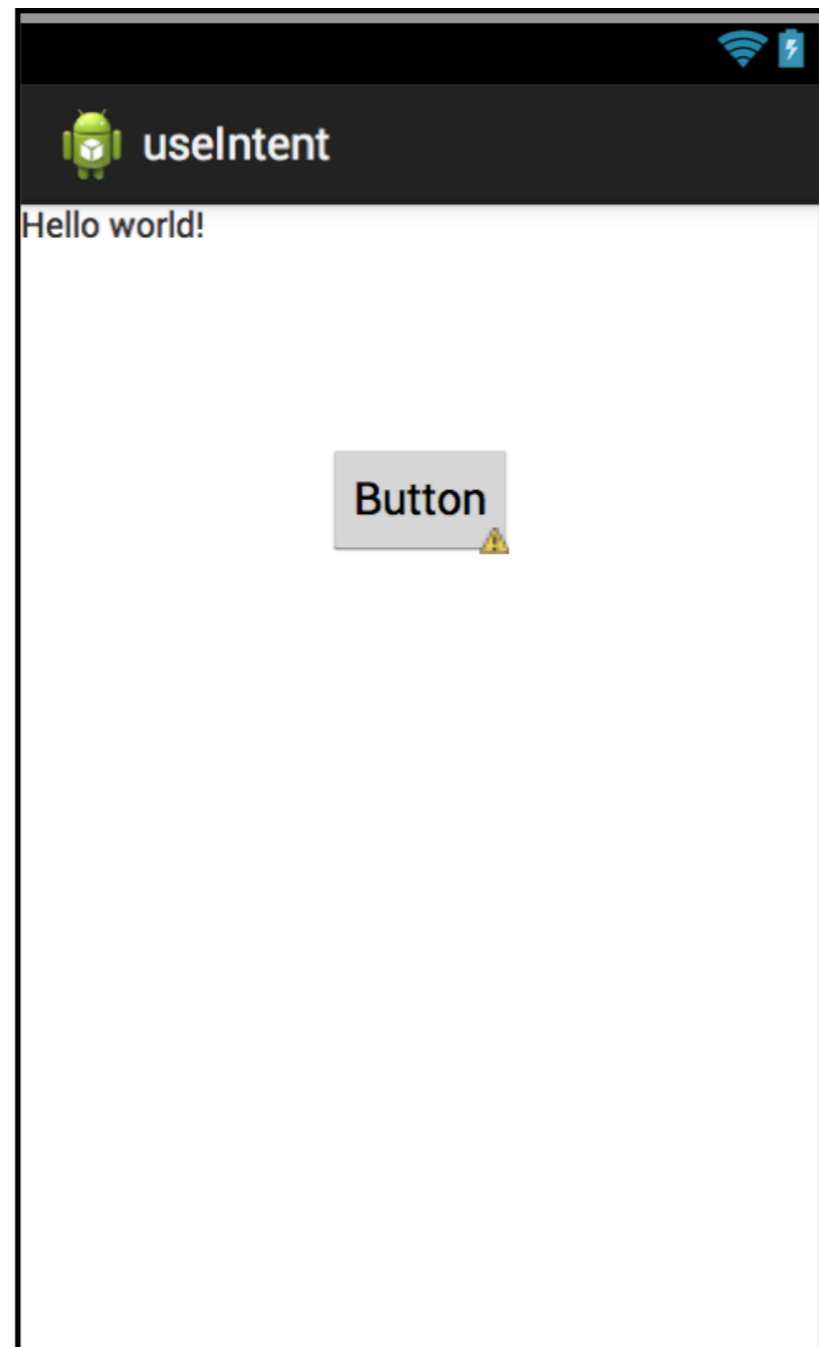
Intent的其他應用

- 開啟網頁
- 撥打電話
- 傳送簡訊
- ...

參考網站:YSL的程式天堂

<http://ysl-paradise.blogspot.tw/2008/12/intent.html>

開啟新專案並設定如下Layout



設定Intent開啟網頁

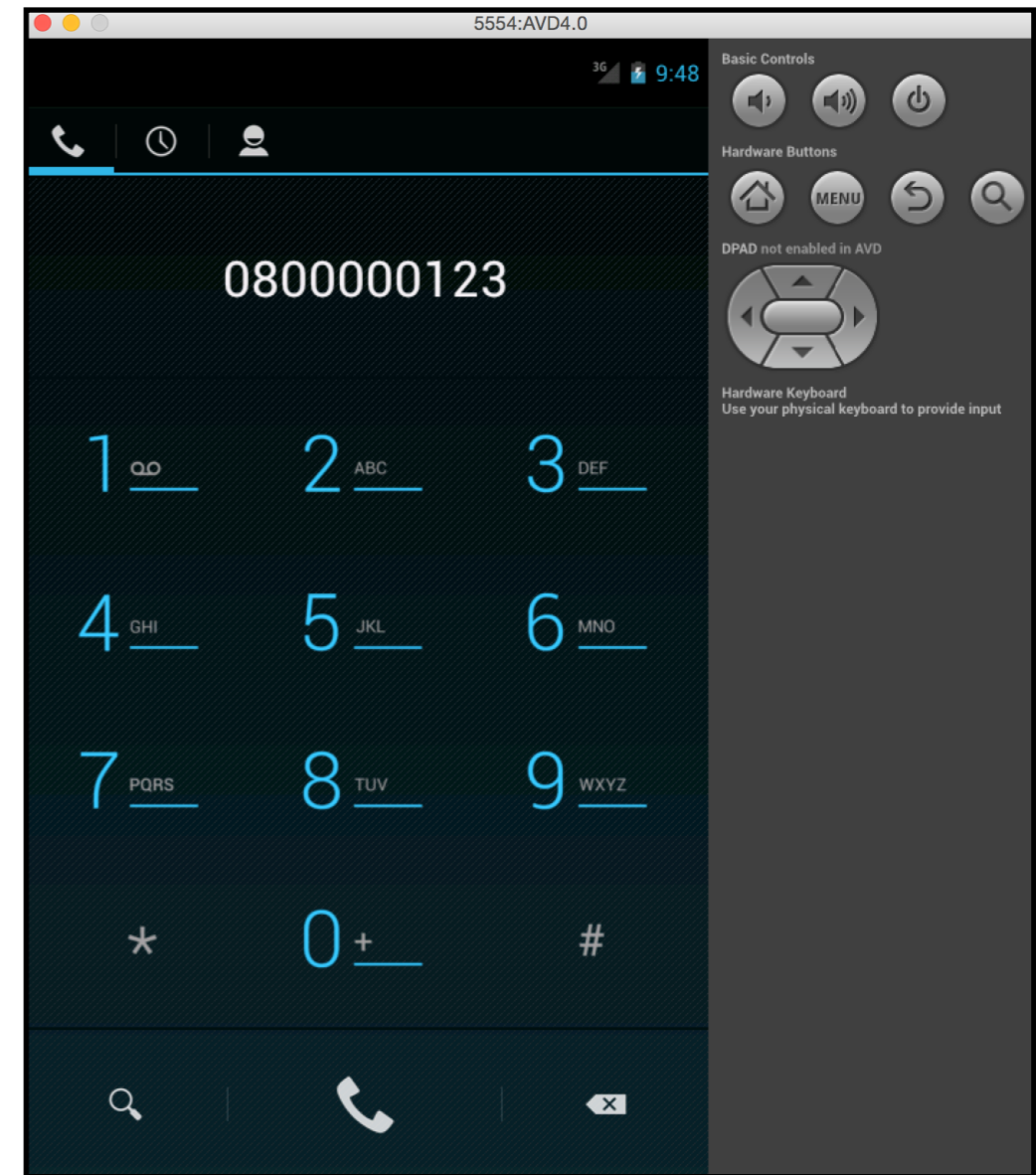
```
public class MainActivity extends Activity {  
  
    private Button myButton;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        myButton=(Button)findViewById(R.id.button1);  
        myButton.setOnClickListener(new OnClickListener() {  
  
            @Override  
            public void onClick(View v) {  
                //開啟網頁  
                Uri uri = Uri.parse("http://google.com");  
                Intent it = new Intent(Intent.ACTION_VIEW, uri);  
                startActivity(it);  
            }  
        });  
    }  
}
```



執行結果

設定Intent呼叫撥號程式

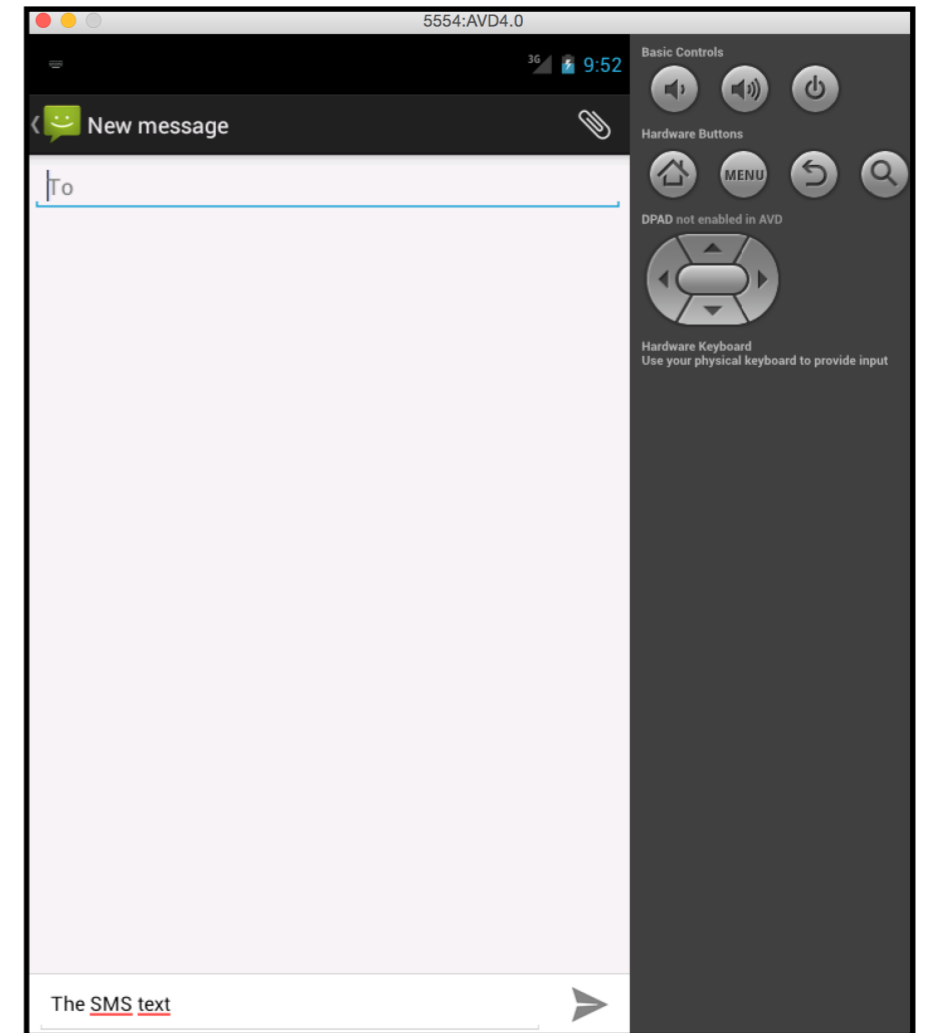
```
public class MainActivity extends Activity {  
  
    private Button myButton;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        myButton=(Button)findViewById(R.id.button1);  
        myButton.setOnClickListener(new OnClickListener() {  
  
            @Override  
            public void onClick(View v) {  
                //叫出撥號程式  
                Uri uri = Uri.parse("tel:0800000123");  
                Intent it = new Intent(Intent.ACTION_DIAL, uri);  
                startActivity(it);  
            }  
        });  
    }  
}
```



執行結果

設定Intent傳送簡訊

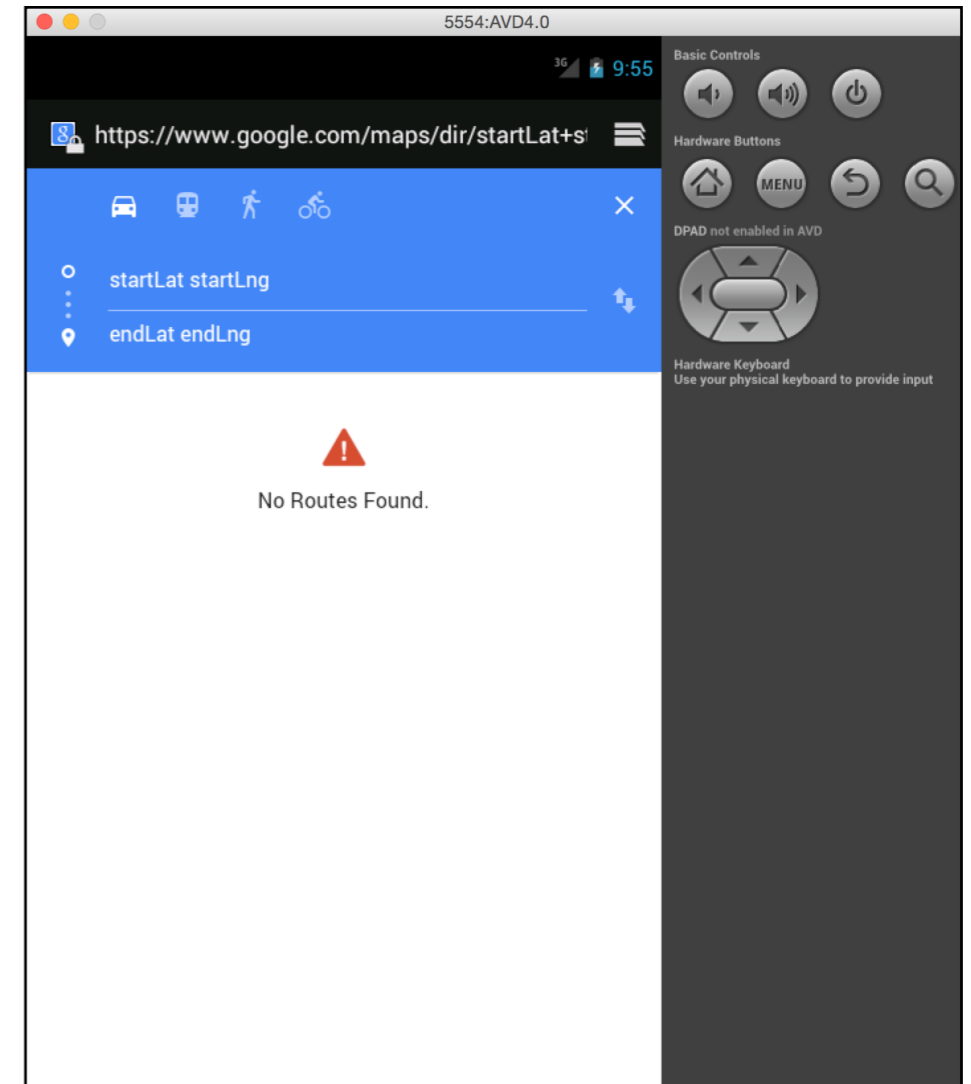
```
public class MainActivity extends Activity {  
  
    private Button myButton;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        myButton=(Button)findViewById(R.id.button1);  
        myButton.setOnClickListener(new OnClickListener() {  
  
            @Override  
            public void onClick(View v) {  
                //叫起簡訊程式  
                Intent it = new Intent(Intent.ACTION_VIEW);  
                it.putExtra("sms_body", "The SMS text");  
                it.setType("vnd.android-dir/mms-sms");  
                startActivity(it);  
            }  
        });  
    }  
}
```



執行結果

設定Intent路徑規劃

```
public class MainActivity extends Activity {  
  
    private Button myButton;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        myButton=(Button)findViewById(R.id.button1);  
        myButton.setOnClickListener(new OnClickListener() {  
  
            @Override  
            public void onClick(View v) {  
                //路徑規劃  
                Uri uri = Uri.parse("http://maps.google.com/maps?  
f=d&saddr=startLat%20startLng&daddr=endLat%20endLng&hl=en");  
                Intent it = new Intent(Intent.ACTION_VIEW, uri);  
                startActivity(it);  
            }  
        });  
    }  
}
```



執行結果(需要實機測試)

廣播Broadcast

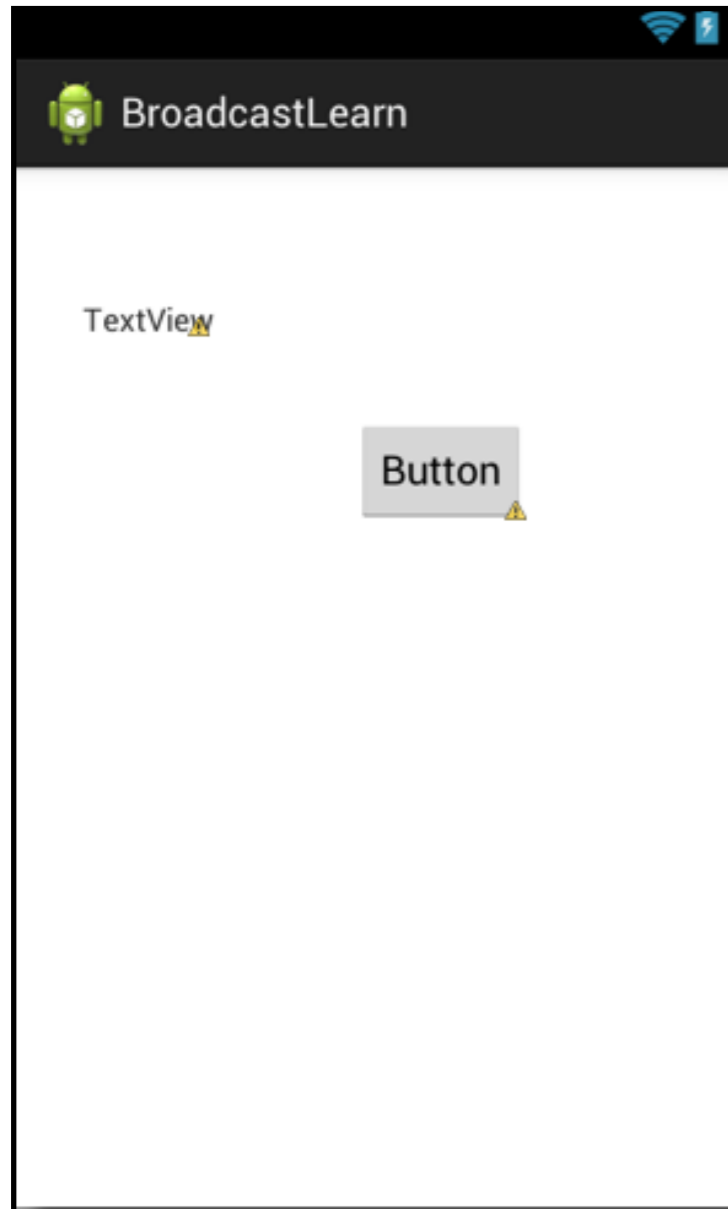
- 可以監控某些事件的發生，並且做相對應的處理
- 例如電池變化、撥號、系統進入等等
- 可以設定自己的狀態並且監控

廣播Broadcast

- 新增IntentFilter並設定監控Action
 - IntentFilter.addAction可以增加監控Action
- 建立BroadcastReceiver
 - onReceiver() 加入對應動作
- 監控的地方需註冊BroadcastReceiver
- 取消監控時要刪除BroadcastReceiver
- 用sendBroadcast送出自訂事件
- 也常用於跨Activity之間的資料傳遞

完成下列Layout

在onCreate()new 一個BroadcastReceiver
在onReceive()內判斷Action，並加入動作



```
//新增BroadcastReceiver並且設定接收動作
myBReceiver = new BroadcastReceiver() {

    @Override
    public void onReceive(Context context, Intent intent) {

        String action=intent.getAction();
        String data=intent.getStringExtra("name");
        if(action.equals("Br1Action"))
            resultTxt.setText(data);

    }

};
```

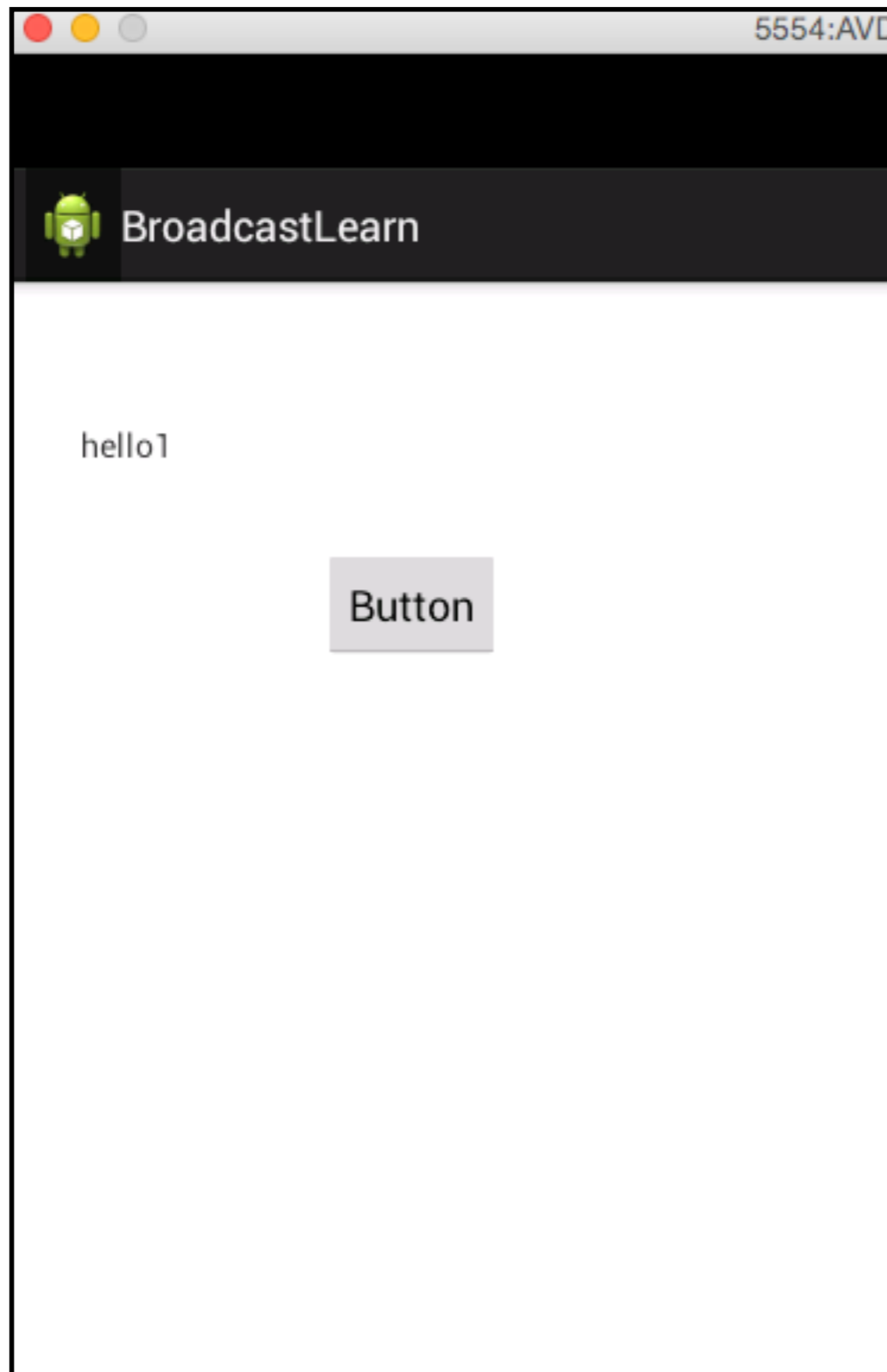
在onCreate()中新增IntentFilter並且註冊此動作

```
IntentFilter filter= new IntentFilter("Br1Action");  
registerReceiver(myBReceiver, filter);
```

設定Button的動作，並且送出Broadcast

```
btn1.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        Intent intent1 = new Intent("Br1Action"); //Br1Action為Action  
        intent1.putExtra("name", "hello1"); //加入key-value  
        sendBroadcast(intent1);  
    }  
});
```

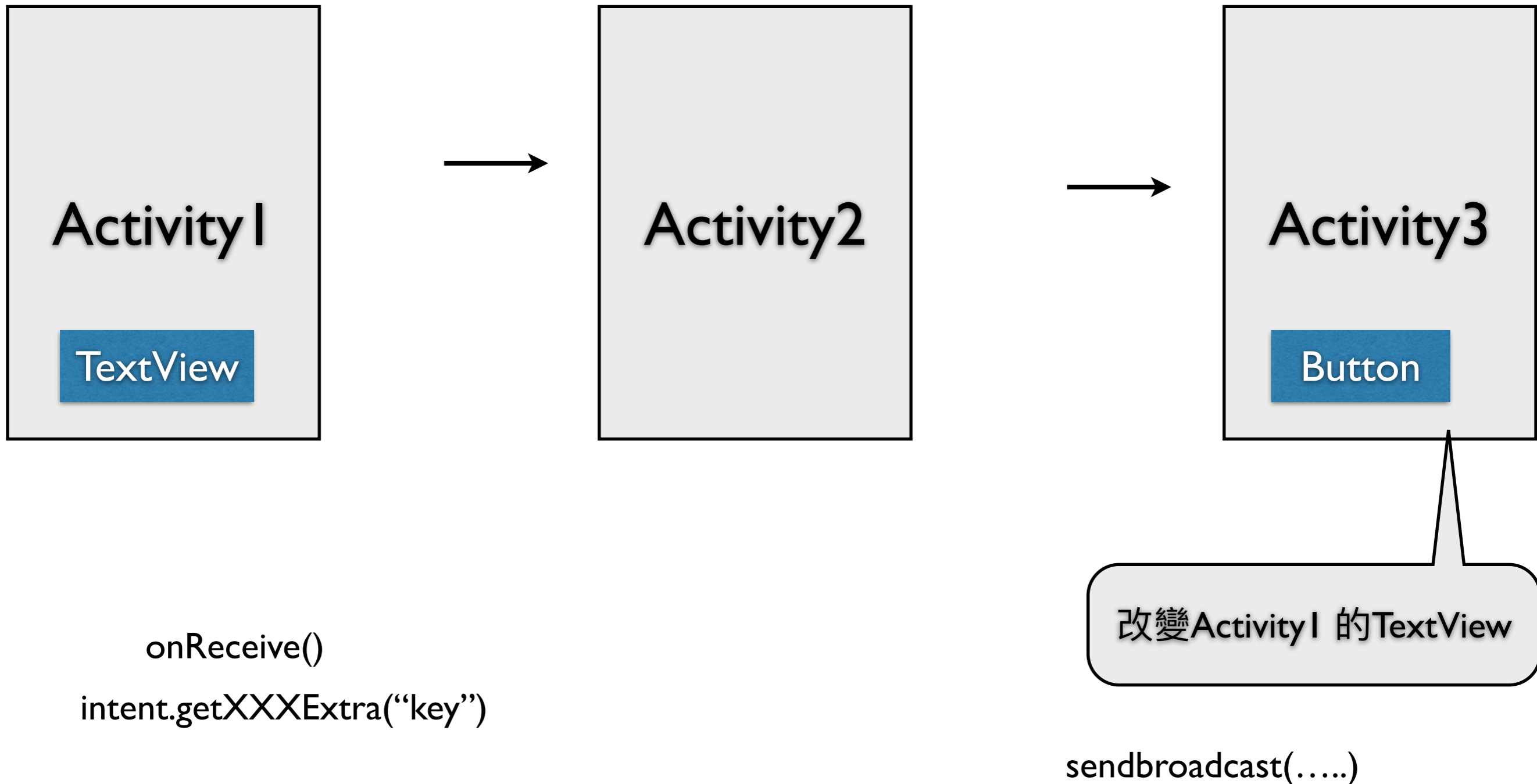
執行結果. 透過Broadcast修改TextView



```
@Override
    protected void onStop() {
        unregisterReceiver(myBReceiver);
        super.onStop();
    }
```

記得不用時要取消此BroadcastReceiver

Exercise: 跨Activity間透過Broadcast做資料傳遞



參考解答

```
//新增BroadcastReceiver並且設定接收動作
BroadcastReceiver myBReceiver = new BroadcastReceiver() {

    @Override
    public void onReceive(Context context, Intent intent) {

        String action=intent.getAction();
        String data=intent.getStringExtra("name");
        if(action.equals("Br1Action"))
            resultTxt.setText(data);

    }

};

//要過濾的Action
IntentFilter filter= new IntentFilter("Br1Action");
registerReceiver(myBReceiver, filter);
```

P3Activity.java

MainActivity.java

```
protected void onCreate(Bundle savedInstanceState) {
    setContentView(R.layout.p3_activity);
    btn=(Button)findViewById(R.id.button1);

    btn.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {

            Intent intent1 = new Intent("Br1Action");
            intent1.putExtra("name", "hello1");
            sendBroadcast(intent1);

        }

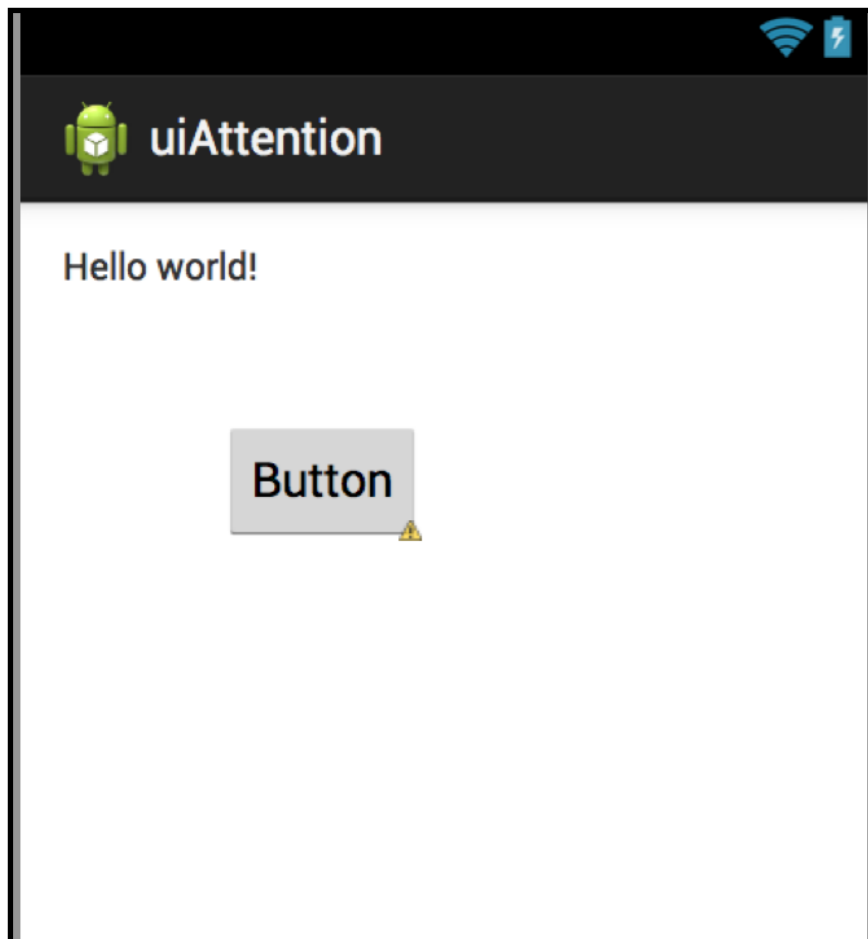
    });
    super.onCreate(savedInstanceState);
}
```

UI使用注意事項

- 主程序超過5秒沒有反應會錯誤
- 耗時的程式使用副程序操做
- 只有主程序Main Thread可以改變UI
- 副程序無法改變UI，但可透過Handler
改變UI

主執行緒超時

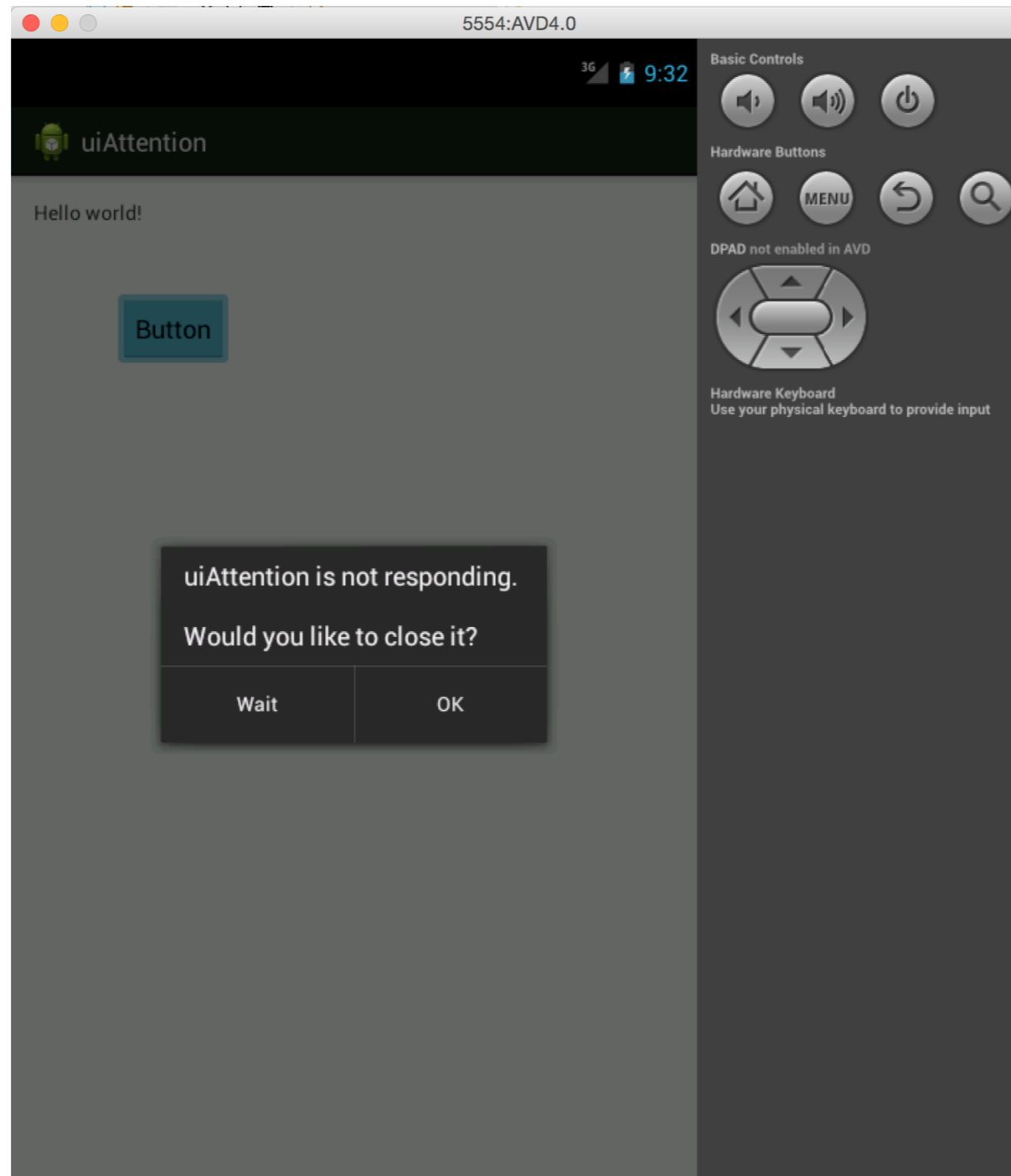
做出以下Layout



```
public class MainActivity extends Activity {  
  
    private Button myBtn;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        myBtn=(Button)findViewById(R.id.button1);  
  
        myBtn.setOnClickListener(new View.OnClickListener() {  
  
            @Override  
            public void onClick(View v) {  
                //讓主線程Main Thread超過5秒  
                while(true)  
                    ;  
            }  
        });  
    }  
}
```

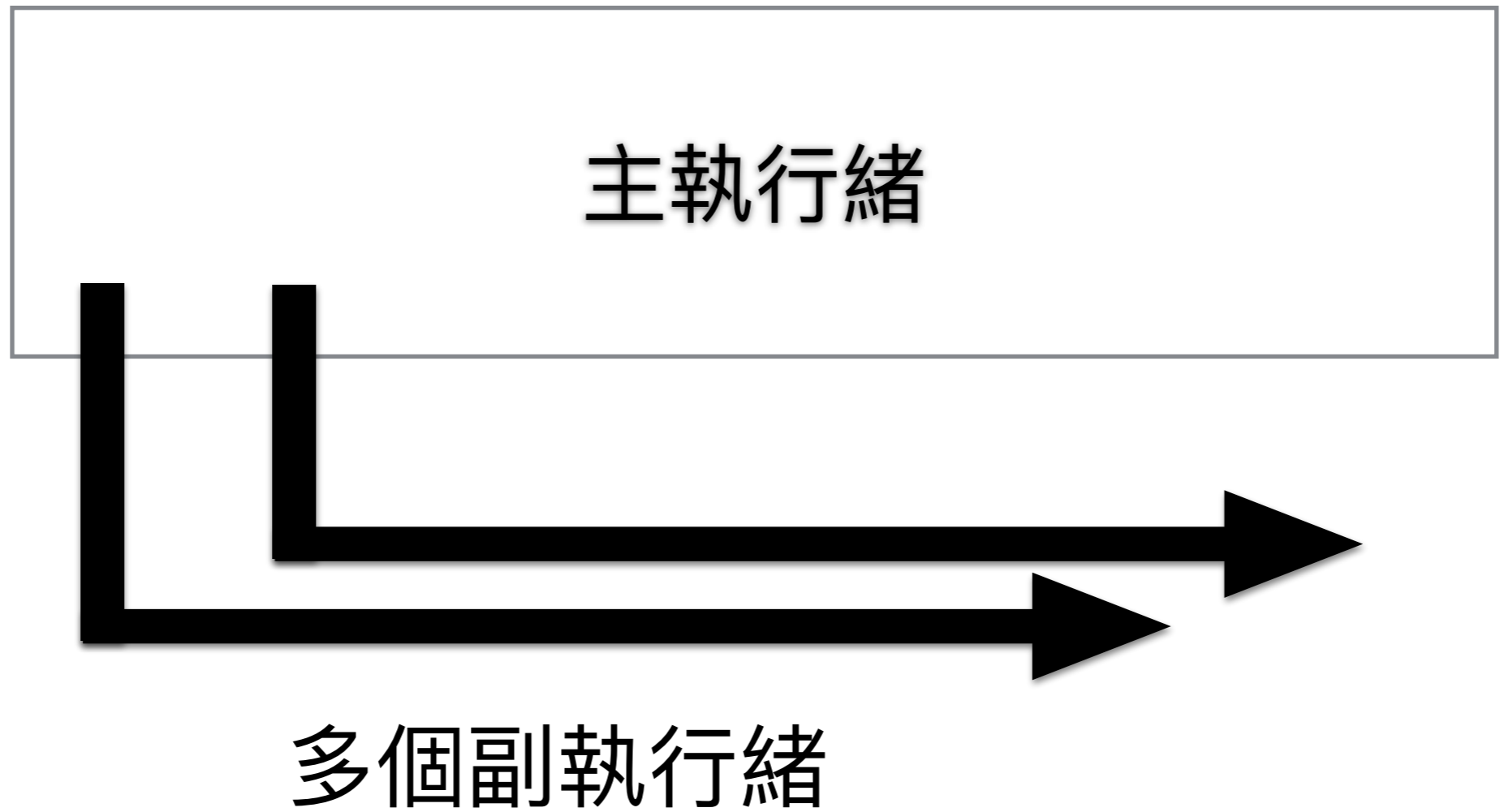
按下按鈕後，跑無窮迴圈

執行結果，發生錯誤

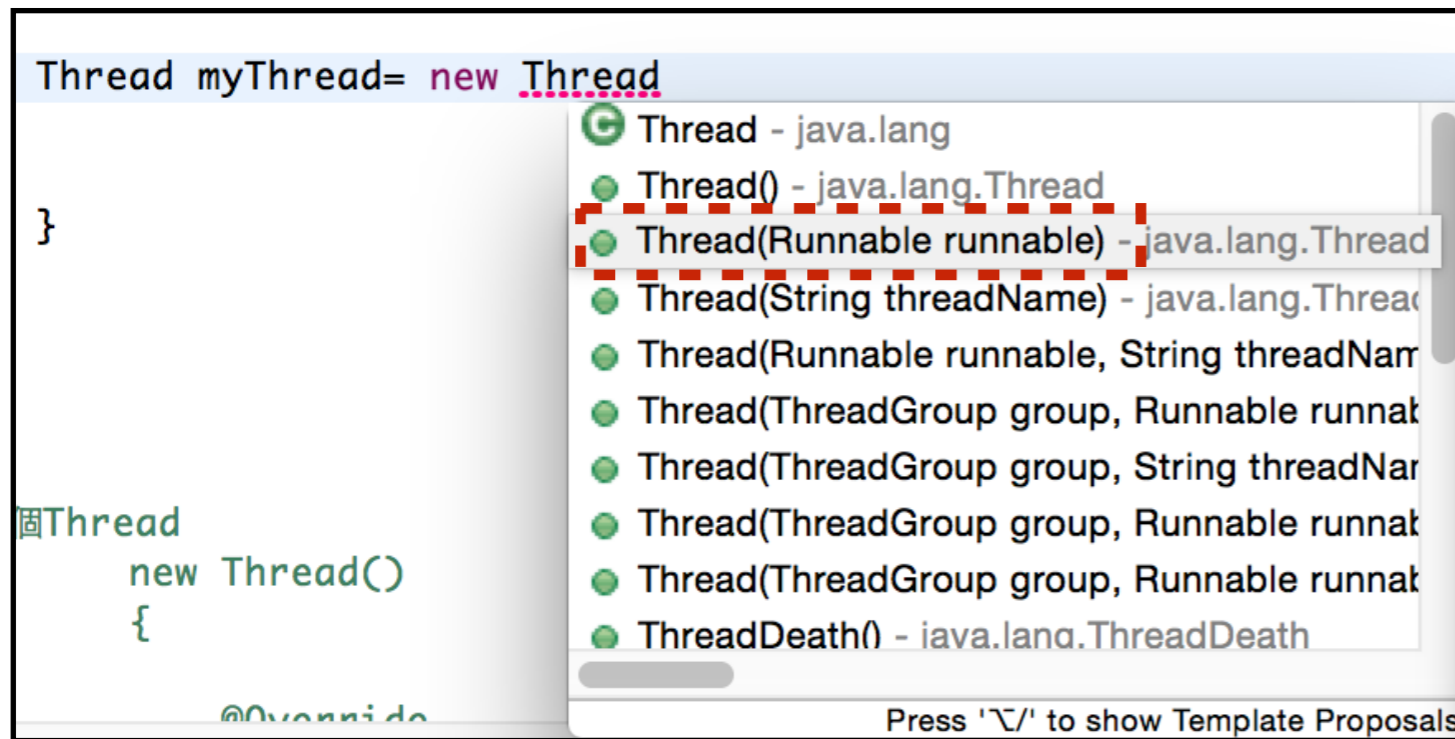


使用副執行緒執行耗時程式 (使用Thread)

執行緒 Thread



Thread使用方法 => 參數為Runnable物件



new一個Runnable物件

```
Runnable myTask=new Runnable() {  
  
    @Override  
    public void run() {  
        // TODO Auto-generated method stub  
    }  
};
```

將原本的方法移到Runnable的Run裡面

```
myBtn.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
  
        Runnable myTask=new Runnable() {  
  
            @Override  
            public void run() {  
                while(true)  
                    ;  
            }  
        };  
        //將myTask Runnable物件交給Thread執行，注意！start才會執行  
        new Thread(myTask).start();  
  
    }  
});
```

start()執行

執行並不會發生錯誤

```
myBtn.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
  
        Runnable myTask=new Runnable() {  
  
            @Override  
            public void run() {  
                while(true)  
                    Log.i("test", "xxx");  
  
            }  
  
        };  
        //將myTask Runnable物件交給Thread執行，注意！start才會執行  
        new Thread(myTask).start();  
  
    }  
});
```

Javadoc Declaration Console LogCat Lint Warnings

no filters)
exprogress (Sessi
uiattention (Sessic

Level	Time	PID	TID	Application	Tag	Text
I	01-04 00:04:52.709	13153	13194	com.example.uiatt...	test	xxx
I	01-04 00:04:52.709	13153	13194	com.example.uiatt...	test	xxx
I	01-04 00:04:52.709	13153	13194	com.example.uiatt...	test	xxx
I	01-04 00:04:52.709	13153	13194	com.example.uiatt...	test	xxx
I	01-04 00:04:52.709	13153	13194	com.example.uiatt...	test	xxx
I	01-04 00:04:52.709	13153	13194	com.example.uiatt...	test	xxx
I	01-04 00:04:52.709	13153	13194	com.example.uiatt...	test	xxx
I	01-04 00:04:52.709	13153	13194	com.example.uiatt...	test	xxx
I	01-04 00:04:52.709	13153	13194	com.example.uiatt...	test	xxx
I	01-04 00:04:52.709	13153	13194	com.example.uiatt...	test	xxx

uiAttention

Hello world!

Button

嘗試在副執行緒改變UI狀態

按下Button改變TextView

```
public class MainActivity extends Activity {

    private Button myBtn;
    private TextView myTxt;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        myTxt=(TextView)findViewById(R.id.textView1);
        myBtn=(Button)findViewById(R.id.button1);

        myBtn.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {

                Runnable myTask=new Runnable() {

                    @Override
                    public void run() {
//                        while(true)
//                            Log.i("test","xxx");
                        myTxt.setText("SSS");
                    }

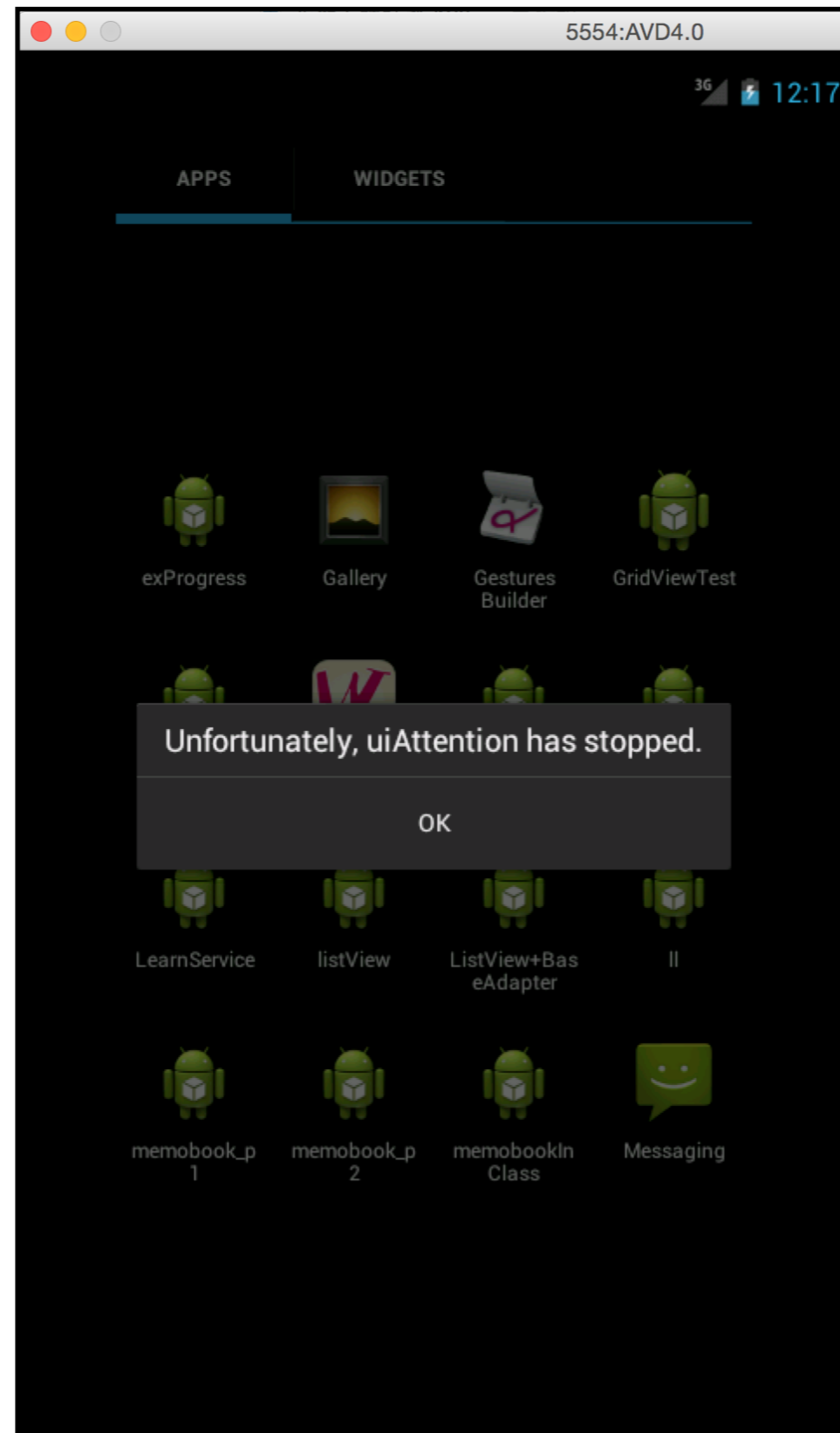
                };
                //將myTask Runnable物件交給Thread執行，注意！start才會執行
                new Thread(myTask).start();

            }

        });

    }
}
```

竟然發生錯誤!!



程式看起來沒有
錯...編譯也沒有錯...

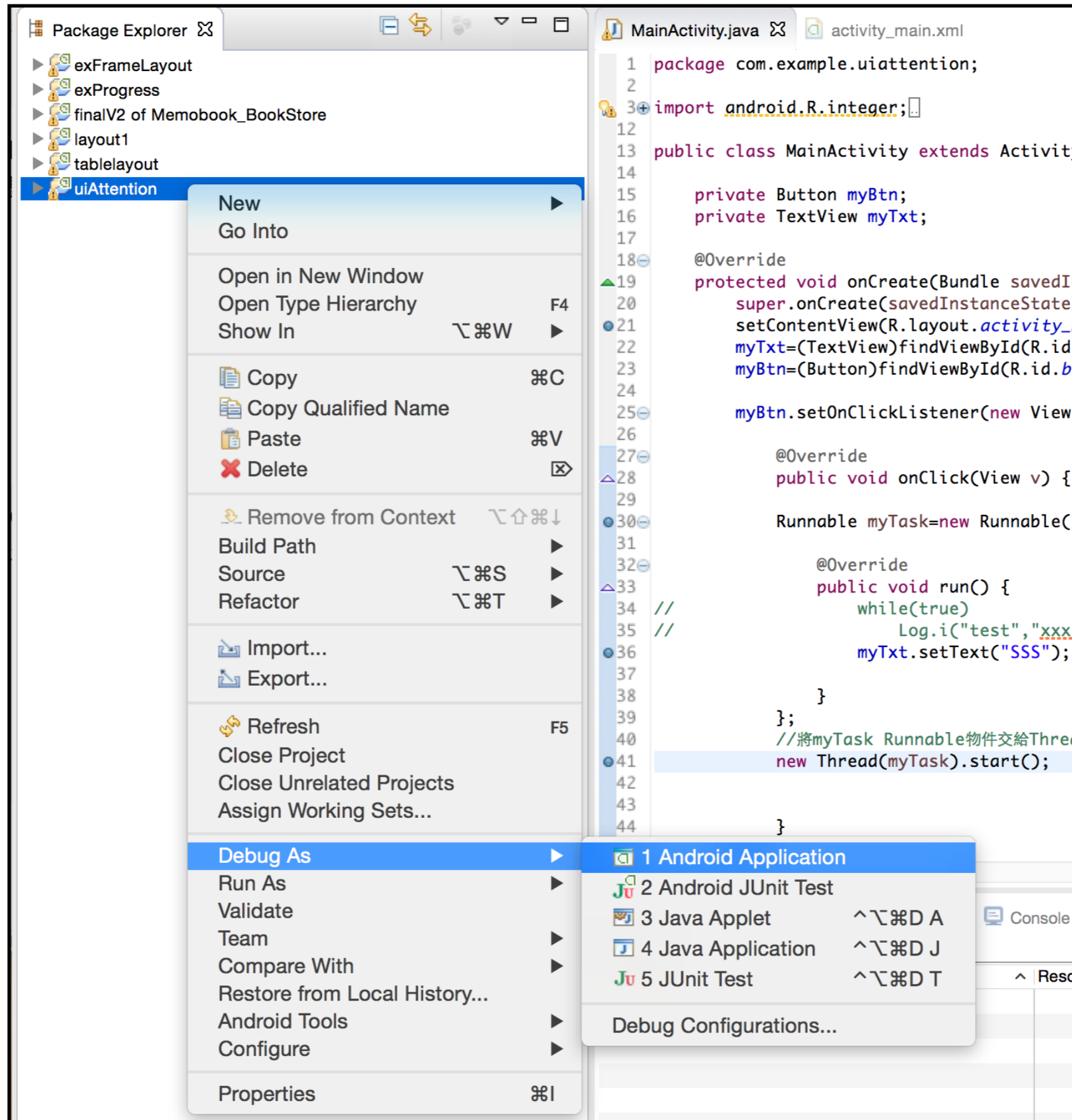


如何除錯?進入除錯模式

```
MainActivity.java activity_main.xml
1 package com.example.uiattention;
2
3 import android.R.integer;
12
13 public class MainActivity extends Activity {
14
15     private Button myBtn;
16     private TextView myTxt;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22         myTxt=(TextView)findViewById(R.id.textView1);
23         myBtn=(Button)findViewById(R.id.button1);
24
25         myBtn.setOnClickListener(new View.OnClickListener() {
26
27             @Override
28             public void onClick(View v) {
29
30                 Runnable myTask=new Runnable() {
31
32                     @Override
33                     public void run() {
34                         //
35                         //
36                         Log.i("test", "xxx");
37                         myTxt.setText("SSS");
38                     }
39                 };
40                 //將myTask Runnable物件交給Thread執行，注意！start才會執行
41                 new Thread(myTask).start();
42
43             }
44         });
45     }
46 }
```

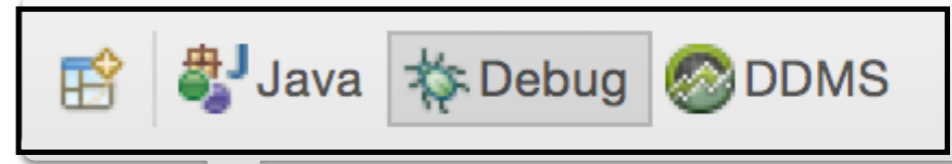
在可疑的地方連點兩下

● Project Name > Debug As > Android Application





執行命令



進入Debug模式

Thread [<1> main] (Suspended (breakpoint at line 21 in MainActivity))

```

16 private TextView myTxt;
17
18 @Override
19 protected void onCreate(Bundle savedInstanceState) {
20     super.onCreate(savedInstanceState);
21     setContentView(R.layout.activity_main);
22     myTxt=(TextView)findViewById(R.id.textView1);
23     myBtn=(Button)findViewById(R.id.button1);
24
25     myBtn.setOnClickListener(new View.OnClickListener() {
26
27         @Override
28         public void onClick(View v) {
29
30             Runnable myTask=new Runnable() {
31
32                 @Override
33                 public void run() {
34                     // while(true)
35                     //     Log.i("test","xxx");
36                     myTxt.setText("SSS");
37
38                 };
39             //將 Task Runnable物件交給Thread執行，注意！start才會執行
40             new Thread(myTask).start();
41

```

Outline:

- com.example.uiattention
 - MainActivity
 - myBtn : Button
 - myTxt : TextView
 - onCreate(Bundle) : void
 - new OnClickListener() {...}
 - onClick(View) : void
 - new Runnable() {...}
 - run() : void

```

17
18 @Override
19 protected void onCreate(Bundle savedInstanceState) {
20     super.onCreate(savedInstanceState);
21     setContentView(R.layout.activity_main);
22     myTxt=(TextView)findViewById(R.id.textView1);
23     myBtn=(Button)findViewById(R.id.button1);
24

```

執行到中斷點

執行命令說明

執行到下一個中斷點

下一步，但不進入子程式

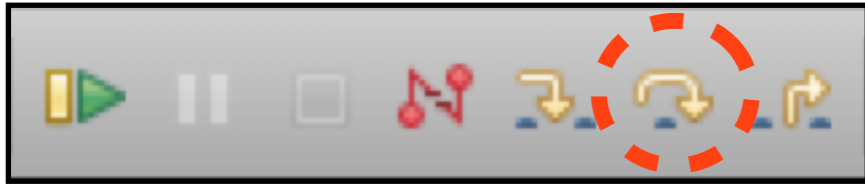


下一步，且進入子程式

從子程式跳開

按下Button,直到第30行仍沒有錯

```
MainActivity.java activity_main.xml
16 private TextView myTxt;
17
18 @Override
19 protected void onCreate(Bundle savedInstanceState) {
20     super.onCreate(savedInstanceState);
21     setContentView(R.layout.activity_main);
22     myTxt=(TextView)findViewById(R.id.textView1);
23     myBtn=(Button)findViewById(R.id.button1);
24
25     myBtn.setOnClickListener(new View.OnClickListener() {
26
27         @Override
28         public void onClick(View v) {
29
30             Runnable myTask=new Runnable() {
31
32                 @Override
33                 public void run() {
34                     // while(true)
35                     //     Log.i("test","xxx");
36                     myTxt.setText("SSS");
37
38                 }
39             };
40             //將myTask Runnable物件交給Thread執行，注意！start才會執行
41             new Thread(myTask).start();
```

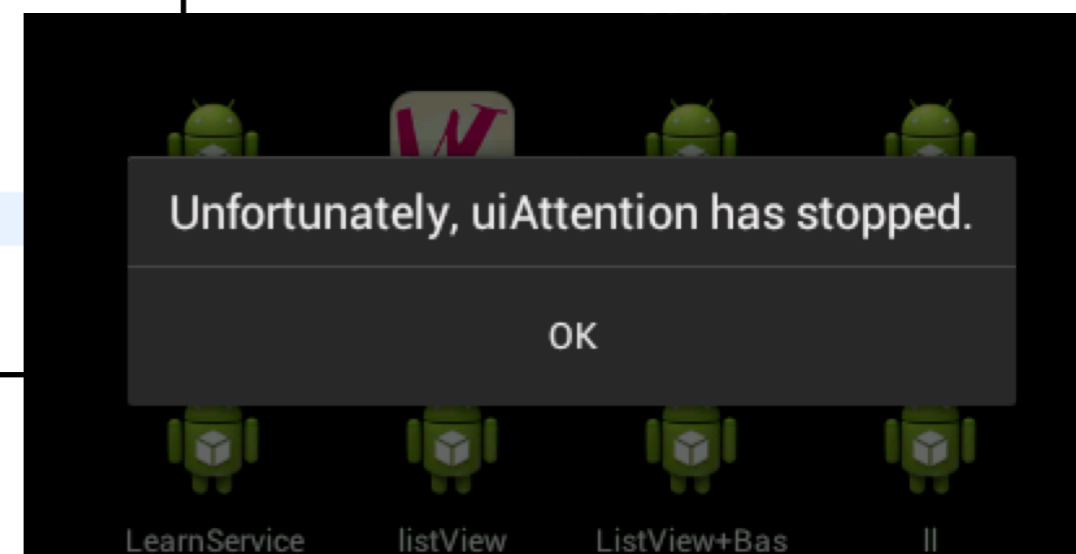


```
MainActivity.java ✕ activity_main.xml
16     private TextView myTxt;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22         myTxt=(TextView)findViewById(R.id.textView1);
23         myBtn=(Button)findViewById(R.id.button1);
24
25         myBtn.setOnClickListener(new View.OnClickListener() {
26
27             @Override
28             public void onClick(View v) {
29
30                 Runnable myTask=new Runnable() {
31
32                     @Override
33                     public void run() {
34                         //         while(true)
35                         //             Log.i("test", "xxx");
36                         myTxt.setText("SSS");
37
38                     }
39                 };
40                 //將myTask Runnable物件交給Thread執行，注意！start才會執行
41                 new Thread(myTask).start();
42
```

跑到不知名的程式檔內



```
MainActivity.java activity_main.xml ViewRootImpl.java ✕
3917 public void dumpGfxInfo(int[] info) {
3918     info[0] = info[1] = 0;
3919     if (mView != null) {
3920         getGfxInfo(mView, info);
3921     }
3922 }
3923
3924 private static void getGfxInfo(View view, int[] info) {
3925     DisplayList displayList = view.mDisplayList;
3926     info[0]++;
3927     if (displayList != null) {
3928         info[1] += displayList.getSize();
3929     }
3930
3931     if (view instanceof ViewGroup) {
3932         ViewGroup group = (ViewGroup) view;
3933
3934         int count = group.getChildCount();
3935         for (int i = 0; i < count; i++) {
3936             getGfxInfo(group.getChildAt(i), info);
3937         }
3938     }
3939 }
3940
3941 public void die(boolean immediate) {
3942     if (immediate) {
```

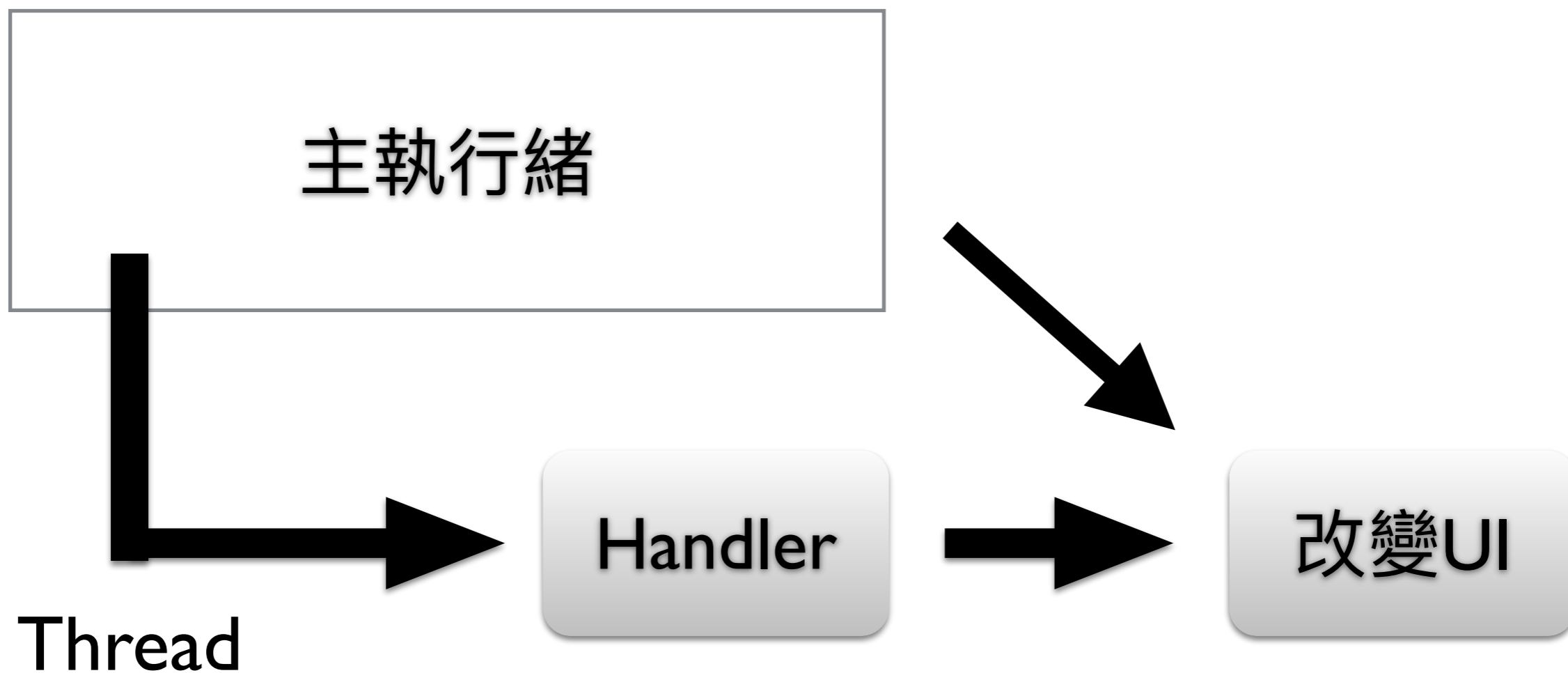


發生錯誤!!

原來副Thread不能改變UI

Android為了考量安全性和執行效能，
不允許副執行緒直接更改主畫面的資料

副執行緒需要透過handler與主程序改變UI

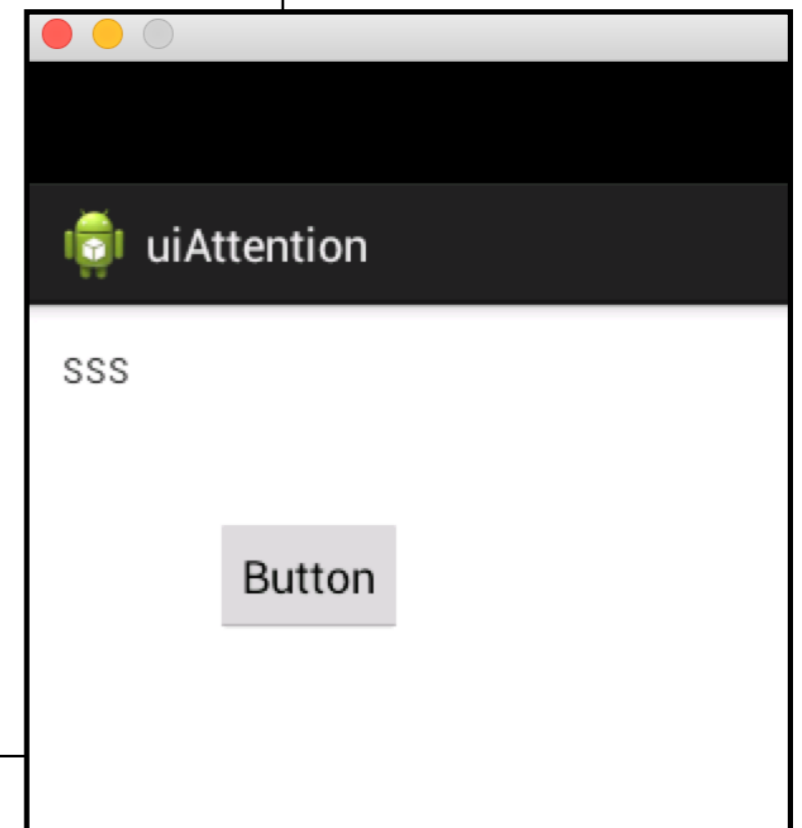


透過Handler從副執行緒改變UI

完整程式

```
public class MainActivity extends Activity {  
  
    private Button myBtn;  
    private TextView myTxt;  
    private Handler myHandler=new Handler();  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        myTxt=(TextView)findViewById(R.id.textView1);  
        myBtn=(Button)findViewById(R.id.button1);  
  
        myBtn.setOnClickListener(new View.OnClickListener() {  
  
            @Override  
            public void onClick(View v) {  
  
                Runnable myTask=new Runnable() {  
  
                    @Override  
                    public void run() {  
                        //myHandler.post(Runnable), 將要執行的程式置於run()  
                        myHandler.post(new Runnable() {  
  
                            @Override  
                            public void run() {  
                                myTxt.setText("SSS");  
                            }  
                        });  
                    }  
                });  
            }  
        });  
        //將myTask Runnable物件交給Thread執行，注意！start才會執行  
        new Thread(myTask).start();  
    }  
};  
}
```

參數為Runnable



執行結果

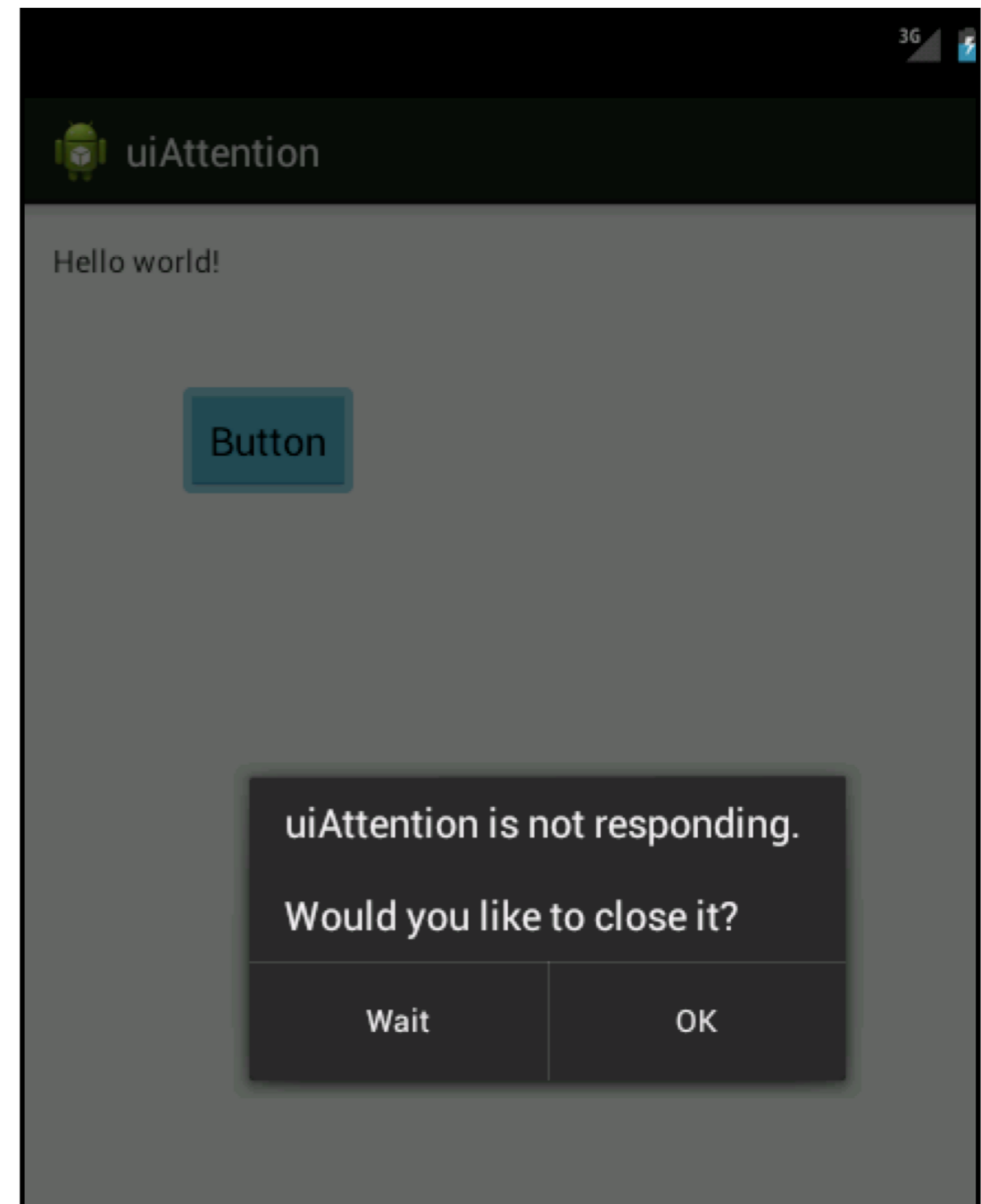
可將Runnable視為一個task，為
Thread、Handler的參數

使用時機：主執行緒不能idle太久 => 子執行緒 => 完成後改變UI

另一種延遲時間的方法Thread.sleep()

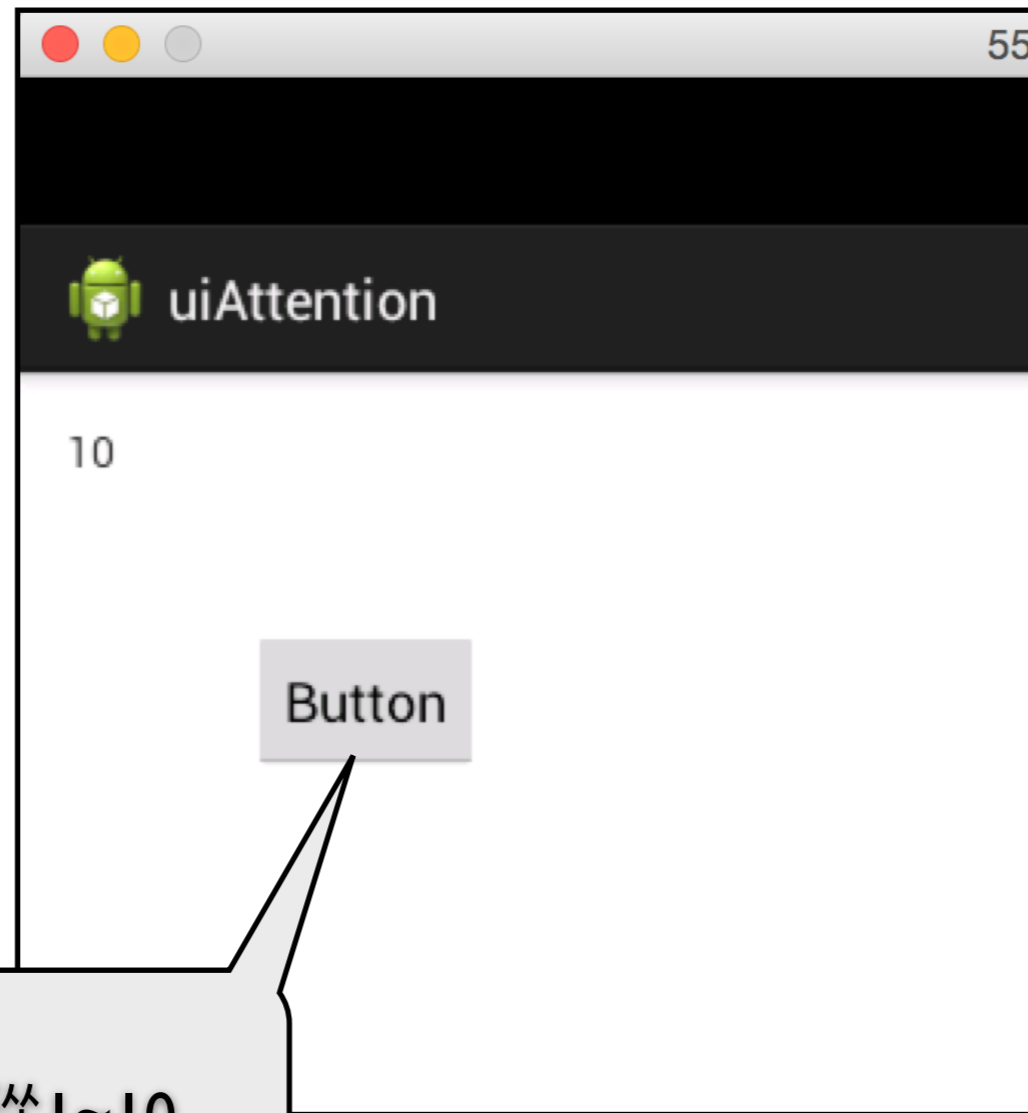
```
try {  
    Thread.sleep(10000); //延遲10秒鐘  
} catch (InterruptedException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

try... catch...(自動產生)
發生狀況時的處置



一樣造成crash

Exercise: 嘗試完成以下程式



按下按鈕, 更改文字從1~10

參考解答

```
public class MainActivity extends Activity {  
  
    private Button myBtn;  
    private TextView myTxt;  
    private Handler myHandler=new Handler();  
    int i;  
    private Runnable countTask;  
    private Runnable updateUITask;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        myTxt=(TextView)findViewById(R.id.textView1);  
        myBtn=(Button)findViewById(R.id.button1);  
        //定義countTask  
  
        //定義upadteUITask  
  
        myBtn.setOnClickListener(new View.OnClickListener() {  
  
            @Override  
            public void onClick(View v) {  
  
                //執行此task  
                new Thread(countTask).start();  
            }  
        });  
    }  
}
```

```
countTask = new Runnable() {  
    public void run() {  
        //延遲的動作在此task內做  
        try {  
            for(i=0;i<10;i++)  
            {  
                Thread.sleep(1000);  
                //每隔一秒透過Handler執行Update UI的工作  
                myHandler.post(updateUITask);  
            }  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
    }  
};
```

```
updateUITask = new Runnable() {  
  
    @Override  
    public void run() {  
        myTxt.setText(Integer.toString(i));  
    }  
};
```

Progress Dialog用法

- setTitle : 設定Title
- setMessage : 設定訊息
- setIndeterminate : 是否無法判斷進度
- setCancelable : 是否可以按返回鍵取消 ;
- show : 顯示ProgressDialog。
- dismiss : 刪除ProgressDialog

```

public class MainActivity extends Activity {

    private Button myBtn;
    private Runnable countTask;
    private ProgressDialog pd;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        pd = new ProgressDialog(MainActivity.this);
        pd.setTitle("Loading"); //設置標題
        pd.setMessage("Please wait..."); //設置body訊息
        pd.setCancelable(false); //設定不可點掉

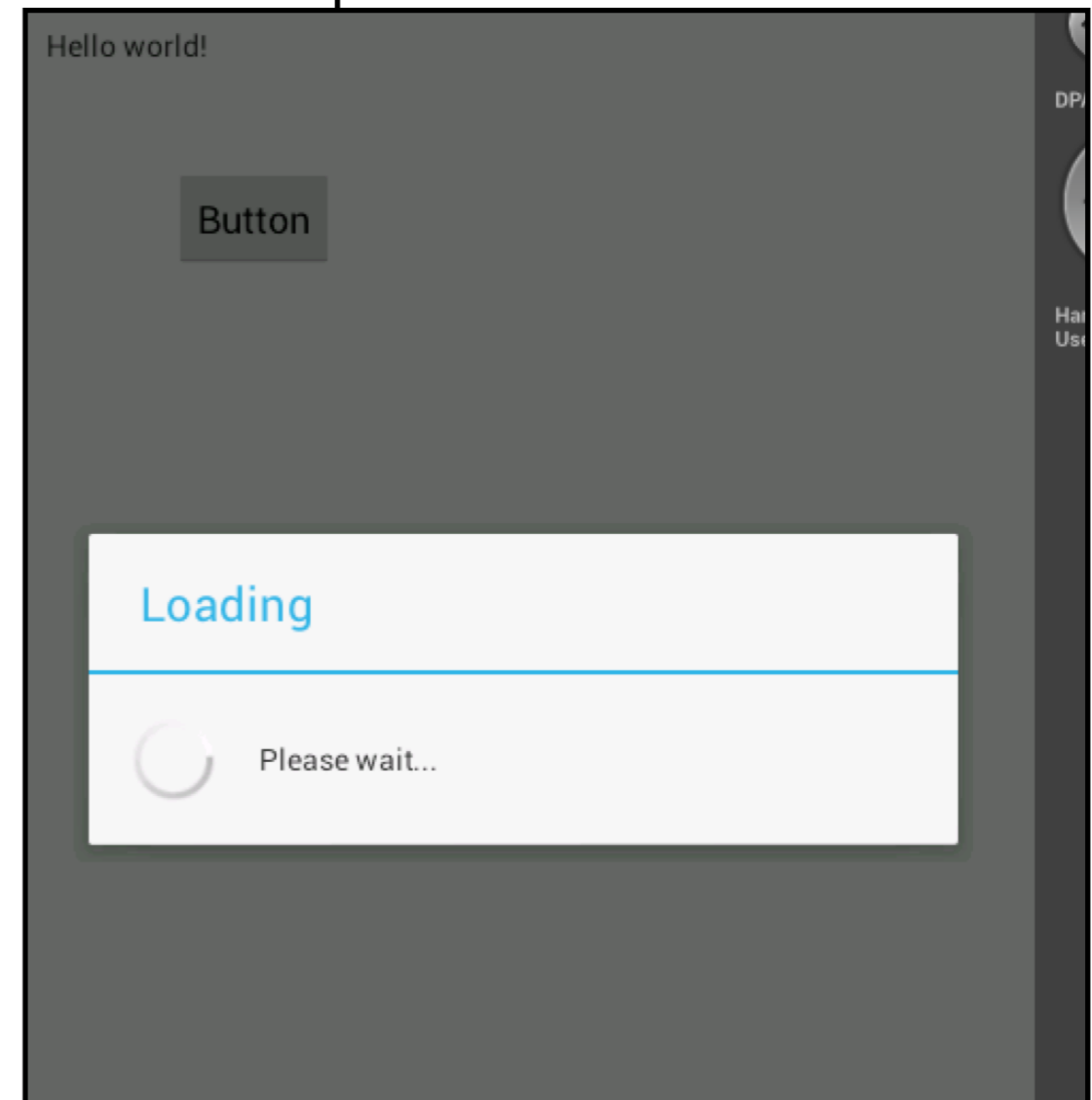
        myBtn=(Button)findViewById(R.id.button1);

        //定義countTask
        countTask = new Runnable() {
            public void run() {
                //延遲的動作在此task內做
                try {
                    Thread.sleep(3000);
                } catch (InterruptedException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
                pd.dismiss();
            }
        };

        myBtn.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                pd.show();
                //執行此task
                new Thread(countTask).start();
            }
        });
    }
}

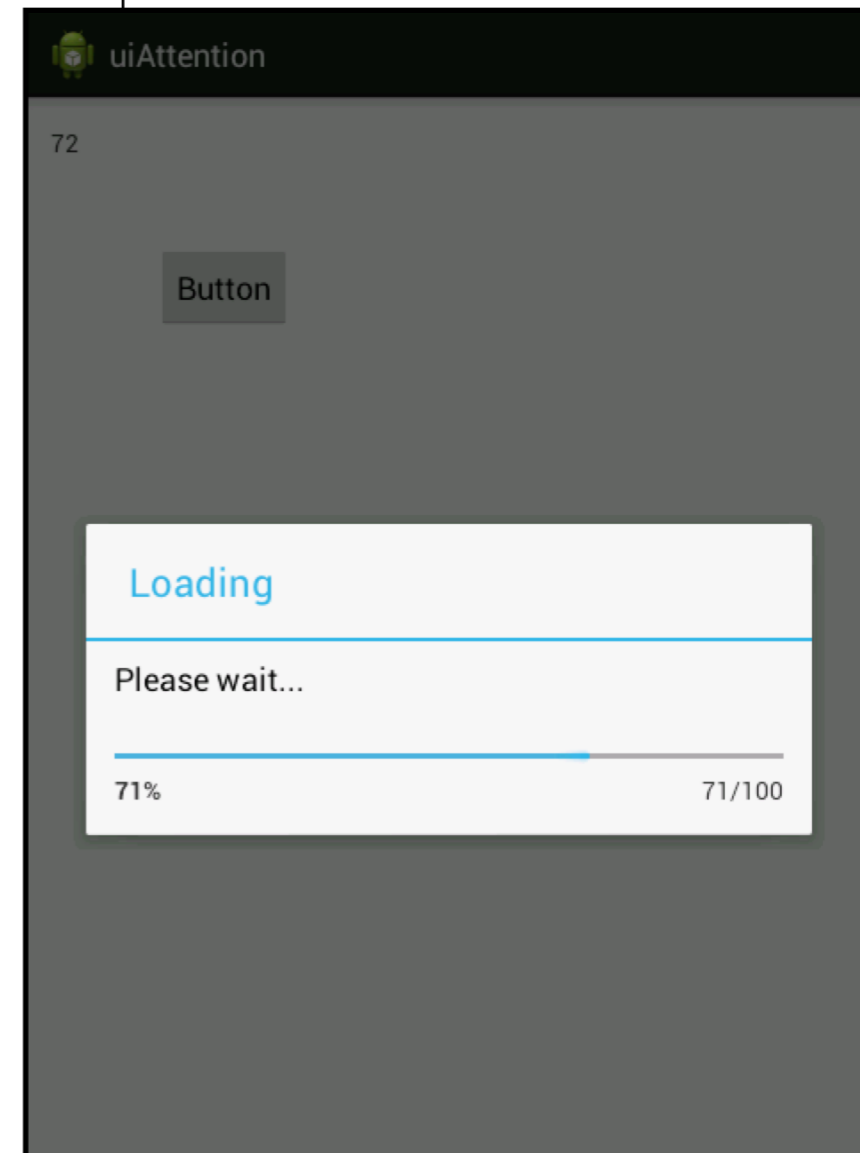
```



執行結果

有進度條樣式

```
..  
..  
pd.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL); //設置進度條樣式  
..  
..  
  
//定義countTask  
countTask = new Runnable() {  
    public void run() {  
        //延遲的動作在此task內做  
        try {  
            for(i=0;i<100;i++)  
            {  
                Thread.sleep(100);  
                pd.setProgress(i);  
                //每隔一秒透過Handler執行Update UI的工作  
                myHandler.post(updateUITask);  
            }  
            if(i==100)  
            {  
                pd.dismiss();  
                pd.setProgress(0);  
            }  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
    }  
};  
  
//定義upadteUITask  
updateUITask = new Runnable() {  
  
    @Override  
    public void run() {  
        myTxt.setText(Integer.toString(i));  
    }  
};
```



執行結果

介面元件介紹

- TextEdit
- RadioGroup & RadioButton
- CheckBox
- Toast
- AlertDialog
- SharedPreferences

Example

- TextView提示，EditText輸入名字，Button送出，用Toast顯示名字
- 用RadioGroup選擇年齡，用TextView顯示
- 用CheckBox選擇喜歡的食物並顯示
- 按下儲存鈕紀錄名字，用AlertDialog確定並儲存

請輸入名字

Jimmy

Done

age 20~25

age 26~30

你的年齡約為

age 26~30

喜歡的食物為

漢堡可樂

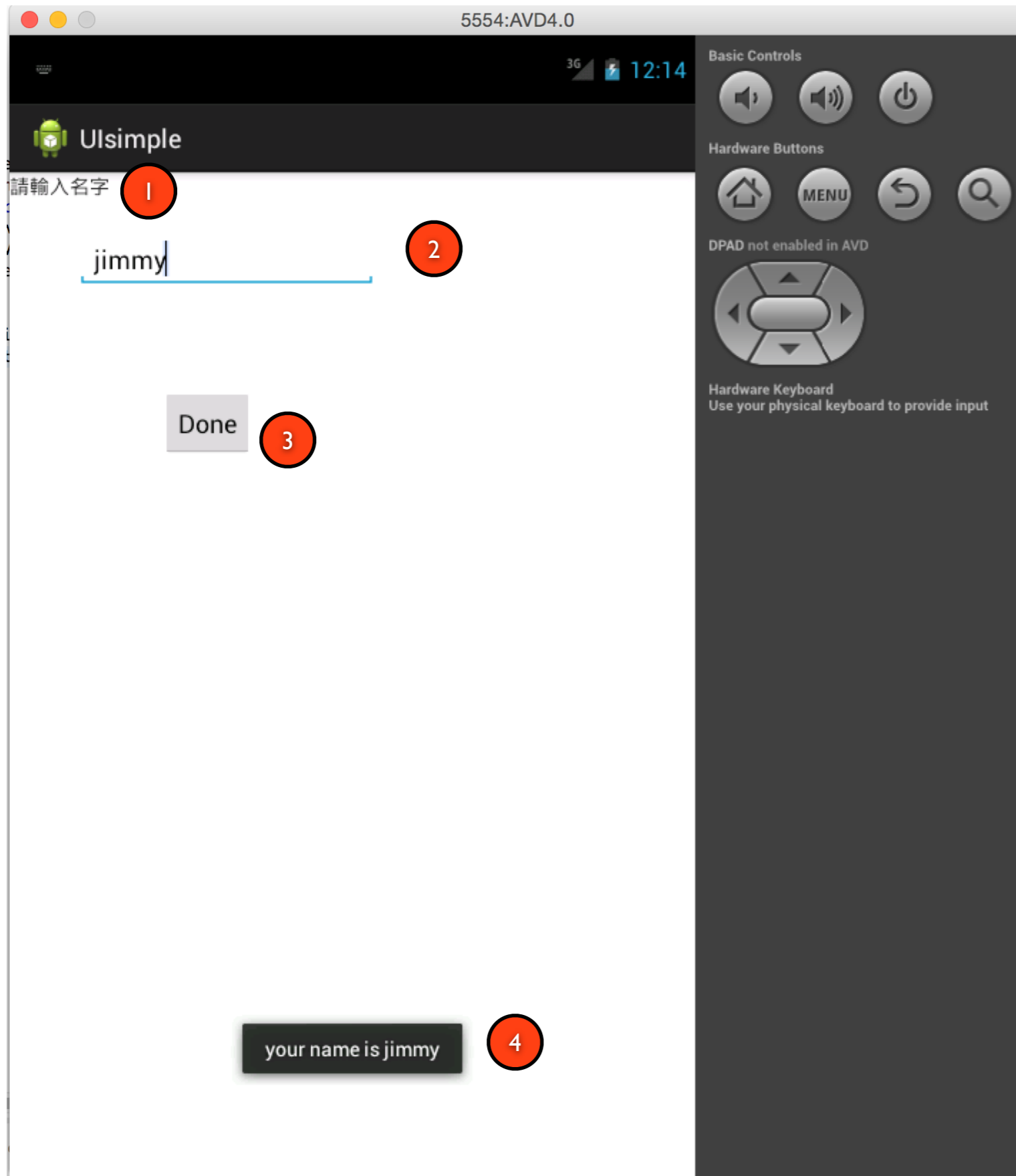
漢堡

可樂

Save

最後結果

簡單介面練習



- TextView : 提示
- TextEdit: 輸入名字
- Button: 按鈕送出
- Toast: 顯示名字

程式碼

```
public class MainActivity extends Activity {  
  
    private Button myButton;  
    private EditText myEditText;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        myButton=(Button) findViewById(R.id.button1);  
        myEditText=(EditText)findViewById(R.id.editText1);  
        myButton.setOnClickListener(new View.OnClickListener() {  
  
            @Override  
            public void onClick(View v) {  
                Toast.makeText(getBaseContext(), "your name is "+ myEditText.getText().toString() ,  
                    Toast.LENGTH_LONG).show();  
            }  
        });  
    }  
}
```

匿名的Listener

使用Toast

EditText：作為輸入的元件

- 常用屬性
 - Input Type: 鍵盤型態
 - MaxLength: 最多文字
 - Lines: 最多顯示的列數
 - Ems: 每列的字元數
 - Editable: 可否編輯

Edit可能的型態

abc

Firstname Lastname

.....

1...2...3

user@domain

(555) 0100

Address

Postal Address
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor

12:00am

1/1/2011

42

-42

42.0

AutoComplete

MultiAutoComplete

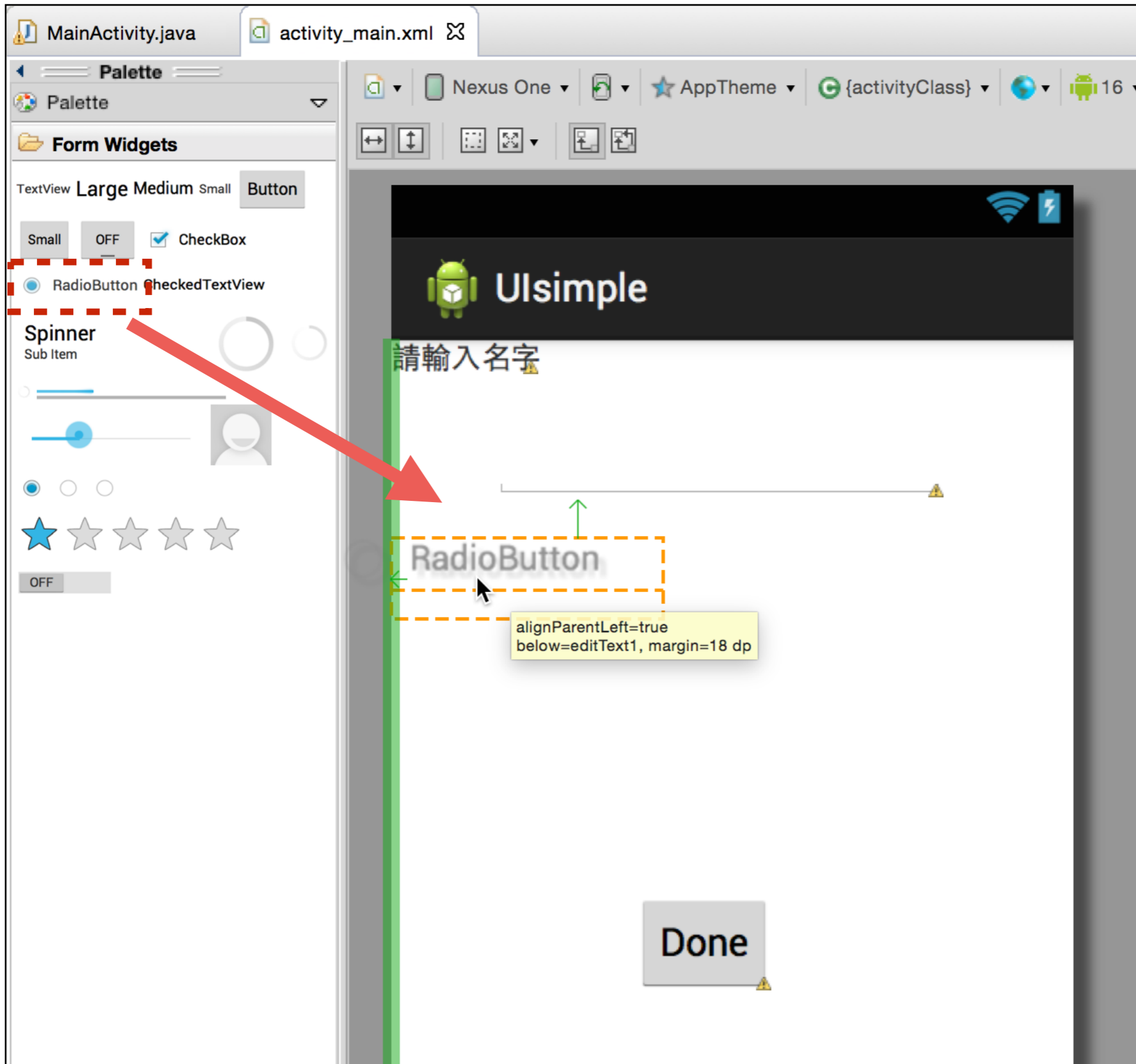
Toast：顯示訊息元件

- 使用方法：
 - `Toast.makeText(Context, 訊息, Time).show()`
- Time
 - `Toast.LENGTH_LONG` 較長時間
 - `Toast.LENGTH_SHORT` 較短時間
- `setGravity` 設定顯示位置
 - `Gravity.CENTER`
 - `Gravity.BOTTOM`

多選一的

RadioButton 與RadioGroup

- RadioGroup裏的RadioButton中，只能選擇一個
- 若沒有RadioGroup選項則手動加入
- 每個元件必定要有width及height
- 設定RadioGroup的Listener
- 在Listener中判斷選擇了哪個RadioButton



MainActivity.java

activity_main.xml

Palette

Palette

Form Widgets

TextView Large Medium Small Button

Small OFF CheckBox

RadioButton CheckedTextView

Spinner

Sub Item

OFF

Nexus One AppTheme {activityClass} 16

Uisimple

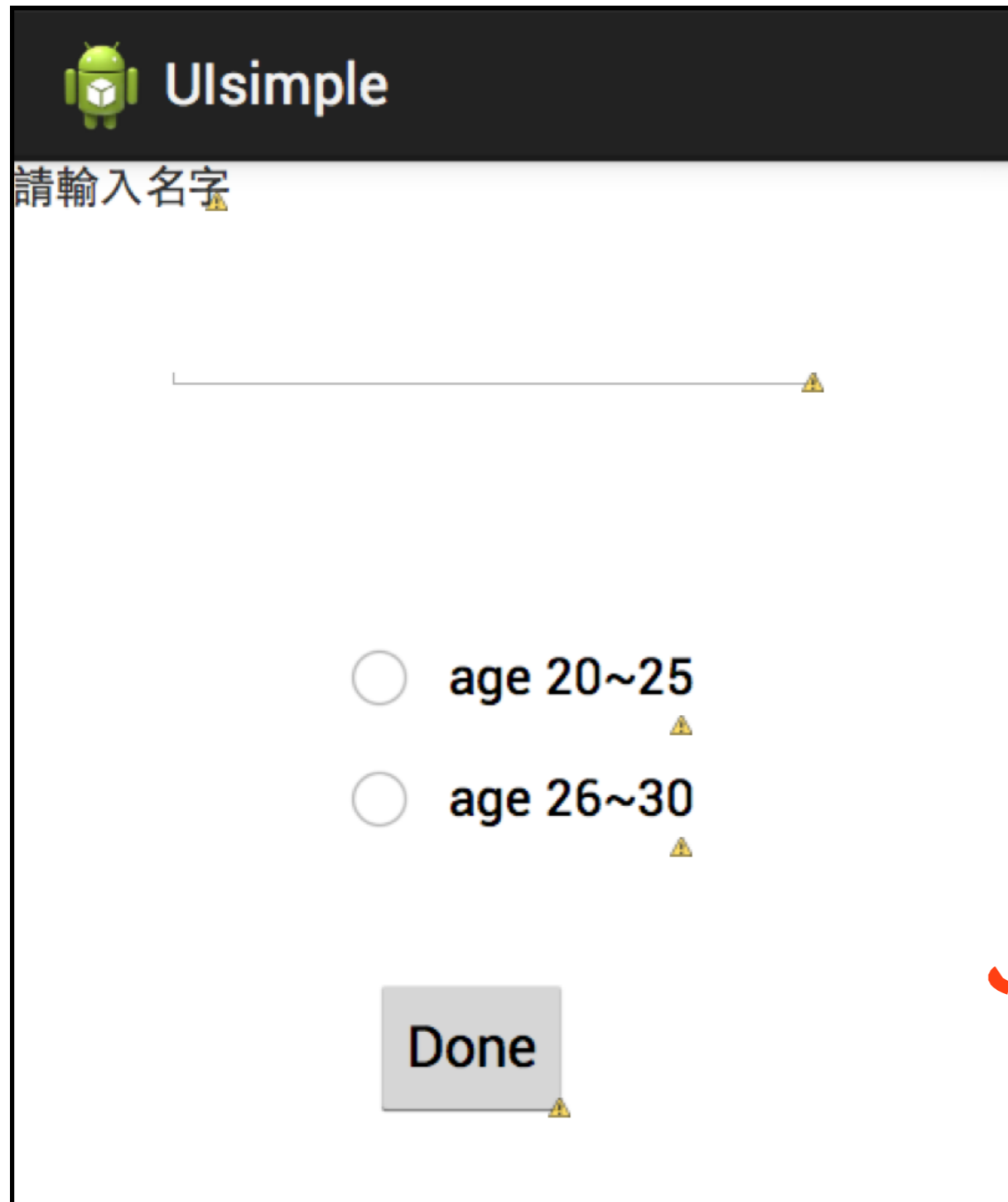
請輸入名字

RadioButton

alignParentLeft=true
below=editText1, margin=18 dp

Done

完成如下Layout並設定RadioGroup的文字



```
radioButton  
    android:id="@+id/radioButton1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@+id/editText1"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="70dp"  
    android:text="age 20~25" />
```

加入RadioGroup

找不到可以由XML檔直接加入

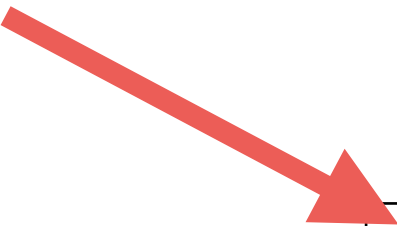
```
android:layout_marginBottom="154dp"  
android:layout_marginLeft="44dp"  
android:layout_toRightOf="@+id/textView1"  
android:text="Done" />
```

```
<RadioButton  
  android:id="@+id/radioButton1"  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:layout_below="@+id/editText1"  
  android:layout_centerHorizontal="true"  
  android:layout_marginTop="70dp"  
  android:text="age 20~25" />
```

```
<RadioButton  
  android:id="@+id/radioButton2"  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:layout_alignLeft="@+id/radioButton1"  
  android:layout_below="@+id/radioButton1"  
  android:text="age 26~30" />
```

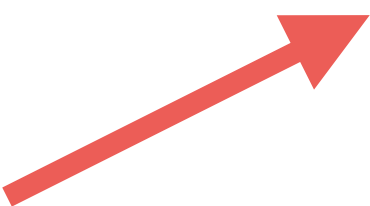
```
</RelativeLayout>
```

RadioGroup需要包
住RadioButton才可
管控

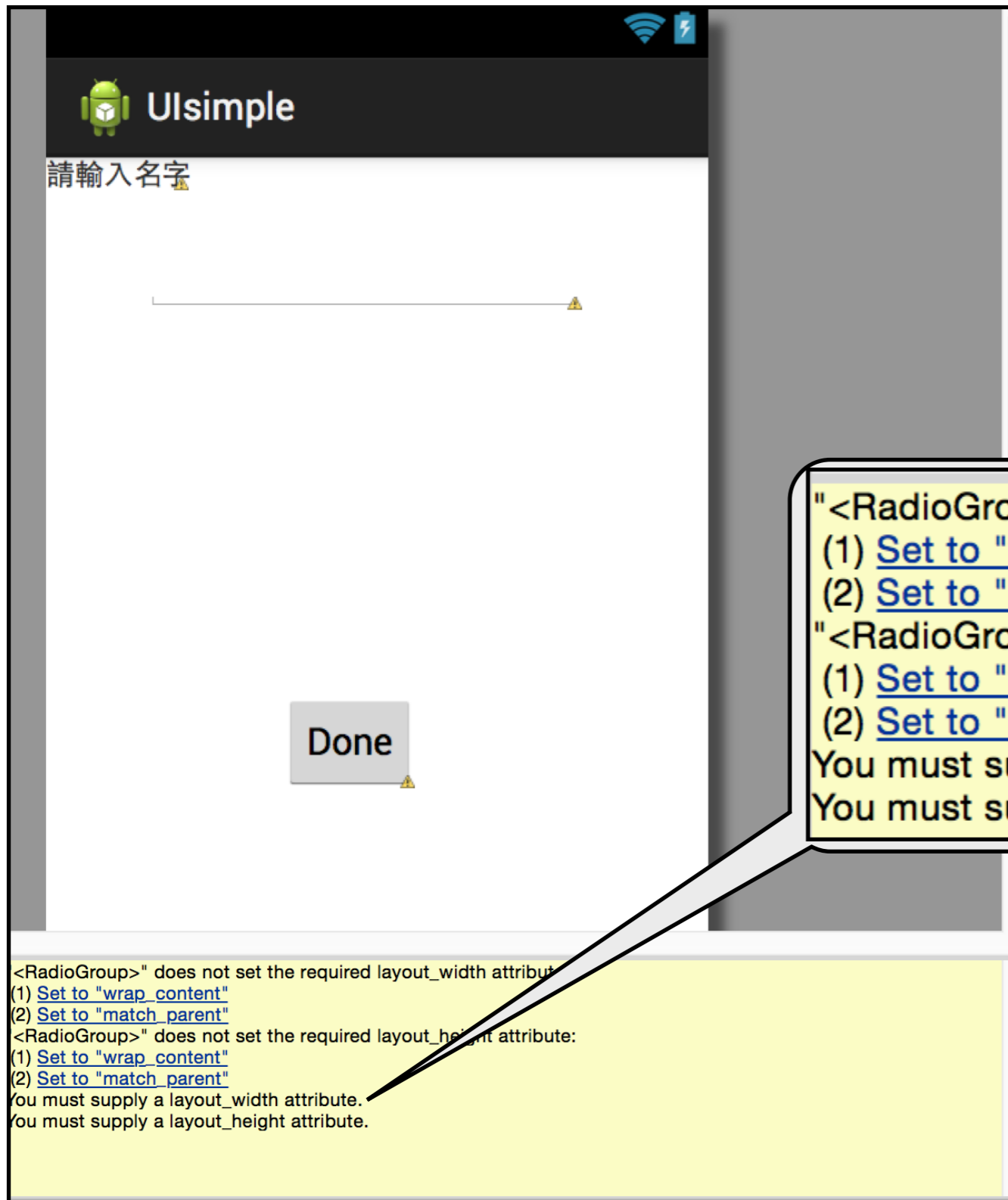


```
<RadioGroup >
  <RadioButton
    android:id="@+id/radioButton1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/editText1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="70dp"
    android:text="age 20~25" />

  <RadioButton
    android:id="@+id/radioButton2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/radioButton1"
    android:layout_below="@+id/radioButton1"
    android:text="age 26~30" />
</RadioGroup>
```

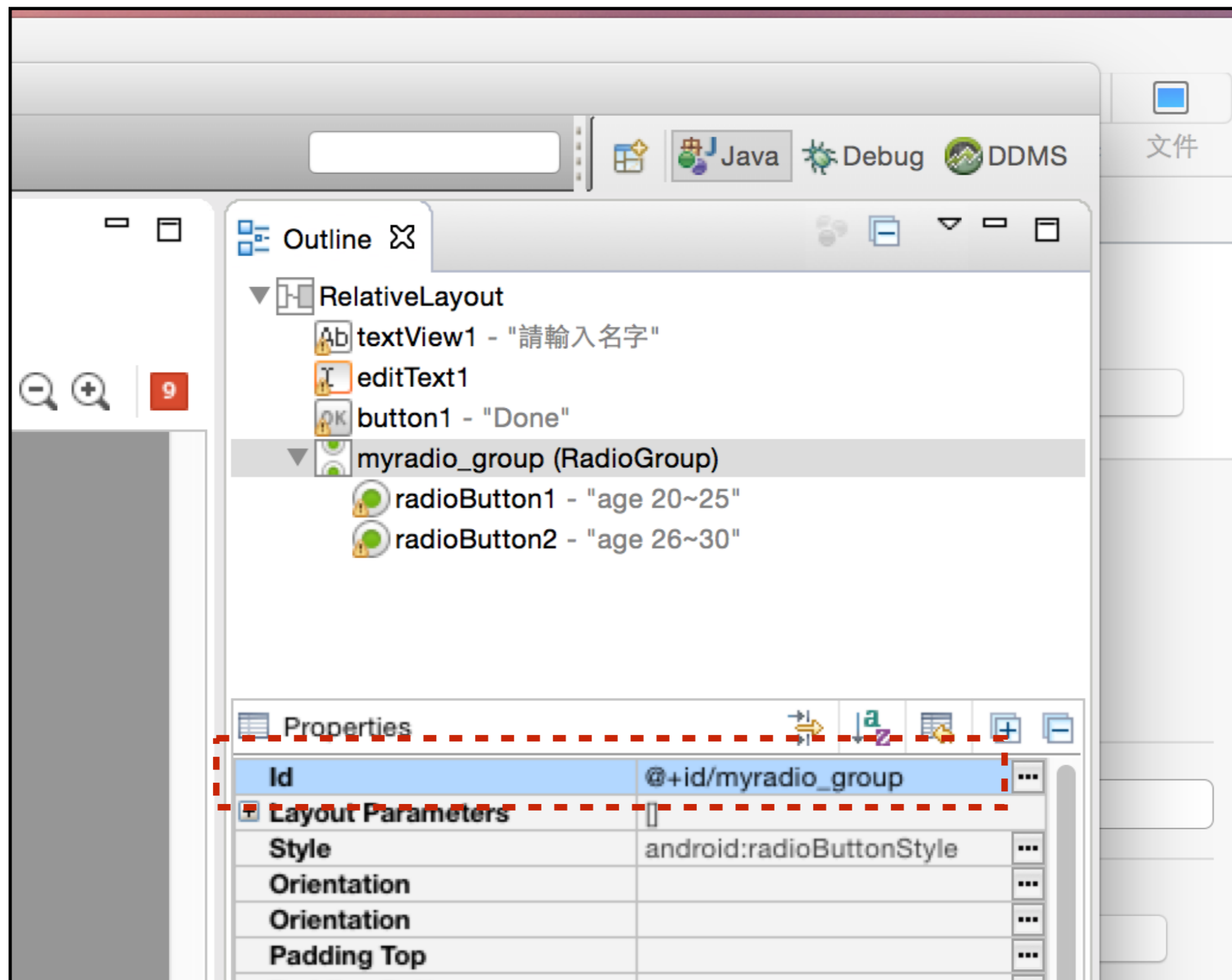


切換到Graphical Layout



需要設定Width及Height

設定RadioGroup的id為myradio_group



加入年齡顯示的TextView



```
private Button myButton;
private EditText myEditText;
private RadioGroup myRadioGroup;
private RadioButton RBage20_25, RBage26_30;
private TextView ageTextView;
```

設定變數

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    myButton=(Button) findViewById(R.id.button1);
    myEditText=(EditText)findViewById(R.id.editText1);
    ageTextView=(TextView)findViewById(R.id.textView3); //顯示年紀的TextView
```

對應id元件

```
myRadioGroup=(RadioGroup)findViewById(R.id.myradio_group);
RBage20_25=(RadioButton)findViewById(R.id.radioButton1);
RBage26_30=(RadioButton)findViewById(R.id.radioButton2);
```

```
myRadioGroup.setOnCheckedChangeListener(new OnCheckedChangeListener() {
```

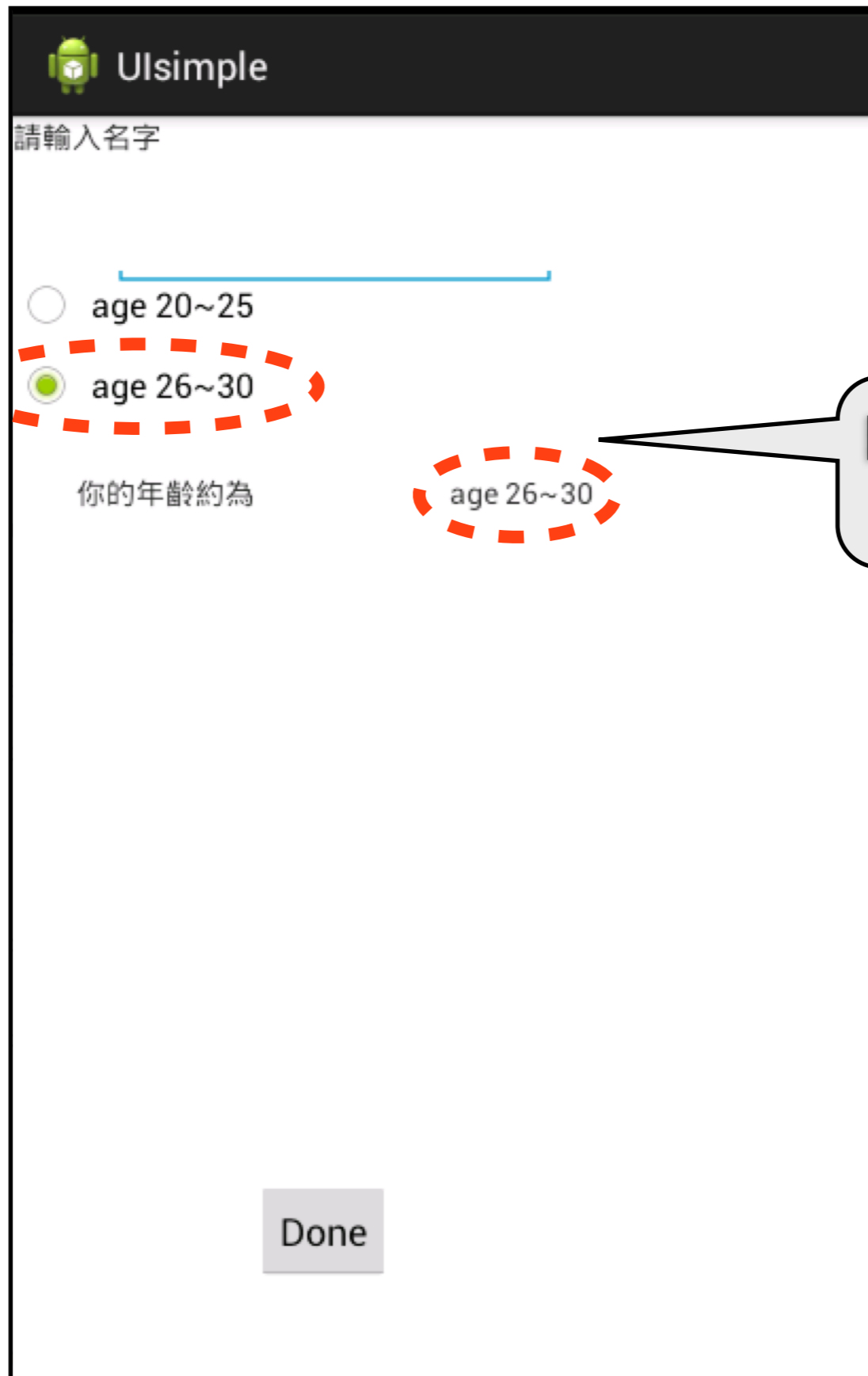
設定ChangeListener

```
@Override
public void onCheckedChanged(RadioGroup group, int checkedId) {
    switch (checkedId) {
        case R.id.radioButton1:
            ageTextView.setText(RBage20_25.getText().toString());
            break;
        case R.id.radioButton2:
            ageTextView.setText(RBage26_30.getText().toString());
            break;

        default:
            break;
    }
}
```

以id判斷選到哪個
radioButton

```
});
```



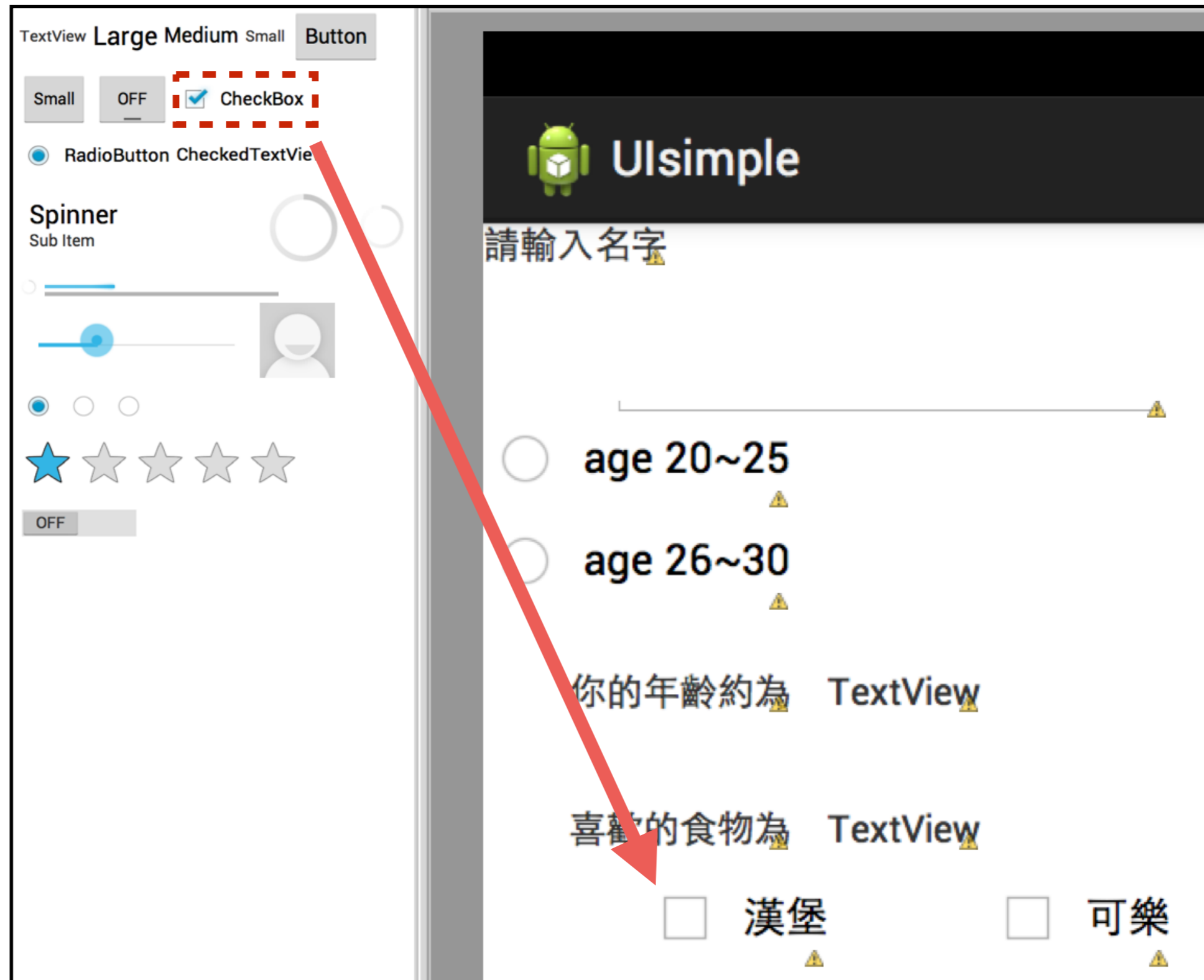
隨著選擇的RadioButton
改變

執行結果

可多選的CheckBox

- CheckBox為多選項目
- 設定CheckBox.OnCheckedChangeListener
- 用isChecked判斷是否選取

加入CheckBox並更改名稱



CheckBox程式部分

```
//for checkbox
private CheckBox CBfood1, CBfood2;
private TextView likeTextView;
private String foodStr;
..
..
CBfood1=(CheckBox)findViewById(R.id.checkBox1);
CBfood2=(CheckBox)findViewById(R.id.checkBox2);
CBfood1.setOnCheckedChangeListener(cbListener);
CBfood2.setOnCheckedChangeListener(cbListener);
..
..
private CheckBox.OnCheckedChangeListener cbListener = new CheckBox.OnCheckedChangeListener() {

    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        foodStr=""; //清除內容

        if(CBfood1.isChecked())
            foodStr=foodStr+CBfood1.getText().toString();
        if(CBfood2.isChecked())
            foodStr=foodStr+CBfood2.getText().toString();

        likeTextView.setText(foodStr);
    }

};
```

同一個Listener

isChecked()判斷是否有
勾選



Uisimple

請輸入名字

age 20~25

age 26~30

你的年齡約為

TextView

喜歡的食物為

漢堡可樂

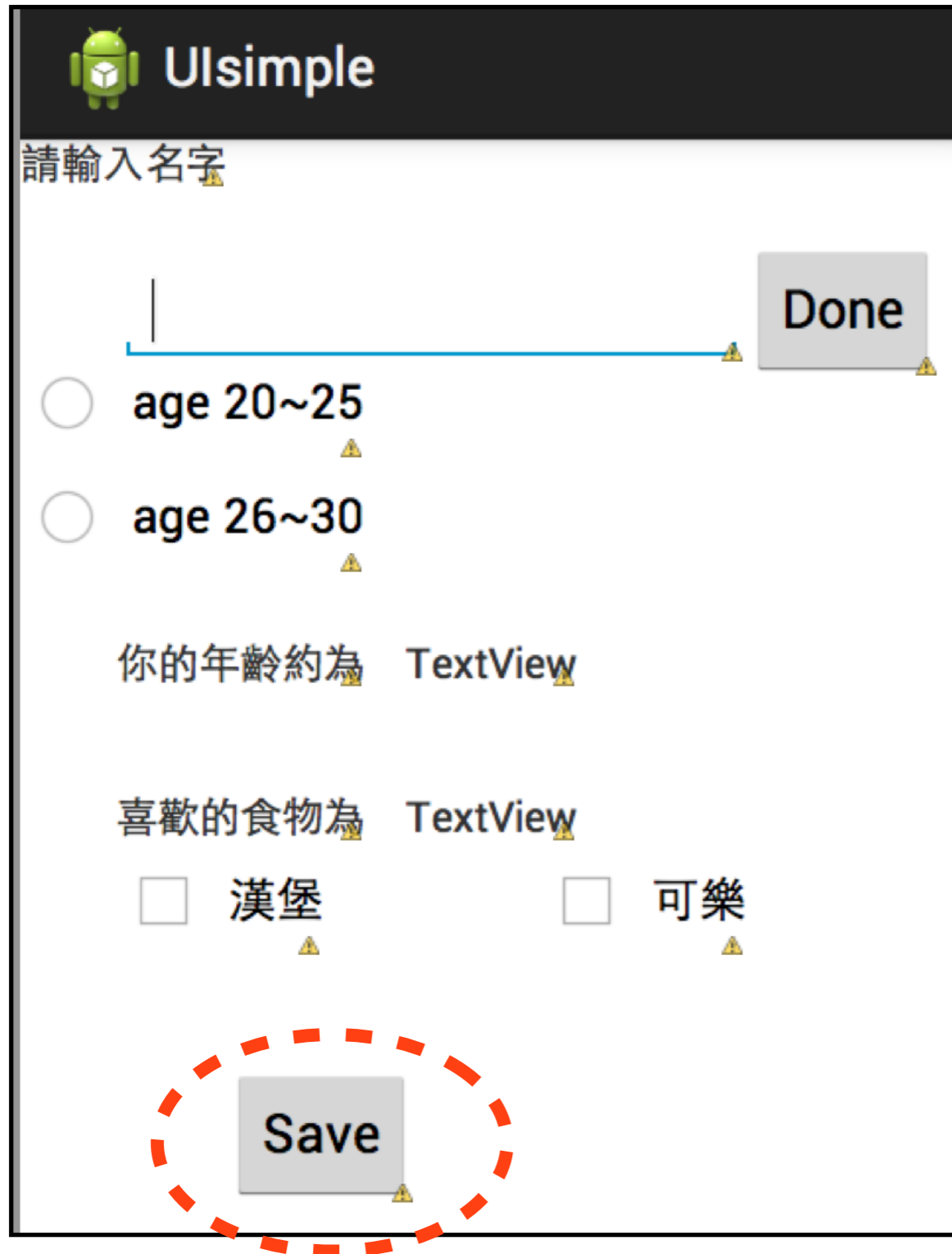
漢堡

可樂

Done

執行結果

加入Save按鈕，用AlertDialog確定是否寫入資料



加入一個Button

AlertDialog對話訊息

- 使用方法 `AlertDialog.Builder(context)`
- 常用屬性
 - `setTitle`：設定顯示title
 - `setIcon`：設定顯示圖片
 - `setMessage`：設定顯示訊息
 - `setPositiveButton`：設定確定動作
 - `setNegativeButton`：設定取消動作
 - `show`：顯示

AlertDialog程式部分

```
private Button saveButton;

//save
saveButton=(Button)findViewById(R.id.button2);

//確定儲存
saveButton.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        new AlertDialog.Builder(MainActivity.this).setTitle("寫入資料").setMessage("是否儲存")
            .setPositiveButton("確定", new OnClickListener() {

                @Override
                public void onClick(DialogInterface dialog, int which) {
                    // 寫入資料
                }
            }).setNegativeButton("取消", new OnClickListener() {

                @Override
                public void onClick(DialogInterface dialog, int which) {
                    // TODO Auto-generated method stub
                }
            }).show();
    }
});
```

確定的動作

取消的動作



執行結果

使用SharedPreferences儲存資料

- 會以key-value的形式儲存
- 儲存方式
 - 利用getSharedPreferences(檔名,權限)取得物件
- 權限: MODE_PRIVATE(只有自己app可使用)
 - Editor editor= SharedPreferences物件.edit()
 - editor.putXXX(key,value)
 - 結尾要加.commit();
- 讀取方式
 - 利用getSharedPreferences(檔名,權限)取得物件
 - 用SharedPreferences物件.getXXX(key,內定值)取得資料

Exercise

按下save鈕時，儲存姓名，在下次進入畫面時直接帶出姓名

離開App重新進入

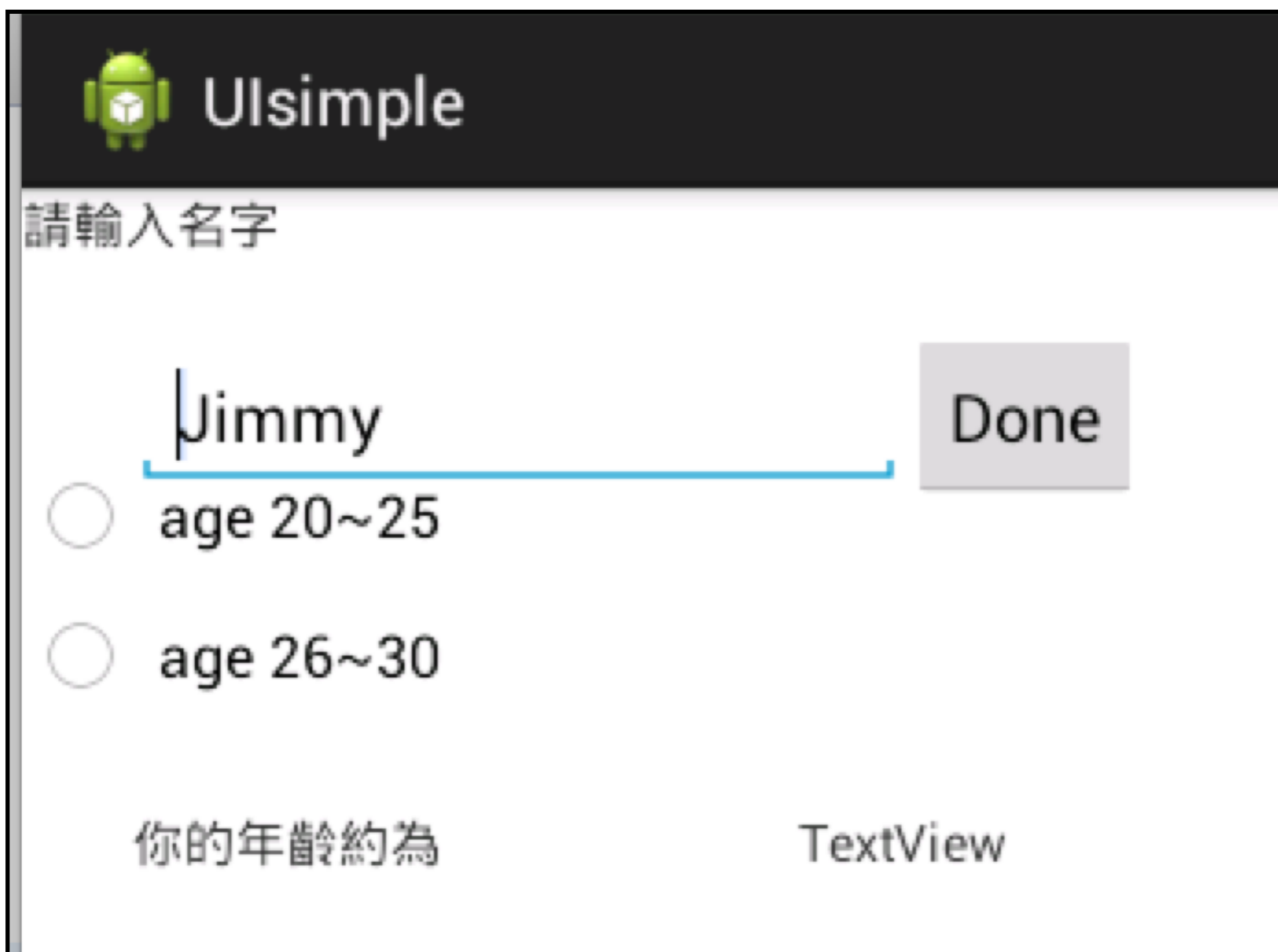


Save後按Home鍵離開



找到App再點App進入

名字自動帶入



The screenshot shows an Android application window with a black title bar containing the Android logo and the text "Uisimple". Below the title bar, the text "請輸入名字" (Please enter name) is displayed. A text input field contains the name "Jimmy" with a blue underline and a cursor. To the right of the input field is a grey button labeled "Done". Below the input field are two radio button options: "age 20~25" and "age 26~30". At the bottom of the screen, the text "你的年齡約為" (Your age is approximately) is followed by the text "TextView".

Uisimple

請輸入名字

Done

age 20~25

age 26~30

你的年齡約為 TextView

參考解答

```
//下次進入時讀取
@Override
protected void onResume() {
    SharedPreferences pref=getSharedPreferences("savename", MODE_PRIVATE);
    String nameString=pref.getString("name", "");
    myEditText.setText(nameString);
    super.onResume();
}
..
..
new AlertDialog.Builder(MainActivity.this).setTitle("寫入資料").setMessage("是否儲存").setPositiveButton("確定", new OnClickListener() {

    @Override
    public void onClick(DialogInterface dialog, int which) {
        // 寫入資料
        SharedPreferences pref=getSharedPreferences("savename",
MODE_PRIVATE);
        Editor editor=pref.edit();
        editor.putString("name", myEditText.getText().toString());
        editor.commit();
        myEditText.setText(""); //寫完後清除EditText資料
    }
})
})
```

Example: 背景播放音樂

- Service
- Spinner選擇音樂
- 播放音樂

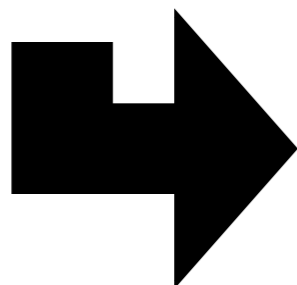
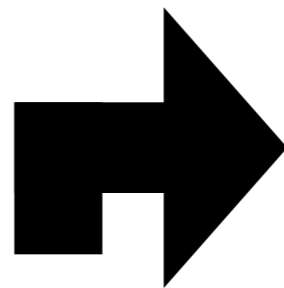
Service服務說明

- Service本身不具有畫面，用於背景執行如音樂、網路通訊
- 可以透過Handler或Broadcast與Activity溝通
- 生命週期為onCreate => onStart => onDestroy
-

專案開發實例



雲端書城

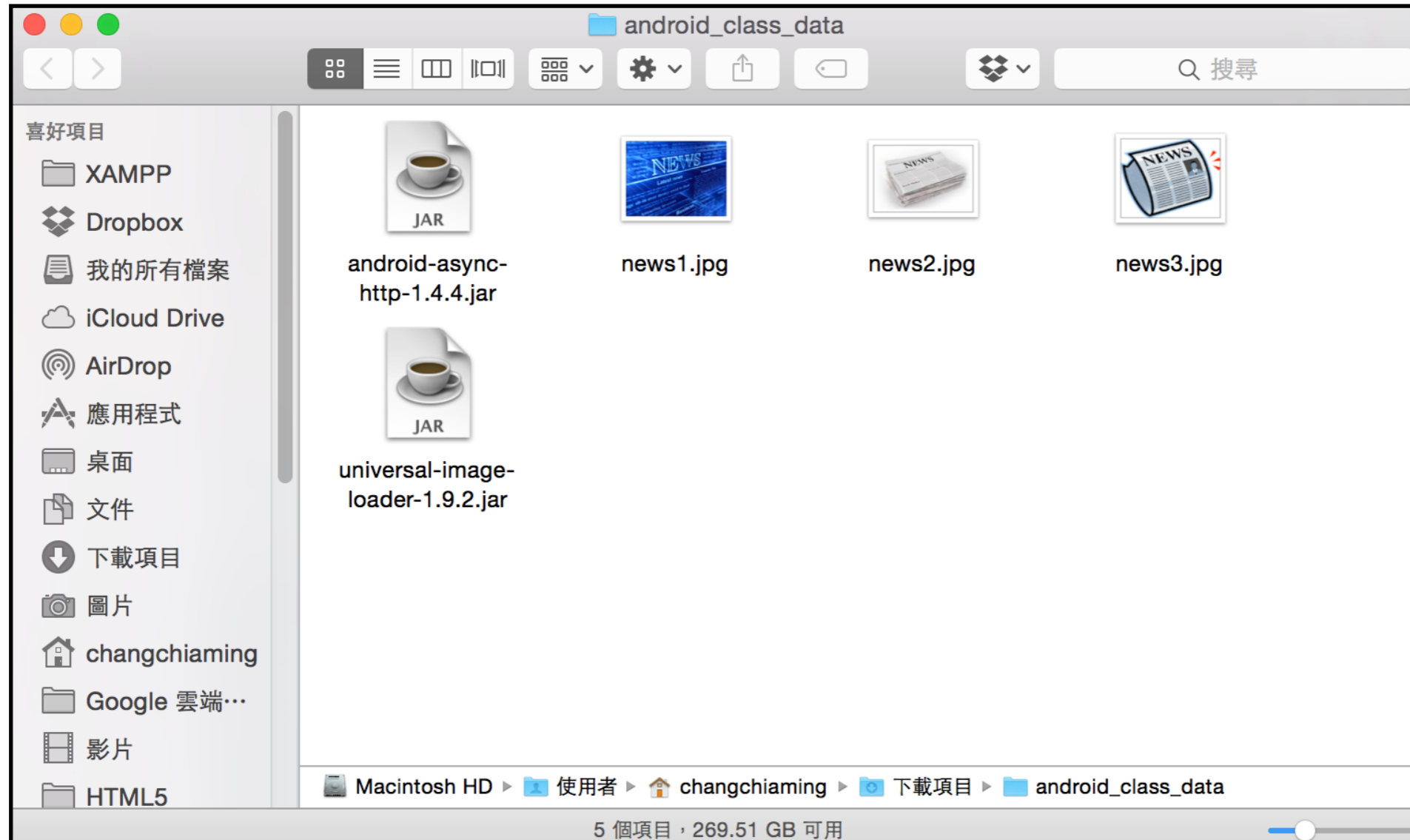


此專案將學習...

- 圖片載入(ImageView)
- 圖片滑動變換 (利用ViewFlipper)
- ListView
- ListView Cell自定layout
- ListView Cell傳遞資料並加入動作
- OpenSource使用 (網路交涉)
- 網路圖片非同步載入

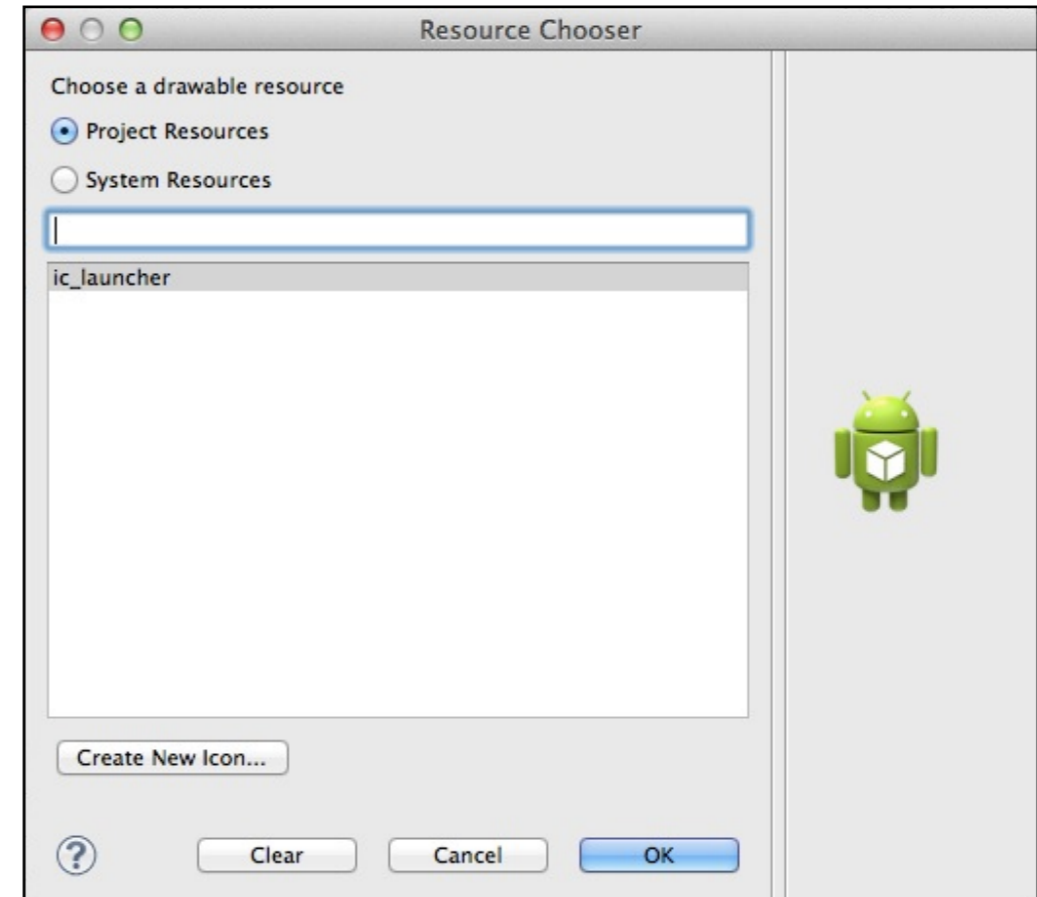
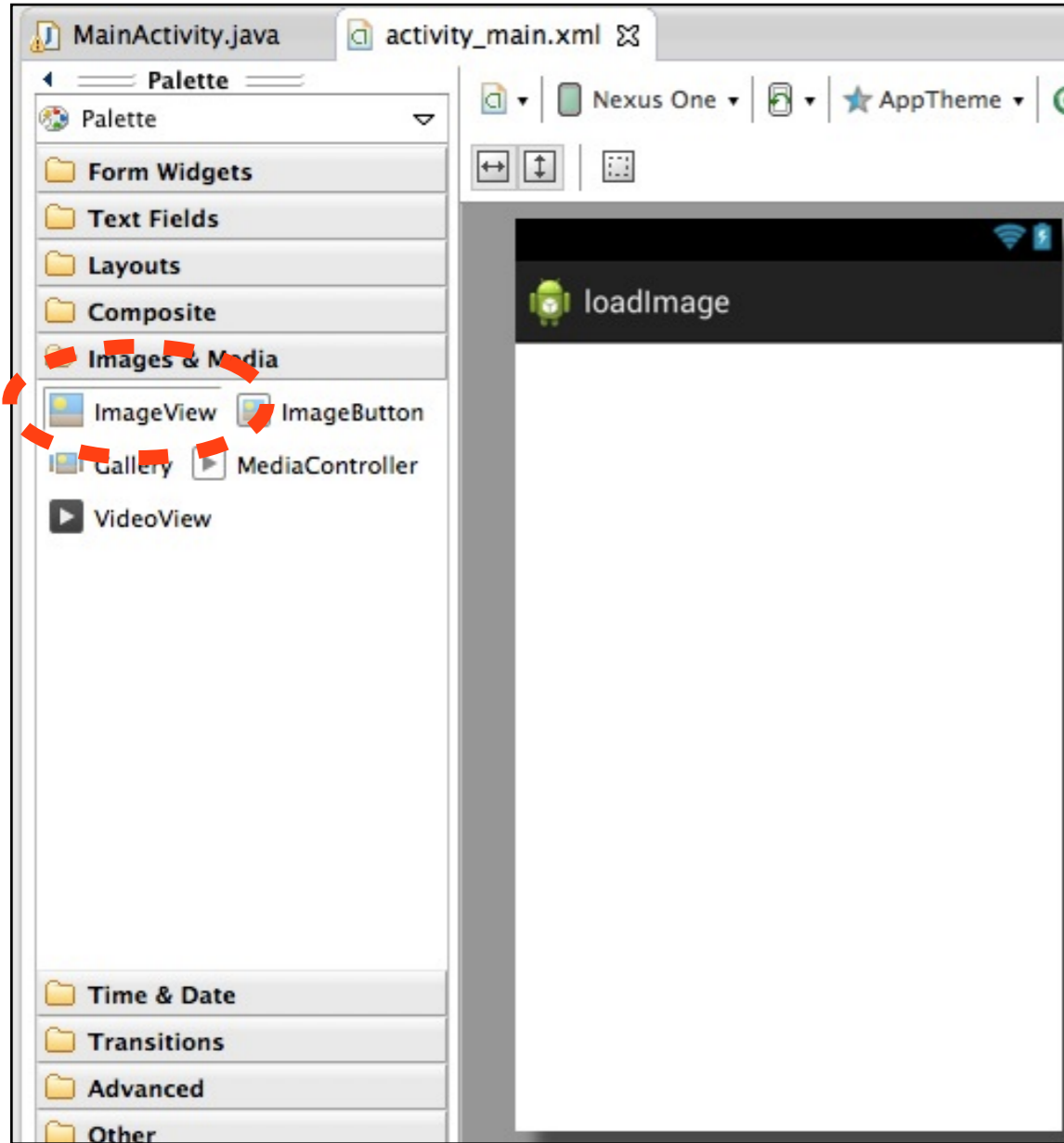
下載使用資料

- <http://goo.gl/iwnL4Y>

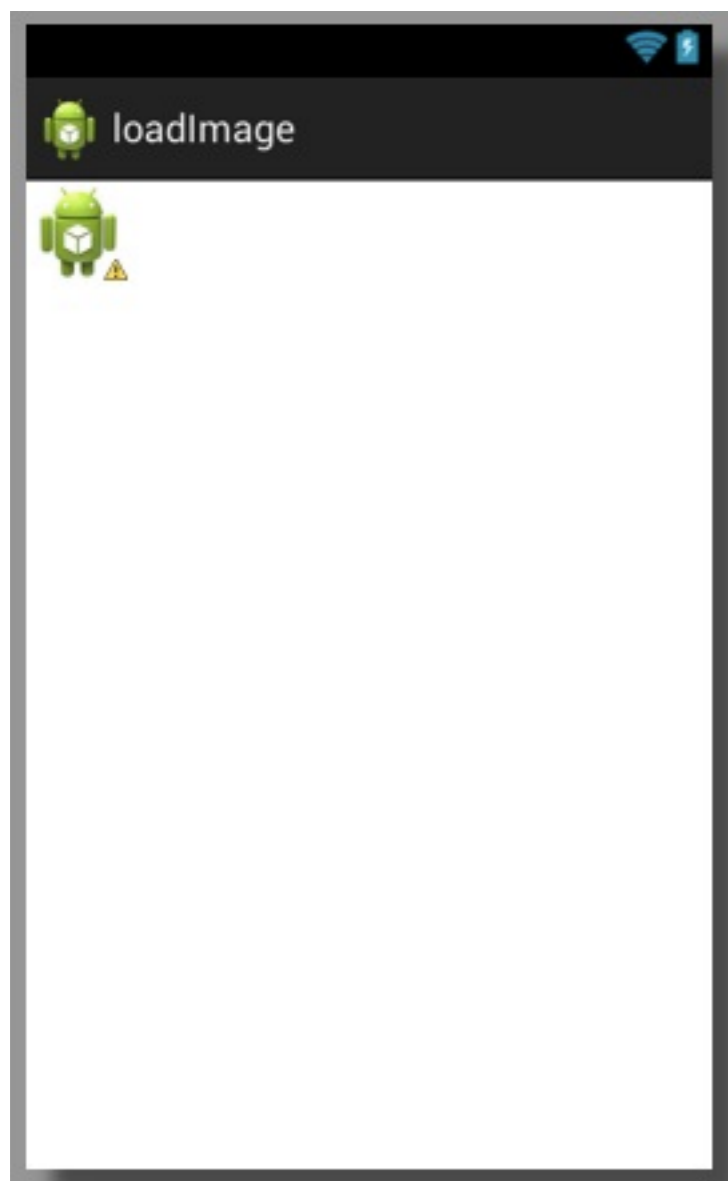


3張圖以及2個opensource

在activity_main.xml裡加入ImageView元件



新建專案LoadImage



Graphical Layout

```
MainActivity.java  *activity_main.xml
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/container"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context="com.example.loadimage.MainActivity"
  tools:ignore="MergeRootFrame" >

  <ImageView
    android:id="@+id/imageView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_launcher" />

</FrameLayout>
```

圖片來源

Text Mode

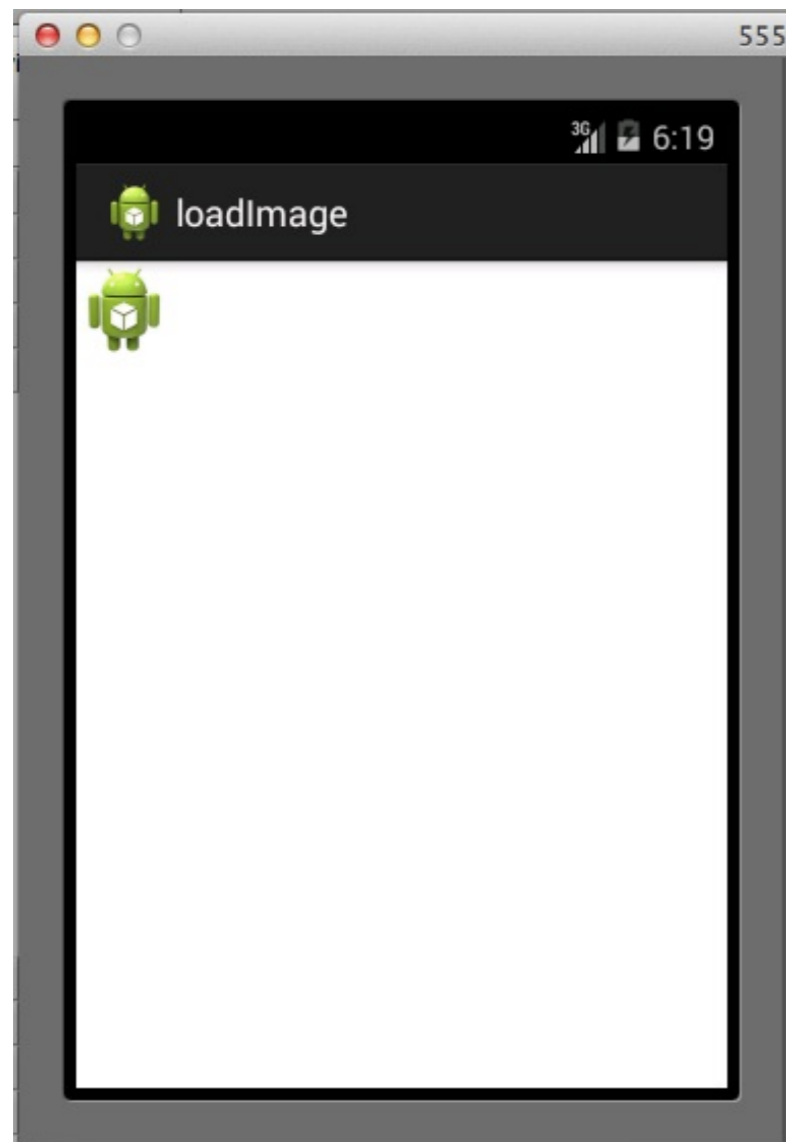


如何從程式加入圖片

使用setImageResource

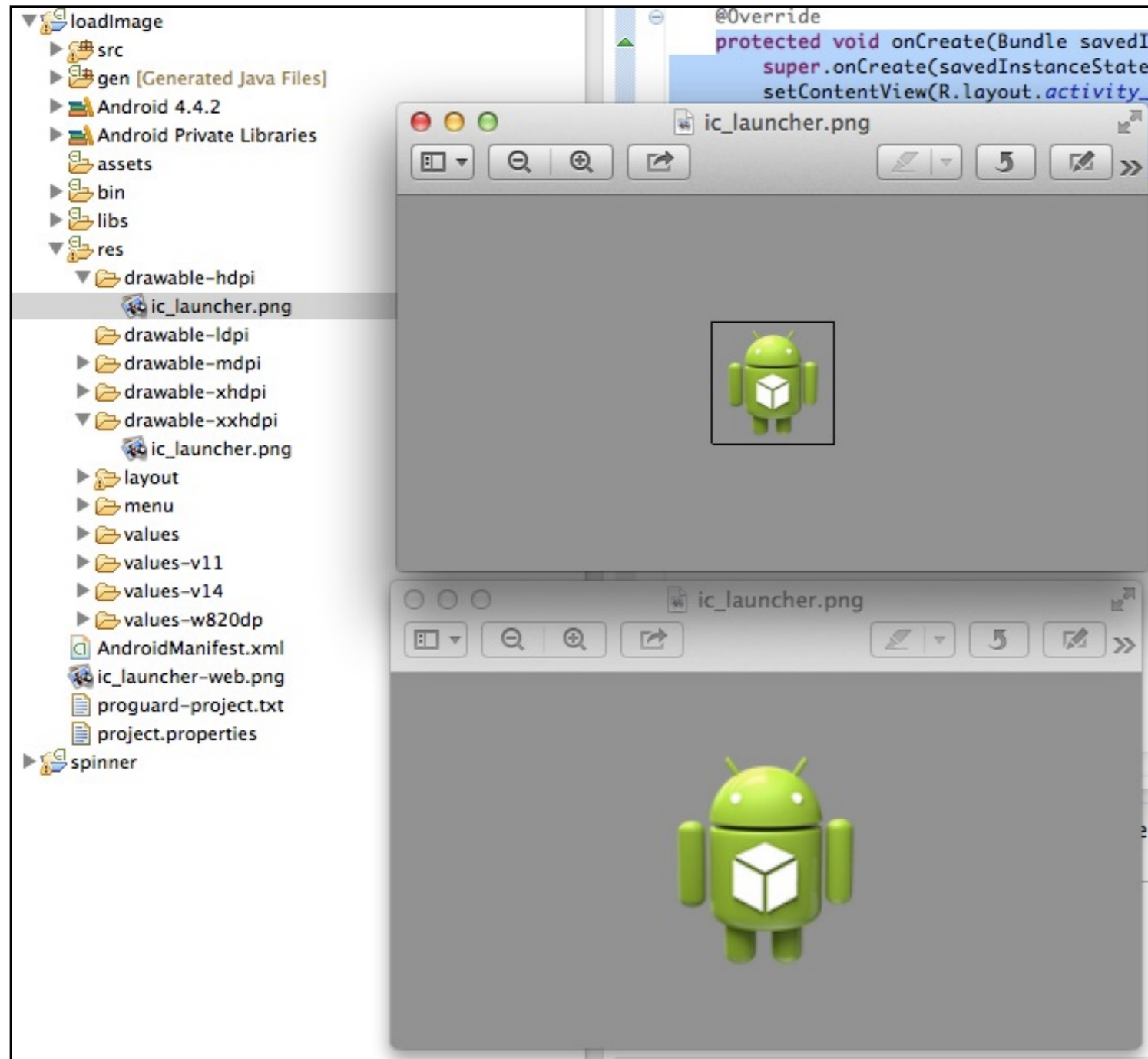
透過setImageResource指定圖片來源

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    myImageView=(ImageView) findViewById(R.id.imageView1); //取得圖片id  
    myImageView.setImageResource(R.drawable.ic_launcher);  
}
```



(執行結果)

針對不同size的drawable

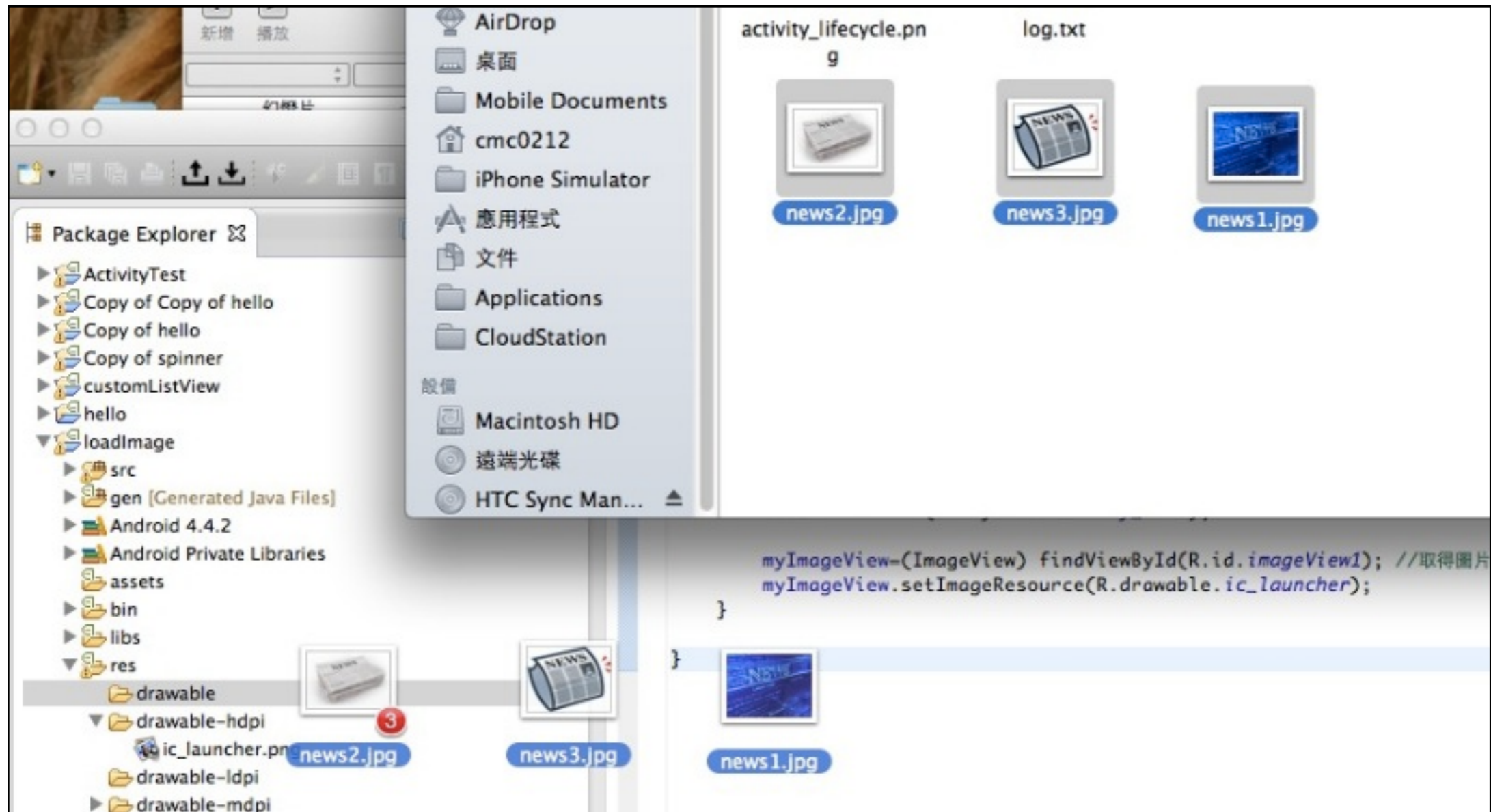


hdpi : 高解析度 【WVGA (480x800),FWVGA (480x854)】

mdpi:中解析度 【HVGA (320x480)】

ldpi: 低解析度 【QVGA (240x320)】

加入自己的圖片



1. 建立一個drawable的資料夾(共用)
2. 拖入檔案

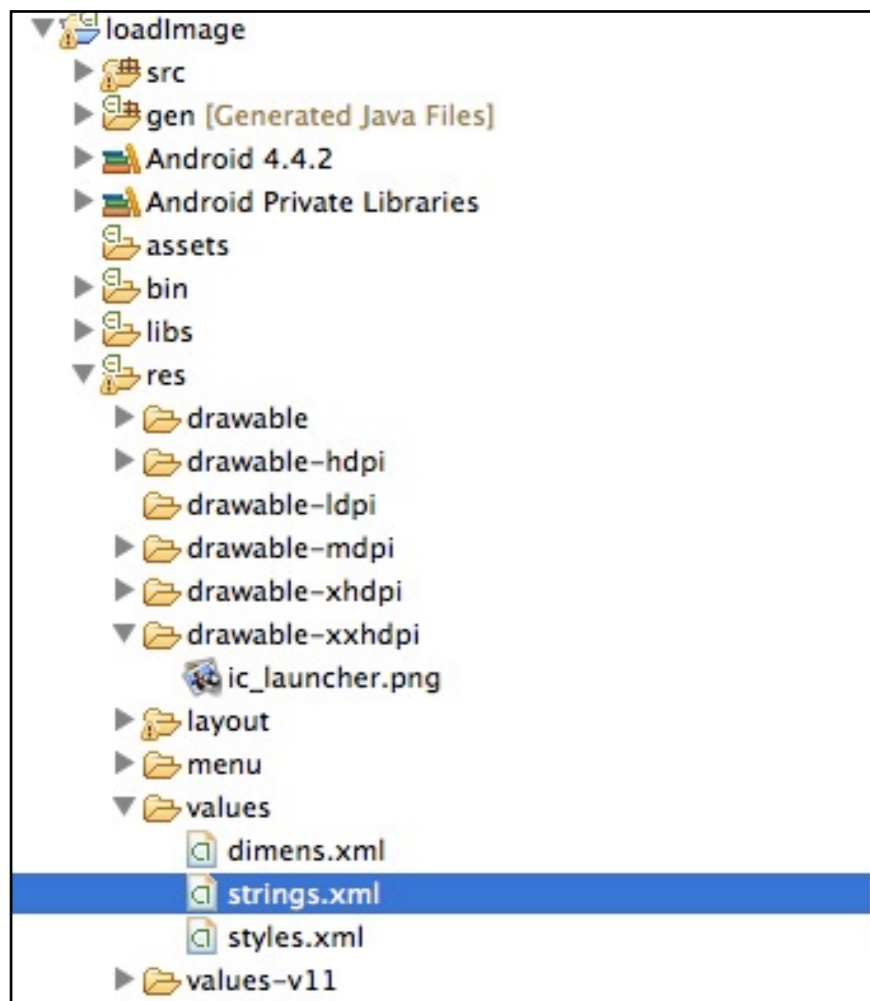
建立字串陣列

方式1: 直接新增

```
private String[] imagePics={"news1", "news2", "news3"};
```

方式2: 透過資源取得

```
imagePics= getResources().getStringArray(R.array.newsImageNames);
```



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

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

    <string-array name="newsImageNames">
        <item >"news1"</item>
        <item >"news2"</item>
        <item >"news3"</item>
    </string-array>

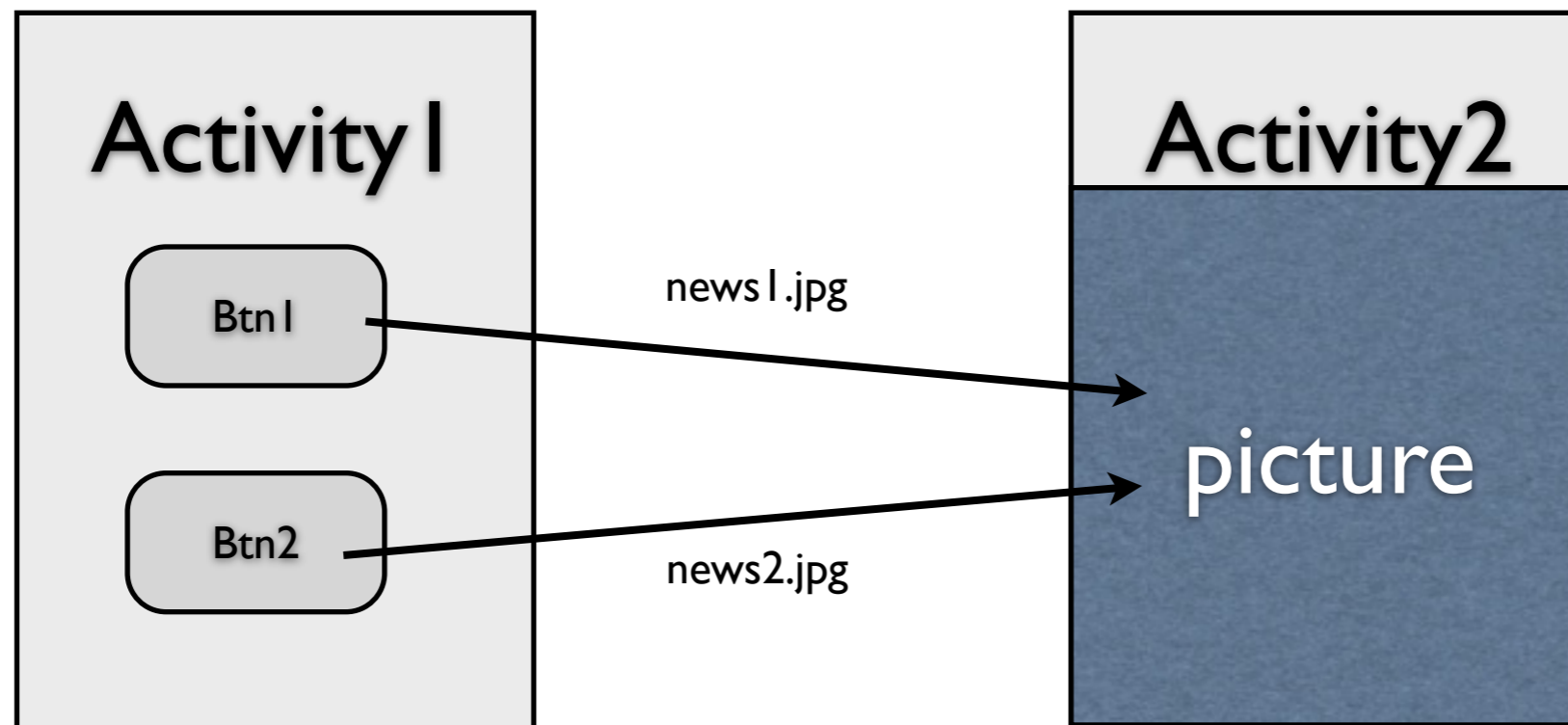
</resources>
```

在values的strings中加入string-array

Exercise

Activity 1 按按鈕+傳參數

Activity 2 顯示圖片



容易犯的錯

- Page2的setContentView加入了嗎?
- setContentView的順序錯了嗎?
- 元件取得了嗎?
- AndroidManifest.xml加入新的class名稱了嗎?

```
int id = getResources().getIdentifier("com.example.loadimage:drawable/" + imageFilename, null, null);  
myImageView.setImageResource(id);
```

Tip: 用變數動態取得資源

參考解答

```
protected void onCreate(Bundle savedInstanceState) {  
    // TODO Auto-generated method stub  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_show_image);  
    myImageView=(ImageView) findViewById(R.id.imageView1); //取得圖片id  
    Intent myIntent=getIntent();  
    String imageFilename=myIntent.getStringExtra("filename");  
    //利用資源getResources取得圖檔，不用加上副檔名  
    int id = getResources().getIdentifier("com.example.loadimage:drawable/" + imageFilename, null, null);  
    myImageView.setImageResource(id);  
}
```



如何作出翻頁效果?

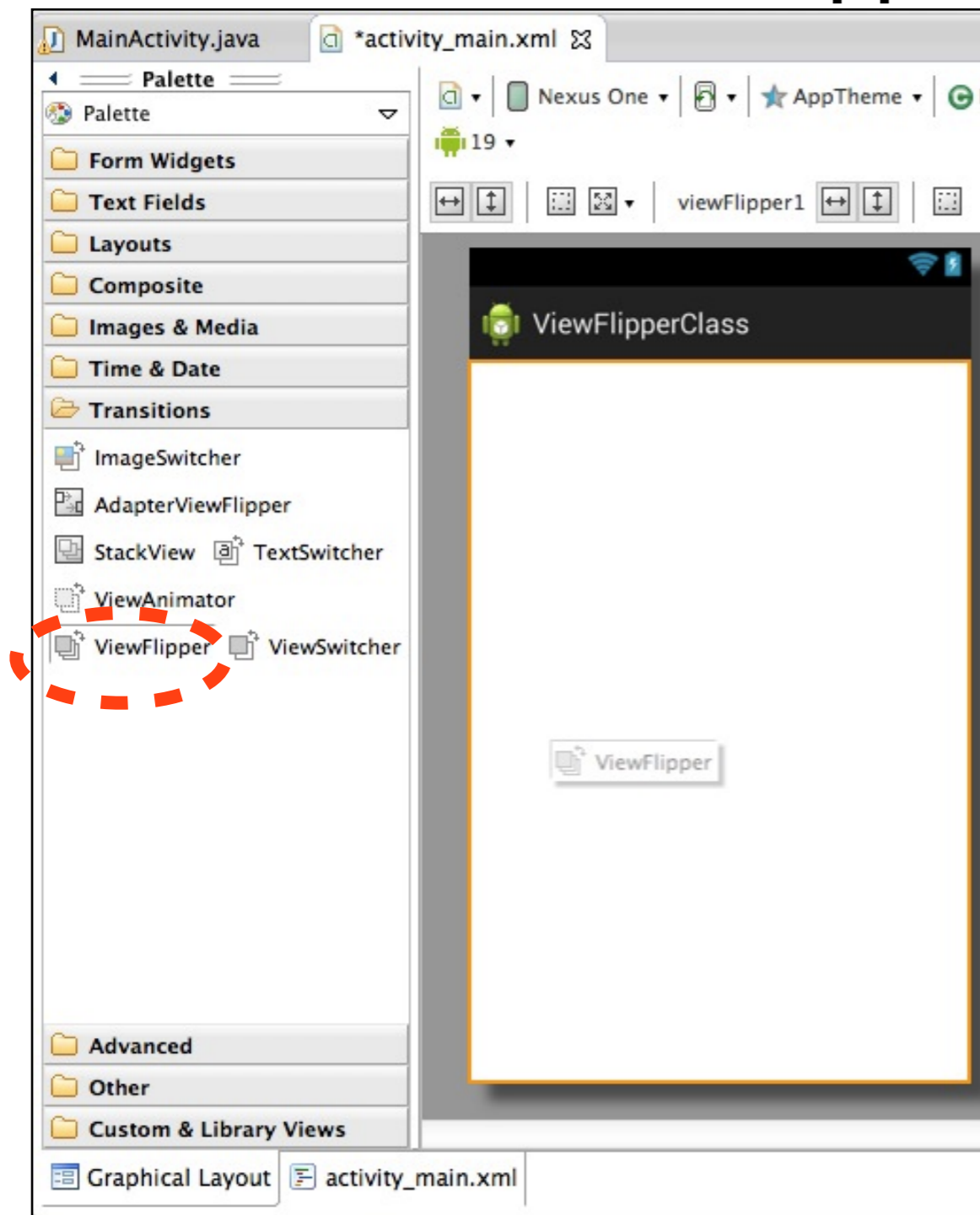
使用ViewFlipper

ViewFlipper

- ViewFlipper可以在Activity中，顯示不同的ViewGroup
- 加入的View即可透過ViewFlipper的showNext()與showPrevious()即可切換不同的View

新增ViewFlipperClass專案

- Transitions > ViewFlipper



刪除多餘的檔案與程式內容後加入ViewFlipper

加入3個ImageView，fill_parent、

The screenshot displays the Android Studio interface for editing the layout of MainActivity. The central canvas shows a preview of the 'ViewFlipperClass' with a single Android icon. The 'Palette' on the left lists various widgets, with 'ImageView' selected. The 'Outline' on the right shows the layout hierarchy: a 'container (FrameLayout)' containing a 'viewFlipper1' which contains three 'imageView' elements (imageView1, imageView2, and imageView3), all with the id 'ic_launcher'. The 'Properties' panel at the bottom right shows the configuration for the selected 'imageView3'. The 'Width' property is set to 'wrap_content', and the 'Height' property is set to 'wrap_content'. The 'Margins' section shows 'fill_parent' and 'match_parent'.

Property	Value
Id	@+id/imageView3
Layout Parameters	{} (expanded)
Gravity	
Width	wrap_content
Height	wrap_content
Margins	fill_parent, match_parent
Src	
Scale Type	
Content Description	
ImageView	{} (expanded)
Src	
Scale Type	
Adjust View Boun...	
Max Width	
Max Height	
Baseline Align Bo...	
Crop To Padding	
View	{} (expanded)

```
public class MainActivity extends Activity {
    private ViewPager myViewPager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

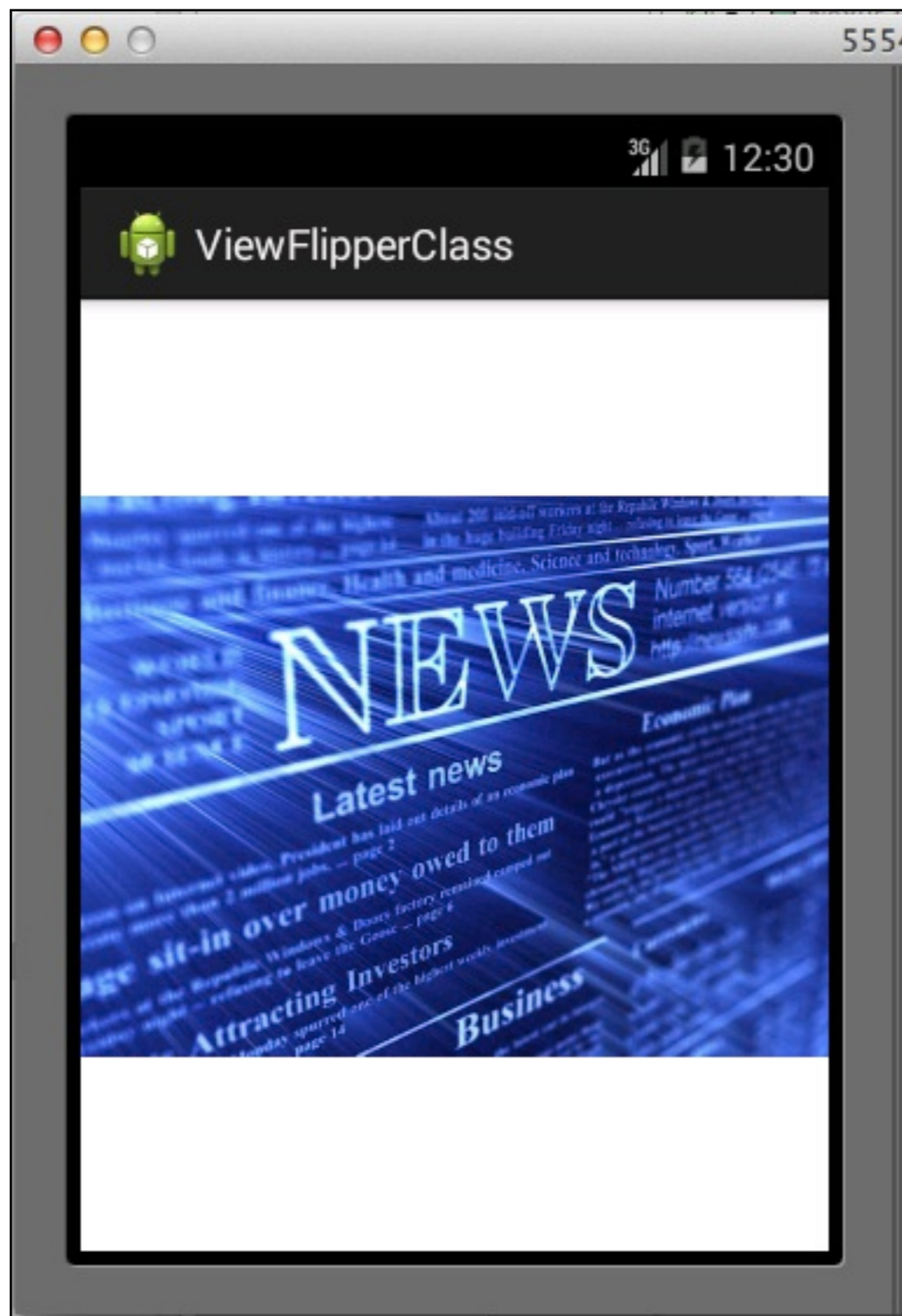
        //取得ViewPager
        myViewPager=(ViewPager) findViewById(R.id.viewPager1);
        //設定touch事件
        myViewPager.setOnTouchListener(myViewPagerListener);
    }

    OnTouchListener myViewPagerListener= new OnTouchListener() {

        @Override
        public boolean onTouch(View v, MotionEvent event) {
            // TODO Auto-generated method stub
            myViewPager.showNext();
            return false;
        }
    };
}
```

設定onTouch事

點擊就show出下一張



執行結果

手勢偵測

GestureDetector

使用自定手勢偵測步驟

- 建立自己的Detector Listener，才可用於整合目前動作行為(ViewFlipper.showNext())
- 產生Listener實體
- 產生Gesture實體並指定Listener
- 將原本的Touch事件委託給自定的Listener
- 注意! onDown必須override並return為true，才可繼續onFling動作

```
private class MyGDListener extends GestureDetector.SimpleOnGestureListener{

    @Override
    public boolean onDown(MotionEvent e) {
        return true; //return true才能讓onFling作用
    }

    @Override
    public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX,
        float velocityY) {

        if(velocityX < -10)
            myViewFlipper.showNext();
        else if(velocityX > 10)
            myViewFlipper.showPrevious();

        return super.onFling(e1, e2, velocityX, velocityY);
    }
}
```

自定Listener是為了能夠存取myViewFlipper變數

自定

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //取得ViewFlipper
    myViewFlipper=(ViewFlipper) findViewById(R.id.viewFlipper1);
    //設定touch事件
    myViewFlipper.setOnTouchListener(myViewFlipperListener);
    //產生Listener實體
    MyGDLListener myGDListener=new MyGDLListener():
    myGestureDetector=new GestureDetector(this, myGDListener);
}
```

產生Listener實體

產生Detector實體並指定Listener

```
OnTouchListener myViewFlipperListener= new OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        return myGestureDetector.onTouchEvent(event); //將事件處理委託給myGestureDetector
    }
};
```

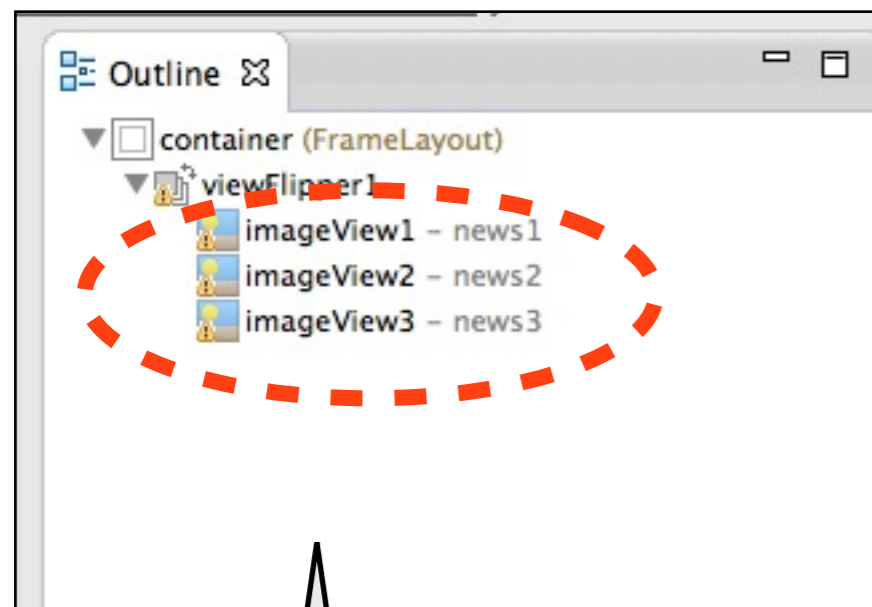
onTouch動作時，交由Detector處



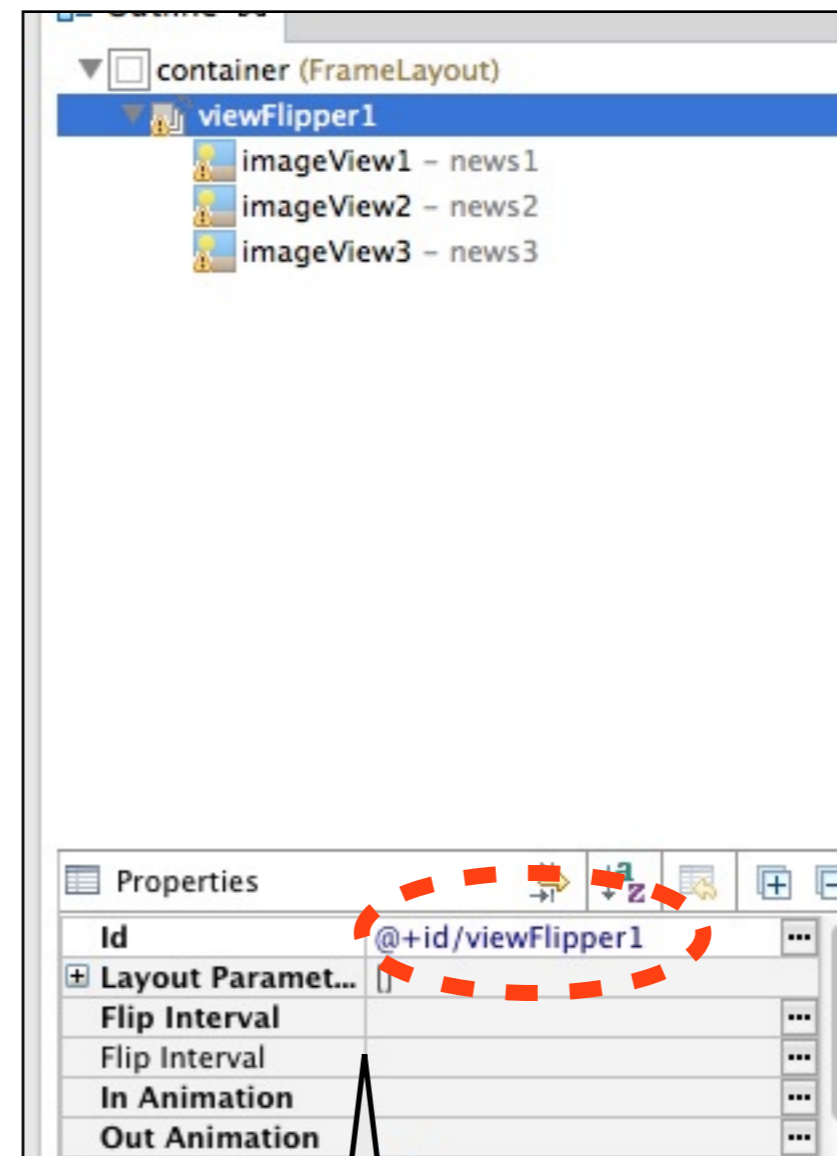
如果我要加入很多頁
呢?

動態產生ImageView

移除ImageView，ViewFlipper設定id



移除ImageView



記得id，我們才知道要加到哪個View

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    //取得ViewFlipper  
    myViewFlipper=(ViewFlipper) findViewById(R.id.viewFlipper1);  
    //設定touch事件  
    myViewFlipper.setOnTouchListener(myViewFlipperListener);  
    //產生Listener實體  
    MyGDListener myGDListener=new MyGDListener();  
    myGestureDetector=new GestureDetector(this, myGDListener);  
    ImageView myImage1= new ImageView(this);  
        myImage1.setImageResource(R.drawable.news1);  
        myViewFlipper.addView(myImage1);  
}
```

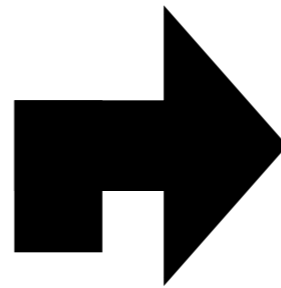
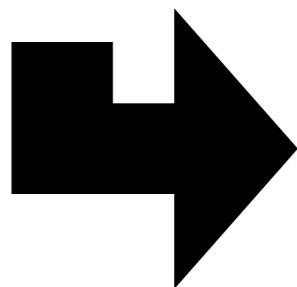
透過new ImageView產生實體，再加
到myViewFlipper裡面

設定一個Resource的陣列

```
int []myImageResource={R.drawable.news1,R.drawable.news2,R.drawable.news3};
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //取得ViewFlipper
    myViewFlipper=(ViewFlipper) findViewById(R.id.viewFlipper1);
    //設定touch事件
    myViewFlipper.setOnTouchListener(myViewFlipperListener);
    //產生Listener實體
    MyGDListener myGDListener=new MyGDListener();
    myGestureDetector=new GestureDetector(this, myGDListener);
    for (int i=0;i<3;i++)
    {
        ImageView tmpImageView=new ImageView(this);
        tmpImageView.setImageResource(myImageResource[i]);
        myViewFlipper.addView(tmpImageView);
    }
}
```

用迴圈動態新增ImageView到myViewFlipper



已完成

List View 探討

(iOS 的 TableView)

List View

- ListView相當於iOS的TableView
- 使用ListView必須透過Adapter(介面)完成資料設定
- ListView也有內建的Layout，讓使用者可以快速的呈現資料
- 常搭配ArrayList (可視為iOS的NSMutableArray)
- 常搭配HashMap<key,value> (可視為iOS的NSMutableDictionary)

BaseAdapter是抽象類別(Abstract)，需要被繼承使用，但是彈性最大

```
public abstract class
BaseAdapter
extends Object
implements ListAdapter SpinnerAdapter

java.lang.Object
↳ android.widget.BaseAdapter

▶ Known Direct Subclasses
  ArrayAdapter<T>, CursorAdapter, SimpleAdapter

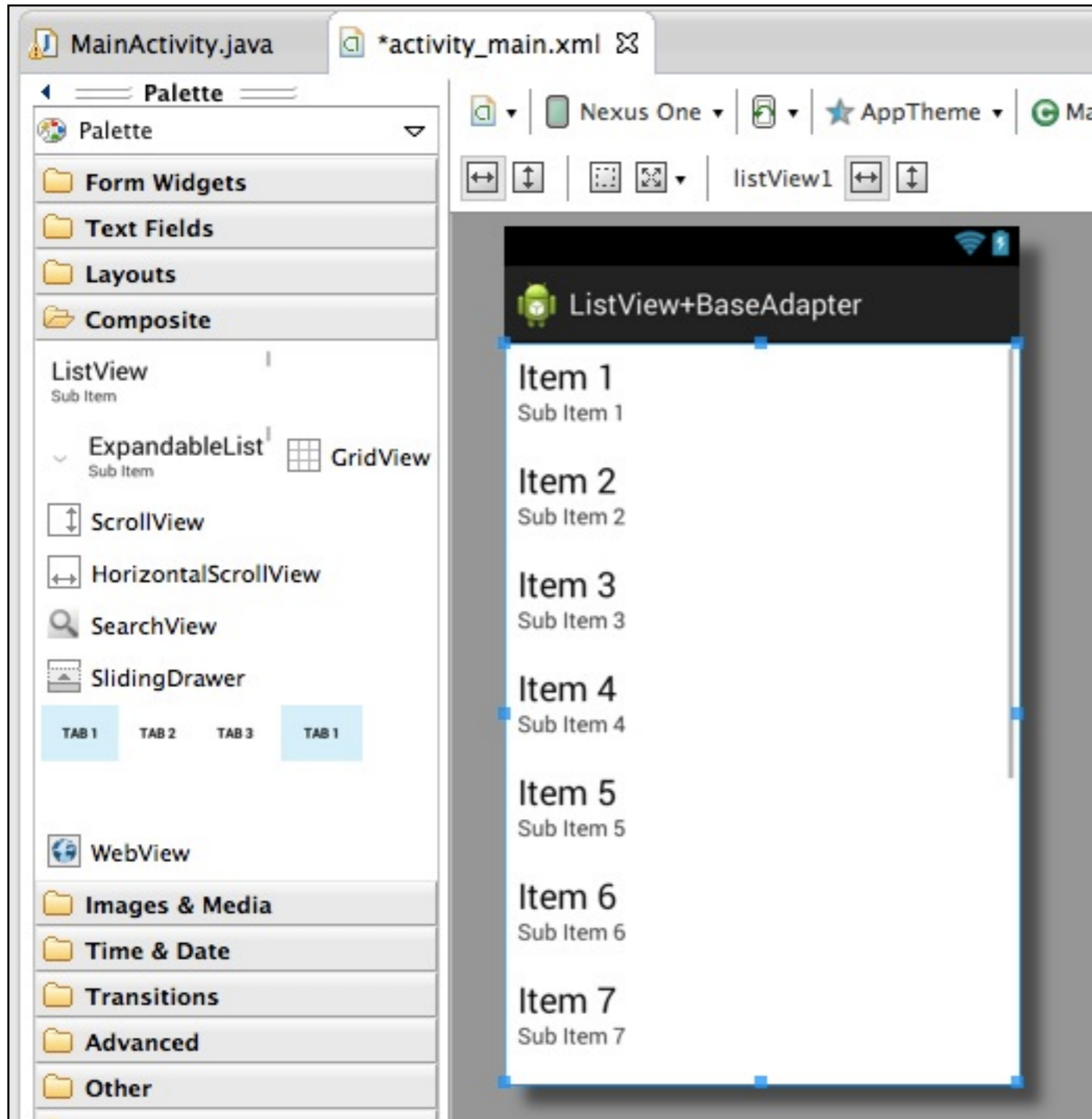
▶ Known Indirect Subclasses
  ResourceCursorAdapter, SimpleCursorAdapter
```

- 需要覆寫四個方法
- `int getCount()` //row總個數
- `Object getItem(int position)` //此row的資料
- `long getItemId(int position)` //此row的id
- `View getView(int position, View view, ViewGroup parent)` //cell view

ListView產生步驟

- 先拉一個ListView元件並取得此元件
- 建立自己要呈現的Cell View
- 建立繼承BaseAdapter的Class
- 在Class傳入(建構子)以下資訊:
 - 1. Context(上下文)，用以產生LayoutInflater
 - 2. 顯示資料
- 實現BaseAdapter需要產生的四個方法
- 採用LayoutInflater取得自定的Cell View並回傳
- 建立ListView和Adapter的關係

Composite > ListView



The Properties panel shows the configuration for a ListView widget. The 'Id' is set to '@+id/listView1'. The 'ListView' section is expanded, showing properties for 'Entries', 'Divider', 'Header Divider', and 'Footer Divider', each with a small green square icon next to it. The 'View' section is also visible at the bottom.

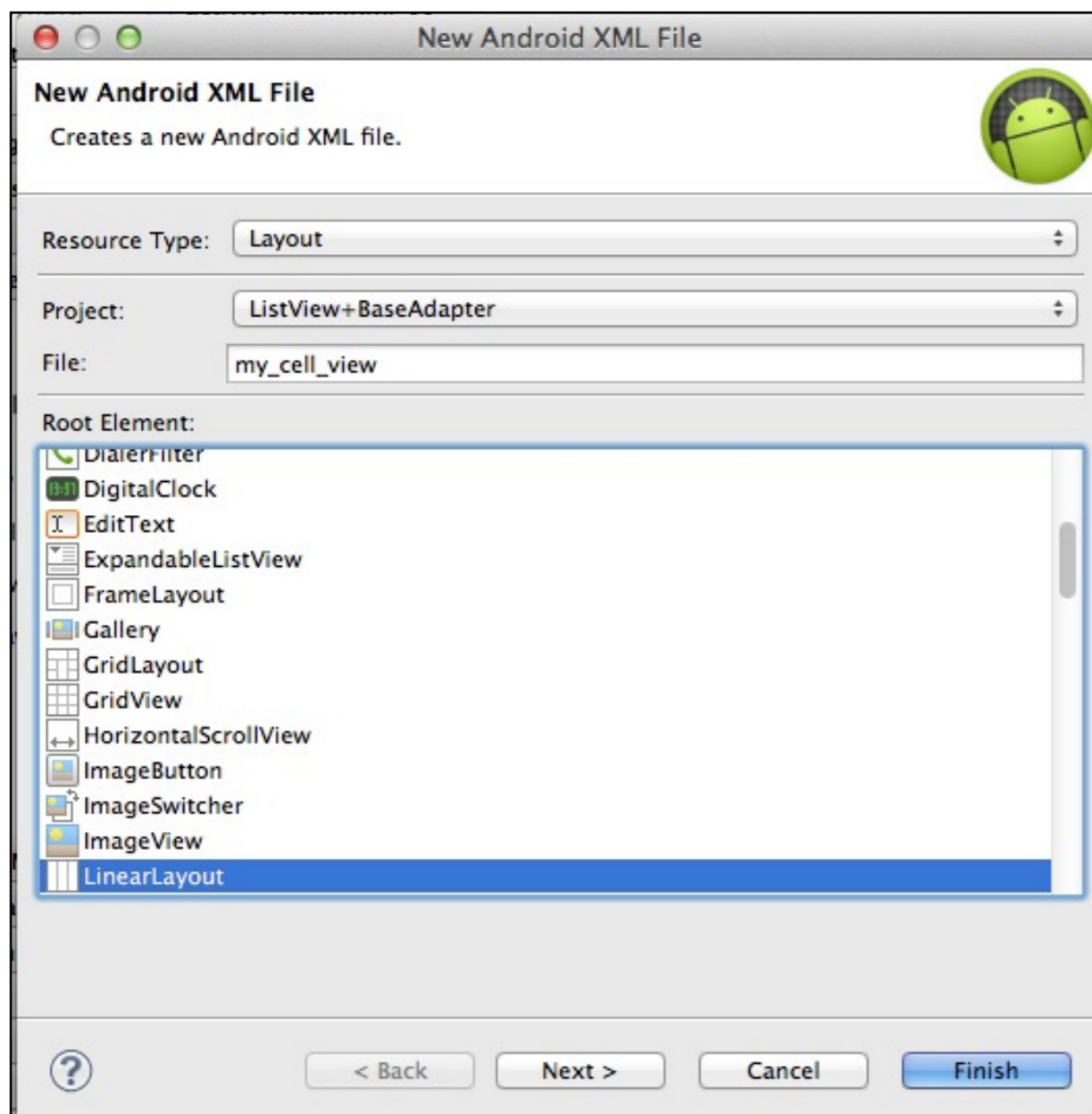
Properties	
Id	@+id/listView1
Layout Paramet...	{}
Divider	?android:attr/listDivider (...)
Draw Selector ...	[Green Square]
Cache Color Hint	?android:attr/colorBackgr...
Background	
Content Descri...	
ListView	{}
Entries	
Divider	?android:attr/listDivider (...)
Divider Height	
Header Divide...	[Green Square]
Footer Divider...	[Green Square]
Over Scroll He...	
Over Scroll Fo...	
View	{}

1 取得ListView

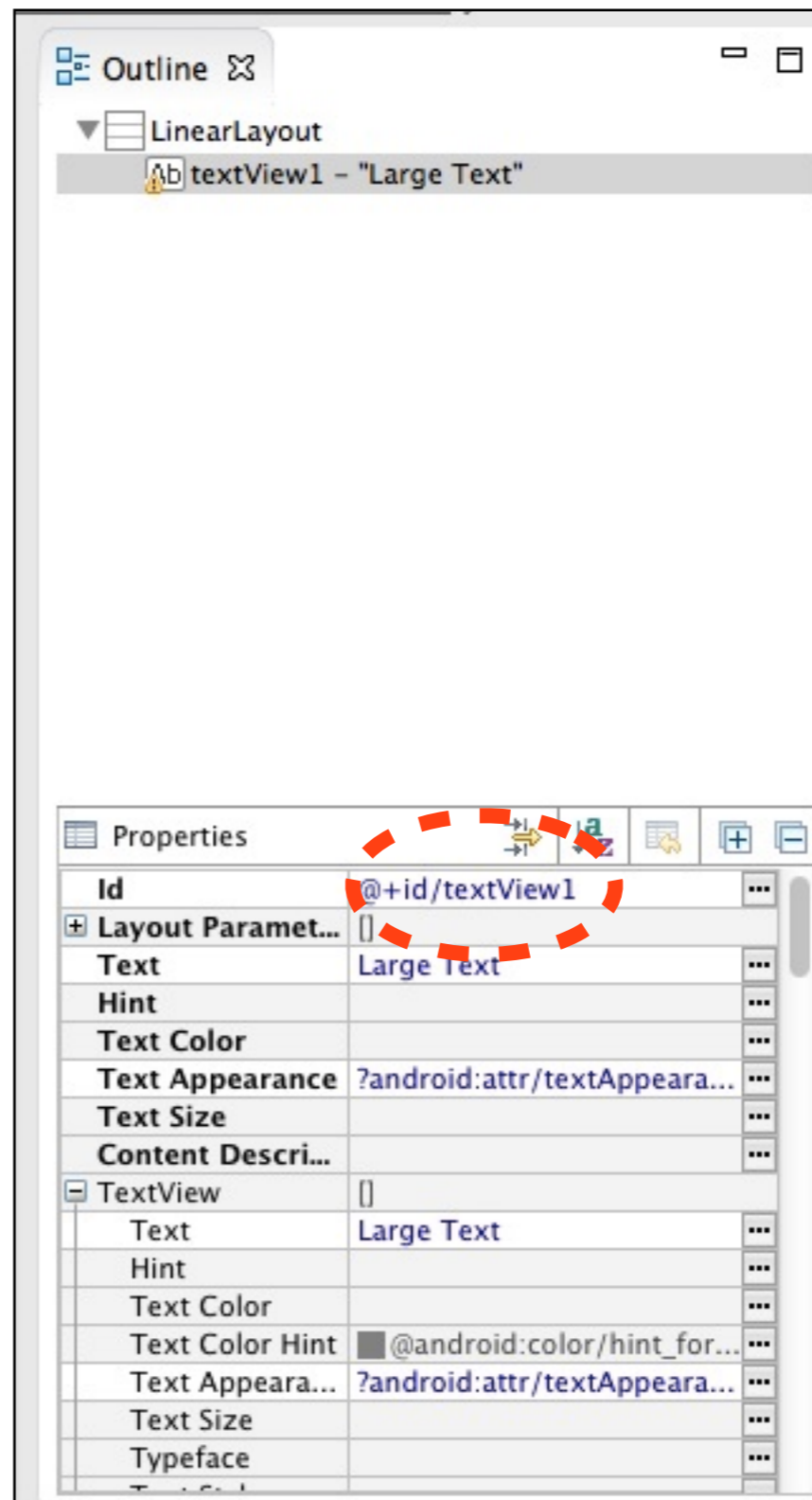
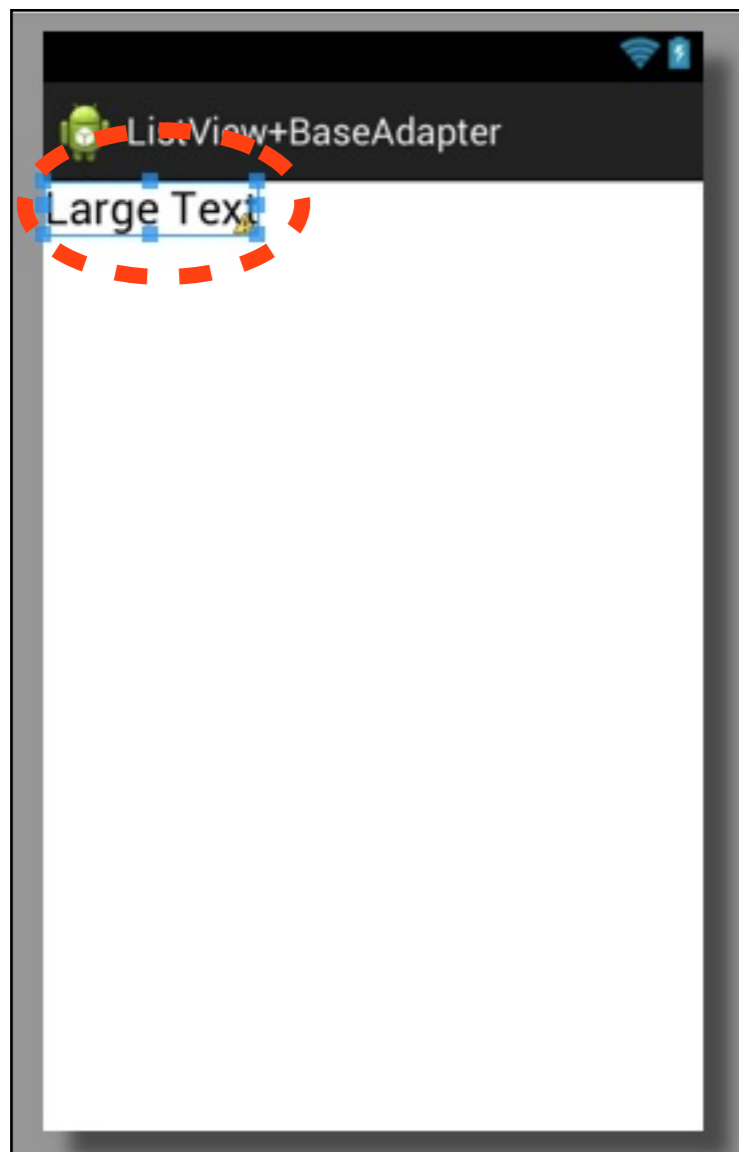
```
private ListView myListView;
```

```
myListView=(ListView)findViewById(R.id.listView1);
```

2 建立自定的Cell.



加入一個TextView即可



目前沒有任何資訊可以這些方法結合

```
private class MyBaseAdapter extends BaseAdapter{

    @Override
    public int getCount() {
        // TODO Auto-generated method stub
        return 0;
    }

    @Override
    public Object getItem(int arg0) {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public long getItemId(int arg0) {
        // TODO Auto-generated method stub
        return 0;
    }

    @Override
    public View getView(int arg0, View arg1, ViewGroup arg2) {
        // TODO Auto-generated method stub
        return null;
    }

}
```

需要的資訊有: 1. 資料來源 2. 作出View的資訊.

自己定義的資料結構

需要Context來做出View



如何取得自己的
CellView?

1. 實現LayoutInflater物件，此物件可由
- 2 由LayoutInflater.inflate方法取得自定cell view的layout

透過建構子初始化Data及LayoutInflater

```
private class MyBaseAdapter extends BaseAdapter{  
  
    private String[] myData;  
    private LayoutInflater myLayoutInflater;  
  
    //利用建構子來作資料初始化  
    public MyBaseAdapter(String[] data, Context context) {  
        myData=data;  
        myLayoutInflater=LayoutInflater.from(context);  
    }  
  
    ...  
    ...  
}
```

建構子不回傳東西

名稱與類別名稱相同

建構子語法

```
class 類別名稱 {  
    存取權限 類別名稱(型別 參數,..) {  
        ...  
    }  
}
```

實現方法 getCount 、 getItem 、 getItemId

```
@Override
public int getCount() {
    return myData.length;
}

@Override
public Object getItem(int position) {

    return myData[position];
}

@Override
public long getItemId(int position) {
    return position;
}
```

實現方法 getView

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {

    if(convertView==null)
        convertView=myLayoutInflater.inflate(R.layout.my_cell_view, null);

    return convertView;
}
```

建立ListView和Adapter的關係

新增自己資料

```
private ListView myListView;  
private String[] data={"Book1","Book2","Book3"}; //設定資料  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    myListView=(ListView)findViewById(R.id.listView1);  
    MyBaseAdapter adapter=new MyBaseAdapter(data, getBaseContext()); //新增MyBaseAdapter  
    myListView.setAdapter(adapter); //建立與ListView關係  
}
```

建立ListView和Adapter關係

執行結果(尚未把資料填入自定Cell View)



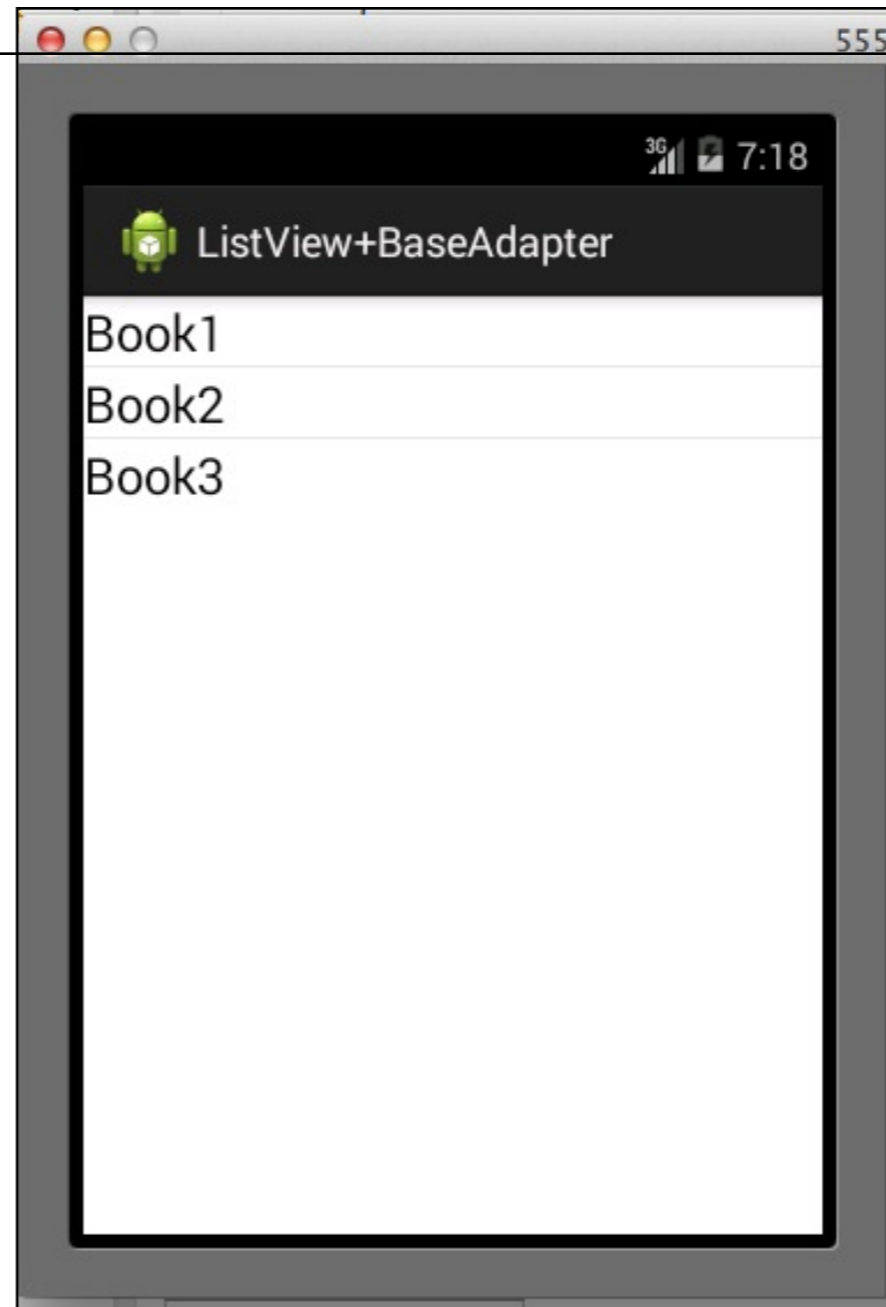
```
@Override
public View getView(int position, View convertView, ViewGroup parent) {

    if(convertView==null)
        convertView=myLayoutInflater.inflate(R.layout.my_cell_view, null);

    //convertView就是我們設定的Cell View
    TextView bookName=(TextView)convertView.findViewById(R.id.textView1);
    String bookNameString=myData[position];
    bookName.setText(bookNameString);

    return convertView;
}
```

由convertView的findViewById找到對應的View並作設定



(執行結果)

為ListView加上item選擇動作

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    myListView=(ListView)findViewById(R.id.listView1);
    MyBaseAdapter adapter=new MyBaseAdapter(data, getBaseContext()); //新增MyBaseAdapter
    myListView.setAdapter(adapter); //建立與ListView關係

    myListView.setOnItemClickListener(itemClickListener);
}

OnItemClickListener itemClickListener=new OnItemClickListener() {

    @Override
    public void onItemClick(AdapterView<?> parentView, View view, int position,
        long arg3) {

    }

};
```

採用Toast做訊息提示

- Toast可以在螢幕上彈出一段訊息
- 最簡單的用法，Class的方式
 - `Toast.makeText(Context,Message,時間長短);`
- 最後要用`show()`才會出現

加在onItemClickListener裡面

```
OnItemClickListener onItemClickListener=new OnItemClickListener() {  
  
    @Override  
    public void onItemClick(AdapterView<?> parentView, View view, int position,  
        long arg3) {  
        Toast.makeText(view.getContext(), "Row Index is"+position, Toast.LENGTH_SHORT).show();  
    }  
}
```

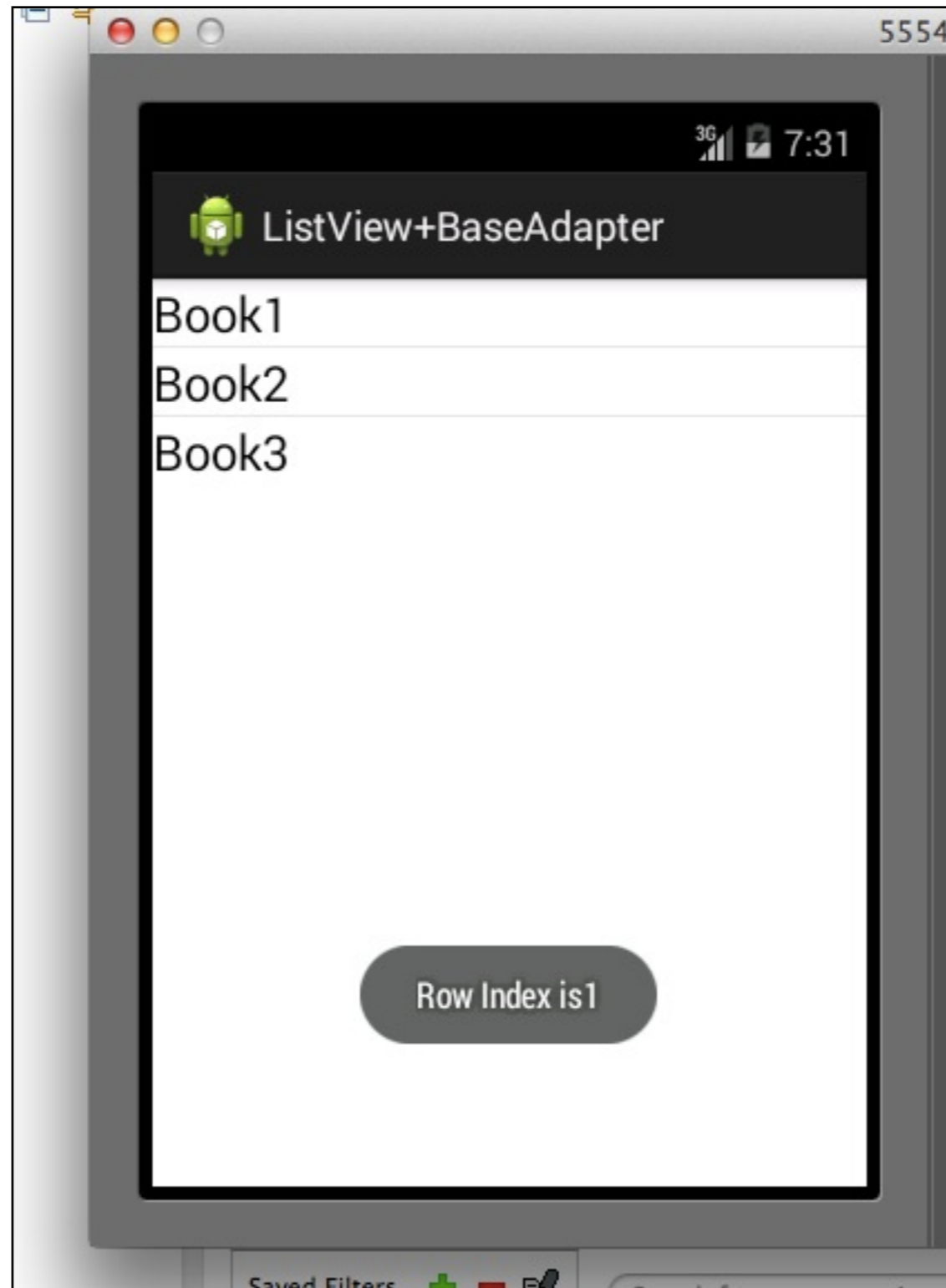
show才會顯示

透過+即可連接字串

時間設定為短

java的與法中，可以用「.」來串接方法

Example: ListView+BaseAdapter



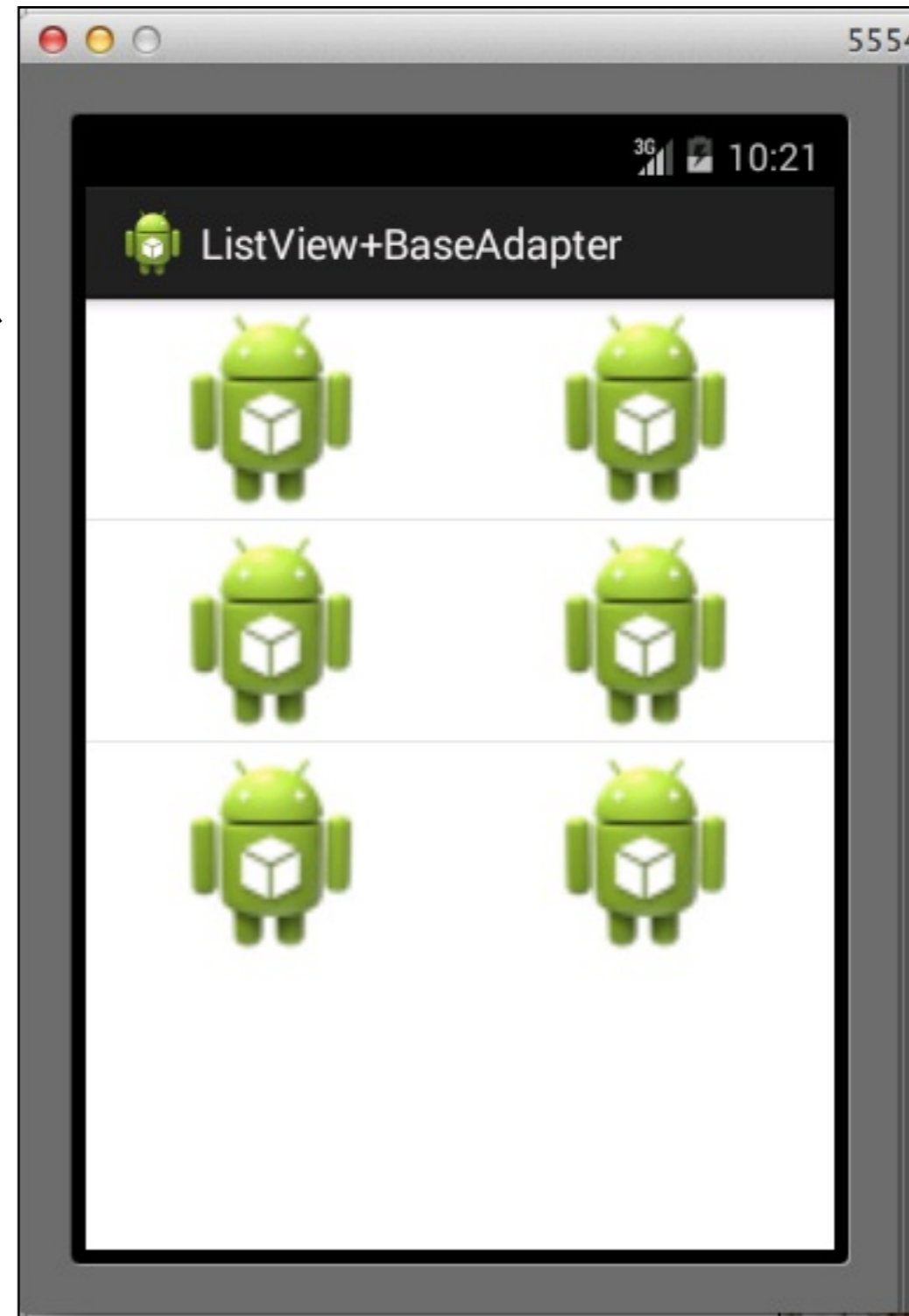
(執行結果)

Exercise: 請作出以下的layout

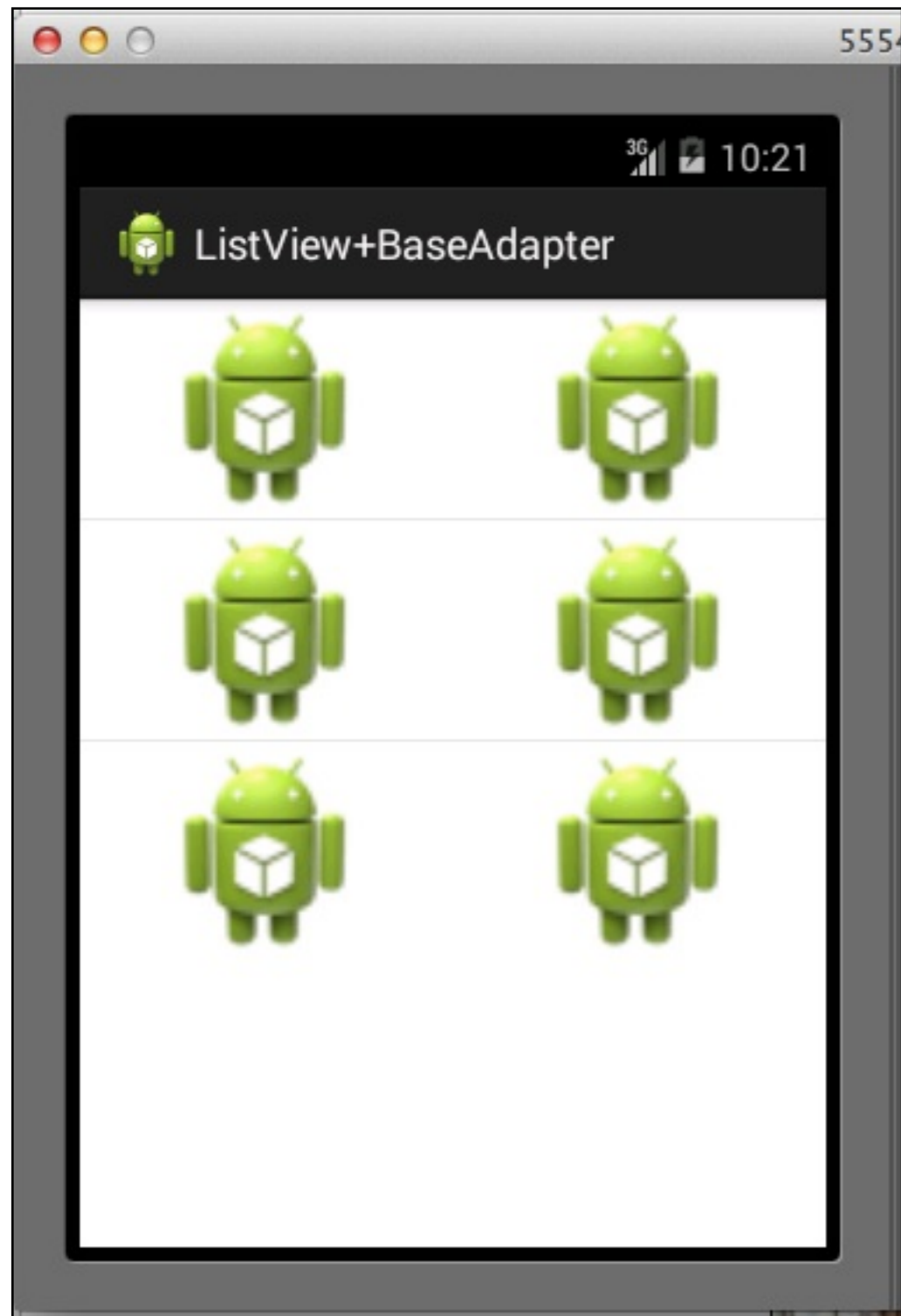
圖片先用ic_launcher.png



2個圖片



(執行結果)



一個
Row有兩個
ImageView，如何區



利用setTag與getTag

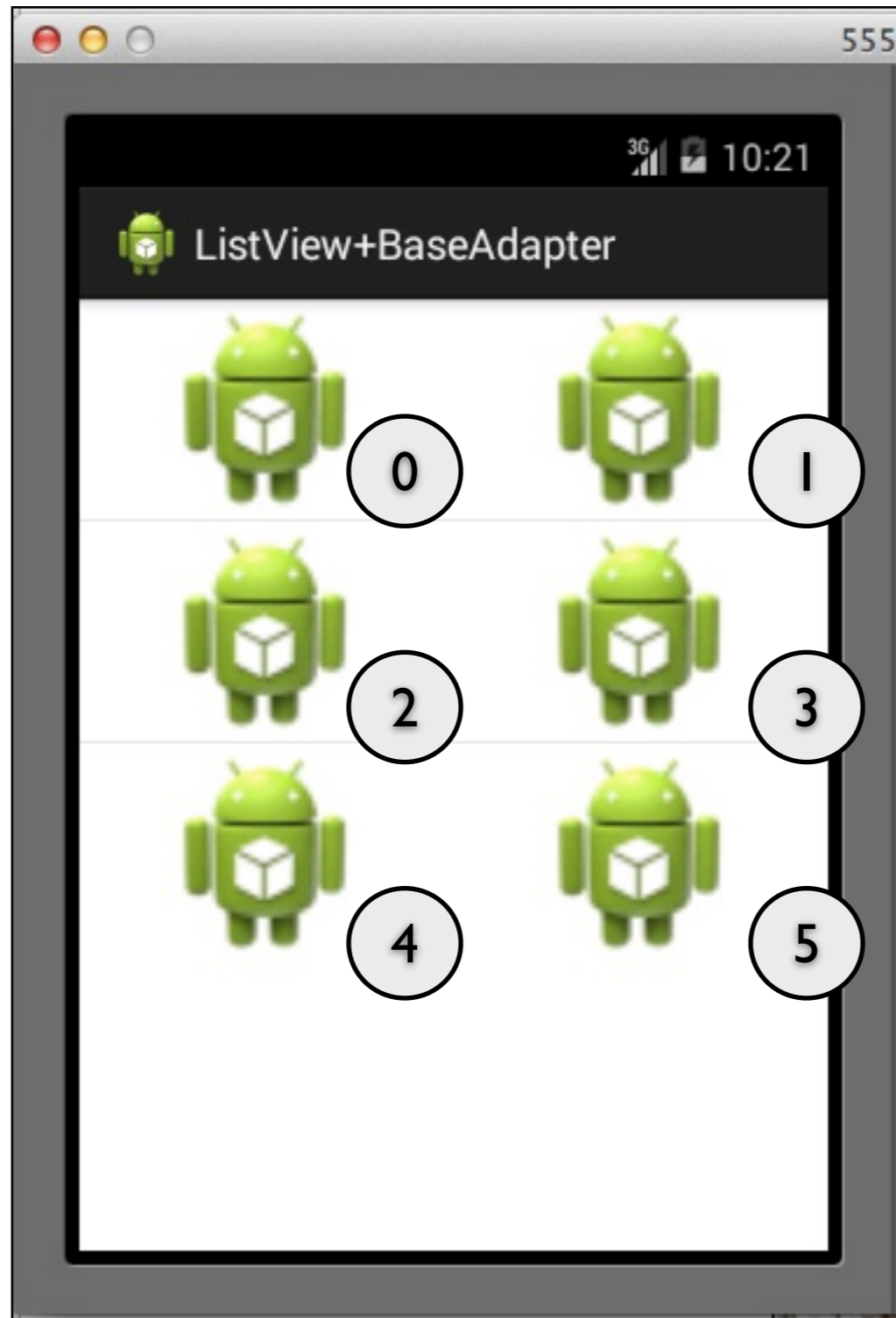
設定Tag為1

```
MyListener listener = new  
MyListener();  
button1.setTag(1);  
button1.setOnClickListener(listener);
```

在Listener的時候透過View v取得tag

```
View.OnClickListener {  
  
    @Override  
    public void onClick(View v) {  
        int tag = (Integer) v.getTag();  
  
    }  
}
```

Exercise: 點擊不同的圖顯示不同的數字



getView改寫後

```
@Override
    public View getView(int position, View convertView, ViewGroup parent) {

        if(contentView==null)
            convertView=myLayoutInflater.inflate(R.layout.my_cell_view, null);

        column1ImageView=(ImageView)convertView.findViewById(R.id.imageView1);
        column2ImageView=(ImageView)convertView.findViewById(R.id.imageView2);

        column1ImageView.setTag(position*2);
        column2ImageView.setTag(position*2+1);

        column1ImageView.setOnClickListener(myImageClickListener);
        column2ImageView.setOnClickListener(myImageClickListener);

        return convertView;
    }
```

相對應的Listener

```
private OnClickListener myImageClickListener=new OnClickListener() {

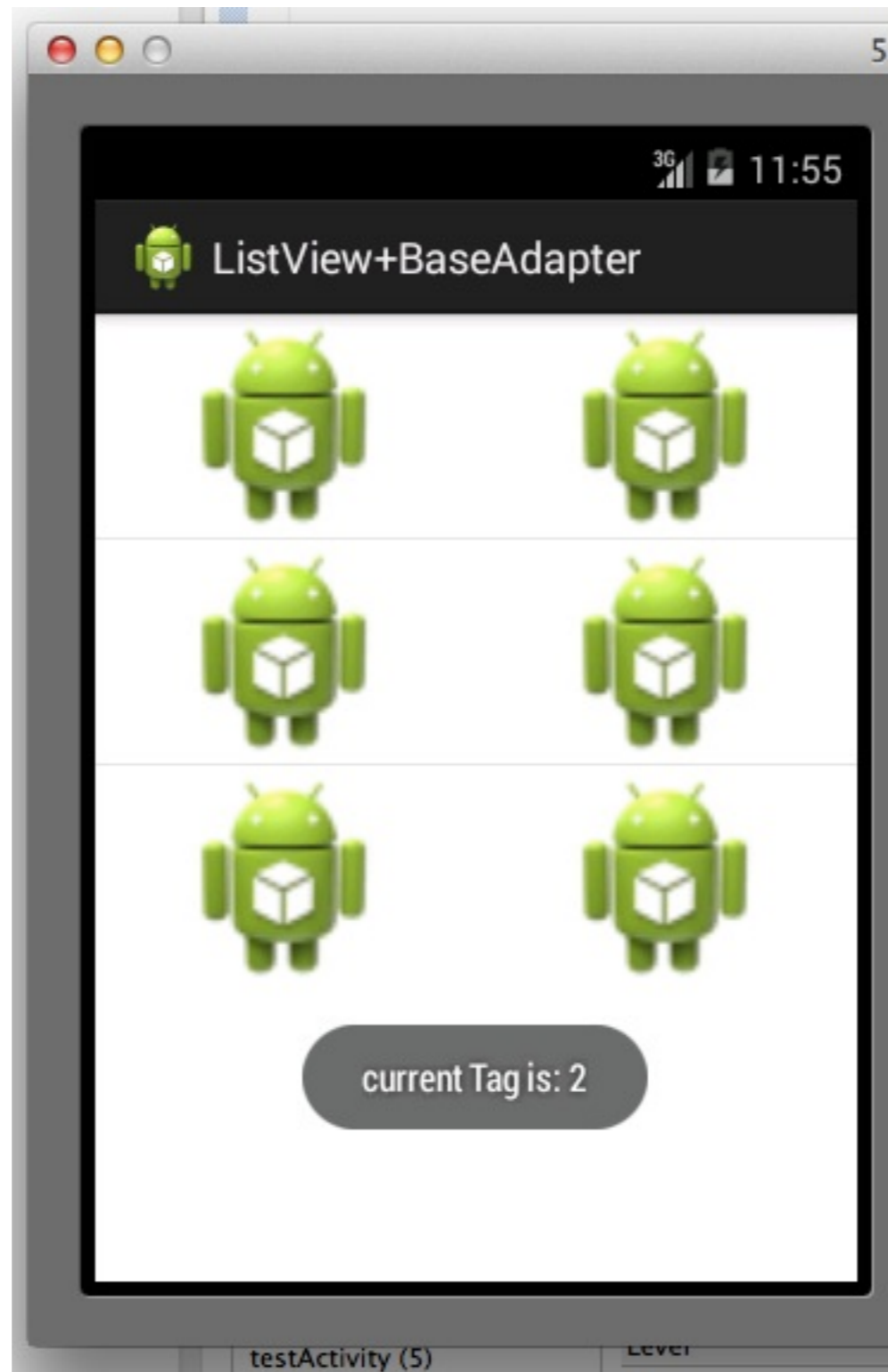
    @Override
    public void onClick(View view) {

        int tag=(Integer) view.getTag();
        Toast.makeText(view.getContext(), "current Tag is: "+tag, Toast.LENGTH_SHORT).show();

    }

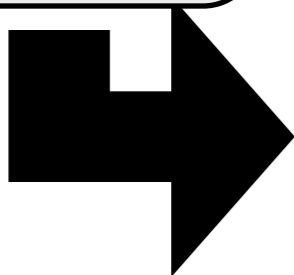
};
```

執行結果，已經可以針對不同的Click做相對應的動作

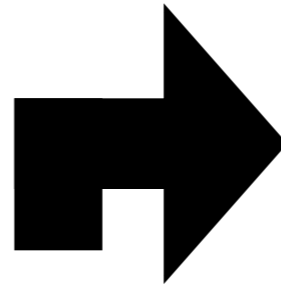




你應該會了...



透過Activity傳參數 Done!



用影像照片一張張記錄與自己最愛的寵物寶貝成長日誌，鏡頭裡自在綻放甜美笑容的魚兒，不時流露出與毛寶貝互動可愛畫面，驗證了狗狗真的是人類忠實的好朋友，養牠就請照顧牠。

螢幕向左滑動欣賞作品

圖片瀏覽、滑動切換 Done

ListView
Done!

資料來源呢?



查看Memobook提供的API

1. 取得所有書城上的分類 (可以直接把網址貼到瀏覽器做測試)

http://images.memobook.com.tw/m/api.php?act=memo_list

參數說明：

act => 呼叫 API 的類型

JSON 回傳格式說明：(無筆數限制)

```
[  
  { "id" : 分類編號 , "title" : 分類標題 , "counts" : 該分類的書籍量 , "icon" : 代表的 ICON , "icon_bg" : 該項目的背景圖片 } ,  
  { "id" : 分類編號 , "title" : 分類標題 , "counts" : 該分類的書籍量 , "icon" : 代表的 ICON , "icon_bg" : 該項目的背景圖片 } ,  
  { "id" : 分類編號 , "title" : 分類標題 , "counts" : 該分類的書籍量 , "icon" : 代表的 ICON , "icon_bg" : 該項目的背景圖片 } ,  
  ..... 以此類推  
]
```

另外關於測試用的 API 網址如下：

<http://images.memobook.com.tw/sample/api.php>

採用OpenSource

Android Asynchronous Http Client

<http://loopj.com/android-async-http/>

Android Asynchronous Http Client

A Callback-Based Http Client Library for Android

 Fork 1,819  Tweet 254

Overview

An asynchronous callback-based Http client for Android built on top of Apache's [HttpClient](#) libraries. All requests are made outside of your app's main UI thread, but any callback logic will be executed on the same thread as the callback was created using Android's Handler message passing.

Download

version 1.4.4 (latest)

[or fork me on github](#)

Features

- Make asynchronous HTTP requests, handle responses in anonymous callbacks
- HTTP requests happen outside the UI thread
- Requests use a **threadpool** to cap concurrent resource usage
- GET/POST **params builder** (RequestParams)
- **Multipart file uploads** with no additional third party libraries
- Tiny size overhead to your application, only **25kb** for everything
- Automatic smart **request retries** optimized for spotty mobile connections
- Automatic **gzip** response decoding support for super-fast requests
- Binary file (images etc) downloading with `BinaryHttpResponseHandler`
- Built-in response parsing into JSON with `JsonHttpResponseHandler`
- **Persistent cookie store**, saves cookies into your app's SharedPreferences

安裝步驟

Installation & Basic Usage

Download the latest .jar file from github and place it in your Android app's libs/ folder.

Import the http package.

```
import com.loopj.android.http.*;
```

由於加入使用INTERNET要求，必須在
AndroidManifest.xml設定使用權限

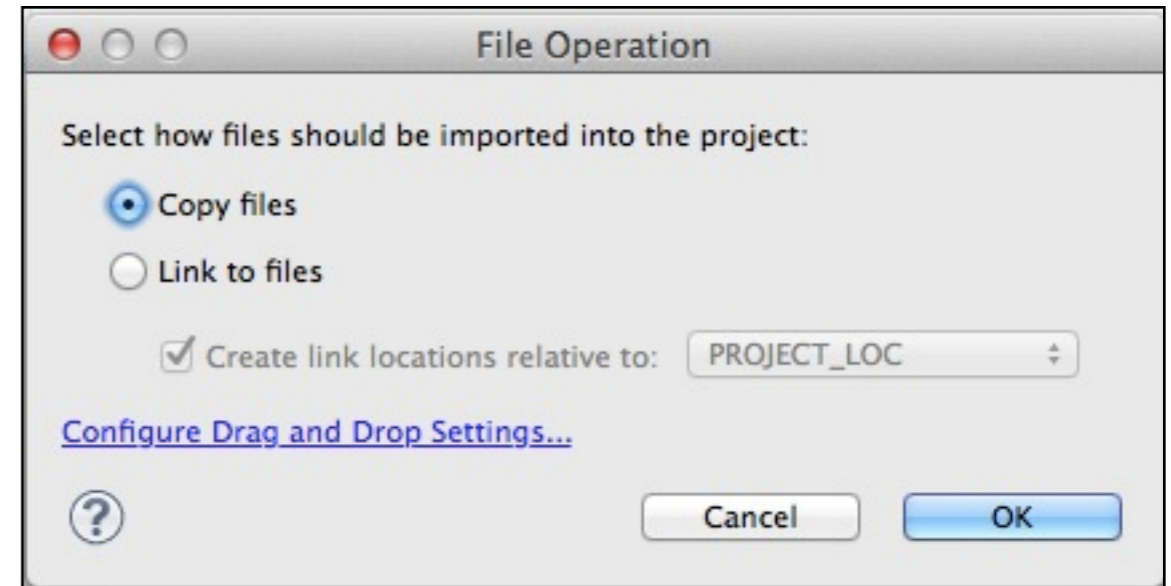
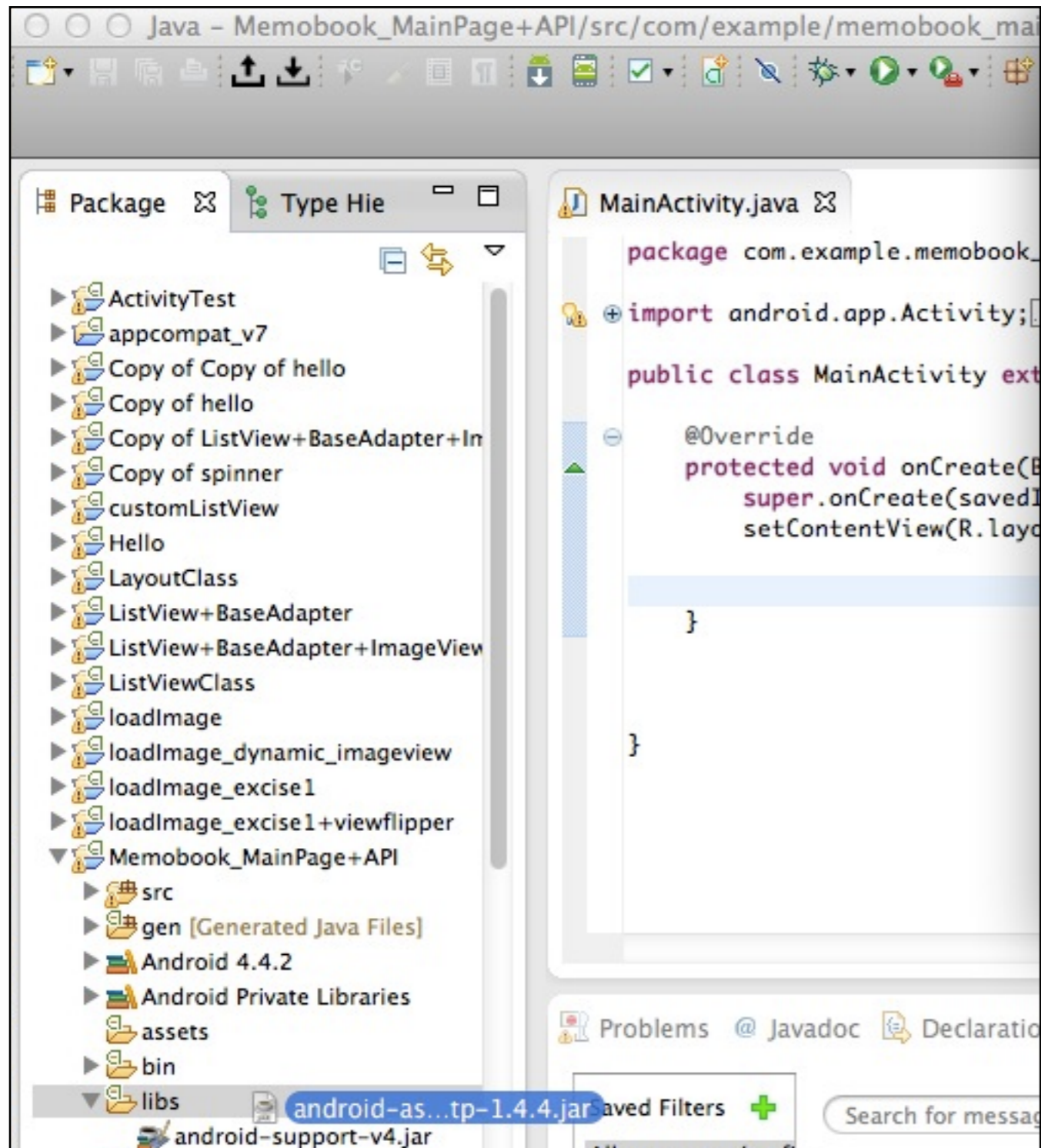
建立專案名稱為:Memobook_MainPage+API

AndroidManifest.xml

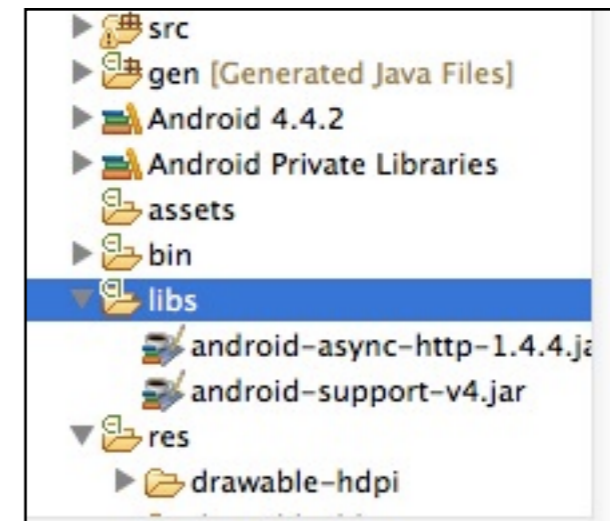
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.memobook_mainpageapi"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="19" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.memobook_mainpageapi.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

加入INTERNET使用權限

jar檔案加入libs



選擇 Copy files



多了一個android-async-

將檔案拖曳到libs

import後使用

```
package com.example.memobook_mainpageapi;

import android.app.Activity;
import android.os.Bundle;
import com.loopj.android.http.*;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

如何加入參數

Adding GET/POST Parameters with RequestParams

The `RequestParams` class is used to add optional GET or POST parameters to your requests. `RequestParams` can be built and constructed in various ways:

Create empty `RequestParams` and immediately add some parameters:

```
RequestParams params = new RequestParams();
params.put("key", "value");
params.put("more", "data");
```

使用步驟

Create a new `AsyncHttpClient` instance and make a request:

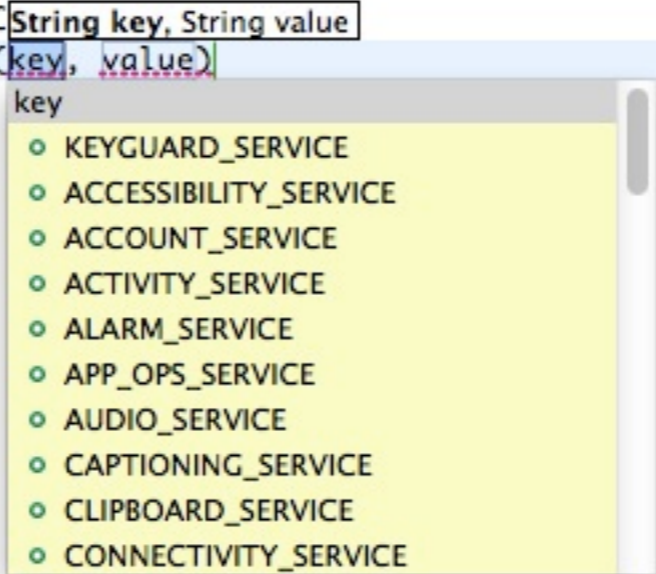
```
AsyncHttpClient client = new AsyncHttpClient();
client.get("http://www.google.com", new AsyncHttpResponseHandler() {
    @Override
    public void onSuccess(String response) {
        System.out.println(response);
    }
});
```

觀察API：http://images.memobook.com.tw/m/api.php?act=memo_list

(Get方式)

有一個參數 Key="act" Value="memo_list"

```
public class MainActivity extends Activity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        AsyncHttpClient client = new AsyncHttpClient();  
        RequestParams params=new RequestParams("key", "value");  
    }  
}
```



產生參數

Get方法中，我們需要url，params以及回傳結果

```
public class MainActivity extends Activity {
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    AsyncHttpClient client = new AsyncHttpClient();
```

```
    RequestParams params=new RequestParams("act", "memo_list"); //產生參數
```

```
    client.get
```

```
}
```

- get(String url, ResponseHandlerInterface responseHandler) : RequestHandle - AsyncHttpClient
- get(Context context, String url, ResponseHandlerInterface responseHandler) : RequestHandle - AsyncHttpClient
- get(String url, RequestParams params, ResponseHandlerInterface responseHandler) : RequestHandle - AsyncHttpClient
- get(Context context, String url, RequestParams params, ResponseHandlerInterface responseHandler) : RequestHandle - AsyncHttpClient
- get(Context context, String url, Header[] headers, RequestParams params, ResponseHandlerInterface responseHandler) : RequestHandle - AsyncHttpClient
- getClass() : Class<?> - Object
- getHttpClient() : HttpClient - AsyncHttpClient
- getHttpContext() : HttpContext - AsyncHttpClient
- getMaxConnections() : int - AsyncHttpClient
- getTimeout() : int - AsyncHttpClient
- getUrlWithQueryString(boolean arg0, String arg1, RequestParams arg2) : String - AsyncHttpClient

Press '\/' to show Template Proposals

提供的JSON回傳範例，用 JsonHttpResponseHandler

This then makes it very easy to work with the Twitter API in your code:

```
import org.json.*;
import com.loopj.android.http.*;

class TwitterRestClientUsage {
    public void getPublicTimeline() throws JSONException {
        TwitterRestClient.get("statuses/public_timeline.json", null, new JsonHttpResponseHandler() {
            @Override
            public void onSuccess(JSONArray timeline) {
                // Pull out the first event on the public timeline
                JSONObject firstEvent = timeline.get(0);
                String tweetText = firstEvent.getString("text");

                // Do something with the response
                System.out.println(tweetText);
            }
        });
    }
}
```

取得JSON回傳程式碼

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    AsyncHttpClient client = new AsyncHttpClient();
    RequestParams params= new RequestParams("act","memo_list");

    client.get("http://images.memobook.com.tw/m/api.php", params, new JsonHttpResponseHandler()
        {

            @Override
            public void onSuccess(JSONArray response) {
                try {
                    JSONObject eachRow=response.getJSONObject(0);

                    String iconbg=eachRow.getString("icon_bg");

                    System.out.print(eachRow);
                } catch (JSONException e) {
                    e.printStackTrace();
                }
                super.onSuccess(response);
            }

        });
}
```

用JSON格式回傳

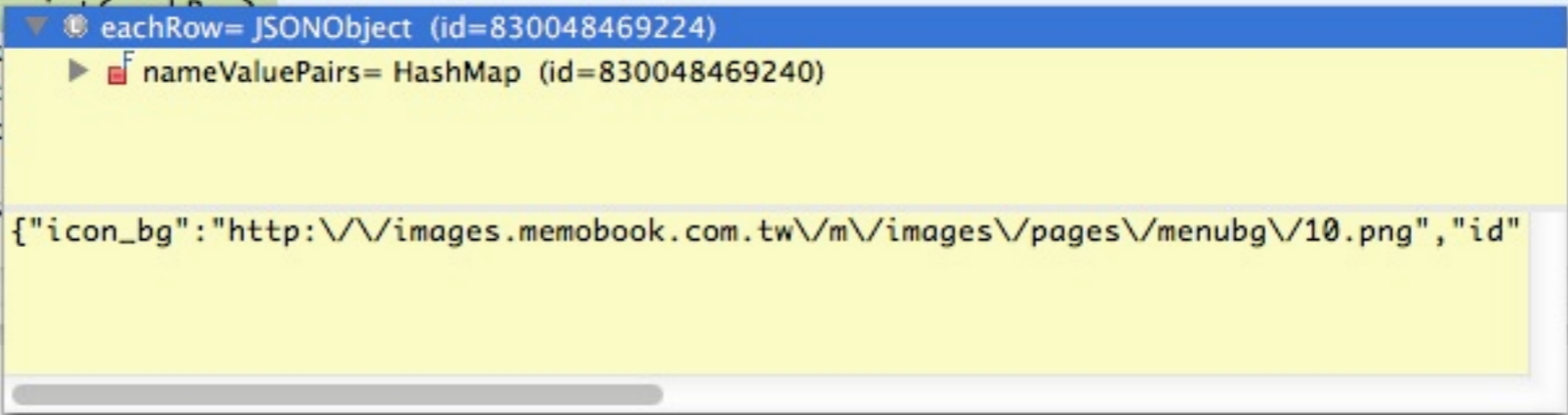
取得其中一個JSON Object

取得對應Key的Value

執行結果

```
AsynchHttpClient client = new AsynchHttpClient();
RequestParams params= new RequestParams("act","memo_list");

client.get("http://images.memobook.com.tw/m/api.php", params, new JsonHttpResponseHandler()
{
    @Override
    public void onSuccess(JSONArray response) {
        // TODO Auto-generated method stub
        try {
            JSONObject eachRow=response.getJSONObject(0);
            System.out.println(eachRow);
        } catch (JSONException e) {
            // TODO Auto-generated method stub
            e.printStackTrace();
        }
        super.onSuccess(response);
    }
}
```



eachRow= JSONObject (id=830048469224)
nameValuePairs= HashMap (id=830048469240)
{"icon_bg":"http://images.memobook.com.tw/m/images/pages/menubg/10.png","id"

取得回傳

如何下載圖片?



採用OpenSource

Android Universal Image Loader

<https://github.com/nostra13/Android-Universal-Image-Loader>



Universal Image Loader for Android

This project aims to provide a reusable instrument for asynchronous image loading, caching and displaying. It is originally based on [Fedor Vlasov's project](#) and has been vastly refactored and improved since then.



Features

- Multithread image loading
- Possibility of wide tuning ImageLoader's configuration (thread executors, downloader, decoder, memory and disk cache, display image options, and others)
- Possibility of image caching in memory and/or on device's file system (or SD card)
- Possibility to "listen" loading process
- Possibility to customize every display image call with separated options
- Widget support

Android 2.0+ support

Downloads

- [universal-image-loader-1.9.2.jar](#) (library; contains *.class files)
- [universal-image-loader-1.9.2-sources.jar](#) (sources; contains *.java files)
- [universal-image-loader-1.9.2-javadoc.jar](#) (Java docs; contains *.html files)
- [universal-image-loader-1.9.2-with-sources.jar](#) (library with sources inside; contains *.class and *.java files)
Prefer to use this JAR so you can see Java docs in Eclipse tooltips.
- [universal-image-loader-sample-1.9.2.apk](#) (sample application)

Latest snapshot of the library - [here](#)

安裝方式

1. Include library

Manual:

- [Download JAR](#)
- Put the JAR in the **libs** subfolder of your Android project

or

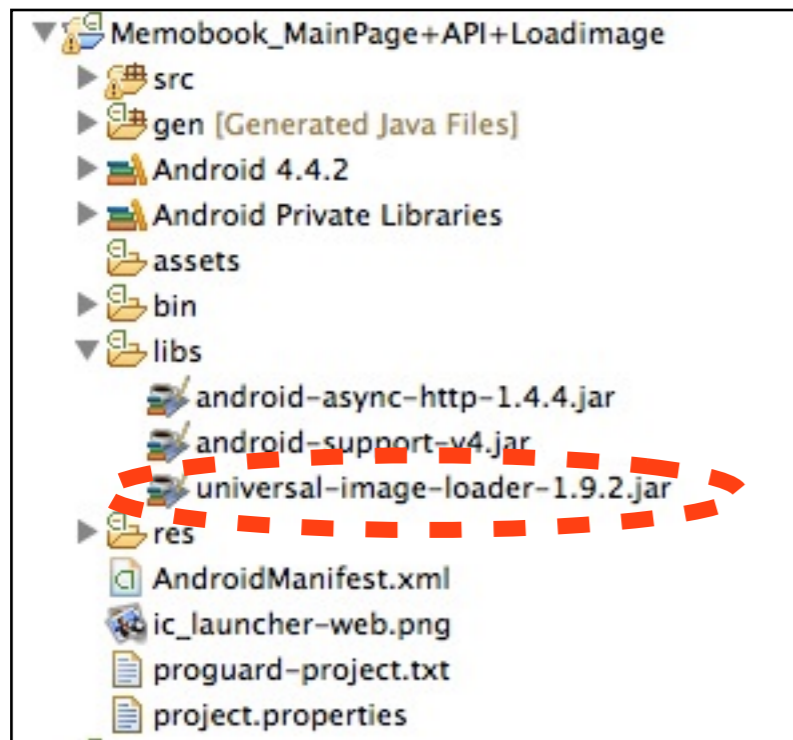
Maven dependency:

```
<dependency>
  <groupId>com.nostra13.universalimageloader</groupId>
  <artifactId>universal-image-loader</artifactId>
  <version>1.9.2</version>
</dependency>
```

2. Android Manifest

```
<manifest>
  <uses-permission android:name="android.permission.INTERNET" />
  <!-- Include next permission if you want to allow UIL to cache images on SD card -->
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
  ...
  <application android:name="MyApplication">
    ...
  </application>
</manifest>
```

專案Memobook_MainPage+API+Loadimage



加入libaray universal-image-loader-1.9.2.jar

提供設定選項

Configuration and Display Options

- ImageLoader **Configuration** (`ImageLoaderConfiguration`) is **global** for application.
- **Display Options** (`DisplayImageOptions`) are **local** for every display task
(`ImageLoader.displayImage(...)`).

Configuration

All options in Configuration builder are optional. Use only those you really want to customize.
See default values for config options in Java docs for every option.

```
// DON'T COPY THIS CODE TO YOUR PROJECT! This is just example of ALL options using.
File cacheDir = StorageUtils.getCacheDirectory(context);
ImageLoaderConfiguration config = new ImageLoaderConfiguration.Builder(context)
    .memoryCacheExtraOptions(480, 800) // default = device screen dimensions
    .diskCacheExtraOptions(480, 800, null)
    .taskExecutor(...)
    .taskExecutorForCachedImages(...)
    .threadPoolSize(3) // default
    .threadPriority(Thread.NORM_PRIORITY - 1) // default
    .tasksProcessingOrder(QueueProcessingType.FIFO) // default
    .denyCacheImageMultipleSizesInMemory()
    .memoryCache(new LruMemoryCache(2 * 1024 * 1024))
    .memoryCacheSize(2 * 1024 * 1024)
    .memoryCacheSizePercentage(13) // default
    .diskCache(new UnlimitedDiscCache(cacheDir)) // default
    .diskCacheSize(50 * 1024 * 1024)
    .diskCacheFileCount(100)
    .diskCacheFileNameGenerator(new HashCodeFileNameGenerator()) // default
    .imageDownloader(new BaseImageDownloader(context)) // default
    .imageDecoder(new BaseImageDecoder()) // default
    .defaultDisplayImageOptions(DisplayImageOptions.createSimple()) // default
    .writeDebugLogs()
    .build();
```

提供顯示選項

Display Options

Display Options can be applied to every display task (`ImageLoader.displayImage(...)` call).

Note: If Display Options wasn't passed to `ImageLoader.displayImage(...)` method then default Display Options from configuration (`ImageLoaderConfiguration.defaultDisplayImageOptions(...)`) will be used.

```
// DON'T COPY THIS CODE TO YOUR PROJECT! This is just example of ALL options using.
DisplayImageOptions options = new DisplayImageOptions.Builder()
    .showImageOnLoading(R.drawable.ic_stub) // resource or drawable
    .showImageForEmptyUri(R.drawable.ic_empty) // resource or drawable
    .showImageOnFail(R.drawable.ic_error) // resource or drawable
    .resetViewBeforeLoading(false) // default
    .delayBeforeLoading(1000)
    .cacheInMemory(false) // default
    .cacheOnDisk(false) // default
    .preProcessor(...)
    .postProcessor(...)
    .extraForDownloader(...)
    .considerExifParams(false) // default
    .imageScaleType(ImageScaleType.IN_SAMPLE_POWER_OF_2) // default
    .bitmapConfig(Bitmap.Config.ARGB_8888) // default
    .decodingOptions(...)
    .displayer(new SimpleBitmapDisplayer()) // default
    .handler(new Handler()) // default
    .build();
```

使用方式

Useful Info

1. **Caching is NOT enabled by default.** If you want loaded images will be cached in memory and/or on disk then you should enable caching in DisplayImageOptions this way:

```
// Create default options which will be used for every
// displayImage(...) call if no options will be passed to this method
DisplayImageOptions defaultOptions = new DisplayImageOptions.Builder()
    ...
    .cacheInMemory(true)
    .cacheOnDisk(true)
    ...
    .build();
ImageLoaderConfiguration config = new ImageLoaderConfiguration.Builder(getApplicationContext())
    ...
    .defaultDisplayImageOptions(defaultOptions)
    ...
    .build();
ImageLoader.getInstance().init(config); // Do it on Application start
```

1. 設定displayImageOptions

2. 設定

```
// Then later, when you want to display image
ImageLoader.getInstance().displayImage(imageUrl, imageView); // Default options will be used
```

3. 使用方法

imageUrl:網址

imageView:被顯示的ImageView

在原本的專案加入步驟1和步驟2

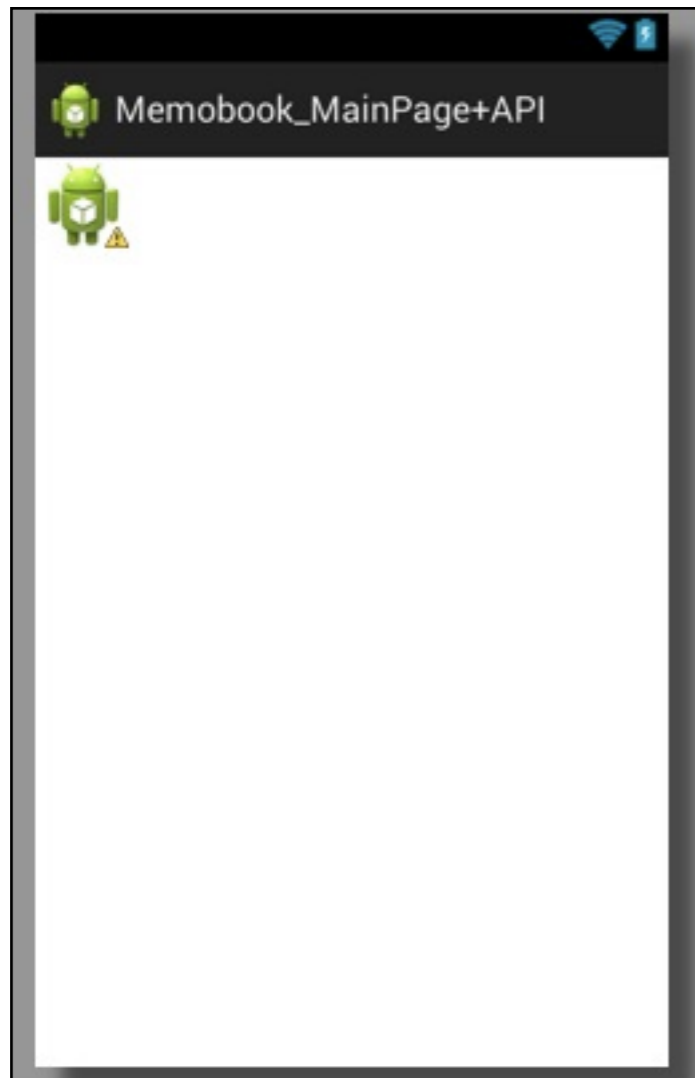
```
//變數宣告
DisplayImageOptions options;
ImageLoader imageLoader = ImageLoader.getInstance();

// 1. 設定display option
options = new DisplayImageOptions.Builder()
    .showImageForEmptyUri(R.drawable.ic_launcher) //若網址不存在的圖片
    .showImageOnFail(R.drawable.ic_launcher) //若載入錯誤的圖片
    .cacheInMemory(true) //用記憶體暫存
    .cacheOnDisk(true) //存在本機
    .build();

//2. 設定imageLoaderConfiguration
ImageLoaderConfiguration config = new ImageLoaderConfiguration.Builder(getBaseContext())
    .defaultDisplayImageOptions(options) //設定內定的display option
    .build();

imageLoader.init(config); //初始化
```

拉一個ImageView用來做網路下載圖片的顯示



步驟3

```
imageLoader.displayImage(iconbg, myImageView);
```

得到的網址

被改變的ImageView

完整程式碼

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    myImageView=(ImageView) findViewById(R.id.imageView1);
    //變數宣告
    DisplayImageOptions options;
    final ImageLoader imageLoader = ImageLoader.getInstance();

    // 1. 設定display option
    options = new DisplayImageOptions.Builder()
        .showImageForEmptyUri(R.drawable.ic_launcher) //若網址不存在的圖片
        .showImageOnFail(R.drawable.ic_launcher) //若載入錯誤的圖片
        .cacheInMemory(true) //用記憶體暫存
        .cacheOnDisk(true) //存在本機
        .build();

    //2. 設定imageLoaderConfiguration
    ImageLoaderConfiguration config = new ImageLoaderConfiguration.Builder(getBaseContext())
        .defaultDisplayImageOptions(options) //設定內定的display option
        .build();

    imageLoader.init(config); //初始化
}
```

inner class的變數要加final，避免被改變

```
//取得JSON Response
AsyncHttpClient client = new AsyncHttpClient();
//設定參數
RequestParams params= new RequestParams("act","memo_list");
//使用get方式
client.get("http://images.memobook.com.tw/m/api.php", params, new JsonHttpResponseHandler()
    {

        @Override
        public void onSuccess(JSONArray response) {
            // TODO Auto-generated method stub
            try {
                JSONObject eachRow=response.getJSONObject(0);
                String iconbg=eachRow.getString("icon_bg");
                //3. 設定要顯示的圖片
                imageLoader.displayImage(iconbg,myImageView);

            } catch (JSONException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            super.onSuccess(response);
        }

    });
```

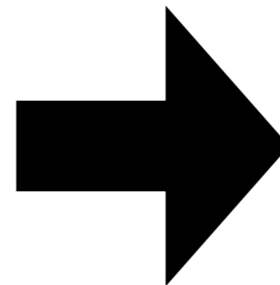
執行結果



專案Memobook_MainPage+API+Loadimage

Exercise: 請整合

List View + BaseAdapter + ImageLoader + AP



只有單個欄位沒其他按鈕

2個圖片個別帶出資料

步驟0. 建立cell的Layout

步驟1. 建立繼承BaseAdapter的class

步驟2. 初始設定ImageLoader

步驟3. 設定網路API回傳資料

步驟4. 在獲得資料後設定ListView的Adapter

提示:API CallBack後才把ListView的Adapter設定好

```
client.get("http://images.memobook.com.tw/m/api.php", params, new JsonHttpResponseHandler()
    {
        @Override
        public void onSuccess(JSONArray response) {
            myPage1Adapter adapter=new myPage1Adapter(getBaseContext(), response);
            myListView.setAdapter(adapter);
            super.onSuccess(response);
        }
    });
```

執行結果



Exercise: 請完成此專案



- .使用ListView+BaseAdapter
- .使用ImageLoader
- .使用Async Http Request

- .傳遞單列參數
- .傳遞雙欄參數
- .Activity資料傳遞

2. 取得某一分類裡面的相片書列表

http://images.memobook.com.tw/m/api.php?act=memo_class&id=10

參數說明：

act => 呼叫 API 的類型

id => 分類編號 (從前一個 API 取得)

JSON 回傳格式說明：(無筆數限制)

```
[  
  {"id": 書籍編號, "img": 書籍縮圖網址, "title": 書籍標題, "author": 作者姓名 },  
  {"id": 書籍編號, "img": 書籍縮圖網址, "title": 書籍標題, "author": 作者姓名 },  
  {"id": 書籍編號, "img": 書籍縮圖網址, "title": 書籍標題, "author": 作者姓名 },  
  ..... 以此類推  
]
```

3. 取得某一書籍的詳細內容及相片書圖片

http://images.memobook.com.tw/m/api.php?act=memo_detail&id=318

參數說明：

act => 呼叫 API 的類型

id => 書籍編號 (從前一個 API 取得)

JSON 回傳格式說明：(只有一筆)

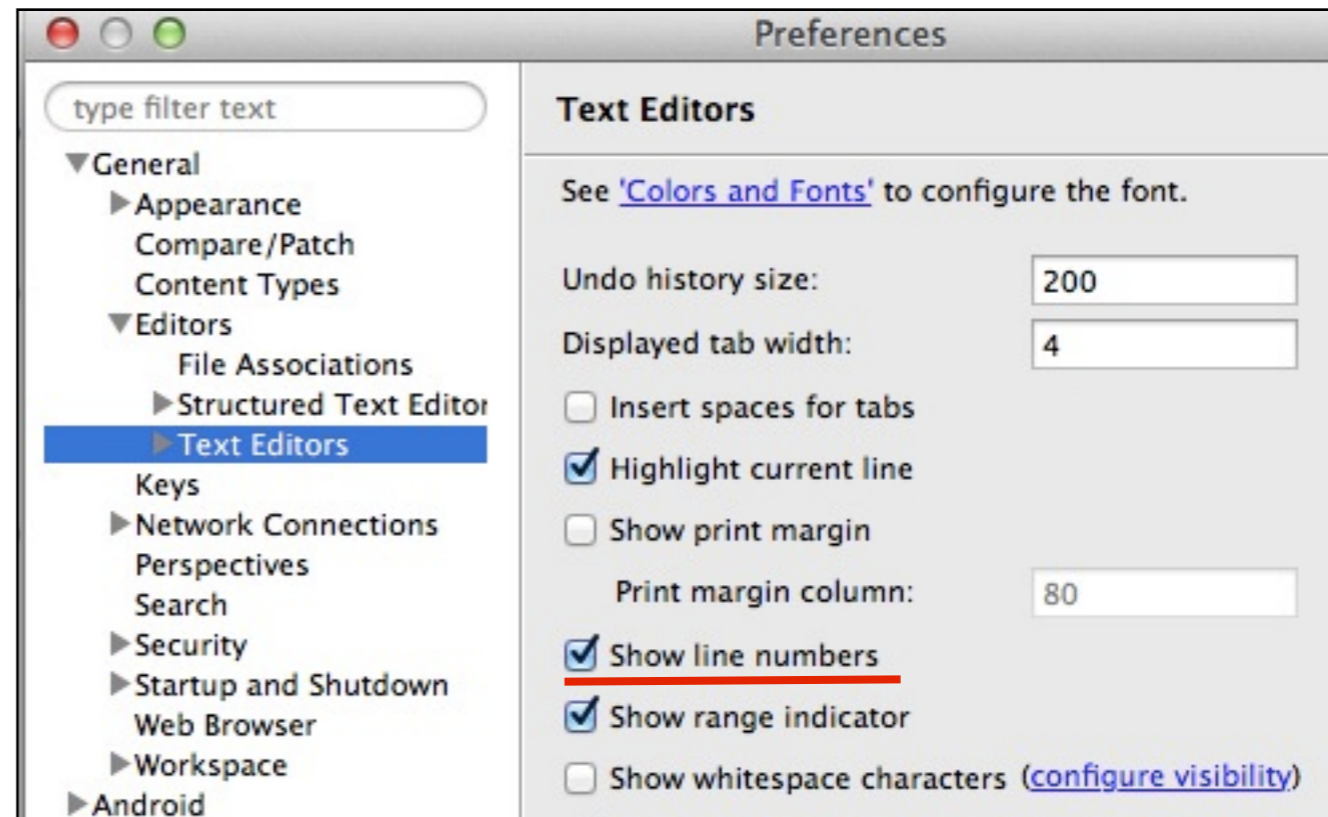
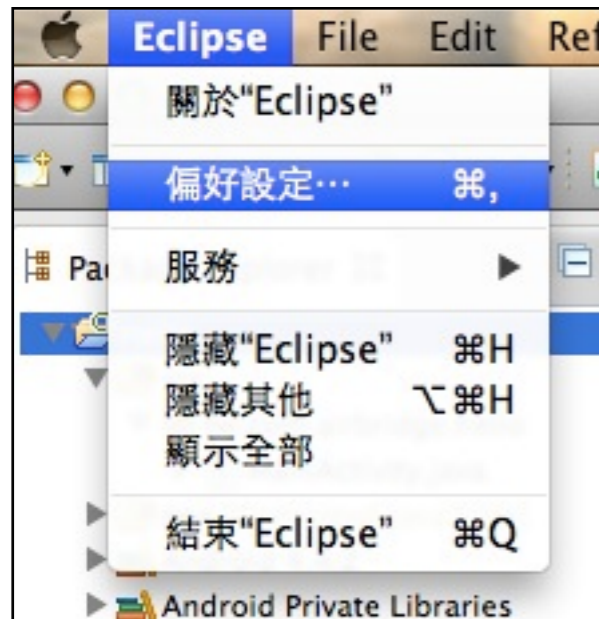
```
{
  "details":{
    "id"      : 書籍編號 ,
    "title"   : 書籍標題 ,
    "author"  : 作者姓名 ,
    "img"     : 書籍縮圖網址 ,
    "content" : 書籍說明內容
  },
  "imgs":[
    "http://www.memobook.com.tw/saved/14/140324/14032415240219/thumb/p1_5ObDvl_1.jpg",
    "http://www.memobook.com.tw/saved/14/140324/14032415240219/thumb/p2_yLkgAa_0.jpg",
    "http://www.memobook.com.tw/saved/14/140324/14032415240219/thumb/p3_byYIN2_5.jpg",
    ..... 上面列表是圖片位置 ( 文字 ) , 以此類推 ( 有多少頁就會撈多少頁 )
  ]
}
```

恭喜!你已經可以完成此專案!

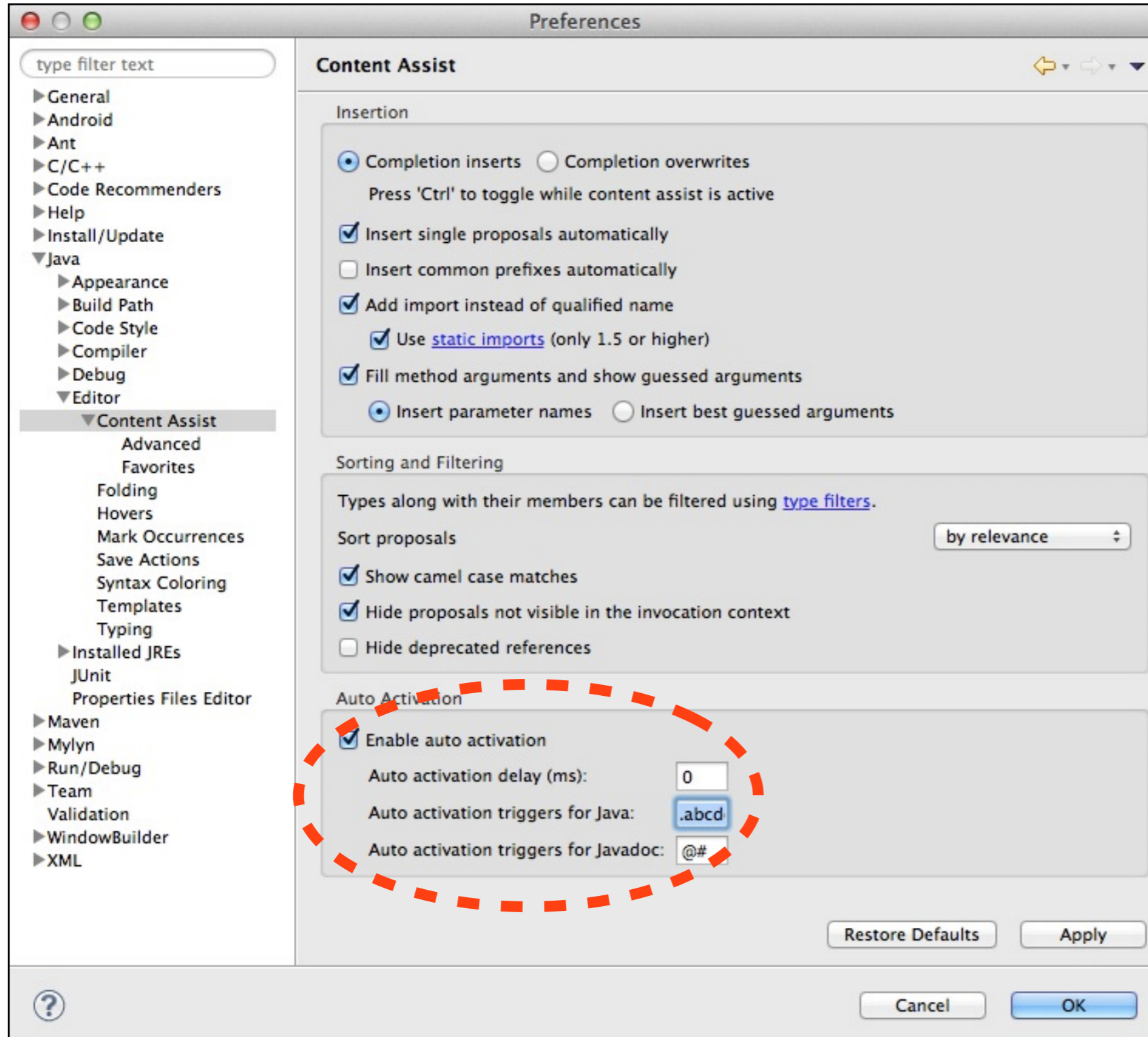
- 圖片載入(ImageView)
- 圖片滑動變換 (利用ViewFlipper)
- ListView (iOS的TableView)
- ListView Cell自定layout
- ListView Cell傳遞資料並加入動作

Tips

Tip: 顯示行號

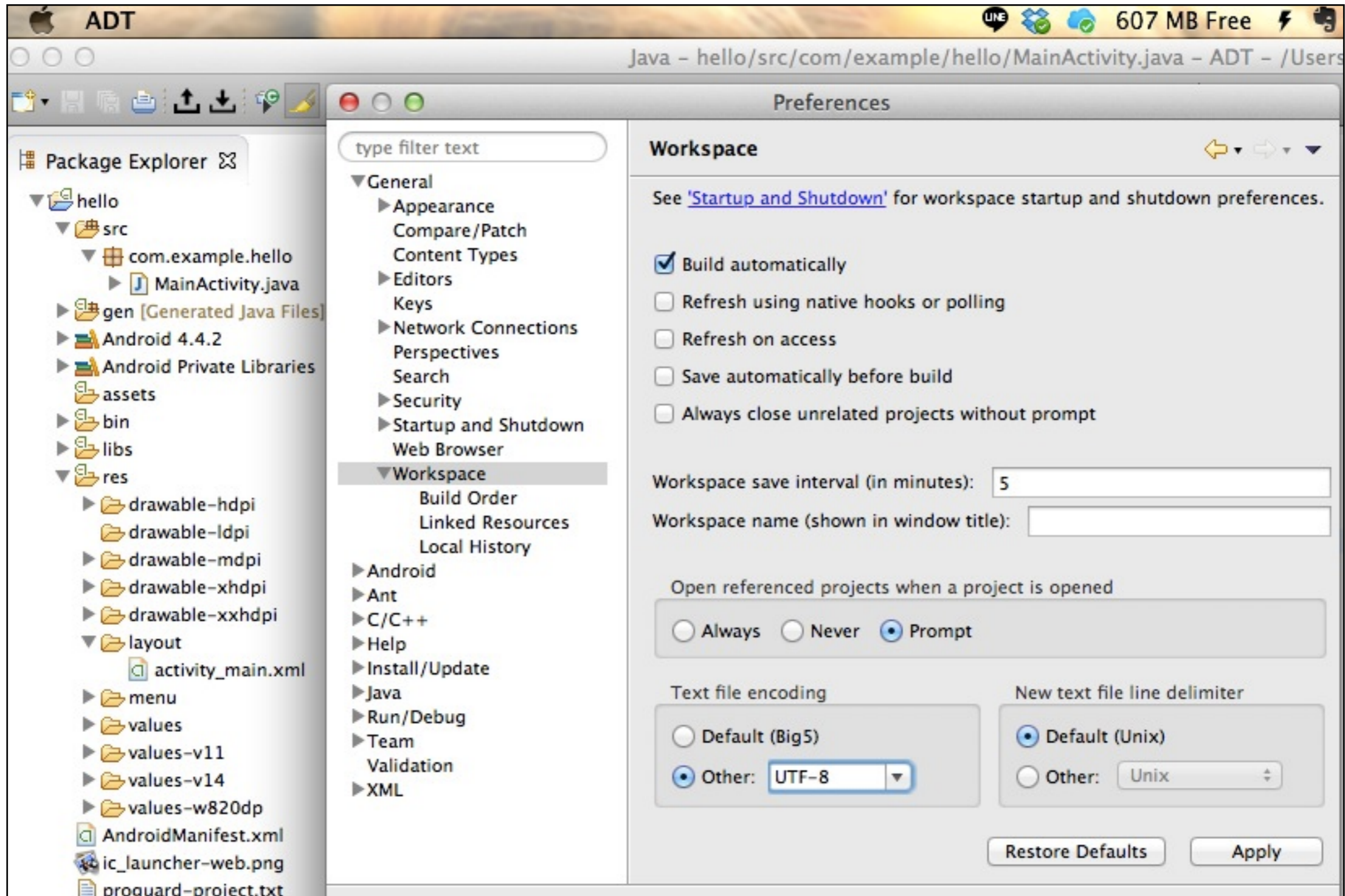


自動完成



or Alt+/

設定編碼為UTF-8



顯示繼承架構

The screenshot displays an IDE interface with two main panels. The left panel, titled 'Type Hierarchy', shows the class hierarchy for 'ImageView - android.widget'. It is structured as follows:

- Object
 - View
 - ImageView
 - ImageButton
 - QuickContactBadge

Below the hierarchy, the 'ImageView' class is selected, and its methods are listed:

- ImageView(Context)
- ImageView(Context, AttributeSet)
- ImageView(Context, AttributeSet, int)
- clearColorFilter() : void
- drawableStateChanged() : void
- getAdjustViewBounds() : boolean
- getBaseline() : int

The right panel shows the 'MainActivity.java' file. The code is as follows:

```
package com.example.loadimage;

import android.app.Activity;

public class MainActivity extends Activity {

    private ImageView myImageView;
    private String[] imagePics = {"img1.jpg", "img2.jpg"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        myImageView = findViewById(R.id.imageView);
        myImageView.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick() {
                myImageView.setImageDrawable(imagePics[0]);
            }
        });
    }
}
```

A context menu is open over the code, listing various actions and their keyboard shortcuts:

- Undo Typing (⌘Z)
- Revert File
- Save (⌘S)
- Open Declaration (F3)
- Open Type Hierarchy (F4)
- Open Call Hierarchy (⌘H)
- Show in Breadcrumb (⌘B)
- Quick Outline (⌘O)
- Quick Type Hierarchy (⌘T)
- Open With
- Show In (⌘W)
- Cut (⌘X)
- Copy (⌘C)
- Copy Qualified Name
- Paste (⌘V)

存入Snippets

The screenshot shows an IDE window with a code editor and a context menu. The code editor contains the following Java code:

```
myListView.setOnItemClickListener(itemClickListener);  
  
//變數宣告  
DisplayImageOptions options;  
final ImageLoader imageLoader = ImageLoader.getInstance();  
  
// 1. 設定display option  
options = new DisplayImageOptions.Builder()  
    .showImageForEmptyUri(R.drawable.ic_launcher) //若網址不存在的圖片  
    .showImageOnFail(R.drawable.ic_launcher) //若載入錯誤的圖片  
    .cacheInMemory(true) //用記憶體暫存  
    .cacheOnDisk(true) //存在本機  
    .build();  
  
//2. 設定imageLoaderConfiguration  
ImageLoaderConfiguration config = new ImageLoaderConfiguration.Builder(g  
    .defaultDisplayImageOptions(options) //設定內定的display option  
    .build();  
  
imageLoader.init(config); //初始化  
  
//取得JSON Response  
AsyncHttpClient client = new AsyncHttpClient();  
//設定參數  
RequestParams params= new RequestParams("act","memo_list");  
//使用get方式  
client.get("http://images.memobook.com.tw/m/api.php" params new JsonHttpRespon
```

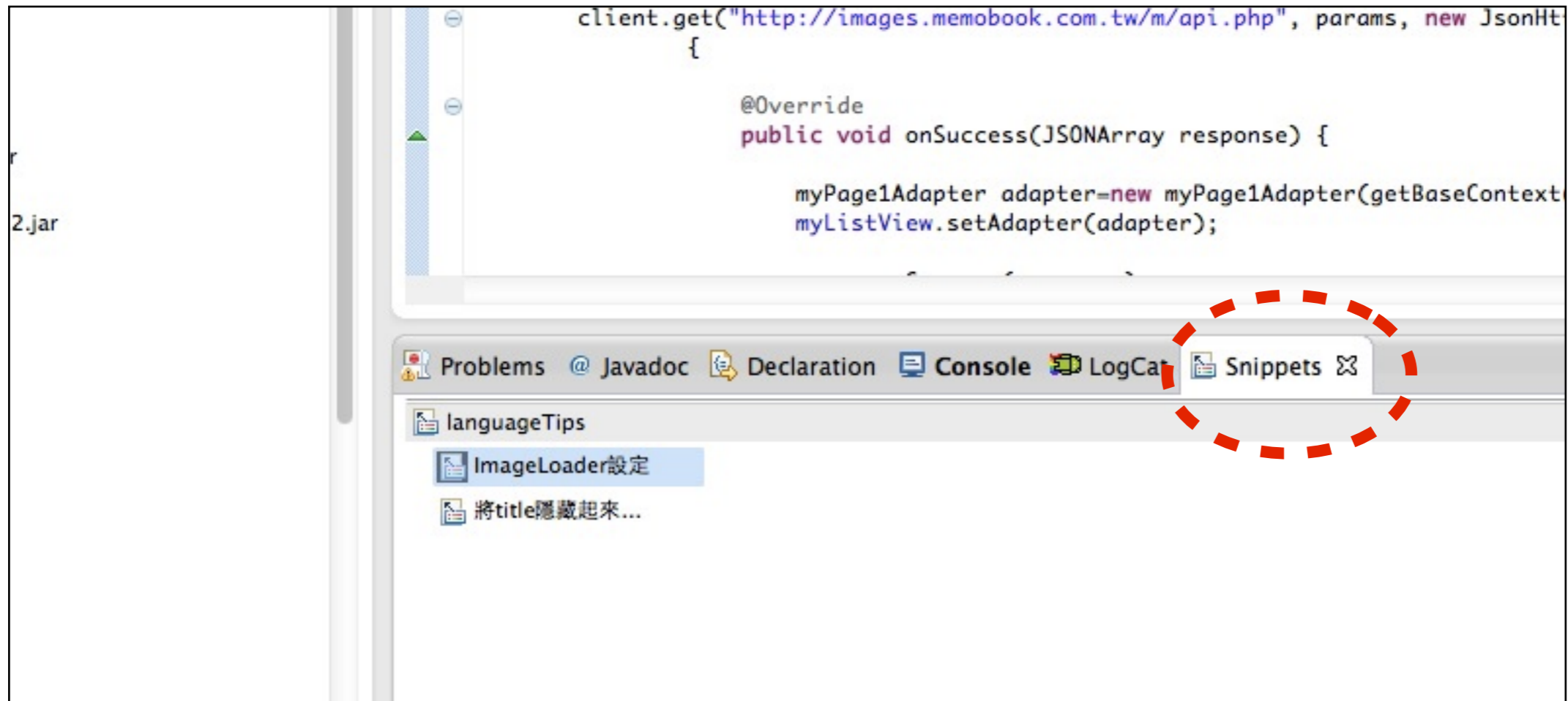
The context menu is open, showing various actions. The "Add to Snippets..." option is highlighted in blue. Other visible options include:

- Undo Typing (⌘Z)
- Revert File
- Save (⌘S)
- Open Declaration (F3)
- Open Type Hierarchy (F4)
- Open Call Hierarchy (^⌘H)
- Show in Breadcrumb (⌘B)
- Quick Outline (⌘O)
- Quick Type Hierarchy (⌘T)
- Open With (▶)
- Show In (⌘W ▶)
- Cut (⌘X)
- Copy (⌘C)
- Copy Qualified Name
- Paste (⌘V)
- Quick Fix (⌘1)
- Source (⌘S ▶)
- Refactor (⌘T ▶)
- Surround With (⌘Z ▶)
- Local History (▶)
- References (▶)
- Declarations (▶)
- Add to Snippets... (highlighted)
- Profile As (▶)

The IDE interface includes a bottom toolbar with tabs for Problems, Javadoc, Declaration, Console, LogCat, and Snippets. A search bar for messages is also visible, along with a table header for LogCat messages:

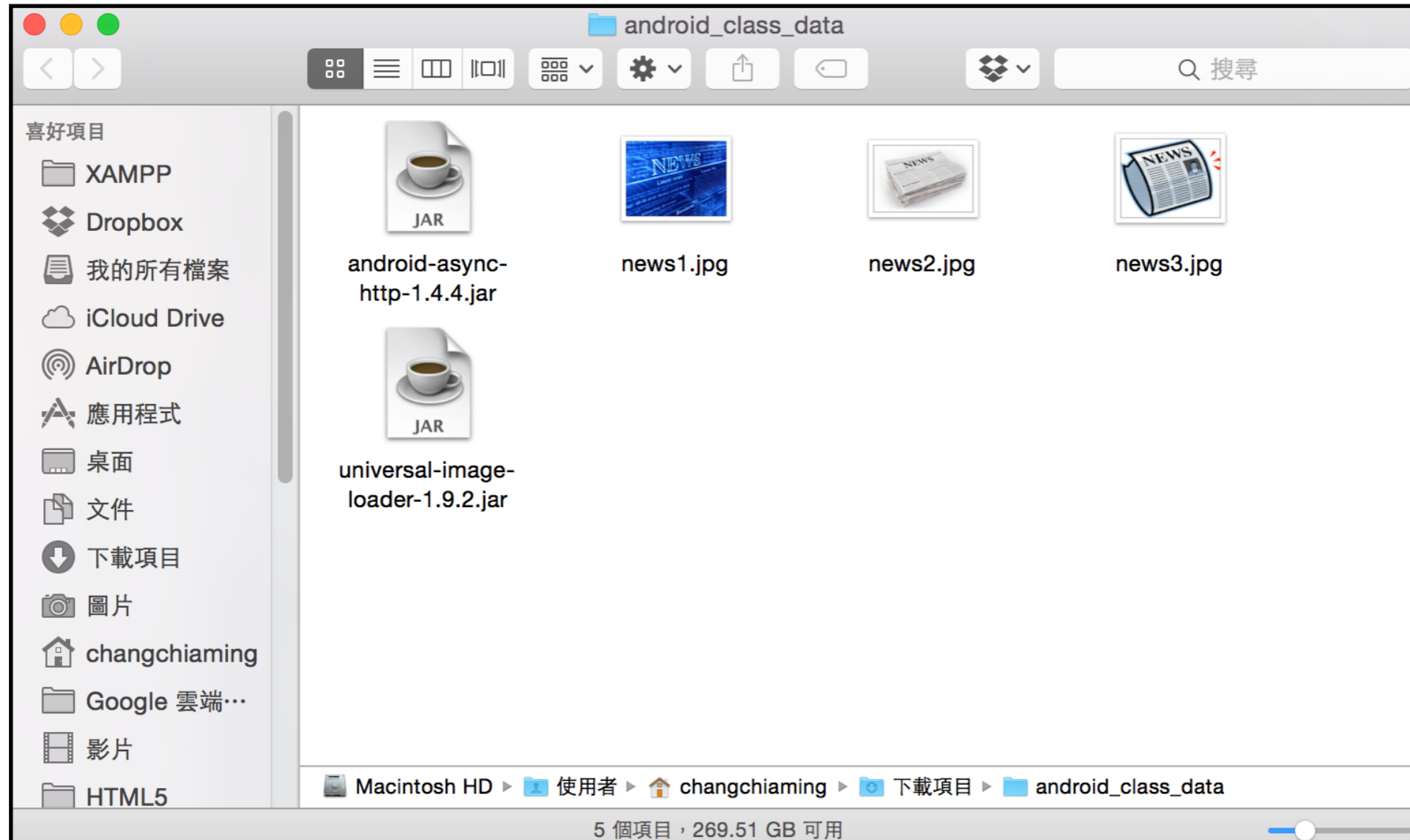
Level	Time	PID	TID	Application
-------	------	-----	-----	-------------

使用Snippets



下載使用資料

- <http://goo.gl/iwnL4Y>



3張圖以及2個opensource