

# 手持式裝置之應用與實作

## ~ Android 行動瀏覽器 ~

王昱景 Brian Wang

[brianwang@isoftstone.com](mailto:brianwang@isoftstone.com)

# WebView 元件

- 在 Android 應用程式可以使用 **WebView** 元件來瀏覽網頁內容
- WebView 是 **android.webkit** 套件 (Package) 的類別
- 使用 **WebKit** 技術的 View

# WebViewDemo

- 開啟和執行 Android 專案
- 建立 WebView 元件的版面配置
- 建立 Activity 活動類別
- 在 AndroidManifest.xml 新增存取 Internet 權限

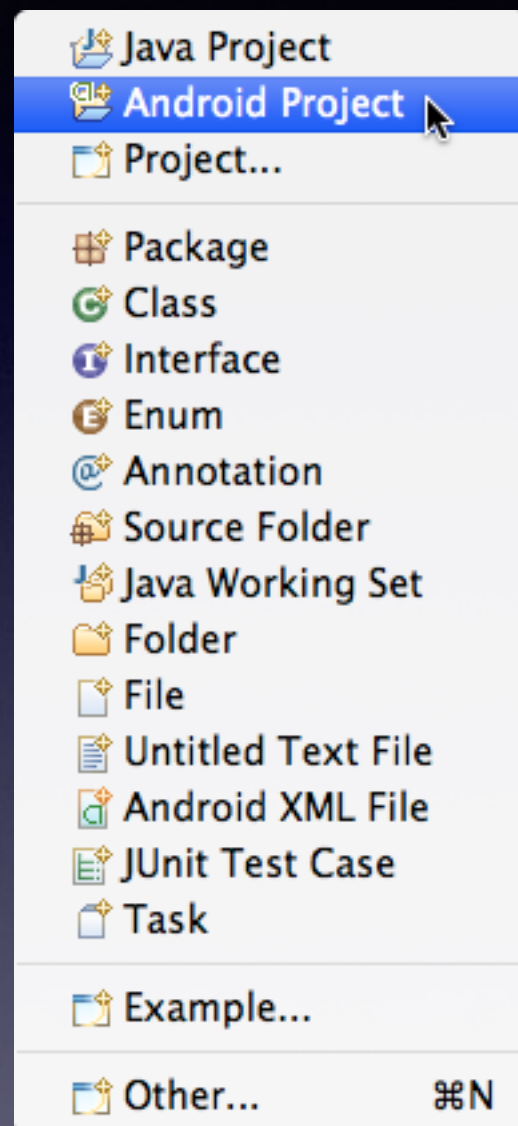
# I. 開啟和執行 Android 專案

- 請啟動 Eclipse IDE
- 建立 Android 專案
  - Project Name: **WebViewDemo**
  - Build Target: **Android 4.0.3**
  - Package Name: **tw.edu.vnu**

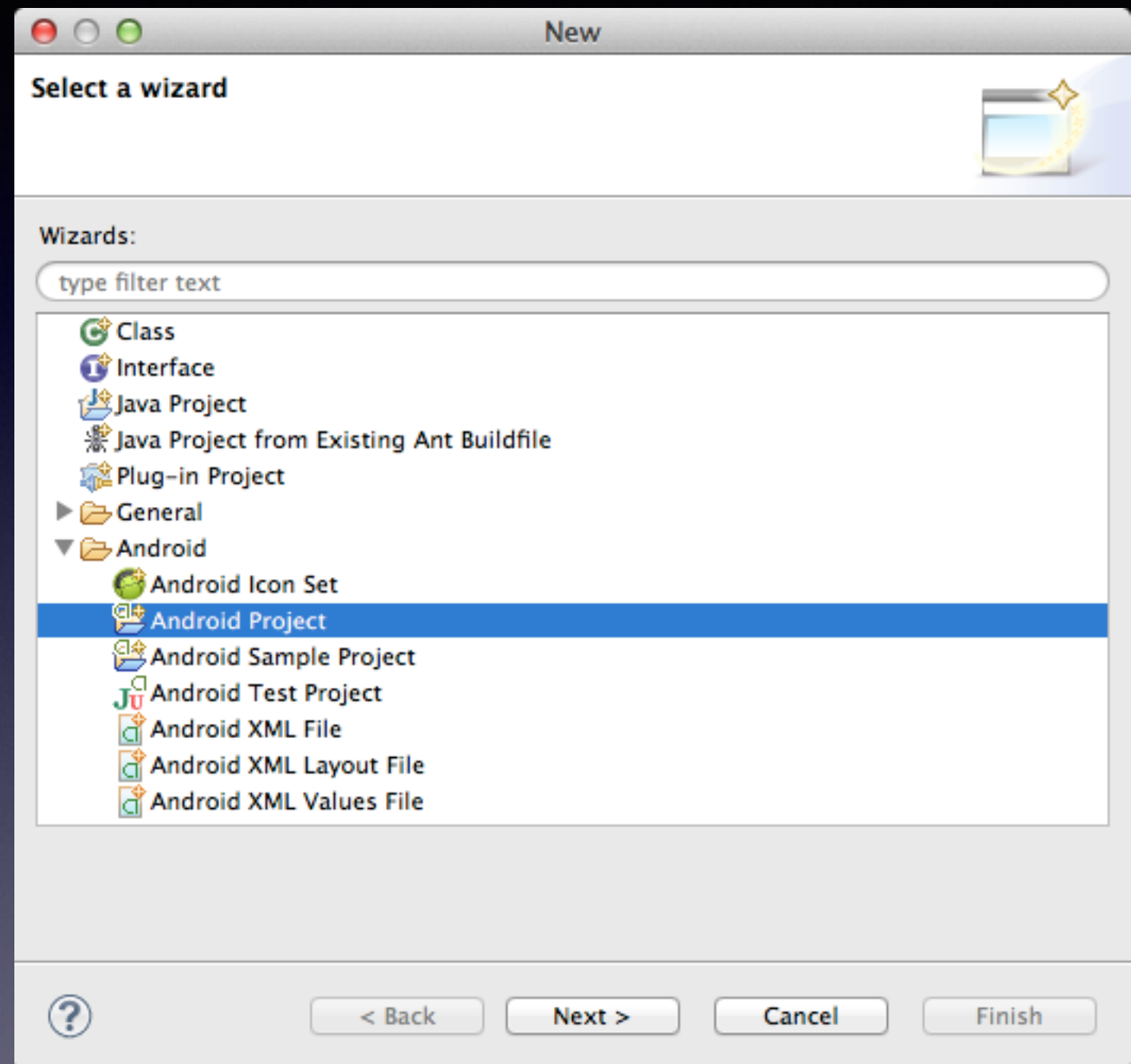




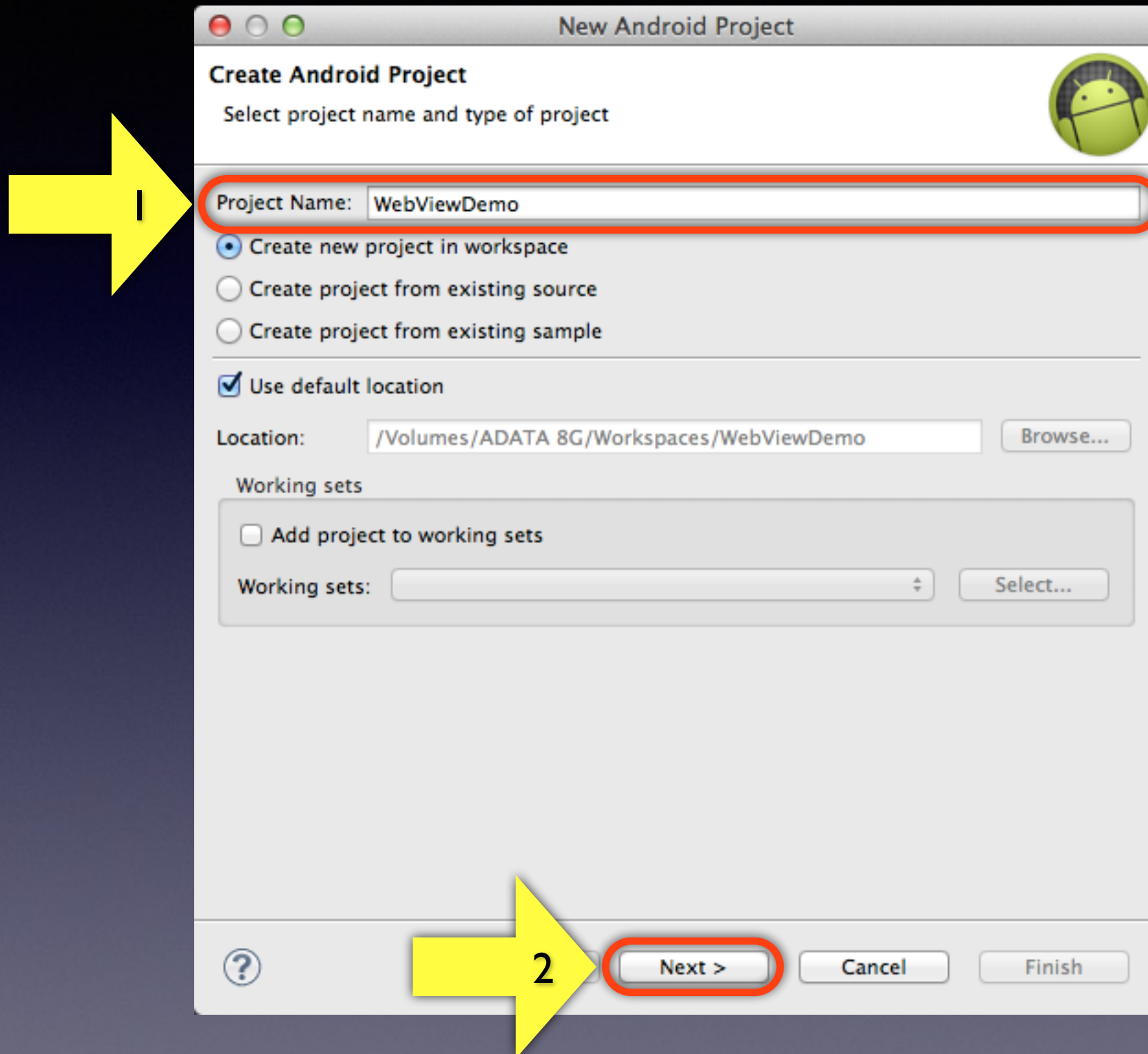




方法1

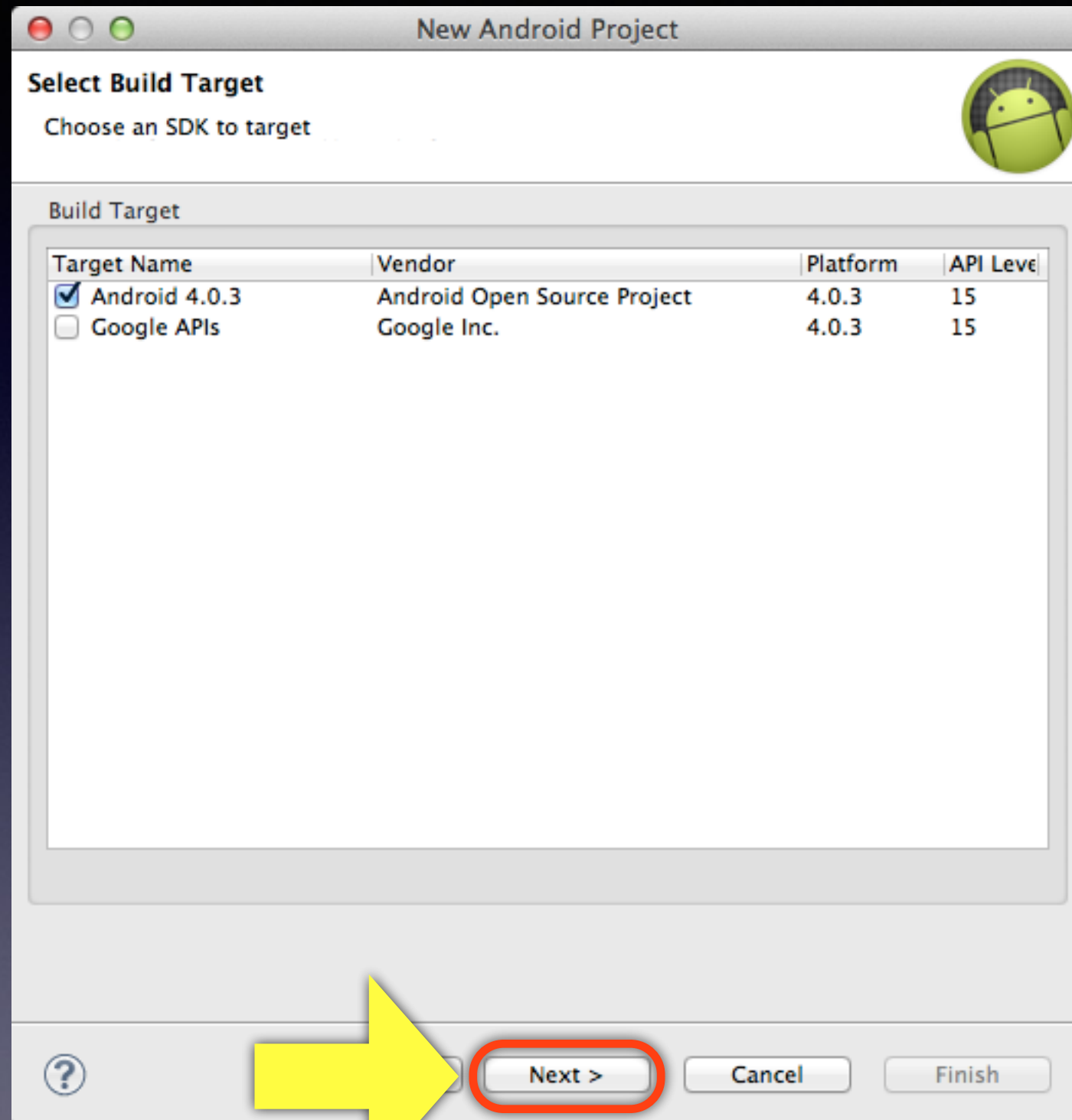


方法2



Project Name: **WebViewDemo**





New Android Project

**Application Info**  
Configure the new Android Project

Application Name: WebViewDemo

**Package Name:** tw.edu.vnu

☒ Create Activity: WebViewDemoActivity

Minimum SDK: 15

☐ Create a Test Project

Test Project Name: WebViewDemoTest

Test Application: WebViewDemoTest

Test Package: tw.edu.vnu.test

? < Back Next > **2** Finish

Package Name: tw.edu.vnu

## 2.建立 WebView 元件的版面配置

- 使用 LinearLayout 垂直編排一個 RelativeLayout 和 WebView 元件
- 在 RelativeLayout 水平編排 EditText 和 Button 元件

Title

LinearLayout

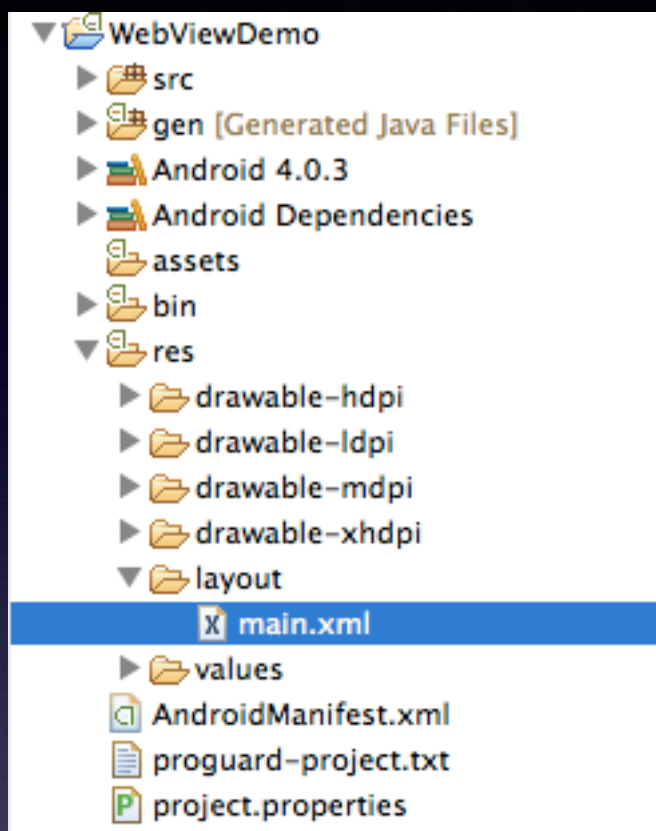
Title

RelativeLayout

WebView

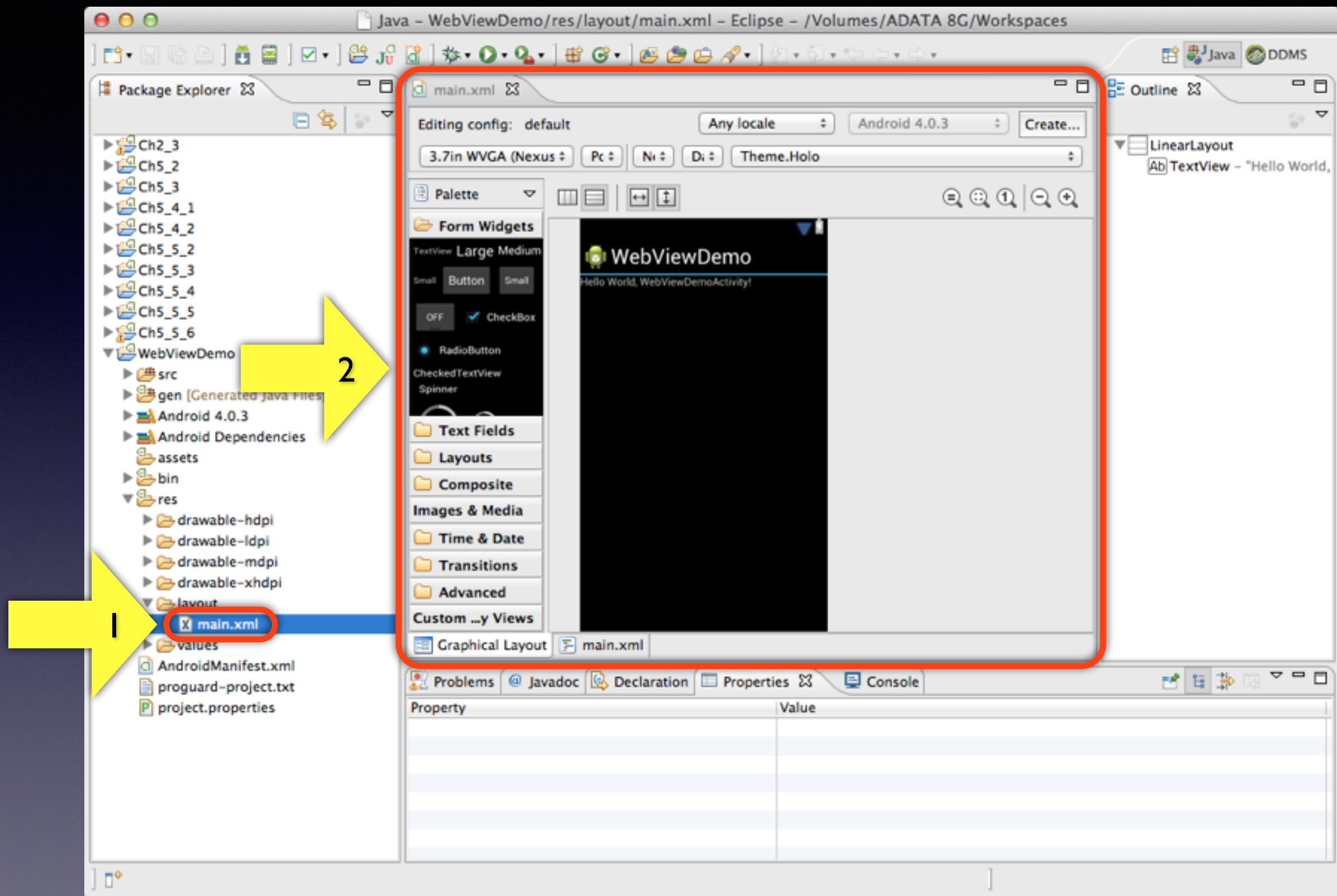


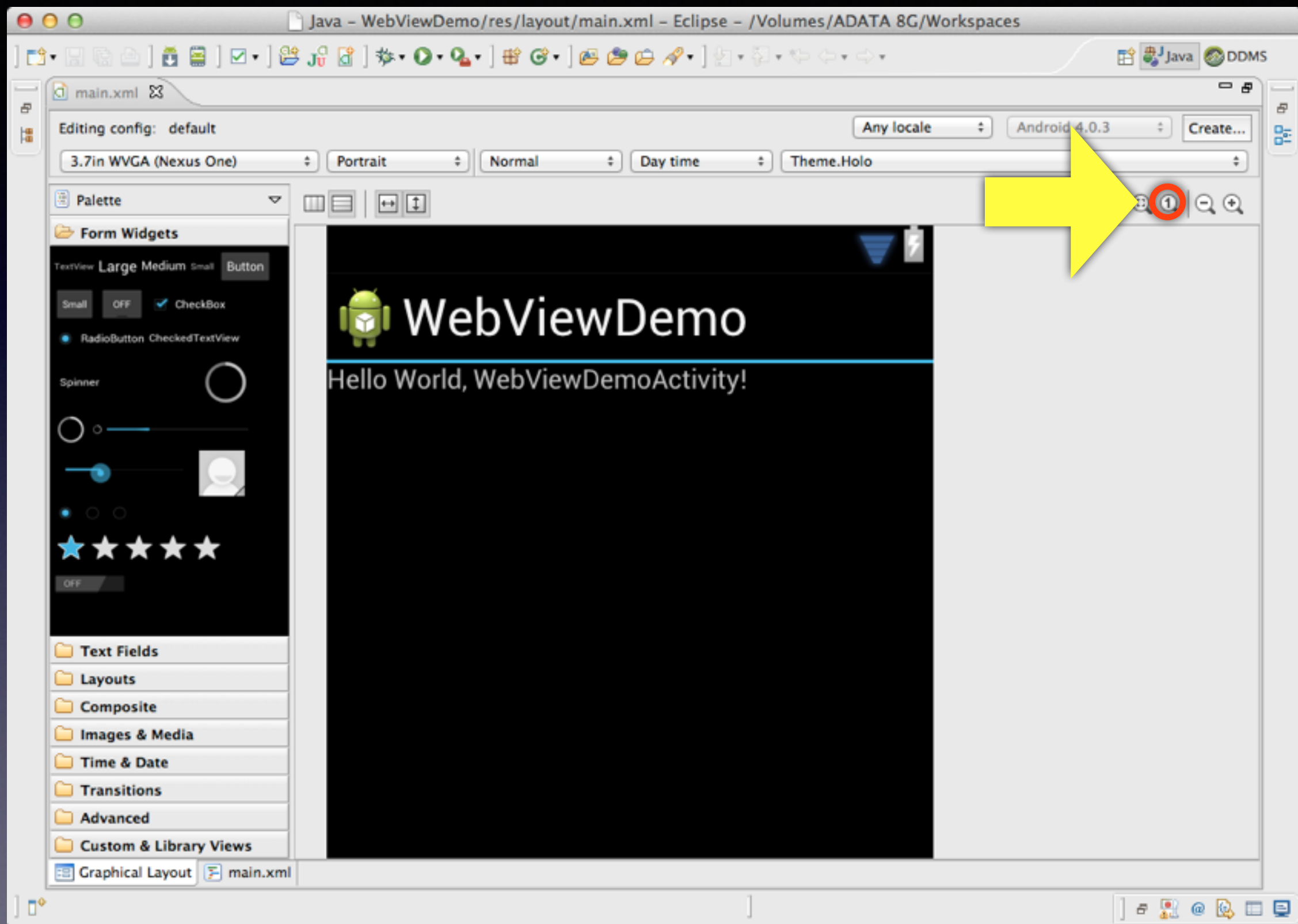


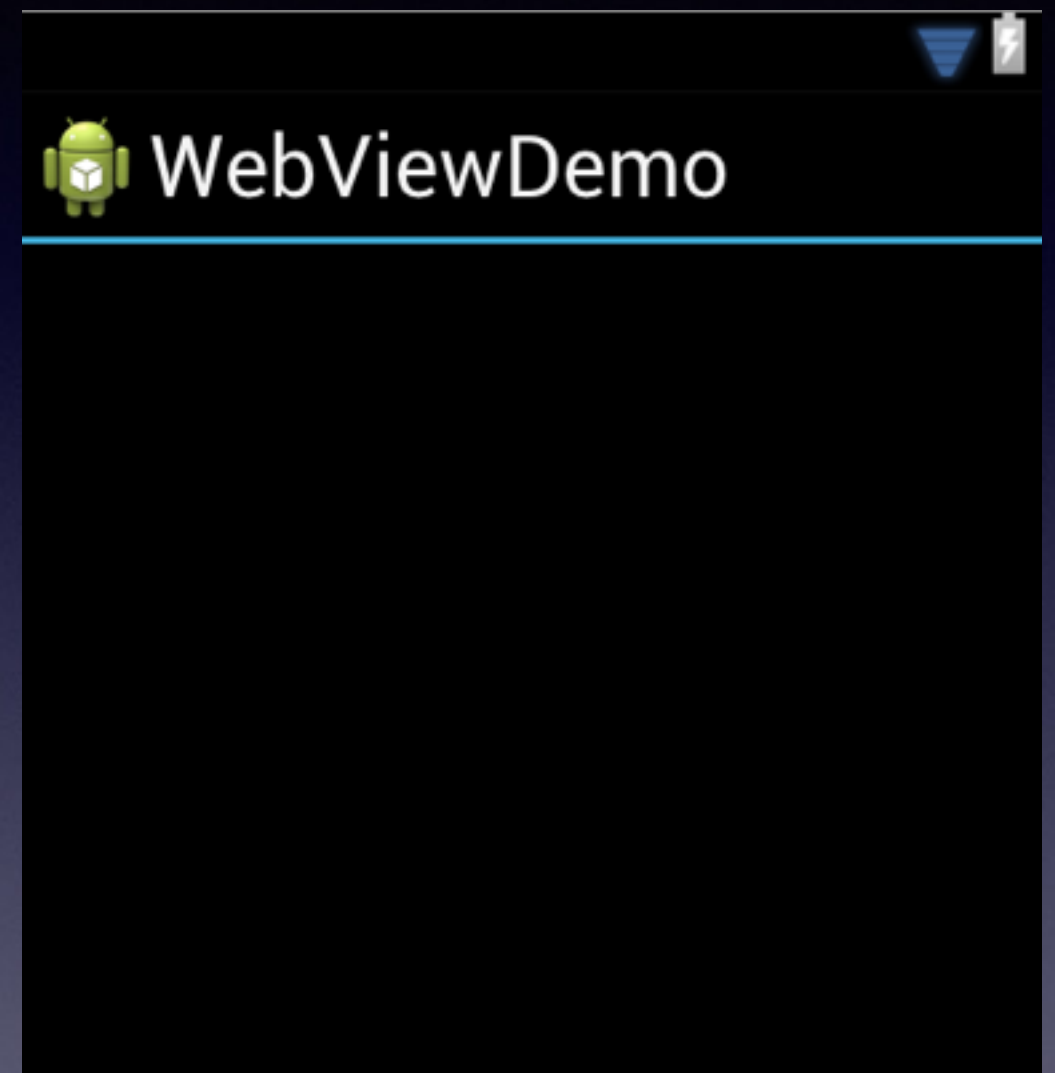
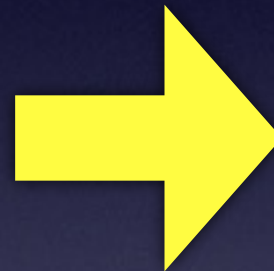
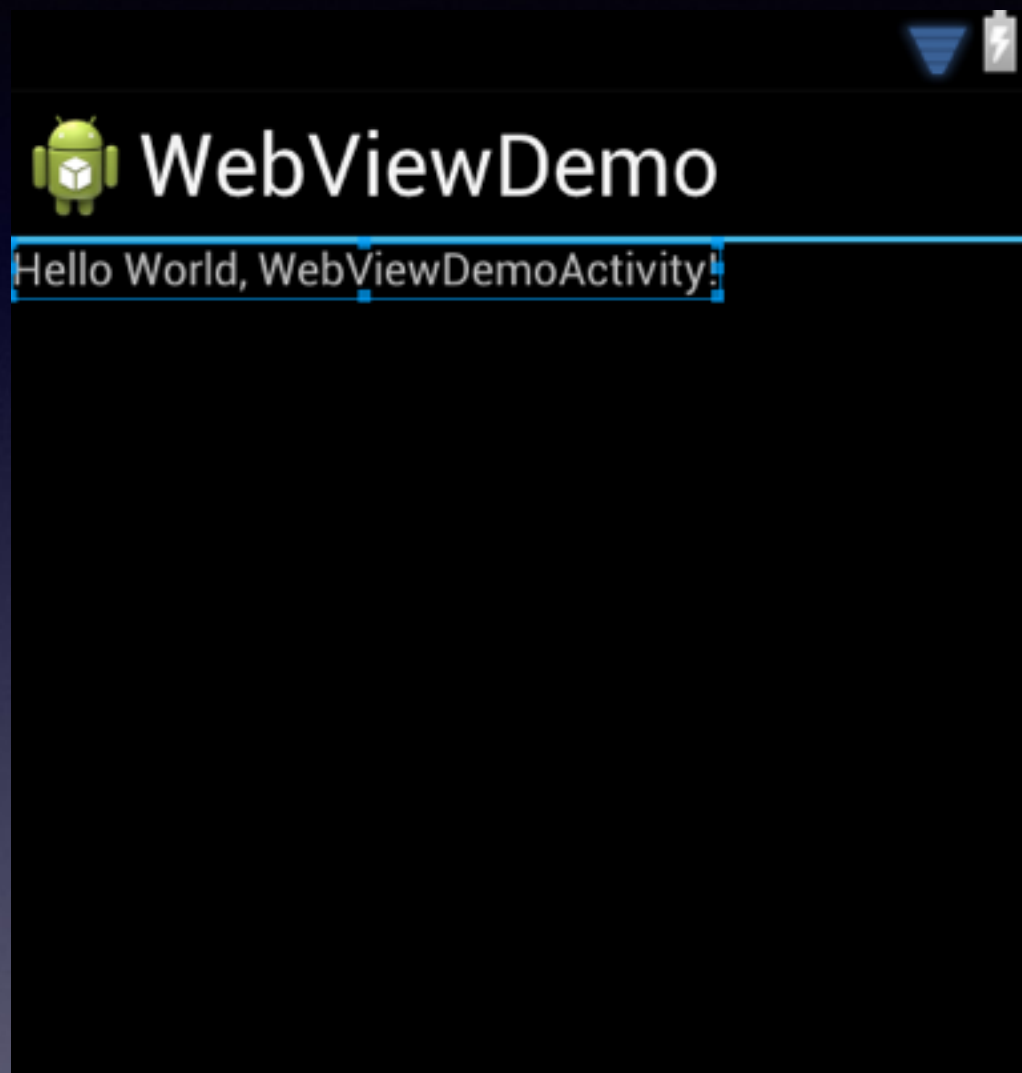


展開 WebViewDemo 專案

開啟 res > layout 下的 **main.xml**

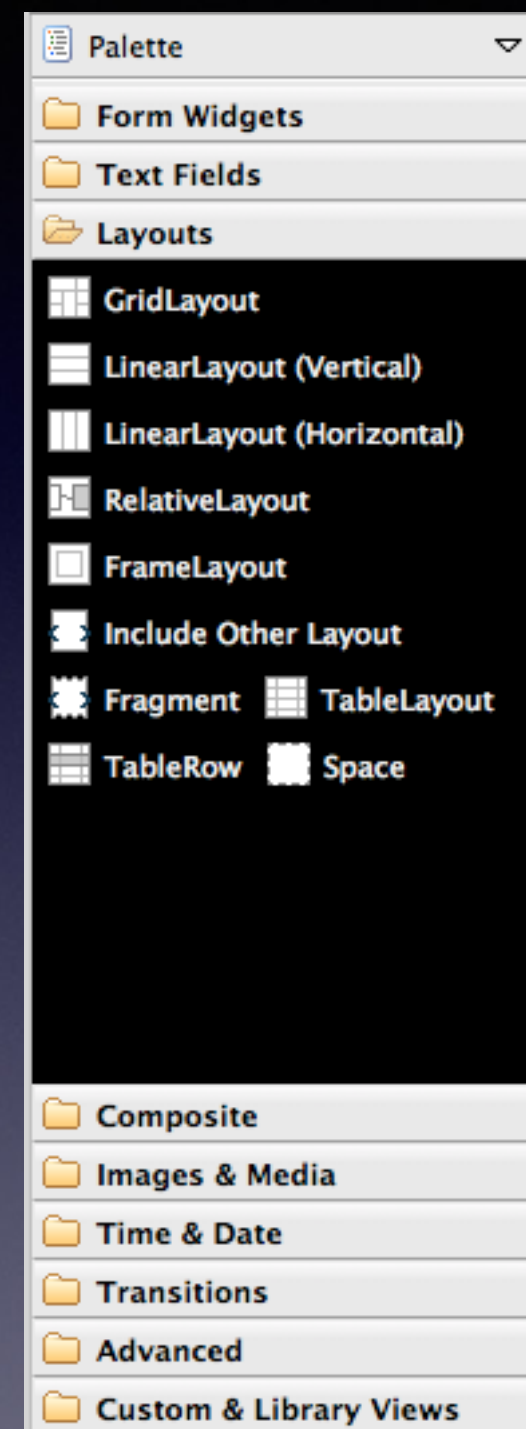
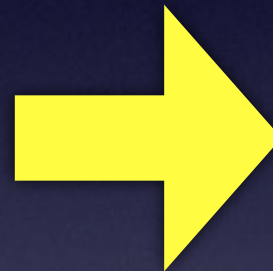
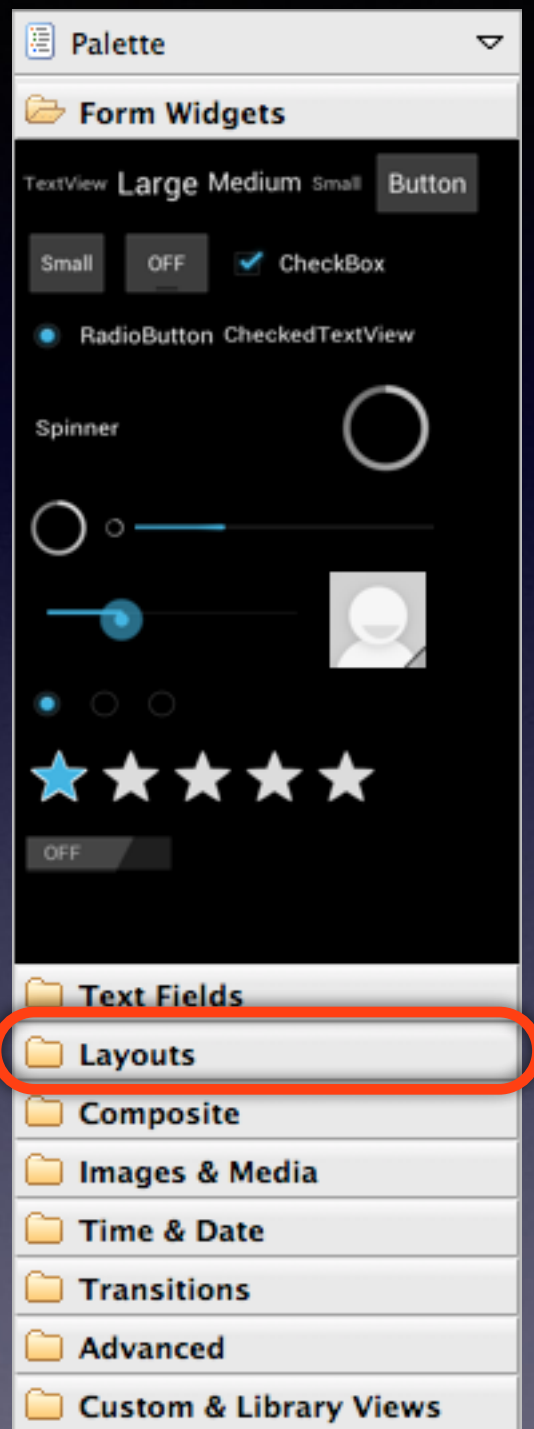




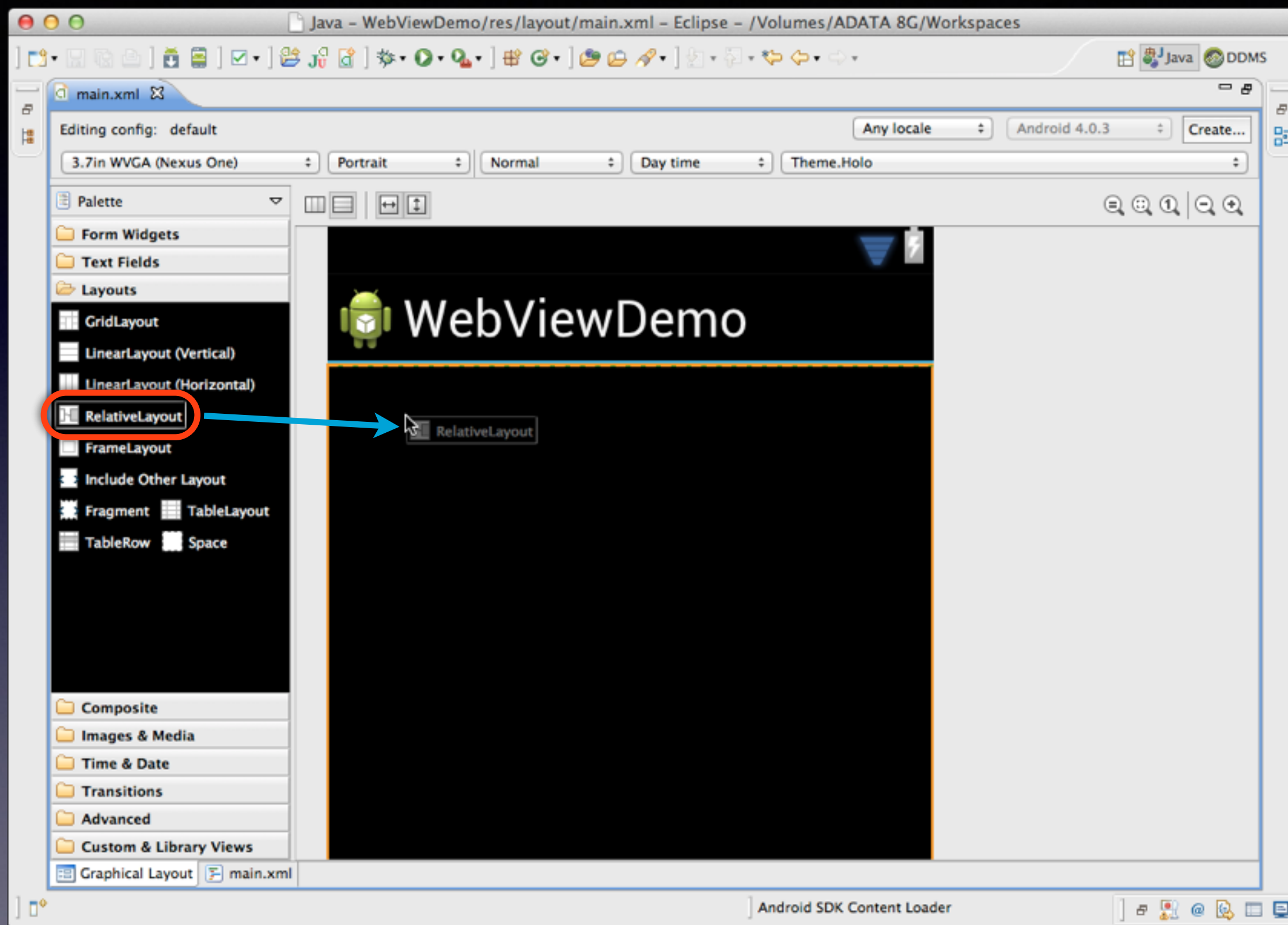


點選 TextView > Del 鍵刪除

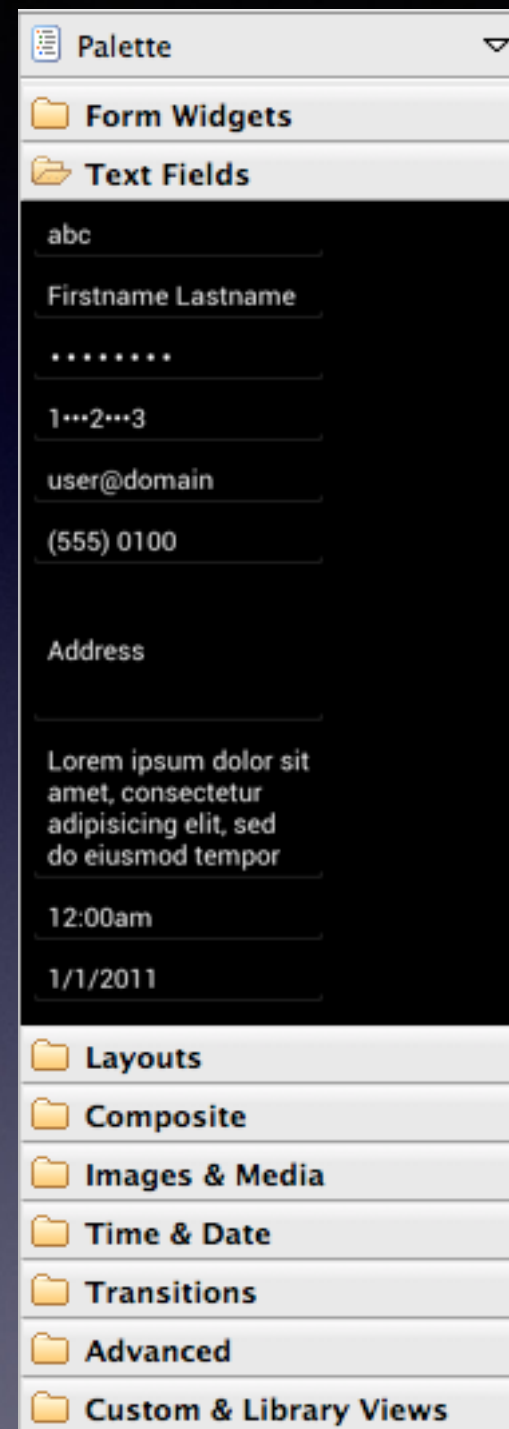
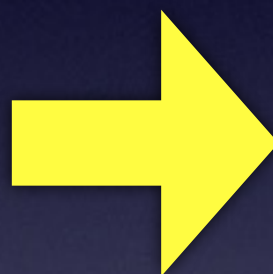
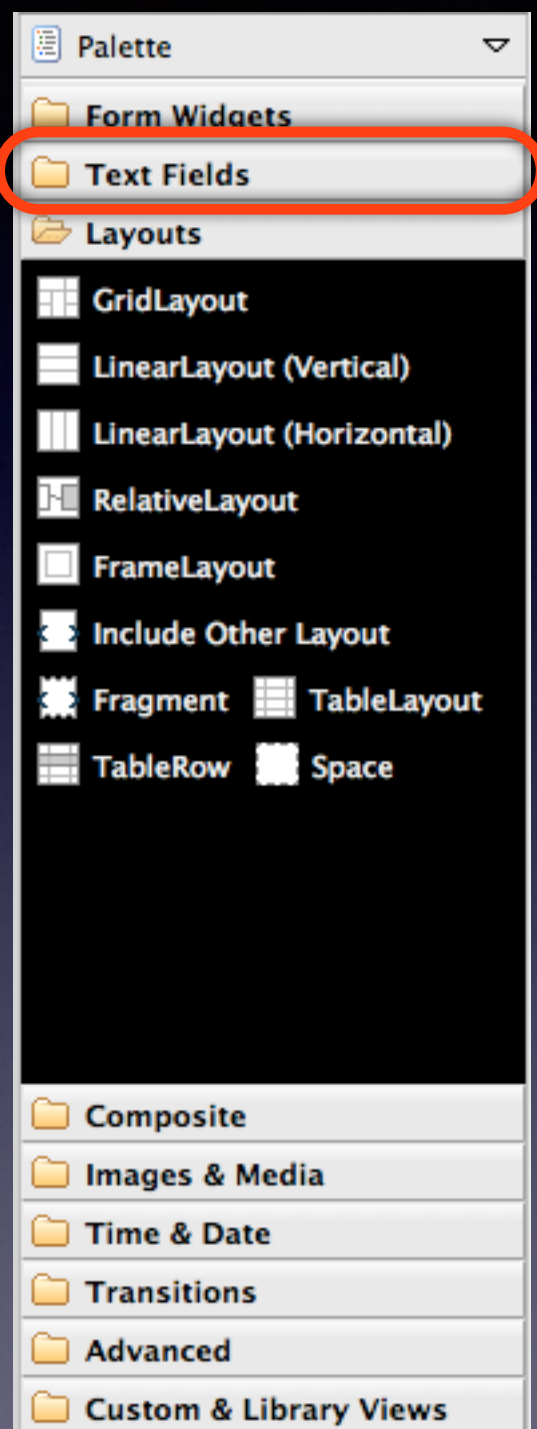




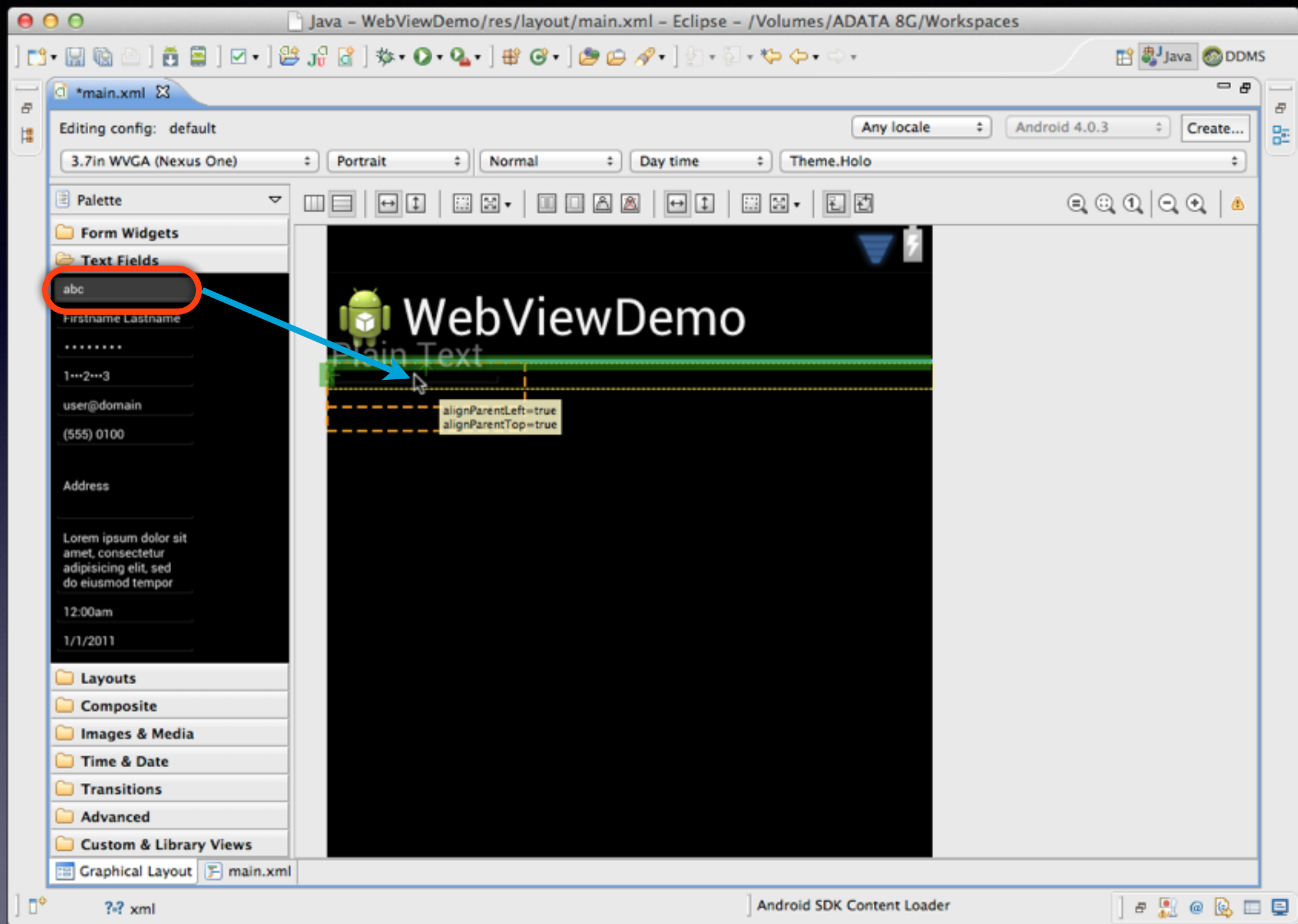
點選 Palette 的 Layouts，將 Layouts 展開



新增 RelativeLayout 至畫面上

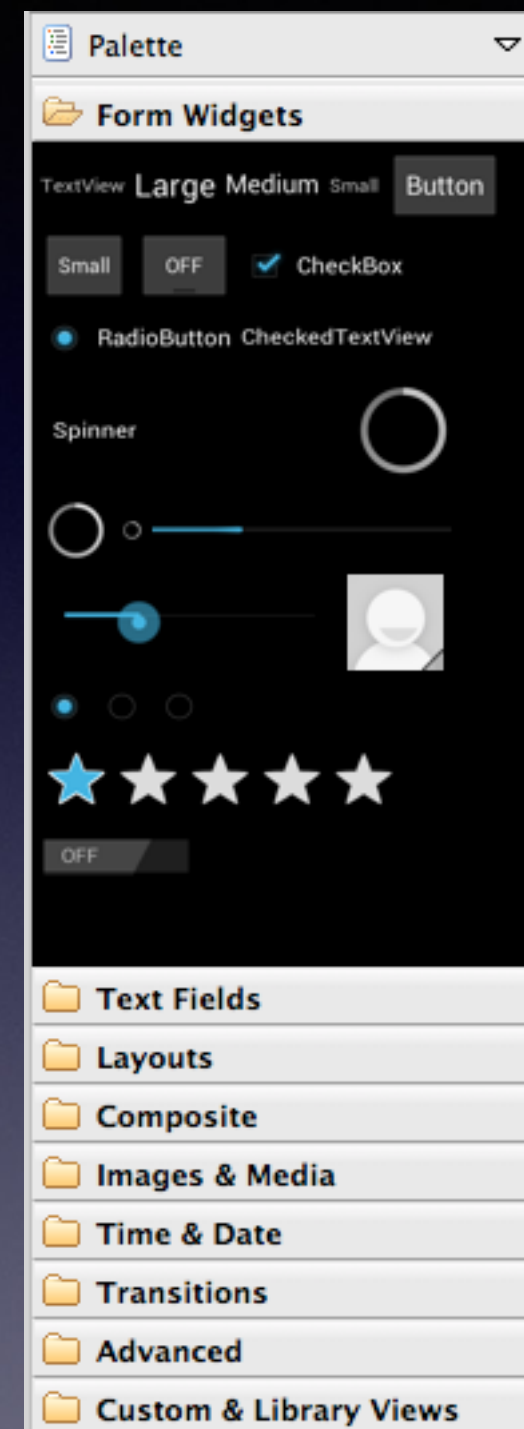
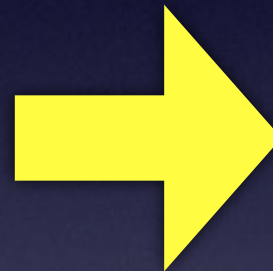
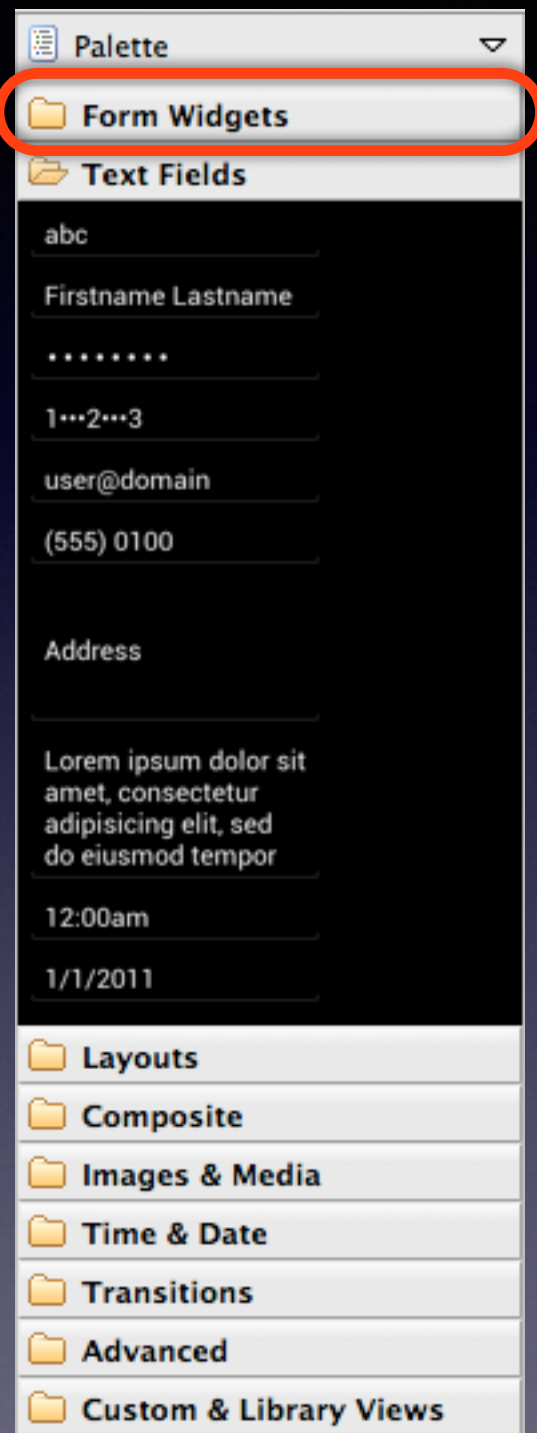


點選 Palette 的 Text Fields，將 Text Fields 展開



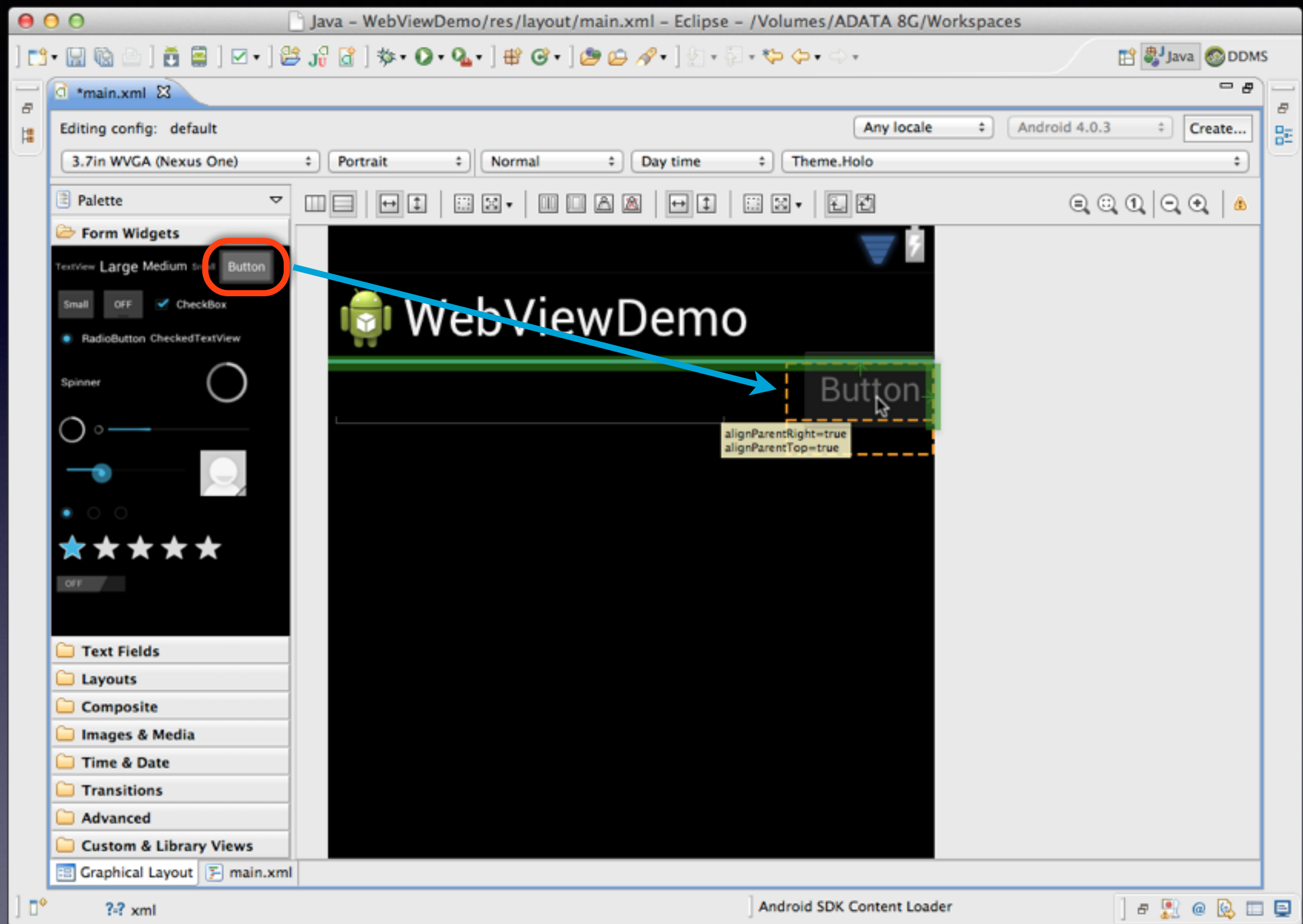
新增 EditText 元件至 RelativeLayout



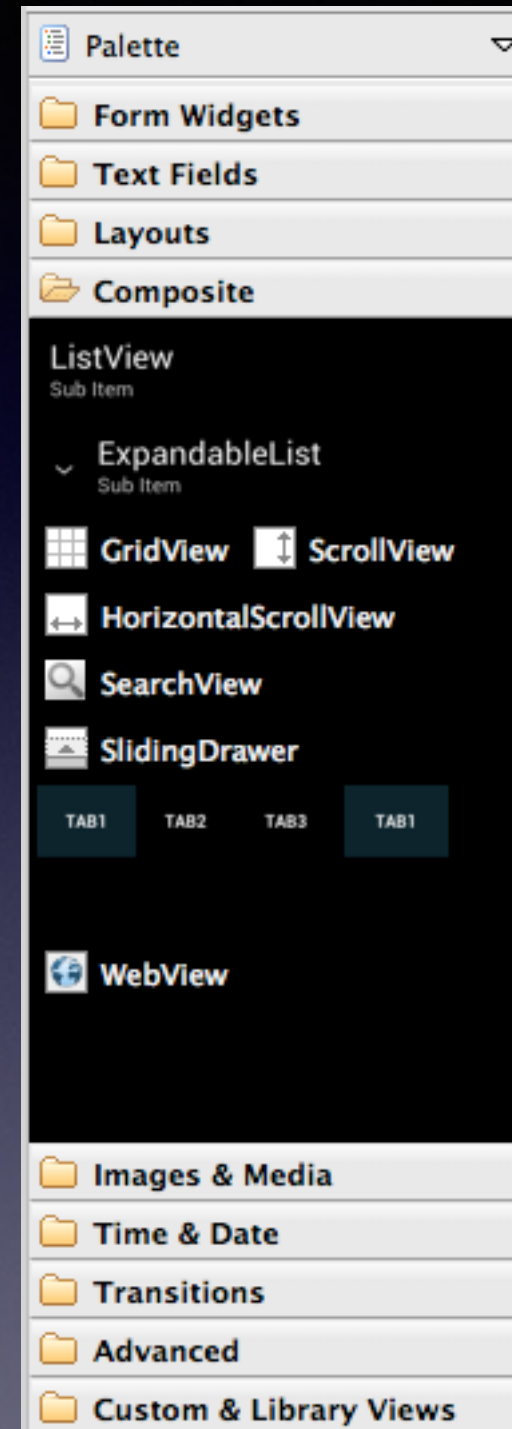
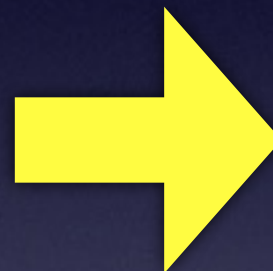
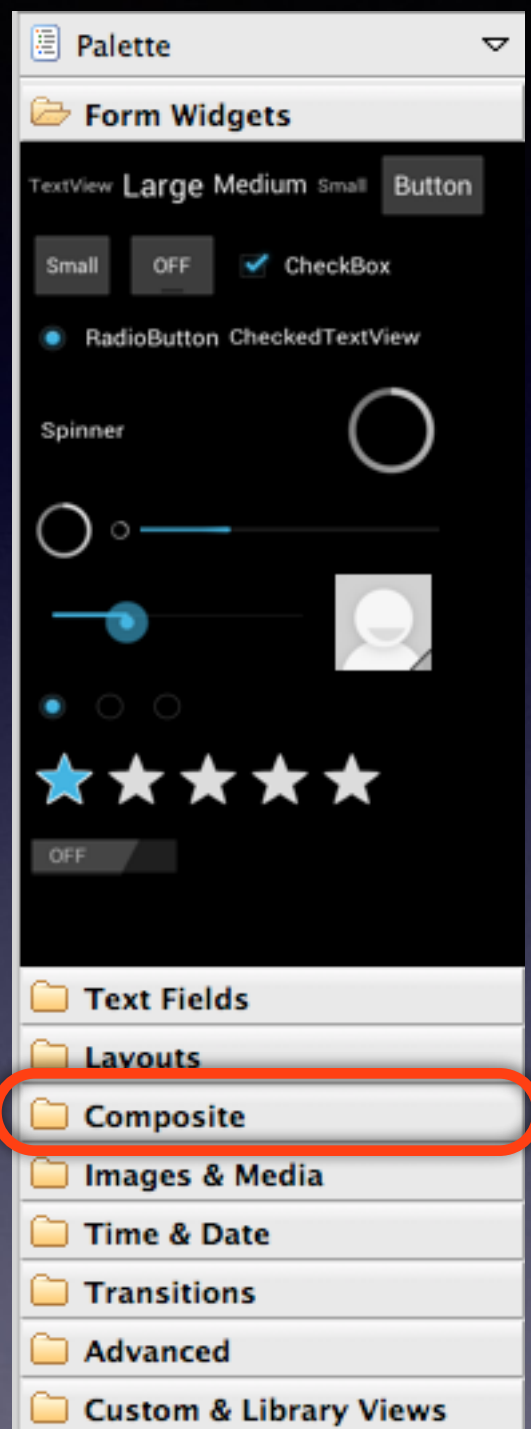


點選 Palette 的 Form Widgets，將 Form Widgets 展開

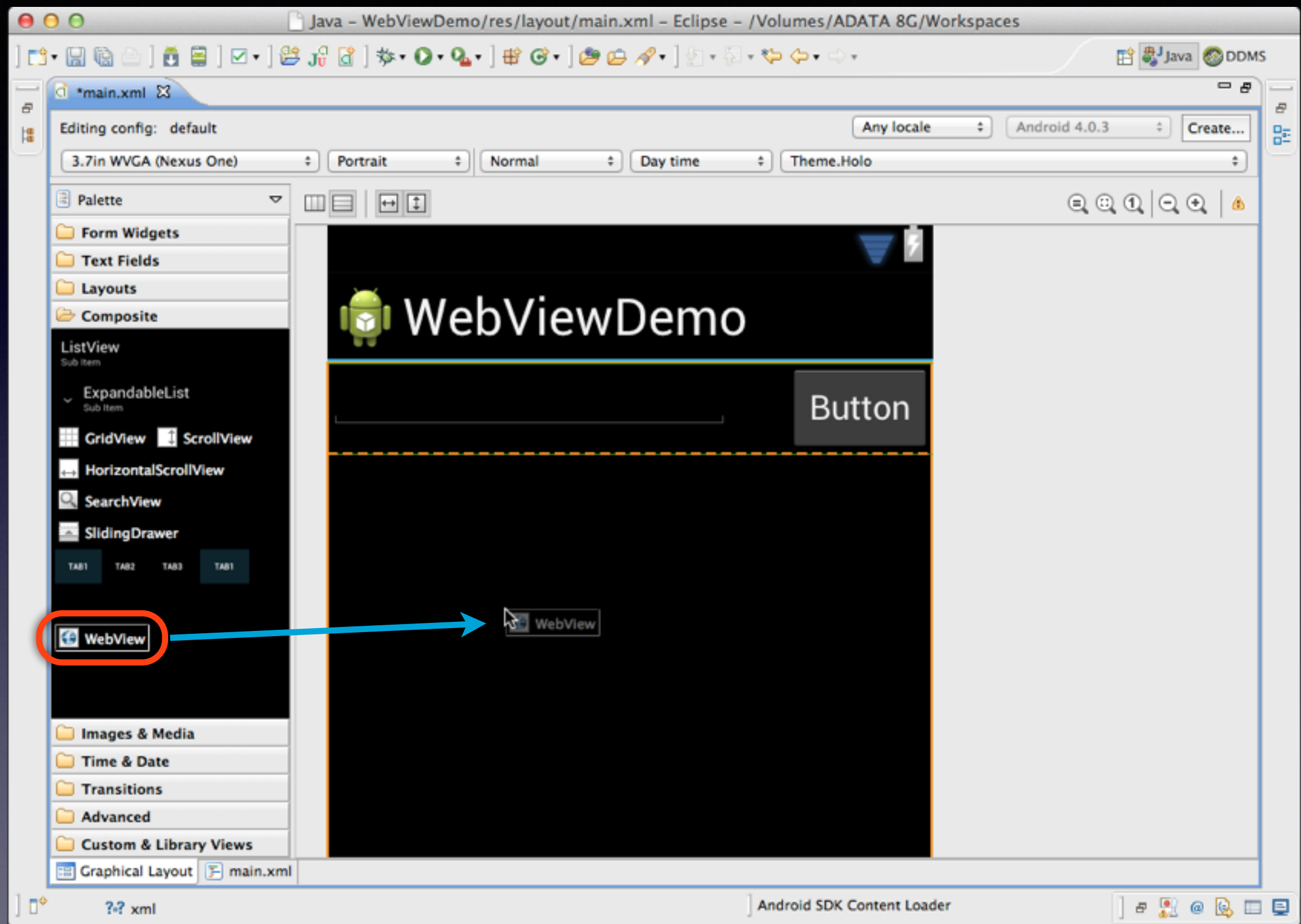




新增 Button 元件於 EditText 右方



點選 Palette 的 Composite，將 Composite 展開

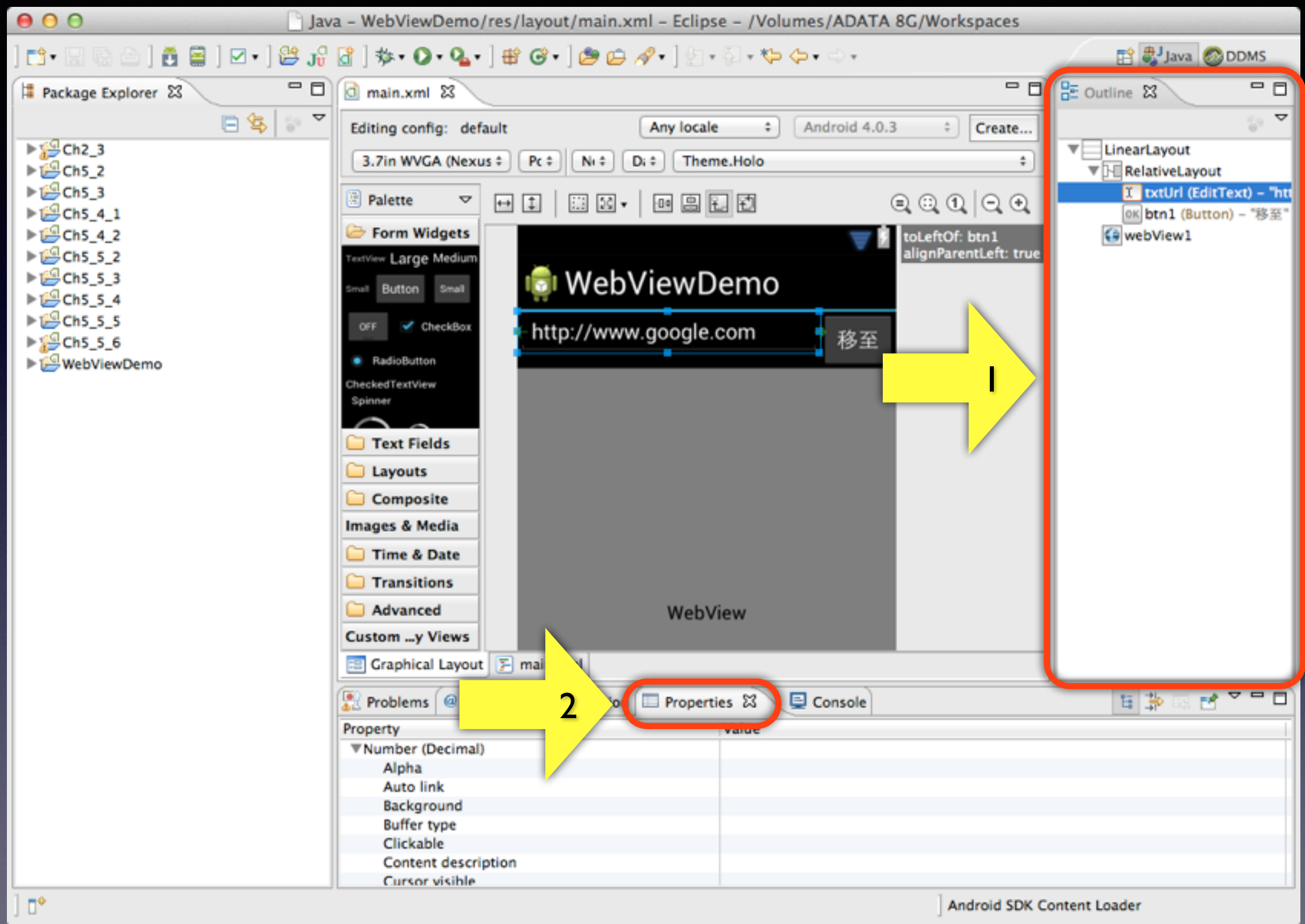


新增 WebView 元件於 RelativeLayout 下方

# 版面配置參數設定

- EditText
  - id: @+id/txtUrl
  - inputType: textUri
  - text: <http://www.google.com>
  - layout\_toLeftOf: @+id/btnI
  - layout\_alignParentLeft: true
- Button
  - id: @+id/btnI
  - onClick: btnI\_Click
  - text: 移至
  - layout\_alignParentRight: true





點選右方 Outline 的 Layout 元件修改元件的 Properties



Java - Eclipse - /Volumes/ADATA 8G/Workspaces

Problems Javadoc Declaration Properties Console

Property	Value
▼ Number (Decimal)	
Alpha	
Auto link	
Background	
Buffer type	
Clickable	
Content description	
Cursor visible	
Digits	
Drawable bottom	
Drawable end	
Drawable left	
Drawable padding	
Drawable right	
Drawable start	
Drawable top	
Drawing cache quality	
Duplicate parent state	
Editor extras	
Ellipsize	
Ems	
Fade scrollbars	
Fading edge length	
Filter touches when obscured	
Fits system windows	
Focusable	
Focusable in touch mode	
Freezes text	
Gravity	
Haptic feedback enabled	
Height	
Hint	
Id	@+id/txtUrl
Ime action id	
Ime action label	
Ime options	
Include font padding	
Input type	textUri
Is scroll container	
Keep screen on	

Supply an identifier name for this view, to later retrieve it with `View.findViewById()` or `Activity.findViewById()`. [reference] Android SDK Content Loader

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" >

        <EditText
            android:id="@+id/txtUrl"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:text="http://www.google.com"
            android:layout_toLeftOf="@+id/btn1"
            android:inputType="textUri"/>

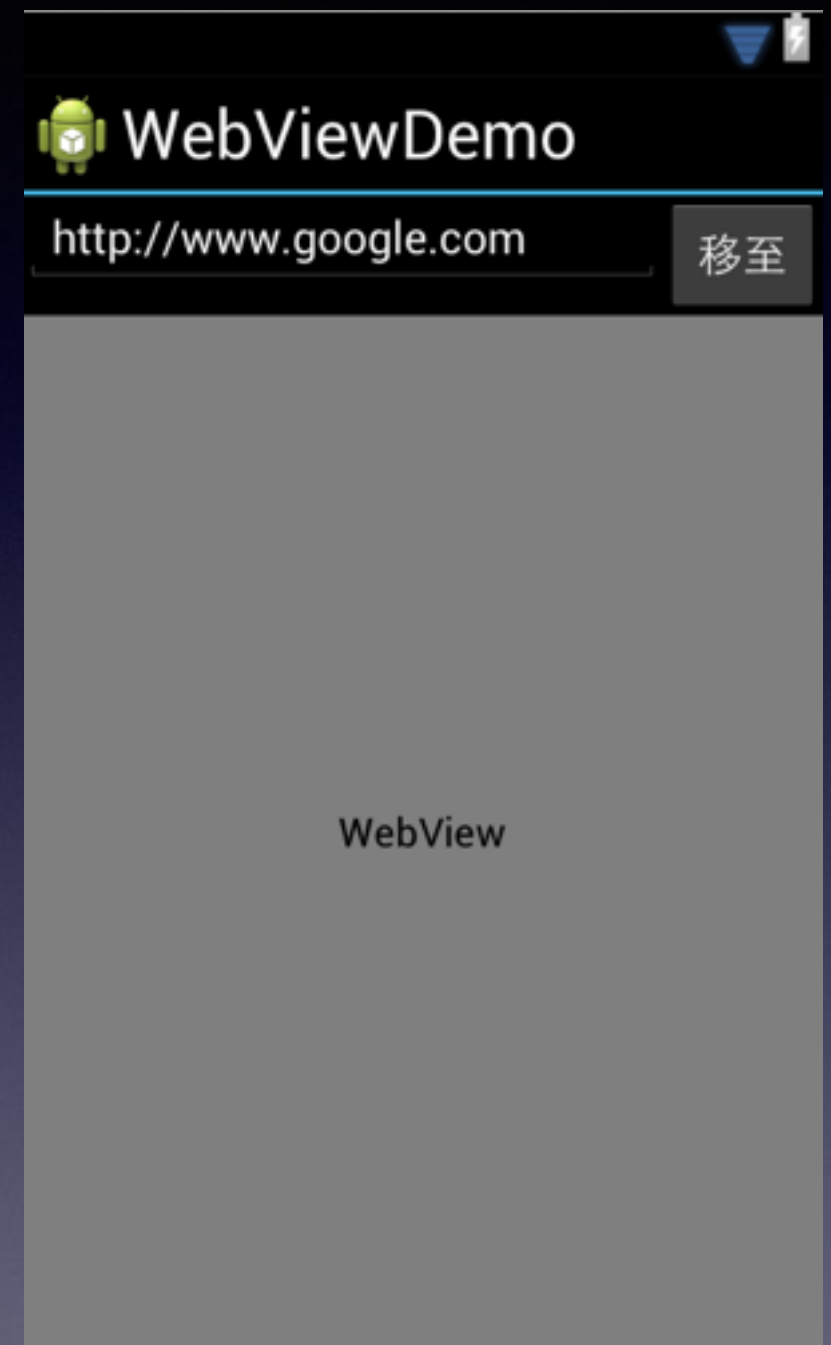
        <Button
            android:id="@+id/btn1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:text="移至"
            android:onClick="btn1_Click"/>

    </RelativeLayout>

    <WebView
        android:id="@+id/webView1"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />

</LinearLayout>

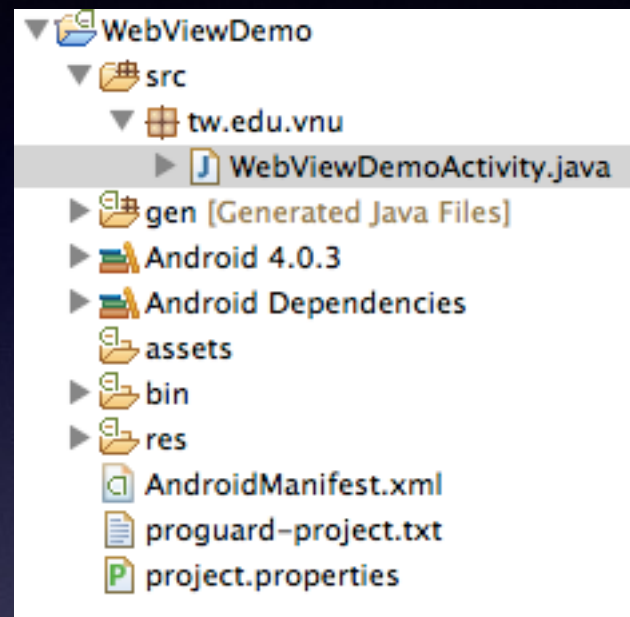
```



完成畫面

# 3. 建立 Activity 活動類別

- 宣告成員變數
- onCreate() 方法
- initWebBrowser() 方法
- loadUrl() 方法
- btnI\_Click() 事件處理方法
- onKeyDown() 方法



開啟 src > tw.edu.vnu 下的  
**WebViewDemoActivity.java**



Java - WebViewDemo/src/tw/edu/vnu/WebViewDemoActivity.java - Eclipse - /Volumes/ADATA 8G/Workspaces

Package Explorer

Ch2\_3  
Ch5\_2  
Ch5\_3  
Ch5\_4\_1  
Ch5\_4\_2  
Ch5\_5\_2  
Ch5\_5\_3  
Ch5\_5\_4  
Ch5\_5\_5  
Ch5\_5\_6  
WebViewDemo  
src  
tw.edu.vnu  
WebViewDemoActivity.java  
gen [Generated Java Files]  
Android 4.0.3  
Android Dependencies  
assets  
bin  
res  
AndroidManifest.xml  
proguard-project.txt  
project.properties

WebViewDemoActivity.java

```
1 package tw.edu.vnu;  
2  
3 import android.app.Activity;  
4  
5  
6 public class WebViewDemoActivity extends Activity {  
7     /** Called when the activity is first created. */  
8     @Override  
9     public void onCreate(Bundle savedInstanceState) {  
10         super.onCreate(savedInstanceState);  
11         setContentView(R.layout.main);  
12     }  
13 }
```

Outline

tw.edu.vnu  
import declarations  
WebViewDemoActivity  
onCreate(Bundle) : void

Properties

Property	Value
Info	
derived	false
editable	true
last modified	2012年4月8日下午8:42:12
linked	false
location	/Volumes/ADATA 8G/Workspaces/WebViewDemo/src/tw/edu/vnu/WebViewDe...
name	WebViewDemoActivity.java
nath	/Web/WebViewDemo/src/tw/edu/vnu/WebViewDemoActivityv.java

Writable Smart Insert 1 : 1 Android SDK Content Loader



# import 載入類別

```
WebViewDemoActivity.java
1 package tw.edu.vnu;
2
3 import android.app.Activity;
4
5
6 public class WebViewDemoActivity extends Activity {
7
8     private WebView web;
9
10    /** Called when the activity is first created. */
11    @Override
12    public void onCreate(Bundle savedInstanceState) {
13        super.onCreate(savedInstanceState);
14        setContentView(R.layout.main);
15    }
16
17 }
```



```
WebViewDemoActivity.java
1 package tw.edu.vnu;
2
3 import android.app.Activity;
4
5
6 public class WebViewDemoActivity extends Activity {
7
8     private WebView web;
9
10    /** Called when the activity is first created. */
11    @Override
12    public void onCreate(Bundle savedInstanceState) {
13        super.onCreate(savedInstanceState);
14        setContentView(R.layout.main);
15    }
16
17 }
```

WebView cannot be resolved to a type

7 quick fixes available:

- Import 'WebView' (android.webkit)
- Create class 'WebView'
- Create interface 'WebView'
- Change to 'WebViewClient' (android.webkit)
- Create enum 'WebView'
- Add type parameter 'WebView' to 'WebViewDemoActivity'

Press 'F2' for focus

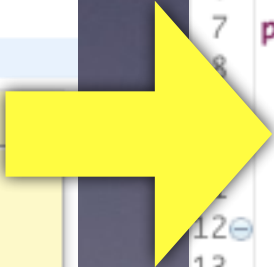


```
WebViewDemoActivity.java
1 package tw.edu.vnu;
2
3 import android.app.Activity;
4
5
6 public class WebViewDemoActivity extends Activity {
7
8     private WebView web;
9
10    /** Called when the activity is first created. */
11    @Override
12    public void onCreate(Bundle savedInstanceState) {
13        super.onCreate(savedInstanceState);
14        setContentView(R.layout.main);
15    }
16
17 }
```

WebView cannot be resolved to a type

7 quick fixes available:

- Import 'WebView' (android.webkit)
- Create class 'WebView'
- Create interface 'WebView'
- Change to 'WebViewClient' (android.webkit)
- Create enum 'WebView'



```
1 package tw.edu.vnu;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.webkit.WebView;
6
7 public class WebViewDemoActivity extends Activity {
8
9     private WebView web;
10
11    /** Called when the activity is first created. */
12    @Override
13    public void onCreate(Bundle savedInstanceState) {
14        super.onCreate(savedInstanceState);
15        setContentView(R.layout.main);
16    }
17
18 }
```

# 宣告成員變數

```
package tw.edu.vnu;

import android.app.Activity;
import android.app.AlertDialog;
import android.os.Bundle;
import android.os.Handler;
import android.webkit.WebView;
import android.widget.EditText;

public class WebViewDemoActivity extends Activity {

    private WebView web;
    private AlertDialog progressDialog;
    private Handler handler;
    private EditText txtUrl;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

# casting 轉型

```
17 /** Called when the activity is first created. */
18 @Override
19 public void onCreate(Bundle savedInstanceState) {
20     super.onCreate(savedInstanceState);
21     setContentView(R.layout.main);
22
23     web = findViewById(R.id.webView1);
24 }
25
26 }
```

```
17 /** Called when the activity is first created. */
18 @Override
19 public void onCreate(Bundle savedInstanceState) {
20     super.onCreate(savedInstanceState);
21     setContentView(R.layout.main);
22
23     web = findViewById(R.id.webView1);
24 }
25
26 }
```

Type mismatch: cannot convert from View to WebView

2 quick fixes available:

- [Add cast to 'WebView'](#)
- [Change type of 'web' to 'View'](#)

Press 'F2' for focus

```
17 /** Called when the activity is first created. */
18 @Override
19 public void onCreate(Bundle savedInstanceState) {
20     super.onCreate(savedInstanceState);
21     setContentView(R.layout.main);
22
23     web = (WebView) findViewById(R.id.webView1);
24 }
25
26 }
```

# onCreate() 方法

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

    web = (WebView) findViewById(R.id.webView1);
    initWebBrowser(); // 初始化WebView
    // 取得EditText元件
    txtUrl = (EditText) findViewById(R.id.txtUrl);
    String strUrl = txtUrl.getText().toString();
    // 預設網址為http://www.google.com
    loadUrl(web, strUrl);
}
```



# initWebBrowser() 方法

```
public void initWebBrowser() {  
    web.getSettings().setJavaScriptEnabled(true); // 啟用JavaScript  
    web.setScrollBarStyle(0); // 顯示捲動軸  
    // 覆寫方法讓使用者點選網頁連結時，直接載入新網頁  
    web.setWebViewClient(new WebViewClient() {  
        @Override  
        public boolean shouldOverrideUrlLoading(WebView view, String url) {  
            loadUrl(view, url);  
            return true;  
        }  
    });  
    // 覆寫方法以便在完全載入網頁後，送出Handler物件的訊息關閉進度對話方塊  
    web.setWebChromeClient(new WebChromeClient() {  
        @Override  
        public void onProgressChanged(WebView view, int progress) {  
            if (progress == 100) { // 全部載入  
                handler.sendEmptyMessage(1); // 隱藏進度對話方塊  
            }  
            super.onProgressChanged(view, progress);  
        }  
    });  
  
    progressDialog = new ProgressDialog(this); // 建立進度對話方塊  
    progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);  
    progressDialog.setMessage("載入網頁中...");  
}
```



# loadUrl() 方法

```
// 在WebView載入網頁
public void loadUrl(final WebView view, final String url) {
    new Thread() {
        @Override
        public void run() {
            handler.sendMessage(0); // 顯示進度對話方塊
            view.loadUrl(url); // 載入網頁
        }
    }.start();
    // 覆寫方法來處理訊息，使用switch條件敘述判斷參數，以便顯示或隱藏進度對話方塊
    handler = new Handler() {
        @Override
        public void handleMessage(Message msg) {
            if (!Thread.currentThread().isInterrupted()) {
                switch (msg.what) {
                    case 0:
                        progressDialog.show(); // 顯示進度對話方塊
                        break;
                    case 1:
                        progressDialog.hide(); // 隱藏進度對話方塊
                        break;
                }
            }
            super.handleMessage(msg);
        }
    };
}
```

# btn1\_Click() 事件處理方法

```
// Button元件的事件處理  
public void btn1_Click(View view) {  
    String strUrl = txtUrl.getText().toString();  
    loadUrl(web, strUrl);  
}
```

# onKeyDown() 方法

```
// 處理使用者按下返回鍵回到上一頁，如果沒有上一頁就顯示對話方塊，確認是否離開行動瀏覽器
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK && web.canGoBack()) {
        web.goBack(); // 上一頁網頁
        return true;
    } else if (keyCode == KeyEvent.KEYCODE_BACK) {
        // 顯示確認離開的對話方塊
        AlertDialog.Builder aDlg = new AlertDialog.Builder(this);
        aDlg.setTitle("離開行動瀏覽器")
            .setMessage("請確認離開行動瀏覽器？").setPositiveButton("是",
            new OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    WebViewDemoActivity.this.finish();
                }
            }).setNegativeButton("否", null).show();
        return true;
    }
    return super.onKeyDown(keyCode, event);
}
```

# 完整程式碼

```
package tw.edu.vnu;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.view.KeyEvent;
import android.view.View;
import android.webkit.WebChromeClient;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.EditText;

public class WebViewDemoActivity extends Activity {

    private WebView web;
    private ProgressDialog progressDialog;
    private Handler handler;
    private EditText txtUrl;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

```

web = (WebView) findViewById(R.id.webView1);
initWebBrowser(); // 初始化WebView
// 取得EditText元件
txtUrl = (EditText) findViewById(R.id.txtUrl);
String strUrl = txtUrl.getText().toString();
// 預設網址為http://www.google.com
loadUrl(web, strUrl);
}

public void initWebBrowser() {
web.getSettings().setJavaScriptEnabled(true); // 啟用JavaScript
web.setScrollBarStyle(0); // 顯示捲動軸
// 覆寫方法讓使用者點選網頁連結時，直接載入新網頁
web.setWebViewClient(new WebViewClient() {
    @Override
    public boolean shouldOverrideUrlLoading(WebView view, String url) {
        loadUrl(view, url);
        return true;
    }
});
// 覆寫方法以便在完全載入網頁後，送出Handler物件的訊息關閉進度對話方塊
web.setWebChromeClient(new WebChromeClient() {
    @Override
    public void onProgressChanged(WebView view, int progress) {
        if (progress == 100) { // 全部載入
            handler.sendMessage(1); // 隱藏進度對話方塊
        }
        super.onProgressChanged(view, progress);
    }
});

progressDialog = new ProgressDialog(this); // 建立進度對話方塊
progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
progressDialog.setMessage("載入網頁中...");
}

```



```

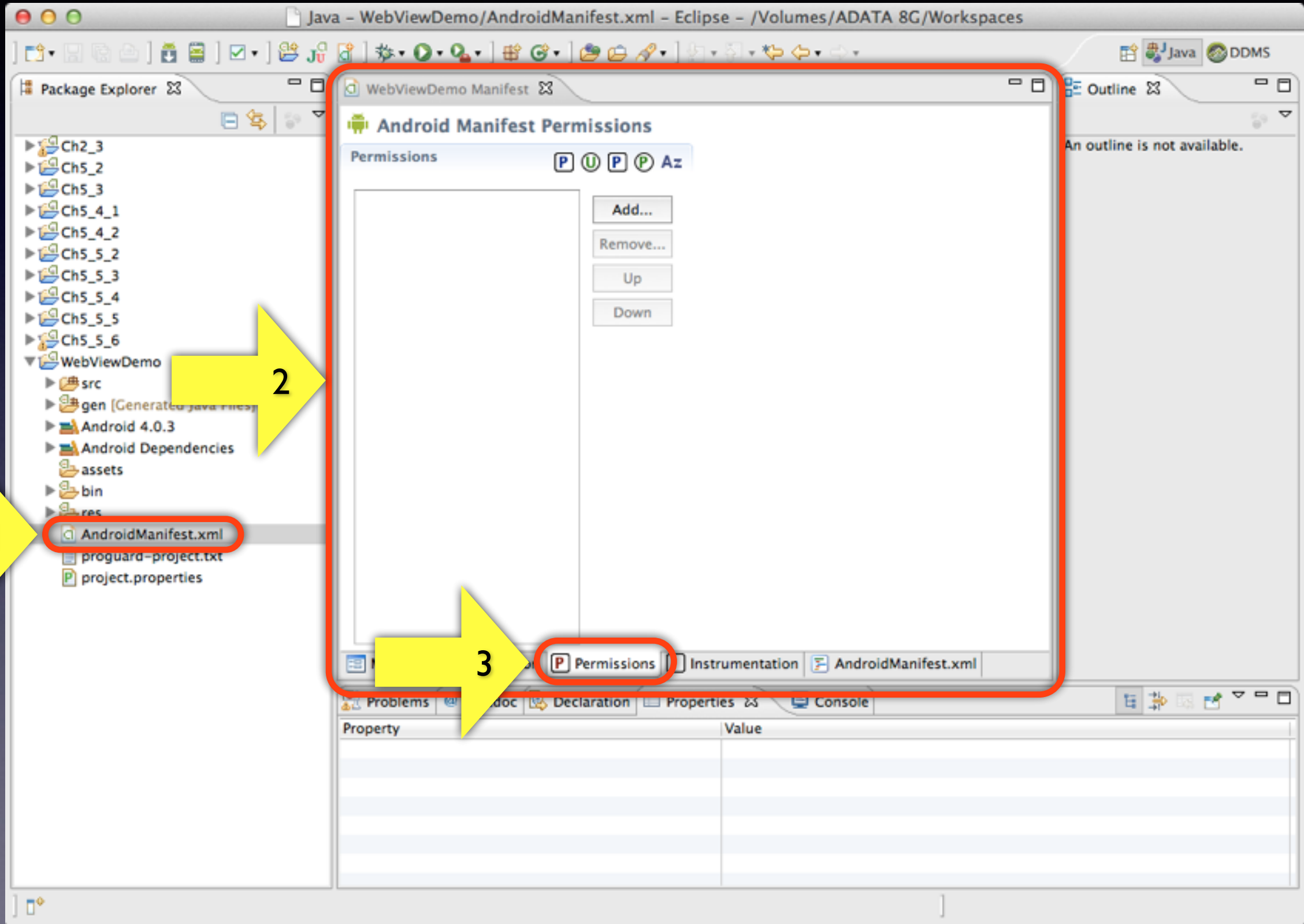
// 在WebView載入網頁
public void loadUrl(final WebView view, final String url) {
    new Thread() {
        @Override
        public void run() {
            handler.sendMessage(0); // 顯示進度對話方塊
            view.loadUrl(url); // 載入網頁
        }
    }.start();
    // 覆寫方法來處理訊息，使用switch條件敘述判斷參數，以便顯示或隱藏進度對話方塊
    handler = new Handler() {
        @Override
        public void handleMessage(Message msg) {
            if (!Thread.currentThread().isInterrupted()) {
                switch (msg.what) {
                    case 0:
                        progressDialog.show(); // 顯示進度對話方塊
                        break;
                    case 1:
                        progressDialog.hide(); // 隱藏進度對話方塊
                        break;
                }
            }
            super.handleMessage(msg);
        }
    };
}

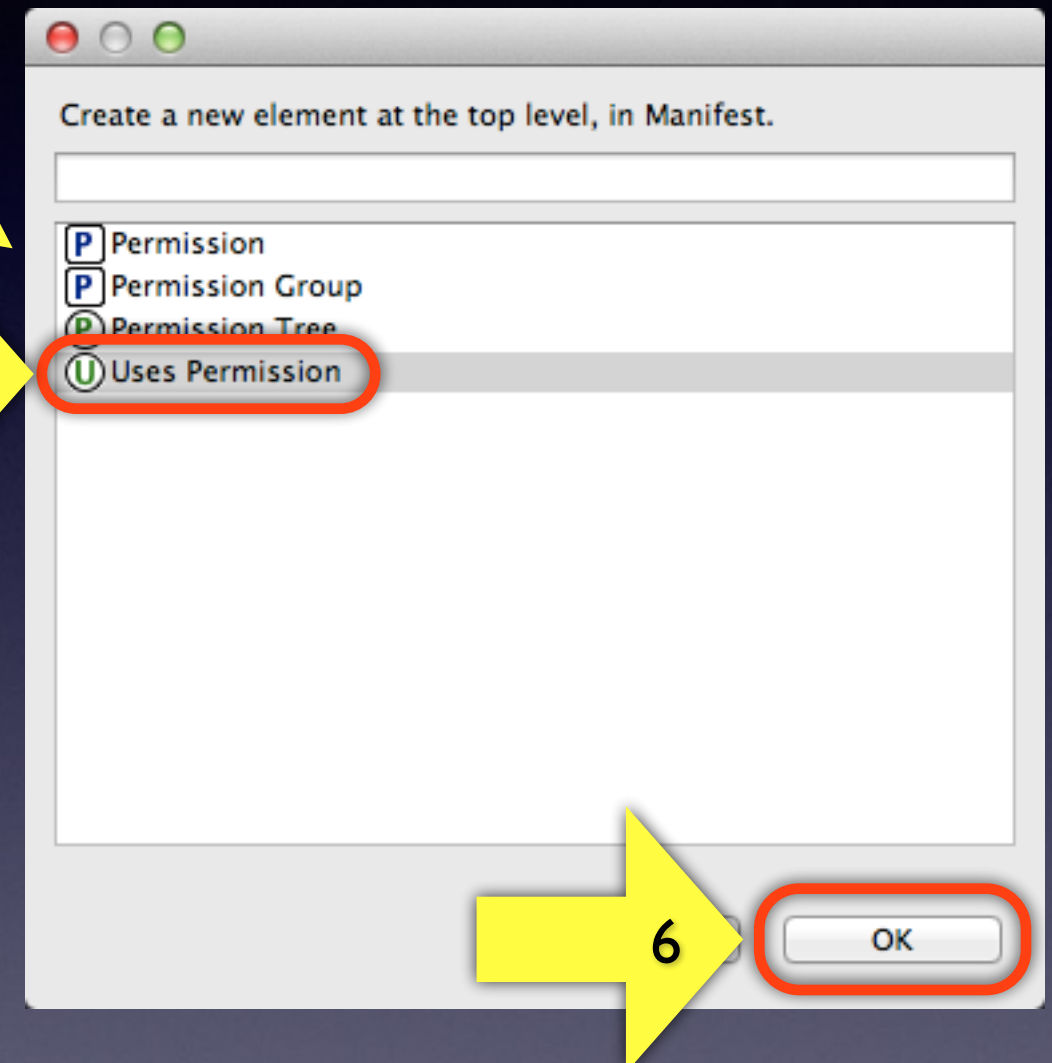
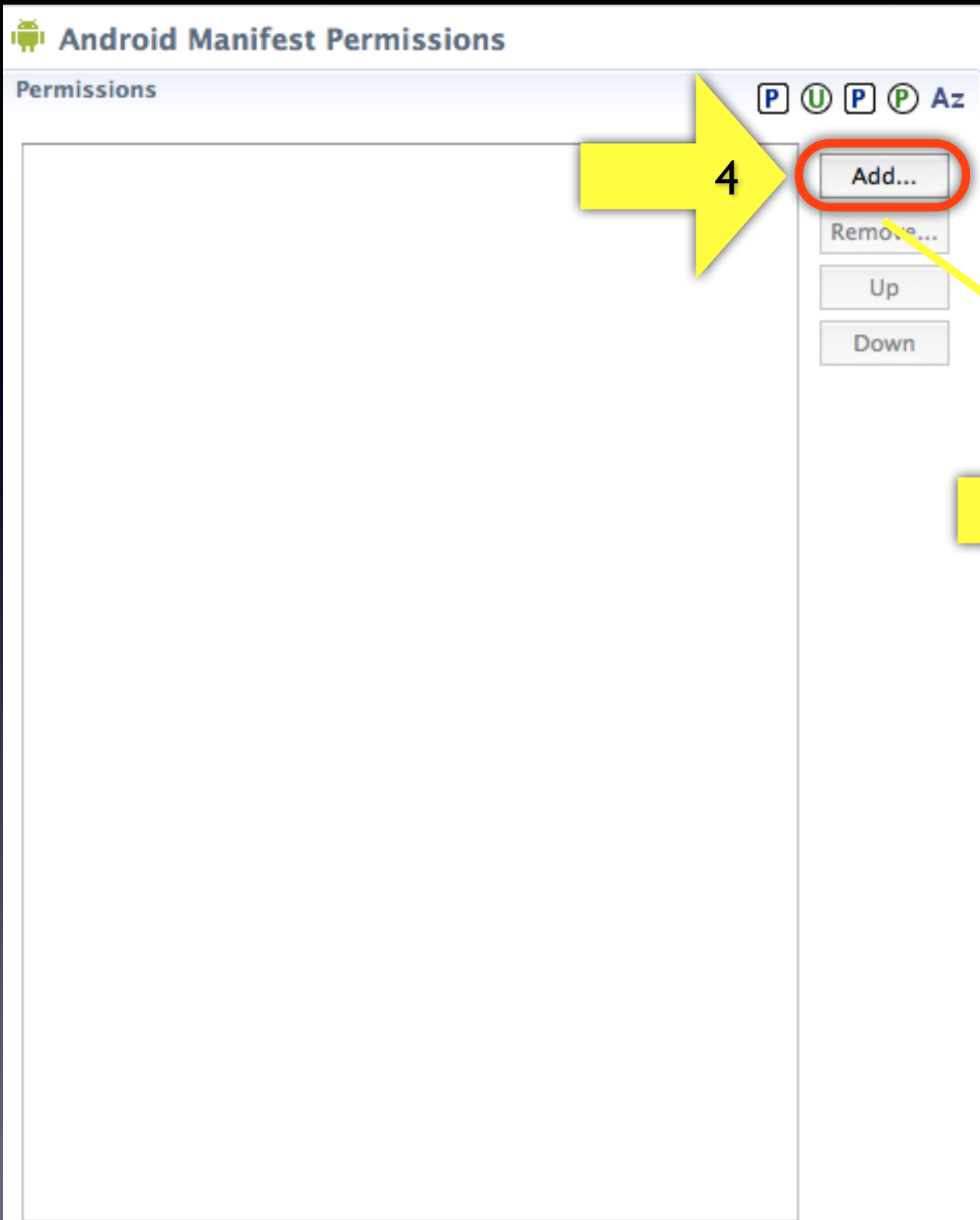
// Button元件的事件處理
public void btn1_Click(View view) {
    String strUrl = txtUrl.getText().toString();
    loadUrl(web, strUrl);
}

```

```
// 處理使用者按下返回鍵回到上一頁，如果沒有上一頁就顯示對話方塊，確認是否離開行動瀏覽器
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK && web.canGoBack()) {
        web.goBack(); // 上一頁網頁
        return true;
    } else if (keyCode == KeyEvent.KEYCODE_BACK) {
        // 顯示確認離開的對話方塊
        AlertDialog.Builder aDlg = new AlertDialog.Builder(this);
        aDlg.setTitle("離開行動瀏覽器")
            .setMessage("請確認離開行動瀏覽器？").setPositiveButton("是", new OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    WebViewDemoActivity.this.finish();
                }
            }).setNegativeButton("否", null).show();
        return true;
    }
    return super.onKeyDown(keyCode, event);
}
}
```

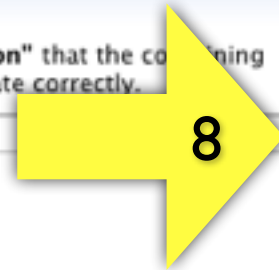
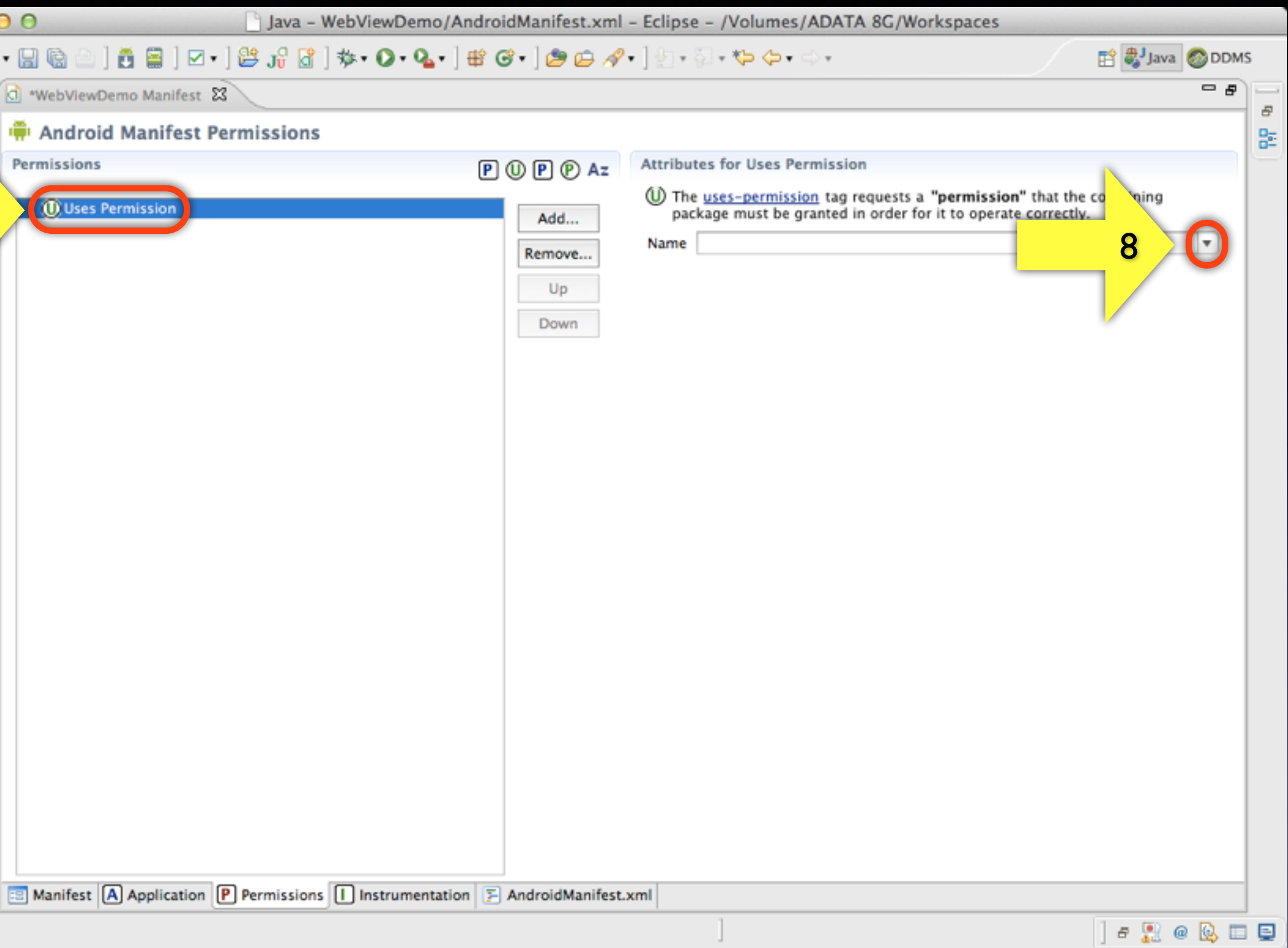
## 4.在 AndroidManifest.xml 新增存取 Internet 權限





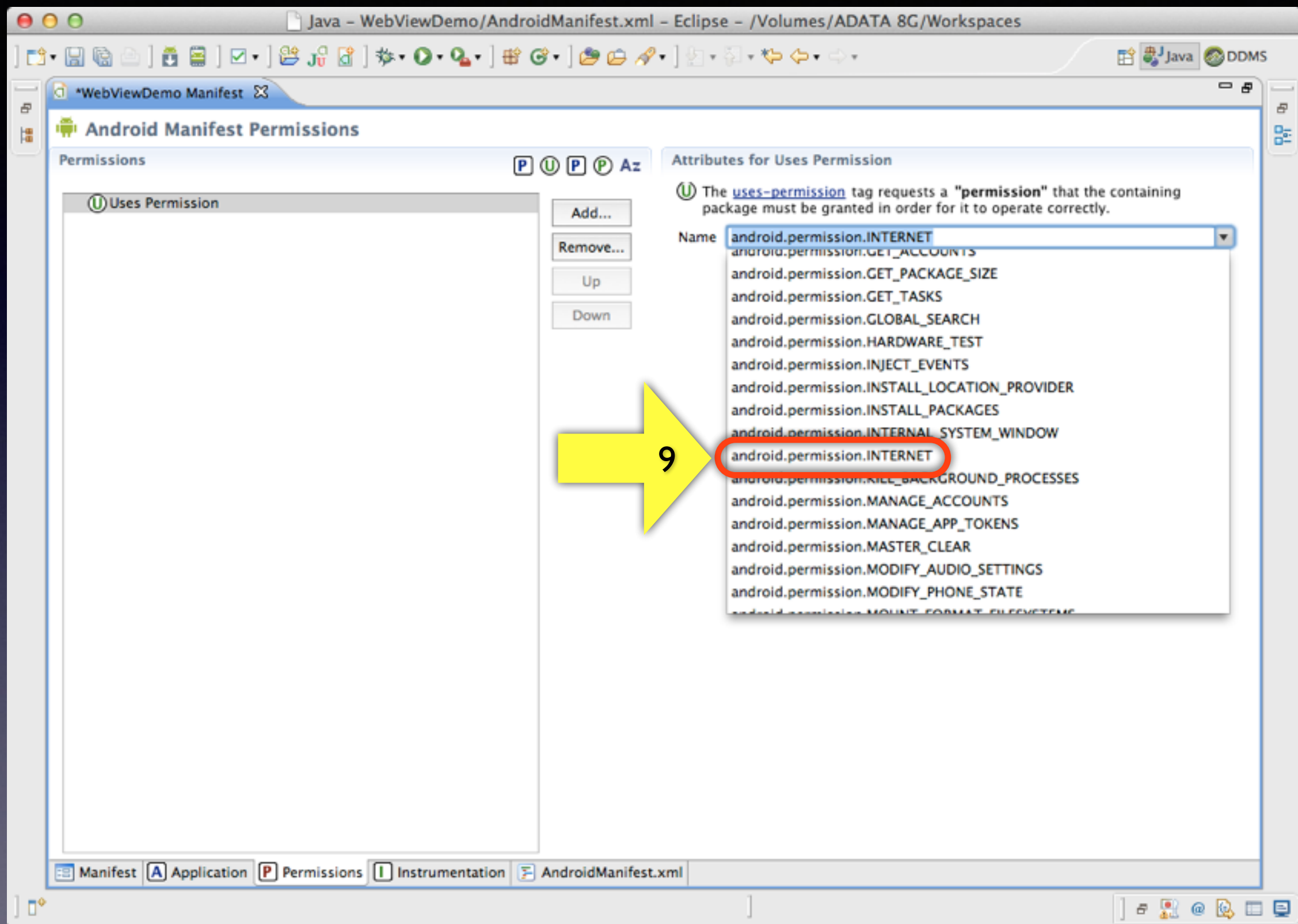


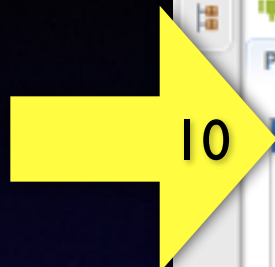
7



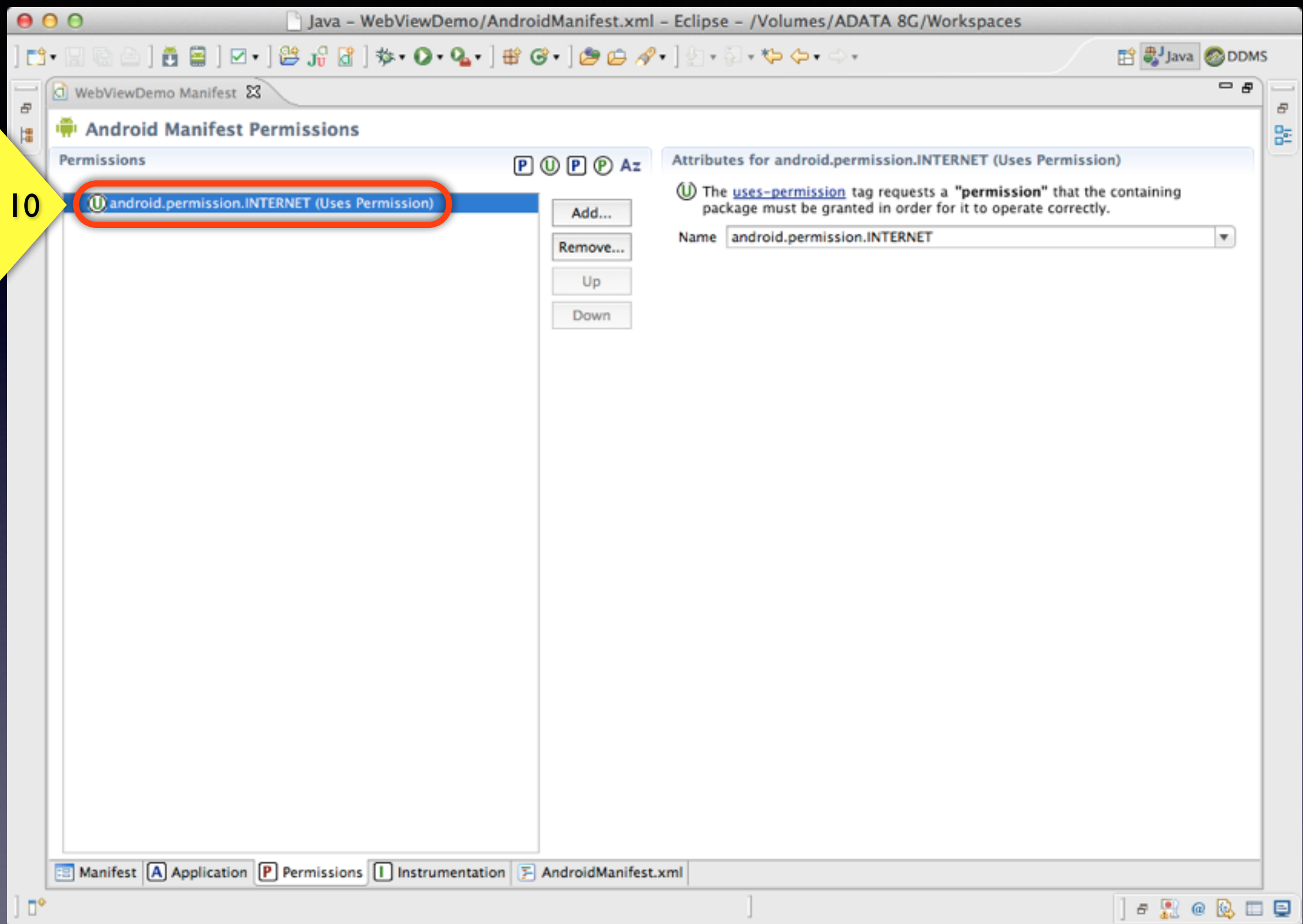
8



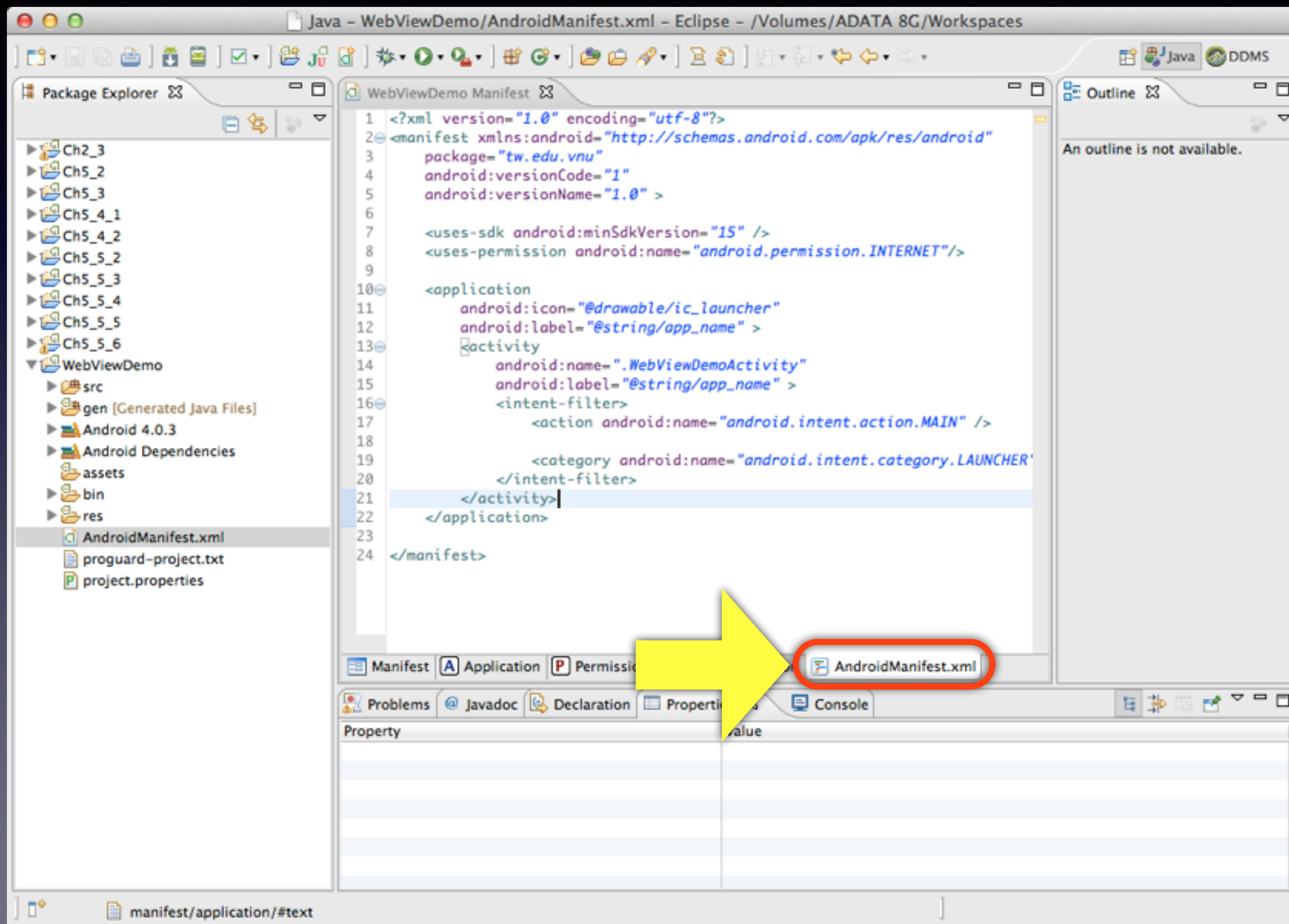


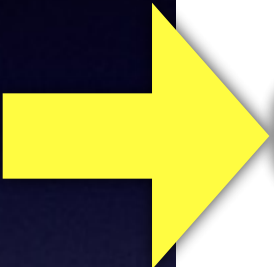


10



# 直接看 xml 內容





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

    <uses-sdk android:minSdkVersion="15" />

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".WebViewDemoActivity"
            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>
```

# 5.執行 Android 模擬器佈署程式

