

Android APP手機程式設計實務



講師：賴貴平



本節課程內容

- 本節課程內容將包含以下教學內容:

Android 系統事件處理與委派機制

- Button 元件
- TextView 元件
- ImageView 元件
- EditText 元件
- Checkbox 元件
- RadioButton 元件



ANDROID 系統事件處理與委派機制

- 系統事件與委派機制處理事件：
 - 介面元件只是在**Android**行動裝置的螢幕上繪出的使用介面圖形，使用者可以觀看；無法產生互動
 - 若要 **Android** 應用程式與使用者可以互動，獲得回應，就必須撰寫**Java**程式碼來建立事件處理
 - **Android**應用程式設計是一種事件驅動模型，而程式設計能產生與使用者互動的基礎就是系統事件處理與委派機制模型
 - **Android**使用的事件處理機制就是委派機制事件處理模型。

```
16  
17  
18  
19  
20
```

```
button1 = (Button) findViewById(R.id.btn_1);  
button2 = (Button) findViewById(R.id.btn_2);  
  
button1.setOnClickListener(new android.view.View.OnClickListener() {  
    ..  
    ..  
})
```



ANDROID 系統事件處理與委派機制

- 系統事件與委派機制處理事件：
 - 「事件」(event) 是指當在執行Android應用程式時，發生系統狀態改變的現象：
 - 觸碰螢幕
 - 按下鍵盤
 - 操作觸發一些動作
 - 例如:控制機器人行走，按下遙控器開關可以產生事件，我們需要按下前進按鈕觸發事件，才能執行事件處理讓機器人開始向前走
 - `setOnClickListener`:
 - Register a callback to be invoked when this view is clicked.



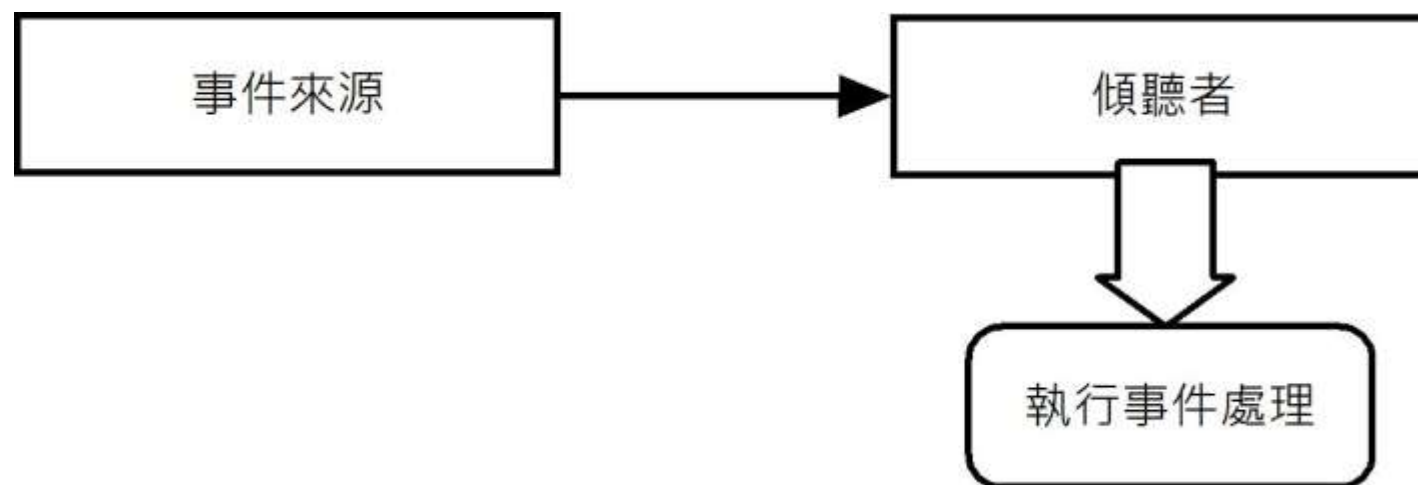
ANDROID 系統事件處理與委派機制

- 系統事件與委派機制處理事件：
 - Android系統事件處理就是利用Java程式碼處理事件，這是一種「委派事件處理模型」（**delegation event model**），分為「事件來源」（**event source**）和處理事件的「傾聽者」（**listener**）物件
 - 在互動式的應用程式裡，往往無法主動地去執行程式，因為程式設計師不確定使用者會不會按下，因此只能被動地等待系統觸發按鈕被按下去的事件
 - 回呼(**callback**)方法：
 - 讓系統去呼叫這個方法，這種類型的方法就稱為回呼方法
 - 按鈕該有的功能會寫在一個回呼方法內



ANDROID 系統事件處理與委派機制

- 系統事件與委派機制處理事件：
 - Listener :
 - 一些介面元件是需要等待使用者的互動，這些介面元件必須要有個傾聽者(Listener)去傾聽互動的方法，而這些傾聽者物件都會有回呼方法(callback)，處理互動事件(例如按下按鈕的事件)





ANDROID 系統事件處理與委派機制

- 系統事件與委派機制處理事件：
 - 處理流程





ANDROID 系統事件處理與委派機制

- 系統事件與委派機制處理事件：
 - Java委託事件處理模型是利用介面（ interface ）來實作，Java介面是一種定義類別行為的Java語法
 - Java介面是在類別繼承架構中定義類別的行為，介面定義需要被實作的函式原型，類別實作介面
 - 類別可以實作一個或多個介面
 - 例如:把介面視為是一張證照，經過類別實作，如同考上證照，擁有專屬能力
 - `setOnClickListener`需要一個參數，這個參數即為一個傾聽者物件，程式透過實作`OnClickListener`介面來設計按鈕的傾聽者



ANDROID 系統事件處理與委派機制

- 實作 **OnClickListener** 介面

- `button1.setOnClickListener(new View.OnClickListener()`
- `@Override`
- `public void onClick(View v) {`
- `button1.setText("已按下按鈕一");`
- `button1.setBackgroundColor(0);`
- `button1.setClickable(false);`
- `button2.setText("我是按鈕二");`
- `button2.setBackgroundResource(R.drawable.button_blue);`
- `button2.setClickable(true);`
- `}`
- `});`



ANDROID 系統事件處理與委派機制

- 系統事件與委派機制處理事件：
 - 步驟一：
 - 在MainActivity活動實作OnClickListener介面
 - MainActivity活動類別不只繼承AppCompatActivity類別，還需實作OnClickListener事件處理介面
 - OnClickListener事件處理介面需實作onClick()介面方法，所以，AppCompatActivity物件本身是一個可以處理按一下事件的傾聽者物件



ANDROID 系統事件處理與委派機制

- 系統事件與委派機制處理事件：
 - 步驟二：
 - 註冊按鈕元件使用的傾聽者物件
 - **MainActivity**活動類別實作**OnClickListener**事件處理介面後，擁有處理按一下事件的功能，可以處理**Button**元件使用者按下的按一下事件
 - 需要在**onCreate()**方法註冊按鈕元件使用的傾聽者物件
 - 事件觸發後即執行**MainActivity**類別的**onClick()**方法，並執行委派知回呼函式



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Button
- 開啟 activity_button_java.xml，並於其中加入以下程式碼
- `<?xml version="1.0" encoding="utf-8"?>`
- `<android.support.constraint.ConstraintLayout`
`xmlns:android="http://schemas.android.com/apk/res/android"`
- `xmlns:app="http://schemas.android.com/apk/res-auto"`
- `android:layout_width="match_parent"`
- `android:layout_height="match_parent"`
- `>`
- `<Button`
- `android:id="@+id/btn_1"`
- `android:layout_width="wrap_content"`
- `android:layout_height="wrap_content"`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Button
- 開啟 activity_button_java.xml，續
 - `android:background="@drawable/button_blue"`
 - `android:text="我是按鈕 I"`
 - `android:textColor="#000000"`
 - `app:layout_constraintBottom_toBottomOf="parent"`
 - `app:layout_constraintLeft_toLeftOf="parent"`
 - `app:layout_constraintRight_toRightOf="parent"`
 - `app:layout_constraintTop_toTopOf="parent"`
 - `app:layout_constraintVertical_bias="0.4" />`
- `<Button`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Button
- 開啟 activity_button_java.xml，續
 - `android:id="@+id/btn_2"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:background="@drawable/button_blue"`
 - `android:text="我是按鈕2"`
 - `android:textColor="#000000"`
 - `app:layout_constraintBottom_toBottomOf="parent"`
 - `app:layout_constraintLeft_toLeftOf="parent"`
 - `app:layout_constraintRight_toRightOf="parent"`
 - `app:layout_constraintTop_toTopOf="parent"`
 - `app:layout_constraintVertical_bias="0.6" />`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Button
- 開啟 Activity_Button_Java.java，並於其中加入以下程式碼
 - import android.view.View;
 - import android.widget.Button;
 - public class Activity_Button_Java extends AppCompatActivity {
 - Button button1, button2;
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_button_java);
 - button1 = (Button) findViewById(R.id.btn_1);



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Button
- 開啟 Activity_Button_Java.java , 續
 - `button2 = (Button) findViewById(R.id.btn_2);`
 - `button1.setOnClickListener(new android.view.View.OnClickListener() {`
 - `@Override`
 - `public void onClick(View v) {`
 - `button1.setText("已按下按鈕一");`
 - `button1.setBackgroundColor(0);`
 - `button1.setClickable(false);`
 - `button2.setText("我是按鈕二");`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Button
- 開啟 Activity_Button_Java.java ， 續
 - `button2.setBackgroundResource(R.drawable.button_blue);`
 - `button2.setClickable(true);`
 - `}`
 - `});`
 - `button2.setOnClickListener(new android.view.View.OnClickListener()`
 - `{`
 - `@Override`
 - `public void onClick(View v) {`
 - `button2.setText("已按下按鈕二");`
 - `button2.setBackgroundColor(0);`
 - `button2.setClickable(false);`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Button
- 開啟 Activity_Button_Java.java ， 續
 - button1.setText("我是按鈕一");
 - button1.setBackgroundResource(R.drawable.button_blue);
 - button1.setClickable(true);
 - }
 - });
 - }



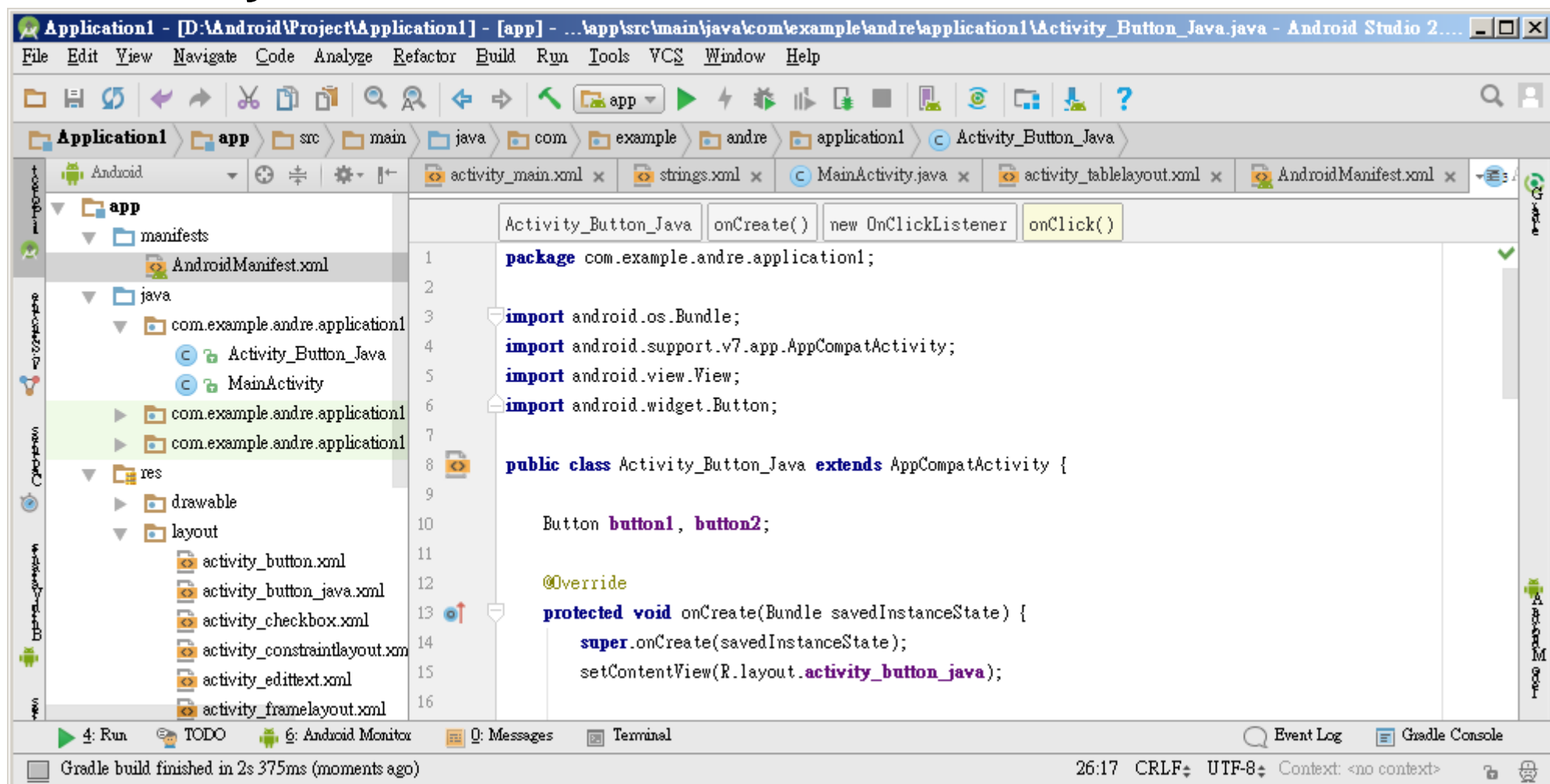
ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Button - 常用函式
 - `setBackgroundColor()`
 - 設定背景顏色
 - `setClickable()`
 - 設定是否可按下
 - `setBackgroundResource`
 - 設定背景來源



ANDROID 元件與程式控制

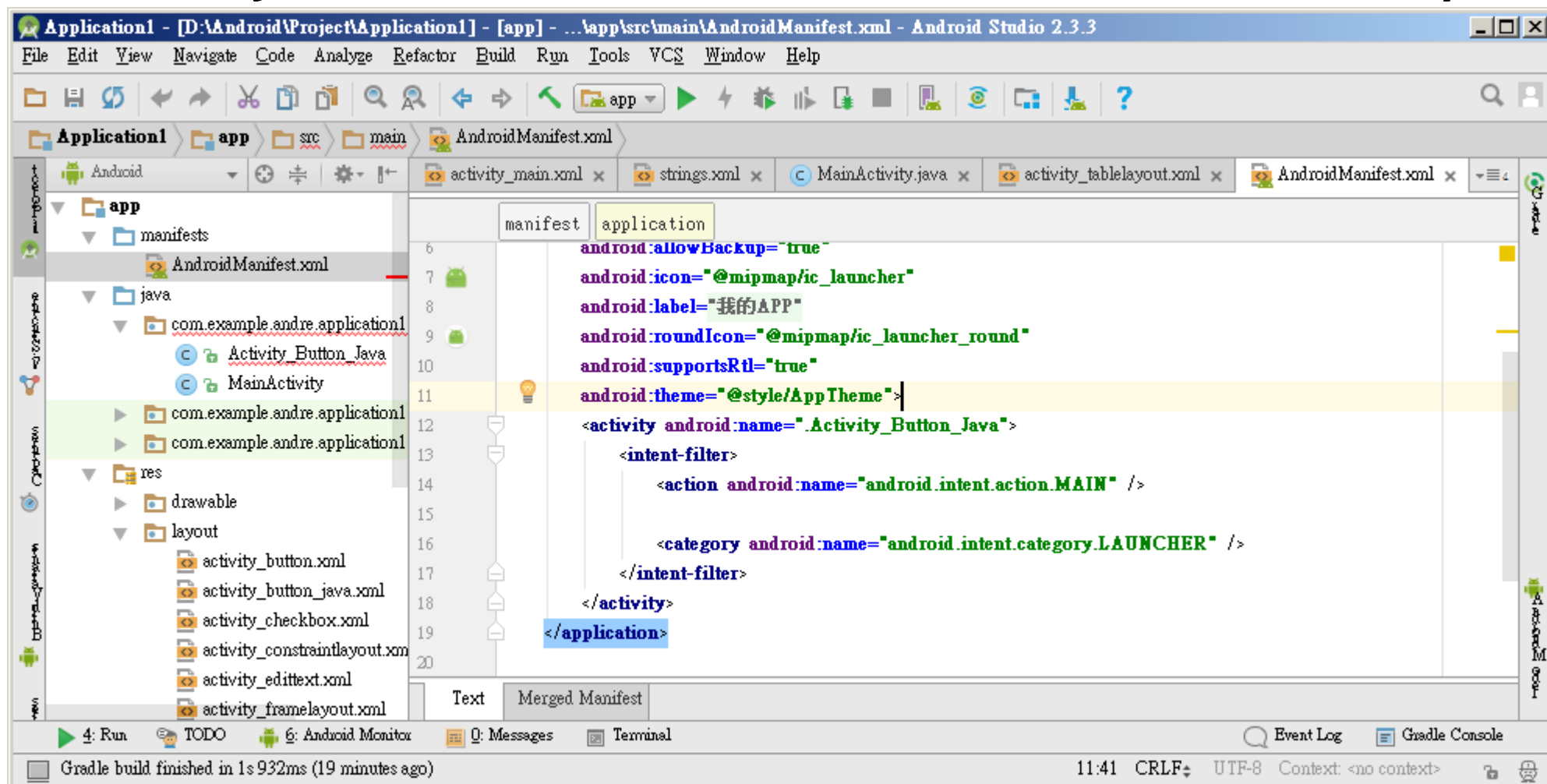
- 利用 Java 程式碼控制 Button – 程式編輯器





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Button - 切換啟動 Activity





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Button - 切換啟動 Activity
- 開啟 AndroidManifest.xml，並於進行修改
 - `<application`
 - `android:allowBackup="true"`
 - `android:icon="@mipmap/ic_launcher"`
 - `android:label="@string/app_name"`
 - `android:roundIcon="@mipmap/ic_launcher_round"`
 - `android:supportRtl="true"`
 - `android:theme="@style/AppTheme">`
 - `<activity android:name=".Activity_Button_Java">`
 - `<intent-filter>`
 - `<action android:name="android.intent.action.MAIN" />`
 - `<category android:name="android.intent.category.LAUNCHER" />`
 - `</intent-filter>`
 - `</activity>`
 - `</application>`



ANDROID 元件與程式控制

- 成果 - 程式控制 Button





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 TextView
- 開啟 `activity_textview_java.xml`，並於其中加入以下程式碼
 - `<LinearLayout`
`xmlns:android="http://schemas.android.com/apk/res/android"`
 - `android:layout_width="match_parent"`
 - `android:layout_height="match_parent"`
 - `android:gravity="center_horizontal"`
 - `android:orientation="vertical"`
 - `android:focusable="true"`
 - `android:focusableInTouchMode="true" >`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 TextView
- 開啟 activity_textview_java.xml，續
 - <TextView
 - android:id="@+id/text_1"
 - android:layout_width="match_parent"
 - android:layout_height="wrap_content"
 - android:text="顯示文字:TextView I"
 - android:textColor="#000000"
 - android:textSize="24sp"
 - android:focusable="true"
 - android:focusableInTouchMode="true"
 - android:clickable="true"/>



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 TextView
- 開啟 activity_textview_java.xml，續
 - <TextView
 - android:id="@+id/text_2"
 - android:layout_width="match_parent"
 - android:layout_height="wrap_content"
 - android:text="顯示文字:TextView2"
 - android:textColor="#000000"
 - android:textSize="24sp"
 - android:focusable="true"
 - android:focusableInTouchMode="true"
 - android:clickable="true"/>



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 TextView
- 開啟 activity_textview_java.xml，續
 - <TextView
 - android:id="@+id/text_3"
 - android:layout_width="match_parent"
 - android:layout_height="wrap_content"
 - android:text="顯示文字:TextView3"
 - android:textColor="#000000"
 - android:textSize="24sp"
 - android:focusable="true"
 - android:focusableInTouchMode="true"
 - android:clickable="true"/>



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 TextView
- 開啟 `Activity_TextView_Java.java`，並於其中加入以下程式碼
 - `import android.view.View;`
 - `import android.widget.TextView;`
 - `public class Activity_TextView_Java extends AppCompatActivity {`
 - `TextView textView1, textView2, textView3;`
 - `String text1;`
 - `Integer colorText1;`
 - `float sizeText3;`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 TextView
- 開啟 Activity_TextView_Java.java ， 續
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_textview_java);
 - textView1 = (TextView) findViewById(R.id.text_1);
 - textView2 = (TextView) findViewById(R.id.text_2);
 - textView3 = (TextView) findViewById(R.id.text_3);



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 TextView
- 開啟 Activity_TextView_Java.java , 續
 - `text1 = textView1.getText().toString();`
 - `colorText1 = textView2.getCurrentTextColor();`
 - `sizeText3 = 24.0;`
 - `textView1.setOnFocusChangeListener(new View.OnFocusChangeListener() {`
 - `@Override`
 - `public void onFocusChange(View v, boolean hasFocus) {`
 - `if(hasFocus) {`
 - `textView1.setText("目前焦點:textView1");`
 - `}`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 TextView
- 開啟 Activity_TextView_Java.java , 續

```
○ else
○ {
○     textView1.setText(text1);
○ }
○ }
○ });

○ textView2.setOnFocusChangeListener(new View.OnFocusChangeListener() {
○     @Override
○     public void onFocusChange(View v, boolean hasFocus) {
○         if(hasFocus) {
○             textView2.setTextColor(0xff0000ff);
○         }
○     }
○ });
```



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 TextView
- 開啟 Activity_TextView_Java.java ， 續
 - else
 - {
 - textView2.setTextColor(colorText1);
 - }
 - }
 - });
 - textView3.setOnFocusChangeListener(new
View.OnFocusChangeListener() {
 - @Override



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 TextView
- 開啟 Activity_TextView_Java.java , 續
 - ```
public void onFocusChange(View v, boolean hasFocus) {
```
  - ```
    if(hasFocus) {
```
 - ```
 textView3.setTextSize(30);
```
  - ```
        //textView3.setTextSize(TypedValue.COMPLEX_UNIT_SP,30);
```
 - ```
 }
```
  - ```
    else
```
 - ```
 {
```
  - ```
        textView3.setTextSize(24);
```
 - ```
 }
```
  - ```
    }
```
 - ```
});
```
  - ```
}
```



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 TextView - 常用函式
 - setText ()
 - 設定文字內容
 - setTextSize
 - 設定文字大小
 - setBackgroundColor()
 - 設定背景顏色
 - setBackgroundResource
 - 設定背景來源



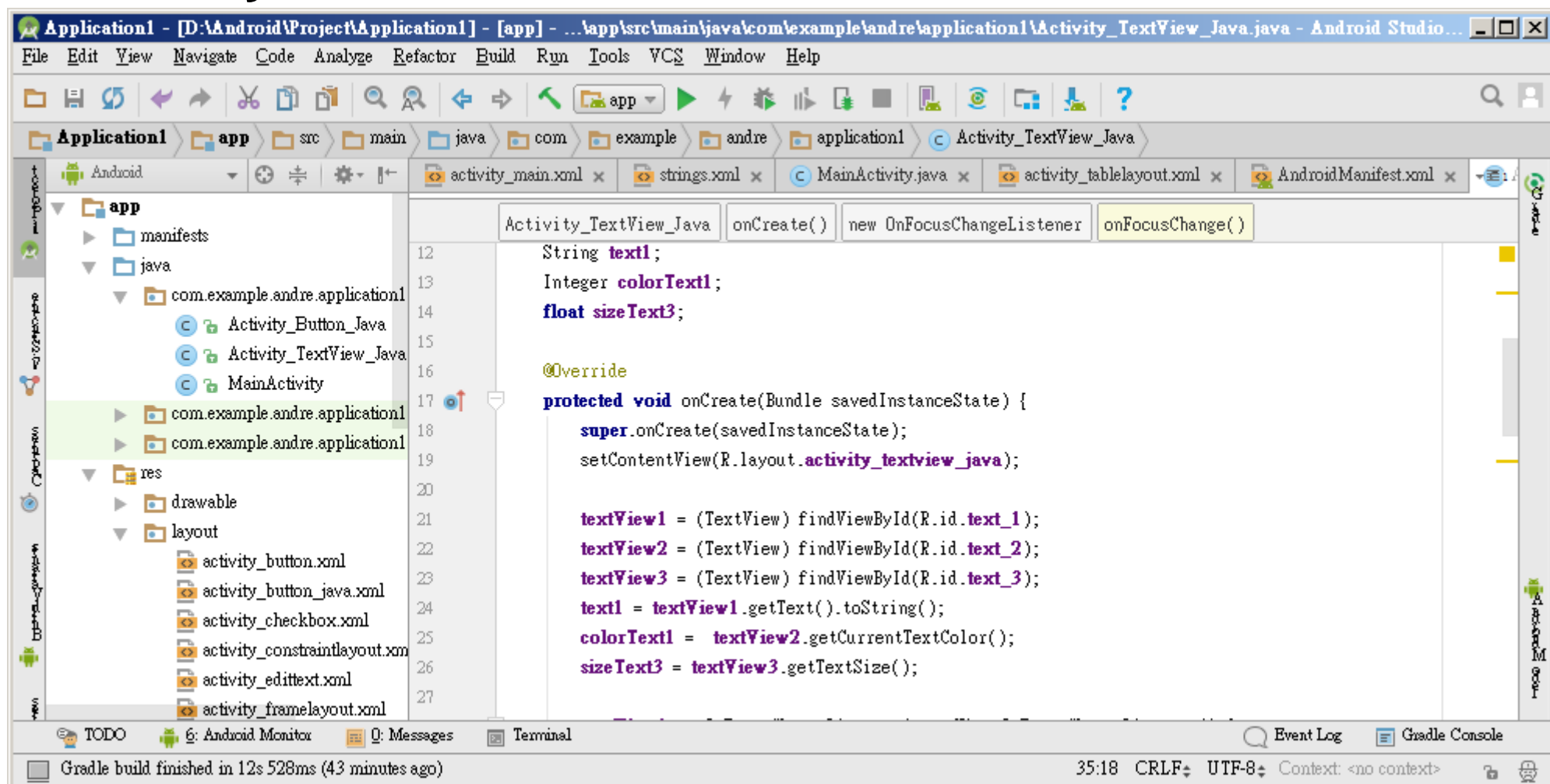
ANDROID 元件與程式控制

- 利用 Java 程式碼控制 TextView - 常用函式
 - getText ()
 - 取回文字內容
 - getTextSize
 - 取回文字大小
 - getBackground()
 - 取回背景drawable Object
 - getTextColors()
 - 取回文字顏色



ANDROID 元件與程式控制

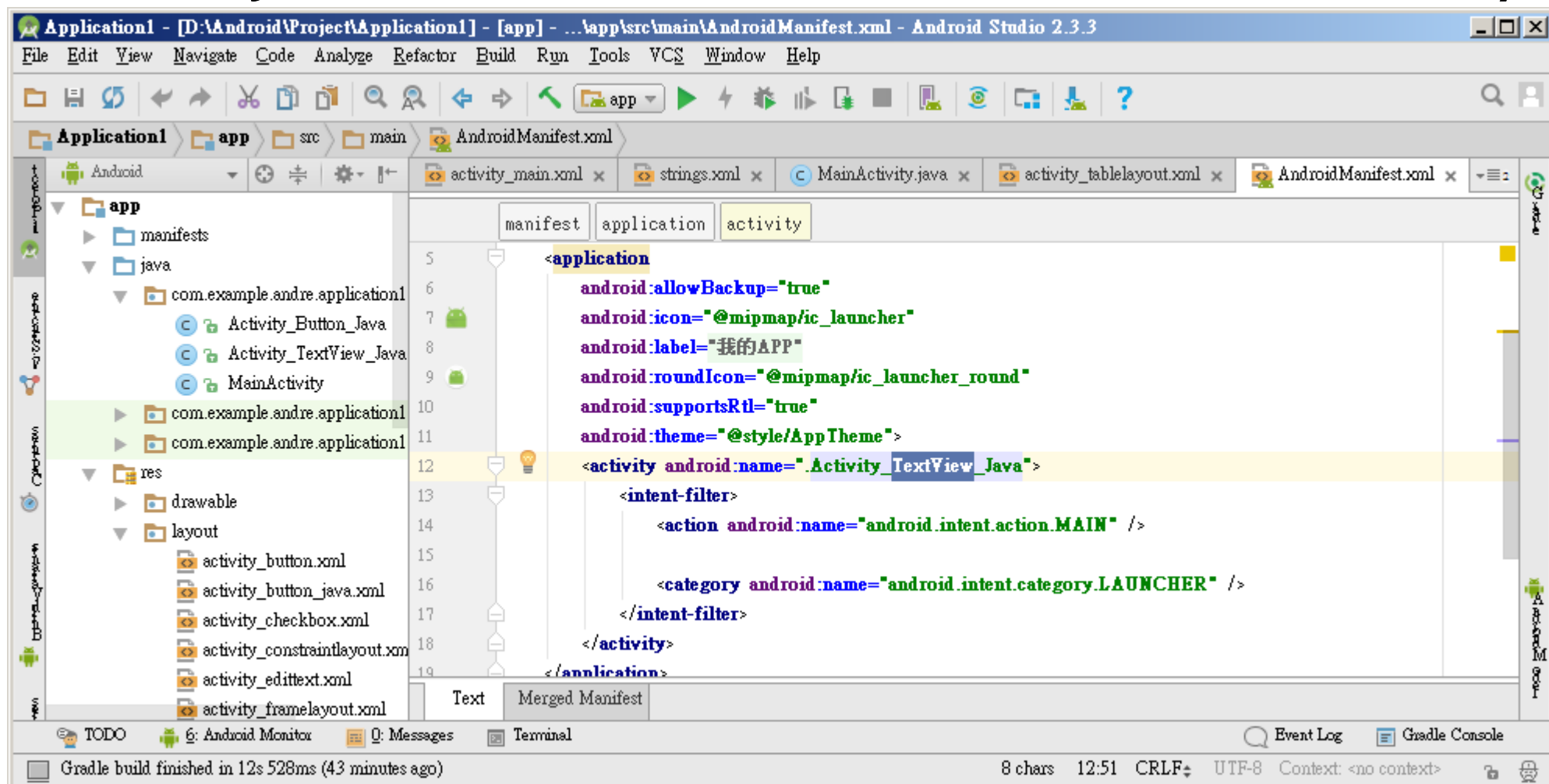
- 利用 Java 程式碼控制 TextView – 程式編輯器





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 TextView - 切換啟動 Activity





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 TextView - 切換啟動 Activity
- 開啟 AndroidManifest.xml，並於進行修改
 - `<application`
 - `android:allowBackup="true"`
 - `android:icon="@mipmap/ic_launcher"`
 - `android:label="@string/app_name"`
 - `android:roundIcon="@mipmap/ic_launcher_round"`
 - `android:supportRtl="true"`
 - `android:theme="@style/AppTheme">`
 - `<activity android:name=".Activity_TextView_Java">`
 - `<intent-filter>`
 - `<action android:name="android.intent.action.MAIN" />`
 - `<category android:name="android.intent.category.LAUNCHER" />`
 - `</intent-filter>`
 - `</activity>`
 - `</application>`



ANDROID 元件與程式控制

- 成果 - 程式控制 TextView





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 ImageView
- 開啟 `activity_imageview_java.xml`，並於其中加入以下程式碼
 - `<LinearLayout`
`xmlns:android="http://schemas.android.com/apk/res/android"`
 - `android:layout_width="match_parent"`
 - `android:layout_height="match_parent"`
 - `android:gravity="center_horizontal"`
 - `android:orientation="vertical"`
 - `android:focusable="true"`
 - `android:focusableInTouchMode="true"`
 - `>`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 ImageView
- 開啟 activity_imageview_java.xml，續
 - <ImageView
 - android:id="@+id/imageview_1"
 - android:layout_width="fill_parent"
 - android:layout_height="300dp"
 - android:scaleType="center"
 - android:src="@drawable/picture01"
 - />



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 ImageView
- 開啟 activity_imageview_java.xml，續
 - <TextView
 - android:id="@+id/text_1"
 - android:layout_width="match_parent"
 - android:layout_height="wrap_content"
 - android:text="切換圖片"
 - android:textColor="#000000"
 - android:textSize="24sp"
 - android:focusable="true"
 - android:focusableInTouchMode="true"
 - android:clickable="true"/>



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 ImageView
- 開啟 activity_imageview_java.xml，續
 - `<TextView`
 - `android:id="@+id/text_2"`
 - `android:layout_width="match_parent"`
 - `android:layout_height="wrap_content"`
 - `android:text="變更顯示方式"`
 - `android:textColor="#000000"`
 - `android:textSize="24sp"`
 - `android:focusable="true"`
 - `android:focusableInTouchMode="true"`
 - `android:clickable="true"/>`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 ImageView
- 開啟 Activity_ImageView_Java.java，並於其中加入以下程式碼
 - import android.view.MotionEvent;
 - import android.view.View;
 - import android.widget.Button;
 - import android.widget.TextView;
 - import android.widget.ImageView;
 - public class Activity_ImageView_Java extends AppCompatActivity {
 - TextView textView1, textView2;



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 ImageView
- 開啟 Activity_ImageView_Java.java , 續
 - `ImageView image;`
 - `Integer imageIndex = 0, scaleIndex = 0;`
 - `@Override`
 - `protected void onCreate(Bundle savedInstanceState) {`
 - `super.onCreate(savedInstanceState);`
 - `setContentView(R.layout.activity_imageview_java);`
 - `textView1 = (TextView) findViewById(R.id.text_1);`
 - `textView2 = (TextView) findViewById(R.id.text_2);`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 ImageView
- 開啟 Activity_ImageView_Java.java , 續
 - `image = (ImageView) findViewById(R.id.imageview_1);`
 - `textView1.setOnTouchListener(new Button.OnTouchListener() {`
 - `@Override`
 - `public boolean onTouch(View v, MotionEvent motionEvent) {`
 - `if (motionEvent.getAction() == MotionEvent.ACTION_DOWN) {`
 - `if (imageIndex == 2) {`
 - `image.setImageResource(R.drawable.imageview_java_1);`
 - `imageIndex = 1;`
 - `} else {`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 ImageView
- 開啟 Activity_ImageView_Java.java，續
 - `image.setImageResource(R.drawable.imageview_java_2);`
 - `imageIndex = 2;`
 - `}`
 - `}`
 - `return false;`
 - `}`
 - `});`
 - `textView2.setOnFocusChangeListener(new View.OnFocusChangeListener()`
 - `{`
 - `@Override`
 - `public void onFocusChange(View v, boolean hasFocus) {`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 ImageView
- 開啟 Activity_ImageView_Java.java , 續
 - if (hasFocus) {
 - if (scaleIndex == 2) {
 - image.setScaleType(ImageView.ScaleType.FIT_CENTER);
 - scaleIndex = 1;
 - } else {
 - image.setScaleType(ImageView.ScaleType.CENTER);
 - scaleIndex = 2;
 - }
 - }
 - }
 - });
 - }



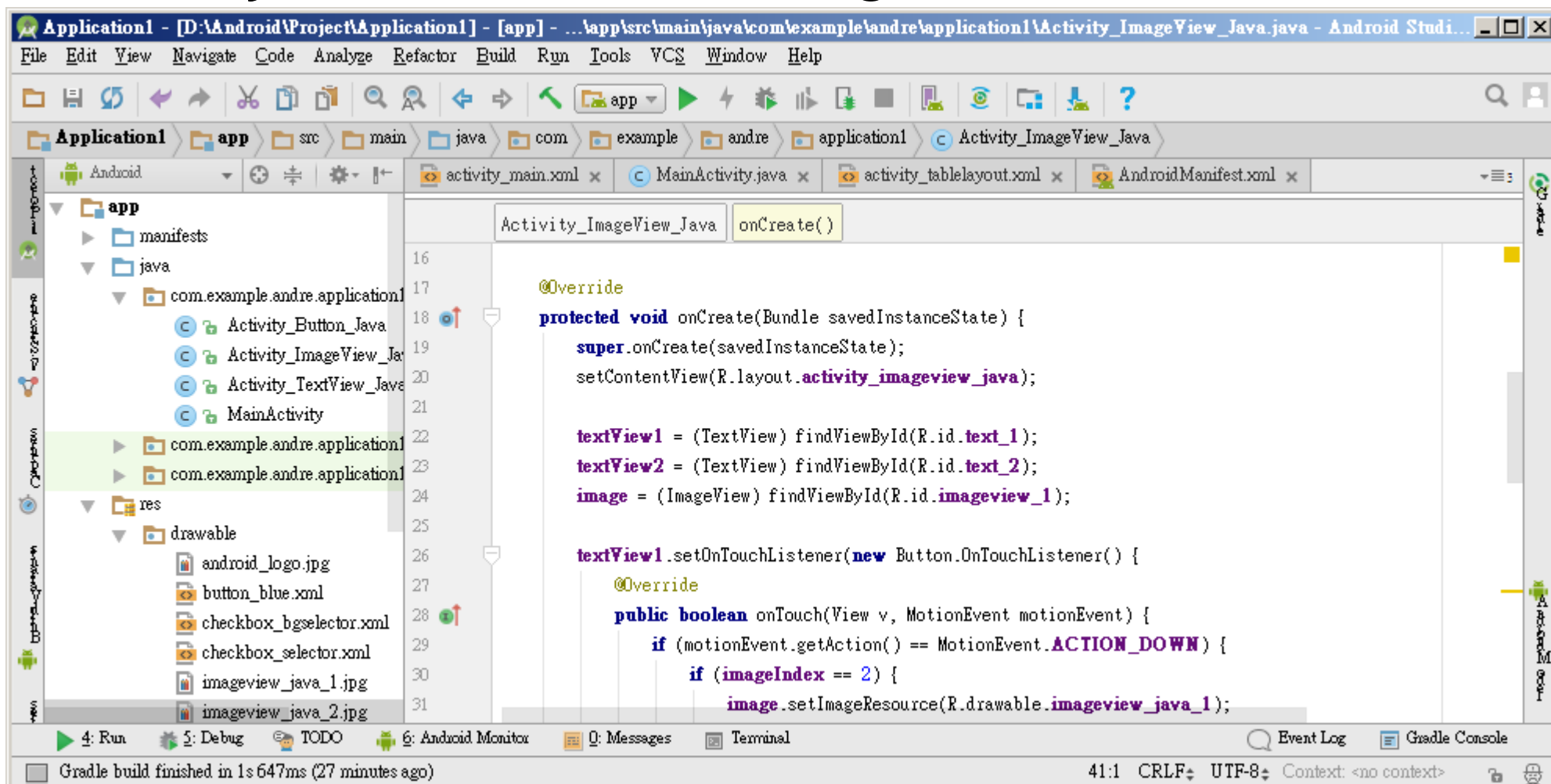
ANDROID 元件與程式控制

- 利用 Java 程式碼控制 ImageView - 常用函式
 - **setScaleType**
 - 設定顯示方式
 - `ImageView.ScaleType.FIT_CENTER`
 - `ImageView.ScaleType.CENTER`
 - `ImageView.ScaleType.CENTER_CROP`
 - `ImageView.ScaleType.FIT_XY`
 - **setImageResource**
 - 設定影像來源



ANDROID 元件與程式控制

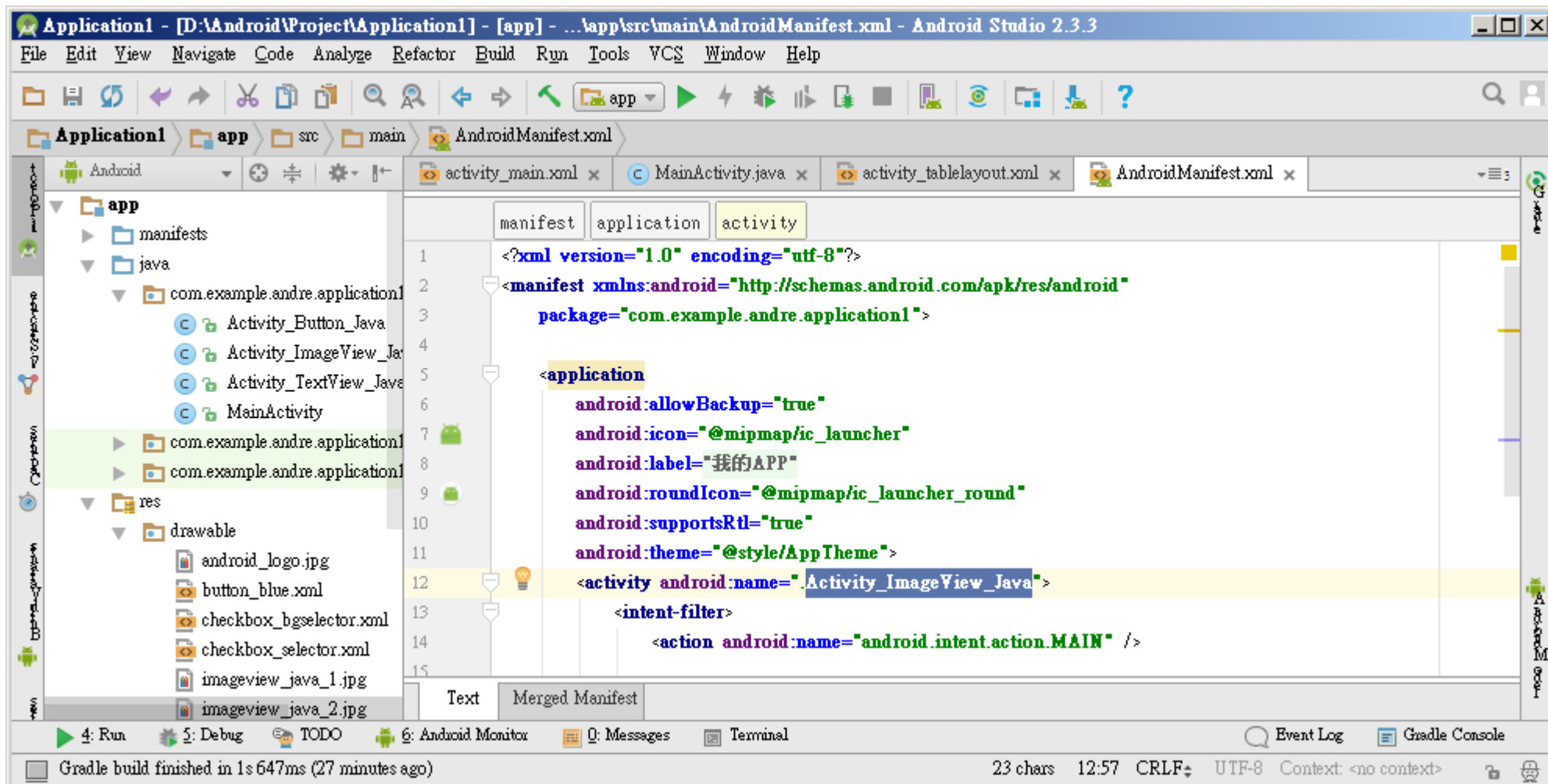
- 利用 Java 程式碼控制 ImageView – 程式編輯器





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 ImageView - 切換啟動 Activity





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 ImageView - 切換啟動 Activity
- 開啟 AndroidManifest.xml，並於進行修改
 - `<application`
 - `android:allowBackup="true"`
 - `android:icon="@mipmap/ic_launcher"`
 - `android:label="@string/app_name"`
 - `android:roundIcon="@mipmap/ic_launcher_round"`
 - `android:supportRtl="true"`
 - `android:theme="@style/AppTheme">`
 - `<activity android:name=".Activity_ImageView_Java">`
 - `<intent-filter>`
 - `<action android:name="android.intent.action.MAIN" />`
 - `<category android:name="android.intent.category.LAUNCHER" />`
 - `</intent-filter>`
 - `</activity>`
 - `</application>`



ANDROID 元件與程式控制

- 成果 - 程式控制 ImageView





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 EditText
- 開啟 activity_edittext_java.xml，並於其中加入以下程式碼
 - `<LinearLayout`
`xmlns:android="http://schemas.android.com/apk/res/android"`
 - `android:layout_width="match_parent"`
 - `android:layout_height="match_parent"`
 - `android:focusable="true"`
 - `android:focusableInTouchMode="true"`
 - `android:orientation="vertical">`
 - `<LinearLayout`
`xmlns:android="http://schemas.android.com/apk/res/android"`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 EditText
- 開啟 activity_edittext_java.xml，續
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:orientation="horizontal">
 - <EditText
 - android:id="@+id/edit_1"
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:hint="數字1"
 - android:maxLines="1"



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 EditText
- 開啟 activity_edittext_java.xml , 續
 - `android:textColor="#ff8c00"`
 - `android:textColorHighlight="#1e2659"`
 - `android:textColorHint="#238745"`
 - `android:inputType="number" />`
 - `<TextView`
 - `android:id="@+id/text_add"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:text="+"`
 - `android:textColor="#000000"`
 - `android:textSize="24sp" />`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 EditText
- 開啟 activity_edittext_java.xml，續
 - <EditText
 - android:id="@+id/edit_2"
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:hint="數字2"
 - android:maxLines="1"
 - android:textColor="#ff8c00"
 - android:textColorHighlight="#1e2659"
 - android:textColorHint="#238745"
 - android:inputType="number" />



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 EditText
- 開啟 activity_edittext_java.xml，續
 - <TextView
 - android:id="@+id/text_answer"
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:text=""
 - android:textColor="#000000"
 - android:textSize="24sp" />
 - </LinearLayout>



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 EditText
- 開啟 activity_edittext_java.xml，續
 - <Button
 - android:id="@+id/btn_add"
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:text="加法" />
 - </LinearLayout>



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 EditText
- 開啟 Activity_EditText_Java.java，並於其中加入以下程式碼
 - import android.view.View;
 - import android.widget.Button;
 - import android.widget.EditText;
 - import android.widget.TextView;
 - public class Activity_EditText_Java extends AppCompatActivity {
 - Button button_Add;
 - TextView text_Answer;



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 EditText
- 開啟 Activity_EditText_Java.java ， 續
 - EditText edit_1, edit_2;
 - Integer int_1, int_2, int_answer;
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_edittext_java);
 - button_Add = (Button) findViewById(R.id.btn_add);



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 EditText
- 開啟 Activity_EditText_Java.java , 續
 - `text_Answer = (TextView) findViewById(R.id.text_answer);`
 - `edit_1 = (EditText) findViewById(R.id.edit_1);`
 - `edit_2 = (EditText) findViewById(R.id.edit_2);`
 - `button_Add.setOnClickListener(new View.OnClickListener() {`
 - `@Override`
 - `public void onClick(View v) {`
 - `int_1 = Integer.parseInt(edit_1.getText().toString());`
 - `int_2 = Integer.parseInt(edit_2.getText().toString());`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 EditText
- 開啟 Activity_EditText_Java.java ， 續
 - `int_1 = Integer.parseInt(edit_1.getText().toString());`
 - `int_2 = Integer.parseInt(edit_2.getText().toString());`
 - `int_answer = int_1 + int_2;`
 - `text_Answer.setText("=" + String.valueOf(int_answer));`
 - `}`
 - `});`
 -



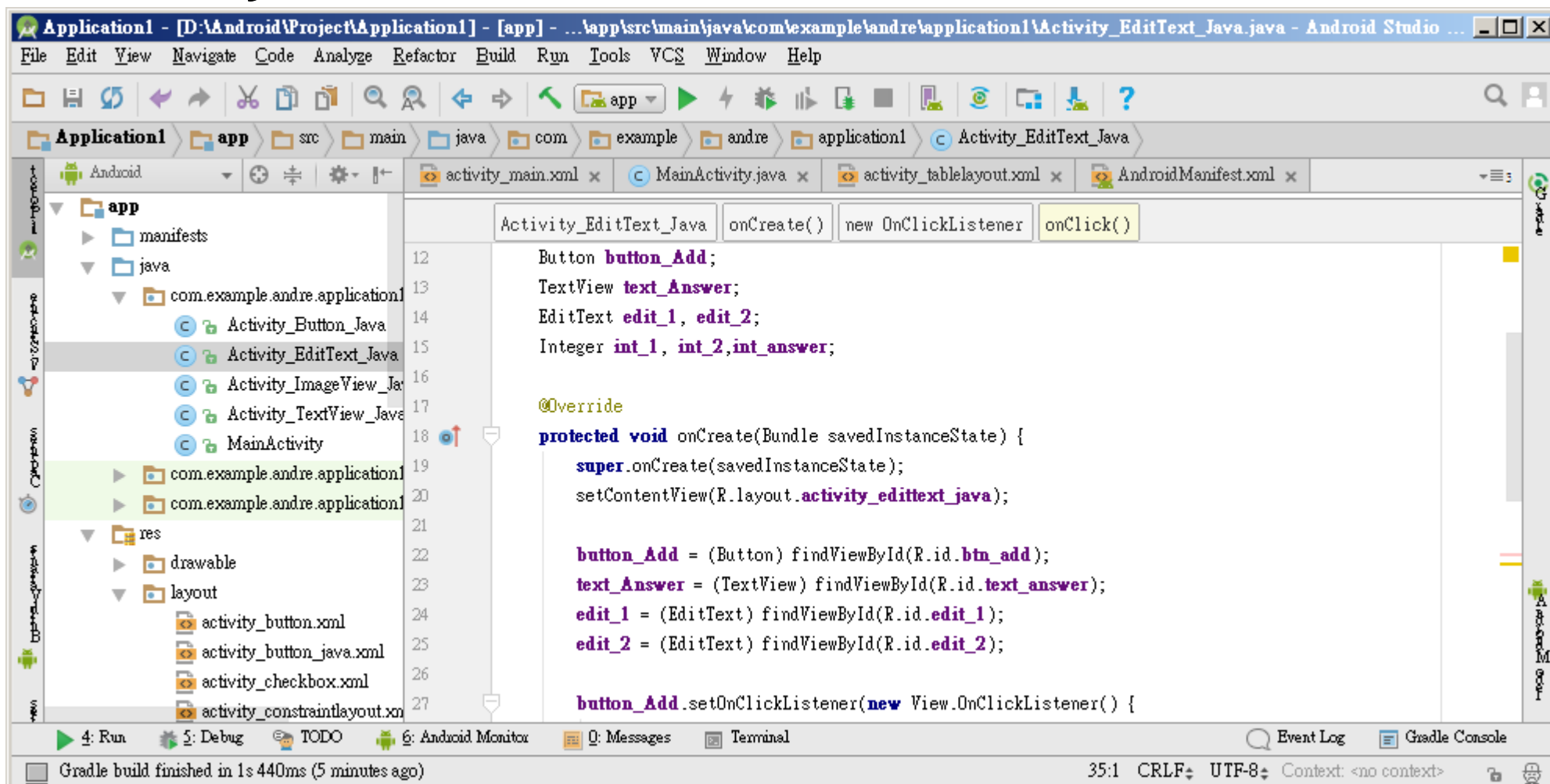
ANDROID 元件與程式控制

- 利用 Java 程式碼控制 EditText - 常用函式
 - setText ()
 - 設定文字內容
 - getText ()
 - 取得文字內容
 - toString()
 - 轉換成字串
 - Integer.parseInt()
 - 轉換成整數



ANDROID 元件與程式控制

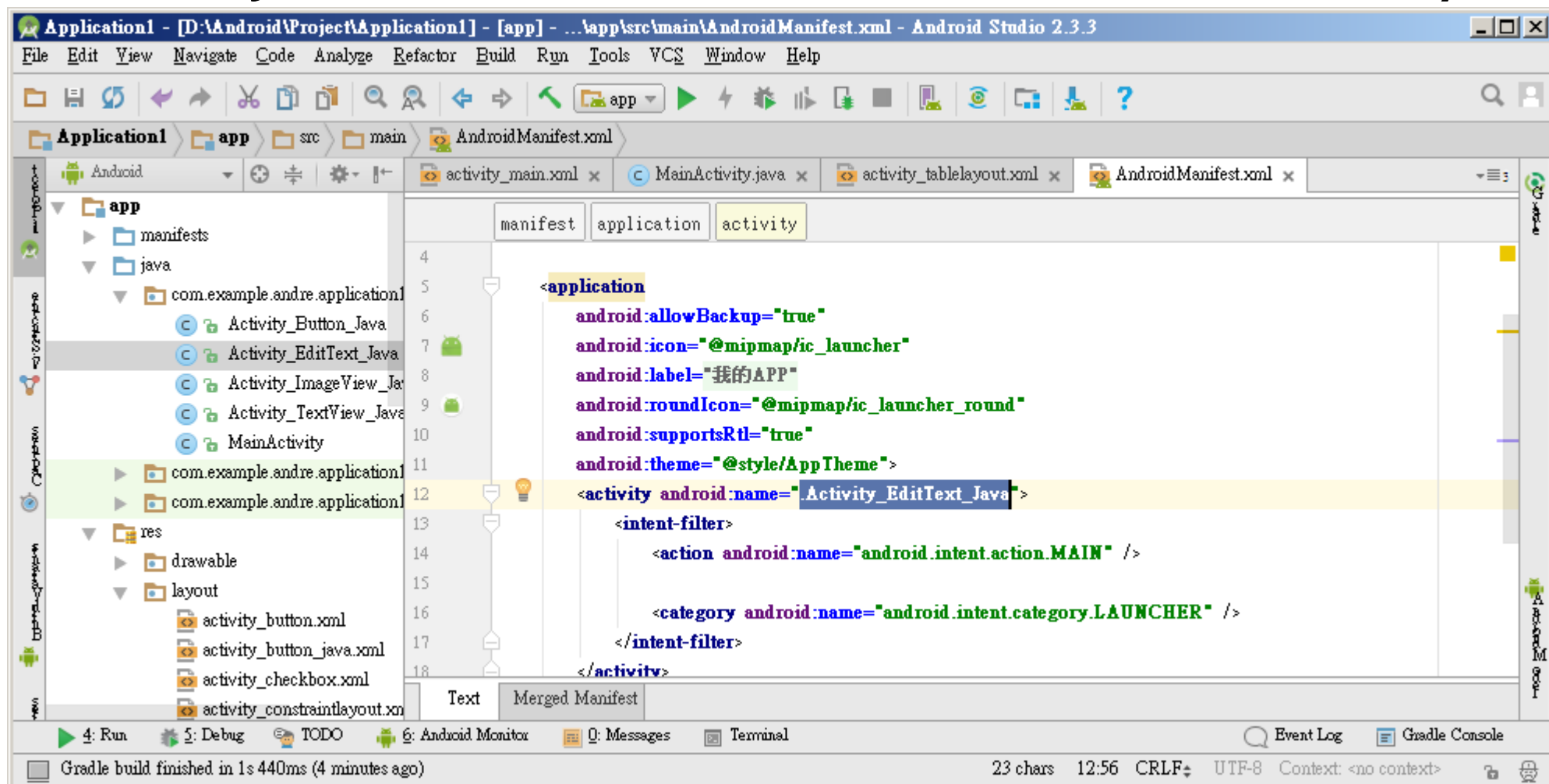
- 利用 Java 程式碼控制 EditText – 程式編輯器





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 EditText - 切換啟動 Activity





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 EditText - 切換啟動 Activity
- 開啟 AndroidManifest.xml，並於進行修改
 - `<application`
 - `android:allowBackup="true"`
 - `android:icon="@mipmap/ic_launcher"`
 - `android:label="@string/app_name"`
 - `android:roundIcon="@mipmap/ic_launcher_round"`
 - `android:supportRtl="true"`
 - `android:theme="@style/AppTheme">`
 - `<activity android:name=".Activity_EditText_Java">`
 - `<intent-filter>`
 - `<action android:name="android.intent.action.MAIN" />`
 - `<category android:name="android.intent.category.LAUNCHER" />`
 - `</intent-filter>`
 - `</activity>`
 - `</application>`



ANDROID 元件與程式控制

- 成果 - 程式控制 EditText





ANDROID 元件與程式控制

- 請嘗試實現攝氏溫度轉華氏溫度的計算
 - 華氏 = 攝氏*(9/5)+32
 - 攝氏 = (華氏-32)*5/9



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 CheckBox
- 開啟 `activity_checkbox_java.xml`，並於其中加入以下程式碼
 - `<LinearLayout`
`xmlns:android="http://schemas.android.com/apk/res/android"`
 - `android:layout_width="match_parent"`
 - `android:layout_height="match_parent"`
 - `android:focusable="true"`
 - `android:focusableInTouchMode="true"`
 - `android:orientation="vertical">`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 CheckBox
- 開啟 activity_checkbox_java.xml，續
 - <TextView
 - android:id="@+id/text_question1"
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:text="1.本學期教學活動方式:"
 - android:textColor="#000000"
 - android:textSize="16sp" />



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 CheckBox
- 開啟 activity_checkbox_java.xml，續
 - <CheckBox
 - android:id="@+id/checkbox_q1"
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:text="實務經驗分享"
 - android:textSize="12sp"
 - android:textColor="#ff0000"
 - android:layout_marginLeft="10dp"/>



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 CheckBox
- 開啟 activity_checkbox_java.xml，續
 - <CheckBox
 - android:id="@+id/checkbox_q2"
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:text="操作教學、實習課程"
 - android:textSize="12sp"
 - android:textColor="#ff0000"
 - android:layout_marginLeft="10dp"/>



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 CheckBox
- 開啟 activity_checkbox_java.xml，續
 - <CheckBox
 - android:id="@+id/checkbox_q3"
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:text="輔導證照考試"
 - android:textSize="12sp"
 - android:textColor="#ff0000"
 - android:layout_marginLeft="10dp"/>



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 CheckBox
- 開啟 activity_checkbox_java.xml，續
 - <CheckBox
 - android:id="@+id/checkbox_q4"
 - android:layout_width="wrap_content"
 - android:layout_height="wrap_content"
 - android:text="校外參觀、體驗學習"
 - android:textSize="12sp"
 - android:textColor="#ff0000"
 - android:layout_marginLeft="10dp"/>
 - </LinearLayout>



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Checkbox
- 開啟 `Activity_Checkbox_Java.java`，並於其中加入以下程式碼
 - `import android.widget.CheckBox;`
 - `import android.widget.CompoundButton;`
 - `import android.widget.Toast;`
 - `public class Activity_CheckBox_Java extends AppCompatActivity {`
 - `CheckBox CB_1, CB_2, CB_3, CB_4;`
 - `String stringChose;`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Checkbox
- 開啟 Activity_Checkbox_Java.java ， 續
 - @Override
 - protected void onCreate(Bundle savedInstanceState) {
 - super.onCreate(savedInstanceState);
 - setContentView(R.layout.activity_checkbox_java);
 - CB_1 = (CheckBox) findViewById(R.id.checkbox_q1);
 - CB_2 = (CheckBox) findViewById(R.id.checkbox_q2);
 - CB_3 = (CheckBox) findViewById(R.id.checkbox_q3);
 - CB_4 = (CheckBox) findViewById(R.id.checkbox_q4);



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Checkbox
- 開啟 Activity_Checkbox_Java.java , 續
 - `CB_1.setOnCheckedChangeListener(checkListener);`
 - `CB_2.setOnCheckedChangeListener(checkListener);`
 - `CB_3.setOnCheckedChangeListener(checkListener);`
 - `CB_4.setOnCheckedChangeListener(checkListener);`
 - `}`
 - `CheckBox.OnCheckedChangeListener checkListener = new`
`CheckBox.OnCheckedChangeListener() {`
 - `@Override`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Checkbox
- 開啟 Activity_Checkbox_Java.java ， 續
 - `public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {`
 - `stringChose = "已選擇:";`
 - `if (CB_1.isChecked() == true) {`
 - `stringChose = stringChose + CB_1.getText().toString();`
 - `}`
 - `if (CB_2.isChecked() == true) {`
 - `stringChose = stringChose + “ , ” + CB_2.getText().toString();`
 - `}`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Checkbox
- 開啟 Activity_Checkbox_Java.java ， 續
 - if (CB_3.isChecked() == true) {
 - stringChose = stringChose + " , " + CB_3.getText().toString();
 - }
 - if (CB_4.isChecked() == true) {
 - stringChose = stringChose + " , " + CB_4.getText().toString();
 - }
 - Toast.makeText(Activity_CheckBox_Java.this, stringChose, Toast.LENGTH_LONG).show();
 - }
 - };



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Checkbox - 常用函式
 - isChecked()
 - 檢查是否 Checkbox 被選取



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Checkbox – 程式編輯器

```
Application1 - [D:\Android\Project\Application1] - [app] - ...app\src\main\java\com\example\andre\application1\Activity_CheckBox_Java.java - Android Studi...
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Application1 app src main java com example andre application1 Activity_CheckBox_Java
activity_main.xml x MainActivity.java x AndroidManifest.xml x activity_edittext_java.xml x activity_checkbox_java.xml x

Activity_CheckBox_Java onCreate()
12 String stringChose;
13
14 @Override
15 protected void onCreate(Bundle savedInstanceState) {
16     super.onCreate(savedInstanceState);
17     setContentView(R.layout.activity_checkbox_java);
18
19     CB_1 = (CheckBox) findViewById(R.id.checkbox_q1);
20     CB_2 = (CheckBox) findViewById(R.id.checkbox_q2);
21     CB_3 = (CheckBox) findViewById(R.id.checkbox_q3);
22     CB_4 = (CheckBox) findViewById(R.id.checkbox_q4);
23
24     CB_1.setOnCheckedChangeListener(checkListener);
25     CB_2.setOnCheckedChangeListener(checkListener);
26     CB_3.setOnCheckedChangeListener(checkListener);
27     CB_4.setOnCheckedChangeListener(checkListener);
28 }
```

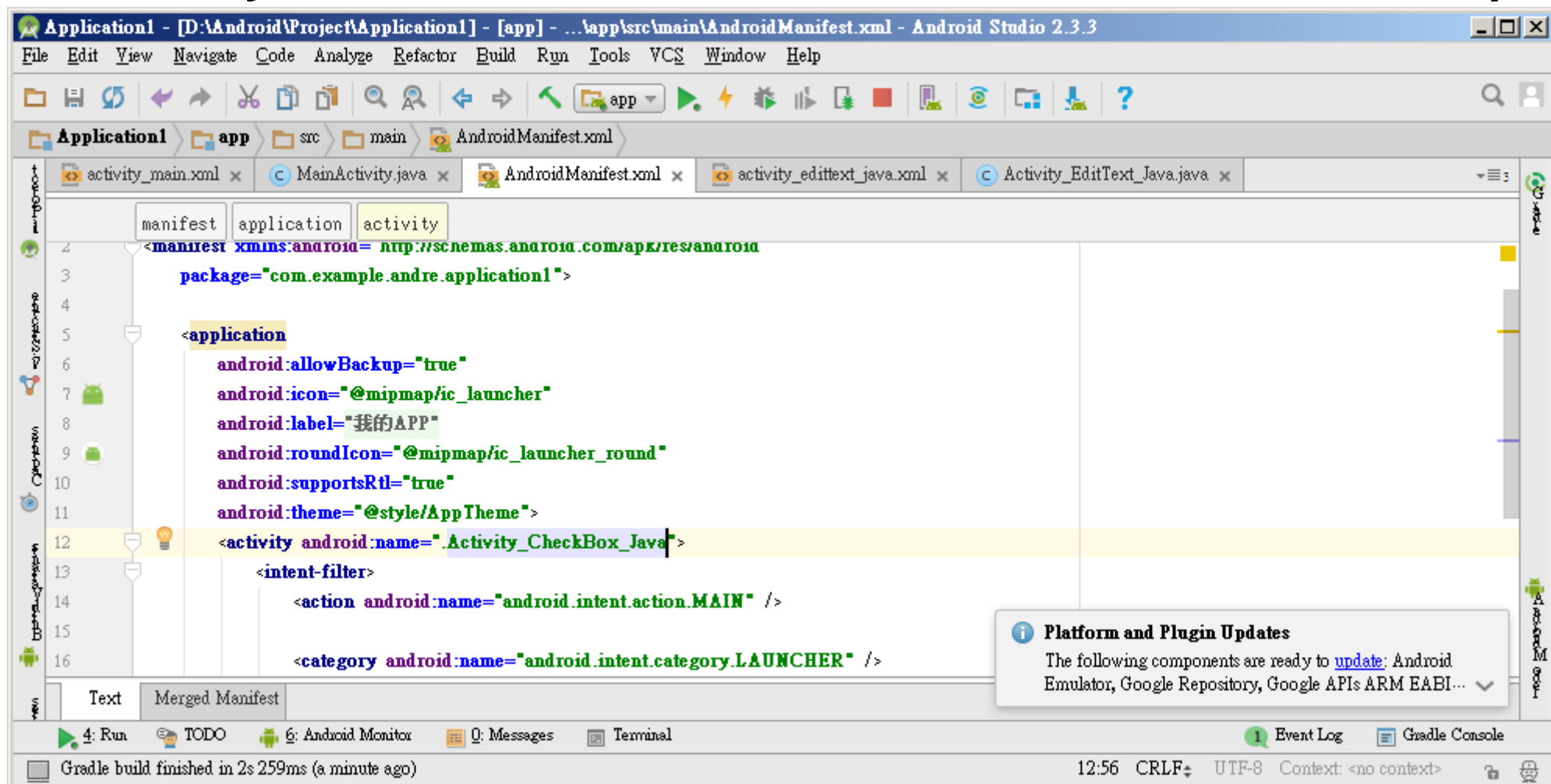
4: Run TODO 6: Android Monitor Q: Messages Terminal Event Log Gradle Console

Gradle build finished in 2s 259ms (4 minutes ago) 14:14 CRLF UTF-8 Context: <no context>



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Checkbox - 切換啟動 Activity





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 Checkbox - 切換啟動 Activity
- 開啟 AndroidManifest.xml，並於進行修改
 - `<application`
 - `android:allowBackup="true"`
 - `android:icon="@mipmap/ic_launcher"`
 - `android:label="@string/app_name"`
 - `android:roundIcon="@mipmap/ic_launcher_round"`
 - `android:supportRtl="true"`
 - `android:theme="@style/AppTheme">`
 - `<activity android:name=".Activity_Checkbox_Java">`
 - `<intent-filter>`
 - `<action android:name="android.intent.action.MAIN" />`
 - `<category android:name="android.intent.category.LAUNCHER" />`
 - `</intent-filter>`
 - `</activity>`
 - `</application>`



ANDROID 元件與程式控制

- 成果 - 程式控制 Checkbox





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton
- 開啟 activity_radiobutton_java.xml，並於其中加入以下程式碼
 - `<LinearLayout`
 - `xmlns:android="http://schemas.android.com/apk/res/android"`
 - `android:layout_width="match_parent"`
 - `android:layout_height="match_parent"`
 - `android:orientation="vertical"`
 - `>`
 - `<TextView`
 - `android:id="@+id/textview_1"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton
- 開啟 activity_radiobutton_java.xml，續
 - `android:text="1.你最常上下面哪一個網站?"`
 - `android:textSize="16sp"/>`
 - `<RadioGroup`
 - `android:id="@+id/radiogroup_1"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_marginLeft="10dp">`
 - `<RadioButton`
 - `android:id="@+id/radiobutton_1"`
 - `android:layout_width="wrap_content"`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton
- 開啟 activity_radiobutton_java.xml，續
 - `android:layout_height="wrap_content"`
 - `android:layout_weight="1"`
 - `android:text="www.google.com" />`
 - `<RadioButton`
 - `android:id="@+id/radiobutton_2"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_weight="1"`
 - `android:text="www.yahoo.com.tw"`
 - `android:textColor="#00ff00" />`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton
- 開啟 activity_radiobutton_java.xml，續
 - `<RadioButton`
 - `android:id="@+id/radiobutton_3"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_weight="1"`
 - `android:text="www.cnyes.com" />`
 - `</RadioGroup>`
 - `<TextView`
 - `android:id="@+id/textview_2"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton
- 開啟 activity_radiobutton_java.xml，續
 - `android:text="2.你最常做的休閒活動是什麼?"`
 - `android:textSize="16sp" />`
 - `<RadioGroup`
 - `android:id="@+id/radiogroup_2"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_marginLeft="10dp" >`
 - `<RadioButton`
 - `android:id="@+id/radiobutton_4"`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton
- 開啟 activity_radiobutton_java.xml，續
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_weight="1"`
 - `android:text="爬山" />`
 - `<RadioButton`
 - `android:id="@+id/radiobutton_5"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_weight="1"`
 - `android:text="打籃球"`
 - `android:textColor="#00ff00" />`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton
- 開啟 activity_radiobutton_java.xml，續
 - `<RadioButton`
 - `android:id="@+id/radiobutton_6"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:layout_weight="1"`
 - `android:text="看電影" />`
 - `</RadioGroup>`
 - `<TextView`
 - `android:id="@+id/textview_ans_1"`
 - `android:layout_width="wrap_content"`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton
- 開啟 activity_radiobutton_java.xml，續
 - `android:layout_height="wrap_content"`
 - `android:text=""`
 - `android:textSize="16sp"/>`
 - `<TextView`
 - `android:id="@+id/textview_ans_2"`
 - `android:layout_width="wrap_content"`
 - `android:layout_height="wrap_content"`
 - `android:text=""`
 - `android:textSize="16sp"/>`
 - `</LinearLayout>`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton
- 開啟 `Activity_RadioButton_Java.java`，並於其中加入以下程式碼
 - `import android.os.Bundle;`
 - `import android.support.v7.app.AppCompatActivity;`
 - `import android.view.View;`
 - `import android.widget.RadioButton;`
 - `import android.widget.RadioGroup;`
 - `import android.widget.TextView;`
 - `public class Activity_RadioButton_Java extends AppCompatActivity {`
 - `TextView textAns1, textAns2;`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton
- 開啟 Activity_RadioButton_Java.java , 續
 - `RadioGroup radioGroup1,radioGroup2;`
 - `RadioButton radioButton4,radioButton5,radioButton6;`
 - `@Override`
 - `protected void onCreate(Bundle savedInstanceState) {`
 - `super.onCreate(savedInstanceState);`
 - `setContentView(R.layout.activity_radiobutton_java);`
 - `radioGroup1 = (RadioGroup)findViewById(R.id.radiogroup_1);`
 - `radioGroup2 = (RadioGroup)findViewById(R.id.radiogroup_2);`
 - `textAns1 = (TextView) findViewById(R.id.textview_ans_1);`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton
- 開啟 Activity_RadioButton_Java.java ， 續
 - `textAns2 = (TextView) findViewById(R.id.textview_ans_2);`
 - `radioButton4 = (RadioButton) findViewById(R.id.radiobutton_4);`
 - `radioButton5 = (RadioButton) findViewById(R.id.radiobutton_5);`
 - `radioButton6 = (RadioButton) findViewById(R.id.radiobutton_6);`
 - `radioGroup1.setOnCheckedChangeListener(radioListener1);`
 - `radioButton4.setOnClickListener(radioListener2);`
 - `radioButton5.setOnClickListener(radioListener2);`
 - `radioButton6.setOnClickListener(radioListener2);`
 - `}`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton
- 開啟 Activity_RadioButton_Java.java ， 續
 - `RadioGroup.OnCheckedChangeListener radioListener1 = new RadioGroup.OnCheckedChangeListener() {`
 - `@Override`
 - `public void onCheckedChanged(RadioGroup radioGroup, int i) {`
 - `int id;`
 - `id=radioGroup.getCheckedRadioButtonId();`
 - `if(id == R.id.radiobutton_1)`
 - `{`
 - `textAns1.setText("第一題選則`
 - `了:"+((RadioButton)findViewById(R.id.radiobutton_1)).getText());`
 - `}`



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton
- 開啟 Activity_RadioButton_Java.java ， 續
 - else if (id == R.id.radiobutton_2)
 - {
 - textAnsI.setText("第一題選則
了:" + ((RadioButton)findViewById(R.id.radiobutton_2)).getText());
 - }
 - else if (id == R.id.radiobutton_3)
 - {
 - textAnsI.setText("第一題選則
了:" + ((RadioButton)findViewById(R.id.radiobutton_3)).getText());
 - }
 - }
 - };



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton
- 開啟 Activity_RadioButton_Java.java ， 續
 - View.OnClickListener radioListener2 = new View.OnClickListener() {
 - @Override
 - public void onClick(View view) {
 - int selectedId;
 - selectedId = radioGroup2.getCheckedRadioButtonId();
 - textAns2.setText(" 第 二 題 選 則
了:" + ((RadioButton)findViewById(selectedId)).getText());
 - }
 - };
 - }
 -



ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton - 常用函式
 - **setText ()**
 - 設定文字內容
 - **setTextSize**
 - 設定文字大小
 - **setBackgroundColor()**
 - 設定背景顏色



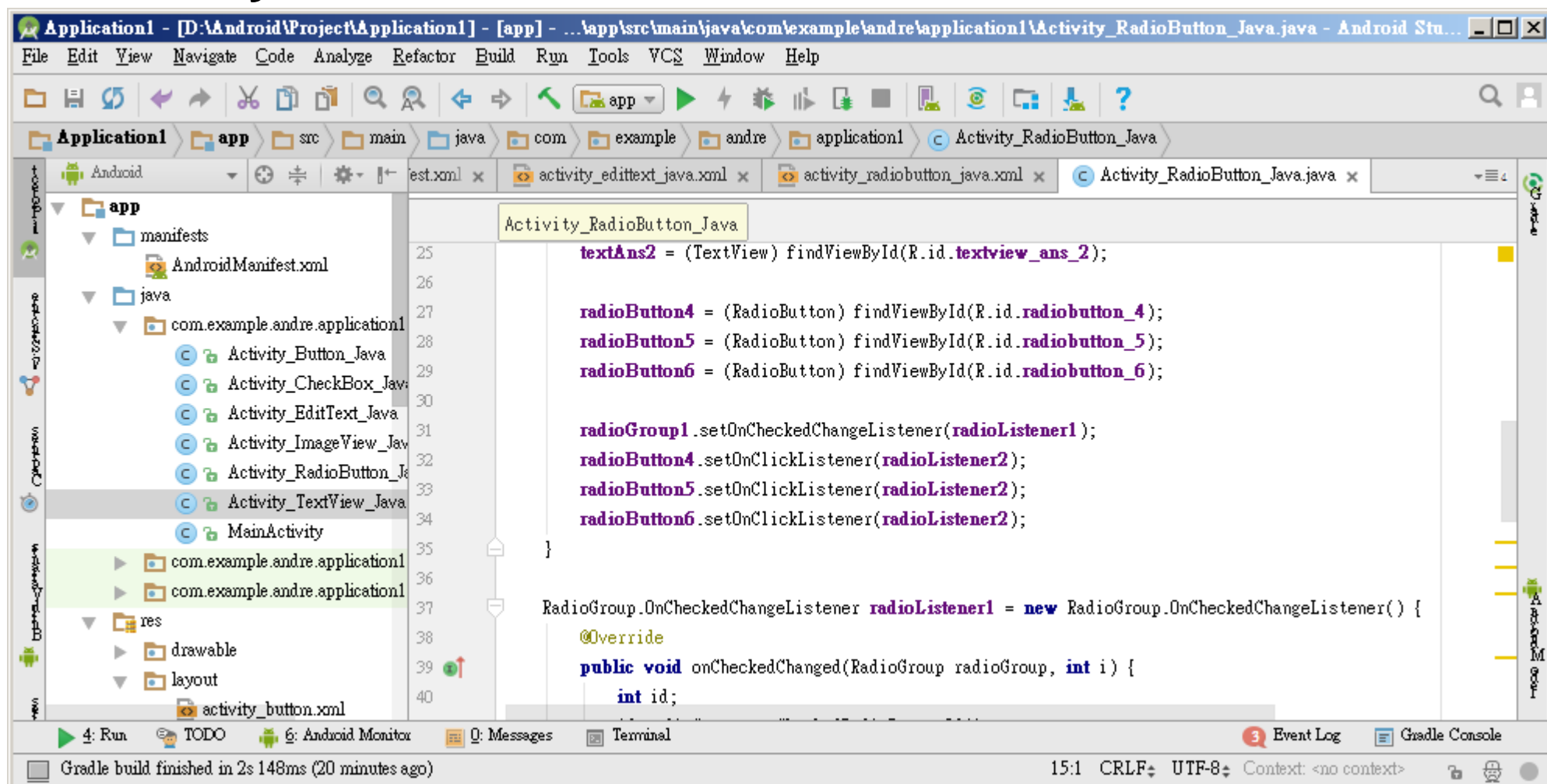
ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton - 常用函式
 - getText ()
 - 取回文字內容
 - getTextSize
 - 取回文字大小
 - getBackground()
 - 取回背景drawable Object
 - getTextColors()
 - 取回文字顏色



ANDROID 元件與程式控制

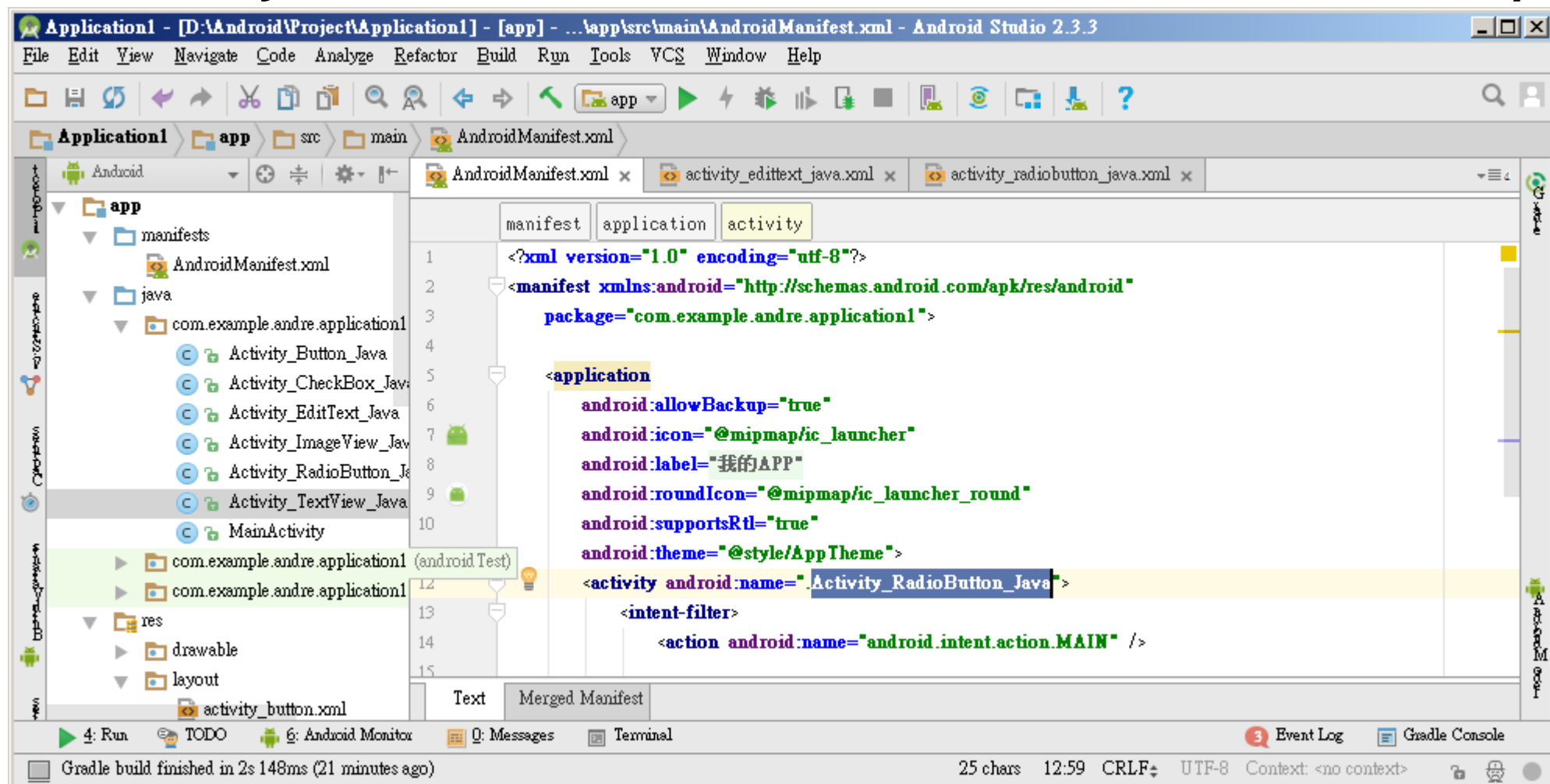
- 利用 Java 程式碼控制 RadioButton— 程式編輯器





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton-切換啟動 Activity





ANDROID 元件與程式控制

- 利用 Java 程式碼控制 RadioButton - 切換啟動 Activity
- 開啟 AndroidManifest.xml，並於進行修改
 - `<application`
 - `android:allowBackup="true"`
 - `android:icon="@mipmap/ic_launcher"`
 - `android:label="@string/app_name"`
 - `android:roundIcon="@mipmap/ic_launcher_round"`
 - `android:supportRtl="true"`
 - `android:theme="@style/AppTheme">`
 - `<activity android:name=".Activity_ RadioButton _Java">`
 - `<intent-filter>`
 - `<action android:name="android.intent.action.MAIN" />`
 - `<category android:name="android.intent.category.LAUNCHER" />`
 - `</intent-filter>`
 - `</activity>`
 - `</application>`



ANDROID 元件與程式控制

- 成果-程式控制 RadioButton





 **THE END**