



行動裝置程式設計

Unit 05 活動 Activity

蘇維宗 (Wei-Tsung Su)
suwt@au.edu.tw
564D



課程目標

認識Activity生命週期

Android使用者介面

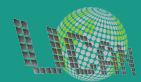
初探Layout

Android程式實戰

新增並啟動Activity

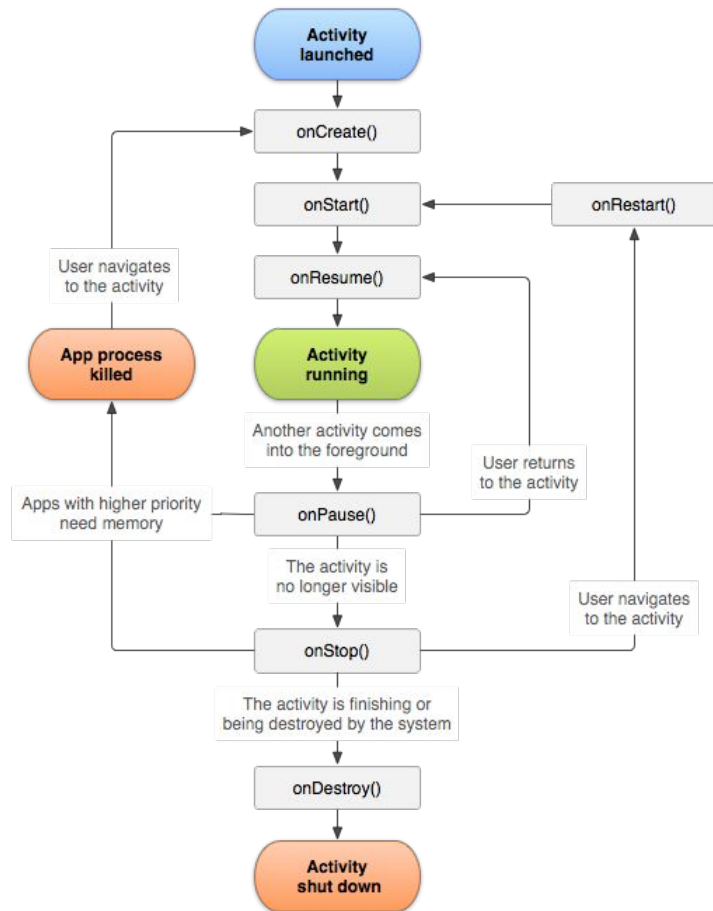


認識Activity生命週期



- Entire lifetime
 - onCreate() 和 onDestroy() 之間
 - 通常用來初始化 (onCreate) 和釋放資源 (onDestroy)
- Visible lifetime
 - onStart() 和 onStop() 之間
 - 使用者可以看到此 Activity 並互動
 - 維護需顯示在 Activity 上的資源
- Foreground lifetime
 - onResume() 和 onPause() 之間
 - 此 Activity 在其他 Activities 的前面, 且使用者可以輸入內容。(當裝置進入睡眠模式或者出現對話框時就會呼叫 onPause)
 - 由於呼叫次數相當頻繁, 在兩個函式內的程式碼應盡量簡化。

Activity 生命週期(重要)





課堂練習

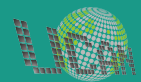
熟習 Activity Life Cycle

提示:
無

請利用LogCat觀察下列事件發生時，Activity生命週期的變化。

- 當目標Activity出現在畫面上時。
- 當目標Activity出現在畫面上，按下Back鍵時。
- 當目標Activity出現在畫面上，另一Activity被執行並出現在畫面上時。
- 當另一Activity結束，目標Activity再度出現在畫面上時。
- 當目標Activity出現在畫面上，手機進入休眠模式時。
- 當手機進入休眠模式，再度開啟手機並顯示目標Activity畫面時。

Android使用者介面



Android使用者介面(UI)

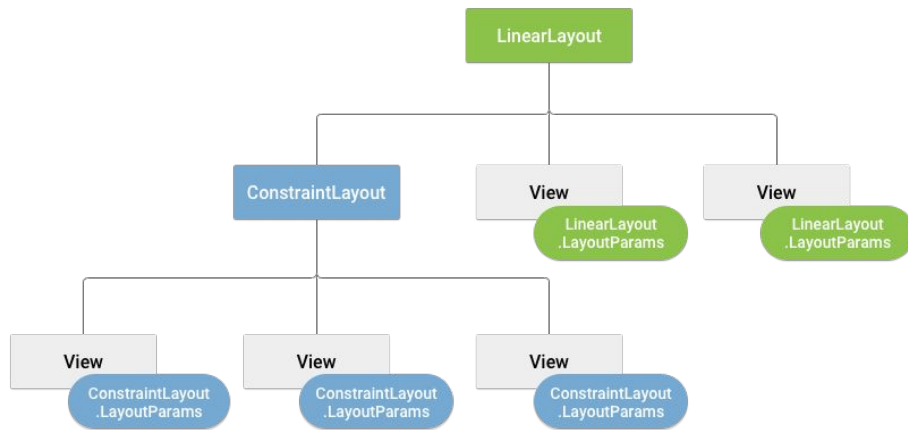
- 在 Android 應用程式中，主要的UI物件，包含
 - Layouts (版面配置)
 - Input Controls (控制項)
 - Menu (選單)
 - Action Bar (API level 11以上)
- 以及其他，包含
 - Status notification
 - Toast notification
 - Dialog
 - ...



使用者介面－應用程式介面是使用能看到且能互動的任何事情。(來源:Android Developers)

Layout

- Layout的種類決定UI的版面配置
- 常用的layout包含
 - LinearLayout
 - RelativeLayout
 - WebView
 - ListView
 - ScrollView
 - ...
- 更多的layout介紹可參考下列網址
 - <http://developer.android.com/guide/topics/ui/declaring-layout.html>



Layout階層架構(來源:Android Developers)

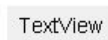
Input Control



AnalogClock



DatePicker



TimePicker



ProgressBar



Menu

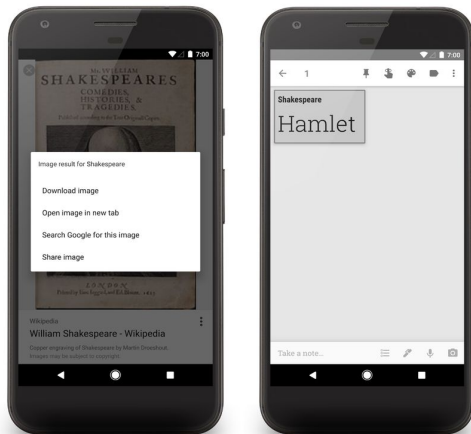


Android 3.0 (API level 11)以上支援Action Bar



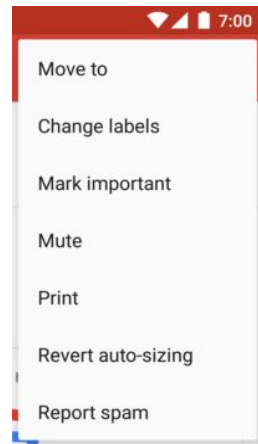
Option Menu

按下MENU鍵所出現的選單(ActionBar取代)



Contextual Menu

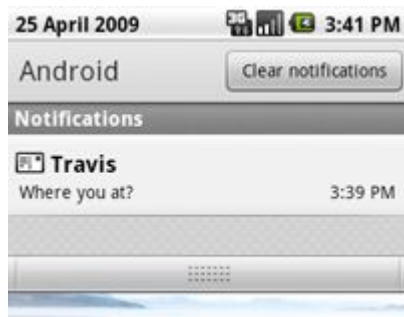
長壓所出現的選單



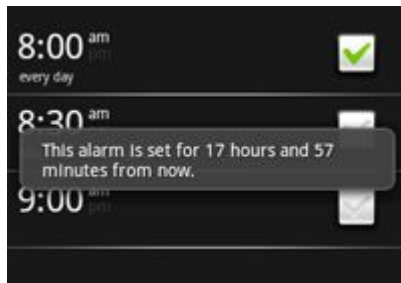
Popup Menu

通常是用來讓使用者選擇選項的子選單

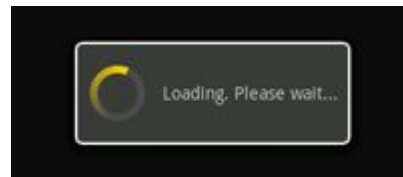
Notification與Dialog



Status notification
顯示在狀態列

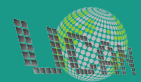


Toast notification
以彈出式窗顯示



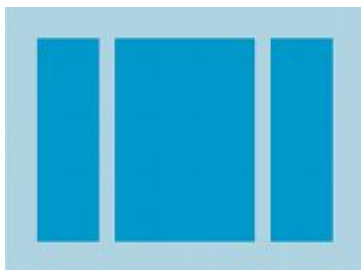
Dialog
以對話視窗顯示

初探Layout



常見的Layout

- Linear Layout: 垂直或水平排列
- Relative Layout: 以相對位置排列
- Web View: 顯示網頁



(a) Linear Layout



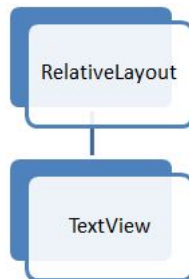
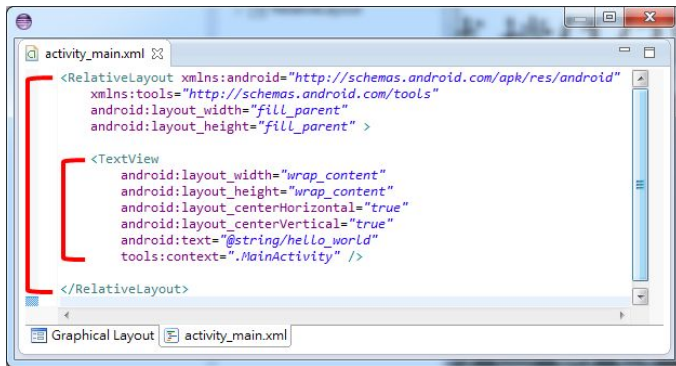
(b) Relative Layout



(b) Web View

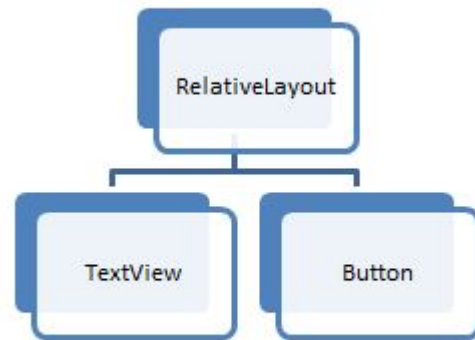
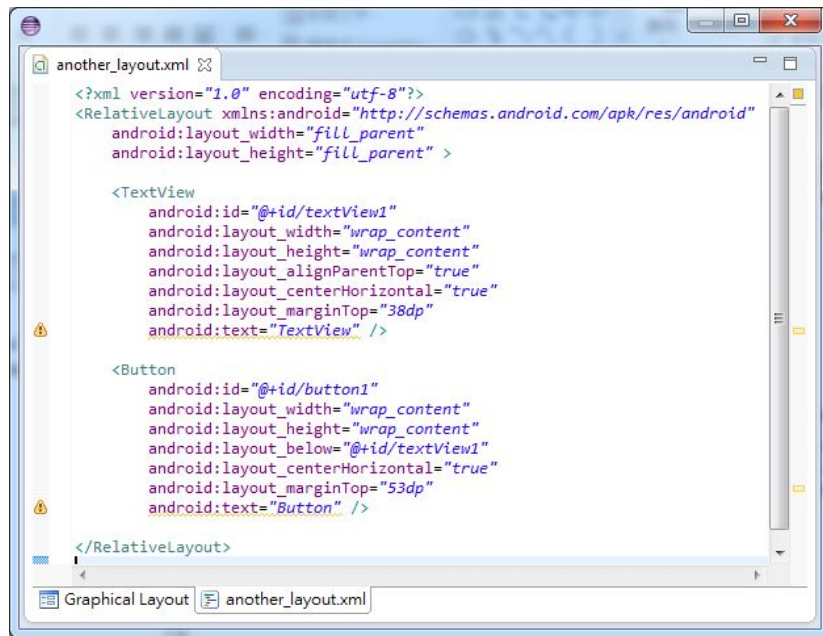
Layout描述檔案

- Layout為Android Activity的UI的描述(可想像成是一張畫布)
- Layout為位於 `res/layout` 底下的XML檔案, Layout上各個元素的格式為
 - `<tag 屬性 ...></tag>` 或可簡化為
 - `<tag 屬性 ... />`
- 例如,



Android使用者介面
階層式架構

Layout描述檔案 (續)



Android 使用者介面
階層式架構



Android如何為Activity載入Layout資源

透過呼叫 `setContentView()` 函式傳入Layout資源檔案來產生介面

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```





Layout屬性

- 不同的Layout或Input Control可能有共同或是自己獨特的屬性。
- 屬性的指定格式為
 - `android:屬性名稱="屬性值"`
- 不同的屬性可以決定不同 Layout或Input Control的樣式, 例如
 - `android:orientation` 屬性決定child的排列方向
 - 值為`vertical`代表child元素以垂直排列
 - 值為`horizontal`代表child元素以水平排列(預設值)
 - `android:layout_width` 屬性決定元素寬度樣式
 - `android:layout_height` 屬性決定元素高度樣式
 - `android:text` 屬性決定顯示的字串
 - ...





android:id屬性

- 為了讓程式可以存取控制項(input controls), 必須透過android:id屬性為控制項指定識別碼(ID)與名稱。
- 例如:
 - `android:id="@+id/自訂名稱"`
- 記住
 - 只要在程式中必須取得或設定此控制項時, 就一定要加上 `android:id` 屬性。
 - 「自訂名稱」在程式設計過程扮演相當重要的角色, **命名時一定要有意義**。



Layout(或其他XML檔)中引用資源

- 引用方式
 - @資源型態 / 資源名稱
- 舉例來說
 - 引用在res/values/strings.xml 內名為app_name的字串資源
 - `android:text="@string/app_name"`
 - 引用在res/drawable-XXXX 內名為ic_launcher.png的圖檔資源
 - `android:src="@drawable/ic_launcher.png"`



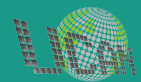
課堂練習

請產生一個hellocalc專案並利用Layout產生如下圖所示的UI。Input Control上的字串必須引用res/values/strings.xml

提示：
無

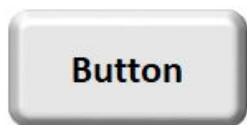


Android程式實戰



事件驅動程式設計模型

這種模型的程式執行流程是由使用者的動作 (如滑鼠的按鍵, 鍵盤的按鍵動作) 或者是由其他程式的訊息來決定的。(來源: wikipedia)



當點擊(click)或觸碰(touch)
等事件發生時, 呼叫對應的事
件處理函式

Event Listeners (事件處理函式)

```
onClick() {  
    // 撰寫點擊事件的處理程式  
    . . .  
}
```

```
onTouch() {  
    // 撰寫觸碰事件的處理程式  
    . . .  
}
```

Hello, Calculator (需求)

- 輸入
 - 兩個整數
- 輸出
 - 使用者按下「兩數相加」按鈕，顯示兩數相加結果。
 - 使用者按下「兩數相減」按鈕，顯示兩數相減結果。



The screenshot shows a mobile application interface for a calculator. At the top, there is a status bar with icons for 3G, signal strength, and battery, along with the time 10:38 AM. Below the status bar is a title bar that says "Hello, Calculator". The main interface has two input fields labeled "數字一" (Number 1) and "數字二" (Number 2). The "數字一" field is currently empty and has an orange border. Below these fields are two buttons: "兩數相加" (Add two numbers) and "兩數相減" (Subtract two numbers). Above the buttons, there are labels "結果" (Result) and "答案" (Answer).

Layout檔案(res/layout/activity_main.xml)

```
<TextView
    android:id="@+id/tv_show_result"
    ... />
<EditText
    android:id="@+id/et_no1"
    ... />
<EditText
    android:id="@+id/et_no2"
    ... />
<Button
    android:id="@+id/bt_add"
    ... />
```





Java原始檔案(src/MainActivity.java)

- 宣告會使用到的 Input Control 物件

```
Button mAdd; //加法按鈕
```

```
EditText mNo1; //第1個輸入數字
```

```
EditText mNo2; //第2個輸入數字
```

```
TextView mResult; //顯示結果
```

- 物件實體化

```
mAdd = (Button) findViewById(R.id.bt_add);
```

```
mNo1 = (EditText) findViewById(R.id.et_no1);
```

```
mNo2 = (EditText) findViewById(R.id.et_no2);
```

```
mResult = (TextView) findViewById(R.id.tv_show_result);
```

- 如何設定點擊事件處理函式？

```
public void setOnClickListener(View.OnClickListener I)
```





Java原始檔案(src/MainActivity.java)

- 設定事件處理函式

```
mAdd.setOnClickListener(new OnClickListener() {  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        int no1 = Integer.valueOf(mNo1.getText().toString());  
        int no2 = Integer.valueOf(mNo2.getText().toString());  
        int result = no1 + no2;  
        mReuslt.setText(String.valueOf(result));  
    }  
});
```





課堂作業

請完成可以執行**加、減、乘、除**的HelloCalc專案。

驗收方式

課堂直接驗收

提示：

無

HelloTouchCalc (需求)

- 輸入
 - 兩個整數
- 輸出
 - 使用者往上滑動，顯示兩數相加結果。
 - 使用者往下滑動，顯示兩數相減結果。
 - 使用者往右滑動，顯示兩數相乘結果。
 - 使用者往左滑動，顯示兩數相除結果。

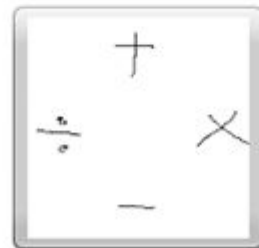


數字一

數字二

結果

答案



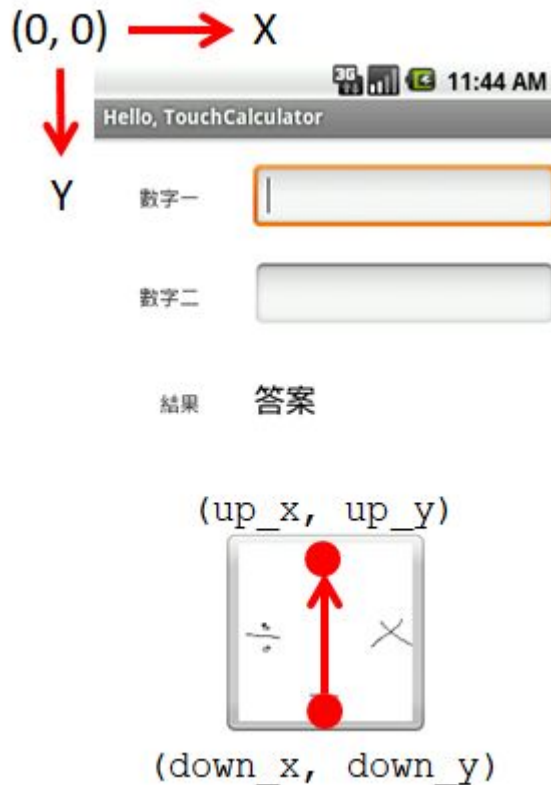


HelloTouchCalc (程式邏輯)

- 使用者在ib_calc上滑動(使用ImageButton)
- 呼叫ib_calc的onTouch()事件處理函式
- 根據滑動方向將et_no1與et_no2的值進行
 - 加(往上滑動)
 - 減(往下滑動)
 - 乘(往右滑動)
 - 除(往左滑動)
- 將運算結果顯示在tv_show_result上

Hello, TouchCalc (程式邏輯)(續)

- 如何判斷使用者滑動方向?
- 以往上滑動為例, 判斷條件?
 - $up_y < down_y$
- 可能誤判的情況?
 - 往右上或左上滑
 - 提高準確度的方法之一
 - $down_y - up_y > Math.abs(down_x - up_x)$
- 程式碼說明
 - `float Math.abs(float value)` 為取絕對值





Java原始檔案(src/MainActivity.java)


```
mCalc.setOnTouchListener(new OnTouchListener() {  
    public boolean onTouch(View v, MotionEvent event) {  
        switch(event.getAction()){  
            case MotionEvent.ACTION_DOWN: //當手指觸控ImageButton時，紀錄按下的座標  
                down_x = event.getX();  
                down_y = event.getY();  
                break;  
        }  
        return false;  
    }  
});
```





Java原始檔案(src/MainActivity.java)

```
mCalc.setOnClickListener(new OnClickListener() {  
    public boolean onTouch(View v, MotionEvent event) {  
        switch(event.getAction()){  
            ...  
            case MotionEvent.ACTION_UP:    /當手指離開ImageButton時，紀錄離開的座標  
                up_x = event.getX();  
                up_y = event.getY();  
                if(down_y - up_y > Math.abs(down_x - up_x)) { // add  
                    int no1 = Integer.valueOf(mNo1.getText().toString());  
                    int no2 = Integer.valueOf(mNo2.getText().toString());  
                    mReuslt.setText(String.valueOf(no1 + no2));  
                }  
                break;  
            }  
        return false;  
    }  
});
```





課堂作業

請完成可以執行**加、減、乘、除**的HelloTouchCalc專案。

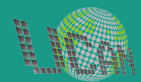
驗收方式

課堂直接驗收

提示：

無

新增並啟動Activity



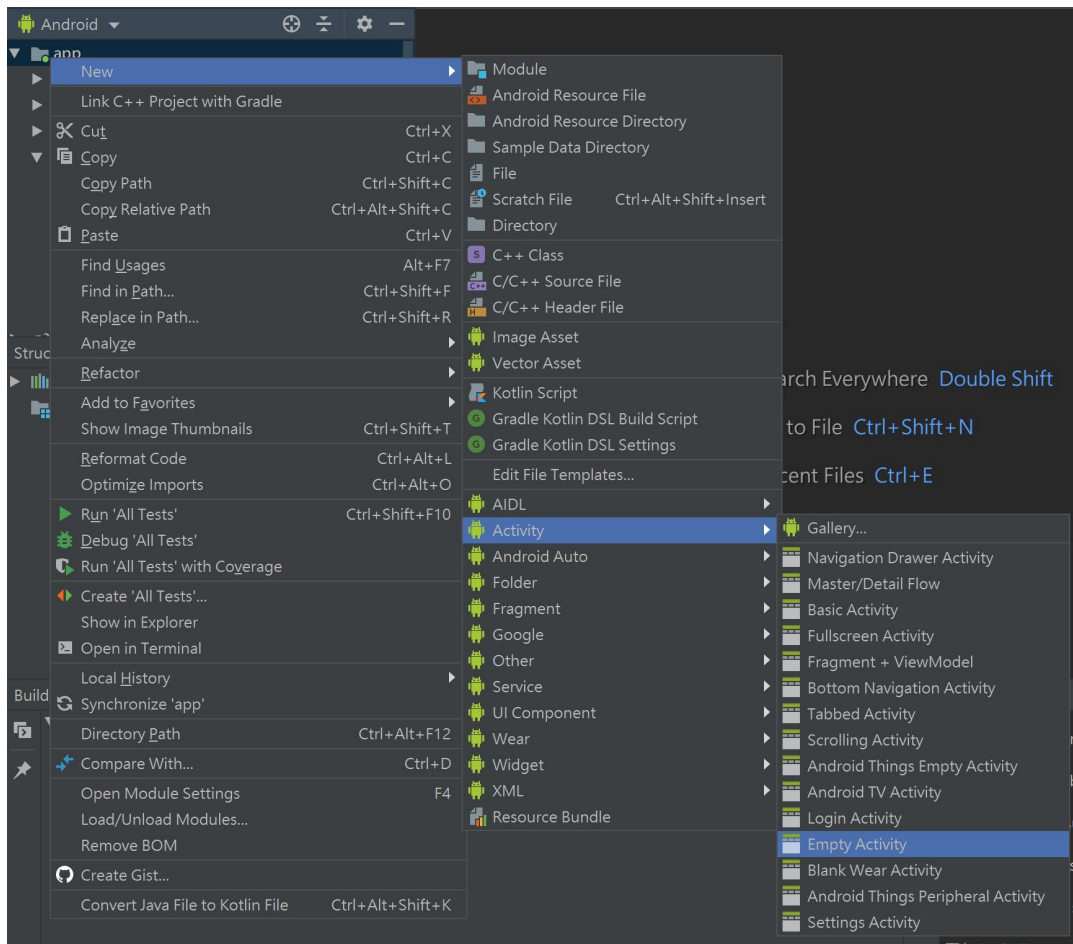


如何新增並啟動Activity?

- 新增Activity的步驟
 - 新增Activity
 - 新增對應的layout (Android Studio自動完成)
 - 在AndroidManifest.xml檔中宣告Activity應用程式元件 (Android Studio自動完成)
- 啟動Activity的步驟
 - 準備Intent
 - 呼叫Activity的啟動函式

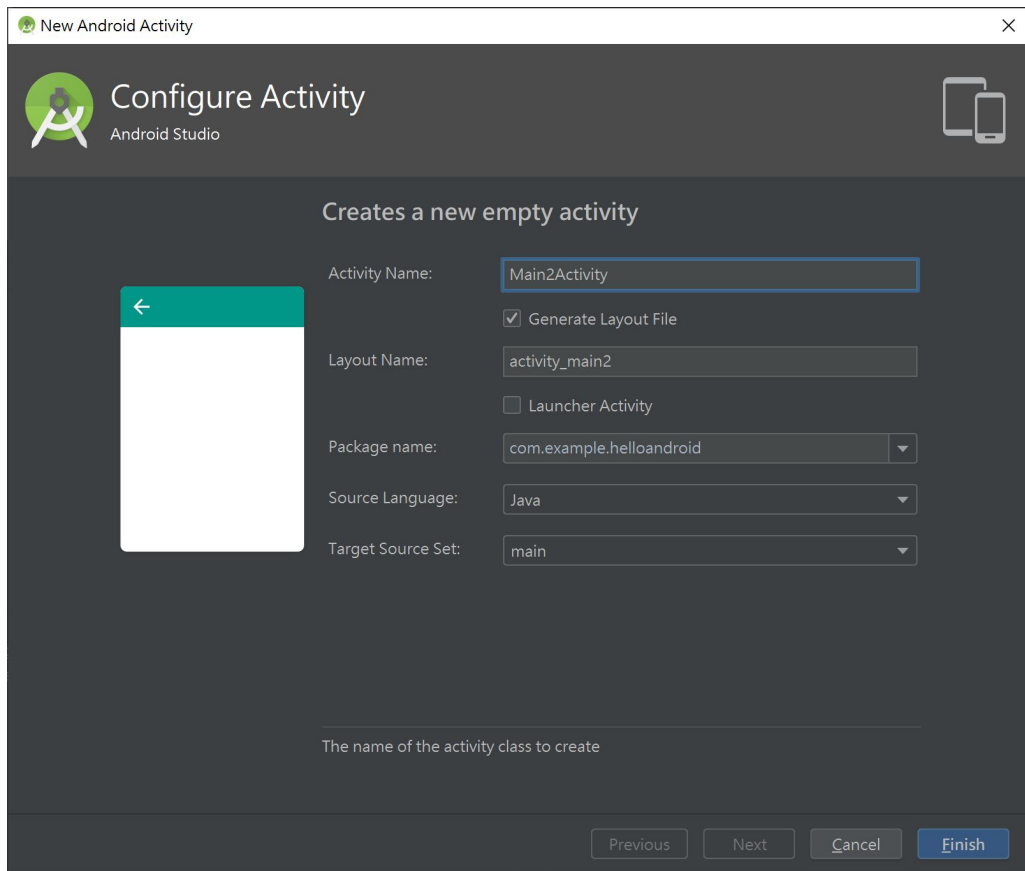
新增Activity

在專案上按滑鼠右鍵並選擇新增Activity



新增Activity (續)

不同的應用程式元件(與其類別)需要輸入不同的資訊



New Android Activity

Configure Activity
Android Studio

Creates a new empty activity

Activity Name:

☒ Generate Layout File

Layout Name:

☐ Launcher Activity

Package name:

Source Language:

Target Source Set:

The name of the activity class to create

Previous Next Cancel Finish



新增Activity專案的變化

1. 在manifests/AndroidManifest.xml中新增了一個標籤
`<activity android:name=".Main2Activity"></activity>`
2. 在src目錄中新增了一個java原始檔
`src/[package]/Main2Activity.java`
3. 在res目錄中新增了一個Layout檔案
`res/layout/activity_main2.xml`



練習: MainActivity啟動Main2Activity

在MainActivity.java中宣告一個Intent物件

```
Intent intent = new Intent();  
intent.setClass(MainActivity.this, Main2Activity.class);
```

送出Intent讓Android啟動Main2Activity

```
startActivity(intent);
```





如何透過Intent傳遞資料?

在MainActivity.java中放入資料

```
intent.putExtra("key", value); // 不同資料型態會使用不同函式
```

在Main2Activity.java中取出資料

```
int vluce = this getIntent().get IntExtra("key", defaultValue);
```

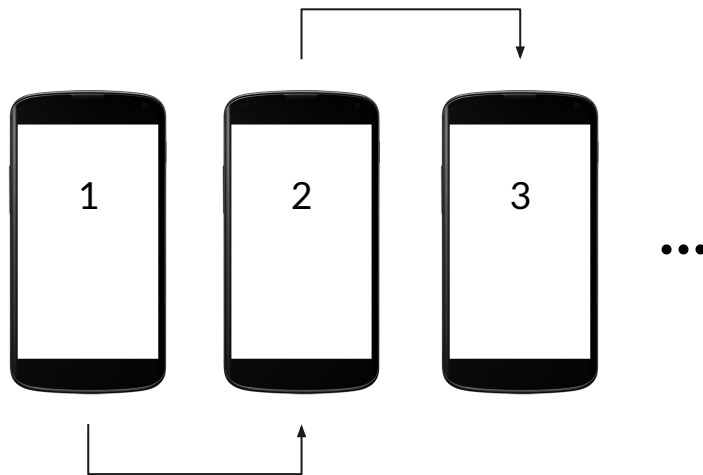


課堂練習

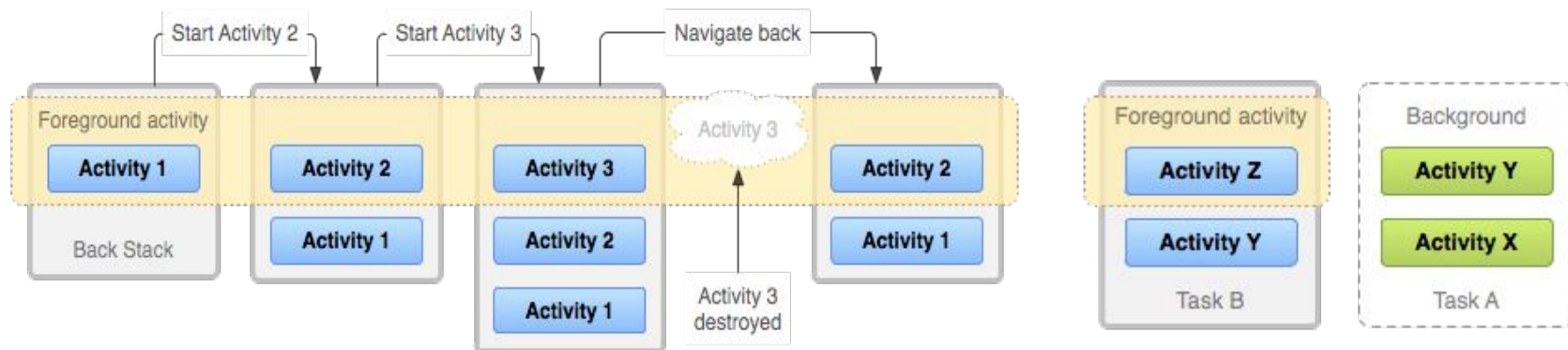
新增並啟動Activity

請完成具有兩個 Activities 的App。

- 第1個Activity上有一個按鈕，按下後會啟動第2個Activity。
- 第2個Activity上有一個按鈕，按下後會啟動第1個Activity。
- 設定一個count從1開始，每次啟動Activity時加1並傳遞給下一個Activity並顯示出來。



Activity Stack



Android Activity Stack (來源: *Android Developers*)

補充說明

除了按下Back鍵外，也可以呼叫 `finish()` 函式可以將Activity結束。



課堂作業

Activity Stack

驗收方式

課堂直接驗收

提示:

無

請完成具有兩個 Activities 的 App。

- 第1個Activity上有一個按鈕，按下後會啟動第2個Activity。
- 第2個Activity上有一個按鈕，按下後會啟動第1個Activity。
- 在Activity Stack上第1個Activity和第2個Activity不能同時存在。

Q & A



Computer History Museum, Mt. View, CA

