

# Android App 專案管理與 Gradle 設定

## ■ 複製 App 專案

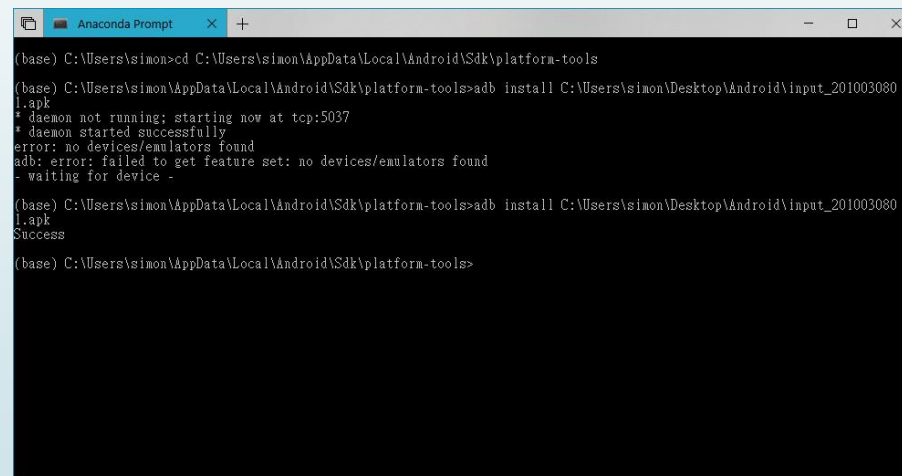
- App 專案中除了程式檔、資源檔之外，還包含許多編譯過程產生的暫存檔與開發環境的暫存檔。
- 複製專案到其他地方時，暫存檔不需要複製。
- 複製專案時，下列項目可不需複製：
  - 程式專案資料夾中的『.gradle』、『.idea』、『build』資料夾。
  - app 資料夾中的『build』子資料夾。
  - 程式專案資料夾中的『.iml』檔案，此檔案的主檔名與專案資料夾名稱相同。

# Android App 設計

- 於模擬器輸入中文：
  - 下載中文安裝檔：
    - <https://code.google.com/archive/p/android-zhuyin-ime/downloads>
  - 開啟命令列，切換到 Android SDK 安裝資料夾中的『platform-tools』資料夾：
    - C:\Users\simon\AppData\Local\Android\Sdk\platform-tools
  - 執行下列安裝指令：(安裝前須開啟模擬器)
    - adb install C:\Users\simon\Desktop\Android\input\_2010030801.apk

# Android App 設計

- ➡ 於模擬器輸入中文：
  - ➡ 畫面出現『Success』即表示安裝成功。



```
(base) C:\Users\simon>cd C:\Users\simon\AppData\Local\Android\Sdk\platform-tools
(base) C:\Users\simon\AppData\Local\Android\Sdk\platform-tools>adb install C:\Users\simon\Desktop\Android\input_201003080
1.apk
* daemon not running; starting now at tcp:5037
* daemon started successfully
error: no devices/emulators found
adb: error: failed to get feature set: no devices/emulators found
- waiting for device -
(base) C:\Users\simon\AppData\Local\Android\Sdk\platform-tools>adb install C:\Users\simon\Desktop\Android\input_201003080
1.apk
Success
(base) C:\Users\simon\AppData\Local\Android\Sdk\platform-tools>
```

# Android App 設計

模擬器中文輸入法設定



模擬器中文輸入法設定



# Android App 設計

## 模擬器中文輸入法設定



## 模擬器中文輸入法設定

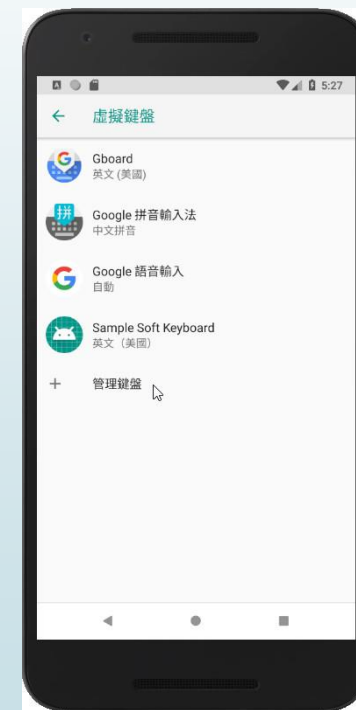


# Android App 設計

## 模擬器中文輸入法設定

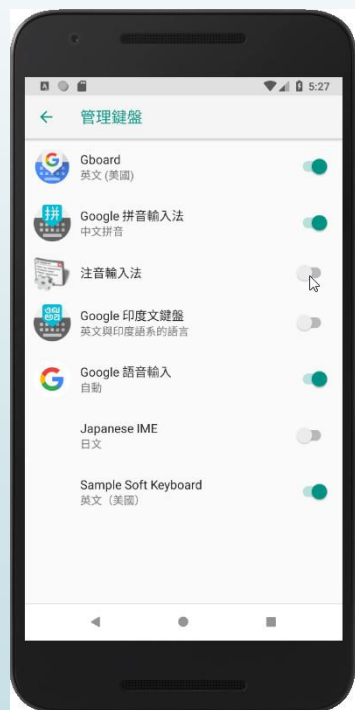


## 模擬器中文輸入法設定



# Android App 設計

## 模擬器中文輸入法設定



## 模擬器中文輸入法設定

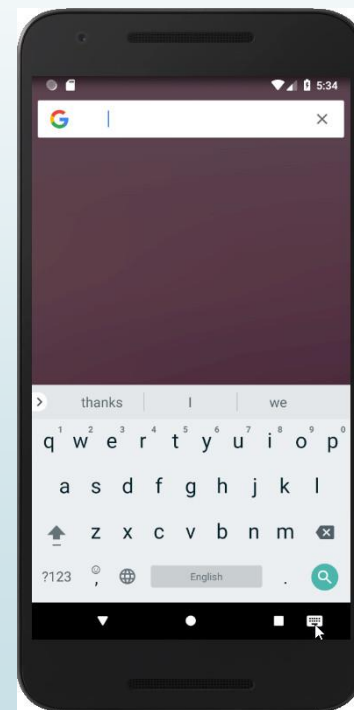


# Android App 設計

## 模擬器中文輸入法設定



## 模擬器中文輸入法設定





# Android App 設計

## 模擬器中文輸入法設定



## 模擬器中文輸入法設定



# Android App 設計

## ➤ MVC 模式：

- Model-View-Controller。
- 軟體工程中的軟體架構模式。
- 將軟體系統分為三個單元：
  - 模型 (Model)。
  - 檢視 (View)。
  - 控制器 (Controller)。

## ➤ 目的：

- 實作動態的程式設計。
- 使後續對程式的修改與擴充功能簡化。
- 使程式的某一部份可以重複利用。
- 使程式結構更加直覺。

# Android App 設計

## ► Android 的 MVC 模式：

- 將程式碼與資料分開。
- 模型：(Model)
  - 將資料儲存在『res/values』資料夾。
- 檢視：(View)
  - 將使用者介面儲存在『res/layout』資料夾中的 xml 檔。
- 控制器：(Controller)
  - 將程式碼 (\*.java) 儲存在『java』資料夾中。

## ► 開發方式建議：

- 先完成資料的定義 (xml 檔)、使用者畫面設計 (xml 檔)。
- 最後撰寫程式碼 (java 檔)。

# 事件處理

- 來源物件
  - 事件發生的來源
- 監聽物件
  - 要處理事件必須要有一個能處理該事件的監聽物件 (Listener)。
  - 再將監聽物件 (Listener) 登錄到來源物件中。
  - 當來源物件有事件發生時，會自動呼叫監聽物件中對應該事件的處理方法進行處理。

# 事件處理

## ► Java 的介面 (Interface)

- 要成為特定事件의監聽物件，必須符合該事件的規範。
- Android 以 Java 的介面 (Interface) 來規範處理事件的方法。
- 如按一下 (onClick) 事件，對應的規範就是 OnClickListener 介面。
- 監聽物件類別一開始就必須宣告要成為哪個事件的監聽物件，方能登錄為監聽物件。此宣告動作稱為【實作(implements) \*\*\* 介面】。

# 事件處理

## ► Java 的介面 (Interface)

- 如要讓 MainActivity 類別物件成為『按一下』(onClick) 事件의監聽物件，在 Java 程式中定義 MainActivity 類別時就必須以下列方式撰寫：

- ```
public class MainActivity extends Activity implements OnClickListener { //實作
    OnClickListener 介面
    .....
    public void onClick(View v) {
    .....          //onClick() 為 OnClickListener 介面中規範必須撰寫的方法
    }
}
```

- 監聽物件準備好後，就可登錄到來源物件中。

- 如 Button 物件就定義有『setOnClickListener()』方法，可用來登錄『按一下』事件의監聽物件。

# 基本元件設計

- ▶ TextView (標籤)
  - ▶ 顯示文字訊息。
  - ▶ Xml 語法：
    - ▶ **<TextView**  
**android:id= "@+id/元件名稱 "**  
**android:屬性 = " 屬性值"**  
**...其他屬性 />**

# 基本元件設計

## ► TextView (標籤)

### ► 常用屬性：

| 屬性 (Attribute) | Xml 程式碼                      | 功能說明           |
|----------------|------------------------------|----------------|
| layout_height  | <b>android:layout_height</b> | 設定文字方塊的高度。     |
| layout_width   | <b>android:layout_width</b>  | 設定文字方塊的寬度。     |
| text           | <b>android:text</b>          | 設定文字方塊的文字內容。   |
| id             | <b>android:id</b>            | 設定文字方塊的 id 名稱。 |
| textColor      | <b>android:textColor</b>     | 設定文字方塊的色彩。     |
| textSize       | <b>android:textSize</b>      | 設定文字方塊的文字大小。   |



# 基本元件設計

- EditText (Plain Text) (文字方塊)
  - 提供使用者輸入文字。
  - Xml 語法：
    - **<EditText**  
**android:id= "@+id/元件名稱 "**  
**android:屬性 = " 屬性值"**  
**...其他屬性 />**

# 基本元件設計

## ▀ EditText (Plain Text) (文字方塊)

### ▀ 常用屬性：

| 屬性 (Attribute) | Xml 程式碼                      | 功能說明           |
|----------------|------------------------------|----------------|
| layout_height  | <b>android:layout_height</b> | 設定文字方塊的高度。     |
| layout_width   | <b>android:layout_width</b>  | 設定文字方塊的寬度。     |
| text           | <b>android:text</b>          | 設定文字方塊的文字內容。   |
| id             | <b>android:id</b>            | 設定文字方塊的 id 名稱。 |
| textColor      | <b>android:textColor</b>     | 設定文字方塊的色彩。     |
| textSize       | <b>android:textSize</b>      | 設定文字方塊的文字大小。   |

# 基本元件設計

## ► EditText (Plain Text) (文字方塊)

► 提供使用者輸入文字。

► 常用屬性：

| 屬性 (Attribute) | Xml 程式碼                   | 功能說明                                           |
|----------------|---------------------------|------------------------------------------------|
| lines          | <b>android:lines</b>      | 設定文字方塊顯示的列數。                                   |
| ems            | <b>android:ems</b>        | 設定文字方塊的每列字元數，必須 width 屬性設定為 wrap_content 才有作用。 |
| editable       | <b>android:editable</b>   | 設定是否可以輸入文字。                                    |
| enabled        | <b>android:enabled</b>    | 設定文字方塊是否有作用。                                   |
| singleLine     | <b>android:singleLine</b> | 設定文字方塊是否輸入單列。                                  |
| inputType      | <b>android:inputType</b>  | 設定文字方塊的輸入資料類型。                                 |

# 基本元件設計

- Button (按鈕)
  - 透過設定的監聽事件，執行指定的工作。
  - Xml 語法：
    - **<Button**  
**android:id= "@+id/元件名稱 "**  
**android:屬性 = " 屬性值"**  
**...其他屬性 />**

# 基本元件設計

## ► Button (按鈕)

### ► 常用屬性：

| 屬性 (Attribute) | Xml 程式碼                      | 功能說明         |
|----------------|------------------------------|--------------|
| layout_height  | <b>android:layout_height</b> | 設定按鈕的高度。     |
| layout_width   | <b>android:layout_width</b>  | 設定按鈕的寬度。     |
| text           | <b>android:text</b>          | 設定按鈕顯示的文字。   |
| id             | <b>android:id</b>            | 設定按鈕的 id 名稱。 |
| textColor      | <b>android:textColor</b>     | 設定按鈕的顯示文字色彩。 |
| textSize       | <b>android:textSize</b>      | 設定按鈕的顯示文字大小。 |
| background     | <b>android:background</b>    | 設定按鈕的色彩。     |

# 基本元件設計

## ■ Button (按鈕)

### ■ 觸發事件的程序：

- 於程式中藉由 id 取得需要使用的按鈕元件。
- 於該按鈕元件設定要『監聽的動作』及觸發時要執行的『方法名稱』。
- 撰寫執行的方法程式內容。

# 基本元件設計

## ➤ Button (按鈕)

### ➤ 加入 Button 元件監聽：

➤ **Button 元件變數.setOnClickListener(監聽方法名稱)；**

### ➤ 加入觸發 Button 元件的監聽方法：

```
private Button.OnClickListener 監聽方法名稱 = new Button.OnClickListener(){  
    @Override  
    public void onClick(View v){  
        方法的程式內容...  
        ...  
    }  
};
```

# 基本元件設計

## ► Button (按鈕)

► 撰寫 Button 按下要執行的方法，如：

```
► public void 方法名稱 (View view){  
    方法的程式內容...  
    ...  
}
```

► 將方法名稱設定至 Button元件的『onClick』屬性。



# 基本元件設計

- Spinner (下拉式選單)
  - 建立選項清單，供使用者選取，可避免輸入文字時的錯誤。
  - Xml 語法：
    - **<Spinner**  
**android:id= "@+id/元件名稱 "**  
**android:屬性 = " 屬性值"**  
**...其他屬性 />**

# 基本元件設計

## ► Spinner (下拉式選單)

### ► 建立 Spinner 元件的程序：

- 於 strings.xml 檔中建立一個字串陣列。
- 將字串陣列設定給 Spinner 元件作為選單。
- 於介面佈局檔中加入 Spinner 元件，並設定其屬性。
- 於主程式檔中為 Spinner 元件加入『OnItemSelectedListener()』事件處理程序：
  - 加入事件監聽。
  - 加入觸發 Spinner 元件的監聽方法。

# 基本元件設計

## ► Spinner (下拉式選單)

### ► 加入 Spinner 元件監聽：

► Spinner 元件變數.setOnItemSelectedListener(監聽方法名稱)；

### ► 加入觸發 Spinner 元件的監聽方法：

```
private Spinner.OnItemSelectedListener 監聽方法名稱 = new Spinner.  
setOnItemSelectedListener(){  
    @Override  
    public void OnItemSelected (View v){  
        方法的程式內容...  
        ...  
    }  
};
```

# 基本元件設計

## ► RadioGroup / RadioButton (單選鈕群組 / 單選鈕)

► 建立單選選項，供使用者選取，在同一個單選鈕群組中的單選鈕只能有一個被選。

► Xml 語法：

```
► <RadioGroup
    android:id= "@+id/元件名稱 "
    android:屬性 = " 屬性值"
    ...其他屬性 >
    <RadioButton
        android:id= "@+id/元件名稱 "
        android:屬性 = " 屬性值"
        ...其他屬性 />
    ...
</RadioGroup>
```

# 基本元件設計

- RadioGroup / RadioButton (單選鈕群組 / 單選鈕)
  - 建立 RadioGroup / RadioButton 元件的程序：
    - 於介面佈局檔中加入 RadioGroup 元件，並設定其屬性。
    - 於介面佈局檔中加入 RadioButton 元件，放在 RadioGroup 元件層次下，並設定其屬性。
    - 於主程式檔中為 RadioGroup 元件加入『OnCheckedChangeListener』事件處理程序。
      - 加入事件監聽。
      - 加入觸發 RadioGroup 元件的監聽方法。

# 基本元件設計

- RadioGroup / RadioButton (單選鈕群組 / 單選鈕)
  - 加入 RadioGroup 元件監聽：
    - **RadioGroup 元件變數.setOnCheckedChangeListener(監聽方法名稱) ;**
  - 加入觸發 RadioGroup 元件的監聽方法：
    - **private RadioGroup.OnCheckedChangeListener 監聽方法名稱 = new RadioGroup.  
OnCheckedChangeListener(){  
    @Override  
    public void OnCheckedChangeListener (View v){  
        方法的程式內容...  
        ...  
    }  
};**