

# Fragment 類別

## ■ 功用

- 架構與 Activity 類似，有專屬的介面佈局檔與程式檔。
- 將 App 畫面分割為數個區域。
- 各區域的程式畫面可以各自隱藏或顯示，以適應不同螢幕尺寸的裝置。

## ■ 特性

- 程式的執行畫面可以由多個 Fragment 組成。
- 每一個 Fragment 都各自有獨立的執行狀態，並且接收各自的處理事件。
- 程式的執行過程中，可以動態加入或移除。

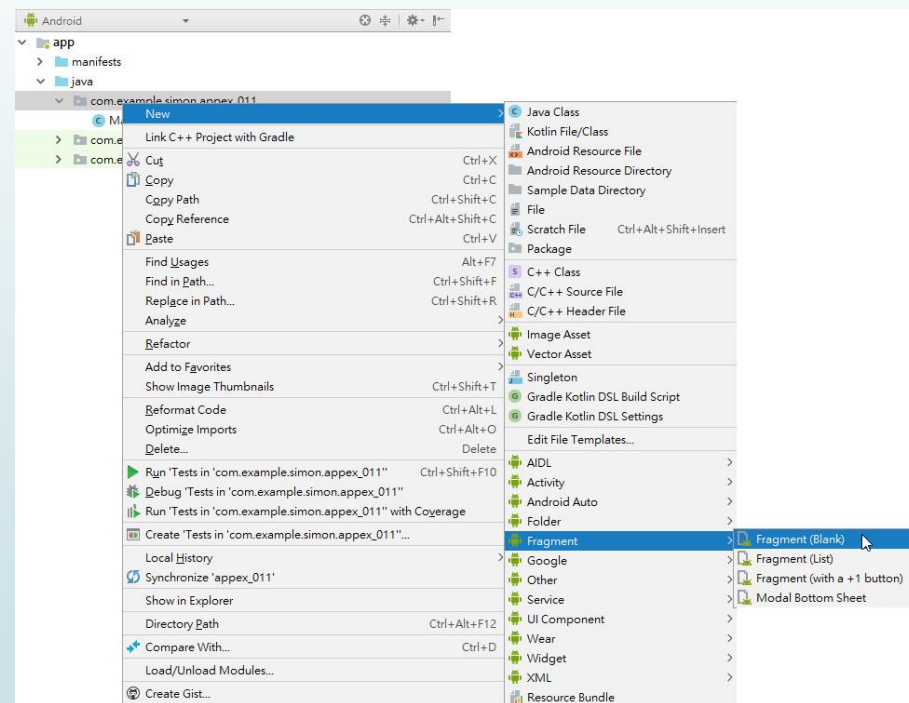
# Fragment 類別

## ■ 版本

- Fragment 是 Android 3.0 後的功能，故 Android 3.0 以前的裝置無法執行具有 Fragment 的程式。
- Google 開發了『android-support-v4.jar』程式庫，載入專案中即可讓舊版以前的裝置使用 Fragment。
- 『android-support-v4.jar』的 Fragment 與 Android 系統內建的 Fragment 不可同時使用，必須擇一使用。

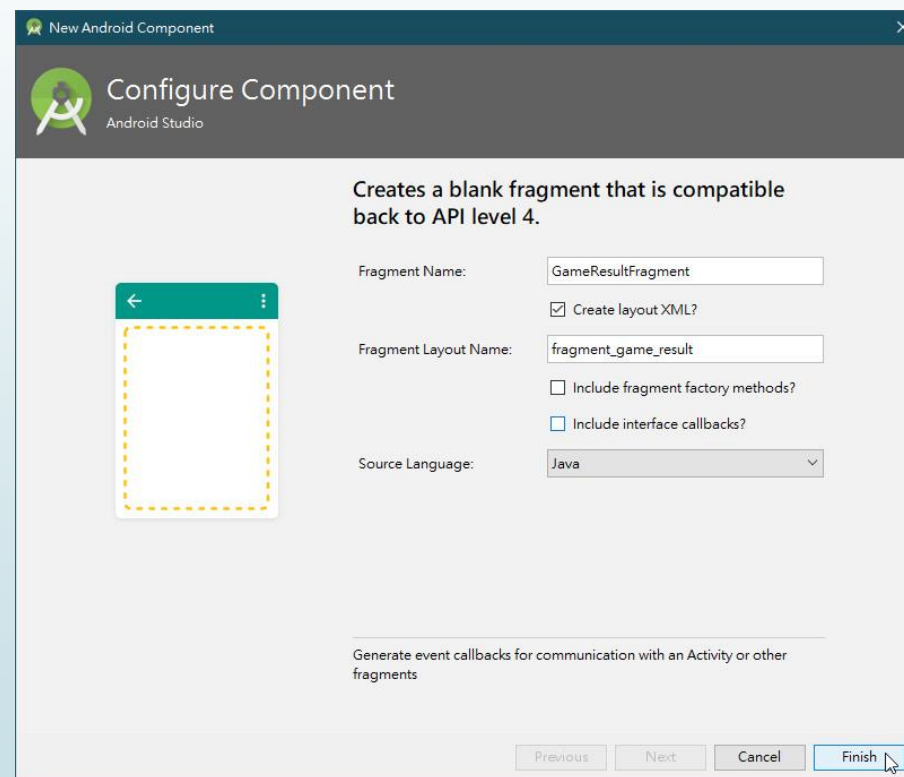
# Fragment 類別

- 使用 Fragment 的步驟
  - 新增 Fragment 檔案：
    - 展開『app>java』資料夾。
    - 於 Java 檔上按右鍵。
    - New>Fragment> Fragment(Blank)



# Fragment 類別

- 使用 Fragment 的步驟
  - 新增 Fragment 檔案：
    - 輸入『Fragment Name』。
    - 取消勾選『include fragment factory methods』與『include interface callbacks』。
    - 按下『finish』。



# Fragment 類別

## ■ 使用 Fragment 的步驟

### ■ 新增 Fragment 檔案：

- 新增的 Fragment 程式檔會自動開啟，並自動產生對應的介面佈局檔(\*.XML)。

### ■ 於Fragment 程式檔中加入需要用到的狀態處理方法：

#### ■ onCreate()：

- 當 Fragment 被建立時會執行的方法，可在此方法中建立必要的初始設定。

#### ■ onCreateView()：

- 當 Fragment 將要顯示在螢幕上時執行的方法，必須在此方法中設定 Fragment 使用的介面佈局檔。

# Fragment 類別

## ■ 使用 Fragment 的步驟

### ■ 於Fragment 程式檔中加入需要用到的狀態處理方法：

#### ■ `onActivityCreated()`：

- 當 Fragment 底層的 Activity 被建立時會執行的方法，必須在此方法中取得 Fragment 的介面元件，並設定給對應的介面物件(如同 Activity 中的 `onCreate()` 方法中的設定)。

#### ■ `onPause()`：

- 當 Fragment 要從螢幕上消失時會執行的方法，可在此方法中儲存使用者的操作狀態和輸入的資料，以便下次 Fragment 重新顯示時，讓使用者繼續進行目前的工作。

### ■ 在主程式的主介面佈局檔中將要使用的 Fragment 加入：

- 以 `<fragment>` 標籤加入。

# Fragment 類別

## Fragment 標籤

### Xml 語法：

- **<fragment**  
    **android:id= "@+id/fragment 名稱 "**  
    **android:屬性 = " 屬性值"**  
    **...其他屬性 />**
- 標籤開頭字母要小寫。
- 每一個<fragment>標籤都要設定『android:id』屬性。
- 『android:name』屬性為指定所使用的 Fragment 類別，必須加上完整的套件路徑名稱。

# 訊息顯示元件

## ► Toast (快顯訊息)

- 顯示訊息，短時間後自動消失。
- 無法與使用者互動，單純顯示訊息。
- 語法：**Toast 變數名稱 = Toast.makeText(主程式類別.this, 顯示的訊息, 顯示時間)**
  - 顯示時間：
    - Toast.LENGTH\_LONG
    - Toast.LENGTH\_SHORT



# 訊息顯示元件

## ► Toast (快顯訊息)

### ► Toast 元件位置：

- 語法：**Toast 變數名稱.setGravity(位置參數, 水平方向偏移量, 垂直方向偏移量)**
- 位置參數：(可同時設定多個，以『|』符號隔開)
  - Gravity.CENTER：水平 / 垂直置中。
  - Gravity.CENTER\_HORIZONTAL：水平置中。
  - Gravity.CENTER\_VERTICAL：垂直置中。
  - Gravity.RIGHT：靠螢幕右邊緣。
  - Gravity.LEFT：靠螢幕左邊緣。
  - Gravity.TOP：靠螢幕上邊緣。
  - Gravity.BOTTOM：靠螢幕下邊緣。

# 訊息顯示元件

## ➤ Toast (快顯訊息)

### ➤ Toast 元件位置：

➤ 語法：**Toast 變數名稱.setGravity(位置參數, 水平方向偏移量, 垂直方向偏移量)**

### ➤ 水平方向偏移量：

➤ 正值：向右偏移。(單位：點)

➤ 負值：向左偏移。(單位：點)

### ➤ 垂直方向偏移量：

➤ 正值：向下偏移。(單位：點)

➤ 負值：向上偏移。(單位：點)