

# Android Tutorial using Kotlin 第二堂

## （1）規劃與建立應用程式需要的資源

[Android Tutorial using Kotlin 第一堂（4）開發Android應用程式的準備工作 << 前情](#)

第二堂的課程內容進入應用程式開發的階段，從規劃與建立應用程式的資源開始，接下來會說明Android最基本的Activity元件有設計應用程式需要的畫面，最後幫應用程式加入讓使用者操作與互動的功能。

Android行動裝置在全世界已經是非常普及的隨身電子設備，Android開發人員把應用程式上傳到Google Play以後，全世界的使用者可以隨時透過網際網路下載與安裝，在這樣的情況下，應用程式就會有多國語言的需求了。應用程式通常會有各種需要顯示文字的面元件，例如TextView和Button，開發人員會希望這些文字可以使用裝置的語言來顯示。以提供確定功能的按鈕元件來說，如果一個應用程式安裝在繁體中文的裝置，按鈕上的文字應該顯示「確定」，如果安裝在英文的裝置，就應該顯示「OK」。

開發Android應用程式應該要先認識資源的規劃與管理，建立好一點的設計習慣，對於應用程式的開發工作會有很大的幫助。你在後續的學習可以得到最好的效果，這一堂課需要建立一個Android應用程式專案，隨著課程的進行，會依照課程內容依序完成完整的應用程式。參考下列的步驟建立一個Android應用程式專案：

1. 啟動Android Studio，選擇「Start a New Android Studio project」。
2. 在「Configure your new project」視窗，依照下列的說明輸入需要的內容後，選擇「Next」：
  - Application Name輸入「ATK」。
  - Company Domain輸入「macdidi.net」，或是你自己的網域名稱。專案的主套件名稱（Package name）自動產生「net.macdidi.myandroidtutorial」。
  - Project location選擇一個儲存專案的位置。
  - 勾選「Include Kotlin support」。
3. 勾選「Phone and Tablet」，Minimum SDK選擇「API 16:Android 4.1(Jelly Bean)」，選擇「Next」。
4. 在「Add an activity to Mobile」視窗選擇「Empty Activity」後選擇「Next」。
5. 在「Create a new empty activity」視窗採用預設的名稱與設定，選擇「Finish」完成建立應用程式的設定。

等候Android Studio啟動新的應用程式視窗，就完成建立Android應用程式專案的工作。這個專案會從這裡開始，一直持續使用。每一段說明的內容，都會在這個專案加入或修改一些內容。

### 5-1 Android應用程式資源介紹

一個應用程式通常會包含許多畫面，每一個畫面中又會包含許多顯示各種資訊的元件。應用程式通常也會提供使用者操作與互動功能，這些工作需要撰寫一些程式碼來執行。如果是一個比較複雜的應用程式，就會包含很多畫面、程式碼和各種需要的內容。

為了簡化應用程式的開發工作，Android平台採用目前比較普遍、也比較好的設計方式，把應用程式需要的程式碼和其它需要分開，開發人員在設計畫面的時候，依照Android的設計規格提供一些「畫面資源」，這些資源是XML格式的文件。應用程式用的文字、陣列、顏色，也都會把它們設計為XML格式的資源檔案。應用程式的程式碼就只剩下基本與流程的工作，需要畫面的時候，就從XML格式的資源檔案讀進來使用。

### 5-1-1 應用程式專案的資源目錄

建立一個Android應用程式專案以後，Android Studio會自動建立一些需要的專案目錄，「res」的目錄用來存放應用程式需要資源，不同的資源要放在規定的目錄下，例如畫面配置檔資源會放在「res/layout」。Android規定所有資源的種類和它們存放目錄，你一定要把資源放在對應的目錄下。這些是Android應用程式專案在「res」目錄下的各種資源目錄：

- anim – 動畫資源，XML格式檔案，檔案名稱就是資源名稱。
- color – 顏色狀態資源，XML格式檔案，檔案名稱就是資源名稱。
- drawable – 圖形與繪圖資源，圖形檔案與XML格式檔案，檔案名稱就是資源名稱。
- layout – 畫面配置資源，XML格式檔案，檔案名稱就是資源名稱。
- menu – 選單資源，XML格式檔案，檔案名稱就是資源名稱。
- raw – 檔案資源，任何應用程式需要的檔案，檔案名稱就是資源名稱。
- values – 一般資源，包含文字、顏色、尺寸、陣列與樣式資源，XML格式檔案，每一個資源的名稱在標籤的設定中決定。

### 5-1-2 Android系統資源

Android系統已經內建許多資源，都是一些比較一般性與常用的資源，你的應用程式應該要優先使用系統提供的資源，如果沒再自己建立需要的資源，這樣可以減少應用程式的大小，而且也比較方便一些。

Android內建的資源都宣告在API的「android.R」套件中，不論是系統或自己建立的資源，它們的分類方式都是一樣的，會依!的種類使用「android.R.資源種類」的格式，例如系統內建的畫面配置資源就放在「android.R.layout」套件。這些是宣告在「android.R」套件中常用的系統資源：

- android.R.anim – 系統動畫資源。
- android.R.color – 系統顏色狀態資源。
- android.R.dimen – 系統尺寸資源。
- android.R.drawable – 系統圖形與繪圖資源。
- android.R.layout – 系統畫面配置資源。
- android.R.menu – 系統選單資源。
- android.R.string – 系統文字資源。
- android.R.style – 系統樣式資源。

在建立應用程式需要的資源之前，應該要先認識系統內建的資源，在Android開發人員網站，提供和Java API文件同樣格式的地方方便開發人員查詢Android平台提供的API，網址是<http://developer.android.com/reference/packages.html>。選擇在「android.R」類別後，就可以看到所有系統內建的資源。例如選擇「R.string」以後，可以看到宣告在這個類別裡面的文字資源，一般應用程式常用的確定、取消文字，變數的名稱是「ok」與「cancel」。

## 5-2 一般資源

Android把應用程式比較常用的一般資源放在「res/values」目錄，這個目錄可以建立需要的XML格式檔案，在檔案中使用規範建立需要的資源，包含文字、顏色、尺寸、陣列和樣式資源。XML檔案名稱只是用來規劃與分類不同的資源，你可以把所有資源都放在同一個檔案，不過一般會習慣使用這樣的規劃方式，在res/values目錄建立這些檔案儲存不同種類的一般資源：

- strings.xml – 文字資源。
- colors.xml – 顏色資源。
- dimens.xml – 尺寸資源。
- arrays.xml – 陣列資源。
- styles.xml – 樣式資源。

在「res/values」目錄的XML檔案，使用規定的格式建立一般資源，最外層使用「resources」標籤，裡面包含一般資源的標籤

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="文字資源名稱">文字資源</string>
    <color name="顏色資源名稱">顏色資源</color>
    <dimen name="尺寸資源名稱">尺寸資源</dimen>
    <array name="一般陣列資源名稱">
        <item>一個陣列資源元素</item>
        ...
    </array>
    <string-array name="文字陣列資源名稱">
        <item>一個陣列資源元素</item>
        ...
    </string-array>
    <integer-array name="數字陣列資源名稱">
        <item>一個陣列資源元素</item>
        ...
    </integer-array>
    <style name="樣式">
        <item name="設定名稱">設定值</item>
        ...
    </style>
</resources>
```

### 5-2-1 建立文字資源

Android應用程式需要的文字，都不應該直接寫在程式碼或設定檔裡面，而是建立好文字資源以後再使用它們。以這裡使用的說，在規劃應用程式的時候，就應該知道需要哪一些文字資源，例如這個新增記事本的畫面：

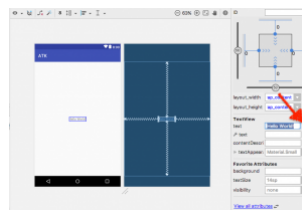


畫面中需要的文字，根據之前的說明，「確定」和「取消」使用Android系統資源就可以了。開啟Android Studio為你建立好「res/values/strings.xml」檔案，裡面已經有一個名稱為app\_name的預設文字資源，依照下列的內容加入需要的文字資源：

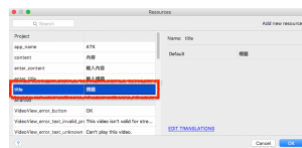
```
<resources>
    <string name="app_name">MAT</string>

    <string name="title">標題</string>
    <string name="enter_title">輸入標題</string>
    <string name="content">內容</string>
    <string name="enter_content">輸入內容</string>
</resources>
```

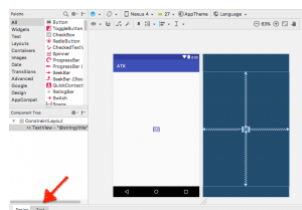
加入文字資源並儲存檔案以後，就可以在其它XML檔案（例如畫面資源）或程式碼中使用。在XML設定檔使用「@string/文字名稱」的格式設定文字資源。開啟MyAndroidTutorial專案「res/layout/activity\_main.xml」，選擇預覽畫面的文字元件，在「Properties」區塊找到「text」，選擇設定值以後，再選擇右側的設定按鈕：



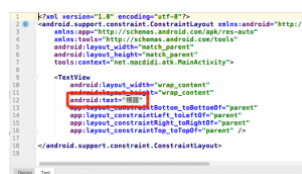
在「Resources」選擇對話框，選擇名為「title」的項目後選擇「OK」：



設定完成後可以看到預覽畫面上的文字已經變成「標題」，這是採用圖形介面的設定方式。在編輯視窗選擇「Text」標籤，檢視這個畫面資源的內容：



Android Studio顯示activity\_main.xml檔案的內容，它是一個XML格式的檔案。建議你選擇功能表「Code -> Reformat Code...」Android Studio會幫你編排內容，看起來會比較整齊一些，這個功能同樣可以適用在Java程式碼。這個畫面是編排後的內容，經顯示剛才設定好的文字資源：



使用資源的部份是直接填入資源的名稱，例如畫面中的「android:text="@string/title"」，如果想要Android Studio直接顯示資源的內容。你可以把滑鼠游標移到這個設定上面，使用快速鍵「Ctrl + .」切換顯示的內容：

```

1 android.support.constraint.ConstraintLayout TextView
2
3 <?xml version="1.0" encoding="utf-8"?>
4 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context="com.example.myapplication.MainActivity">
9
10     <TextView
11         android:id="@+id/textView"
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="Hello World!"
15         app:layout_constraintLeft_toLeftOf="parent"
16         app:layout_constraintRight_toRightOf="parent"
17         app:layout_constraintTop_toTopOf="parent"
18         app:layout_constraintBottom_toBottomOf="parent" />
19
20 </android.support.constraint.ConstraintLayout>

```

執行這個應用程式，在模擬裝置啟動應用程式以後，可以在畫面看到「標題」訊息。如果在開發應用程式之前經過完整的規劃，在就可以建立好大部份需要的文字資源。在開發的過程需要新增或修改文字資源的時候，再開啟文字資源檔案執行異動的工作。

## 5-2-2 建立顏色資源

如果設計應用程式畫面的時候，沒有特別設定元件的顏色，畫面看起來應該會很單調。需要為畫面設定顏色的時候，也不應該寫在程式碼與設定檔。例如瀏覽與檢視記事本內容的畫面：



顏色資源的內容使用下面這些設定格式：

- #RGB – 使用0~9、A~F設定紅綠藍的配色，共256種顏色。
- #RRGGBB – 使用00~FF設定紅綠藍的配色，共65535種顏色。
- #ARGB – 第一碼使用0~9、A~F設定透明度，0表示完全透明，就是看不到了，F表示完全不透明。
- #AARRGGBB – 使用00~FF設定透明度，00表示完全透明，FF表示完全不透明。

這個畫面需要一些灰階的顏色資源，用來設定元件的背景顏色。開啟Android Studio為你建立好的「res/values/colors.xml」檔裡面已經有colorPrimary、colorPrimaryDark與colorAccent預設的顏色資源，依照下列的內容加入需要的顏色資源：

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>

```

```
<color name="light_grey">#CCCCCC</color>
<color name="grey">#AAAAAA</color>
<color name="divider_color">#DD999999</color>
</resources>
```

在XML設定檔使用「@color/顏色資源名稱」的格式設定顏色資源。開啟「res/layout/activity\_main.xml」，修改這個檔案的內容，為文字元件加入背景顏色的設定：

```
<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

    <!-- 加入背景顏色設定 -->
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/light_grey"
        android:text="@string/title"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

你可以在預覽畫面檢視，或是執行這個應用程式，在模擬裝置啟動應用程式以後，可以看到「標題」文字元件的背景顏色已經灰色。

## 5-2-2 建立尺寸資源

應用程式的畫面通常需要設定元件的大小與排列間隔，例如一些比較重要的文字，應該會希望顯示在畫面的時候可以大一些。元件與元件之間，通常會設定一些間隔，這樣畫面才不會擠成一團。尺寸資源的內容使用下面這些單位：

- px – 螢幕畫素。
- dp – 每英吋畫面，160dp為一英吋。
- sp – 和dp一樣，不過會根據裝置設定的字型大小自動調整。
- in – 英吋。
- mm – 公厘。

依照下列的步驟建立尺寸資源檔案：

1. 在「res/values」目錄上按滑鼠右鍵，選擇「New -> Values resources file」。
2. 在「File name」輸入「dimens」後選擇「OK」。

建立尺寸資源檔以後，加入應用程式需要的尺寸資源：

```
<resources>
    <!-- 加入尺寸資源 -->
    <dimen name="default_padding">6dp</dimen>
    <dimen name="title_txt_size">24sp</dimen>
</resources>
```

在XML設定檔中使用「@dimen/尺寸資源名稱」格式設定尺寸資源。開啟「res/layout/activity\_main.xml」，修改這個檔案的

```
<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="net.macdidi.atk.MainActivity">

    <!-- 加入背景顏色設定 -->
    <!-- 加入邊界與文字大小的設定 -->
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/light_grey"
        android:text="@string/title"
        android:padding="@dimen/default_padding"
        android:textSize="@dimen/title_txt_size"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

你可以在預覽畫面檢視，或是執行這個應用程式，在模擬裝置啟動應用程式以後，可以看到訊息的大小與邊界已經改變。

## 5-3 圖形資源

為了要提供使用者更清楚和美觀的應用程式畫面，通常會在設計畫面的時候使用一些圖形，例如按鈕或選單的圖示。應用程式在裝置以後，顯示在應用程式列表中的圖示也很重要，一個適合又好看的圖示，通常會比只有使用文字好一些。應用程式需要的圖形資源都會放在應用程式專案的「res/drawable」目錄。

Android系統的圖形資源可以接受PNG、JPG和GIF格式的圖檔，檔案名稱必須是小寫的英文字母和底線，沒有包含副檔名的檔名就是資源名稱。在res/drawable目錄也可以建立XML格式的圖形資源，提供應用程式一些簡單的幾何圖形和繪圖效果，例如使用層色彩繪製的矩形，可以使用在設定畫面元件的背景。

### 5-3-1 圖檔圖形資源

經過應用程式規劃的階段，應該已經知道需要哪些圖形資源，例如這個畫面需要的按鈕圖示：

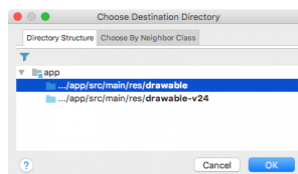


應用程式需要的基本圖形檔案，可以在Android開發人員網站下載，網址是<https://github.com/google/material-design-icons>，

儲存並解壓縮下載的檔案，你可以參考官方網站「<https://www.google.com/design/icons/>」，在分類好的資料夾裡面找到需要的圖檔。如果是提供給按鈕使用的圖示，選擇「drawable-mdpi」目錄下的圖檔，大小為48 X 48。或是你可以在這裡<https://github.com/macdidi5/Android-Tutorial-Kotlin/tree/master/images/icons>下載這個課程需要的圖示檔案。

準備好圖示的檔案以後，依照下列的步驟，把圖檔複製到應用程式專案：

1. 在檔案總管選擇需要的圖檔以後，選擇「複製」。
2. 回到Android Studio，在「res/drawable」目錄上按滑鼠右鍵，選擇「Paste」。
3. 如果出現這個對話框，選擇「.../app/src/main/res/drawable」以後選擇「OK」：

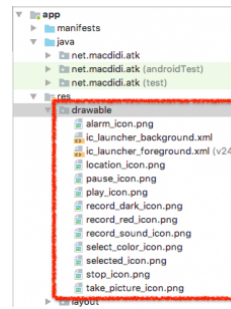


4. 在這個對話框選擇「OK」：



5. 複製完成後，在res/drawable目錄下就會出現圖檔的名稱：





將應用程式需要的圖檔複製到res/drawable目錄以後，就可以在XML資源檔案中用「@drawable/資源名稱」的格式使用這些圖源。開啟「res/layout/activity\_main.xml」，修改這個檔案的內容：

```
<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="net.macdidi.atk.MainActivity">

    <TextView ... />

    <!-- 加入圖形元件 -->
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="32dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        app:layout_constraintBottom_toTopOf="@+id/textView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:srcCompat="@drawable/alarm_icon" />

</android.support.constraint.ConstraintLayout>
```

你可以在預覽畫面檢視，或是執行這個應用程式，在模擬裝置啟動應用程式以後，可以看到畫面多一個圖示。

### 5-3-2 繪圖圖形資源

一個應用程式提供的功能最好是能夠很實用，讓使用者操作與瀏覽資料的畫面美觀也非常重要，使用者當然希望應用程式的畫面以好看一些。在設計畫面配置資源的時候，你可以為畫面元件設定大小尺寸、文字與背景的颜色，讓應用程式的畫面看起來不會那麼單調，不過這些基本的設定所呈現的效果也是有限的。想要讓畫面的設計比較不一樣，可以不用到處找一些圖形，也不學習繪圖軟體，Android提供一種使用XML格式設計的圖形資源，可以讓你設計簡單的繪圖圖形資源，把這種圖形資源設定給畫面當作背景，就可以讓畫面元件擁有獨特的外觀。

繪圖圖形資源是一個XML格式的檔案，儲存在專案的「res/drawable」目錄，檔案名稱就是資源名稱。這是設計繪圖圖形資源式：

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape=["rectangle" | "oval" | "line" | "ring"] >
    <corners
        android:radius="integer"
        android:topLeftRadius="integer"
        android:topRightRadius="integer"
        android:bottomLeftRadius="integer"
        android:bottomRightRadius="integer" />
    <gradient
        android:angle="integer"
        android:centerX="integer"
        android:centerY="integer"
        android:centerColor="integer"
        android:endColor="color"
        android:gradientRadius="integer"
        android:startColor="color"
        android:type=["linear" | "radial" | "sweep"]
        android:useLevel=["true" | "false"] />
    <padding
        android:left="integer"
        android:top="integer"
        android:right="integer"
        android:bottom="integer" />
    <size
        android:width="integer"
        android:height="integer" />
    <solid
        android:color="color" />
    <stroke
        android:width="integer"
        android:color="color"
        android:dashWidth="integer"
        android:dashGap="integer" />
</shape>
```

這個繪圖圖形資源檔案的內容，最外層是「shape」標籤，標籤中的「android:shape」設定決定繪圖的種類，可以設定為下列設定值：

- **Rectangle** – 繪製矩形，可以搭配「corners」標籤設定四邊的圓角。
- **Oval** – 繪製橢圓形。
- **Line** – 繪製線條。
- **Ring** – 繪製圓環，在shape標籤中控制繪製的效果。

你可以使用繪圖圖形資源，為畫面元件設定一個特殊的樣式。依照下列的步驟建立一個繪圖圖形資源檔案：

1. 在「res/drawable」目錄上按滑鼠右鍵，選擇「New -> Drawable resource file」。
2. 在「File name」輸入「rectangle\_drawable」後選擇「OK」。

修改rectangle\_drawable.xml為下面的內容：

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >

    <corners
        android:topLeftRadius="5sp"
        android:topRightRadius="5sp"
        android:bottomLeftRadius="5sp"
        android:bottomRightRadius="5sp" />

    <solid android:color="#AAAAAA"/>

</shape>
```

儲存建立好的繪圖圖形資源檔案，就可以在XML資源檔案中用「@drawable/資源名稱」的格式使用這些資源。開啟「res/layout/activity\_main.xml」，修改這個檔案的內容：

```
<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="net.macdidi.atk.MainActivity">

    <!-- 加入邊界與文字大小的設定 -->
    <!-- 修改android:background的設定 -->
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/rectangle_drawable"
        android:text="@string/title"
        android:textSize="@dimen/title_txt_size"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```
<ImageView ... />

</android.support.constraint.ConstraintLayout>
```

儲存檔案以後，你可以在預覽畫面檢視，或是執行這個應用程式後執行這個應用程式，在模擬裝置啟動應用程式以後，可以看見呈現圓角的圖形背景。

## 5-4 選單資源

一般的應用程式通常使用畫面元件提供使用者執行功能操作，例如按鈕或下拉式選單，這些元件會顯示在畫面上，讓使用者可以隨時操作它們。你可以在應用程式畫面的最上方提供圖示或文字選單，選單資源是一個XML格式的檔案，放在專案的「res\menu」目錄，檔案名稱就是它的資源名稱，設定檔最外層的標籤是「menu」，裡面可以包含設定選單項目的「item」標籤，選單項目下面有這些設定值：

- **android:id** – 選單項目的資源名稱。
- **android:title** – 選單項目的文字。
- **android:icon** – 選單項目的圖示，指定一個圖形資源。
- **android:showAsAction** – 設定選單項目的樣式，可以設定為「ifRoom」、「never」、「withText」、「always」和「collapseActionView」。設定為**never**表示使用一般選單的樣式，其它的設定將選單顯示在Action Bar。
- **android:onClick** – 選擇選單以後呼叫的函式名稱。

依照下列的步驟建立一個選單資源檔案：

1. 在「res」目錄上按滑鼠右鍵，選擇「New -> Android resource file」。
2. 在「File name」輸入「menu\_main」。
3. 在「Resource type」選擇「Menu」。
4. 選擇「OK」。

把建立好的選單資源檔案修改為下面的內容：

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/search_item"
        android:icon="@android:drawable/ic_menu_search"
        android:title="SEARCH"
        app:showAsAction="always" />

    <item
        android:id="@+id/add_item"
        android:icon="@android:drawable/ic_menu_add"
        android:title="ADD"
        app:showAsAction="always" />
```

```

        <item
            android:id="@+id/revert_item"
            android:icon="@android:drawable/ic_menu_revert"
            android:title="REVERT"
            app:showAsAction="always" />

        <item
            android:id="@+id/delete_item"
            android:icon="@android:drawable/ic_menu_delete"
            android:title="DELETE"
            app:showAsAction="always" />
    </menu>

```

開啟專案的「**MainActivity.java**」，依照下列的說明加入載入選單資源的函式：

```

package net.macdidi.atk

import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.view.Menu

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    // 加入載入選單資源的函式
    override fun onCreateOptionsMenu(menu: Menu): Boolean {
        menuInflater.inflate(R.menu.menu_main, menu)
        return true
    }

}

```

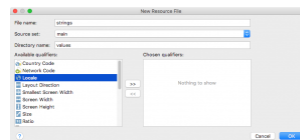
選單不會在預覽畫面顯示，所以執行這個應用程式，在模擬裝置啟動應用程式以後，可以看到設定好的選單。目前還沒有為選定選擇以後要執行的工作，所以在選擇以後都沒有任何反應。



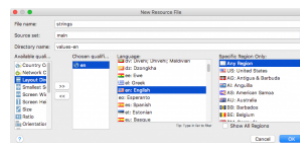
## 5-5 設計與使用動態資源

以文字資源來說，Android裝置可以設定為不同的語言與地區，如果希望應用程式可以根據裝置設定的語言與地區，自動使用文字資源，就需要另外建立其它語言的文字資源。目前建立的文字資源是繁體中文的內容，如果需要提供英文的文字資源，作列的步驟新增資源檔案：

1. 在「res/values」目錄上按滑鼠右鍵，選擇「New -> Values resources file」。
2. 在「File name」輸入「strings」，在Available qualifiers選擇「Locale」後選擇「>>」：



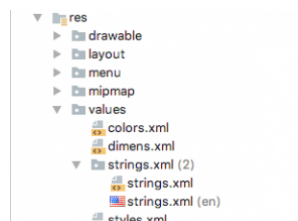
3. 在Language選擇「en:English」，Specific Region Only選擇「Any Region」，最後選擇「OK」：



完成上面的步驟以後，Android Studio建立一個英文文字資源檔，把「res/values/strings.xml」的內容複製過來，把它改為英文：

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="title">Title</string>
    <string name="enter_title">Enter title</string>
    <string name="content">Content</string>
    <string name="enter_content">Enter content</string>
</resources>
```

在資源管理目錄可以看到文字資源有兩個檔案：



目前的應用程式專案有預設的繁體中文與英文文字資源。執行這個應用程式，在模擬裝置執行「設定」應用程式，選擇「語言」->「語言」，選擇「English(United States)」以後，在重新執行應用程式，畫面上的訊息會根據裝置的語言設定顯示對應文字，原來畫面上顯示的「標題」會變成「Title」。如果希望應用程式支援更多語言，可以使用同樣的作法新增其它語言的文字資源。

完整的資源設定請參考Android開發人員網站，網址是：<http://developer.android.com/guide/topics/resources/index.html>



相關的檔案都可以在GitHub瀏覽與下載：

# GitHub


<https://github.com/macdidi5/Android-Tutorial-Kotlin>

[後續 >> Android Tutorial using Kotlin 第二堂（2）設計應用程式使用者介面](#)

Does Clearly work fine?



Shortcuts: **SHIFT+CTRL+C** to Toggle, **ESC** to Close.

 Give us feedback

Build upon ♥ with Clearly