

# 圖片相關介面元件

**ImageView** 介面元件主要用以顯示影像，**Gallery** 以圖片列表顯示所有圖片，支援手指左右拖曳滑動的效果，並且可以選取指定的圖片。**GridView** 以欄列二維的方式顯示表格，並在表格中放置圖片。

## 學習重點

- **ImageView** 介面元件
- **Gallery** 介面元件
- 繼承 **BaseAdapter**
- 影片循環的 **Gallery**
- **GridView** 介面元件

# 06



碁峯

www.gotop.com.tw



## 6.1 ImageView 介面元件—顯示影像

ImageView 介面元件主要用以顯示影像，讓顯示畫面更加美觀和生動。

### 6.1.1 ImageView 的語法與常用屬性

#### ImageView 的語法

在 Android 程式中使用影像圖片有幾個注意事項：

1. 影像圖片必須依使用的解析度，分別放置在 `<res/drawable-hdpi>`、`<res/drawable-ldpi>`、`<res/drawable-mdpi>`、`<res/drawable-xhdpi>` 或 `<res/drawable-xxhdpi>` 中。但如果沒有那麼講究，可以直接放置 `<res/drawable>` 目錄中。
2. 程式的影像檔格式可以使用 `png`、`jpg` 或 `gif`，要注意的是檔案命名時無論檔名或副檔名都必須使用小寫字元，否則執行時會產生應用程式的錯誤。

ImageView 介面元件的標準語法如下：

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

例如：建立 ImageView 元件，名稱為「imgPhoto」，影像的圖片為「R.drawable.img01」，圖片按原來的尺寸居中顯示。

```
<ImageView android:id="@+id/imgPhoto"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/img01"  
    android:scaleType="fitCenter"/>
```

在程式中設定 ImageView 元件顯示指定的圖片，可以使用 `setImageResource()` 方法設定 ImageView 顯示的影像。例如我們要指定名稱為「imgPhoto」，影像的圖片為「R.drawable.img01」，方式如下：

```
imgPhoto.setImageResource(R.drawable.img01);
```

## ImageView 的常用屬性

屬性名稱	對應的 xml 程式碼	說明	
Src	android:src	設定圖片來源。 圖片放置在指定的資料夾後，系統會自動在資源類別檔 <R.java> 中註冊，一般來源檔會由其中的 R.drawable 取得 id。	
Scale type	android:scaleType	設定顯示圖片的縮放方式。以下是常用類型：	
		center	按原來尺寸，將圖片居中顯示。
		centerCrop	按原始比例縮放圖片，並將圖片居中顯示，超過部份裁切。
		centerInside	將圖片的內容完整居中顯示
		fitCenter	按比例居中完整顯示圖片
		fitEnd	按比例靠右或下完整顯示圖片
		fitStart	按比例靠左或上完整顯示圖片
		fitXY	不按比例依元件大小顯示圖片
matrix	依 matrix 矩陣來顯示圖片		

## 在 Eclipse 專案中加入圖片檔案

剛才曾說過 Android 圖片的位置為 <res/drawable> 目錄，但是新的專案下預設並沒有這個資料夾，所以我們必須先新增資料夾後再將圖片放進去，步驟如下：

1. 在 Eclipse 中開啟 Package Explorer 視窗，開啟編輯的專案後，在 <res> 資料夾按右鍵，選按 **New / Folder**，在對話方塊的 **Folder name** 欄輸入「drawable」後按 **Finish** 鈕完成。
2. 在 Windows 檔案總管中選取要加入的圖片檔後按 **Ctrl + C** 鍵複製，然後在 Eclipse 中剛新增的 <res/drawable> 目錄上按右鍵選按 **Paste** 貼上，即可將選取的圖片加入至 drawable 目錄中。

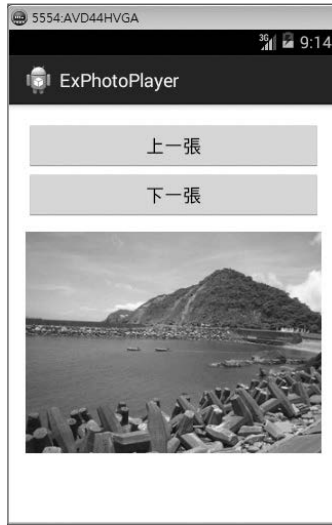


如果使用者是在 Windows 檔案總管下新增或刪除影像檔案，則必須在 Eclipse 中，再選取該專案並在主功能中以 **File \ Refresh** 更新資源類別檔。



## 6.1.2 範例：影像播放器

利用 `ImageView` 顯示影像，並可以 上一張、下一張 按鈕顯示其他的影像。



圖片的切換是利用按鈕，所以程式的觸發要寫在按鈕上。



### 新增專案並完成版面配置

請新增 `<ExPhotoPlayer>` 專案，在 `<activity_main.xml>` 中佈建兩個 `Button` 介面元件、一個 `ImageView` 介面元件，並設定相關屬性如下：

```
<ExPhotoPlayer/res/layout/activity_main.xml>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <Button android:id="@+id/btnPrev"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text=" 上一張 " />
```

```

<Button android:id="@+id/btnNext"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/btnPrev"
        android:text="下一張" />

<ImageView android:id="@+id/imgPhoto"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:layout_centerHorizontal="true"
           android:layout_below="@+id/btnNext"
           android:scaleType="fitCenter"
           android:src="@drawable/img01" />

</RelativeLayout>

```

版面配置使用預設的 `RelativeLayout`，三個元件名稱分別是 `btnPrev`、`btnNext`、`imgPhoto`，`imgPhoto` 預設的圖像檔是「`R.drawable.img01`」，寬和高依圖片原來大小居中顯示。

## 加入圖片

請在專案 `<res>` 目錄下建立 `<drawable>` 資料夾，並加入附書光碟中 `<drawable>` 的原始圖檔 `<img01.jpg>` ~ `<img06.jpg>` 共 6 張，請注意檔名必須使用小寫字元。

## 加入執行的程式碼

請開啟 `<src/com.ehappy.exphotoplayer/MainActivity.java>`。程式中 `import` 套件程式省略，程式中也加入一些註解，幫助讀者了解程式。

```

<ExPhotoPlayer/src/com.ehappy.exphotoplayer/MainActivity.java>
...略
9  public class MainActivity extends Activity {
10     // 將所有的圖片儲存至陣列中
11     int[] imgId={R.drawable.img01,R.drawable.img02,R.drawable.img03,
12                 R.drawable.img04,R.drawable.img05,R.drawable.img06};
13     private Button btnPrev,btnNext;
14     private ImageView imgPhoto;
15     int p=0; // 圖片的索引 (第幾張圖片)
16     int count=imgId.length; // 共有多少張圖片

```



```
17  @Override
18  protected void onCreate(Bundle savedInstanceState) {
19      super.onCreate(savedInstanceState);
20      setContentView(R.layout.activity_main);
21
22      // 取得資源類別 R 中的介面元件
23      btnPrev=(Button)findViewById(R.id.btnPrev);
24      btnNext=(Button)findViewById(R.id.btnNext);
25      imgPhoto=(ImageView)findViewById(R.id.imgPhoto);
26
27      // 設定 button 元件 Click 事件的 listener
28      btnPrev.setOnClickListener(btnPrevListener);
29      btnNext.setOnClickListener(btnNextListener);
30  }
31
32  // 定義 btnPrev 按鈕的 onClick() 方法
33  private Button.OnClickListener btnPrevListener=
34      new Button.OnClickListener(){
35      public void onClick(View v){
36          p--;
37          if (p<0)
38              p=count-1;
39          imgPhoto.setImageResource(imgId[p]);
40          setTitle("第 " + (p+1) + "/" + count);
41      }
42  };
43
44  // 定義 btnNext 按鈕的 onClick() 方法
45  private Button.OnClickListener btnNextListener=
46      new Button.OnClickListener(){
47      public void onClick(View v){
48          p++;
49          if (p==count)
50              p=0;
51          imgPhoto.setImageResource(imgId[p]);
52          setTitle("第 " + (p+1) + "/" + count);
53      }
54  }
55 }
```

❶ 宣告全域變數：因為 btnPrev、btnNext 和 imgPhoto 物件變數必須在不同的方法中存取，必須宣告為全域變數。

第 11 列，將圖檔儲存在至 `imgId` 陣列中，再透過陣列來存取。

第 15 列，設定圖片的索引 `p` (第幾張圖片)，初始值是 0。

第 16 列，利用 `count` 記錄共有多少張圖片，`imgId.length` 可取得陣列的長度，本例是 6。

- ❷ 接著在 `onCreate` 方法中利用 `findViewById()` 取得的 `btnPrev`、`btnNext` 和 `imgPhoto` 介面元件並儲存到剛才宣告的三個變數中。
- ❸ 為按鈕 `btnPrev` 元件加入按下的偵聽動作在觸發時執行 `btnPrevListener` 自訂方法，`btnNext` 元件加入按下的偵聽動作在觸發時執行 `btnNextListener` 自訂方法。
- ❹ 自訂 `btnPrevListener` 方法來處理在觸發按鈕被按下後的功能。  
按下「上一張」按鈕，將 `p` 減 1，並判斷是否 `p < 0`，也就是目前顯示的照片數如果是第一張，則將 `p` 的索引設為 `imgId.length-1` 跳到最後一張。  
第 37 列，因為是 `p` 圖片的索引值，必須再減 1 作修正。  
第 38 列，`imgPhoto.setImageResource(imgId[p])` 設定顯示的圖檔。  
第 39 列，以 `setTitle()` 在標題列上顯示目前圖片是第幾張，共幾張的資訊。
- ❺ 自訂 `btnNextListener` 方法來處理在觸發按鈕被按下後的功能。  
按下「下一張」按鈕，將 `p` 加 1，並判斷是否 `p == count`，也就是目前顯示的照片數如果是最後一張，則將 `p` 的索引設為 0 返回第一張。

儲存專案後請按下 **Ctrl + F11** 執行專案，請選按上方的按鈕來切換圖片，在標題處會顯示圖片共幾張、目前在第幾張的訊息。





## 6.2 Gallery 介面元件—畫廊展示

Gallery 元件的展示方式是將圖片由左至右方式擺放，如同畫廊放置作品一樣。但是較為特殊的是 Gallery 元件支援手指左右拖曳滑動的效果，並且可以選取指定的圖片，對於版面佈置來看是相當靈活而有用的元件。

要注意的是：Android 4.1.2 版後 (API 16) 宣布不再支援 Gallery 的更新，並推薦以 HorizontalScrollView 和 ViewPager 元件來取代，但操作過於繁瑣。其實也可以用轉換專案版本和編輯器的方式處理，這也是讓專案向下相容的常用技巧。

### 6.2.1 Gallery 語法範例與常用的屬性

例如：我們要建立一個 Gallery 元件，名稱為「Gallery01」，Gallery 元件和邊界的距離是 5dp，圖片間の間隔是 5dp，寬度填滿螢幕，高度依圖片高度調整。

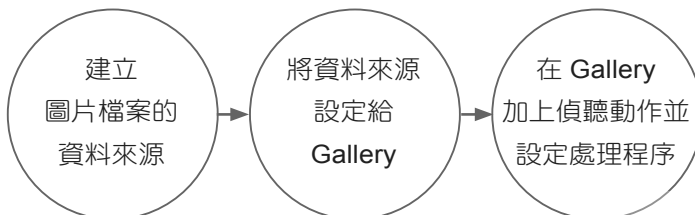
```
<Gallery android:id="@+id/Gallery01"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:spacing="5dp"/>
```

Gallery 提供許多屬性來設定其特性，常用的屬性如下：

屬性名稱	對應的 xml 程式碼	說明
Animation duration	android:animationDuration	設定動畫過渡時間。
Gravity	android:gravity	圖片對齊方式。
Unselected alpha	android:unselectedAlpha	選中圖片的透明度。
Spacing	android:spacing	圖片間の間隔。

### 6.2.2 Gallery 程式執行的流程

一般來說 Gallery 程式的執行流程重點如下：





Gallery 元件所使用圖片的資料來源是以 `BaseAdapter` 物件的型式存在，所以在使用前必須要先建立 `BaseAdapter` 類別。

### 6.2.3 繼承 `BaseAdapter`

使用 Gallery 元件的資料來源必須建立一個繼承自 `BaseAdapter` 的物件來放置想要呈現的圖片資料。在程式中繼承 `BaseAdapter` 後，必須實作 `getCount()`、`getItem()`、`getItemId()`、`getView()` 等方法，並將 Gallery 元件欲顯示的圖片建立在 `getView()` 方法中。

#### 建立繼承 `BaseAdapter` 的類別

以下我們以建立 `MyAdapter` 類別來繼承 `BaseAdapter` 為例，產生的程式架構如下，我們最主要的工作就是在 `getView()` 方法加入要顯示的內容，當然預設的參數命名也可改為較易辨識的名稱。`getCount()` 取得共有多少張圖片，`getItem()` 是取得目前的選項，`getItemId()` 是取得目前選項的 `Id`，`getView()` 方法定義要顯示的內容。

```
class MyAdapter extends BaseAdapter { ←繼承 BaseAdapter 類別
    @Override
    public int getCount() {
        // TODO Auto-generated method stub
        return 0;
    }

    @Override
    public Object getItem(int position) {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public long getItemId(int position) {
        // TODO Auto-generated method stub
        return 0;
    }

    @Override
    public View getView(int position, View convertView,
        ViewGroup parent) {
```

這裡是第一次接觸 `BaseAdapter`，未來有許多元件都必須利用它來顯示內容，務必要多多練習。





```
// TODO Auto-generated method stub  
return null;  
}  
}
```

例如：定義 Gallery 元件要顯示的圖片是一個 ImageView 元件，ImageView 寬、高為 120\*90，圖像置中，圖像來源為 imageIds 陣列。

```
// 定義顯示的圖片  
public View getView(int position, View convertView, ViewGroup  
parent)  
{  
    ImageView iv = new ImageView(mContext);  
    iv.setImageResource(imageIds[position]);  
    iv.setScaleType(ImageView.ScaleType.FIT_CENTER);  
    iv.setLayoutParams(new Gallery.LayoutParams(120,90));  
    return iv;  
}
```

## 6.2.4 範例：以 Gallery 元件顯示圖片列

以 Gallery 元件顯示圖片列，當在 Gallery 元件中選取指定的圖片，同時會在下方的 ImageView 元件顯示完整圖片。



這個範例很精彩喔！

