

# Android App

## 程式設計教本 之無痛起步

引領入門

最簡單、最易懂的初學教材



## 第 5 章

# 使用者界面的基本元件

本投影片（下稱教用資源）僅授權給採用教用資源相關之旗標書籍為教科書之授課老師（下稱老師）專用，老師為教學使用之目的，得摘錄、編輯、重製教用資源（但使用量不得超過各該教用資源內容之80%）以製作為輔助教學之教學投影片，並於授課時搭配旗標書籍公開播放，但不得為網際網路公開傳輸之遠距教學、網路教學等之使用；除此之外，老師不得再授權予任何第三人使用，並不得將依此授權所製作之教學投影片之相關著作物移作他用。

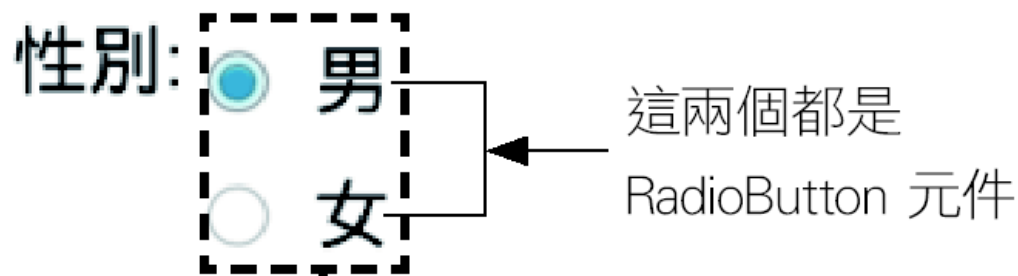
# 前言

- 5-1 多選一的單選鈕 (RadioButton)
- 5-2 可複選的核取方塊 (CheckBox)
- 5-3 顯示圖形的 ImageView

# 5-1 多選一的單選鈕 (RadioButton)

- RadioButton 與 RadioGroup 元件
- isChecked() : 讀取單選鈕狀態
- onCheckedChangeListener() : 選項改變的事件

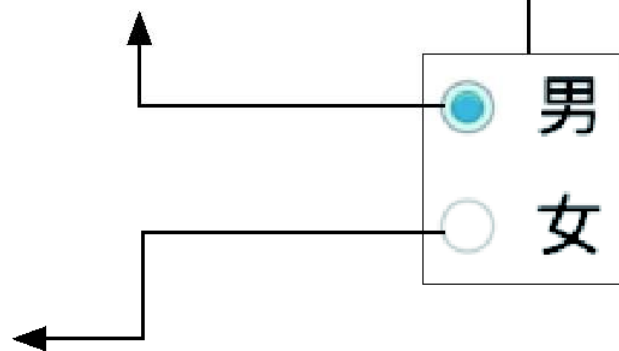
# RadioButton 與 RadioGroup 元件



外層有個『看不見』的 RadioGroup  
包住兩個 RadioButton

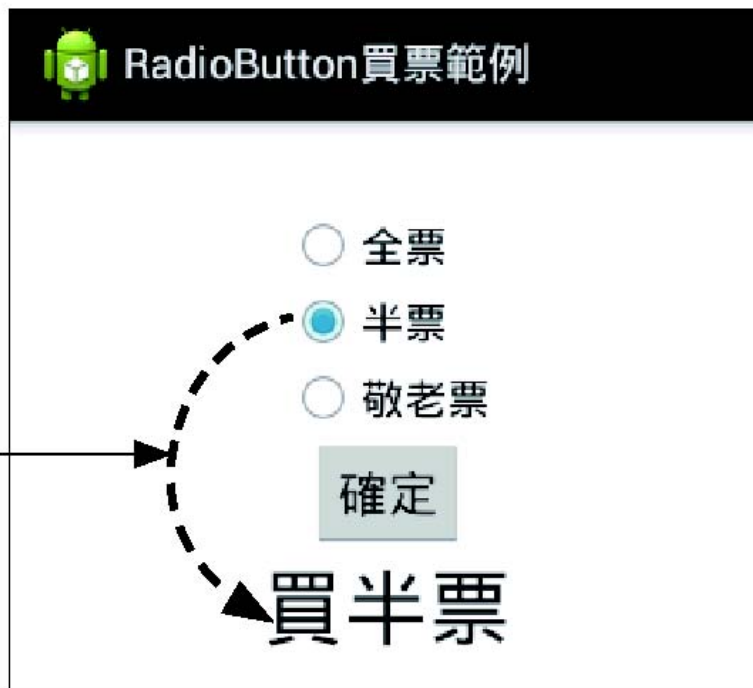
# getCheckedRadioButtonId() : 讀取 單選鈕狀態

```
// 取得 RadioGroup 之物件
RadioGroup sex = (RadioGroup) findViewById(...) ;
...
// 判斷選擇的項目資源 ID 是不是『男』選項的資源 ID
if (sex.getCheckedRadioButtonId() == R.id.male) {
    // 選『男』時要執行的程式片段
    ...
}
// 因為是 2 選 1，所以不是選『男』就是『女』
else {
    // 選『女』時要執行的程式片段
    ...
}
```



# 範例5-1：讀取 RadioGroup 的選取項目

按下按鈕, 程式就會讀取、  
顯示使用者選取的項目



# 讀取 RadioGroup 的選取項目

step 1

The screenshot displays the Android Studio IDE. The main preview window shows a mobile device screen with the title "RadioButton買票範例" and a "Hello world!" text view. A button with the text "確定" is positioned below the text. A callout box points to the button, showing its text as "確定" and its on-click action as "show". Another callout box points to the text view, showing its ID as "@+id/txv" and its text size as "30sp". The right-hand side of the IDE shows the "Structure" and "Properties" panels. The "Structure" panel lists a "RelativeLayout" containing a "TextView" with the text "Hello world!" and a "button1" with the text "確定". The "Properties" panel shows various attributes for the selected view, including "Content ...", "On Click" (set to "show"), "Over Scr...", "Alpha", "Text Alig...", "Important...", "Label For", and "Deprecated" (set to "(Deprecated Proper...").

Text	確定
On Click	show

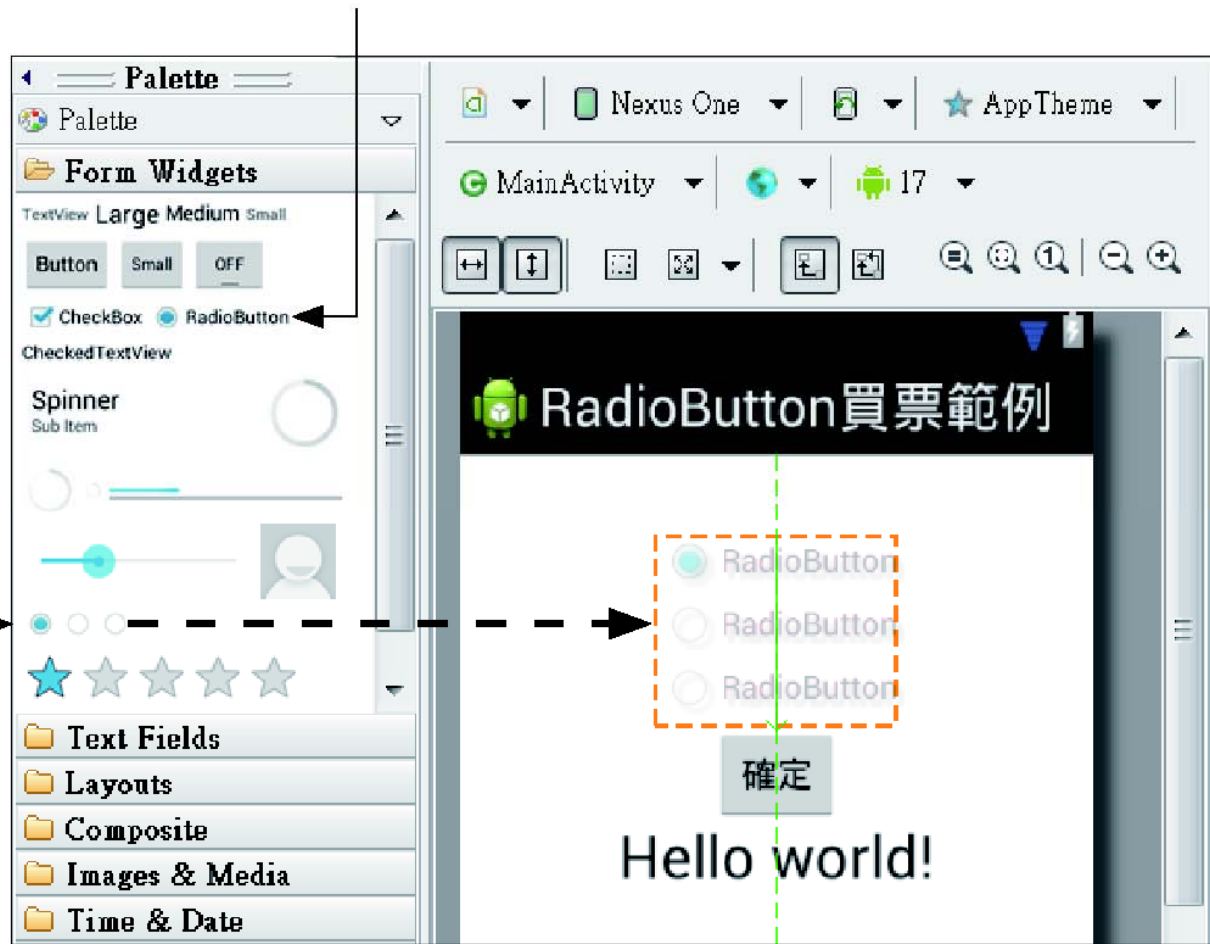
Id	@+id/txv
Text Size	30sp

# 讀取 RadioGroup 的選取項目

## step 2

此圖示代表單個 RadioButton

1 將 RadioGroup 元件  
(3 個圓圈的圖示)  
拉曳到按鈕之上





# 讀取 RadioGroup 的選取項目

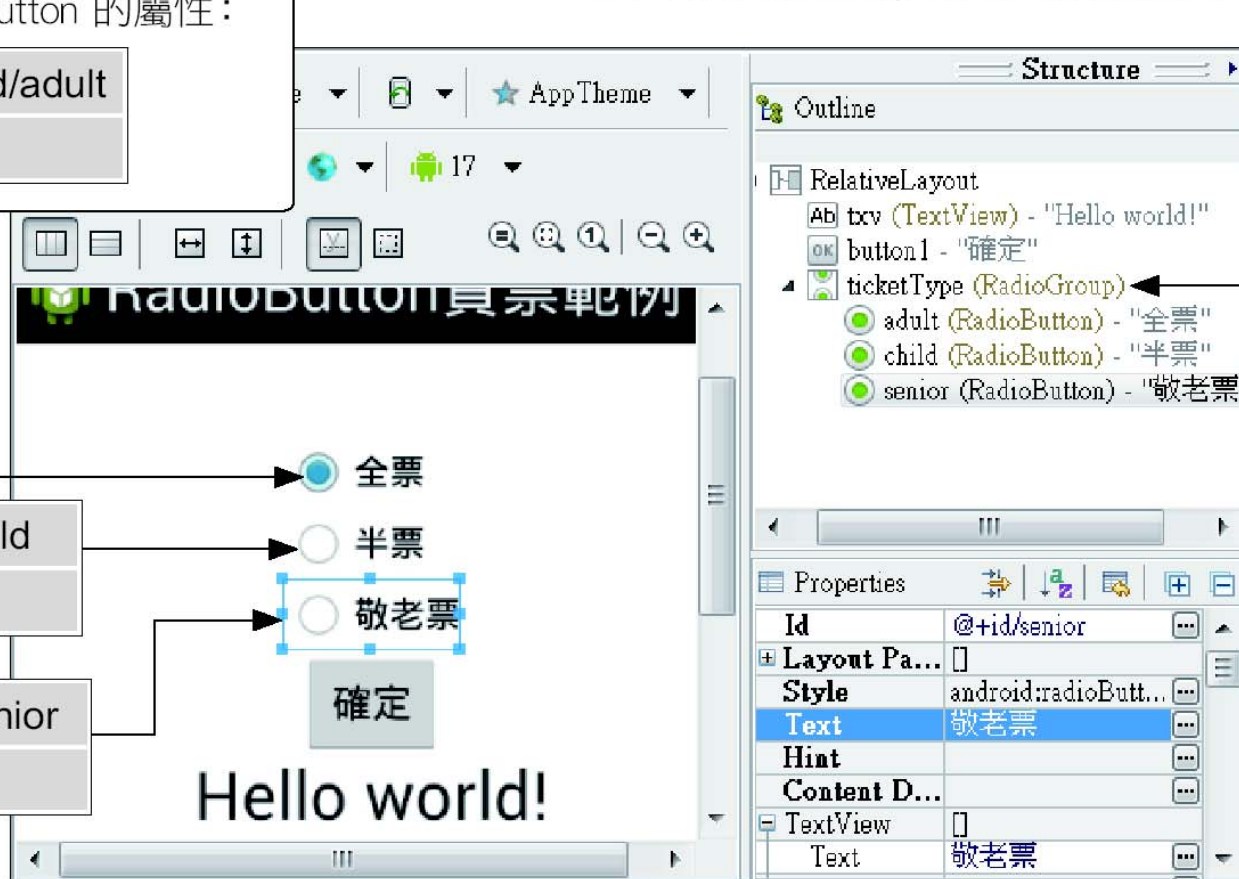
2 將 RadioGroup 的 Id 屬性值設為 "ticketType"

3 修改各 RadioButton 的屬性：

Id	@+id/adult
Text	全票

Id	@+id/child
Text	半票

Id	@+id/senior
Text	敬老票



# 讀取 RadioGroup 的選取項目

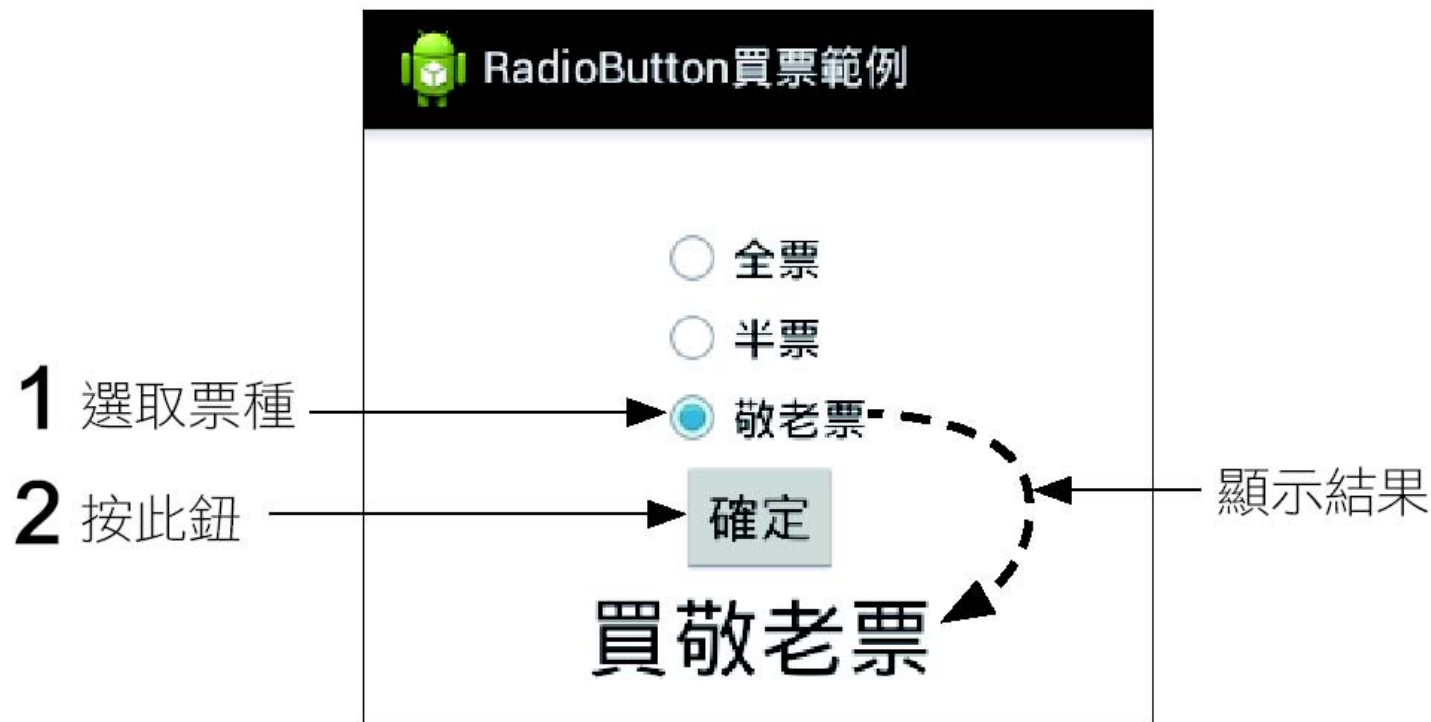
## step 3

```
01 public void show(View v){
02     TextView txv=(TextView)findViewById(R.id.txv);
03     RadioGroup ticketType =
04         (RadioGroup) findViewById(R.id.ticketType);
05
06     // 依選取項目顯示不同訊息
07     switch(ticketType.getCheckedRadioButtonId()){
08     case R.id.adult:           ← 選全票
09         txv.setText("買全票");
10         break;
11     case R.id.child:         ← 選半票
12         txv.setText("買半票");
13         break;
14     case R.id.senior:       ← 選敬老票
15         txv.setText("買敬老票");
16         break;
17     }
18 }
```

別忘了, R.id.XXX 都是在 R.java 中以 final 宣告的常數, 其值是固定不變的, 因此可用在 switch 敘述中用來區別選取的單選鈕。

# 讀取 RadioGroup 的選取項目

## step 4

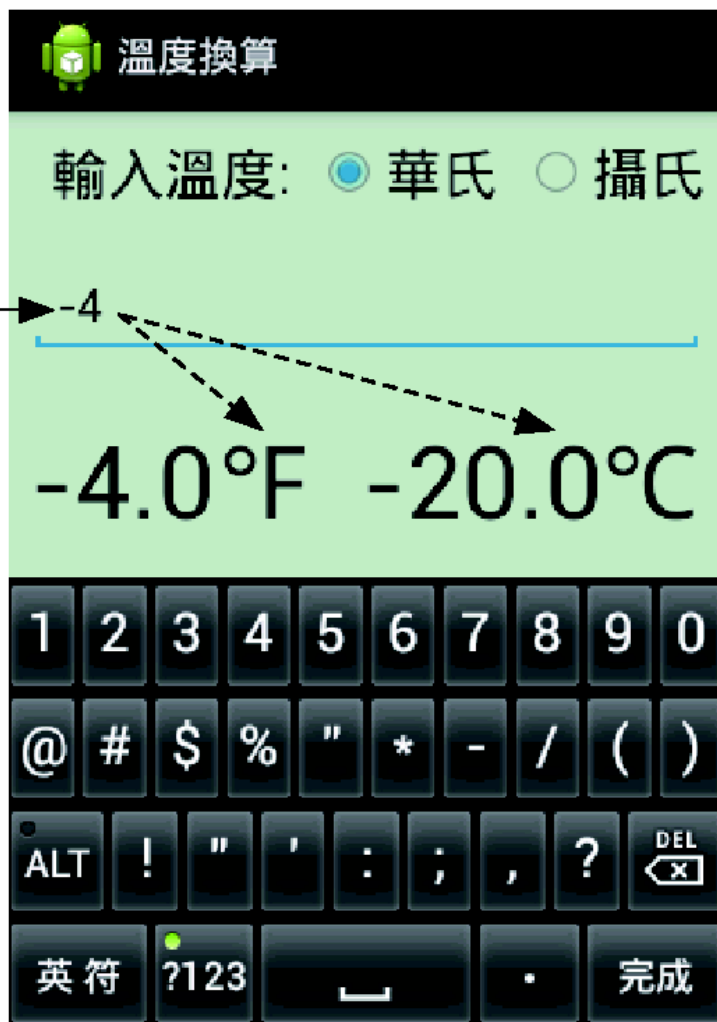


# onCheckedChanged() : 選項改變的事件



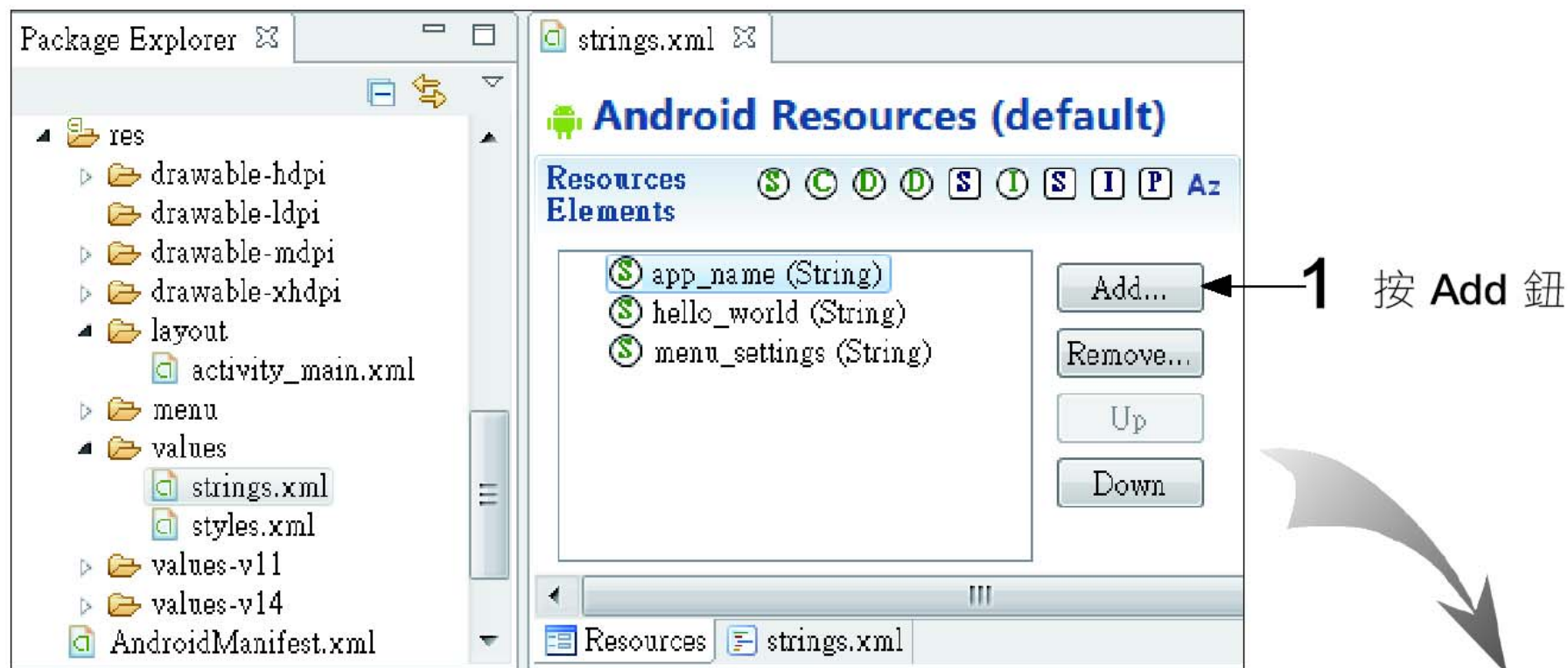
# 範例5-2：利用 RadioButton 選擇溫度轉換單位

輸入溫度值，  
程式就會立即  
換算並顯示  
結果



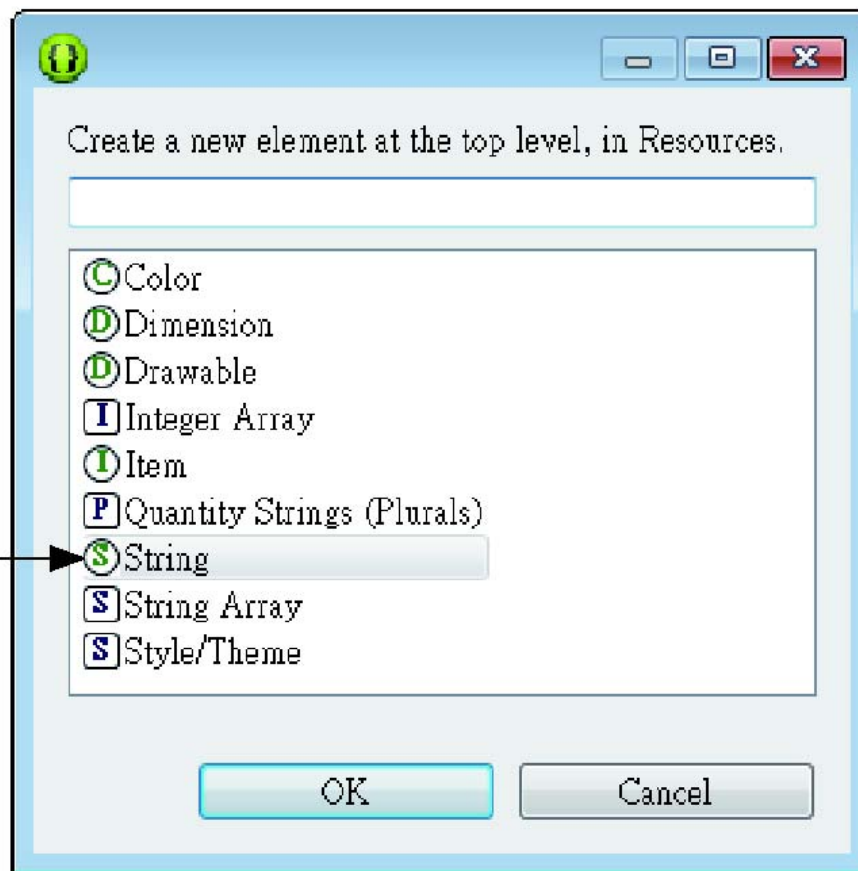
# 利用 RadioButton 選擇溫度轉換單位

## step 1

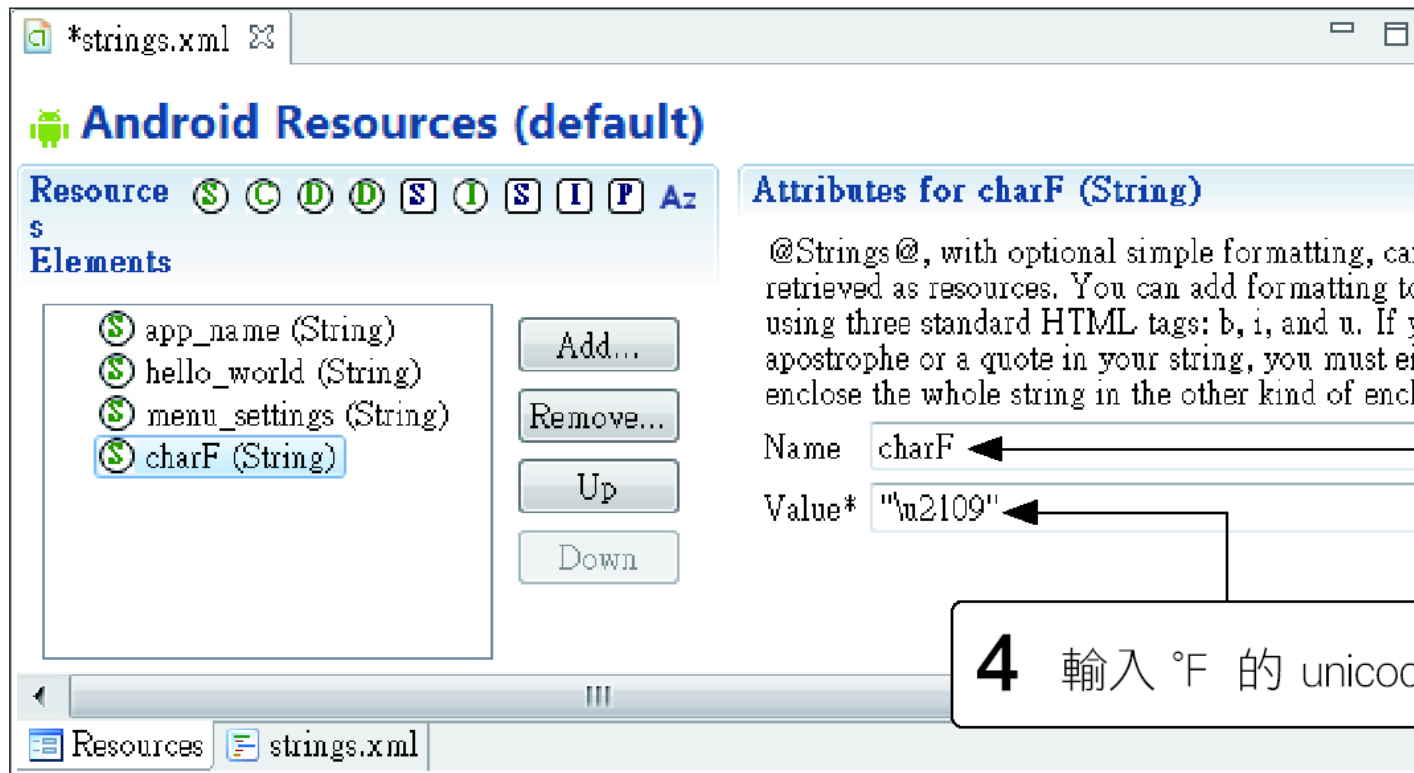


# 利用 **RadioButton** 選擇溫度轉換單位

2 雙按 **String** 表示  
要建立字串資源



# 利用 RadioButton 選擇溫度轉換單位

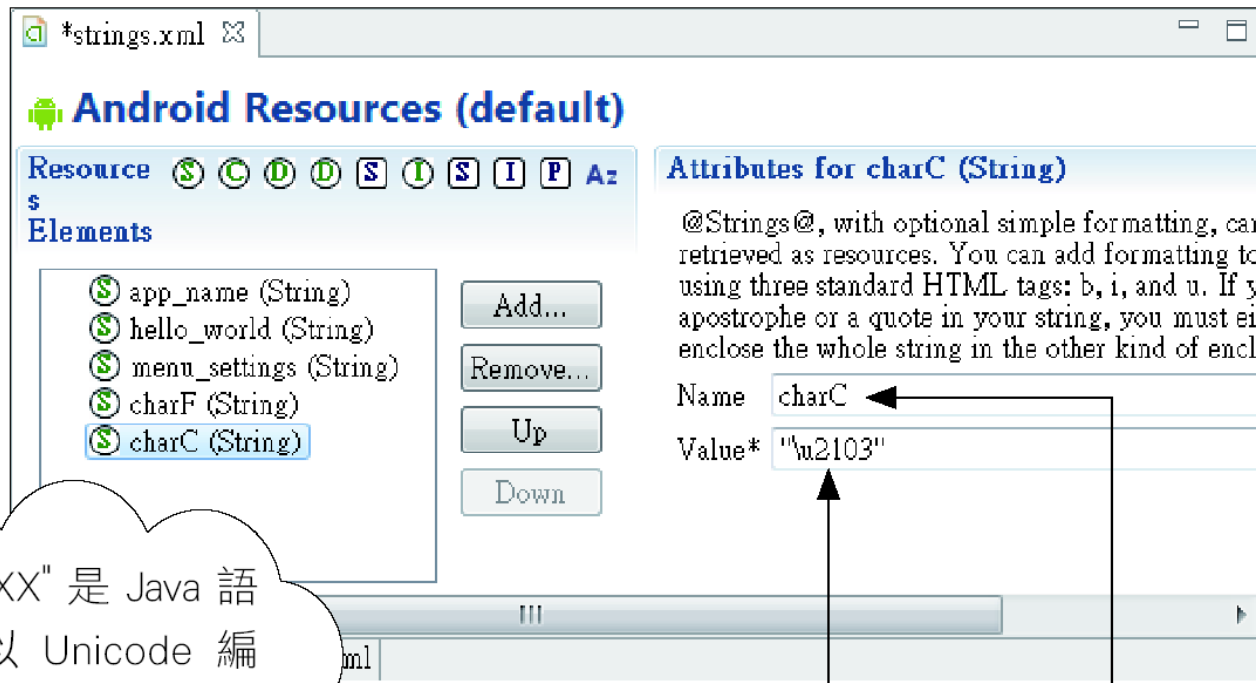


**3** 輸入字串名稱 "charF"

**4** 輸入 °F 的 unicode 字碼 "&u2109"



# 利用 RadioButton 選擇溫度轉換單位



"\uXXXX" 是 Java 語言中以 Unicode 編碼表示字元的語法

**6** 輸入 °C 的 unicode 字碼 "\u2103", 完成後記得要存檔

**5** 重複第 1、2 步加入另一個字串, 設定字串名稱 "charC"

# 利用 RadioButton 選擇溫度轉換單位

## step 2

RadioGroup

Id	@+id/unit
Orientation	horizontal

RadioButton

Id	@+id/unitF
Text	華氏
Text Size	22sp

RadioButton

Id	@+id/unitC
Text	攝氏
Text Size	22sp

"Large" TextView

Text	輸入溫度:
------	-------

AppTheme | MainActivity | Android 17

溫度換算

輸入溫度:  華氏  攝氏

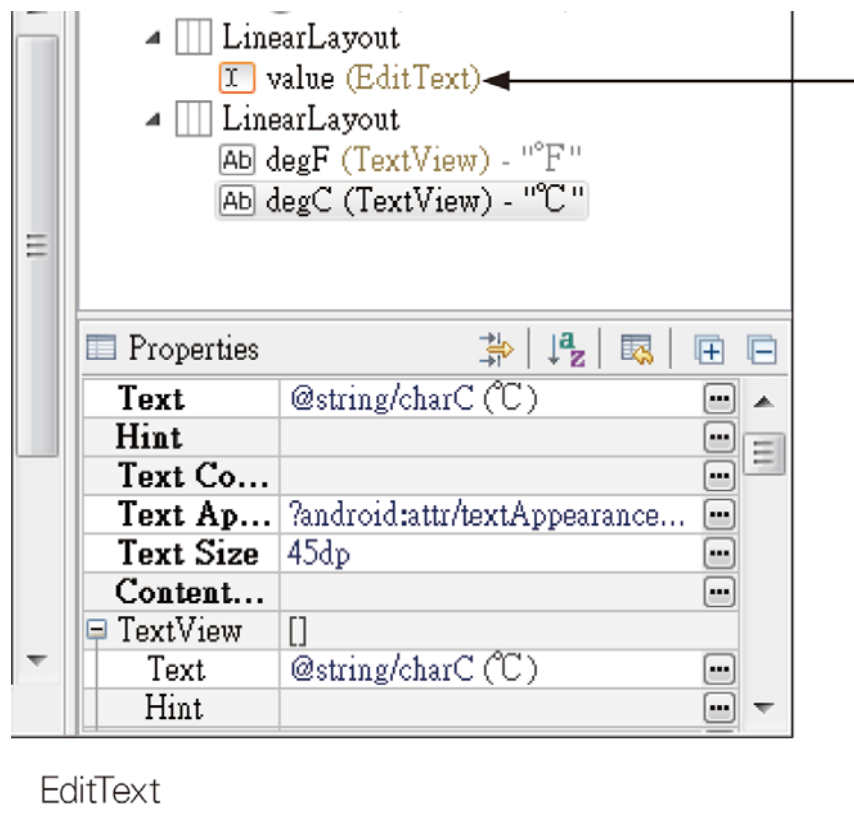
°F      °C

```
LinearLayout1  
├── LinearLayout  
│   ├── textView1 - "輸入溫度:"  
│   └── unit (RadioGroup)  
│       ├── unitF (RadioButton) - "華氏"  
│       └── unitC (RadioButton) - "攝氏"  
├── LinearLayout  
│   └── value (EditText)  
└── LinearLayout  
    ├── degF (TextView) - "°F"  
    └── degC (TextView) - "°C"
```

Properties

Text	@string/charC (°C)
Hint	
Text Co...	
Text Ap...	?android:attr/textAppearance...
Text Size	45dp
Content...	
TextView	
Text	@string/charC (°C)
Hint	

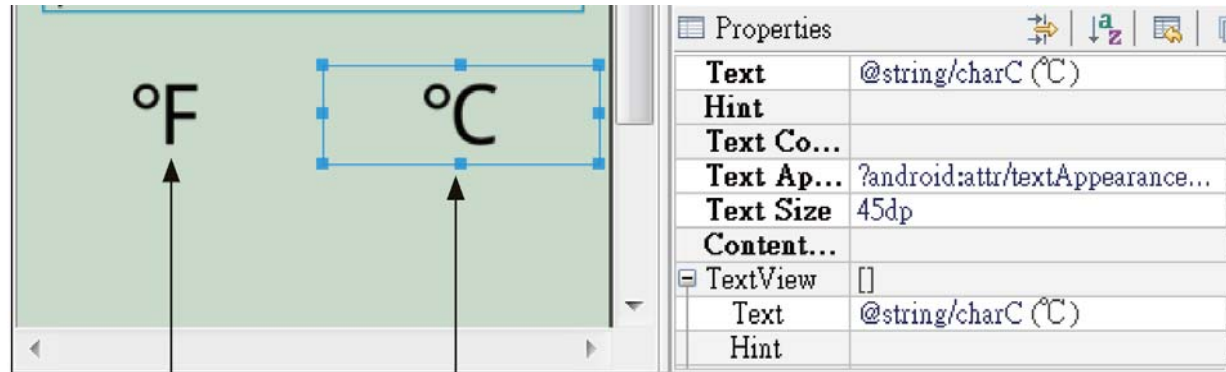
# 利用 RadioButton 選擇溫度轉換單位



EditText

Id	@+id/value
Single Line	true
Max Length	3
Input Type	numberSigned numberDecimal

# 利用 RadioButton 選擇溫度轉換單位



TextView

Id	@+id/degF
Text	@string/charF
TextView/Gravity	center
Max Lines	1
Text Size	45sp
Weidht	1

TextView

Id	@+id/degC
Text	@string/charC
TextView/Gravity	center
Max Lines	1
Text Size	45sp
Weidht	1

# 利用 **RadioButton** 選擇溫度轉換單位

```
01 public class MainActivity extends Activity
    implements OnCheckedChangeListener, TextWatcher {
02     RadioGroup unit;      ← 單選鈕群組欄位
03     EditText value;     ← 使用者輸入的溫度值欄位
04
05     @Override
06     protected void onCreate(Bundle savedInstanceState) {
07         super.onCreate(savedInstanceState);
```

因為有同名的  
介面，所以按  
**Ctrl + Shift + O**  
鍵時，ADT 會  
詢問要匯入哪一  
個套件，請參  
考下面圖解

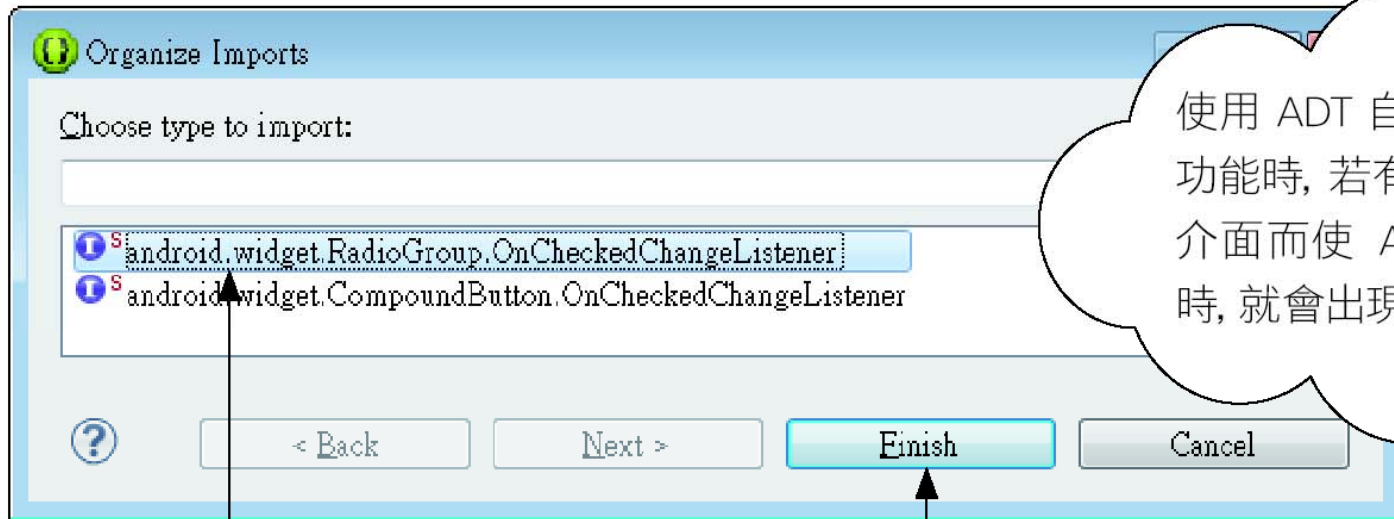
# 利用 **RadioButton** 選擇溫度轉換單位

## step 3

```
08     setContentView(R.layout.activity_main);
09
10     unit = (RadioGroup) findViewById(R.id.unit); ← 取得單選鈕群組
11     unit.setOnCheckedChangeListener(this); ← 設定 this 為監聽器
12
13     value = (EditText) findViewById(R.id.value); ← 取得輸入欄位
14     value.addTextChangedListener(this); ← 設定 this 為監聽器
15     }
...

```

# 利用 RadioButton 選擇溫度轉換單位



**1** 此處是要監聽 RadioGroup 的選取事件, 所以選此項

**2** 按 **Finish** 鈕完成匯入動作

# 利用 **RadioButton** 選擇溫度轉換單位

## step 4

```
01 public void afterTextChanged(Editable arg0) {
02     calc();
03 }
04
05 public void beforeTextChanged(CharSequence arg0, int arg1, int arg2,
06     int arg3) {
07     // TextWatcher 介面的方法, 此處不會用到
08 }
09
10 public void onTextChanged(CharSequence arg0, int arg1, int arg2, int arg3) {
11     // TextWatcher 介面的方法, 此處不會用到
12 }
13
14 public void onCheckedChanged(RadioGroup arg0, int arg1) {
15     calc();
16 }
```



# 利用 **RadioButton** 選擇溫度轉換單位

## step 5

```
01 protected void calc(){
02     // 取得文字方塊
03     TextView degF = (TextView) findViewById(R.id.degF);
04     TextView degC = (TextView) findViewById(R.id.degC);
05
06     double f=0, c=0; ← 儲存溫度值換算結果
07
08     if(unit.getCheckedRadioButtonId()==R.id.unitF){
09         f = Double.parseDouble(value.getText().toString());
10
11         c = (f-32)*5/9; ← 華氏 => 攝氏
                                ↑ 若選擇輸入華氏溫度
```

# 利用 `RadioButton` 選擇溫度轉換單位

```
12     }else{                                ← 若選擇輸入攝氏溫度
13         c = Double.parseDouble(value.getText().toString());
14
15         f = c*9/5+32;                    ← 攝氏 => 華氏
16     }
17
18     degC.setText(String.format("%.1f", c) + ← 只顯示到小數點後 1 位
19         getResources().getString(R.string.charC)); ← 從字串資源載入 °C 符號
20     degF.setText(String.format("%.1f", f) + ← 只顯示到小數點後 1 位
21         getResources().getString(R.string.charF)); ← 從字串資源載入 °F 符號
22 }
```

# 利用 `RadioButton` 選擇溫度轉換單位

轉換完成後, 傳回轉換後的字串

```
String.format("%.1f", x);
```



x 的值代入格式化參數, 並轉成指定的格式

例：

```
String.format("%.1f", 3.14159);
```

 → 傳回字串 "3.1"

```
String.format("%d 是質數", 17);
```

 → 傳回字串 "17 是質數"

↑ %d 代表要轉換的是整數

如果字串中要格式化多個變數, 可在格式化字串中列出各變數的格式化參數, 而各個變數也可一一列在 `format()` 方法的參數列中。

# 利用 RadioButton 選擇溫度轉換單位

## step 6

1 輸入 55 預設選『華氏』



溫度換算

輸入溫度:  華氏  攝氏

55

55.0°F 12.8°C

2 立即顯示換算後的攝氏溫度值

1 2 3 4  
@ # \$ %  
ALT ! " ' : ; , ? DEL  
英符 ?123 \_ . 完成

Detailed description: This screenshot shows the app interface with the input field containing '55'. The '華氏' (Fahrenheit) radio button is selected. The output area displays '55.0°F' and '12.8°C'. A dashed arrow points from the input '55' to the Celsius value. A callout box labeled '2' points to the Celsius value, stating '立即顯示換算後的攝氏溫度值' (Immediately display the converted Celsius temperature value).

3 改選『攝氏』



溫度換算

輸入溫度:  華氏  攝氏

55

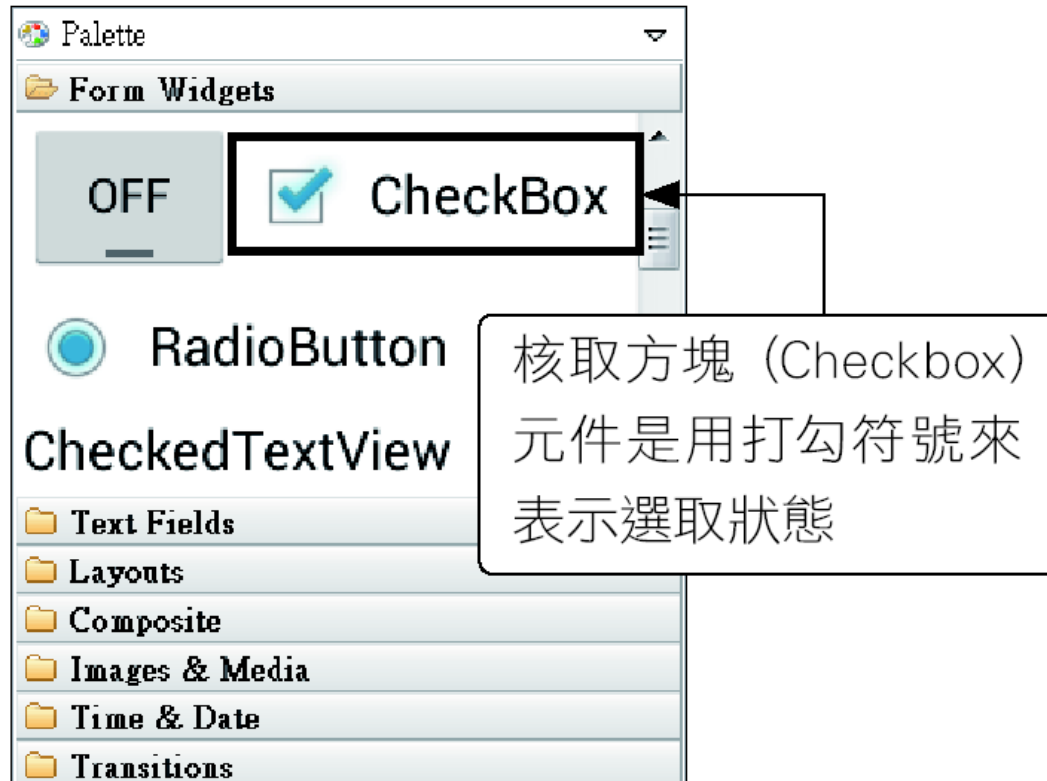
131.0°F 55.0°C

4 立即顯示換算後的華氏溫度值

1 2 3 4 5  
@ # \$ % "  
ALT ! " ' : ; , ? DEL  
英符 ?123 \_ . 完成

Detailed description: This screenshot shows the app interface after the user has selected the '攝氏' (Celsius) radio button. The input field still contains '55'. The output area now displays '131.0°F' and '55.0°C'. A dashed arrow points from the input '55' to the Fahrenheit value. A callout box labeled '4' points to the Fahrenheit value, stating '立即顯示換算後的華氏溫度值' (Immediately display the converted Fahrenheit temperature value).

## 5-2 可複選的核取方塊 (CheckBox)



# 可複選的核取方塊 (CheckBox)

- `isChecked()`：檢查是否被選取
- `onCheckedChanged()`：選取/取消核取方塊的事件

# isChecked()：檢查是否被選取

漢堡 ← isChecked() 傳回 true

可樂 ← isChecked() 傳回 false

# 範例5-3：以核取方塊建立餐點選單

1 可任意點選/  
取消餐點



2 按訂購鈕時，程式  
會列出所點的餐點



# 以核取方塊建立餐點選單

step 1

step 2

1 加入 2 個 Linear Layout (Horizontal),  
並設定 Margin 屬性為 5dp

設計完成後, 記得按  
**Ctrl + S** 存檔

2 在 Linear Layout 中分別  
加入 4 個 CheckBox, 並  
設定屬性

Id	@+id/chk1、@+id/chk2、 @+id/chk3、@+id/chk4
Text	漢堡、薯條、可樂、玉米濃湯
Text Size	30sp

4 最下方加入文字方塊, 並設定屬性

Id	@+id/showOrder
Text	(空白)
Text Size	30sp
Width	match_parent

3 加入按鈕, 並設定屬性

Text	訂購
On Click	takeOrder
Text Size	30sp
Width	match_parent

# 以核取方塊建立餐點選單

## step 3

```
01 public void takeOrder(View v) {
02     CheckBox chk;
03     String msg="";           ← 存放要顯示在畫面上的文字訊息
04     // 用陣列存放所有 CheckBox 元件的 ID
05     int[] id={R.id.chk1, R.id.chk2, R.id.chk3, R.id.chk4};
06
07     for(int i:id){          ← 以迴圈逐一檢視各 CheckBox 是否被選取
08         chk = (CheckBox) findViewById(i);
09         if(chk.isChecked()) ← 若有被選取
10             msg+="\n"+chk.getText(); ← 將換行字元及選項文字
11     }                       ← 附加到 msg 字串後面
```

# 以核取方塊建立餐點選單

```
12
13     if(msg.length()>0) ← 有點餐 (字串長度大於0)
14         msg ="你點購的餐點是："+msg;
15     else
16         msg ="請點餐!";
17
18     // 在文字方塊中顯示點購的項目
19     ((TextView) findViewById(R.id.showOrder)).setText(msg);
20 }
```

# 以核取方塊建立餐點選單

## step 4

1 任意選取項目



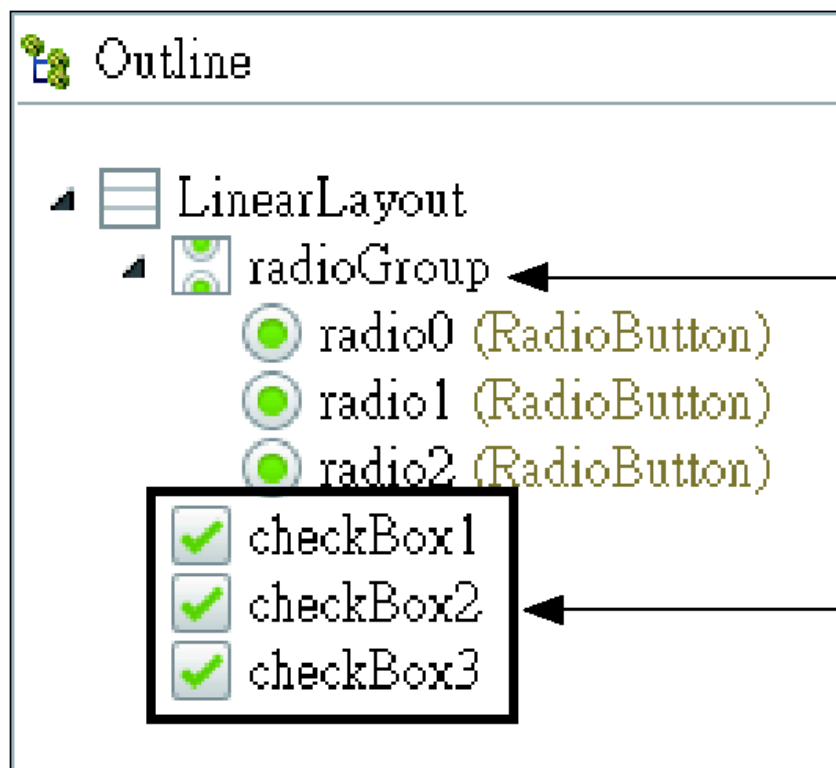
2 按此鈕



可繼續重複前面  
步驟進行測試

點選的項目  
會列在此處

# onCheckedChanged()：選取/取消核 取方塊的事件



只要對 RadioGroup 物件設定監聽物件，即可監聽各 RadioButton 的選取動作

每個 CheckBox 物件都要『個別設定』監聽物件

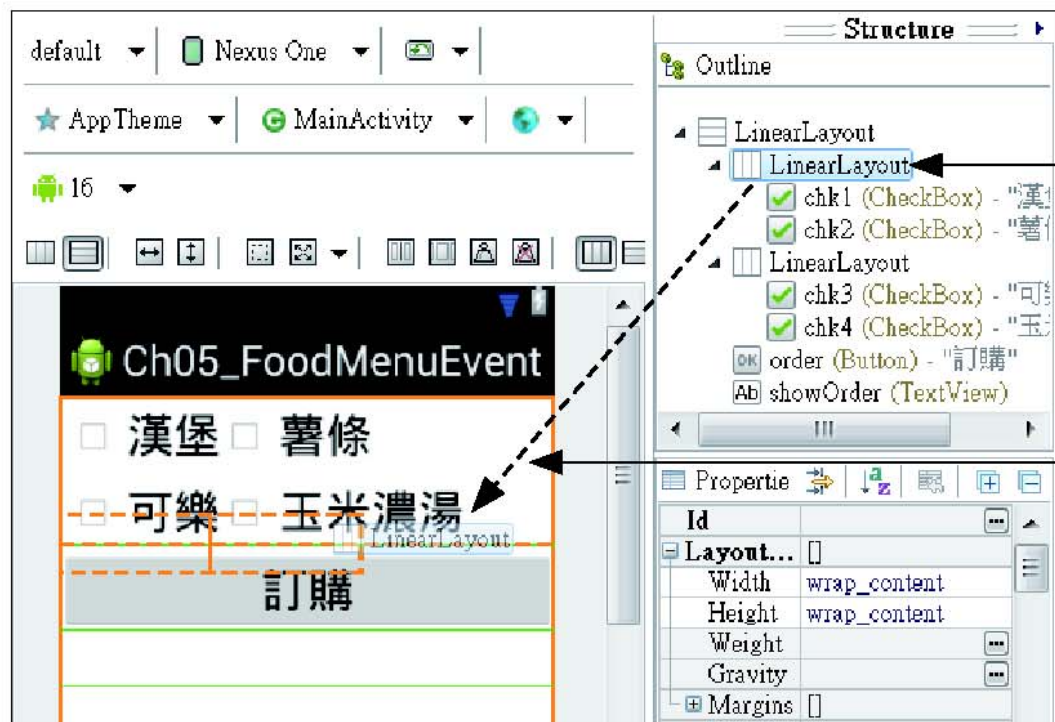
# onCheckedChanged() : 選取/取消核 取方塊的事件



# 範例5-4：利用選取事件即時修改訂單

step 1

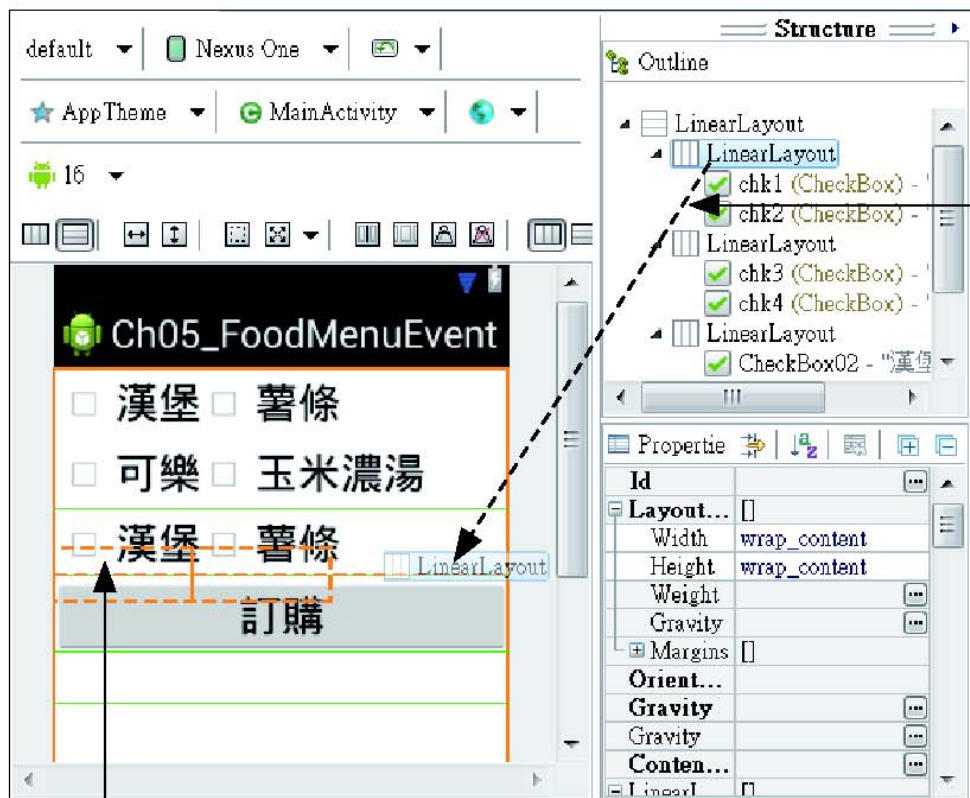
step 2



1 用滑鼠選取包含 2 個 CheckBox 的 Linear Layout (Horizontal)

2 按住 **Ctrl** 鍵不放，將選取的 Linear Layout 拉曳到畫面中、按鈕之上放開

# 利用選取事件即時修改訂單



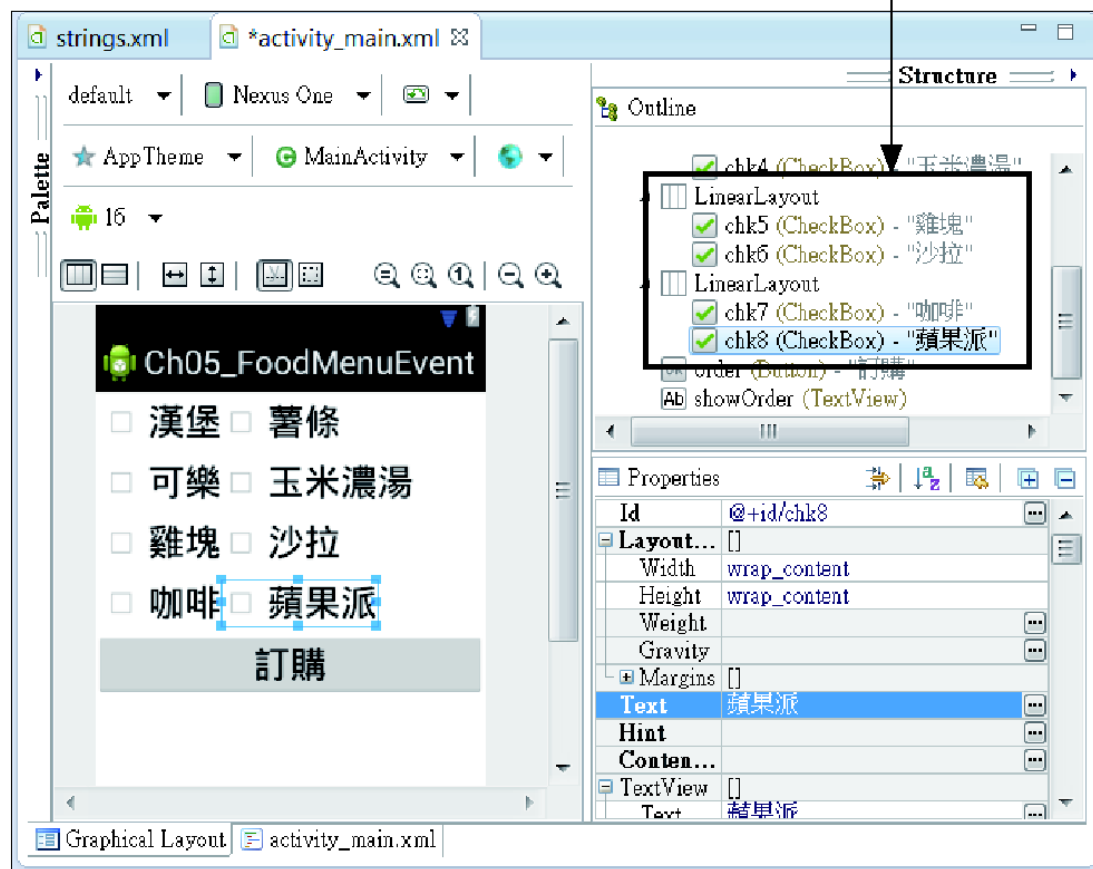
3 再依相同方式加入另一列元件

產生一系列相同的元件



# 利用選取事件即時修改訂單

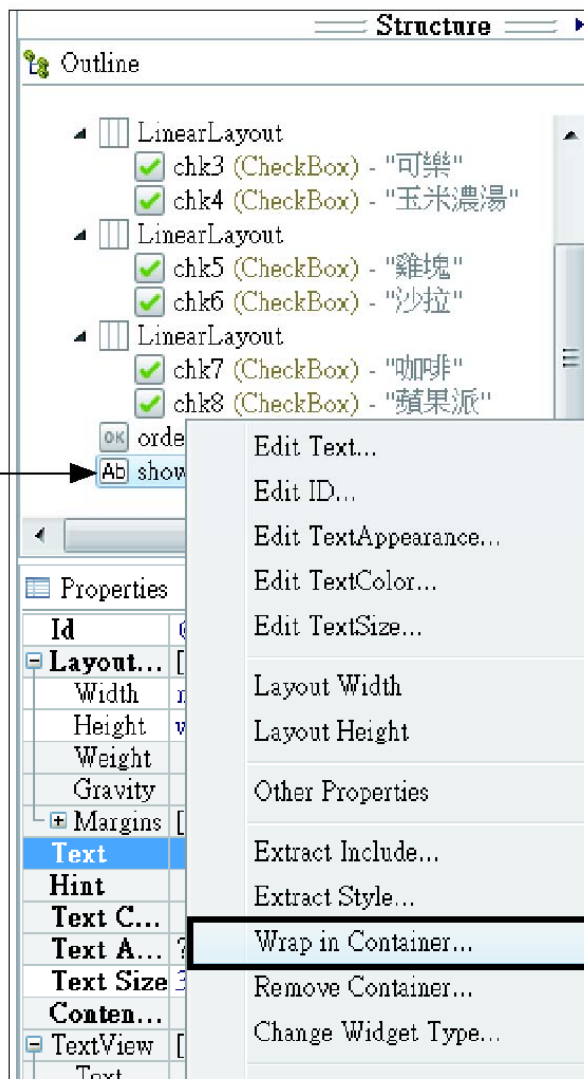
- 4 將新加入的 CheckBox 的 Id 及 Text 屬性修改成如圖所示的內容



# 利用選取事件即時修改訂單

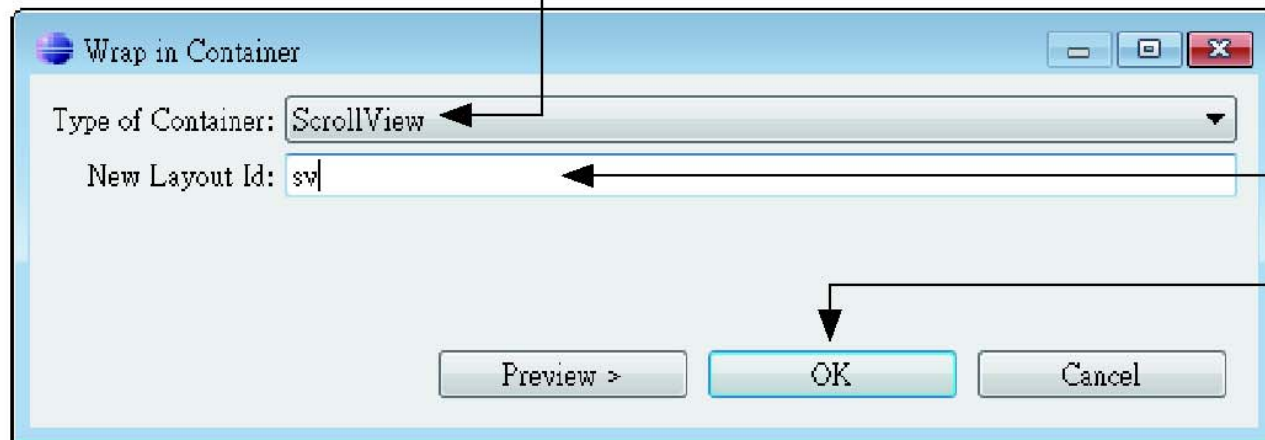
## step 3

- 1 在 showOrder (TextView) 上按滑鼠右鈕, 執行『Wrap in Container』命令 (包在容器內)



# 利用選取事件即時修改訂單

- 2** 選 ScrollView (在下拉選單較下方, 若找不到可按 **S** 鍵, 反白項目就會跳到以 "S" 開頭的項目)

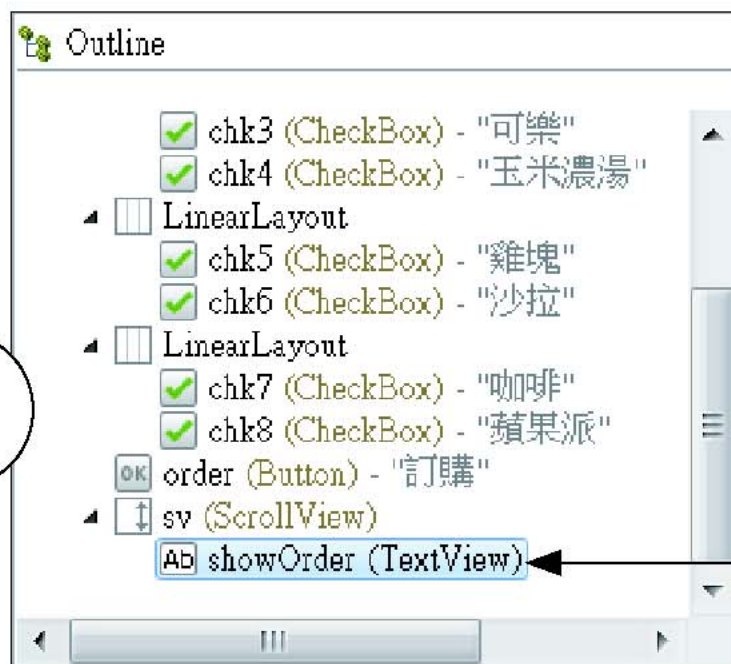


- 3** 輸入任意 Id 名稱

- 4** 按 OK 鈕

# 利用選取事件即時修改訂單

修改完成後，請按  
**Ctrl** + **S** 存檔



TextView 現在是包  
在 ScrollView 元件中

# 利用選取事件即時修改訂單

## step 4

```
01 public class MainActivity extends Activity
                                implements OnCheckedChangeListener {
02
03     @Override
04     protected void onCreate(Bundle savedInstanceState) {
05         super.onCreate(savedInstanceState);
06         setContentView(R.layout.activity_main);
07
```

# 利用選取事件即時修改訂單

```
08      // 所有核取方塊 ID 的陣列
09      int[] chk_id={R.id.chk1, R.id.chk2, R.id.chk3, R.id.chk4,
10                  R.id.chk5, R.id.chk6, R.id.chk7, R.id.chk8};
11
12      for(int id:chk_id){ ← 用迴圈替所有核取方塊註冊監聽物件
13          CheckBox chk=(CheckBox) findViewById(id);
14          chk.setOnCheckedChangeListener(this);
15      }
16  }
...

```

# 利用選取事件即時修改訂單

## step 5

```
01 // 用來儲存已選取項目的集合物件
02 ArrayList<CompoundButton> selected=new ArrayList<CompoundButton>();
03
04 public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
05
06     if (isChecked==true)           ← 若項目被選取
07         selected.add(buttonView); ← 加到集合之中
08     else                             ← 若項目被取消
09         selected.remove(buttonView); ← 自集合中移除
10 }
```

# 利用選取事件即時修改訂單

## step 6

```
01 public void takeOrder(View v) {
02     String msg=""; ← 儲存訊息的字串
03
04     for(CompoundButton chk:selected) ← 以迴圈逐一將換行字元及
05         msg += "\n"+chk.getText(); ← 選項文字附加到 msg 字串後面
06
07     if(msg.length()>0) ← 有點餐
08         msg ="你點購的餐點是："+msg;
09     else
10         msg ="請點餐!";
11
12     // 在文字方塊中顯示點購的項目
13     ((TextView) findViewById(R.id.showOrder)).setText(msg);
14 }
```



# 利用選取事件即時修改訂單

## step 7

The diagram illustrates the process of selecting items and scrolling the order list in an Android application. It consists of two screenshots of the 'FoodMenu 事件處理' screen, connected by a right-pointing arrow.

**Step 1:** 選取餐點 (Select items). The first screenshot shows a list of items with checkboxes: 漢堡 (checked), 薯條 (checked), 可樂 (unchecked), 玉米濃湯 (unchecked), 雞塊 (unchecked), 沙拉 (checked), 咖啡 (checked), and 蘋果派 (checked). A bracket on the left indicates that items 1 through 4 are selected.

**Step 2:** 按此鈕 (Press this button). An arrow points to the '訂購' (Order) button at the bottom of the list.

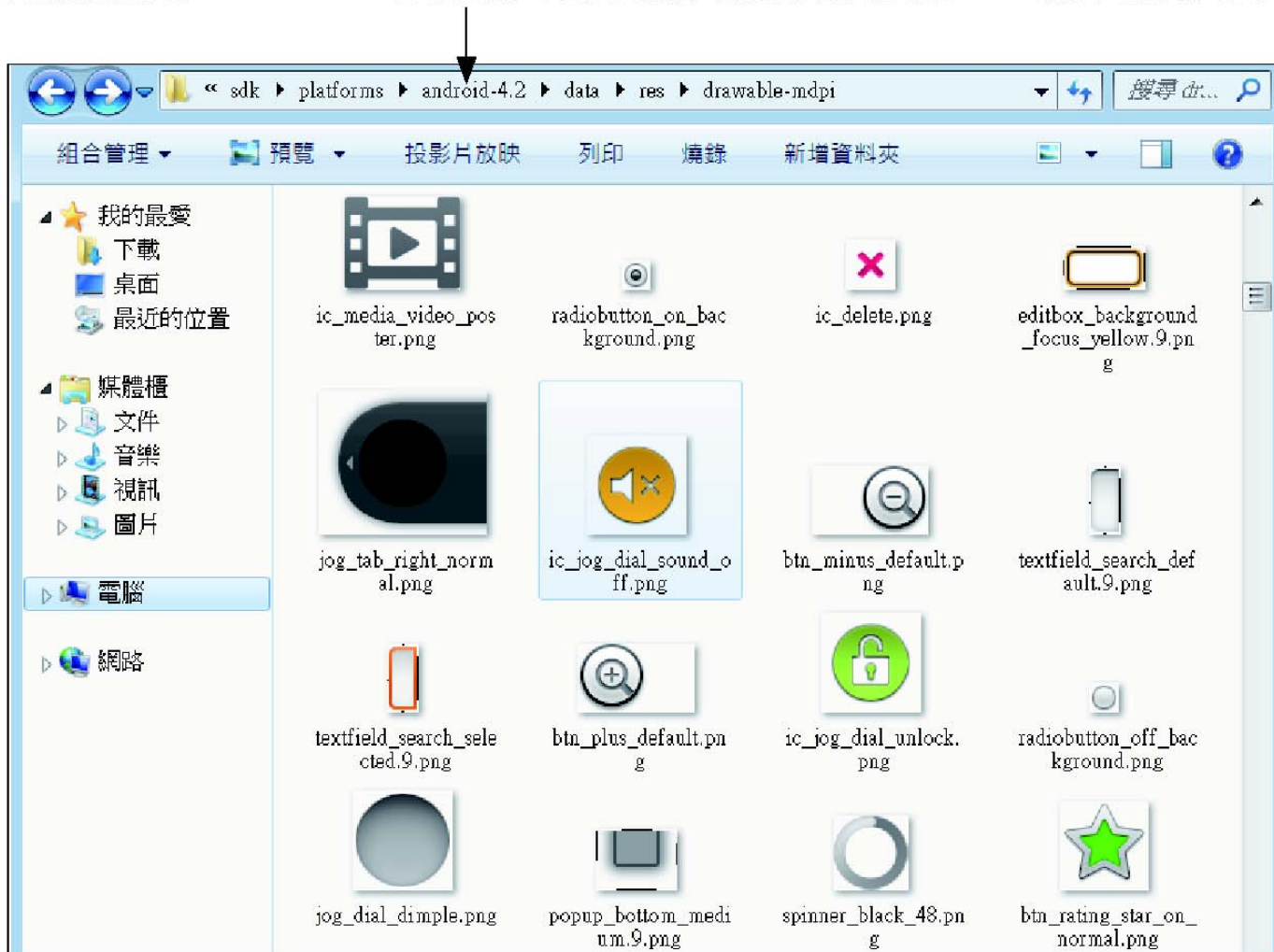
**Step 3:** 若內容超出畫面範圍，可按住文字方塊上下捲動 (If content exceeds screen range, you can scroll the text block up and down). The second screenshot shows the '訂購' button and a list of items below it: 漢堡, 蘋果派, 薯條, and 咖啡. A callout box on the right explains that if the content exceeds the screen range, the user can scroll the text block up and down.

## 5-3 顯示圖形的 ImageView

- 使用 Android 系統內建的圖形資源
- 使用非 Android 內建(自行提供)的圖形資源
- 圖形的縮放控制

# 使用 Android 系統內建的圖形資源

本例檢視的是 android-4.2 中的資源 (您的電腦可能會看到其它 SDK 版本的資料夾名稱)

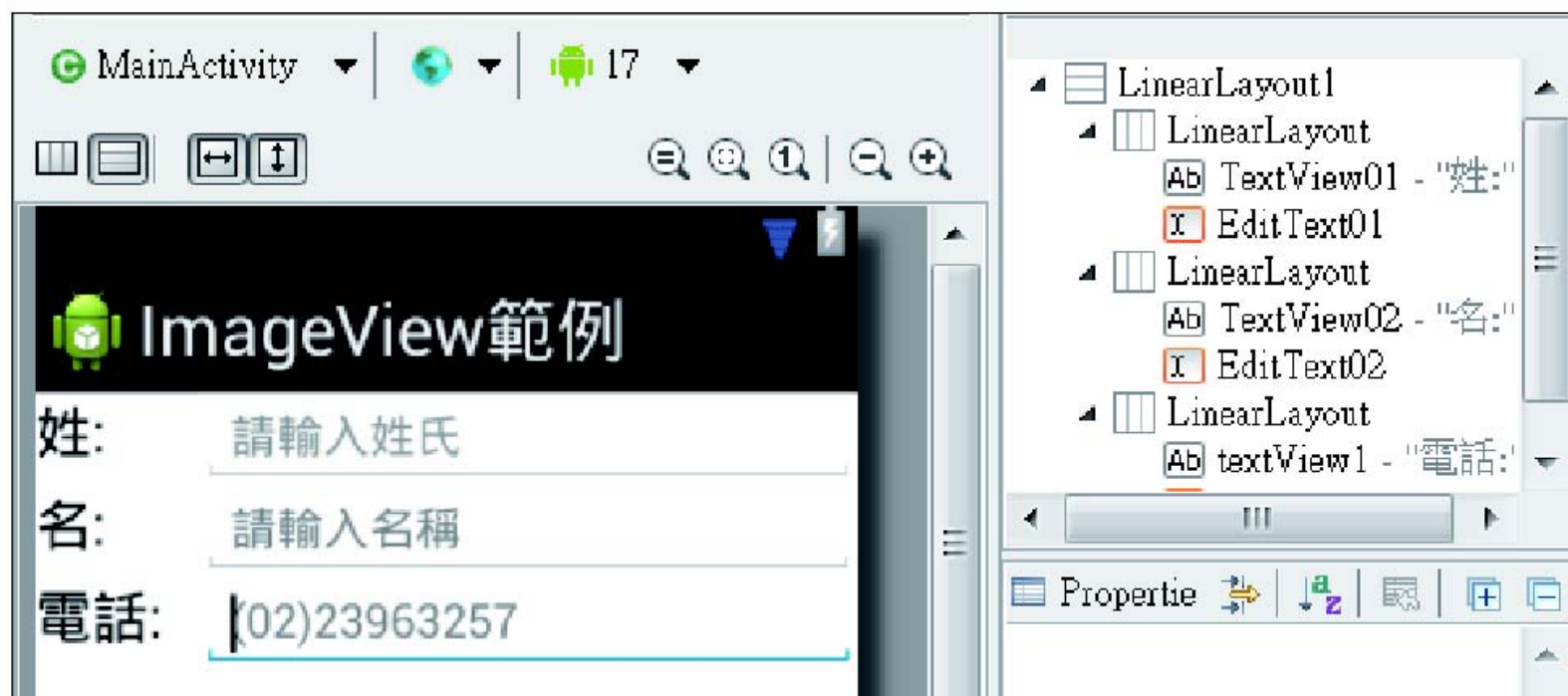


# 範例5-5：顯示系統內建圖形



# 顯示系統內建圖形

## step 1



# 顯示系統內建圖形

## step 2

1 展開 Images & Media 項目



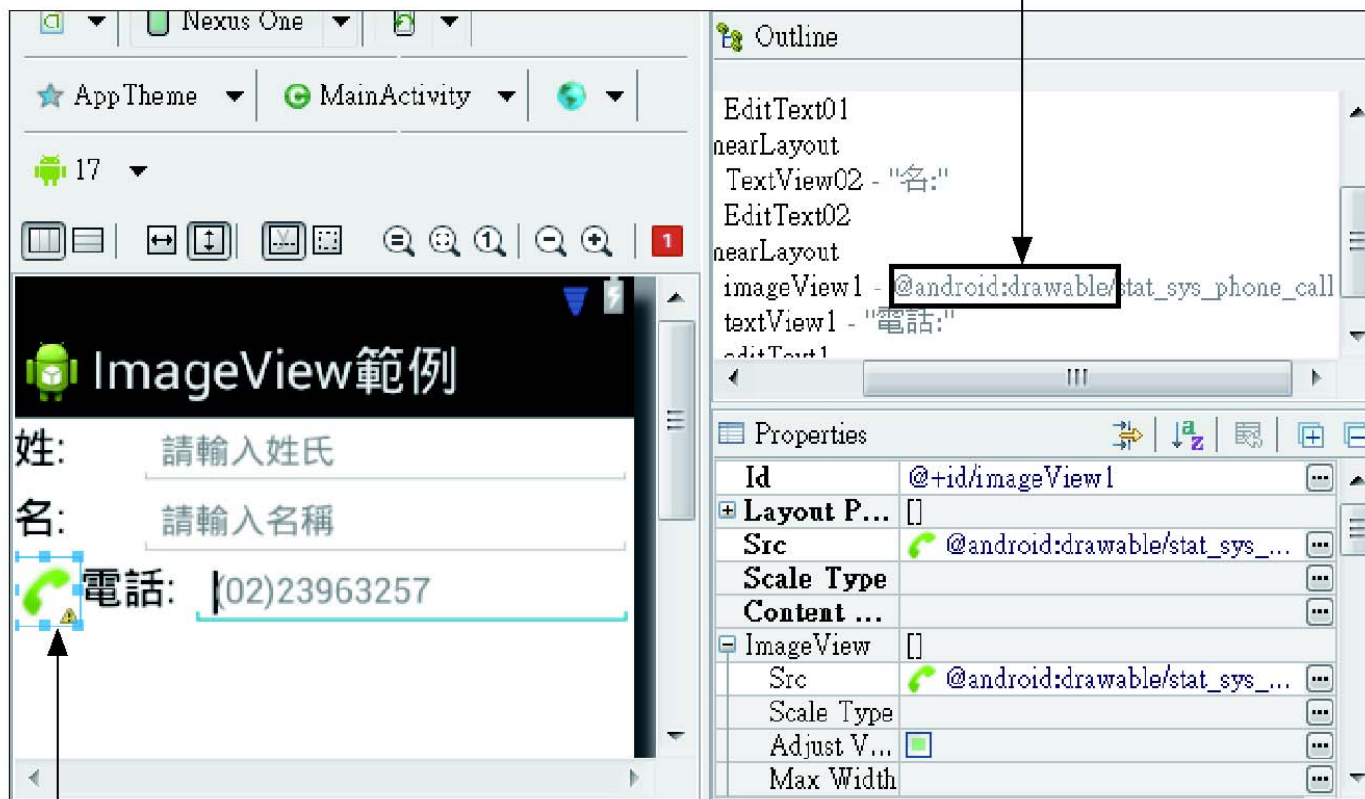
# 顯示系統內建圖形

4 可在此輸入關鍵字進行篩選，本例輸入 "\*phone" (\* 表示萬用字元，"phone" 代表任何名稱含有 "phone" 的內建圖形)



# 顯示系統內建圖形

內建圖形資源是以 "@android:drawable/" 開頭的格式來指定

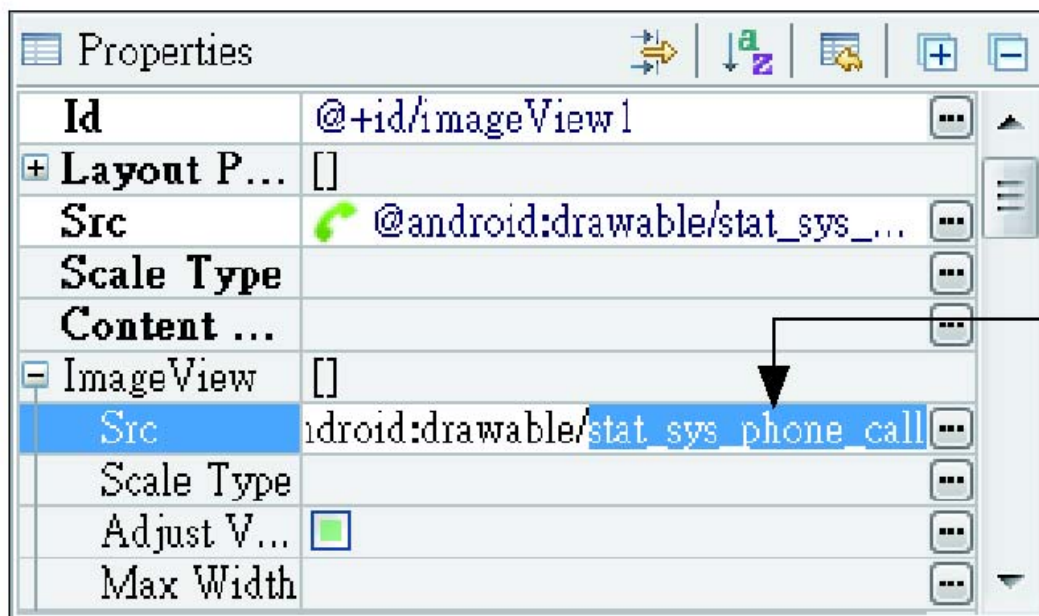


7 可用滑鼠拉曳調整大小, 或在屬性窗格設定



# 顯示系統內建圖形

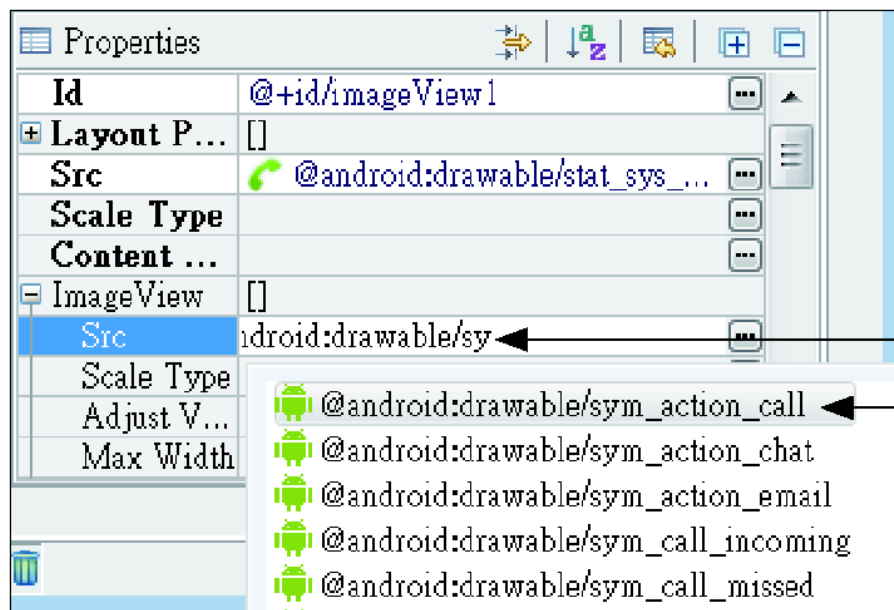
## step 3



1 用滑鼠選取 Src 欄位  
"/" 之後的內容

2 按 **Delete** 鍵

# 顯示系統內建圖形

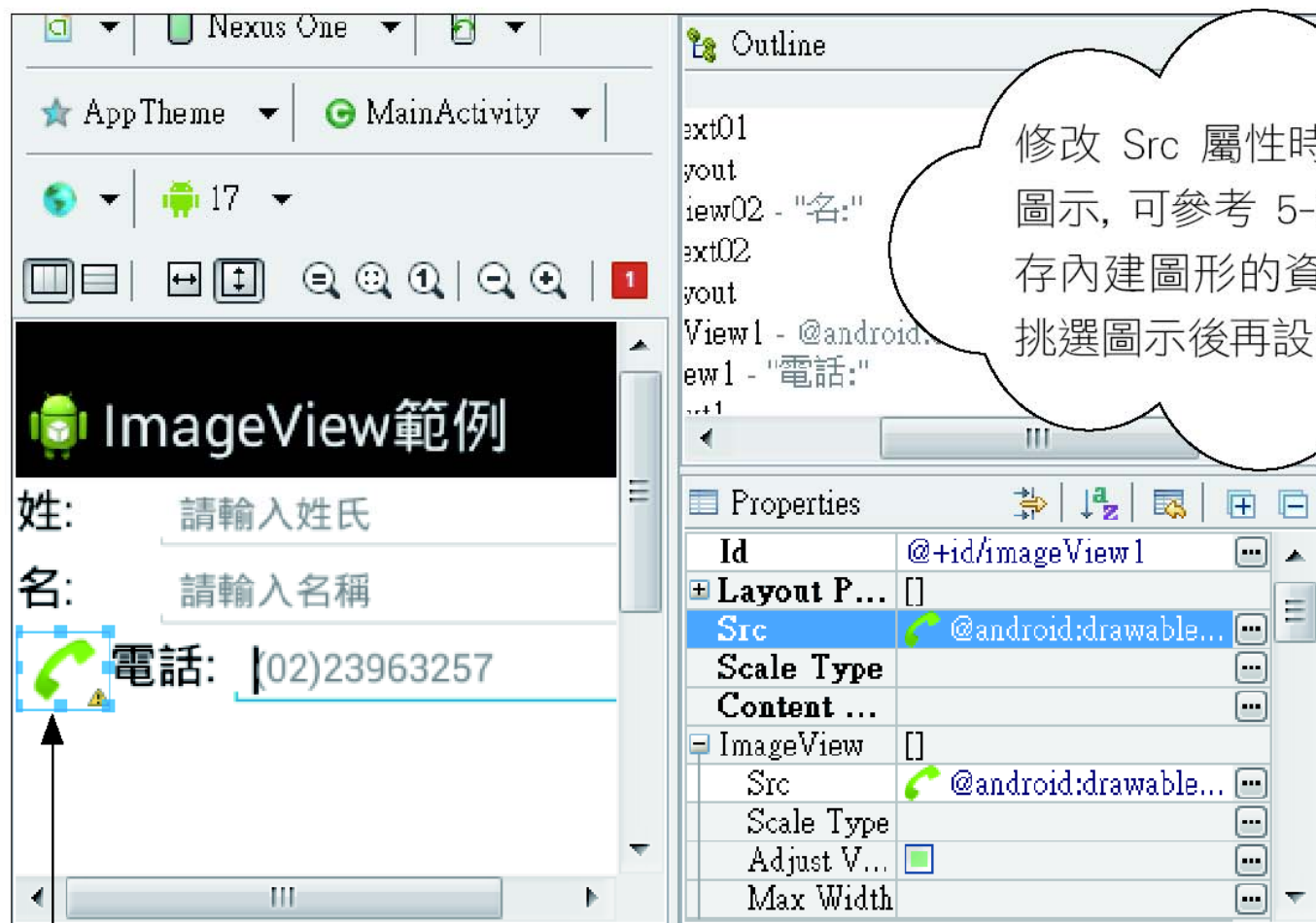


**3** 輸入 "sy" 篩選內容

**4** 雙按 "sym\_action\_call"

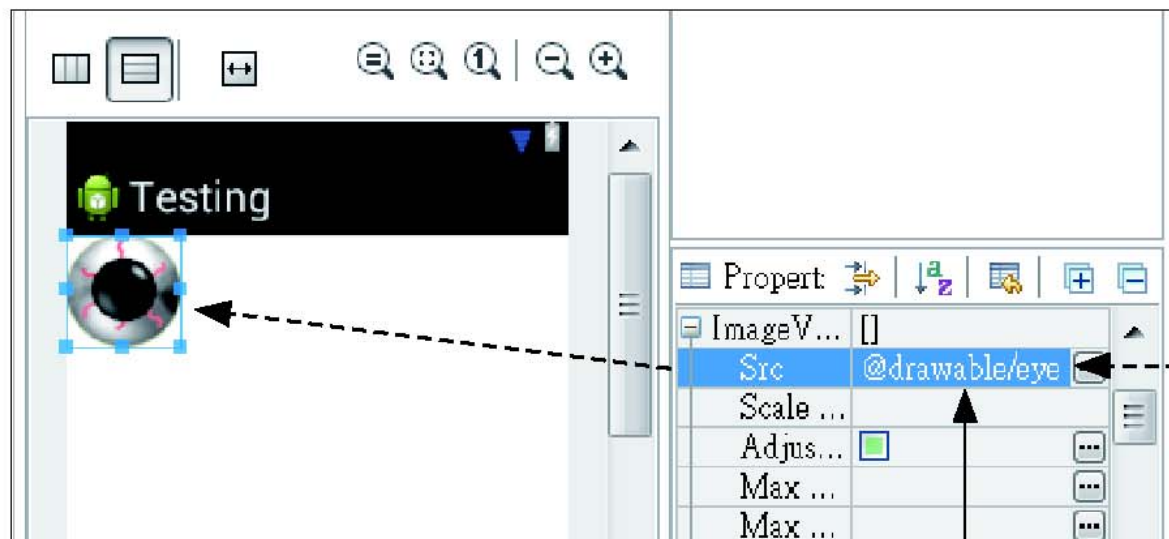
下拉清單會列出  
所有可選的資源

# 顯示系統內建圖形



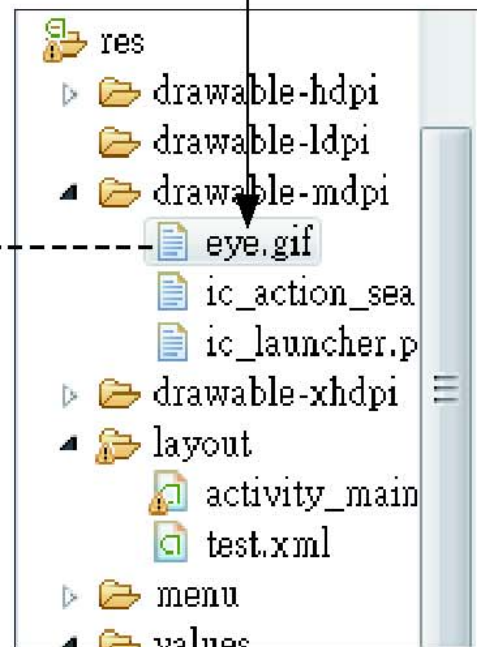
另一個 "電話" 圖示

# 使用非 Android 內建(自行提供)的圖形資源

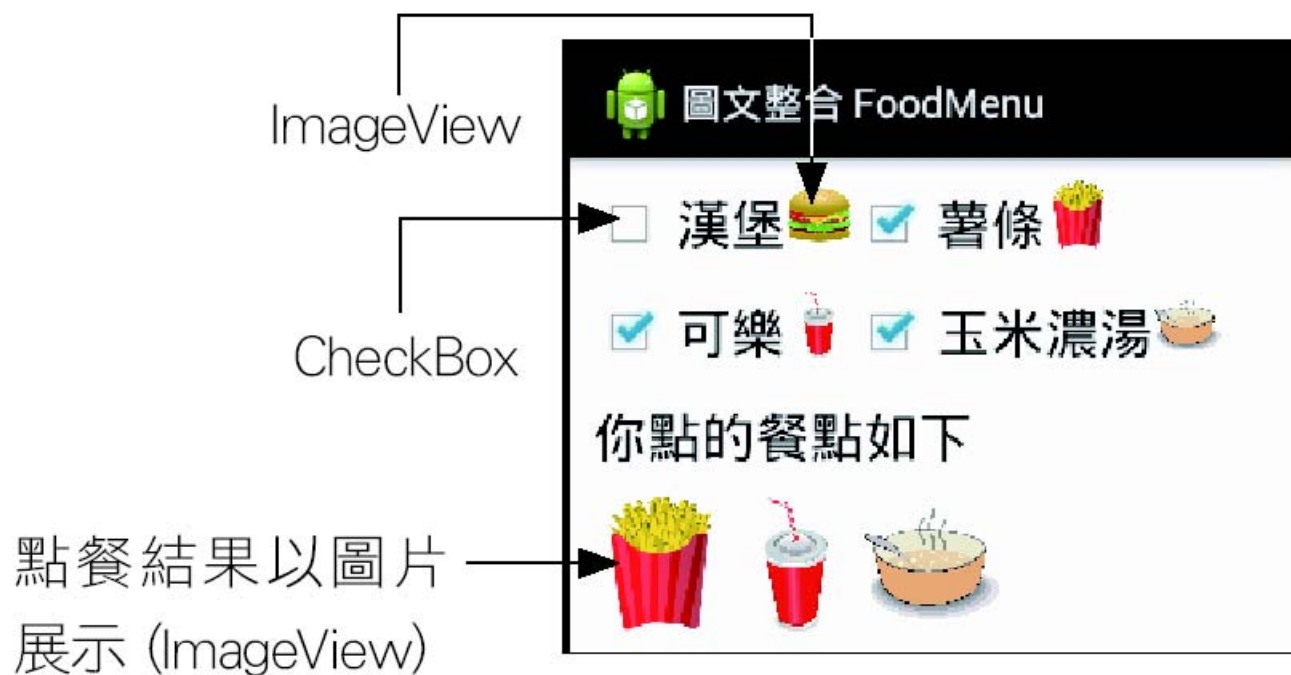


Src 之屬性值設為 "@drawable/圖檔主檔名"

可使用 gif、jpg、png 格式的檔案



# 範例5-6：替選單加上圖片



# 替選單加上圖片

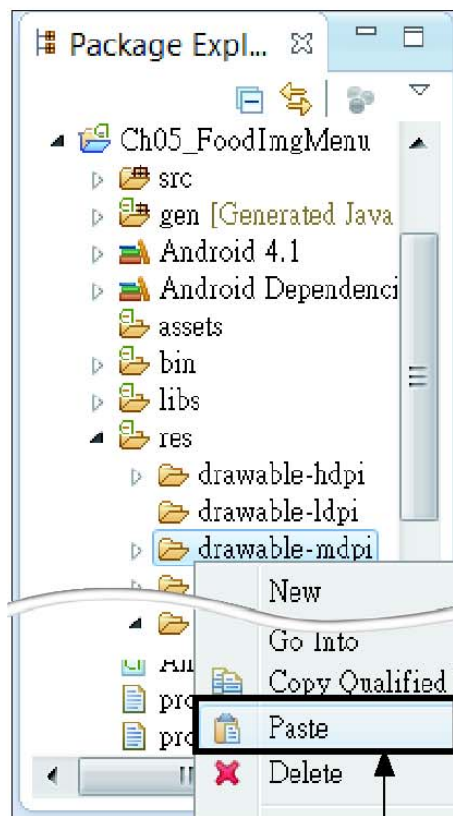
step 1

step 2

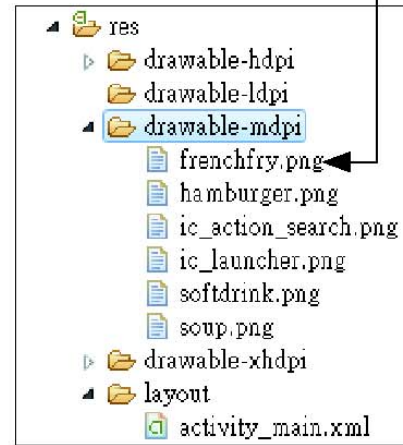
您可在光碟的範例資料夾 Ch05\_FoodImgMenu\res\drawable-mdpi 資料夾下找到圖檔



1 在檔案總管中選取圖檔後，按右鈕執行『複製』命令



圖檔主檔名就是資源名稱



2 在新專案的 res/drawable-mdpi 上按滑鼠右鈕，執行『Paste』命令貼上檔案

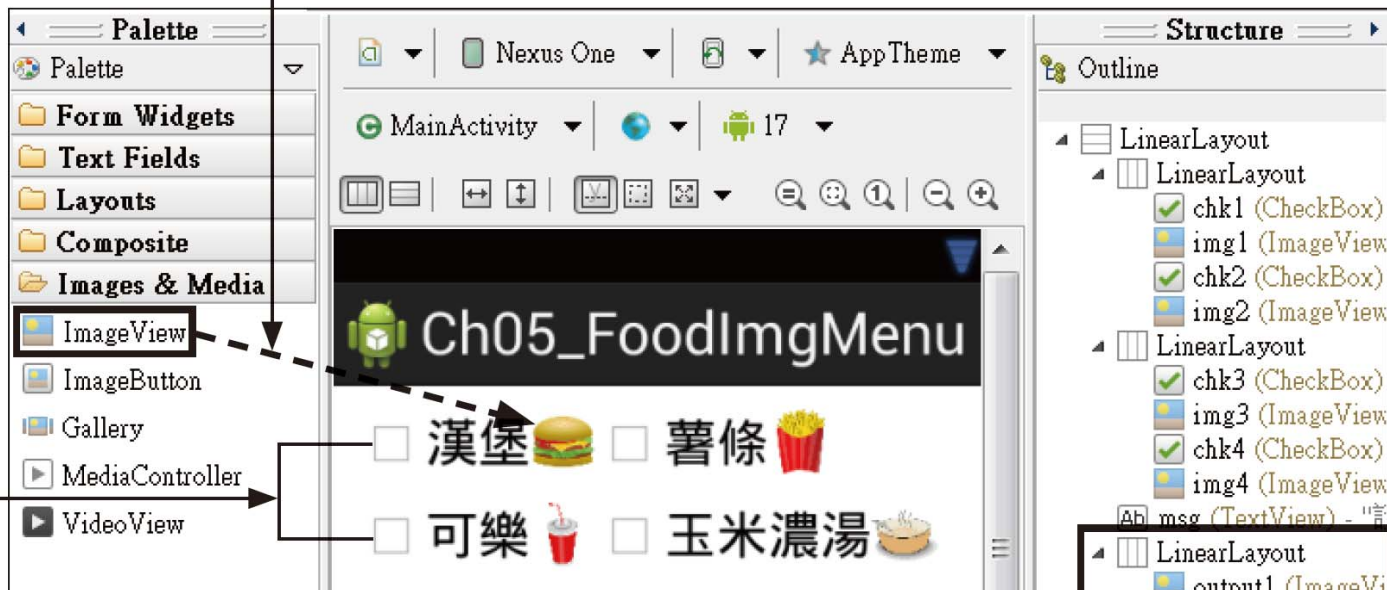
# 替選單加上圖片

## step 3

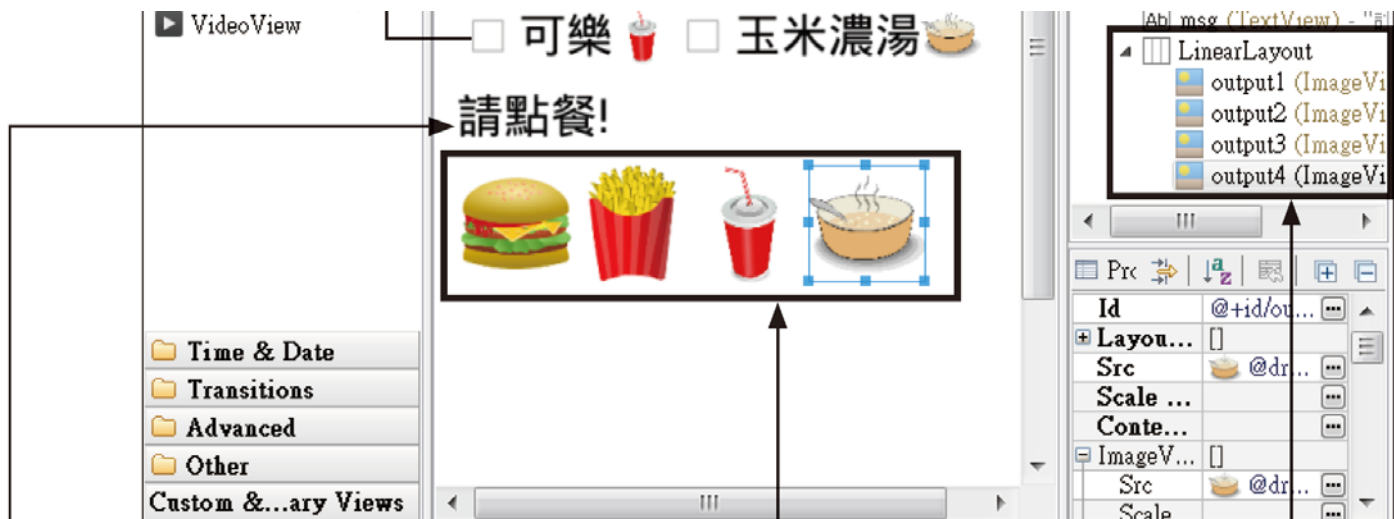
2 在每個 CheckBox 後面加入 ImageView, 並設定屬性

Src	@drawable/hamburger、@drawable/frenchfry、 @drawable/softdrink、@drawable/soup
Width	30dp
Height	30dp

1 加入 4 個  
CheckBox 並  
設定如圖的  
Id 和文字



# 替選單加上圖片



**3** 加入 TextView,  
並設定屬性

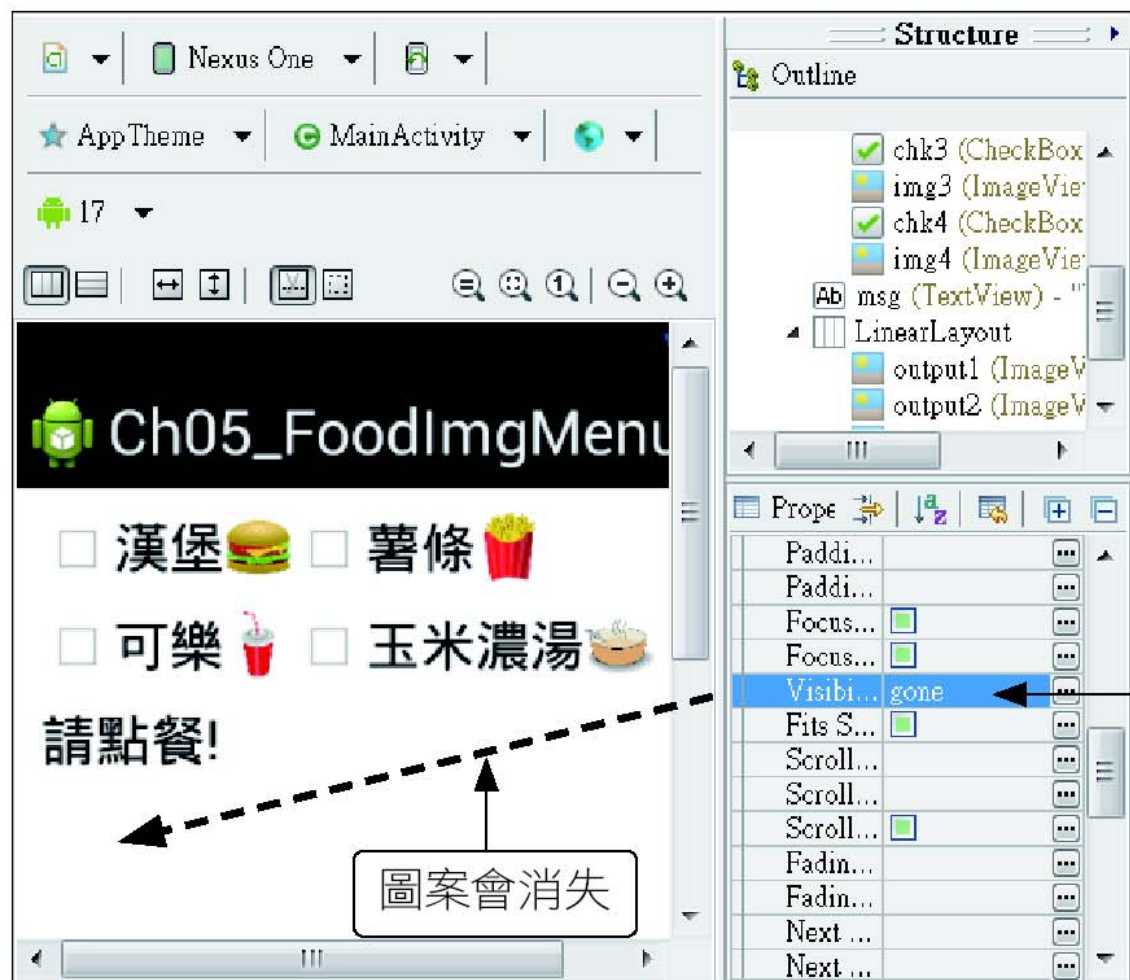
Id	@+id/showOrder
Text	請點餐!
Text Size	24sp

**4** 加入 Linear Layout (Horizontal), 在其中加入 4 個 ImageView  
並設定屬性

Id	@+id/output1、@+id/output2、@+id/output3、 @+id/output4
Src	@drawable/hamburger、@drawable/frenchfry、 @drawable/softdrink、@drawable/soup
Width	60dp
Height	60dp



# 替選單加上圖片



**5** 將下面 4 個 ImageView 的 Visibility 屬性都設為 gone

# 替選單加上圖片

## step 4

```
01 public class MainActivity extends Activity
02         implements OnCheckedChangeListener {
03     @Override
04     protected void onCreate(Bundle savedInstanceState) {
05         super.onCreate(savedInstanceState);
06         setContentView(R.layout.activity_main);
07
08         int[] chk_id={R.id.chk1,R.id.chk2,R.id.chk3,R.id.chk4};
09         for (int id:chk_id)    ← 為每個 CheckBox 註冊監聽物件
10             ((CheckBox) findViewById(id)).setOnCheckedChangeListener(this);
```

# 替選單加上圖片

```
11     }
12
13     int items=0;    ← 記錄目前選取餐點的數量
14     public void onCheckedChanged(CompoundButton chk,
                                   boolean isChecked) {
15         int visible;    ← CheckBox 顯示狀態
16         if(isChecked){ ← 被選取時
17             items++;    ← 數量加 1
18             visible=View.VISIBLE; ←
                                   應使用的 Visibility 屬性值為 VISIBLE (可見)
```

Next

# 替選單加上圖片

```
19     }
20     else{           ← 被取消時
21         items--;   ← 數量減 1
22         visible=View.GONE; ← 應使用的 Visibility 屬性值為 GONE (不可見)
23     }
24
25     switch (chk.getId()){ ← 依選取項目的資源 ID,
26                                     決定要更改的 ImageView
27     case R.id.chk1:
28         findViewById(R.id.output1).setVisibility(visible);
29         break;
30     case R.id.chk2:
31         findViewById(R.id.output2).setVisibility(visible);
32         break;
33     case R.id.chk3:
```

# 替選單加上圖片

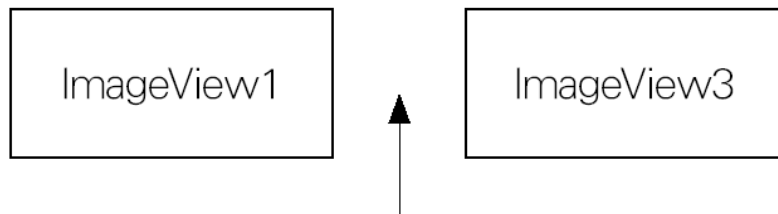
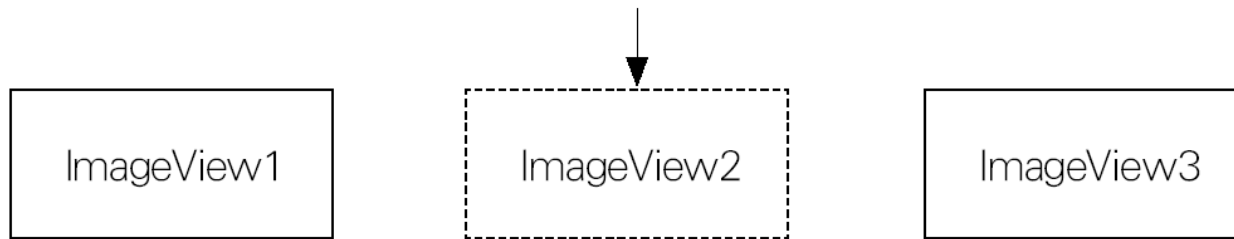
```
33         findViewById(R.id.output3).setVisibility(visible);
34         break;
35     case R.id.chk4:
36         findViewById(R.id.output4).setVisibility(visible);
37         break;
38     }
39
40     String msg;
41     if(items>0) ← 選取項目大於 0, 顯示有點餐的訊息
42         msg= "你點的餐點如下:";
43     else ← 否則顯示請點餐的訊息
44         msg= "請點餐!";
45     ((TextView) findViewById(R.id.showOrder)).setText(msg);
46 }
47 ...
```

# 替選單加上圖片

setVisibility() 參數值	Visibility 屬性值	說明
VIEW.VISIBLE	visible	預設值, 表示元件會正常顯示。
VIEW.INVISIBLE	invisible	元件看不到, 但會占用佈局空間, 例如 3 個 TextView 排在一起, 若中間的被設為 VIEW.INVISIBLE, 則其前後 TextView 之間將留白。
VIEW.GONE	gone	元件看不到, 也『不占用』佈局空間, 同樣 3 個 TextView 排列的情況, 若中間的被設為 VIEW.GONE, 則其前後 TextView 將會變成相鄰的元件。

# 替選單加上圖片

ImageView2 的 Visibility 屬性為 INVISIBLE, 元件會隱藏但會佔用原有空間



TextView2 的 Visibility 屬性為 GONE, 元件會隱藏且不佔用空間

# 替選單加上圖片

## step 5

1 選取任意項目



立即出現對應的圖案

2 取消已勾選的項目



被取消的項目圖案  
會立即消失



# 圖形的縮放控制

