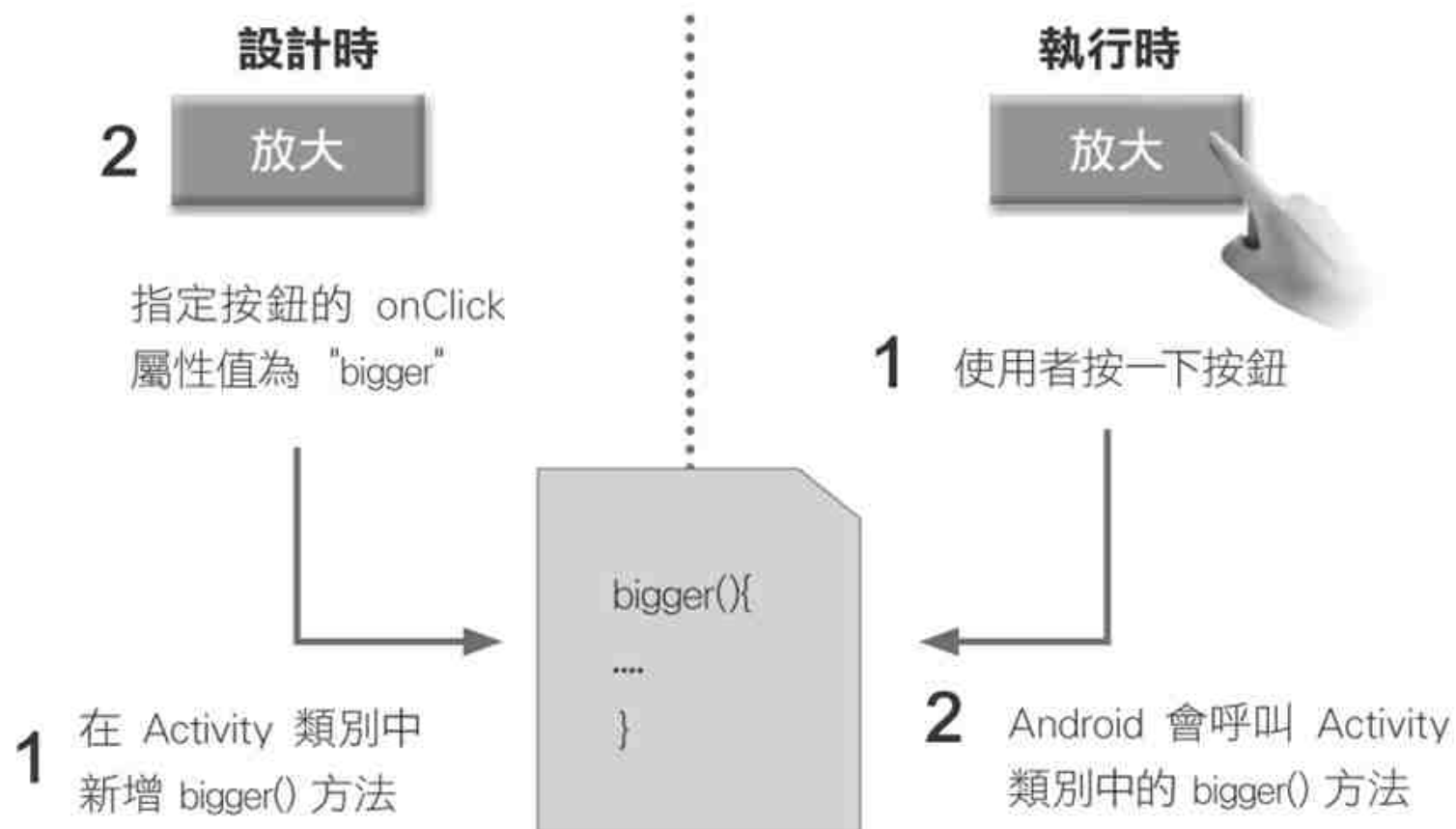


1102課程大綱

- 擴充上週第一個程式範例
- 第二個程式範例
- 隨堂測驗
- 事件處理機制
- 處理按一下事件
- 處理長按事件

程式範例一



擴充上週第一個程式範例

- TextView中的文字大小預設為20 sp 。
- 同時有放大及縮小按鈕。
- 按放大鈕字體增加1 sp 。
- 按放大鈕字體縮小1 sp 。
- TextView上的字體最大為 40 sp ，最小為10 sp 。

第二個程式範例 (1)



第二個程式範例 (2)



第二個程式範例 (3)

- 請參考P.2-51~P.2-55 Step 1到Step 5的操作。
- 在onCreate()中要加入的程式碼，請參考P. 2-55 Step 5的下面。

擴充第二個程式範例

- 若在可編輯文字框中輸入的內容為空白，則在文字顯示框中顯示“請輸入大名！”。
- ✓ 主要的程式碼請見 P. 2-56 下面。
- ✓ 第5行的trim()方法是Java String類別內建的方法，它會除去String物件裡前後的空白字元，再將字串傳回。

事件處理機制(1)

- 先前我們已經介紹過可透過在視覺元件上設定onClick屬性來處理按一下按鈕的事件，但這樣的方法卻無法處理長按等其它事件。
- 本週課程將學習可以處理各種事件的標準方法，即“事件處理機制”。

事件處理機制(2)

- 當使用者對手機進行各種操作時，會產生對應的事件(Event)。
- 例如當使用者按一下按鈕時，會產生onClick事件：



事件處理機制(3)

- 事件處理機制：
 - 事件發生的來源，例如某個按鈕，稱為來源物件。
 - 我們必須準備一個能處理該事件的監聽物件 (Listener)，並登錄到來源物中。
 - 當來源物件有事件發生時，就會自動呼叫監聽物件中對應該事件的處理方法來進行處理。

事件處理機制(4)

- 在程式啟動時：



- 當程式持續執行時：



事件處理機制(5)

- 要成為特定事件的監聽物件，首先必須符合該事件的規範。
 - 在Android中，以Java的介面(Interface)來規範處理事件的方法。
 - 例如按一下(onClick)事件，對應的規範是OnClickListener介面。
 - 該介面規定了onClick()方法的規格，包括方法名稱、輸出及輸入參數的型態。
 - 我們在監聽物件中必須依其規格撰寫onClick()方法。

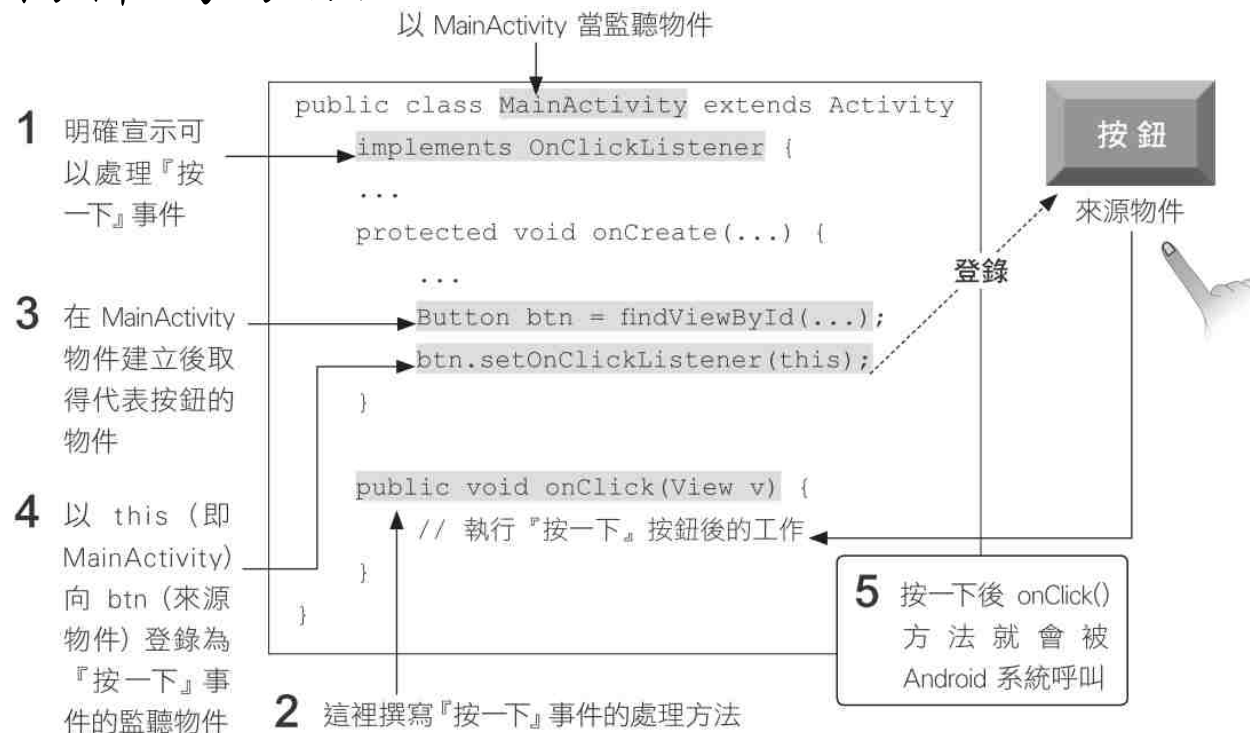
事件處理機制(6)

- 監聽物件類別一開始要宣告成那個事件的監聽物件，才能登錄為監聽物件。這個宣告的動作就稱為實作(implement XX介面)：

```
public class MainActivity extends Activity
    implements OnClickListener { ← 明確宣告要成為『按一下』事件的監聽物件
    ...
    public void onClick(View v) { ← 然後在監聽物件中依介面規格撰寫能夠
                                   處理『按一下』事件的方法
        ...
    }
}
```

事件處理機制(7)

- 準備好監聽物件後，接著登錄到來源物件中。
Android已事先為各個常用物件定義出許多登錄監聽物件的方法：



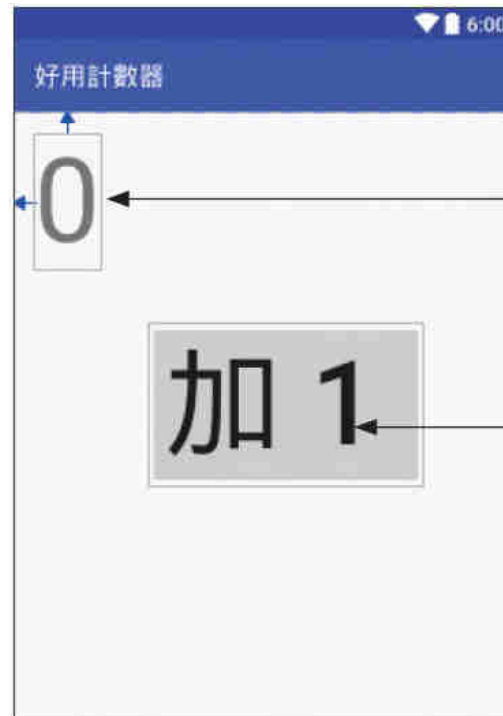
處理按一下事件 (1)

- 程式範例請參考P.4-6~P.4-12 Step 1到Step 5的操作。



處理按一下事件(2)

- Step 1: 新增一個專案，並將應用程式名稱設定為 "好用計數器"。
- Step 2:



1 設定 TextView 元件屬性：

屬性	值
id	textView
text	0
textSize	90sp

2 設定 Button 元件屬性：

屬性	值
id	button
text	加 1 (請在 1 的前後各留一空白)
textSize	90sp

處理按一下事件(3)

- Step 3-1:

```
01 package tw.com.flag.ch04_ezcounter;
02
03 import ...
10
11 public class MainActivity extends AppCompatActivity
12     implements View.OnClickListener { ←
                                宣告要實作 OnClickListener 介面成為監聽物件
13     TextView txv;           ← 用來操作 textView 元件的變數
14     Button btn;             ← 用來操作 button 元件的變數
15     int counter = 0;        ← 用來儲存計數的值，初值為 0
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21
```

處理按一下事件(4)

- Step 3-2:

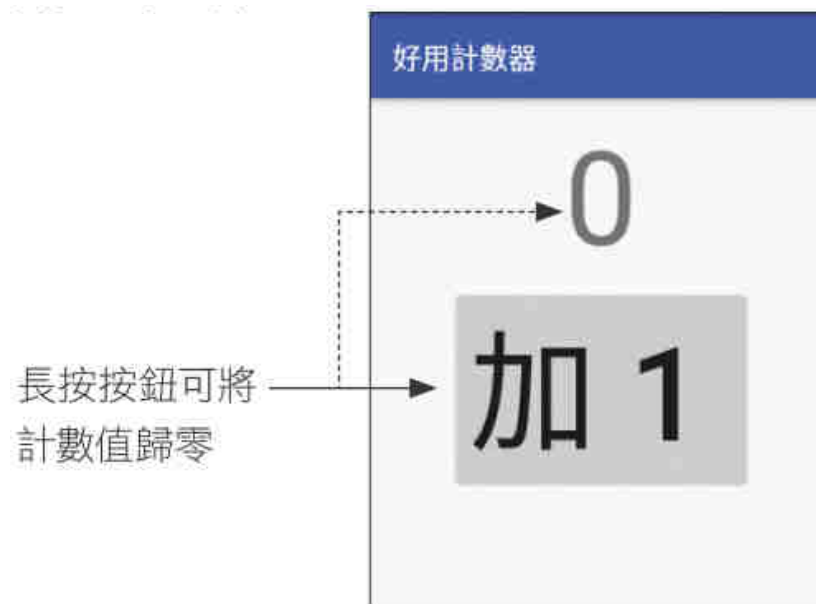
```
22      txv = (TextView) findViewById(R.id.textView); ← 找出要操作的物件
23      btn = (Button) findViewById(R.id.button); ← 找出要操作的物件
24
25      btn.setOnClickListener(this); ← 登錄 (set) 監聽物件,
                                     this 表示 MainActivity 物件本身
26  }
27
28  @Override
29  public void onClick(View v) { ← 在這裡撰寫監聽器介面中定義的 onClick 方法
30      txv.setText(String.valueOf(++counter)); ← 將計數值加 1, 然後
                                                轉成字串顯示出來
31  }
32 }
```

處理按一下事件 (5)

- 若想要在按下文字框後，所顯示的數字也能加1，則只需在前述程式的第25行的前後加入以下指令：
➤ `txv.setOnClickListener(this)`。

處理長按事件(1)

- 前述好用計數器只有按一下加1的功能。
- 請思考如何加入在按鈕上長按(按住不放約1秒)，將計數器歸零的功能？



處理長按事件(2)

- 要處理長按事件，必須實作OnLongClickListener介面的監聽物件，並且設計介面定義的onLongClick()方法。

```
public boolean onLongClick(View v) {  
    ...  
}
```

處理長按事件(3)

- 先前介紹的onClick()不需傳回任何值，但長按動作一定包含按一下動作。
- onLongClick()必須傳回一個布林值，代表是否只要引發長按事件，或在手指放開時引發按一下事件。
 - 若傳回true，表示不會引發按一下事件。
 - 若傳回false，會在使用者放開手指時立刻引發按一下事件。

處理長按事件(4)

```
...
11
12 public class MainActivity extends AppCompatActivity
13     implements View.OnClickListener, View.OnLongClickListener
    {
14     TextView txv;      ← 用來操作 textView 元件的變數
15     Button btn;       ← 用來操作 button1 元件的變數
16     int counter = 0;  ← 用來儲存計數的值，初值為 0
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22
23         txv = (TextView) findViewById(R.id.textview); ← 找出要操作的物件
24         btn = (Button) findViewById(R.id.button); ← 找出要操作的物件
```

處理長按事件(5)

```
25
26     btn.setOnClickListener(this); ← 登錄監聽物件, this 表示活動物件本身
27     btn.setOnLongClickListener(this); ←
                                   將 MainActivity 物件登錄為按鈕的長按監聽器
28 }
29
30 @Override
31 public void onClick(View v) { ← 實作監聽器介面中定義的 onClick 方法
32     txv.setText(String.valueOf(++counter)); ←
                                   將計數值加 1, 然後轉成字串顯示出來
33 }
34
35 @Override                    實作長按 (OnLongClickListener) 介面定義的方法
36 public boolean onLongClick(View v) { ←
37
38     counter = 0;
39     txv.setText("0");
40     return true;
41 }
```