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

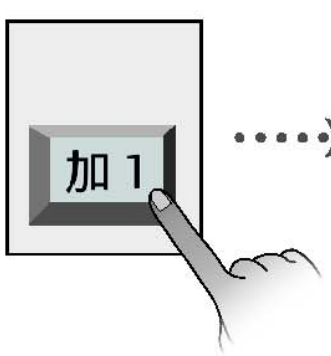
第 4 章

與使用者互動–事件處理

前言

- 4-1 事件處理的機制
- 4-2『按一下』事件的處理
- 4-3 監聽『長按』事件
- 4-4 處理不同來源物件的相同事件
- 4-5 監聽『觸控』事件讓手機震動

4-1 事件處理的機制



使用者按一下按鈕

產生 onClick 事件

```
public class MainActivity extends Activity {
    TextView txv;       // 參照 textView1 元件的變數
    Button btn;         // 參照 button1 元件的變數
    int counter = 0;    // 用來儲存計數的值，初值為 0

    // 定義實作 OnClickListener 介面的類別
    class ButtonOnClick implements OnClickListener {
        public void onClick(View v) {
            txv.setText(String.valueOf(++counter));
        }
    }
```

手機就會執行我們預先寫好的事件處理程式

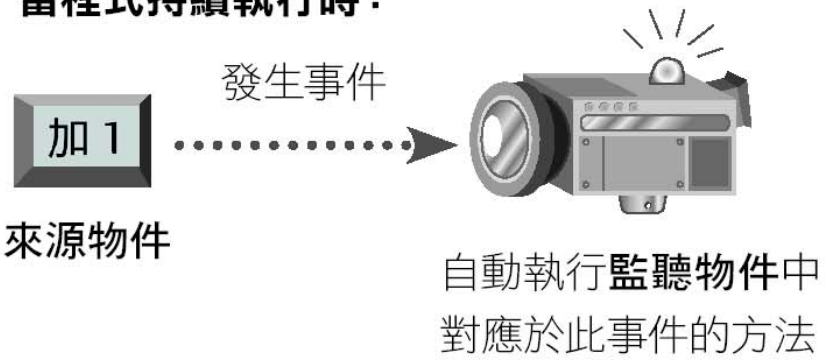
事件處理的機制

- 來源物件與監聽物件
- Java 的介面(Interface)

來源物件與監聽物件

- **在程式啟動時：**

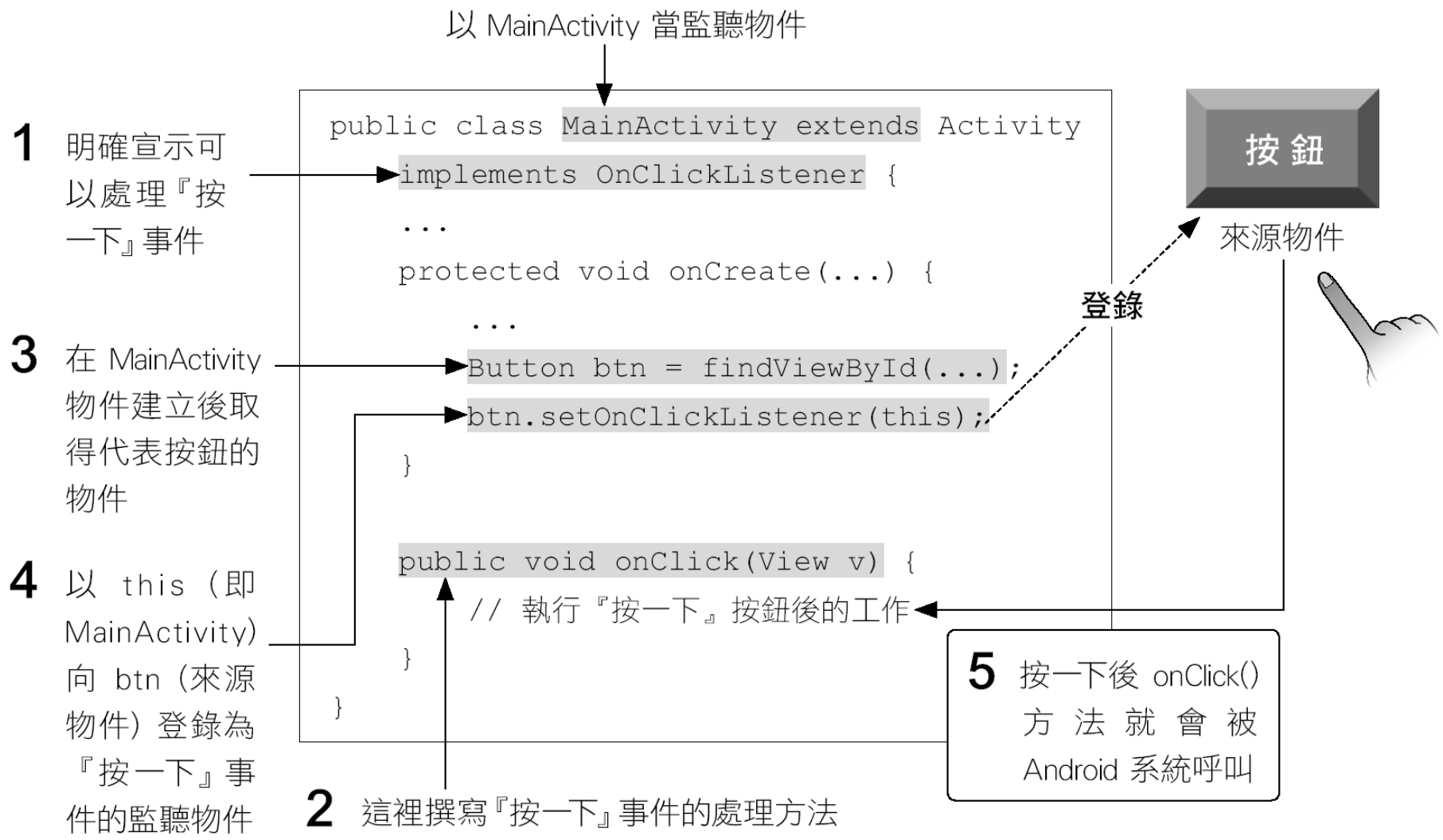
- **當程式持續執行時：**



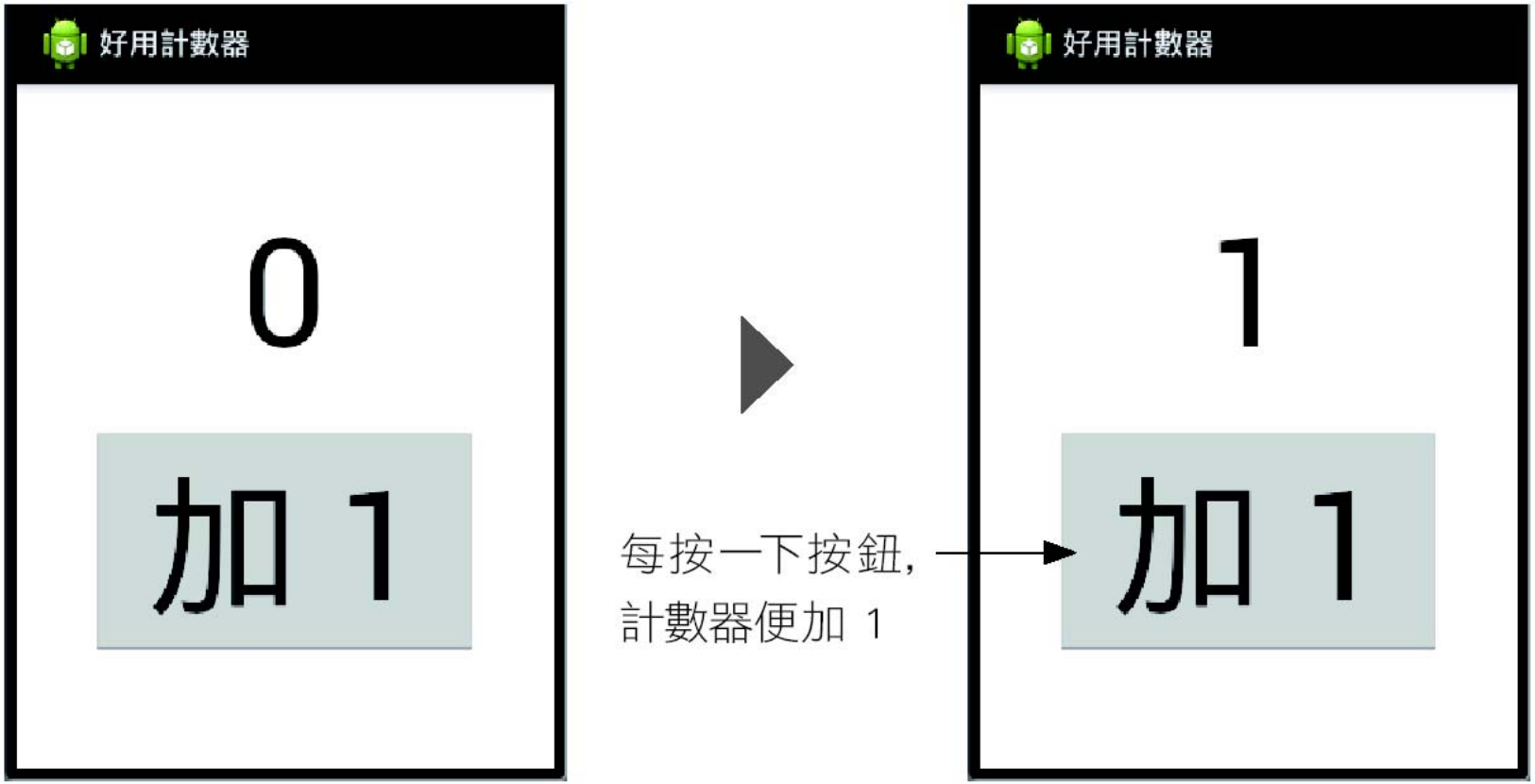
Java 的介面 (Interface)

```
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
        ...
    }
}
```

Java 的介面 (Interface)



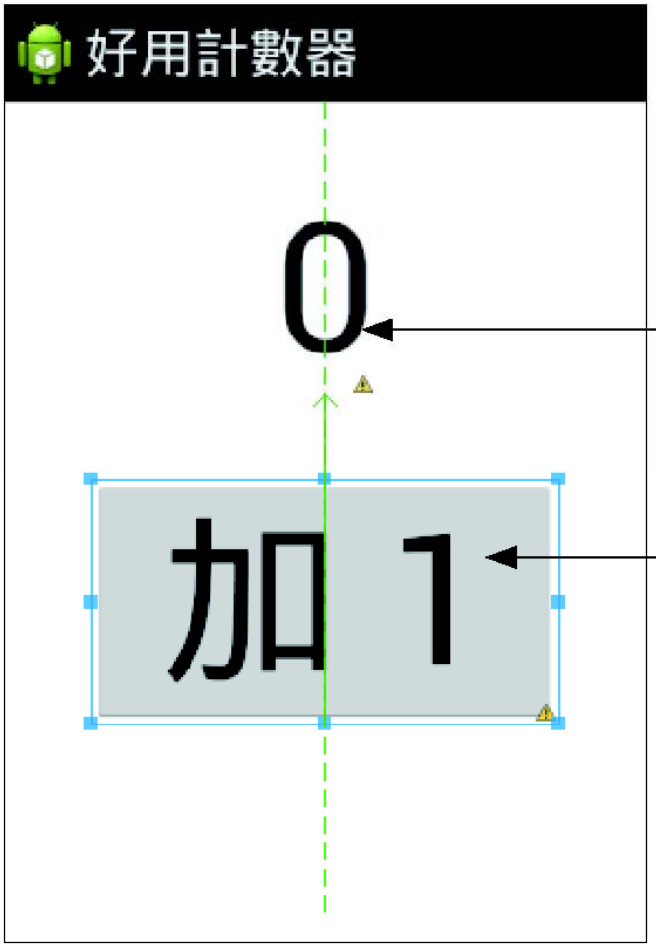
4-2『按一下』事件的處理



範例4-1 ：每按一下按鈕, 就讓計數器加1

step 1

step 2



1 將 TextView 元件上移並水平置中, 設定屬性：

屬性	值
Id	@+id/textView1
Text	0
Text Size	90dp

2 將 Button 元件移到 TextView 之下並水平置中, 設定屬性：

屬性	值
Id	@+id/button1
Text	加 1 (請在前後各留一空白)
Text Size	90dp

每按一下按鈕, 就讓計數器加1

step 3

```
01 package tw.com.flag.ch04_ezcounter;
02
03 import android.os.Bundle;
04 import android.app.Activity;
05 import android.view.Menu;
06 import android.view.View;
07 import android.view.View.OnClickListener; ◄— 匯入監聽器介面
08 import android.widget.Button;
09 import android.widget.TextView;
10
11 public class MainActivity extends Activity
12         implements OnClickListener {◄— 宣告要實作 OnClickListener
                                          介面成為監聽物件
13     TextView txv;          ◄— 用來操作 textView1 元件的變數
```

每按一下按鈕, 就讓計數器加1

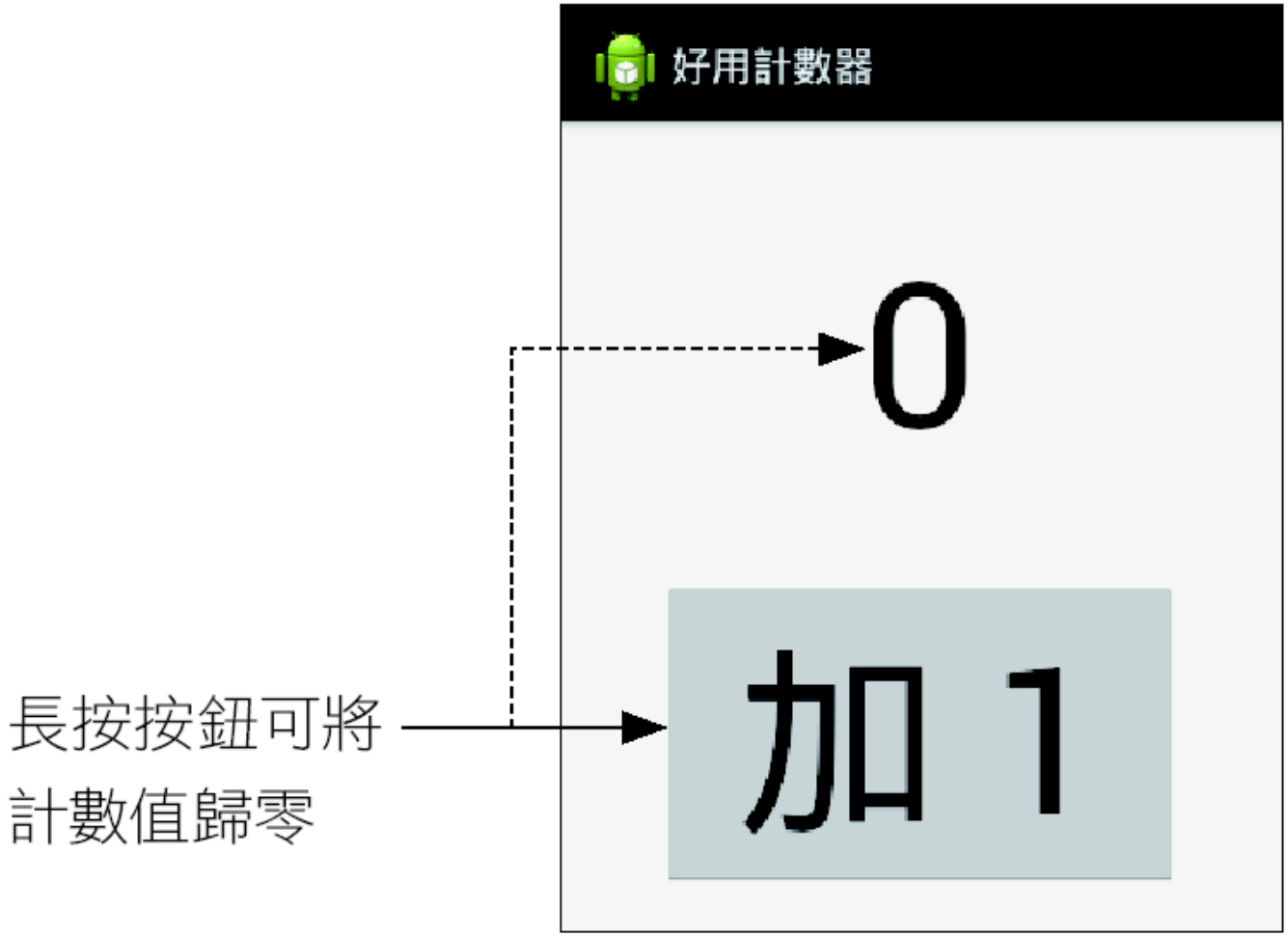
```
14      Button btn;          ◄── 用來操作 button1 元件的變數
15      int counter = 0;     ◄── 用來儲存計數的值, 初值為 0
16
17      @Override
18      public void onClick(View v) { ◄── 在這裡撰寫監聽器介面中定義的 onClick 方法
19          txv.setText(String.valueOf(++counter));◄── 將計數值加 1,然後
                                                       轉成字串顯示出來
20      }
21
22      @Override
23      protected void onCreate(Bundle savedInstanceState) {
24          super.onCreate(savedInstanceState);
25          setContentView(R.layout.activity_main);
26
```

Next

每按一下按鈕, 就讓計數器加1

```
27          txv = (TextView) findViewById(R.id.textView1);   ← 找出要操作的物件
28          btn = (Button) findViewById(R.id.button1);       ← 找出要操作的物件
29
30          btn.setOnClickListener(this);   ← 登錄 (set) 監聽物件, this 表示 MainActivity 物件本身
31      }
32
33      @Override
34      public boolean onCreateOptionsMenu(Menu menu) {
35          getMenuInflater().inflate(R.menu.activity_main, menu);
36          return true;
37      }
38  }
```

4-3 監聽『長按』事件



監聽『長按』事件

- onLongClick()：處理『長按』事件

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

範例4-2 ：長按按鈕將計數值歸零

step 1

```
 ...
06 import android.view.View;
07 import android.view.View.OnClickListener;
08 import android.view.View.OnLongClickListener;
09 import android.widget.Button;
10 import android.widget.TextView;
11
12 public class MainActivity extends Activity
13         implements OnClickListener, OnLongClickListener {
14     TextView txv;
15     Button btn;
```

匯入監聽器介面
(按 Ctrl + Shift + O)

實作兩個介面

實作 OnLongClickListener 介面

用來操作 textView1 元件的變數

用來操作 button1 元件的變數

長按按鈕將計數值歸零

```
16      int counter = 0;      ← 用來儲存計數的值, 初值為 0
17
18      @Override
19      public void onClick(View v) {  ← 實作監聽器介面中定義的 onClick 方法
20          txv.setText(String.valueOf(++counter));
                                        將計數值加 1, 然後轉成字串顯示出來
21      }
22
23      @Override
24      public boolean onLongClick(View v) {
                              實作長按 (OnLongClickListener) 介面定義的方法
25          counter = 0;
26          txv.setText("0");
27          return true;
28      }
```

長按按鈕將計數值歸零

```
29
30      @Override
31      protected void onCreate(Bundle savedInstanceState) {
32          super.onCreate(savedInstanceState);
33          setContentView(R.layout.activity_main);
34
35          txv = (TextView) findViewById(R.id.textView1);◄── 找出要操作的物件
36          btn = (Button) findViewById(R.id.button1);◄── 找出要操作的物件
37
38          btn.setOnClickListener(this);◄── 登錄監聽物件, this 表示活動物件本身
39          btn.setOnLongClickListener(this);◄┐
                                  將 MainActivity 物件登錄為按鈕的長按監聽器
40      }
 ...
47  }
```

4-4 處理不同來源物件的相同事件

- getId()：判斷事件的來源物件

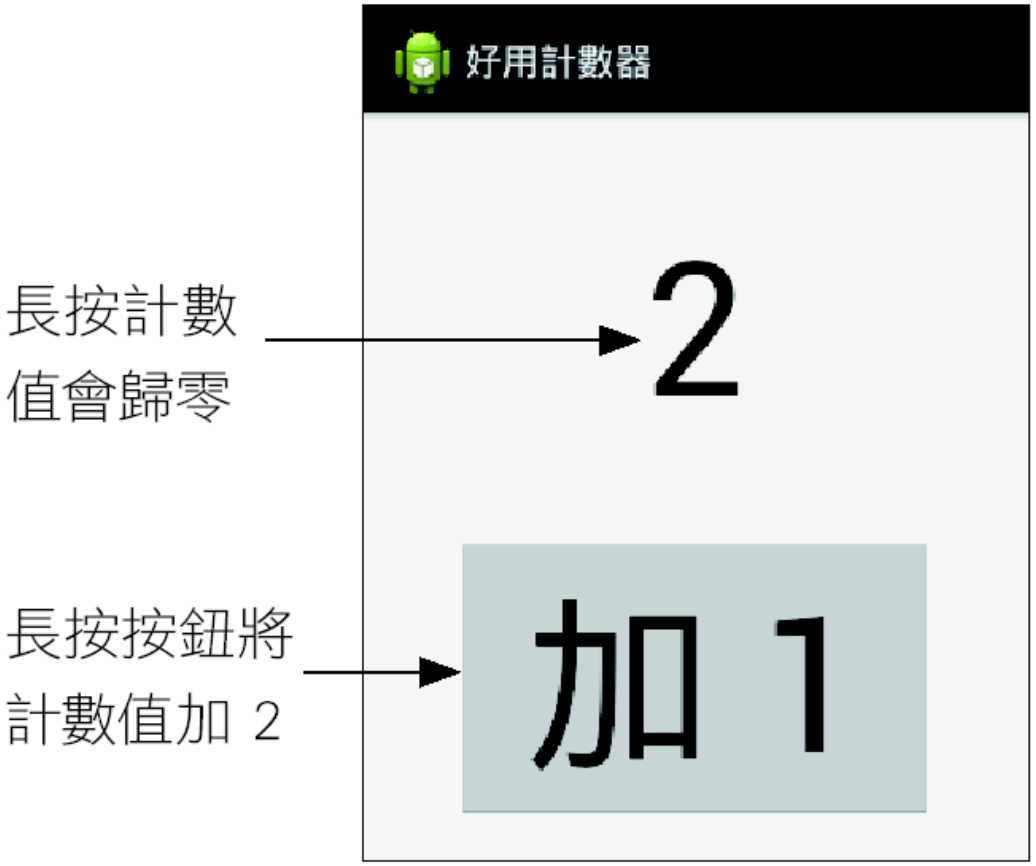
```
public void onClick(View v) {
    ...
}

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

getId()：判斷事件的來源物件

```
public boolean onLongClick(View v) {
    if(v.getId() == R.id.textView1) {
        // 使用者在顯示計數值的 txv 上長按, 將計數歸零
    }
    else {
        // 使用者在按鈕長按, 將計數加 2
    }
}
```

範例4-3 ：長按按鈕計數加 2, 長按計數值可歸零

step 1



長按按鈕計數加 2, 長按計數值可歸零

```
01 public class MainActivity extends Activity
02         implements OnClickListener, OnLongClickListener {   ← 實作 OnLongClickListener 介面
03     TextView txv;          ← 用來操作 textView1 元件的變數
04     Button btn;            ← 用來操作 button1 元件的變數
05     int counter = 0;       ← 用來儲存計數的值, 初值為 0
06
...
12     @Override                    實作長按 (OnLongClickListener 介面) 的方法
13     public boolean onLongClick(View v) {
14         if(v.getId() == R.id.textView1) {   ← 判斷來源物件是否為顯示計數值的 TextView, 若是就將計數歸零
15             counter = 0;
```

長按按鈕計數加 2, 長按計數值可歸零

```
16              txv.setText("0");
17          }
18          else { ◄── 來源物件為按鈕, 將計數值加 2
19              counter += 2;
20              txv.setText(String.valueOf(counter));
21          }
22          return true;
23      }
24
25      @Override
26      protected void onCreate(Bundle savedInstanceState) {
27          super.onCreate(savedInstanceState);
28          setContentView(R.layout.activity_main);
```

Next

長按按鈕計數加 2, 長按計數值可歸零

```
29
30          txv = (TextView) findViewById(R.id.textView1);
31          btn = (Button) findViewById(R.id.button1);
32
33          btn.setOnClickListener(this);
34          btn.setOnLongClickListener(this);
35          txv.setOnLongClickListener(this);
36      }
 ...
43  }
```

找出要操作的物件

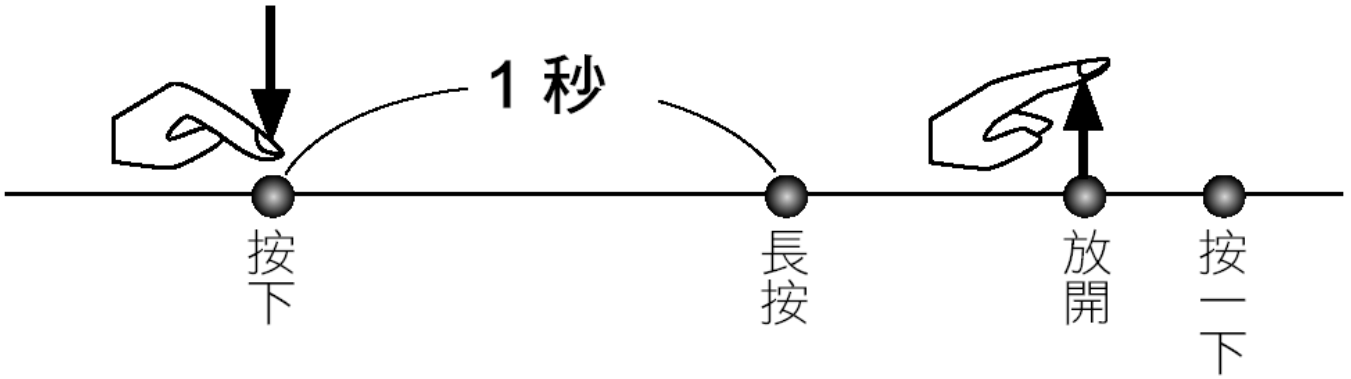
找出要操作的物件

登錄監聽物件, this 表示 MainActivity 物件本身

將 MainActivity 物件登錄為按鈕的長按監聽器

將 MainActivity 物件登錄為文字標籤的長按監聽器

4-5 監聽『觸控』事件讓手機震動



監聽『觸控』事件讓手機震動

- onTouch()：觸控事件的處理
- 如何讓手機震動
- 在程式中登記『震動』的使用權限

onTouch()：觸控事件的處理

```
public boolean onTouch(View v, MotionEvent e) {
    ...
}
```

如何讓手機震動

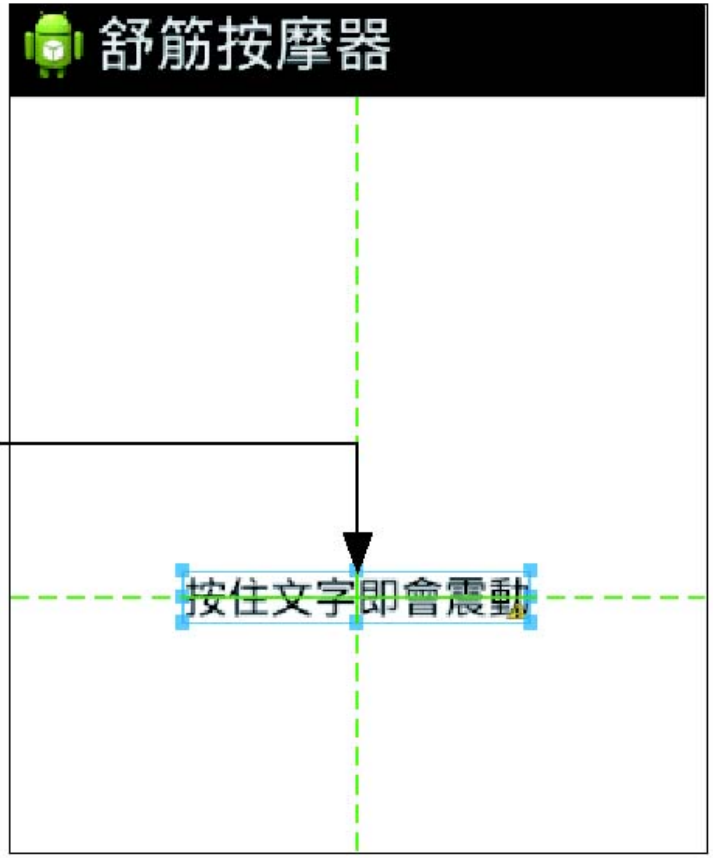
```
//取得震動物件
Vibrator vb = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);

//震動與停止
vb.vibrate(5000);   ◄── 震動 5 秒 (5000ms)
vb.cancel();        ◄── 停止震動
```

範例4-4 ：監聽 TextView 的觸控事件

step 1

將預建的 TextView 元件依下表修改：

屬性	值
Id	@+id/textView1
Text	按住文字即會震動
Text Size	20dp



監聽 TextView 的觸控事件

step 2

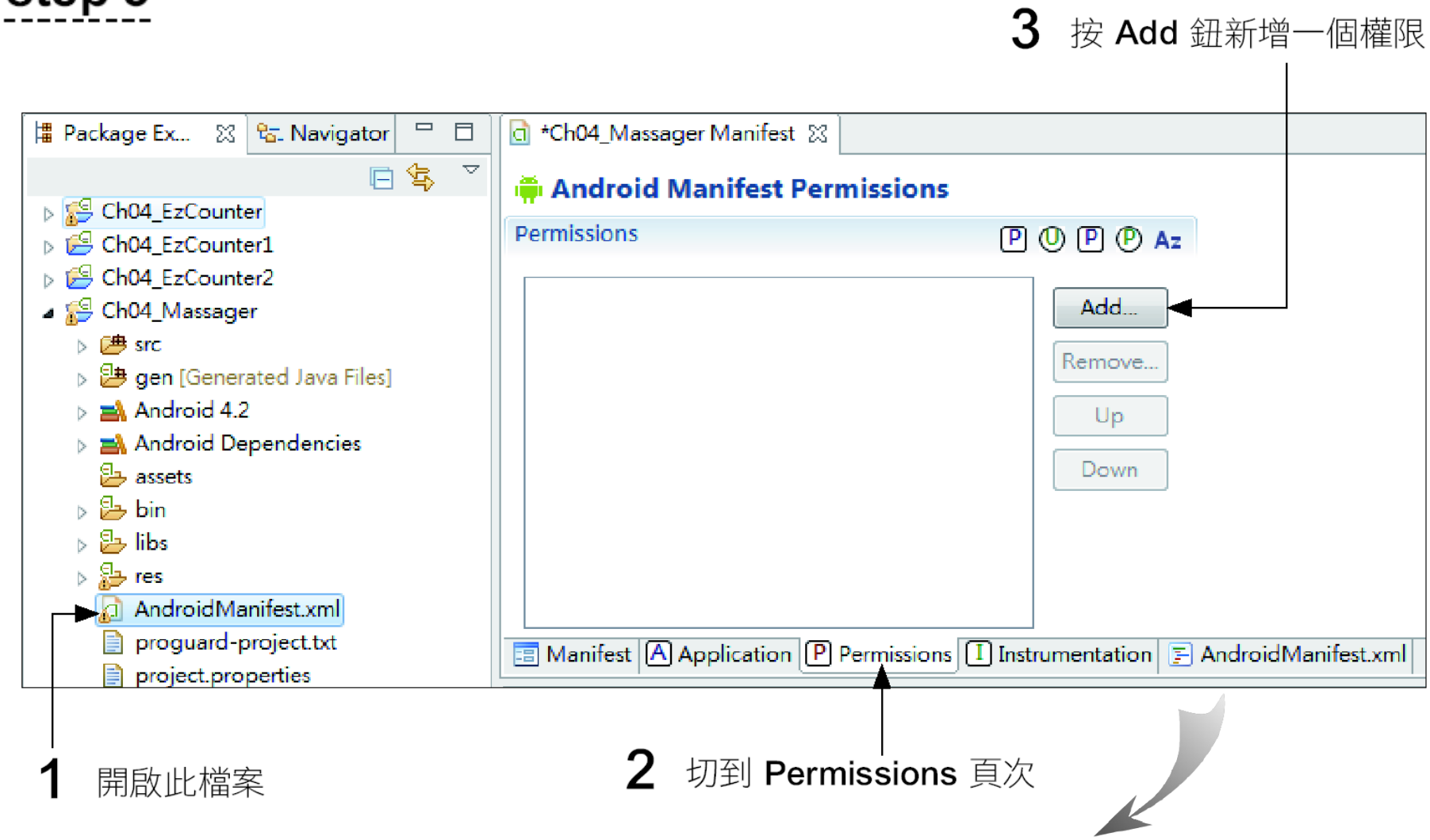
```
01 package tw.com.flag.ch04_massager;
   ...
10 public class MainActivity extends Activity implements OnTouchListener {
11     @Override
12     public boolean onTouch(View v, MotionEvent e) {
                        實作 OnTouchListener 觸控監聽器介面的方法
13         Vibrator vb = (Vibrator) getSystemService(
               Context.VIBRATOR_SERVICE);
14         if(e.getAction() == MotionEvent.ACTION_DOWN) {
                                                  按下螢幕中間的文字
15             vb.vibrate(5000); ← 震動 5 秒
16         }
17         else if(e.getAction() == MotionEvent.ACTION_UP){
                                                  放開螢幕中間的文字
```

監聽 TextView 的觸控事件

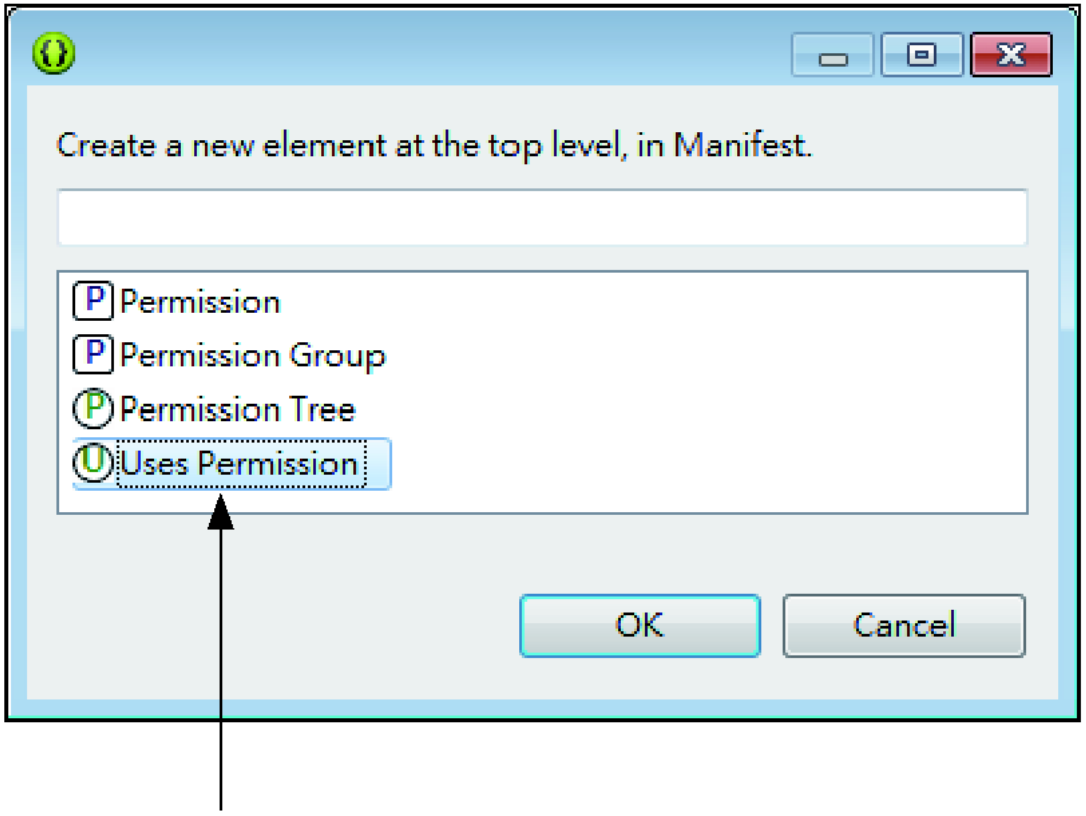
```
18              vb.cancel();          ◄— 停止震動
19          }
20          return true;
21      }
22
23      @Override
24      protected void onCreate(Bundle savedInstanceState) {
25          super.onCreate(savedInstanceState);
26          setContentView(R.layout.activity_main);
27
28          TextView txv = (TextView) findViewById(R.id.textView1);
29          txv.setOnTouchListener(this); ◄— 登錄觸控監聽物件
30      }
    ...
36  }
```

在程式中登記『震動』的使用權限

step 3



在程式中登記『震動』的使用權限



4 選擇 **Uses Permission**, 再按 **OK** 鈕

在程式中登記『震動』的使用權限

5 按下列示窗, 選取 android.permission.VIBRATE 權限

在程式中登記『震動』的使用權限

在程式中登記『震動』的使用權限

step 4

按住文字即會震動，直到放開、或過了 5 秒才停止