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 2 章

Android 程式設計基礎講座

前言

- 2-1 Android App 的主角： Activity
- 2-2 Android 程式的設計流程
- 2-3 認識 Activity 的基本程式邏輯
- 2-4 元件的佈局與屬性設定
- 2-5 開始動手寫程式
- 2-6 輸入欄位 EditText 元件
- 2-7 使用 USB 線將程式部署到手機上執行

2-1 Android App 的主角：Activity

1. Activity (活動)
2. Service (背景服務)
3. Content Provider (內容提供者)
4. Broadcast Receiver (廣播接收端)

Android App 的主角：Activity

- Activity (活動)
- 實際的狀況是這樣

Activity (活動)

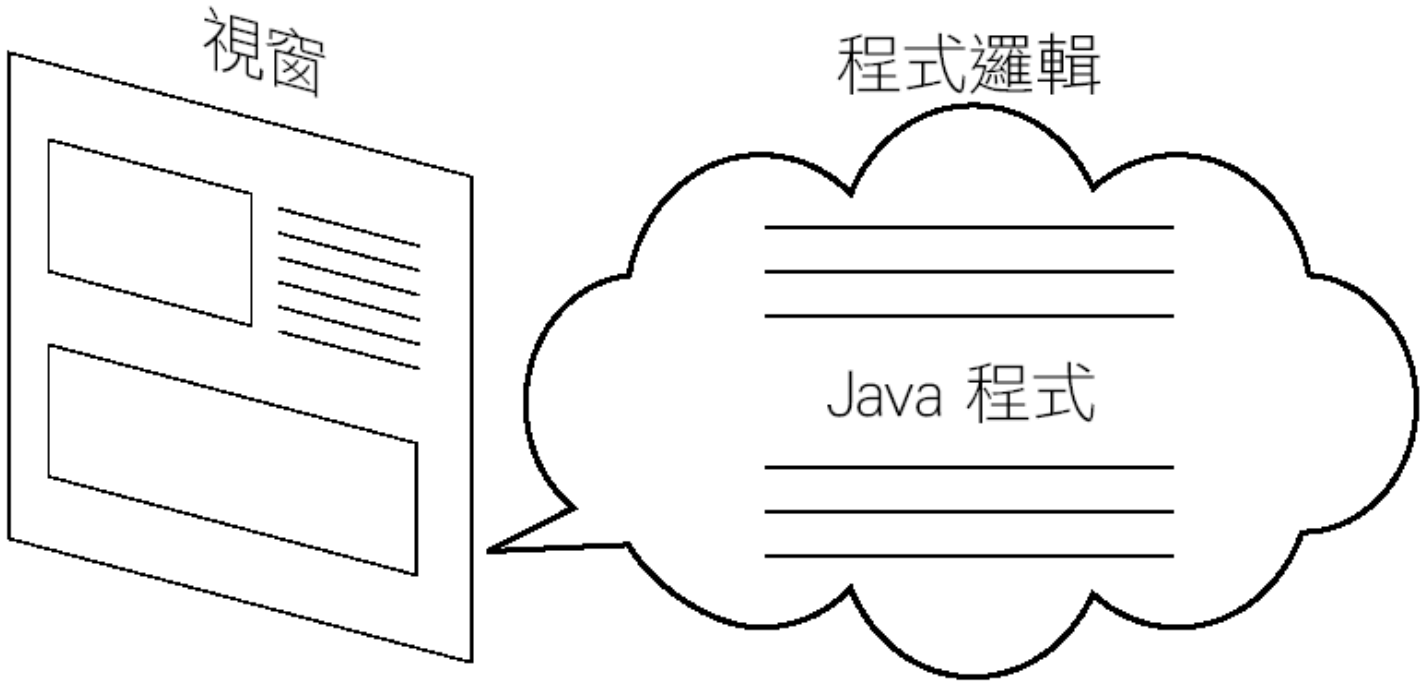
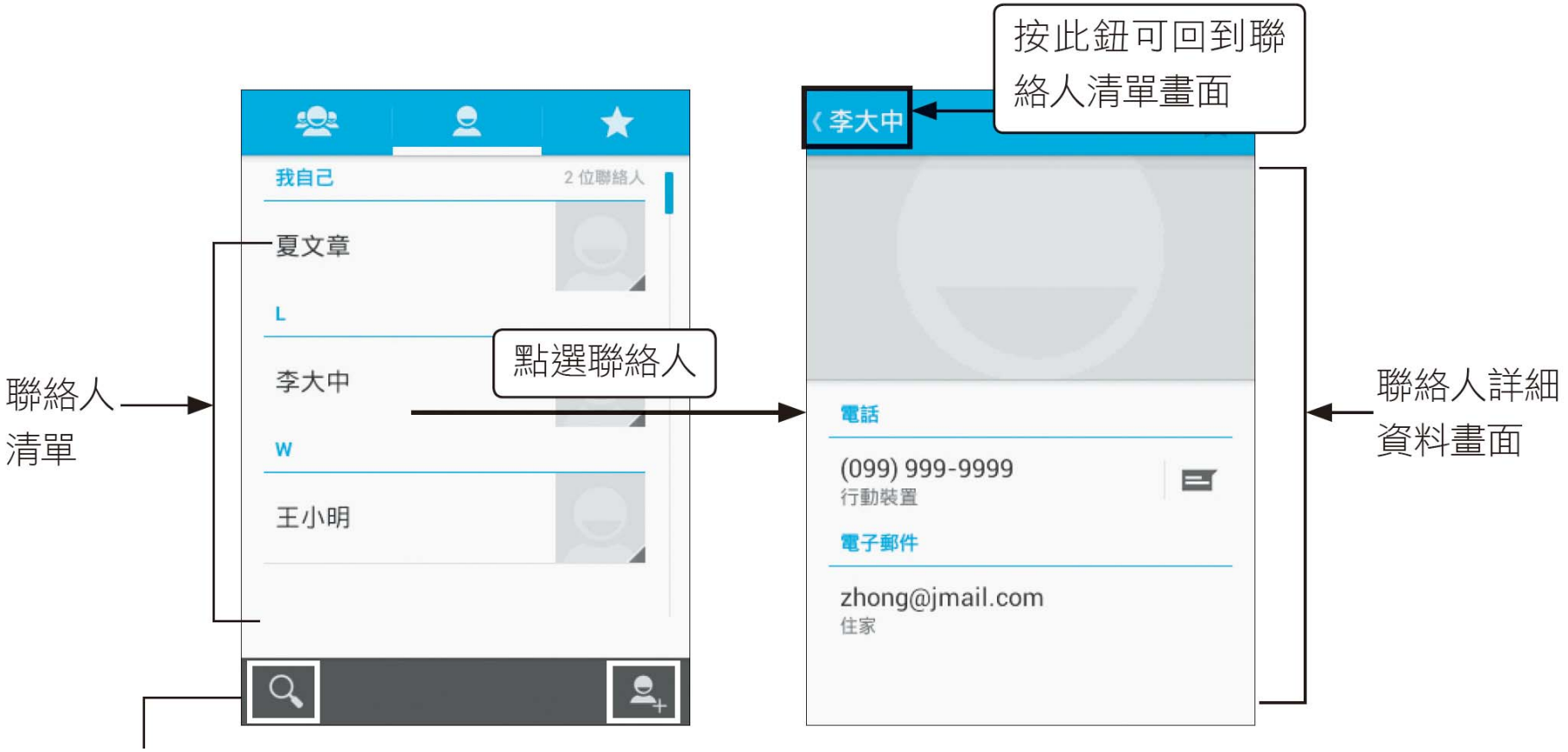
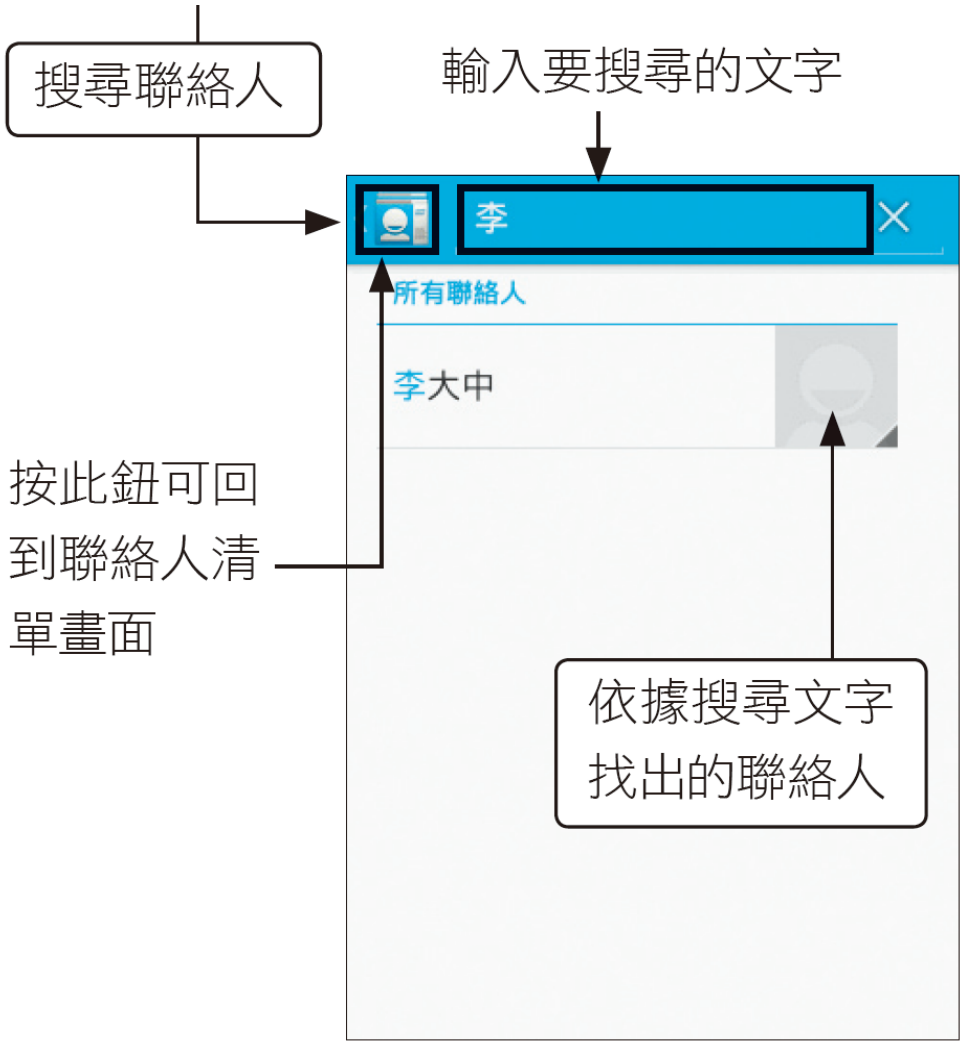


圖 2-1　一個典型的 Activity

實際的狀況是這樣


聯絡人
清單
我自己
2 位聯絡人
夏文章
L
李大中
W
王小明
點選聯絡人
按此鈕可回到聯
絡人清單畫面
李大中
電話
(099) 999-9999
行動裝置
電子郵件
zhong@jmail.com
住家
聯絡人詳細
資料畫面

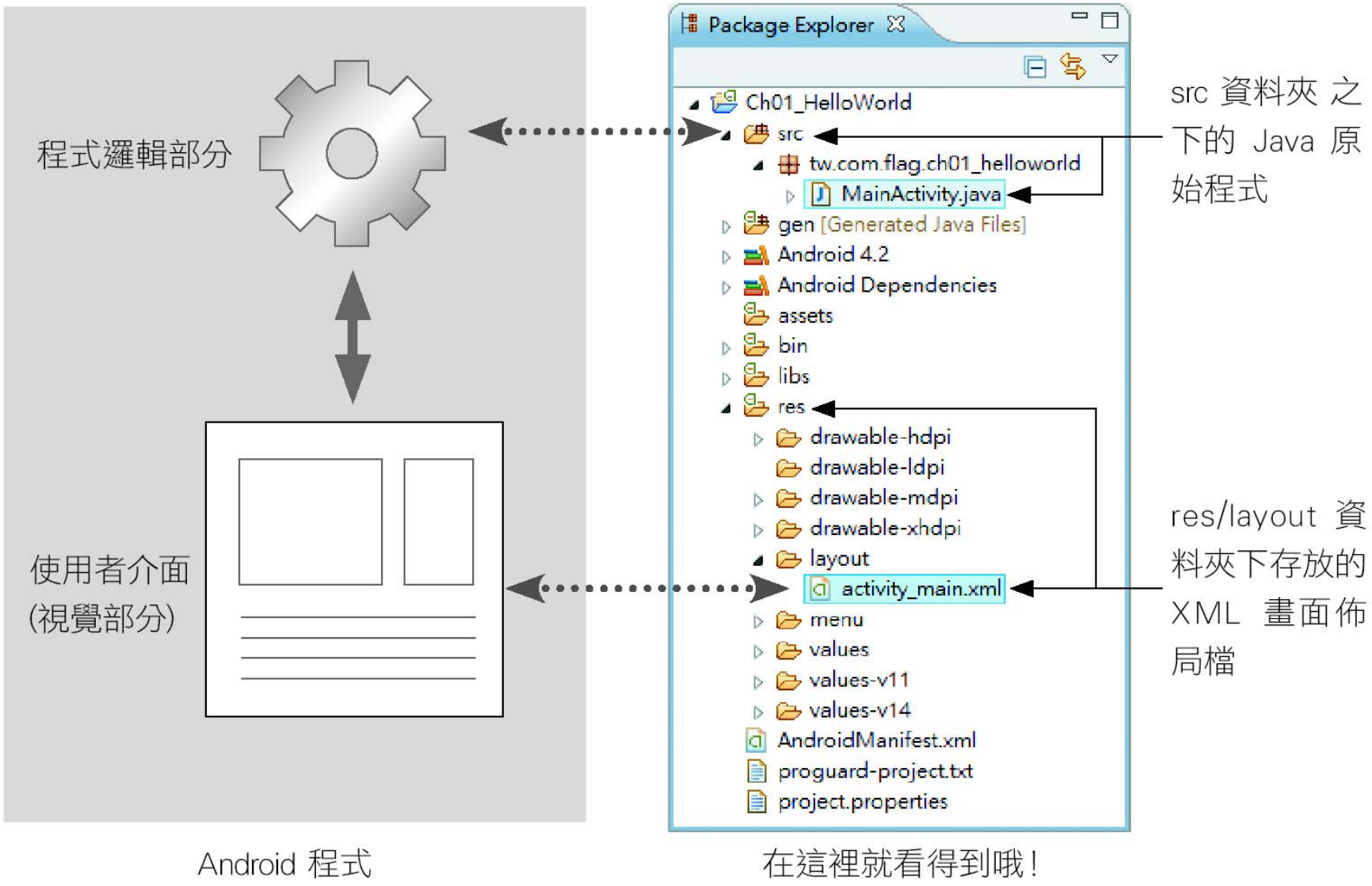
實際的狀況是這樣



2-2 Android 程式的設計流程

- 視覺設計 + 程式邏輯
- 用圖形化介面來做視覺設計
- 用 Java 來寫程式邏輯
- 最後把視覺設計與程式碼組建 (Build) 起來

視覺設計 + 程式邏輯


程式邏輯部分
使用者介面
(視覺部分)
Package Explorer
Ch01_HelloWorld
src
tw.com.flag.ch01_helloworld
MainActivity.java
gen [Generated Java Files]
Android 4.2
Android Dependencies
assets
bin
libs
res
drawable-hdpi
drawable-ldpi
drawable-mdpi
drawable-xhdpi
layout
activity_main.xml
menu
values
values-v11
values-v14
AndroidManifest.xml
proguard-project.txt
project.properties
src 資料夾 之下的 Java 原始程式
res/layout 資料夾下存放的 XML 畫面佈局檔
Android 程式
在這裡就看得到哦!

視覺設計 + 程式邏輯

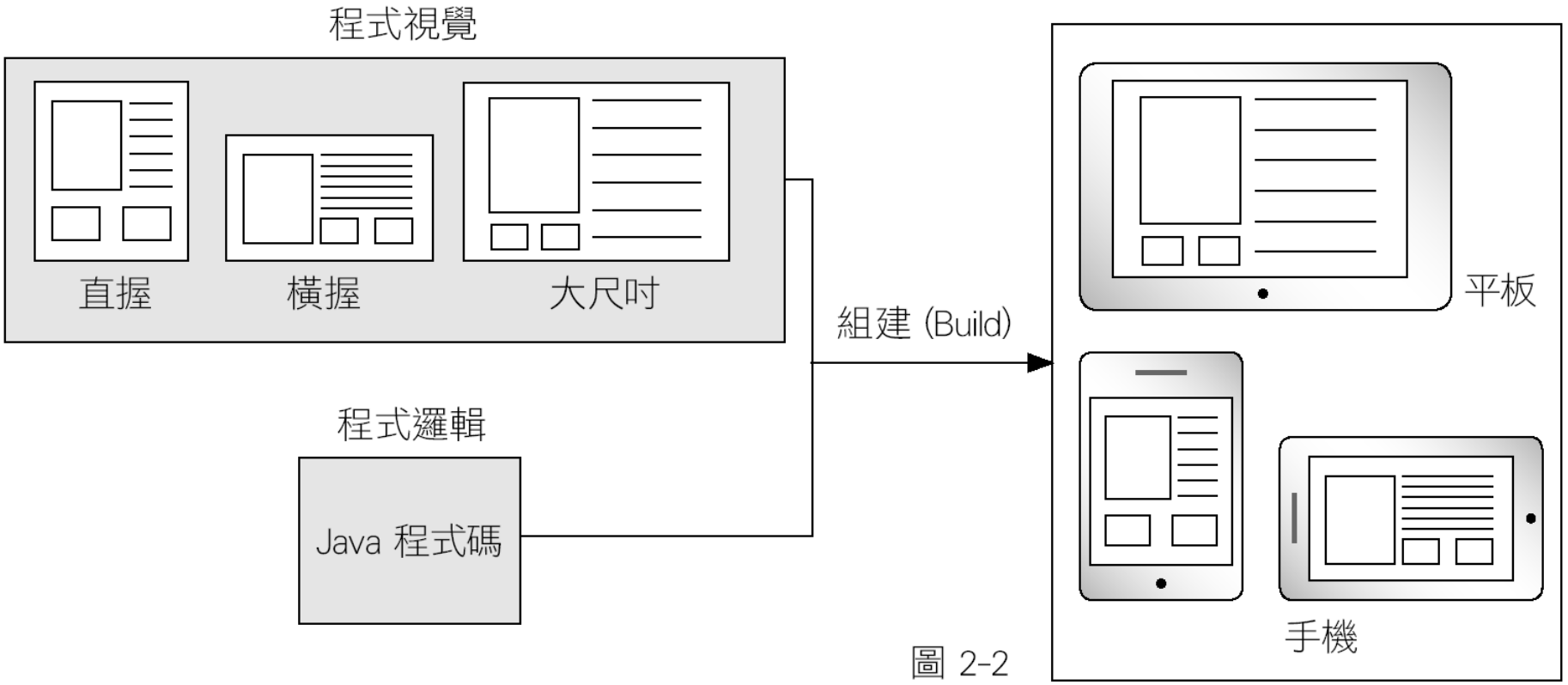
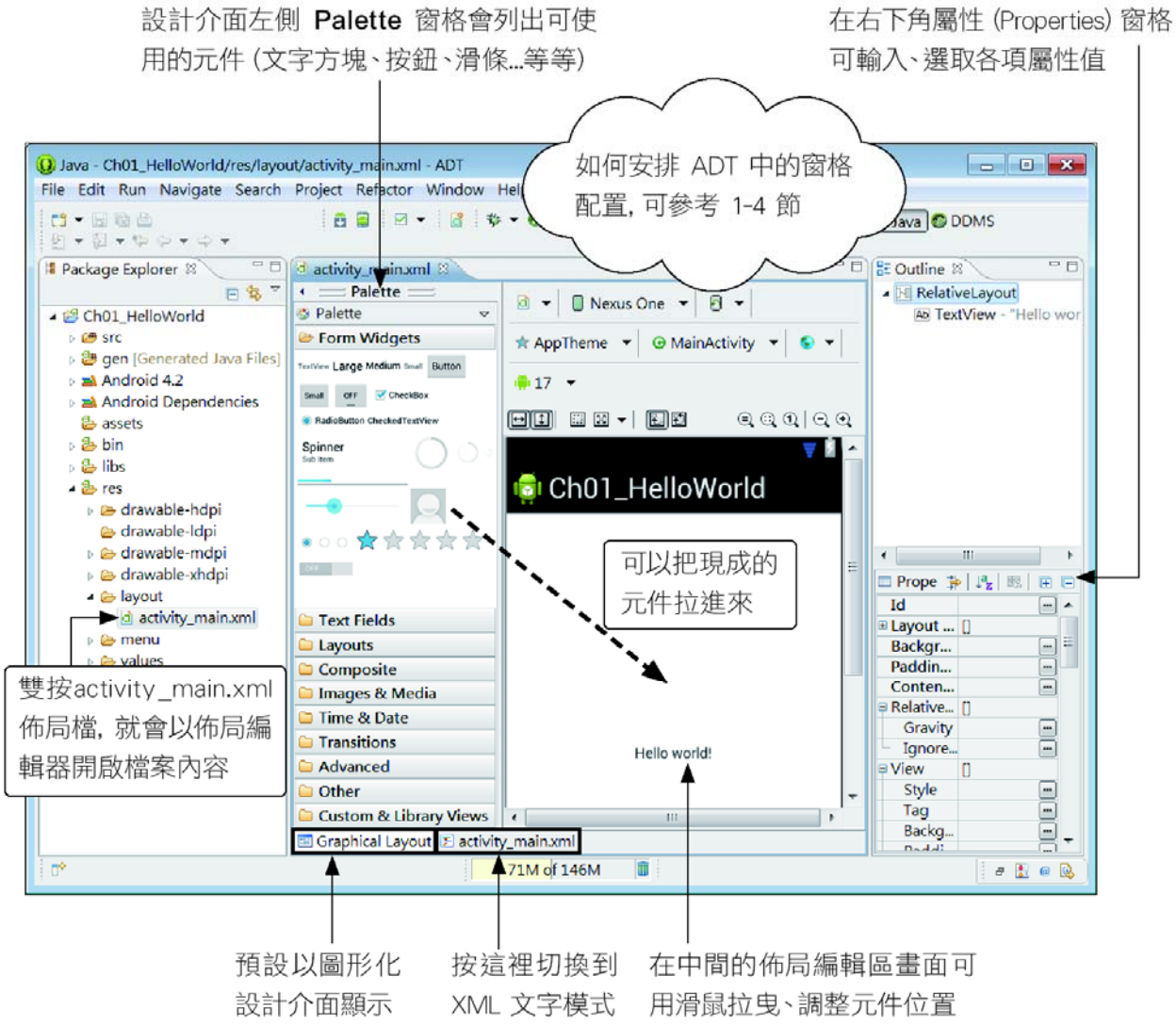
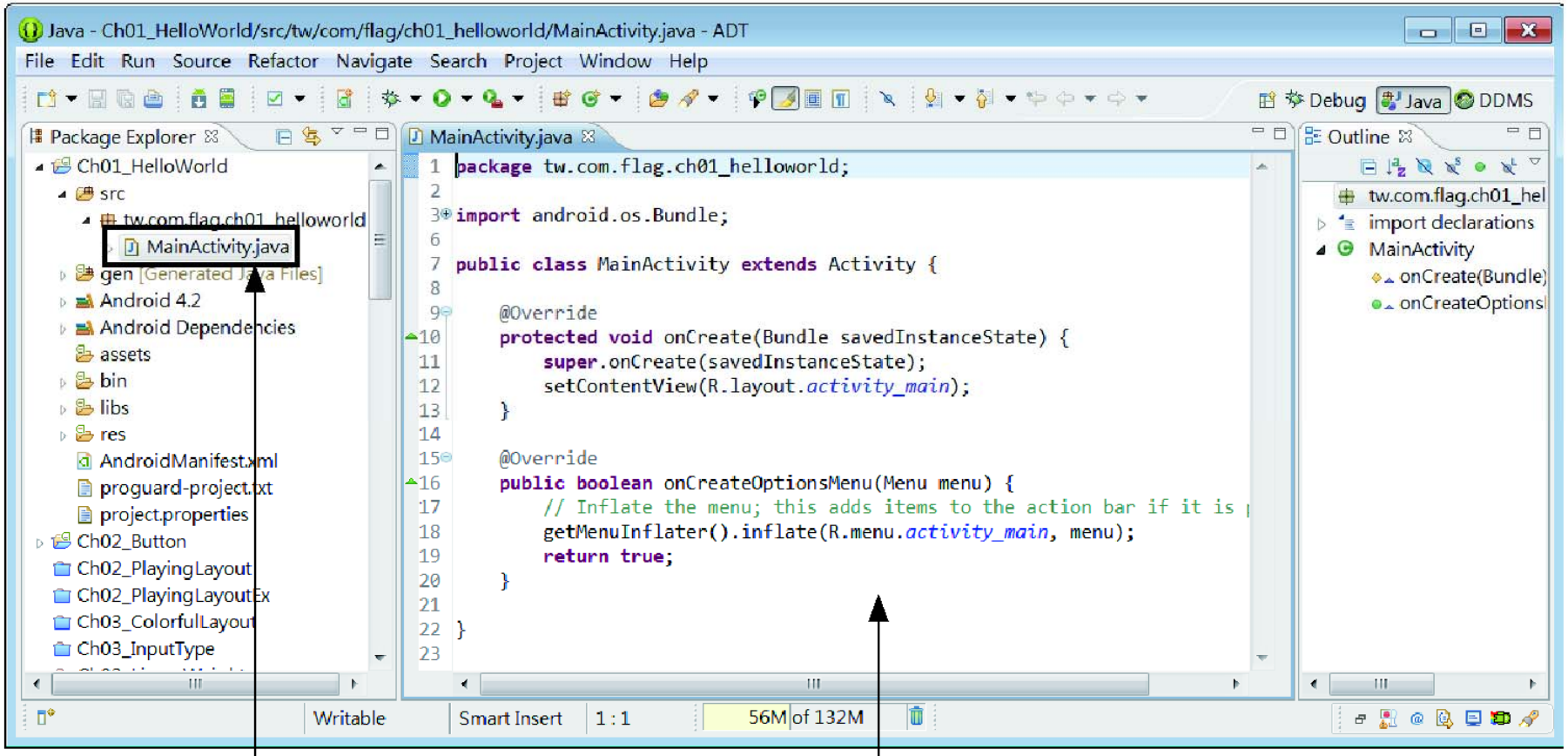


圖 2-2

用圖形化介面來做視覺設計



用Java 來寫程式邏輯



自動建立的 Java 程式檔

Java 程式內容

最後把視覺設計與程式碼組建 (Build) 起來

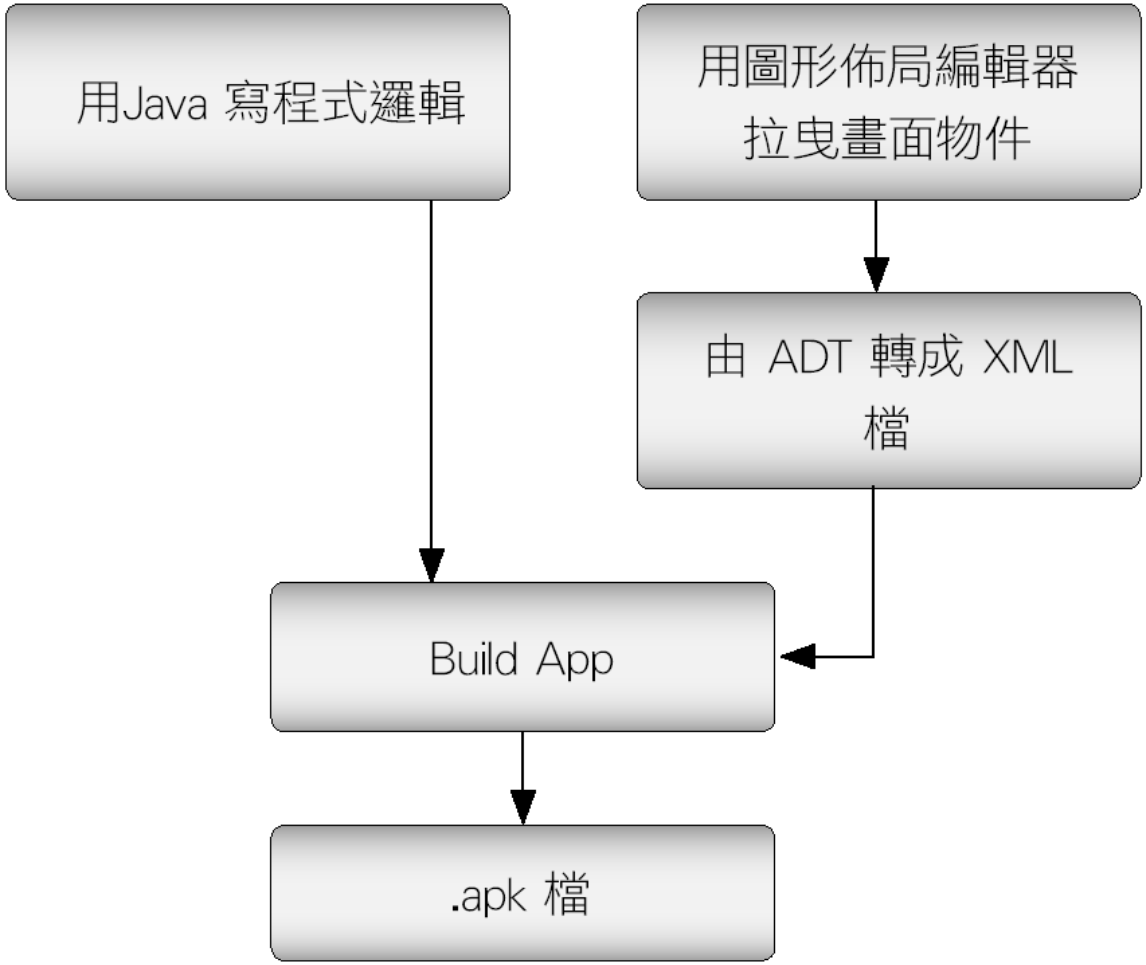
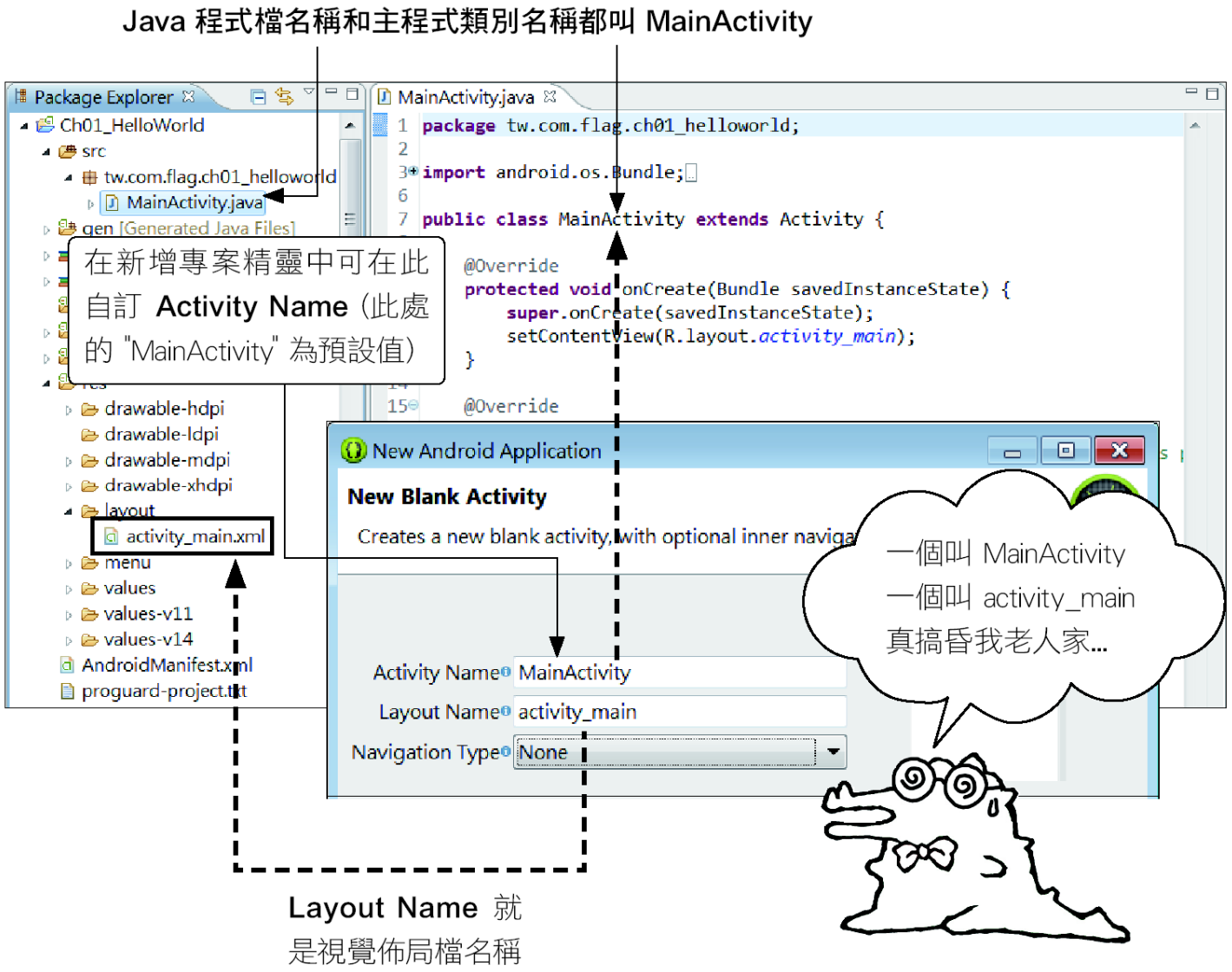


圖 2-3 Android 程式設計的基本流程

2-3 認識 Activity 的基本程式邏輯

- 初識 MainActivity 框架
- onCreate(): MainActivity 第一件要做的事
- setContentView()：載入佈局檔
- 資源的 ID

初識 MainActivity 框架



初識 MainActivity 框架

onCreate(): MainActivity 第一件要做的事

程式 2-1 新建專案時 ADT 自動產生的 Java 程式框架

```
01 package tw.com.flag.ch01_helloworld;
02
03 import android.os.Bundle;
04 import android.app.Activity;
05 import android.view.Menu;
06
07 public class MainActivity extends Activity {
08
09     @Override
10     protected void onCreate(Bundle savedInstanceState) {
11         super.onCreate(savedInstanceState);
```

← 我們之前設定的套件(package)名稱，這是 Java 的標準語法

← 先呼叫父類別的 onCreate()來做該做的事

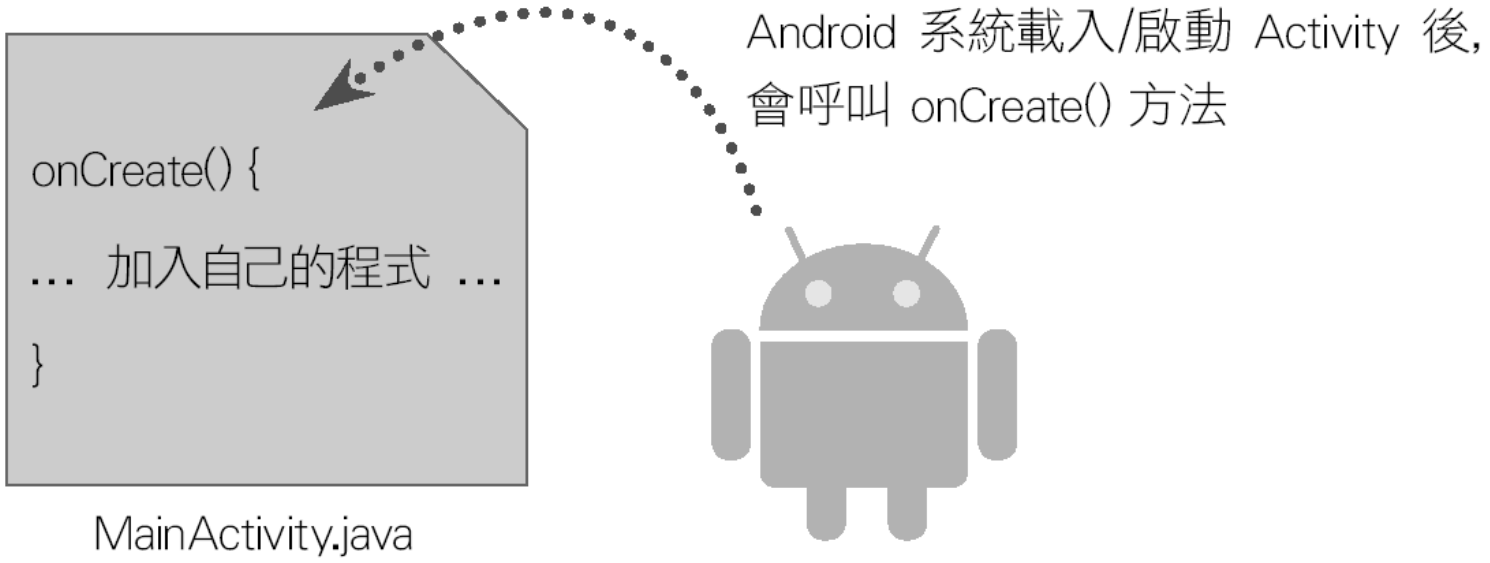
onCreate(): MainActivity 第一件要做的事

```
12          setContentView(R.layout.activity_main);
13      }
14
15      @Override
16      public boolean onCreateOptionsMenu(Menu menu) {
17          // Inflate the menu; this adds items to the
            // action bar if it is present.
18          getMenuInflater().inflate(R.menu.activity_main, menu);
19          return true;
20      }
21
22  }
```

然後我才來做我要做的

這一部份暫時用不到...

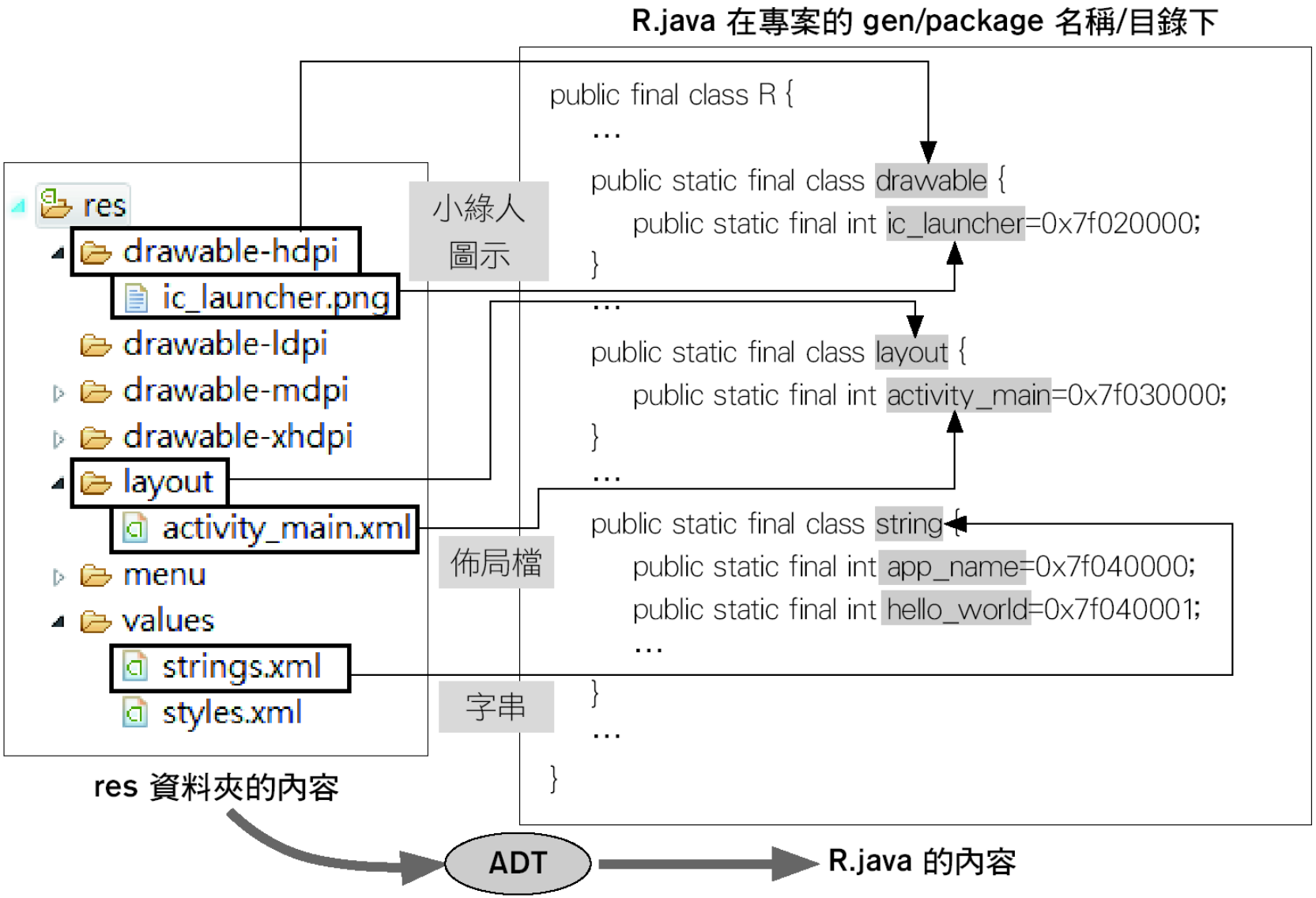
onCreate(): MainActivity 第一件要做的事



setContentView()：載入佈局檔

```
setContentView(R.layout.activity_main);
```

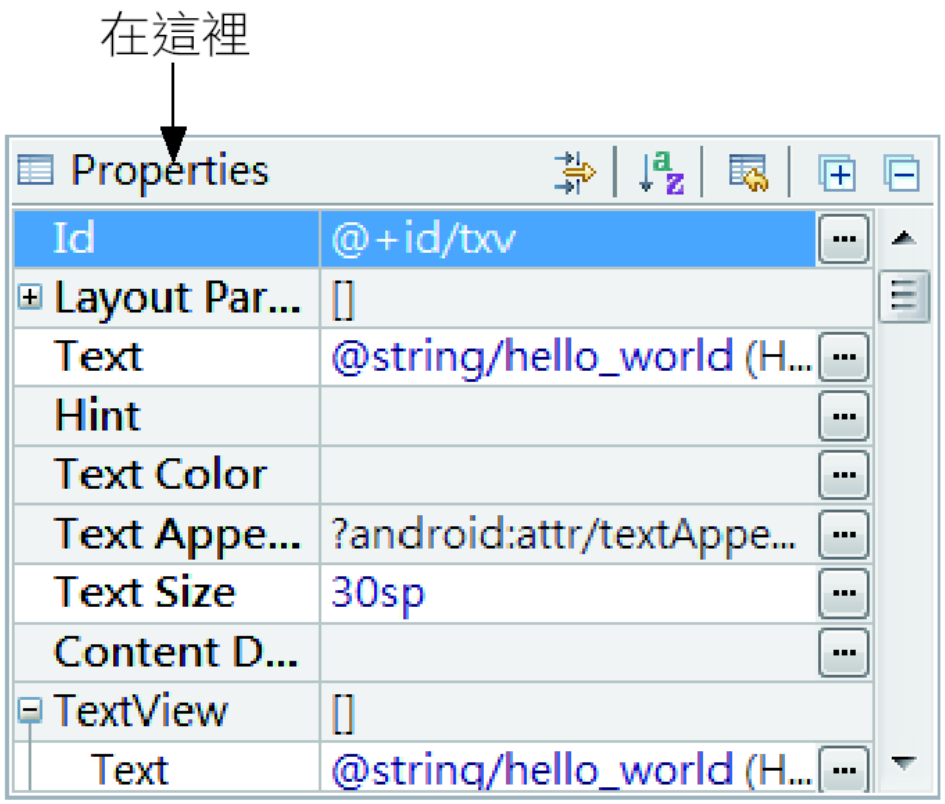
資源的 ID


R.java 在專案的 gen/package 名稱/目錄下
public final class R {
...
public static final class drawable {
public static final int ic_launcher=0x7f020000;
}
...
public static final class layout {
public static final int activity_main=0x7f030000;
}
...
public static final class string {
public static final int app_name=0x7f040000;
public static final int hello_world=0x7f040001;
...
}
...
}
res
drawable-hdpi
ic_launcher.png
drawable-ldpi
drawable-mdpi
drawable-xhdpi
layout
activity_main.xml
menu
values
strings.xml
styles.xml
小綠人圖示
佈局檔
字串
res 資料夾的內容
ADT
R.java 的內容

資源的 ID

```
R.layout.activity_main   ◄── 可取得 layout 中的 activity_main.xml 佈局檔
R.string.app_name        ◄── 可取得 strings.xml 中的 app_name 字串內容
```

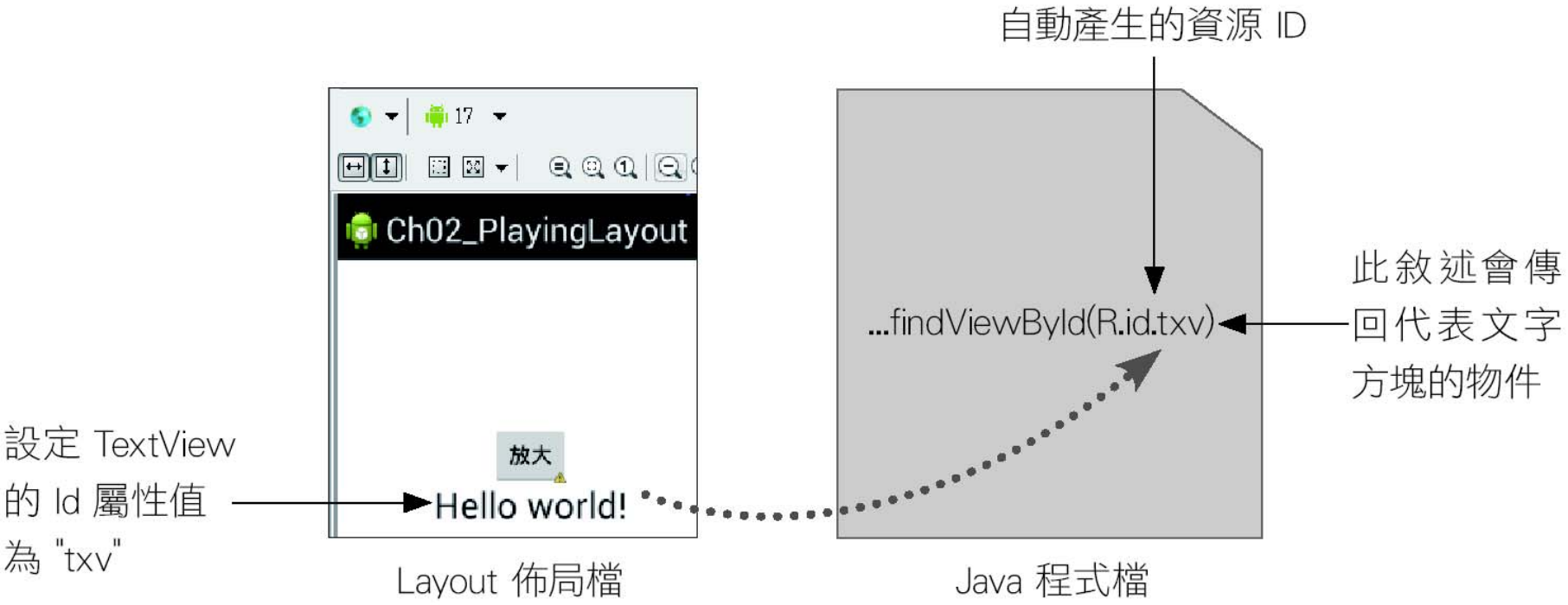
2-4 元件的佈局與屬性設定



元件的佈局與屬性設定

- Id 屬性
- findViewById() 方法
- 常見的屬性類型

findViewById() 方法



findViewById() 方法

```
 // 取得 TextView 物件                    ← 它傳回的是 View 基礎類別的物件
TextView myTxv = (TextView) findViewById(R.id.txv);
                  ↑ 轉換成 TextView 類別
// 呼叫 TextView 類別的 setText() 方法
myTxv.setText("請問芳名");   ← 設定 (set) 元件顯示的文字 (Text)為 "請問芳名"
```

常見的屬性類型

屬性	屬性值舉例	說明
Id	@+id/button1	通知 ADT 建立新的資源 ID
Above	@id/txv	將元件放置在畫面中名稱為 txv 的元件上方
On Click	bigger	表示當元件被『按一下』後，要執行 Activity 中定義的 bigger() 方法
Text	@string/hello_world	將 res/values 資料夾下的 strings.xml 檔中名稱為 hello_world 的字串內容顯示在元件上
Text Size	30sp	將元件上顯示的文字設定為 30sp 大小
Src	@drawable/ic_launcher	將 res/drawable-XXX 資料夾下主檔名為 ic_launcher 的圖形顯示在元件上

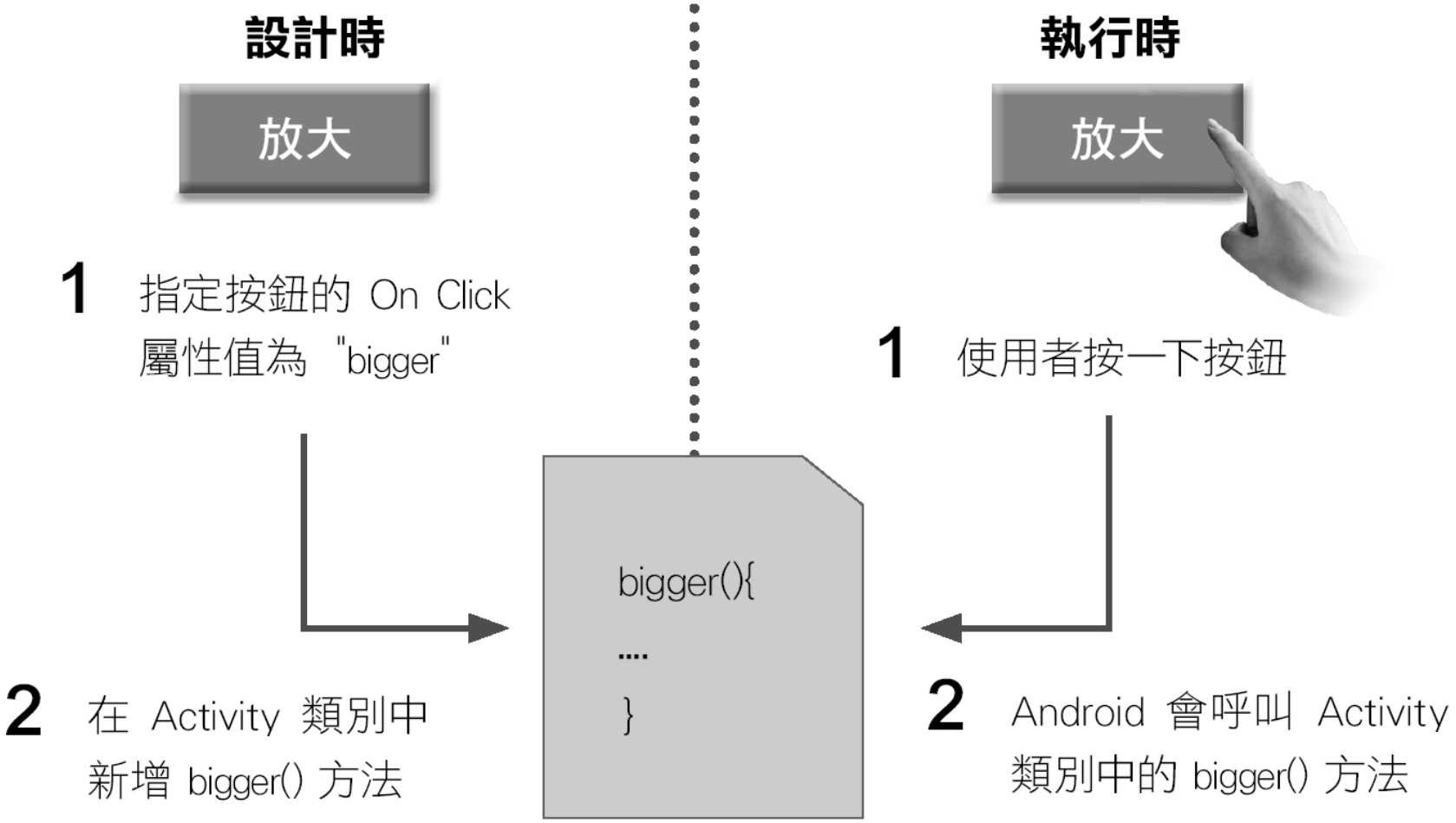
＊"@+id/名稱" 代表如果名稱不存在，就要建立此 id 名稱。而 "@id/名稱" 則是要使用此名稱。

常見的屬性類型

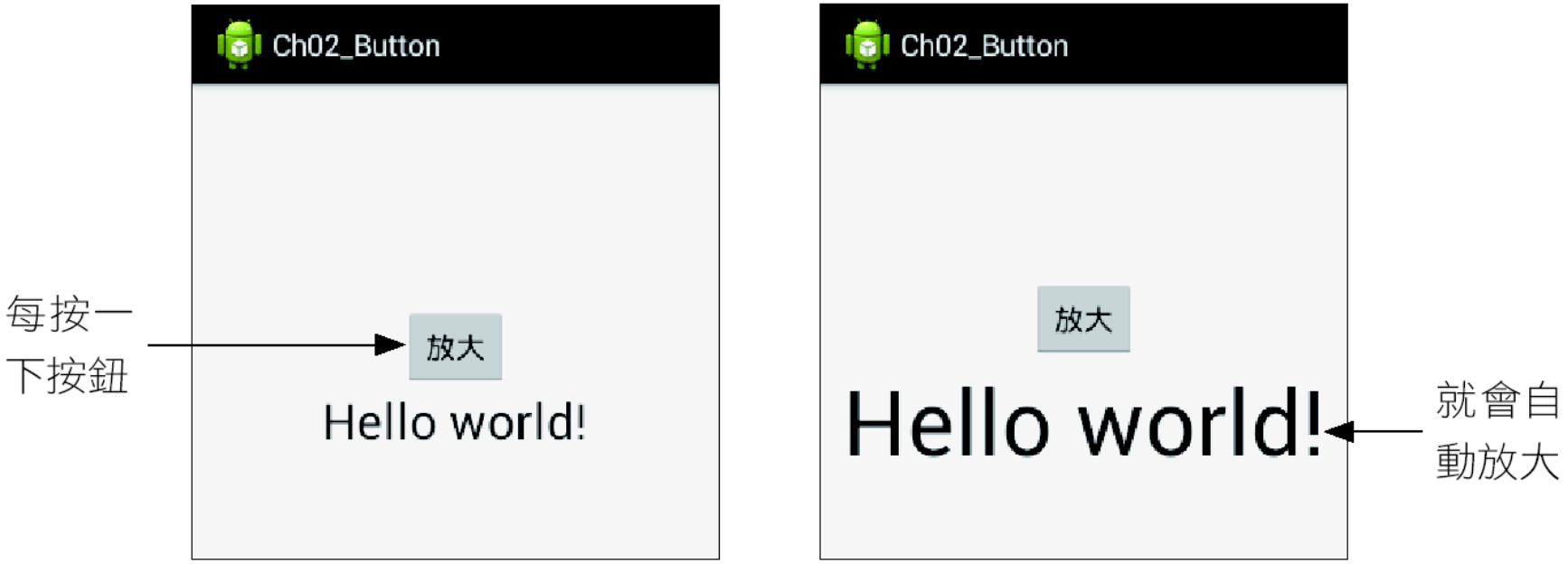
- 在屬性中直接設值
- 在屬性中設參照 (Reference)
- 在屬性中設方法的名稱

```
public void bigger(View v) {
   ...// 放大 TextView 文字的程式碼
}
```

常見的屬性類型


設計時
放大
1 指定按鈕的 On Click 屬性值為 "bigger"
2 在 Activity 類別中新增 bigger() 方法
bigger(){
....
}
執行時
放大
1 使用者按一下按鈕
2 Android 會呼叫 Activity 類別中的 bigger() 方法

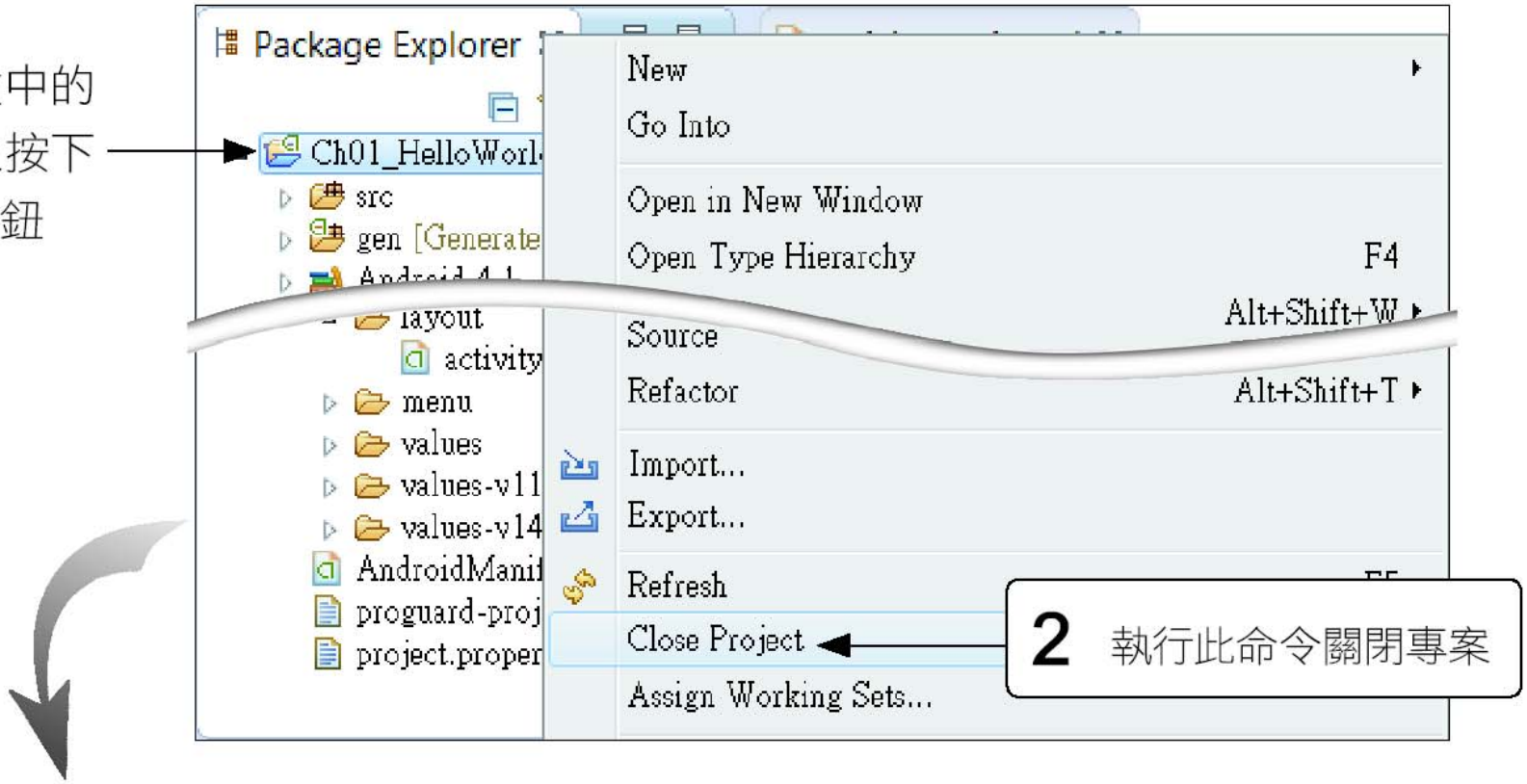
2-5 開始動手寫程式


Ch02_Button
每按一
下按鈕
放大
Hello world!
Ch02_Button
放大
Hello world!
就會自
動放大

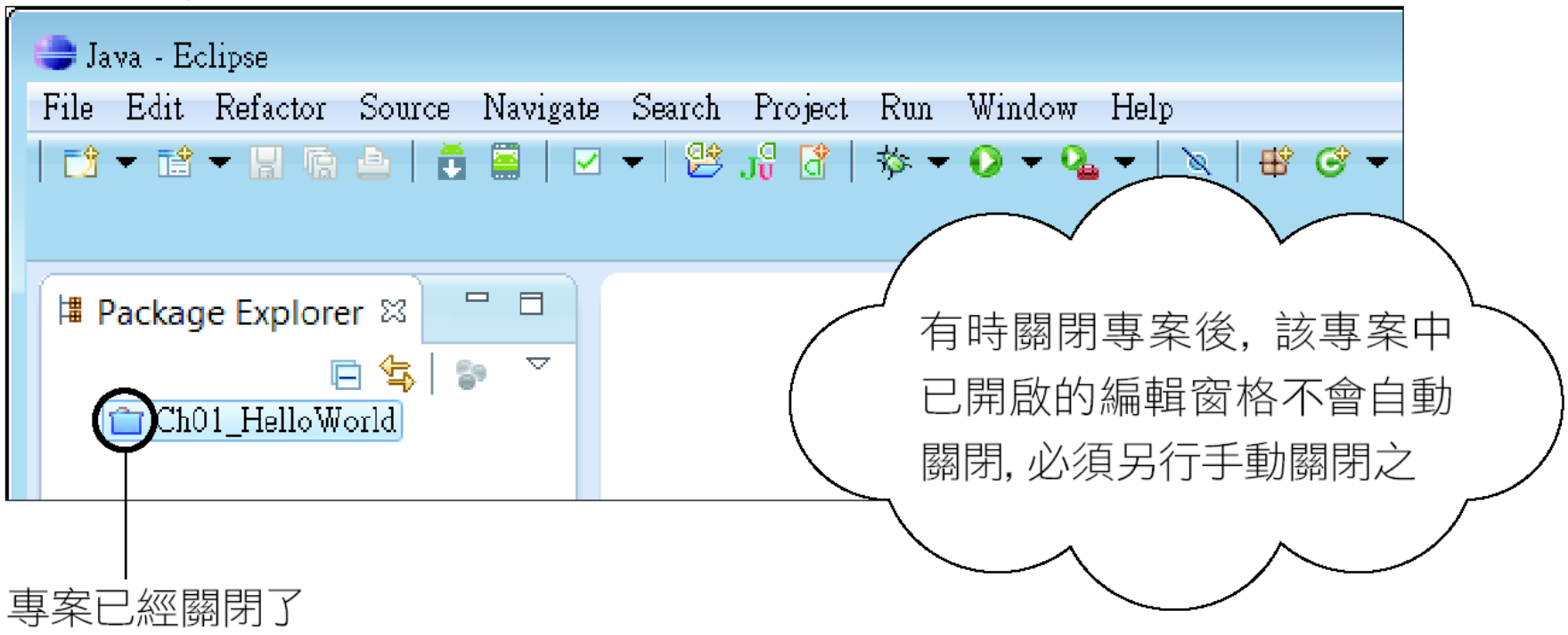
範例2-1 ：按一下按鈕就放大顯示的文字

step 1

1 在開啟中的專案上按下滑鼠右鈕



按一下按鈕就放大顯示的文字

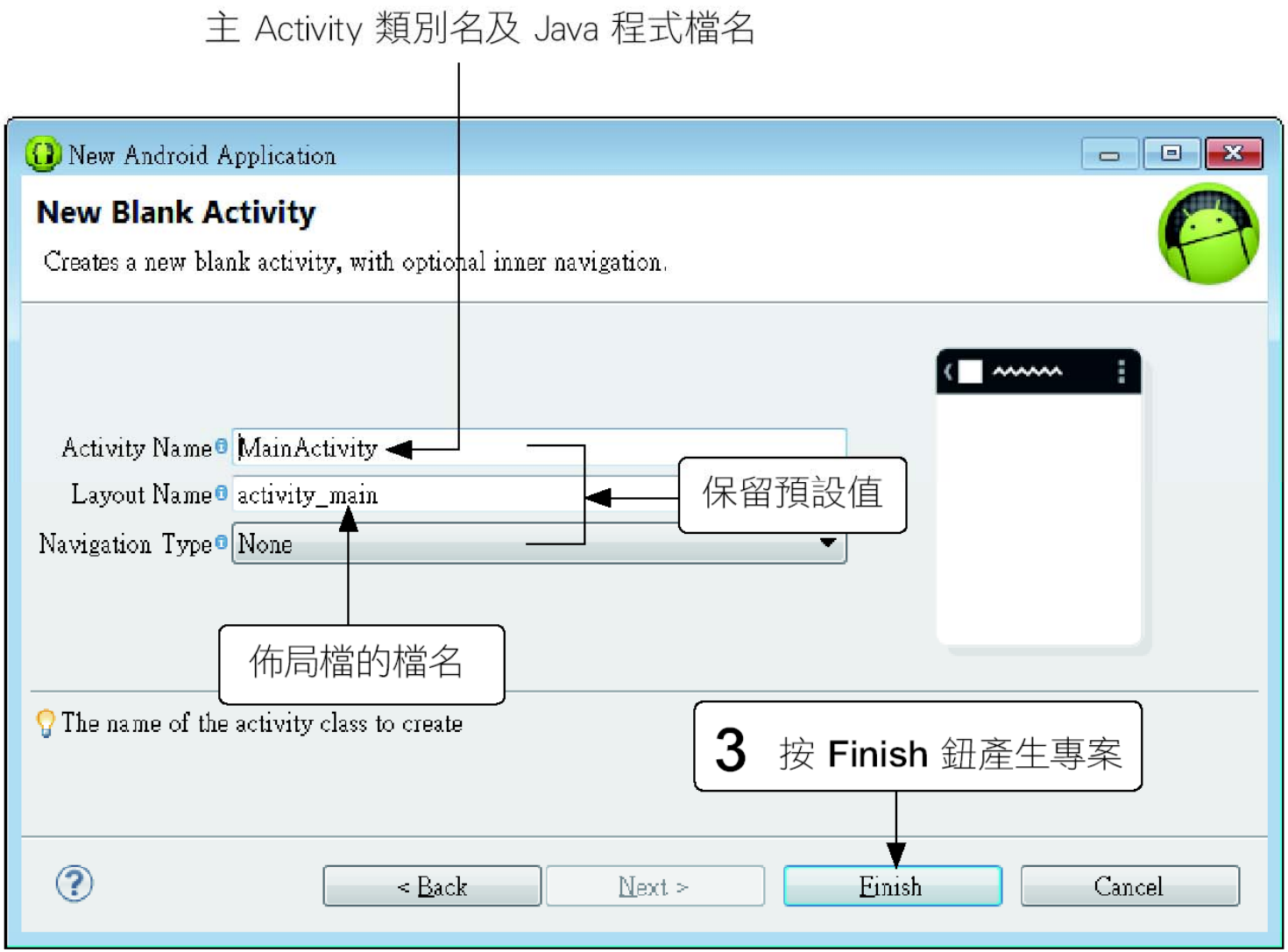


按一下按鈕就放大顯示的文字

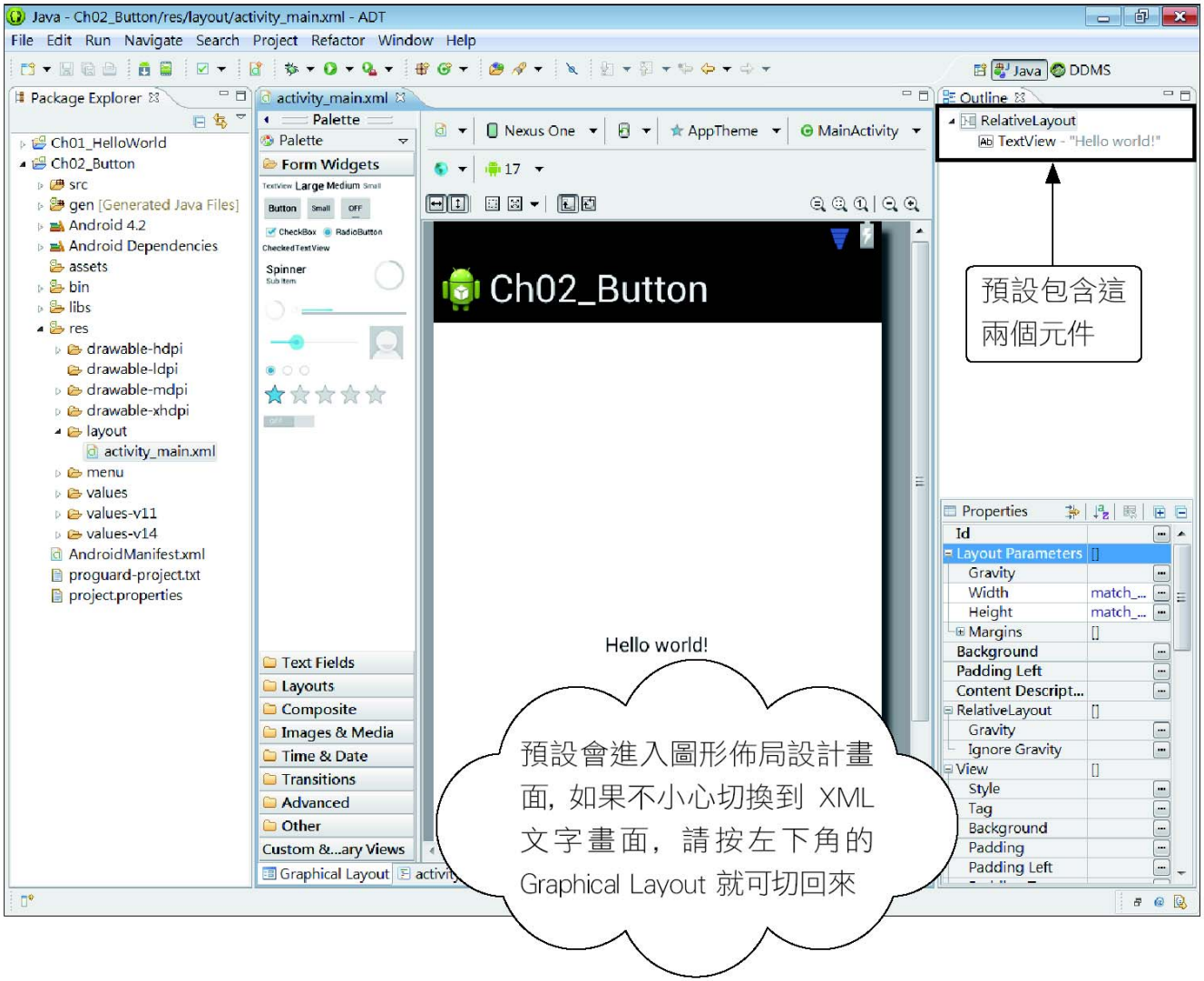
step 2

Application Name	Ch02_Button
Project Name	Ch02_Button
Package Name	tw.com.flag.ch02_button

按一下按鈕就放大顯示的文字



按一下按鈕就放大顯示的文字

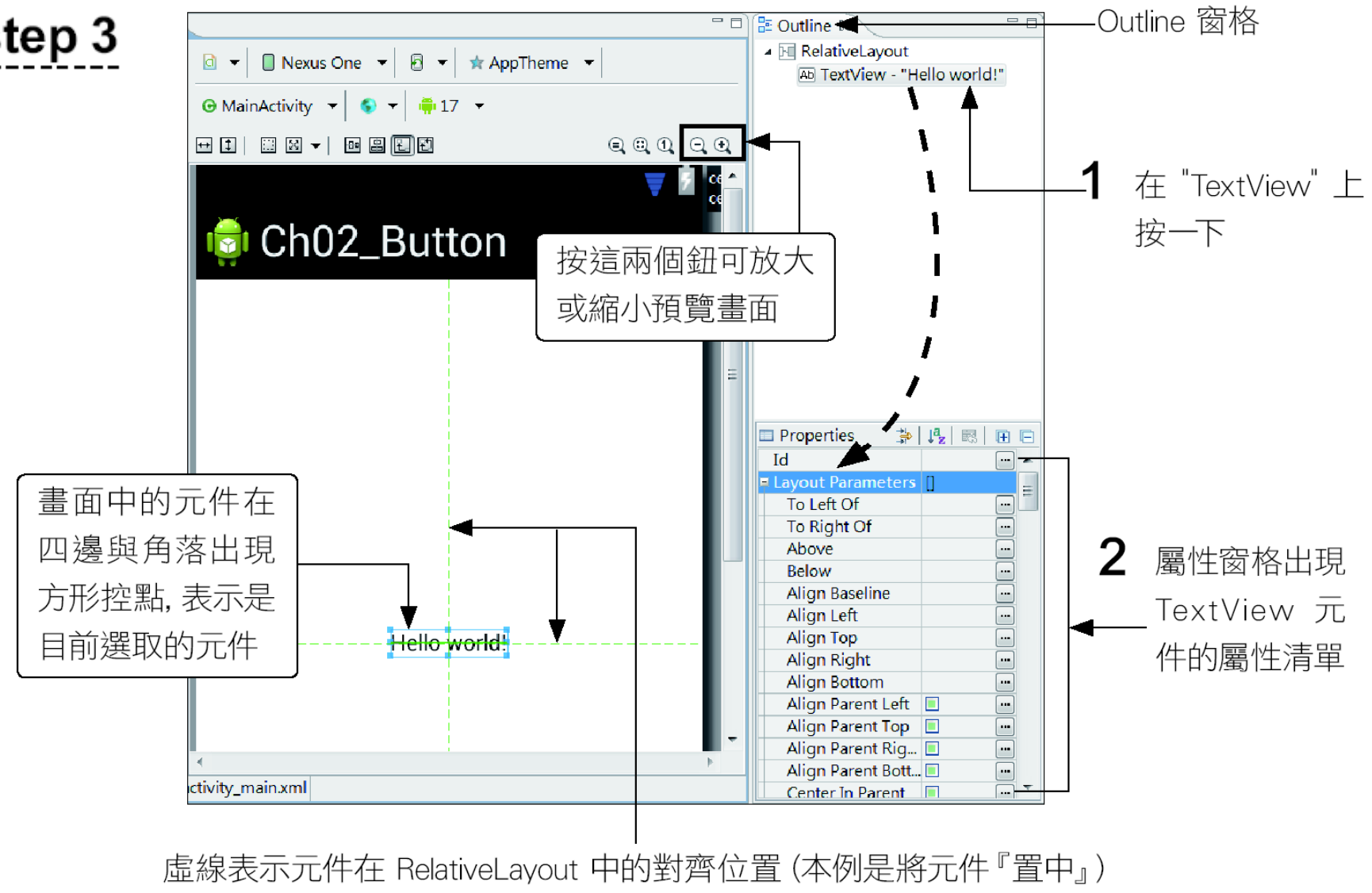


按一下按鈕就放大顯示的文字

- **RelativeLayout** ：RelativeLayout 表示是透過『相對 (Relative) 位置』來規劃畫面上元件的位置 (包括『元件與元件』或『元件與 Layout』之間的相對位置)。下一章會介紹另一種佈局元件。
- **TextView 元件**：可稱為『文字標籤』、『文字方塊』，它的用途就是用來顯示一段文字，例如預設顯示 “Hello world!" 字串。若您畫面中的 "Hello world!” 是出現在其他位置，請先用滑鼠將其拖曳到畫面正中央，以利後續操作。

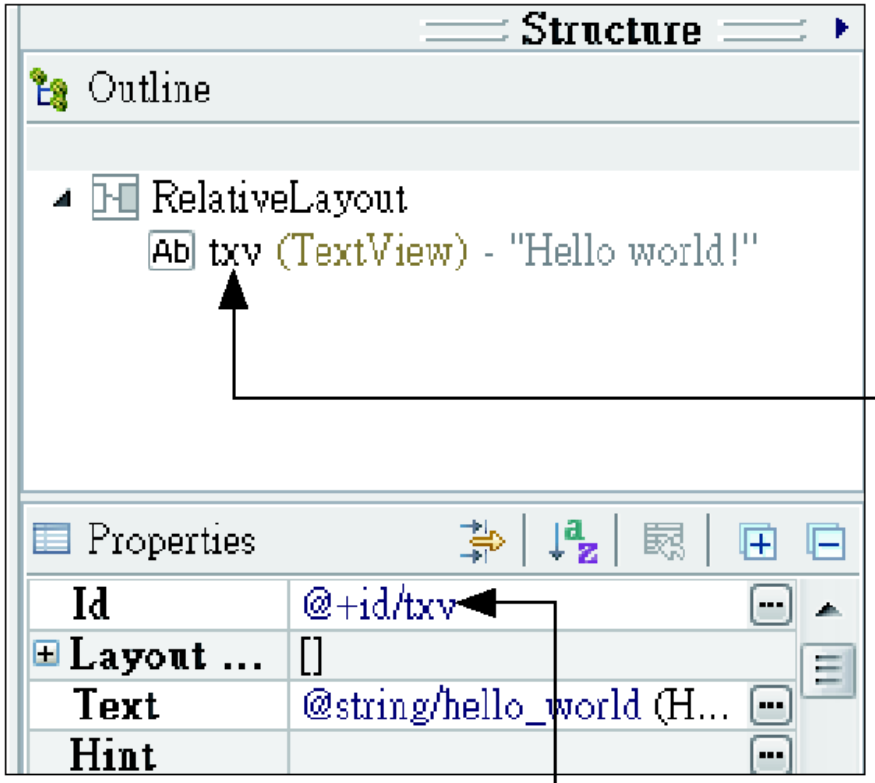
按一下按鈕就放大顯示的文字

step 3



按一下按鈕就放大顯示的文字

step 4



2 在 **Outline** 窗格會顯示新的名稱："txv"

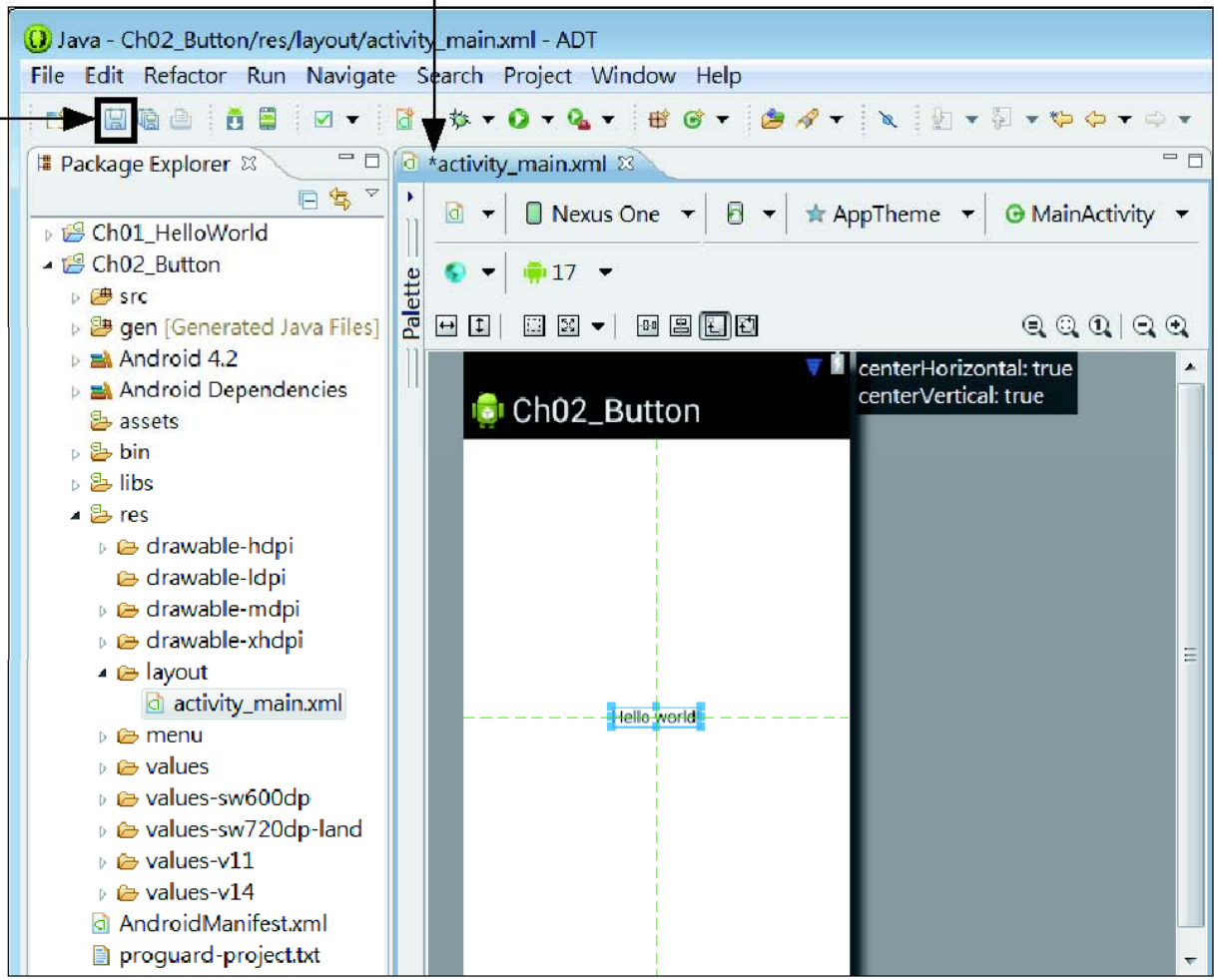
1 在 Id 屬性欄位填入 "@+id/txv" 將 TextView 元件命名為 "txv"

按一下按鈕就放大顯示的文字

step 5

存檔後, 前頭的 '*' 符號會消失

請按此鈕 (或按 Ctrl + S 鍵) 存檔

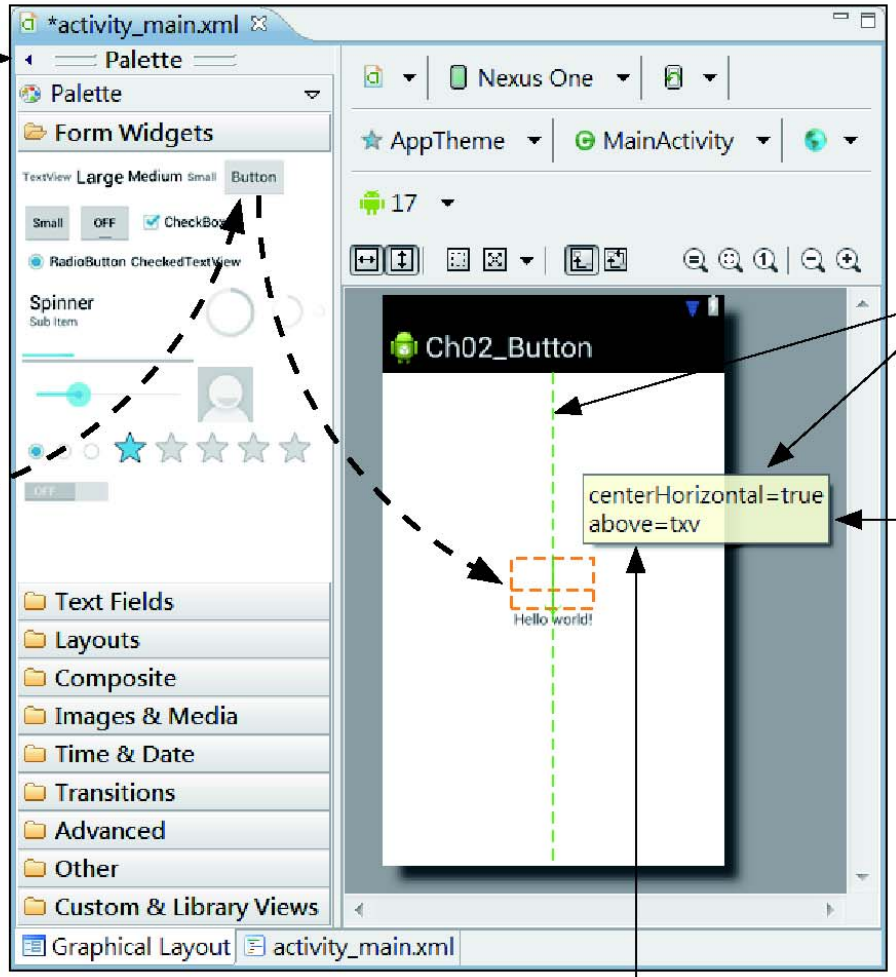


按一下按鈕就放大顯示的文字

step 6

如果 **Palette** 是被收起的狀態，可按標題上的三角箭頭圖示將其展開

1 按住 **Button** 不放，將元件拉曳到 TextView (就是 Hello World 文字) 的『正』上方放開

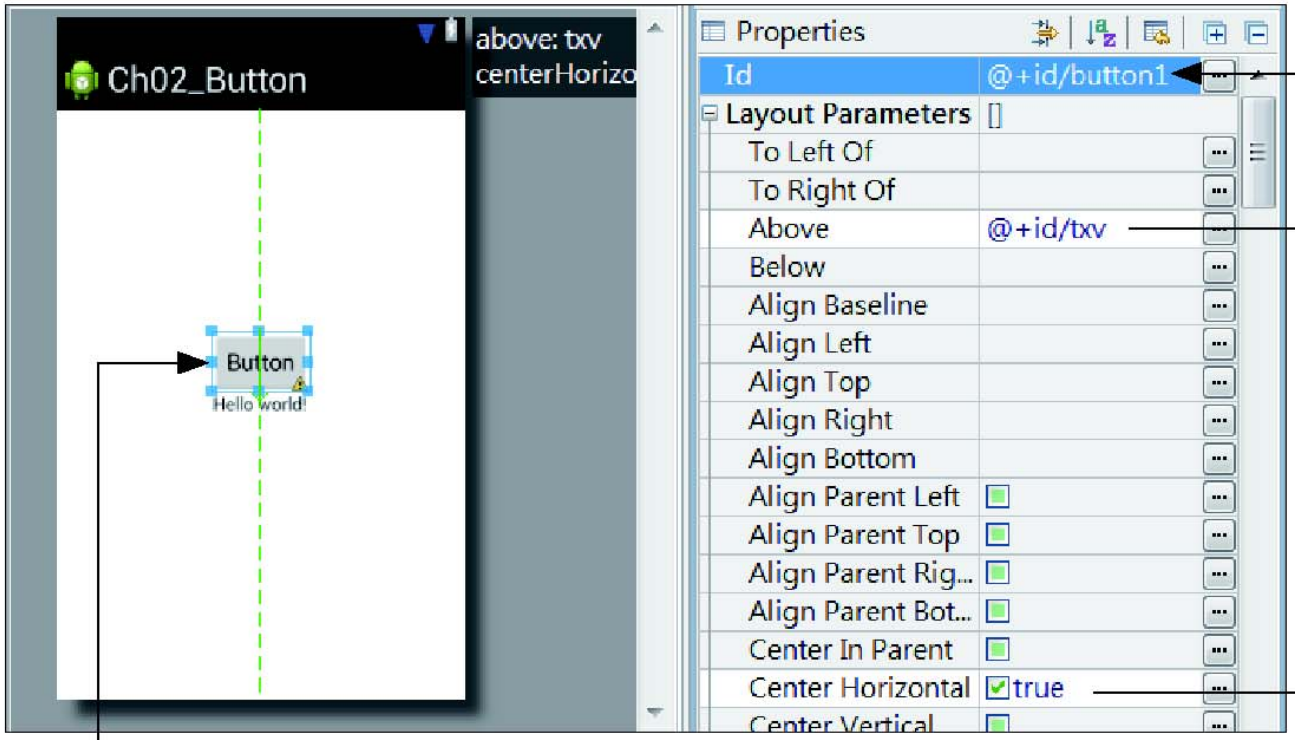


放開滑鼠鈕前，編輯器會用線條及浮動視窗提示元件的 Layout 相關屬性

表示元件『水平置中』

表示元件置於 txv 元件之上 (above)

按一下按鈕就放大顯示的文字

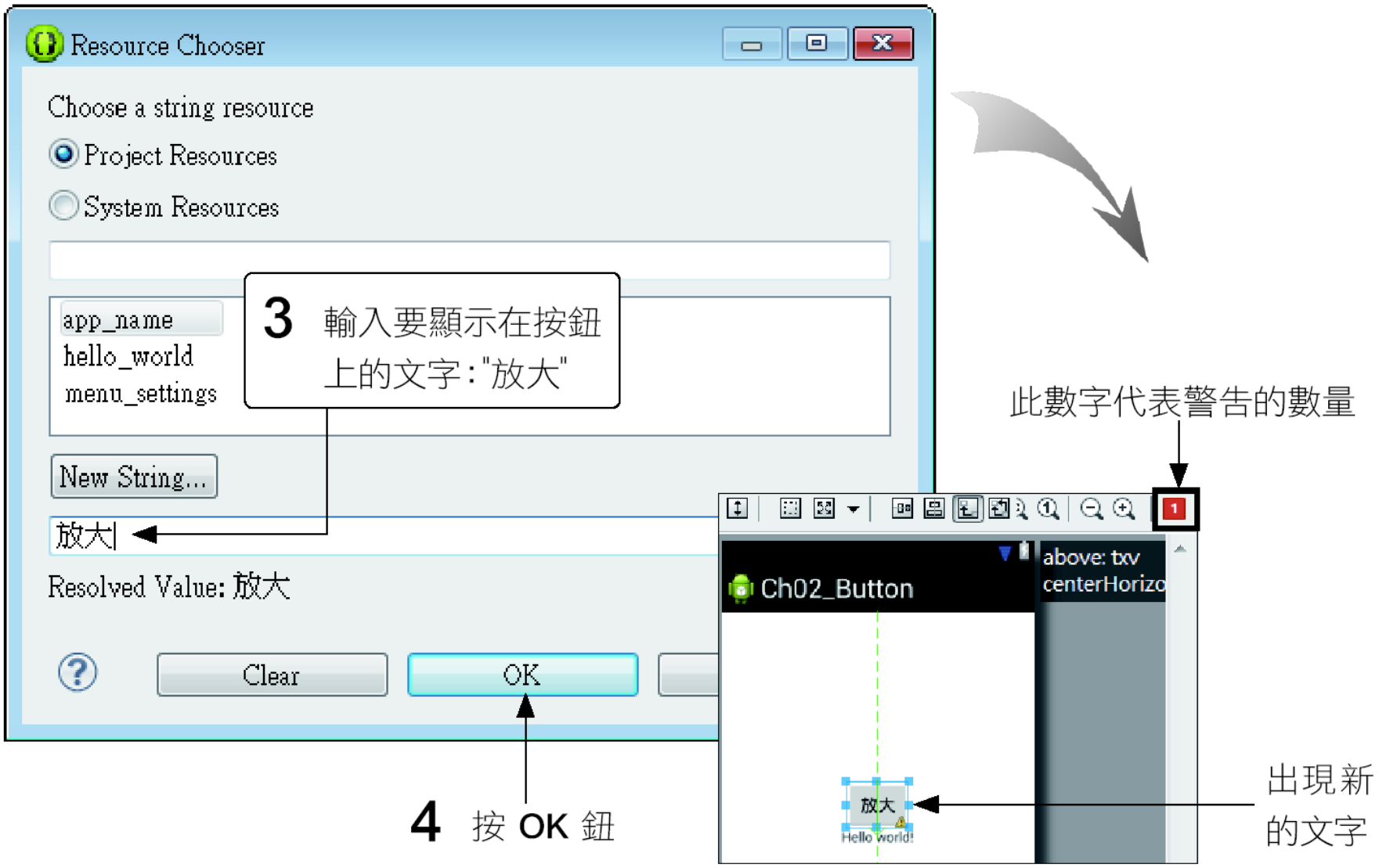


新加入元件的預設 Id (通常是『元件類型』加『流水號』1, 2, 3...)

放開滑鼠鈕, 屬性窗格會出現和先前浮動視窗內相同的設定 (若看不到, 可按 **Layout Parameter** 前的加號圖示)

2 在新加入的元件上按滑鼠右鈕, 執行『**Edit Text**』命令

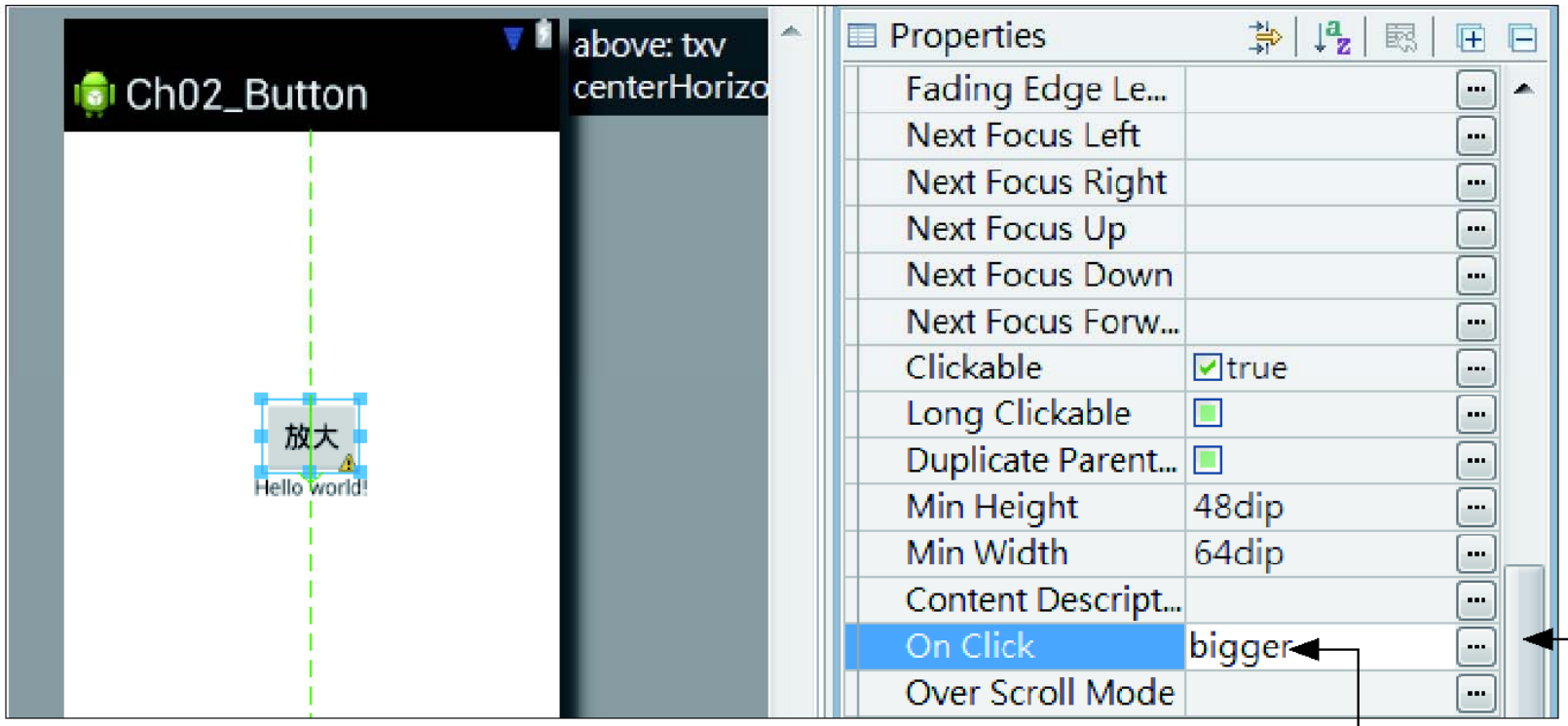
按一下按鈕就放大顯示的文字



按一下按鈕就放大顯示的文字

step 7

1 將屬性清單捲動到最後面, 找到 **On Click** 屬性

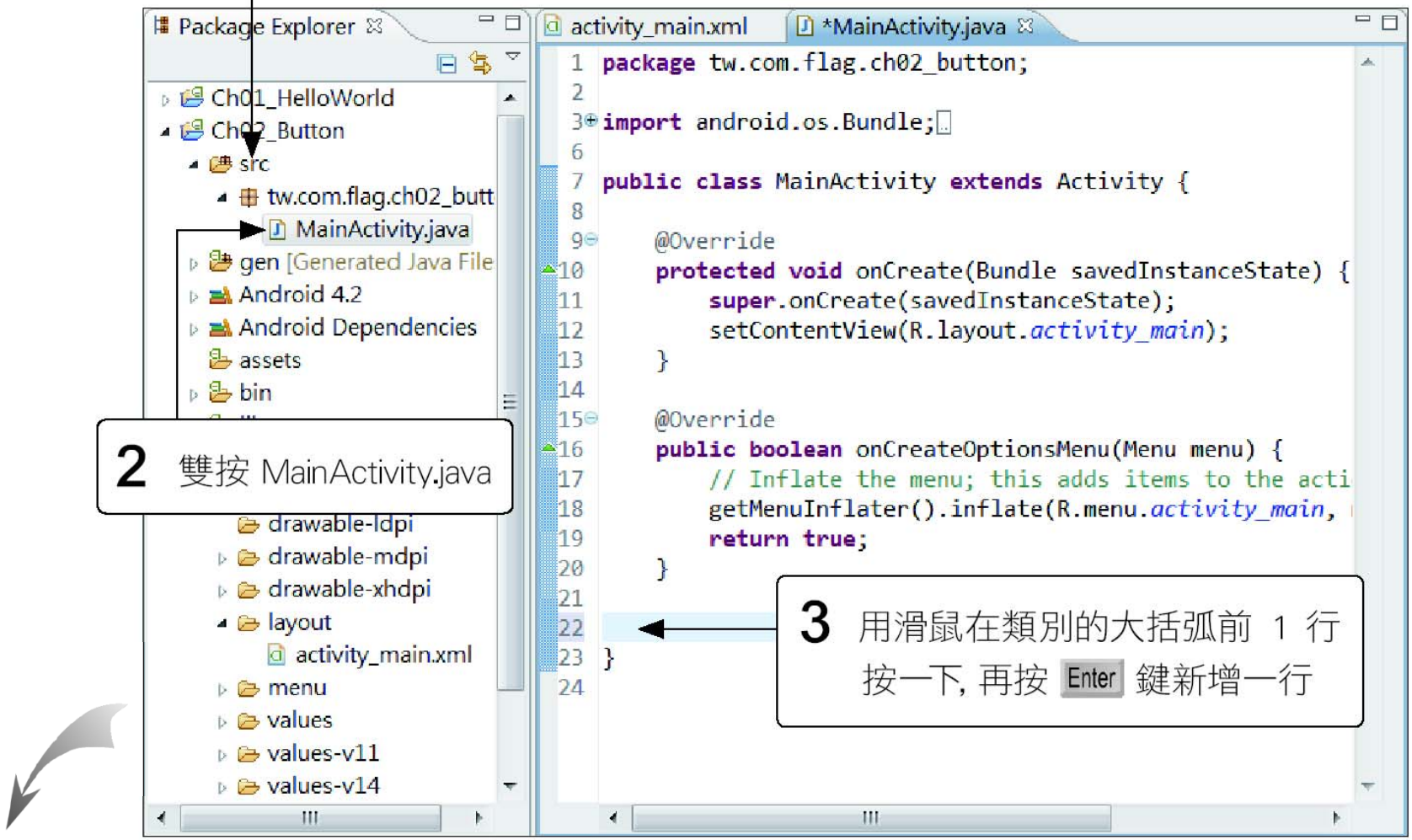


2 在此欄位輸入 "bigger"

按一下按鈕就放大顯示的文字

step 8

1 展開 src 及其下的套件名稱



按一下按鈕就放大顯示的文字

activity_main.xml *MainActivity.java

```
1  package tw.com.flag.ch02_button;
2
3  import android.os.Bundle;
6
7  public class MainActivity extends Activity {
8
9      @Override
10     protected void onCreate(Bundle savedInstanceState) {
11         super.onCreate(savedInstanceState);
12         setContentView(R.layout.activity_main);
13     }
14
15     @Override
16     public boolean onCreateOptionsMenu(Menu menu) {
17         // Inflate the menu; this adds items to the acti
18         getMenuInflater().inflate(R.menu.activity_main,
19         return true;
20     }
21
22     int size=30; // 字型大小, 初始值 30(sp)
23 }
24
```

4 加入一個變數用來記錄字型大小

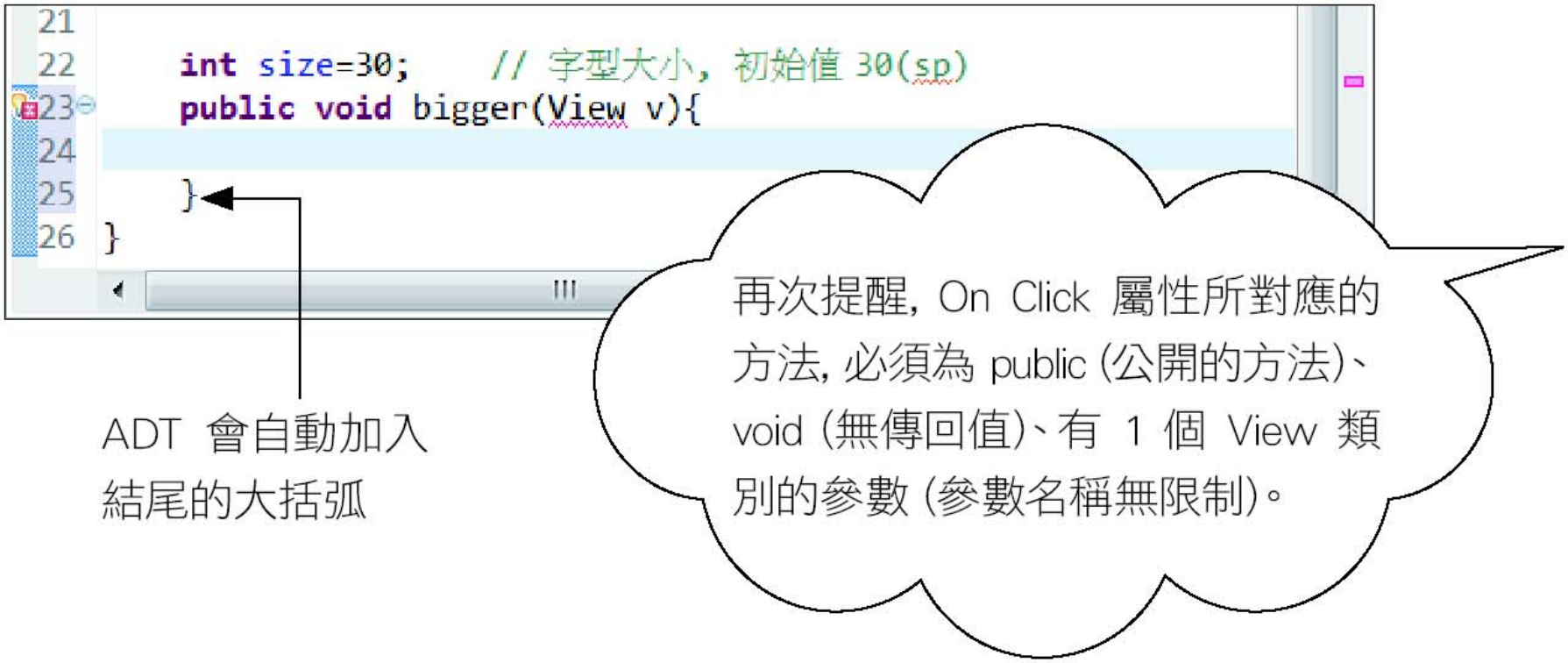
按一下按鈕就放大顯示的文字

輸入過程中, 會出現紅色波浪表示錯誤, 稍後再排除

5 輸入方法的定義

6 按 Enter 鍵

按一下按鈕就放大顯示的文字



按一下按鈕就放大顯示的文字

step 9

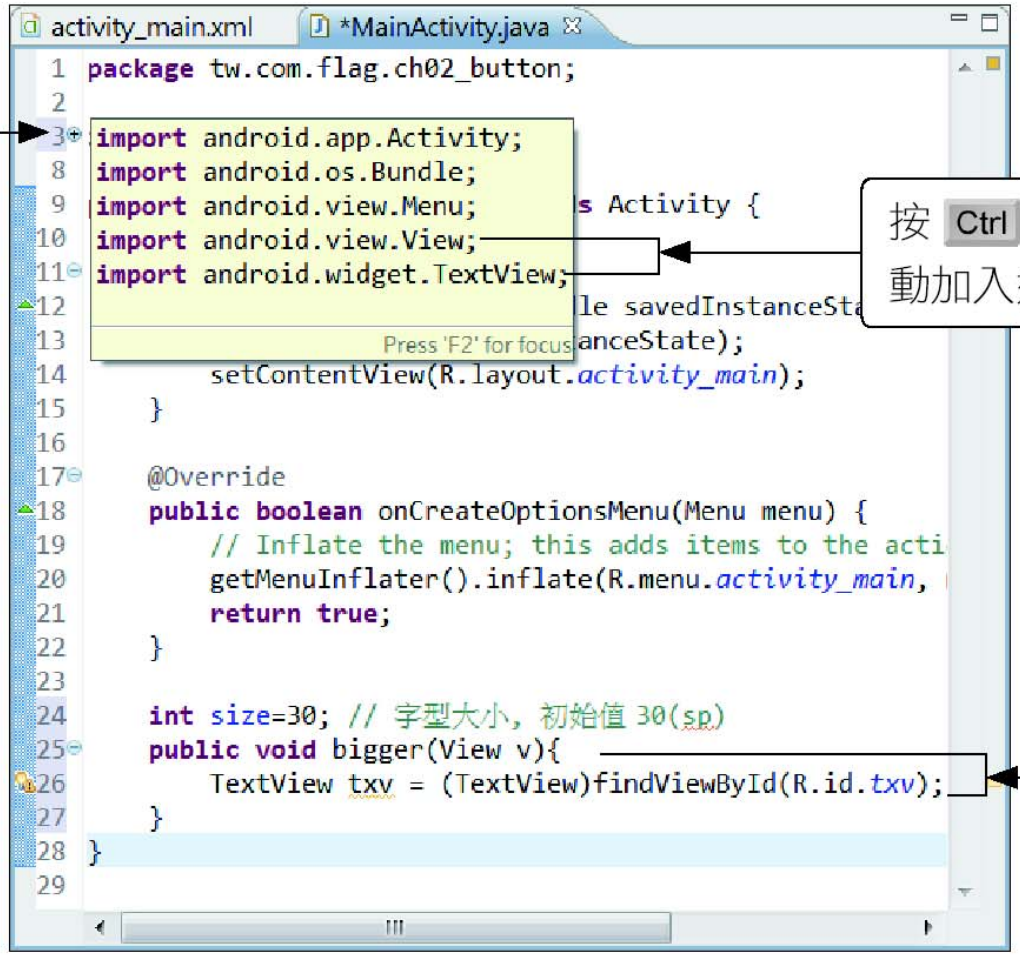
```
22      int size=30; // 字型大小，初始值 30(sp)
23      public void bigger(View v){
24          TextView txv = (TextView)findViewById(R.id.txv);
25      }
26  }
27
```

1 輸入這一行程式

2 按 Ctrl ＋ Shift ＋ O 鍵

按一下按鈕就放大顯示的文字

3 將滑鼠移到第 3 行程式前面的加號圖示，會顯示所有的 import 敘述

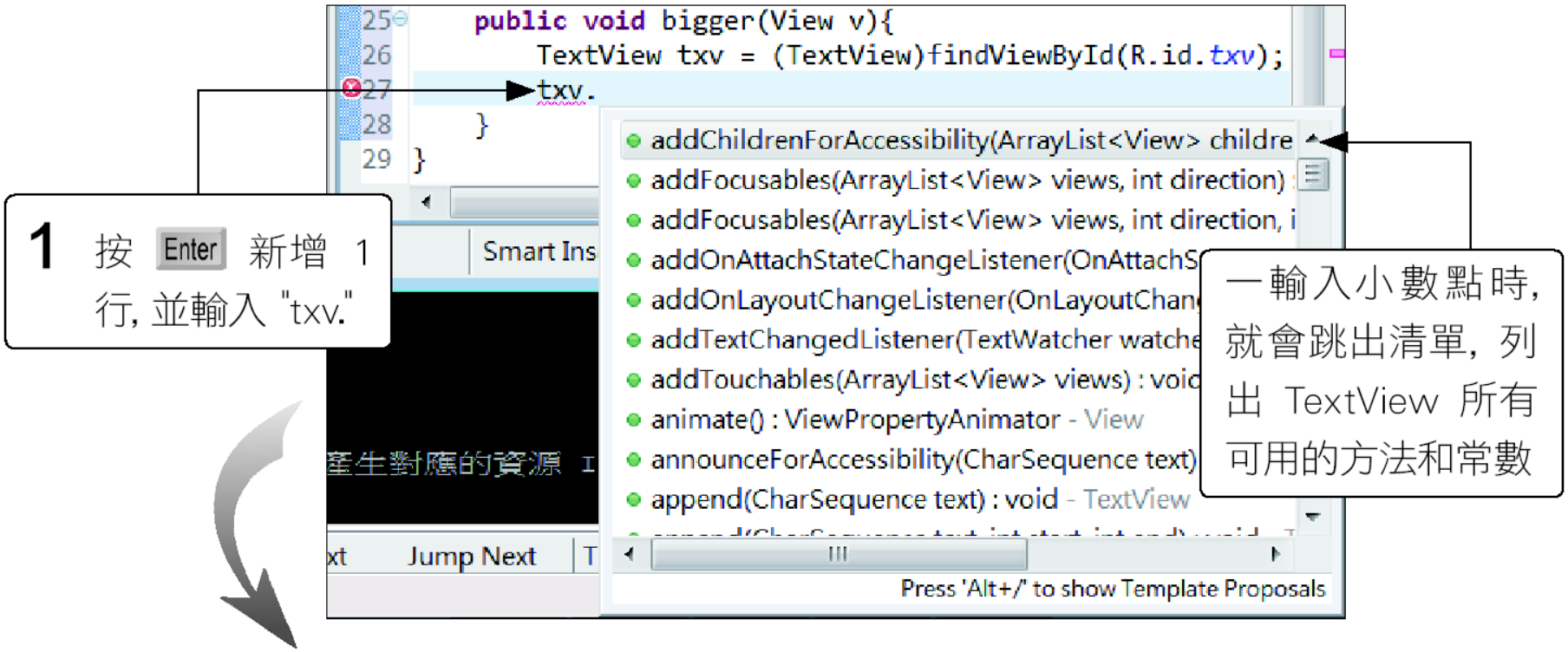


按 Ctrl + Shift + O 鍵會自動加入這 2 行 import 敘述

這 2 行的紅色波浪線都消失了

按一下按鈕就放大顯示的文字

step10

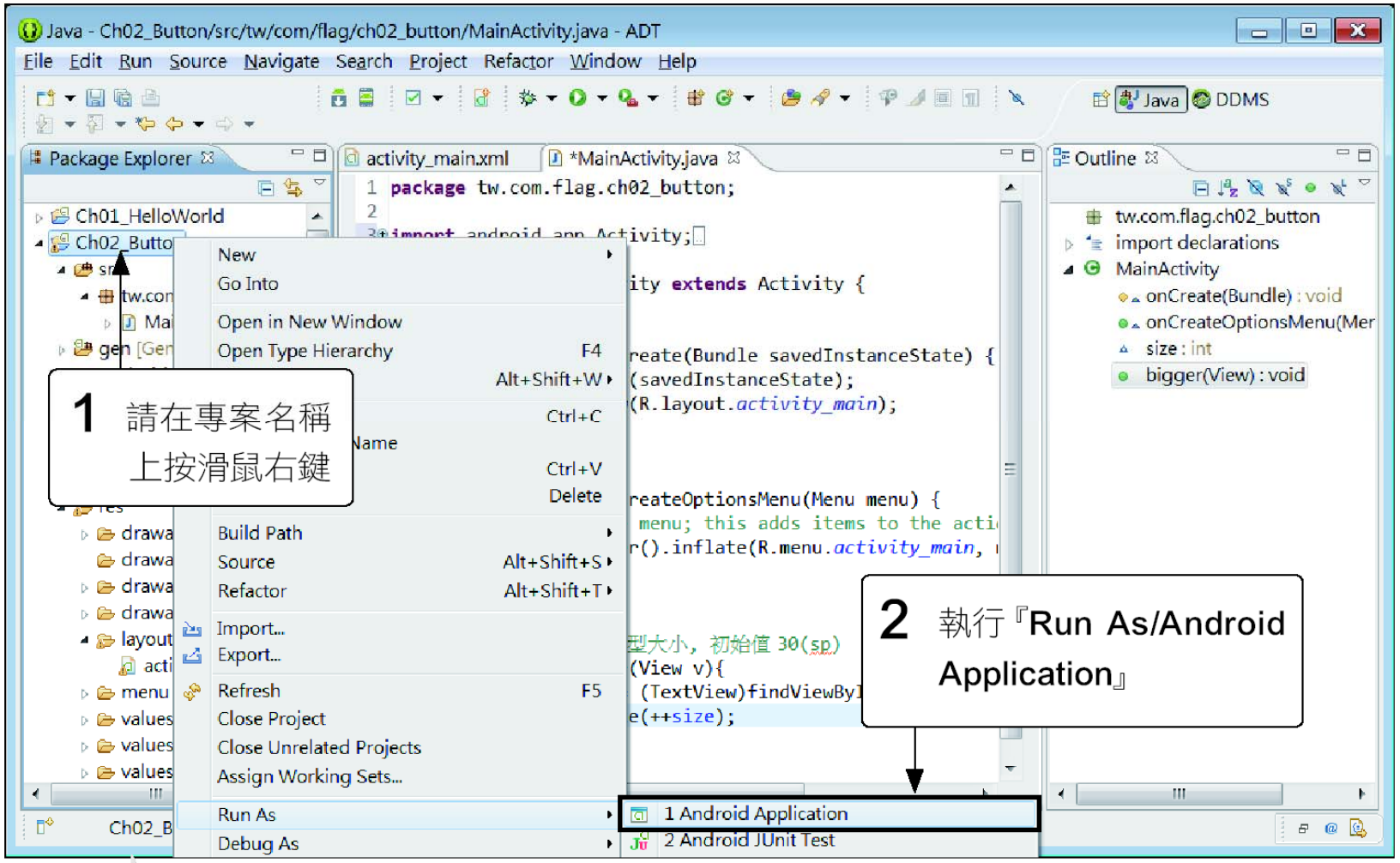


按一下按鈕就放大顯示的文字

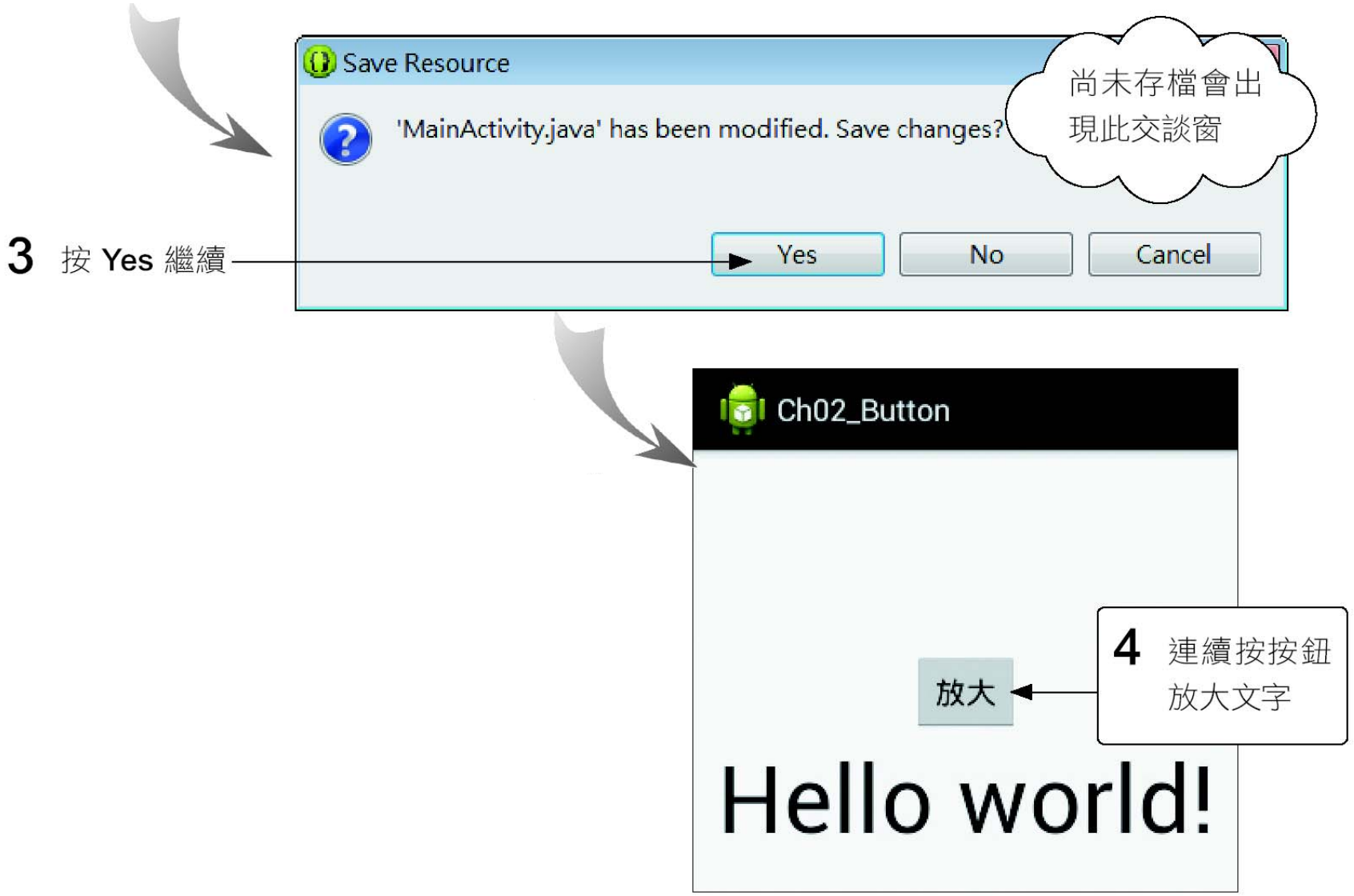
按一下按鈕就放大顯示的文字

按一下按鈕就放大顯示的文字

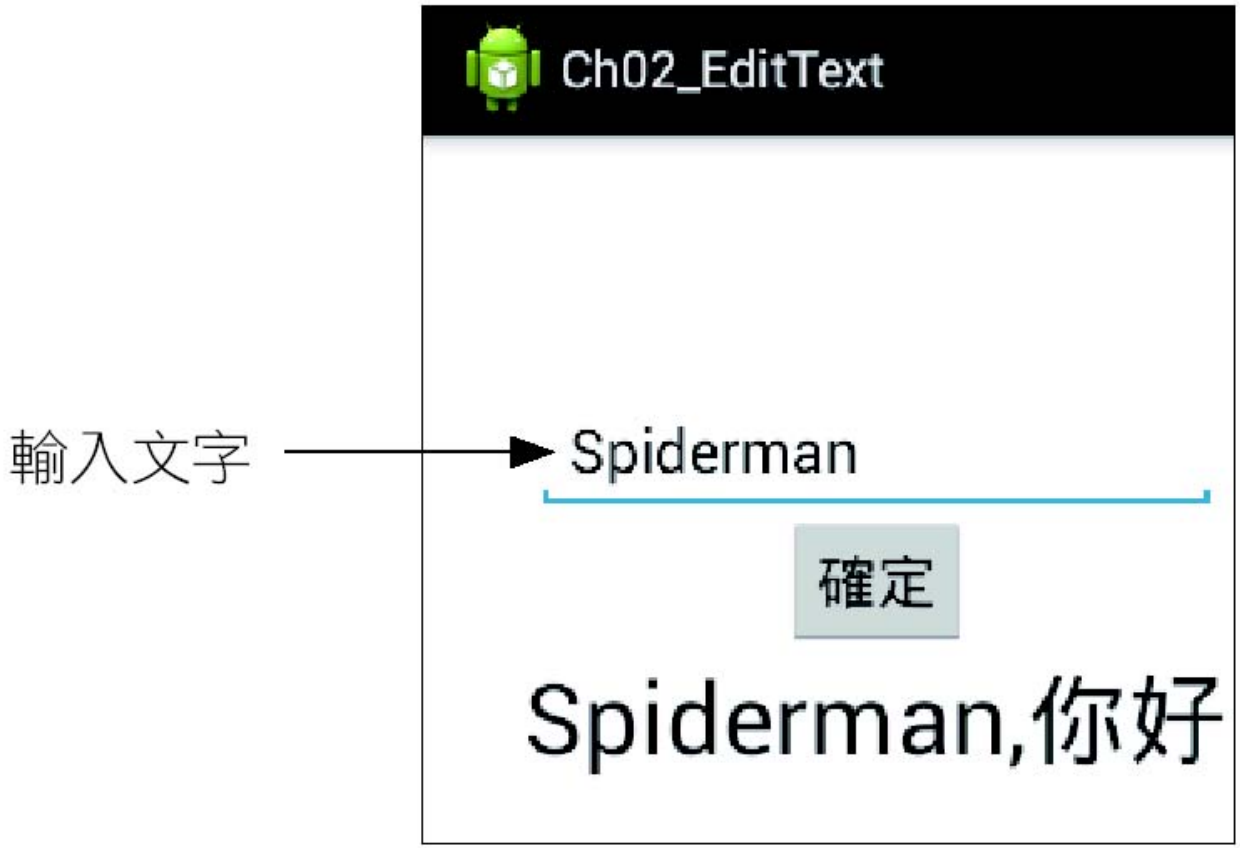
step11



按一下按鈕就放大顯示的文字



2-6 輸入欄位 EditText 元件



輸入欄位 EditText 元件

- getText()：取得使用者輸入的文字
- setText()：設定 TextView 顯示的文字

getText()：取得使用者輸入的文字

```
EditText edit = (EditText) findViewById(R.id.edit);
String str = edit.getText().toString();
```

若佈局中的元件 Id 屬性命名為 edit

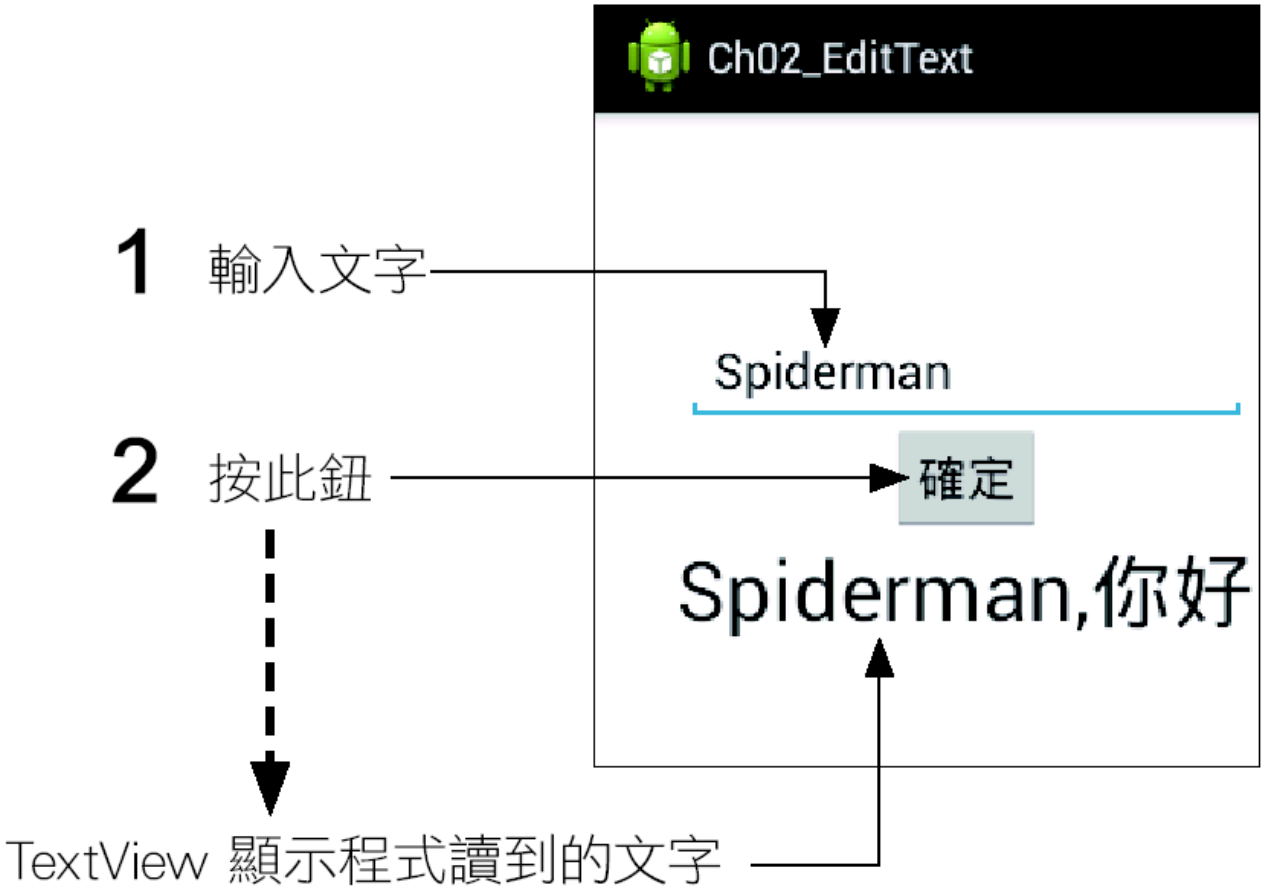
取得文字

有在佈局中設定 Id 屬性（命名）的元件才會自動產生資源 ID 喔！

setText()：設定TextView 顯示的文字

```
txv.setText("您好！");
```

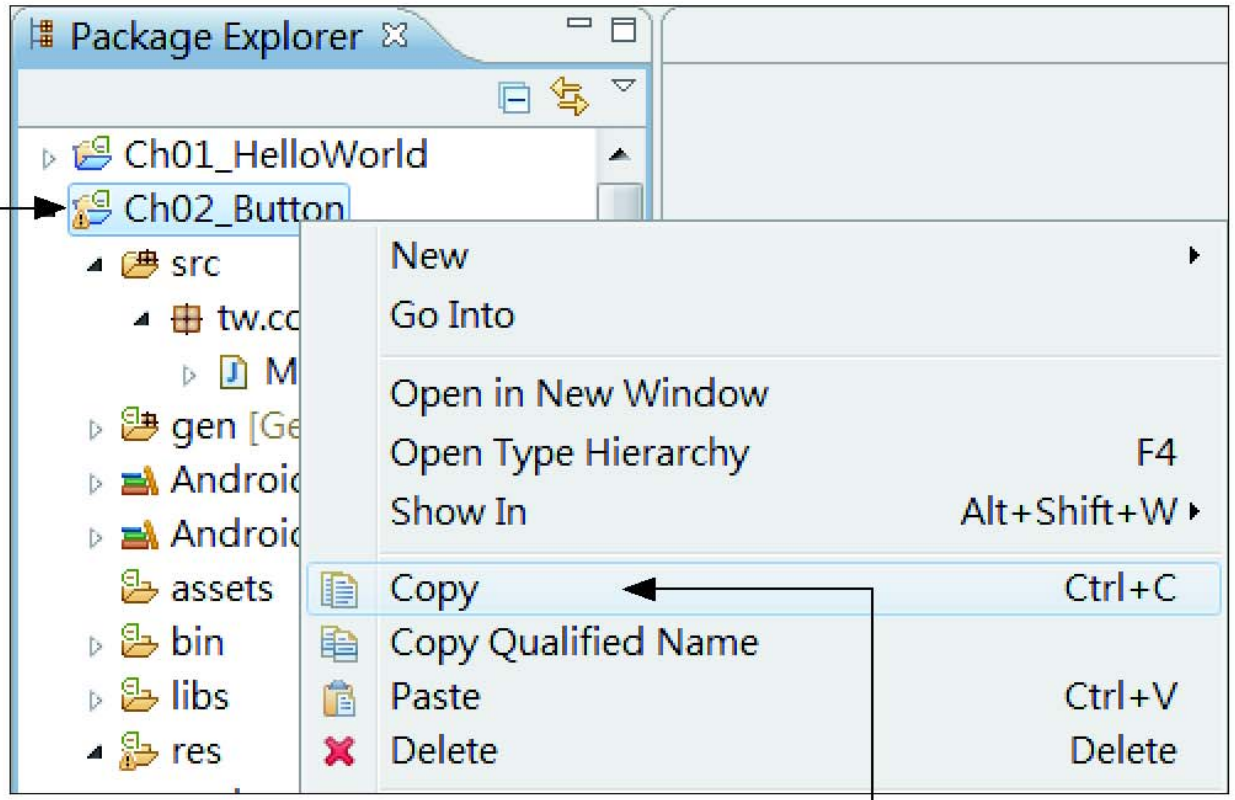
範例2-2 ：加入 EditText 元件



加入 EditText 元件

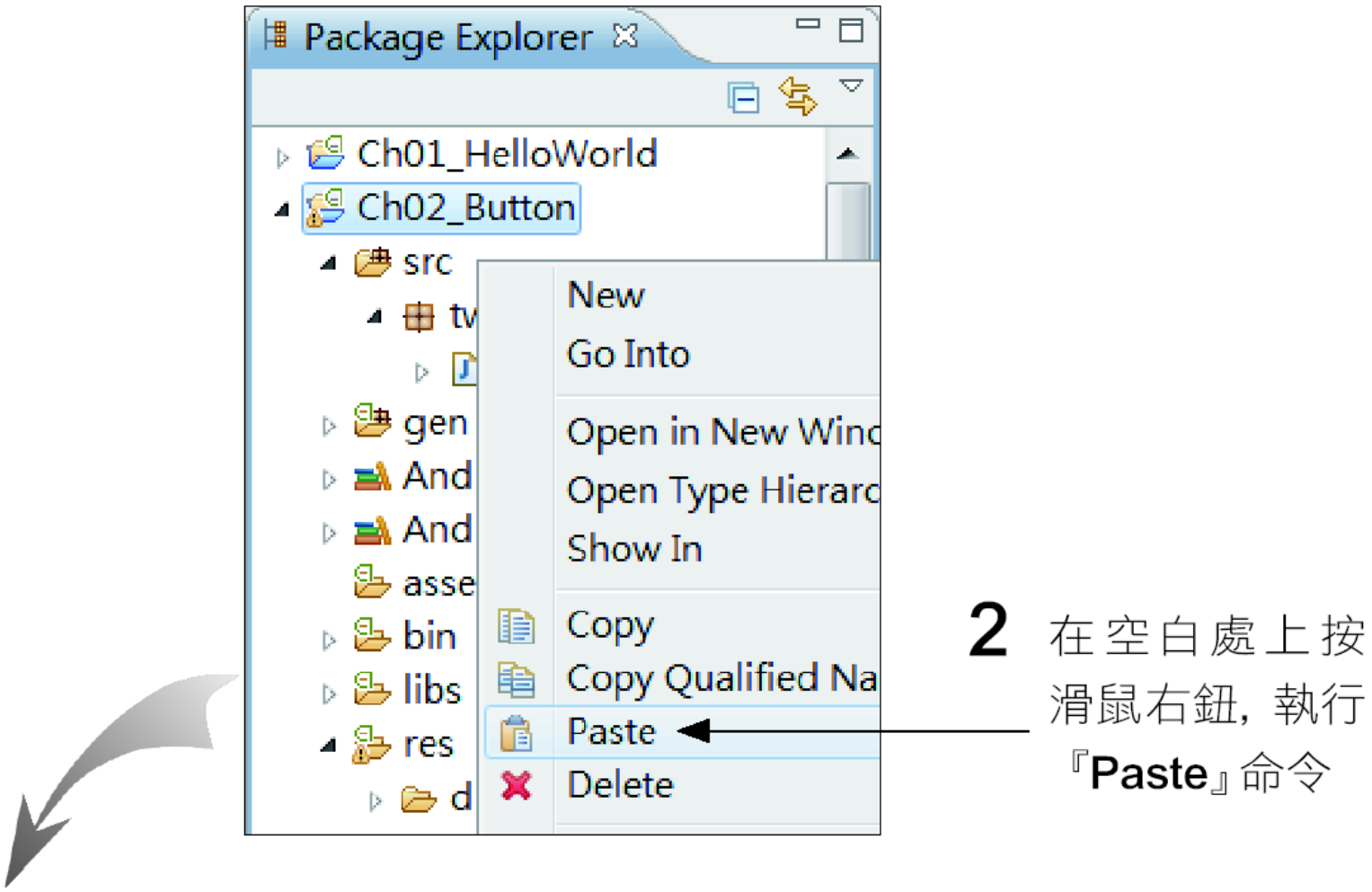
step 1

專案開啟時，才能進行複製的動作

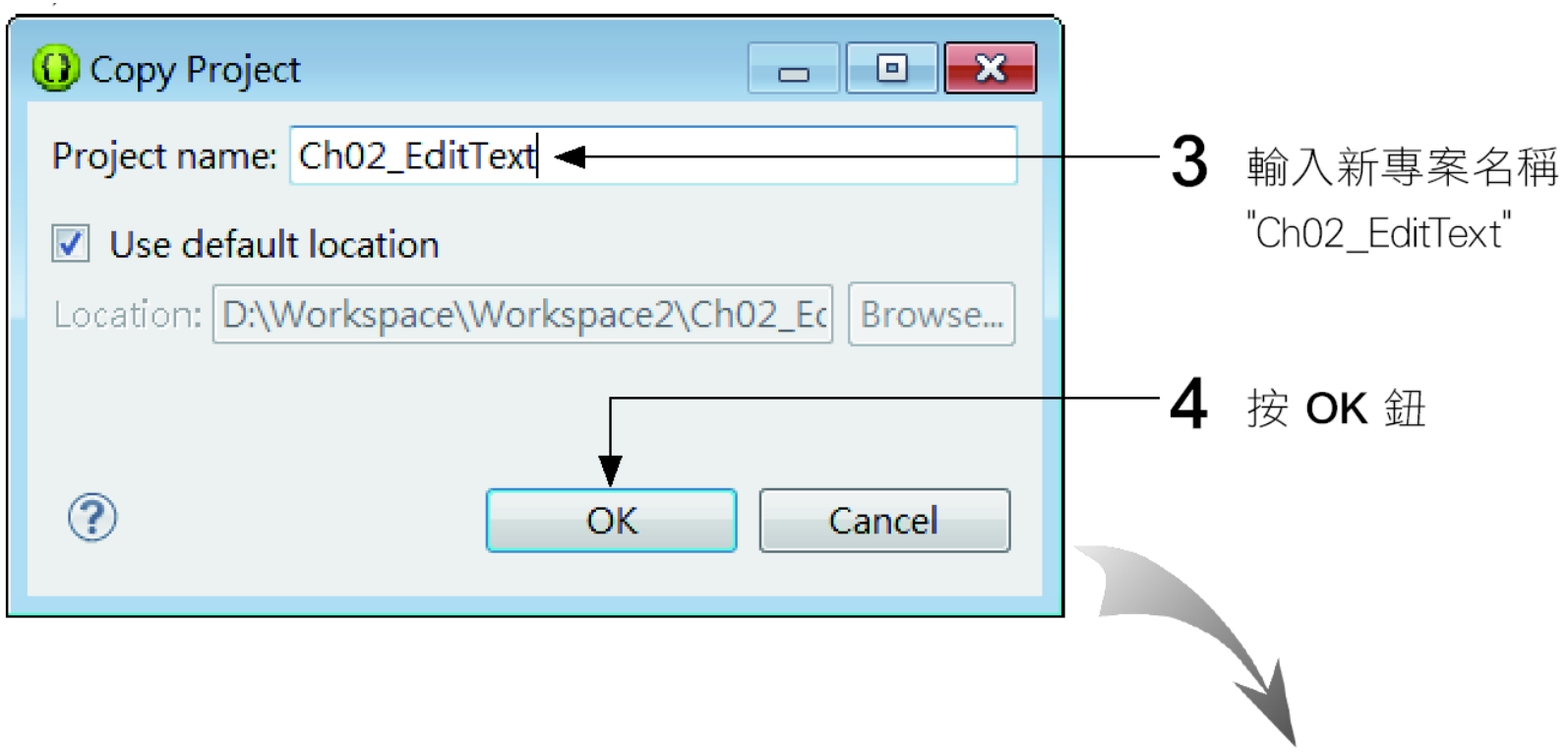


1 在現在專案上按滑鼠右鈕，執行『**Copy**』命令

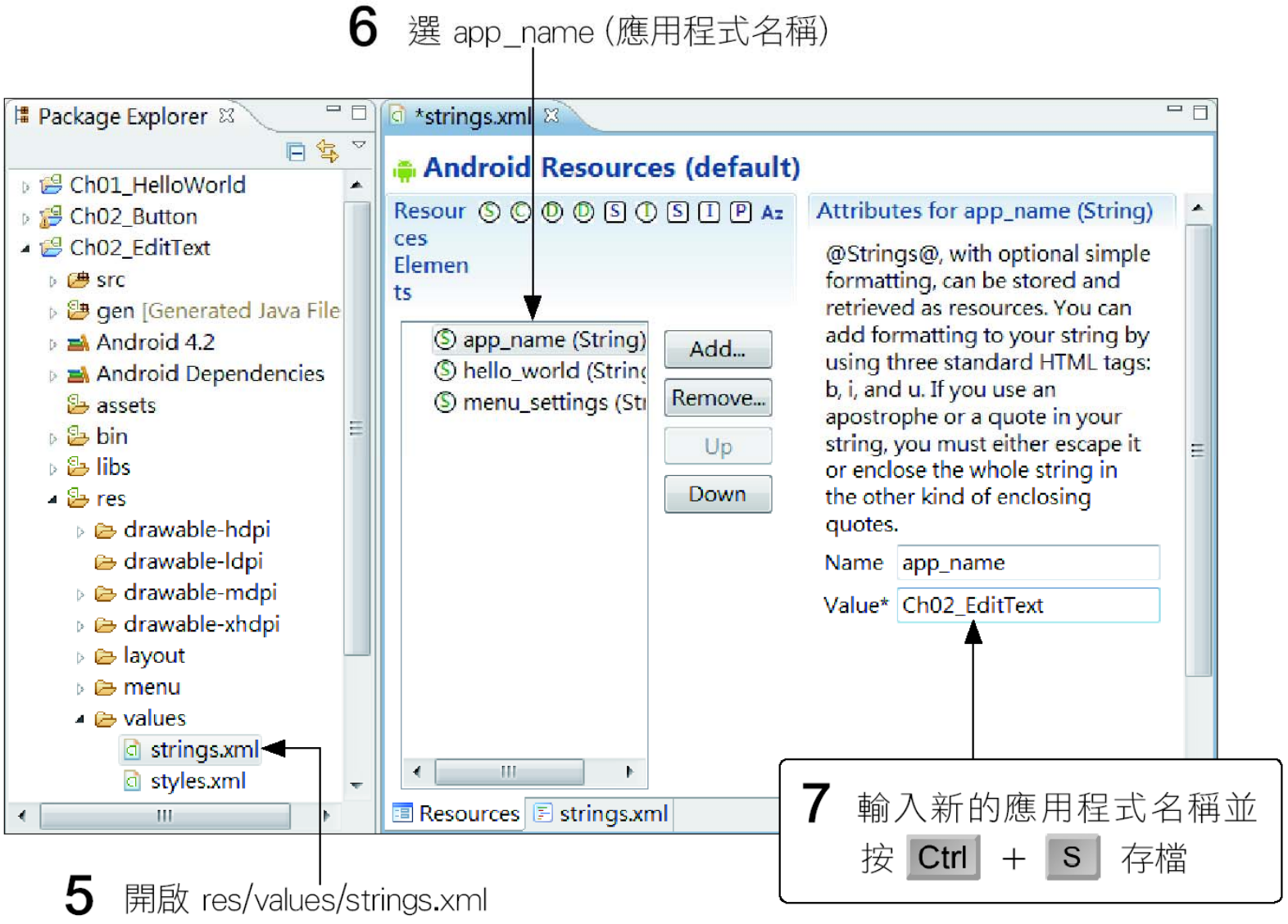
加入 EditText 元件



加入 EditText 元件



加入 EditText 元件

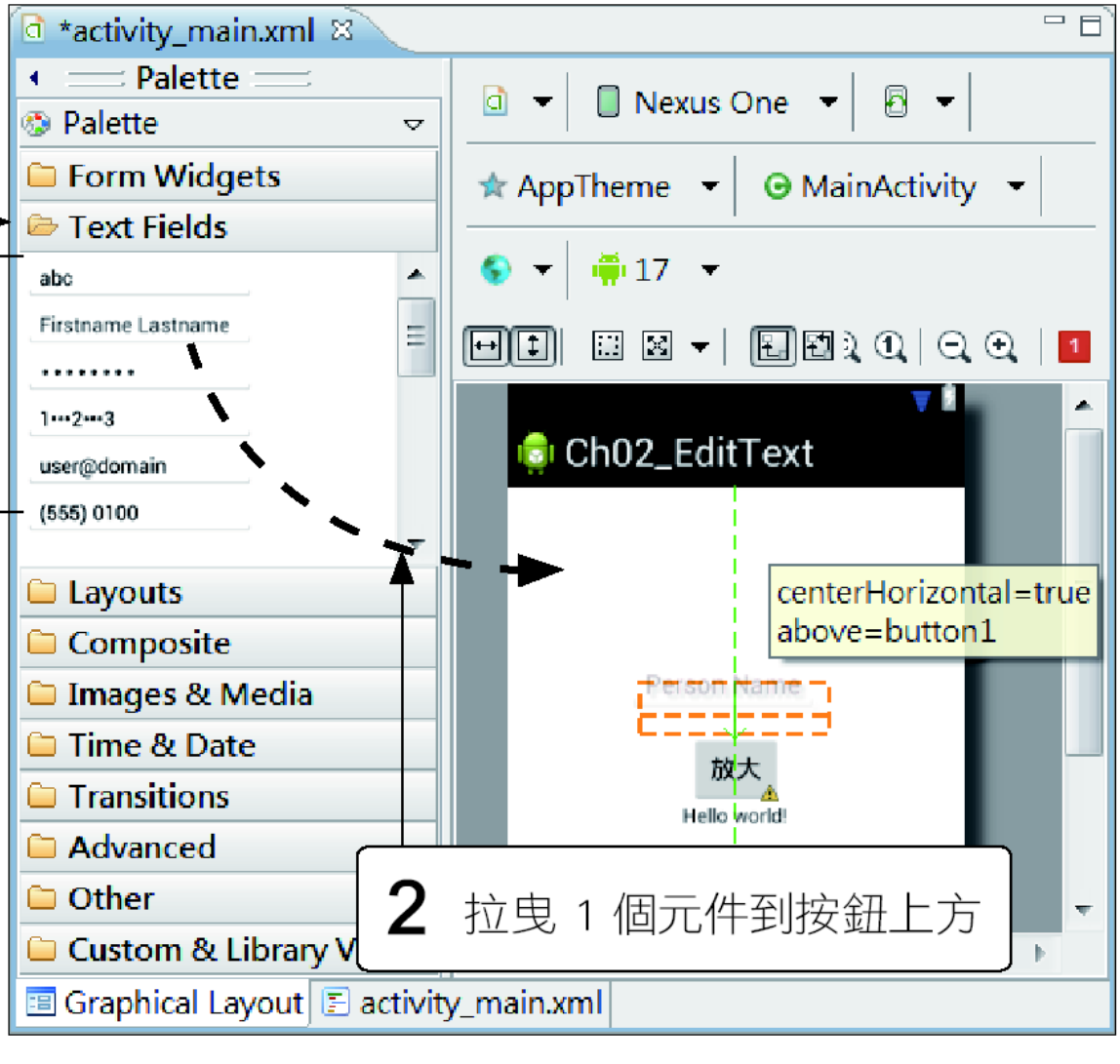


加入 EditText 元件

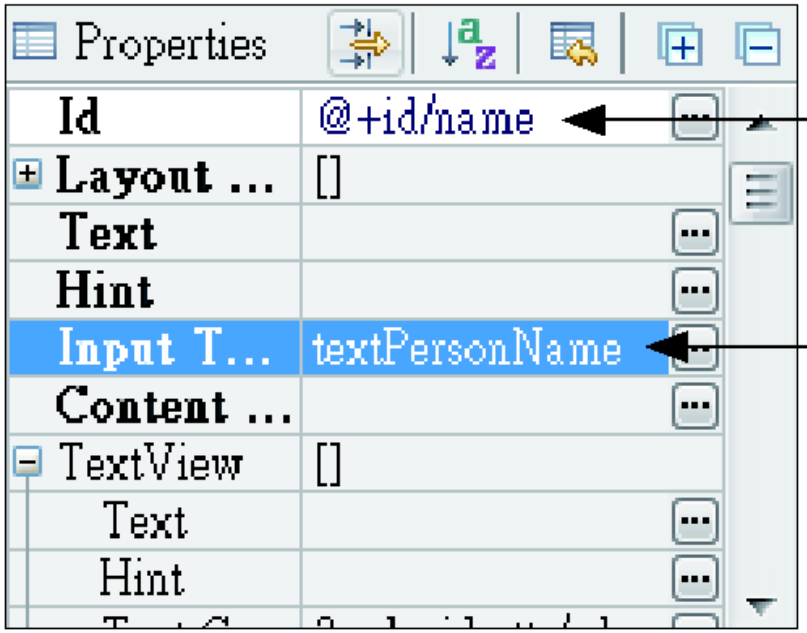
step 2

1 在 **Text Fields** 項目上按一下展開其內容

這些都是 EditText 元件, 但 Input Type 屬性不同 (參見下圖)



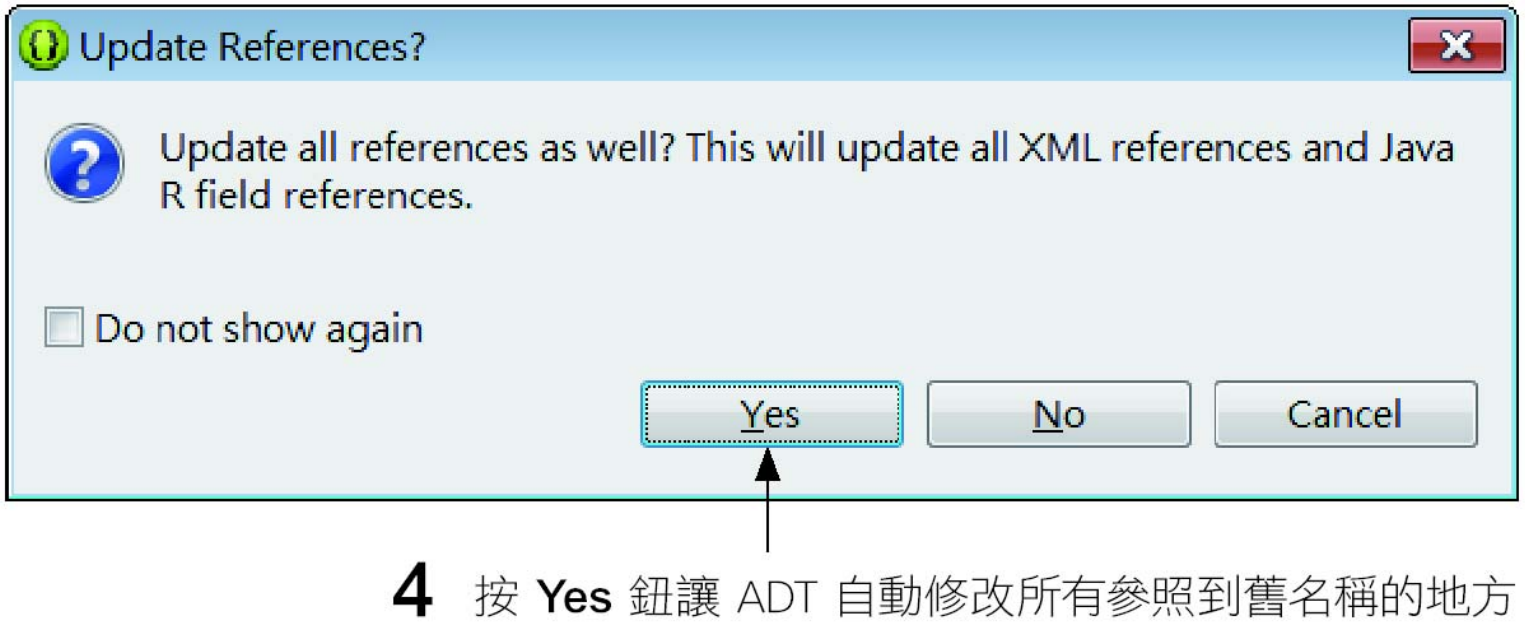
加入 EditText 元件



3 修改 Id 為 name

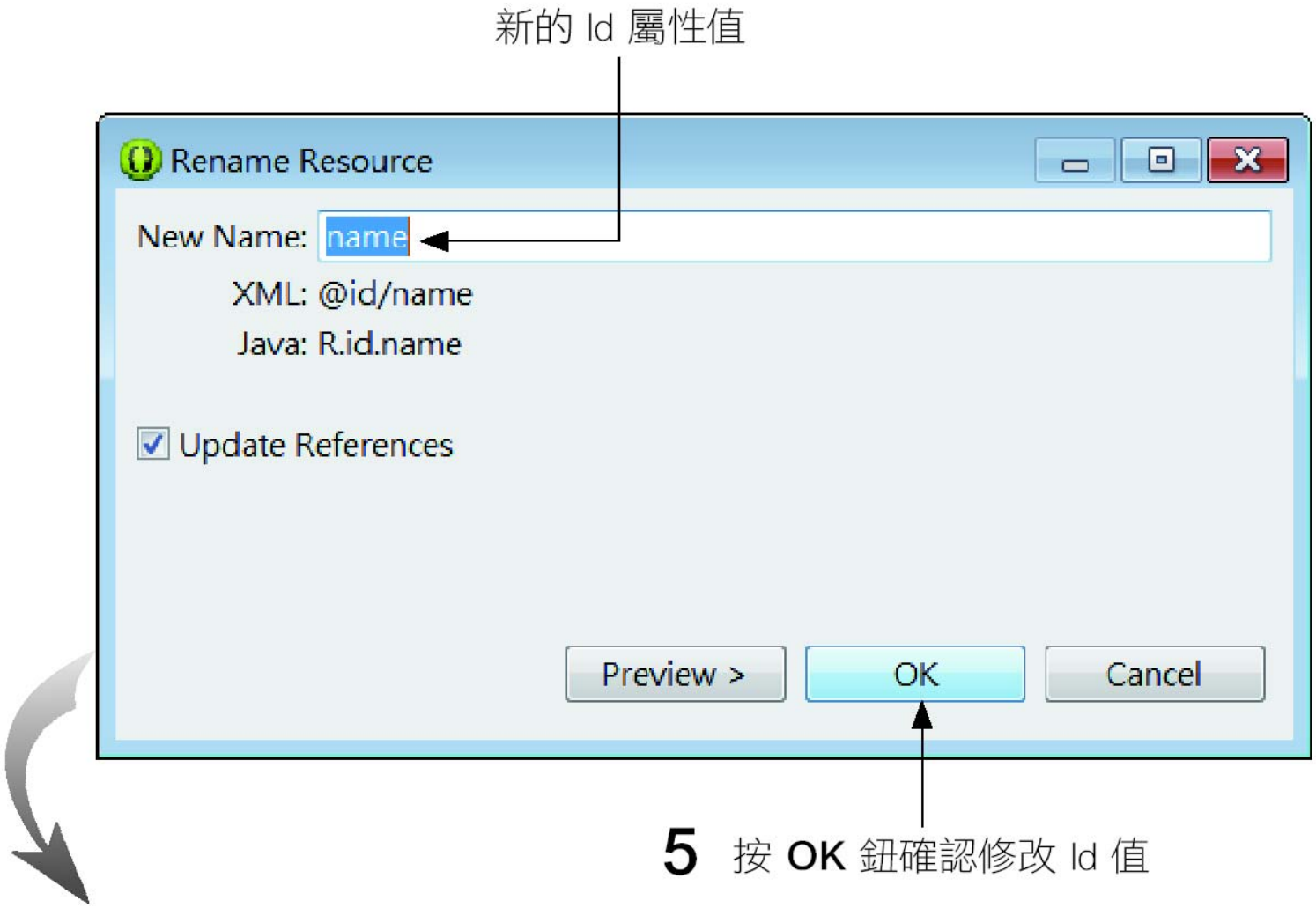
剛剛拉曳加入的項目，其 Input Type 屬性預設為 "textPersonName"，表示是用於輸入『人名』的欄位

加入 EditText 元件

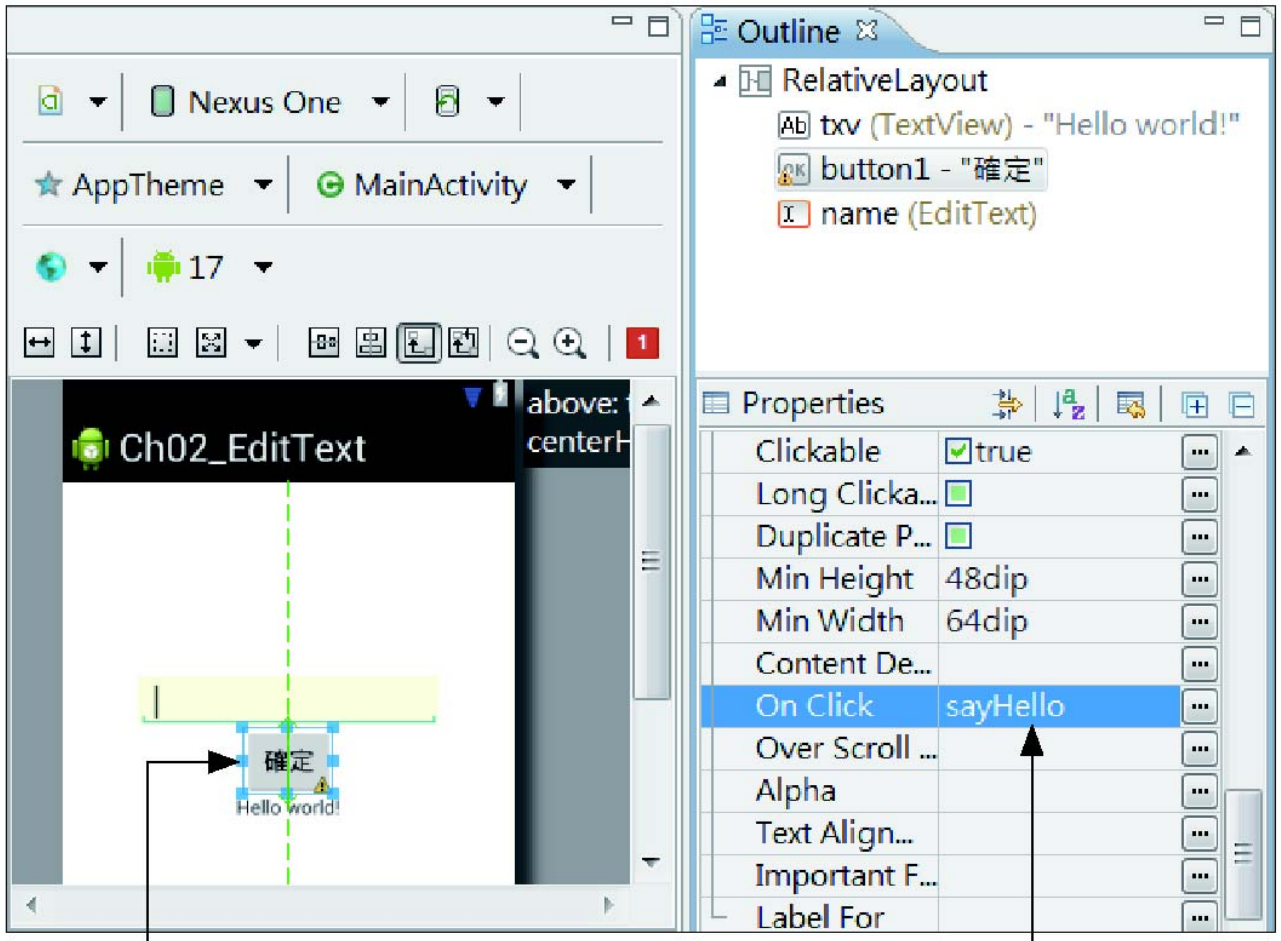


4 按 **Yes** 鈕讓 ADT 自動修改所有參照到舊名稱的地方

加入 EditText 元件



加入 EditText 元件



6 修改按鈕文字

7 設定按鈕的 On Click 屬性為 "sayHello"

加入 EditText 元件

step 3

自訂的 sayHello() 方法

```
25      public void sayHello(View v){
26          // 取得代表佈局中 EditText 及 TextView 的物件
27          EditText name = (EditText)findViewById(R.id.name);
28          TextView txv = (TextView)findViewById(R.id.txv);
29          // 設定 TextView 文字大小為 30sp,
30          // 並將 EditText 文字串接自訂訊息再設定給 TextView
31          txv.setTextSize(30);
32          txv.setText(name.getText().toString() + ", 您好！");
33      }
34  }
```

在輸入程式的過程中，可按 Ctrl + Shift + O 加入必要的 import 敘述匯入所需的類別

2-7 使用USB 線將程式部署到手機上執行

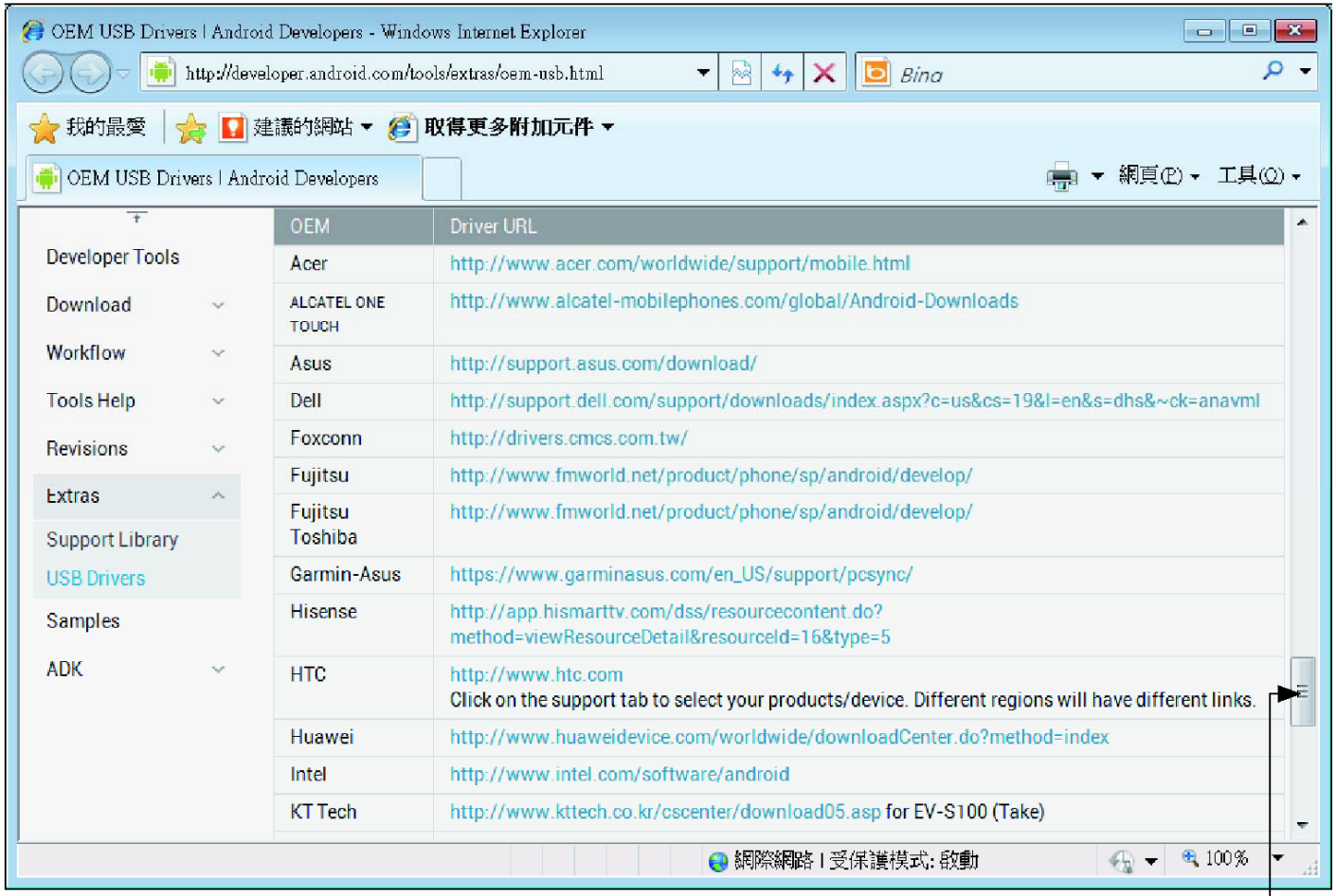
- 開啟手機偵錯功能
- 透過 USB 將 Android App 傳送到手機安裝與執行
- 執行已安裝的程式
- 修改專案的套件名稱

開啟手機偵錯功能

1 在設定項目清單中，選**開發人員選項**項目

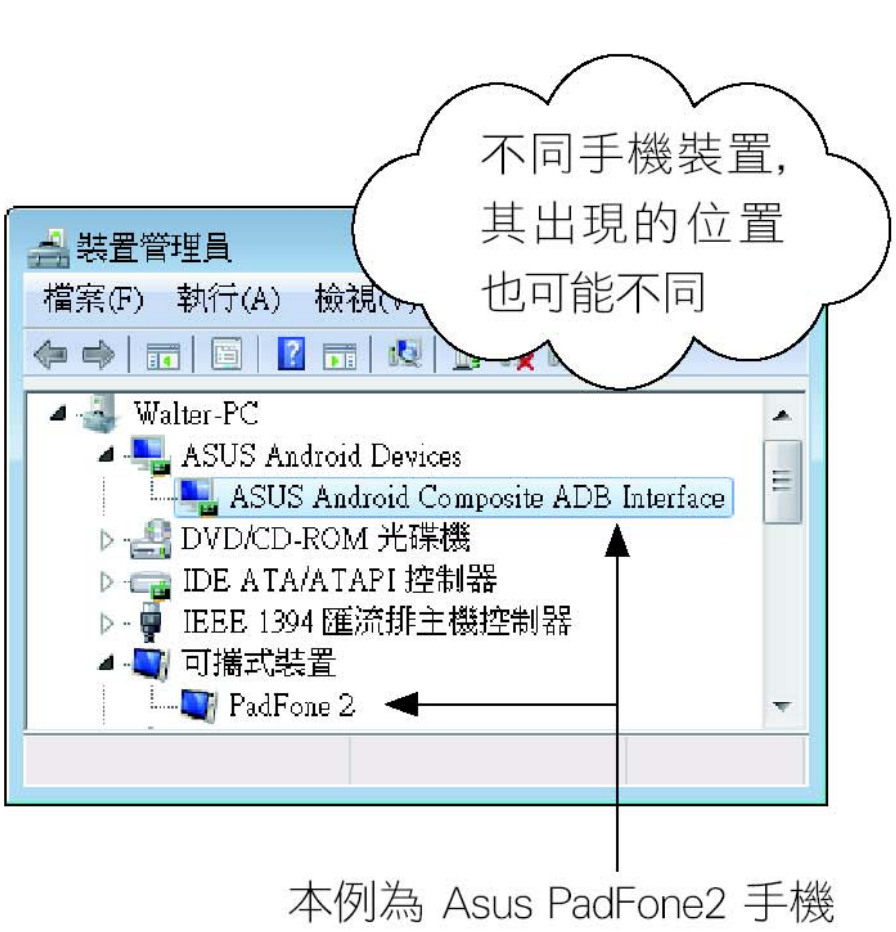
2 勾選 **USB 偵錯**並按**確定**鈕，啟用此設定表示允許 ADT 透過 USB 將程式傳送到手機

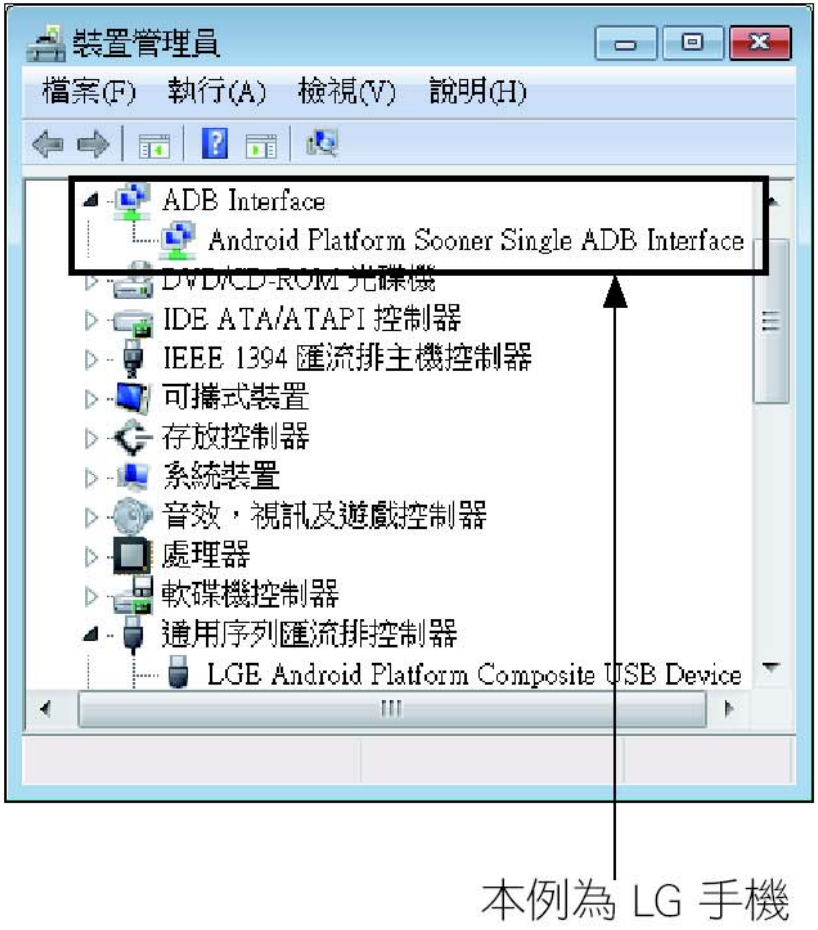
透過 USB 將 Android App 傳送到手機安裝與執行



將網頁向下捲動, 即可看到各廠商的下載網址連結

透過 USB 將 Android App 傳送到手機安裝與執行

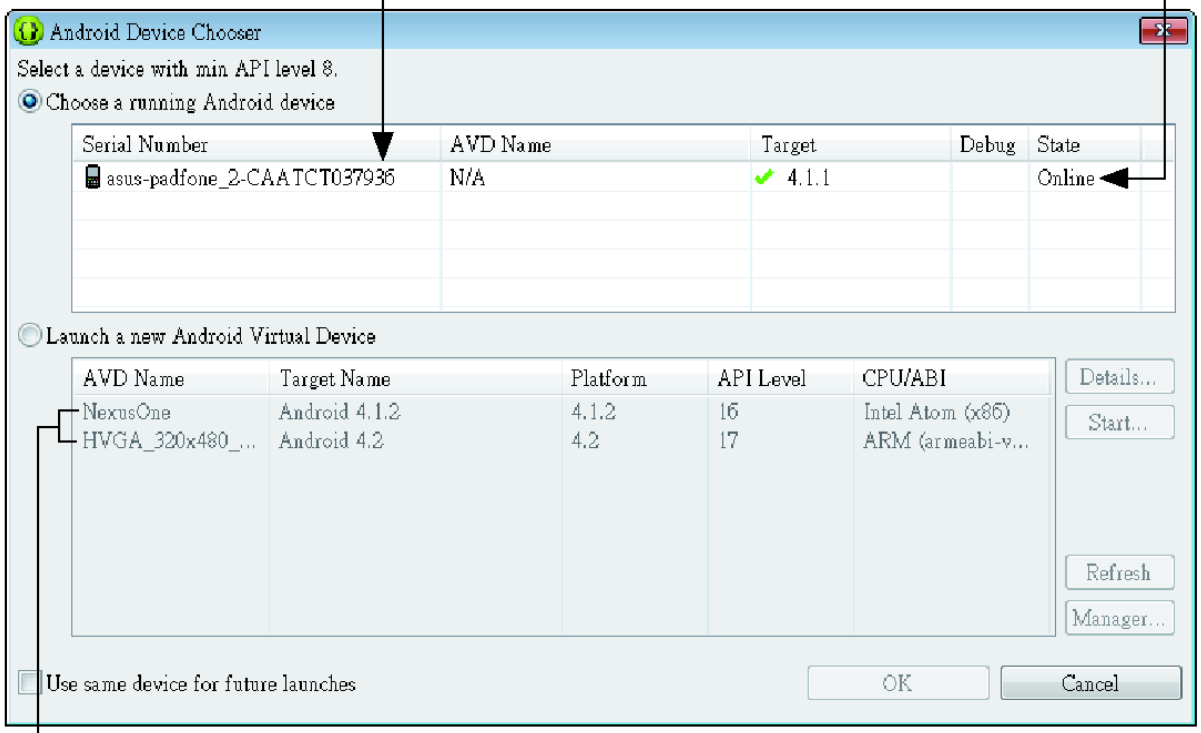




透過 USB 將 Android App 傳送到手機安裝與執行

雙按手機裝置名稱即可將程式安裝到手機上, 且會自動執行

若此處顯示 "Offline", 請將連接手機的 USB 線拔除, 再重新接回, 待顯示 "Online" 才能上傳程式



稍等一下, 就可在手機螢幕上看到程式的執行畫面了

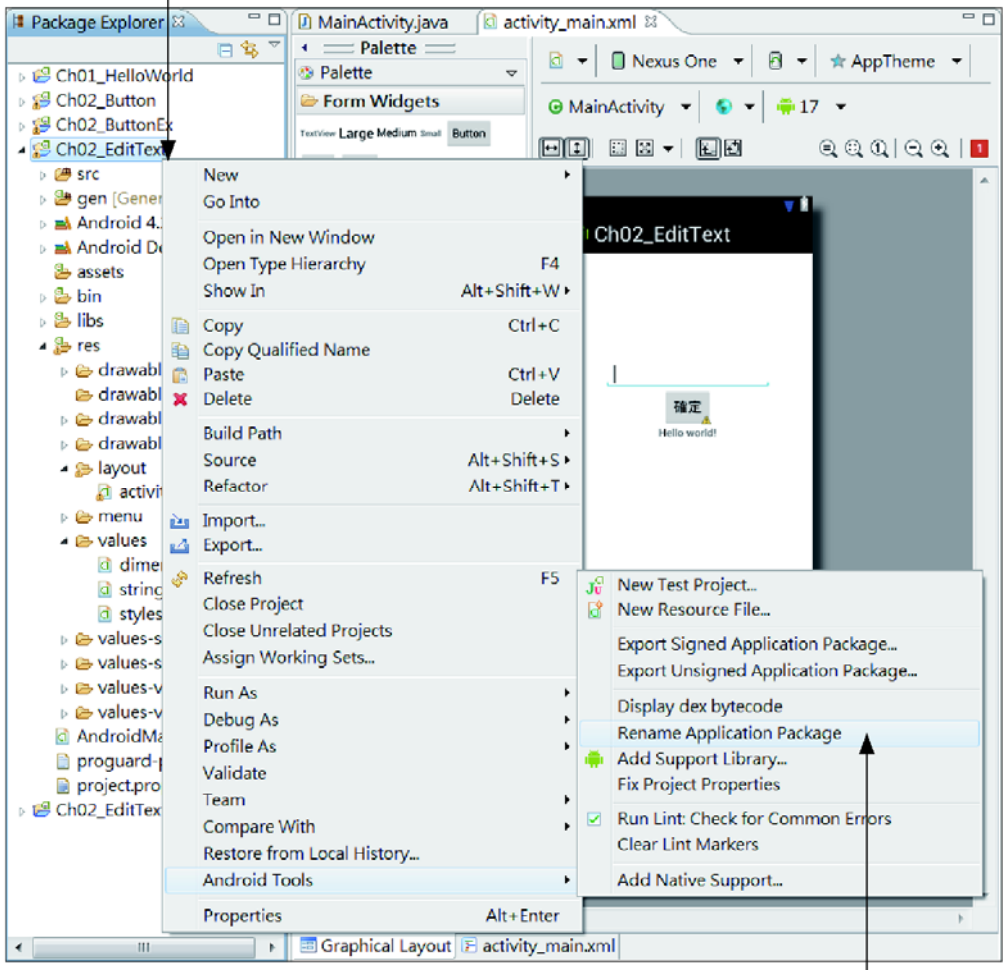
此為尚未啟動的模擬器, 若模擬器已啟動, 也會列在視窗上半的清單

執行已安裝的程式


5:19
應用程式
PAD專用
小工具
Ch02_EditText
Chrome
Compass & Leveler
Crocodile Roulette
Editor Of Music...
ES 任務管理器
ES 檔案瀏覽器
Evernote

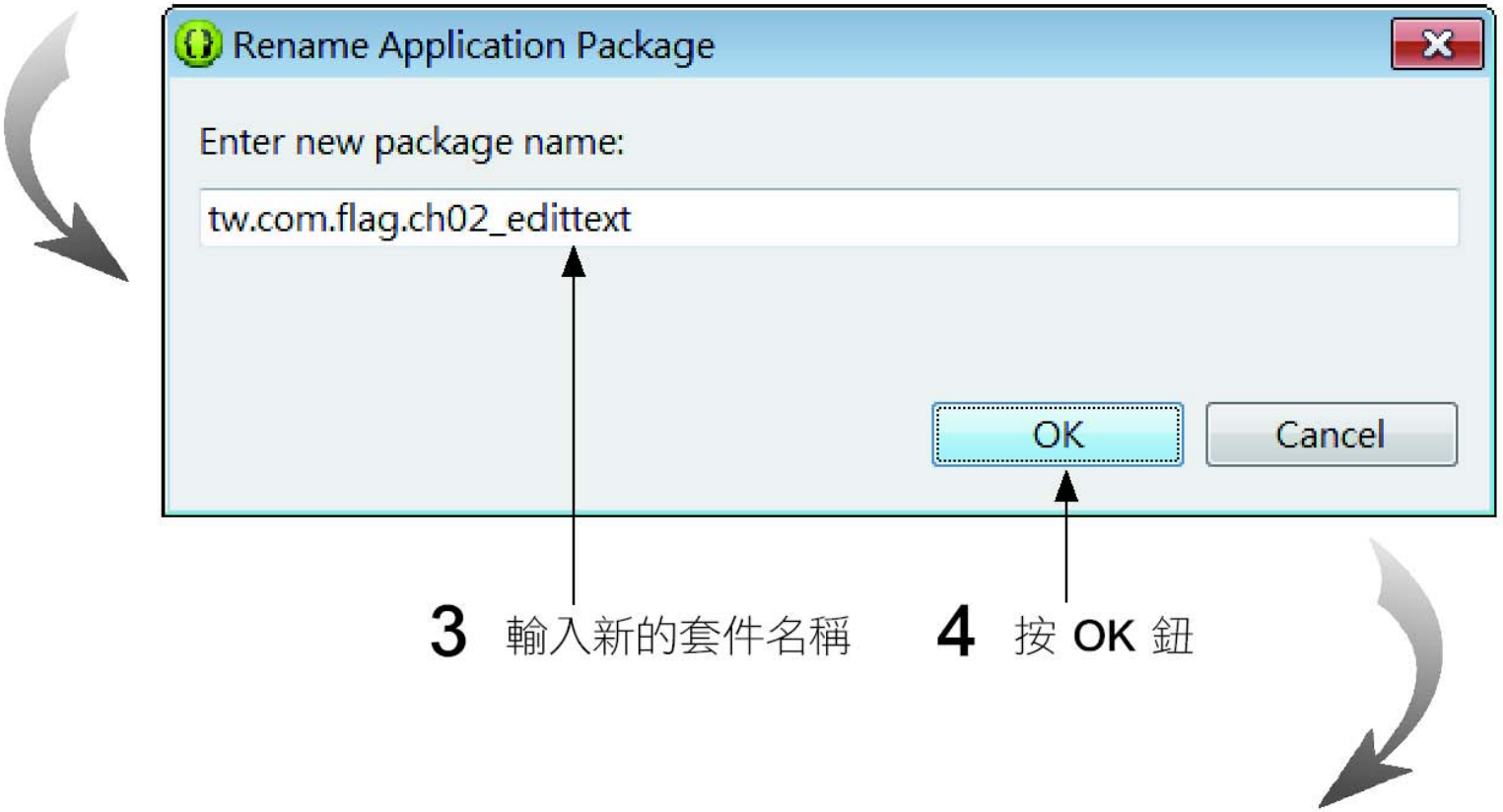
修改專案的套件名稱

1 在複製產生的新專案上按滑鼠右鈕

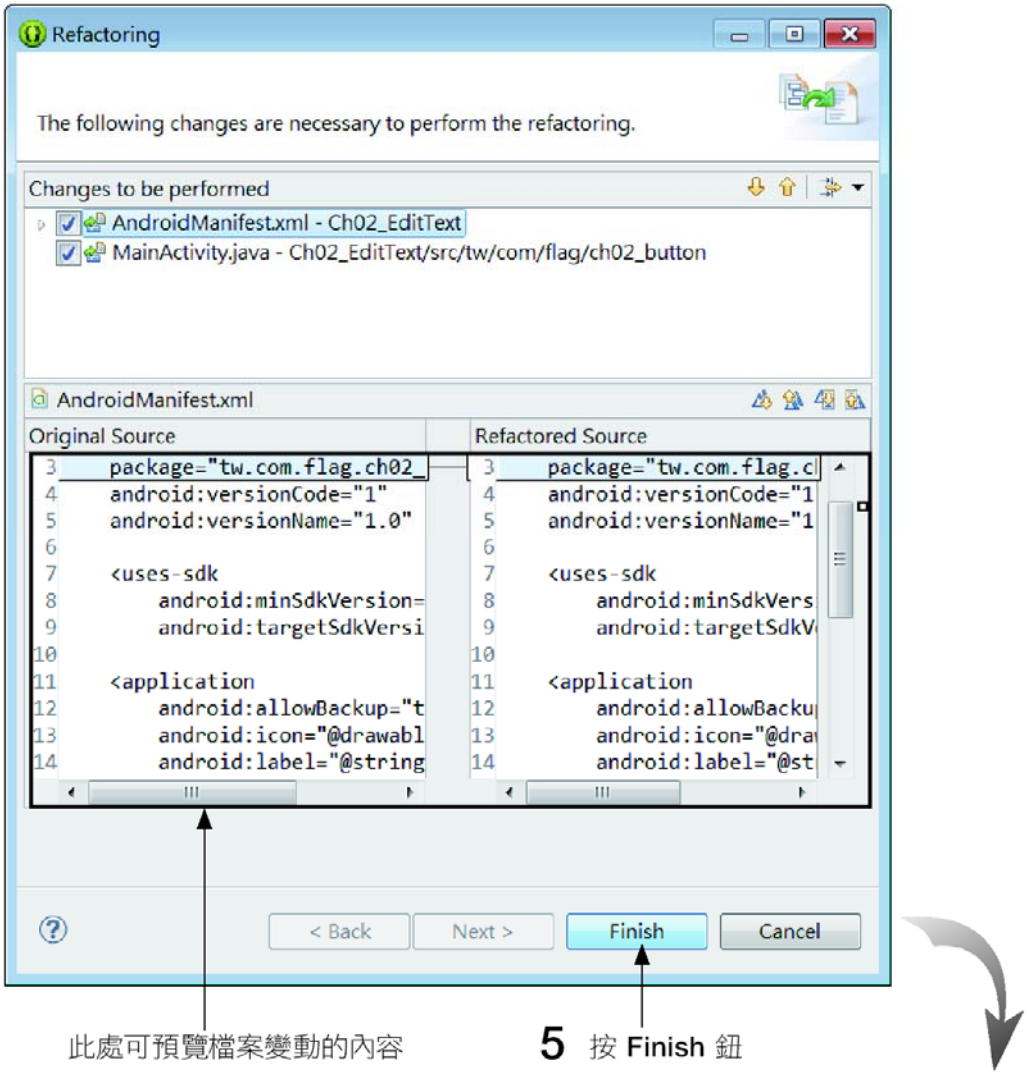


2 執行『Android Tools/Rename Application Package』命令

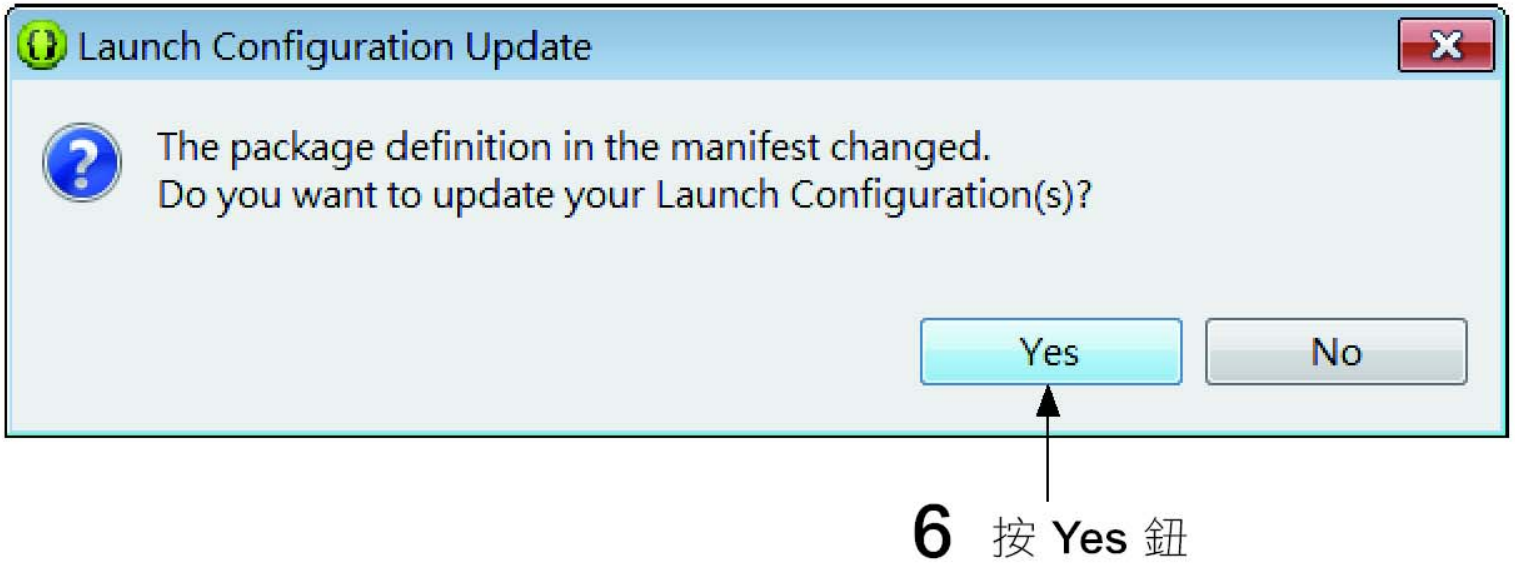
修改專案的套件名稱


Rename Application Package
Enter new package name:
tw.com.flag.ch02_edittext
OK
Cancel
3 輸入新的套件名稱
4 按 OK 鈕

修改專案的套件名稱

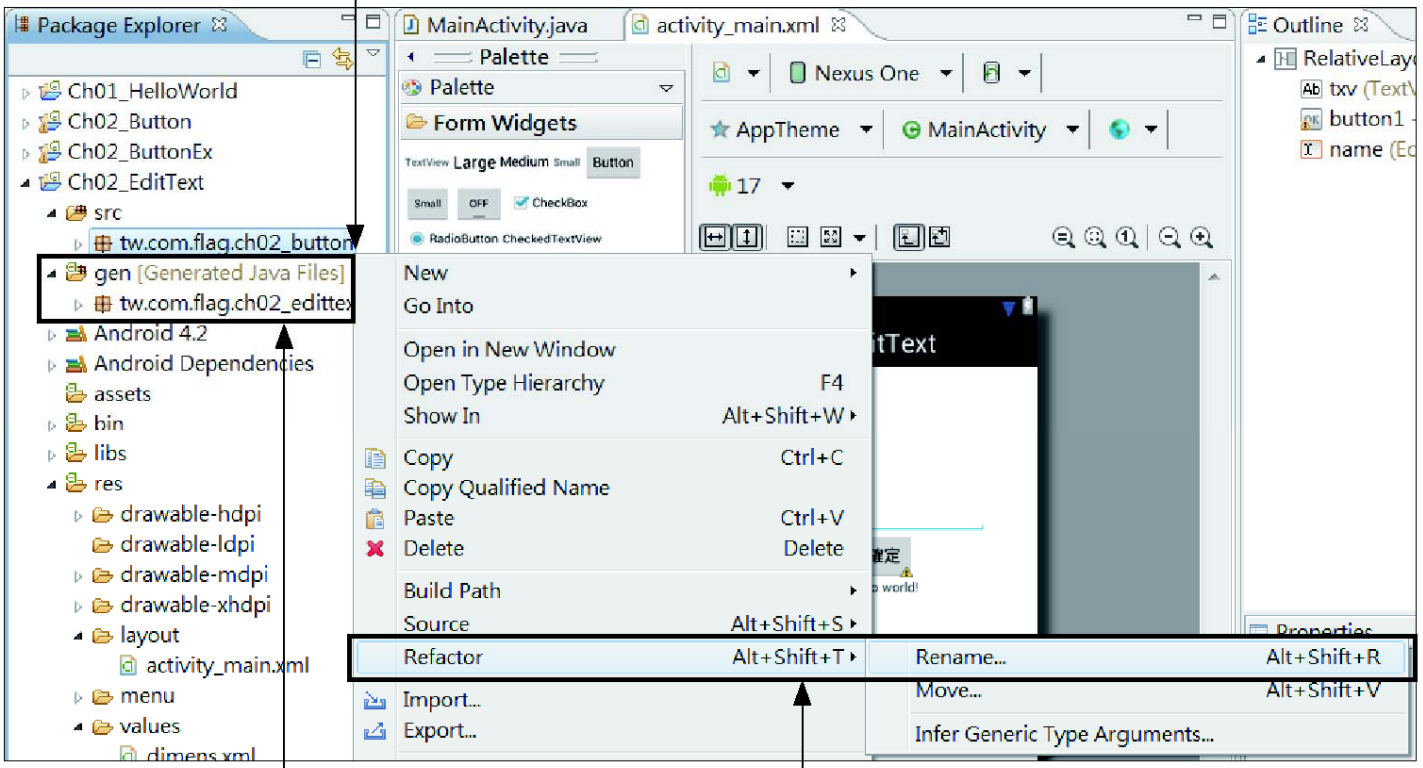

Refactoring
The following changes are necessary to perform the refactoring.
Changes to be performed
AndroidManifest.xml - Ch02_EditText
MainActivity.java - Ch02_EditText/src/tw/com/flag/ch02_button
AndroidManifest.xml
Original Source
Refactored Source
< Back
Next >
Finish
Cancel
此處可預覽檔案變動的內容
5 按 Finish 鈕

修改專案的套件名稱


Launch Configuration Update
The package definition in the manifest changed.
Do you want to update your Launch Configuration(s)?
Yes
No
6 按 Yes 鈕

修改專案的套件名稱

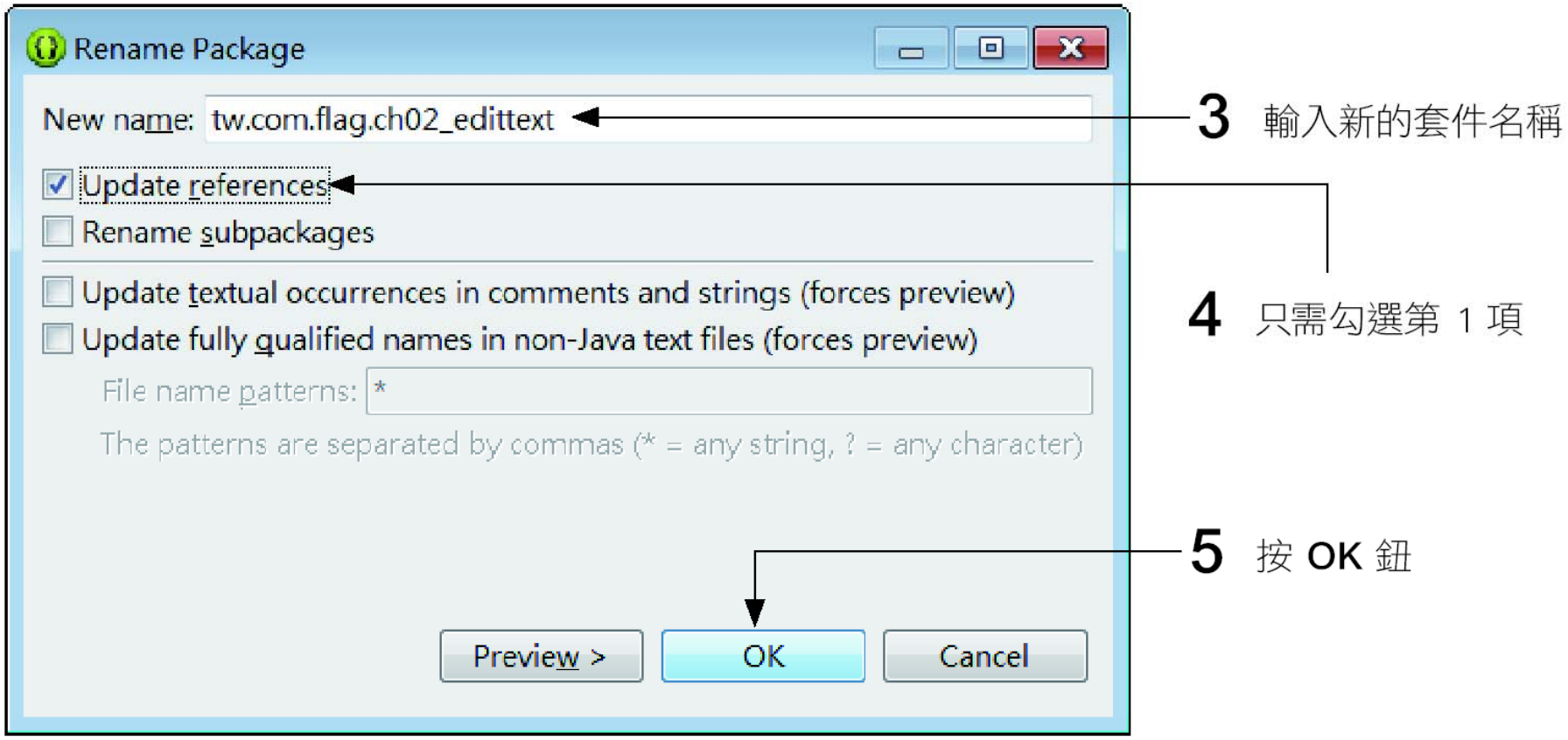
1 展開 src 資料夾, 在套件名稱上按滑鼠右鈕



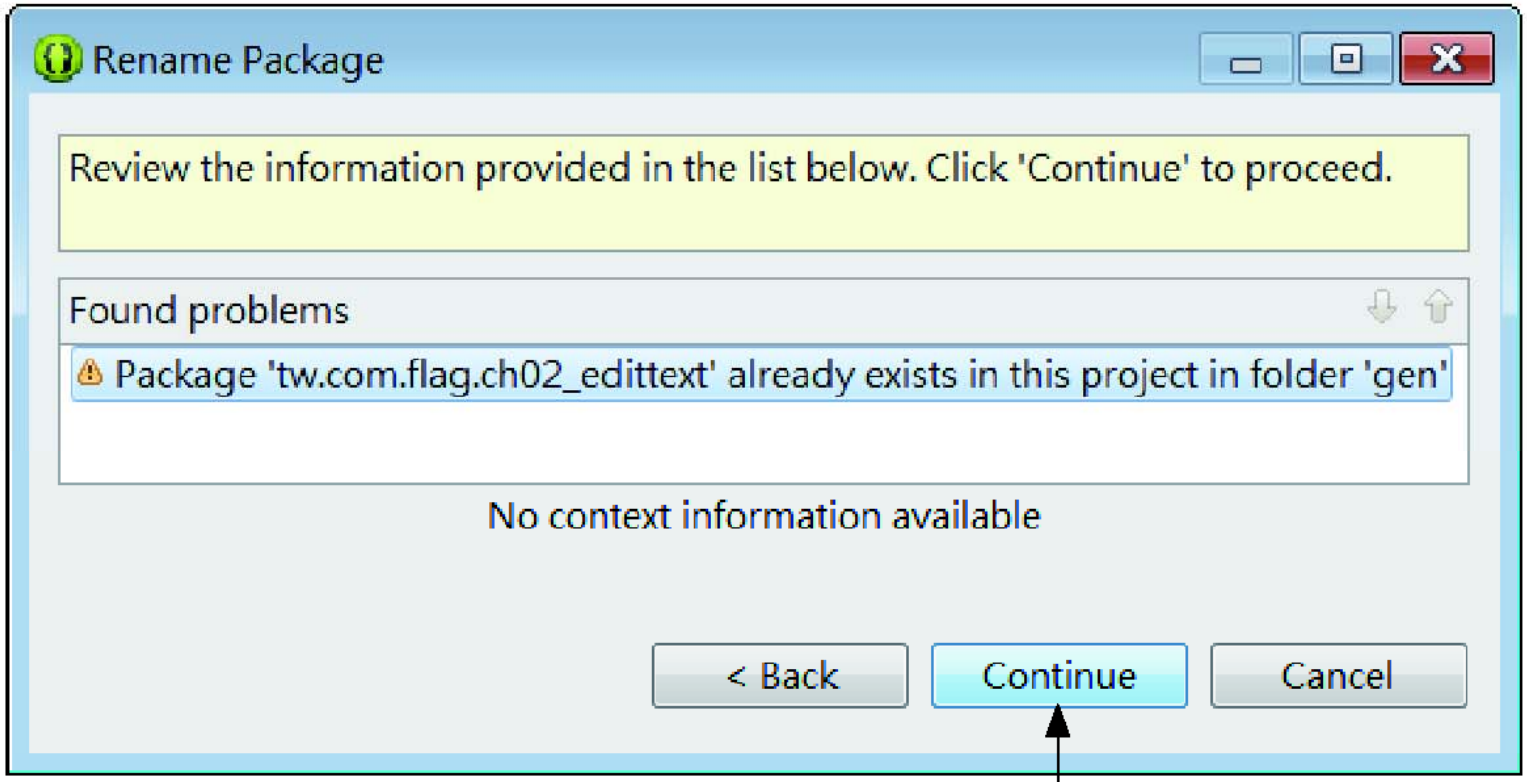
gen 資料夾是由 ADT 維護,
所以其內的套件名稱已更新

2 執行『**Refactor/Rename**』命令

修改專案的套件名稱

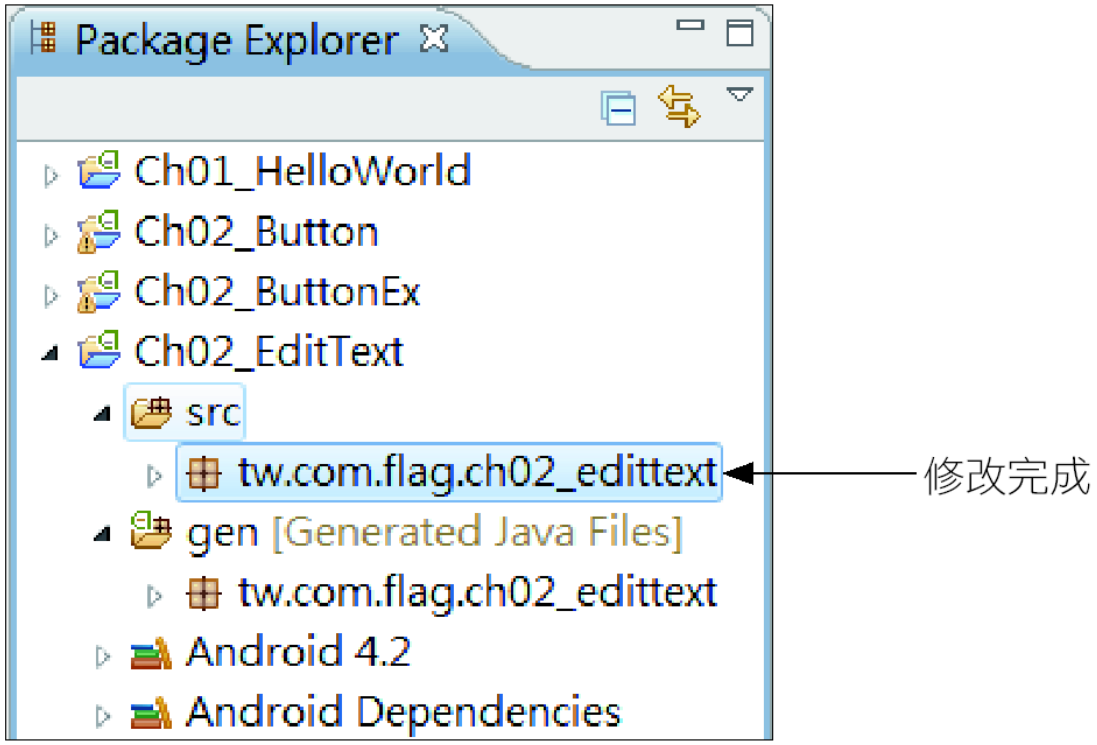

Rename Package
New name: tw.com.flag.ch02_edittext
Update references
Rename subpackages
Update textual occurrences in comments and strings (forces preview)
Update fully qualified names in non-Java text files (forces preview)
File name patterns: *
The patterns are separated by commas (* = any string, ? = any character)
Preview >
OK
Cancel
3 輸入新的套件名稱
4 只需勾選第 1 項
5 按 OK 鈕

修改專案的套件名稱



6 按 **Continue** 鈕確認修改套件名稱

修改專案的套件名稱


Package Explorer
Ch01_HelloWorld
Ch02_Button
Ch02_ButtonEx
Ch02_EditText
src
tw.com.flag.ch02_edittext
gen [Generated Java Files]
tw.com.flag.ch02_edittext
Android 4.2
Android Dependencies
修改完成