

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 3 章

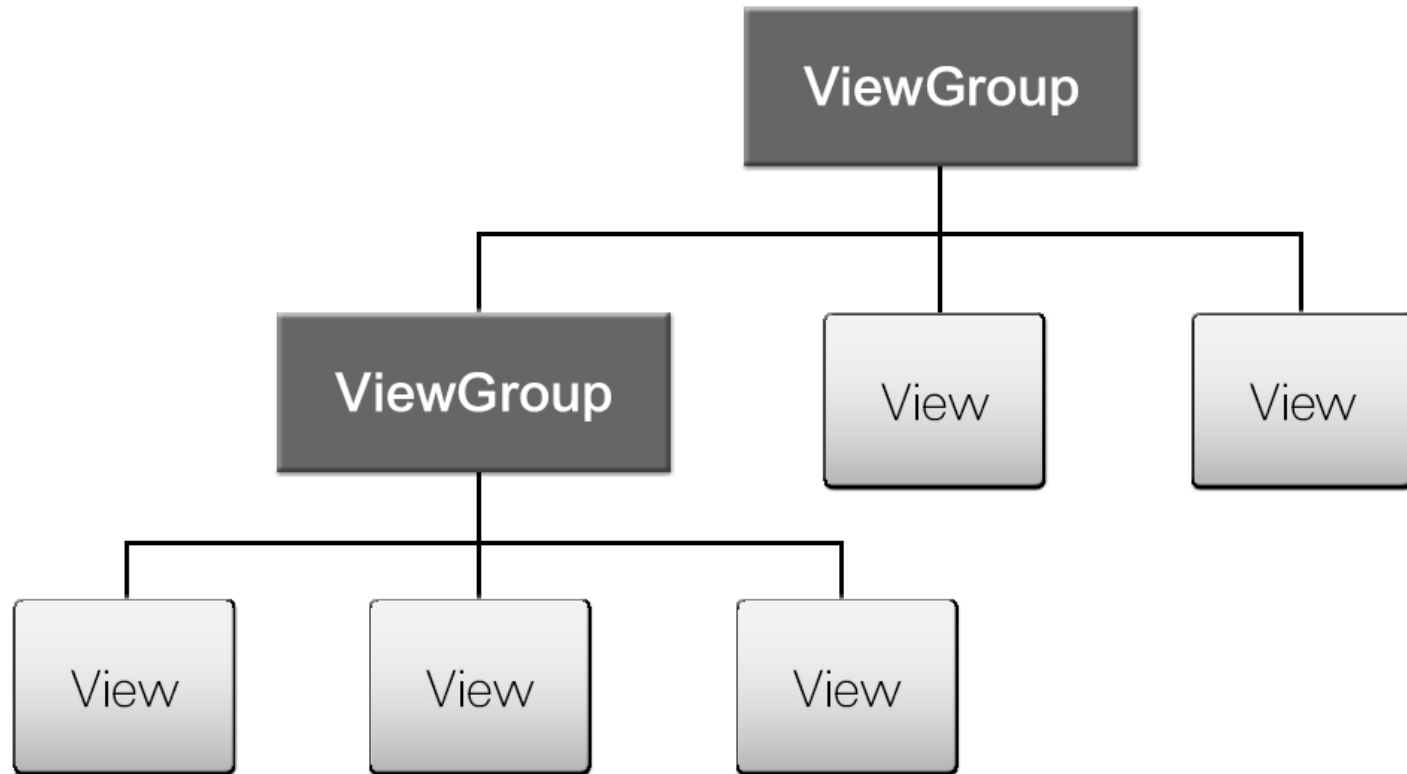
Android App 介面設計

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。

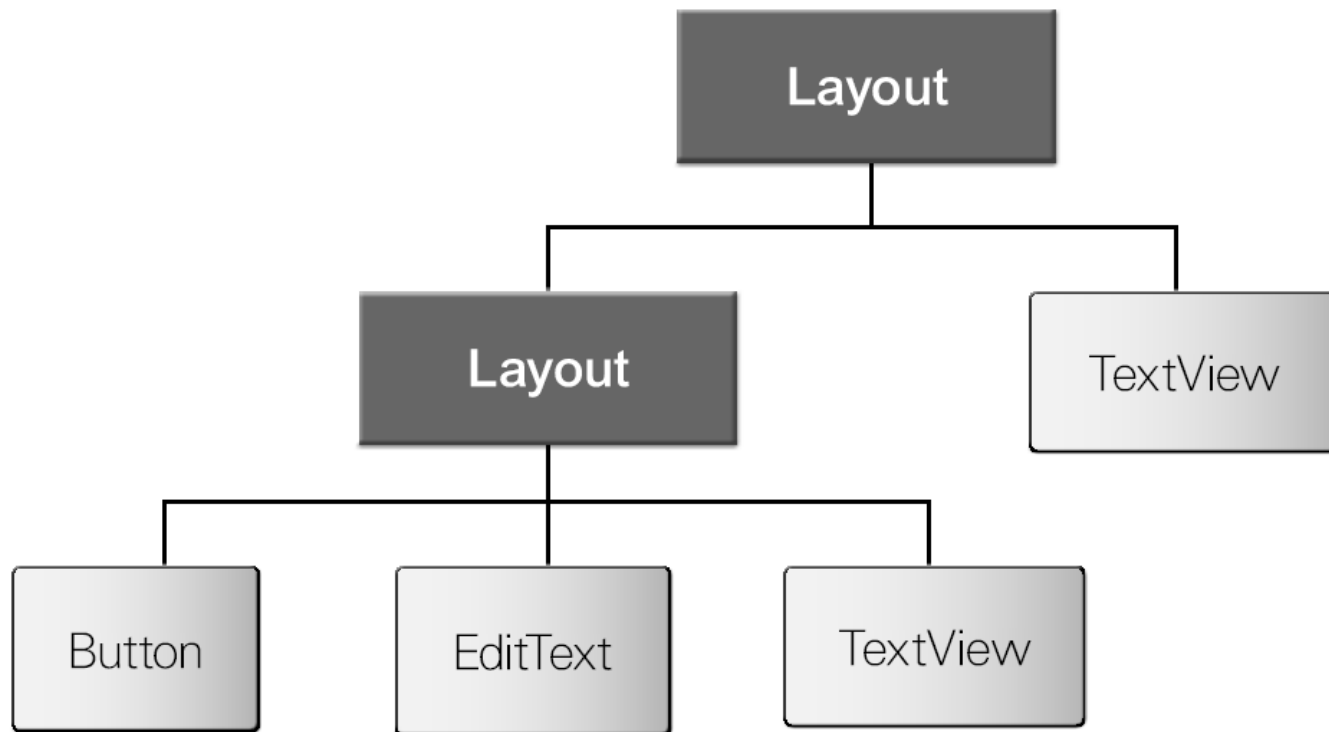
前言

- 3-1 View 與 ViewGroup(Layout)：元件與佈局
- 3-2 使用 LinearLayout 建立畫面佈局
- 3-3 使用 Weight 屬性控制元件在 LinearLayout 中的寬/高
- 3-4 透過屬性美化外觀
- 3-5 以程式設定元件的外觀屬性
- 3-6 使用自訂佈局樣版取代 ADT 預設佈局樣版
- 3-7 使用 Gmail 將程式寄給朋友測試

3-1 View 與 ViewGroup (Layout) : 元件與佈局



View 與 ViewGroup (Layout) : 元件與佈局

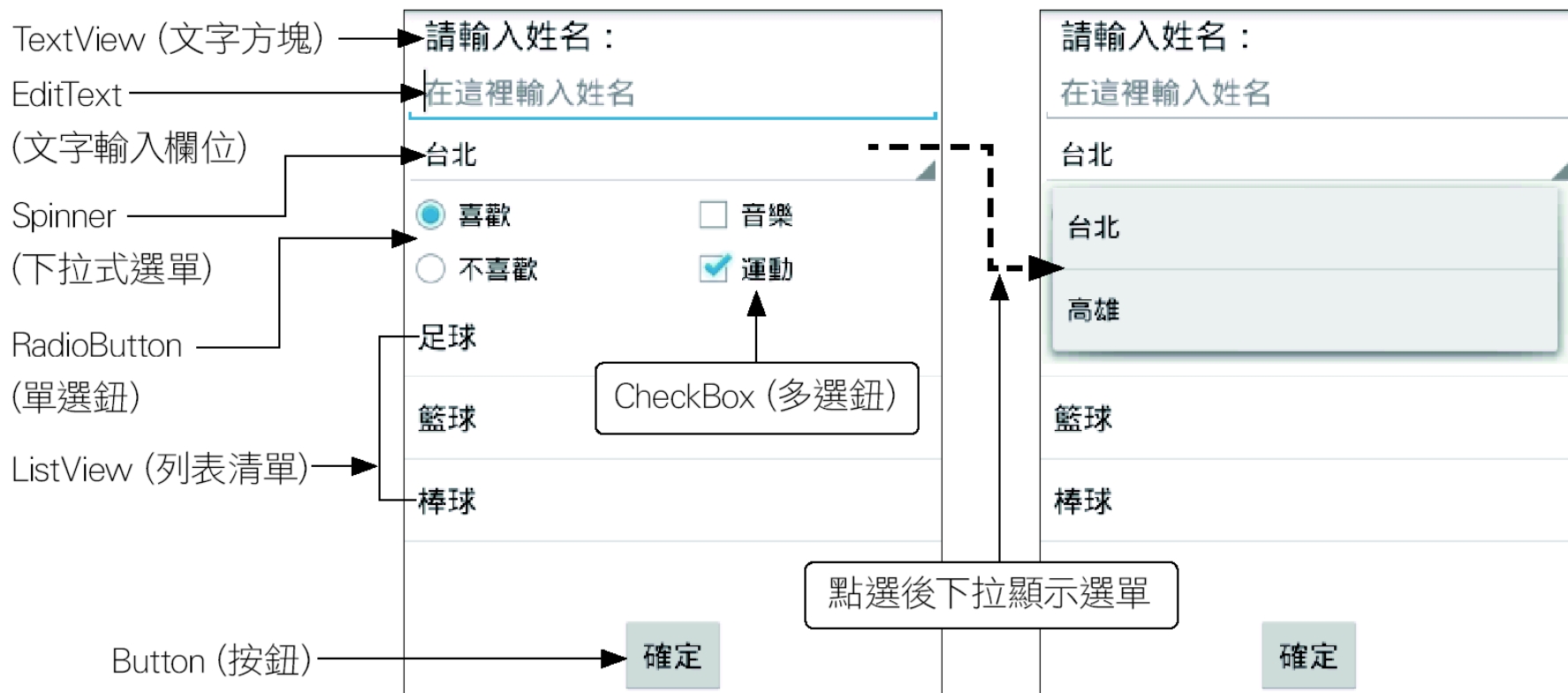


View 與 ViewGroup (Layout) :

元件與佈局

- View：視覺元件
- View 元件的屬性與設定
- 再談Id 屬性
- Layout：畫面佈局

View：視覺元件



View：視覺元件

Form Widgets
資料夾當中有
這些元件可使用

各類元件都分
門別類的放在
各資料夾當中

拉到編輯區

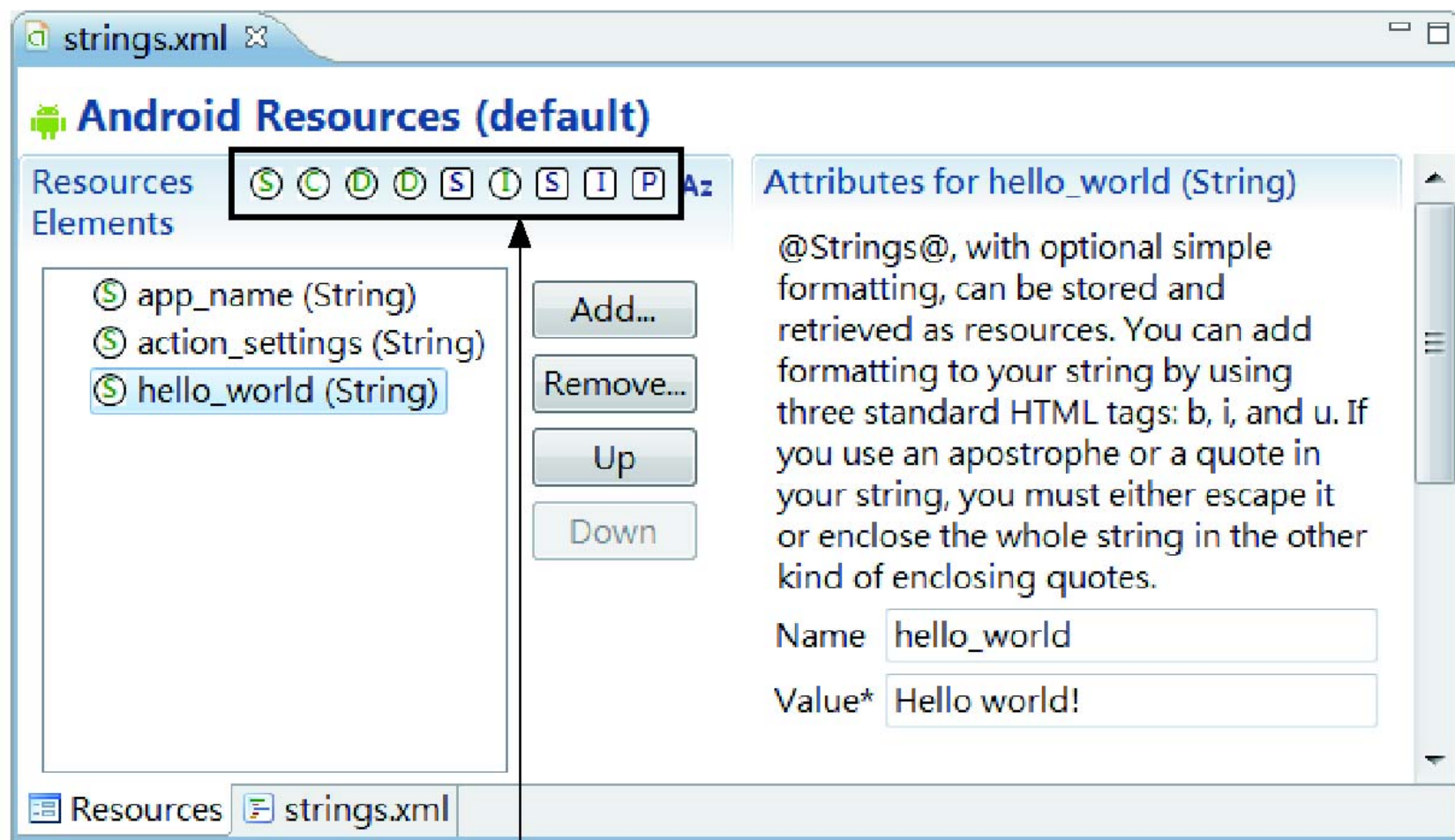
目前的 Layout 層次

Id	@+id/b...
Layout...	[]
Style	android:...
Text	確定
Hint	
Conte...	
TextVie...	[]
Text	確定
Hint	

*我怎麼看不到這個畫面？請雙按 Package Explorer 的 res/layout/activity_main.xml 檔就可以看到了

屬性 (Properties) 設定區

View 元件的屬性與設定



這些項目表示要顯示的資源種類, 開啟 strings.xml 時預設顯示字串資源

Layout：畫面佈局

- **RelativeLayout (相對佈局)**：是以元件與元件之間的相對關係來安排佈局，在上一章的範例都是使用這種佈局方式。
- **LinearLayout (線性佈局)**：是從上到下或從左到右依序擺放元件的佈局方式。

Layout：畫面佈局



Layout：畫面佈局

- **TableLayout (表格佈局)**：是以表格方式將個別元件放置在指定儲存格的佈局方式。

	第 1 行	第 2 行	第 3 行
第 1 列	元件 1	元件 2	元件 3
第 2 列	元件 4	元件 5	元件 6

Layout：畫面佈局

文字方塊

文字輸入欄位

按鈕

單選鈕

多選鈕

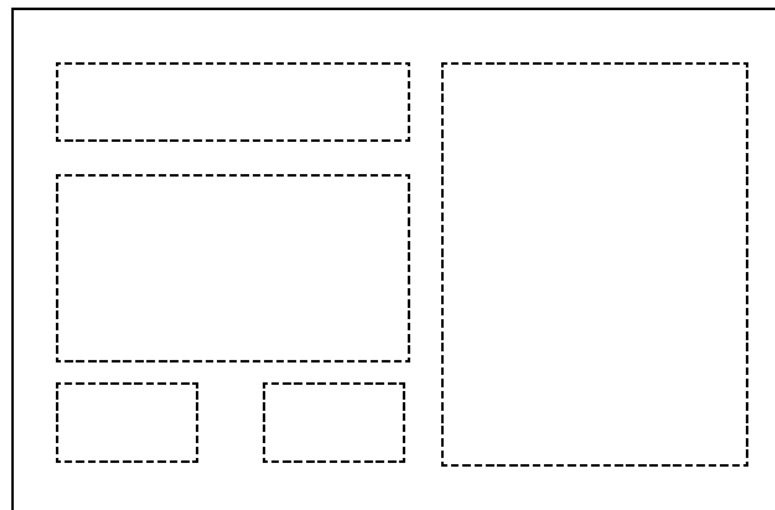
列表項目 1

列表項目 2

列表項目 3

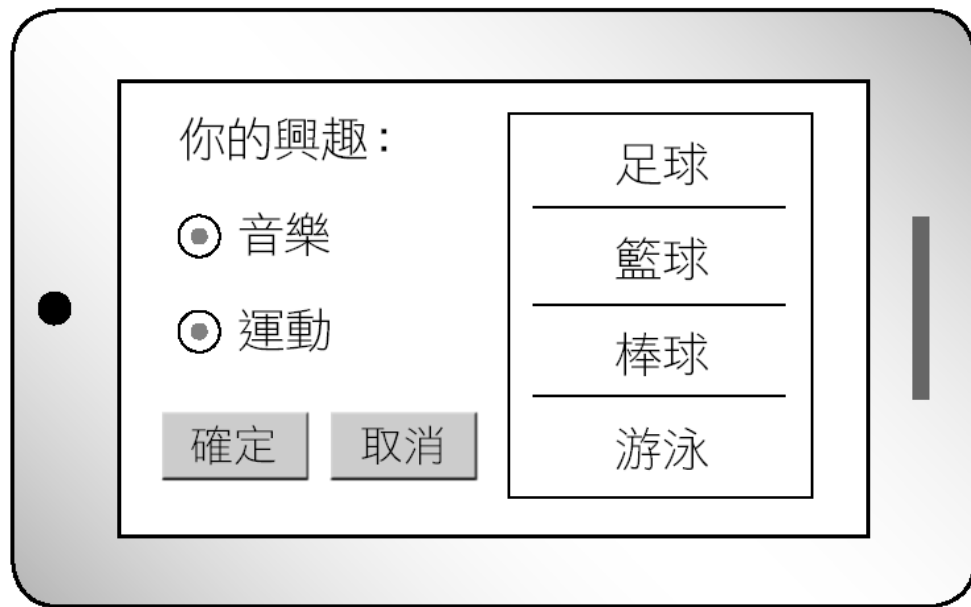
列表項目 4

在圖形編輯器選用、拉曳各種 View 元件, 並設定其屬性

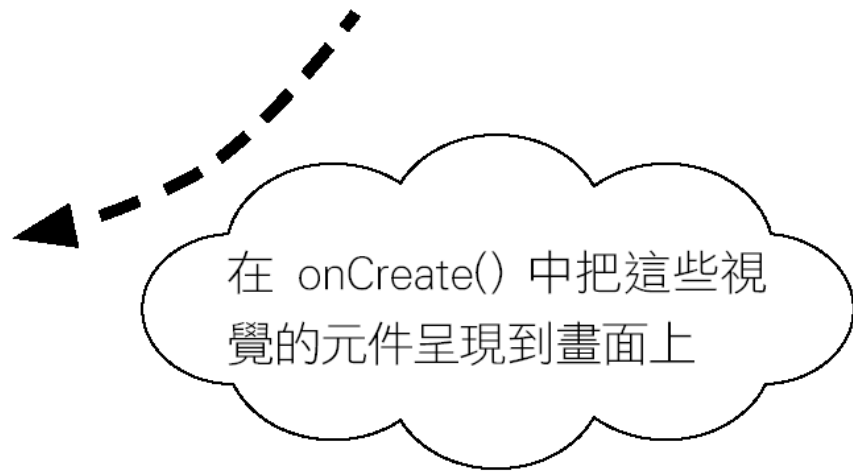


由 Layout 規範元件的排列原則

Layout：畫面佈局



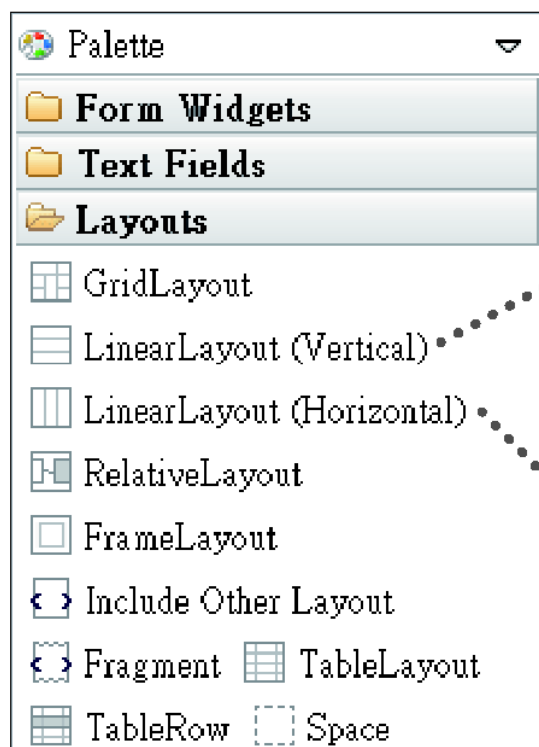
在手機畫面上呈現出來的外觀



3-2 使用 **LinearLayout** 建立畫面佈局

- **LinearLayout**：依序排列元件
 - ▣ **LinearLayout z (Horizontal)**：將其內的元件依水平方向，由左至右直線排列。
 - ▣ **LinearLayout z (Vertical)**：將其內的元件依垂直方向，由上而下直線排列。
- **Input Type** 屬性：設定輸入欄位種類

LinearLayout：依序排列元件



LinearLayout (Vertical)
內的元件是由上至下
垂直排列



LinearLayout (Horizontal)
內的元件是由左至右水
平排列

LinearLayout：依序排列元件

最外層 (最上層) 是 LinearLayout (Vertical)

3 個 LinearLayout (Horizontal)

TextView 元件 EditText 元件

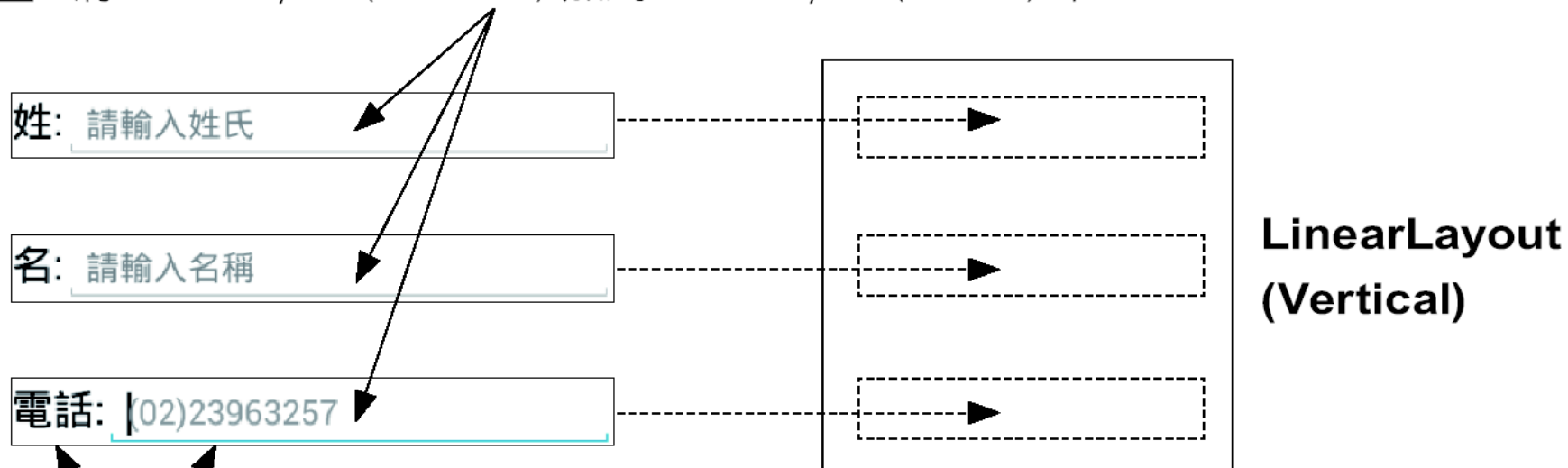
下層放 3 個水平的 LinearLayout

Outline

- LinearLayout1
 - LinearLayout
 - TextView01 - "姓"
 - EditText01
 - LinearLayout
 - TextView02 - "名"
 - EditText02
 - LinearLayout
 - textView1 - "電話"
 - editText1

LinearLayout：依序排列元件

2 將 LinearLayout (Horizontal) 加到 LinearLayout (Vertical) 中



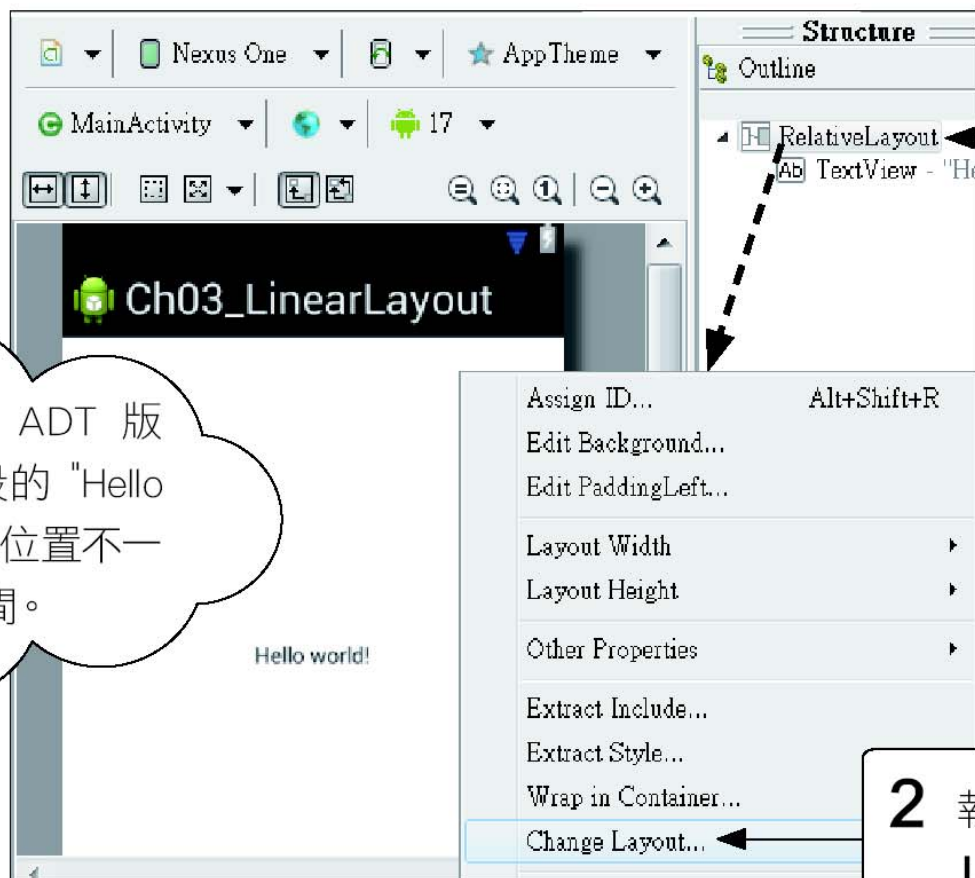
3 把元件加到水平的
LinearLayout (Horizontal) 中

1 Activity 的最上層佈局
為 LinearLayout (Vertical)

圖 3-1 使用垂直與水平的 LinearLayout 來製作輸入畫面

範例3-1：在佈局中使用LinearLayout

step 1

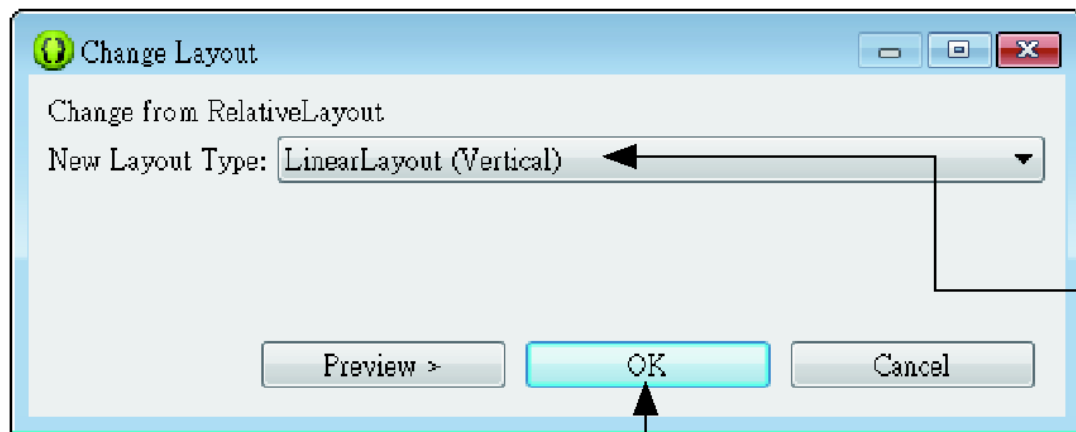


1 在 RelativeLayout
上按滑鼠右鈕

若使用不同的 ADT 版本，畫面中預設的 "Hello world!" 文字，其位置不一定會在畫面中間。

2 執行『Change
Layout』命令

在佈局中使用 **LinearLayout**



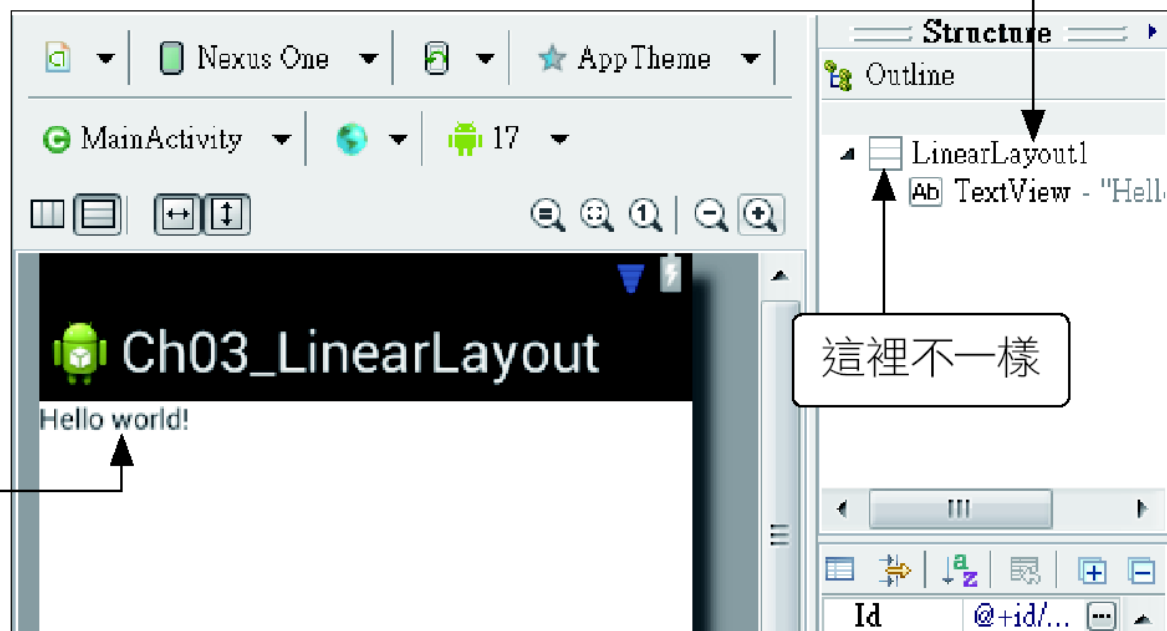
3 在下拉清單中選取
LinearLayout (Vertical)

4 按 OK 鈕

在佈局中使用 LinearLayout

最上層元件轉換成 LinearLayout (Vertical) 了

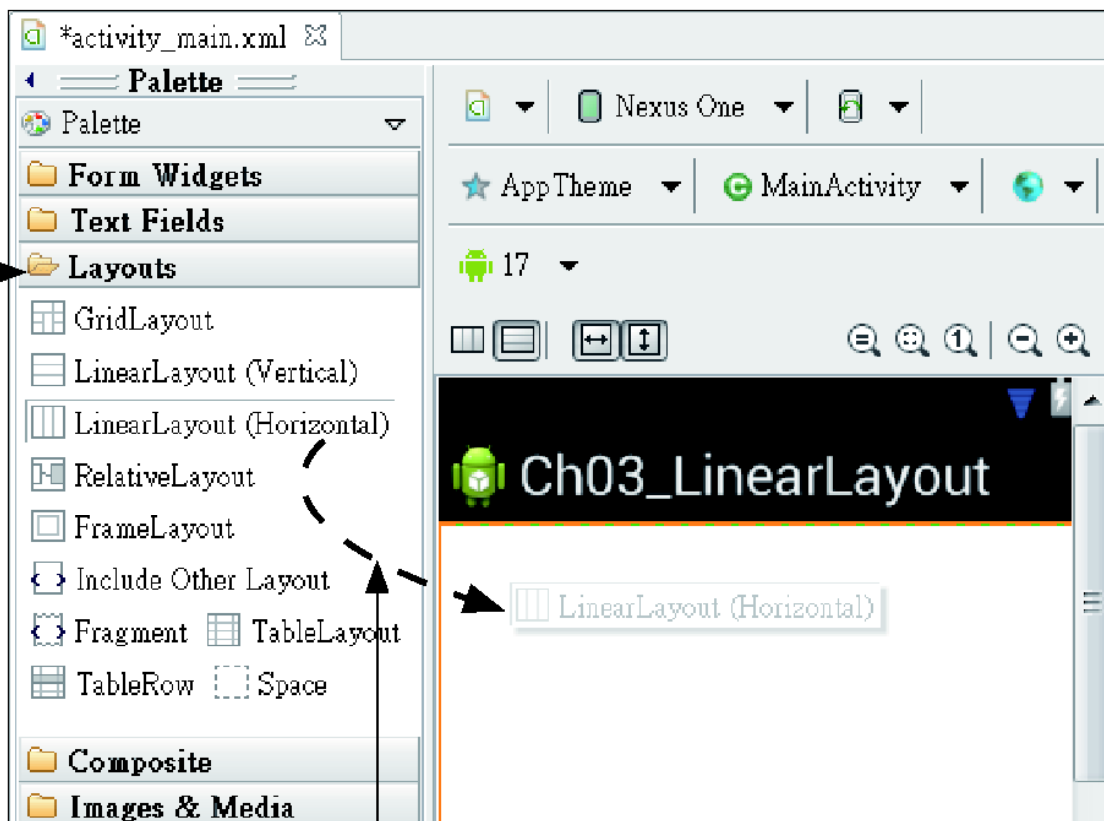
因為 LinearLayout (Vertical) 內的元件是由上而下排列，所以原本就存在的 TextView 元件會移到畫面最上方



範例3-2：使用 LinearLayout (Horizontal) 建立表單

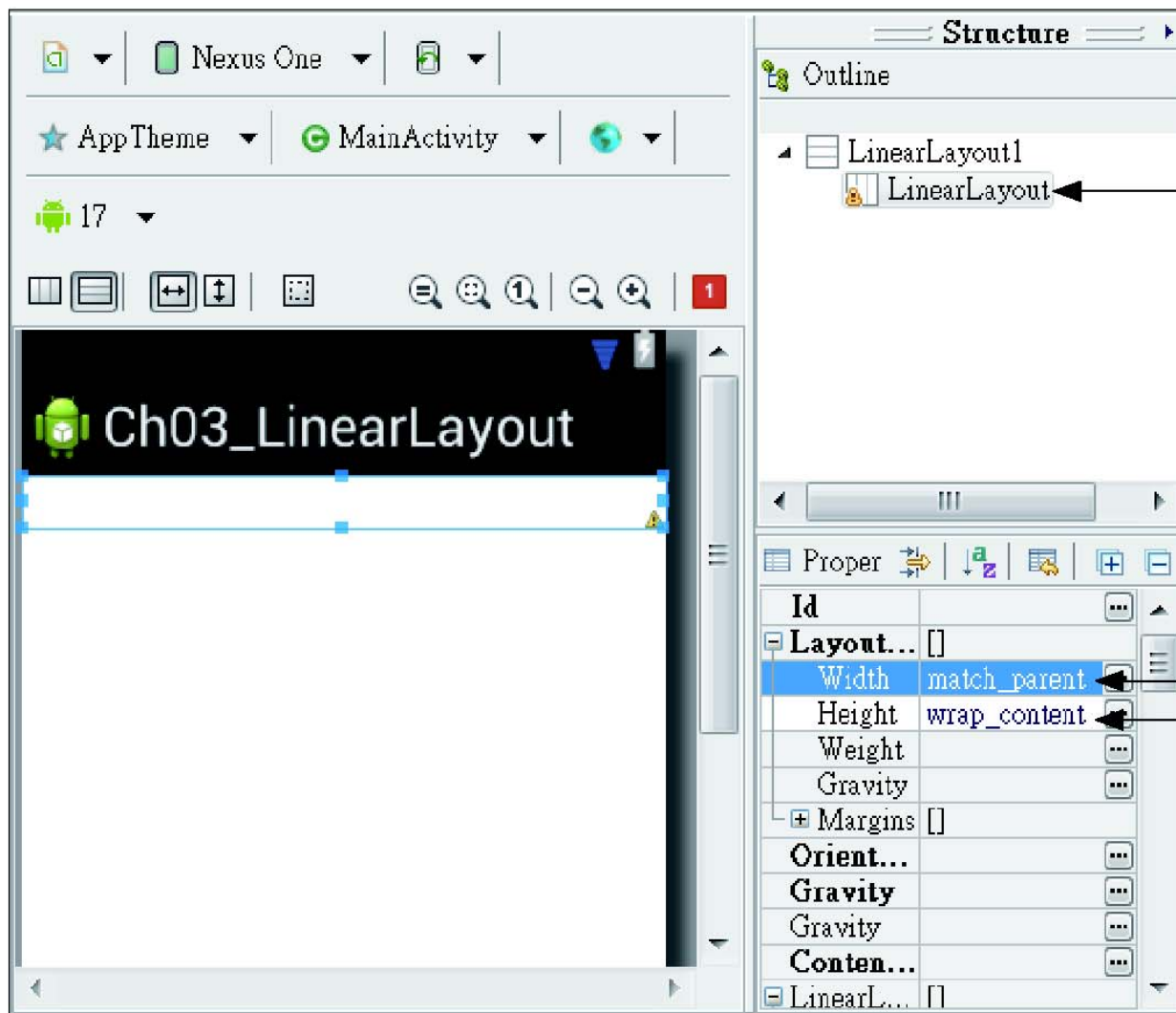
step 1

- 1 在 Layouts 項目上按一下，展開其內容



- 2 按住 LinearLayout (Horizontal) 不放，將它拉曳到空白處 (也就是 LinearLayout (Vertical) 的空間) 放開

使用 LinearLayout (Horizontal) 建立表單



還沒在 Layout 中加入任何元件時，會出現警告，可暫不理會（等加入元件後，警告就會消失）

寬度屬性，"match_parent" 表示『與上層的元件同寬』

高度屬性，"wrap_content" 表示『符合所含元件的高度』

使用 **LinearLayout (Horizontal)** 建立表單

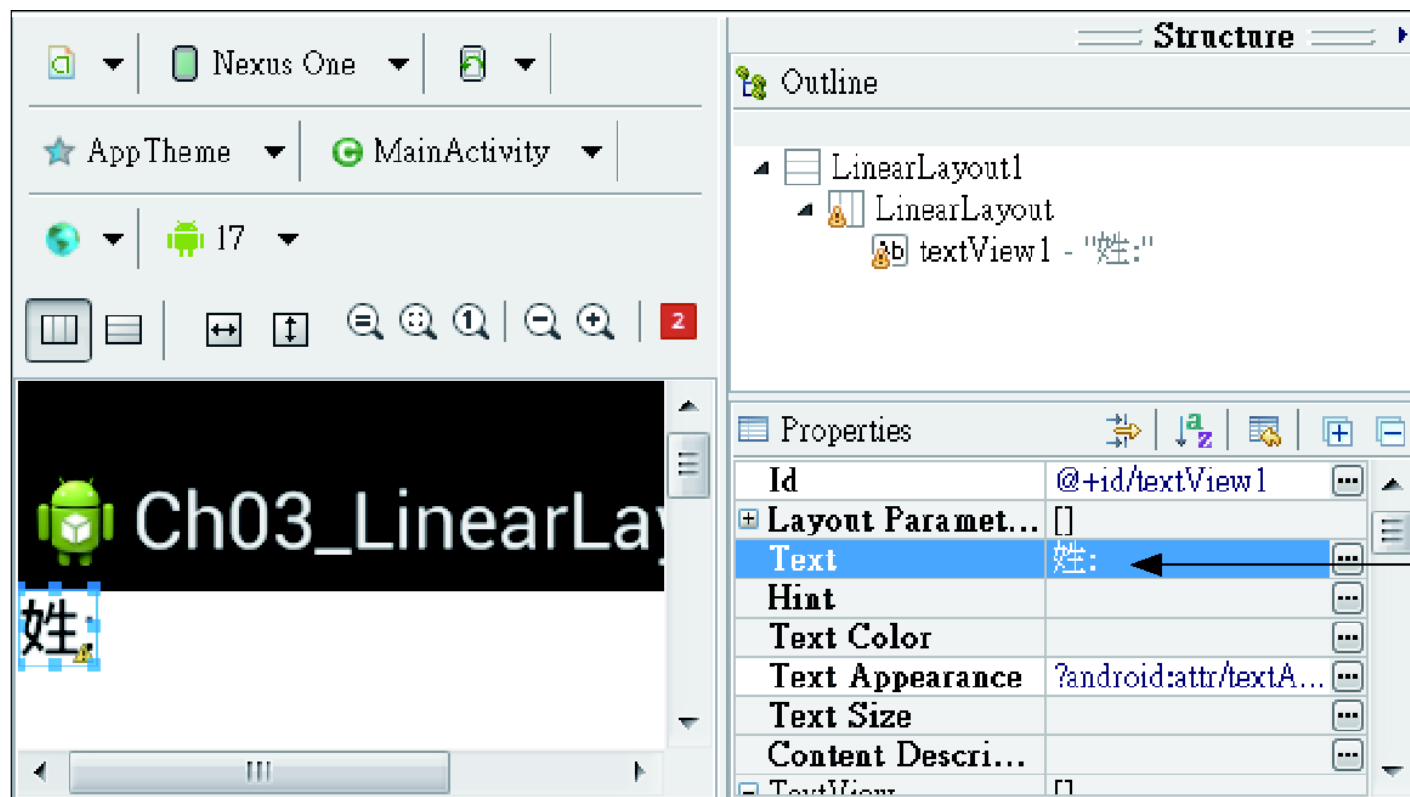
step 2

- 1 在 **Form Widgets** 上按一下, 展開其內容
這一系列都是 **TextView** 元件, Large 表示使用大字型、Medium 使用中字型, Small 使用小字型

2 將 **Large** 拉曳到剛剛加入的 **LinearLayout (Horizontal)** 之中

如果拉曳到外面, 則 **TextView** 會變成加到上層 **LinearLayout (Vertical)** 中

使用 LinearLayout (Horizontal) 建立表單



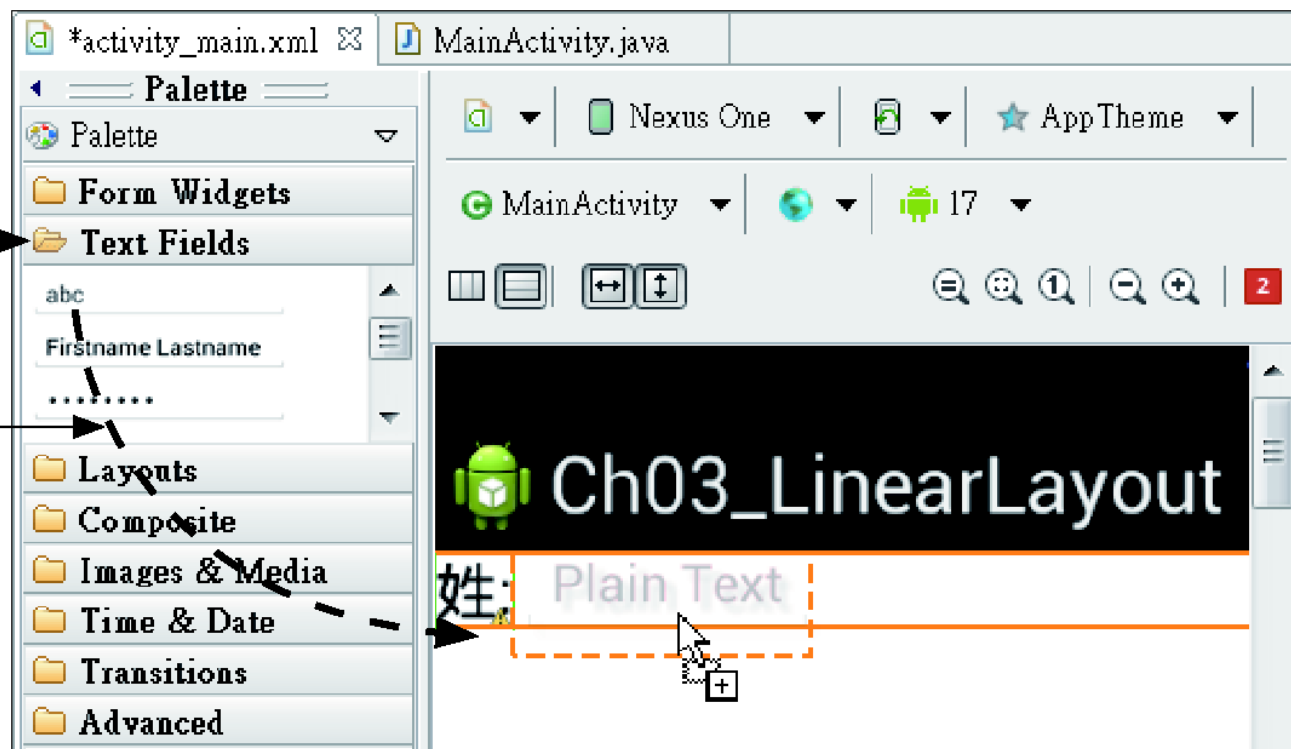
3 將 TextView 的文字改成『姓:』

使用 **LinearLayout (Horizontal)** 建立表單

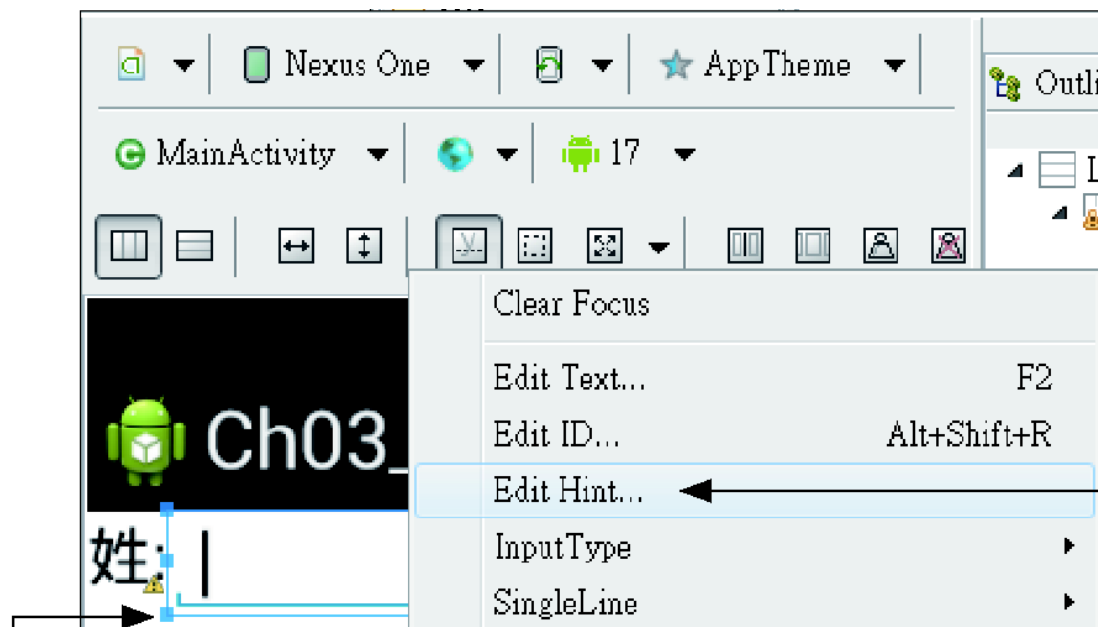
step 3

1 在 **Text Fields** 項目上按一下, 展開其內容

2 將 "abc" 項目拉曳到 **TextView** 右邊放開



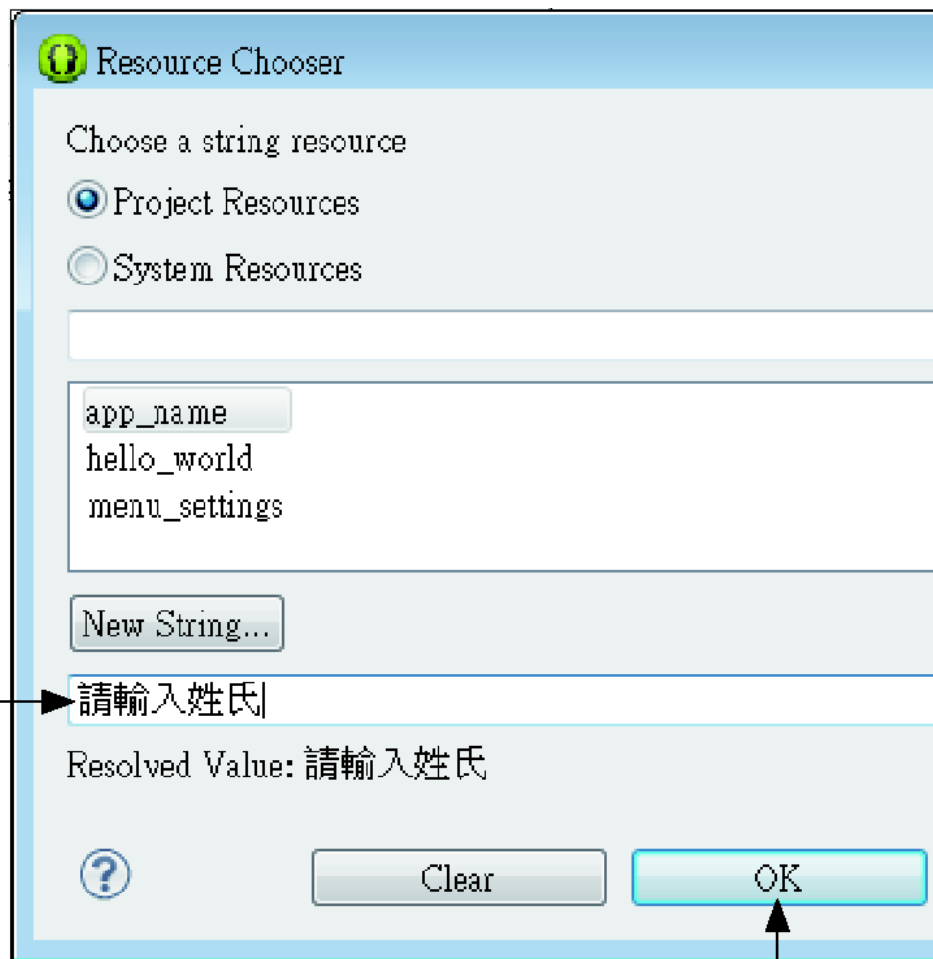
使用 **LinearLayout (Horizontal)** 建立表單



3 在 EditText 元件上按滑鼠右鈕，執行『Edit Hint』命令

元件在水平的 LinearLayout 中由左至右排列

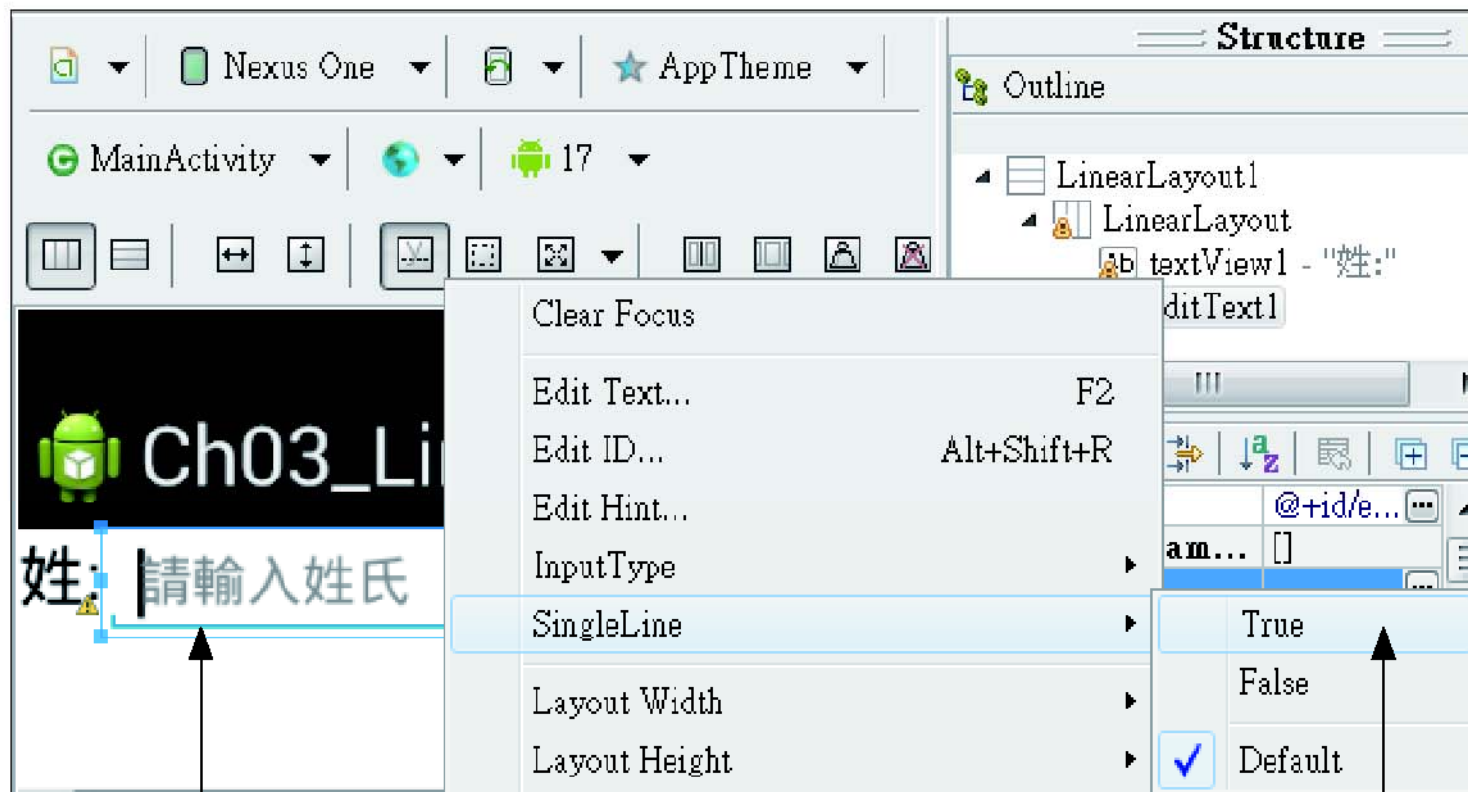
使用 **LinearLayout (Horizontal)** 建立表單



4 輸入提示文字

5 按 OK 鈕

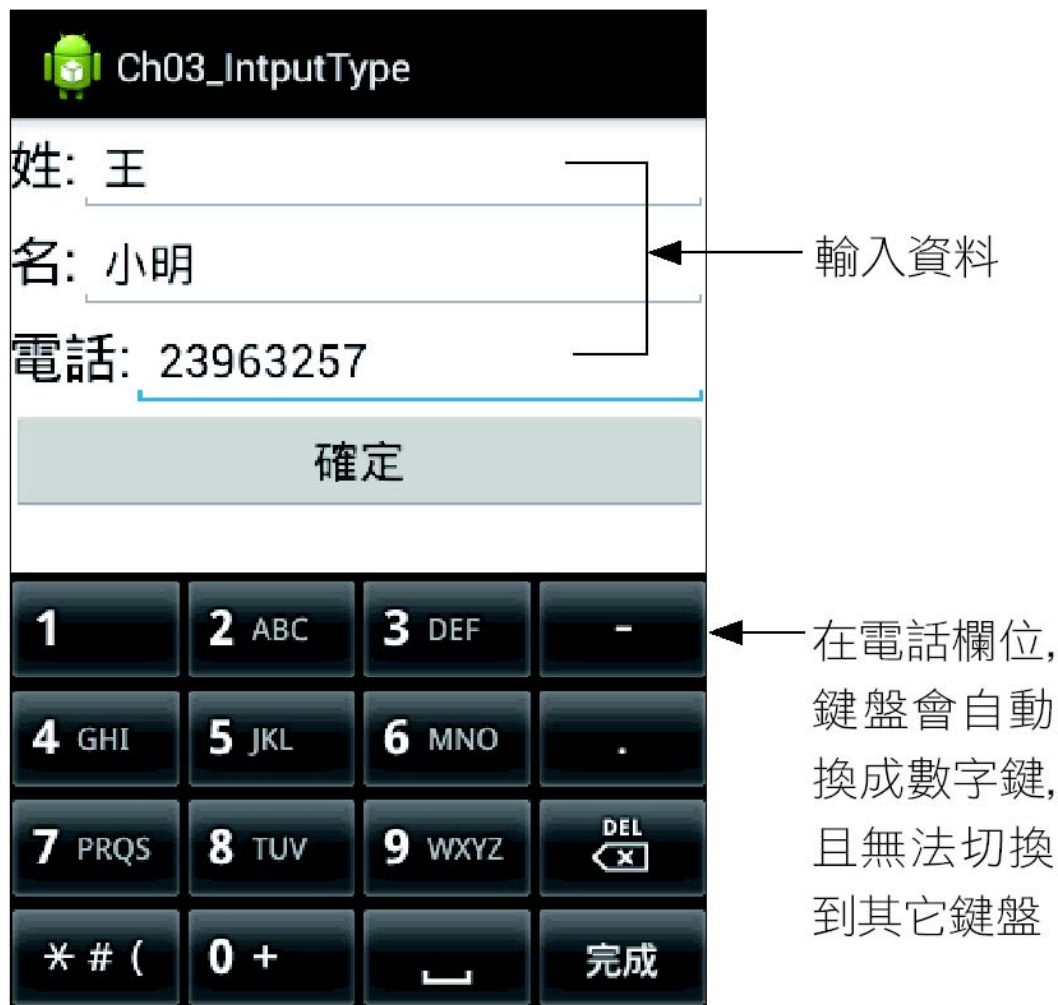
使用 **LinearLayout (Horizontal)** 建立表單



提示的文字會
以淺灰色顯示

6 在 EditText 元件上按滑鼠右鈕, 執行『**Single Line/True**』命令, 將之設為只限單行

範例3-3：加入輸入電話專用的 EditText



加入輸入電話專用的 EditText

step 1

1 複製出新專案

Resources Elements

- app_name (String)
- hello_world (String)
- menu_settings (String)

Attributes for app_name (String)

Name: app_name

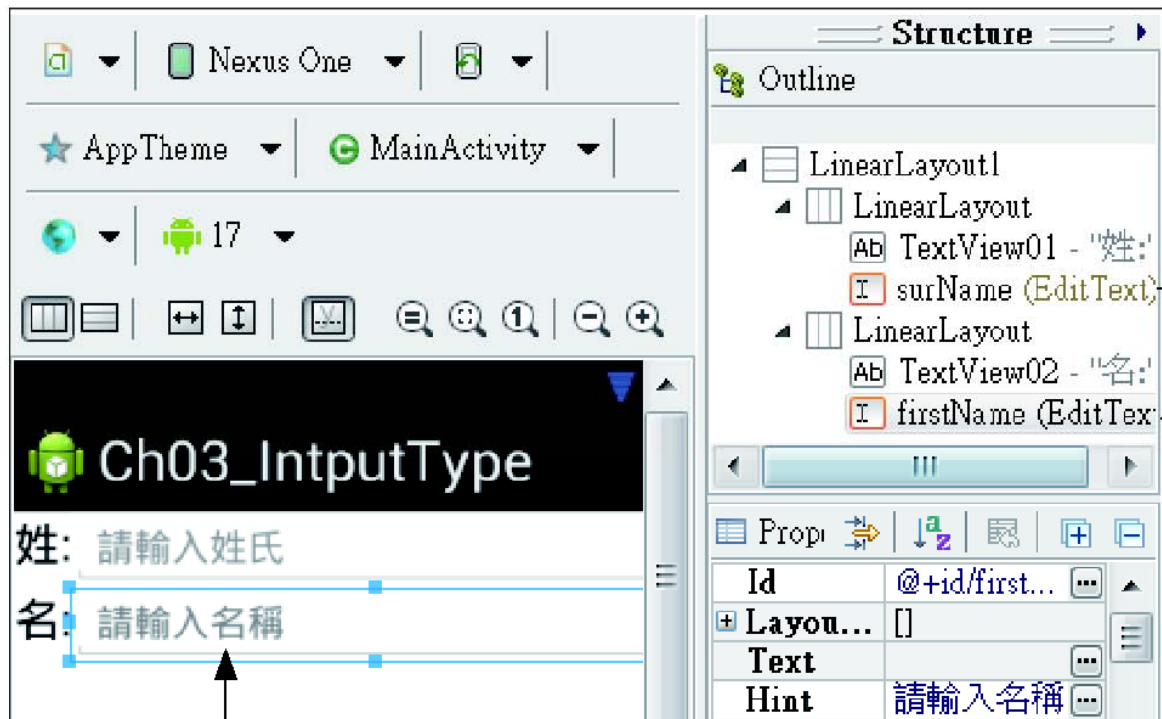
Value*: Ch03_InputType

3 修改 app_name 字串, 完成後按 **Ctrl** + **S** 存檔

2 開啟字串資源檔

加入輸入電話專用的 EditText

step 2

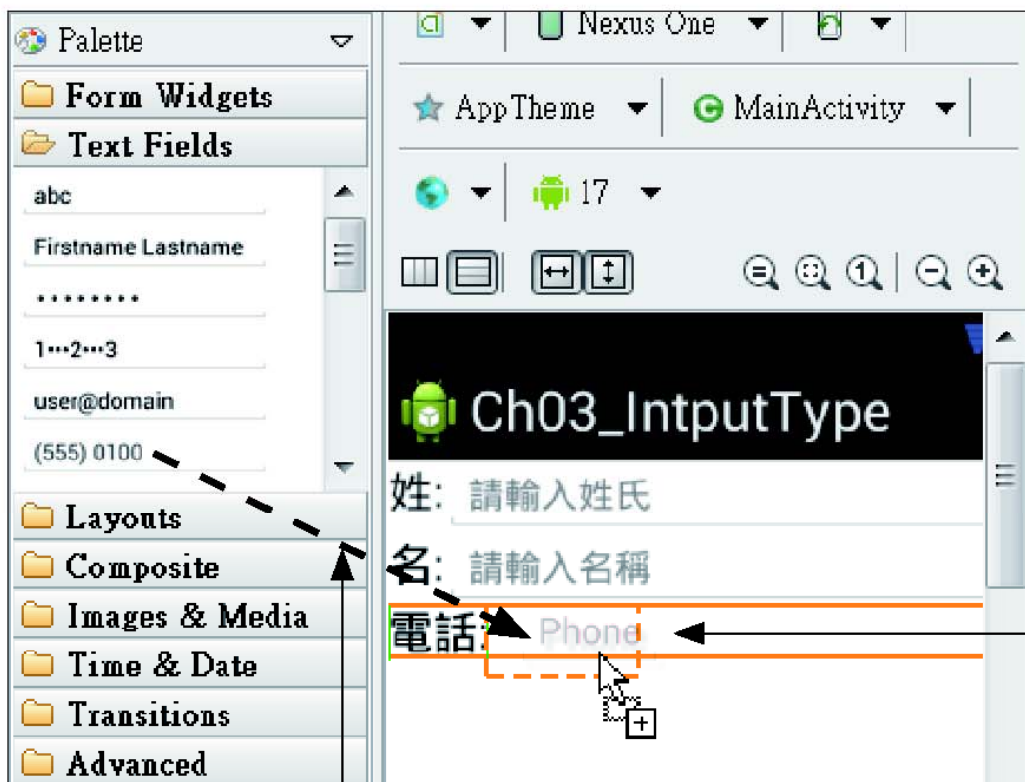


1 加入 LinearLayout (Horizontal), 內含 TextView、EditText

2 將 EditText 的 Id 屬性分別改為 surName、firstName。還記得嗎? 我們必須為元件取名稱 (設 Id 屬性值), 才能從程式中存取到相對應的物件, 然後才能透過其方法來使用物件

加入輸入電話專用的 EditText

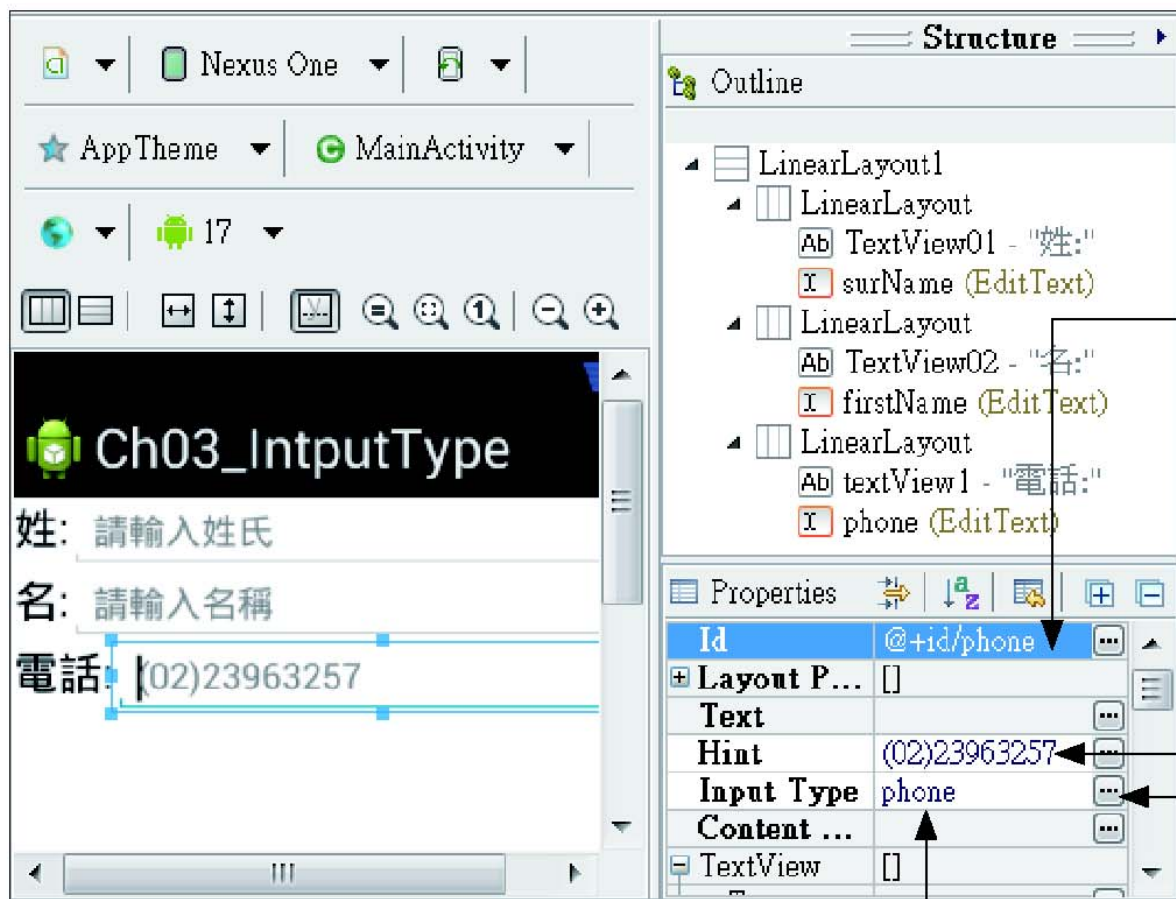
step 3



1 先加入 LinearLayout (Horizontal), 並在其中加入 TextView

2 加入電話號碼欄位

加入輸入電話專用的 EditText



3 設定 Id 為 "phone" (在屬性欄位要輸入"@+id/phone"), 設了 Id 之後才能在程式中使用 findViewById() 取得物件 (見下頁的程式 12~15 行)

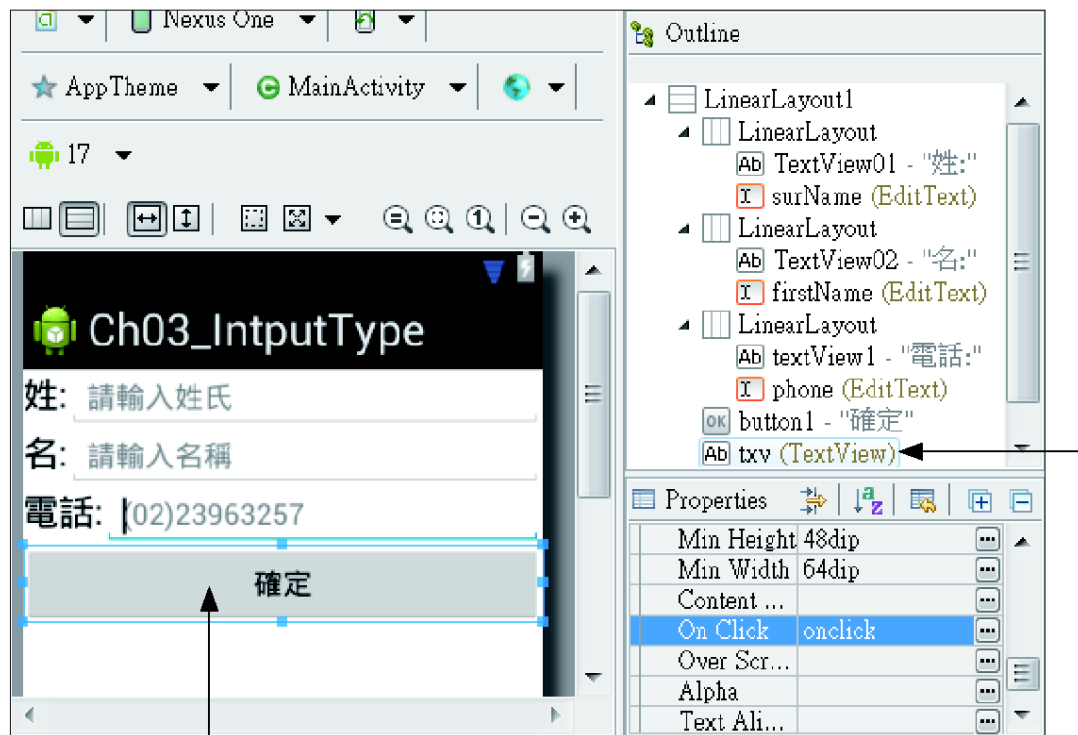
4 輸入 Hint, 並設定 Single Line 屬性為 "true" (方法參見 3-15 頁)

按此鈕可展開交談窗, 選擇其它類型

Input Type 屬性值為 phone

加入輸入電話專用的 EditText

step 4



1 加入 Button 元件, 並設定屬性

屬性	屬性值
Layout Parameters/Width	match_parent
Text	確定
On Click	onclick

2 加入 TextView 元件至 Button 元件之下, 並設定屬性

屬性	屬性值
ID	@+id/txv
Text	(空白)
Text Size	30sp

加入輸入電話專用的 EditText

step 5

```
01 public class MainActivity extends Activity {
02     // 宣告代表 UI 元件的變數
03     EditText sname, fname, phone;
04     TextView txv;
05
06     @Override
07     protected void onCreate(Bundle savedInstanceState) {
08         super.onCreate(savedInstanceState);
09         setContentView(R.layout.activity_main);
10
11         // 初始化變數
```

加入輸入電話專用的 EditText

```
12     sname = (EditText) findViewById(R.id.surName);
13     fname = (EditText) findViewById(R.id.firstName);
14     phone = (EditText) findViewById(R.id.phone);
15     txv = (TextView) findViewById(R.id.txv);
16 }
...
25 public void onclick(View v){
26     txv.setText(sname.getText().toString()+    ← 取得姓
27               fname.getText()+              ← 取得名
28               "的電話是 "+ phone.getText()); ← 取得電話
29 }
30 }
```

加入輸入電話專用的 EditText

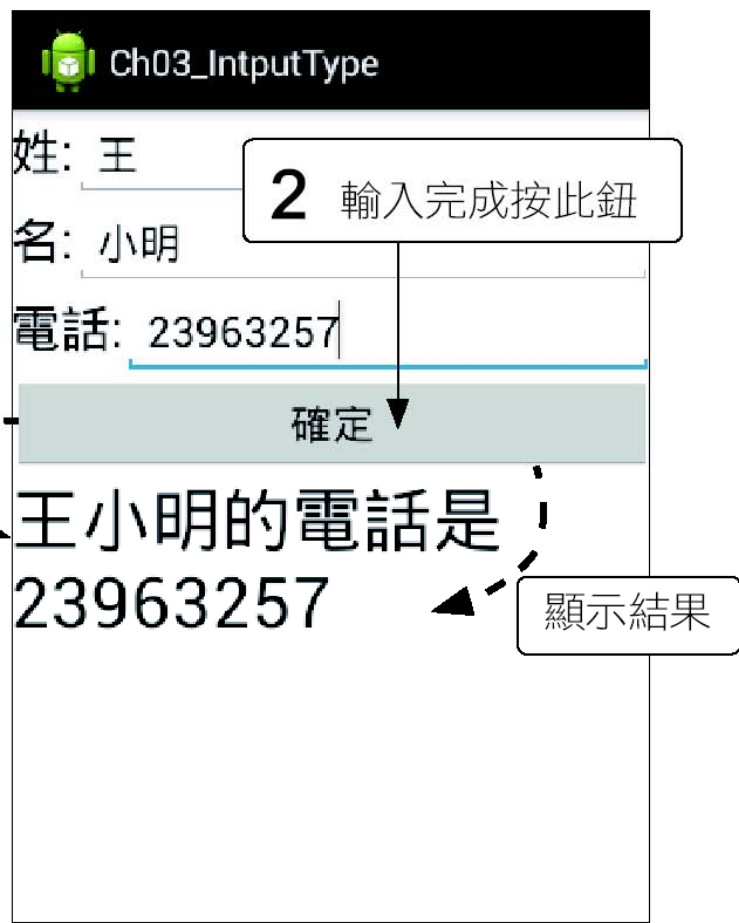
step 6



1 輸入資料



在電話欄位，
鍵盤會自動
換成數字鍵，
且無法切換
到其它鍵盤



2 輸入完成按此鈕

顯示結果

3-3 使用 Weight 屬性控制元件的寬/高



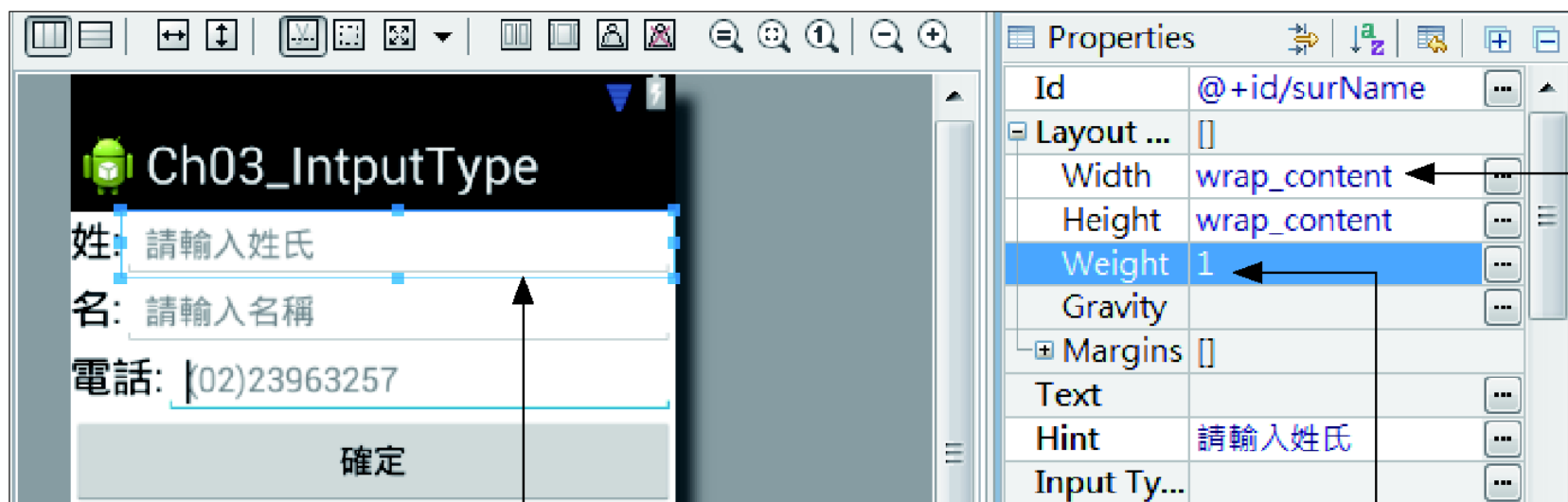
因 TextView 寬度不一，
所以元件未對齊



利用 Weight 屬性可讓
寬度不一的元件對齊

使用 Weight 屬性控制元件的寬/高

- 1 Width 屬性預設為 wrap_content, 不會自動延伸到與上層的 Layout元件同寬

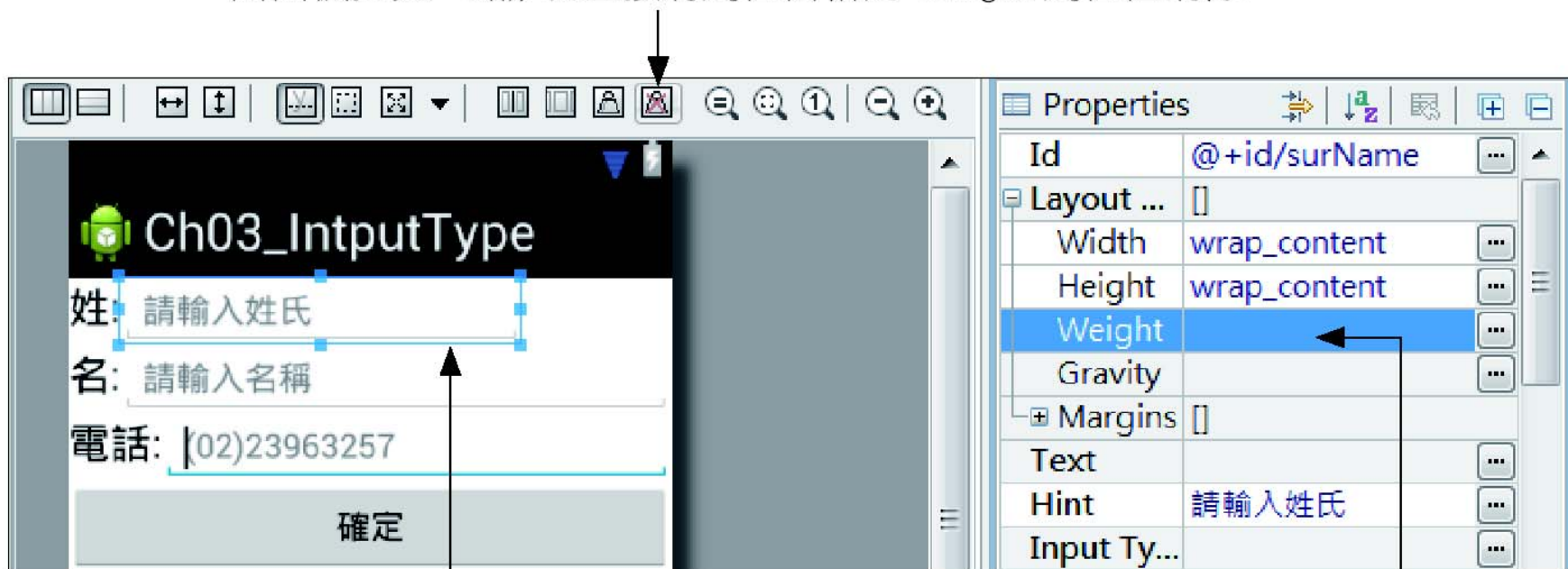


- 3 剩餘的空間都分配給 TextEdit 元件, 使得寬度還是會延伸到畫面右側

- 2 但 Weight 屬性預設為 1

使用 Weight 屬性控制元件的寬/高

按此鈕可清除 Weight 屬性值 (若沒看到這個按鈕, 請將預覽畫面面板拉寬一點), 或直接將屬性窗格的 Weight 屬性值清除



EditText 元件變短了

Weight 屬性值被清除

範例3-4：利用 Weight 屬性對齊元件

step 1

step 2

2 按此鈕 (Distribute Weight Evenly) 可自動平分空間

The screenshot shows the Android Studio IDE with a form layout titled "Ch03 Weight屬性範例". The form contains three input fields: "姓: 請輸入姓氏", "名: 請輸入名稱", and "電話: (02)23963257", followed by a "確定" button. A callout box labeled "1" points to the first input field with the text "先選取第 1 列的 TextView 元件". Another callout box labeled "2" points to the "Distribute Weight Evenly" button in the toolbar with the text "按此鈕 (Distribute Weight Evenly) 可自動平分空間". The right-hand side of the IDE shows the Hierarchy and Properties panels. The Hierarchy panel shows a tree structure of LinearLayouts and TextViews. The Properties panel shows the Weight property of the selected TextView set to wrap_content.

1 先選取第 1 列的 TextView 元件

2 按此鈕 (Distribute Weight Evenly) 可自動平分空間

Id	@+id/TextVie...
Layout ...	[]
Width	wrap_content
Height	wrap_content
Weight	wrap_content
Gravity	

利用 Weight 屬性對齊元件

第 1 列的 TextView、EditText 元件變成等寬了

TextView 元件的 Width 屬性被設為 0dp

Weight 屬性被設為 1

MainActivity | Android 17

姓: 請輸入姓氏
名: 請輸入名稱
電話: (02)23963257
確定

```
LinearLayout1
├── LinearLayout
│   ├── TextView01 - "姓:"
│   └── surName (EditText)
├── LinearLayout
│   ├── TextView02 - "名:"
│   └── firstName (EditText)
├── LinearLayout
│   ├── textView1 - "電話:"
│   ├── phone (EditText)
│   ├── button1 - "確定"
│   └── txv (TextView)
```

Property	Value
Id	@+id/TextVie...
Layout ...	
Width	0dp
Height	wrap_content
Weight	1
Gravity	

利用 Weight 屬性對齊元件

3 選取 EditText 元件

Id	@+id/surName
Layout ...	
Width	0dp
Height	wrap_content
Weight	1
Gravity	

可以發現 Width、Weight 屬性同樣被設為 0dp、1

利用 Weight 屬性對齊元件

step 3

1 選取第 2 排中的 TextView

2 按此鈕

3 選取 EditText, 設定效果也相同

4 按此鈕

Properties

Id	@+id/TextVie...	...
Layout ...	[]	...
Width	0dp	...
Height	wrap_content	...
Weight	1	...
Gravity		...

3-4 透過屬性美化外觀

 元件外觀屬性範例

姓:

名:

電話:

未設定邊界,
版面較緊密

 元件外觀屬性範例

姓:

名:

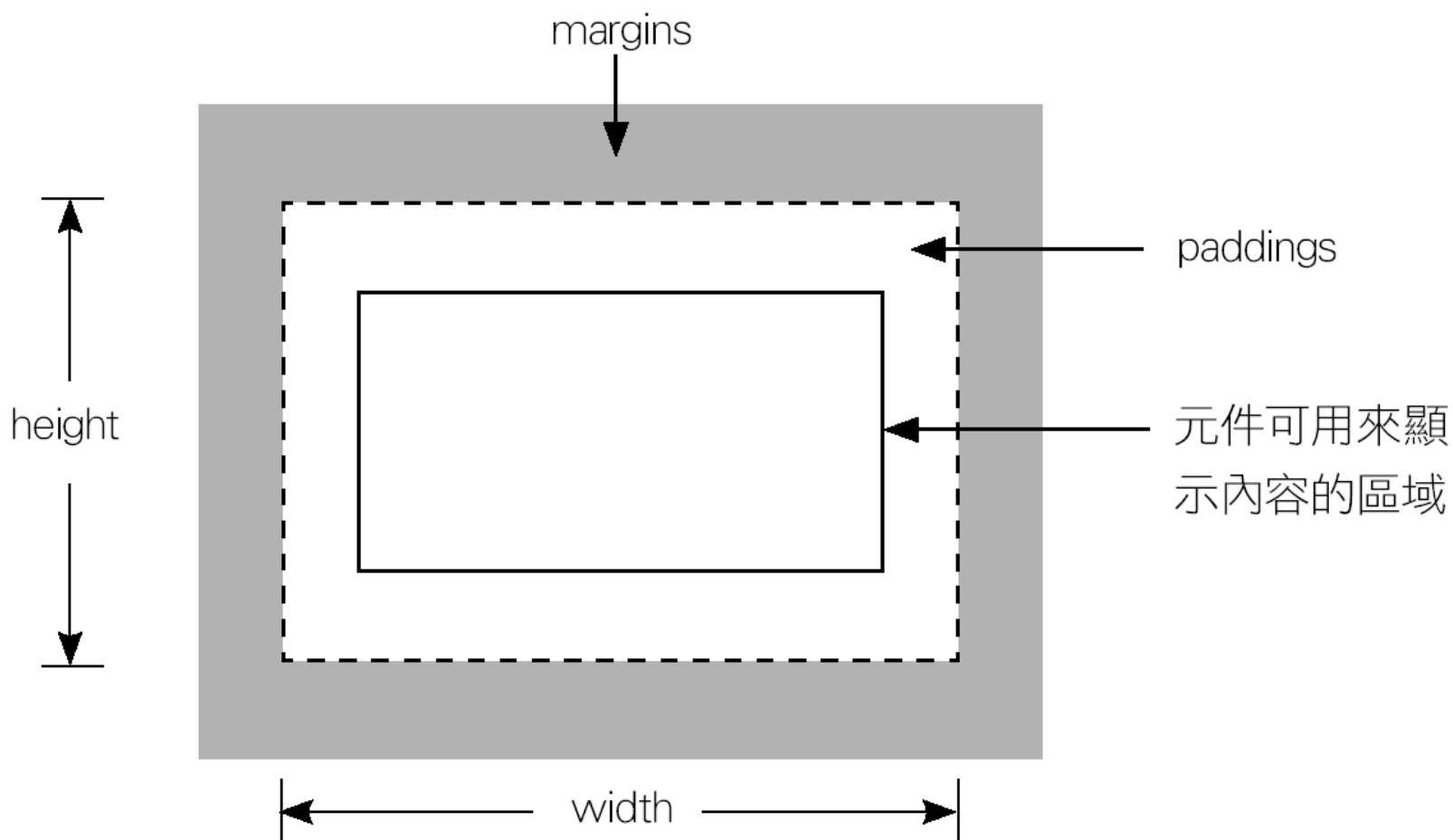
電話:

有設定邊界,
版面較寬鬆

透過屬性美化外觀

- 元件的邊界：margins 與 paddings
- 顏色：以RGB 值指定文字或背景顏色

元件的邊界：margins 與 paddings

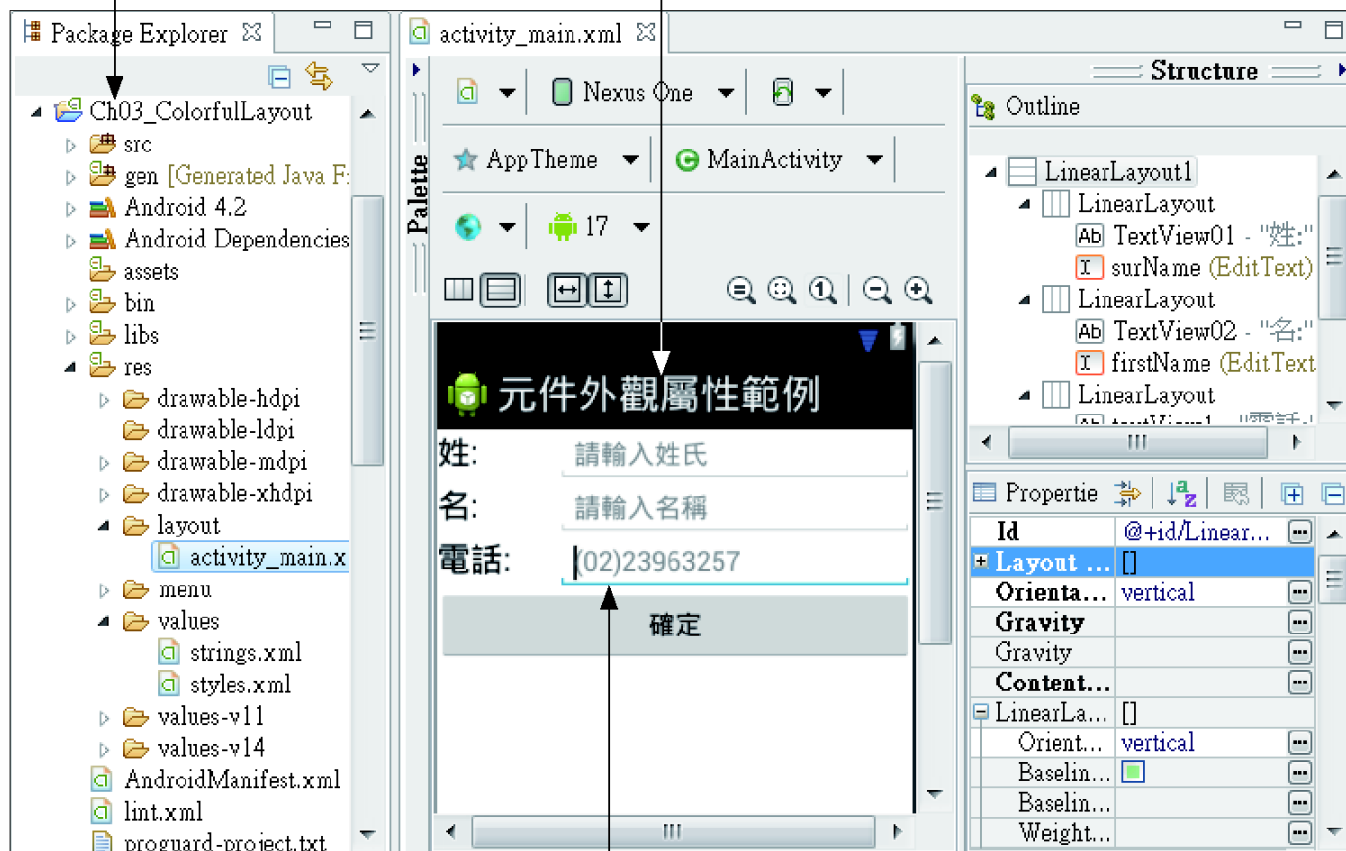


範例3-5：設定邊界讓輸入表單版面變寬鬆

step 1

1 複製出新專案

2 修改 res/values/strings.xml 資源中的 app_name 字串即可更改預覽畫面上的 Activity 標題



3 請將 3 個 EditText 的 Weight 屬性值都設為 3

設定邊界讓輸入表單版面變寬鬆

step 2

1 選取第 2 列的 LinearLayout (Horizontal)

2 展開 Layout Parameters/ Margins 項目

3 在 Margin 屬性欄輸入 "10dp" 表示上、下、左、右邊界都是 10 dp

Id	Value
Layout P...	
Width	match_parent
Height	wrap_content
Weight	
Gravity	
Margins	
Margin	10dp
Left	10dp
Top	10dp
Right	

設定邊界讓輸入表單版面變寬鬆



4 按 **Enter** 或 **Tab** 鍵

這一系列上下左右都空
出一些空間了

顏色：以 RGB 值指定文字或背景顏色

以 # 符號開頭

#FF7F00

紅色的值 綠色的值 藍色的值

(16 進位 FF 等於 10 進位 255) (16 進位 7F 等於 10 進位 127)

→ 橘色 (= 紅色 + 淺綠)

顏色：以 RGB 值指定文字或背景顏色

3 記下 (或複製) 16 進位數值

1 在此選取想要的色系

ColorPicker.com : Quick Online Color Picker Tool | HTML Color Codes - ...

http://www.color... Bina

我的最愛 建議的網站 取得更多附加元件

ColorPicker.com : Quick Online Color Pick... 網頁(P) 工具(O)

ColorPicker.com #7BAFB3

顏色預覽區

H: 184
S: 31
B: 70
R: 123
G: 175
B: 179

十進位的 RGB 值

Add to My Colors

網際網路 | 受保護模式: 啟動 100%

2 在此選取所要顏色的濃淡、明暗

顏色：以 RGB 值指定文字或背景顏色

4 在元件的 "Text Color" 欄
輸入以 "#" 開頭的顏色值

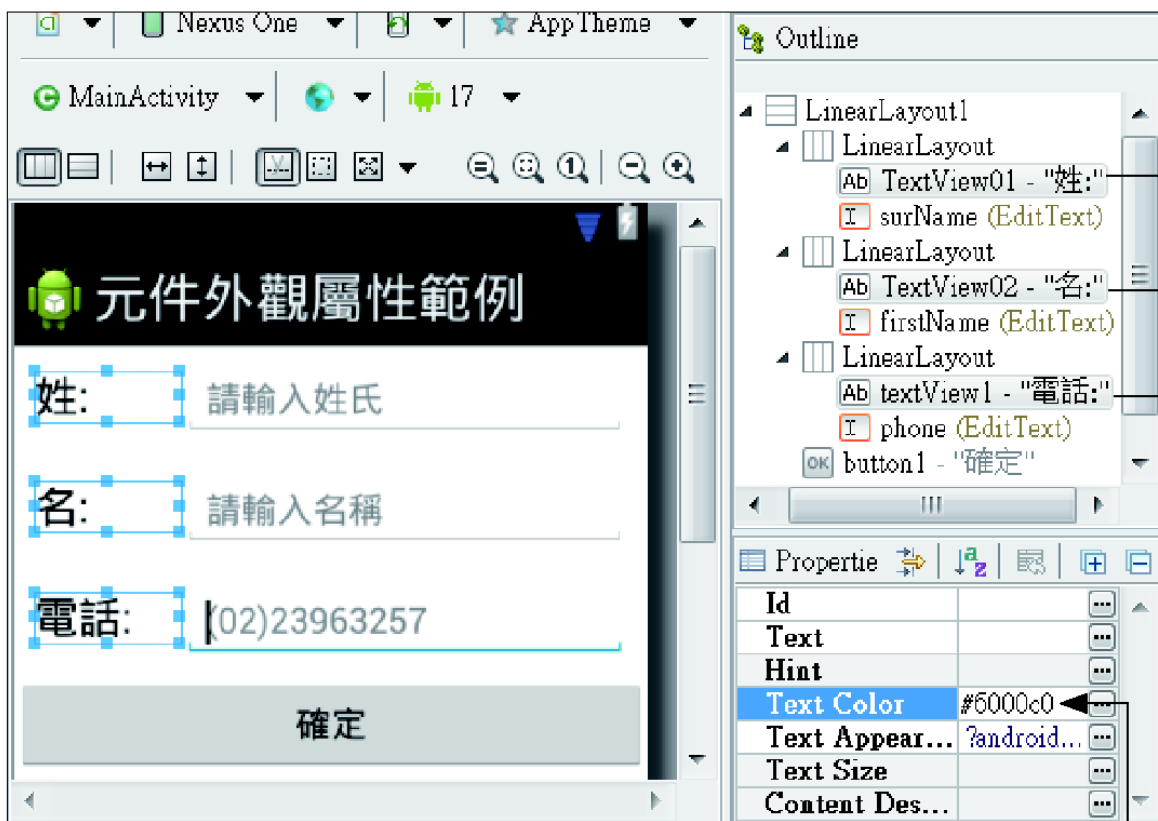


輸入後在欄位中會
出現該顏色的色塊

範例3-6：設定文字及背景顏色

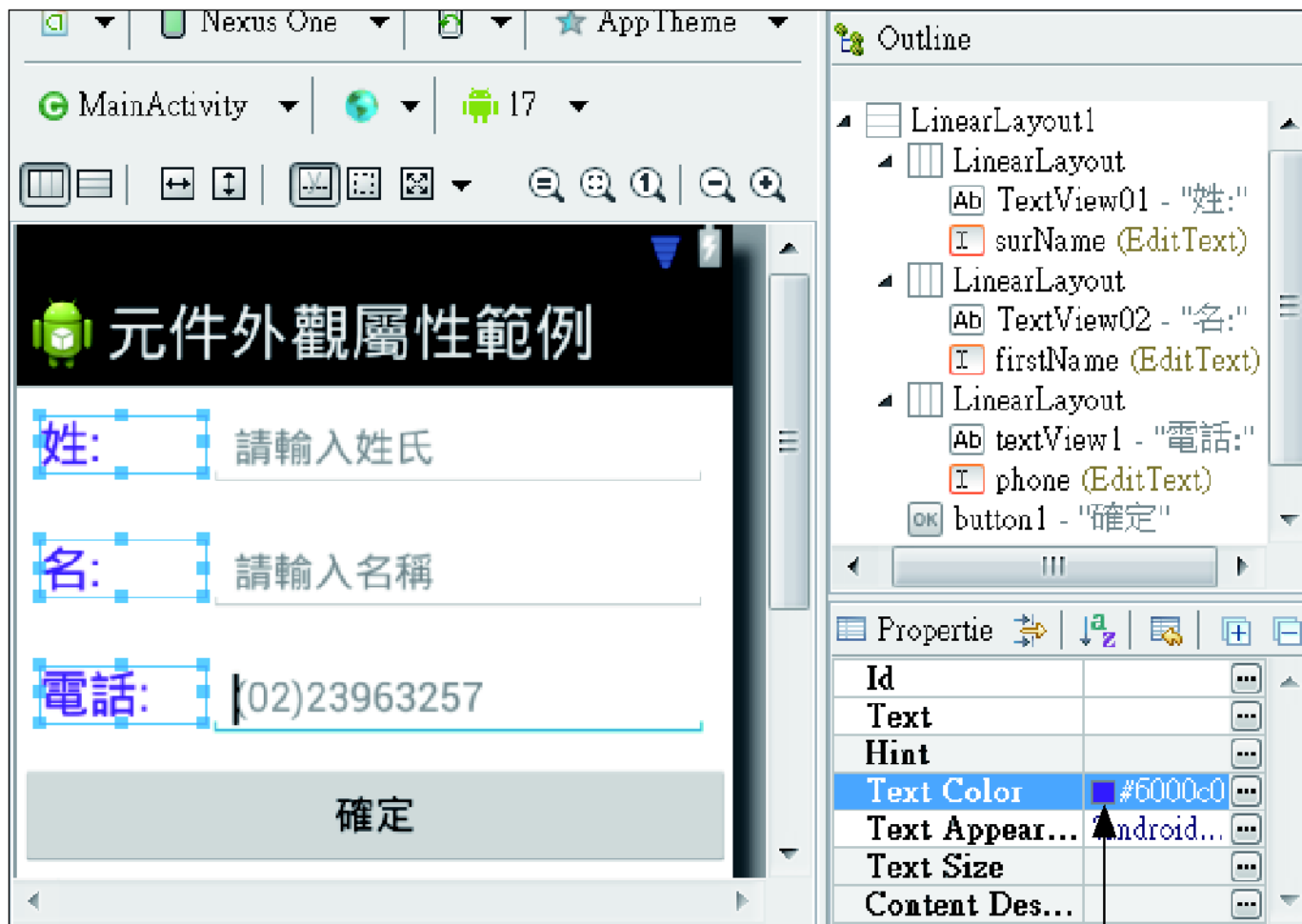
step 1

1 按住 **Ctrl** 鍵不放, 選取 3 個 TextView 項目



2 將 **Text Color** 屬性設為 "#6000c0"
(英文字母大小寫均可), 再按 **Tab** 鍵

設定文字及背景顏色

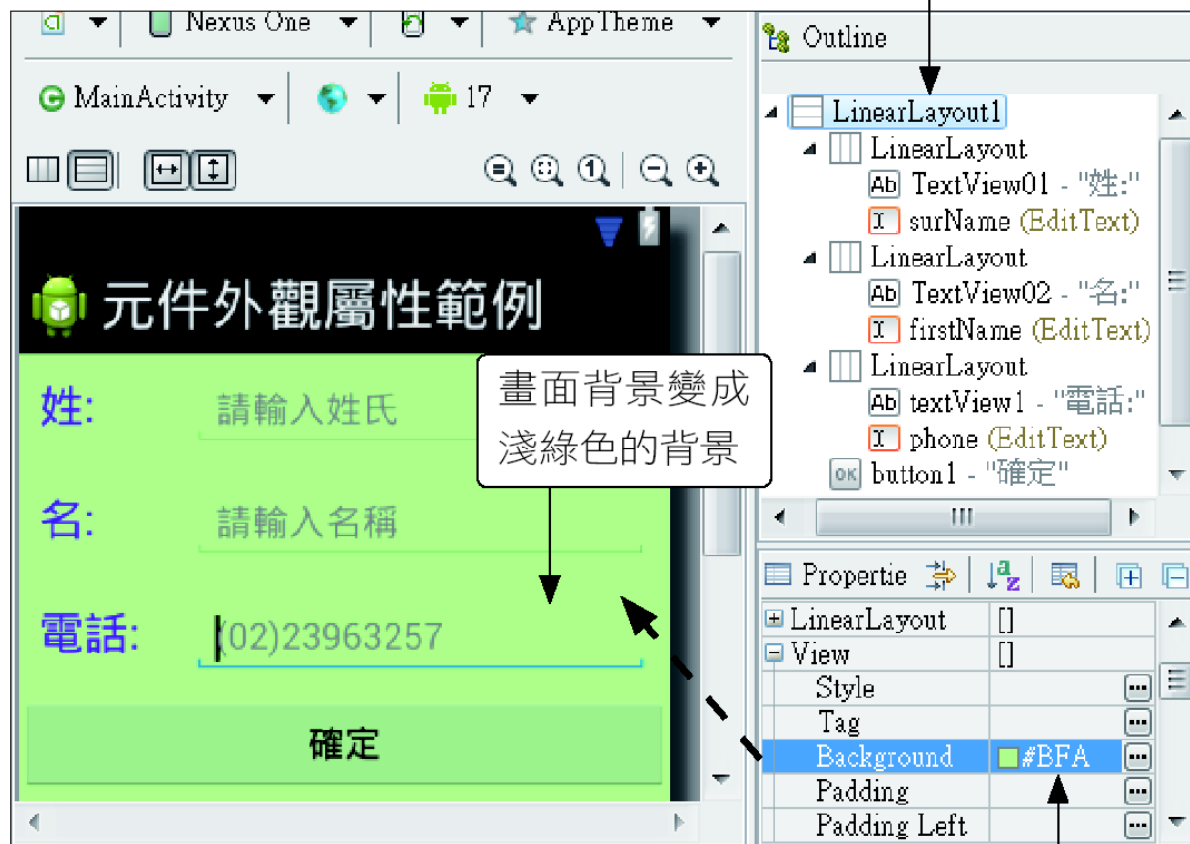


屬性值前面會有色塊

設定文字及背景顏色

step 2

1 選最上層的 LinearLayout1



2 在 View 項目下的 Background 欄輸入 "#BFA"

3-5 以程式設定元件的外觀屬性

屬性名稱	在程式中設定的方式
padding	<code>(View 類別的物件).setPaddings(5,10,5,10);</code> // 參數順序為：左、上、右、下 // (順時針)
Text Color	<code>(TextView 類別的物件).setTextColor(Color.Red);</code> // 將文字設為紅色
Background	<code>(View 類別的物件).setBackgroundColor(顏色值);</code>
Text Size	<code>(TextView 類別的物件).setTextSize(30)</code> // 設定文字大小 (單位為 sp)

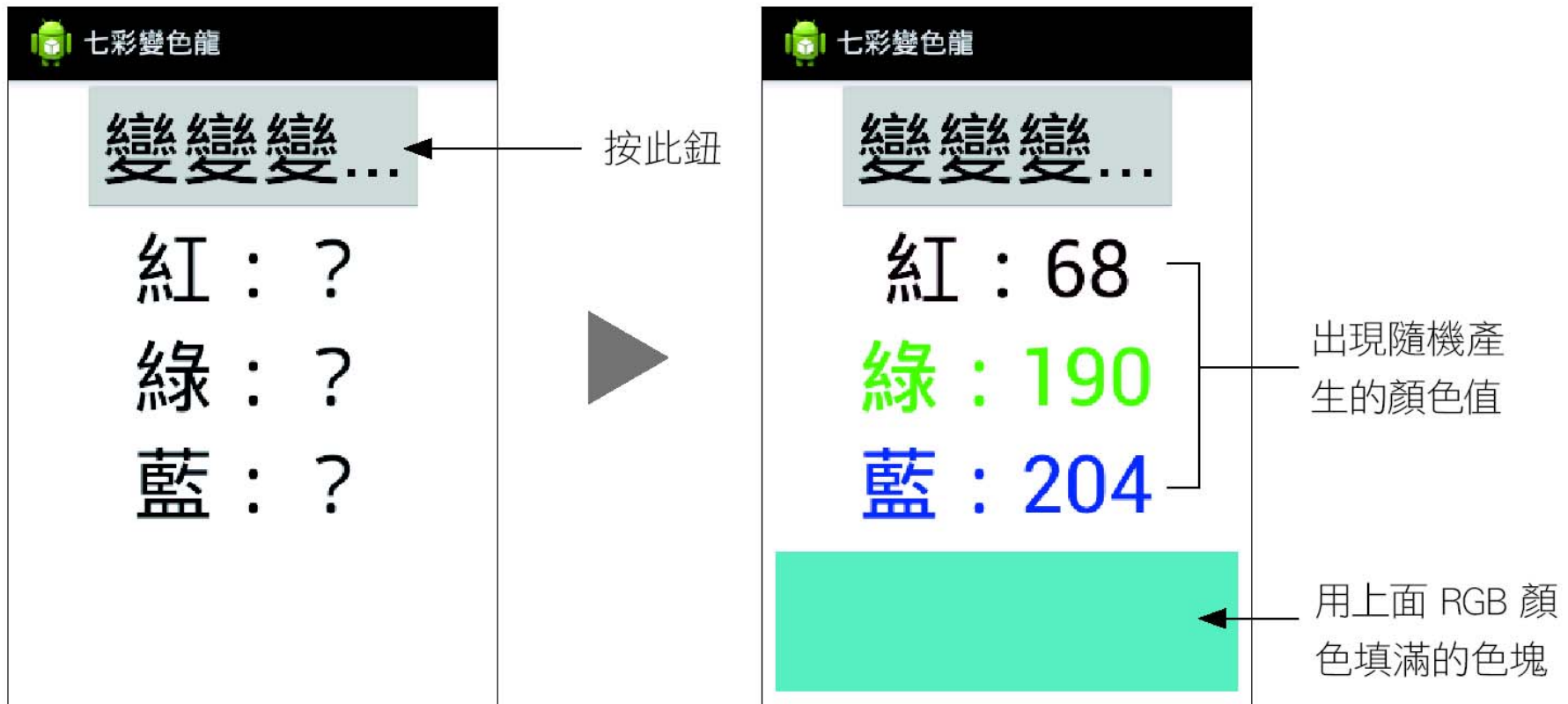
以程式設定元件的外觀屬性

- `setTextColor()`：改變文字顏色

```
TextView txv = (TextView) findViewById(R.id.txv);  
// 以下兩行都是將文字設為紅色  
txv.setTextColor(Color.Red);  
txv.setTextColor(Color.rgb(255, 0, 0)); ← rgb() 三個參數分別是紅、綠、藍  
                                         三原色的強度，可為 0~255  
  
// 以下兩行都是將文字設為黃色  
txv.setTextColor(Color.Yellow);  
txv.setTextColor(Color.rgb(255, 255, 0));
```

```
// Color.rgb(255, 0, 255), 紫色  
Color.parseColor("#FF00FF"); ← 參數必須以字串表示
```

範例3-7：變色龍- 以亂數設定顏色屬性



變色龍- 以亂數設定顏色屬性

step 1

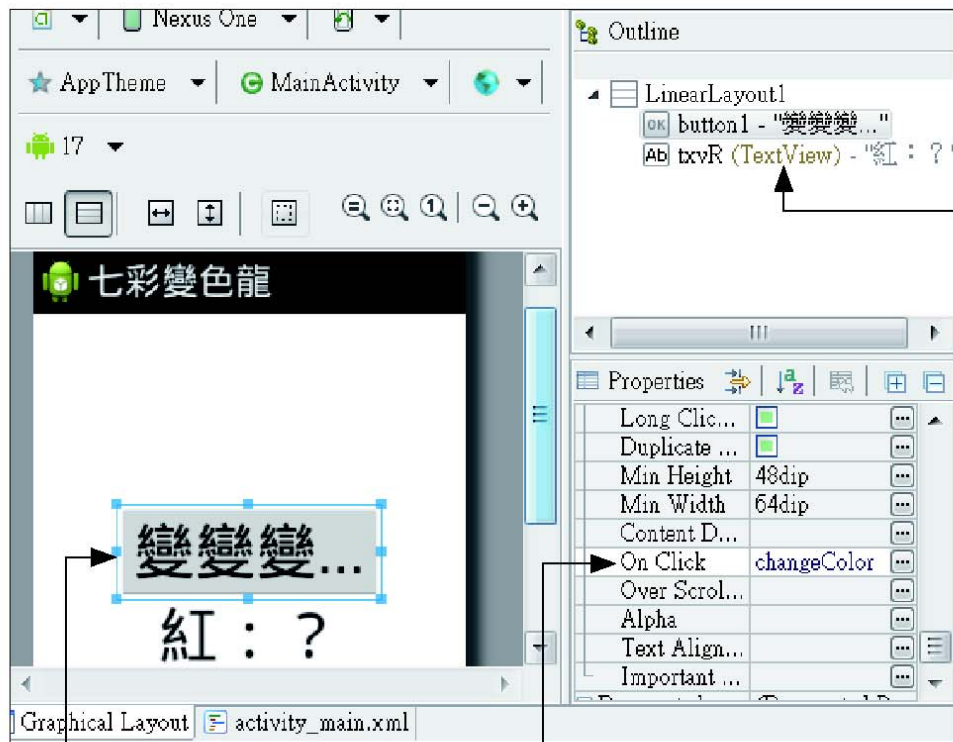
1 先將最上層的 Layout 元件換成 LinearLayout (Vertical)

2 按一下開啟選單

3 將 Gravity 屬性設為 center, 表示稍後加入的元件都會自動『置中』

這 2 個是同一個屬性

變色龍- 以亂數設定顏色屬性



3 修改預設 "Hello world!" 元件的屬性:

屬性	值
Id	@+id/txvR
Text	紅: ?
Text Size	45sp

4 加入一個按鈕元件:

屬性	值
Text	變變變...
Text Size	45sp
On Click	changeColor

On Click 屬性是列在屬性清單最後段

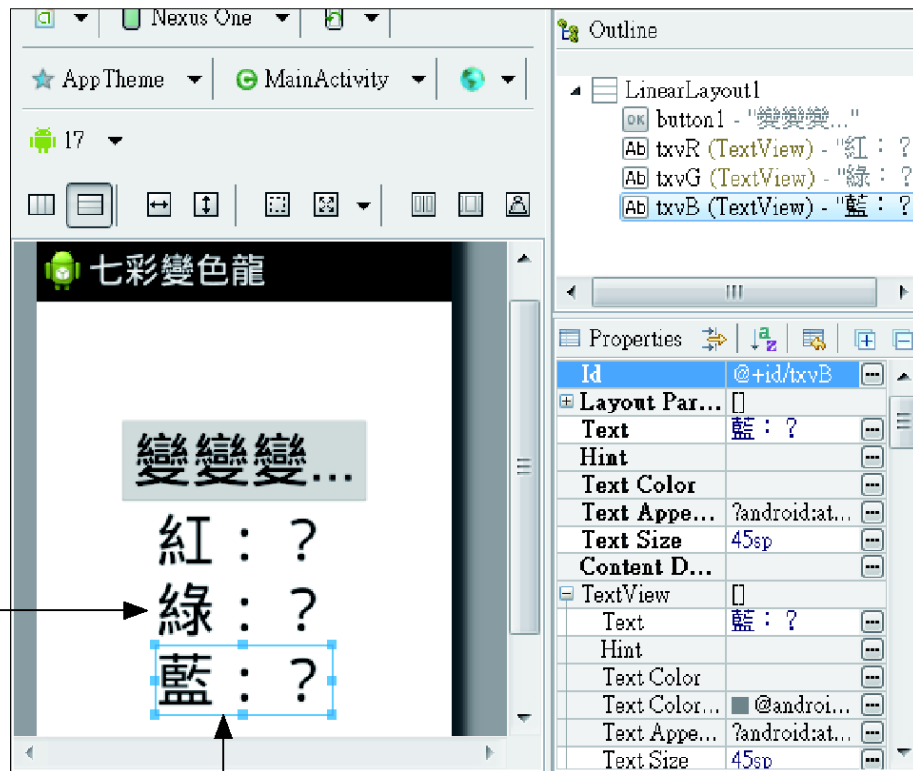
變色龍- 以亂數設定顏色屬性

5 加入另一個 TextView
並設定屬性：

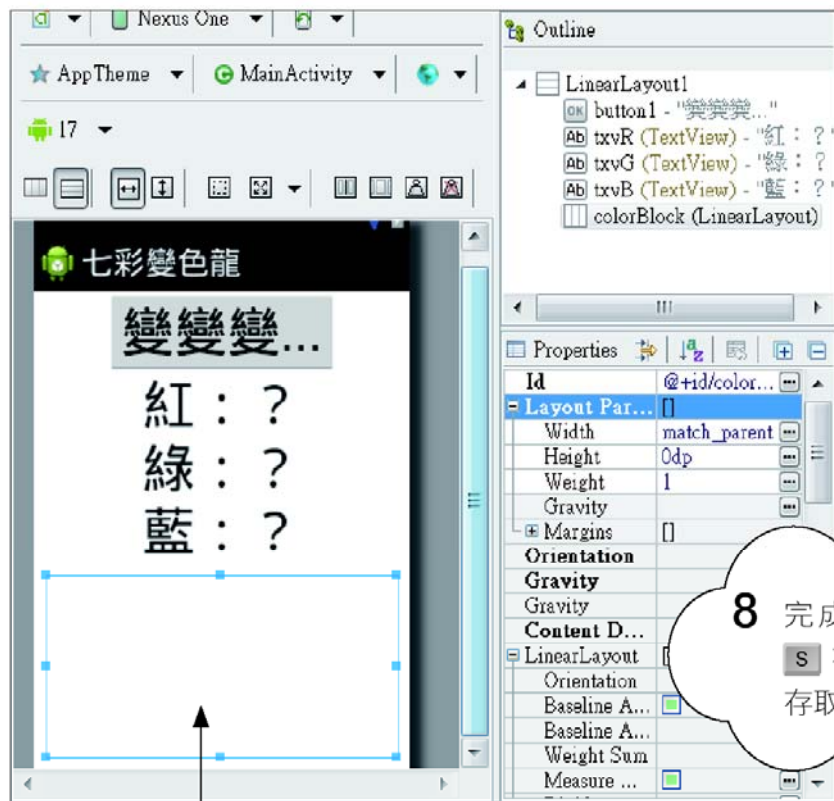
屬性	值
Id	@+id/txvG
Text	綠：？
Text Size	45sp

6 再加第 3 個 TextView
並設定屬性：

屬性	值
Id	@+id/txvB
Text	藍：？
Text Size	45sp



變色龍- 以亂數設定顏色屬性



8 完成後記得按 **Ctrl + S** 存檔, 在程式中才能存取元件的資源 ID

7 在最下方加入一個 LinearLayout (Horizontal)

屬性	值
Id	@+id/colorBlock
Margins/Margin	10dp
Height	0dp
Weight	1

變色龍- 以亂數設定顏色屬性

step 2

```
01 public class MainActivity extends Activity {
02     TextView txvR,txvG,txvB;
03     View colorBlock;
04     @Override
05     protected void onCreate(Bundle savedInstanceState) {
06         super.onCreate(savedInstanceState);
07         setContentView(R.layout.activity_main);
08
09         // 取得 3 個 TextView 的物件, 及畫面最下方的 LinearLayout
10         txvR = (TextView) findViewById(R.id.txvR);
```

Next

變色龍- 以亂數設定顏色屬性

```
11         txvG= (TextView) findViewById(R.id.txvG);  
12         txvB= (TextView) findViewById(R.id.txvB);  
13         colorBlock = findViewById(R.id.colorBlock);  
14     }  
...
```

變色龍- 以亂數設定顏色屬性

step 3

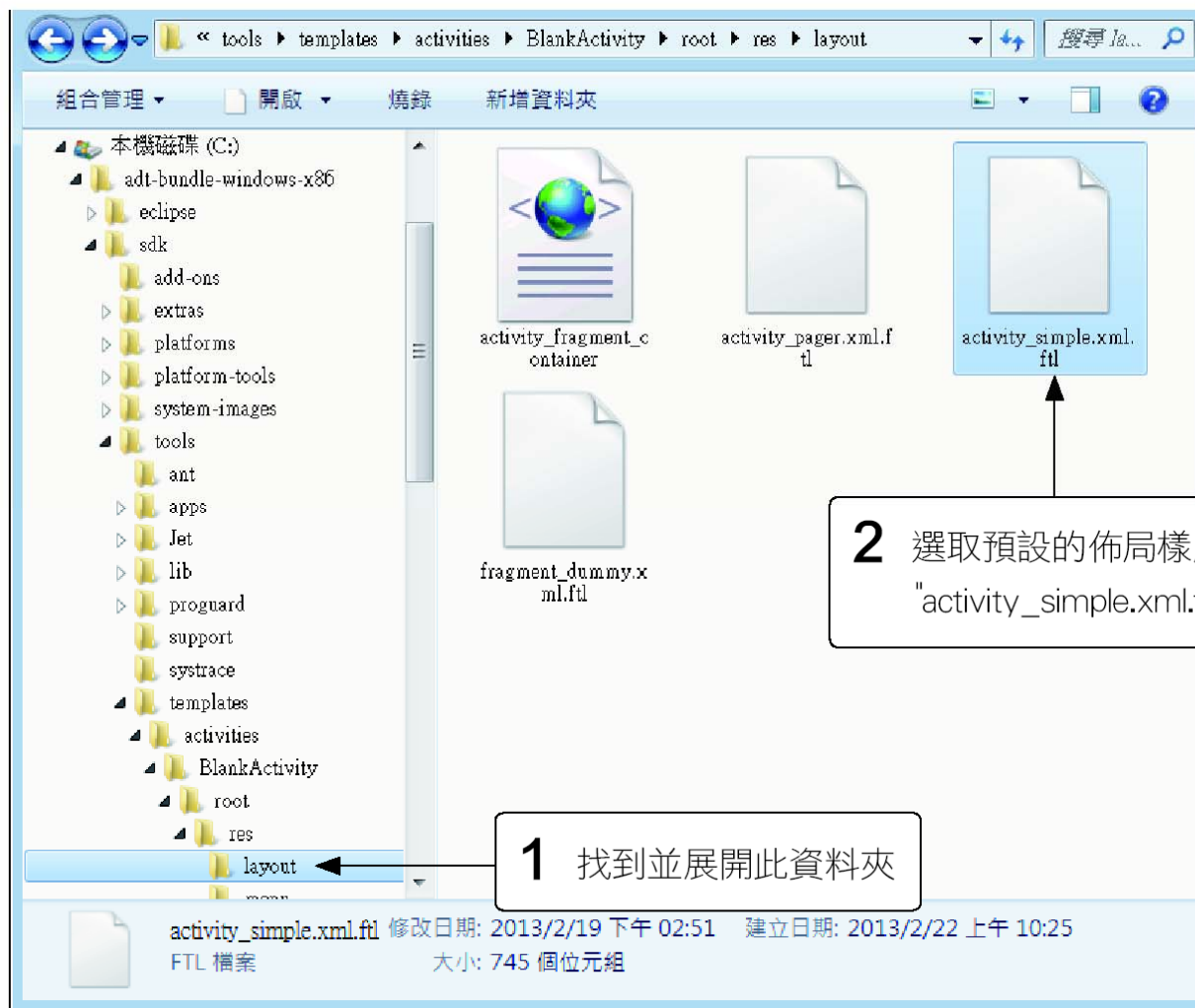
```
01 public void changeColor(View v){
02     // 取得亂數物件, 產生3個亂數值(rgb值)
03     Random x=new Random();
04     int red=x.nextInt(256);           ← 取0~255之間的亂數
05     txvR.setText("紅:" + red);      ← 顯示亂數值
06     txvR.setTextColor(Color.rgb(red,0,0)); ← 將文字設為亂數顏(紅)色值
07
08     int green=x.nextInt(256);
09     txvG.setText("綠:" + green);
10     txvG.setTextColor(Color.rgb(0,green,0)); ← 將文字設為亂數顏(綠)色值
```

變色龍- 以亂數設定顏色屬性

```
11
12     int blue=x.nextInt(256);
13     txvB.setText("藍："+ blue);
14     txvB.setTextColor(Color.rgb(0,0,blue)); ← 將文字設為亂數顏(藍)色值
15
16     // 設定畫面最下方的空白 LinearLayout 之背景顏色
17     colorBlock.setBackgroundColor(Color.rgb(red, green, blue));
18 }
```

3-6 使用自訂佈局樣版取代 ADT 預設佈局樣版

step 1



2 選取預設的佈局樣版檔
"activity_simple.xml.ftl"

1 找到並展開此資料夾

3 按 **F2** 鍵

使用自訂佈局樣版取代 ADT 預設佈局樣版



使用自訂佈局樣版取代 ADT 預設佈局樣版

step 2

The screenshot shows two Windows Explorer windows. The top window displays the project's file structure: `tools > templates > activities > BlankActivity > root > res > layout`. The bottom window shows a CD-ROM drive (G:) containing a folder named `樣版` (Templates). A dashed arrow indicates the transfer of files from the CD-ROM to the project's layout folder.

1 開啟書附光碟上的 樣版 資料夾

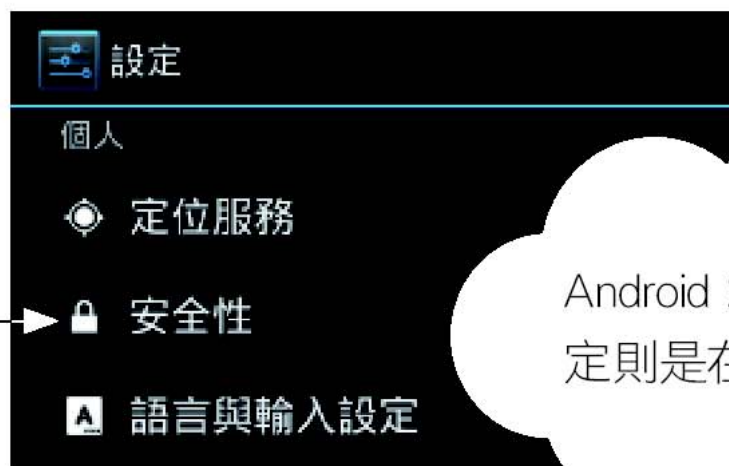
2 將光碟上的樣版檔複製到 ADT 的資料夾中

3-7 使用 Gmail 將程式寄給朋友測試

- 設定可以安裝非 Google Play 商店下載的程式
- 將程式寄給朋友安裝

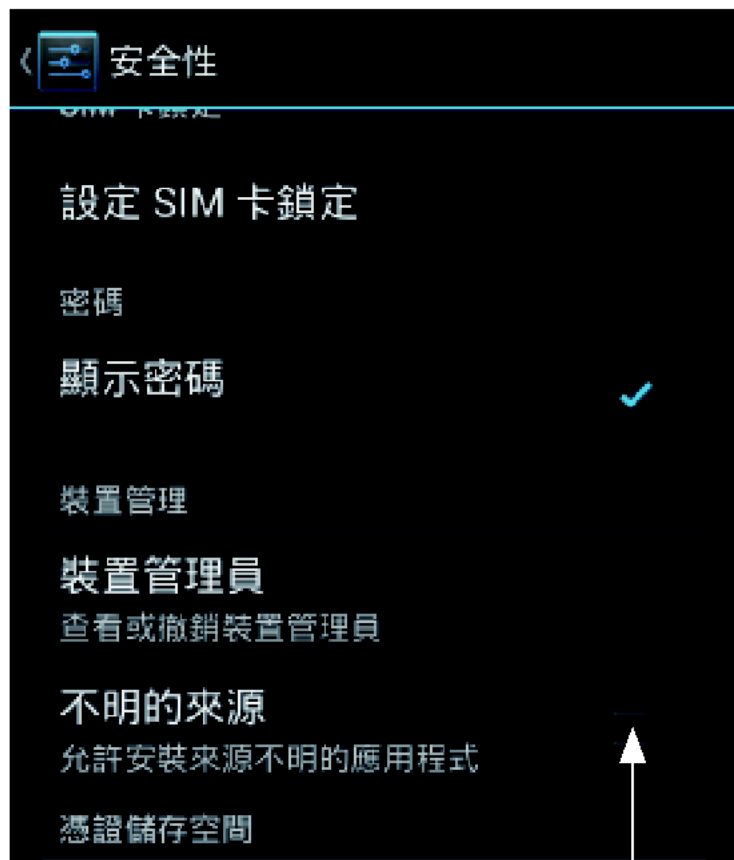
設定可以安裝非 Google Play 商店下載的程式

- 1 在設定項目清單中，選擇**安全性**項目

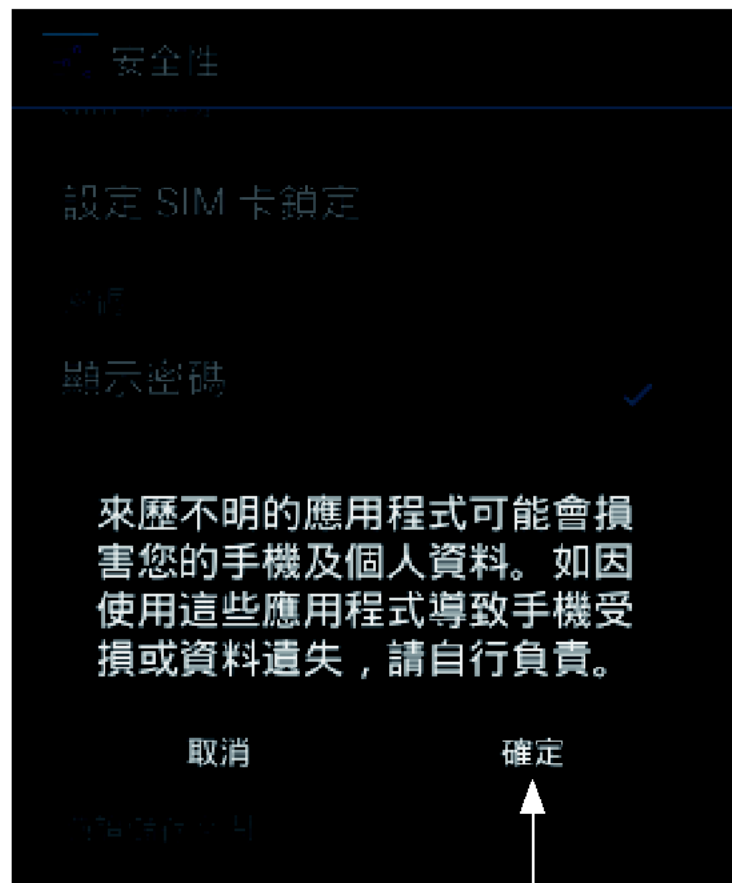


Android 2.X 手機的相關設定則是在**應用程式**項目下

設定可以安裝非 Google Play 商店下載的程式

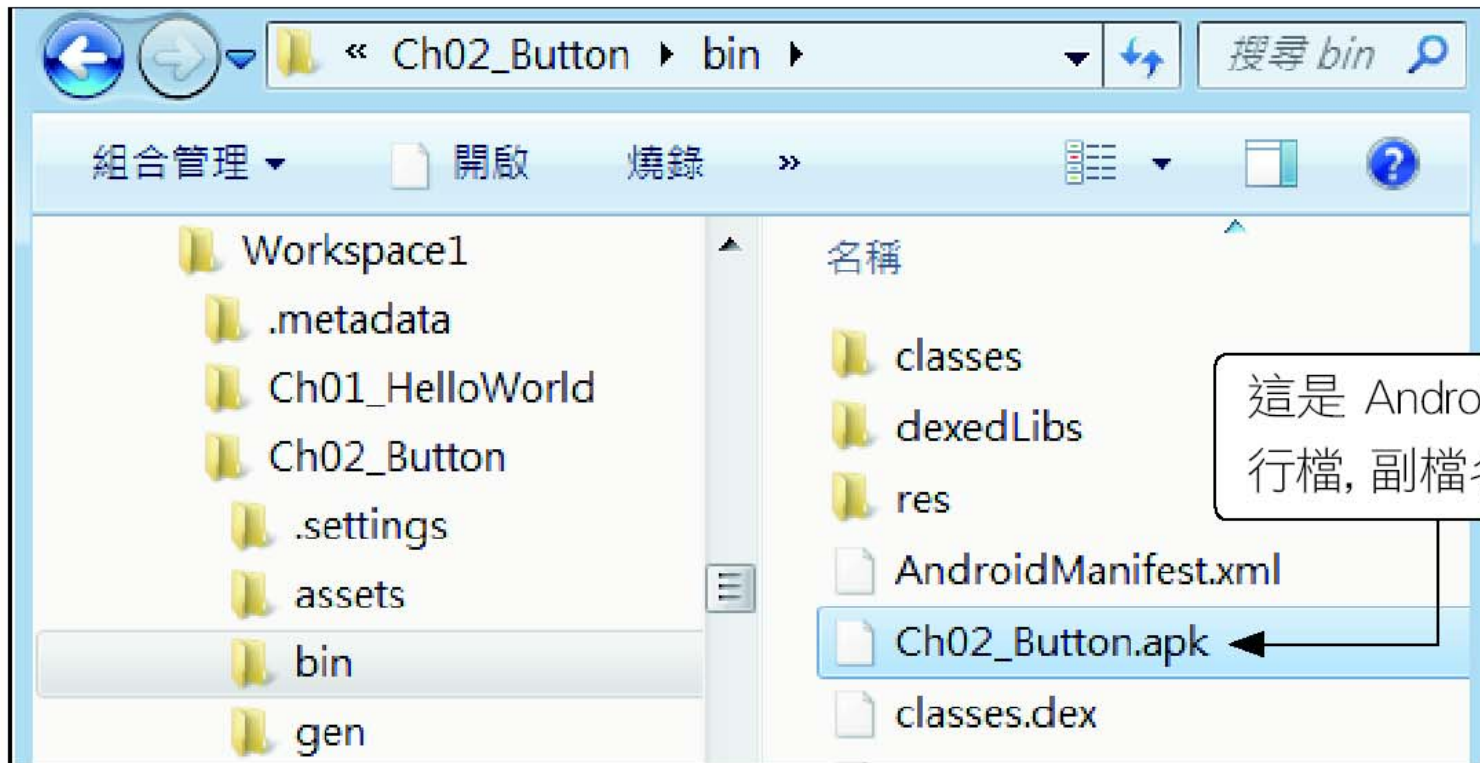


2 勾選此項



3 按確定鈕

將程式寄給朋友安裝

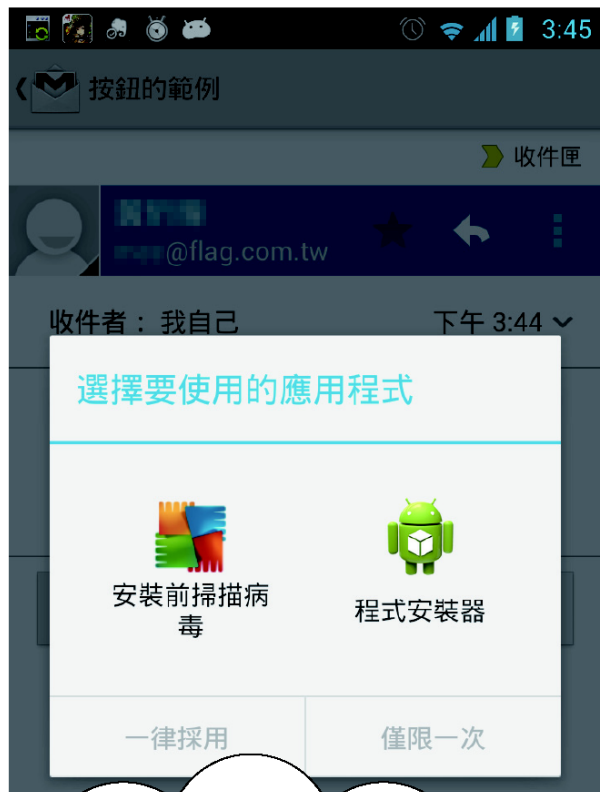


將程式寄給朋友安裝

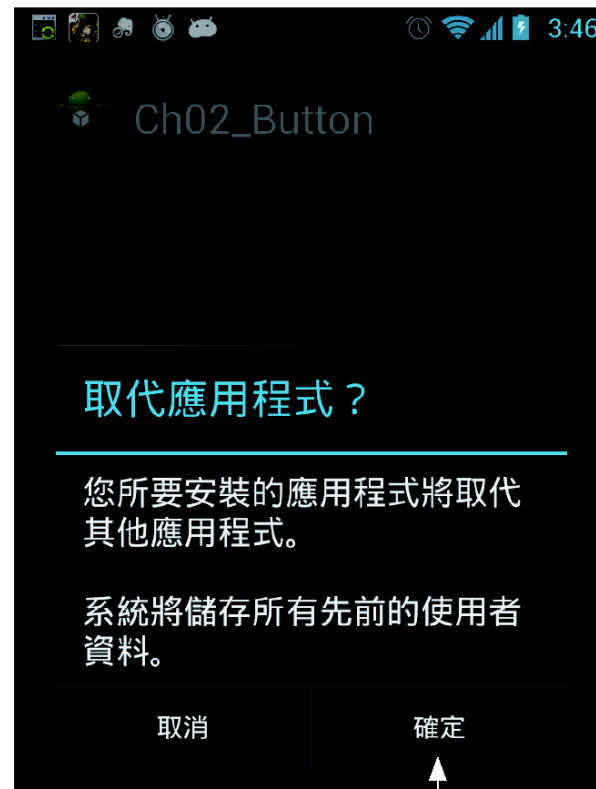


1 按此鈕安裝

將程式寄給朋友安裝



若手機有安裝防毒軟體等工具，則系統會詢問開啟檔案的方式，請選『程式安裝器』



- 2 按**確定**鈕 (如果已安裝過同一程式，例如先前用 USB 上傳，才會進入此畫面)

將程式寄給朋友安裝



3 按安裝鈕



4 按開啟鈕就會啟動 Android App