

# 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 7 章

# 即時訊息與交談窗

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

- 7-1 使用 Toast 顯示即時訊息
- 7-2 使用 Alert 交談窗
- 7-3 使用日期、時間交談窗

# 7-1 使用 Toast 顯示即時訊息

錄音功能已啟用

Toast 即時訊息

 Android 問卷調查		
你喜歡 Android 手機嗎？		
討厭	沒意見	喜歡

Alert 交談窗

# 使用 Toast 顯示即時訊息



1 連續點選任意的問題

2 即會依序顯示點選問題的答案, 每個答案顯示 2 秒後即自動消失, 然後顯示下一個

# 使用 Toast 顯示即時訊息

- Toast 類別
- Toast 訊息的取消顯示與更新顯示

# Toast 類別

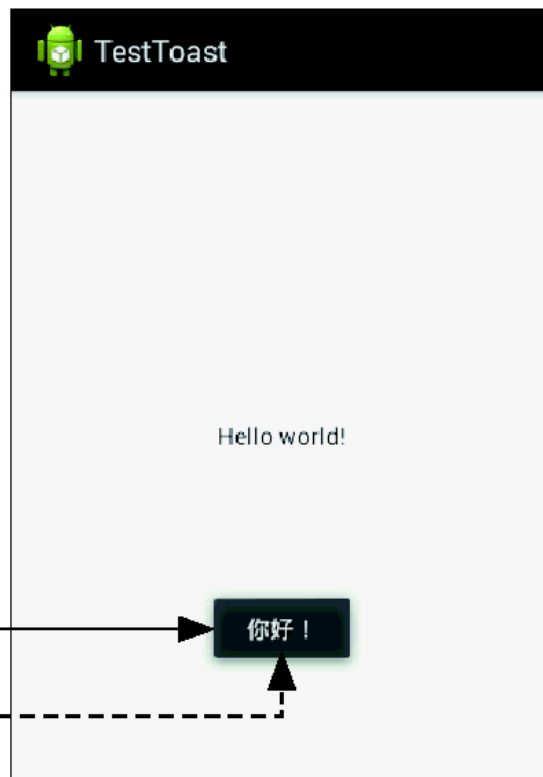
```
Toast tos = Toast.makeText(this, "你好!", Toast.LENGTH_SHORT);
```

第 1 個參數傳入代表 MainActivity 物件的 this, 表示要在 MainActivity 中顯示訊息

要顯示的訊息

tos.show(); ← 將訊息顯示出來

顯示的時間長度



# Toast 類別

```
Toast.makeText(this, "你好!", Toast.LENGTH_SHORT).show();
```

# 範例7-1：腦筋急轉彎-用 Toast 顯示答案



1 在使用者點選問題時觸發的 `onItemClick()` 方法中, 從答案陣列中取出答案字串

aA 陣列

答案1
答案2
....
⋮
⋮
⋮
⋮
⋮
⋮

2 建立 Toast 物件將答案顯示在畫面上



# 腦筋急轉彎-用 Toast 顯示答案

step 1

step 2

將 ListView 拉曳到佈局左上角 (預設會填滿畫面), 並設定 Id 屬性值為 "lv"



# 腦筋急轉彎-用 Toast 顯示答案

## step 3

```
01 public class MainActivity extends Activity
02         implements OnClickListener {
03
04     String[] queArr = {"什麼門永遠關不上？","什麼東西沒人愛吃？",
05                       "什麼瓜不能吃？","什麼布切不斷？",
06                       "什麼鼠最愛乾淨？","偷什麼不犯法？"};
07     String[] ansArr = { "球門", "虧",
08                        "傻瓜","瀑布",
09                        "環保署","偷笑" };
10
```

建立問題陣列

← 建立答案陣列

# 腦筋急轉彎-用 Toast 顯示答案

```
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15
16         ArrayAdapter<String> adapter = new ArrayAdapter<String>(
17             this,
18             android.R.layout.simple_list_item_1,
19             queArr);
20
21         ListView lv = (ListView) findViewById(R.id.lv);
```

建立供 ListView 使用的 ArrayAdapter 物件

使用內建的佈局資源

以 queArr 陣列當資料來源

取得 ListView

Next

# 腦筋急轉彎-用 Toast 顯示答案

```
23         lv.setAdapter(adapter);           ← 設定 ListView 使用的 Adapter
24         lv.setOnItemClickListener(this); ← 設定 ListView 項目
25     }                                       被按時的事件監聽器
26
27     @Override
28     public void onItemClick(AdapterView<?> a, View v,
29                             int pos, long id) {
30         Toast.makeText(this,
31             "答案：" + ansArr[pos], ← 由 ansArr 陣列取得答案
32             Toast.LENGTH_SHORT).show();
33     }
34     ...
```

# 腦筋急轉彎-用 Toast 顯示答案

## step 4

按下題目, 就會出現對應的答案



Toast 訊息約 2 秒後會消失 (程式中設定 Toast.LENGTH\_SHORT)

# 範例7-2：即時顯示答案的腦筋急轉彎

**1** 在 onCreate() 先建立好 Toast 物件，以便之後重複利用，即時顯示答案



**2** 在使用者選取題目後會執行的 onItemClick() 方法中，首先從答案陣列中取出答案字串

**aA 陣列**

答案1
答案2
...
...
...

**3** 利用事先建好的 Toast 物件立即顯示答案，不用等待之前顯示的答案消失

# 即時顯示答案的腦筋急轉彎

## step 1

## step 2

```
01 public class MainActivity extends Activity
...
07     // 建立答案陣列
08     String[] ansArr = { "球門", "虧",
09                         "傻瓜", "瀑布",
10                         "環保署", "偷笑"};
11     Toast tos; ← 宣告 Toast 物件
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
```

# 即時顯示答案的腦筋急轉彎

```
...
27     tos = Toast.makeText(this, "", Toast.LENGTH_SHORT); ← 建立 Toast 物件
28 }
29
30 @Override
31 public void onItemClick(AdapterView<?> a, View v,
32     int pos, long id) {
33     tos.setText("答案：" + ansArr[pos]); ← 變更 Toast 物件的文字內容
34     tos.show(); ← 立即重新顯示
35 }
36 }
```



# 7-2 使用 Alert 交談窗

 **Android 問卷調查**

---

你喜歡 Android 手機嗎？

討厭	沒意見	喜歡
----	-----	----

AlertDialog

**2013/1/18週五**

2012	12月	17
2013	1月	18
2014	2月	19

完成

DatePickerDialog

**設定時間**

14	05
15	: 06
16	07

完成

TimePickerDialog

# 使用 Alert 交談窗

- AlertDialog 類別
- AlertDialog.Builder：設定與建立 Alert 交談窗
- setCancelable()：設定可按返回鍵關閉交談窗
- show()：建立並顯示交談窗
- 建立 Alert 交談窗的簡潔寫法
- 在交談窗中加入按鈕

# AlertDialog 類別



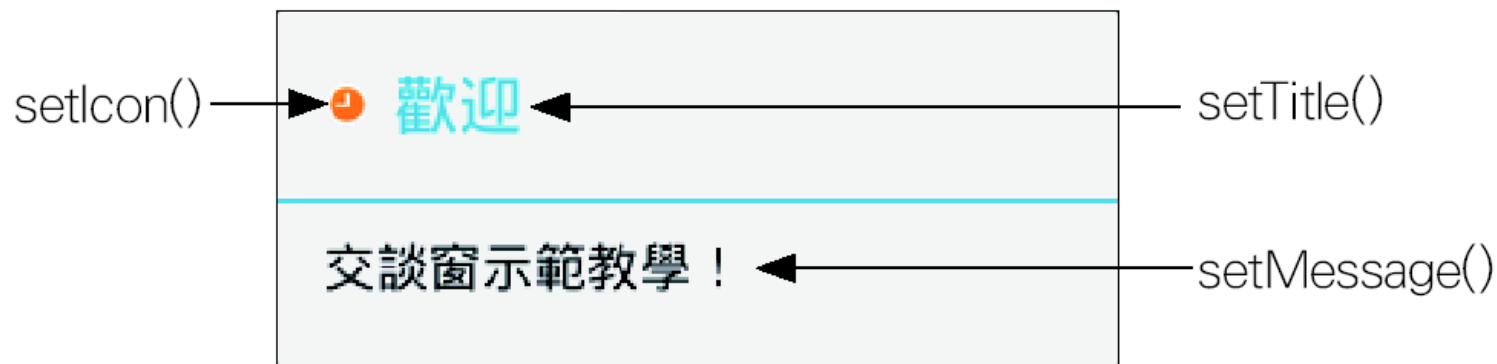
**標題:**可包含文字及圖示  
(也可都沒有)

**內容:**一段文字訊息、或  
一個可選擇的清單方塊

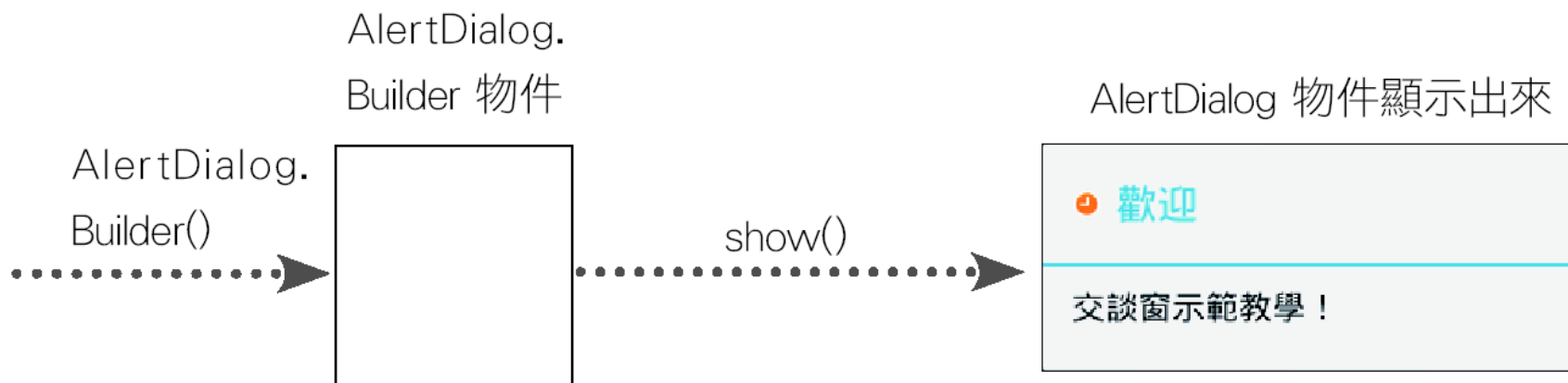
**按鈕:**可包含 0~3 個按鈕  
(也可都沒有)

# AlertDialog.Builder：設定與建立 Alert 交談窗

```
AlertDialog.Builder bdr = new AlertDialog.Builder(this);  
bdr.setMessage("交談窗示範教學！");  
bdr.setTitle("歡迎");  
bdr.setIcon(android.R.drawable.presence_away);
```



# show() : 建立並顯示交談窗

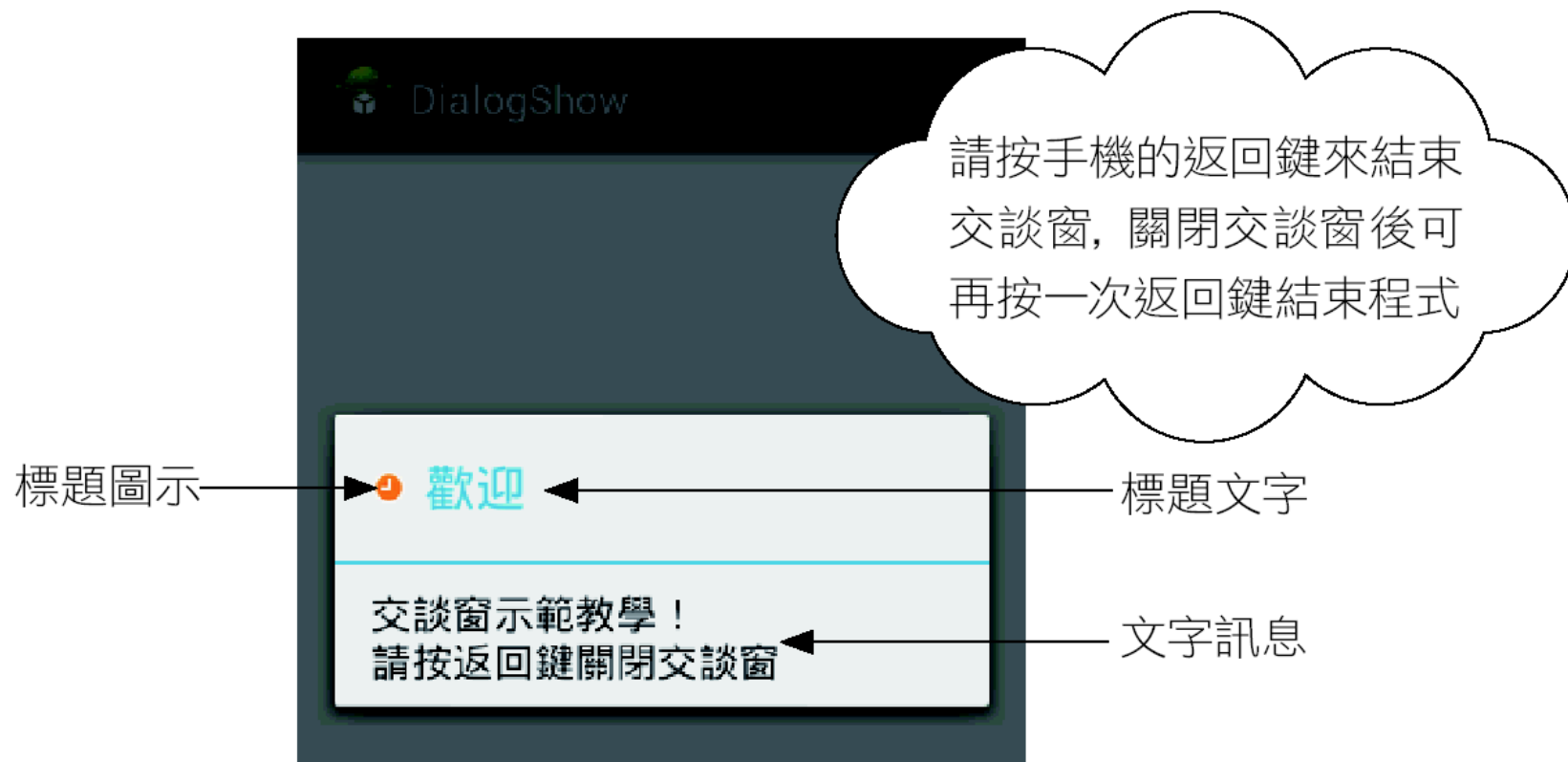


# 建立 Alert 交談窗的簡潔寫法

```
AlertDialog.Builder bdr = new AlertDialog.Builder(this);  
bdr.setMessage("交談窗示範教學！")  
    .setTitle("歡迎");  
bdr.setIcon(android.R.drawable.presence_away)  
    .show;
```

```
new AlertDialog.Builder(this).  
    .setMessage("交談窗示範教學！")  
    .setTitle("歡迎");  
new AlertDialog.Builder(this).  
    .setIcon(android.R.drawable.presence_away)  
    .show();
```

# 範例7-3：顯示歡迎訊息的交談窗



# 顯示歡迎訊息的交談窗

step 1

step 2



修改預設的 TextView 元件

Text	請再按一次返回鍵結束程式
Text Size	20sp



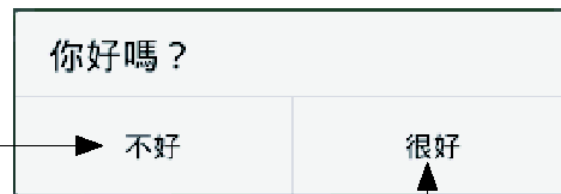
# 顯示歡迎訊息的交談窗

## step 3

```
01 protected void onCreate(Bundle savedInstanceState) {
02     super.onCreate(savedInstanceState);
03     setContentView(R.layout.activity_main);
04
05     AlertDialog.Builder bdr = new AlertDialog.Builder(this);
06     bdr.setMessage("交談窗示範教學！\n"    ← 加入文字訊息
07         + "請按返回鍵關閉交談窗");
08     bdr.setTitle("歡迎");                ← 加入標題
09     bdr.setIcon(android.R.drawable.presence_away); ← 加入圖示
10     bdr.setCancelable(true);          ← 允許按返回鍵關閉交談窗
11     bdr.show();
12 }
```

# 在交談窗中加入按鈕

```
new AlertDialog.Builder(this)
    .setMessage("你好嗎？")
    .setNegativeButton("不好", null)
    .setPositiveButton("很好", null);
```



```
public class MainActivity extends Activity
    implements DialogInterface.OnClickListener {
```

# 在交談窗中加入按鈕

```
public void onClick(DialogInterface dialog, int id) {  
    if(id == DialogInterface.BUTTON_POSITIVE) {  
        ... // 按下『是』按鈕時的處理  
    }  
    else if(id == DialogInterface.BUTTON_NEGATIVE) {  
        ... // 按下『否』按鈕時的處理  
    }  
}
```

# 在交談窗中加入按鈕

## 代表按鈕的常數

常數名稱	意義
BUTTON_NEGATIVE	按了代表『否』的按鈕
BUTTON_NEUTRAL	按了代表『中性』的按鈕
BUTTON_POSITIVE	按了代表『是』的按鈕

# 範例7-4：Android 問卷調查



交談窗中會顯示 3 個按鈕, 依序為 Negative、Neutral、及 Positive 鈕

按任一個按鈕  
.....➔



顯示使用者的選擇結果

# Android 問卷調查

step 1

step 2



**1** 修改預設的 TextView 元件：

Id	@+id/answer
Text	你對 Android 手機沒表示意見
Text Size	20dp

# Android 問卷調查

## step 3

```
01 package tw.com.flag.ch07_dialogask;
02
03 import android.os.Bundle;
04 import android.app.Activity;
05 import android.app.AlertDialog;
06 import android.content.DialogInterface;
07 import android.view.Menu;
08 import android.widget.TextView;
09
10 protected class MainActivity extends Activity
11     implements DialogInterface.OnClickListener { ← 實作監聽介面
12
13     TextView txv; ← 記錄預設的 TextView 元件
```

# Android 問卷調查

```
14
15     @Override
16     public void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_main);
19
20         txv = (TextView) findViewById(R.id.answer); ← 找出預設的 TextView
21         new AlertDialog.Builder(this) ← 建立 Builder 物件
22             .setMessage("你喜歡 Android 手機嗎?") ← 設定顯示訊息
23             .setCancelable(false) ← 禁用返回鍵關閉交談窗
24             .setIcon(android.R.drawable.ic_menu_edit) ← 採用內建的圖示
25             .setTitle("Android 問卷調查") ← 設定交談窗的標題
26             .setPositiveButton("喜歡", this) ← 加入肯定按鈕並監聽事件
27             .setNegativeButton("討厭", this) ← 加入否定按鈕並監聽事件
```

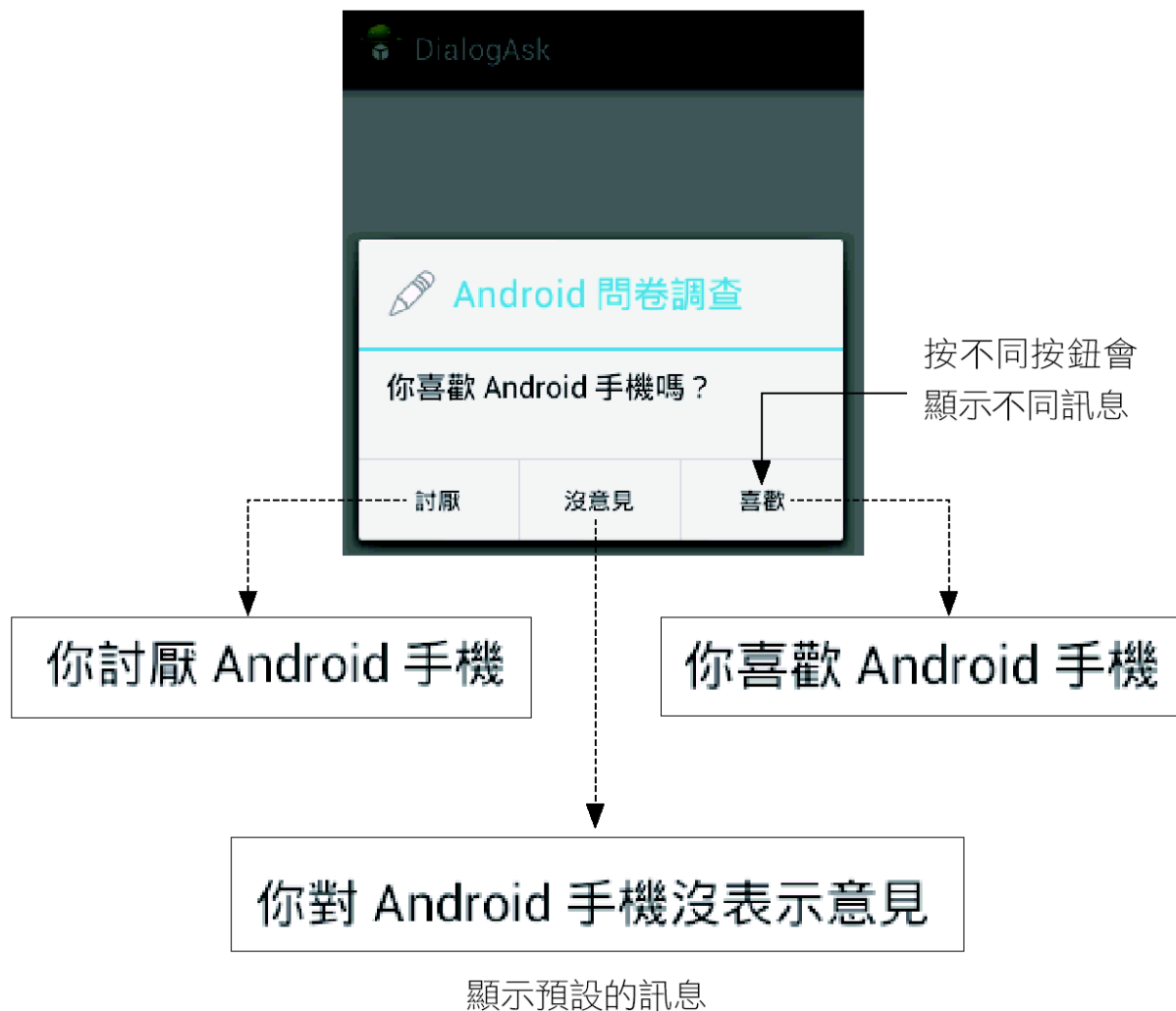


# Android 問卷調查

```
28     .setNeutralButton("沒意見", null) ← 不監聽中性按鈕
29     .show(); ← 顯示交談窗
30 }
31
32 @Override
33 public void onClick(DialogInterface dialog, int id) { ← 實作監聽介面定義的方法
34     if(id == DialogInterface.BUTTON_POSITIVE) { ← 如果按下肯定的『喜歡』
35         txv.setText("你喜歡 Android 手機");
36     } ← 如果按下否定的『討厭』
37     else if(id == DialogInterface.BUTTON_NEGATIVE) { ←
38         txv.setText("你討厭 Android 手機");
39     }
40 }
```

# Android 問卷調查

## step 4



## 7-3 使用日期、時間交談窗

- DatePickerDialog 與 TimePickerDialog 類別
- onDateSet() 與 onTimeSet()：取得選取的日期與時間

# DatePickerDialog 與 TimePickerDialog 類別

傳入 this 表示要在 MainActivity 顯示交談窗

由 MainActivity 監聽選取日期完成時的事件

```
new DatePickerDialog(this, this, 2013, 0, 18)  
.show();
```

預設選取的年份

預設選取的月份 (從 0 起算)

預設選取的日數



# DatePickerDialog 與 TimePickerDialog 類別

傳入 this 表示要在 MainActivity 顯示交談窗

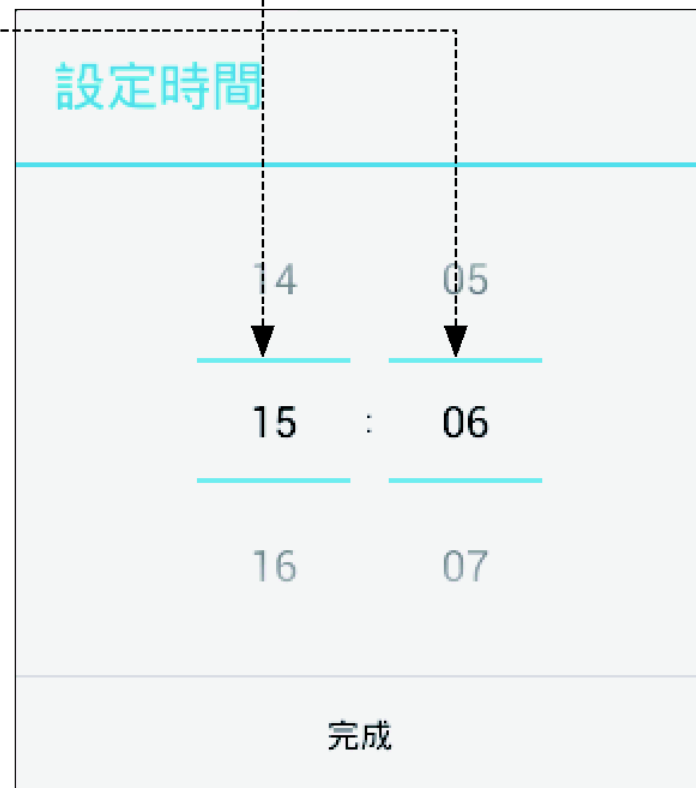
由 MainActivity 監聽選取時間完成時的事件

```
new TimePickerDialog(this, this, 15, 6, true)  
.show();
```

預設選取的時數

預設選取的分鐘數

使用 24 小時制



# DatePickerDialog 與 TimePickerDialog 類別

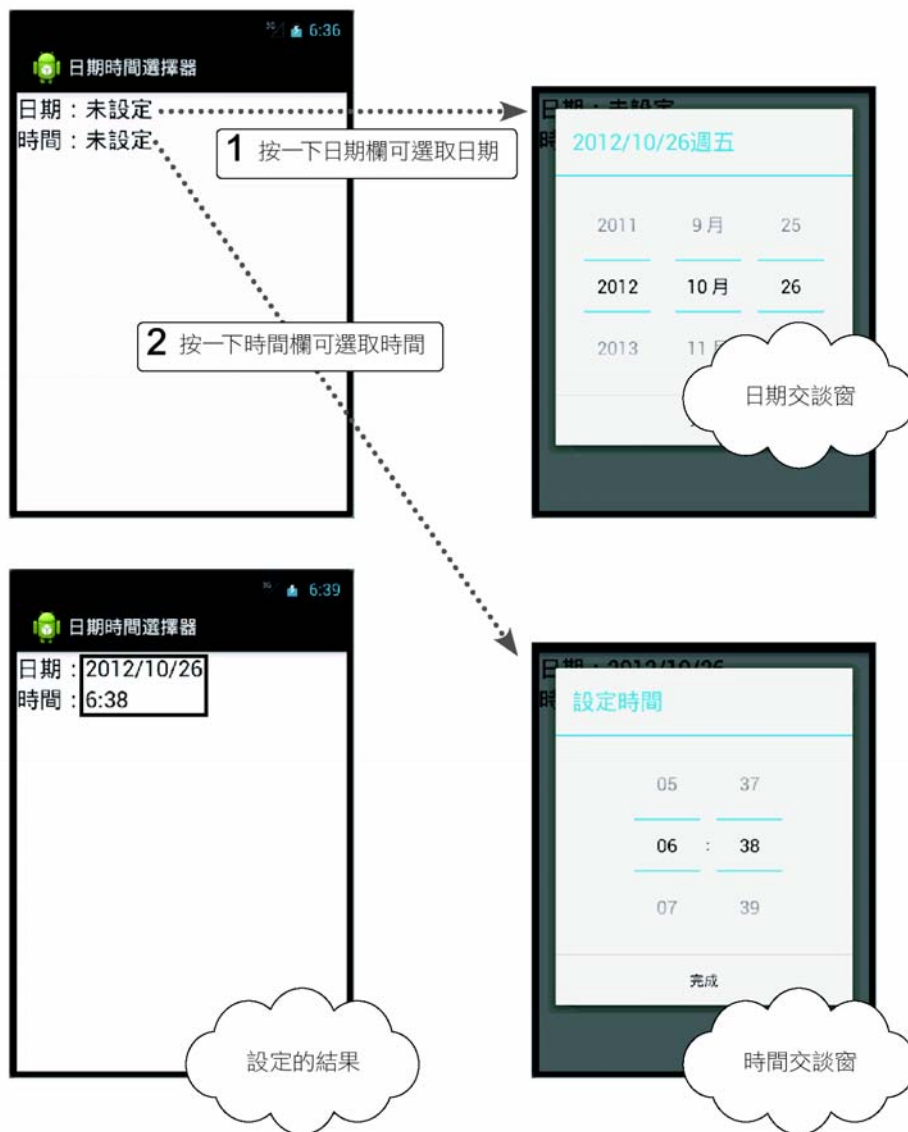


# onDateSet() 與 onTimeSet() : 取得 選取的日期與時間

```
public void onDateSet(DatePicker v, int y, int mon, int d) {  
    ...  
}
```

```
public void onTimeSet(TimePicker v, int h, int m) {  
    ...  
}
```

# 範例7-5：日期時間選擇器





# 日期時間選擇器

step 1

step 2



**1** 將原來的 Layout 換成垂直的 LinearLayout(Vertical)

**2** 加入 2 個 Large TextView



Id	@+id/textView2
Text	時間：未設定

Id	@+id/textView1
Text	日期：未設定

# 日期時間選擇器

## step 3

```
01 public class MainActivity extends Activity
02     implements OnClickListener, ←出現紅色波浪線時, 要 import android.
                                view.View.OnClickListener
03         DatePickerDialog.OnDateSetListener, ←
                                實作監聽日期交談窗事件的介面
04         TimePickerDialog.OnTimeSetListener { ←
                                實作監聽時間交談窗事件的介面
05
06     Calendar c = Calendar.getInstance(); ← 建立日曆物件
07     TextView txDate; ← 記錄日期文字的元件
08     TextView txTime; ← 記錄時間文字的元件
```

# 日期時間選擇器

## step 4

```
01     protected void onCreate(Bundle savedInstanceState) {
02         super.onCreate(savedInstanceState);
03         setContentView(R.layout.activity_main);
04
05         txDate = (TextView) findViewById(R.id.textView1); ←取得用來顯示日期的 TextView
06         txTime = (TextView) findViewById(R.id.textView2); ←取得用來顯示時間的 TextView
07
08         txDate.setOnClickListener(this); ←設定按下日期文字時的監聽物件
09         txTime.setOnClickListener(this); ←設定按下時間文字時的監聽物件
10     }
```

# 日期時間選擇器

## step 5

```
01     public void onClick(View v) {
02         if(v == txDate) { ← 按的是日期文字
03             //建立日期選擇交談窗, 並傳入設定完成時的監聽物件
04             new DatePickerDialog(this, this, ← 由 MainActivity
05                 c.get(Calendar.YEAR), ← 由 Calendar 物件取得目前的西元年
06                 c.get(Calendar.MONTH), ← 取得目前月 (由 0 算起)
07                 c.get(Calendar.DAY_OF_MONTH)) ← 取得目前日
```

Next

# 日期時間選擇器

```
08         .show(); ← 顯示交談窗
09     }
10     else if(v == txTime) { ← 按的是時間文字
11         //建立時間選擇交談窗, 並傳入設定完成時的監聽物件
12         new TimePickerDialog(this, this, ← 由 MainActivity 監聽事件
13             c.get(Calendar.HOUR_OF_DAY), ← 取得目前的時(24小時制)
14             c.get(Calendar.MINUTE), ← 取得目前的分
15             true) ← 使用 24 小時制
16         .show(); ← 顯示交談窗
17     }
18 }
```

# 日期時間選擇器

## step 6

```
01     public void onDateSet(DatePicker v, int y, int m, int d) { ←┐
                                           選定日期後的處理方法
02         txDate.setText("日期：" + y + "/" + (m+1) + "/" + d); ←┐
03     }                                     將選定日期顯示在螢幕上
04
05     @Override                             選定時間後的處理方法
06     public void onTimeSet(TimePicker v, int h, int m) { ←┐
07         txTime.setText("時間：" + h + ":" + m); ←┐
08     }                                     將選定的時間顯示在螢幕上
```