

# 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 15 章

# SQLite 資料庫

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

- 15-1 認識 SQLite 資料庫
- 15-2 查詢資料及使用 Cursor 物件
- 15-3 熱線通訊家

# 15-1 認識 SQLite 資料庫

- 資料庫、資料表、資料欄位
- 使用 CREATE TABLE 敘述建立資料表
- 使用 openOrCreateDatabase() 建立資料庫
- 用 execSQL() 方法執行 "CREATE TABLE" 敘述
- 用 insert() 方法及 ContentValues 物件新增資料

# 資料庫、資料表、資料欄位

欄位 (Column) 名稱 →

欄位 ↓

name	phone	email
孫小小	(02)23963257	small@flag.com.tw
盧拉拉	(02)23211271	lala@flag.com.tw
陳章章	(02)23975215	chacha@flag.com.tw

← 記錄 (Record)

# 使用 CREATE TABLE 敘述建立資料表

```
CREATE TABLE 資料表名稱 (欄位名稱 資料型別 PRIMARY KEY ,  
                           欄位名稱 資料型別,  ↖ 主索引鍵  
                           ...)  
                           ↖ 可有多組 "欄位名稱 資料型別"
```

```
CREATE TABLE customers (name VARCHAR(32),  
                          phone VARCHAR(16),  
                          email VARCHAR(32))
```

name	phone	email
孫小小	(02)23963257	small@flag.com.tw
盧拉拉	(02)23211271	lala@flag.com.tw
陳章章	(02)23975215	chacha@flag.com.tw

# 使用 openOrCreateDatabase() 建立 資料庫

```
openOrCreateDatabase("customer",  
                    Context.MODE_PRIVATE,  
                    null);
```

← 資料庫名稱  
← 建立資料庫檔的模式  
← 傳回查詢結果的類別

# 用 `execSQL()` 方法執行 "CREATE TABLE" 敘述

```
SQLiteDatabase db = openOrCreateDatabase(...); ← 取得資料庫物件
String sql = "CREATE TABLE test " + ← 建立 "test" 資料表的 SQL 語法字串
            "(name VARCHAR(32), " +
            "phone VARCHAR(32), " +
            "email VARCHAR(32))";
db.execSQL(sql); ← 執行上面的 CREATE TABLE 敘述
```

# 用 insert() 方法及 ContentValues 物件新增資料

```
ContentValues cv = new ContentValues(3); ← 建立含 3 個欄位的  
ContentValues 物件  
cv.put("name", "孫小小"); ← name 欄位為 "孫小小"  
cv.put("phone", "(02)23963257"); ← phone 欄位為 "(02)23963257"  
cv.put("email", "small@flag.com.tw"); ← email 欄位為 "small@flag.com.tw"  
db.insert("customers", null, cv) ← 將上列內容寫入 customers 資料表,  
新增 1 筆記錄
```



孫小小	(02)23963257	small@flag.com.tw
-----	--------------	-------------------

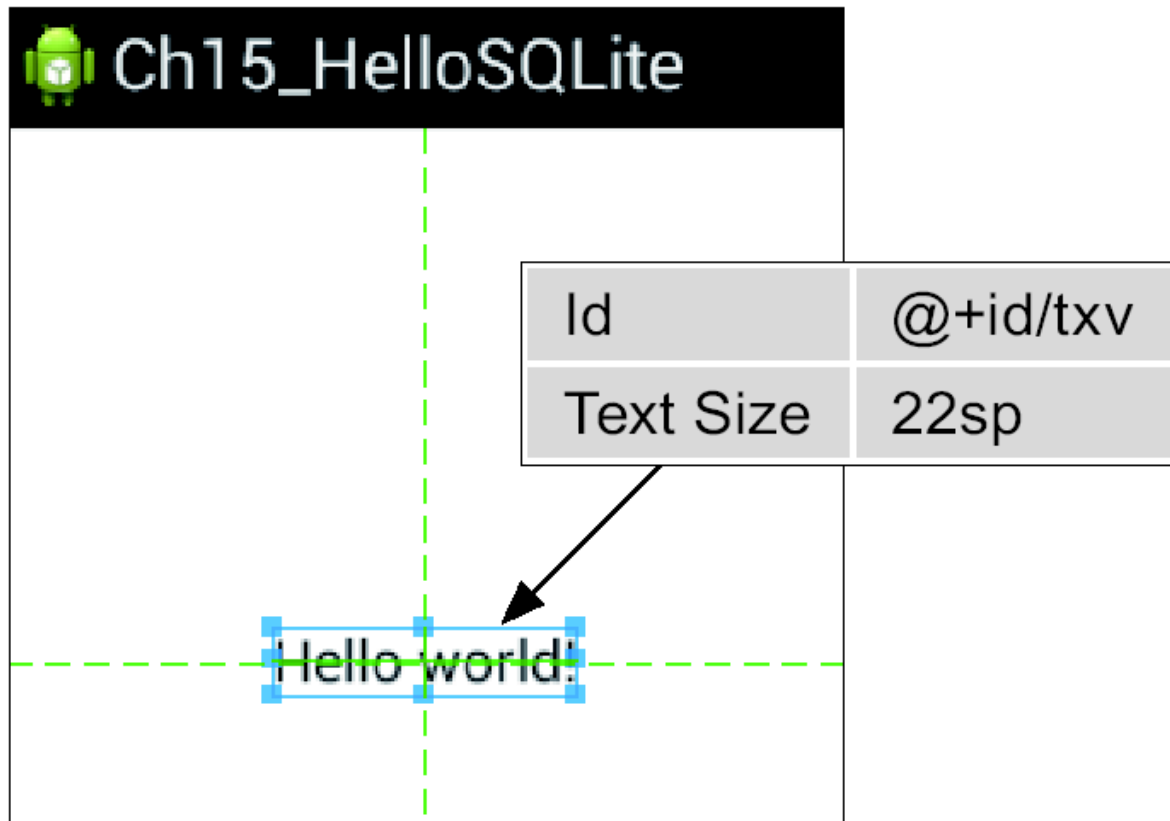
customers 資料表



# 範例15-1：建立資料庫及資料表

step 1

step 2



# 建立資料庫及資料表

## step 3

```
01 public class MainActivity extends Activity {
02     static final String db_name="testDB";    ← 資料庫名稱
03     static final String tb_name="test";     ← 資料表名稱
04     SQLiteDatabase db;                      ← 資料庫物件
05     @Override
06     protected void onCreate(Bundle savedInstanceState) {
07         super.onCreate(savedInstanceState);
08         setContentView(R.layout.activity_main);
09
```

# 建立資料庫及資料表

```
10 // 開啟或建立資料庫
11 db = openOrCreateDatabase(db_name,
                             Context.MODE_PRIVATE, null);
12
13 String createTable="CREATE TABLE IF NOT EXISTS " +
14     tb_name +           ← 資料表名稱
15     "(name VARCHAR(32), " + ← 姓名欄位
16     "phone VARCHAR(16), " + ← 電話欄位
17     "email VARCHAR(64))"; ← Email欄位
18 db.execSQL(createTable); ← 建立資料表
```

Next

# 建立資料庫及資料表

```
19
20     // 呼叫自訂的 addData() 方法寫入 2 筆資料
21     addData("Flag Publishing Co.", "02-23963257", "service@flag.
        com.tw");
22     addData("PCDIY Magazine", "02-23214335", "service@pcdiy.com.tw");
23
24     TextView txv=(TextView) findViewById(R.id.txv);           取得及顯
25     txv.setText("資料庫檔路徑: "+db.getPath()+"\n" + ←示資料庫
        資訊
26                 "資料庫分頁大小: "+db.getPageSize() + " Bytes\n" +
27                 "資料量上限: "+db.getMaximumSize() + " Bytes\n");
28
29     db.close();           ← 關閉資料庫
```

# 建立資料庫及資料表

```
30     }
31
32     private void addData(String name, String phone, String email) {
33         ContentValues cv=new ContentValues(3); ←
34
35         cv.put("name", name);
36         cv.put("phone", phone);
37         cv.put("email", email);
38
39         db.insert(tb_name, null, cv); ← 將資料加到資料表
40     }
41
42     ...
```

建立含3個資料項目的物件

← 將資料加到資料表

# 建立資料庫及資料表

## step 4



Ch15\_HelloSQLite

資料庫檔路徑: /data/data/  
tw.com.flag.ch15\_hellosqlite/  
databases/testDB

資料庫分頁大小: 4096 Bytes

資料庫大小上限:  
4398046507008 Bytes

← 資料庫檔在手機 (模擬器) 上完整的路徑

← 視手機規格, 分頁檔 (SQLite 配置儲存空間的基本單位) 大小不一定相同

## 15-2 查詢資料及使用 Cursor 物件

- 使用 SELECT 敘述進行資料查詢
- 使用 Cursor 物件取得查詢結果
- 使用 Cursor 物件的 getXXX() 方法讀取資料

# 使用 **SELECT** 敘述進行資料查詢

```
SELECT 欄位名稱 FROM 資料表名稱
```

```
SELECT * FROM test ← 直接用 * 代表『所有欄位』
```

```
SELECT name,phone,email FROM test
```



# 使用 Cursor 物件取得查詢結果

```
rawQuery("SELECT * FROM test", null); ← 傳回 test 資料表的所有記錄
```

moveToFirst()	移到第一筆	moveToLast()	移到最後一筆
moveToNext()	移到下一筆	moveToPrevious()	移到前一筆

```
Cursor cur=db.rawQuery("SELECT * FROM test"); ← 執行查詢  
  
if( cur.moveToFirst() ) ← 查詢結果中有資料才繼續  
do { ← 利用 do/while 迴圈逐筆讀取  
    ... // 讀取『指標』所指的資料  
} while (cur.moveToNext()); ← 若還有下一筆，就繼續 do/while 迴圈
```

# 使用 Cursor 物件的 getXXX() 方法讀取資料

getDouble (欄位索引)	←	讀取 double	資料
getFloat (欄位索引)	←	讀取 float	資料
getInt (欄位索引)	←	讀取 int	資料
getLong (欄位索引)	←	讀取 long	資料
getShort (欄位索引)	←	讀取 short	資料
getString (欄位索引)	←	讀取字串	

```
Cursor c = db.rawQuery("SELECT * FROM test");  
...  
c.getString(0); ← 讀 "name" 欄位的值  
c.getString(1); ← 讀 "phone" 欄位的值  
c.getString(2); ← 讀 "email" 欄位的值
```

# 使用 **Cursor** 物件的 **getXXX()** 方法讀取資料

```
c.getString(getColumnIndex("name"));  
c.getString(getColumnIndex("phone"));  
c.getString(getColumnIndex("email"));
```

# 範例15-2：使用 Cursor 物件讀取查詢結果

顯示查詢到的資料筆數

```
Ch15_HelloCursor
總共有 2筆資料
-----
name:Flag Publishing Co.
phone:02-23963257
email:service@flag.com.tw
-----
name:PCDIY Magazine
phone:02-23214335
email:service@pcdiy.com.tw
-----
```

逐筆列出查詢結果

# 使用 Cursor 物件讀取查詢結果

## step 1

## step 2

```
01 protected void onCreate(Bundle savedInstanceState) {
02     super.onCreate(savedInstanceState);
03     setContentView(R.layout.activity_main);
04
05     // 開啟或建立資料庫
06     db = openOrCreateDatabase(db_name, Context.MODE_PRIVATE, null);
07
08     String createTable="CREATE TABLE IF NOT EXISTS " +
09         tb_name +
```

# 使用 Cursor 物件讀取查詢結果

```
10         "(name VARCHAR(32), " +
11         "phone VARCHAR(16), " +
12         "email VARCHAR(64))";
13 db.execSQL(createTable);           ← 建立資料表
14
15 Cursor c=db.rawQuery("SELECT * FROM "+tb_name, null); ←
16                                     查詢tb_name資料表中的所有資料
17 if (c.getCount()==0){           ← 若無資料, 則立即新增 2筆資料
18     addData("Flag Publishing Co.,"02-23963257","service@flag.
19             com.tw");
20     addData("PCDIY Magazine","02-23214335","service@pcdiy.com.tw");
```

Next

# 使用 Cursor 物件讀取查詢結果

```
19     c=db.rawQuery("SELECT * FROM "+tb_name, null); ← 重新查詢
20 }
21
22 if (c.moveToFirst()) { ← 移到第 1 筆資料 (若有資料才繼續)
23     String str="總共有 "+c.getCount()+"筆資料\n";
24     str+="-----\n";
25
26     do{ ← 逐筆讀出資料, 並串接成訊息字串
27         str+="name:"+c.getString(0)+"\n"; ← name 欄位
28         str+="phone:"+c.getString(1)+"\n"; ← phone 欄位
29         str+="email:"+c.getString(2)+"\n"; ← email 欄位
```

# 使用 Cursor 物件讀取查詢結果

```
30         str+="-----\n";
31     } while(c.moveToNext());           ← 有下一筆就繼續迴圈
32
33     TextView txv=(TextView)findViewById(R.id.txv);
34     txv.setText(str);                 ← 顯示訊息字串
35 }
36
37 db.close();                          ← 關閉資料庫
38 }
```



# 使用 Cursor 物件讀取查詢結果

## step 3

Ch15\_HelloCursor

總共有 2筆資料 ← 以 getCount() 方法  
取得的資料筆數

-----

name:Flag Publishing Co. ← 用迴圈配合 getString()  
phone:02-23963257 讀到的查詢結果  
email:service@flag.com.tw

-----

name:PCDIY Magazine  
phone:02-23214335  
email:service@pcdiy.com.tw

-----

# 15-3 熱線通訊家

在 ListView 使用自訂的佈局顯示由 SQLite 資料庫查詢到的資料

可輸入連絡資訊，亦可修改、刪除資料

可立即撥電話或發送電子郵件

公司名稱	電話	電子郵件
旗標公司	02-23963257	service@flag.com.tw
旗訊公司	02-23214335	service@pcdiy.com.tw

姓名: 台北市

電話: 02-27708889

Email: service@mail.taipei.gov.tw

新增 修改 刪除 [撥電話] [發送電子郵件]

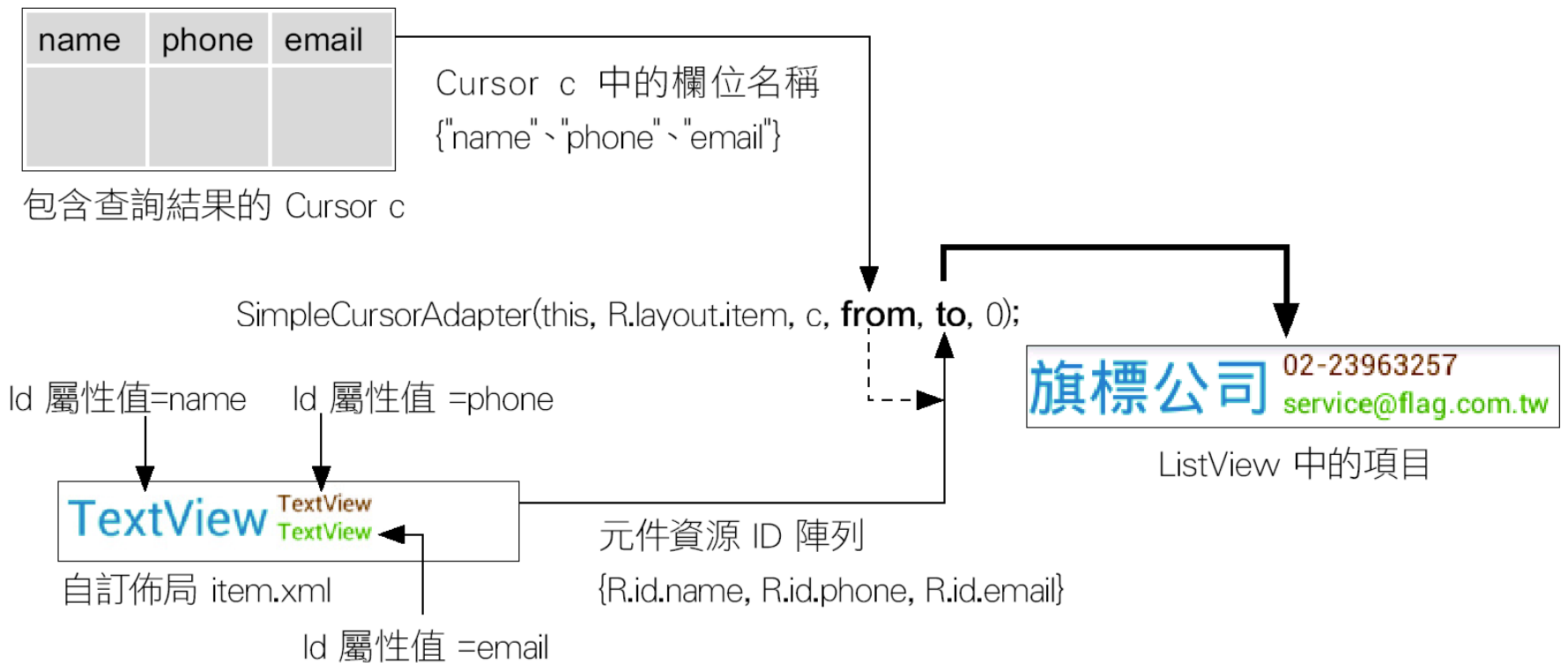
# 熱線通訊家

- 使用 SimpleCursorAdapter 自訂 ListView 版面
- 資料表的 '\_id' 欄位

# 使用 SimpleCursorAdapter 自訂 ListView 版面

```
SimpleCursorAdapter (MainActivity.this,  
    int layout,      ← 在 ListView 中顯示每個資料項目  
                    所用的佈局  
    Cursor c,        ← 以 Cursor 物件為資料來源  
    String[] from,   ← Cursor 中的欄位名稱陣列  
    int[] to,        ← layout (第 2 個參數) 中的元件 ID 陣列  
    0);              ← 指定 Adapter 運作模式, 本例不使用, 故設為 0
```

# 使用 SimpleCursorAdapter 自訂 ListView 版面



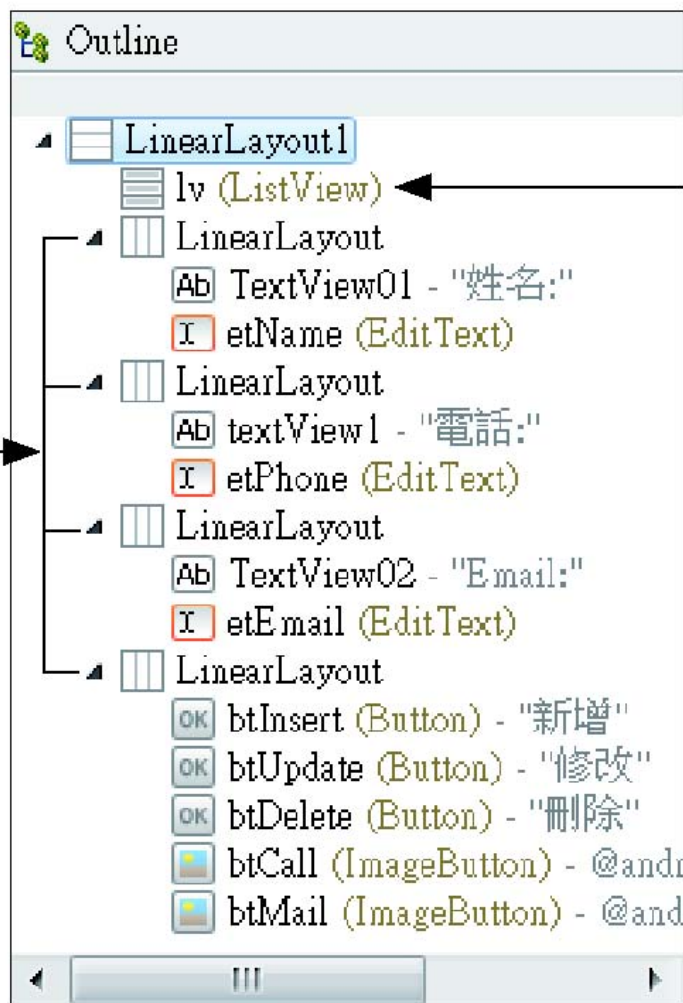
# 資料表的 '\_id' 欄位

```
CREATE TABLE hotlist (_id INTEGER PRIMARY KEY AUTOINCREMENT,  
                        name VARCHAR(32),  
                        phone VARCHAR(16),  
                        email VARCHAR(32))
```

# 範例15-3：熱線通訊家

## step 1

加入 4 個  
LinearLayout  
(Horizontal)



加入 ListView

# 熱線通訊家

設定 3 個 TextView 的 Text 屬性為如圖所示文字

ListView:

Id	@+id/lv
Height	0dp
Weight	1

EditText

Id	@+id/etName
Hint	請輸入姓名
Input Type	txtPersonName

EditText

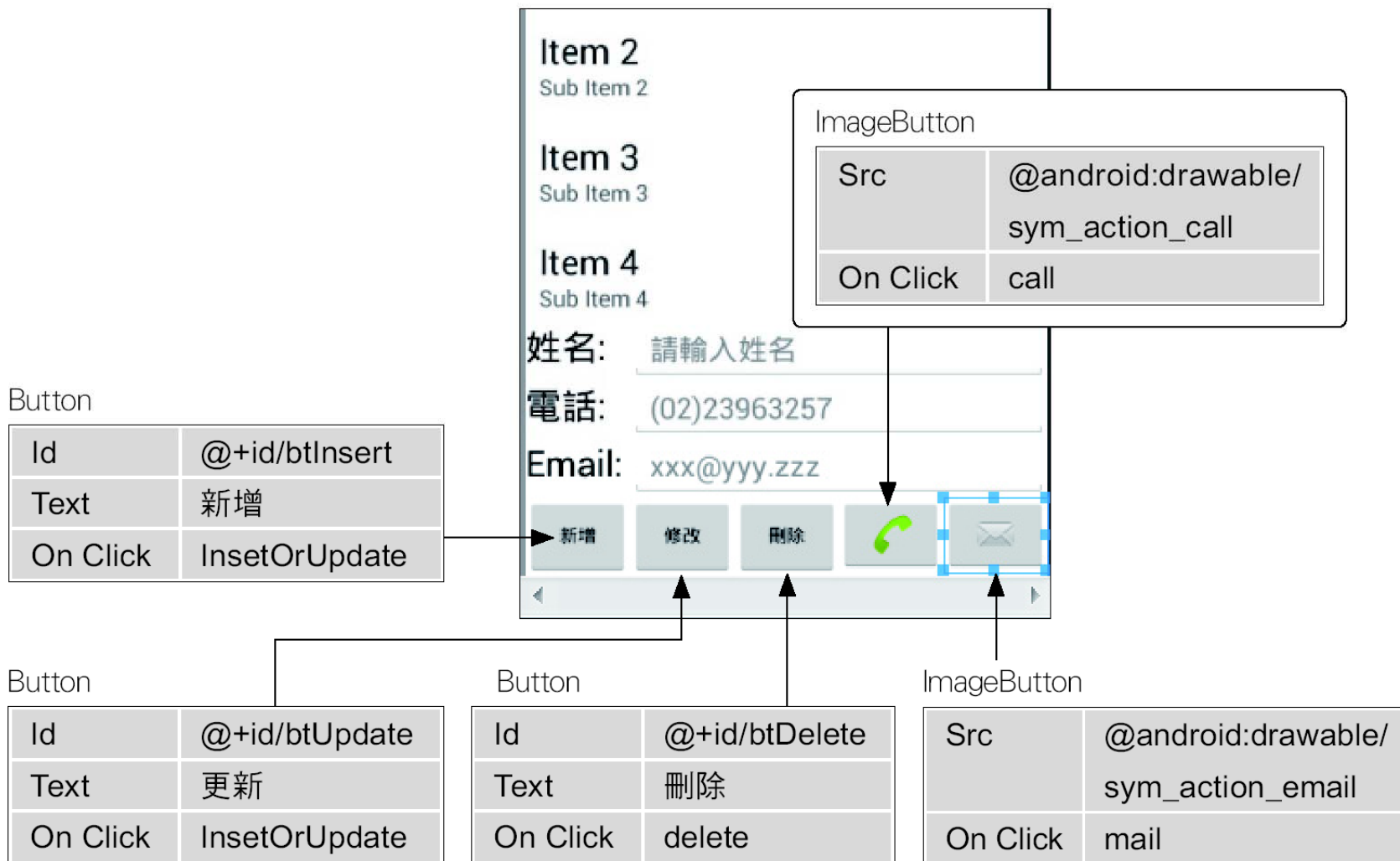
Id	@+id/etPhone
Hint	(02)-23963257
Input Type	phone

EditText

Id	@+id/etEmail
Hint	xxx@yyy.zzz
Input Type	txtEmailAddress

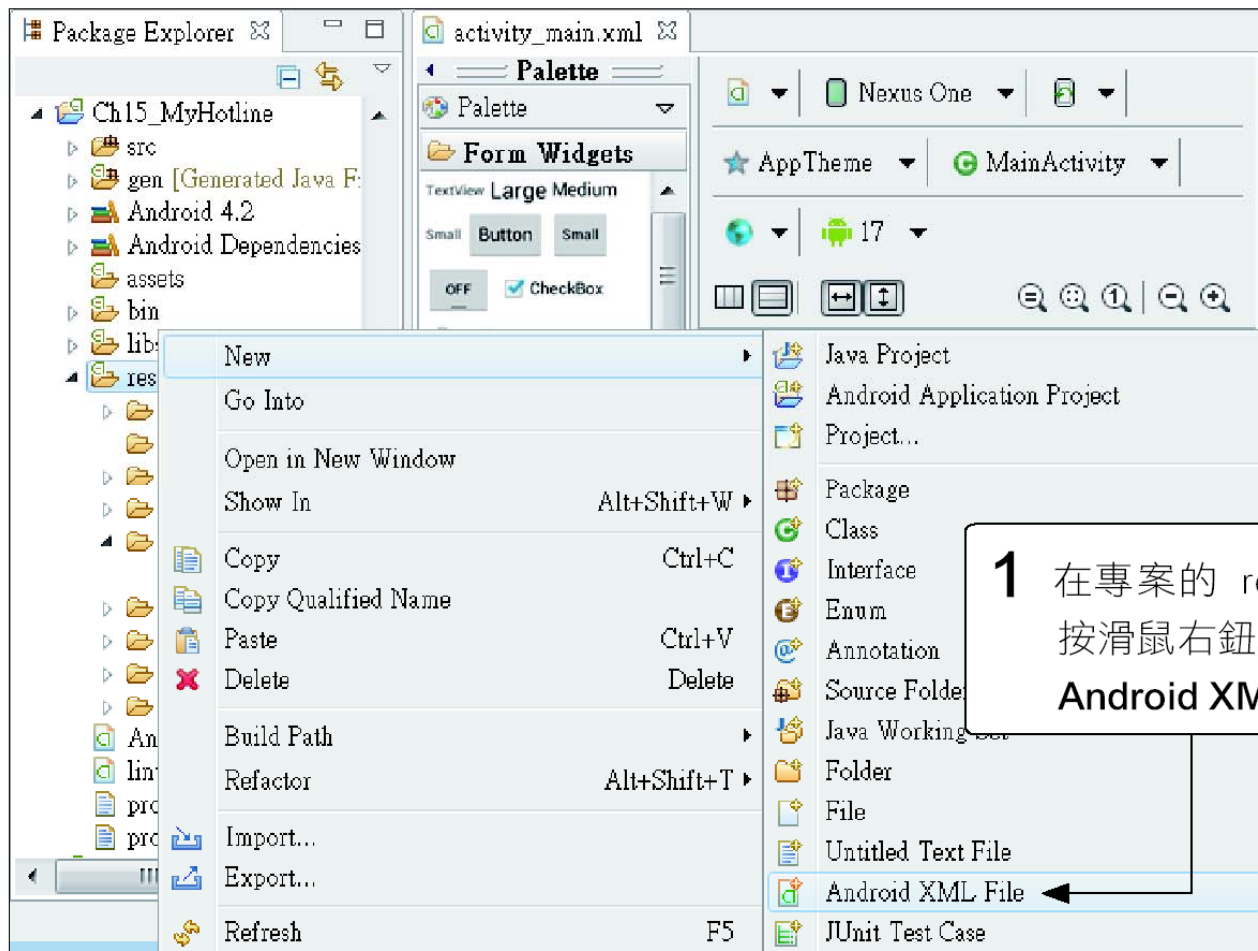


# 熱線通訊家



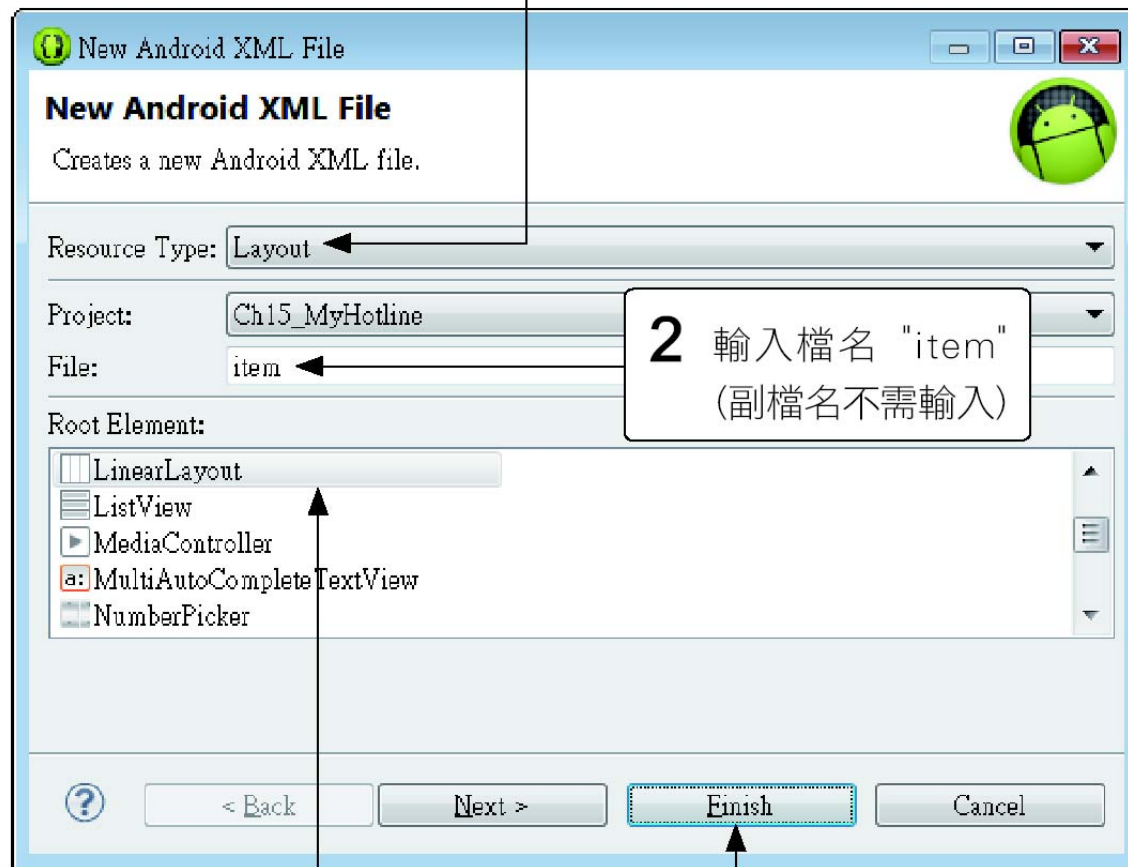
# 熱線通訊家

## step 2



# 熱線通訊家

預設就是選取 Layout 項目



2 輸入檔名 "item"  
(副檔名不需輸入)

3 選 LinearLayout

4 按 Finish 鈕

# 熱線通訊家



6 將 Margin 設為 5dp

# 熱線通訊家

8 加入 LinearLayout (Vertical)

7 加入 TextView

Id	@+id/name
Margins/Right	5dp
Text Size	30sp
Text Color	#07a

10 加入 TextView

Id	@+id/email
Text Size	14sp
Text Color	#480

9 加入 TextView

Id	@+id/phone
Text Size	14sp
Text Color	#840

Outline

- LinearLayout
  - name (TextView)
  - LinearLayout
    - phone (TextView)
    - email (TextView)

# 熱線通訊家

## step 3

```
01 public class MainActivity extends Activity
           implements AdapterView.OnItemClickListener{
02     static final String DB_NAME="HotlineDB";    ← 資料庫名稱
03     static final String TB_NAME="hotlist";     ← 資料表名稱
04     static final int MAX=8;                   ← 程式記錄的通訊資料筆數上限
05     static final String[] FROM =              ← 資料表欄位名稱字串陣列
06         new String[] {"name","phone","email"};
07     SQLiteDatabase db;
```

Next

# 熱線通訊家

```
08     Cursor cur; ← 存放查詢結果的 Cursor 物件
09     SimpleCursorAdapter adapter;
10     EditText etName,etPhone,etEmail;← 用於輸入姓名、電話、Email 的欄位
11     Button btInsert, btUpdate, btDelete; ← 新增、修改、刪除按鈕
12     ListView lv;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18
```

# 熱線通訊家

```
19     etName=(EditText) findViewById(R.id.etName);
20     etPhone=(EditText) findViewById(R.id.etPhone);
21     etEmail=(EditText) findViewById(R.id.etEmail);
22     btInsert =(Button) findViewById(R.id.btInsert);
23     btUpdate =(Button) findViewById(R.id.btInsert);
24     btDelete =(Button) findViewById(R.id.btInsert);
25
26     // 開啟或建立資料庫
27     db = openOrCreateDatabase(db_name, Context.MODE_PRIVATE, null);
28
29     // 建立資料表
30     String createTable="CREATE TABLE IF NOT EXISTS " + tb_name +
```

← 取得畫面  
上的元件



# 熱線通訊家

```
31         ┌─▶ "(_id INTEGER PRIMARY KEY AUTOINCREMENT, " +
32     索引欄位     "name VARCHAR(32), " +
33                 "phone VARCHAR(16), " +
34                 "email VARCHAR(64))";
35     db.execSQL(createTable);
36
37     cur=db.rawQuery("SELECT * FROM "+ tb_name, null); ◀─ 查詢資料
38
39     // 若是空的則寫入測試資料
40     if (cur.getCount() == 0) {
41         addData("旗標公司", "02-23963257", "service@flag.com.tw");
42         addData("旗訊公司", "02-23763257", "service@pcdiy.com.tw");
43     }
44
```

Next

# 熱線通訊家

```
45     // 建立 Adapter 物件
46     adapter=new SimpleCursorAdapter(this,
47         R.layout.item, cur, ← 自訂的 Layout, Cursor 物件
48         FROM, ← 欄位名稱陣列
49         new int[] {R.id.name,R.id.phone,R.id.email},
50         0); ← TextView 資源 ID 陣列
51
52     lv=(ListView) findViewById(R.id.lv);
53     lv.setAdapter(adapter); ← 設定 Adapter
54     lv.setOnItemClickListener(this); ← 設定按下事件的監聽器
55     requery(); ← 呼叫自訂方法重新查詢及設定按鈕狀態
56 }
```

# 熱線通訊家

## step 4

```
01 private void addData(String name, String phone, String email) {
02     ContentValues cv=new ContentValues(3);
03     cv.put(FROM[0], name);
04     cv.put(FROM[1], phone);
05     cv.put(FROM[2], email);
06
07     db.insert(TB_NAME, null, cv);
08 }
09
10 private void update(String name, String phone,
11                     String email, int id) {
12     ContentValues cv=new ContentValues(3);
13     cv.put(FROM[0], name);
14     cv.put(FROM[1], phone);
```

# 熱線通訊家

```
14     cv.put(FROM[2], email);
15
16     db.update(TB_NAME, cv, "_id="+id, null);    ← 更新 _id 所指的記錄
17 }
18
19 private void requery() {    ← 重新查詢的自訂方法
20     cur=db.rawQuery("SELECT * FROM "+tb_name, null);
21     adapter.changeCursor(cur);    ← 更改 Adapter 的 Cursor
22     if(cur.getCount()==max) ← 已達上限, 停用新增鈕
23         btInsert.setEnabled(false);
24     else
25         btInsert.setEnabled(true);
26     btUpdate.setEnabled(false); ← 停用更新鈕, 待使用者選取項目後再啟用
27     btDelete.setEnabled(false); ← 停用刪除鈕, 待使用者選取項目後再啟用
28 }
```

# 熱線通訊家

```
db.update(TB_NAME,           ← 要更新的資料表
          cv,                 ← 內含新資料的 ContentValues 物件
          "_id="+id,        ← 更新的條件式
          null);             ← 此參數本例不使用，故設為 null
```

# 熱線通訊家

## step 5

```
01 @Override
02 public void onItemClick(AdapterView<?> parent, View v,
                           int position, long id) {
03     cur.moveToPosition(position); ← 移動 Cursor 至使用者選取的項目
04     // 讀出姓名,電話,Email資料並顯示
05     etName.setText(cur.getString(
06                     cur.getColumnIndex(FROM[0])));
07     etPhone.setText(cur.getString(
08                     cur.getColumnIndex(FROM[1])));
09     etEmail.setText(cur.getString(
10                     cur.getColumnIndex(FROM[2])));
11
```

Next

# 熱線通訊家

```
12     btUpdate.setEnabled(true);           ← 啟用更新鈕
13     btDelete.setEnabled(true);          ← 啟用刪除鈕
14 }
15
16 public void onInsertUpdate(View v){
17     String nameStr=etName.getText().toString().trim();
18     String phoneStr=etPhone.getText().toString().trim();
19     String emailStr=etEmail.getText().toString().trim();
20     if(nameStr.length()==0 ||           ← 任一欄位沒有內容即返回
21         phoneStr.length()==0 ||
22         emailStr.length()==0) return;
23
24     if(v.getId()==R.id.btUpdate)       ← 按更新鈕
```

# 熱線通訊家

```
25         update(nameStr, phoneStr, emailStr, cur.getInt(0));
26     else                                     ← 按新增鈕
27         addData(nameStr, phoneStr, emailStr);    取得 _id 值, 更新
28                                                     含此 _id 的記錄
29     requery();    ← 更新 Cursor 內容
30 }
31
32 public void onDelete(View v){    ← 刪除鈕的On Click事件方法
33     db.delete(tb_name, "_id="+cur.getInt(0), null);
34     requery();    ← 更新 Cursor 內容
35 }
36
37 public void call(View v){    ← 打電話
```



# 熱線通訊家

```
38     String uri="tel:" + cur.getString(           ↙ "phone" 欄位
39                                     cur.getColumnIndex(FROM[1]));
40     Intent it = new Intent(Intent.ACTION_VIEW, Uri.parse(uri));
41     startActivity(it);
42 }
43
44 public void mail(View v) {           ← 寄送電子郵件
45     String uri="mailto:"+cur.getString(       ↙ "email" 欄位
46                                     cur.getColumnIndex(FROM[2]));
47     Intent it = new Intent(Intent.ACTION_SENDTO, Uri.parse(uri));
48     startActivity(it);
49 }
```

# 熱線通訊家

```
db.delete(tb_name,  
         "_id="+cur.getInt(0),  
         null);
```

← 資料表名稱

← 刪除的條件式

← 此參數本例不使用，故設為 null

# 熱線通訊家

## step 6

熱線通訊家

旗標公司	02-23963257 service@flag.com.tw
旗訊公司	02-23214335 service@pcdiy.com.tw

姓名: 台北市

電話: 02-27708889

Email: service@mail.taipei.gov.tw

新增 修改 刪除

1 先輸入一筆新資料

2 按此鈕

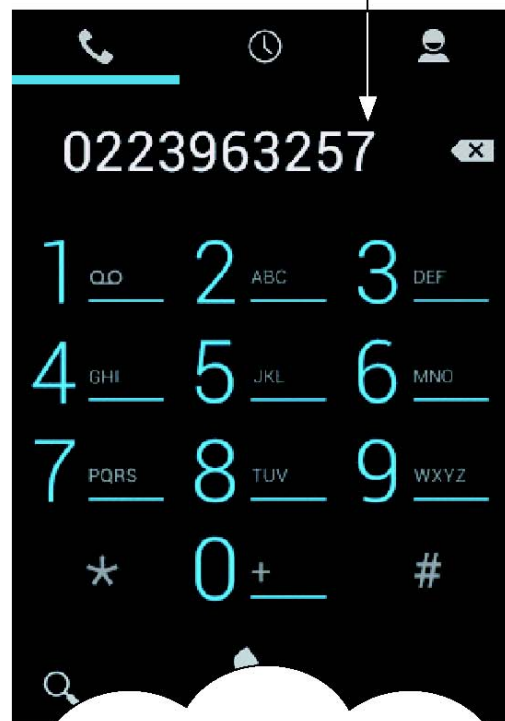
# 熱線通訊家

3 選取任一項



4 按電話鈕

啟動撥號的 Activity



若使用模擬器，由於模擬器無電子郵件軟體，所以按發信鈕會出現『不支援的操作』訊息。