



國立宜蘭大學

National Ilan University NIU

行動裝置互動系統設計與應用

授課老師：黃朝曦



教學單元

- SharedPreferences(偏好設定)
- ListView
- SQLite資料庫
- 小專題-電話簿

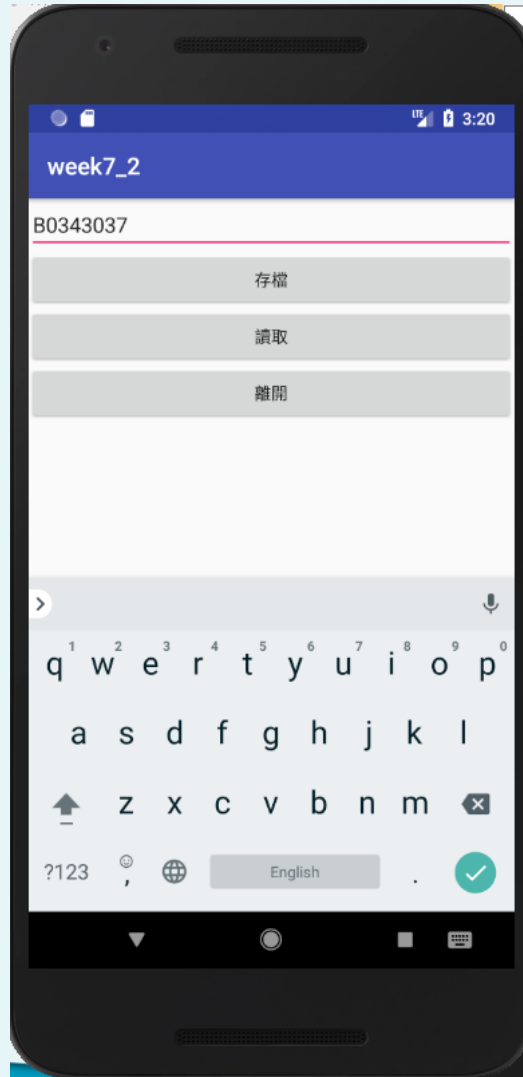


國立宜蘭大學

National Ilan University NIU

Shared preferences (偏好設定)

Shared preferences 範例



Sharedpreferences

- Sharedpreferences提供一個簡易且快速存取資料的方法，方便下次應用程式被啟動時，載入偏好設定，讓應用程式自動回復到前一次設定值。
- 以key-value pair方式儲存應用程式狀態，通常我們會在應用程式的onPause()方法中進行偏好設定儲存作業。

UI介面

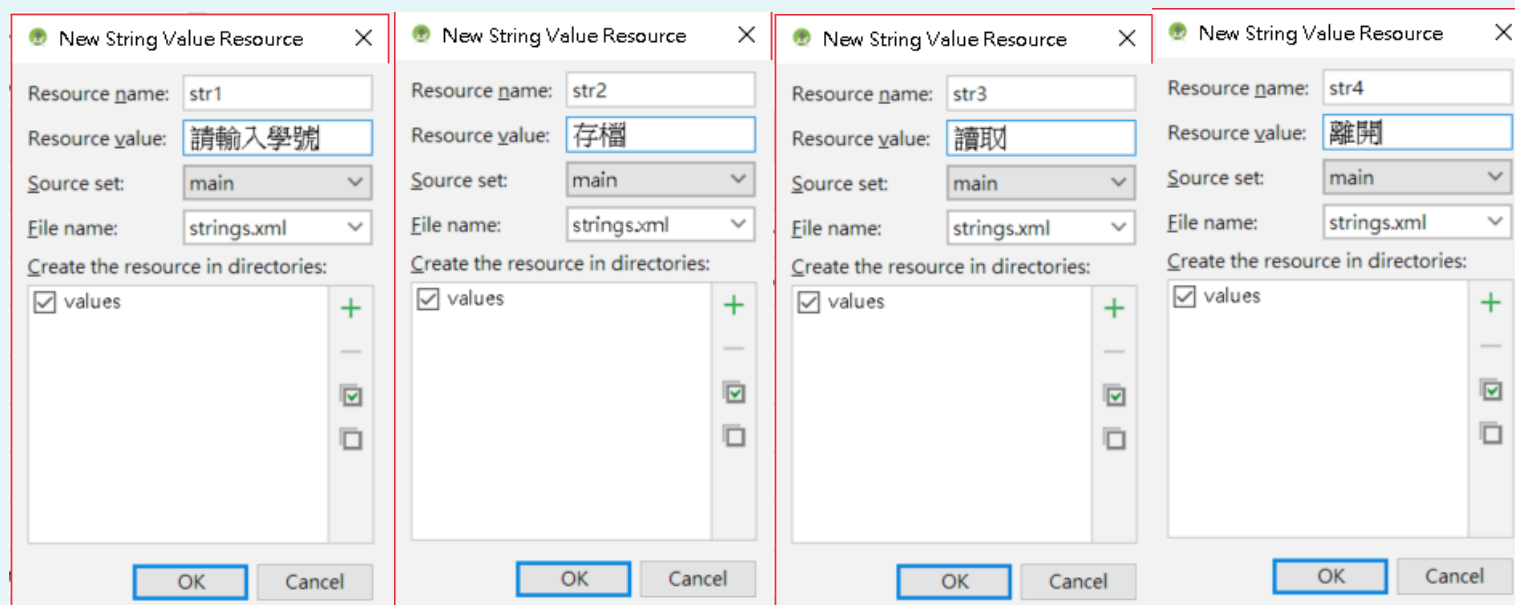


國立宜蘭大學

National Ilan University NIU

activity_main.xml

- 步驟1. 拉一個EditText和三個Button。
- 步驟2. 在EditText的Hint新增String(請輸入學號)
- 步驟3. 在Button的Text分別新增String(存檔、讀取、離開)



開始打程式 MainActivity.java

- 步驟1. 宣告UI物件。

```
Button mBtn1, mBtn2, mBtn3;  
EditText mEditText;
```

- 步驟2. 宣告SharedPreferences。

```
SharedPreferences sharedPreferences;
```

- 步驟3. 建立程式與元件的關係。

```
mBtn1 = (Button) findViewById(R.id.button1);  
mBtn2 = (Button) findViewById(R.id.button2);  
mBtn3 = (Button) findViewById(R.id.button3);  
mEditText = (EditText) findViewById(R.id.editText);
```


開始打程式 MainActivity.java

- 步驟4. 將三個按鈕分別建立一個監聽動作

```
mBtn1.setOnClickListener(new Button.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
  
    }  
});
```

- 步驟5. 在存檔按鈕的監聽裡面建立存檔指令

```
mSharedPreferences = getSharedPreferences("data", 0);  
mSharedPreferences.edit().putString("name", EditText.getText().toString()).commit();
```

開始打程式 MainActivity.java

- 步驟6.在讀取按鈕的監聽裡面建立讀取指令。

```
mSharedPreferences = getSharedPreferences("data", 0);  
mEditText.setText(mSharedPreferences.getString("name", ""));
```

- 步驟7.在離開按鈕的監聽裡面建立離開指令

```
MainActivity.this.finish();
```

完整onCreat()程式碼

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    mBtn1 = (Button) findViewById(R.id.button1);
    mBtn2 = (Button) findViewById(R.id.button2);
    mBtn3 = (Button) findViewById(R.id.button3);
    mEditText = (EditText) findViewById(R.id.editText);

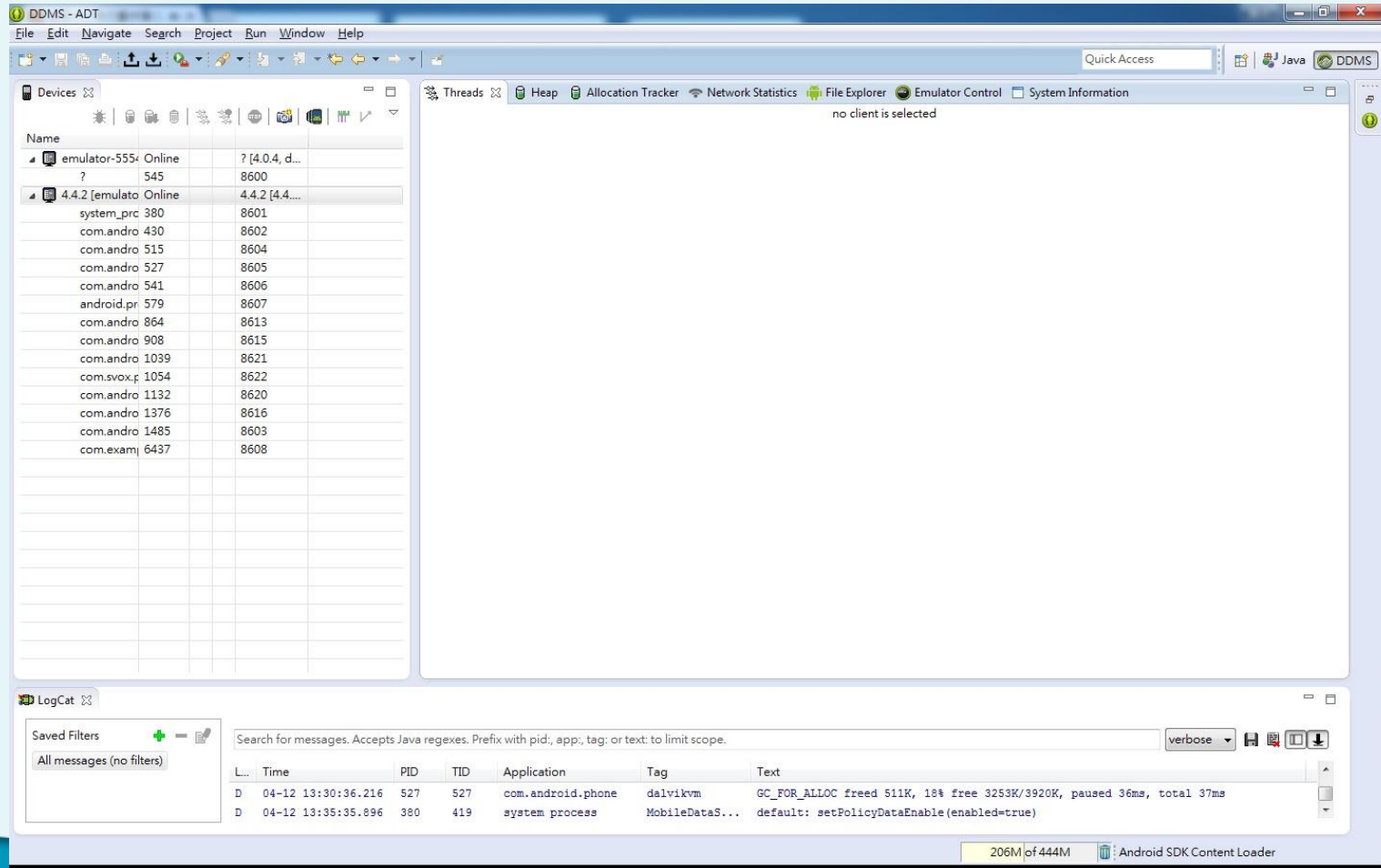
    mBtn1.setOnClickListener(new Button.OnClickListener() {
        @Override
        public void onClick(View v) {
            mSharedPreferences = getSharedPreferences( name: "data", mode: 0);
            mSharedPreferences.edit().putString("name", mEditText.getText().toString()).commit();
        }
    });

    mBtn2.setOnClickListener(new Button.OnClickListener() {
        @Override
        public void onClick(View v) {
            mSharedPreferences = getSharedPreferences( name: "data", mode: 0);
            mEditText.setText(mSharedPreferences.getString( key: "name", defValue: ""));
        }
    });

    mBtn3.setOnClickListener(new Button.OnClickListener() {
        @Override
        public void onClick(View v) {
            MainActivity.this.finish();
        }
    });
}
```

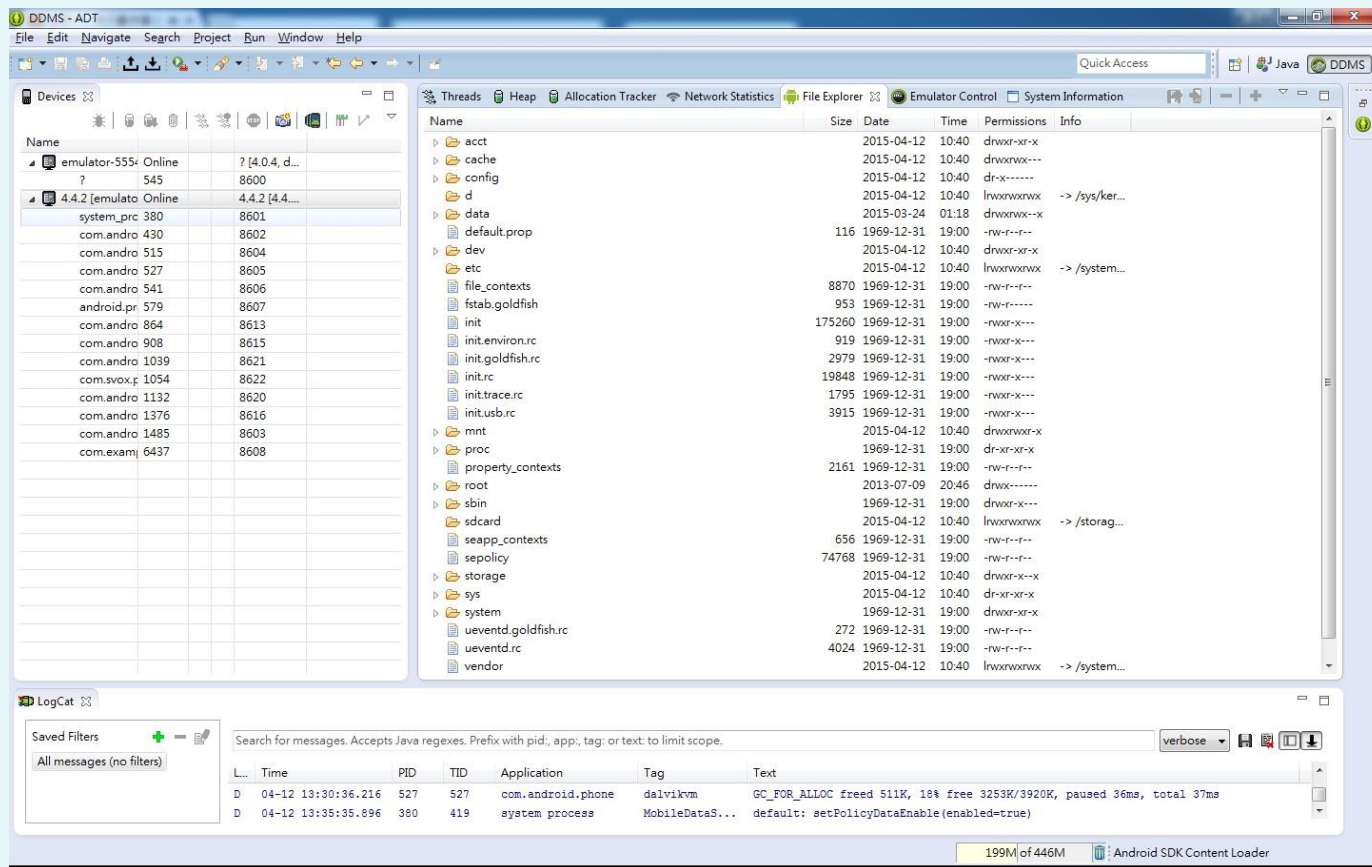
查詢SQLite資料

- 步驟1. 進入DDMS。



查詢SQLite資料

- 步驟2. 點擊Devices模擬器的名字後，在右邊找到File Explorer。

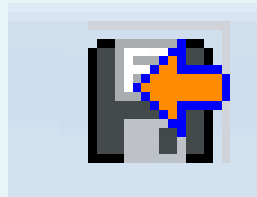


查詢Sharedpreferences資料

- 步驟3. 在data/data/**pagename**/shared_prefs/**檔案名.xml**。

└─ com.example.sharedpreferences	2015-04-12	11:38	drwxr-x--x	
└─┬─ cache	2015-04-12	11:01	drwxrwx--x	
└─┬─ lib	2015-04-12	11:38	lrwxrwxrwx	-> /data/a...
└─┬─ shared_prefs	2015-04-12	11:39	drwxrwx--x	
└─┬─ data.xml	111	2015-04-12	11:39	-rw-rw----

- 步驟4. 點選**檔案名.xml**後，找到右上方存取的按鈕，即可下載並用記事本，就可以看到剛剛儲存的資料。



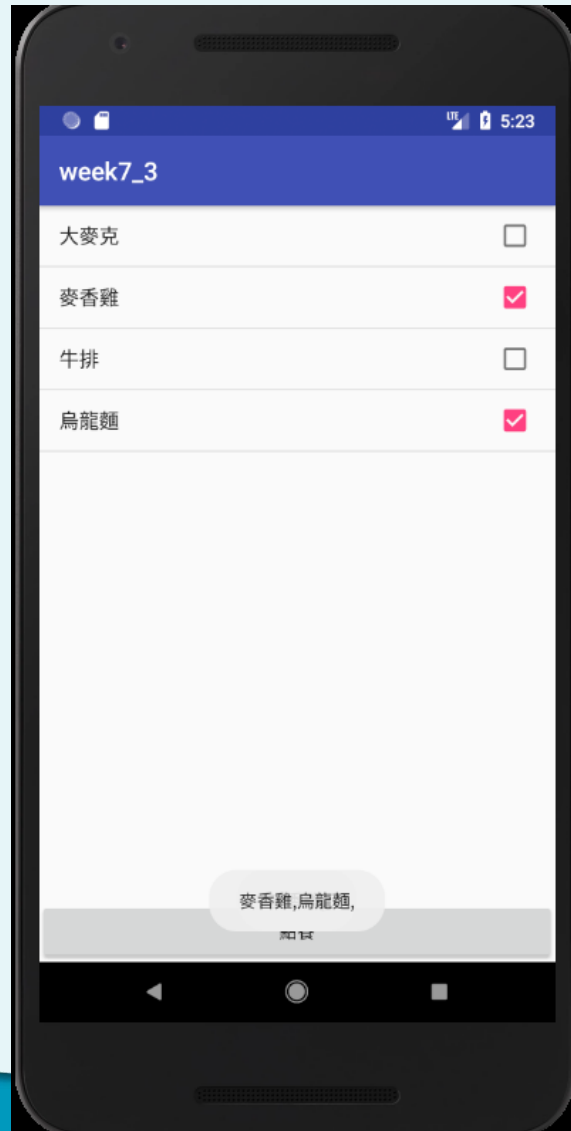


國立宜蘭大學

National Ilan University NIU

ListView

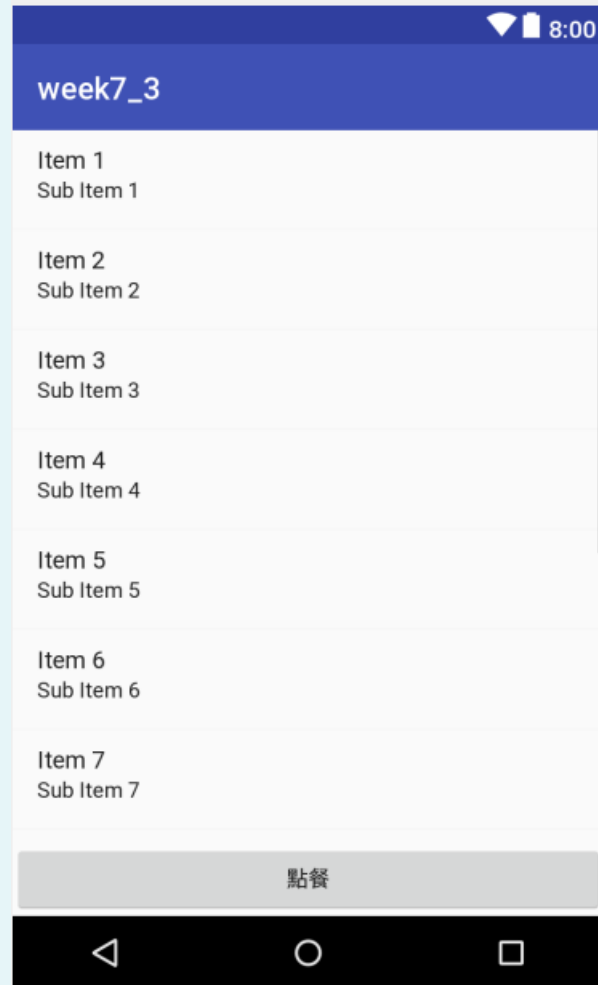
List View 範例



List View

- ListView是Android應用程式很常使用的畫面元件，它可以顯示多筆資料項目讓使用者瀏覽、選擇與執行後續的操作，當資料過多時，使用者可以透過捲動畫面來瀏覽。
- 最常用的有幾種Adapter：
 - **ArrayAdapter**：將一組數組連繫到ListView。
 - **SimpleAdapter**：適用於自訂ListView外觀的應用。
 - **BaseAdapter**：適用於需要自訂ListView外觀等靈活應用的場合。
 - **SimpleCursorAdapter**：將一組現有的資料庫或聯絡人等資料查詢的結果連繫到ListView中。

UI介面

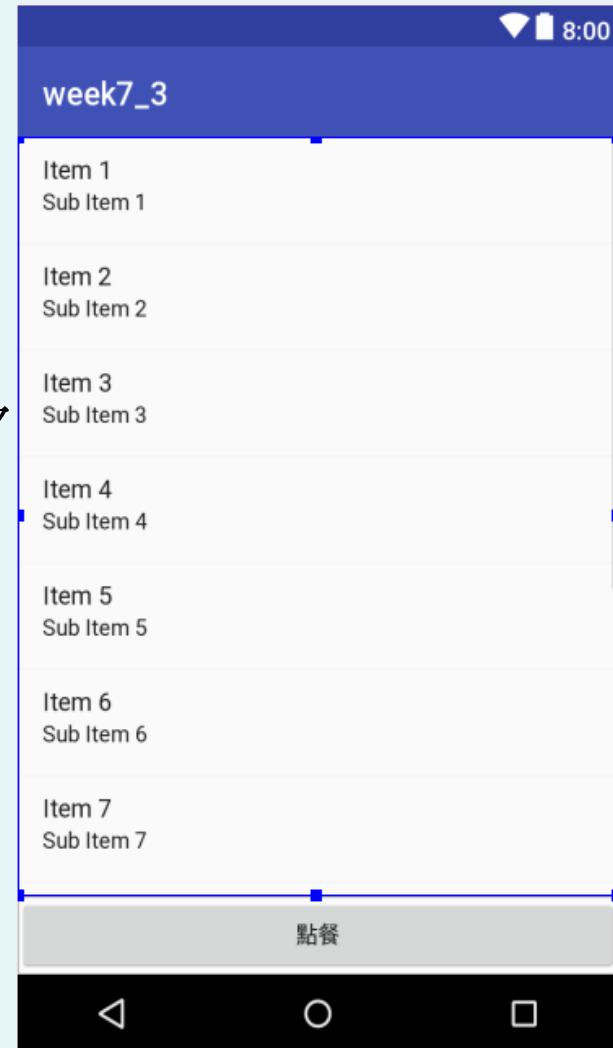


國立宜蘭大學

National Ilan University NIU

activity_main.xml

- 步驟1. 拉一個置底的Button和一個ListView(注意：**ListView**要將對齊點放在畫面的最上面和**Button**的上方，這樣才不會**ListView**顯示太多時蓋過**Button**)。
- 步驟2. 將Button的text設定成“點餐”。



開始打程式 MainActivity.java

- 步驟1. 宣告UI物件。

```
ListView lv;  
Button btn;
```

- 步驟2. 宣告菜單陣列。

```
String[] array = new String[] { "大麥克", "麥香雞", "牛排", "烏龍麵" };
```

- 步驟3. 建立程式與元件的關係。

```
lv = (ListView) findViewById(R.id.listView1);  
btn = (Button) findViewById(R.id.button1);
```

開始打程式 MainActivity.java

- 步驟4. 將陣列資料放入ListView中，並以多選呈現ListView。

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(
    MainActivity.this,
    android.R.layout.simple_list_item_multiple_choice, array);
lv.setAdapter(adapter);
```

↑
android官方提供內建多選介面。

- 步驟5. 將ListView點選方式改為多選模式。

```
lv.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
```

開始打程式 MainActivity.java

- 步驟6. 將按鈕建立一個監聽動作。

```
btn.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
  
    }  
});
```

開始打程式 MainActivity.java

- 步驟7. 在按鈕的監聽裡面建立多選列表，並用 Toast 呈現結果。

```
String list = "";  
for (int i = 0; i < array.length; i++)  
{  
    if (lv.getCheckedItemPositions().get(i))  
        list += array[i] + ",";  
}  
Toast.makeText(MainActivity.this, list, Toast.LENGTH_LONG)  
    .show();
```

判斷ListView是否被打勾。

完整按鈕程式碼

```
btn.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        String list = "";
        for (int i = 0; i < array.length; i++)
        {
            if(lv.getCheckedItemPositions().get(i))
                list += array[i] + ",";
        }
        Toast.makeText(MainActivity.this, list, Toast.LENGTH_LONG)
            .show();
    }
});
```




國立宜蘭大學

National Ilan University NIU

SQLite資料庫

SQLite資料庫範例



SQLite資料庫

- **SQLite**是個輕量化的關聯式資料庫，它輕薄短小，無須設定或管理，沒有伺服器及組態檔，不需要資料庫管理員，它只是一個xml的檔案，可以依須求四處移動，對行動裝置來說是非常好用的資料庫。
- 在建立資料表時，通常第一個資料欄位會命名為_id，並指定為主鍵，而且使用AUTOINCREMENT，當資料有新增時會自動遞增加1。

UI介面



國立宜蘭大學

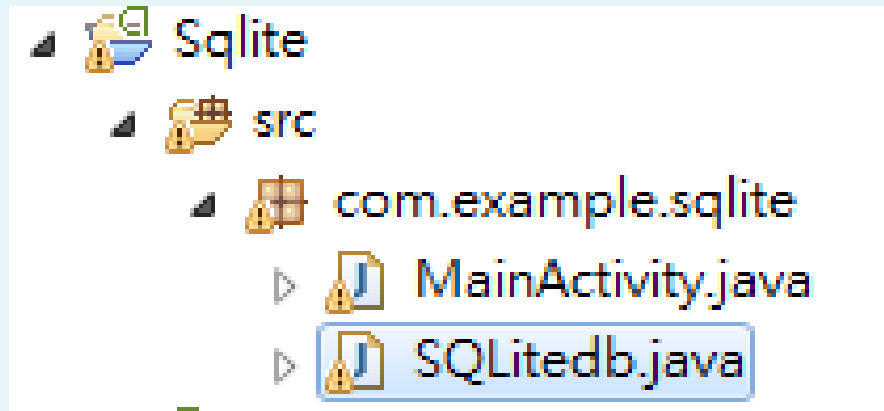
National Ilan University NIU

activity_main.xml

- 步驟1. 先設定Layout變更為RelativeLayout。
- 步驟2. 拉一個EditText、三個按鈕和ListView，並對齊如UI介面圖。
- 步驟3. 將左邊Button的text設定成“新增”；中間Button的text設定成“刪除”；右邊Button的text設定成“修改”。

開始打程式 MainActivity.java

- 步驟1. 載入資料庫(助教提供)。
 - 將SQLiteDb.java直接拉入src的資料夾裡面，程式碼的解釋皆附在檔案中。



- 步驟2. 宣告UI物件。

```
Button btn1, btn2, btn3;  
EditText ed;  
ListView lv;
```

開始打程式 MainActivity.java

- 步驟3. 將載入的SQLitedb.java與程式建立關係。

```
SQLitedb db = new SQLitedb(MainActivity.this);
```

- 步驟4. 宣告id來記錄游標點選的_id位置。

```
int id;
```

- 步驟5. 建立程式與元件的關係。

```
btn1 = (Button) findViewById(R.id.button1);  
btn2 = (Button) findViewById(R.id.button2);  
btn3 = (Button) findViewById(R.id.button3);  
ed = (EditText) findViewById(R.id.editText1);  
lv = (ListView) findViewById(R.id.listView1);
```

開始打程式 MainActivity.java

- 步驟6. 取得資料。

```
db.curosr();
```

- 步驟7. 建立ListView畫面，將資料庫資料放入ListView中，並把EditText的內容初始化，由於會使用很多次，因此使用附程式。

```
private void listviewupdate() {  
    ArrayAdapter<String> listAdapter = new ArrayAdapter<String>(  
        MainActivity.this, android.R.layout.simple_list_item_1,  
        db.select());  
    lv.setAdapter(listAdapter);  
    ed.setText("");  
}
```

↑
android官方提供內建標準介面。

↑
查詢資料庫資料。

開始打程式 MainActivity.java

- 步驟8. 回到onCreat()，在任一空白處加入剛剛的附程式。

```
listviewupdate();
```

- 步驟9. 在ListView建立一個選項按下的監聽動作。

```
lv.setOnItemClickListener(new OnItemClickListener() {  
  
    @Override  
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,  
        long arg3) {  
        // TODO Auto-generated method stub  
  
    }  
});
```

開始打程式 MainActivity.java

- 步驟10. 當按下ListView的選項時，會將游標指向當筆資料，並將其_id和_text1放入id和EditText中。

```
db.cursor.moveToPosition(arg2);  
id = db.cursor.getInt(0);  
ed.setText(db.cursor.getString(1));
```

完整ListView選項按鈕監聽程式碼

```
lv.setOnItemClickListener(new OnItemClickListener() {  
  
    @Override  
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,  
        long arg3) {  
        // TODO Auto-generated method stub  
        db.cursor.moveToPosition(arg2);  
        id = db.cursor.getInt(0);  
        ed.setText(db.cursor.getString(1));  
    }  
});
```

開始打程式 MainActivity.java

- 步驟11. 在新增的按鈕加入資料庫新增和更新 ListView 附程式。

```
db.insert(ed.getText().toString());  
listviewupdate();
```

- 步驟12. 在刪除的按鈕加入資料庫刪除和更新 ListView 附程式。

```
db.delete(id);  
listviewupdate();
```

- 步驟13. 在更新的按鈕加入資料庫更新和更新 ListView 附程式。

```
db.update(id, ed.getText().toString());  
listviewupdate();
```



新增按鈕完整程式碼

```
btn1.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View arg0) {  
        // TODO Auto-generated method stub  
        db.insert(ed.getText().toString());  
        listviewupdate();  
    }  
});
```

刪除按鈕完整程式碼

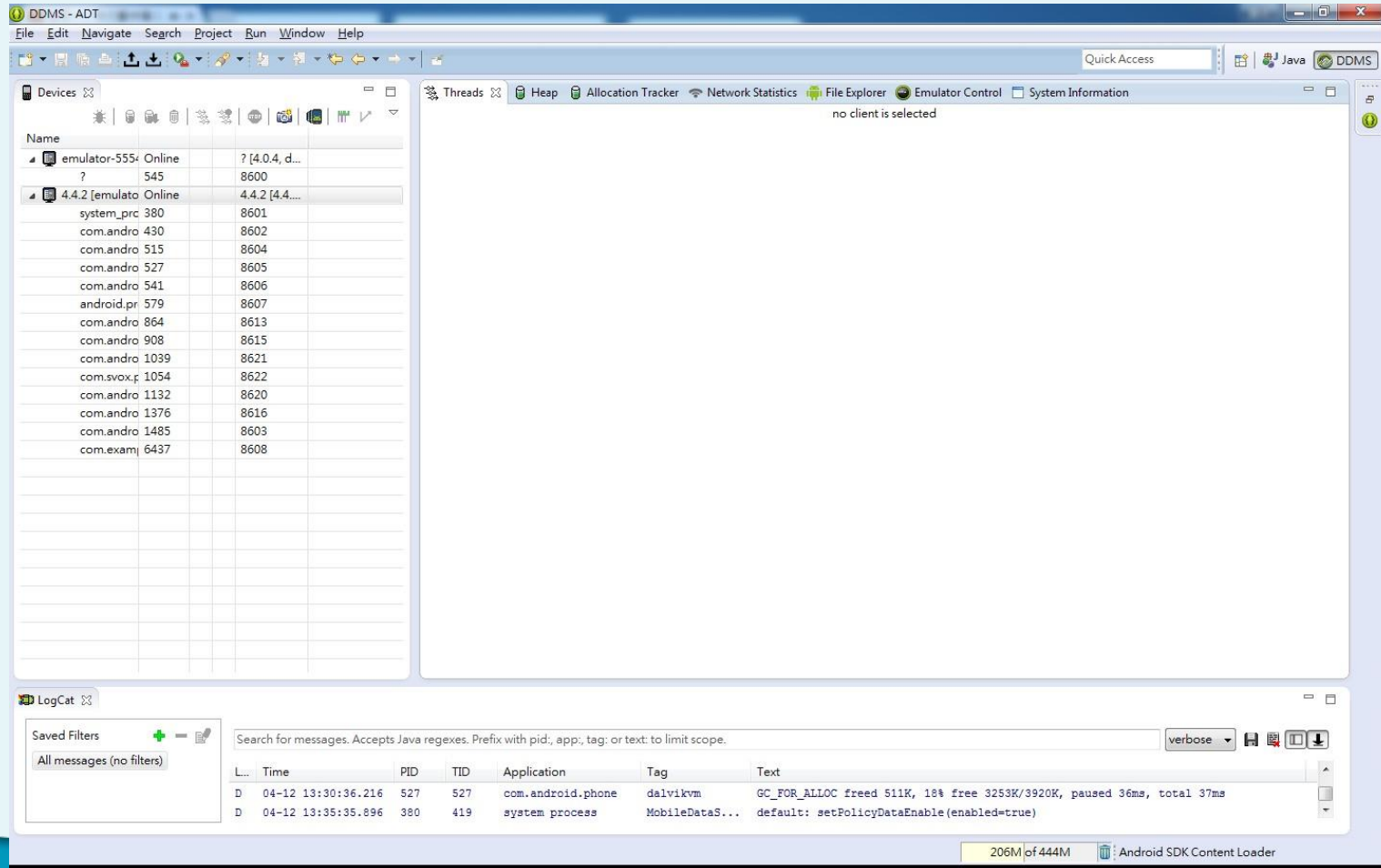
```
btn2.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View arg0) {  
        // TODO Auto-generated method stub  
        db.delete(id);  
        listviewupdate();  
    }  
});
```

更新按鈕完整程式碼

```
btn3.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        db.update(id, ed.getText().toString());  
        listviewupdate();  
    }  
});
```

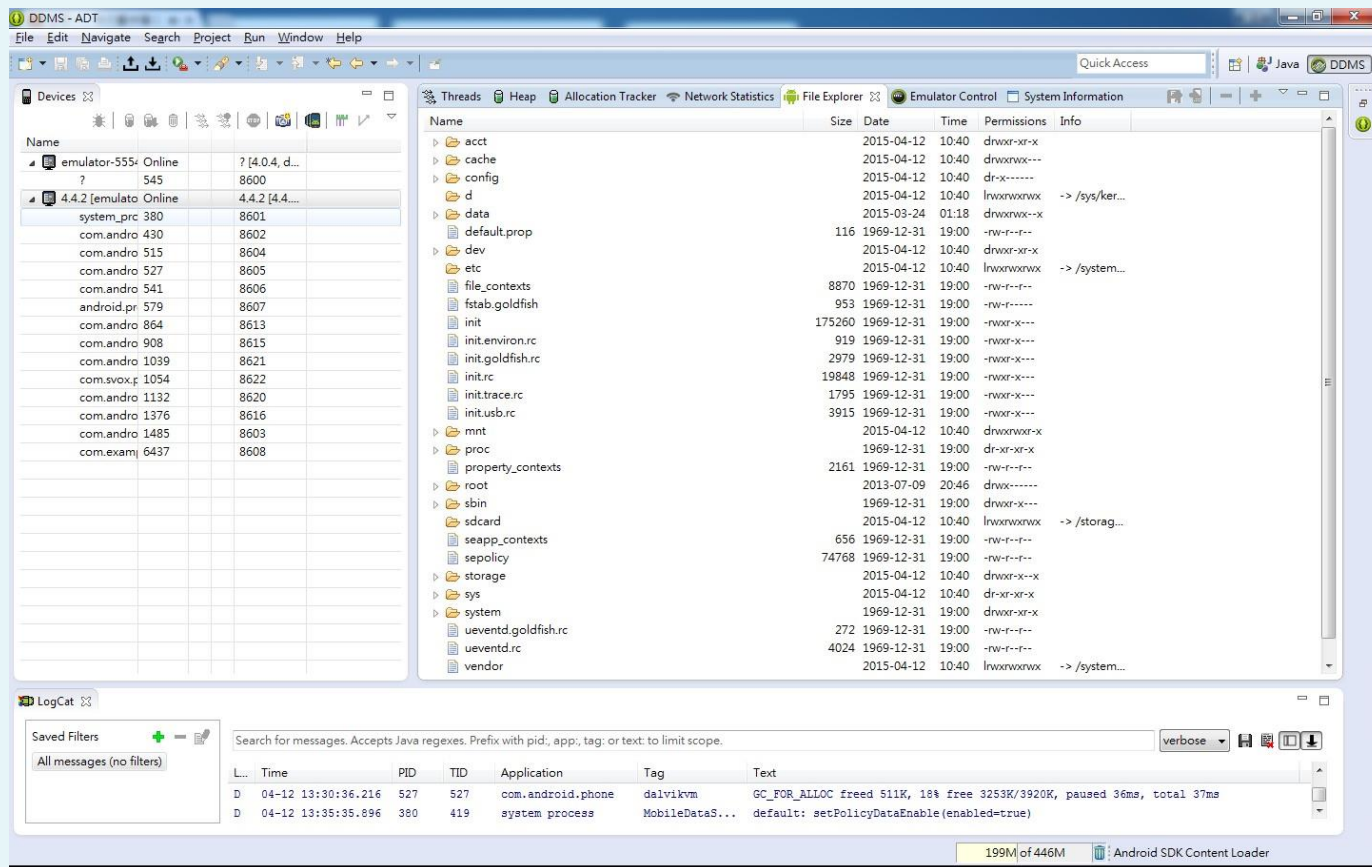
查詢SQLite資料

- 步驟1. 進入DDMS。



查詢SQLite資料

- 步驟2. 點擊Devices模擬器的名字後，在右邊找到File Explorer。



查詢SQLite資料

- 步驟3. 在data/data/**pagename**/databases/**資料庫名.xml**。

com.example.sqlite		2015-04-13	11:43	drwxr-x--x
cache		2015-04-13	11:43	drwxrwx--x
databases		2015-04-13	11:43	drwxrwx--x
DB	16384	2015-04-13	11:44	-rw-rw----
DB-journal	8720	2015-04-13	11:44	-rw-----
lib		2015-04-13	11:43	lrwxrwxrwx -> /data/a...

- 步驟4. 點選**資料庫名.xml**後，找到右上方存取的按鈕，即可下載並用記事本，就可以看到剛剛儲存的資料。





國立宜蘭大學

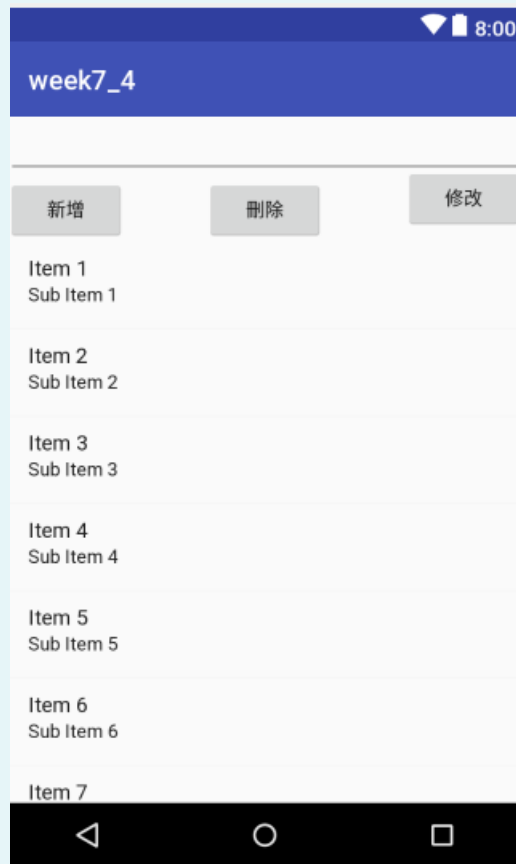
National Ilan University NIU

小專題- 電話簿

電話簿完成畫面



UI介面

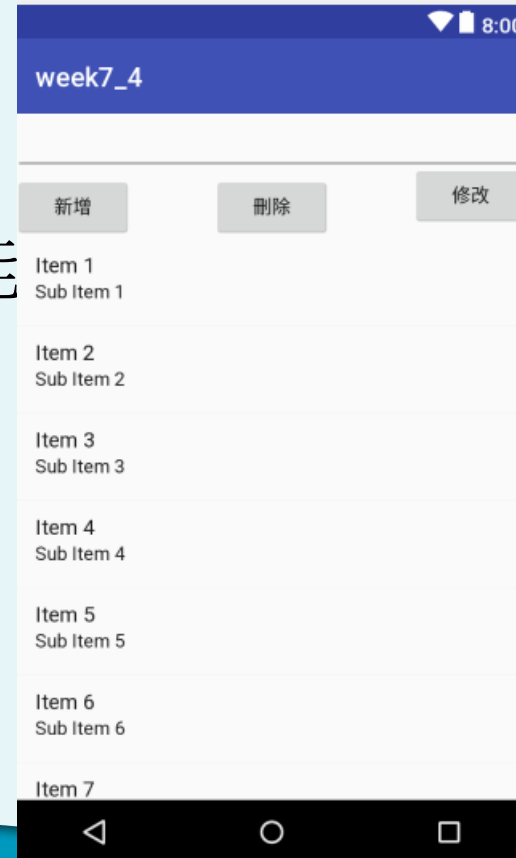


國立宜蘭大學

National Ilan University NIU

activity_main.xml

- 步驟1. 先設定Layout變更為RelativeLayout。
- 步驟2. 拉兩個EditText，三個Button和ListView。
- 步驟3. 在第一個EditText的Edit Hint設定姓名，第二個EditText的Edit Hint設定電話，並在InputType設定成Phone，這樣就只能輸入數字。



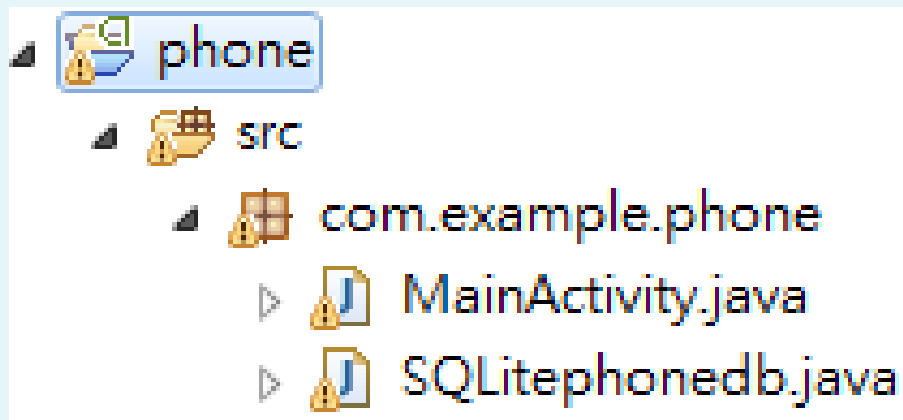


activity_main.xml

- 步驟4. 將左邊Button的text設定成“新增”；中間Button的text設定成“刪除”；右邊Button的text設定成“修改”。

開始打程式 MainActivity.java

- 步驟1. 載入資料庫(助教提供)。
 - 將SQLitephonedb.java直接拉入src的資料夾裡面，程式碼的解釋皆附在檔案中。



- 步驟2. 宣告UI物件。

```
EditText ed1, ed2;  
Button btn1, btn2, btn3;  
ListView lv;
```


開始打程式 MainActivity.java

- 步驟3. 將載入的SQLitephonedb.java與程式建立關係。

```
SQLitephonedb db = new SQLitephonedb(MainActivity.this);
```

- 步驟4. 宣告id來記錄游標點選的_id位置。

```
int id;
```

- 步驟5. 宣告SharedPreferences。

```
SharedPreferences sharedPreferences;
```

開始打程式 MainActivity.java

- 步驟6. 建立程式與元件的關係。

```
ed1 = (EditText) findViewById(R.id.editText1);  
ed2 = (EditText) findViewById(R.id.editText2);  
btn1 = (Button) findViewById(R.id.button1);  
btn2 = (Button) findViewById(R.id.button2);  
btn3 = (Button) findViewById(R.id.button3);  
lv = (ListView) findViewById(R.id.listView1);
```

- 步驟7. 找出onPause()生命週期的名字，並在裡面建立SharedPreferences。

```
sharedPreferences = getSharedPreferences("data", 0);  
sharedPreferences.edit().putInt("id", id)  
    .putString("name", ed1.getText().toString())  
    .putString("phone", ed2.getText().toString()).commit();
```



完整onPause()程式碼

```
@Override
protected void onPause() {
    // TODO Auto-generated method stub
    super.onPause();
    sharedPreferences = getSharedPreferences("data", 0);
    sharedPreferences.edit().putInt("id", id)
        .putString("name", ed1.getText().toString())
        .putString("phone", ed2.getText().toString()).commit();
}
```

開始打程式 MainActivity.java

- 步驟8. 回到onCreat()，取得SharedPreferences的data資料，並進行讀取資料，再將資料放入指定的EditText中。

```
sharedPreferences = getSharedPreferences("data", 0);  
id = sharedPreferences.getInt("id", 0);  
ed1.setText(sharedPreferences.getString("name", ""));  
ed2.setText(sharedPreferences.getString("phone", ""));
```

- 步驟9. 取得資料。

```
db.curosr();
```

開始打程式 MainActivity.java

- 步驟10. 建立ListView畫面，由於需要一次出現兩筆資料，所以先將資料庫的資料放入list二維陣列中。

```
ArrayList<HashMap<String, String>> list = new ArrayList<HashMap<String, String>>();  
for (int i = 0; i < db.select(0).size(); i++) {  
    HashMap<String, String> item = new HashMap<String, String>();  
    item.put("name", db.select(1).get(i));  
    item.put("phone", db.select(2).get(i));  
    list.add(item);  
}
```

HashMap

- HashMap也是以key-value的方式來取得資料，不同的是他的value可以應使用者的需求進行變更，可以是String、Integer或是Object。
- HashMap裡資料的排列與key-value加入順序都無關HashMap有自己獨特的排列方式。
- 在步驟10我們透過HashMap將值放入HashMap中，再透過ArrayList進行儲存，讓它形成二維陣列的樣子，當步驟11要求取得key資料就能快速取得其對應的value。

開始打程式 MainActivity.java

- 步驟11. 最後利用利用android內建的
simple_list_item_2排列呈現，並且載入置
ListView，再將它附程式化，方便多次使用。

```
SimpleAdapter adapter = new SimpleAdapter(MainActivity.this, list,  
    android.R.layout.simple_list_item_2, new String[] { "name",  
        "phone" }, new int[] { android.R.id.text1,  
        android.R.id.text2 });  
lv.setAdapter(adapter);
```


完整附程式程式碼

```
private void listviewupdate() {
    ArrayList<HashMap<String, String>> list = new ArrayList<HashMap<String, String>>();
    for (int i = 0; i < db.select(0).size(); i++) {
        HashMap<String, String> item = new HashMap<String, String>();
        item.put("name", db.select(1).get(i));
        item.put("phone", db.select(2).get(i));
        list.add(item);
    }
    SimpleAdapter adapter = new SimpleAdapter(MainActivity.this, list,
        android.R.layout.simple_list_item_2, new String[] { "name",
            "phone" }, new int[] { android.R.id.text1,
            android.R.id.text2 });
    lv.setAdapter(adapter);
}
```


開始打程式 MainActivity.java

- 步驟12. 回到onCreat()，在任一空白處加入剛剛的附程式。

```
listviewupdate();
```

- 步驟13. 在ListView建立一個選項按下的監聽動作。

```
lv.setOnItemClickListener(new OnItemClickListener() {  
  
    @Override  
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,  
        long arg3) {  
        // TODO Auto-generated method stub  
  
    }  
});
```

開始打程式 MainActivity.java

- 步驟14. 當按下ListView的選項時，會將游標指向當筆資料，並將其_id、_text1、_text2放入id和指定的EditText中。

```
db.cursor.moveToPosition(arg2);  
id = db.cursor.getInt(0);  
ed1.setText(db.cursor.getString(1));  
ed2.setText(db.cursor.getString(2));
```

完整ListView選項按鈕監聽程式碼

```
lv.setOnItemClickListener(new OnItemClickListener() {  
  
    @Override  
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,  
        long arg3) {  
        // TODO Auto-generated method stub  
        db.cursor.moveToPosition(arg2);  
        id = db.cursor.getInt(0);  
        ed1.setText(db.cursor.getString(1));  
        ed2.setText(db.cursor.getString(2));  
    }  
});
```

開始打程式 MainActivity.java

- 步驟15. 在新增的按鈕加入資料庫新增和更新 ListView 附程式，以及將 EditText 初始化。

```
db.insert(ed1.getText().toString(), ed2.getText().toString());  
listviewupdate();  
ed1.setText("");  
ed2.setText("");
```

- 步驟16. 在更新的按鈕加入資料庫更新和更新 ListView 附程式，以及將 EditText 初始化。

```
db.update(id, ed1.getText().toString(), ed2.getText()  
        .toString());  
listviewupdate();  
ed1.setText("");  
ed2.setText("");
```

開始打程式 MainActivity.java

- 步驟17. 在刪除的按鈕加入資料庫刪除和更新ListView附程式，以及將EditText初始化。

```
db.delete(id);  
listviewupdate();  
ed1.setText("");  
ed2.setText("");
```

- P.S. EditText初始化不放在附程式是因為要取得SharedPreferences的資料，所以與SQLite教學時不同。



新增按鈕完整程式碼

```
btn1.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        db.insert(ed1.getText().toString(), ed2.getText().toString());
        listviewupdate();
        ed1.setText("");
        ed2.setText("");
    }
});
```

更新按鈕完整程式碼

```
btn2.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View arg0) {  
        // TODO Auto-generated method stub  
        db.update(id, ed1.getText().toString(), ed2.getText()  
            .toString());  
        listviewupdate();  
        ed1.setText("");  
        ed2.setText("");  
    }  
});
```

刪除按鈕完整程式碼

```
btn3.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        db.delete(id);  
        listviewupdate();  
        ed1.setText("");  
        ed2.setText("");  
    }  
});
```