

# 0104課程大綱

- 用Intent啟動APP中的其它Activity
  - 在APP中新增Activity
  - 用Intent啟動APP中的Activity
  - 包含2個Activity的APP範例
- 在Intent中夾帶資料傳給新的Activity
- 程式範例-迷你備忘錄
- 程式作業

# 在APP中新增Activity(1)

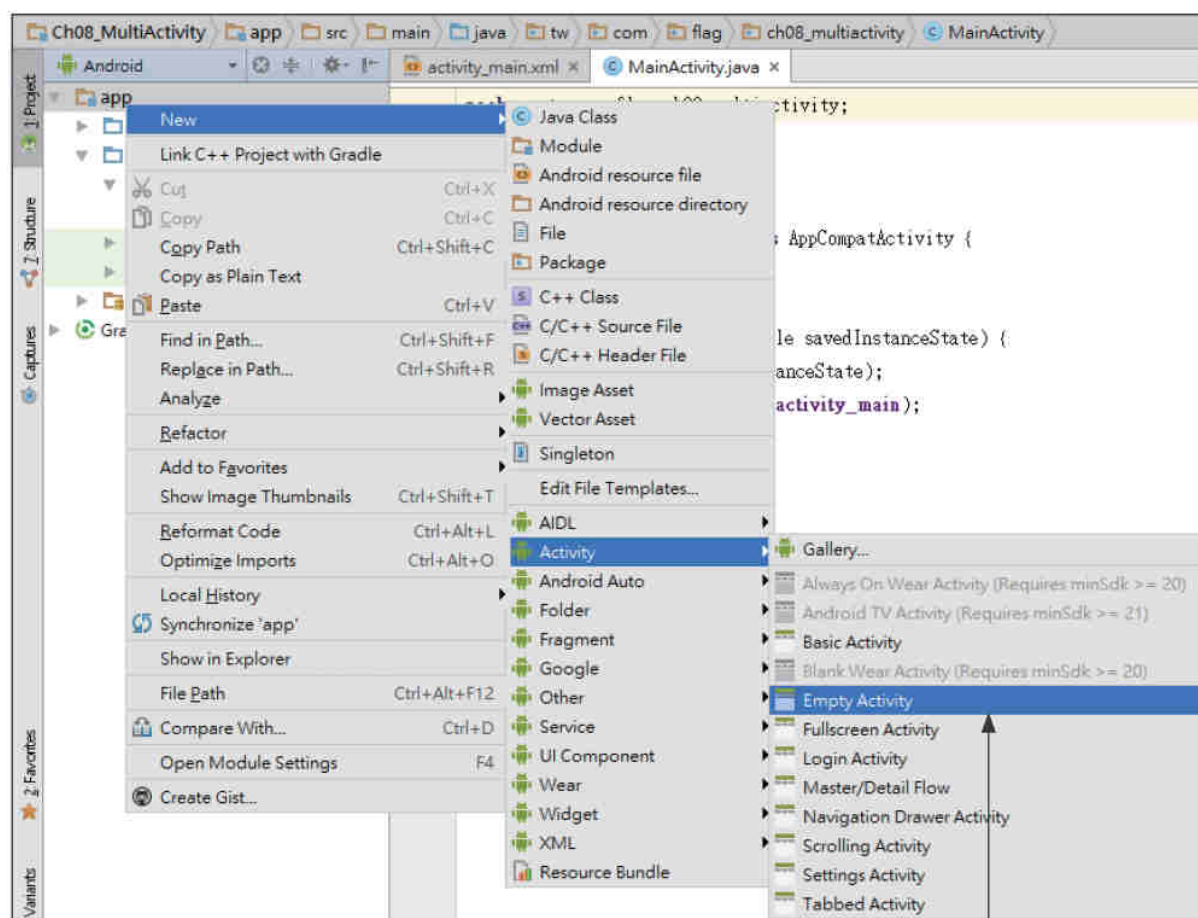
- 先前曾經介紹過，Android APP通常是由Activity組成，每一個Activity就代表一個畫面，也是一個可執行的獨立單元。
  - 程式如果需要多個畫面，就需要建立多個Activity。

# 在APP中新增Activity(2)

Step 1: 建立一個專案。

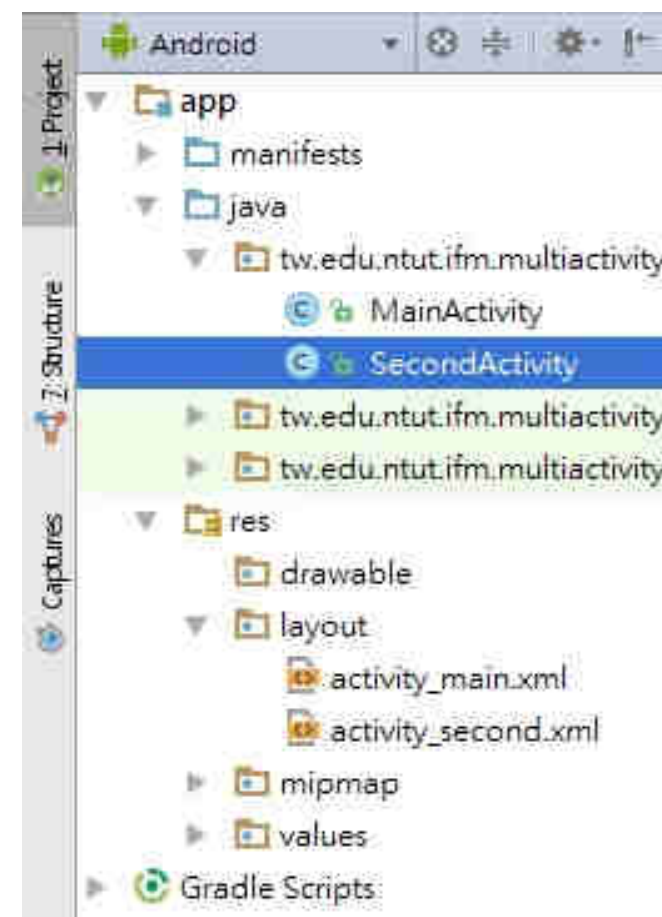
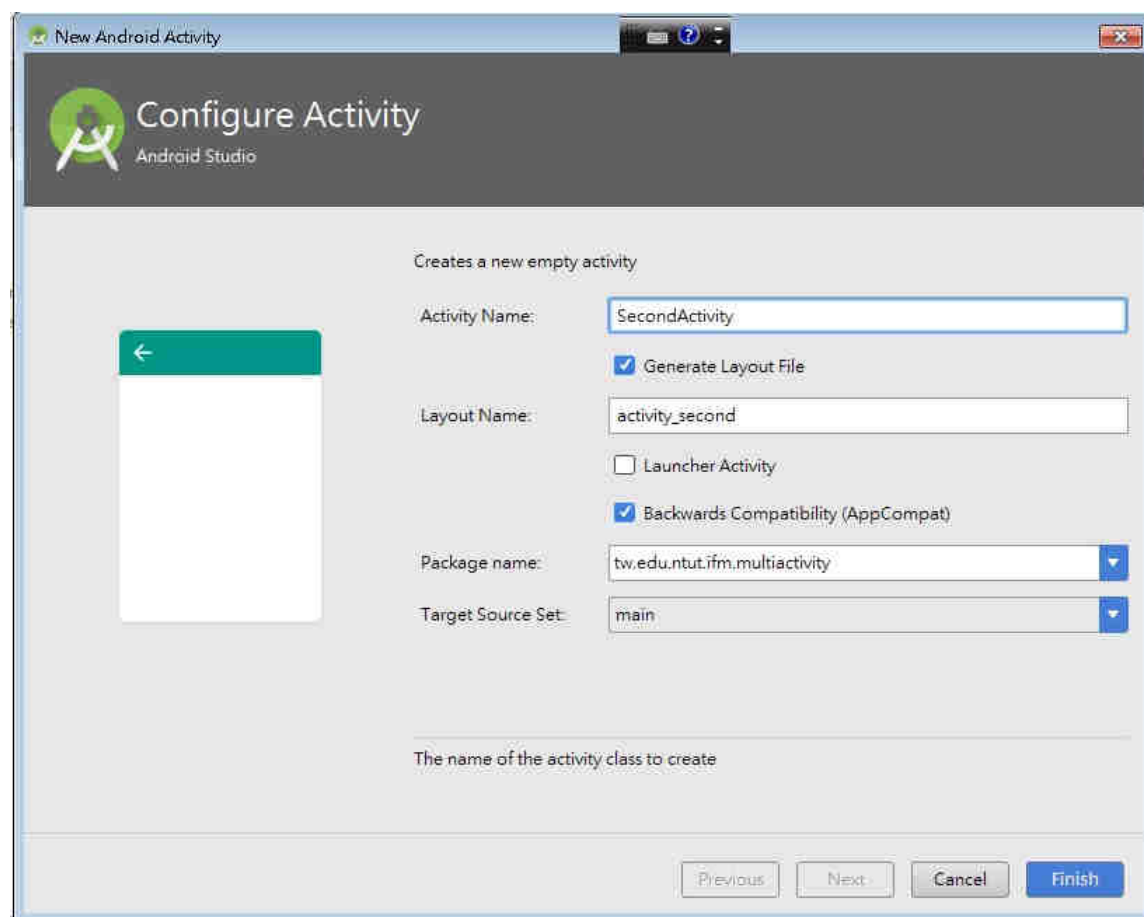
Step 2: 在此專案Project窗格的app模組上按下滑鼠右鍵，執行 New->Activity->Empty Activity命令。

# 在APP中新增Activity(3)



1 執行此命令

# 在APP中新增Activity(4)



# 用 Intent 啟動 APP 中的 Activity(1)

- 我們可以用 Intent(意圖)來啟動手機中的 Activity，其啟動方式可分成二種：
  - Explicit Intent：就是直接以類別名稱來指定要啟動那一個 Activity，例如，SecondActivity。
  - Implicit Intent：就是只在 Intent 中指出想要進行的動作(如：撥號、顯示、編輯)及資料(如：電話號碼、E-mail地址、網址)，由系統幫我找出適合的 Activity 來執行相關操作。

## 用 Intent 啟動 APP 中的 Activity(2)

- 假設啟動同一 APP 中，類別名稱為 Act2 的 Activity，Explicit Intent 的用法如下：

```
Intent it = new Intent();
```

← 建立 Intent 物件

```
it.setClass(this, Act2.class);
```

← 設定要啟動的 Activity 類別

```
startActivity(it);
```

← 啟動目標 Activity

## 用 Intent 啟動 APP 中的 Activity(3)

- 合併的寫法如下：

```
Intent it = new Intent(this, Act2.class);  
startActivity(it);
```

← 建立 Intent 物件並設定要啟動的 Activity 類別  
← 啟動目標 Activity

---

- 以下為更精簡的寫法：

```
startActivity(new Intent(this, Act2.class));
```

← 就地建立 Intent 物件來啟動

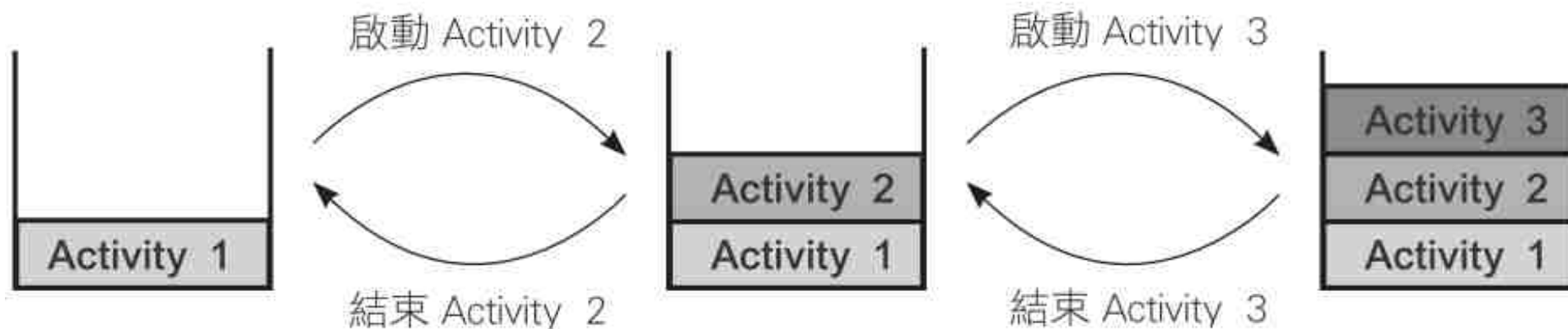
---



## 用 Intent 啟動 APP 中的 Activity(4)

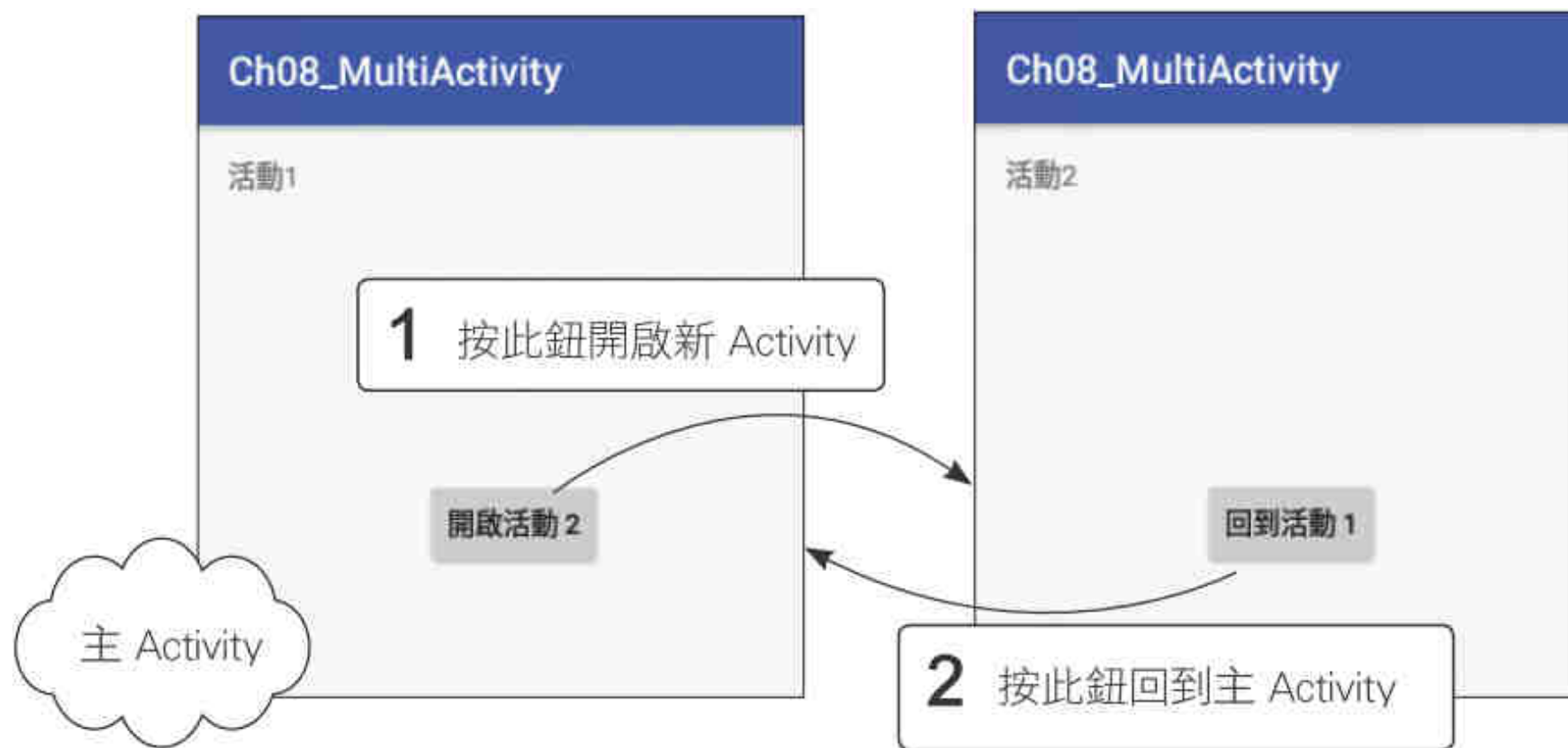
- startActivity(Intent)，可啟動新的 Activity，而 finish() 則可結束目前的 Activity。
  - 當啟動新的 Activity 時，新 Activity 會在前景執行。
  - 原本的 Activity 則會被推到背景中暫停執行。
  - 當新的 Activity 結束時，在背景中的 Activity 又會被提到前景來執行。

## 用 Intent 啟動 APP 中的 Activity(5)



# 包含2個Activity的APP範例(1)

請參考課本P.8-7~8-8：



## 包含2個Activity的APP範例(2)

```
01 public class MainActivity extends AppCompatActivity {  
02  
03     @Override  
04     protected void onCreate(Bundle savedInstanceState) {  
05         super.onCreate(savedInstanceState);  
06         setContentView(R.layout.activity_main);  
07     }  
08  
09     public void gotoSecondActivity(View v) {  
10         Intent it = new Intent(this, SecondActivity.class); ←  
11         startActivity(it); ← 啟動 Intent 中的目標 Activity  
12     }  
13  
14 }
```

## 包含2個Activity的APP範例(3)

```
01      public void goBack(View v) {  
02          finish();    ← 結束 Activity , 即可回到前一個 Activity  
03      }
```

## 在Intent中夾帶資料傳給新的Activity(1)

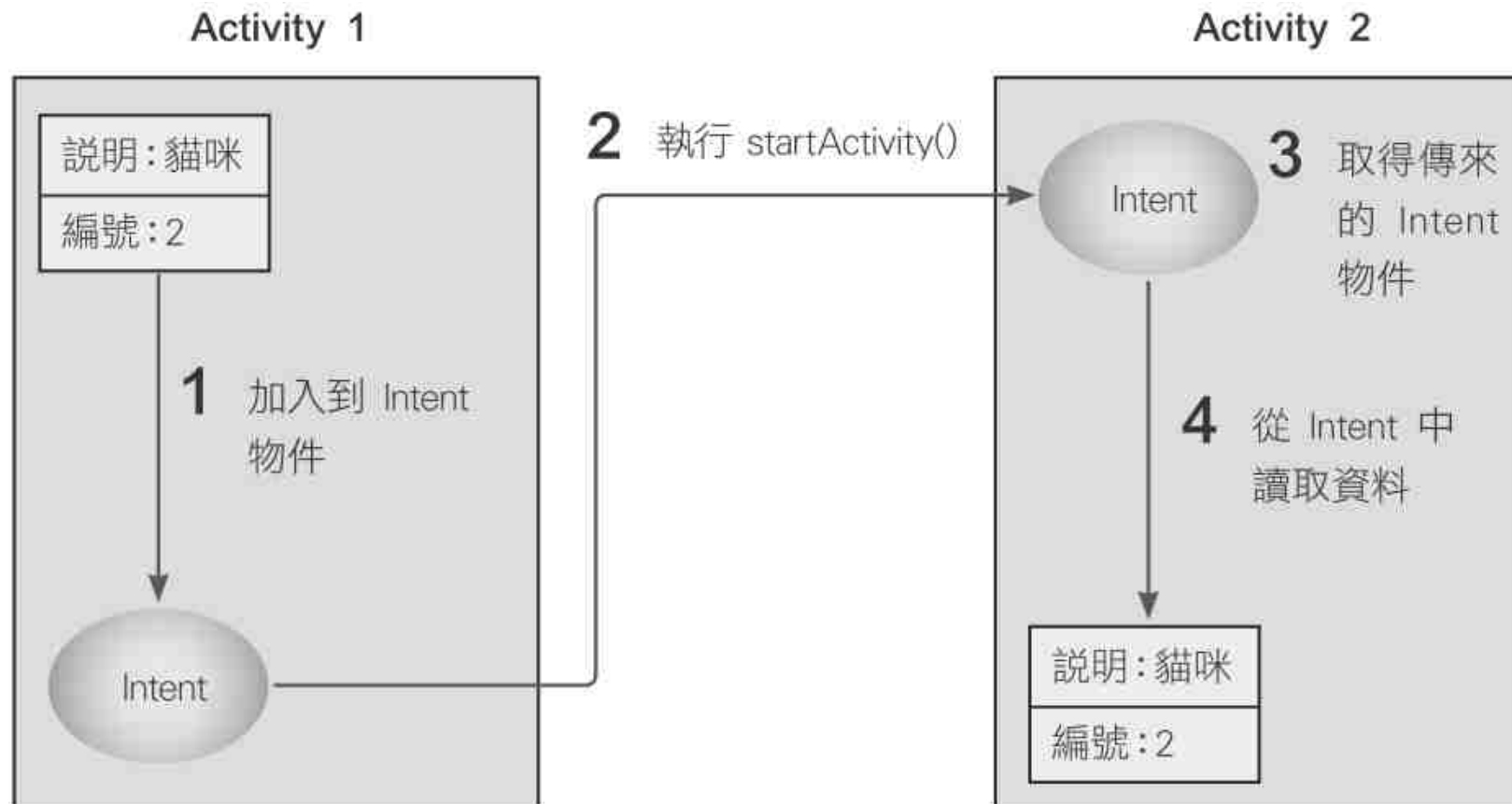


圖 8-1 夾帶資料傳給新 Activity

## 在Intent中夾帶資料傳給新的Activity(2)

### 附加資料到Intent：

```
String favor[] = { "魚肉", "小魚干", "貓餅干" };  
Intent it = new Intent(this, Act2.class);  
it.putExtra("編號", 2);      ← 入名稱為 "編號" 的 2 (Int 型別)  
it.putExtra("說明", "貓咪"); ← 加入名稱為 "說明" 的 "貓咪" (String 型別)  
it.putExtra("愛吃", favor); ← 加入名稱為 "愛吃" 的 String 陣列型別  
startActivity(it);           ← 啟動新 Activity Act2
```

## 在Intent中夾帶資料傳給新的Activity(3)

從Intent中取出資料：

```
Intent it = getIntent();           ← 取得傳入的 Intent 物件
int no = it.getIntExtra("編號", 0); ← 讀出名為 "編號" 的 Int 資料,
                                     若沒有則傳回 0

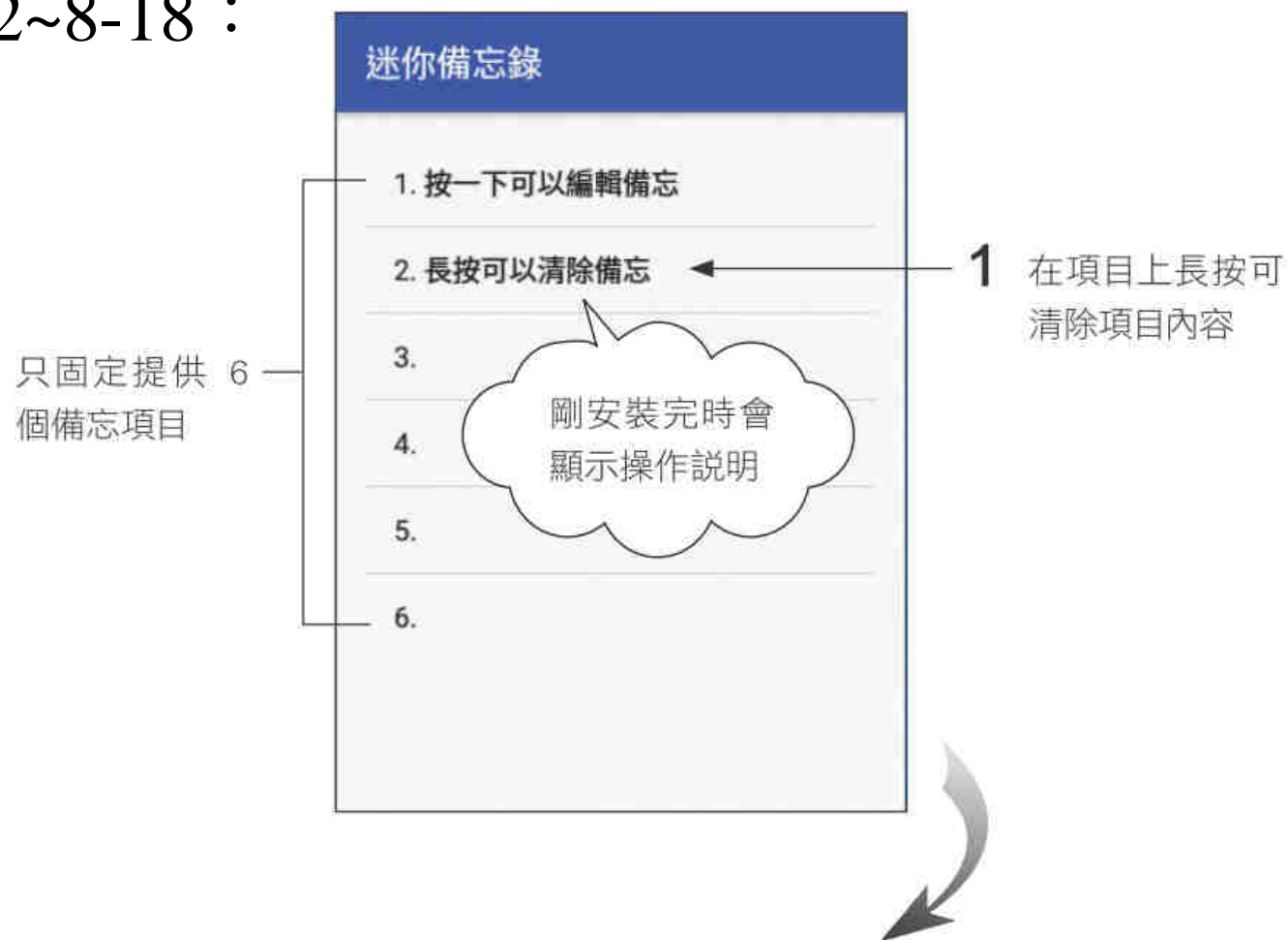
String da = it.getStringExtra("說明"); ←
                                     讀出名為 "說明" 的 String 資料, 若沒有則傳回 null

String a[] = it.getStringArrayExtra("愛吃"); ←
                                     讀出名為 "愛吃" 的 String[] 資料, 若沒有則傳回 null
```

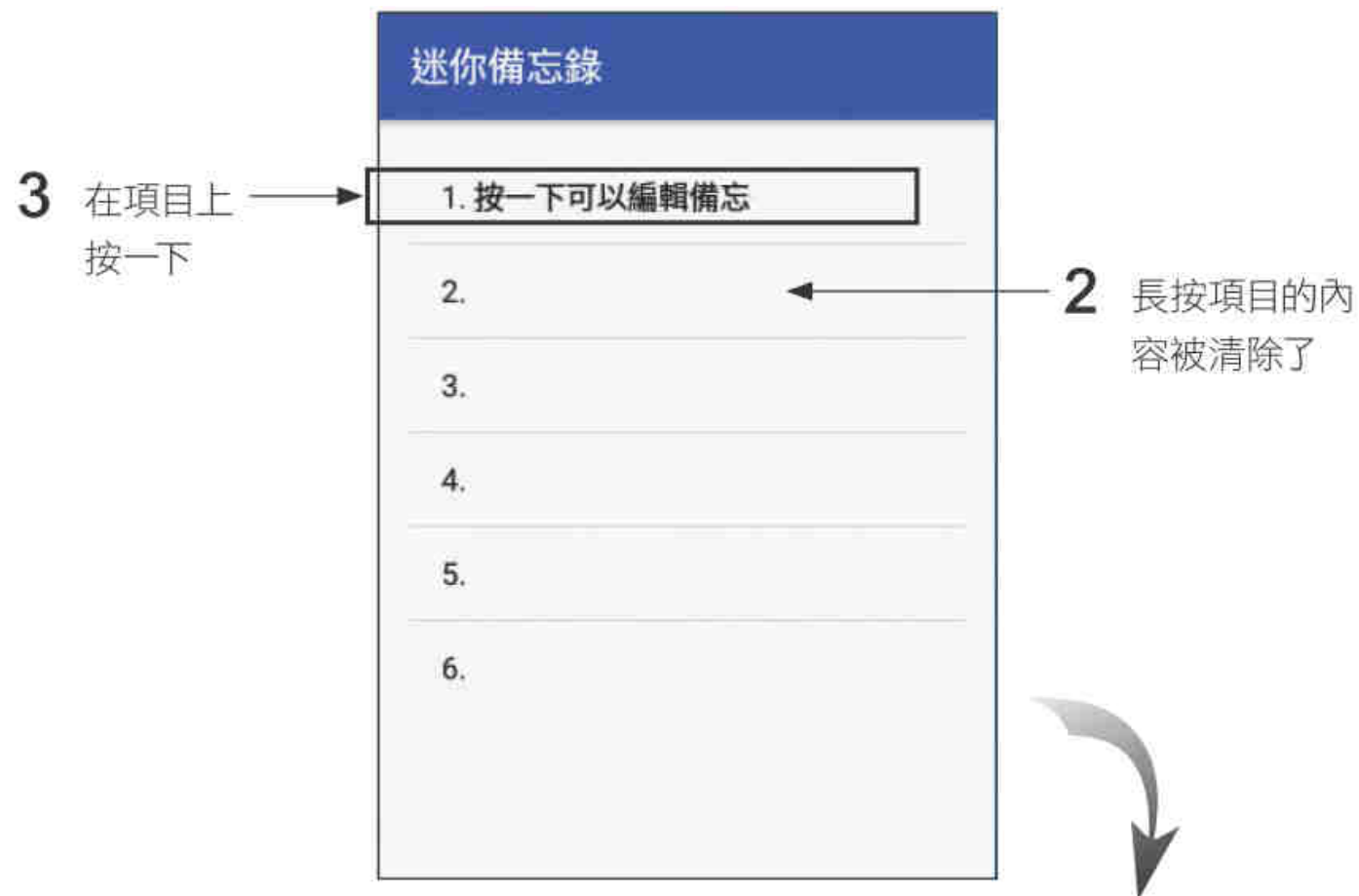


## 程式範例-迷你備忘錄(1)

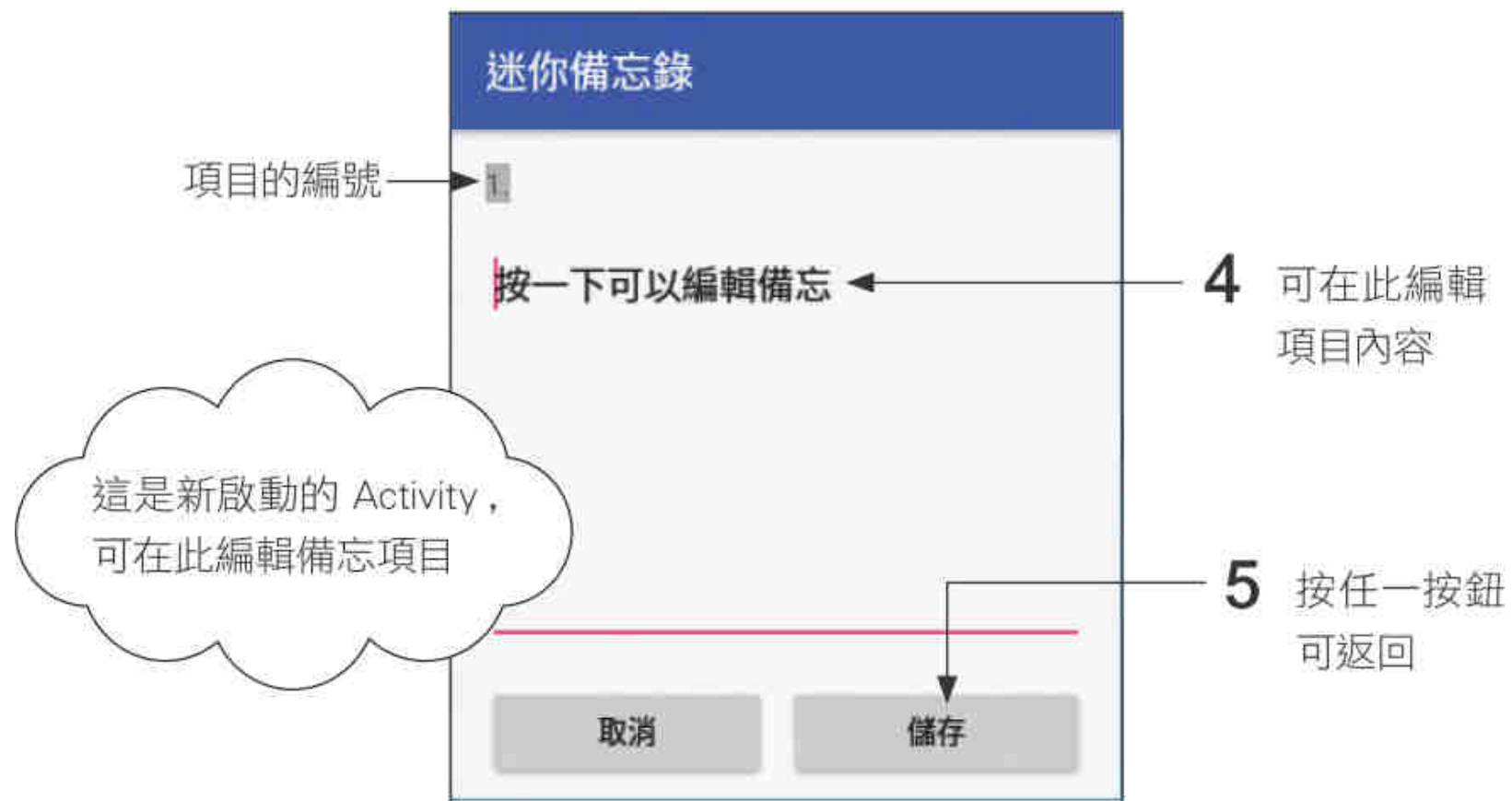
請參考課本P.8-12~8-18：



## 程式範例-迷你備忘錄(2)



## 程式範例-迷你備忘錄(3)



# Java語法-substring()

- `substring(int start)`：截取由 `start` 位置(由 0 算起)開始，到字串結束的所有字元用 Intent 啟動 APP 中的其它 Activity。
- `substring(int start, int end)`：截取由 `start` 到 `(end-1)` 之間的字元。

# 更新顯示內容

- notifyDataSetChanged()：可用來更新 ListView 或 Spinner 的顯示內容。
  - 直接修改作為 ArrayAdapter 物件資料來源的陣列內容，再呼叫 ArrayAdapter 的 notifyDataSetChanged() 方法。

## 程式範例-迷你備忘錄(4)

```
01 public class MainActivity extends AppCompatActivity
02     implements OnItemClickListener, OnItemLongClickListener{
03
04     String[] aMemo = { ← 預設的備忘內容
05         "1. 按一下可以編輯備忘",
06         "2. 長按可以清除備忘", "3.", "4.", "5.", "6." };
07     ListView lv; ← 顯示備忘錄的 ListView
08     ArrayAdapter<String> aa; ← ListView 與備忘資料 aMemo 的橋樑
09
```

## 程式範例-迷你備忘錄(5)

```
01      @Override
02      protected void onCreate(Bundle savedInstanceState) {
03          super.onCreate(savedInstanceState);
04          setContentView(R.layout.activity_main);
05
06          lv = (ListView)findViewById(R.id.listView);
07          aa = new ArrayAdapter<>(this,
08              android.R.layout.simple_list_item_1, aMemo);
09
10          lv.setAdapter(aa);    ← 設定 listView 的內容
11
12          //設定 listView 被按一下的監聽器
13          lv.setOnItemClickListener(this);
14          //設定 listView 被長按的監聽器
15          lv.setOnItemLongClickListener(this);
16      }
```

## 程式範例-迷你備忘錄(6)

```
01     public void onItemClick(AdapterView<?> a,  
02                             View v, int pos, long id) {  
03         Intent it = new Intent(this, Edit.class);  
04         it.putExtra("編號", pos+1);           ← 附加編號  
05         it.putExtra("備忘", aMemo[pos]);      ← 附加備忘項目的內容  
06         startActivity(it);                   ← 啟動 Edit 活動  
07     }  
08  
09     public boolean onItemClick(AdapterView<?> a,  
10                             View v, int pos, long id) {  
11         aMemo[pos] = (pos+1) + ".";          ← 將內容清除 (只剩編號)  
12         aa.notifyDataSetChanged();          ← 通知 ListView 要更新顯示的內容  
13         return true;                         ← 傳回 true 表示此事件已處理  
14     }
```



## 程式範例-迷你備忘錄(7)

```
01  protected void onCreate(Bundle savedInstanceState) {  
02      super.onCreate(savedInstanceState);  
03      setContentView(R.layout.activity_edit);  
04  
05      Intent it = getIntent();           ← 取得傳入的 Intent 物件  
06      int no = it.getIntExtra("編號", 0); ←  
                                         讀出名為 "編號" 的 Int 資料, 若沒有則傳回 0  
07      String s = it.getStringExtra("備忘"); ←  
                                         讀出名為 "備忘" 的 String 資料  
08  
09      TextView txv = (TextView)findViewById(R.id.textView);  
10      txv.setText(no + ".");           ← 在畫面左上角顯示編號
```

## 程式範例-迷你備忘錄(8)

```
11      EditText edt = (EditText)findViewById(R.id.editText);  
12      if(s.length() > 3)  
13          edt.setText(s.substring(3)); ←  
                                     將傳來的備忘資料去除前3個字，然後填入 EditText 元件中  
14  }  
15  
16  public void onCancel(View v) { ← 按取消鈕時  
17      finish(); ← 結束 Activity  
18  }  
19  public void onSave(View v) { ← 按儲存鈕時  
20      finish(); ← 結束 Activity  
21  }
```

# 程式作業

- 包含3個Activity的APP：(P8-9)
  - 活動1包含了一個按鈕“開啟活動2”。
  - 活動2包含了二個按鈕，分別是“開啟活動3”及“回到活動1”。
  - 活動3包含了一個按鈕“回到活動2”。
- 完成迷你備忘錄。
  - 請多傳目前時間日期給Edit，顯示在其EditText中(P.8-19及P.8-20)。