



# 行動裝置程式設計

## Unit 06 其它使用者介面 Other UI

蘇維宗 (Wei-Tsung Su)  
suwt@au.edu.tw  
564D



# 課程目標

Toast

Option Menu

Dialog



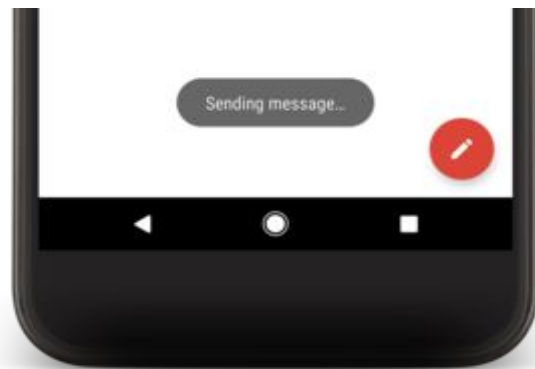
---

# Toast



# Toast

提供以彈出視窗的方式給使用者在操作上的回饋。  
例如，操作成功或失敗的訊息。



Toasts (來源: Android Developers)

1. `Toast toast = Toast.makeText(context, text, duration);`
2. `toast.show();`

## 程式碼說明

1. **context:** toast 出現的目標(通常是一個Application或Activity)
2. **text:** toast 所顯示的文字/或資源id
3. **duration:** toast 持續的時間(`Toast.LENGTH_LONG`, `Toast.LENGTH_SHORT`)



## 改變Toast出現的位置

可以利用`setGravity()`來改變Toast出現的位置, 例如

```
1. Toast toast = Toast.makeText(context, text, duration);  
2. toast.setGravity(Gravity.TOP|Gravity.LEFT, 0, 0); // 左上  
3. toast.show();
```

### 程式碼說明

```
1. public void setGravity(int gravity, int xOffset, int yOffset)  
2. Gravity.TOP, Gravity.BOTTOM, Gravity.LEFT, Gravity.RIGHT  
3. 可利用 | 並用上述參數值
```



---

# Option Menu



## Option Menu (1/2)

Option Menu為按下系統MENU鍵後所出現的選單。  
建議將對應用程式有全域影響的動作放在此，例如搜尋、設定等。

註1: Android 3.0 (API Level 11)以上可以Action Bar取代。

註2: 未來硬體設計可能會取消硬體 MENU鍵([相關文件](#))。





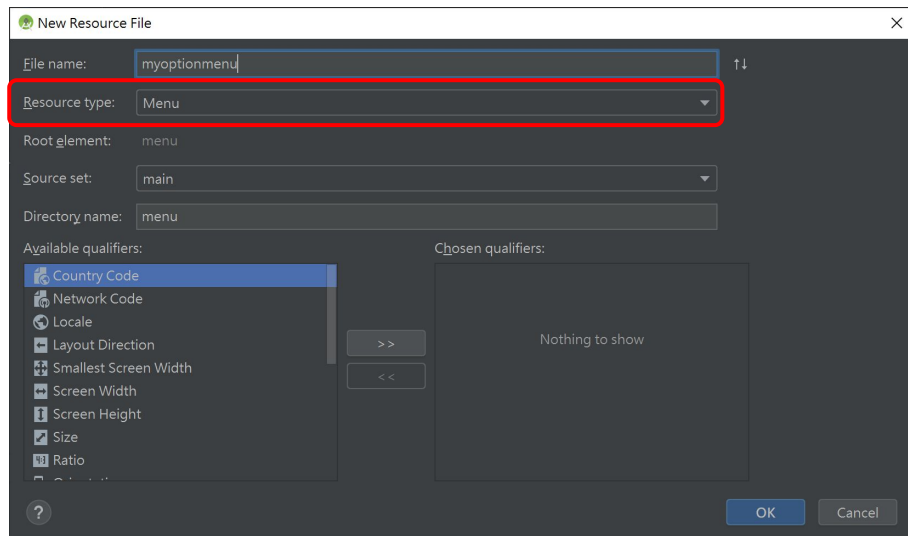
## Option Menu (2/2)

- **Step 1: 新增menu的layout**
  - 在res/menu/底下新增menu資源檔
- **Step 2: 產生Option Menu**
  - 在程式碼中實作 [onCreateOptionsMenu\(\)](#) 方法載入 menu layout 來產生 Option Menu。
- **Step 3: 撰寫點擊menu item的事件處理函式**
  - 在程式碼中實作 [onOptionsItemSelected\(\)](#) 方法撰寫使用者點擊 menu item 的事件處理函式。



## Step 1: 新增menu的layout

在res上按右鍵，選擇「New」「Android Resource File」



## Step 1: 新增menu的layout (續)



### Menu Item屬性

- `android:id="@+id/ 名稱 "`
- `android:title=" 字串或字串資源 "`
- `android:icon=" 圖檔資源 "`
- `android:showAsAction="Action Bar 屬性"`



## Step 1: 新增menu的layout (續)

利用 `android:showAsAction` 屬性決定 Action Bar 顯示方式, 屬性值有

- `never` (不以 Action Bar 方式呈現)
- `ifRoom` (如果有空間的話才以 Action Bar 方式呈現)
- `always` (以 Action Bar 方式呈現)
- `collapseActionView` (可折疊的 item - API Level 14 以上才支援。例如, Search 框)
- `withText` (顯示文字, 可利用 | 與上述屬性並用)
  - 例如, `android:showAsAction="always|withText"`





## Step 2: 產生Option Menu

在`onCreateOptionsMenu(...)` 中以`getMenuInflater().inflate()` 來產生Option Menu(或Action bar), 如下:

```
1.  @Override
2.  public boolean onCreateOptionsMenu(Menu menu) {
3.      getMenuInflater().inflate(R.menu.mymenu, menu);
4.      return super.onCreateOptionsMenu(menu);
5.  }
```

### 程式碼說明

1. `getMenuInflater().inflate(R.menu.mymenu, menu)` 是產生如在`res/menu/mymenu.xml`中所描述的Option Menu(或Action Bar)





## Step 3: 撰寫事件處理函式

實作onOptionsItemSelected 並撰寫不同 menu item被按下時的處理程式，如下：

```
1.  switch(item.getItemId()){  
2.      case R.id.item_add: //撰寫按下menu_add時所要執行的程式  
3.          break;  
4.      case R.id.menu_sub: //撰寫按下menu_sub時所要執行的程式  
5.          break;  
6.  }
```

### 程式碼說明

1. `item.getItemId()` 可以取得被按下之 menu item 的 id



## 課堂作業

請完成透過 Option Menu (或Action Bar)  
來執行加、減、乘、除的 MenuCalc專案。

## 驗收方式

課堂直接驗收

提示：  
無



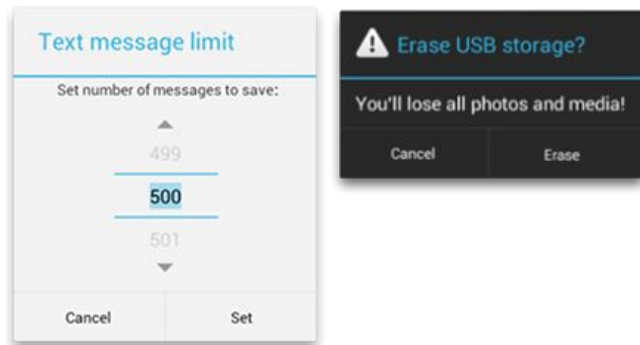
---

# Dialog



# Dialogs

- Dialog是一個小型的彈跳視窗，可以讓使用者用做決定或輸入其它資訊之用。
- 預設的dialog型態有AlertDialog、DatePickerDialog和TimePickerDialog。
- Dialog的建立方式
  - 建立dialog的builder
  - 利用builder設定dialog的格式
  - 利用builder來產生dialog
  - 利用dialog.show()顯示dialog



Dialogs(來源 : Android Developers)







# AlertDialog

- 建立dialog的builder
  - `AlertDialog.Builder builder = new AlertDialog.Builder(context);`
- 利用builder設定dialog的格式
  - `builder.setTitle(抬頭文字);`
  - `builder.setMessage(訊息文字);`
  - `builder.setPositiveButton(按鈕文字, OnClickListener事件處理函式);`
  - `builder.setNegativeButton(按鈕文字, OnClickListener事件處理函式);`
- 利用builder來產生dialog
  - `AlertDialog dialog = builder.create();`
- 利用dialog.show()顯示dialog
  - `dialog.show();`





## \* AlertDialog其他樣式

透過取代`builder.setMessage()`方法, 可以讓AlertDialog支援加入選項List、單選List或多選List的樣式。

- 多選List: `builder.setMultiChoiceItems()`
- 單選List: `builder.setSingleChoiceItems()`
- 選項List: `builder.setItems()`



## \*單選List的AlertDialog

利用String Array設定選項

```
1. String[] calcAdaptor = {"Add", "Sub", "Mul", "Div"};
```

產生具有單選List的AlertDialog

```
2. AlertDialog.Builder builder = new AlertDialog.Builder(context);  
3. builder.setTitle(抬頭文字);  
4. builder.setSingleChoiceItem(calcAdaptor, 0, new OnClickListener() {...});  
5. builder.setPositiveButton(...);  
6. builder.setNegativeButton(...);  
7. AlertDialog dialog = builder.create();  
8. dialog.show();
```



## \*單選List的AlertDialog (續)

利用String Array設定選項

```
1. String[] calcAdaptor = {"Add", "Sub", "Mul", "Div"};
```

產生具有單選List的AlertDialog

```
2. AlertDialog.Builder builder = new AlertDialog.Builder(context);  
3. builder.setTitle(抬頭文字);  
4. builder.setSingleChoiceItem(calcAdaptor, 0, new OnClickListener() {...});  
5. builder.setPositiveButton(...);  
6. builder.setNegativeButton(...);  
7. AlertDialog dialog = builder.create();  
8. dialog.show();
```



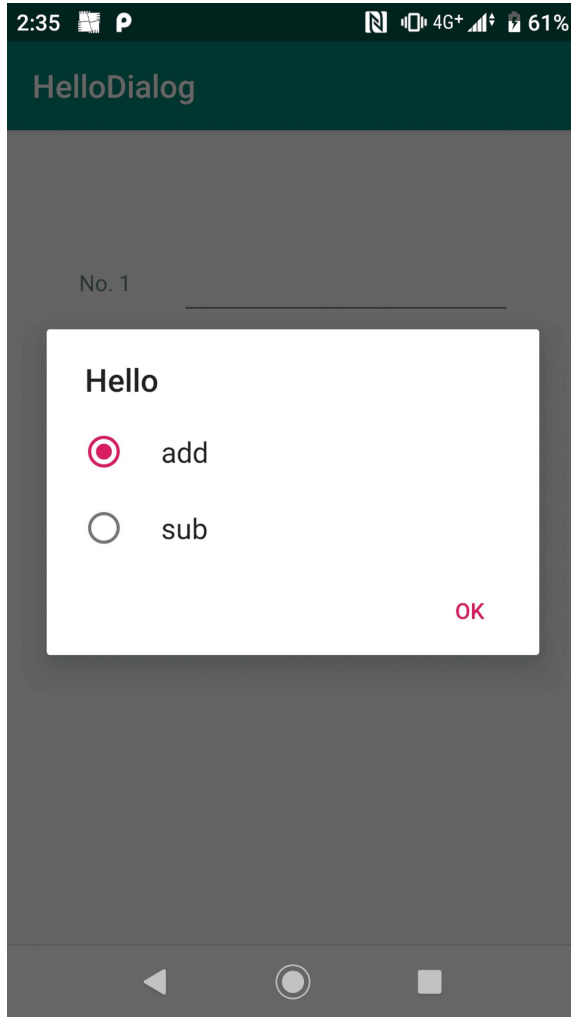
## 課堂作業

請完成透過AlertDialog來執行加、減、乘、除的DialogCalc專案。

## 驗收方式

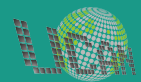
課堂直接驗收

提示：  
無



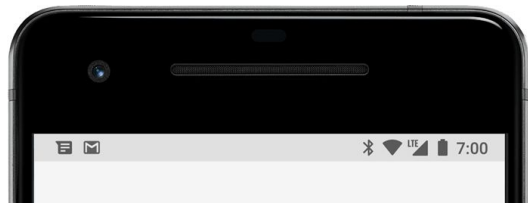
---

# 還有哪些UI元件可以選用?

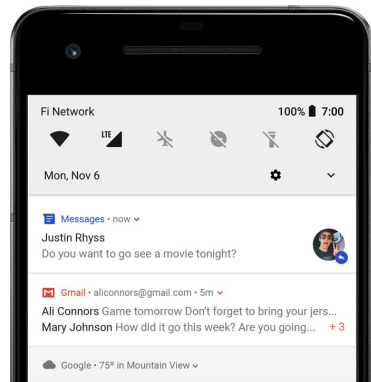


# 通知(Notification)

- Notification可以讓應用程式在系統的notification area中顯示圖示來提供訊息給使用者。訊息的詳細資料可以在notification drawer中顯示。



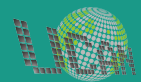
Notification area



Notification drawer

---

# 進階主題







## \*客製化Toast

- Step 1: 產生Toast Layout
  - 在res/layout底下產生客製的Toast layout
  - Layout必須加入android:id屬性
- Step 2: 產生客製化Toast
  - 在程式碼中產生客製化Toast



## \*Step 1: 產生Toast Layout

在res/layout底下產生layout, 以custom\_toast\_layout.xml 為範例

```
1.  <?xml version="1.0" encoding="utf-8"?>
2.  <LinearLayout xmlns:android=" http://schemas.android.com/apk/res/android "
3.      android:id="@+id/toast_layout"
4.      android:layout_width="match_parent"
5.      android:layout_height="match_parent"
6.      android:orientation="vertical" >
7.      <ImageView
8.          android:layout_width="match_parent"
9.          android:layout_height="match_parent"
10.         android:src="@android:drawable/ic_delete"/>
11.      <TextView
12.          android:layout_width="match_parent"
13.          android:layout_height="match_parent"
14.          android:text="Custom Toast"/>
15.  </LinearLayout>
```





## \*Step 2: 產生客製化Toast

以下列程式碼產生客製化Toast

```
1.  LayoutInflater inflater = getLayoutInflater();
2.  View layout = inflater.inflate(
3.      R.layout.toast_layout,
4.      (ViewGroup) findViewById(R.id.toast_layout)
5.  );
6.  Toast toast = new Toast(getApplicationContext());
7.  toast.setDuration	Toast.LENGTH_LONG);
8.  toast.setView(layout);
9.  toast.show();
```





## \*客製化Dialog

- Step 1: 產生Dialog Layout
  - 在res/layout底下產生客製的Dialog layout
- Step 2: 產生客製化Dialog
  - 在程式碼中產生客製化Dialog



## \*Step 1: 產生Dialog Layout

在res/layout底下產生layout, 以custom\_dialog\_layout.xml為範例

```
1.  <?xml version="1.0" encoding="utf-8"?>
2.  <LinearLayout xmlns:android=http://schemas.android.com/apk/res/android
3.      android:layout_width="match_parent"
4.      android:layout_height="match_parent"
5.      android:orientation="vertical" >
6.      <Button
7.          android:id="@+id/dialog_add"
8.          android:layout_width="match_parent"
9.          android:layout_height="match_parent"
10.         android:text="Add"/>
11.  </LinearLayout>
```





## \*Step 2: 產生客製化Dialog

取得Layout Inflater

```
1.  LayoutInflater inflater = getLayoutInflater();
```

產生客製化的AlertDialog

```
2.  AlertDialog.Builder builder = new AlertDialog.Builder(context);  
3.  builder.setView(inflater.inflate(R.layout.custom_dialog_layout, null));  
4.  AlertDialog dialog = builder.create();  
5.  dialog.show();
```



# Q & A

---



Computer History Museum, Mt. View, CA

