Module III
Android Basic User Interface

- UI設計基本觀念
- Widget基礎元件介紹
- 版面配置與Layout元件介紹

Android UI設計基本觀念 (1/5)

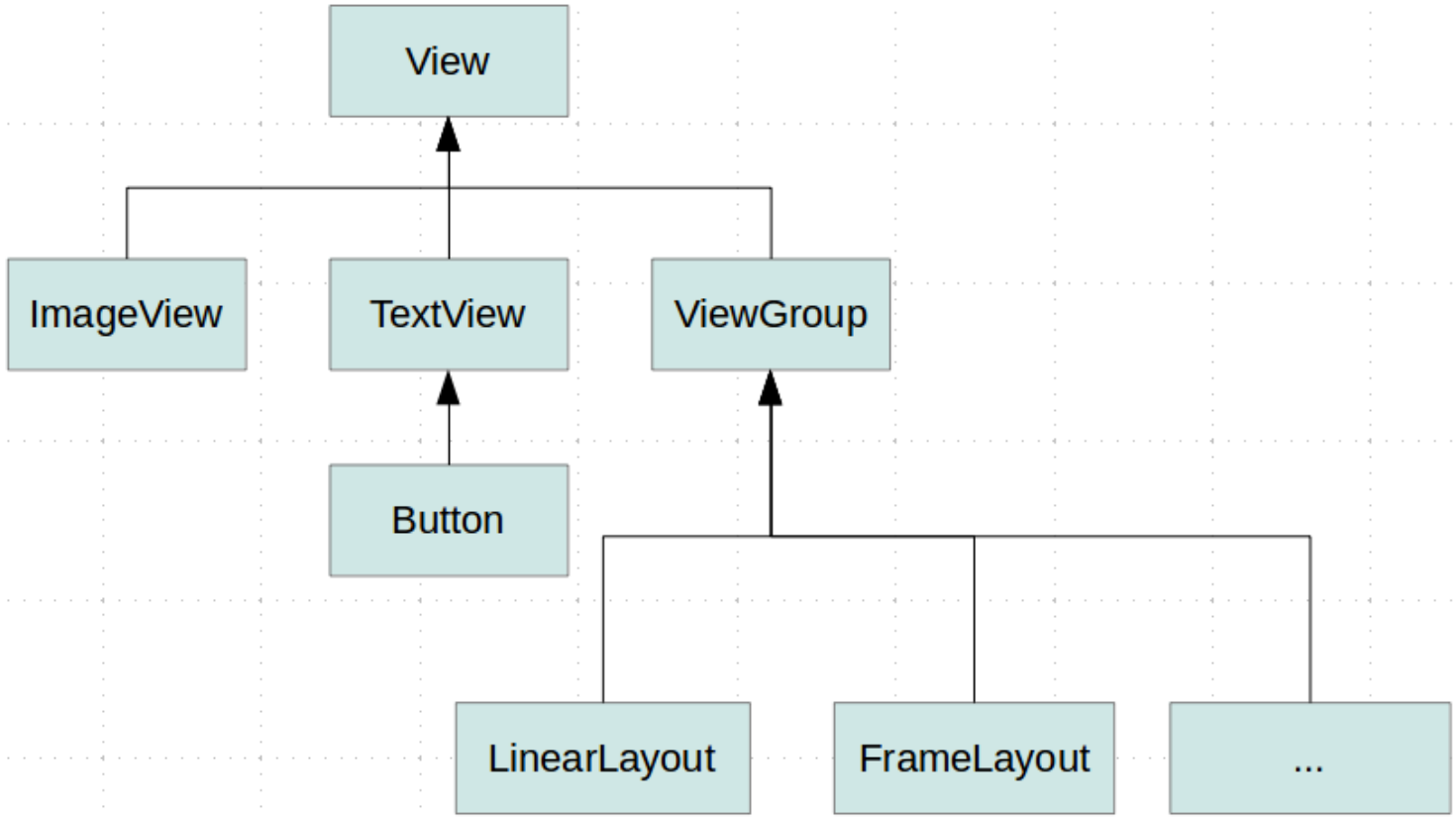
- 使用者與應用程式互動的畫面稱為UI (User Interface – 使用者介面)，包含了各種元件，如按鈕、文字輸入方塊，下拉選單等，我們稱之為UI元件 (UI elements)。

- 所有Android應用程式的UI元件都是繼承View類別，並可分成以下兩大類：
  - Widget元件 – 即最基本單位，如按鈕，文字輸入方塊等。
  - Layout元件 – 可再置入其它元件，如RelativeLayout, LinearLayout等

- 可使用XML檔案進行版面配置，或是利用程式碼動態產生元件，建議用XML檔案管理UI，原因上課時說明

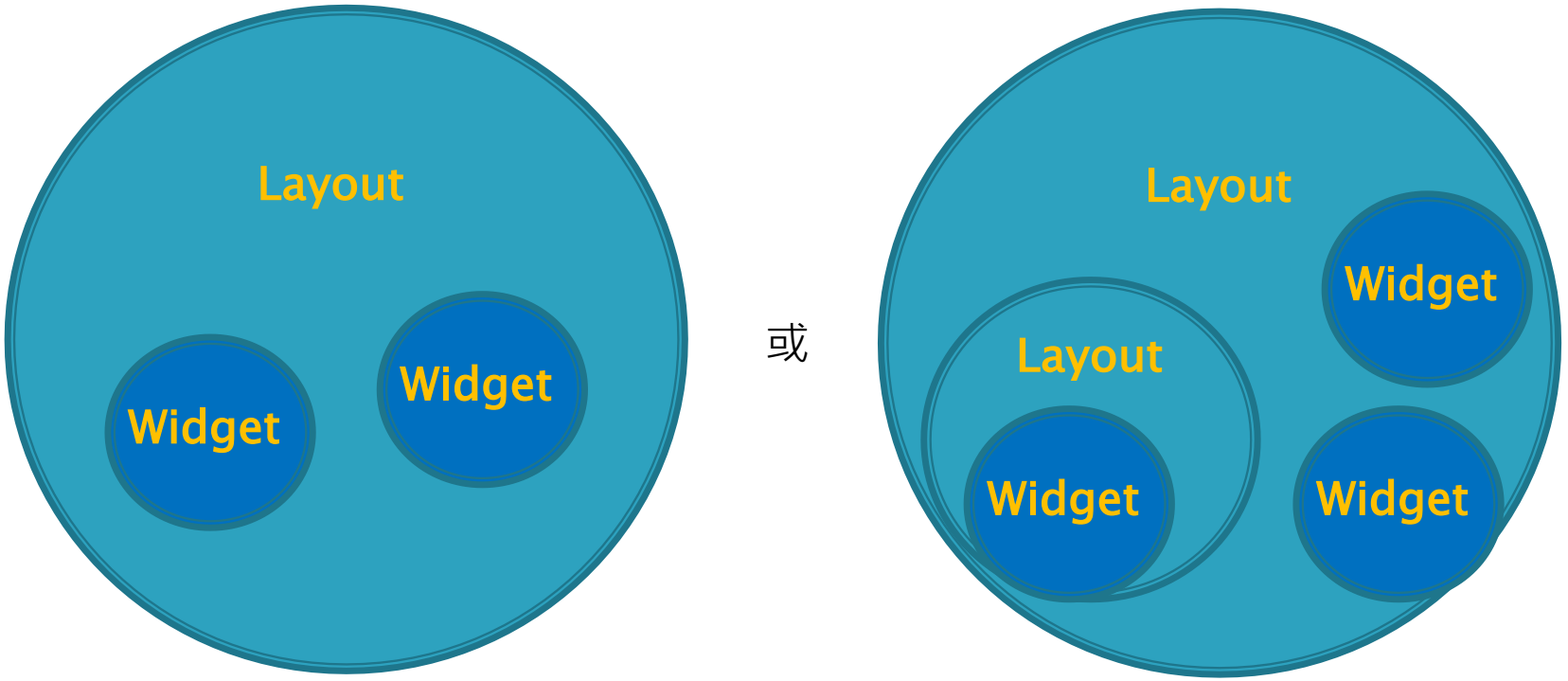
Android UI設計基本觀念 (2/5)

- UI class diagram



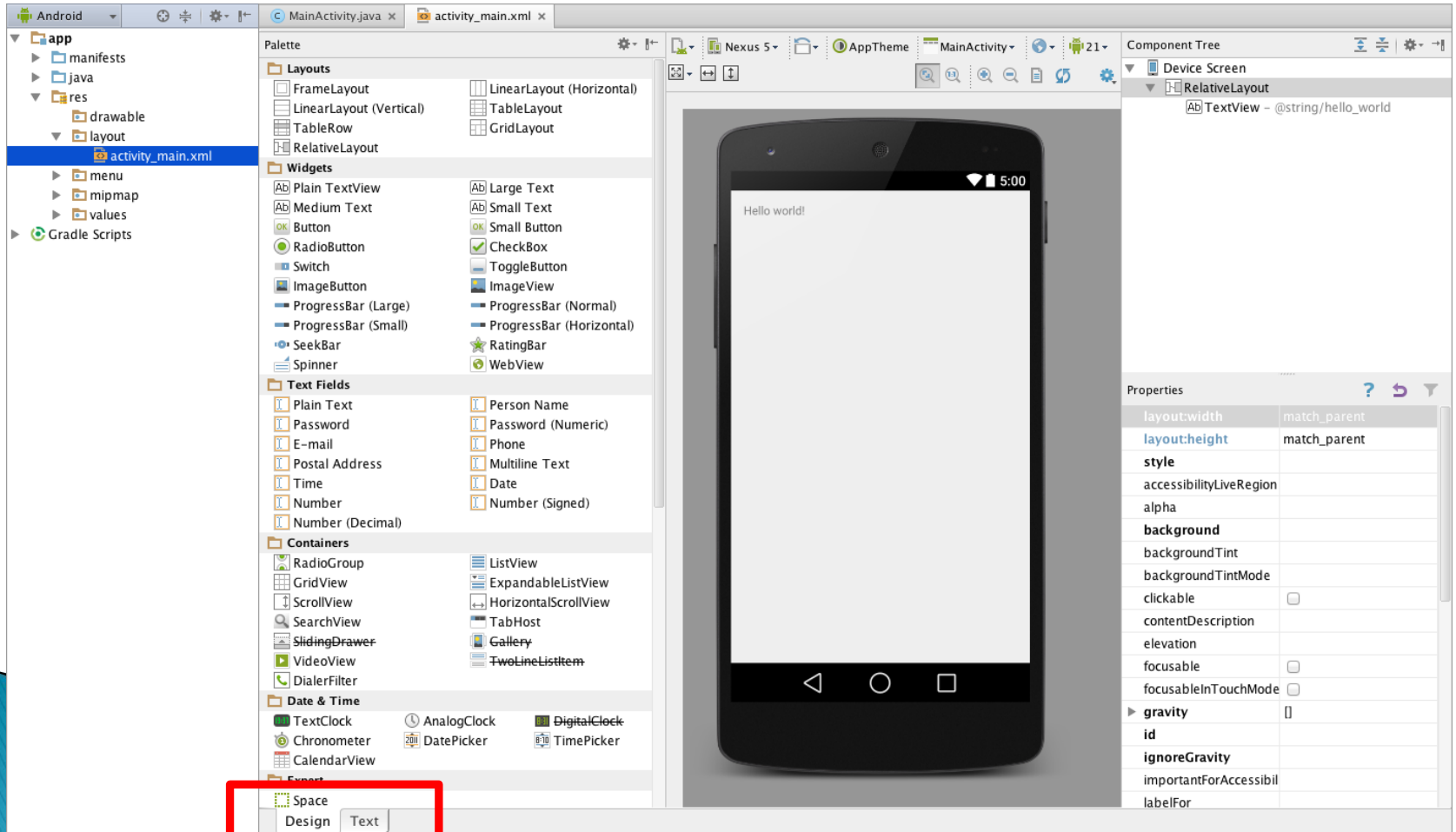
Android UI設計基本觀念 (3/5)

- Layout與Widget關係如下圖：

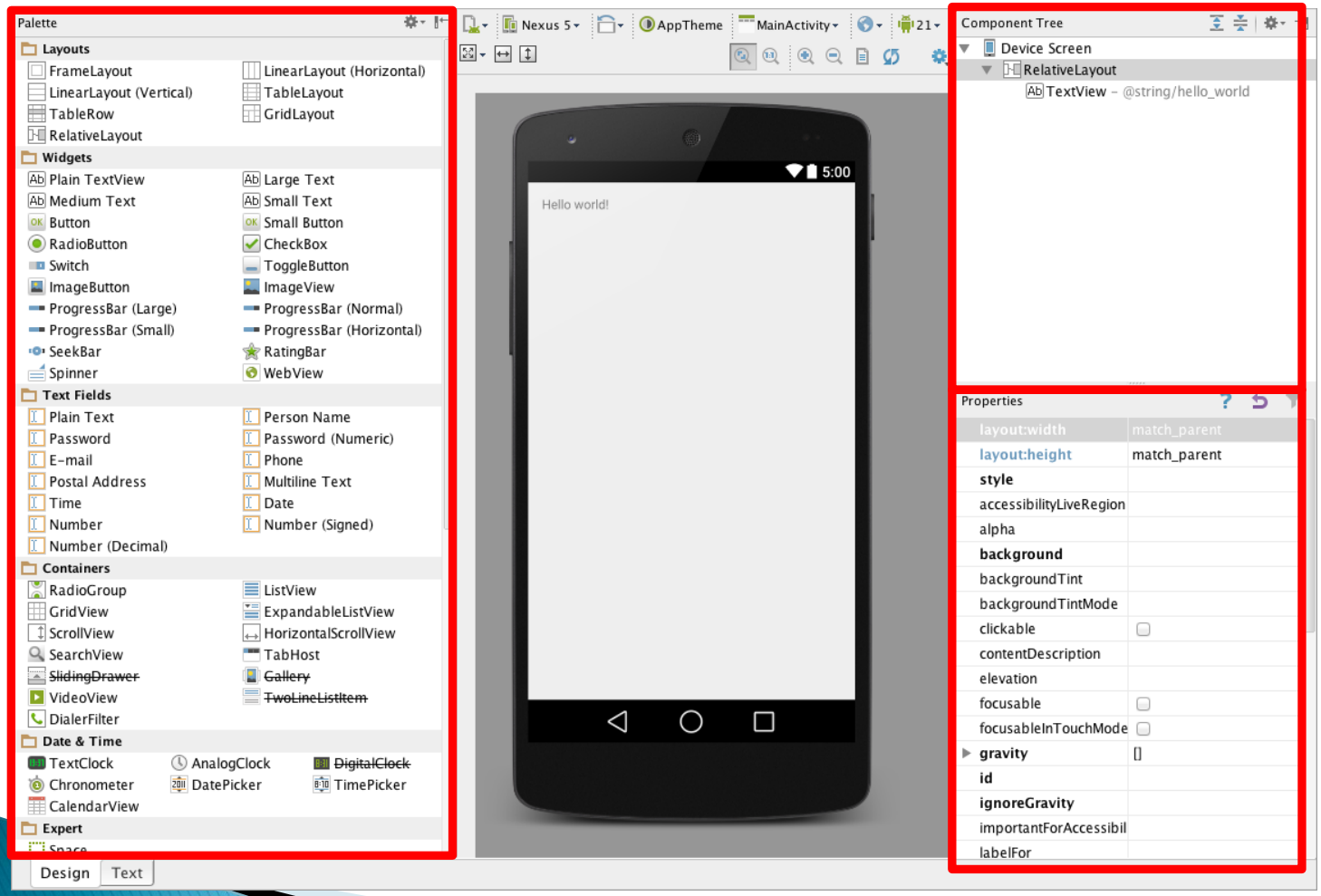


Android UI設計基本觀念 (4/5)

- 雙點layout檔案即可開啟視覺化設計工具，下方Design可拖拉元件擺設，而點選Text會開啟xml格式的原始碼設定



Android UI設計基本觀念 (5/5)



Palette
拖拉指定元件到畫面區

Component Tree
UI元件架構圖

Properties
設定元件屬性

事件處理機制

- 事件處理機制(event handling)是一種委任機制，透過設定對應的監聽器(listener)，當事件一發生，監聽器即會自動呼叫對應的處理結果，本範例以按鈕事件處理為例，有兩種實作的方式：
  - Java UI事件處理：程式碼較複雜，但方便管理與維護
  - layout檔案設定：不易於程式碼的管理與維護，但方便實作

- 建議使用第一項，Java UI事件處理，為了日後專案的管理維護與增加程式碼的可閱讀性，而這樣的做法也符合MVC設計原則

事件處理機制

- Java UI事件處理機制
  - Button物件向監聽器註冊：Button物件必須先呼叫setOnClickListener()，向OnClickListener監聽器註冊，這樣監聽器才會開始監聽
  - 實作OnClickListener的onClick()：當按鈕被按下時，OnClickListener監聽器便會知道，並自動呼叫onClick()以回應使用者按下按鈕的動作

```
btnSubmit = (Button) findViewById(R.id.btnSubmit);
btnSubmit.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        //按鈕按下後執行的內容
        String text = ((Button) v).getText().toString();
        Toast.makeText(MainActivity.this, text, Toast.LENGTH_SHORT)
                .show();
    }
});
```

範例：ButtonEx

事件處理機制

- layout檔案設定
  - android:onClick屬性指定觸擊事件發生時欲呼叫的方法

```
<Button
    android:id="@+id/btnTwo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onButtonClick"
    android:text="@string/btnTwoStr" />
```

  - 在Activity裡的方法簽章就得如以下：

```
public void onButtonClick(View v) {
    String text = ((Button) v).getText().toString();
    Toast.makeText(MainActivity.this, text, Toast.LENGTH_SHORT)
            .show();
}
```

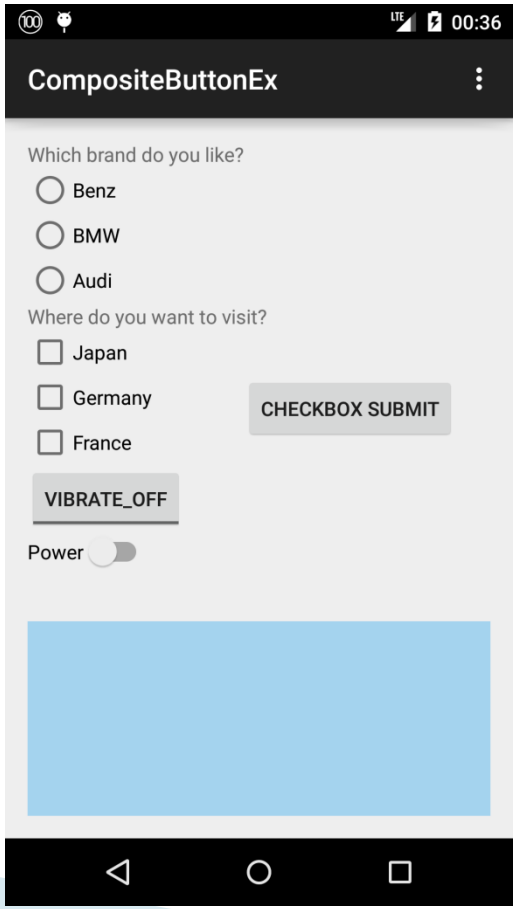
TextView與EditText

- TextView：呈現提示文字或訊息文字
- EditText：文字輸入方塊
- EditText元件的重要XML屬性
  - android:inputType=“textPassword” – 輸入文字以密碼方式呈現
  - android:hint 提示文字
  - android:maxLength 字數長度
  - android:inputType=“phone” – 只能輸入電話號碼
  - android:inputType=“number” – 只能輸入數字
  - android:inputType=“numberSigned” – 可輸入正、負整數
  - android:inputType=“numberDecimal” – 可輸入整數或小數
  - android:digits 限制能輸入的數字

範例：InputEx

Composite Button

- Radio Button
  - 單選按鈕，通常會搭配RadioGroup以達成多選一的功能
- CheckBox
  - 適合用在多選
- ToggleButton
  - 類似開關
- Switch
  - 就是開關



範例：CompositeButtonEx

RatingBar / SeekBar

- RatingBar可以讓使用者藉由點選星星數量來評分，當評分改變時會觸發事件，處理如下：
  - 呼叫setOnRatingBarChangeListener()註冊監聽器
  - 實作onRatingChanged()：當評分改變時會自動執行

- SeekBar是一種捲軸式的調整元件，事件處理如下：
  - 呼叫setOnSeekBarChangeListener()註冊監聽器
  - 實作OnSeekBarChangeListener的3個方法
    - onProgressChanged()：SeekBar值正在改變時會自動執行
    - onStopTrackingTouch()：SeekBar值結束改變時會自動執行
    - onStartTrackingTouch()：SeekBar值開始改變時會自動執行

範例：TwoBarsEx

ImageView/ImageButton

- ImageButton與Button功能大同小異，差別就在ImageButton顯示的是影像，可以跟使用者產生較多有趣的互動

- ImageView即是顯示影像給使用者，若是在程式執行過程要改變影像，可以呼叫ImageView的setImageResource()方法，該方法可以從專案的res資源載入指定的影像檔

範例：ImageEx

Layout說明

- 常見Layout如LinearLayout, RelativeLayout, FrameLayout與TableLayout等，而AbsoluteLayout在API level 3時，就已經被官方deprecated，不建議使用

- LinearLayout以線性方式呈現UI元件，也就是以水平或是垂直方式進行排列，相關XML屬性：
  - android:orientation 設定垂直或水平排列 (預設為水平)

- FrameLayout又稱為幀佈局，所有UI元件會疊在一起，layout檔案子元件最前者，顯示時會在最下層

- RelativeLayout以相對位置來呈現UI元件，相關XML屬性：
  - android:layout_above 放在給定元件的上方
  - android:layout_alignBottom 下邊對齊給定元件的下邊
  - android:layout_alignParentBottom 下邊對齊父元件的下邊
  - android:layout_centerHorizontal 放在父元件的水平中央
  - android:layout_centerInParent 放在父元件的正中央

範例：LayoutEx

ScrollView

- ScrollView是FrameLayout的子類別，可以使用捲動的方法瀏覽資料
- 通常會在ScrollView裡再置入其它layout元件，常用如LinearLayout，當資料超過行動裝置的實際螢幕時，即可用捲動的方式瀏覽
- XML的相關屬性：
  - android:scrollbars
    - none：不顯示捲軸
    - vertical：顯示垂直捲軸

範例：ScrollViewEx

Style/Theme (1/2)

- Style就是將跟UI外觀有關的屬性從layout檔獨立出來並放在一個<style>標籤裡，以方便重覆給各種UI元件套用，包括寬、高、文字顏色與大小等等

- 這些style設定都會放在一個styles.xml檔案，而這個檔案要放在res/values目錄內

- 相較與style是套用在各UI元件，Theme則是套用整個應用程式或是Activity，所以在設定上較style多且複雜，但Theme其實也是多個sytle的集合

Style/Theme (2/2)

- Style/Theme都提供類似Java繼承(inheritance)的功能，<style>標籤搭配parent屬性即可繼承指定的父Style/Theme，而子Style/Theme也能自行新增自己的屬性樣式或改變父Style/Theme的樣式

- Theme的套用較為複雜，一般都是採用SDK提供的預設Theme，如要指定應用程式或是Activity的Theme，需到AndroidManiFest.xml檔進行設定

範例：StyleThemeEx

WebView

- 瀏覽網頁使用WebView元件，說明如下：
  - layout檔案建立WebView
  - 讓JavaScript語法可以在WebView元件上運行
    webView.getSettings().setJavaScriptEnabled(true);
  - 設定WebView元件欲連結的網址
    webView.loadUrl("http://www.developer.android.com");
- 瀏覽網頁必須使用到Internet，設定android.permission.INTERNET
- 要使用自製的WebView來處理URL請求，必須改寫WebViewClient的shouldOverrideUrlLoading()，並呼叫setWebViewClient設定
- 建議對返回鍵的按下事件加以處理

範例：WebViewEx