Android 基礎教學

吳柏翰

FreeIdea-Tech

Outline

- Android歷史起源
- Android優勢與市場未來趨勢
- 安裝Android SDK
- 建立專案
- 模擬器的執行與操作
- 使用Log及IDE除錯工具
- 實機運作

FreeIdea-Tech

Android歷史起源

- 起源
  - 隨著手機與網際網路結合，上網普及化之後，使用者對手機的功能有截然不同的需求。
  - 透過網際網路，各種新的服務與應用應運而生，使用者也希望手機平台能像電腦一樣，讓使用者能發揮設計創意，自主地開發服務。
  - 因此智慧型手機隨之上市，於是微軟公司以其在電腦終端作業系統霸主的地位，推出智慧型手機專用之WinCE及Window Mobile作業平台。

FreeCarTech

Android歷史起源

- 由於微軟公司的作業系統並不開放原始程式碼，因此除有版權問題外，也因無法深入作業系統核心而限縮了應用軟體發展的空間。

- 有鑑於此，以開放程式碼Linux作業系統為基礎發展Android手機平台的構想應運而生。Android字面上的意義是指科技小說裏面的機器人，隱含有新奇、能接受指示，且具有智慧的意思。

- 它是Google在2007年鼓吹它為手機系統平台，初期由Google開發，後由開放手機聯盟（Open Handset Alliance）接手後續發展推廣。

Energia-Tech

Android歷史起源

- 由於Android開放式架構的特性，開發人員可以很容易地進行系統擴充。

- Android平台具有以下特點：
  - 系統的開放性
  - 應用程式的對等性
  - 跨應用程式的整合性
  - 開發環境的效率性

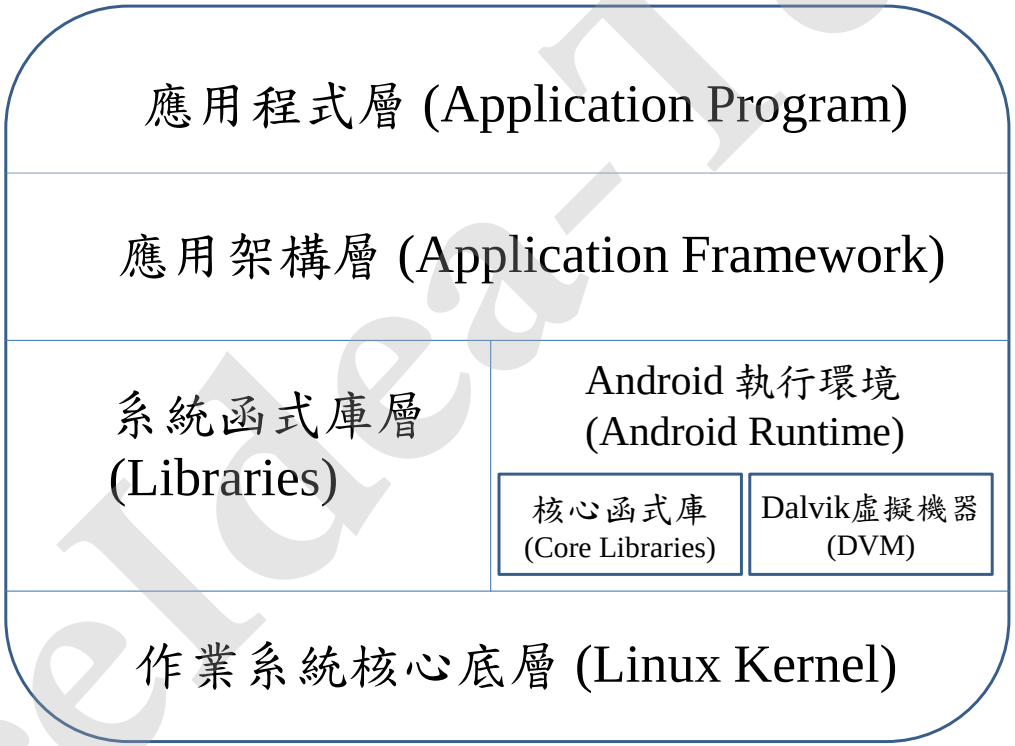
Free Tea-Tech

Android歷史起源

- 架構簡介
  - Android使用堆層（Stack）做為軟體平台的建構方式。
  - 堆疊主要分為四層，分別為作業系統核心層、系統函式庫層、應用架構層、及應用程式層，如下頁圖所示。
  - 作業系統核心層以Linux核心工作為基礎，提供作業系統基本功能及一些硬體驅動程式。

Freida-Tech

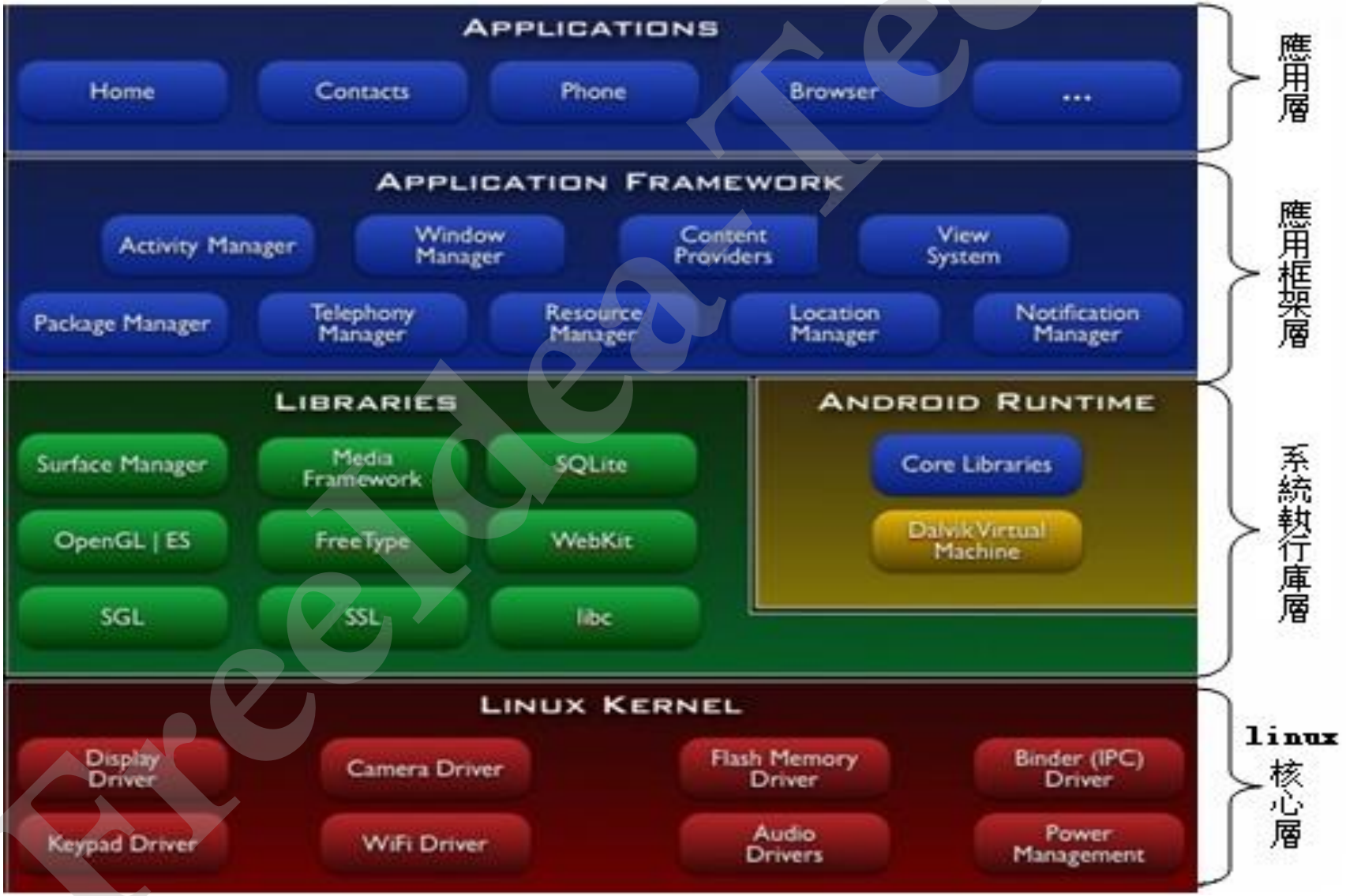
Android歷史起源



Android平台架構圖

FreeIdea-Tech

Android歷史起源



FreeIdeaTech

Android歷史起源

- 要注意到的一點是，目前Android的Linux核心與常用的GNU/Linux並不完全相同，GNU/Linux有的東西在Android Linux核心中並不一定有。

- 為提供在手機環境使用，Android的Linux核心特別加強了程式間通訊(Inter Process Communication, IPC)及電源管理(Power Management)的功能。

FreeIdea-Tech

Android歷史起源

- 作業系統核心底層之上是函式庫及Android 執行環境(Android Runtime)，函式庫有許多開放原始碼的有用函式，例如，OpenGL、libc (bionic)、WebKit等等。

- 而Android Runtime主要包括核心函式庫(Core Libraries)及Android專有之虛擬機器Dalvik (Dalvik Virtual Machine, DVM)。

- 應用架構層主要是提供Android應用程式的API，包括位置管理、資源管理、電話管理、提示管理等一些軟體套件，應用架構層再透過JNI去呼叫下層函式庫。

Fredea-Tech

Android歷史起源

- 若有新的函式加到函式庫裏，則應用架構層就需要加以擴充。透過應用架構層的規範，可以對軟體函式的增加做較好的管理。

- 有了上述三層的支援，系統已將硬體元件抽象化(Hardware Abstraction)，並以層次的架構提供各種有用且方便取用的函式庫，再經由其DVM建構的跨平台Java開發環境，使用者即可在應用程式層設計Android平台上的各種好玩的應用服務。

Free Idea-Tech

Android歷史起源

- 系統特性
  - 應用程式框架
    - 支援元件的重用與替換
  - Dalvik虛擬機
    - 專門爲移動設備做了優化
  - 內部整合瀏覽器
    - 該瀏覽器基於開源的WebKit 引擎
  - 優化的圖形庫
    - 包括2D和3D圖形庫，3D圖形庫基於OpenGL ES 1.0 (硬體加速可選)
  - SQLite
    - 結構化的資料存儲
  - 多媒體支援
    - 包括常見的音效、視訊和靜態印象檔格式
    - MPEG4、H.264、MP3、AAC、AMR、JPG、PNG、GIF
  - 豐富的開發環境
    - 模擬器，除錯工具，記憶體及性能分析，和Eclipse整合開發環境

Fresidea-Tech

Android優勢與市場未來趨勢

- 開發工作
  - Android移植開發的最終目的是讓手機或其他移動產品能夠執行強大的Android系統，使之成爲一個移動作業平台。
  - 主要的工作集中在以下兩個方面：
    - Linux中的相關設備驅動程式
    - Android本地框架中的硬體抽象層

FreeIdeaTech

Android優勢與市場未來趨勢

- 系統開發
  - Android系統開發的一個比較典型的實例就是當系統需要某種功能時，爲了給Java層次的應用程式提供讀取的介面，需要從底層到上層的整體開發，步驟如下：
    - 增加C或者C++本地庫(如底層driver)
    - 定義Java層所需要的類(系統API)
    - 將所需要的程式碼封裝成JNI
    - 結合Java類和JNI
    - 應用程式讀取Java類(上層應用)

FreeLearTech

安裝Android SDK

- Only Idea
  - Android 提供免費且跨平台的整合開發環境，只要電腦能連接上網路，我們隨時都能下載相關工具下來，並開始開發 Android 應用程式。
  - 除了不用擔心平台相容性問題，Android也提供相當容易的開發工具，讓任何使用者都可以更專心於程式設計上。

FreeIdeaTech

安裝Android SDK

- 支援的作業系統
  - Windows XP 或 Vista
  - Mac OS X 10.4.8 或之後版本 (適用 x86 架構的 Intel Mac)
  - Linux (Ubuntu 6.10)
- Android開發工具
  - JDK 6
    - Java程式語言開發工具
    - http://java.sun.com/javase/downloads/index.jsp
  - Android SDK 1.5r2
    - Android 程式開發套件，包含 Android 手機模擬器(Emulator)。
    - http://developer.android.com/sdk/
  - Eclipse 3.4.2 IDE (Ganymede)
    - 程式撰寫工具
    - http://www.eclipse.org/downloads/
  - ADT 0.9.1
    - Eclipse Android 擴充套件
    - http://developer.android.com/sdk/adt_download.html

Free Tea-Tech

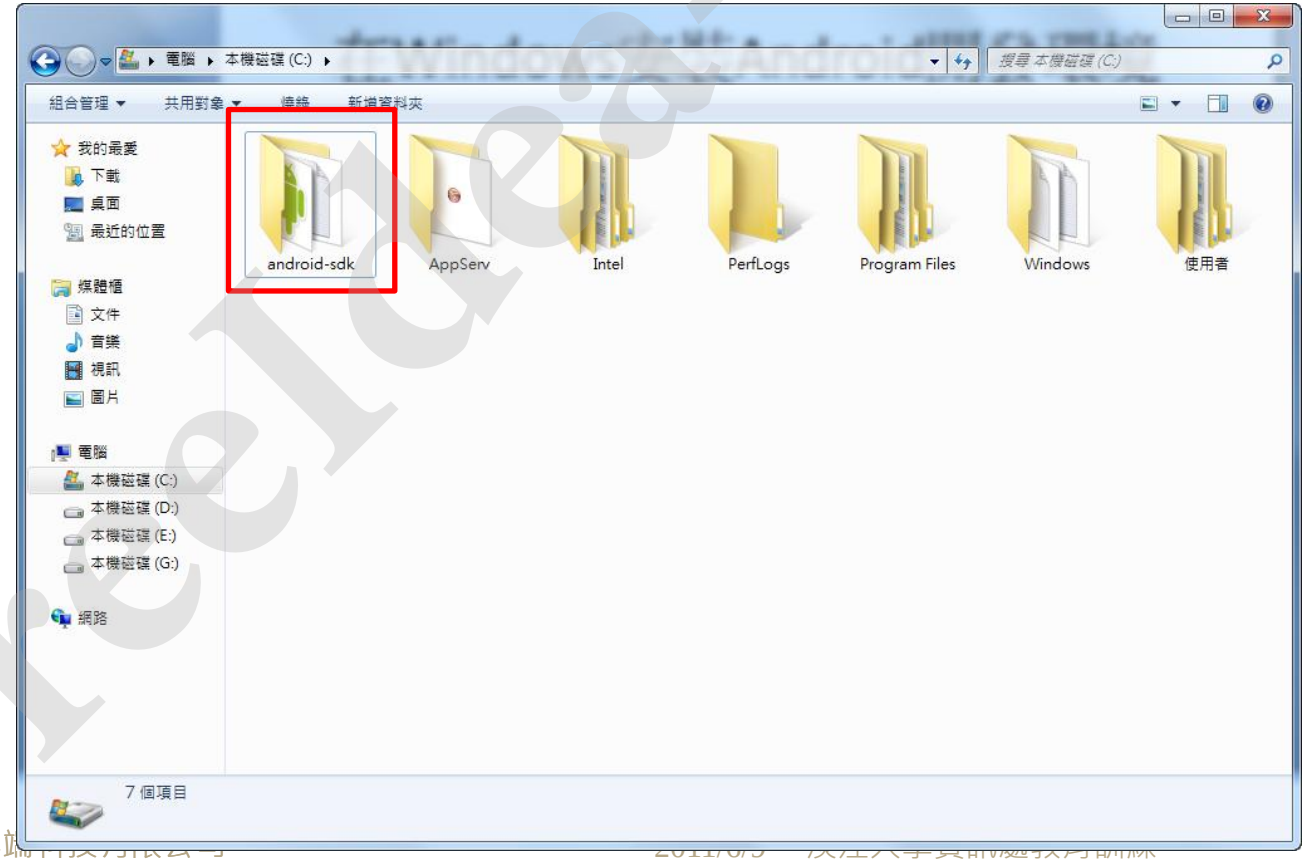
安裝Android SDK

- 安裝步驟
  - 安裝Java (JDK 6)
  - 安裝Eclipse ADT擴充套件
    - 更新 ADT 擴充套件
      - Help → Software Update
      - 選擇安裝方式
      - Available Software →Add Site → Local or 輸入網址
      - http://dlssl.google.com/android/eclipse/site.xml
    - 以Local 方式安裝
      - 解壓縮ADT-0.9.1 →尋找解壓縮資料夾→點選OK搜尋ADT套件
    - 安裝ADT套件
    - 安裝ADT套件
      - 勾選I accept the … → 點選Finish開始安裝
    - 檢查ADT是否安裝成功
      - Window → Preferences → 檢查Android標籤是否出現

FreeIdea-Tech

安裝Android SDK

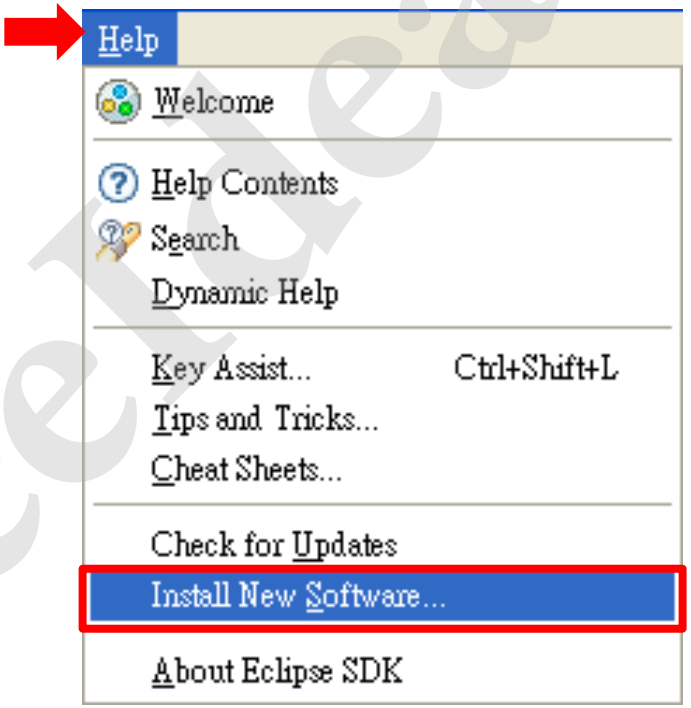
- 安裝Android SDK
  - 解壓縮後重新命名為”android-sdk”，SDK資料夾可放於任意位置，故本教材統一放於C槽底下



FreeEagle Tech

安裝Android SDK

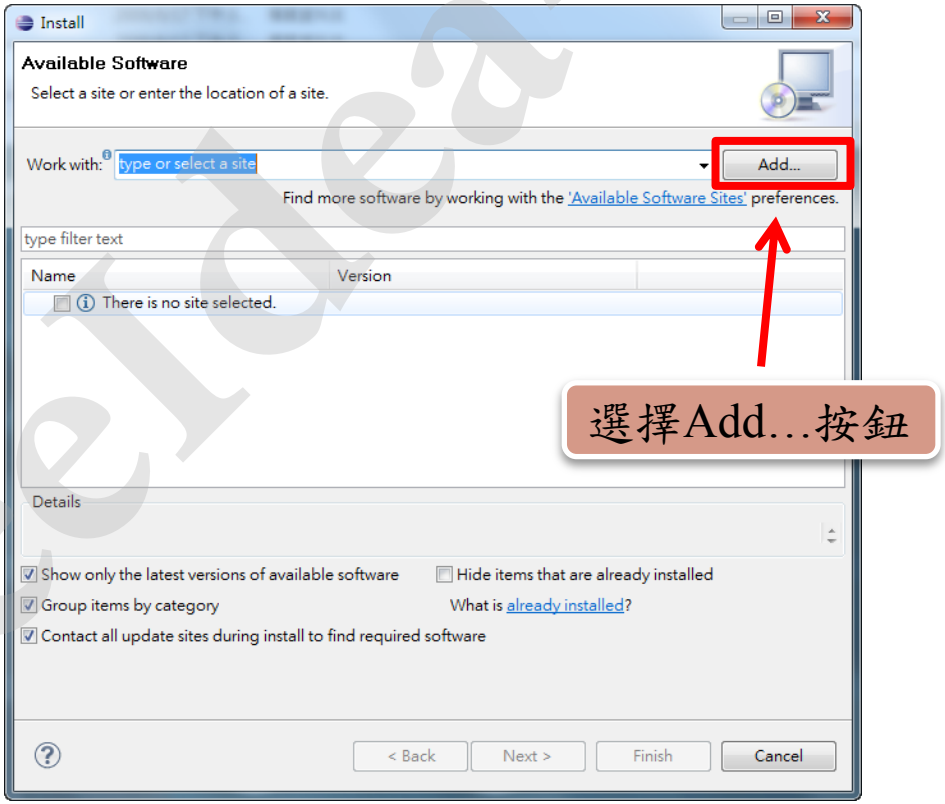
- 安裝Android Development Tools (ADT)
  - 開啟Eclipse
  - 選取Help → Install New Software…



FreeIdea-Tech

安裝Android SDK

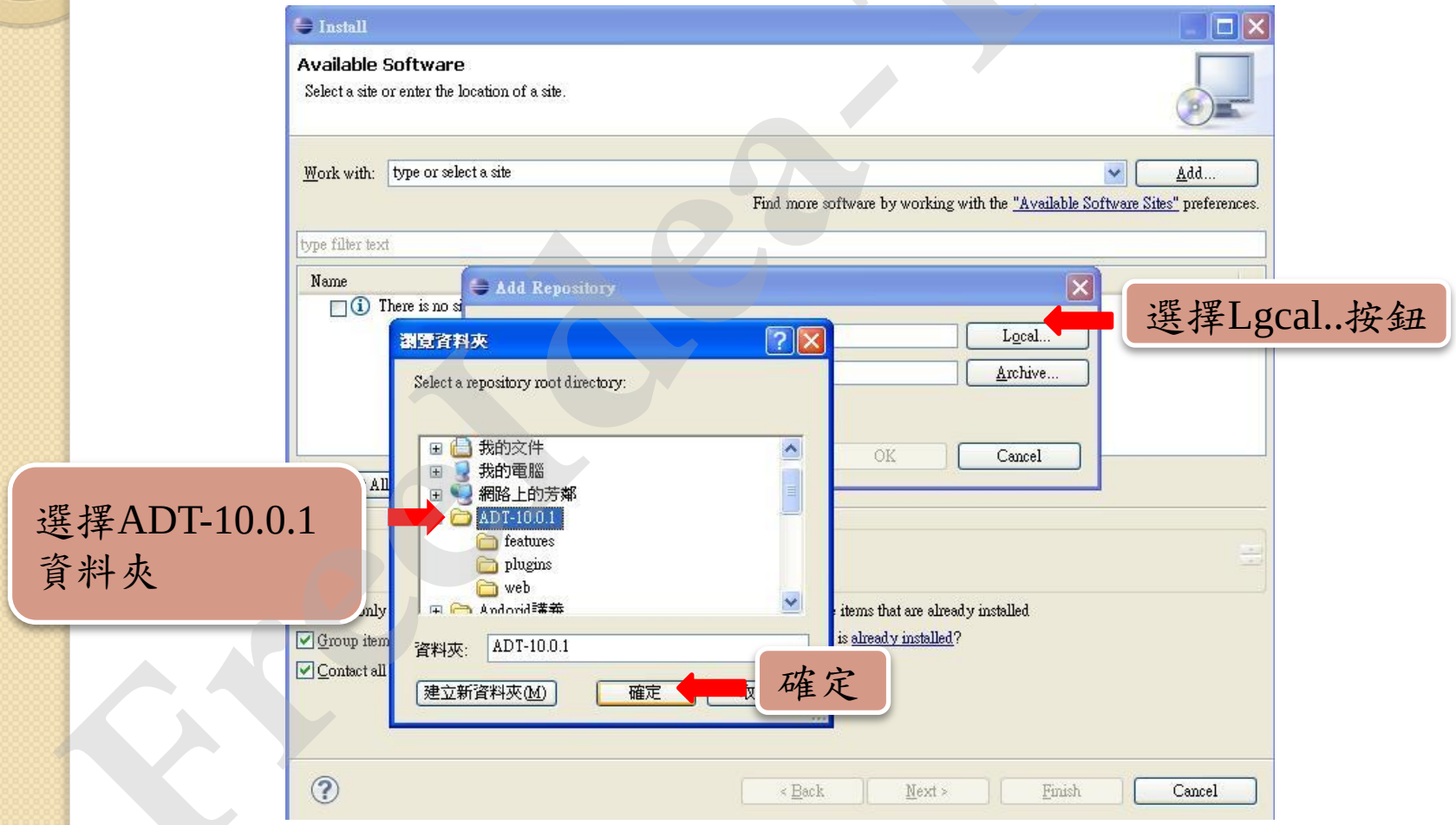
- 安裝Android Development Tools (ADT)
  - 開啟Eclipse
  - 選取Help → Install New Software…



FreeIdea Tech

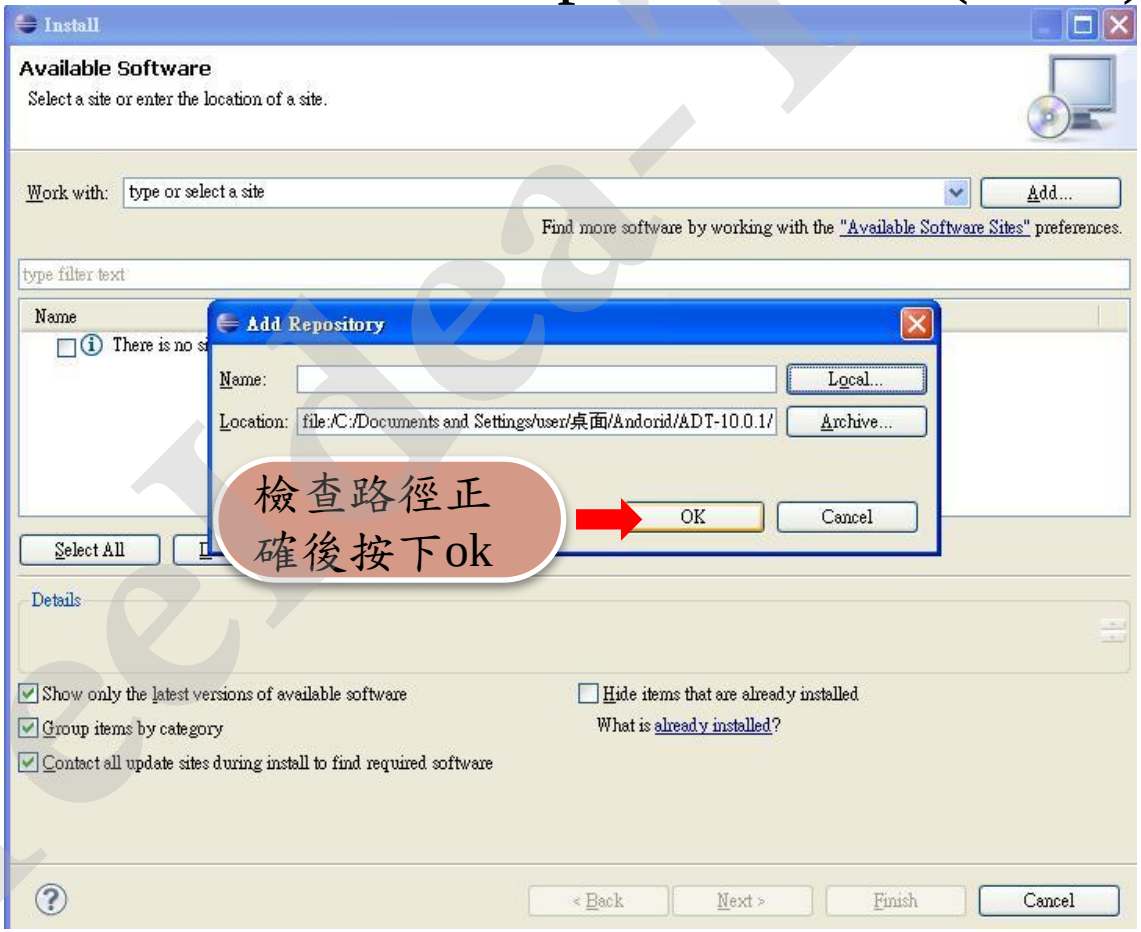
安裝Android SDK

- 安裝Android Development Tools (ADT)



安裝Android SDK

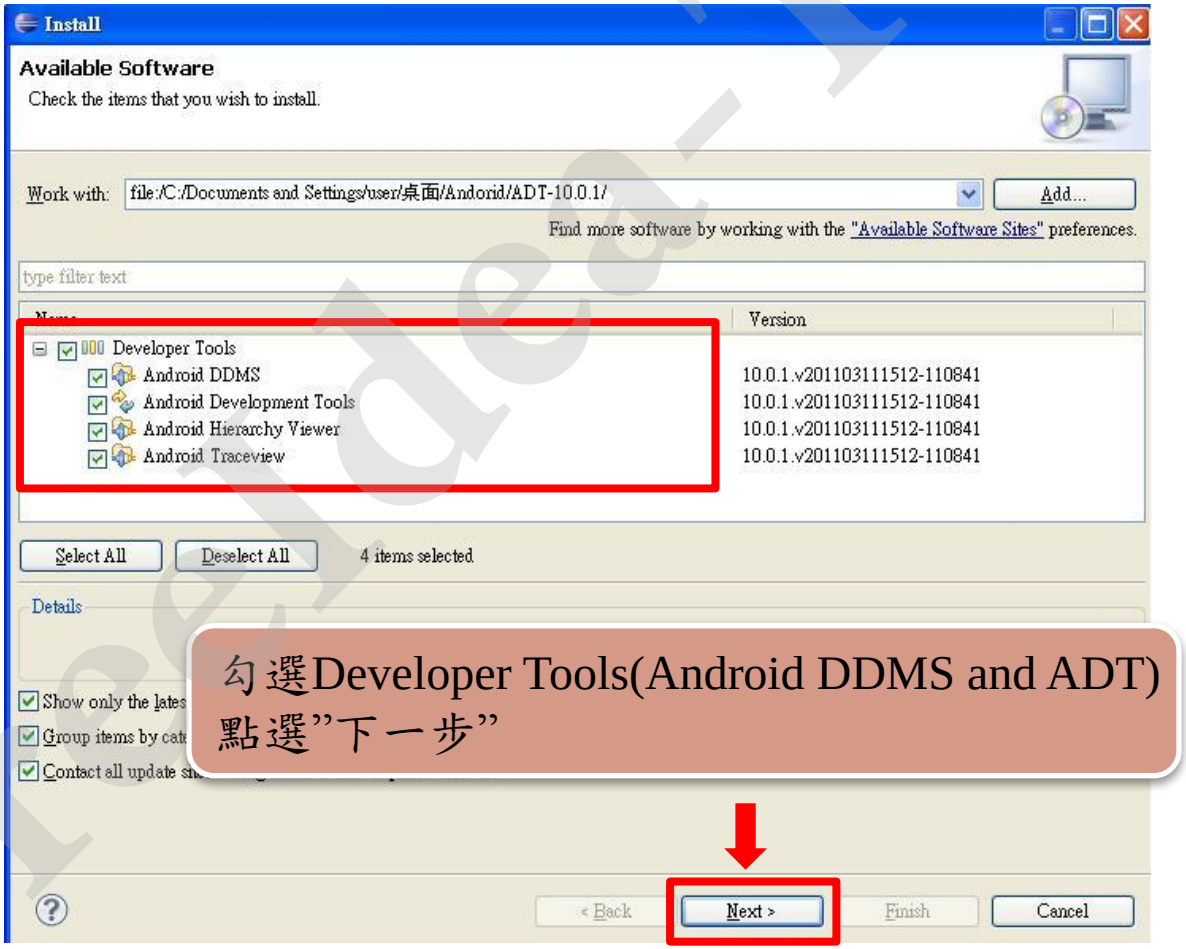
- 安裝Android Development Tools (ADT)



FreeTea-Tech

安裝Android SDK

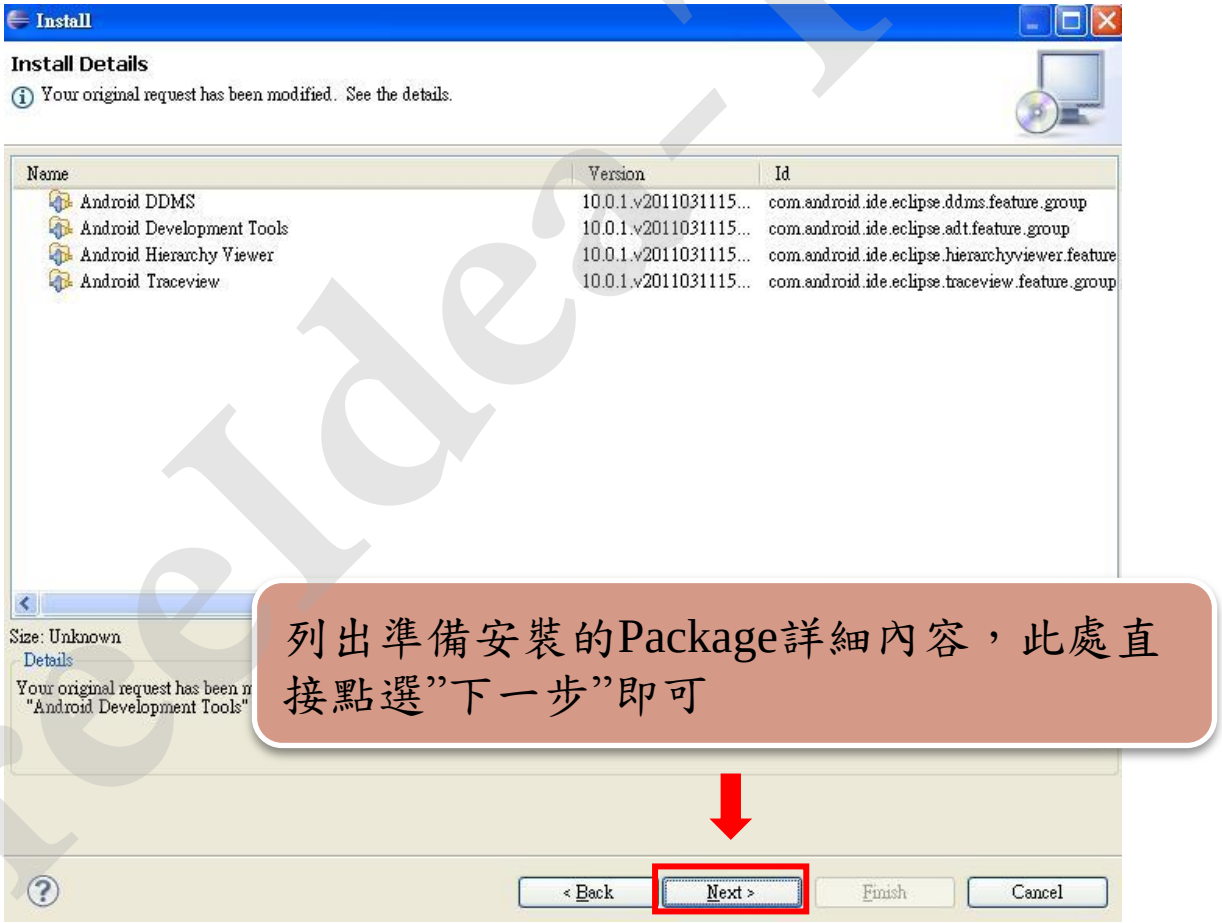
- 安裝Android Development Tools (ADT)



Freelea-Tech

安裝Android SDK

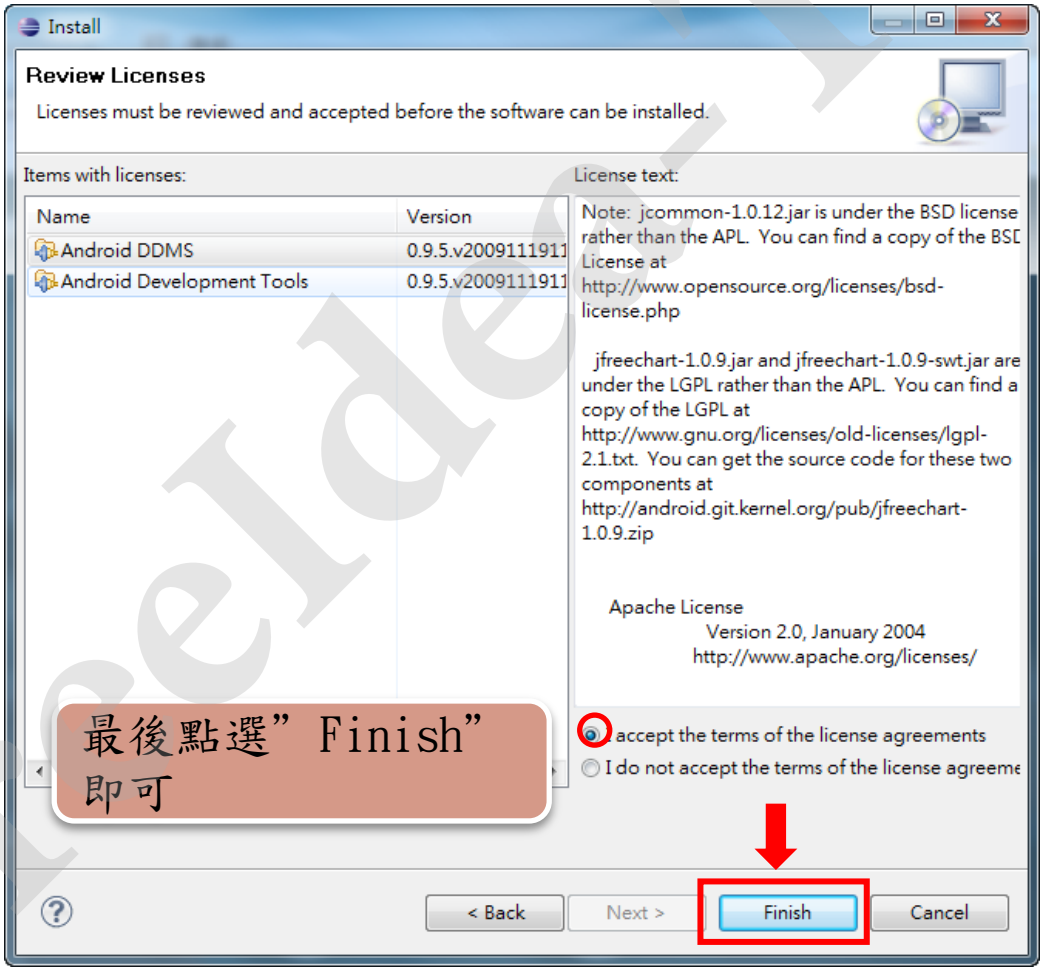
- 安裝Android Development Tools (ADT)



FreeIdea-Tech

安裝Android SDK

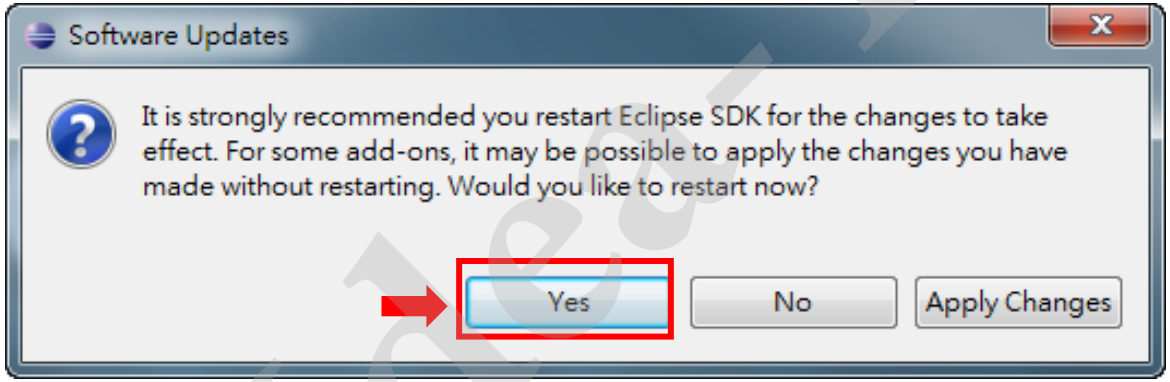
- 安裝Android Development Tools (ADT)



FreeIdea Tech

安裝Android SDK

- 安裝Android Development Tools (ADT)

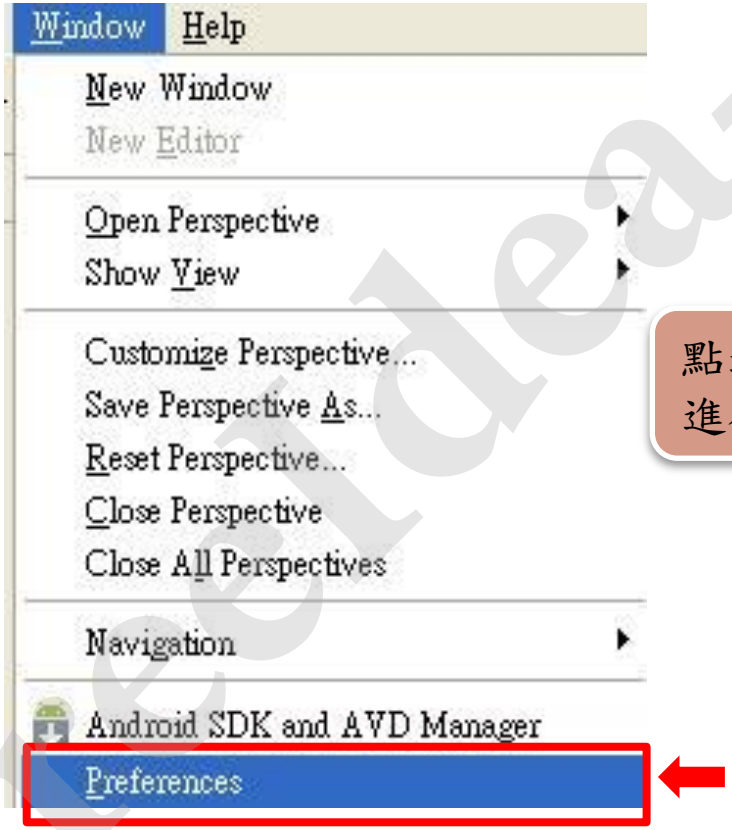


安裝完成後需要重新啟動Eclipse

FreeIdea Tech

安裝Android SDK

- 設定Android SDK細項



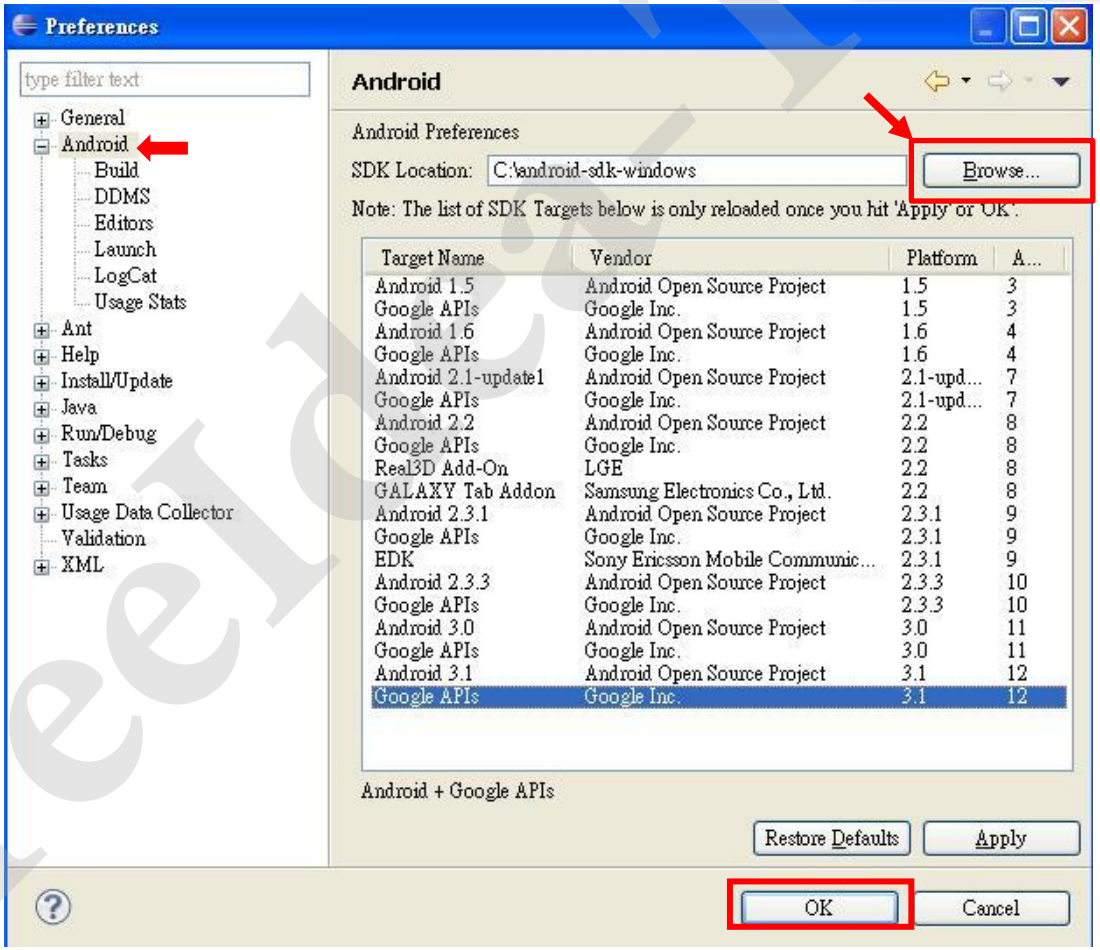
點選 Window → Preferences
進行Android SDK細項設定

Freeidea-Tech

安裝Android SDK

- 設定Android SDK細項

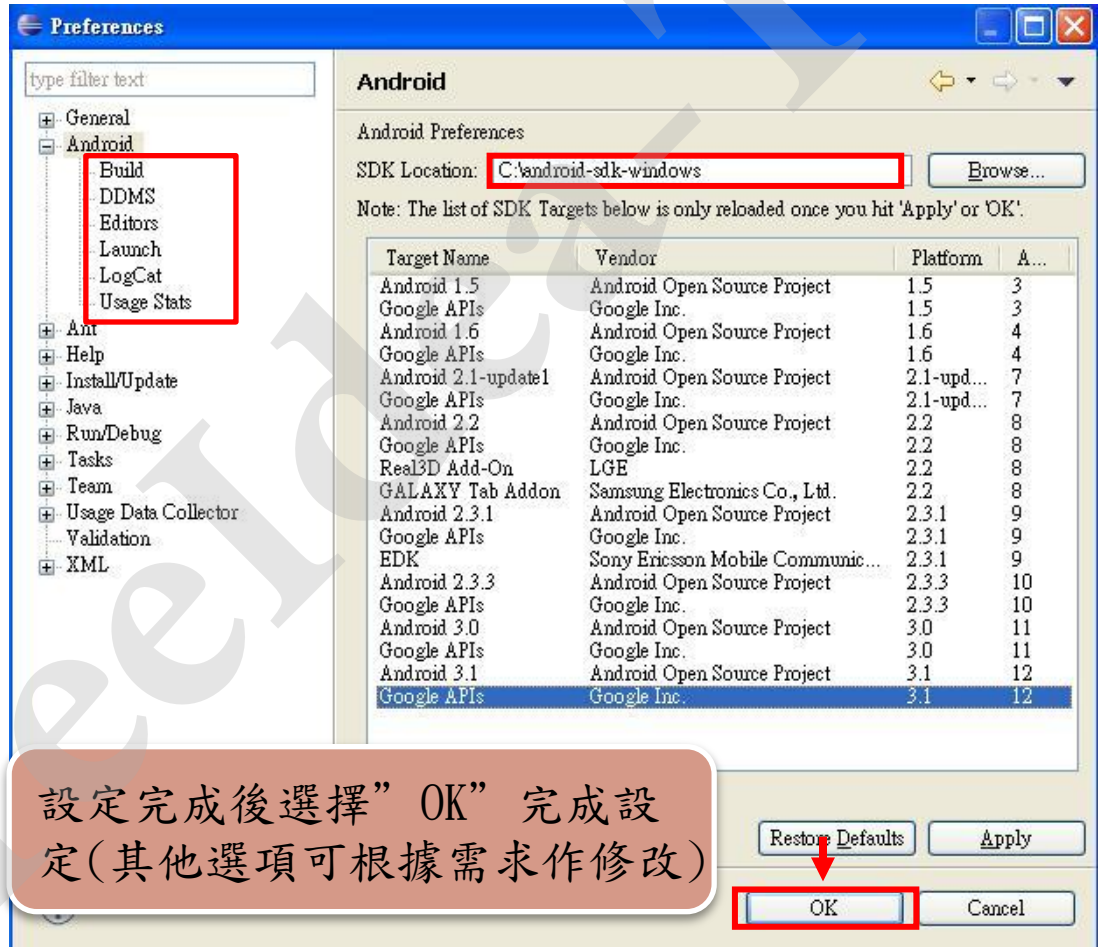
點選“Browse…”並選擇Android SDK存放位置



FreeIdea Tech

安裝Android SDK

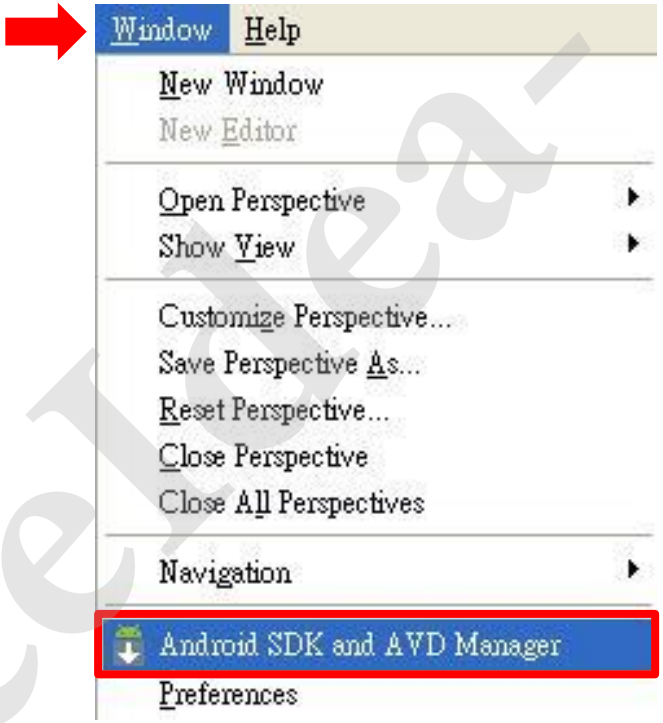
- 設定Android SDK細項



FreeIdea Tech

安裝Android SDK

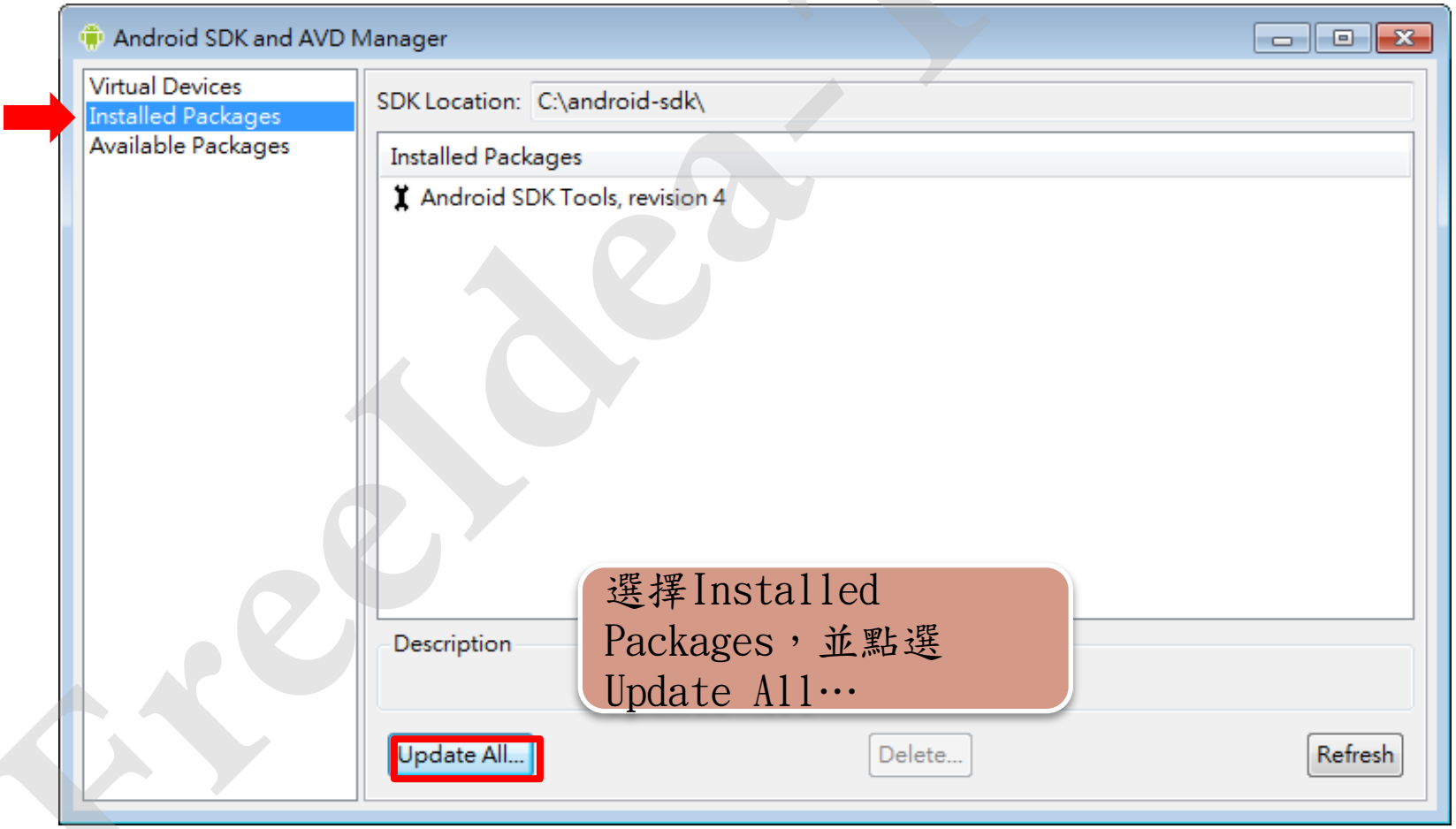
- 安裝 Android SDK



FreeIdea-Tech

安裝Android SDK

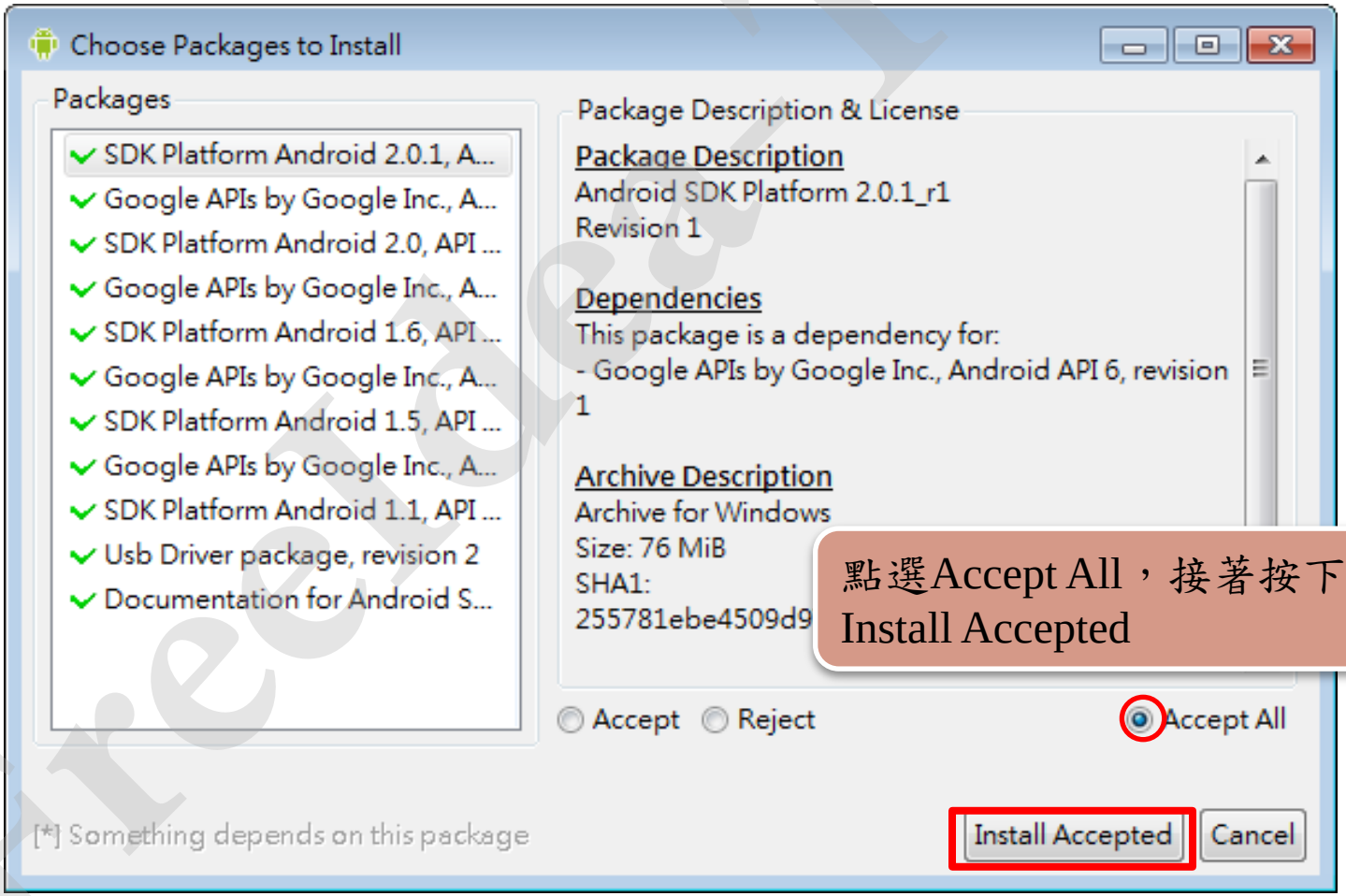
- 安裝 Android SDK



FreeIdea-Tech

安裝Android SDK

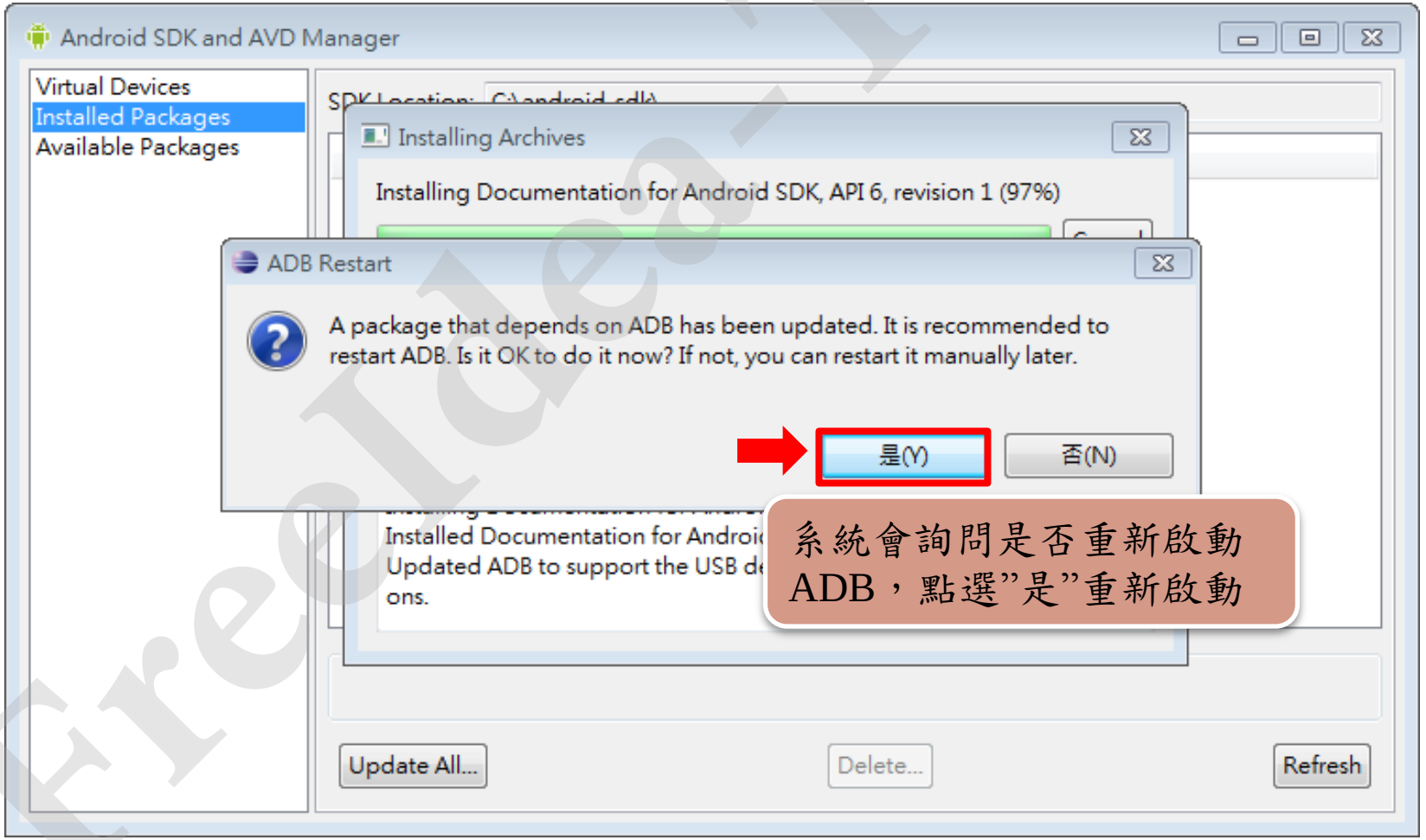
- 安裝 Android SDK



Freelance Tech

安裝Android SDK

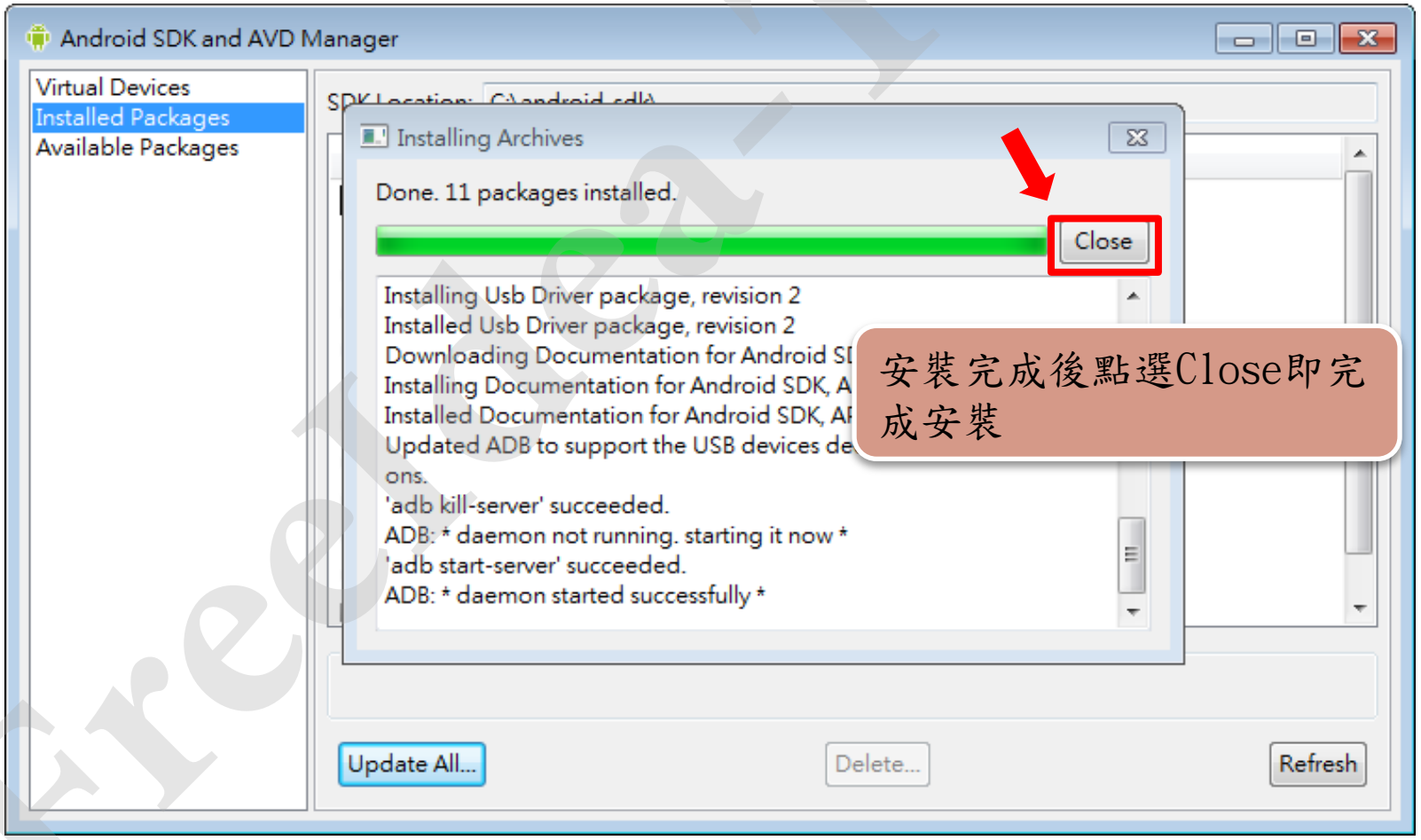
- 安裝 Android SDK



FreeIdea-Tech

安裝Android SDK

- 安裝 Android SDK



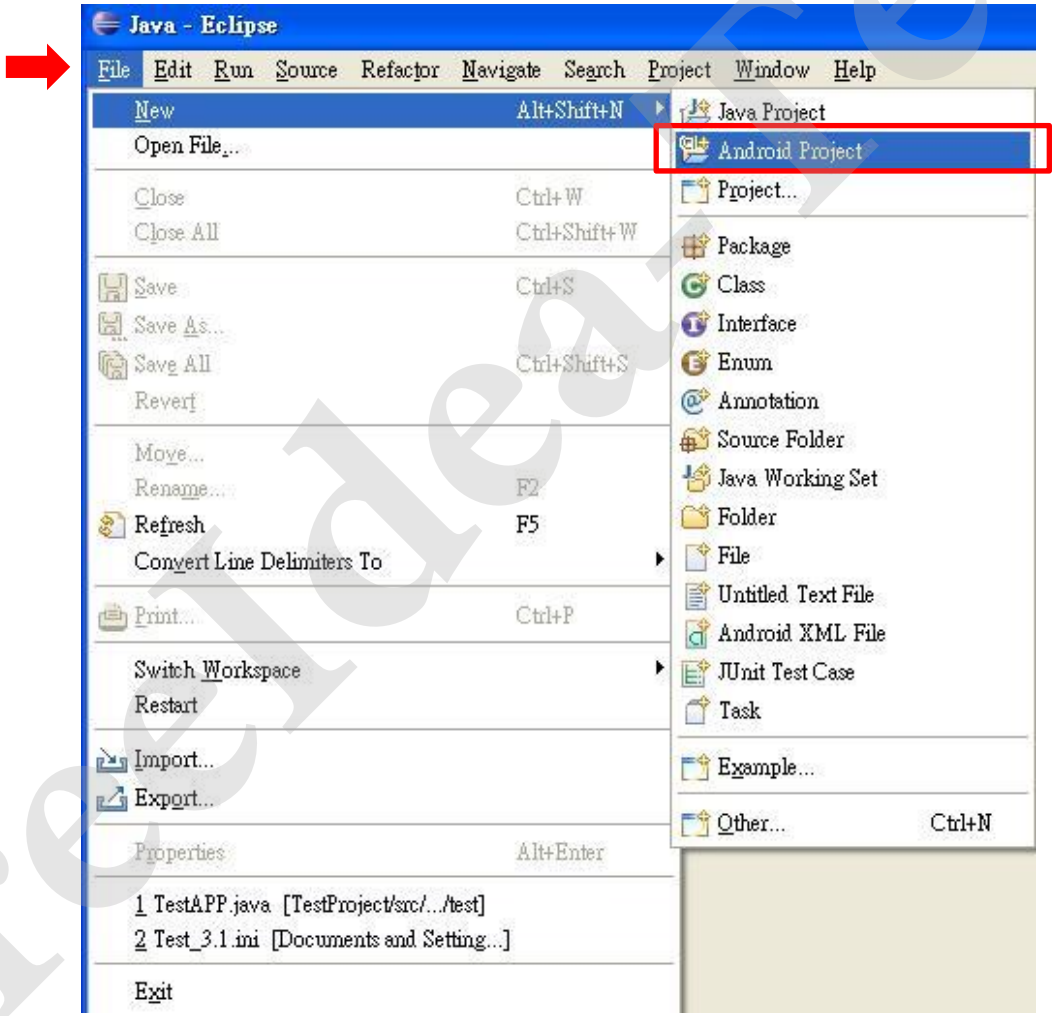
FreeTea-Tech

建立專案

- 建立Android專案
  - File → Other…
  - 選擇Android Project → Next進入設定頁面
- 設定Android專案
  - 選擇Build Target
    - 選擇用來編譯專案的 SDK 版本
  - 輸入Project name
    - 自訂Android專案名稱
  - 輸入Application name
    - 自訂顯示在應用程式上的標題
  - 輸入Package Name
    - 套件(Package)名稱，用於辨別不同類別(class)。
  - 輸入Create Activity
    - 建立並自訂Activity 子類別名稱，Activity用於啟動程式和控制程式流程，或是根據需要控制螢幕、界面。
  - Min SDK Version
    - 專案所支援的最低 SDK 版本。
- Debug編譯器相關設定
  - Debug → Debug Configurations
  - 點兩下Android Application
- Debug相關設定
  - 選擇Target標籤 → 勾選剛剛建立的模擬器名稱 → Apply
- 編譯器測試
  - Debug → 點選剛剛建立的編譯器

Freedea-Tech

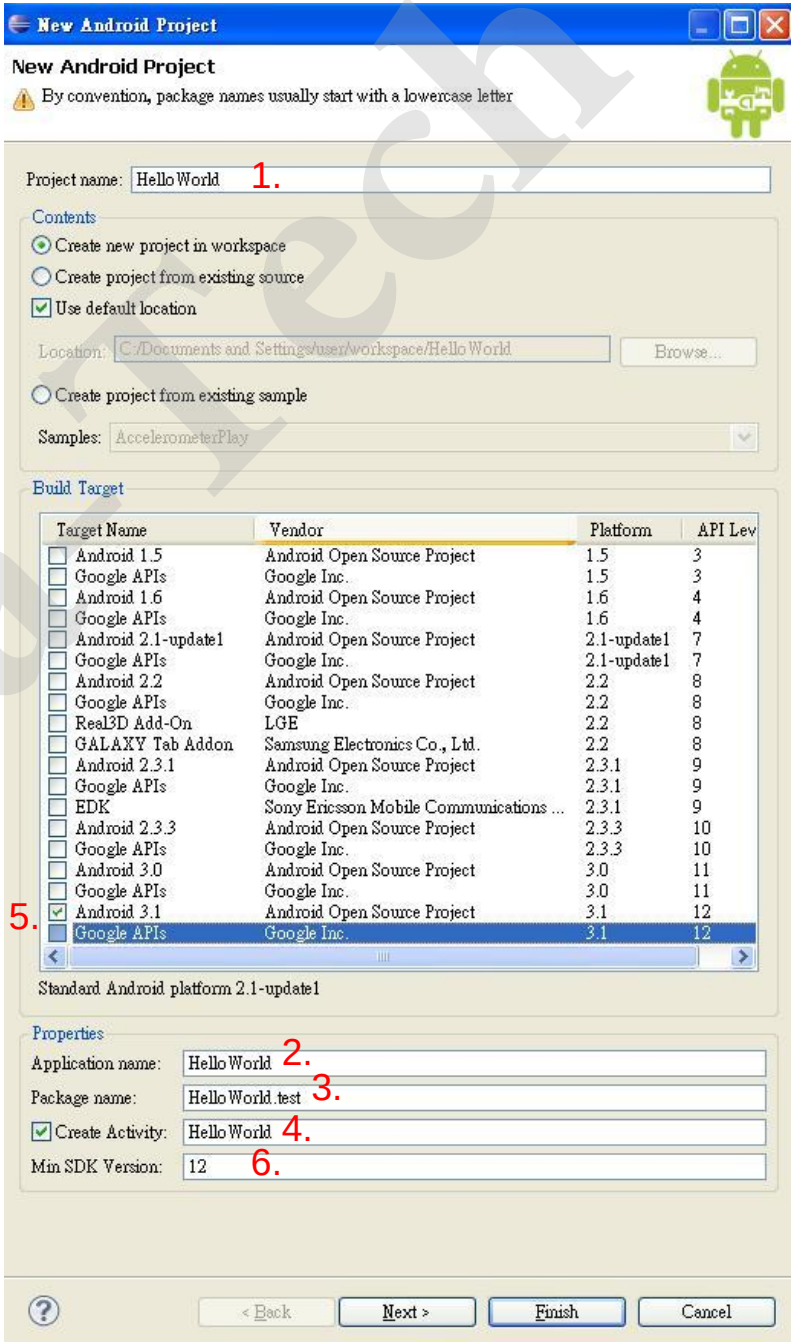
建立專案



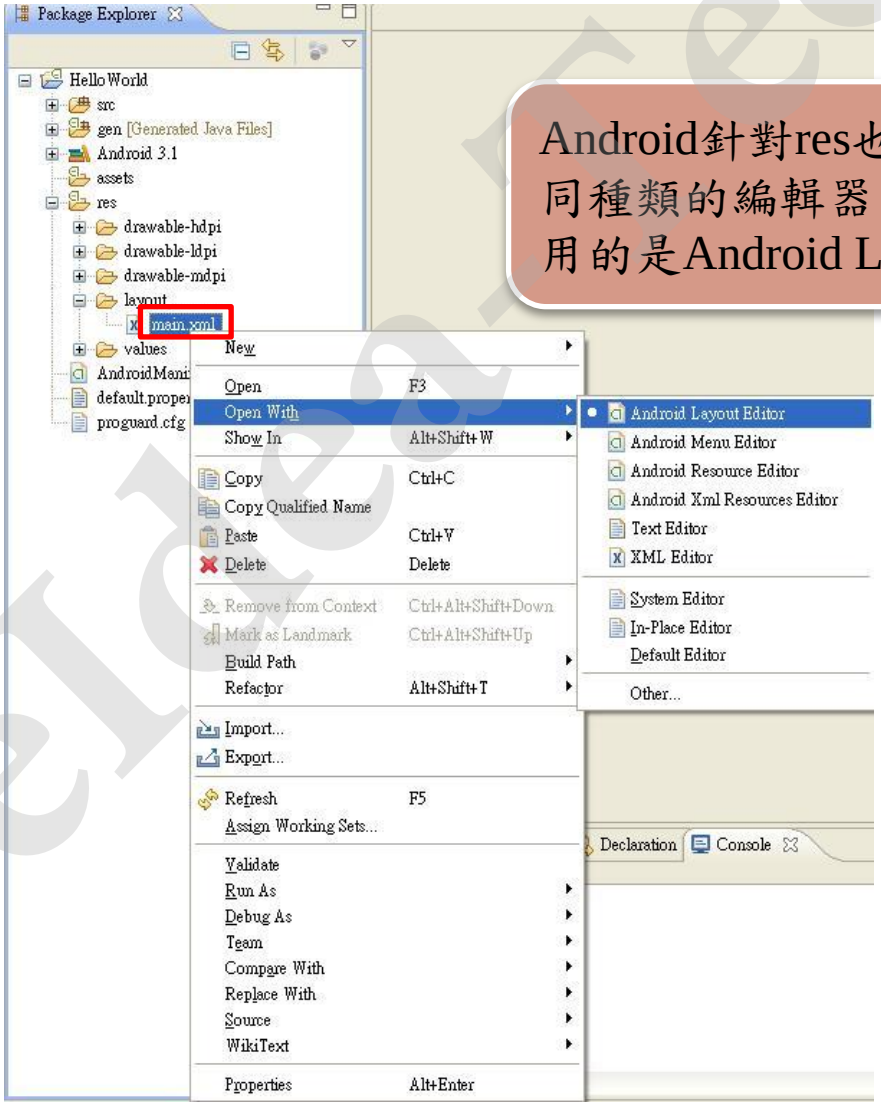
FreeIdeaTech

建立專案

名稱	描述
1 Project Name	專案資料夾名稱。
2 Application Name	應用程式標題。
3 Package Name	套件名稱，例如: java.net、java.io，可自行命名。
4 Create Activity	是否建立這個是項目的主要類別。
5 Build Target	選擇編譯專案的SDK版本，選擇後會填入下列的Min SDK Version中。
6 Min SDK Version	應用程式支援的最低 SDK 版本。



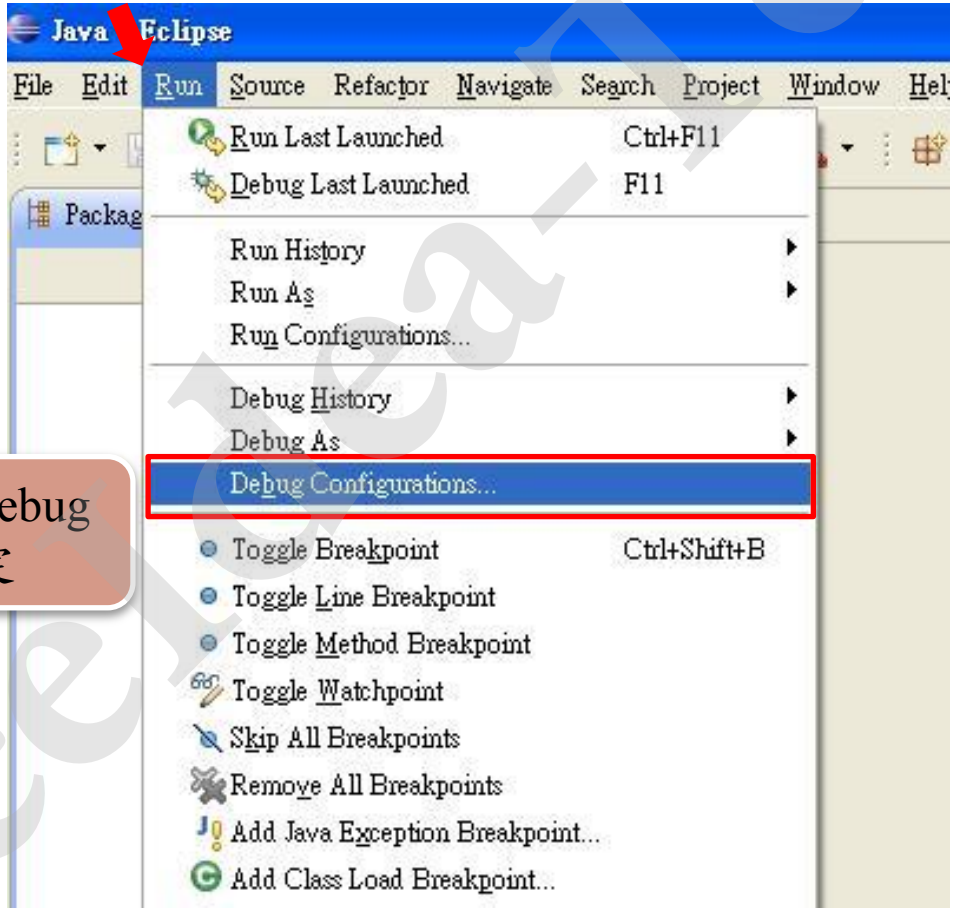
建立專案



Android針對res也提供了不同種類的編輯器，畫面上使用的是Android Layout Editor

FreeIdea Tech

建立專案

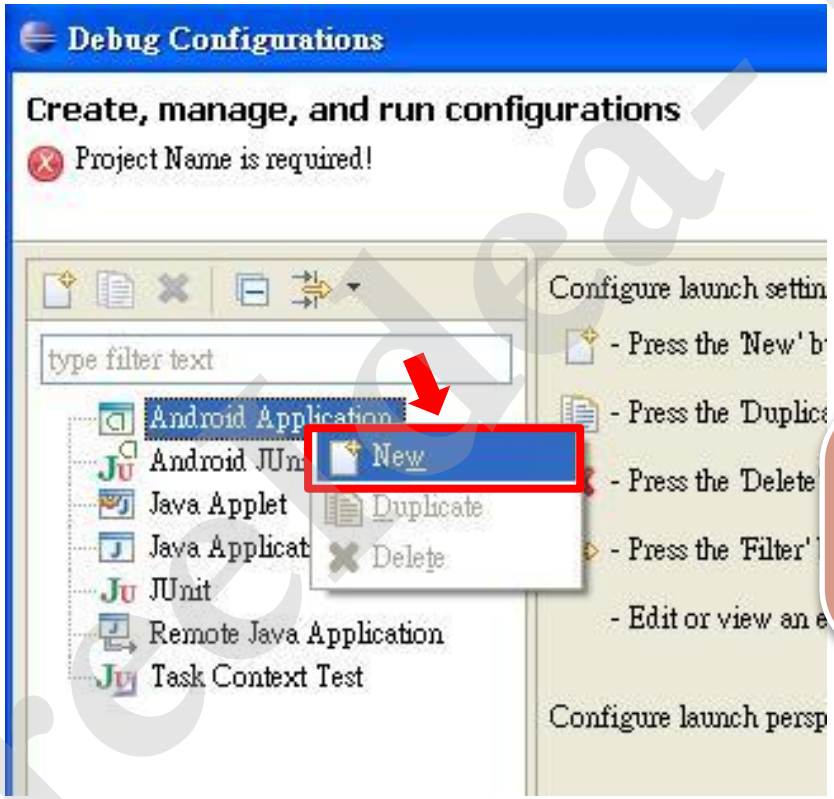


選擇"RUN" → "Debug Configuration"設定

FreeTea-Tech

建立專案

- 建立Android模擬器

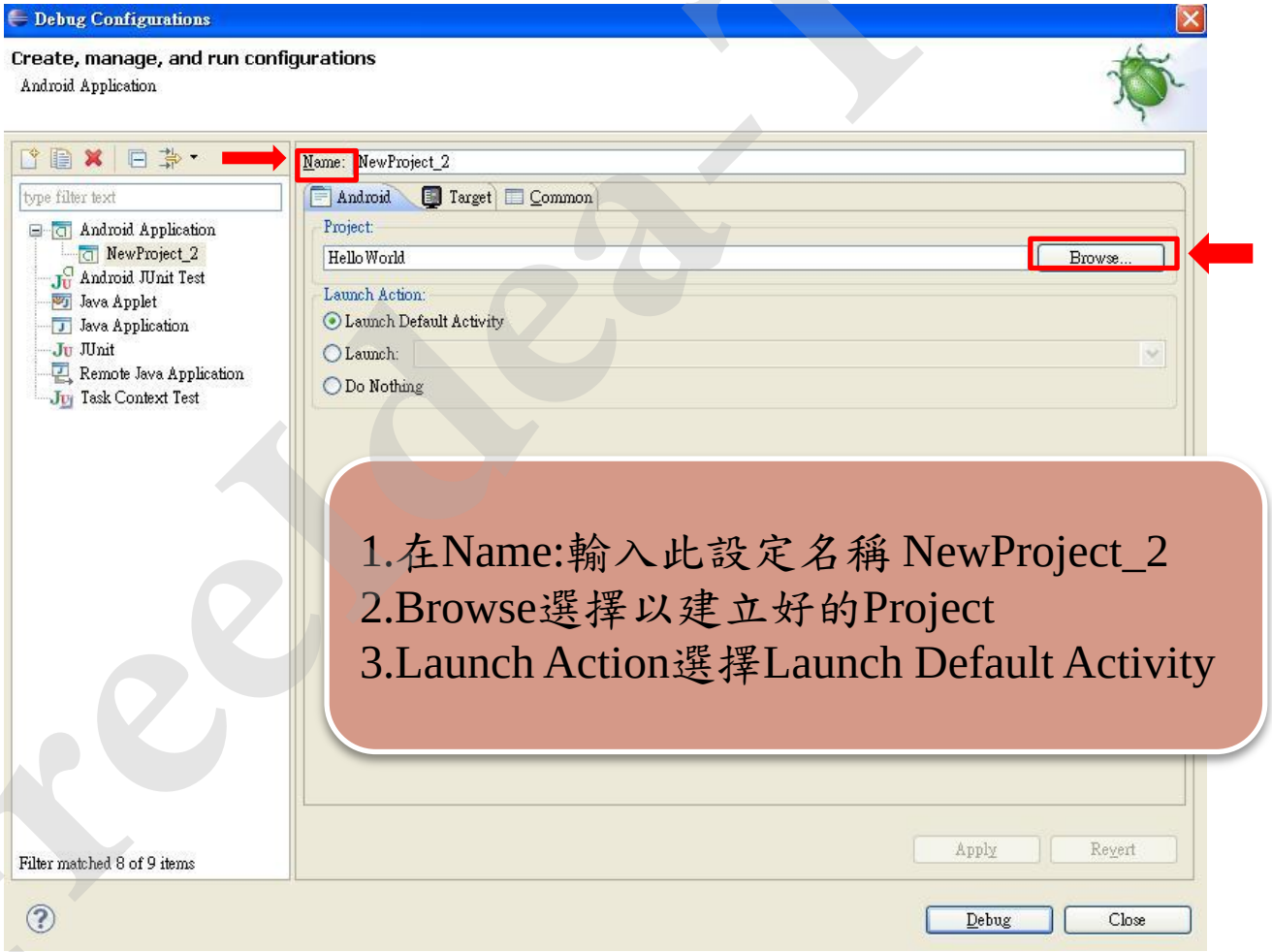


點選Android Application，右鍵點選”New”新增一個新設定

FreeIdeaTech

建立專案

- 建立Android模擬器



1.在Name:輸入此設定名稱 NewProject_2
2.Browse選擇以建立好的Project
3.Launch Action選擇Launch Default Activity

FreeIdea Tech

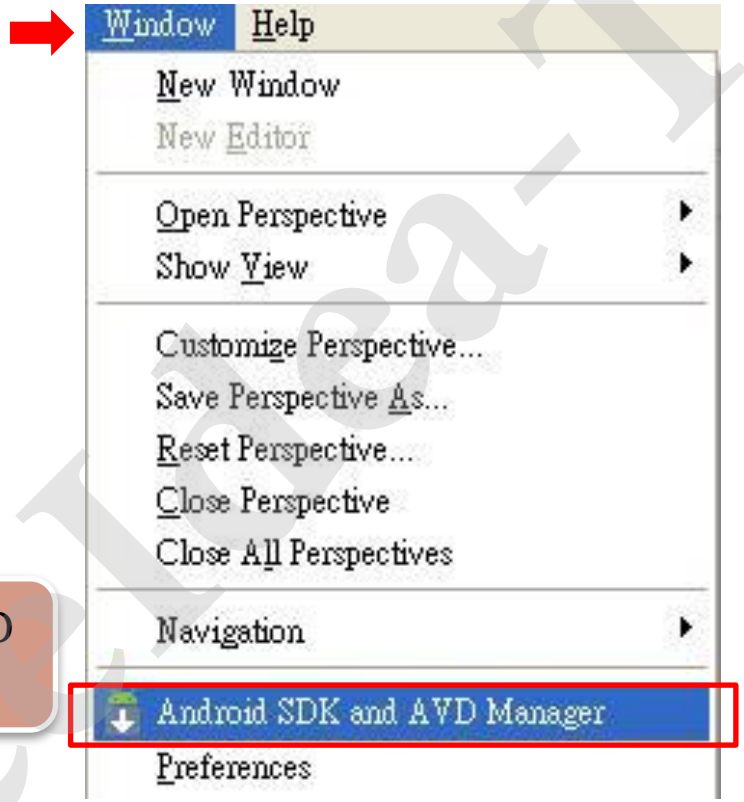
模擬器的執行與操作

- 建立Android模擬器
  - 進入模擬器設定頁面
  - Window → Android AVD Manager
- 設定與建立模擬器
  - 輸入Name
  - 自訂模擬器名稱
  - 選擇Target(建議為Google APIs-3.1)
  - 使用模擬器API
  - 選擇Skin
  - 模擬器佈景大小與顯示方式
  - 點選Finish → 顯示模擬器於清單中

FreeIdea-Tech

模擬器的執行與操作

- 建立Android模擬器



選擇"Window" → 設定AVD模擬器環境

Free Idea-Tech

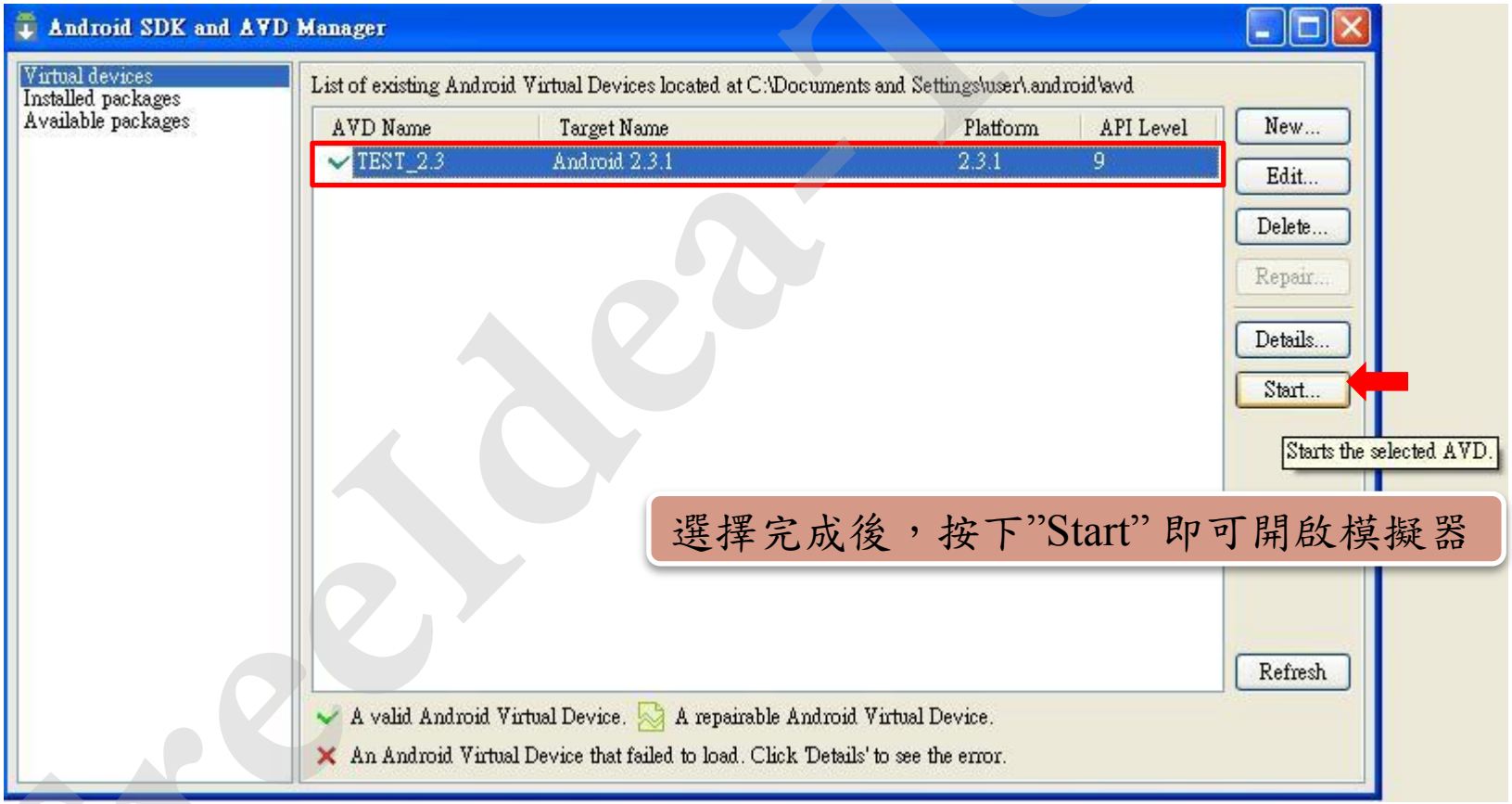
模擬器的執行與操作

Name: → 打上模擬器名稱
Target:→ 選擇Android 2.3.1- API Level 9
選擇完成後，按下”Create AVD” 即完成設定

FreeLife-Tech

模擬器的執行與操作

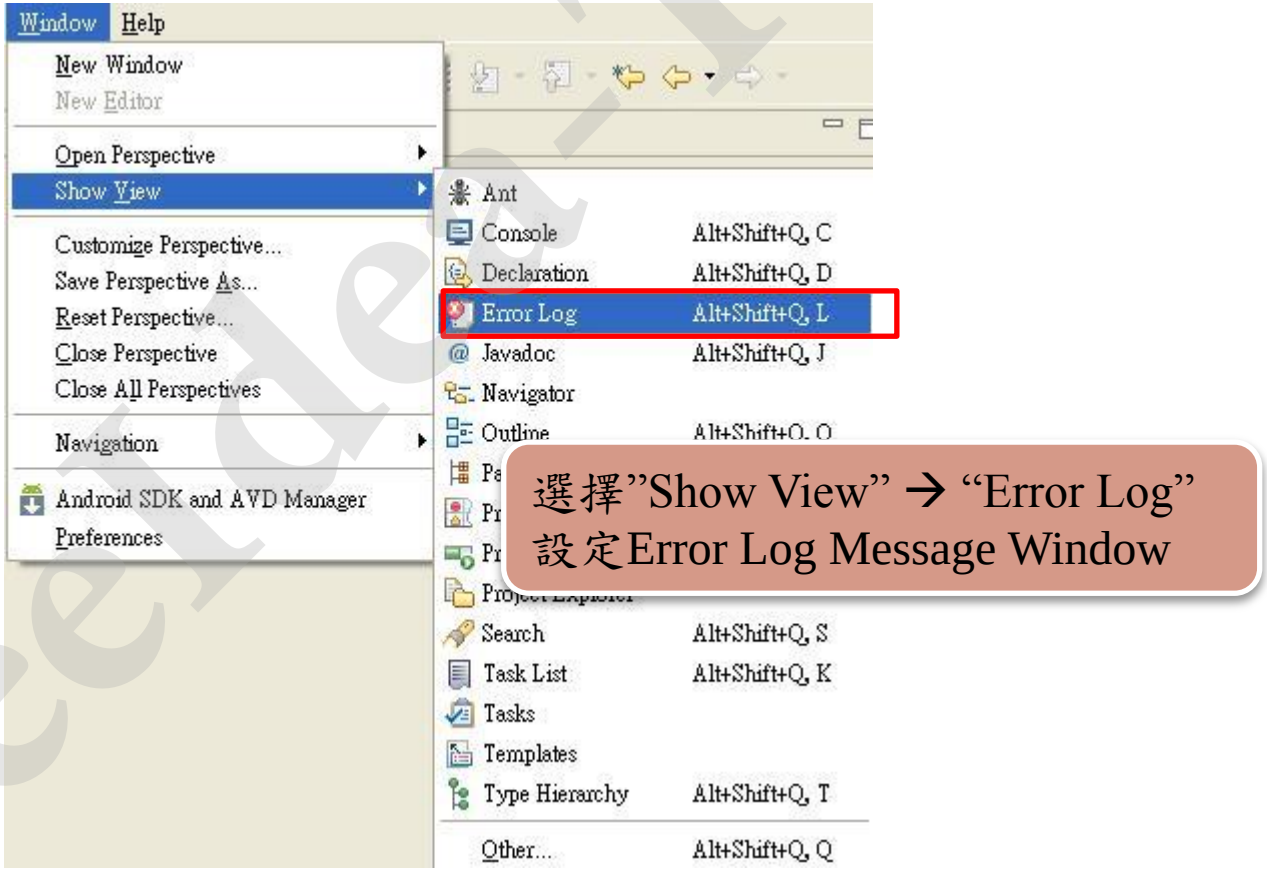
- 建立Android模擬器



FreeIdeaTech

使用Log及IDE除錯工具

- Error Log

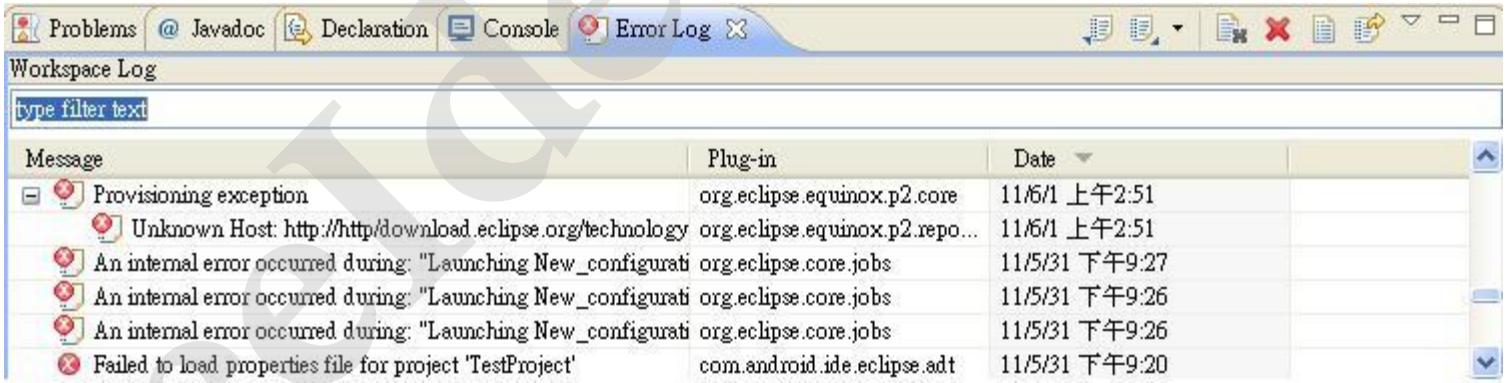


FreeIdea-Tech

使用Log及IDE除錯工具

- Error Log

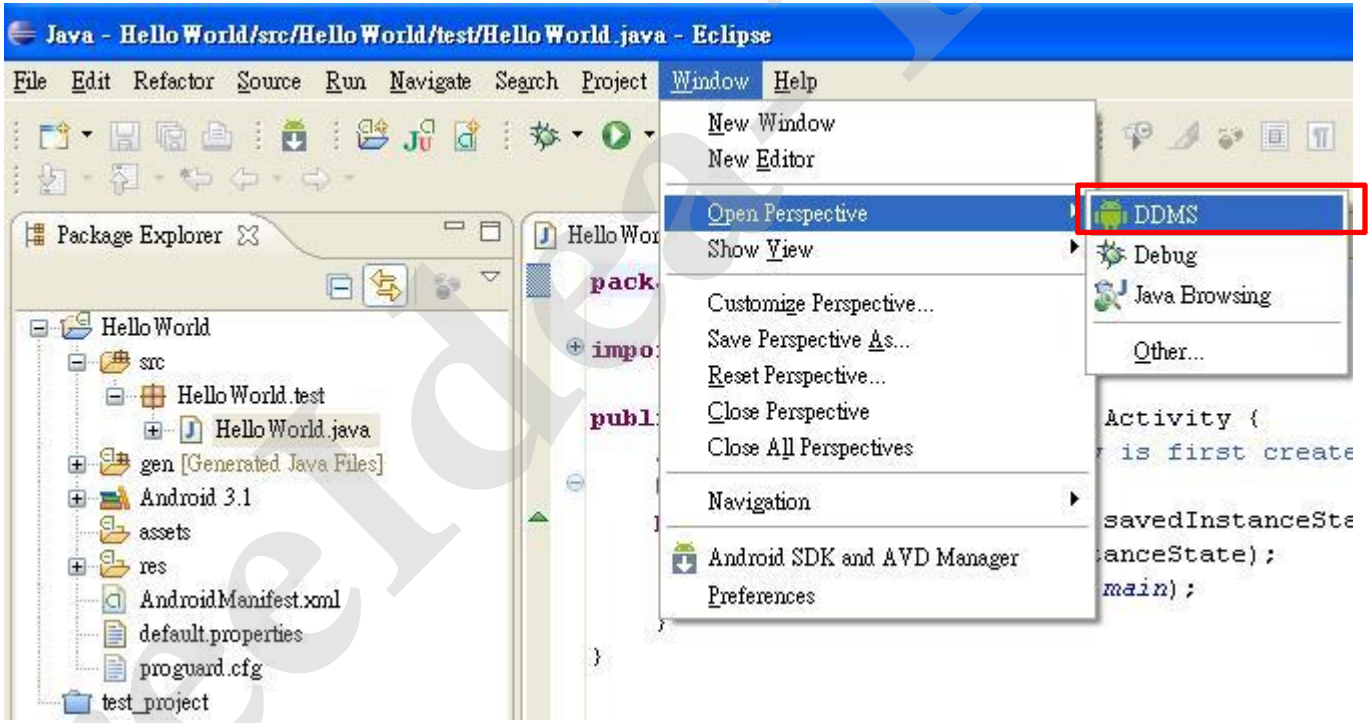
此時下方即會出現
Error Log Message Window



FreeIdea-Tech

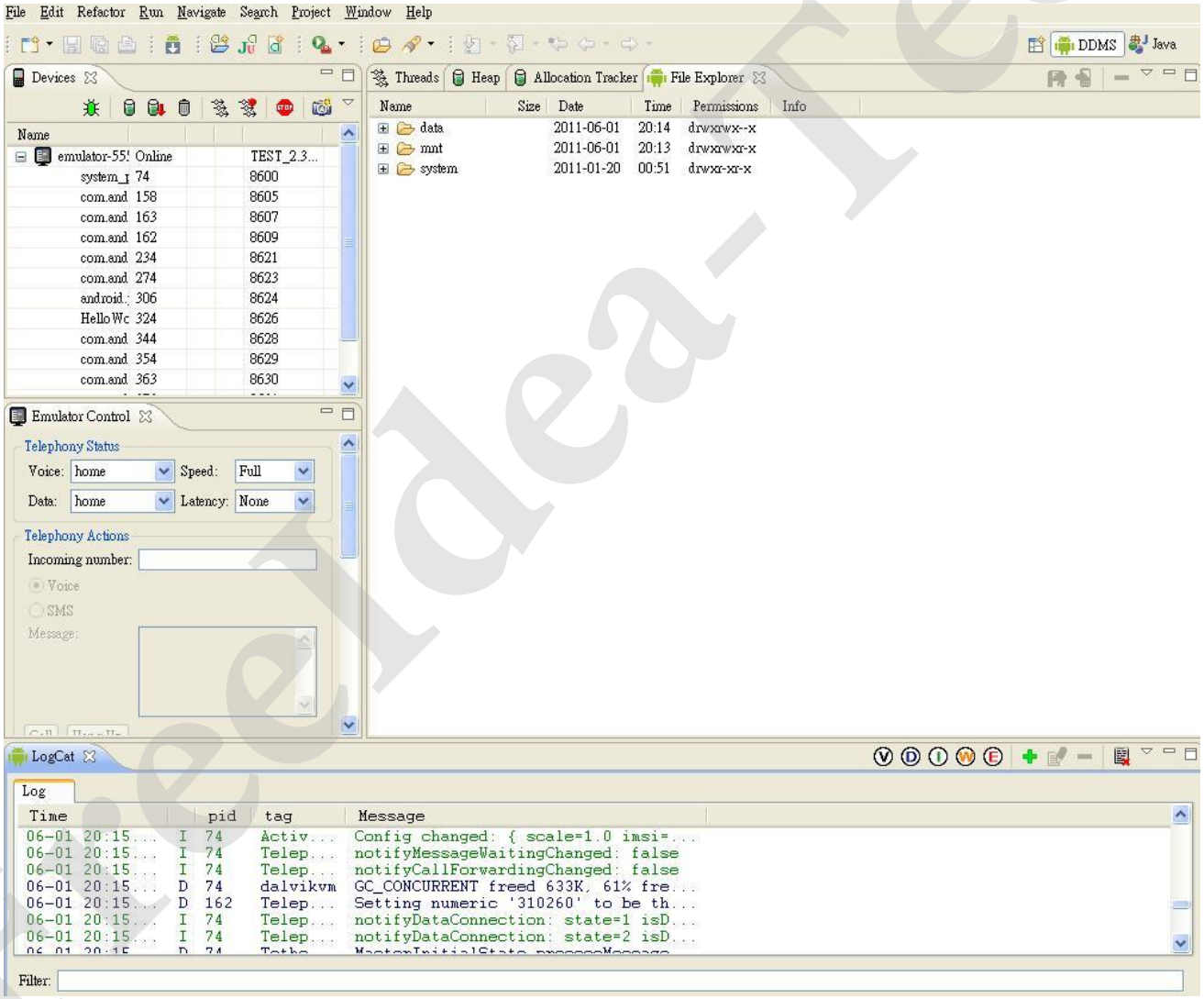
使用Log及IDE除錯工具

- IDE 除錯工具



FreeIdeaTech

使用Log及IDE除錯工具



FreeIdea-Tech

實機運作

- Android SDK 安裝
- 建立第一支 Hello World 專案
- Run 模擬器，並可以成功秀出Hello World！

FreeIdea Tech

實機運作

FreeIdea-Tech

Android 圖形介面設計

吳柏翰

FreeIdea-Tech

Outline

- 圖形操作介面設計
- 相關常用API
- 核心框架
- 程式生命週期
- 畫面切換的流程

FreeIdea-Tech

圖形操作介面設計

- Android與UI的關係
- Style樣式設計
- 視窗事件處理
- Event Listeners
- Event Handlers
- TouchMode
- Handling Focus

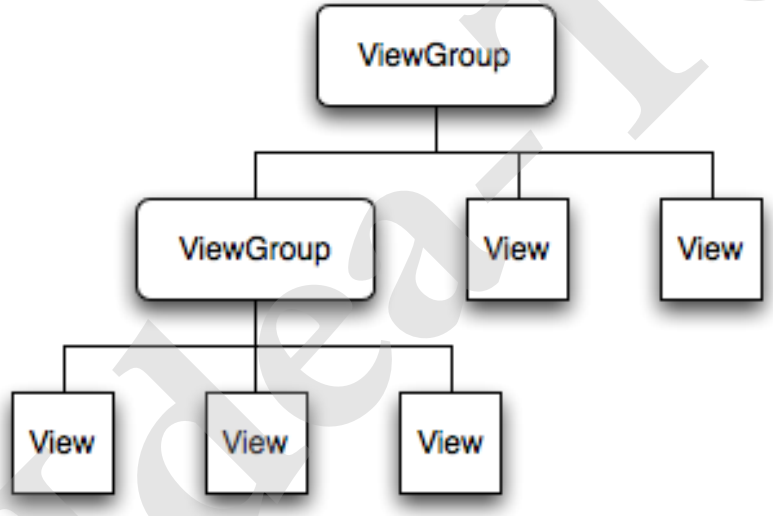
Freeidea-Tech

Android與UI的關係

- 使用者介面有兩種方式可以呈現
  - 與主程式混合寫在一起
  - 寫在XML中
- 用來顯示資料、影像或是其他訊息的元件，都被叫做View。
- View是大部分UI的父類別

FreeIdea-Tech

Android與UI的關係



ViewGroup是一種View容器，本身也是一種View，但是可以包含View及其他ViewGroup元件的View。

FriendTech

Style樣式設計

- 設計使用者介面的兩種方式
  - Styles
    - 是一個包含一種或者多種格式化屬性的集合，可以將其套用在佈局XML的單一元素中。
  - Themes
    - 是一個包含一種或者多種格式化屬性的集合，可以將其套用在應用程式中所有的活動當中或其中的某個活動。

FreeIdea Tech

Style樣式設計

- Android系統中有提供一些預設的style
- 如果要產生客製化的style，首先要在res/values目錄下建立一個名為style.xml的文件

FreeIdea Tech

Style樣式設計

布局文件(res/values/style.xml)：

style.xml定義內容如下

```
<?xml version="1.0" encoding="utf-8" ?>
<resources>
    <style name="StyleText1">
        <item name="android:textSize">18sp</item>
        <item name="android:textColor">#EC9237</item>
    </style>
    <style name="StyleText2">
        <item name="android:textSize">14sp</item>
        <item name="android:textColor">#FF7F7C</item>
        <item name="android:fromAlpha">0.0</item>
        <item name="android:toAlpha">0.0</item>
    </style>
</resources>
```

Freeidea-Tech

Style樣式設計

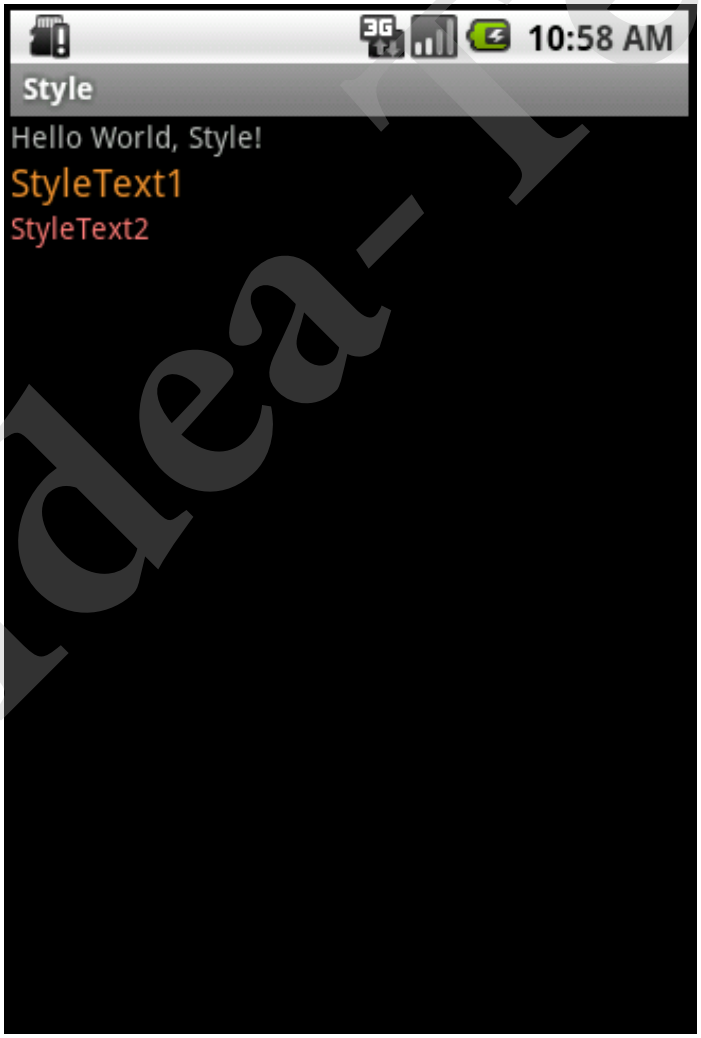
布局文件(res/layout/main.xml)：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    />
<TextView style = "@style/StyleText1"
        android:text="StyleText1"
        android:id="@+id/TextView01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
</TextView>
<TextView  style = "@style/StyleText2"
        android:text="StyleText2"
        android:id="@+id/TextView02"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
</TextView>
</LinearLayout>
```

接著加入兩個TextView，修改res/layout/main.xml

FreeIdea-Tech

Style樣式設計



FreeIdea-Tech

視窗事件處理

- 利用UI的事件處理(UI Events)，和使用者「互動」：
  - 在Android中有超過一種以上監聽使用者與應用程式互動的方法。
  - 當事件發生在使用者介面上時，截取使用者與特定View互動的事件，在View類別中就提供了此類的方法。

Fred de Tech

視窗事件處理

- Android framework呼叫不同的view中，針對許多UI事件的方法。
  - 例如：當一個View(按鈕)被點選，這時onTouchEvent()方法就會被呼叫。
- 為了監聽這個動作，必需繼承及覆寫這個方法，但為了處理這類事件，每個View都需要被繼承，實際上來說並不實用。
- View類別也包含巢狀介面的集合，使得這些事件可以更容易去定義，這些介面就被稱為event listeners。

FreeCatTech

Event Listeners

- 一個event listener在View類別中是一個介面並包含一個callback的方法。
- 當使用者與View的物件互動，此listener將會被註冊，這些方法將會由Android framework所呼叫。

FreeIdea-Tech

Event Listeners

- event listener有下列幾種callback方法：
  - onClick()
  - onLongClick()
  - onFocusChange()
  - onKey()
  - onTouch()
  - onCreateContextMenu()

FreeIdea-Tech

Event Listeners

程式碼(src/ncu/bnlab/OnClickListenerExample.java)：

```
public OnClickListener mSendListener = new OnClickListener()
{
   public void onClick(View v)
   {
      /* 當按鈕被按下所需執行的動作 */
      Toast.makeText(
            OnClickEX2.this,
      "Yes.",
      Toast.LENGTH_LONG).show();
    }
};
@Override
public void onCreate(Bundle savedInstanceState)
{
   super.onCreate(savedInstanceState);
   setContentView(R.layout.main);
   Button button = (Button)findViewById(R.id.send);
   /* 註冊onClick listener */
   button.setOnClickListener(mSendListener);
}
```

此處針對Event Listeners撰寫一範例，改寫onClick事件。

FreeIdea-Tech

Event Listeners

針對按鈕撰寫點擊事件，當按下按鈕時，下方會出現Toast的訊息。

FreeIdea-Tech

Event Listeners

程式碼(src/ncu/bnlab/OnClickListenerExample2.java)：

```
public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    Button button = (Button)findViewById(R.id.send);
    // 註冊onClick listener
    button.setOnClickListener(this)
}
public void onClick(View v)
{
     // 當按鈕被按下所需執行的動作
    Toast.makeText(
            this,
            "Yes.",
            Toast.LENGTH_LONG).show();
}
```

另外可以把OnClickListener作為活動的一部分來實作會更方便，這將避免額外的問題。

FreeIdea Tech

Event Listeners

- 第二個範例中onClick()沒有回傳值，但是一些其它event listener必需回傳一個布林值。

- 對於其中一些callback，原因與event相關，詳細說明如下。

  - onLongClick()
    - 回傳一個布林值表示是否已經處理了這個事件而不再進一步處理。

FreeIdea-Tech

Event Listeners

- onKey()
  - 回傳一個布林值來表示是否已經處理了這個事件而不再進一步處理。

- onTouch()
  - 回傳一個布林值來表示是否已經處理了這個事件。重要的是這個事件可以有多個彼此跟隨的動作。

FreeIdea-Tech

Event Listeners

- 按鍵事件總是傳遞給當前焦點所在的View。
- 它們從View層次的頂層開始被分發，然後依次向下，直到到達適合的目標。

FreeIdea-Tech

Event Handlers

- 如果從View建立一個客製化元件，那麼將定義一些callback方法當作預設的事件處理器。
  - onKeyDown(int, KeyEvent)
  - onKeyUp(int, KeyEvent)
  - onTrackballEvent(MotionEvent)
  - onTouchEvent(MotionEvent)
  - onFocusChanged(boolean, int, Rect)

FreeIdea-Tech

Event Handlers

- 不屬於View類別的一部分，但可以直接影響處理事件的方式：
  - Activity.dispatchTouchEvent(MotionEvent)
  - ViewGroup.onInterceptTouchEvent(MotionEvent)
  - ViewParent.requestDisallowInterceptTouchEvent (boolean)

FreeIdeaTech

TouchMode

- 觸控模式(TouchMode)
  - 設備具有觸控功能，而且使用者透過觸控來和UI互動，不需要針對焦點項目做提醒，或者設定焦點到一個特定的View，使用者也可以得知哪個項目將接受輸入。

FreeIdea Tech

TouchMode

- isFocusableInTouchMode()
  - 對於一個具備觸控功能的裝置，當使用者觸碰螢幕，設備就會進入觸控模式。因此只有isFocusableInTouchMode()方法回傳true，View才可取得焦點。
- isInTouchMode()
  - 為了查詢目前狀態，可以呼叫isInTouchMode()來查看此裝置是否處於觸控模式中。

FreeIdea Tech

Handling Focus

- Framework將根據使用者輸入去處理一般的焦點移動，這包含當View刪除、隱藏或者新View出現時改變焦點。

- isFocusable()
  - View透過isFocusable()方法表示取得焦點的意圖。

- setFocusable()
  - 呼叫setFocusable()，可以改變View是否可接受焦點。

FreeIdeaTech

Handling Focus

- isFocusableInTouchMode()
  - 在觸控模式下，可以透過isFocusableInTouchMode()查詢一個View是否允許接受焦點。
- setFocusableInTouchMode()
  - 透過setFocusableInTouchMode()，來改變isFocusableInTouchMode()查詢的view

FreeIdeaTech

Handling Focus

Handling Focus：

```
<LinearLayout
   android:orientation="vertical"
   ... >
  <Button android:id="@+id/top"
        android:nextFocusUp="@+id/bottom"
        ... />
  <Button android:id="@+id/bottom"
        android:nextFocusDown="@+id/top"
        ... />
</LinearLayout>
```

透過覆寫佈局文件中的XML屬性：
nextFocusDown、*nextFocusLeft*、*nextFocusRight*和*nextFocusUp*。
為失去焦點的View增加這些屬性，屬性值為擁有焦點的ViewID。

FreeIdea-Tech

相關常用API

- 元件佈局
- 文字元件
- 表單元件
- 對話視窗元件
- 選單元件
- 顯示元件

Freeidea-Tech

元件佈局

- LinearLayout
- RelativeLayout
- TableLayout

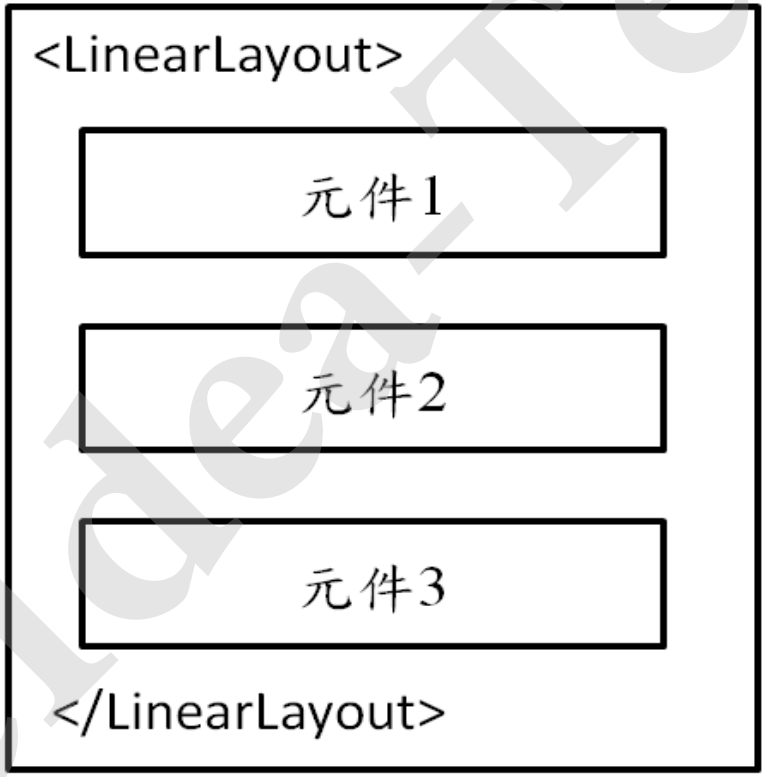
FreeIdea-Tech

LinearLayout

- 線性佈局共有兩個方向：
  - 垂直（vertical）
  - 水平（horizontal）
- 決定垂直或是水平的屬性為Orientation
  - android:orientation="vertical"

FreeIdea - Tech

LinearLayout



線性佈局就是將在< LinearLayout >內的元件以線性的方式來呈現

FreeIdeaTech

LinearLayout

在這個範例中，使用了<TextView>元件，共使用了三個<TextView>，這三個元件都以垂直向下的方式來呈現

FreeIdea-Tech

LinearLayout

布局文件(res/layout/main.xml)：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
        <TextView
        android:text="這是第一列"
        android:textSize="22sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
        <TextView
        android:text="這是第二列"
        android:textSize="22sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
        <TextView
        android:text="這是第三列"
        android:textSize="22sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

FreeIdea-Tech

LinearLayout

- android:orientation
  - 在<LinearLayout>中，此屬性代表元件的排列是垂直或水平佈局
- android:layout_width
  - 代表此元件佈局的寬度，若值為fill_parent則會填滿parent的寬度；若值為wrap_content則元件寬度會依照內容大小而調整

FreeIdeaTech

LinearLayout

- android:layout_height
  - 代表此元件佈局的高度，若值為fill_parent則會填滿parent的高度；若值為wrap_content則元件高度會依照內容大小而調整
- android:layout_margin
  - 指定這個view距離上下左右的額外距離

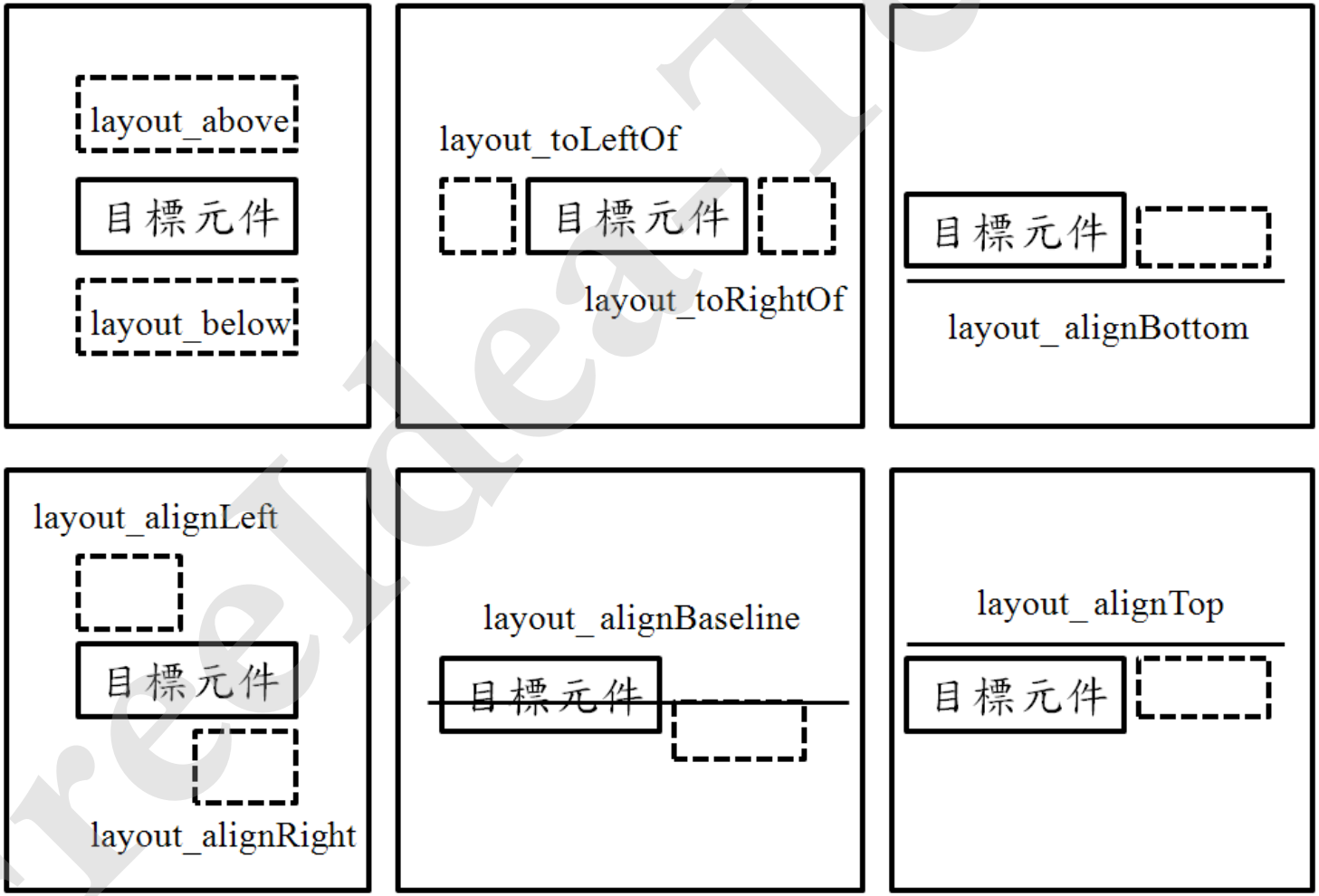
FreeIdeaTech

LinearLayout

- android: layout_marginBottom
  - 指定這個view距離下方的額外距離
- android: layout_marginLeft
  - 指定這個view距離左方的額外距離
- android: layout_marginRight
  - 指定這個view距離右方的額外距離
- android: layout_marginTop
  - 指定這個view距離上方的額外距離

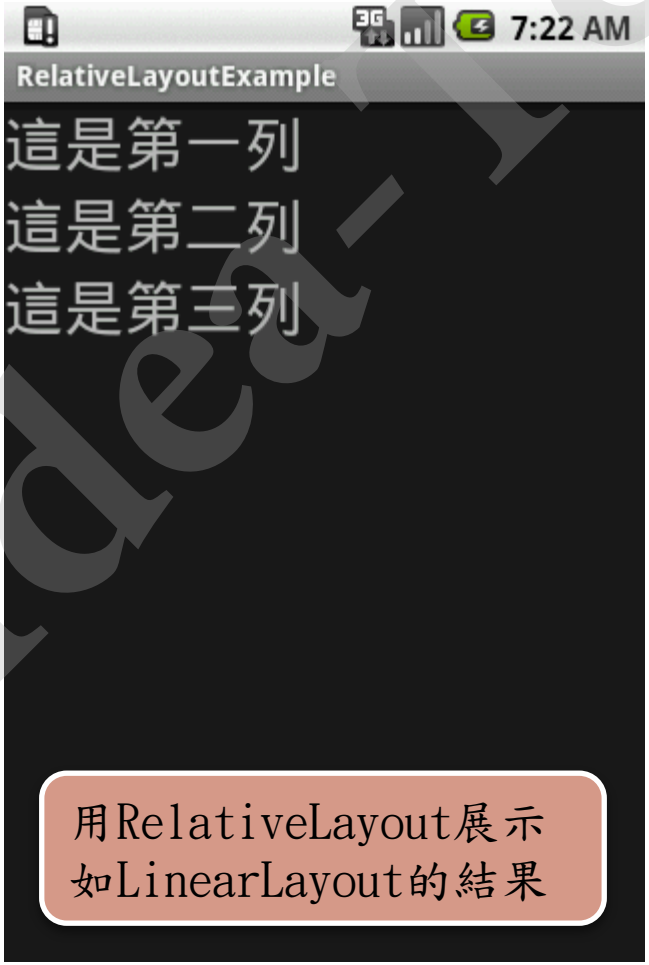
FreeDa-Tech

RelativeLayout



FreeIdea Tech

RelativeLayout



FreeIdea-Tech

RelativeLayout

布局文件架構(res/layout/main.xml)：

```
<!-- 顯示第一行的TextView -->
android:id="@+id/TextView01"

<!-- 顯示第二行的TextView -->
android:id="@+id/ TextView02"
android:layout_below="@id/ TextView01"
<!-- 顯示第三行的TextView -->
android:id="@+id/ TextView03"
android:layout_below="@id/ TextView02"
```

程式以TextView01為參考，TextView02置放在TextView01下方，TextView03放置在TextView02下方

FreeIdea-Tech

RelativeLayout

布局文件架構(res/layout/main.xml)：

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent">

        <TextView
        android:text="這是第一列"
        android:id="@+id/TextView01"
        android:textSize="30sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

        <TextView
        android:text="這是第二列"
        android:id="@+id/TextView02"
        android:textSize="30sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/TextView01" />
        <TextView
        android:text="這是第三列"
        android:id="@+id/TextView03"
        android:textSize="30sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/TextView02" />
</RelativeLayout>
```

此範例用到三個TextView，利用三個TextView作RelativeLayout

Freeidea-Tech

RelativeLayout

- android:layout_above
  - 置於目標id元件的上方
- android:layout_alignBaseline
  - 置於與目標id元件同樣的基準線上
- android:layout_alignBottom
  - 讓自己的下邊界與目標id元件的下邊界在同一個位置

Free de Tech

RelativeLayout

- android:layout_alignLeft
  - 讓自己的左邊界與目標id元件的左邊界在同一位置
- android:layout_alignParentBottom
  - 若為true，讓自己的下邊界與Parent的下邊界同位置
- android:layout_alignParentLeft
  - 若為true，讓自己的左邊界與Parent的左邊界同位置

Freeidea-Tech

RelativeLayout

- android:layout_alignParentRight
  - 若為true，讓自己的右邊界與Parent的右邊界同位置
- android:layout_alignParentTop
  - 若為true，讓自己的上邊界與Parent的上邊界同位置
- android:layout_alignRight
  - 讓自己的右邊界與目標id元件的右邊界在同一位置

Freeidea-Tech

RelativeLayout

- android:layout_alignTop
  - 讓自己的上邊界與目標id元件的上邊界在同一個位置
- android:layout_alignWithParentIfMissing
  - 若設為true，當參考的目標id不可用時，會以Parent為參考目標
- android:layout_below
  - 置於目標id元件的下方

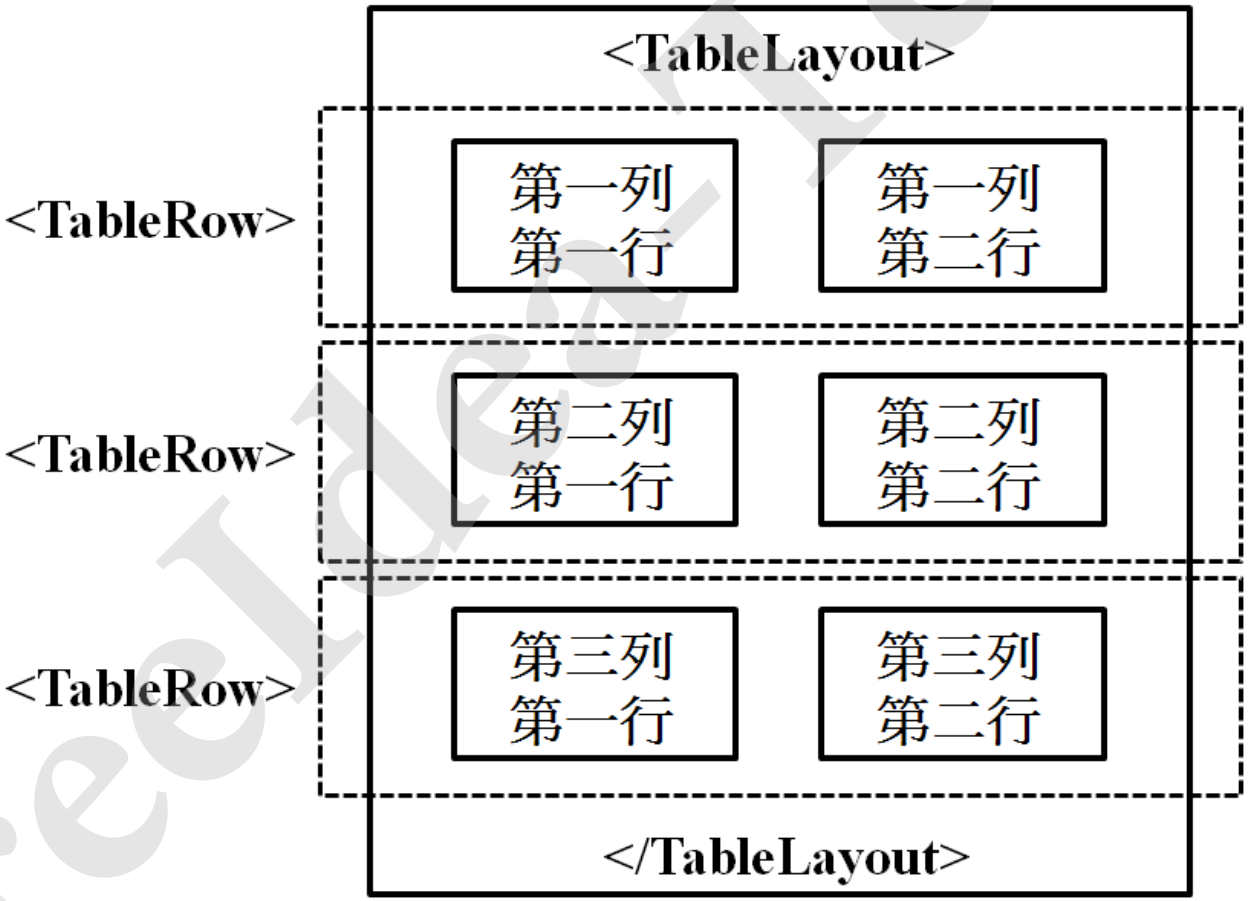
FreeIdea-Tech

RelativeLayout

- android:layout_centerHorizontal
  - 若為true，置於Parent水平位置的中心
- android:layout_centerInParent
  - 若為true，置於Parent水平以及垂直位置的中心
- android:layout_centerVertical
  - 若為true，置於Parent垂直位置的中心
- android:layout_toLeftOf/toRightOf
  - 置於目標id元件的左方/右方

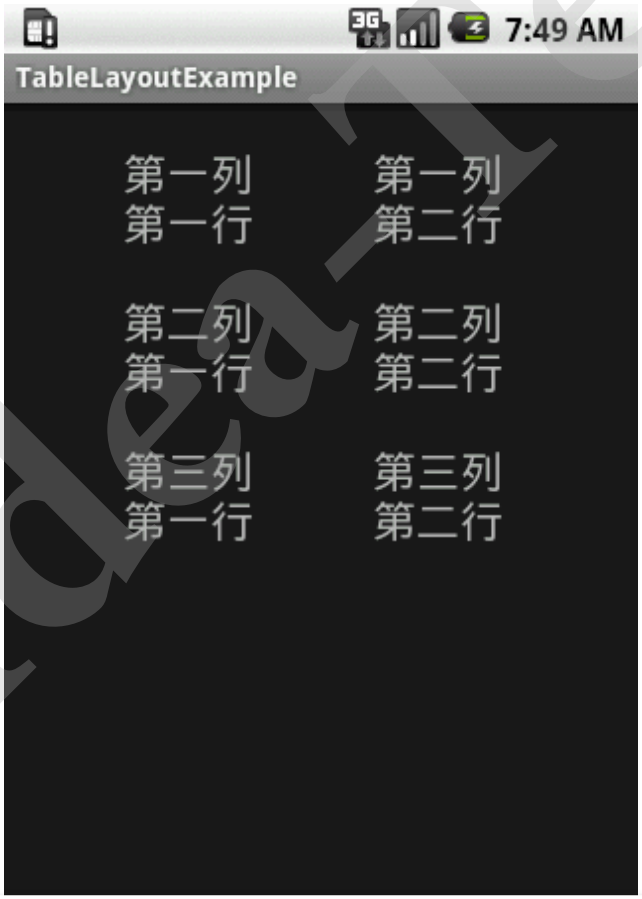
FreeIdea Tech

TableLayout



FreeIdeaTech

TableLayout



範例使用了三個<TableRow>，意即有三個row。

FreeIdea-Tech

TableLayout

布局文件-1(res/layout/main.xml)：

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent">
   <TableRow android:layout_marginTop="20px">
      <TextView
      android:layout_column="0"
      android:text="第一列\n第一行"
      android:textSize="22sp"
      android:layout_marginLeft="60px"/>
      <TextView
      android:layout_column="1"
      android:text="第一列\n第二行"
      android:textSize="22sp"
      android:layout_marginLeft="60px"/>/>
   </TableRow>
<TableRow android:layout_marginTop="20px">
      <TextView
```

FreeIdea Tech

TableLayout

布局文件-2(res/layout/main.xml)：

```
android:layout_column="0"
        android:text="第二列\n第一行"
        android:textSize="22sp"
        android:layout_marginLeft="60px"/>
        <TextView
        android:layout_column="1"
        android:text="第二列\n第二行"
        android:textSize="22sp"
        android:layout_marginLeft="60px"/>/>
    </TableRow>
     <TableRow android:layout_marginTop="20px">
        <TextView
        android:layout_column="0"
        android:text="第三列\n第一行"
        android:textSize="22sp"
        android:layout_marginLeft="60px"/>
        <TextView
        android:layout_column="1"
        android:text="第三列\n第二行"
        android:textSize="22sp"
        android:layout_marginLeft="60px"/>/>
    </TableRow>
</TableLayout>
```

FreeIdea-Tech

文字元件

- TextView
- AutoCompleteTextView

FreeIdea-Tech

TextView

- TextView是個基本常用的元件
  - 可使用XML來操作
  - 可使用程式碼中的Method方法來操作
- 下列敘述XML與Method相對應之屬性功能
  - 格式為 XML <-> Method

FreeIdea Tech

TextView

- android:autoLink <-> setAutoLinkMask(int)
  - 可讓文字上的連結自動變成可點擊的連結
- android:gravity <-> setGravity(int)
  - 設定文字在View中x軸和y軸相關數值
- android:height <-> setHeight(int)
  - 設定TextView的高度
- android:width <-> setWidth(int)
  - 設定TextView的寬度

FreeDaTech

TextView

- android:hint <-> setHint(int)
  - 當Text是空的時候，就會顯示hint中的文字
- android:lines <-> setLines(int)
  - 設置TextView高度為幾個Line的高度，值必須為整數型別
- android:maxLength <-> setFilters(InputFilter)
  - 設定TextView文字的最大長度

FreeIdea Tech

TextView

- android:password <-> setTransformationMethod(Transformation)
  - 讓Text顯示成其他符號，常用於輸入或顯示密碼時
- android:text <-> setText(CharSequence)
  - 顯示的文字
- android:textStyle <-> setTypeface(Typeface)
  - 設定文字樣式

FreeIdeaTech

TextView

- android:textColor <-> setTextColor(ColorStateList)
  - 設定文字的顏色
- android:textColorLink <-> setLinkTextColor(int)
  - 設定連結的顏色
- android:textSize <-> setTextSize(float)
  - 設定文字大小

Freeria-Tech

TextView

若點擊上述的網址，則會自動開啟瀏覽器跳至指定之網頁

FreeIdea-Tech

TextView

純粹用XML語法產生連結(res/layout/main.xml)：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent">
   <TextView
         android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:autoLink="web"
            android:text="Google - www.google.com" />
            <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
         android:autoLink="web"
         android:text="Yahoo - www.yahoo.com.tw" />
</LinearLayout>
```

設置autoLink為web

設置autoLink為web

Freelea-Tech

TextView

布局文件(res/layout/main.xml)：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/my_layout" >
</LinearLayout>
```

用程式碼產生連結 – 1(TextView.java)：

```
package ncu.bnlab;
import android.app.Activity;
import android.os.Bundle;
import android.text.util.Linkify;
import android.view.Gravity;
import android.widget.LinearLayout;
import android.widget.TextView;
```

Freida-Tech

TextView

用程式碼產生連結－2(TextView.java)：

```
public class TextViewExample extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        LinearLayout layout = (LinearLayout)findViewById(R.id.my_layout);
        TextView tv1 = new TextView(this);
        tv1.setGravity(Gravity.CENTER);
        tv1.setAutoLinkMask(Linkify.WEB_URLS);
        tv1.setText("Google - www.google.com");
        layout.addView(tv1);

        TextView tv2 = new TextView(this);
        tv2.setGravity(Gravity.CENTER);
        tv2.setAutoLinkMask(Linkify.WEB_URLS);
        tv2.setText("Yahoo - www.yahoo.com.tw");
        layout.addView(tv2);
    }
}
```

Freida-Tech

TextView

- android:autoLink <-> setAutoLinkMask(int)
  - none
    - autoLink預設值
  - all <-> Linkify.ALL
    - 目前所有連結種類
  - email <-> Linkify. EMAIL_ADDRESSES
    - Email連結
  - phone <-> Linkify. PHONE_NUMBERS
    - 電話號碼連結
  - web <-> Linkify.WEB_URLS
    - 網址連結

設置autoLink屬性

Freeidea-Tech

AutoCompleteTextView

- AutoCompleteTextView可達到簡易自動完成Text的功能
- 假設已有定義台灣的縣市英文名稱，當輸入Ta時，程式會自動將符合Ta的縣市名稱給列出來，如範例所展示。

FreeIdea Tech

AutoCompleteTextView

FreeIdea-Tech

AutoCompleteTextView

布局文件(res/layout/main.xml)：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent" >
                <AutoCompleteTextView
                android:text=""
                android:id="@+id/city"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content" />
</LinearLayout>
```

FreeIdea-Tech

AutoCompleteTextView

程式碼(TextView.java)：

```
package ncu.bnlab;
import android.app.Activity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
public class AutoCompleteTextViewExample extends Activity {
   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);
      ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
      android.R.layout.simple_dropdown_item_1line, TAIWAN);
      AutoCompleteTextView TaiwanCity =
        (AutoCompleteTextView)findViewById(R.id.city);
      TaiwanCity.setAdapter(adapter);
   }
private static final String[] TAIWAN = new String[] {
      "Keelung", "Taipei", "Taoyuan", "Hsinchu", "Miaoli", "Taichung", "Changhua",
      "Nantou", "Yunlin", "Chiayi", "Tainan", "Kaohsiung", "Pingtung ", "Yilan",
      "Hualien",  "Taitung"
      };
}
```

Freetea-Tech

表單元件

- Button
- ImageButton
- RadioButton
- CheckBox
- DatePicker
- ProgressBar
- RatingBar

FreeIdea-Tech

Button

- Button的layout方面會有兩種屬性
  - wrap_content
    - layout_width為wrap_content時，這個button會依據button上的text長度為基準
  - fill_parent
    - layout_width為fill_parent時，則會以parent最寬的長度為主

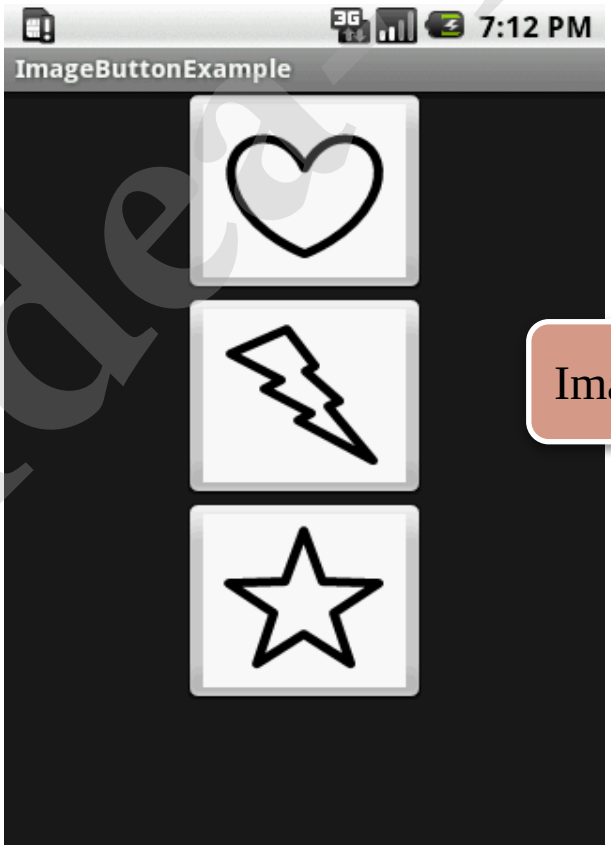
Freeder Tech

Button

FreeIdea-Tech

ImageButton

- ImageButton可以將圖片當作button的背景
  - 利用此屬性 layout:src="圖片位置"



ImageButton 範例

FreeIdea Tech

ImageButton

布局文件(res/layout/main.xml)：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
        <ImageButton android:id="@+id/ImageButton01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:src="@drawable/heart" />

        <ImageButton android:id="@+id/ImageButton02"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:src="@drawable/lightning" />

        <ImageButton android:id="@+id/ImageButton03"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:src="@drawable/star" />
</LinearLayout>
```

Freeida-Tech

RadioButton

- RadioButton為個別的一個按鈕
- 若要做成有多選一這種功能時，則需將這些RadioButton放置一個<RadioGroup>中

FreeIdeaTech

RadioButton

FreeIdea-Tech

RadioButton

布局文件-1(res/layout/main.xml)：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <TextView
    android:text="您對此次教學感到?"
    android:id="@+id/TextView01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

    <RadioGroup
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:orientation="horizontal“>
```

Free Idea-Tech

RadioButton

布局文件-2(res/layout/main.xml)：

```
                <RadioButton
                android:text="滿意"
                android:id="@+id/RadioButton01"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:checked="true"/>

                <RadioButton
                android:text="普通"
                android:id="@+id/RadioButton02"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />

                <RadioButton
                android:text="不滿意"
                android:id="@+id/RadioButton03"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
</RadioGroup>
</LinearLayout>
```

FreeIdea-Tech

CheckBox

- CheckBox只有兩種型態
  - checked
  - unchecked
- CheckBox可用在勾選多項選擇時

FreeIdea-Tech

CheckBox

FreeIdea-Tech

CheckBox

布局文件-1(res/layout/main.xml)：

```
<TextView
                android:text="你喜歡的顏色? (可複選)"
                android:id="@+id/TextView01"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
                <LinearLayout
                android:orientation="horizontal"
android:layout_width="fill_parent"
   android:layout_height="wrap_content">
                                <CheckBox
                                android:text="紅色"
                                android:id="@+id/CheckBox01"
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content"  />

                                <CheckBox
                                android:text="黃色"
                                android:id="@+id/CheckBox02"
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content" />
                                <CheckBox
                                android:text="綠色"
                                android:id="@+id/CheckBox02"
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content" />
                </LinearLayout>
```

FreeIdea-Tech

CheckBox

布局文件-2(res/layout/main.xml)：

```
		<TextView
				android:text="平常的休閒活動?(可複選)"
				android:id="@+id/TextView02"
				android:layout_width="wrap_content"
				android:layout_height="wrap_content" />
		<LinearLayout
		android:orientation="horizontal"
	android:layout_width="fill_parent"
	android:layout_height="wrap_content">
				<CheckBox
				android:text="打球"
				android:id="@+id/CheckBox04"
				android:layout_width="wrap_content"
				android:layout_height="wrap_content"  />

				<CheckBox
				android:text="游泳"
				android:id="@+id/CheckBox05"
				android:layout_width="wrap_content"
				android:layout_height="wrap_content" />
				<CheckBox
				android:text="上網"
				android:id="@+id/CheckBox06"
				android:layout_width="wrap_content"
				android:layout_height="wrap_content" />
		</LinearLayout>
</LinearLayout>
```

FreeIdea-Tech

DatePicker

- DatePicker可讓使用者選擇年、月、日

當使用者按下button後，TextView會顯示DatePicker所選的日期。

FreeIdea Tech

DatePicker

布局文件-1(res/layout/main.xml)：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <DatePicker
    android:id="@+id/DatePicker01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="15sp"/>
     <Button
     android:text="確定"
     android:id="@+id/Button01"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:layout_gravity="center_horizontal"
     android:layout_marginTop="25sp"/>
     <TextView
     android:text=""
     android:id="@+id/TextView01"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:layout_gravity="center_horizontal"
     android:layout_marginTop="35sp"/>
</LinearLayout>
```

Freeida-Tech

DatePicker

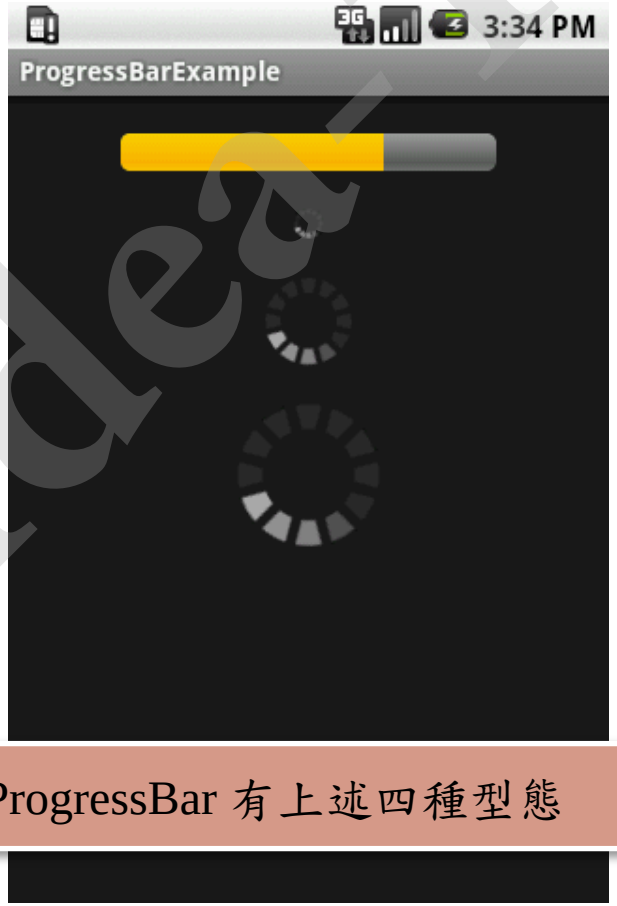
程式碼(DatePickerExample.java)：

```
public class DatePickerExample extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        final DatePicker datePicker = (DatePicker) findViewById(R.id.DatePicker01);
        final Button myButton = (Button) findViewById(R.id.Button01);
        final TextView textView = (TextView) findViewById(R.id.TextView01);

        myButton.setOnClickListener(new OnClickListener(){
            public void onClick(View v) {
            int year = datePicker.getYear();
            int month = datePicker.getMonth() + 1;
            int day = datePicker.getDayOfMonth();
            textView.setText( year + "-" + month + "-" + day );
        }
        });
    }
}
```

Free-Tech

ProgressBar

- ProgressBar可用於顯示程式執行進度



ProgressBar 有上述四種型態

FreeIdea Tech

ProgressBar

布局文件(res/layout/main.xml)：

```
<ProgressBar
android:id="@+id/ProgressBar01"
android:layout_height="wrap_content"
android:layout_width="200px"
android:layout_gravity="center_horizontal"
android:layout_marginTop="20sp"
style="?android:attr/progressBarStyleHorizontal" />
<ProgressBar
android:id="@+id/ProgressBar02"
android:layout_height="wrap_content"
android:layout_width="wrap_content"
android:layout_gravity="center_horizontal"
android:layout_marginTop="20sp"
style="?android:attr/progressBarStyleSmall" />
<ProgressBar
android:id="@+id/ProgressBar03"
android:layout_height="wrap_content"
android:layout_width="wrap_content"
android:layout_gravity="center_horizontal"
android:layout_marginTop="20sp"
style="?android:attr/progressBarStyle" />
<ProgressBar
android:id="@+id/ProgressBar04"
android:layout_height="wrap_content"
android:layout_width="wrap_content"
android:layout_gravity="center_horizontal"
android:layout_marginTop="20sp"
style="?android:attr/progressBarStyleLarge" />
```

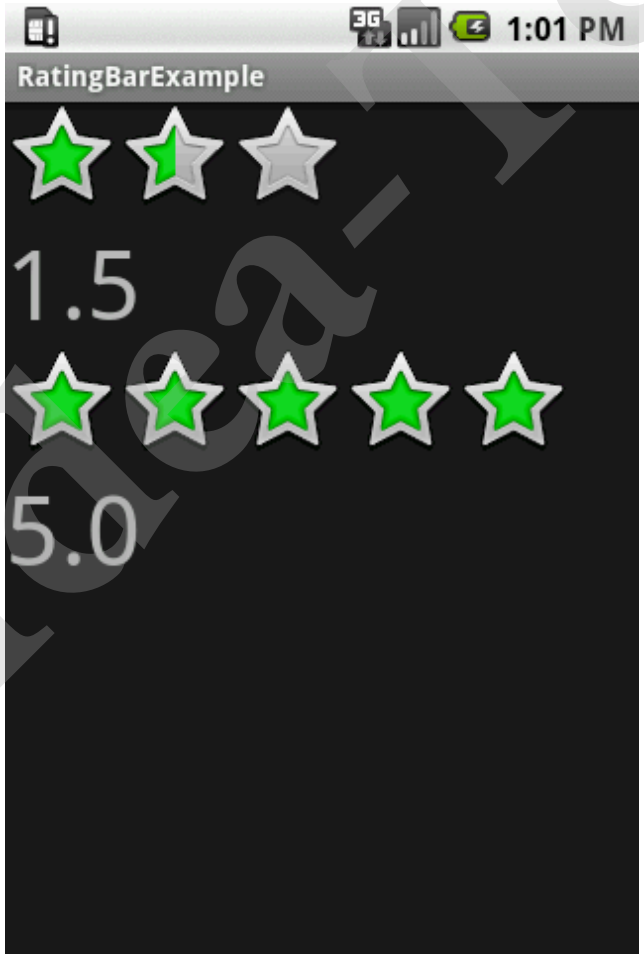
FreeIdea-Tech

RatingBar

- RatingBar可用來製作評分系統
  - 使用者可碰觸RatingBar的星星來達到評分
    - 若碰觸星星的右半部則會加分
    - 碰觸左半部則會扣分
    - 最多為五顆星，星星間隔為0.5顆星

FreeIdea Tech

RatingBar



FreeIdea-Tech

RatingBar

布局文件(res/layout/main.xml)：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent">
                <RatingBar android:id="@+id/RatingBar01"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:numStars="3" />
                <TextView
                android:text=""
                android:id="@+id/TextView01"
                android:textSize="50sp"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content" />
                <RatingBar android:id="@+id/RatingBar02"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
                <TextView
                android:text=""
                android:id="@+id/TextView02"
                android:textSize="50sp"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content" />
</LinearLayout>
```

FreeIdea Tech

RatingBar

程式碼(RatingBarExample.java)：

```
public class RatingBarExample extends Activity {
   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);

      final TextView textView01 = (TextView) findViewById(R.id.TextView01);
      final TextView textView02 = (TextView) findViewById(R.id.TextView02);
      final RatingBar ratingBar01 = (RatingBar) findViewById(R.id.RatingBar01);
      ratingBar01.setOnRatingBarChangeListener(new OnRatingBarChangeListener(){
          public void onRatingChanged(RatingBar ratingBar, float rating, boolean fromUser) {
              textView01.setText(Float.toString(ratingBar01.getRating()));
          }});
       final RatingBar ratingBar02 = (RatingBar) findViewById(R.id.RatingBar02);
       ratingBar02.setOnRatingBarChangeListener(new OnRatingBarChangeListener(){
          public void onRatingChanged(RatingBar ratingBar, float rating,boolean fromUser) {
                                  textView02.setText(Float.toString(ratingBar02.getRating()));

           }});
      }
}
```

Prestige-Tech

對話視窗元件

- AlertDialog
- ProgressDialog

FreeIdea-Tech

AlertDialog

- AlertDialog為警告對話視窗，像是離開程式或是刪除檔案時會跳出的對話視窗元件

FreeIdeaTech

AlertDialog

程式碼-1(AlertDialogExample.java)：

```
public class AlertDialogExample extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        final Button button01 = (Button) findViewById(R.id.Button01);
        final Button button02 = (Button) findViewById(R.id.Button02);
        button01.setOnClickListener(new OnClickListener(){
        public void onClick(View arg0) {
            About();
        }
        });
        button02.setOnClickListener(new OnClickListener(){
         public void onClick(View arg0) {
            Leave();
        }
        });
    }
    private void About() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("關於")
            .setMessage("這是 Alert Dialog")
            .show();
    }
```

Freeida-Tech

AlertDialog

程式碼-2(AlertDialogExample.java)：

```
 private void Leave() {
          AlertDialog.Builder builder = new AlertDialog.Builder(this);
          builder.setMessage("確定要離開本程式嗎?")
            .setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
           public void onClick(DialogInterface dialog, int which) {
              // TODO Auto-generated method stub
              AlertDialogExample.this.finish();
            }
            })
            .setNegativeButton("No", new
DialogInterface.OnClickListener() {
             public void onClick(DialogInterface dialog, int which) {
                // TODO Auto-generated method stub
               dialog.cancel();
              }
            });
           AlertDialog about_dialog = builder.create();
            about_dialog.show();
      }
}
```

FreeLeaTech

AlertDialog

- create()
  - 創建一個AlertDialog
- setCancelable(boolean)
  - 設為false時，使用者無法使用其他方式關閉這個對話視窗，只可使用對話視窗上給的操作方式來操作
- setIcon(Drawable or int)
  - 設置Title上的Icon
- setTitle(CharSequence or int)
  - 設定要顯示的Title

FreeIdea-Tech

AlertDialog

- setMessage(CharSequence or int)
  - 設定要顯示的內容
- setPositiveButton(CharSequence text, DialogInterface.OnClickListener listener)
  - 設定正向(左邊)的按鈕
- setNegativeButton(CharSequence text, DialogInterface.OnClickListener listener)
  - 設定反向(右邊)的按鈕
- show()
  - 顯示AlertDialog

FreeIdea Tech

ProgressDialog

- ProgressDialog可用於在等待其他程式或是在等待上傳時使用

- ProgressDialog有兩種形式
  - 圓型 ProgressDialog
  - 長條型 ProgressDialog

FreeIdea-Tech

ProgressDialog

FreeIdea-Tech

ProgressDialog

程式碼(AlertDialogExample01.java)：

```
public class ProgressDialogExample extends Activity {
   int nowProgressStatus = 0;
    Handler myHandler = new Handler();
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);
      final ProgressDialog dialog01 = ProgressDialog.show(this, "", "程式正在開啟中...", false);
      Thread thread01 = new Thread(new Runnable() {
       public void run() {
            while (nowProgressStatus < 100) {
               try {
            Thread.sleep(500);
               }
               catch (InterruptedException e) {
                  // TODO Auto-generated catch block
                    e.printStackTrace();
               }
                nowProgressStatus+=5;
            } // While
            myHandler.post(new Runnable() {
               public void run() {
                              dialog01.cancel();
               }
            });
         }
      });
      thread01.start();
   }
}
```

圓形的ProgressDialog

FreeIdea-Tech

ProgressDialog

FreeIdea-Tech

ProgressDialog

程式碼(AlertDialogExample02.java)：

```
public class ProgressDialogExample02 extends Activity {
    int nowProgressStatus = 0;
    Handler myHandler = new Handler();
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);
      final ProgressDialog dialog02 = new ProgressDialog(this);
      dialog02.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
      dialog02.setMessage("讀取中...");
      dialog02.show();
      Thread thread02 = new Thread(new Runnable() {
         public void run() {
            while (nowProgressStatus < 100) {
               try {
                     Thread.sleep(500);  }
               catch (InterruptedException e) {
                    e.printStackTrace(); }
                nowProgressStatus+=5;
                myHandler.post(new Runnable() {
                public void run() {
                    dialog02.setProgress(nowProgressStatus); }
               });
            } // While
            myHandler.post(new Runnable() {
               public void run() {
                     dialog02.cancel();
            }});
      }});
      thread02.start();
   }
}
```

FreeIdeaTech

選單元件

- Options Menu
- Context Menu
- Sub Menu
- Spinner

FreeIdea-Tech

Options Menu

- Options Menu就是透過按手機或模擬器上的menu鍵顯示的功能表
  - 最多可以顯示六項，稱為Icon Menu
  - 超過六項就會以More的功能項來表示其餘的選項，稱為Expanded Menu。

FreeIdeaTech

Options Menu

此種型式的Menu就稱為OptionsMenu

FreeIdea-Tech

Options Menu

程式碼-1(OptionsMenuEX.java)：

```
public class OptionsMenu extends Activity
{
   public static final int aboutBtnID = Menu.FIRST;
   public static final int exitBtnID = Menu.FIRST + 1;
   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);
   }

   public boolean onCreateOptionsMenu(Menu menu)
   {
      menu.add(0, aboutBtnID, 0, "About");
      menu.add(0, exitBtnID, 0, "Exit");
      return true;
   }
```

FreeIdea-Tech

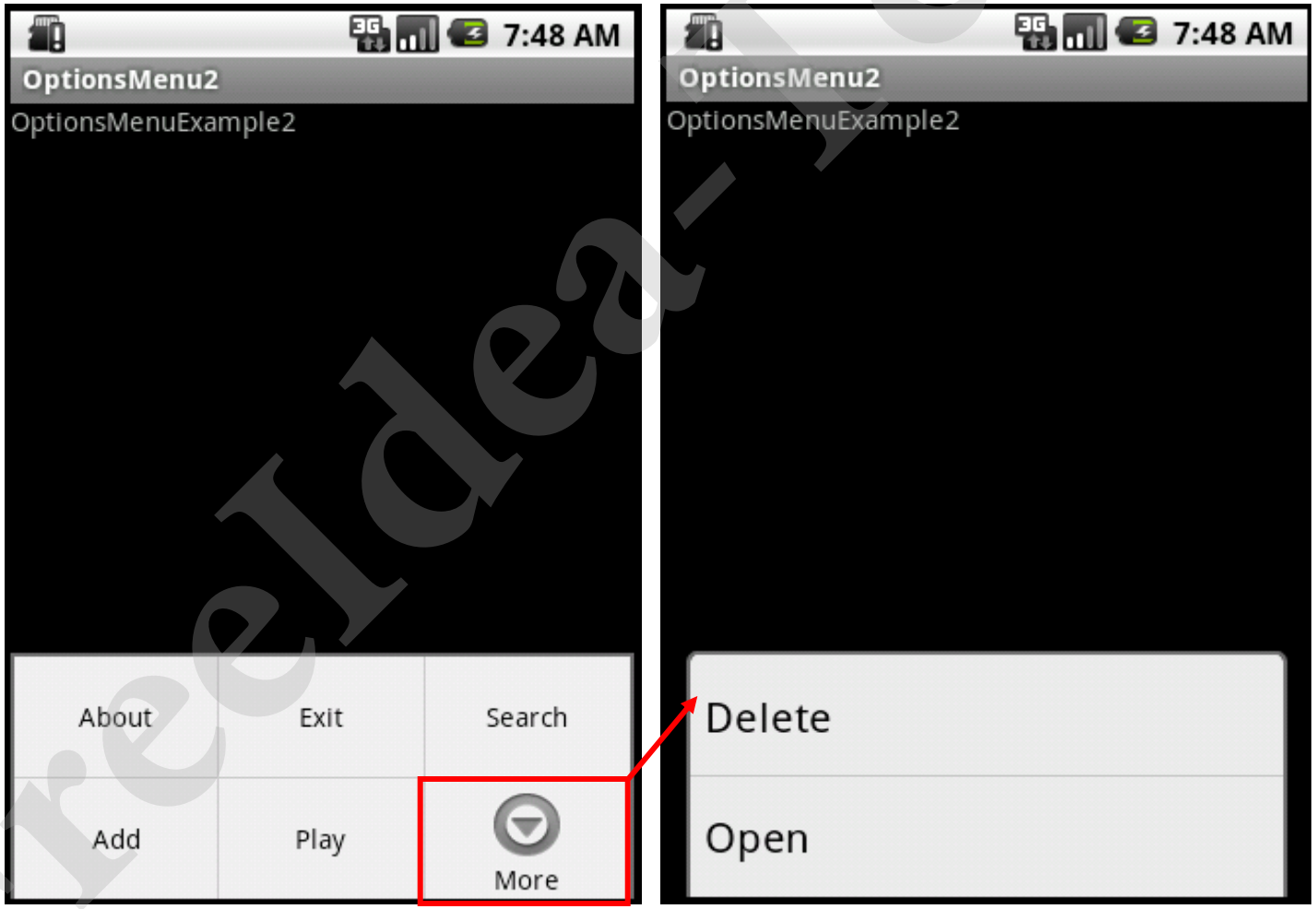
Options Menu

程式碼-2(OptionsMenuEX.java)：

```
public boolean onOptionsItemSelected(MenuItem item)     {
        super.onOptionsItemSelected(item);
        switch( item.getItemId() )  {
           case aboutBtnID:
                    openDialog();
                    break;
           case exitBtnID:
                    finish();
                    break;
        }
        return true;
    }
    public void openDialog()   {
           AlertDialog.Builder builder = new AlertDialog.Builder(this);
           builder.setMessage("About OptionsMenu");
           builder.setTitle("About");
           builder.setPositiveButton("OK",
           new DialogInterface.OnClickListener()  {

              @Override
              public void onClick(DialogInterface dialog, int which) {
              }

           }).show();
    }
}
```

Free Idea-Tech

Options Menu



FreeIdea-Tech

Options Menu

程式碼-1(OptionsMenuEX.java)：

```
public class OptionsMenu2 extends Activity
{
    public static final int aboutBtnID = Menu.FIRST;
    public static final int exitBtnID = Menu.FIRST + 1;
    public static final int searchBtnID = Menu.FIRST + 2;
    public static final int addBtnID = Menu.FIRST + 3;
    public static final int playBtnID= Menu.FIRST + 4;
    public static final int delBtnID = Menu.FIRST + 5;
    public static final int openBtnID = Menu.FIRST + 6;
  /** Called when the activity is first created. */
  @Override
  public void onCreate(Bundle savedInstanceState)
  {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
  }

  public boolean onCreateOptionsMenu(Menu menu)
  {
    menu.add(0, aboutBtnID, 0, "About");
    menu.add(0, exitBtnID, 0, "Exit");
    menu.add(0, searchBtnID, 0, "Search");
    menu.add(0, addBtnID, 0, "Add");
    menu.add(0, playBtnID, 0, "Play");
    menu.add(0, delBtnID, 0, "Delete");
    menu.add(0, openBtnID, 0, "Open");
    return true;
  }
```

FreeIdeaTech

Options Menu

程式碼-2(OptionsMenuEX2.java)：

```
public boolean onOptionsItemSelected(MenuItem item)
   {
      super.onOptionsItemSelected(item);
      switch( item.getItemId() )
      {
         case aboutBtnID:
            openDialog();
                  break;
         case exitBtnID:
                  finish();
                  break;
      }
      return true;
   }
   public void openDialog()
   {
       AlertDialog.Builder builder = new AlertDialog.Builder(this);
       builder.setMessage("About OptionsMenu");
       builder.setTitle("About");
       builder.setPositiveButton("OK",  new DialogInterface.OnClickListener()   {

          @Override
           public void onClick(DialogInterface dialog, int which)    {
           }

         }).show();
   }
}
```

Free Idea-Tech

Options Menu

FreeIdea-Tech

Options Menu

程式碼(OptionsMenuEX.java)：

```
public boolean onCreateOptionsMenu(Menu menu) {
    menu.add(0, aboutBtnID, 0, "About").setIcon(R.drawable.icon);
    menu.add(0, exitBtnID, 0, "Exit").setIcon(R.drawable.icon);
    return true;
}
```

透過setIcon()方法在功能表中增加圖示

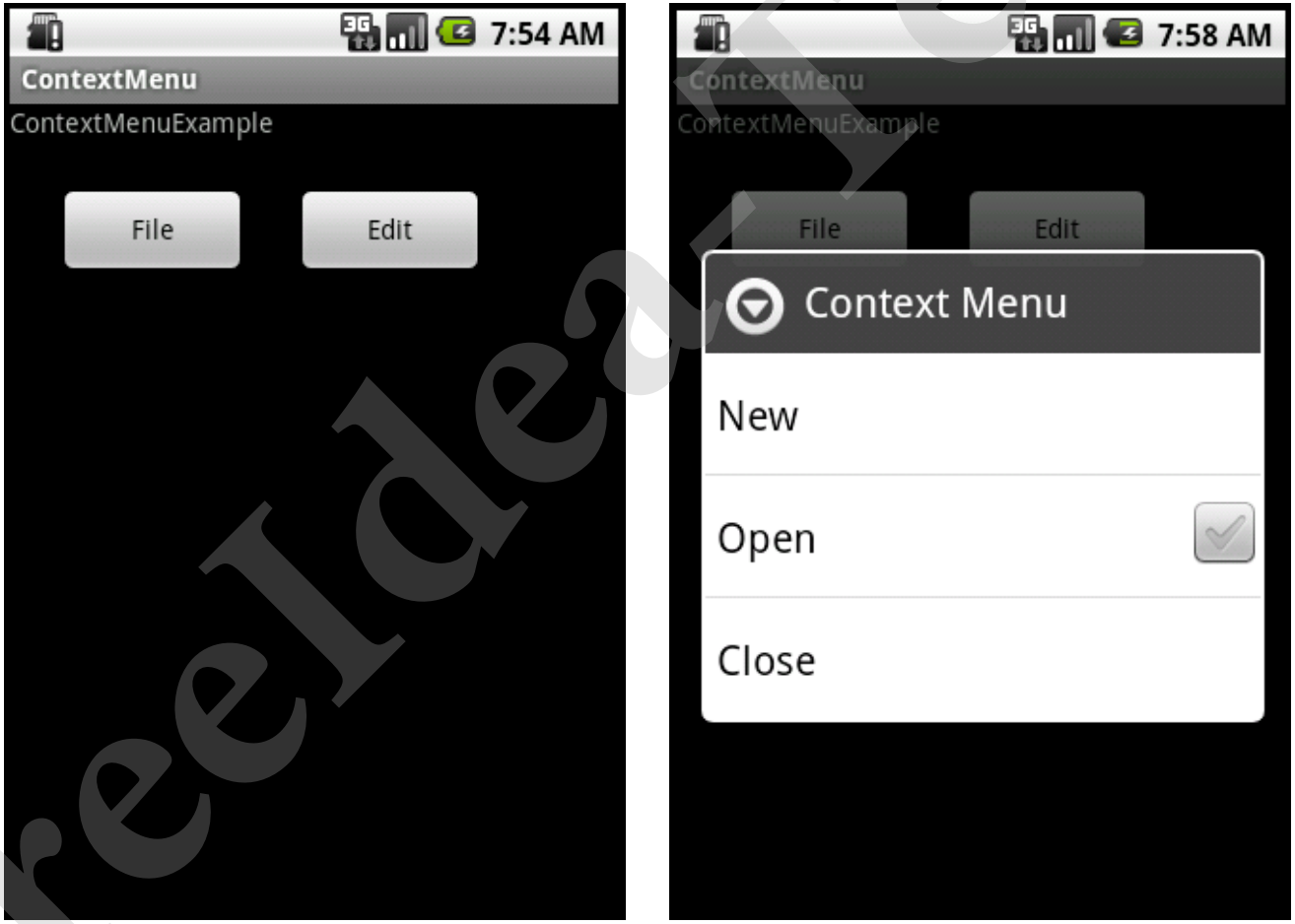
FreeIdeaTech

Context Menu

- Android中的Context Menu與一般個人電腦中的滑鼠右鍵非常類似
  - 當在View上，使用者長按螢幕不放兩秒，將會出現一個浮動功能表
  - Context Menu不支援圖示或快捷鍵

FreeIdeaTech

Context Menu



FreeIdea Tech

Context Menu

程式碼(ContextMenuEX.java)：

```
public class ContextMenu extends Activity {
    public String checkedItem = "";
    public static final int newBtnID = Menu.FIRST;
    public static final int openBtnID = Menu.FIRST + 1;
    public static final int closeBtnID = Menu.FIRST + 2;
    @Override
    public void onCreate(Bundle savedInstanceState)          {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button fileBtn = (Button)findViewById(R.id.fileBtn);
        Button editBtn = (Button)findViewById(R.id.editBtn);
        registerForContextMenu(fileBtn);
        registerForContextMenu(editBtn);
    }
    public void onCreateContextMenu(android.view.ContextMenu conMenu, View v, ContextMenuInfo menuInfo)  {
        super.onCreateContextMenu(conMenu, v, menuInfo);
        conMenu.setHeaderTitle("Context Menu");
        conMenu.add(0, newBtnID, 0, "New").setNumericShortcut('1');
        conMenu.add(0, openBtnID, 0, "Open").setCheckable(true);
        conMenu.add(0, closeBtnID, 0, "Close").setIcon(R.drawable.icon);
    }
    public boolean onContextItemSelected(MenuItem item)
    {
        super.onContextItemSelected(item);
        return false;
    }
}
```

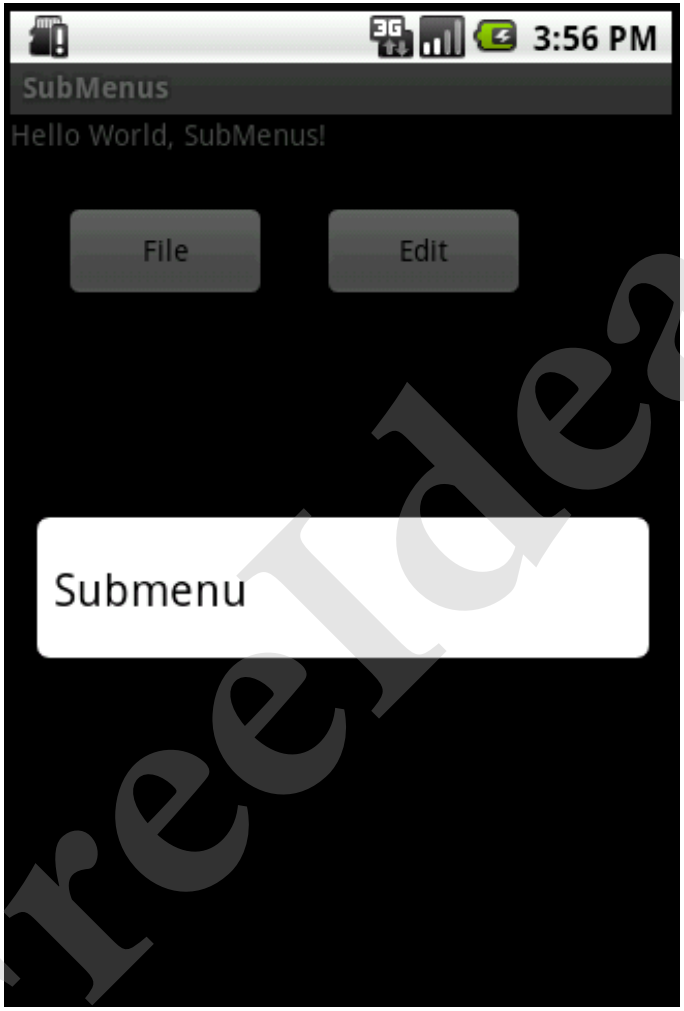
FreeIdea-Tech

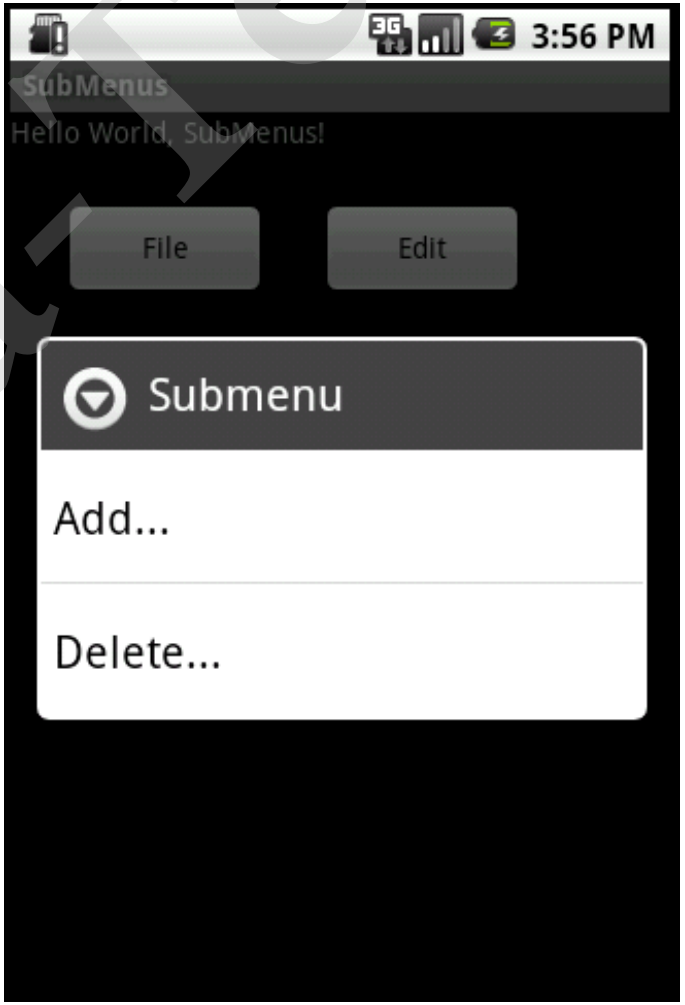
Sub Menu

- 一個子功能表可以被加入任何功能表中，但不能加入另外的子功能表中。
- 當應用程式有很多功能可以被分類時，會非常好用，就像是一般電腦的功能表。

FreeIdea-Tech

Sub Menu





FreeIdea Tech

Sub Menu

程式碼-1(SubMenuEX.java)：

```
public class SubMenus extends Activity
{
    public String checkedItem = "";
    public static final int newBtnID = Menu.FIRST;
    public static final int openBtnID = Menu.FIRST + 1;
    public static final int closeBtnID = Menu.FIRST + 2;
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button fileBtn = (Button)findViewById(R.id.fileBtn);
        Button editBtn = (Button)findViewById(R.id.editBtn);
        registerForContextMenu(fileBtn);
        registerForContextMenu(editBtn);
    }
```

FreeIdea Tech

Sub Menu

程式碼-2(SubMenuEX.java)：

```
        public void onCreateContextMenu(android.view.ContextMenu conMenu,View v, ContextMenuInfo menuInfo)
        {
                super.onCreateContextMenu(conMenu, v, menuInfo);

        /*      conMenu.setHeaderTitle("Context Menu");
                conMenu.add(0, newBtnID, 0, "New").setNumericShortcut('1'); //利用數字選擇
                conMenu.add(0, openBtnID, 0, "Open").setCheckable(true); //設定checkbox
                conMenu.add(0, closeBtnID, 0, "Close").setIcon(R.drawable.icon); //ContextMenu不支援icon*/
                    SubMenu sub = conMenu.addSubMenu("Submenu");
                sub.add("Add...");
                sub.add("Delete...");
        }
        public boolean onContextItemSelected(MenuItem item)
        {
                super.onContextItemSelected(item);
                return false;
        }
}
```

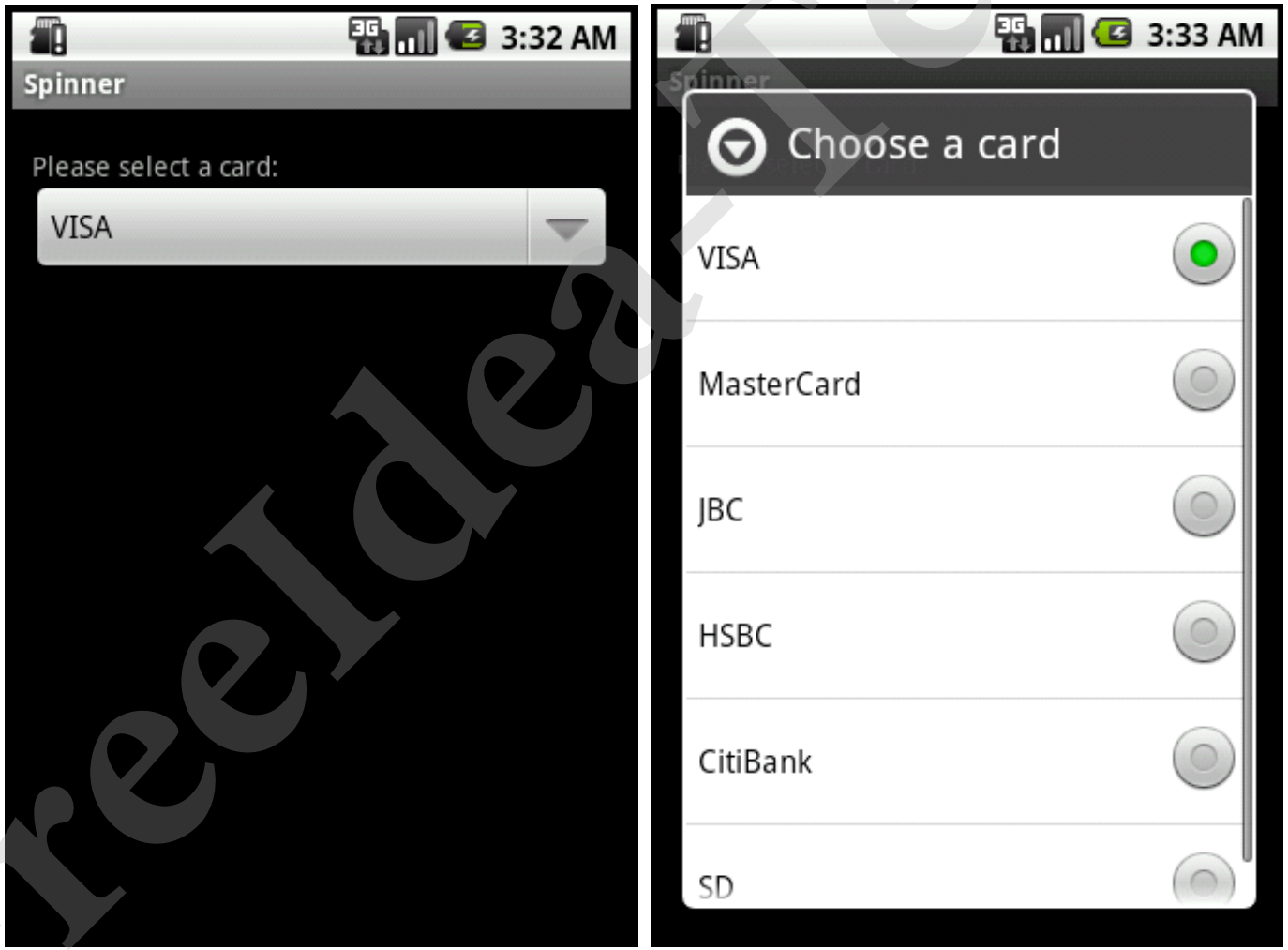
FreeEdaTech

Spinner

- Spinner是一個widget，允許使用者從一群選項中選出其中一個。
- Spinner就類似下拉式選單，當列表超過螢幕大小時也可允許捲動。

FreeIdea-Tech

Spinner



FreeIdea Tech

Spinner

布局文件-1(res/layout/main.xml)：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="10dip"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dip"
        android:text="Please select a planet:"
    />
    <Spinner
        android:id="@+id/spinner"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:drawSelectorOnTop="true"
        android:prompt="@string/planet_prompt"
    />
</LinearLayout>
```

首先我們將main.xml改寫成下列範例中的程式碼

FreeIdea-Tech

Spinner

布局文件(res/value/strings.xml)：

```
<string name="planet_prompt">Choose a item</string>
```

其中我們新增了Spinner的標籤，
接著新增這段程式碼至strings.xml中。

布局文件(res/value/arrays.xml)：

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
   <string-array name="cards">
      <item>VISA</item>
      <item>MasterCard</item>
      <item>JBC</item>
      <item>HSBC</item>
      <item>CitiBank</item>
      <item>SD</item>
   </string-array>
</resources>
```

接著在res/value/中建立一下名為
arrays.xml的文件，arrays.xml程式碼如下

Freeda-Tech

Spinner

程式碼-1(SpinnerEX.java)：

```
package ncu.bnlab.SpinnerExample;

import android.app.Activity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.Spinner;

public class SpinnerEX extends Activity
{
   /** Called when the activity is first created. */
   @SuppressWarnings("unchecked")
      @Override
public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);

      Spinner s = (Spinner)findViewById(R.id.spinner);
```

最後將Spinner的JAVA原始檔開啟，在onCreate()中加入程式碼。

FreeIdea-Tech

Spinner

程式碼-2(SpinnerEX.java)：

```
        /* 設定功能表項目陣列，使用createFromResource方法 */
        ArrayAdapter adapter = ArrayAdapter.createFromResource(
                this, R.array.cards, android.R.layout.simple_spinner_item);

        /* 設定選單 */
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

        /* 設定Adapter */
        s.setAdapter(adapter);
    }
}
```

FreeIdea-Tech

顯示元件

- ImageView
- ListView
- GridView
- WebView

FreeIdea-Tech

ImageView

- 用於顯示一個圖樣，例如說一個圖示或圖片。
- ImageView類別可以載入來自不同來源的圖片(可能來自資源或內容提供商)，它可以用於任何佈局管理器，並提供多種顯示選項，例如：縮放、著色。

FreeIcon Tech

ImageView

FreeIda-Tech

ImageView

程式碼-1(ImageViewEX.java)：

```
public class ImageViewEX extends Activity
{
   @Override
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      ImageView imageView = new ImageView(this);
      imageView.setImageResource(R.drawable.christmas);
      setContentView(imageView);
   }
}
```

將drawable中的圖檔資源置入imageView的圖片來源。

FreeIdea-Tech

ListView

- Android 除了提供CheckBox與RadioButton的物件之外，還提供另一種更為方便且表現多樣化的物件 – ListView

FreeIdea-Tech

ListView

FreeIdea-Tech

ListView

程式碼-1(ListViewEX.java)：

```
public class ListViewEX extends Activity
{
   @Override
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      String[] Data = {"網路概論","通訊網路實驗","通訊概論"};
      ListView listView = new ListView(this);
      // 設定ListView選擇的方式：
      // 單選：ListView.CHOICE_MODE_SINGLE
      // 多選：ListView.CHOICE_MODE_MULTIPLE
      listView.setChoiceMode( ListView.CHOICE_MODE_SINGLE );
      ArrayAdapter vArrayData = new ArrayAdapter(
            this
            , android.R.layout.simple_list_item_single_choice
            , Data
            );
      // 設定ListView的接收器，做為選項的來源
      listView.setAdapter( vArrayData );
      // ListView 設定為 ContentView
      setContentView(listView);
   }
}
```

FreeIdea-Tech

ListView

程式碼-2(ListViewEX.java)：

```
public class ListViewEX2 extends ListActivity
{
   @Override
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      String[] Data = { "葷食","素食","兩者皆可" };
      ListView listView = this.getListView();
      listView.setChoiceMode( ListView.CHOICE_MODE_SINGLE );
      // ListView 的選項來源由陣列提供
      this.setListAdapter( new ArrayAdapter( this
                                          ,
android.R.layout.simple_list_item_single_choice
                                 , Data
                                 ));
   }
}
```

Freeida Tech

GridView

- GridView
  可以顯示一個二維滾動格框，也可搭配其他元件，呈現類似一般手機上的功能表。

FreeIdeaTech

GridView

FreeIdea Tech

GridView

GridView程式碼-1(GridViewEX.java)：

```
public class GridViewEX extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
        {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        GridView gridview = (GridView) findViewById(R.id.gridview);
        gridview.setAdapter(new ImageAdapter(this));
    }
}
```

FreeIdea-Tech

GridView

ImageAdapter程式碼-1(ImageAdapter.java)：

```
public class ImageAdapter extends BaseAdapter
{
    private Context mContext;
    public ImageAdapter(Context c)
    {
        mContext = c;
    }
    public int getCount()
    {
        return mThumbIds.length;
    }
    public Object getItem(int position)
    {
        return null;
    }
    public long getItemId(int position)
    {
        return 0;
    }
```

Freeidea-Tech

GridView

ImageAdapter程式碼-2(ImageAdapter.java)：

```
public View getView(int position, View convertView, ViewGroup parent)
    {
        ImageView imageView;
        if (convertView == null)
        {
            imageView = new ImageView(mContext);
            imageView.setLayoutParams(new GridView.LayoutParams(85, 85));
            imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
            imageView.setPadding(8, 8, 8, 8);
        }
        else
        {
            imageView = (ImageView) convertView;
        }
        imageView.setImageResource(mThumbIds[position]);
        return imageView;
    }
    private Integer[] mThumbIds =
    {
            R.drawable.icon, R.drawable.christmas,
            R.drawable.favorite, R.drawable.ipod
    };
}
```

Freeder-Tech

WebView

- Webview是用於顯示web頁面的視景。
- 在活動中的任何一個瀏覽器或簡易地顯示一些線上內容。
- 它使用Webkit繪圖引擎去顯示頁面且包含
  - 上一頁
  - 下一頁
  - 放大縮小
  - 文字搜尋…等等方法

FreIdea-Tech

WebView

FreeIdea-Tech

WebView

```
<uses-permission android:name="android.permission.INTERNET" />
```

為了使活動可以在WebView存取Internet及讀取web頁面，需在AndroidManifest.xml中加入INTERNET權限

佈局文件(res/layout/main.xml)：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <WebView
        android:id="@+id/webview"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
    />
</LinearLayout>
```

FreeDart Tech

WebView

程式碼(WebViewEX.java)：

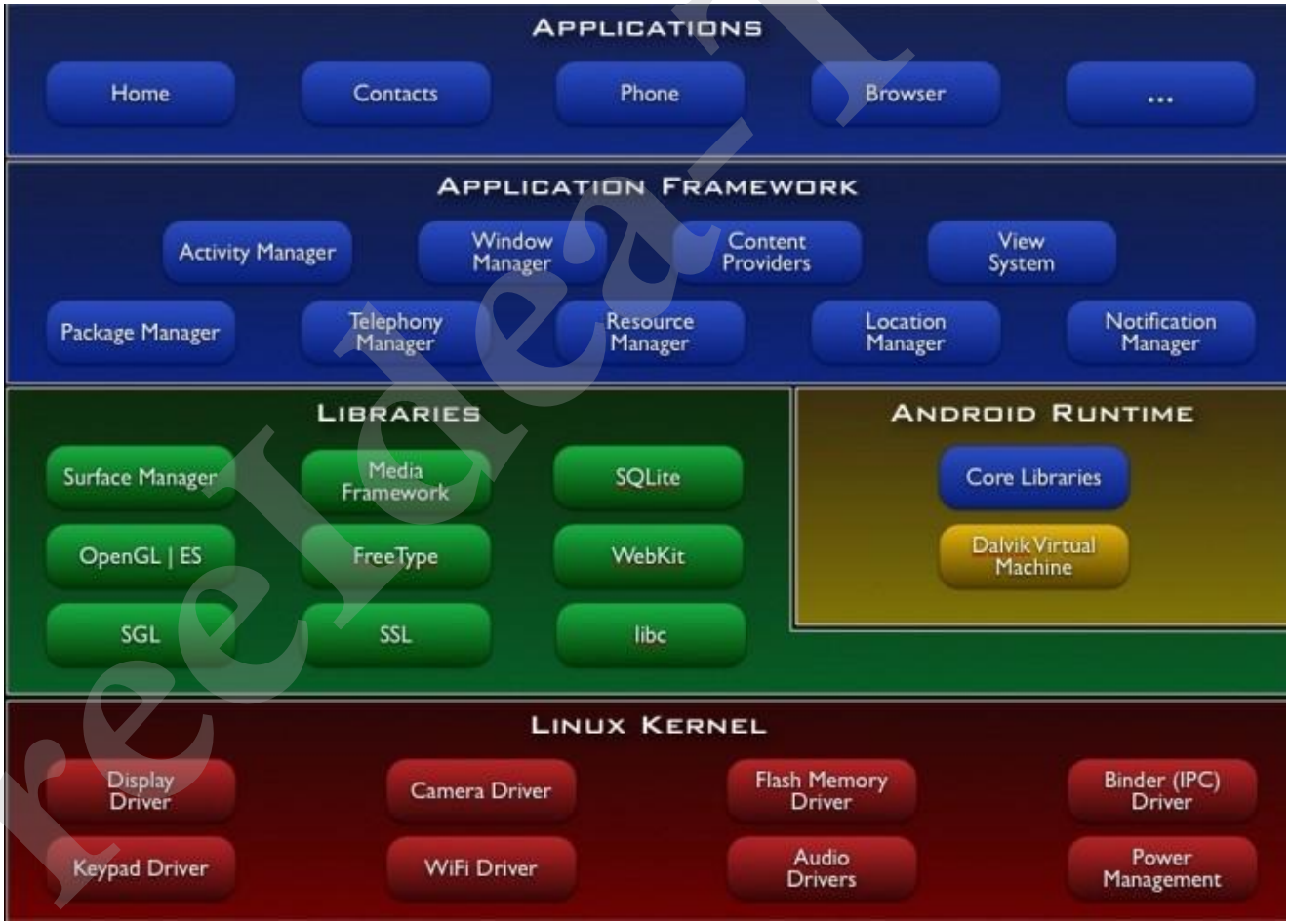
```
public class WebViewEX extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        WebView webview;
        webview = (WebView) findViewById(R.id.webview);
        webview.getSettings().setJavaScriptEnabled(true);
        webview.loadUrl("http://www.google.com");
    }
}
```

此處以Google網站作為連接範例

FreeIdea-Tech

核心框架

Android系統架構



FreeLearTech

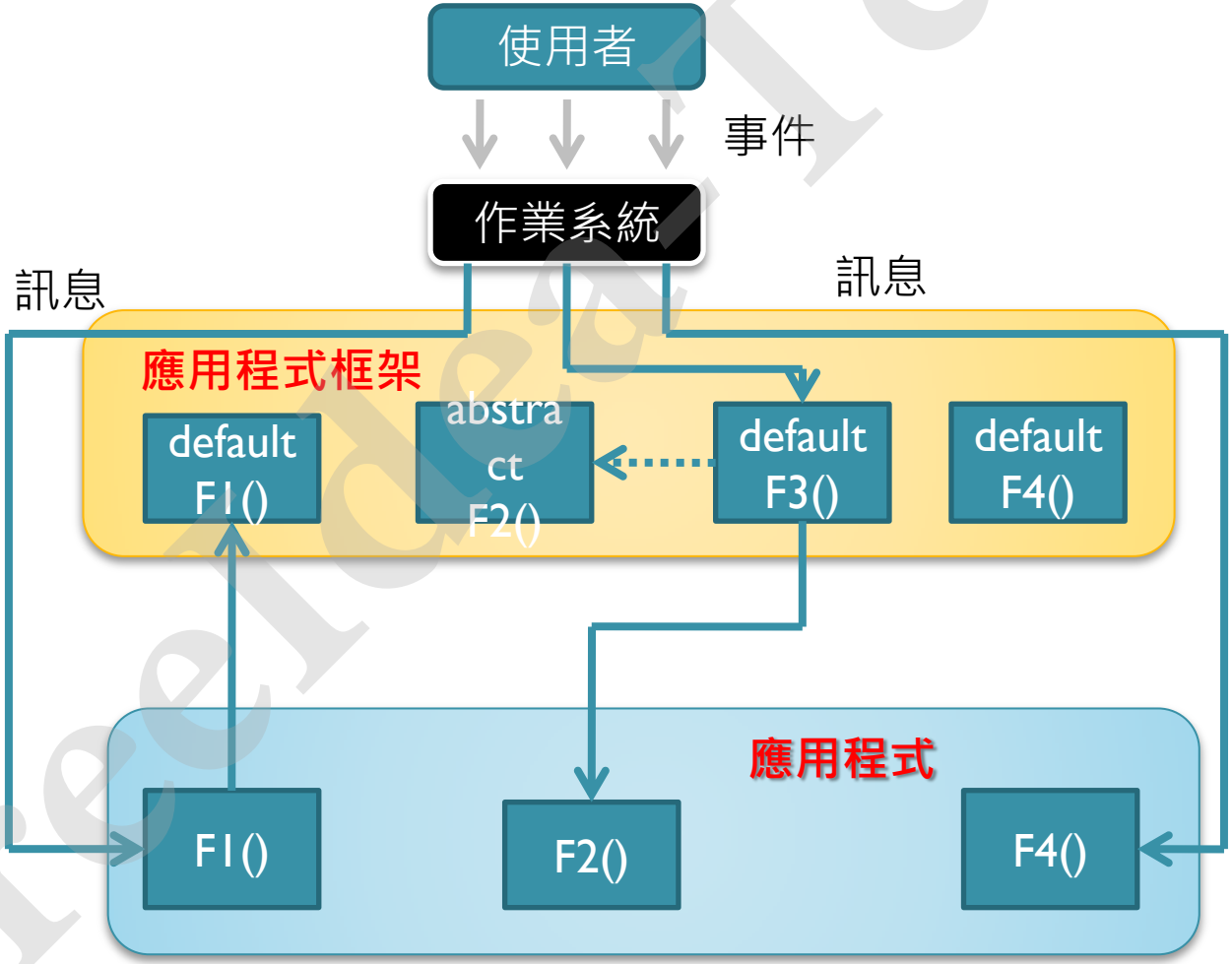
核心框架

(Application Framework)

- 開發人員可以使用核心應用程式所提供的API框架。
- 應用程式框架構簡化元件的重用性。
- 任何一個應用程式都可以發佈它的功能，且其他應用程式也可以使用其所發佈的功能。
- 應用程式重用機制讓使用者方便地替換程式元件。

FreeIca-Tech

核心框架



FreeIdea Tech

核心框架

- 隱藏在每個應用程式背後是一系列的服務(Services)和活動(Activity)。
- 豐富的視圖(Views) 用來構建應用程式，包括：清單(Lists)、網格(Grids)、文字方塊(Text Boxes)、按鈕(Buttons)、可嵌入的Web瀏覽器。
- 內容提供器(Content Providers)使應用程式可以獲得另一個應用程式的資料(如聯繫人資料庫)，或者共用資料。
- 資源管理器(Resource Manager)提供非程式碼資源的訪問，如本地字串、圖形、和佈局檔(layout files)。
- 通知管理器(Notification Manager)使得應用程式可以在狀態列中顯示自訂的提示資訊。
- 活動管理器(Activity Manager)管理應用程式生命週期並提供常用的導航返回功能。

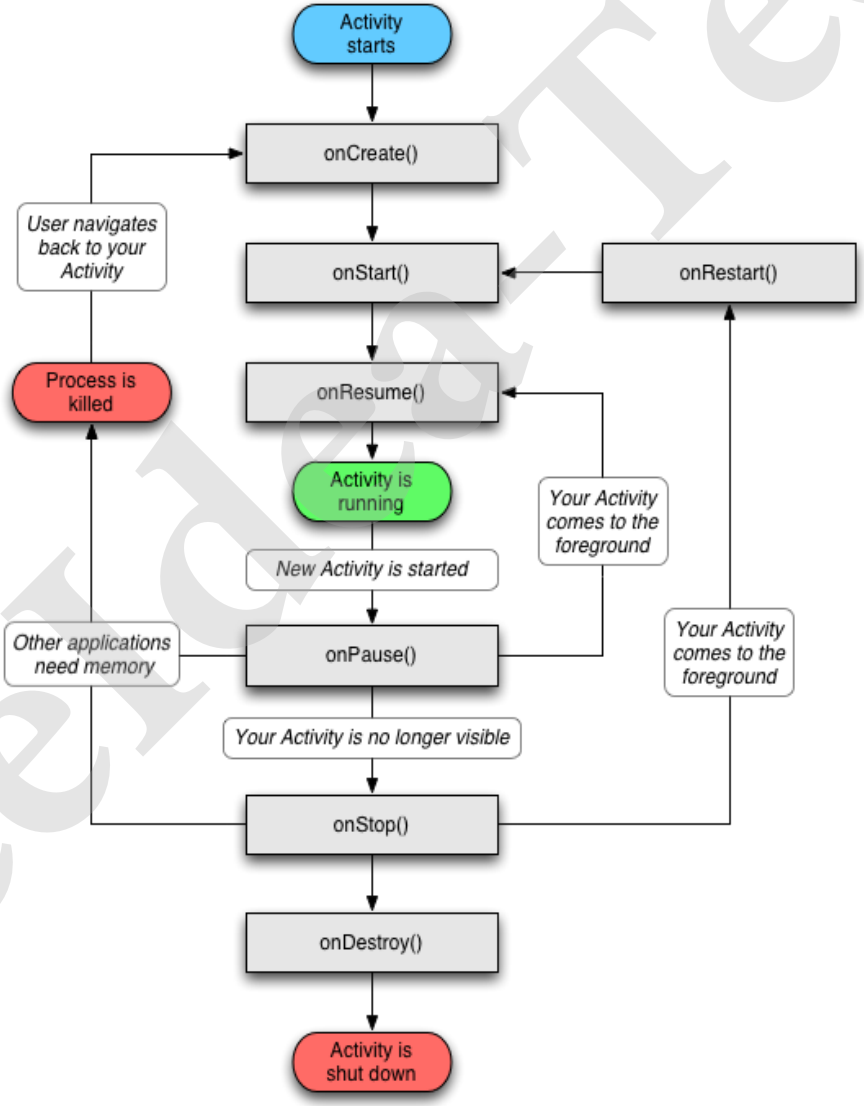
Feeder Tech

程式生命週期

- **程式的生命週期(Life Cycle)**
  - **活動 (Activity)**
    - 一個Activity基本上有三個生命狀態：
      - active或running
      - Paused
      - Stop
    - 當一個Activity處於Pause或Stop的狀態時，系統可以要求Activity結束或移除它，當它再度呈現在使用者面前時，要能完整的重新啟動及回復先前的狀態。
    - 應用程式存在與否並非由應用程式所自行決定，而是由Android系統透過運行機制決定。

Free Idea Tech

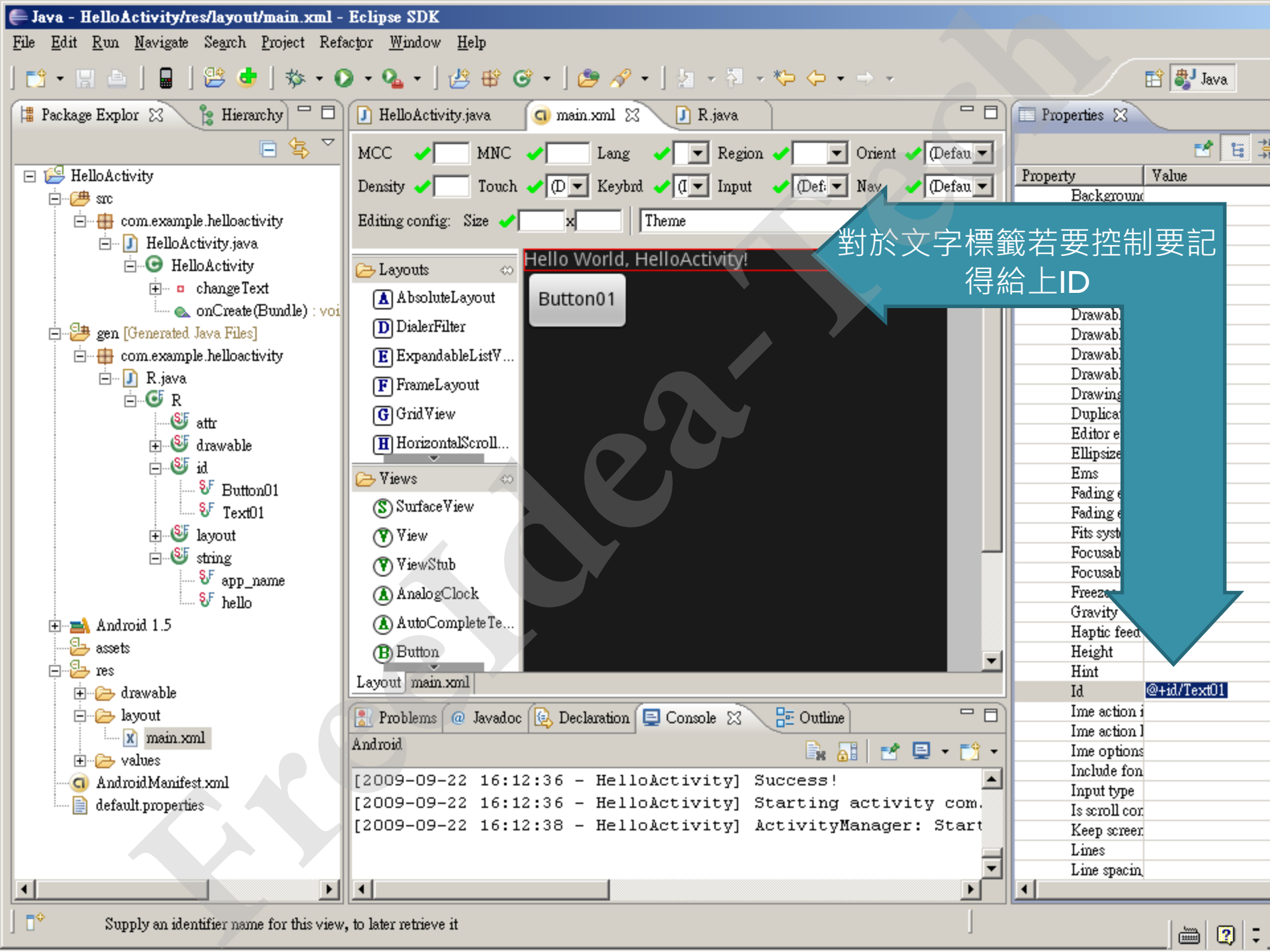
程式生命週期

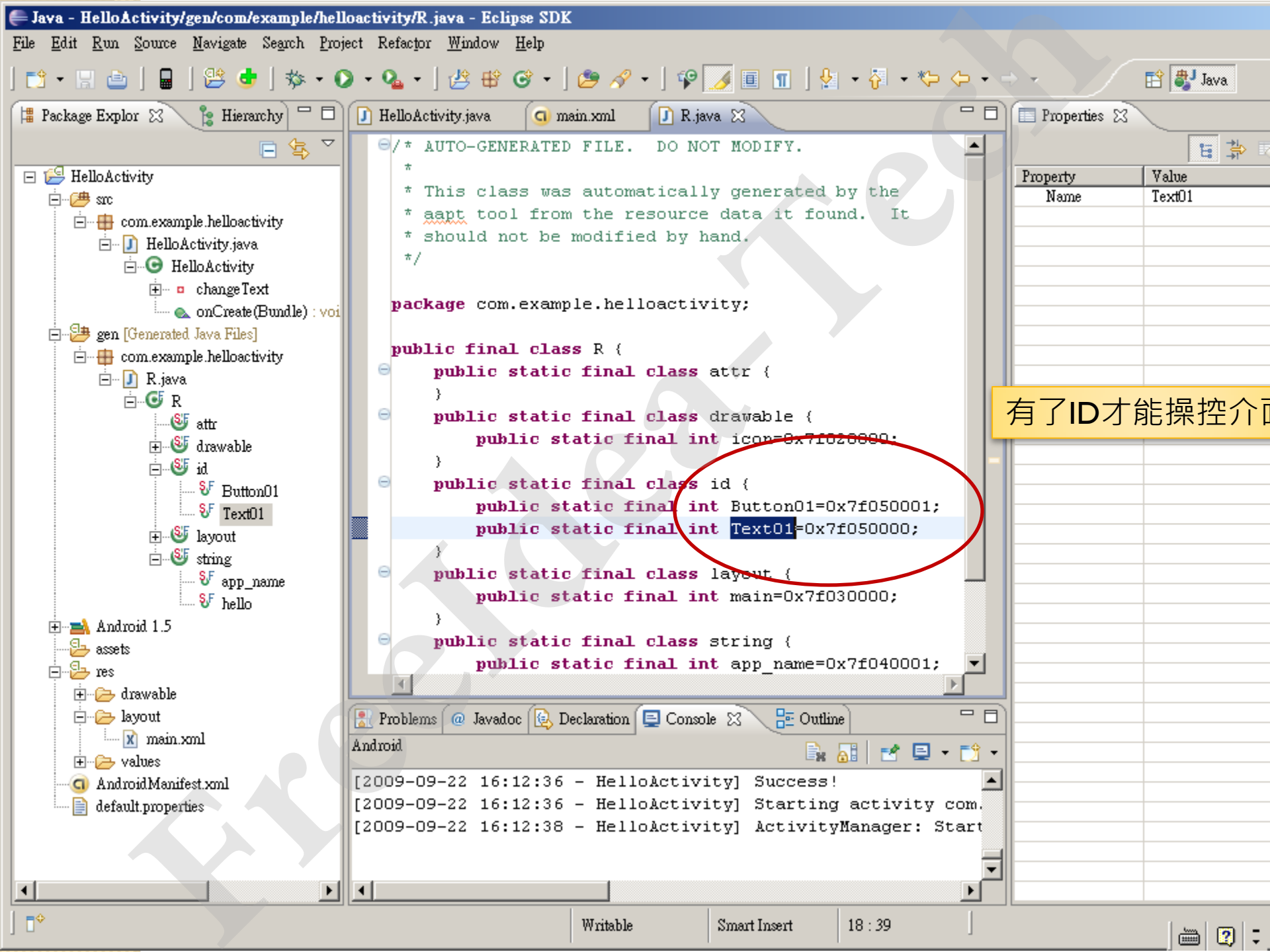


FreeIdeaTech

畫面切換的流程

FreeIdea-Tech


Java - HelloActivity/res/layout/main.xml - Eclipse SDK
File Edit Run Navigate Search Project Refactor Window Help
Package Explor
Hierarchy
HelloActivity
src
com.example.helloactivity
HelloActivity.java
HelloActivity
changeText
onCreate(Bundle) : voi
gen [Generated Java Files]
com.example.helloactivity
R.java
R
attr
drawable
id
Button01
Text01
layout
string
app_name
hello
Android 1.5
assets
res
drawable
layout
main.xml
values
AndroidManifest.xml
default.properties
HelloActivity.java
main.xml
R.java
MCC MNC Lang Region Orient
Density Touch Keybrd Input Nav
Editing config: Size x Theme
Layouts
AbsoluteLayout
DialerFilter
ExpandableListV...
FrameLayout
GridView
HorizontalScroll...
Views
SurfaceView
View
ViewStub
AnalogClock
AutoCompleteTe...
Button
Hello World, HelloActivity!
Button01
Layout
main.xml
Problems
Javadoc
Declaration
Console
Outline
Android
[2009-09-22 16:12:36 - HelloActivity] Success!
[2009-09-22 16:12:36 - HelloActivity] Starting activity com.
[2009-09-22 16:12:38 - HelloActivity] ActivityManager: Start
Properties
Property
Value
Id
@+id/Text01
Supply an identifier name for this view, to later retrieve it
對於文字標籤若要控制要記得給上ID


Java - HelloActivity/gen/com/example/helloactivity/R.java - Eclipse SDK
File Edit Run Source Navigate Search Project Refactor Window Help
Package Explor
Hierarchy
HelloActivity
src
com.example.helloactivity
HelloActivity.java
HelloActivity
changeText
onCreate(Bundle) : voi
gen [Generated Java Files]
com.example.helloactivity
R.java
R
attr
drawable
id
Button01
Text01
layout
string
app_name
hello
Android 1.5
assets
res
drawable
layout
main.xml
values
AndroidManifest.xml
default.properties
HelloActivity.java
main.xml
R.java
/* AUTO-GENERATED FILE. DO NOT MODIFY.
*
* This class was automatically generated by the
* aapt tool from the resource data it found. It
* should not be modified by hand.
*/
package com.example.helloactivity;
public final class R {
public static final class attr {
}
public static final class drawable {
public static final int icon=0x7f020000;
}
public static final class id {
public static final int Button01=0x7f050001;
public static final int Text01=0x7f050000;
}
public static final class layout {
public static final int main=0x7f030000;
}
public static final class string {
public static final int app_name=0x7f040001;
Properties
Property
Value
Name
Text01
有了ID才能操控介面
Problems
Javadoc
Declaration
Console
Outline
Android
[2009-09-22 16:12:36 - HelloActivity] Success!
[2009-09-22 16:12:36 - HelloActivity] Starting activity com.
[2009-09-22 16:12:38 - HelloActivity] ActivityManager: Start
Writable
Smart Insert
18 : 39

畫面切換的流程

- 註冊按鍵事件處理程序

```
// Create an anonymous implementation of OnClickListener
private OnClickListener mCorkyListener = new OnClickListener() {
    public void onClick(View v) {
      // do something when the button is clicked
    }
};

protected void onCreate(Bundle savedValues) {
    ...
    // Capture our button from layout
    Button button = (Button)findViewById(R.id.corky);
    // Register the onClick listener with the implementation above
    button.setOnClickListener(mCorkyListener);
    ...
}
```

方法一
不採用繼承介面

```
public class ExampleActivity extends Activity implements OnClickListener {
    protected void onCreate(Bundle savedValues) {
        ...
        Button button = (Button)findViewById(R.id.corky);
        button.setOnClickListener(this);
    }

    // Implement the OnClickListener callback
    public void onClick(View v) {
      // do something when the button is clicked
    }
    ...
}
```

方法二
採用繼承介面

FreeIdea-Tech

畫面切換的流程

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
public class HelloActivity extends Activity {
    /** Called when the activity is first created. */
@Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button button = (Button)findViewById(R.id.Button01);
        button.setOnClickListener(changeText);
        }
private OnClickListener changeText = new OnClickListener()
    {
                public void onClick(View v)
               {
                            TextView result = (TextView)findViewById(R.id.Text01);
                            result.setText("You are happy!");
               }
    };
}
```

紅色粗體是新增程式

方法一
不採用繼承介面

Freelea-Tech

畫面切換的流程

```
package com.example.helloactivity;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
public class HelloActivity extends Activity implements  OnClickListener {
    /** Called when the activity is first created. */
@Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button button = (Button)findViewById(R.id.Button01);
        button.setOnClickListener(this);
        }
     public void onClick(View v) {
     TextView result = (TextView)findViewById(R.id.Text01);
    result.setText("You are happy!");
    }
}
```

紅色粗體是新增程式

方法二
採用繼承介面

Freelea-Tech

```
package com.example.hello;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
public class HelloActivity extends Activity implements OnClickListener{
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button button1=(Button) findViewById(R.id.Button01
        button1.setOnClickListener(this);
        Button button2=(Button) findViewById(R.id.Button02
        button2.setOnClickListener(this);
    }
    public void onClick(View v){
     switch(v.getId())
     {
     case R.id.Button01:
         TextView result = (TextView) findViewById(R.id.Text
         result.setText("123");
         break;
       :
    }
}
```

利用getId()及switch
來處理不同按鈕的功能

```
public void onClick(View v){
 switch(v.getId())
 {
 case R.id.Button01:
    TextView result = (TextView) findViewById(R.id.Text01);
    result.setText("123");
    break;
 case R.id.Button02:
        setContentView(R.layout.first);
        Button button3=(Button) findViewById(R.id.Button03);
        button3.setOnClickListener(this);
        break;
 case R.id.Button03:
        setContentView(R.layout.main);
        Button button4=(Button) findViewById(R.id.Button01);
        button4.setOnClickListener(this);
        Button button5=(Button) findViewById(R.id.Button02);
        button5.setOnClickListener(this);
        break;
 }
}
```

FreeEdaTech

分組練習

- 每組1~3人
- 實作下列相關常用API各單元內任選一主題

表單元件	選單元件	顯示元件
Button	Options Menu	ImageView
ImageButton	Context Menu	ListView
RadioButton	Sub Menu	GridView
CheckBox	Spinner	WebView
DatePicker		
ProgressBar		
RatingBar		

- 相同的主題至多只能 2 組重複
- 內容可依各組能力做出屬於自己的style

FreeIdea-Tech

Android 圖形介面設計

吳柏翰

FreeIdea-Tech

Outline

- SQL Lite Database
- Activities
- Services
- Content Providers
- Broadcast Receivers
- Activating components: Intents
- 補充內容
- 專題製作

Freeida-Tech

SQL Lite Database

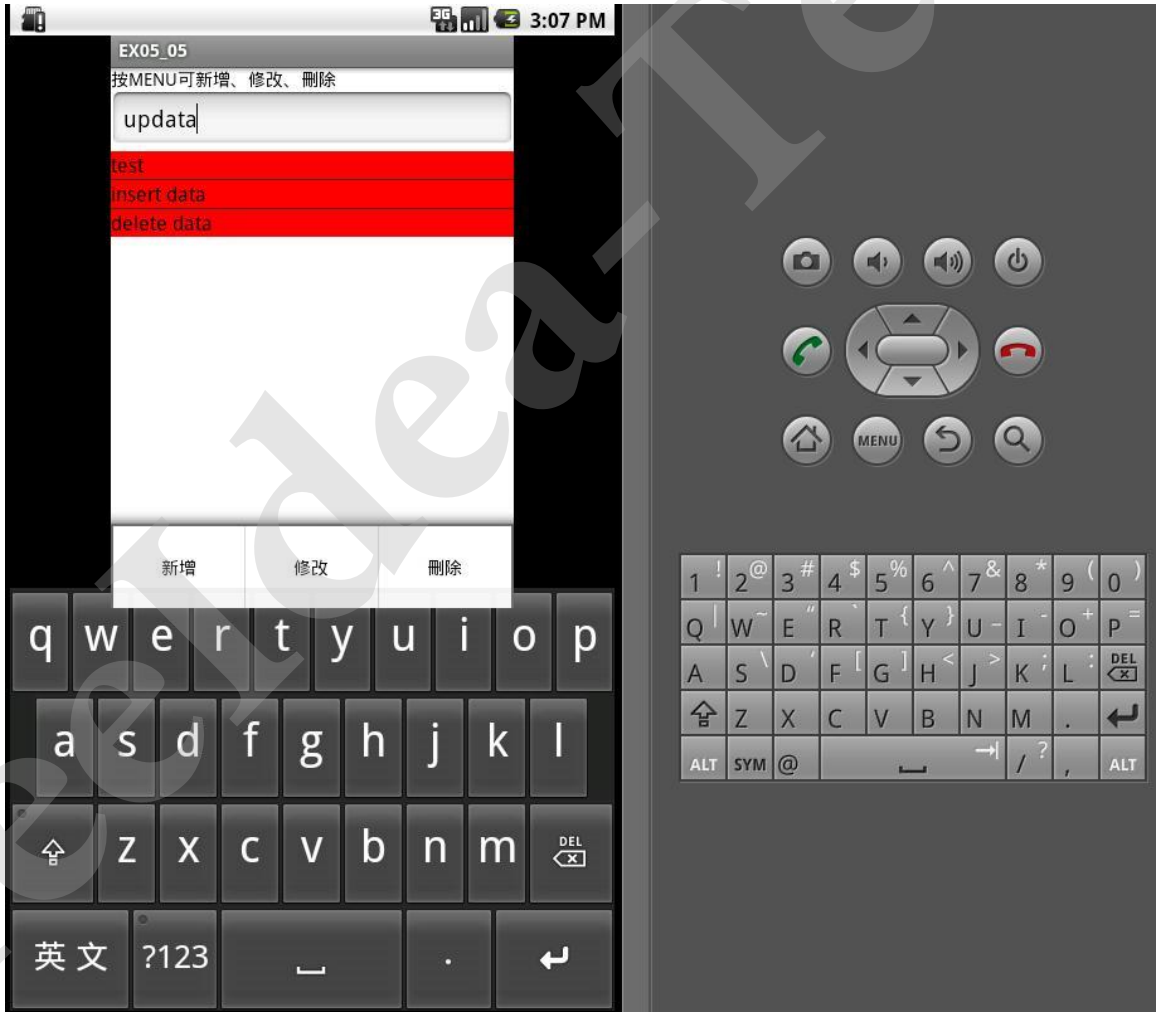
- 手機上Porttable(可攜式) SQLite(資料庫)，不外乎包含Query(查詢)、Insert(新增)、Update(修改) 和 Delete(刪除)，可用來設計放置行事曆或提醒使用者還有哪些工作事項。

- 以下範例為提供簡易Menu鍵做出資料庫的增、修、改、查功能，同時設計單層的Menu按鈕，使操作更直覺化，搭配ListView Wideget 來觸發點選事件。

- 範例中與User 互動的唯一管道是EditText Widget，在設計新增、修改的功能時，資料庫會參考EditText裡的值，並可以在EditText編輯工作事項，透過ListView顯示所有的工作事項。

FreeIdea-Tech

SQL Lite Database



FreeIdea-Tech

SQL Lite Database

EX05_05.java 範例

```
package  irdc.ex05_05;
import android.app.Activity;import android.database.Cursor;
import android.database.sqlite.SQLiteCursor;import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
public class EX05_05 extends Activity{
private ToDoDB myToDoDB;
private Cursor myCursor;
private ListView myListView;
private EditText myEditText;
private int _id;
 protected final static int MENU_ADD = Menu.FIRST;
protected final static int MENU_EDIT = Menu.FIRST + 1;
protected final static int MENU_DELETE = Menu.FIRST + 2;

@Override  public boolean onOptionsItemSelected(MenuItem item)  {
super.onOptionsItemSelected(item);
switch (item.getItemId())    {
case MENU_ADD:       this.addTodo();
       break;
    case MENU_EDIT:       this.editTodo();
       break;
    case MENU_DELETE:       this.deleteTodo();
       break;
   }
 return true;
}
```

Freeidea-Tech

SQL Lite Database

EX05_05.java 範例

```
@Override  public boolean onCreateOptionsMenu(Menu menu)  {
  super.onCreateOptionsMenu(menu);
   /* 新增三個MENU */
  menu.add(Menu.NONE, MENU_ADD, 0, R.string.strAddButton);
   menu.add(Menu.NONE, MENU_EDIT, 0, R.string.strEditButton);
  menu.add(Menu.NONE, MENU_DELETE, 0, R.string.strDeleteButton);
   return true;
 }
 /** Called when the activity is first created. */
@Override  public void onCreate(Bundle savedInstanceState)  {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.main);
   myListView = (ListView) this.findViewById(R.id.myListView);
   myEditText = (EditText) this.findViewById(R.id.myEditText);
  myToDoDB = new ToDoDB(this);
   /* 取得DataBase裡的資料 */
   myCursor = myToDoDB.select();
   /* new SimpleCursorAdapter並將myCursor傳入，顯示資料的欄位為todo_text */
   SimpleCursorAdapter adapter = new SimpleCursorAdapter(this, R.layout.list,       myCursor, new String[]       { ToDoDB.FIELD_TEXT }, new int[]
        { R.id.listTextView1 });
   myListView.setAdapter(adapter);
  /* 將myListView加入OnItemClickListener */
  myListView.setOnItemClickListener(new AdapterView.OnItemClickListener()   {
    @Override     public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,         long arg3)     {
      /* 將myCursor移到所點選的值 */
     myCursor.moveToPosition(arg2);
     /* 取得欄位_id的值 */
     _id = myCursor.getInt(0);
      /* 取得欄位todo_text的值 */
      myEditText.setText(myCursor.getString(1));
     }
  });
```

FreeIdea-Tech

SQL Lite Database

EX05_05.java 範例

```
myListView
        .setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()       {
         @Override
         public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2,          long arg3)
         {
           /* getSelectedItem所取得的是SQLiteCursor */
           SQLiteCursor sc = (SQLiteCursor) arg0.getSelectedItem();
           _id = sc.getInt(0);
           myEditText.setText(sc.getString(1));
         }
         @Override
        public void onNothingSelected(AdapterView<?> arg0)
         {
         }
       });
  }
```

FreeIdea-Tech

SQL Lite Database

ToDoDB.java 範例

```
private void addTodo()
 {
   if (myEditText.getText().toString().equals(""))
     return;
   /* 新增資料到資料庫 */
   myToDoDB.insert(myEditText.getText().toString());
   /* 重新查詢 */
   myCursor.requery();
   /* 重新整理myListView */
   myListView.invalidateViews();
   myEditText.setText("");    _id = 0;
 }
 private void editTodo()
 {
   if (myEditText.getText().toString().equals(""))
     return;
   /* 修改資料 */
   myToDoDB.update(_id, myEditText.getText().toString());
   myCursor.requery();
   myListView.invalidateViews();
   myEditText.setText("");    _id = 0;
 }
private void deleteTodo()
 {
   if (_id == 0)      return;
   /* 刪除資料 */
   myToDoDB.delete(_id);
  myCursor.requery();
  myListView.invalidateViews();
   myEditText.setText("");
   _id = 0;
 }}
```

FreeIdea-Tech

SQL Lite Database

ToDoDB. java 範例

```
package irdc.ex05_05;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class ToDoDB extends SQLiteOpenHelper  {
  private final static String DATABASE_NAME = "todo_db";
  private final static int DATABASE_VERSION = 1;
  private final static String TABLE_NAME = "todo_table";
  public final static String FIELD_id = "_id";
  public final static String FIELD_TEXT = "todo_text";

  public ToDoDB(Context context)  {
   super(context, DATABASE_NAME, null, DATABASE_VERSION);
  }
  @Override
  public void onCreate(SQLiteDatabase db)  {
   /* 建立table */
   String sql = "CREATE TABLE " + TABLE_NAME + " (" + FIELD_id
       + " INTEGER primary key autoincrement, " + " " + FIELD_TEXT + " text)";
   db.execSQL(sql);
  }
  @Override
  public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)  {
   String sql = "DROP TABLE IF EXISTS " + TABLE_NAME;
   db.execSQL(sql);
   onCreate(db);
  }
```

FreeIdea-Tech

SQL Lite Database

ToDoDB. java 範例

```
public Cursor select() {
  SQLiteDatabase db = this.getReadableDatabase();
  Cursor cursor = db.query(TABLE_NAME, null, null, null, null, null, null);
  return cursor;
 }
 public long insert(String text)  {
  SQLiteDatabase db = this.getWritableDatabase();
  /* 將新增的值放入ContentValues */
  ContentValues cv = new ContentValues();
  cv.put(FIELD_TEXT, text);
  long row = db.insert(TABLE_NAME, null, cv);
  return row;
 }
 public void delete(int id) {
  SQLiteDatabase db = this.getWritableDatabase();
  String where = FIELD_id + " = ?";
  String[] whereValue =
  { Integer.toString(id) };
  db.delete(TABLE_NAME, where, whereValue);
 }
 public void update(int id, String text) {
  SQLiteDatabase db = this.getWritableDatabase();
  String where = FIELD_id + " = ?";
  String[] whereValue =
  { Integer.toString(id) };
  /* 將修改的值放入ContentValues */
  ContentValues cv = new ContentValues();
  cv.put(FIELD_TEXT, text);
  db.update(TABLE_NAME, cv, where, whereValue);
 } }
```

Freeidea-Tech

Activities

- 一般所指的活動(Activity)是使用者介面。
  一支應用程式可能有一個或以上的活動存在，每個活動也都會有自己的View。

- 所有的活動在系統裡由活動堆疊所管理，當一個新的活動被執行後，它將會被放置到堆疊的最頂端，並且變成"running activity"，而先前的活動原則上還是會存在於堆疊中，但它此時不會是在前景的情況，除非新加入的活動離開。

Freeda-Tech

Activities

活動元件說明

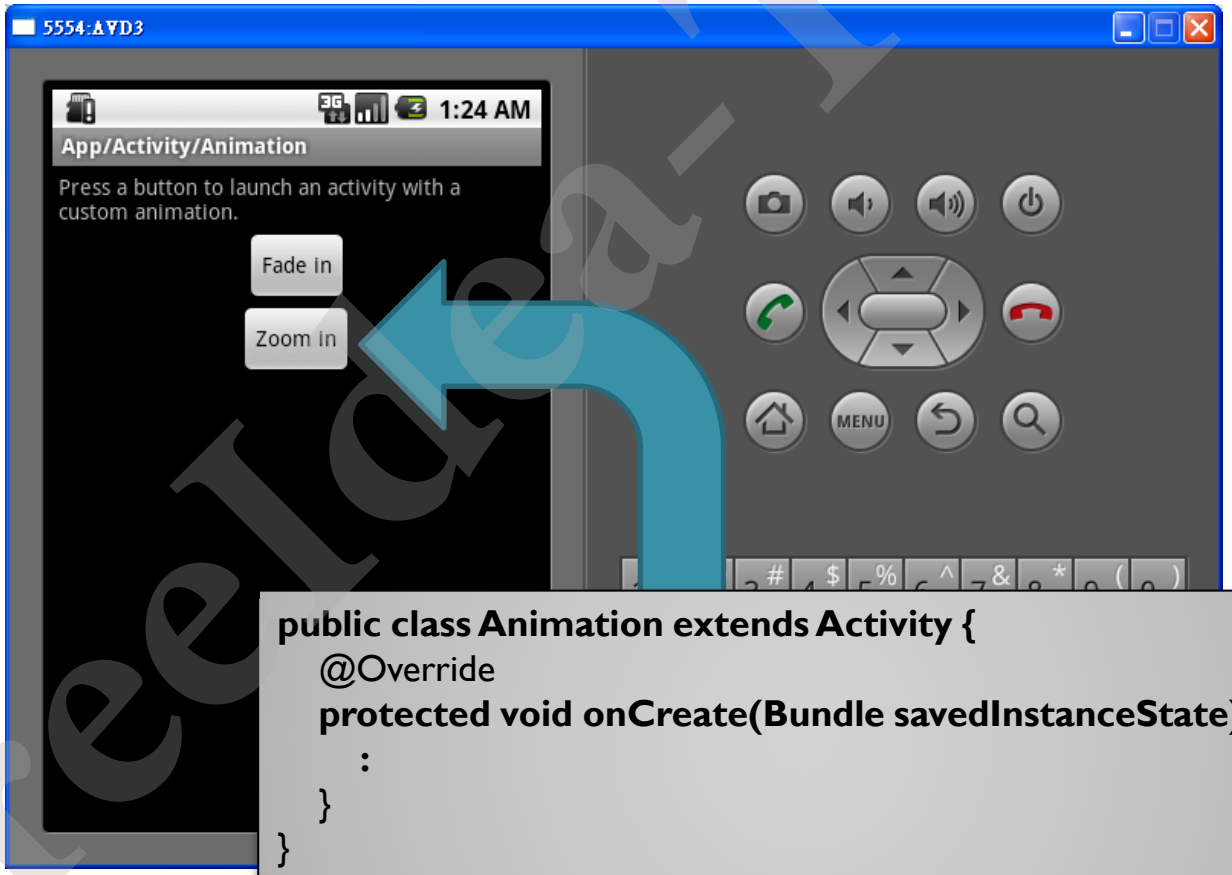
- 活動元件是回應用戶操作而展示的視覺化使用者介面。
- 應用程式可以只有一個Activity，或像簡訊應用程式那樣包含多個。
- 每個Activity都被給予一個預設的視窗以進行繪製。

- extends
  - ContextThemeWrapper
- implements
  - ComponentCallbacks
  - KeyEvent.Callback
  - LayoutInflater.Factory
  - View.OnCreateContextMenuListener
  - Window.Callback

FreeIdea-Tech

Activities

操控介面程式就是活動元件




```
public class Animation extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        :
    }
}
```

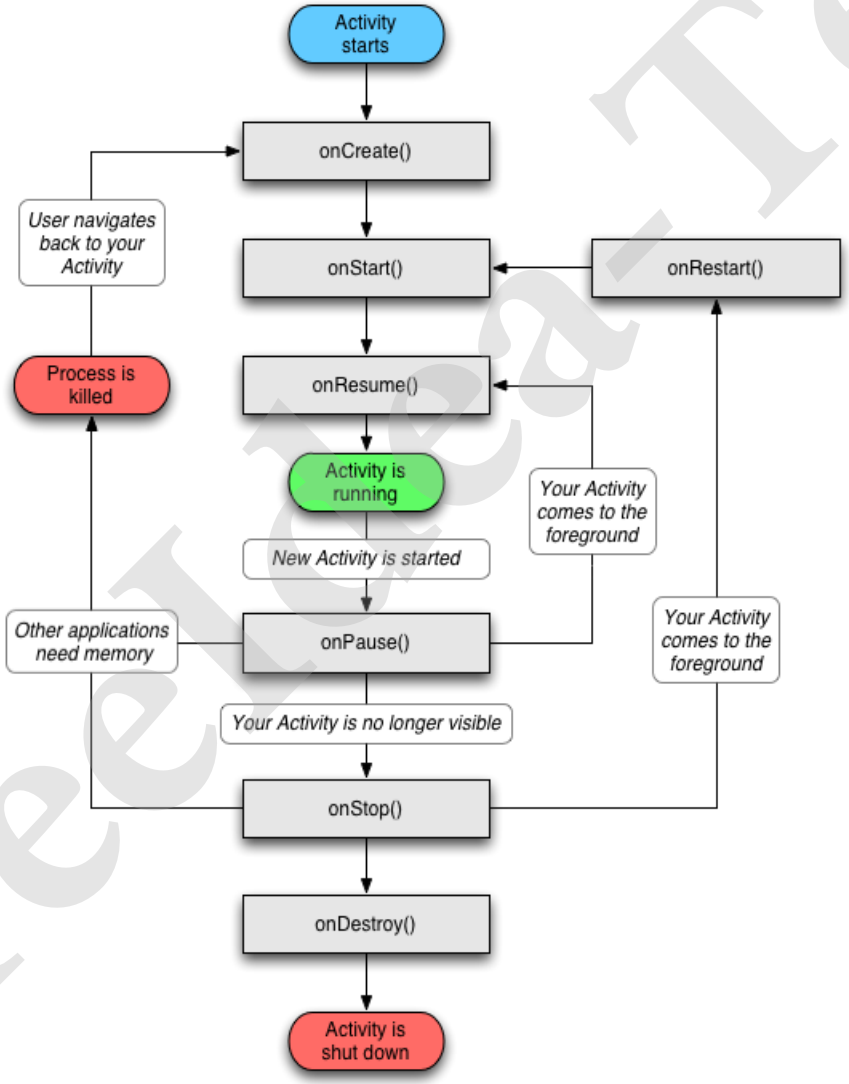
FreeIdea-Tech

Activities

- 活動狀態間的切換包含了呼叫下列幾種方法：
  - void onCreate(Bundle *savedInstanceState*)
  - void onStart()
  - void onRestart()
  - void onResume()
  - void onPause()
  - void onStop()
  - void onDestroy()

FreeIdea-Tech

Activities



活動元件生命週期 (Activity Lifecycle)

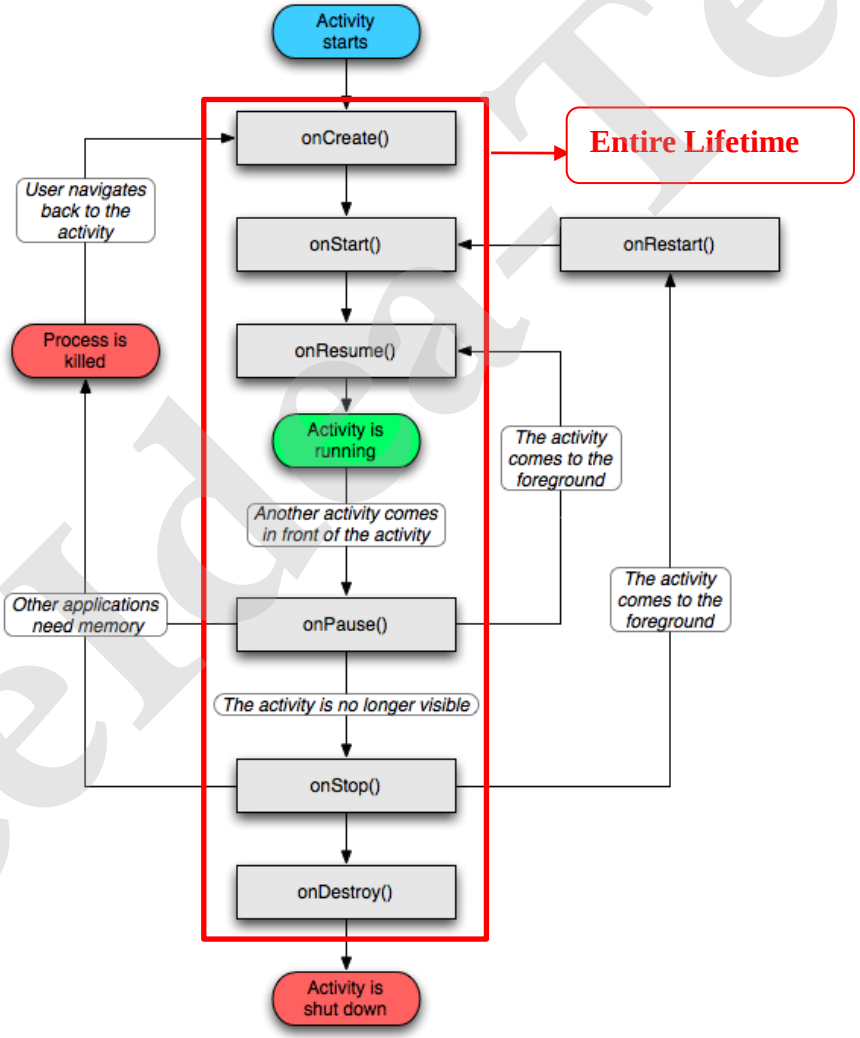
FreeLea Tech

Activities

- **活動的Entire Lifetime**
  - 一個活動的Entire Lifetime是由onCreate(Bundle)開始，直到onDestroy()結束。
  - 一個活動可以把所有的資源設定寫在onCreate中，直到onDestroy()時，再釋放出來。

FreeIdea Tech

Activities



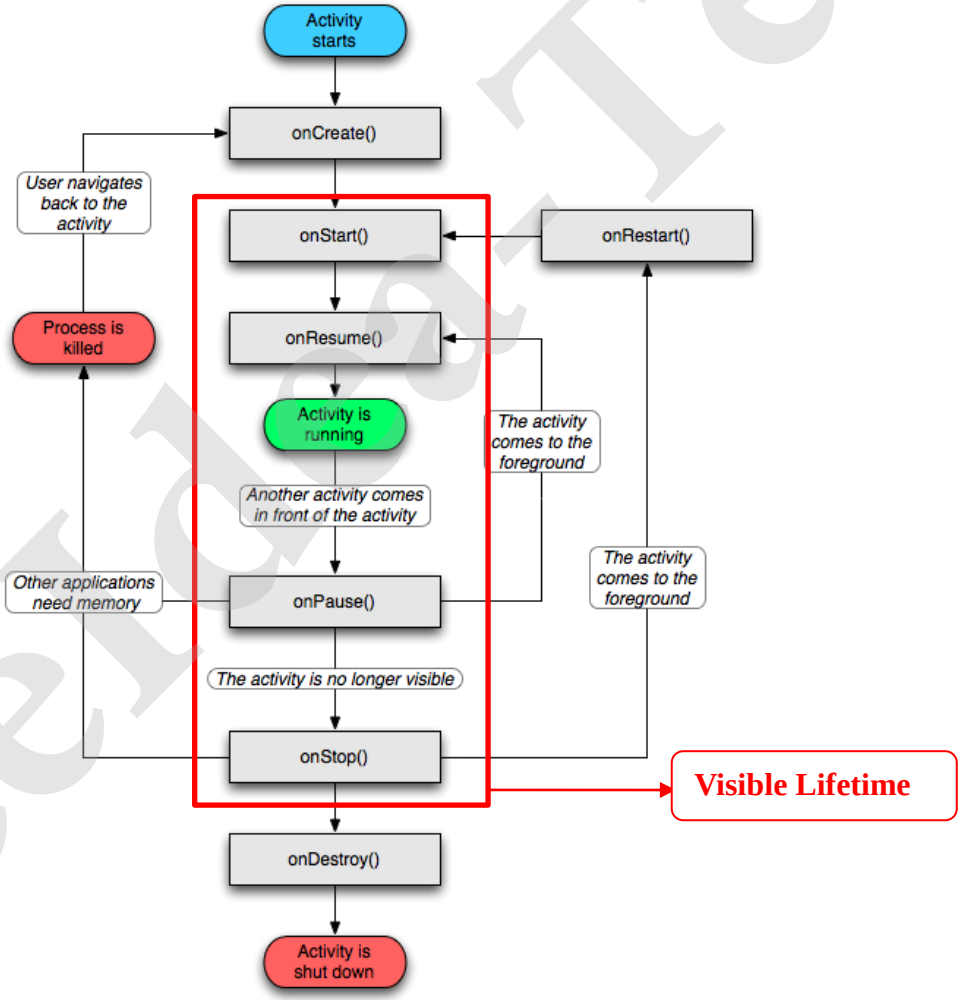
FreeDataTech

Activities

- **活動的Visible Lifetime：**
  - 一個活動的Visible Lifetime則是指在onStart() → onStop()之間，稱為“可視生命週期”，在這段時間內，使用者可以在螢幕上看見Activity，但這個Activity不見得一定在前景跟使用者直接互動。

FreeIdea Tech

程式生命週期



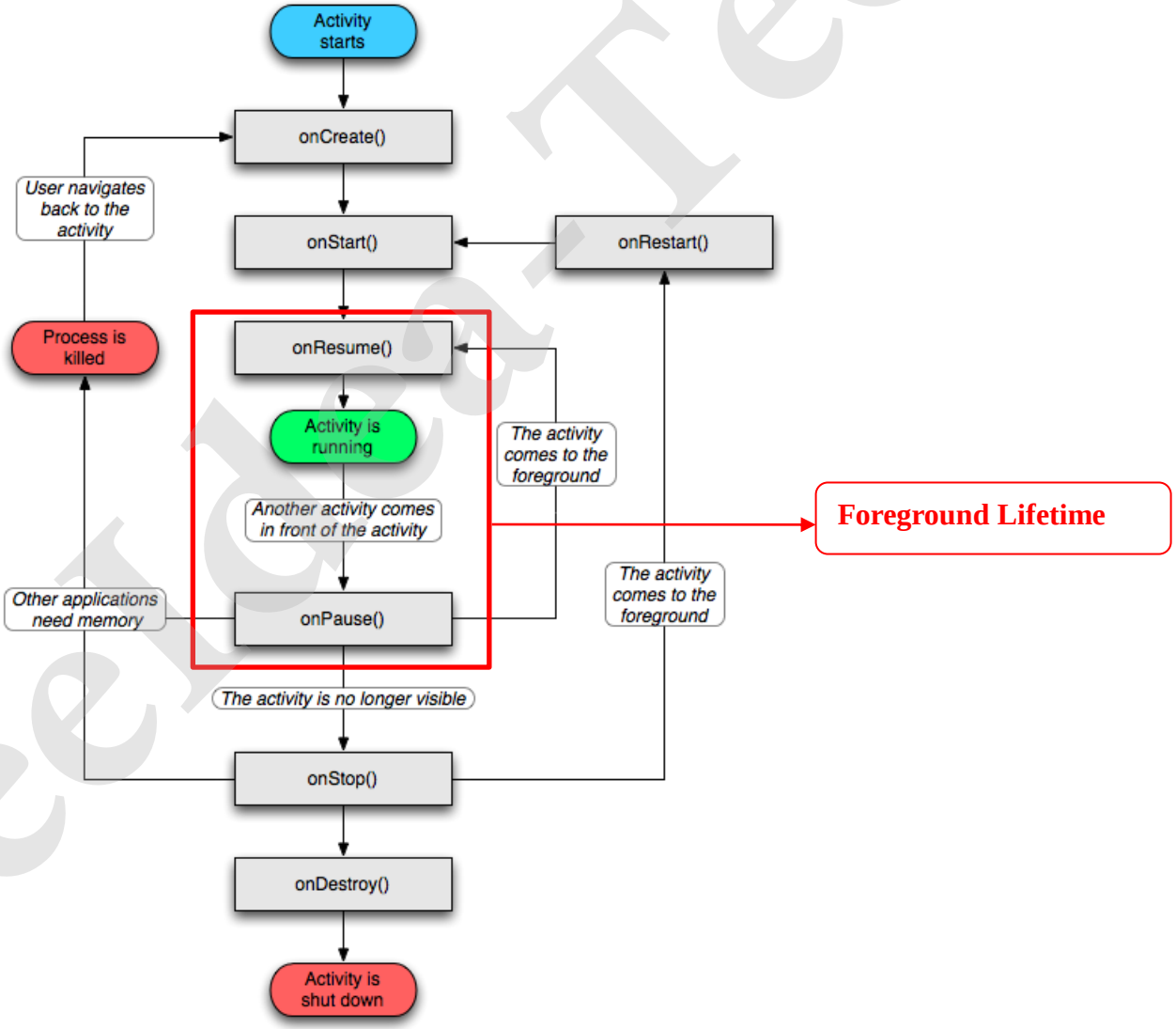
FreeEdgeTech

程式生命週期

- **活動的Foreground Lifetime：**
  - Foreground Lifetime則是指onResume() → onPause()之間，在這個時期的活動是在所有的活動的前面，並且直接跟使用者進行互動。
  - 一個活動能很頻繁的在Resume及Pause這兩個狀態切換，所以在onResume()及onPause()中實作的程式應盡量精簡。

FreeDaeTech

程式生命週期



FreeIdea-Tech

Services

- 在每一個包含服務元件的AndroidManifest.xml文件中，必須有一個相應的<service>宣告。

- 服務可以用兩種方式呼叫：
  - 服務自行啟動和運行，直到某項操作停止或自行停止時。
  - 它可被提供給對外使用者的介面所操作。用戶端與服務物件建立連接並呼叫服務。

Freelea-Tech

Services

- 上述兩種模式不是完全分開的，且可以綁定一個被startService()啟動的服務。
- 這種情況下，stopService()不會停止服務，一直到最後一個綁定的連接關閉時。
- 與活動相似，服務也有一些生命週期方法，程式設計師可以實作它們去改變狀態，但方法比活動要少。

Free Idea-Tech

Services

- 服務是在背景長時間運行的應用元件，不和使用者直接進行互動。
- 例如：某服務可能在後台播放音樂，而用於在執行其他的操作，或者它透過網絡抓取資料或者執行某些計算，將結果提供給活動。

FreeIdea-Tech

Services

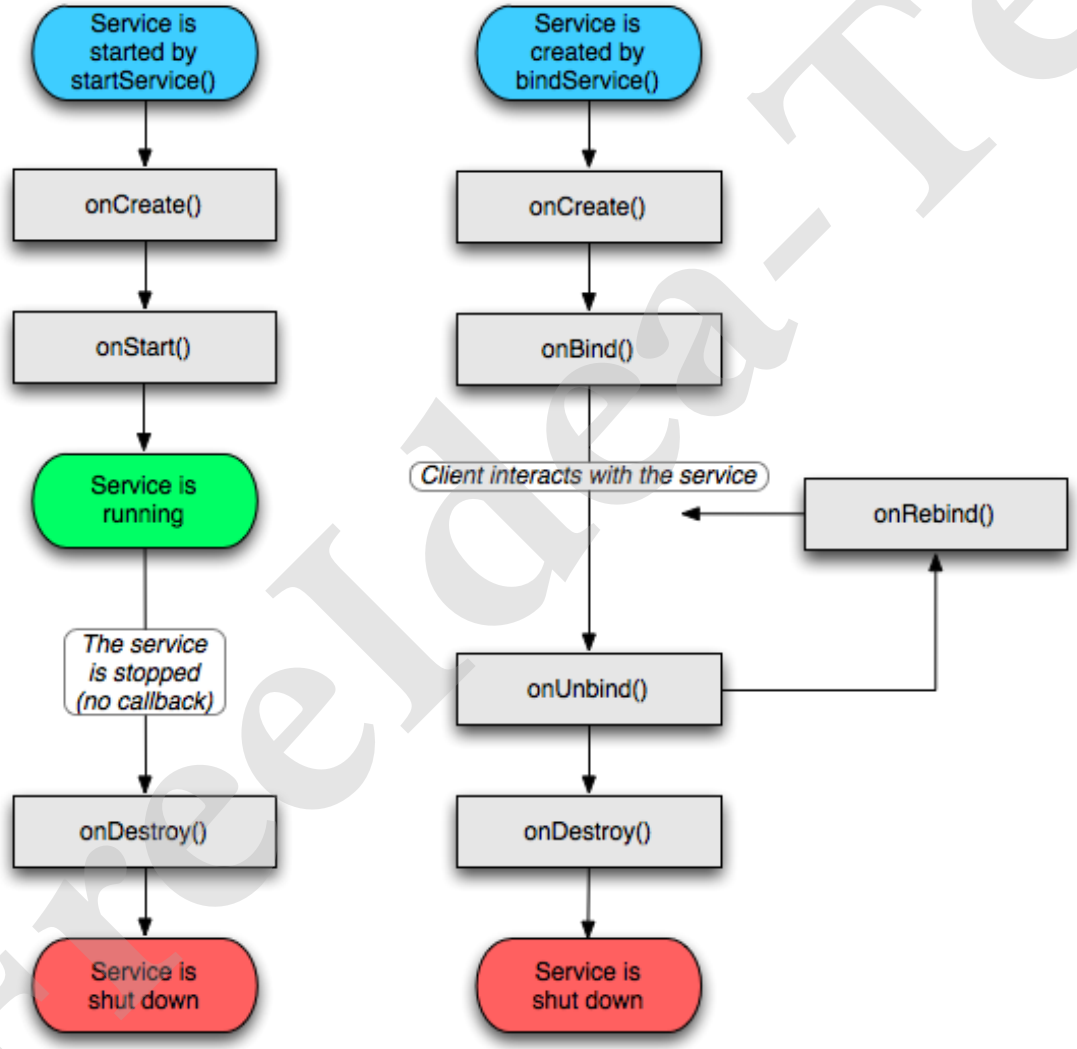
服務元件說明

- 無視覺化使用者介面
- 在後台執行
- 執行於應用程式行程的主執行緒內。

- extends
  - ContextWrapper
- implements
  - ComponentCallbacks

FreeIdea-Tech

Services



服務元件生命週期 (Service Lifecycle)

FreeIdea-Tech

Services

- **服務的Entire Lifetime**
  - 服務的Entire Lifetime是在onCreate()開始，於onDestroy()結束。
  - 類似活動，服務在onCreate()方法中進行初始化，在onDestroy()方法中釋放所有系統資源。
  - 例如，音樂播放服務可能在onCreate()方法中建立音樂播放執行緒，在onDestroy()方法中停止執行緒。

Free Idea-Tech

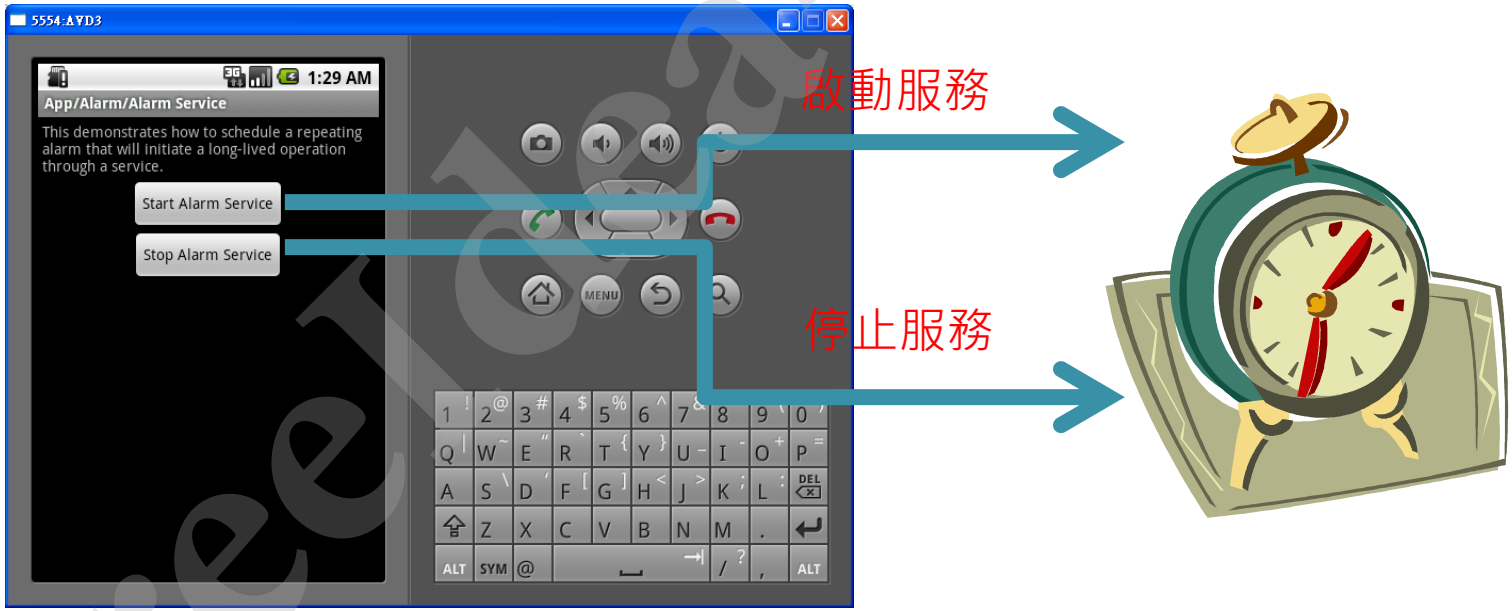
Services

- **服務的Active Lifetime**
  - 在呼叫startService()方法後，此方法被Intent對象指示去呼叫onStart() 方法。
  - 音樂服務將分析這個Intent來確定播放什麼音樂，並開始播放。

FreeIdea-Tech

Services

利用活動元件來操控服務元件



FreeIdea-Tech

Services

活動及服務繼承關係

FreeIdea Tech

Content Providers

- 內容管理器將應用程式資料組合成特定的集合供其它應用程式使用。資料可以是儲存在檔、SQLite資料庫，或是其它任何使用者可以存取資料的地方。
- 內容管理器繼承於內容管理器基礎類別，並實作一組標準的方法，使應用程式可以檢索和儲存它控制的資料。
- 應用程式不是直接呼叫這些實作方法。而是透過內容解析器(ContentResolver)對象呼叫方法。內容解析器能夠通知任何的內容管理器，並可以參與這些內容管理器行程間的管理。

Fridea-Tech

Content Providers

內容管理器說明

- 內容管理器將一些特定的應用程式資料提供給其它應用程式使用。資料可以存儲於檔案系統、SQLite資料庫或其它方式。

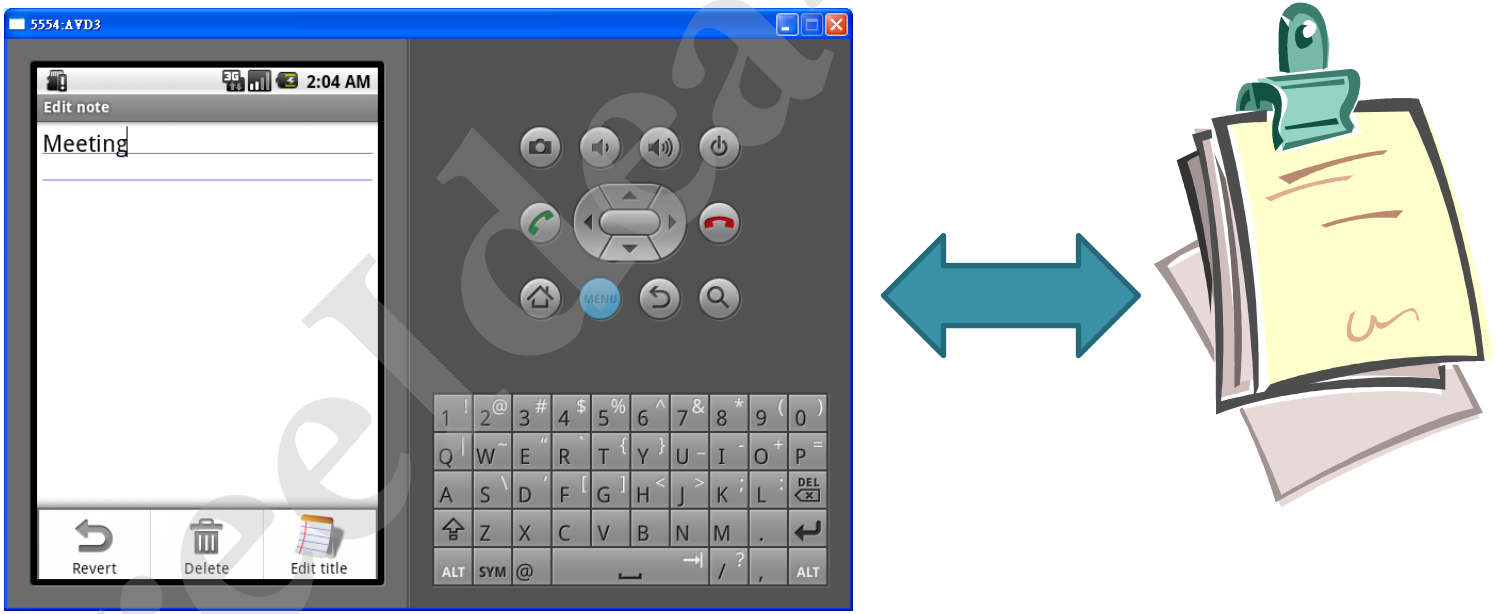
- public abstract class ContentProvider

- extends Object implements ComponentCallbacks

FreeIdea-Tech

Content Providers

使用內容提供者來操控資料庫



FreeIdea-Tech

Broadcast Receivers

- 廣播接收器只有一個生命週期方法:
  - void onReceive(Context curContext, Intent broadcastMsg)
- 當廣播消息到達接收者時，Android呼叫onReceive()方法，並傳遞保存著訊息的Intent對象。
- 當呼叫該方法時，廣播接收者被認為是active的。當onReceive()結束時，它就是inactive的。
- 擁有活動的廣播接收者行程，會被保護不被清除。但是只擁有inactive的廣播元件時，當系統認為記憶體應該被其他行程使用時，可以在任何時間被系統清除。

FreeLearTech

Broadcast Receivers

- 當與廣播元件頻繁互動時，會產生一些問題，因此，某些任務在獨立的執行緒中操作，與管理其他的用戶元件的主執行緒是分開的。
- 如果onReceive()產生執行緒然後返回，這個完整的行程，包括新產生的執行緒，將被判斷為inactive的，會被提報為將被清除的行程。
- 解決這個問題的辦法是為onReceive()啟動一個Service，讓Service接手這工作，因此系統會認為在行程中仍然有active的工作在進行。

Freelance-Tech

Broadcast Receivers

- 廣播接收器負責接受和回應通知，很多通知源自於系統所發送的，例如：發送時區變換的通知，電池電量不足，或使用者改變語言設置。

- 應用程式也可以發出廣播通知，舉例來說，通知其它應用程式，資料已下載完畢，可供使用。

- 應用程式可以擁有任意數量的廣播接收器來接收任何的通知。另外也可以啟動活動去回應接收到的通知，或利用通知管理器(NotificationManager)來通知使用者。

Freida-Tech

Broadcast Receivers

廣播接收器說明

- 專注於接收廣播通知資訊。
- 並做出對應處理的元件。
- 無使用者介面。
- public abstract class BroadcastReceiver
- extends Object

FreeIdea-Tech

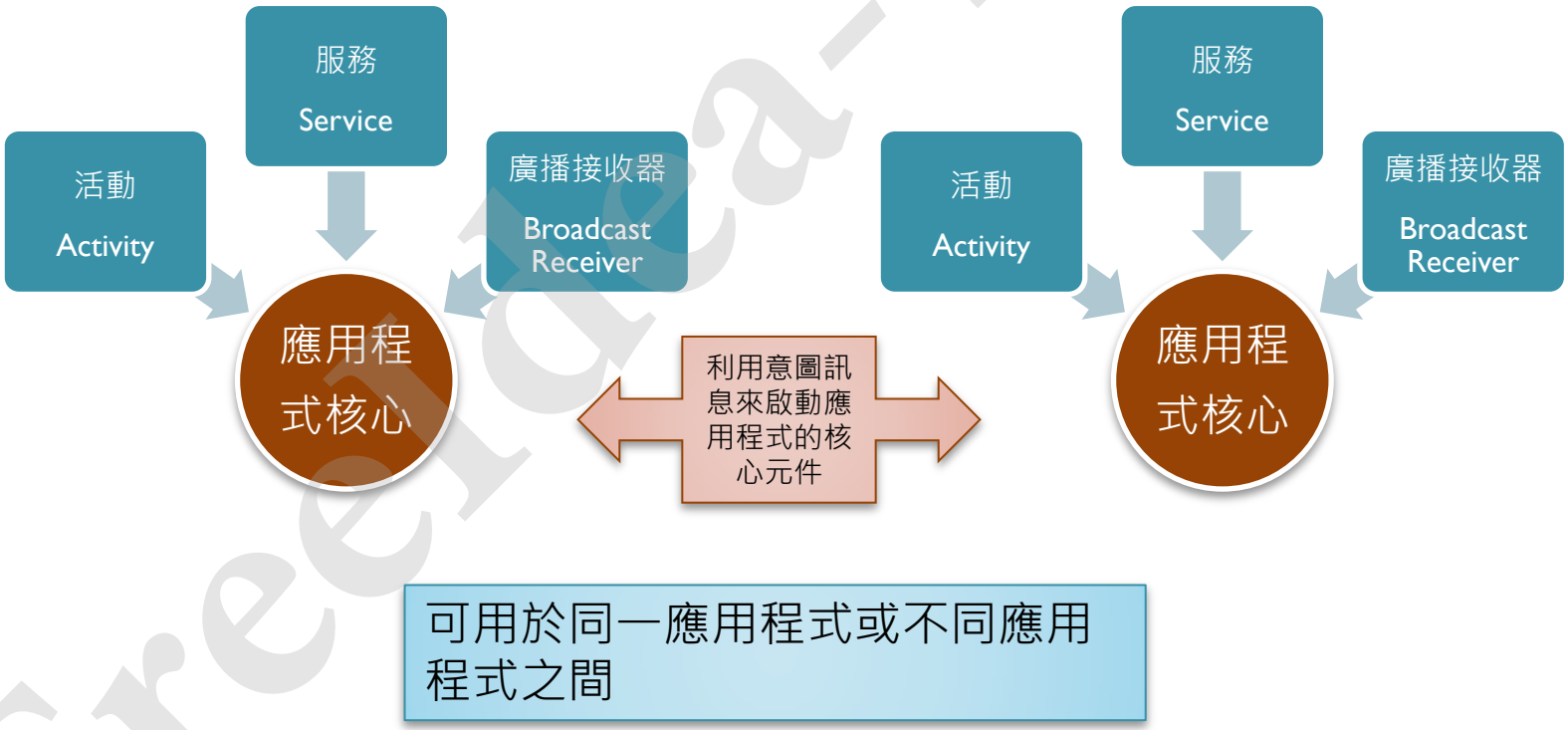
Activating components: Intents

- Intent是一次即將操作的抽象描述。

- Intent作用是：
  (1) 啟動一個新Activity並且可以攜帶資料。
  (2) 透過Intent來啟動一個服務(Service)。
  (3) 透過Intent來廣播一個事件。

FreeIdea-Tech

Activating components: Intents



可用於同一應用程式或不同應用程式之間

Freelea-Tech

Activating components: Intents

三個傳遞意圖訊息的機制

活動

- Context.startActivity() – 啟動活動
- Activity.startActivityForResult() – 啟動活動含結果
- Activity.setResult() – 設定結果

服務

- Context.startService() – 啟動服務
- Context.bindService() – 建立應用程式和服務間的相依關係。

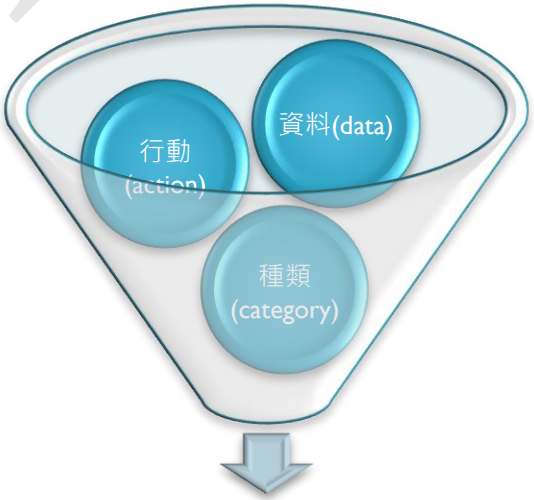
廣播接收器

- Context.sendBroadcast() -
- Context.sendOrderedBroadcast()
- Context.sendStickyBroadcast()

ReIdea-Tech

Activating components: Intents

- 意圖依指定或不指定目的名稱可分成兩類
  - 明顯意圖
  - 隱含意圖
- 意圖濾器可用來宣告元件的能力並限定意圖訊息的使用
- 當元件沒有意圖濾器時只能用明顯意圖



意圖濾器(Intent Filter)

FreeIdea-Tech

Activating components: Intents

```
<intent-filter . . . >
<action android:name="com.example.project.SHOW_CURRENT" />
<action android:name="com.example.project.SHOW_RECENT" />
<action android:name="com.example.project.SHOW_PENDING" />
. . .
</intent-filter>
```

Action test

```
<intent-filter . . . >
<category android:name="android.intent.category.DEFAULT" />
<category android:name="android.intent.category.BROWSABLE" />
. . .
</intent-filter>
```

Category test

```
<intent-filter . . . >
<data android:mimeType="video/mpeg" android:scheme="http" . . . />
<data android:mimeType="audio/mpeg" android:scheme="http" . . . />
. . .
</intent-filter>
```

Data test

Free Idea-Tech

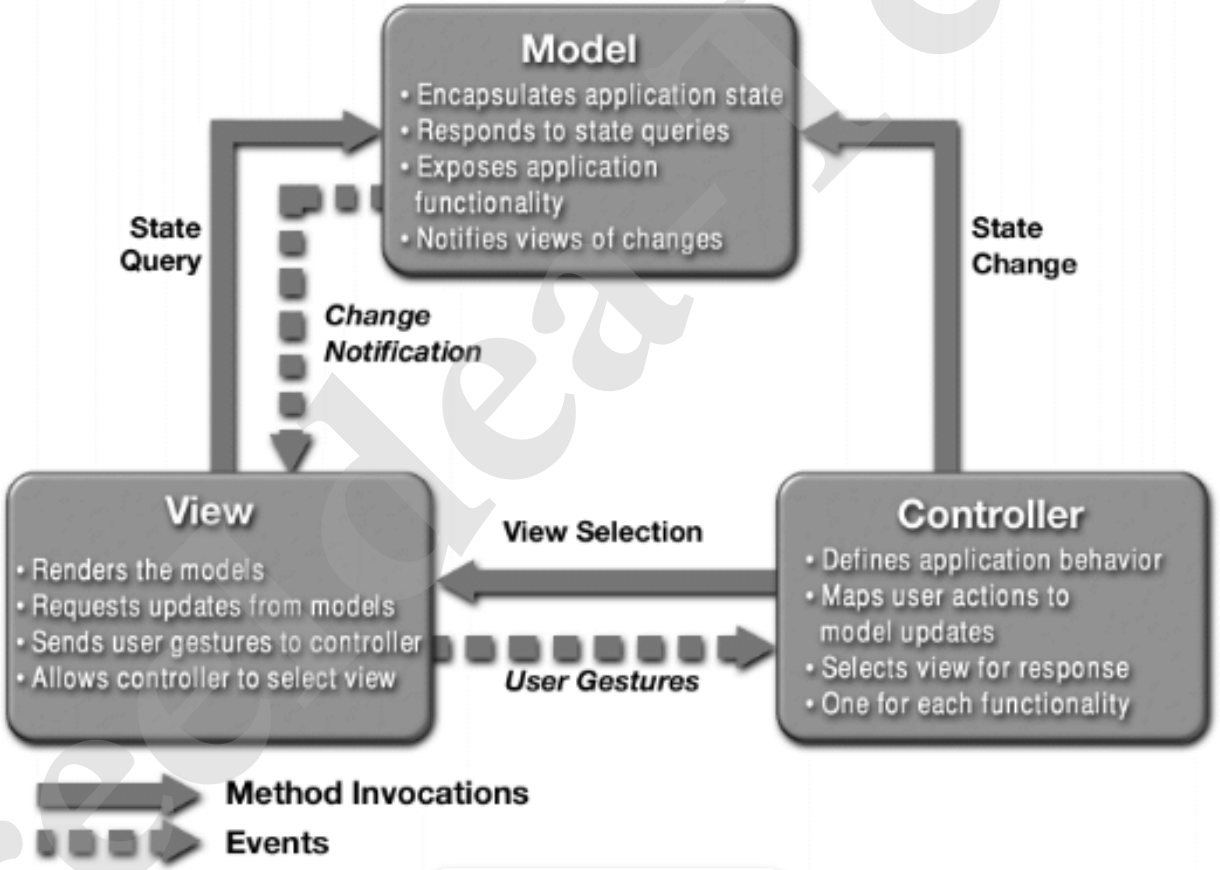
補充資料(MVC)

- MVC(Model-View-Controller)
  用於表示一種檢視軟體模型架構模式

- MVC把軟體分為三個基本部分：
  - 模型(Model)
  - 檢視(View)
  - 控制器(Controller)

FreeIdea-Tech

補充資料(MVC)



MVC架構

Free Learn Tech

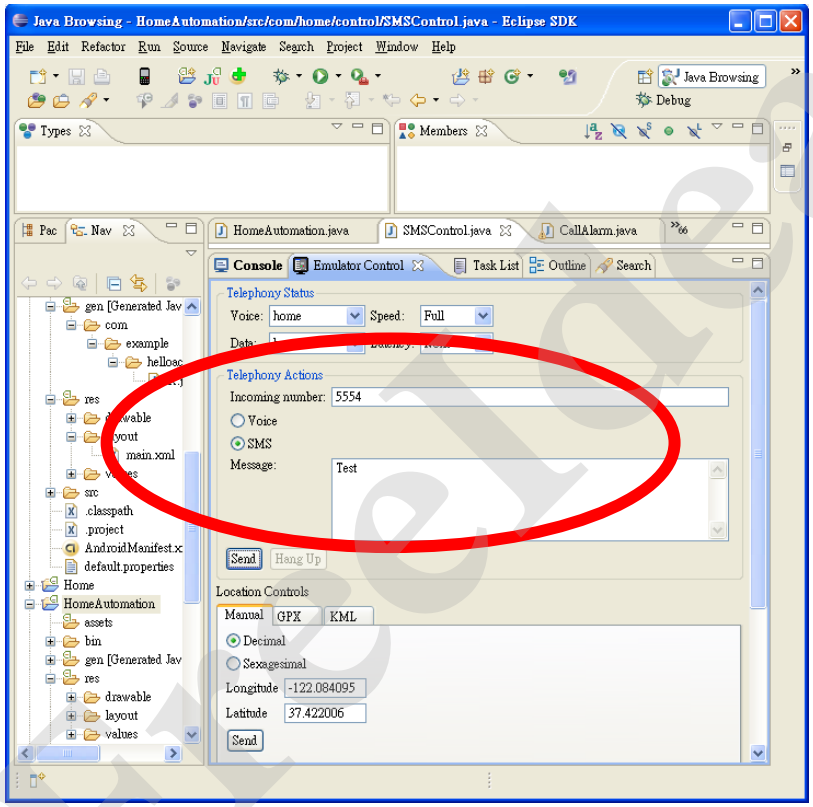
補充資料(MVC)

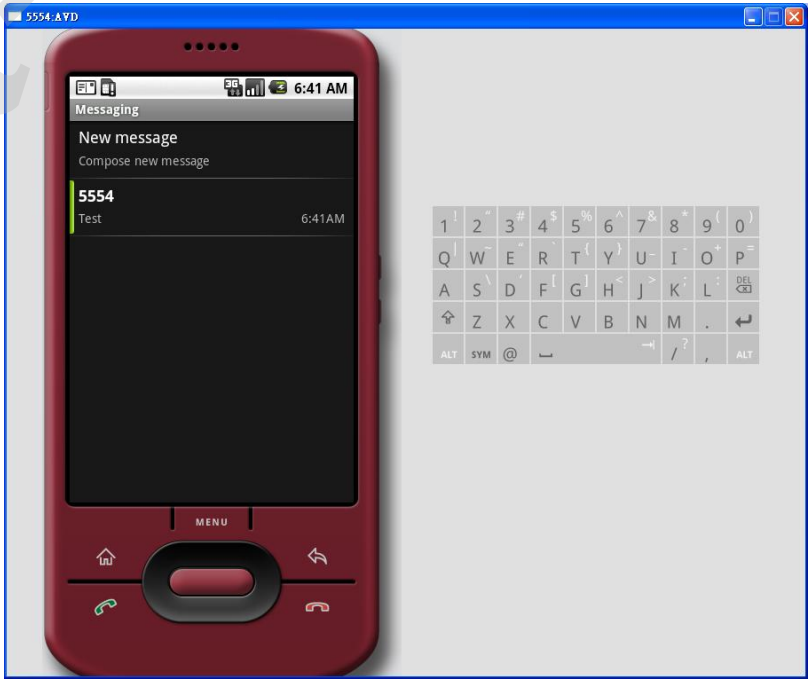
- MVC架構起始於一個GUI(graphical user interface design patter)原型。
- 其目的是實作動態程式設計，使日後對於程式修改及擴展更加便利，並使某些程式碼可重複利用。
- 另外透過對複雜度的簡化，使程式結構更加直覺。

FreeIdea-Tech

補充資料

- 利用模擬器控制視窗來進行簡訊測試

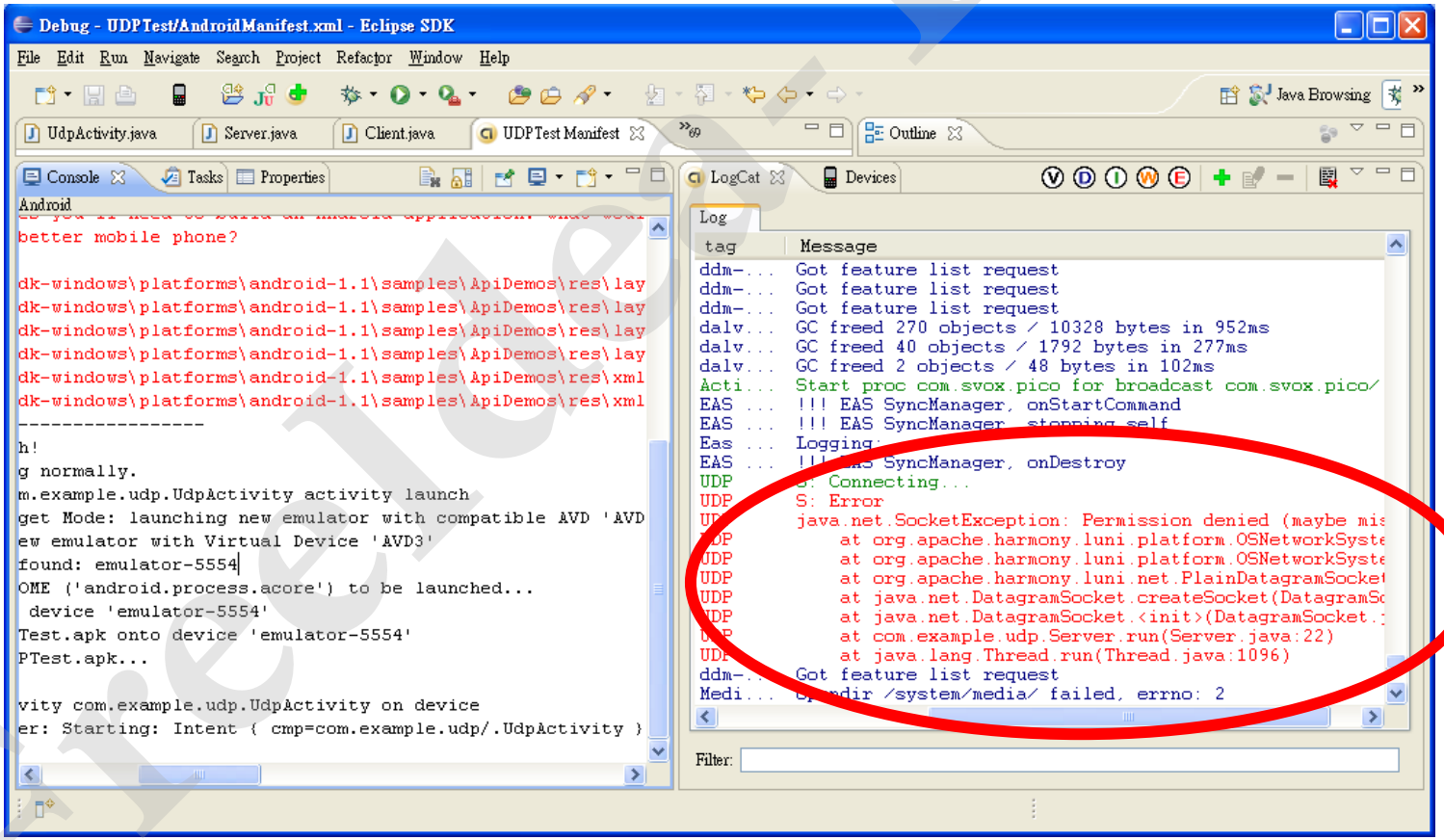




Free Tech

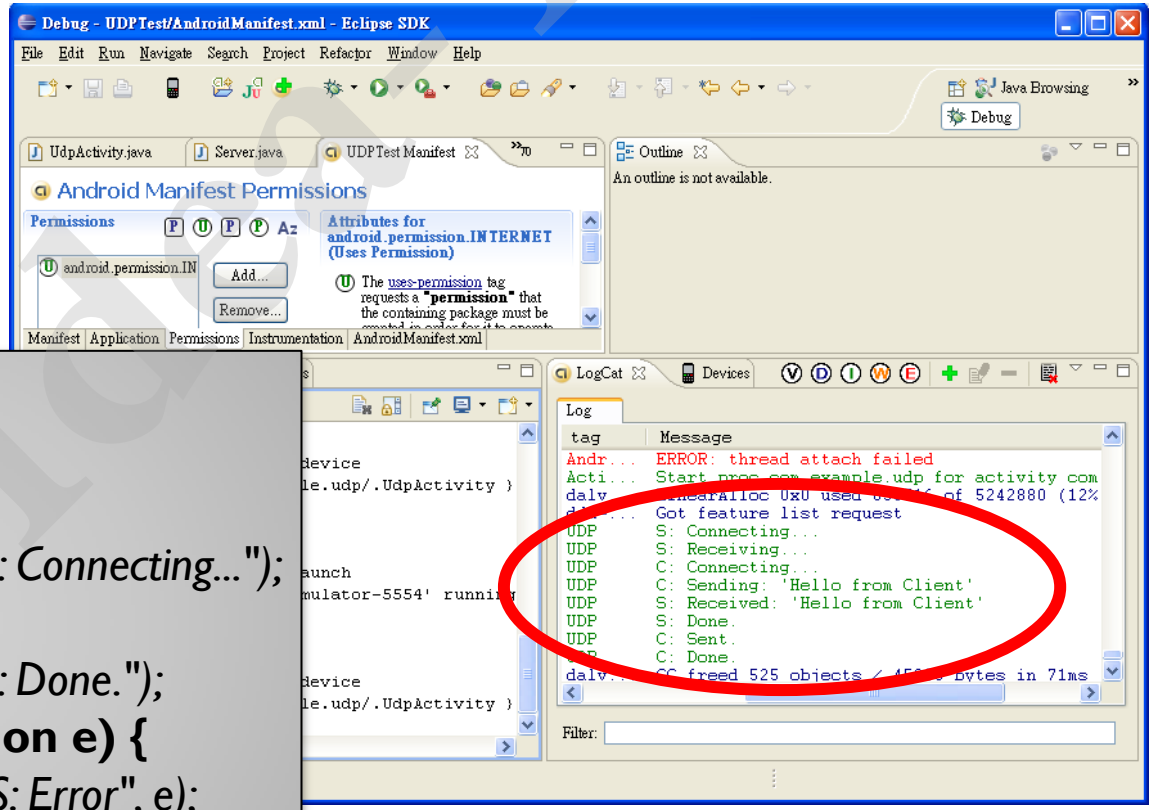
補充資料

- 利用LogCat來輔助除錯



補充資料

- 利用Log物件來協助除錯

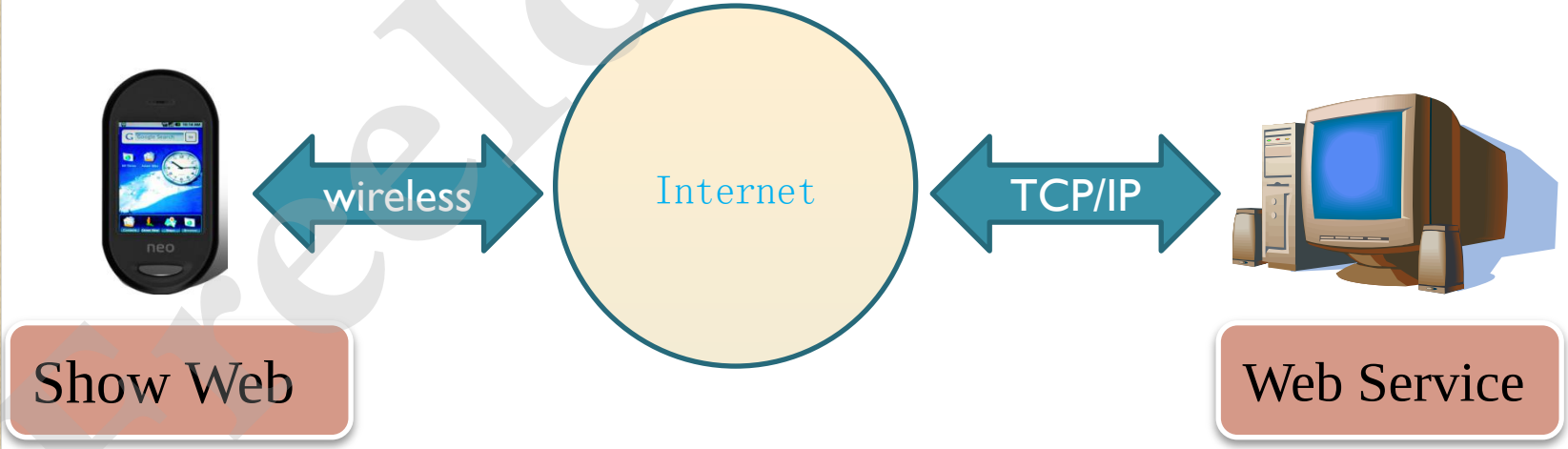



```
public void run() {
        try {
            :
            Log.i("UDP", "S: Connecting...");
            :
            Log.i("UDP", "S: Done.");
        } catch (Exception e) {
            Log.e("UDP", "S: Error", e);
        }
```

FreeLife Tech

專題製作(一)

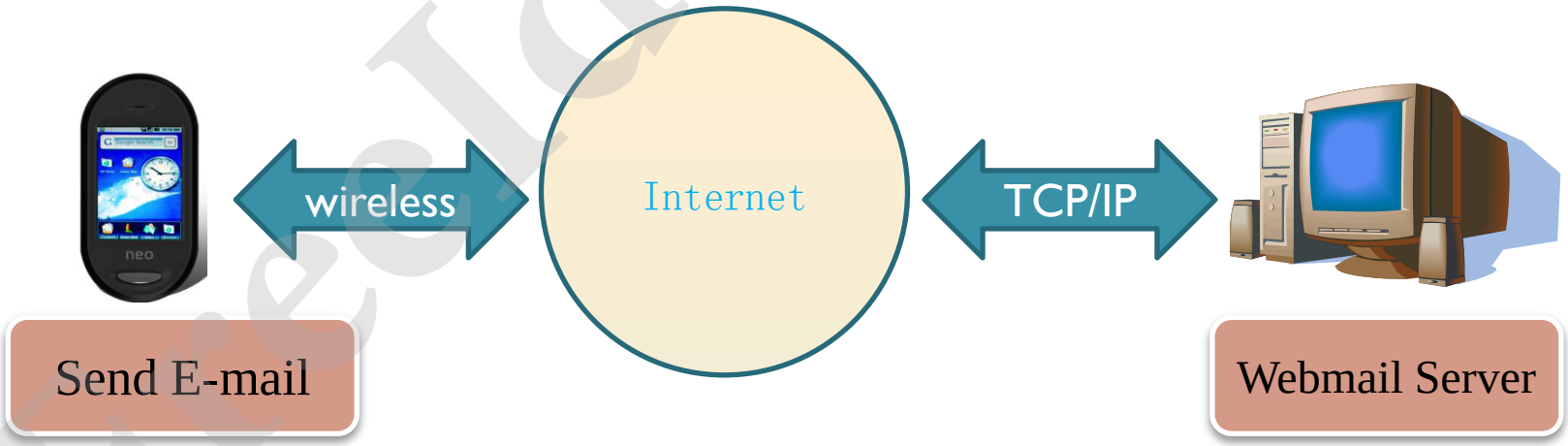
- 每組1~3人
- 目的：利用Android 設計出，整合多種選擇表單可瀏覽Http網頁內容的APP程式(如Facebook、Yahoo新聞、GoogleMap等)
- 需求：底圖、表單元件、對話視窗



FreeIdeaTech

專題製作(二)

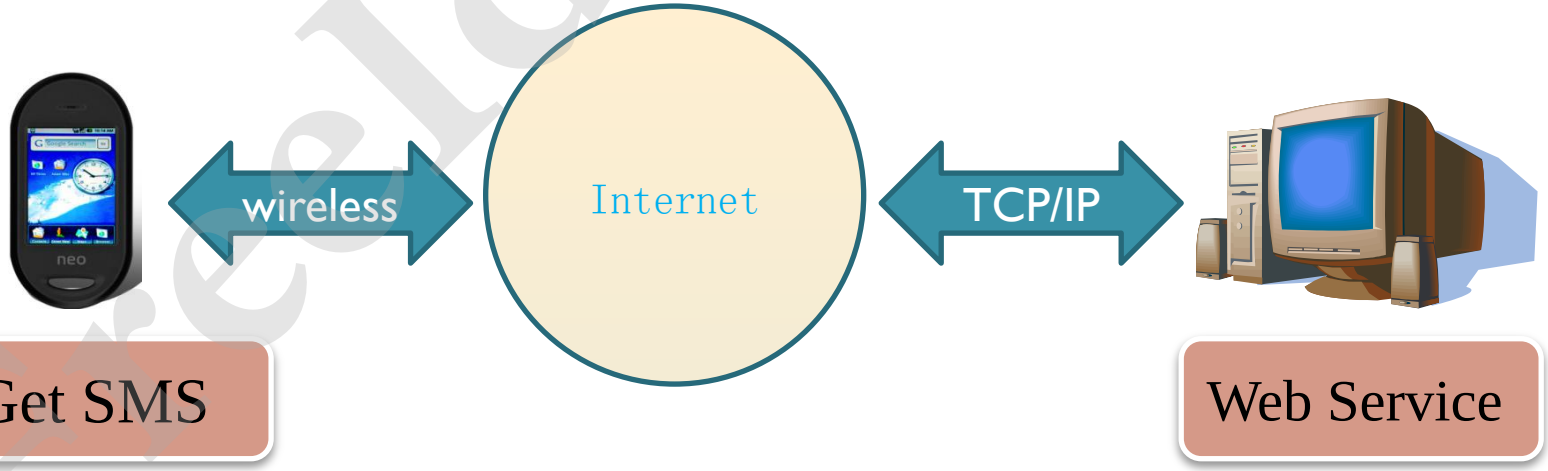
- 每組1~3人
- 目的：利用Android設計出，由手機透過Intent發送E-mail至WebMail的APP程式
- 需求：文字元件、表單元件、對話視窗



FreeIdeaTech

專題製作(三)

- 每組1~3人
- 目的：利用Android設計出，即時衛星雲圖的APP程式
- 需求：底圖、表單元件、對話視窗



FreeIdea-Tech