學習Android之路

這是為了讓許多有心想學習如何開發Android APP的朋友們設計的內容，從頭以正確的方式學習，這個過程並不是無痛的，只是，有付出，一定會有收獲的。請切記，你練習過的內容若在日後無法自己完成它，就不算程式設計能力。程式設計是需要不斷重複練習與應用後，所獲得的經驗所組成的。

系列文章使用Android 7.1並以Android Studio 2.3版為基礎

0. 開始之前

使用Github分享在學習、開發過程的程式碼是一個潮流，未來的工作面試時，你可能只需要一個如github的網址，對方就能知道你的程式設計能力，也能看出你在學習過程中參與有興趣的專案歷程。我在撰寫本書時，使用github來管理各個章節的程式碼，因此，使用它來說明一個專案的開發過程是再也合適不過了。

0.1  [什麼是git、github? 為什麼要版本控制，從安裝必要的軟體開始](https://litotom.com/2017/03/03/android-book-chapter0-1/)

0.2  [取得範例專案程式碼，註冊github後取得專案複本](https://litotom.com/2017/03/03/android-book-chapter0-2/)

0.3  [切換各個版本，與遠端檔案庫切斷連結](https://litotom.com/2017/03/03/android-book-chapter0-3/)

1. 由最基本的環境、工具、架構、開發工具，從一開始就做對的事吧

1.1 [學Android的起點，把環境建置好並掌握每個角色後，什麼都好做](https://litotom.com/2017/02/22/android-effective-02-env/)

1.2 [Android Studio開發環境準備與安裝必要工具](https://litotom.com/2017/03/01/android-studio-start/)

1.3 [建立新Android專案，認識Android Studio環境](https://litotom.com/2017/03/09/android-effective-02-env-3/)

1.4 [Android SDK，SDK管理工具的使用](https://litotom.com/2017/03/09/android-effective-02-env-4/)

1.5 [使用內建模擬器映像檔，建立Android模擬器](https://litotom.com/2017/03/09/android-effective-02-env-5/)

1.6 Genymotion模擬器

2. 開始寫第一個APP囉，建立一個專案後，先認識專案架構與開發工具

2.1 [Android 專案架構，什麼是AndroidManifest.xml?](https://litotom.com/2017/03/12/android-effective-03-1st-app-01/)

2.2 [畫面配置Layout，使用者看到的畫面設計](https://litotom.com/2017/03/12/android-effective-03-1st-app-1/)

2.3 [Activity是什麼? Java類別，沒有它就沒有互動了](https://litotom.com/2017/03/12/android-effective-03-1st-app-2/)

2.4 [專案的資源，Layout、字串與圖檔等，但沒有R類別就麻煩了](https://litotom.com/2017/03/12/android-effective-03-1st-app-3/)

2.5 [Android Support Library—支援函式庫](https://litotom.com/2017/03/12/android-effective-03-1st-app-4/)

3. 如何設計一個畫面? 什麼是Layout與Activity類別

3.1 [Layout版面配置能做什麼? 有那些Layout可以選擇，談LinearLayout與RelativeLayout](https://litotom.com/2017/03/27/book-ch4-1/)

3.2 [元件的屬性，高度、寬度、對齊與權重分配](https://litotom.com/2017/03/28/book-4-2/)

3.3 [Layout新成員ConstraintLayout，組織要扁平化，Layout也要扁平化?](https://litotom.com/2017/04/01/book-ch4-3/)

3.4 [上戰場了，寫個Bmi APP吧，專案Layout與元件設計](https://litotom.com/2017/04/03/book-4-4/)

3.5 [Bmi專案功能與事件處理，什麼是onClick屬性?](https://litotom.com/2017/04/03/book-ch4-5/)

3.6 [Bmi專案功能之顯示資訊，包括Toast、AlertDialog](https://litotom.com/2017/04/03/book-ch4-6/)

3.7 [Bmi專案功能之多國語言，資源抽出之重要性](https://litotom.com/2017/03/24/book-ch4-7/)

3.8 [Bmi專案功能與事件處理，什麼是匿名類別?](https://litotom.com/2017/04/11/book-ch4-8/)

3.9 使用ConstraintLayout設計Bmi專案畫面

4. 建立與使用第二個Activity

4.1 [建立第二個Acitivity，自動產生了那些東西?](https://litotom.com/2017/04/27/ch5-1-2nd-activity)

4.2 [使用Intent轉換Activity並傳遞資料，什麼是Intent意圖?](https://litotom.com/2017/04/27/ch5-2-intent/)

4.3 [將字串抽取出成為資源，重要的能力](https://litotom.com/2017/05/25/ch5-3-extract-string/)

4.4 [Activity的生命週期，它會自動執行那些方法? 方法如何覆寫?](https://litotom.com/2017/05/25/ch5-4-activity-lifecycle/)

4.5 [生命週期的測試情境—使用Bmi專案](https://litotom.com/2017/05/25/ch5-5-lifecycle-bmi/)

4-6 [來證明你會了吧，Atm專案之登入功能實務](https://litotom.com/2017/06/03/ch5-6-atm-login/)

4-7 [轉換畫面後，按返回鍵回到前一個Activity時怎麼取得結果? onActivityResult方法](https://litotom.com/2017/06/04/ch5-7-activity-result/)

5. Material Design，什麼是實感設計? 有那些元件符合

5.1 [Material Design實感設計，浮動按鈕與它的事件處理](https://litotom.com/2017/06/05/ch6-1-material-design/)

5.2 [SnackBar提示訊息元件](https://litotom.com/2017/06/05/ch6-2-snackbar/)

5.3 什麼是CoordinatorLayout?

5-4 太好用的輸入方塊元件，TextInputLayout

5.5 那RecyclerView呢?

6. APP中可以存取資料嗎? 使用ADB工具與設計ActionBar

6.1 [使用SharedPreferences存取設定資料](https://litotom.com/2017/06/27/ch7-1-sharedpreferences/)

6.2 [使用ADB登入模擬器](https://litotom.com/2017/07/05/ch7-2-adb/)

6.3 [實務練習，怎麼讓APP記住我的登入帳號? Atm專案](https://litotom.com/2017/07/11/ch7-3-save-account/)

6.4 [實作Activity上方選單Menu與下拉項目功能](https://litotom.com/2017/07/31/ch7-4-menu/)

Android高效入門，0-1 什麼是git、github? 為什麼要版本控制，從安裝必要的軟體開始

Git版本控制系統

Git是一個分散式版本控制系統，起初是用在管理複雜且龐大的Linux核心原始碼，它可以保存原始碼在特定時期的的狀態，並指定版本與註解文字，也可以讓開發人員將目前的原始碼回復到特定的版本狀態。

Git使用檔案庫(Repository)來記錄檔案的狀態與資料，可在檔案庫中管理、儲存與修改版本資料，並可在檔案庫中建立開發的分支(Branch)，進行實驗功能的開發，未來成功後可以與主幹(trunk)版本合併，成為一個新版本。

GitHub網站

GitHub([http://github.com](http://github.com/))是一個使用Git版本控制，並用來存放軟體原始碼的共享虛擬主機服務，由GitHub公司所提供。使用者可申請免費或付費帳號，建立公開的始碼檔案庫，目前，GitHub已有超過9百萬註冊使用者與二千多萬個檔案庫,成為最大的開發原始碼聚集的部落。

筆者在設計本書的範例時，已將每個範例專案的重要階段的原始碼狀態送交至GitHub網站中，因此，使用git工具即可取得專案在每個時期的狀態，可觀察其差異，並協助讀者找到問題並解決。GitHub上的網址為:

<https://github.com/android66>

GitHub範例專案

本書的範例專案都可以由github網站上取得，Android Studio已支援Git版本管理，唯需要先在本機中安裝Git工具執行程式。

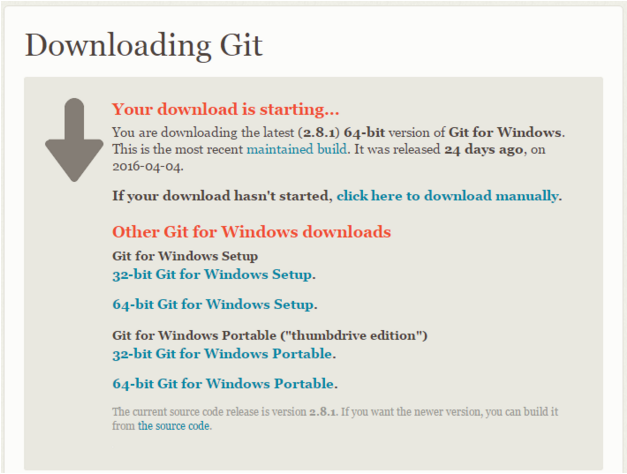
  Git工具軟體安裝

請依以下步驟安裝Git工具軟體：

Step1 以瀏覽器打開「Git for Windows」官方網址：https://git-scm.com/download/，請點擊欲安裝的作業系統類型，如下圖Windows：



Step2 再點擊64位元或32位元的安裝檔下載，如下圖「64-bit Git for Windows Setup」：



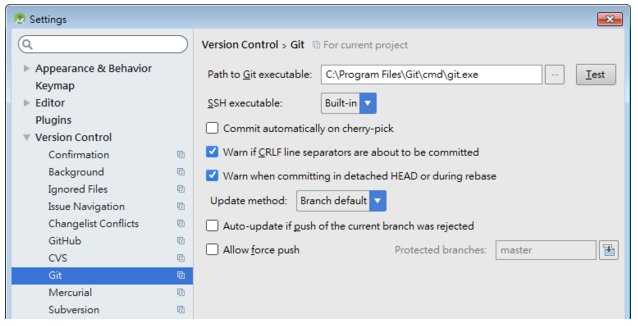
Step3 安裝檔案

請依照預設方式執行檔案安裝即可，64位元版本的安裝資料夾為「C:\Program Files\Git\」，若安裝的是32位元則是「C:\Program Files(x86)\Git\」，執行檔都放置在安裝位置的「bin」與「cmd」子資料夾下，如下圖：

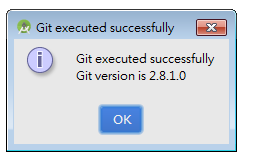


Step4 在Andorid Studio中設定

再打開Android Studio，點擊功能表中的「File/Settings」，左方選擇「Version Control/Git」，在右方的Git執行檔路徑「Path to Git executable」中開啟檔案瀏覽指向git.exe的位置，如下圖：



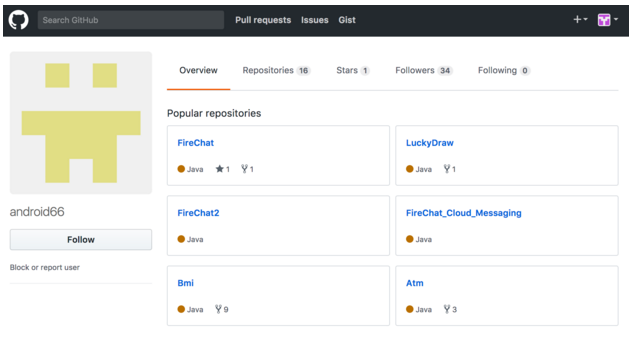
上圖指向的路徑為「C:\Program Files\Git\cmd\git.exe」，選擇完成後，可按下右方的「Test」測試鈕，測試是否能夠正常執行，如下圖:



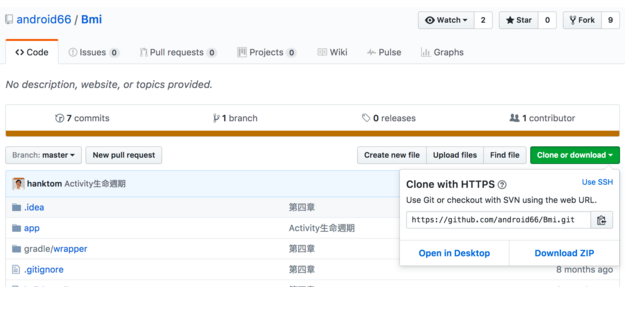
測試成功後，請按下「OK」完成環境設定。

 由Github網站取得本書範例

本書的範例專案網址為 <https://github.com/android66>，請以瀏覽器開啟，如下圖：



在「Repositories」檔案庫中點擊其中一個專案，如「Bmi」，顯示Bmi檔案庫的內容，如下圖：



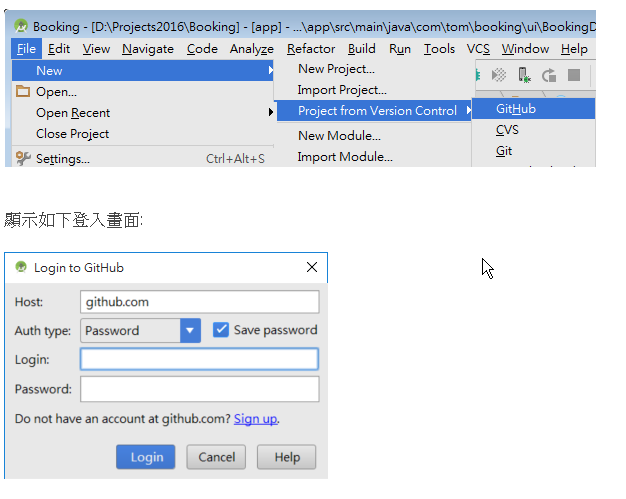
在上圖框線處即是Bmi檔案庫的取得網址，為： <https://github.com/android66/Bmi.git>

請複製此檔案庫網址，使用這個網址即可取得Bmi完整專案原始碼。

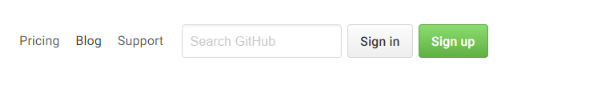
Android高效入門，0-2 取得範例專案程式碼，註冊github後取得專案複本

 取得Bmi範例專案

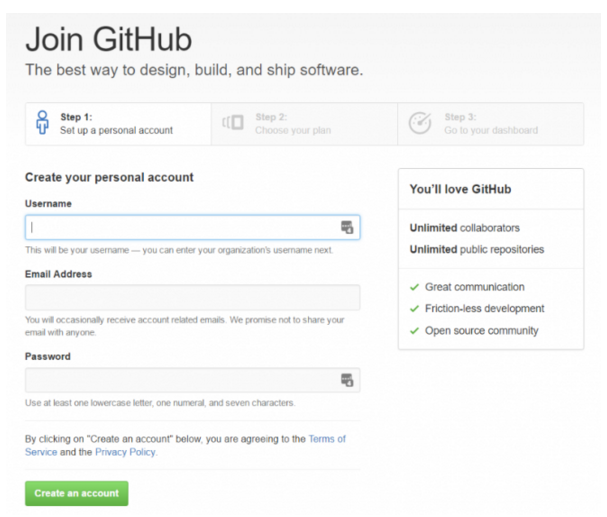
在Android Studio中點擊功能表「File/New/Project from Version Control/GitHub」由GitHub版本控制取得專案，如下圖：



若已有GitHub帳號請填入帳號與密碼，若無，則請點擊上圖的「Sign up」可開啟瀏覽器在github.com申請帳號，如下圖：



點擊網頁右上方的「Sign up」連結申請一個帳號，再輸入想要使用的帳號名稱（Username）、電子郵件(Email Address)與要使用的密碼(Password)，如下圖：



完成後請按下「Create an account」建立帳號，並收信驗證後，日後即可使用該帳號名稱(Username)與密碼登入GitHub取得檔案庫的專案資料。

帳號申請成功後，請回到Android Studio，在對話框中輸入帳號與密碼，並勾選儲存密碼（Save password），並按下「Login」登入，如下圖：



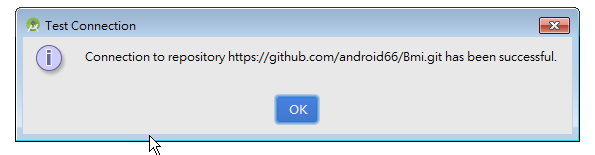
在對話框中輸入或貼上Github中的檔案庫網址，如Bmi專案的檔案庫網址：

<https://github.com/android66/Bmi.git>

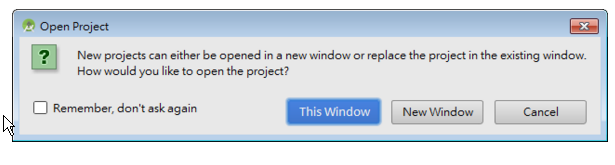
再指定下載專案回來的資料夾，請在本機中建立一個或指定一個已存在儲存專案的位置，如「D:\android\project2」，在該資料夾中不可有Bmi的資料夾，如下圖：



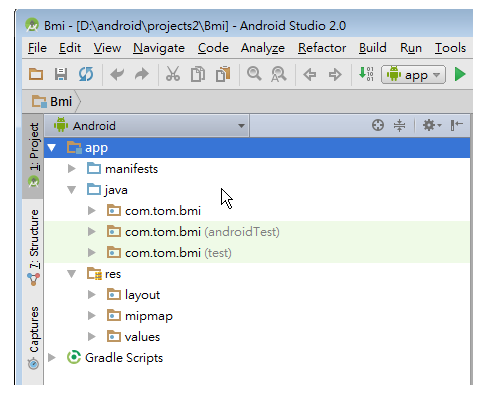
在開始取得專案複本前，請按下右上方的「Test」鈕測試一下是否能夠正確連線，如下圖：



測試成功後，請按下「Clone」取得複本，可下載GitHub上的檔案庫，在本機中建立一套複本，完成後會出現詢問視窗，要將專案在目前的Android Stduio視窗中開啟，或是為Bmi專案複本另開一個新視窗，如下圖：



若選擇「This Window」本視窗，會在原視窗開啟Bmi專案，如下圖：



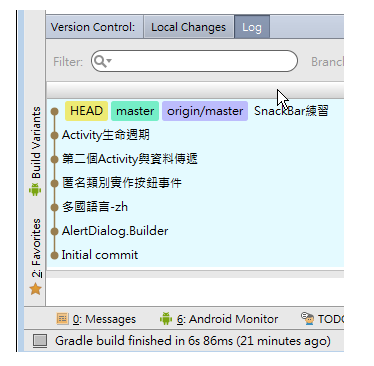
這個複本就是本書最後完成的Bmi專案範例，可執行、測試本書所實作的應用程式功能。

Android高效入門，0-3 切換各個版本，與遠端檔案庫切斷連結

切換至特定版本

版本控制的特色是能夠讓開發人員將每次階段的變動訂定其版本，並記錄專案在當下階段的程式碼，假如讀者想從某章節開始練習，但又不想從頭建立專案，可以先取得該專案的複本後，切換到專案的某一特定階段版本。

筆者以Bmi專案為例，先由GitHub中取得最新版本的專案複本後，按下「Alt+9」在Android Studio下方開啟「Version Control」視窗，再點擊視窗上緣的「Log」，如下圖：



上圖中的黃色「HEAD」字樣代表目前專案複本是最新的版本，該版本的資訊註名為「SnackBar練習」，由上圖可看到Bmi專案複本目前有幾個階段：

Initial commit

AlertDialog.Builder

多國語言-zh

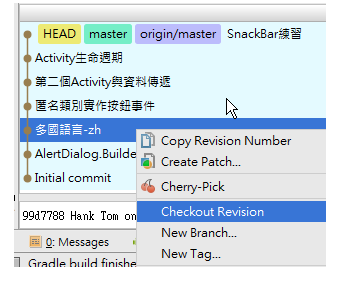
匿名類別實作按鈕事件

第二個Activity與資料傳遞

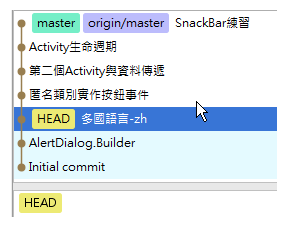
Activity生命週期

SnackBar練習

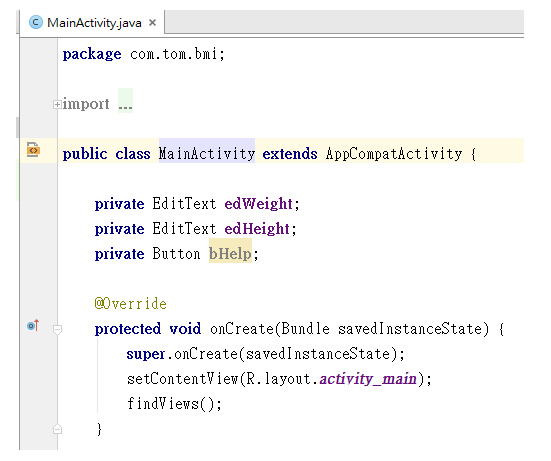
假設讀者想練習「匿名類別實作」，可以將Bmi專案複本切換至它的前一個版本「多國語言-zh」，這個名稱上按下滑鼠右鍵，選擇「Checkout Revision」，如下圖：



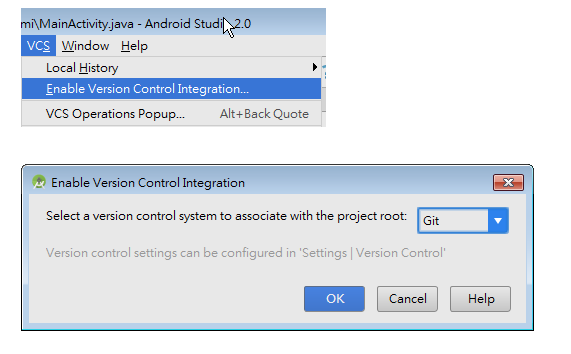
Bmi專案複本馬上會切換至「多國語言-zh」所完成的狀態版本下，如下圖的HEAD標示：



此時專案中的程式碼就是該版本的樣貌，如MainActivity的部份程式碼如下：



若切換到「Initial commit」初版送交版本後，原版本控制視窗會消失，原因是因為專案複本中的設定跑掉了，請再設定一下Bmi專案的版本控制，點擊功能表中的「VCS/Enable Version Control Integration」，再選擇「Git」，如下圖：



即可再按下「Alt+9」開啟版本控制視窗了。

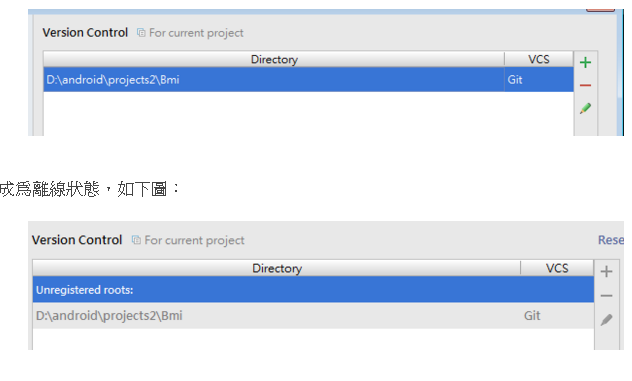
卸離檔案庫

由GitHub下載回來的Bmi專案是一份與版本控制系統連結的複本，在複本上的修改或新增功能假如沒有正確的帳號與密碼，是無法將修改的專案送交至檔案庫上的，因為你並不是GitHub該檔案庫的擁有者。

假設你從GitHub中取得複本後，想要卸離檔案庫並成為一般的獨立本機專案，請點擊功能表中的「File/Settings」，再選擇左方的「Version Control」，如下圖：



在右方點擊目前連結的專案「D:\android\project2\Bmi」，它的右方使用的是「Git」版本控制，再點擊一下右邊的減號卸離，如下圖：



再點擊對話框的「Apply」套用按鈕後，再按下「OK」完成設定。

完成後，本專案即成為一般的本機專案，讀者可以自由在這個專案中新增或修改功能，未來也可以將專案送至另一個版本控制系統中。

 協助與額外資訊

搜尋Facebook粉絲團「Android綠豆湯」，按讚以隨時得到更多有關Android開發新知與新版本的資訊，粉絲團網址為： <https://www.facebook.com/Androidpa>

相關文章:

[Android高效入門，0-2 取得範例專案程式碼，註冊github後取得專案複本](https://litotom.com/2017/03/03/android-book-chapter0-2/)

[Android高效入門，0-1 什麼是git、github? 為什麼要版本控制，從安裝必要的軟體開始](https://litotom.com/2017/03/03/android-book-chapter0-1/)

[Android Support Library—支援函式庫](https://litotom.com/2017/03/12/android-effective-03-1st-app-4/)

[Android SDK，SDK管理工具的使用](https://litotom.com/2017/03/09/android-effective-02-env-4/)

[畫面配置Layout，使用者看到的畫面設計](https://litotom.com/2017/03/12/android-effective-03-1st-app-1/)

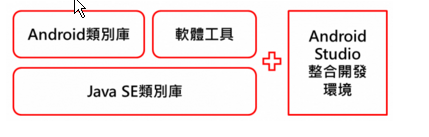
學Android的起點，把環境建置好並掌握每個角色後，什麼都好做

環境建置

想成為程式開發人員，最重要的就是開發環境的建置，沒有了環境什麼事都做不了，環境不熟悉，事事都不會順利。本文一開始將介紹如何在Windows作業系統中建置Android開發環境，包括JDK、Android SDK、Android Studio與模擬器，接著會帶領讀者熟悉Android Studio這個由Google主導的整合開發環境，最後建立一個新應用程式專案Bmi。在後續的文章中將會使用這個專案為例，進行Android應用程式開發的初步介紹與練習。

基礎環境－JDK與Android SDK

Android Studio是Google與IntelliJ IDEA合作開發專為開發人員設計的整合開發工具，必須有Java執行環境才能執行，先安裝JDK後，再安裝Android Studio。開發環境的架構中包括了JDK、Android Studio、Android SDK與模擬器，如下圖：



安裝JDK

Java 是由美商昇陽公司（於 2009 年被 Oracle 收購）所設計出的程式語言，開發 Java 程式所必需的 JDK 可由 Oracle Java 的官方網站下載，網址：

<http://www.oracle.com/technetwork/java/index.html>。

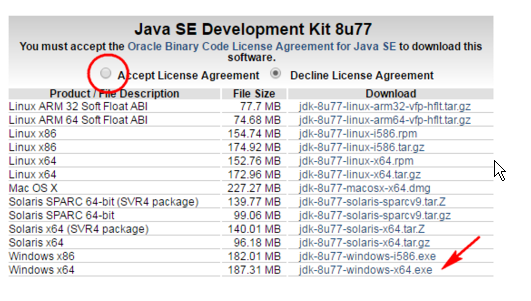
進入後可由上方的下載標籤或右方的 Software Downloads 中的 Java SE 進入下載頁面：



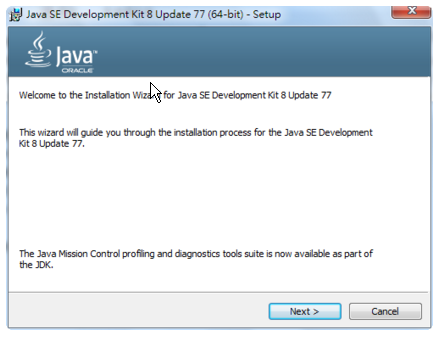
在下載分類中找到 Java SE，點擊「Download JDK」，如下圖：



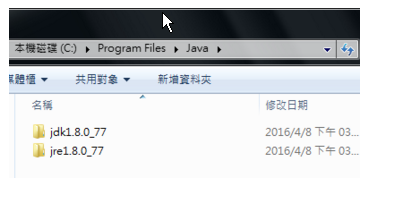
先勾選同意授權（Accept License Agreement），再點擊欲安裝平台的安裝檔（如 windows-x64）即可直接下載，如下圖：



執行本機下載完成的執行檔，進行安裝步驟：



安裝完成後會在 C:\Program Files\Java\目錄中產生兩個目錄， 一 個是JDK，另一個是 JRE 子目錄：



整合開發工具－Android Studio

Android Studio開發工具與SDK的下載與安裝過程中需要連線網路下載必要的SDK與模擬器，故需要一個良好的網路環境。在安裝前請確認目前主要磁碟機可用空間是否有6GB，Android Studio加上Android SDK的安裝需要不少空間，且在安裝過程中必須使用網路連線下載數GB的資料，請預先準備可用的空間與網路環境。

Android SDK

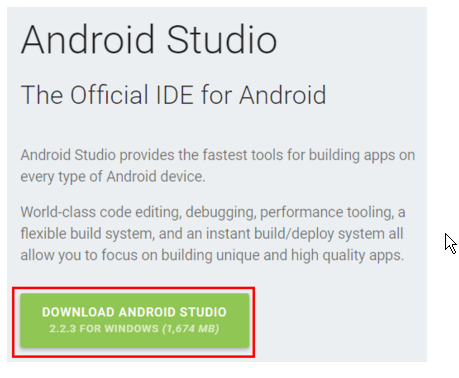
Android SDK是Google為了Android手機而設計並免費提供開發人員的工具庫，其中包括類別庫與工具程式，可從[http://developer.android.com](http://developer.android.com/)中下載取得，為了方便開發人員，Android Studio整合開發工具中也一併包裹了基礎的SDK，讀者可在安裝Android Studio時指定SDK要存放在那個目錄下。

Android Studio

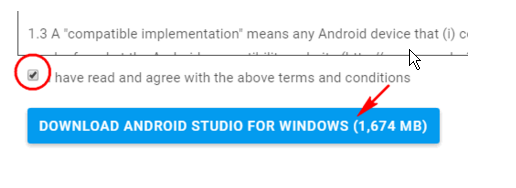
Android Studio是由Google與IntelliJ合作所開發的新一代Android整合開發工具，提供專案管理、編譯、程式設計、除錯、版面設計等功能，可由以下網址下載安裝：

<https://developer.android.com/sdk/index.html>

如下圖，請點擊「DOWNLOAD ANDROID STUDIO」：



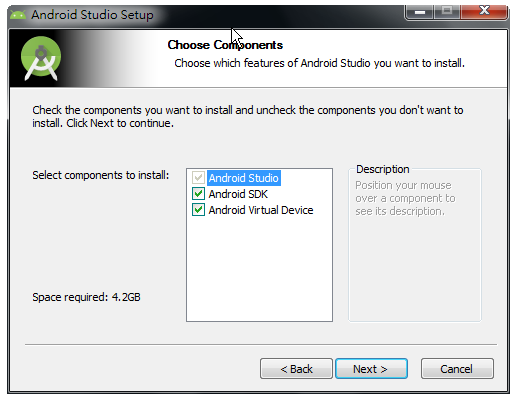
再勾選同意授權後，按下下載連結，將檔案下載儲存在本機，如下圖：



執行下載後的安裝檔，進行安裝步驟，首先是歡迎畫面，按下「Next」繼續，如下圖：



說明將安裝Android Studio、Android SDK與模擬器三項，按下Next繼續即可，如下圖：

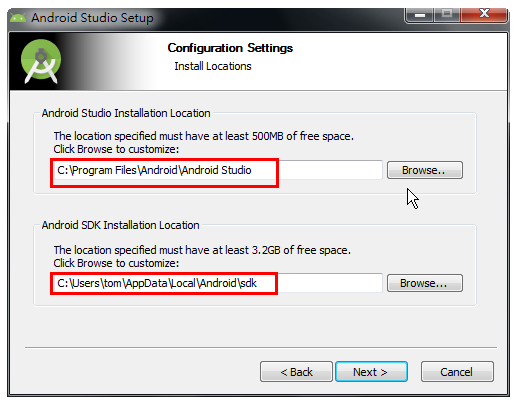


請按下授權同意「I Agree」，如下圖：

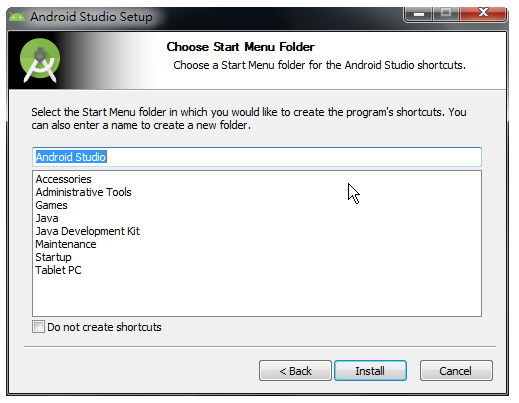


此時應記得安裝Android Studio的目錄，預設是「C:\Program Files\Android\Android Studio\」，另外，SDK則預設會裝在所登入使用者的家目錄下的「AppData\Local\Android\sdk」，會預設選擇這個目錄的原因是SDK經常需要更新，因此將它設定在使用者目錄下，可避免無法寫入系統權限的問題。這個位置並不好記憶，如果開發環境是自己能完全控制的（擁有管理者權限），可更換sdk安裝至別的目錄，如c:\Program Files\Android\sdk或C:\android\sdk\都是不錯的選擇。

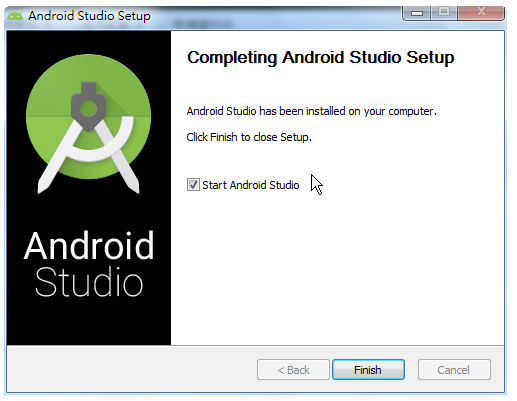
按下Next進行安裝即可，如下圖：



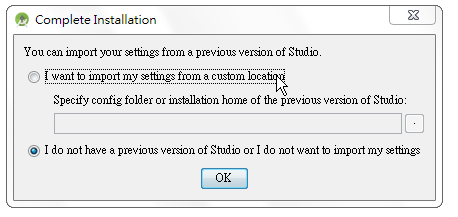
接著請按下「Install」進行安裝，如下圖：



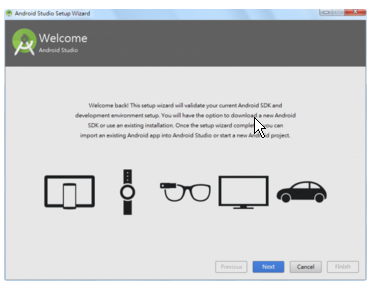
安裝完成後按下「Finish」，會自動執行「Android Studio」，如下圖：



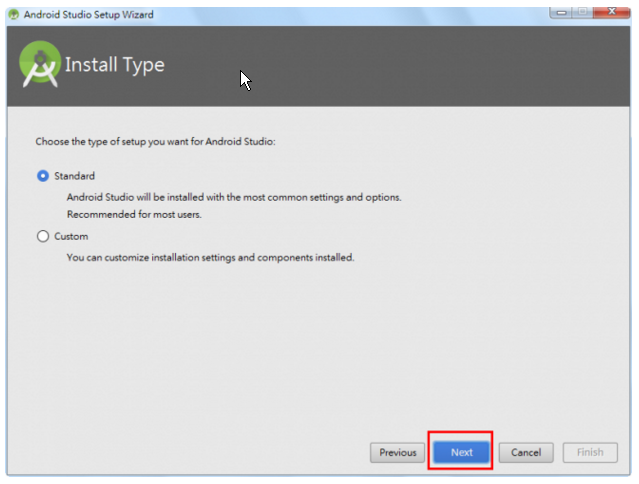
首次執行會自動檢查目前電腦中有無之前安裝過舊版Android Studio，若有可自動匯入設定值，若無，則選擇預設即可，請按下OK，如下圖：



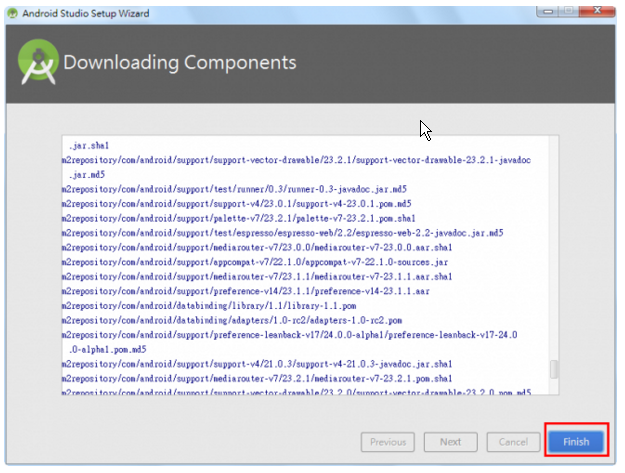
接下來進入設定精靈，此設定能根據所選擇開發的要件，自動判別從網路下載的軟體套件，請按下「Next」進入下一步，如下圖：



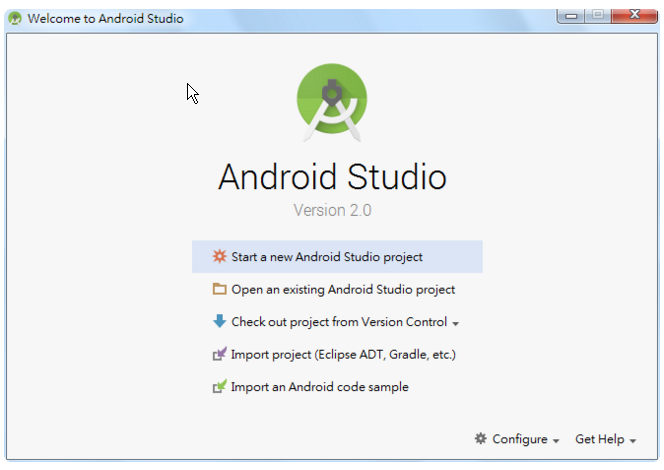
請使用預設的「Standard」標準安裝即可，請按下「Next」進入下一步，如下圖：



接著確認需要下載的項目，如Android Support Repository，按下「Finish」完成設定並開始下載工具函式庫，如等它下載完畢後，按下「Finish」完成。畫面如下圖：



自動進入Android Studio中的專案選擇畫面，如下圖:



相關文章:

[使用ADB登入模擬器](https://litotom.com/2017/07/05/ch7-2-adb/)

[使用內建模擬器映像檔，建立Android模擬器](https://litotom.com/2017/03/09/android-effective-02-env-5/)

[Android SDK，SDK管理工具的使用](https://litotom.com/2017/03/09/android-effective-02-env-4/)

[Android高效入門，0-1 什麼是git、github? 為什麼要版本控制，從安裝必要的軟體開始](https://litotom.com/2017/03/03/android-book-chapter0-1/)

[建立新的Android專案，認識Android Studio環境](https://litotom.com/2017/03/09/android-effective-02-env-3/)

Android Studio開發環境準備與安裝必要工具

Android Studio初始畫面

安裝完成後，Android Studio初次執行時，因為尚未建立任何專案，會出現Android Studio的歡迎畫面，在此畫面提供幾項功能：

建立一個新的Android專案(Start a new Android Stduio project)

開啟已存在的Android專案(Open an existing Android Studio projet)

從版本控制系統中取出專案(Check out project from Version Control)

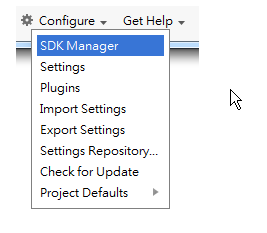
從Eclipse ADT、Gradle或其他匯入專案(Import project(Eclipse ADT, Gradle, etc)

匯入Android範例(Import an Android code sample)

下方還有：

設定(Configure)

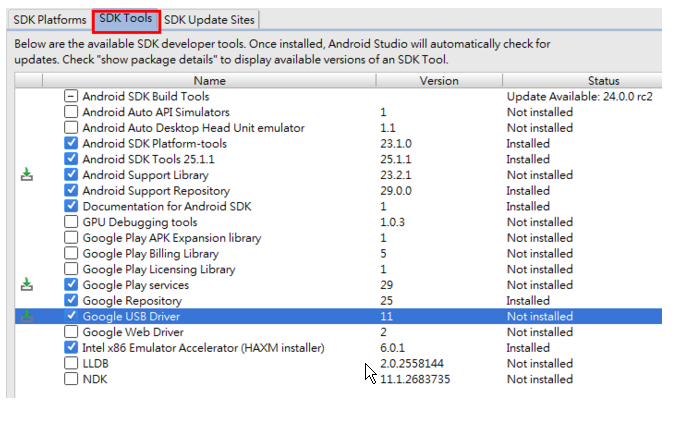
在設定中還有數個子設定項目，如下圖：



文件與操作說明(Docs and How-Tos)

補充安裝必要工具

雖然Android Studio與Android SDK已安裝完成，但在本書中會使用到必要的工具與函式庫，請讀者點擊畫面下方的「Configure」內的「SDK Manager」，開啟SDK管理器，切換至「SDK Tools」頁籤後，勾選必要的函式工具，如下圖：



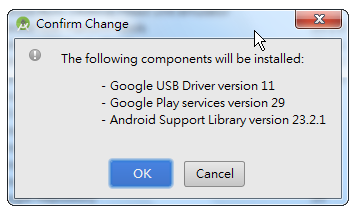
請勾選：

Android Support Library

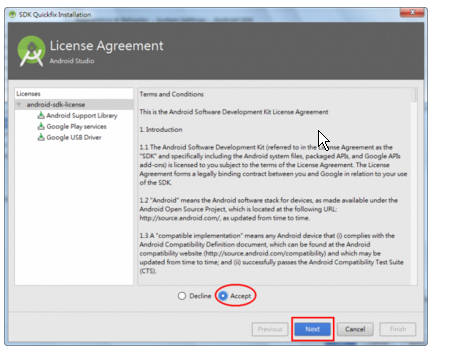
Google Play services

Google USB Driver

接著請按下方的「Apply」套用鍵，出現確認對話框後按下「OK」，如下：



出現授權畫面時，請按選擇「Accept」接受後，再按下「Next」進行網路下載與安裝，如下圖：



安裝完成後，請按下「Finish」結束，再按下「OK」結束SDK管理器。

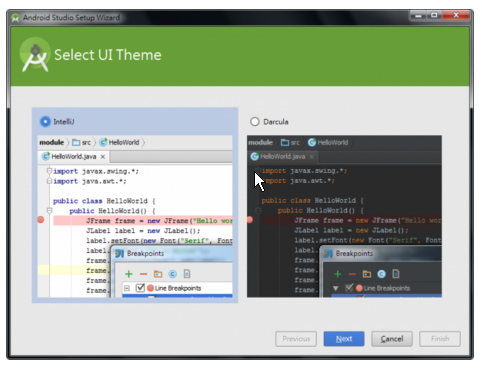
此時，整個Android Studio已安裝完成，亦安裝好可用的Android SDK，SDK中包含一套Android 6.0的平台函式庫與最新的工具程式，讀者可直接到2-4節建立第一個專案。

安裝注意事項

在安裝過程中，若有開啟防火牆，請在防火牆警示對話框出現時，按下「允許存取」，如下圖：



安裝完成後若出現使用者介面畫面，使用者可兩種環境配色，左邊是預設的IntelliJ白色基底，右邊則是Darcula暗黑色系，可依習慣選擇，選擇完成後按下「Next」，如下圖：



接下來，可以建立一個新專案了

相關文章:

[Android Studio 3，舊專案的gradle問題怎麼解決，為什麼?](https://litotom.com/2017/10/29/as3-gradle/)

[使用 Android Studio 3 開發新世代旗艦雲端資料庫 Cloud Firestore 之首部曲，類別庫導入與設定](https://litotom.com/2017/11/08/as3-firestore-1st/)

[Android高效入門-範例出現 Packages Unavaliable 對話框的解法 （已修正）](https://litotom.com/2016/06/30/android-packages-unavaliable/)

[在Android專案導入Firebase Crash Reporting機制，讓APP在出錯時回報除錯訊息](https://litotom.com/2017/04/19/firebase-crash-reporting/)

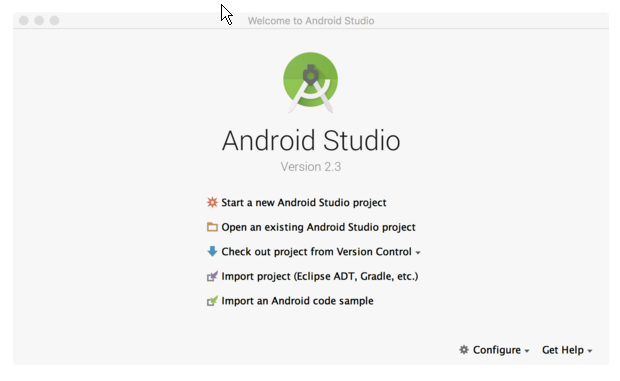
[使用第三方類別庫 ButterKnife 及外掛套件加速開發 (2017/1/3 更新)](https://litotom.com/2016/07/22/butterknife/)

建立新的Android專案，認識Android Studio環境

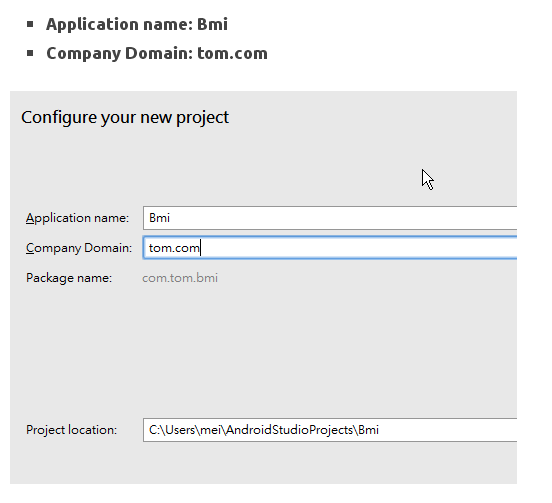
建立應用程式專案

筆者將建立一個新專案做為後續章節的範例，此專案目的是設計一個計算BMI身體質量指數的應用程式(APP)，步驟說明如下。

選擇建立一個新的Android專案，如下圖：

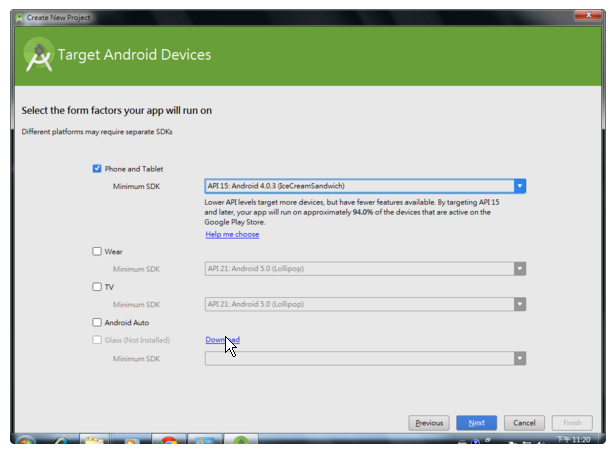


接著輸入應用程式名稱(Application name)、公司域名(Company Domain)，如下：



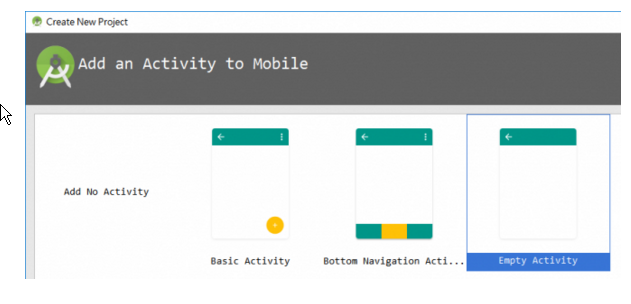
在公司域名下方會自動產生合適的套件名稱「com.tom.bmi」，這是Java語言的package名稱，先將公司域名倒過來寫，最後再加上專案名稱，若公司為abc.org，建立的專案叫xyz，則套件名稱為「org.abc.xyz」。

輸入完成後按下「Next」，進入專案執行環境的選擇，如下圖：

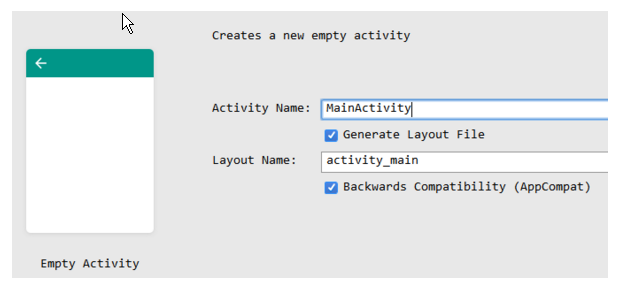


預設選擇為手機與平版類，最低支援Android版本為4.0.3(IceCreamSandwich)，其API編號為15，在此選擇預設即可，請按下「Next」進入下一步驟。

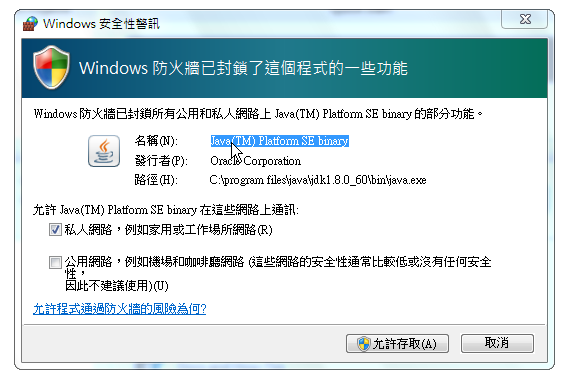
接著選擇自動產生的畫面類型，在Android中稱之為Activity(活動)，預設選擇的是空白畫面 (Empty Activity)，選擇預設即可，請按下「Next」，如下圖：



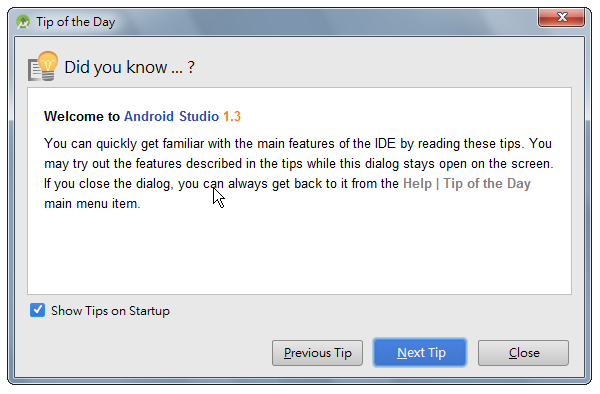
最後列出將會自動產生的檔案，有一個Activity類別名稱與標題都是「MainActivity」，畫面配置定義檔(Layout Name)「activity\_main.xml」，如下圖，此時皆使用預設值即可，請按下「Finish」完成專案建立。



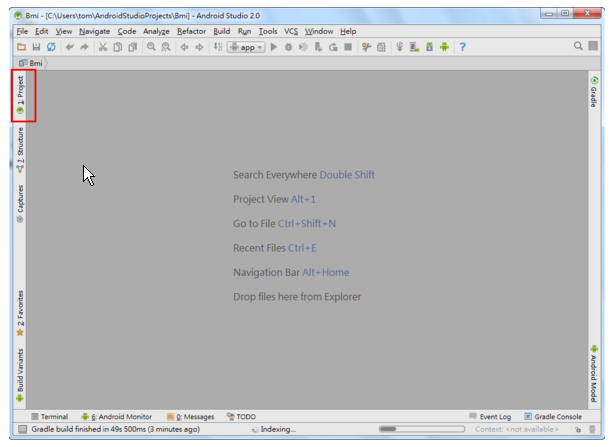
專案建立過程中若出現防火牆提示對話框，請按下「允許存取」，讓程式未來能夠通過防火牆，如下圖：



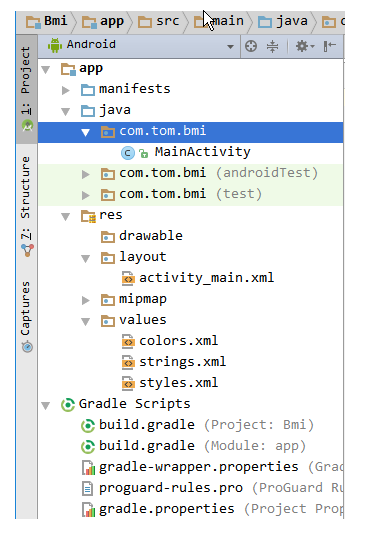
第一次執行Android Studio時預設都會打開今日提示對話框，可按下「Close」關閉它，若未來不想再看到提示，可取消勾選，如下圖：



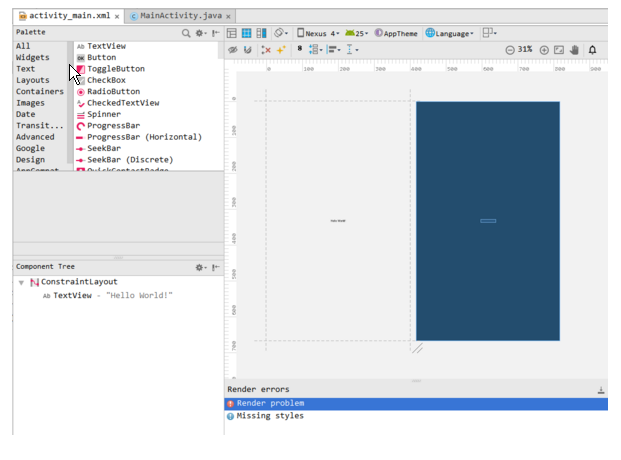
關閉提示後，就是Android Studio的整體設計畫面了，預設若未開啟專案區，請在畫面左方點擊「Project」如下圖：



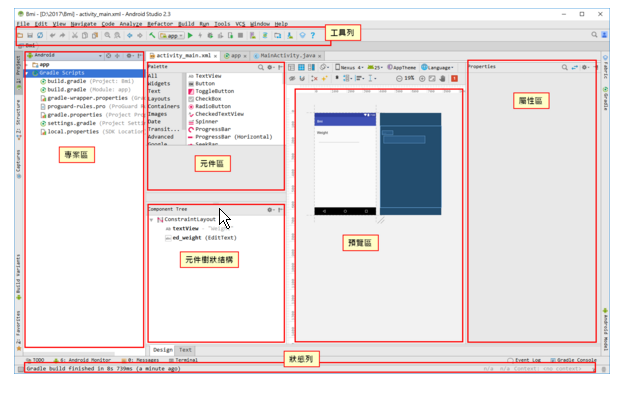
請在專案區中展開app、再展開app/java與app/res/layout的各個子目錄，如下圖所示：



請先開啟「app/res/layout/activity\_main.xml」，它是應用程式的主畫面版面設計檔，如下圖：



畫面區分為「工具列」、「專案區」、「預覽區」、「元件樹狀結構」、「屬性」與下方的「狀態列」，如下圖：



專案建立後，預設產生一個手機應用程式的畫面，畫面中有一個文字方塊元件，在這個畫面上方顯示「Hello world」。關於專案的架構與類別將於下一章詳細說明，在此時，筆者先就Android SDK所提示的工具進行介紹。

相關文章:

[建立第二個Activity，自動產生了那些東西?](https://litotom.com/2017/04/27/ch5-1-2nd-activity/)

[專案的資源，Layout、字串與圖檔等，但沒有R類別就麻煩了](https://litotom.com/2017/03/12/android-effective-03-1st-app-3/)

[Android 專案架構，什麼是AndroidManifest.xml?](https://litotom.com/2017/03/12/android-effective-03-1st-app-01/)

[Bmi專案功能之多國語言，資源抽出之重要性](https://litotom.com/2017/04/14/book-ch4-7/)

[畫面配置Layout，使用者看到的畫面設計](https://litotom.com/2017/03/12/android-effective-03-1st-app-1/)

Android SDK，SDK管理工具的使用

Android SDK工具與模擬器

Android Studio畫面上方的工具列已整合了Android SDK的三樣工具程式，分別為「SDK Manager」SDK管理器、「AVD Manager」模擬器管理與「Android Device Monitor」，其用途說明如下：

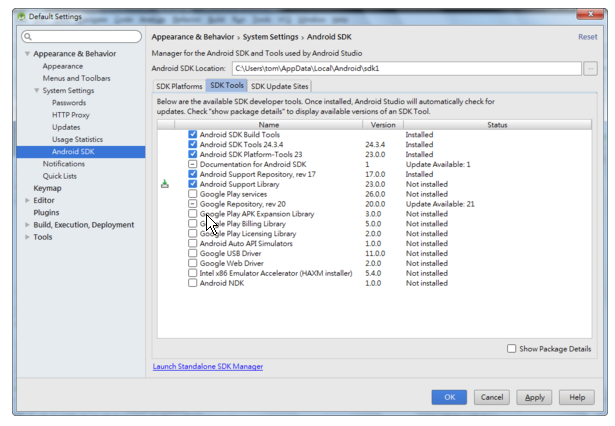
SDK管理器（SDK Manager）

由於不斷的改版與進步，Android有著多種版本的演進，如4.0、4.1、4.2、5.0、5.1與6等版本號碼，每個主要版本其都被指定一個唯一的編號值，稱為「API Level」，例如Android 5.1的API Level為22，Android 6.0為23，7.0為24，7.1則為25，每個不同的編號都擁有其專有的工具程式、平台(platform)與元件。Android依編號將這些檔案加以分類，方便使用者下載各種不同版本的套件，SDK管理器即是管理這些不同版本檔案的工具。

工具列圖示說明，請點擊SDK管理器圖示，如下：



開啟後會出現視窗管理介面，如下圖：



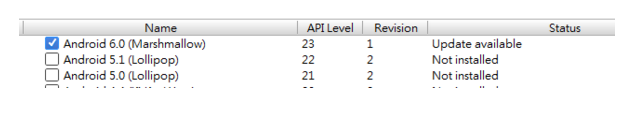
Platforms

第一個標籤SDK Platform，也就是目前可下載的Android平台，至少必須擁有一個平台才能開始開發應用程式，因為應用程式需要一個平台來編譯、執行，如下圖:

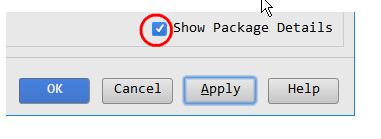


圖中已下載了最新版的Android 7.1，代號為「Nougat」牛軋糖，其編號為25。Android Studio的安裝檔會預先包裹最新一版的平台，本書使用的是7.1為例。由於它不是安裝7.1的所有套件，因此在右邊顯示為部份安裝(Partially installed)。

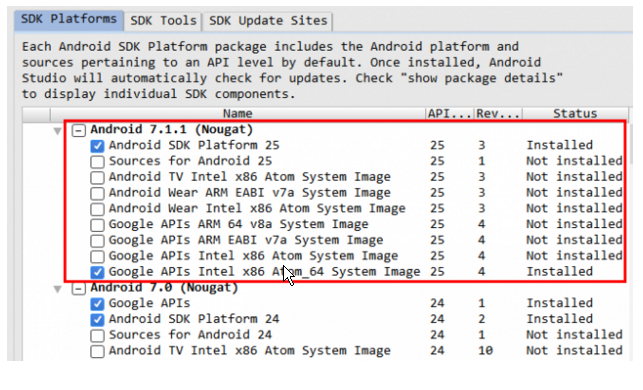
平台下載後，仍會有更新或改版的情形，會在這個畫面顯示，下圖為6.0目前有資訊可供更新：



每個Android版本下又有它的子項目，可點擊右下方的「Show  Package Details」顯示每個版本的詳細項目，如下圖:



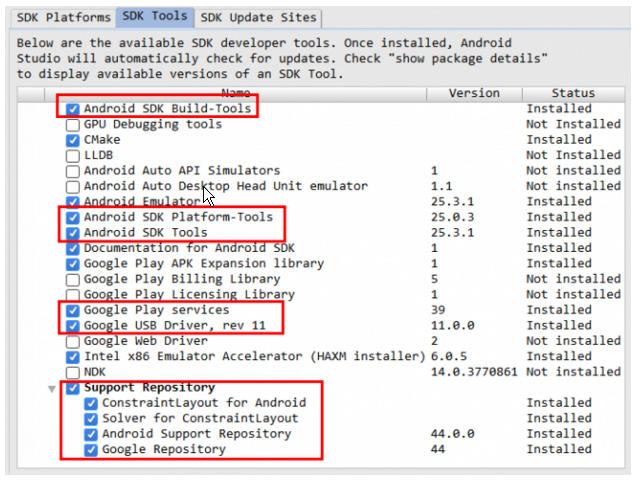
以「Android SDK Platform 版號」為必要項目，若想建立該版本的模擬器，則需再勾選一種模擬器映像檔，如下圖的「Google APIs Intel x86 Atom\_64 System Image」:



如果網路與磁碟空間許可，讀者也可試著下載其他的平台，勾選後再按下Apply套用即可，如7.0、6.0等，每個平台下載後大約會占到2GB磁碟空間，請先確認磁碟空用是否足夠。

SDK Tools

SDK工具是開發時需要的工具軟體、類別庫， 預設已經自動安裝了一些工具。開發工具組是很重要的項目，請勾選下圖所標式之項目，並完成安裝:



相關文章:

[使用內建模擬器映像檔，建立Android模擬器](https://litotom.com/2017/03/09/android-effective-02-env-5/)

[學Android的起點，把環境建置好並掌握每個角色後，什麼都好做](https://litotom.com/2017/02/22/android-effective-02-env/)

[使用ADB登入模擬器](https://litotom.com/2017/07/05/ch7-2-adb/)

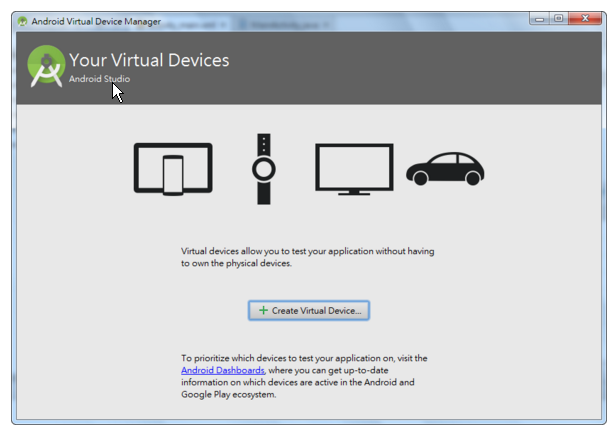
[Android高效入門，0-1 什麼是git、github? 為什麼要版本控制，從安裝必要的軟體開始](https://litotom.com/2017/03/03/android-book-chapter0-1/)

[Bmi專案功能之多國語言，資源抽出之重要性](https://litotom.com/2017/04/14/book-ch4-7/)

使用內建模擬器映像檔，建立Android模擬器

模擬器管理（AVD Manager）

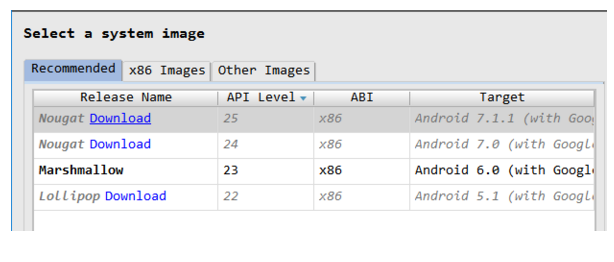
使用功能表「Tools/Android/AVD Manager」或按下功能表圖示可啟動模擬器管理工具，按下工具下方的「Create Virtual Device」可建立一個新的模擬器，如下圖：



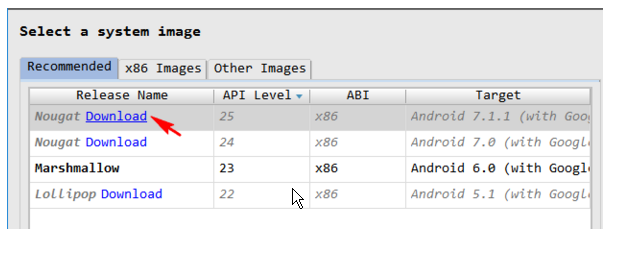
選擇模擬的裝置，如下圖中選擇「Nexus 5」，再按下「Next」：



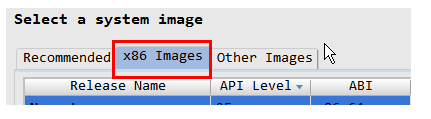
 再選擇建議項目中的「Nougat，API Level 25」後，按下「Next」進入模擬器的設定，如下圖：



未下載的項目可點擊Download直接下載，如下:



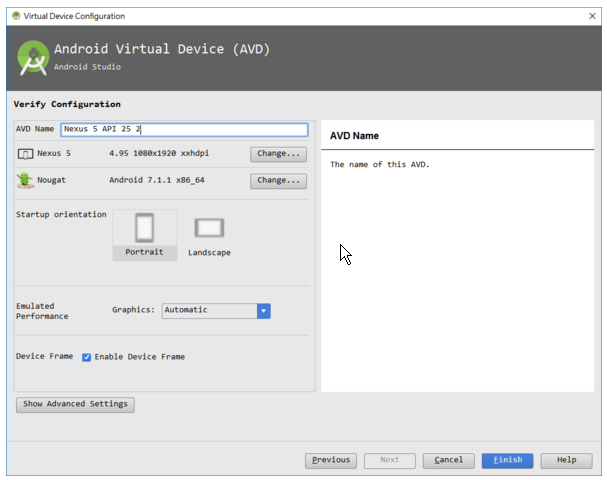
64位元的模擬器映像檔可點擊「x86 Images」頁籤選擇:





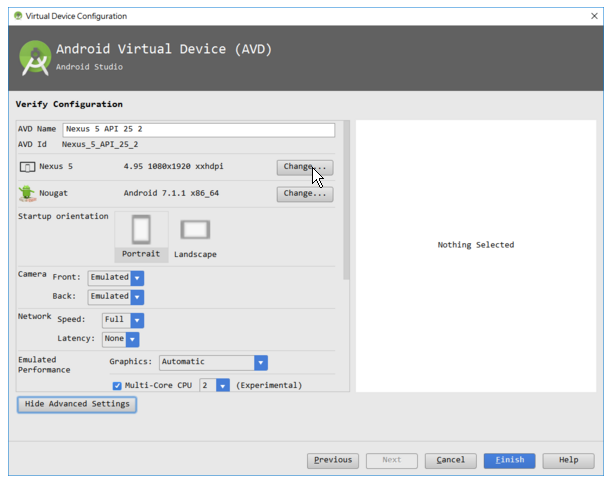
設定模擬器

進入模擬器的設定頁面，如下：

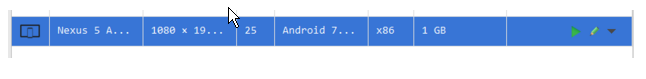


進階設定

按下畫面左下方的Show Advanced Settings可顯示進階設定值，如下圖：

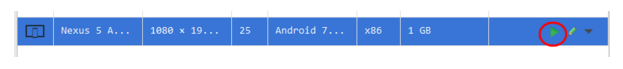


在此可調整模擬器使用的記憶體與儲存空間等設定，目前暫時不需要更改設定，請按下「Finish」完成模擬器建立，回到管理器的主頁面，日後需要再設定時，請在AVD Manager畫面中點擊模擬器的編輯鈕，如下圖：

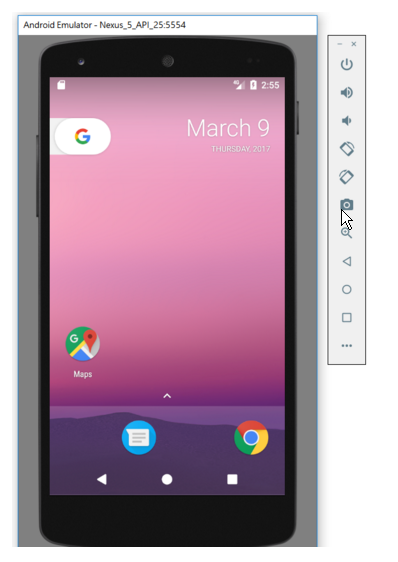


啟動模擬器

按下模擬器的啟動鈕，可啟動它，如下：

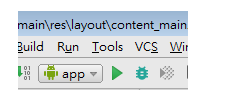


最後啟動完成的模擬器畫面，右方為功能鍵：



執行應用程式專案

模擬器啟動後，請回到Android Studio，執行專案的方式可按下上方的功能圖示，或點擊功能表「Run/Run ‘app`」，如下圖：





執行結果

在數秒鐘後，待Android Studio使用Gradle工具進行編譯、包裝後，會自動將APK應用程式安裝檔傳送至模擬器安裝、執行，模擬器的執行畫面如下圖：



這就是Bmi專案在Android裝置中執行的結果，開發Android應用程式的步驟為：

在Android Studio中建立新專案

設計畫面與功能

在模擬器中執行與測試

不斷重覆第2與第3步驟，直到功能都符合需求。

[版權聲明]

本文章版權為湯秉翰所有，授權範圍僅限綠豆湯網站使用，除Facebook之類社群等未更改本文章出處之分享行為不在此限，其他個人或公司未經作者同意，不得任意將本文章內容轉載至其他網站，或以任何形式重製，為以免觸犯著作權法，請尊重作者之智慧財產權。

相關文章:

[Bmi專案功能之多國語言，資源抽出之重要性](https://litotom.com/2017/04/14/book-ch4-7/)

[使用ADB登入模擬器](https://litotom.com/2017/07/05/ch7-2-adb/)

[建立新的Android專案，認識Android Studio環境](https://litotom.com/2017/03/09/android-effective-02-env-3/)

[生命週期的測試情境—使用Bmi專案](https://litotom.com/2017/05/25/ch5-5-lifecycle-bmi/)

[Android高效入門，0-2 取得範例專案程式碼，註冊github後取得專案複本](https://litotom.com/2017/03/03/android-book-chapter0-2/)

Android 專案架構，什麼是AndroidManifest.xml?

一個APP就是一個應用程式專案，在Android Studio中可進行APP的畫面設計、程式設計與測試等工作。一個專案有三個要項，說明如下：

AndroidManifest.xml載運清單檔案

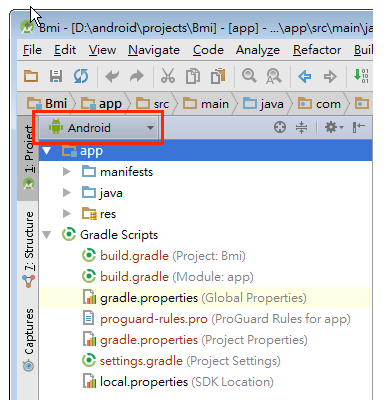
Java類別

資源

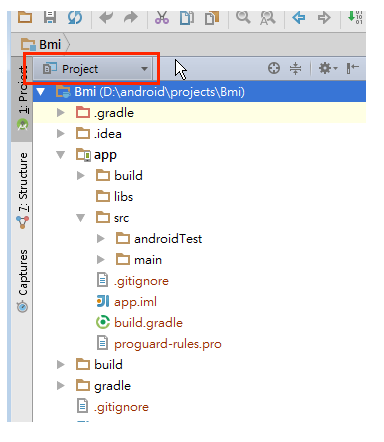
AndroidManifest.xml 載運清單

在每一個應用程式專案中一定有一個AndroidManifest.xml檔案，這個XML格式檔案記錄了這個APP內的重要資料，如package名稱、版本、Android版本資訊、權限與所使用到的類別（如Activity、Receiver、Service、Provider等）。

在Android Studio畫面左上方預設是以「Android」專用顯示配置，如下圖所標示處：



Android專案顯示配置會以開發APP時最合適的樹狀圖顯示，但實際的檔案與目錄結構並不一定是這樣的，請讀者點擊該下拉選單，並改選為「Project」以顯示專案真實的檔案結構，再展開專案、app、src等目錄後，觀察一個APP專案的實際情形，如下圖：



在開發Android應用程式時，大都使用「Android」專用顯示配置，在瞭解了不同的顯示配置的切換方式後，請再將其切換回Android配置，本書大部份都在這個配置下操作。

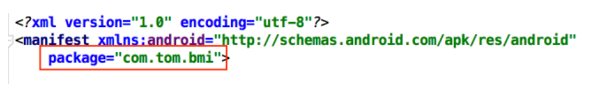
請再切換回「Android」專案配置後，再連擊「app」下的「manifests」，或展開「manifests」後再連擊「AndroidManifest.xml」以開啟該檔案，AndroidManifest.xml的根元素為「manifest」，其內部一定有一個「application」元素，而在application元素內會定義在這個應用程式會使用到的所有Android類別，如下圖：



上圖的application元素（圓孤方框）中有一個activity元素（虛線方框），代表這個APP目前只有一個畫面類別，類別名稱為MainActivity。

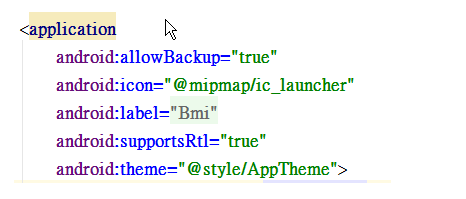
package屬性

每一個應用程式都必須定義一個與目前整個「Google Play商店」中上架的APP不重複的package名稱，但這個設定值是可以未來要上架時再更改的，不需要在一開始就要決定一個不重複的名稱，本例中使用的是「com.tom.bmi」，如下圖：



application元素的屬性

「application元素」擁有許多的屬性可供設定，「android:icon」屬性可指定這個應用程式所使用的代表圖示，可使用其他的圖檔替換成自訂的圖示。「android:label」屬性代表的是APP的名稱，本例為Bmi，另一個常用的是「android:theme」屬性，theme意思是「主題」，在Android中將元件外觀的資訊稱為樣式（style），而將樣式套用在整個應用程式或單一畫面中，就稱為theme，本例的主題設定為「@style/AppTheme」，如下圖：



activity元素的屬性

activity元素是用來宣告(定義)在這個應用程式中將會使用到的畫面控制元件，也就是Android的Activity類別，必要的屬性「android:name」是類別名稱，應使用類別的全名，如「com.tom.bmi.MainActivity」，但如果第一個字元為小數點時，會自動在類別名稱前補上這個專案的package名稱，因此，本例的「.MainActivity」就代表「com.tom.bmi.MainActivity」，如下圖：



上圖的activity元素內還有「intent-filter」元素，代表該Activity類別會在本應用程式啟動時被第一個執行，也就是一個APP的第一個出現的畫面類別。

[版權聲明]

本文章版權為湯秉翰所有，授權範圍僅限綠豆湯網站使用，除Facebook之類社群等未更改本文章出處之分享行為不在此限，其他個人或公司未經作者同意，不得任意將本文章內容轉載至其他網站，或以任何形式重製，為以免觸犯著作權法，請尊重作者之智慧財產權。

相關文章:

[建立第二個Activity，自動產生了那些東西?](https://litotom.com/2017/04/27/ch5-1-2nd-activity/)

[建立新的Android專案，認識Android Studio環境](https://litotom.com/2017/03/09/android-effective-02-env-3/)

[專案的資源，Layout、字串與圖檔等，但沒有R類別就麻煩了](https://litotom.com/2017/03/12/android-effective-03-1st-app-3/)

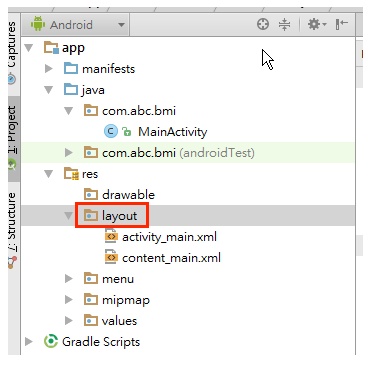
[畫面配置Layout，使用者看到的畫面設計](https://litotom.com/2017/03/12/android-effective-03-1st-app-1/)

[實作Activity上方選單Menu與下拉項目功能](https://litotom.com/2017/07/31/ch7-4-menu/)

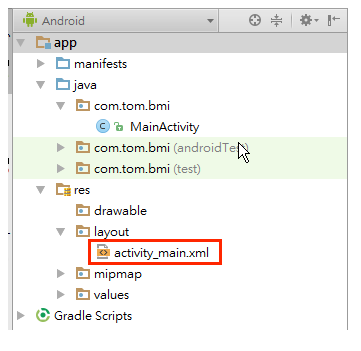
畫面配置Layout，使用者看到的畫面設計

畫面配置－Layout

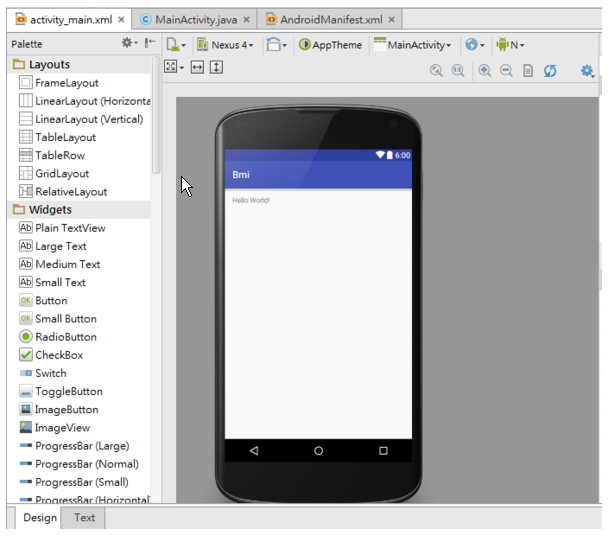
一個手機畫面是由XML檔案描述構成的，這個檔案稱為「Layout」畫面配置檔，統一放在專案下的「res/layout/」資料夾下，如下圖：



資料夾內有一個xml檔「activity\_main.xml」，這個檔案是在建立專案時選擇了空白活動(Empty Activity)時，由Android Studio自動產生出來，代表APP的第一個畫面。



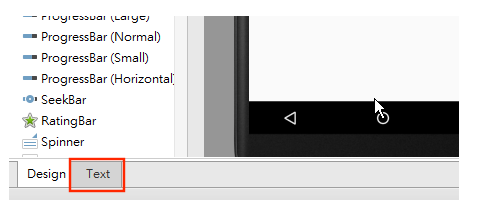
請連擊「activity\_main.xml」，會在編輯區域中開啟檔案，顯示如下圖：



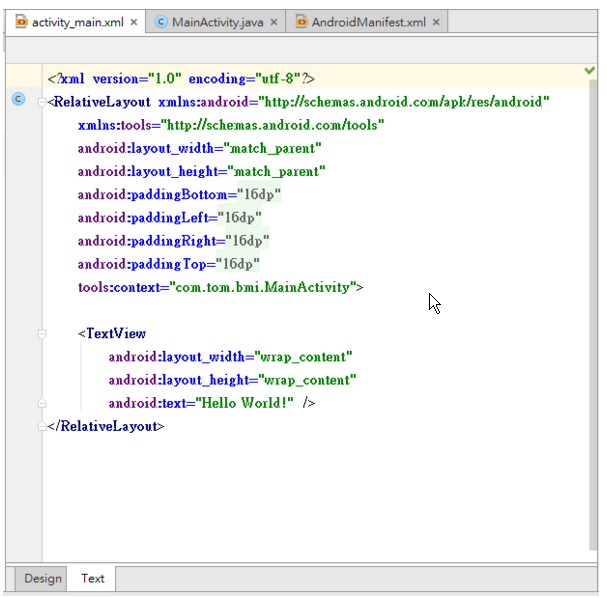
雖然我們點擊的是XML文字格式的檔案，但因為它屬於Android的畫面配置檔，因此預設以預覽方式顯示在畫面中，開發或設計人員可在這個介面下進行元件配置、調整與設計。如上圖所示，一個畫面的最上方是一條橫向長條狀的「Toolbar」，在其下方整塊是主要畫面區塊。

XML檔的原貌

Layout檔案的內容是XML格式，請點擊編輯區塊的左下方的標籤「Text」，就在「Design」標籤旁，如下圖：



點擊Text標籤後，編輯區域會換成該檔案的原始碼格式，也就是XML檔案的內容，如下圖：



畫面配置檔中的元素

辨認XML的架構對於進階的開發人員是必要的能力，以便日後能夠自行微調畫面配置中的元件資訊，請切換回設計畫面「Design」，在右邊有一個「Component Tree」元件樹，其樹狀結構如下圖：

整個畫面內容主要由一個「RelativeLayout」容器元件所構成，容器內有一個「TextView」元件，也就是畫面上顯示「Hello World!」的文字方塊元件。

[版權聲明]

本文章版權為湯秉翰所有，授權範圍僅限綠豆湯網站使用，除Facebook之類社群等未更改本文章出處之分享行為不在此限，其他個人或公司未經作者同意，不得任意將本文章內容轉載至其他網站，或以任何形式重製，為以免觸犯著作權法，請尊重作者之智慧財產權。

相關文章:

[專案的資源，Layout、字串與圖檔等，但沒有R類別就麻煩了](https://litotom.com/2017/03/12/android-effective-03-1st-app-3/)

[Android 專案架構，什麼是AndroidManifest.xml?](https://litotom.com/2017/03/12/android-effective-03-1st-app-01/)

[建立新的Android專案，認識Android Studio環境](https://litotom.com/2017/03/09/android-effective-02-env-3/)

[建立第二個Activity，自動產生了那些東西?](https://litotom.com/2017/04/27/ch5-1-2nd-activity/)

[Activity是什麼? Java類別，沒有它就沒有互動了](https://litotom.com/2017/03/12/android-effective-03-1st-app-2/)

Activity是什麼? Java類別，沒有它就沒有互動了

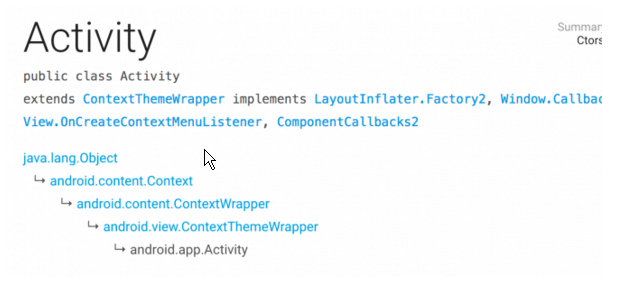
Java類別

所有在APP中會使用到的Java類別全都放在「src」目錄，在設計一個Android應用程式時，最常用到的是Activity類別、資料庫存取類別與其他因為需求所設計出的類別。

Activity類別

Android團隊為了讓開發者能夠快速開發自己能夠控制的畫面，設計了「android.app.Activity」類別，類別中已設計許多有關畫面控制、取得各類工具的方法，開發者設計類別時先繼承Activity類別，就能在這個類別中撰寫客製化的程式碼，開發符合自己需求的畫面功能。

Activity類別的繼承情形如下圖：

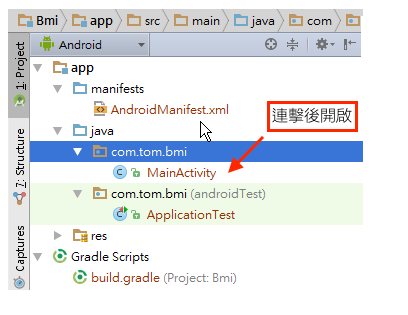


上圖可看出Activity的父類別為ContextThemeWrapper，再上層父類別為ContextWrapper，再上層則為「Context」類別。

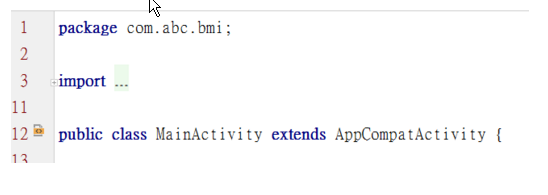
Activity的子類別

Android版本變化非常快速，往往每年都有新的功能的加入，這也讓原本釋出的類別庫不再能夠滿足新的功能。利用Java語言的繼承特色，可快速地擴充舊的元件，加入新的功能。例如在Android 3.0時，加入了Fragment功能，便釋出FragmentActivity擴充原Acitivity類別的功能，另外，也釋出ActionBarActivity以符合ActionBar功能。

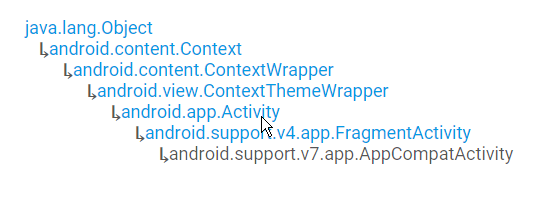
接下來請展開「app」下的「java」，再展開套件「com.tom.bmi」，並連擊「MainActivity」類別，如下：



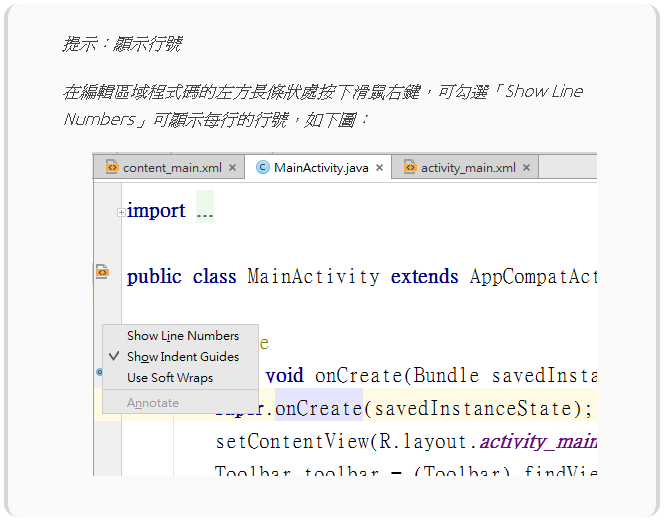
MainActivity.java的第一行是宣告這MainActivity的package套件名稱為「com.tom.bmi」，接著是import語法，重要的則是類別的宣告，如下圖：



上圖的第12行代表MainActivity繼承了「AppCompatActivity」類別，AppCompatActivity是新版的類別，它的套件為「android.support.v7.app」，屬於support v7類別庫（support v7類別庫的導入在本章最後會詳細說明）。其主要目的是「向下相容」，自從Android 3.0以來使用的ActionBarActivity已不建議使用，因此，設計一個Activity時，建議都繼承AppCompatActivity，才能讓開發的APP都能在Android 3.0、4.x、5.x與6.x的環境中正常執行。下圖是AppCompatActivity的類別繼承圖：

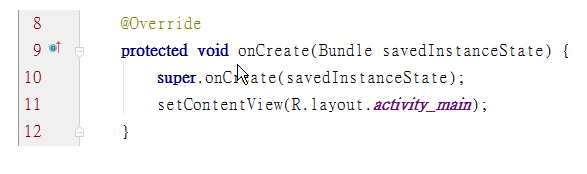


因為MainActivity繼承了AppCompatActivity，也代表MainActivity也具備了上層所有類別的方法與能力，如Activity、Context等類別。



MainActivity的onCreate方法

一個Activity類別最重要的是「onCreate」方法，在產生Activity時會自動執行onCreate方法中的程式碼，之後才會顯示在手機畫面上，



上圖在onCreate方法中的第10行呼叫父類別的onCreate方法，主要是進行資料設定，接下來的程式碼設計了一項工作，說明如下：

設定類別使用的畫面配置檔



|  |  |
| --- | --- |
|  |  |

setContentView方法是android.app.Activity類別提供的方法，用來設定Activity本身的畫面配置資源，在Activity中存取資料需透過「R」類別，R類別代表應用程式內的資源，有關資源會在下一節詳細介紹。

[版權聲明]

本文章版權為湯秉翰所有，授權範圍僅限綠豆湯網站使用，除Facebook之類社群等未更改本文章出處之分享行為不在此限，其他個人或公司未經作者同意，不得任意將本文章內容轉載至其他網站，或以任何形式重製，為以免觸犯著作權法，請尊重作者之智慧財產權。

相關文章:

[轉換畫面後，按返回鍵回到前一個Activity時怎麼取得結果? onActivityResult方法](https://litotom.com/2017/06/04/ch5-7-activity-result/)

[Bmi專案功能與事件處理，什麼是匿名類別?](https://litotom.com/2017/04/14/book-ch4-8/)

[建立第二個Activity，自動產生了那些東西?](https://litotom.com/2017/04/27/ch5-1-2nd-activity/)

[來證明你會了吧，Atm專案之登入功能實務](https://litotom.com/2017/06/03/ch5-6-atm-login/)

[生命週期的測試情境—使用Bmi專案](https://litotom.com/2017/05/25/ch5-5-lifecycle-bmi/)

專案的資源，Layout、字串與圖檔等，但沒有R類別就麻煩了

資源

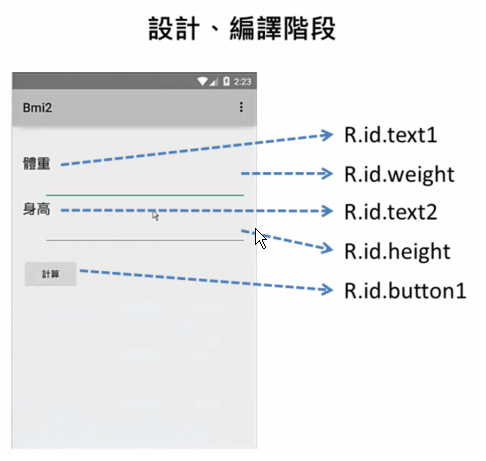
一個應用程式除了Java類別外，一定會用到其他如畫面配置、圖檔、樣式甚至字串資料等，這些在Android中稱之為「Resources資源」，資源依其類型放置在專案的「res」資料夾中。在一個Java程式中存取檔案的代價是高的，因為要經過開啟檔案、讀取、解析、辨識等工作，但程式碼又經常需要存取畫面中的元件。

在現代的手持裝置程式架構中，先整理一個應用程式中所使用的資源，如圖檔、畫面XML檔、字串等，為每個資源「造冊」、「分類」，產生一個「資源造冊類別」，讓程式直接利用資源造冊類別取得資源。在Android中，這個類別的名稱就叫「R」，其套件名稱就是應用程式的套件名稱，如本例為「com.tom.bmi」，類別的全名為「com.tom.bmi.R」。

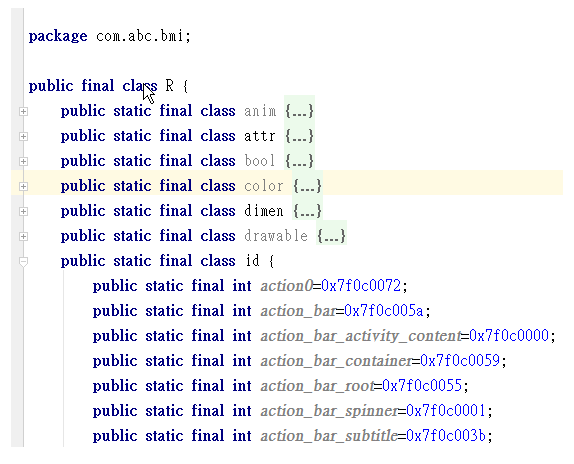
R類別

R類別是自動產生的，不需要手動加入，也不需要修改它，當新增或修改資源時，皆會自動產生新的R類別，例如新加入一個圖檔、修改畫面配置檔中的元件屬性或加入一個元件時，只要按下儲存後，會立即產生對應最新資源狀態的R類別。

在Layout版面內的每個元件所設定的id屬性的用途為辨識，一個應用程式的所有版面(Layout)中的每一個元件都會被自動掃描，依照id值製成一個對照表，成為一個R類別，未來可在Activity類別中使用這個R類別快速存取到元件。



每個在資源中的值都會自動產生，自動指派不重覆的int整數值，畫面中的元件(View)所設定的id值，可使用R.id.xxx存取，如果是字串資源(在/rec/values/strings.xml)則會自動分配為R.string.xxx，圖檔資源則會是R.drawable.xxx，以圖檔檔名為命令方式。自動產生的R類別原始碼如下圖：



除了上述常用的資源外，還有如R.layout.xxx(版面)、R.array.xxx(陣列)、R.anim.xxx(動畫)與R.dimen.xxx(大小)等資源。

圖檔－drawable

在應用程式中使用的圖檔，如圖示檔、按鈕的底圖與在畫面中使用到的所有小圖示等，都應放置在這個目錄下。

畫面配置－layout

活動畫面的版面配置設定檔，有關畫面設置，如按鈕、輸入方塊、文字區塊、下拉選單的的畫面設計檔，使用XML格式儲存在這個目錄下。

選單－menu

顯示在畫面上方的橫向列的功能選單，使用者以XML檔案方式設計選單的項目與內容，儲存在此目錄下，可以在活動中使用、設定這些選單。

資料－values

字串(string)、顏色(color)、方塊區域大小(dimension)或字型大小(text size)在Android中都可以設定為資料(value)，設定後會成為R類別的項目之一，例如R.string.字串名、R.color.顏色名等資源，可使用在類別的程式中。

[版權聲明]

本文章版權為湯秉翰所有，授權範圍僅限綠豆湯網站使用，除Facebook之類社群等未更改本文章出處之分享行為不在此限，其他個人或公司未經作者同意，不得任意將本文章內容轉載至其他網站，或以任何形式重製，為以免觸犯著作權法，請尊重作者之智慧財產權。

相關文章:

[建立新的Android專案，認識Android Studio環境](https://litotom.com/2017/03/09/android-effective-02-env-3/)

[Bmi專案功能之多國語言，資源抽出之重要性](https://litotom.com/2017/04/14/book-ch4-7/)

[建立第二個Activity，自動產生了那些東西?](https://litotom.com/2017/04/27/ch5-1-2nd-activity/)

[上戰場了，寫個Bmi APP吧，專案Layout與元件設計](https://litotom.com/2017/04/03/book-4-4/)

[Activity是什麼? Java類別，沒有它就沒有互動了](https://litotom.com/2017/03/12/android-effective-03-1st-app-2/)

Android Support Library—支援函式庫

Android Support Library－支援函式庫

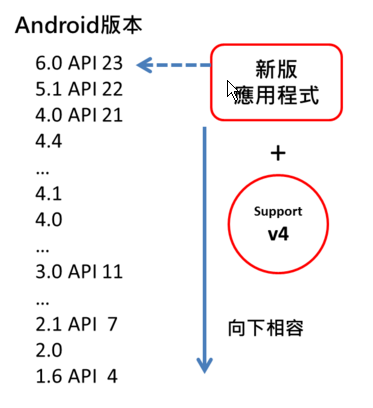
Android的版本的更新頻率大概是一年兩個版號，每推出一個新版本時，往往會加入一些舊版本沒有的功能，而這些功能都是新加入的類別。因此，假設開發人員使用Android 6.0開發應用程式時使用了新類別，但舊的手機內的版本卻是4.0.3時，在開發完成時，舊版本的手機中並不具有新類別，那怎麼安裝、執行依照新版本所開發的應用程式呢？

Google提供了「Android Support Library」套件包來解決這個問題，用來保証用新版本開發的應用程式可以在舊版本的裝置上正常執行。你可稱它為「Android支援函式庫」，但筆者比較偏好稱為「Android照顧包」，因為它的用途就是用來「照顧」舊版裝置能執行新版的應用程式啊。

Android Support v4支援函式庫

Android的API Level 4的版本對應為Android 1.6版，這個函式庫是為了照顧1.6版與更高版本所設計的，如果你想要向下支援到1.6版的裝置，應該在專案中加入這個函式庫，加入函式庫的方法會在後續內容說明。

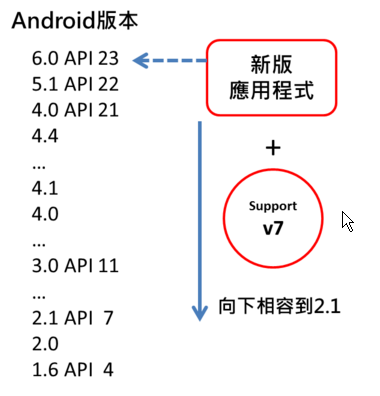
所以，如果加入support v4函式庫，代表若「引用正確的類別」就能夠在1.6版以上的裝置正確執行具有新功能的應用程式，如下圖：



Android Support v7支援函式庫

目前Android手機裝置幾乎沒在使用1.6版了，但假設只需要向下支援到Android 2.1(API Level 7)，以下就不再支援了。此時，可使用「Android Support v7」支援函式庫，如果不考慮執行裝置為1.6-2.0時，可導入此函式庫達到支援到Android 2.1。

v7函式庫是以v4為基礎設計的，因此v4為v7的依賴項目，因此應用程式開發時，要導入Android Supportv4與v7兩個函式庫，向下相容版本如下圖：



appcompat v7函式庫

支援使用Action Bar與新的使用者介面設計模式所推出的「appcompat v7」是為了要讓Android 2.1(API Level 7)以上的裝置都能正常執行Android 3.0才提出的ActionBar、AppCompatActivity、AppCompatDialog等新類別。

v7 carview函式庫

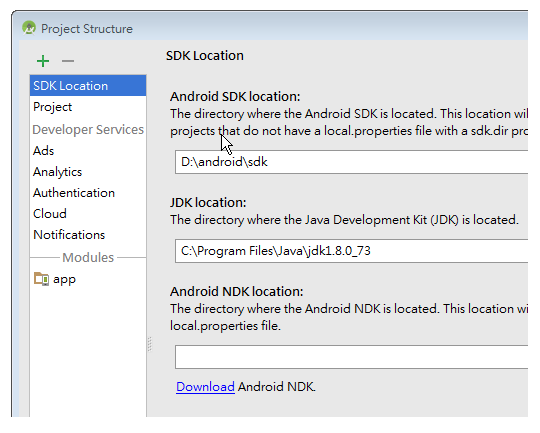
支援舊版本(Android 2.1以上)能夠使用「CardView」元件，這是Android　5.0開始推出的新版元件，能在如名片般的卡片式方塊中顯示資訊，符合最新的「Marterial設計」規範。

v7 recyclerview函式庫

支援舊版本(Android 2.1以上)能夠使用「RecyclerView」元件，是Android　5.0開始推出的新版元件，提供更有效率地以清單方式顯示大量的資料，需要以清單列表方式顯示數量很多的內容時，使用RecyclerView利用限定的資料項目，並重覆使用這些項目物件，以避免過多資源的耗用。

專案中導入函式庫

可使用工具介面加入或直接修改專案區「Gradle Scripts」下的「build.gradle(Module:app)」，筆者先以工具設定後，再打開該檔案觀察所加入的內容，請點擊功能表的「File/Project Structure」，這是專案架構配置設定工具頁面，如下圖：

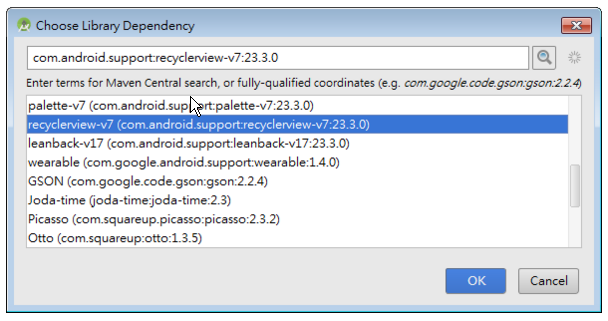


在專案架構的左方請先點擊Modules模組內的「app」，再點擊「Dependencies」相依函式庫設定，如下圖：



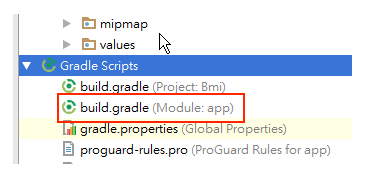
相依函式庫代表本專案除了內建的類別外，還加入了額外需要的類別庫，目前預設已有「junit」與「com.android.support:appcompat-v7」兩個類別庫，

可按下右方的加號「+」圖示加入新的函式庫，請點擊加號後再點擊「Libaray denpendency」，新增函式庫的選擇畫面如下：

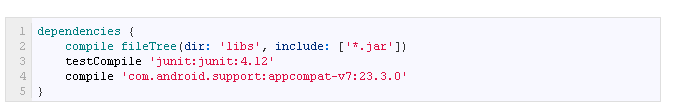


可在最上方輸入欲篩選的字，或直接找到需要的函式庫，上圖為選擇recyclerview v7的例子。

在這畫面中已加入的函式庫會更動到專案的app模組的Gradle設定檔，請在專案區展開「Gradle Scripts」，開啟「build.gradle(Module:app)」檔案，如圖：



在其下方的「dependencies」區塊中，就是本專案會導入的額外函式庫，如下：



使用compile命令，可在本機的Gradle檔案庫中自動導入到專案內，當本機的Gradle檔案庫中沒有這個函式庫時，會自動連到網路上的Maven檔案庫，下載必要的函式庫後，並儲存在本機的使用者家目錄下的「.gradle」子目錄內。

[版權聲明]

本文章版權為湯秉翰所有，授權範圍僅限綠豆湯網站使用，除Facebook之類社群等未更改本文章出處之分享行為不在此限，其他個人或公司未經作者同意，不得任意將本文章內容轉載至其他網站，或以任何形式重製，為以免觸犯著作權法，請尊重作者之智慧財產權。

相關文章:

[建立新的Android專案，認識Android Studio環境](https://litotom.com/2017/03/09/android-effective-02-env-3/)

[Activity是什麼? Java類別，沒有它就沒有互動了](https://litotom.com/2017/03/12/android-effective-03-1st-app-2/)

[使用內建模擬器映像檔，建立Android模擬器](https://litotom.com/2017/03/09/android-effective-02-env-5/)

[Android 專案架構，什麼是AndroidManifest.xml?](https://litotom.com/2017/03/12/android-effective-03-1st-app-01/)

[學Android的起點，把環境建置好並掌握每個角色後，什麼都好做](https://litotom.com/2017/02/22/android-effective-02-env/)