Layout版面配置能做什麼? 有那些Layout可以選擇，談LinearLayout與RelativeLayout

本章節仍然繼續使用第三章所產生的Bmi專案，主要內容為設計Bmi的畫面元件，以及當使用者按下按鈕進行計算時，MainActivity類別內的程式設計，包括了取得畫面元件、轉型與事件處理相關的知識。

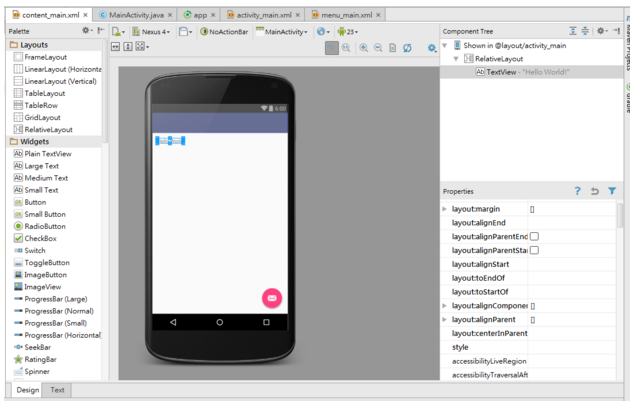
版面配置Layout

Android提供了幾個常見的Layout類別供放置需要的元件，如輸入方塊、文字方塊、按鈕等元件。元件在畫面上的呈現方式與Layout類別有很大的關係，選擇合適的Layout後，再加上調整元件本身的屬性後，才能設計出符合需求的畫面。

本節先介紹設計環境後，再著手使用RelativeLayout(相對)與LinearLayout(線性)兩個常用的Layout，最後再介紹其他的Layout類別。

版面配置設計畫面

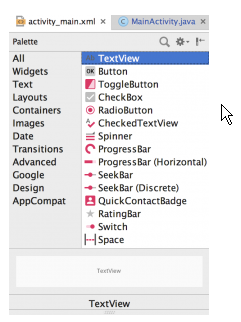
請先開啟「res/layout/content\_main.xml」，預設以一個版面配置的設計畫面，讓開發或設計人員可在這個環境下進行設計，整體畫面如下圖：



畫面中的每個區塊都有其用途，詳細說明如下：

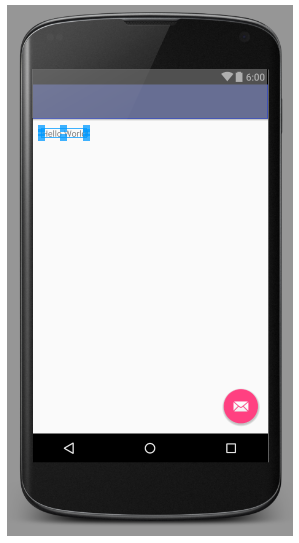
左方為可使用元件區(palette)

元件區是展示目前能夠使用的Android元件，使用滑鼠點擊需要的元件後，再放置到中間預覽設計區中，左方為功能性質為元件分類，下方是選取元件的預覽示意圖。



中間為預覽設計區

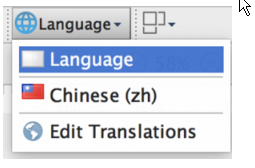
目前版面配置的預覽區塊，也是編輯區域，使用滑鼠單擊可選擇元件，若在元件上雙擊滑鼠，會顯示該元件最常使用的屬性對話框，可在對話框中設定id或顯示文字等屬性。



上方則為快速功能圖示

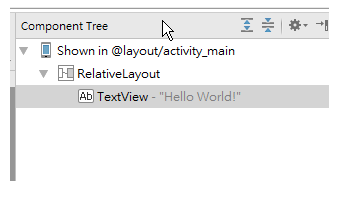


上排為版面的外觀顯示設定，可設定如預覽手機的外貌、翻轉、樣式與本版面使用在那一個類別等，另外，亦可為本版面新增多國語言的文字資源，如下圖：

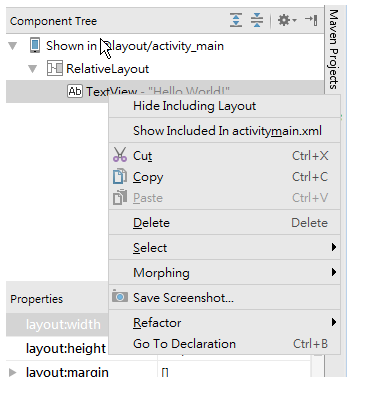


多國語言資源設計將在本書後續內容中介紹。

左下方為元件樹（Component Tree）



元件樹以樹狀結構顯示版面中的元件，亦可將欲新增的元件置放在此，選擇某特定元件後，可在元件上按下滑鼠右鍵，選擇最後一項「Go To Declaration」可快速切換為XML原始碼並移至該元件處，如下：



原始碼如下：



亦可使用快速鍵「Ctrl+B」，快速切換至原始碼定義處。

右下為元件屬性調整區

被選擇元件常用的屬性設定區，很常用的功能。

RelativeLayout

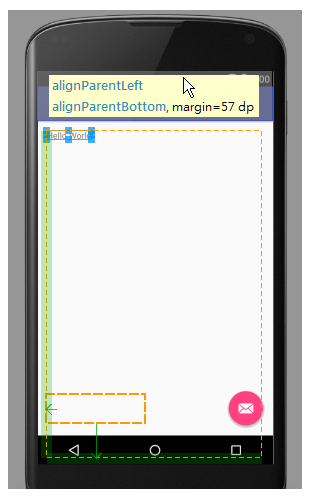
RelativeLayout是Android Studio在產生Activity時使用的預設Layout，在此版面的元件都需要參考一個對象，這個對象可以是任一個元件或是元件所在位置的元件，在本例也就是面版本身(parent)，也可稱為元件所在容器。

對齊對象為面版(parent)

比如說，如果想要在目前面版的最下方加入一個按鈕：

先在左方的元件找到Button，點擊它。

再將滑鼠移到中間的預覽區的下方偏左處，如下圖



在移動的時候，畫面上的虛線方塊是元件位置的預覽，重點是上方會有一區塊即時顯示元件的屬性提示，上圖代表在若此時按下滑鼠決定放置後，會為元件設定屬性為：

對齊面版(Parent)的左邊界(alignParentLeft)

對齊面版的下緣(alignParentBottom)，並設定邊界距離為57個點(margin=57dp)

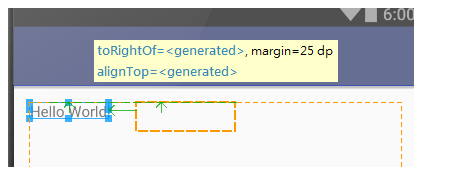
本例將在XML中插入Button元件定義，如下：

對齊對象為元件

假設想在目前的文字的右方放入一個輸入方塊時，

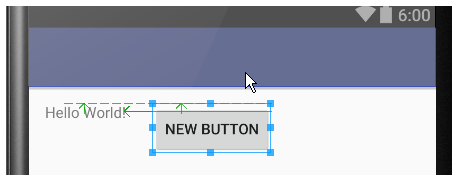
先在左方的元件找到Button，點擊它。

再將滑鼠移到中間的預覽區中的文字方塊右方，如下圖：



在移動的時候，虛擬方塊元件的週圍會出現綠色箭頭，箭頭指向對齊的對象，上方會有一區塊即時顯示元件的屬性提示。上圖的屬性提示意指，在對象的右方(toRightOf)，與其邊界距離25點，且元件上緣對齊(alignTop)。

最後點擊滑鼠後，元件即放置完成，如下圖：



本例將在XML中插入Button元件定義，如下：



元件相對位置屬性有幾個常用的屬性，對象為父元件(容器)的相關屬性:

android:layout\_alignParentTop：若為true，對齊所在容器的頂邊

android:layout\_alignParentBottom：若為true，對齊所在容器的下緣

android:layout\_alignParentLeft：若為true，對齊所在容器的左邊緣

android:layout\_alignParentRight：若為true，對齊所在容器的右邊緣

android:layout\_alignParentCenter：若為true，元件放在容器的中央

當對象為一般元件時的相關屬性:

android:layout\_above：元件放在對象之上

android:layout\_below：元件放在對象之下

android:layout\_toLeftOf：元件的右邊緣對齊對象的左邊緣

android:layout\_toRightOf：元件的左邊緣對齊對象的右邊緣

android:layout\_alignTop：元件的上緣對齊對象的上緣

android:layout\_alignBottom：元件的下緣對齊對象的下緣

android:layout\_alignLeft：元件的左邊緣對齊對象的左邊緣

android:layout\_alignRight：元件的右邊緣對齊對象的右邊緣

android:layout\_centerHorizontal：元件水平置中

android:layout\_centerVertical：元件垂直置中

相關文章:

[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/)

[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/)

[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/)

[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/)

[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/)

元件的屬性，高度、寬度、對齊與權重分配

LinearLayout

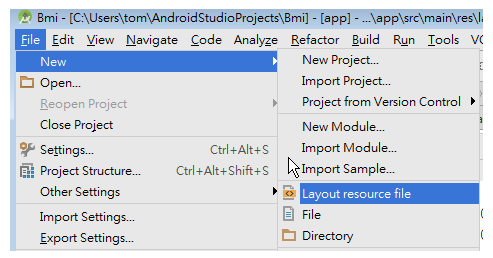
就字面意義來說LinearLayout可稱為線性Layout，但「流水式Layout」比較合適，在LinearLayout內的元件皆以水平(horizontal)或垂直(vertical)排列，在使用之前，可先定義其「orientation」屬性，可設定值為「horizontal」水平或「vertical」垂直。

接下來在專案中，新增一個版面配置(Layout)檔，當作是練習的標的。

Step 1 . 先點擊專案的「app」，或確認app是被選取的狀態，如下圖：



Step 2 . File/New/Android Resource File



Step 3 . 版面檔案名稱

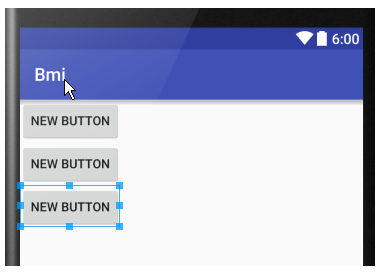
輸入檔名(File name)「activity\_main2.xml」，在第二個下拉選項改選「Layout」代表將產生新的版面配置檔案，預設是使用LinearLayout，完成後按下「OK」，即可產生新的版面檔案，如下圖。





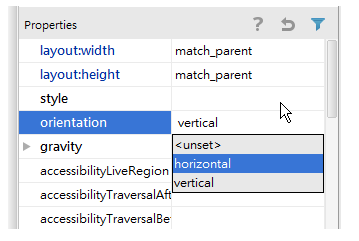
新的版面配置檔案activity\_main2.xml預設使用LinearLayout，且設定為垂直排列，請在版面中加入三個Button按鈕元件，可觀察其配置結果，如下圖：

垂直流水版面

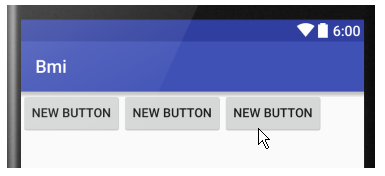


水平流水版面

接著將LinearLayout的排列方式改為水平，先選擇畫面中的LinearLayout(可在預覽區中點擊，或在右上方的元件樹中選擇)，再到右下方的屬性區中找到「orientation」屬性，由「vertical」更改為「horizontal」水平，如下圖：



預覽圖將更改為水平流水版面，如下圖：

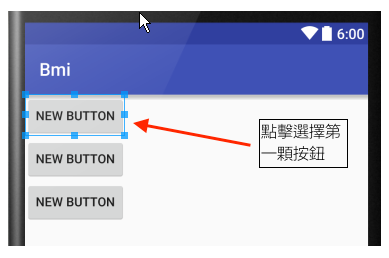


元件的高度與寬度

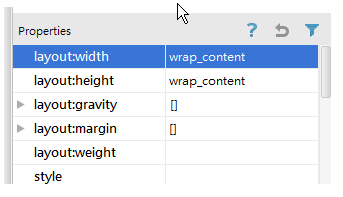
Android的所有元件都具備兩個屬性，也就是元件的高與寬，屬性名稱為「layout:height」與「layout:width」，許多人直覺想要以螢幕的畫素為元件訂定個別的高度與寬度，其實，為元件訂定固定的高與寬是不適合的，因為Android手機產品非常多元，有的手機的解析度是480×720，有的是640×960、640×1136、1080×1920或甚至1440×2560，如此眾多的差異性。假設我們的APP訂的元件大小是固定的，那是無法滿足所有的螢幕需求的。因此，Android建議開發者在訂定元件大小時，若希望只占用元件內容資料時，使用「wrap\_content」值，當想要元件占滿所在區域可用空間時，使用「match\_parent」值，筆者使用上節的LinearLayout垂直配置的範例，說明如下。

符合元件內容物－wrap\_content

意義是元件只占可顯示其內容物的大小，請選擇第一顆按鈕，如下圖：



觀察右下角屬性區，其layout:width寬度屬性與layout:height高度屬性都是「wrap\_content」代表元件的高與度都只占必要的空間，屬性如下圖：

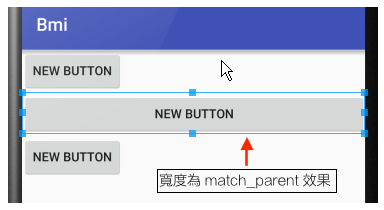


以元件所在的容器為主－match\_parent

意義是所在容器有多大就占多大，繼續使用同一個範例，為第二個按鈕元件設定其寬度屬性為「match\_parent」，如下：



結果如下圖：

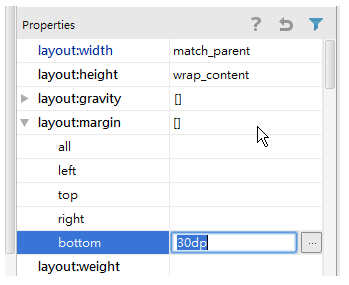


第二顆按鈕雖然內容文字不需要那麼寬的空間，但其元件寬度仍占滿所在的LinearLayout區域。

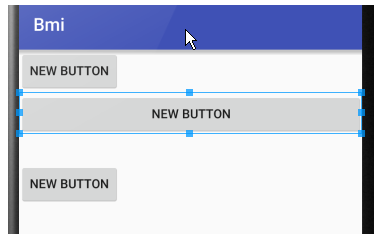
元件的間距－margin

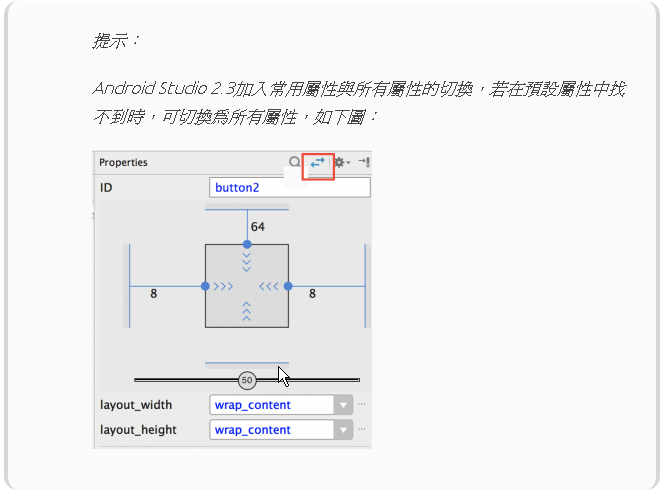
每一個元件都可指定其上下左右與其他元件的距離，稱之為「Margin」間距，「android:layout\_margin」為其屬性名稱，可以統一設定四個方向的間距，亦可個別設定。

假設希望本例的第二顆按鈕與下方第三顆按鈕有30點的間距，請先選擇第二顆按鈕，在其屬性表中找到「layout:margin」屬性後展開，可設定bottom值為「30dp」，如下圖：



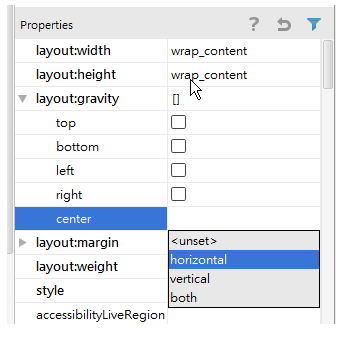
輸入完後按下Enter鍵，即完成設定，預覽圖如下：



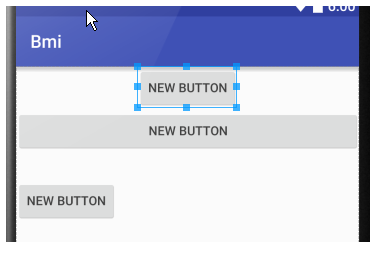


元件的對齊

為元件設定「layout:gravity」屬性，可決定元件在容器中向那個方位靠攏，請選擇第一顆按鈕，找到屬性表中的layout:gravity屬性後展開，在center子屬性中選擇「horizontal」，代表本按鈕對齊為水平置中，如下圖：

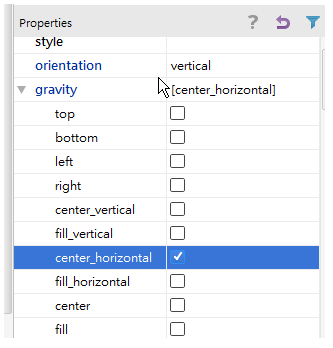


預覽圖如下，第一顆按鈕的layout:gravity屬性在設定為水平置中後：

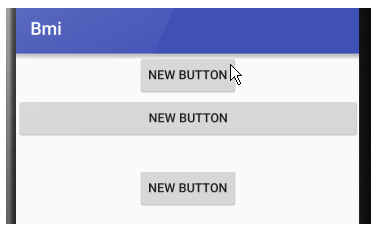


另一個與layout:gravity屬性很像的屬性為「gravity」，假如在一個版面中有很多個元件，想要每個元件都以同一種方式對齊，要一個個去設定實在太繁複了，「gravity」屬性提供給版面配置或容器統一設定裏面元件的對齊方式，只要設定版面即可。

請選擇範例的LinearLayout，找到屬性表中的gravity屬性後展開，勾選「center\_horizontal」如下圖：



預覽效果如下，所有在LinearLayout中的元件都水平置中了：



LinearLayout中的權重weight

當我們在LinearLayout中放入多個元件時，可利用屬性「weight」權重，為元件分配不同的權重值，其預設值為0。透過分別設定在LinearLayout內的元件的權值值，可依權重分配元件可以占用的空間，例如一個水平的LinearLayout人的三個Button，權值為分別為1,2,3時，其配置結果如下圖：



原始碼如下：

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="horizontal">

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="BUTTON1" />

<Button

android:id="@+id/button2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="2"

android:text="BUTTON2" />

<Button

android:id="@+id/button3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="3"

android:text="BUTTON3" />

</LinearLayout>

那如果需要每個元件平均分配，占據整個Layout的區域呢? 可將每個元件的權重值設定為非零的同一個值，如1.0或3.0都可以，平均分配的結果預覽如下圖，三個Button元件的權重值都是1.0：



相關文章:

[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/)

[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/)

[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/)

[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/)

[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/)

Layout新成員ConstraintLayout，組織要扁平化，Layout也要扁平化?

Android的版面(layout)設計一直是Google想要精進的，在Android Studio 2.2始推出新的版面編輯器(layout editor)可以說是為了配合Layout的新成員ConstraintLayout所設計的，原本在iOS中功能非常強大的AutoLayout的設計理念是讓畫面元件的階層規劃能夠盡量扁平化，Android也看到階層扁平所帶來的效能成果，推出類似原RelativeLayout功能的ConstraintLayout元件，首次登場的ConstraintLayout元件將帶來更豐富的畫面設計體驗。

新成員ConstraintLayout

在之前的版面設計，大都使用LinearLayout與RelativeLayout能夠實作出大部份的版面，但Layout層級太多會造成設計更動上的困擾，取用元件時也較耗費資源。因此，如何能將複雜的版面設計的層級較為扁平化呢? 組織要扁平化，Layout也要扁平化才有效率。Android正式推出ConstraintLayout元件，它就是為了這個目的而誕生的。

更新SDK Tools

ConstraintLayout以外加的方式，請打開SDK Manager，在SDK Tools中勾選ConstraintLayout for Android與Solver for ConstraintLayout，並安裝，如下圖：



加入support-constraint類別庫

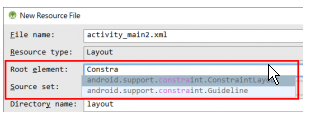
ConstraintLayout元件是外加的，它可支援到Android 2.3的手機，使用前必須先導入support.constraint類別庫，請開啟Gradle Scripts中的app模組層級的build.gradle，在dependencies區塊中加入:



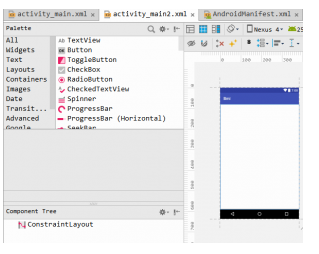
上述使用的是1.0.1的版本，未來新版或正式版推出時可再依其版號更換。完成後記得要Sync project with gradle files。

新增一個layout檔

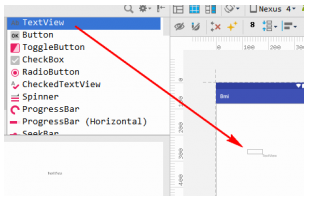
加入類別庫後，先建立一個以ConstraintLayout為根元素的layout檔，點擊專案中的app模組後，功能表File/New/Android resource file，為這個layout訂定名稱為activity\_login2.xml，目的是設計一個登入畫面，再將Resource type下拉更換為Layout後，在Root Element中鍵入ConstraintLayout的部份名稱，如下:



立即會自動出現下拉選單，讓我們選擇android.support.constraint.ConstraintLayout類別，完成後如下:



產生layout檔後的畫面如下，左方是元件選擇區，新版在點擊元件時與舊版點一下後再到畫面中點一下不同，要以按住元件後拖拉的方式，如下圖:

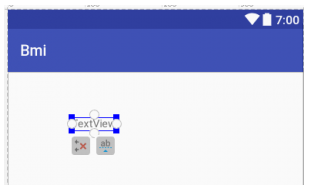


什麼是Constraint?

Constraint的中文翻譯是限制、條件、約束，由劍橋英語字典對它的解釋為「something that controls what you do by keeping you within particular limits」，意思是一些讓你能夠控制的東西。假如我們想要一個元件依照我們的意思配置在版面中的特定位置，就應該為它加上一些控制條件，而這些限制條件就稱為Constraints。

一個無任何Constraint的元件，它預設會配置在layout的最左上角(0,0)處，如果一個畫面中有多個元件都未設定任何Constraint時，它們全都會擠在畫面的左上角。

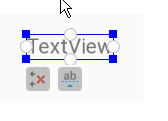
請在左方元件區點擊TextView元件並拖拉到畫面中，任何一個位置，結果如下圖:



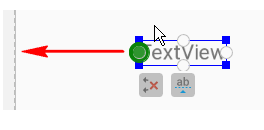
當一個元件被放到ConstraintLayout時，預設會先放置在滑鼠放開的位置，那沒任何Constraint的元件不是應該在畫面的左上方嗎? 這是為了方便設計者，讓設計者能再為它設定Constraints，但如果在設計時都未為元件設定Constraint，執行時元件還是會擠到畫面左上方的。

為元件設定Constraints

ConstraintLayout和RelativeLayout相似，元件應找一個對象對齊，對象可以是parent容器，但功能更有彈性。當點擊選取一個元件時，元件的四週會出現可調的圓點，可用來對齊元件的四個邊界，如下圖:



上圖中TextView的文字下緣有一個橢圓水平的長條，也是用來對齊內容物下緣(Baseline Constraint)。而四個角落的四方型則可改變元件的大小。同樣的，元件的Constraints設定的對象可以是另一個元件，或是容器本身，也就是ConstraintLayout。請按住左邊圓點後拉動設定第一個Constraint到容器的左邊，如下圖:



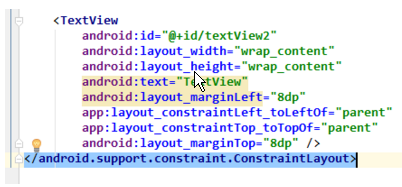
此時已為TextView元件設定一個向左邊的Constraint，向parent左邊對齊並距離8dp，原始碼如下:



ConstraintLayout中的元件規定一個元件至少要有兩個Constraints，分別是水平與垂直方向的Constraints，因此如上圖箭頭指處，這個TextView並未符合規定，所以出現錯誤提示訊息。請再拉動TextView上方的圓點，向容器的上緣靠近，再設定一個垂直方向的Constraint，如下圖:



由於元件現在擁有一個水平與一個垂直的Constraints，已符合最低限度，原始碼也不再有錯誤訊息，如下:



設定一個簡單的元件並不足以展示ConstraintLayout的特色，但算是起個頭，而且登入畫面的設計也不需要用牛刀，未來有機會再以一些特別的畫面配置為例，以展現新成員的能耐。

相關文章:

[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/)

[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/)

[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/)

[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/)

[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/)

上戰場了，寫個Bmi APP吧，專案Layout與元件設計

Bmi專案功能設計

現在終於要實作計算BMI值的應用程式了，一個人的BMI指數的計算公式為：

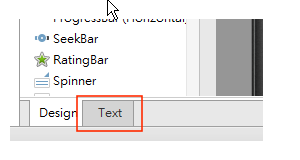
BMI指數 ＝ 體重（kg）/ 身高（m）2

Bmi專案的MainActivity畫面應該有兩個文字輸入元件(EditText)，讓使用者可以輸入體重與身高，另外要有個Button元件，供使用者按下後計算BMI指數。

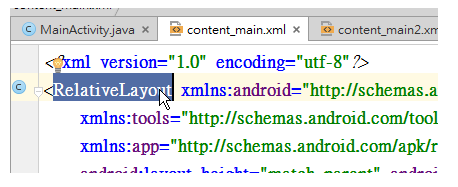
Bmi畫面設計－更換版面配置

MainActivity使用的版面配置為「res/layout/activity\_main.xml」，在這個畫面中會有兩個供使用者輸入的文字方塊與一個計算按鈕，適合的版面配置為LinearLayout，但自動產生的畫面中使用的是RelativeLayout，將其更改為LinearLayout的方式如下：

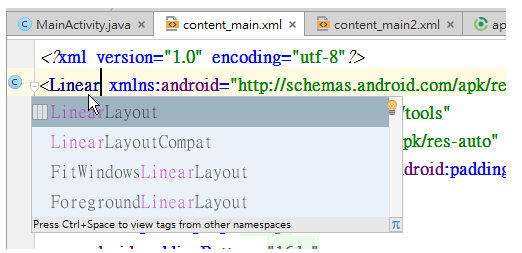
Step 1  開啟xml並切換至原始碼原始碼如下圖：



Step 2  選取RelativeLayout元素名稱，如下：



Step 3  先輸入Linear(L大寫)，還未完全打完時就會出現符合的下拉選單，如下圖：

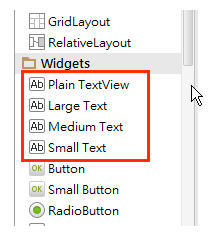


當LinearLayout出現在下拉選單中的第一個時，按下Enter鍵，或使用上下鍵移動再按下Enter，Android Studio會自動更改元素的開始與結尾為LinearLayout。

接著切換回「Design」預覽畫面，選擇LinearLayout後，設定其orientation屬性為vertical垂直排列，再將畫面所有元件先刪除，選擇元件後再按下「Delete鍵」刪除。

新增元件

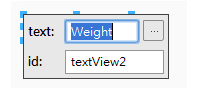
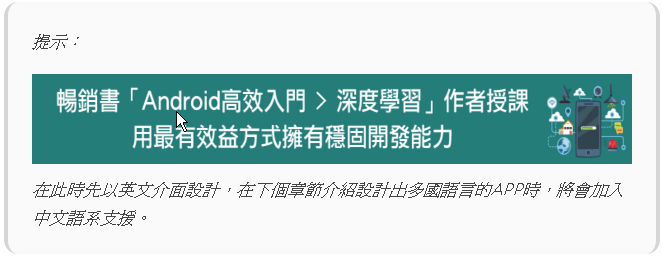
在Android中，文字方塊的類別為「TextView」，在左方元件區找到「Widgets」類別，在此類別下是常用的元件，如下圖：



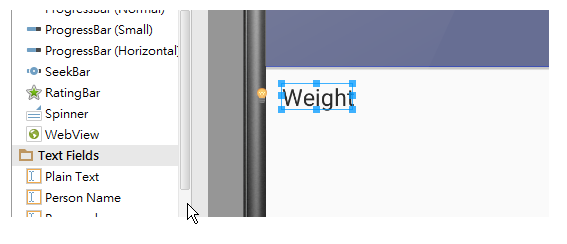
圖中的前四個都是文字方塊，只是字的大小不同，請點擊上圖中的「Large Text」並放到預覽圖中的版面中，結果如下圖：



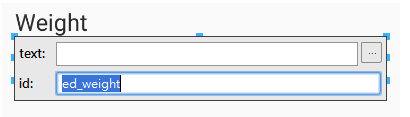
連擊TextView元件可快速設定文字，也就是text屬性，輸入「Weight」代表體重，如下圖：

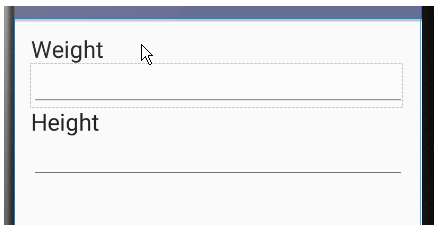
輸入方塊的類別名稱為「EditText」，在左方找到「Text Fields類別」，點擊「Plain Text」元件，並放到版面中以新增一個輸入方塊，如下圖：



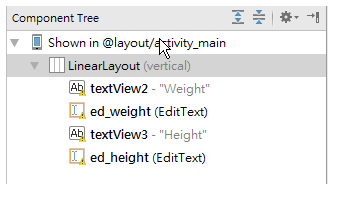
因為未來程式需要取得使用者在此EditText內鍵入的資料，因此先為其設計id屬性為「ed\_weight」，請連擊輸入方塊後，在快速屬性設定對話框中設定id值，如下圖：



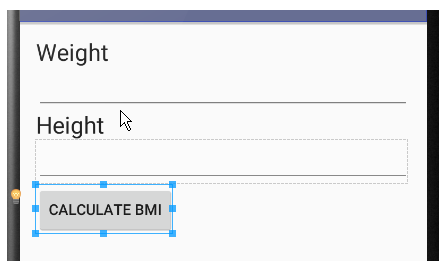
接著請依序新增身高的TextView與EditText元件，並為身高的輸入方塊設定id屬性為「ed\_height」，結果如下：



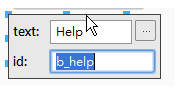
右上方的元件樹中的兩個EditText具有ed\_weight與ed\_height兩個id值，如下圖所示，：



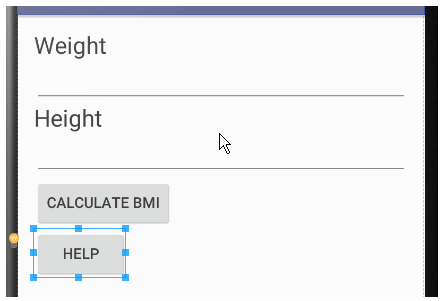
最後在版面加入Button元件，連擊Button元件設定text屬性為「Calculate BMI」，如下圖：



最後，再多加入一個說明按鈕，能讓使用者初次使用時，顯示用法介紹，再為這個說明按鈕多設定一個id屬性為「b\_help」，如下圖：



最後畫面設計完成如下圖：



在MainActivity中取得畫面元件

開啟MainActivity.java，在onCreate方法中的setContentView語法的下一行，新增以下程式碼：

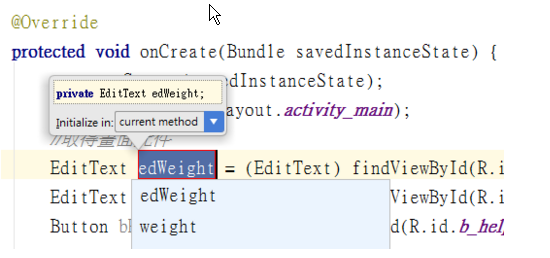


Activity提供的findViewById方法，可利用先前在畫面上為元件設定的id值，取得元件，回傳的型態為View，再經過轉型後可得到元件原本樣貌（Java語言的多型應用）。畫面上的Help按鈕，也使用同樣方法取得bHelp物件。



區域變數提昇為屬性

可是，上述的「edWeight」、「edHeight」與「bHelp」三個物件都是在onCreate方法中宣告的區域變數，在onCreate方法外是無法被存取的，應該將這三個變數提昇為類別的屬性。Android Studio提供快速鍵「Refactor」重構功能，請將游標放在edWeight區域變數上，再按下「Ctrl」+「Alt」+「F」Refactor快速鍵，出現選單後按下Enter鍵，如下圖：



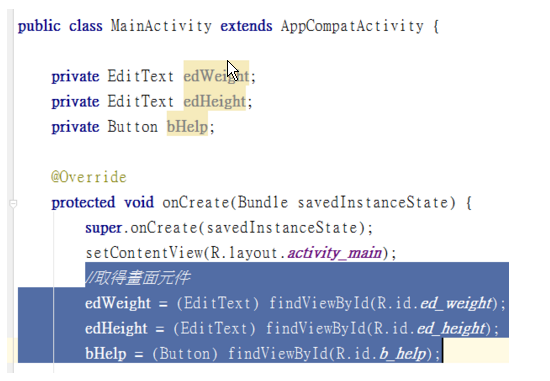
按下Enter鍵後，edWeight即從區域變數提昇為類別的屬性，如下圖：



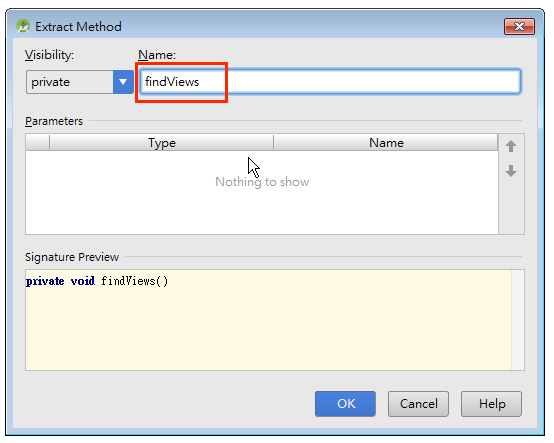
請使用同樣方法將edHeight與bHelp也提昇為屬性。

抽取程式碼成為方法

當需從一個畫面中取得很多個元件時，這些在onCreate方法中的程式碼會略顯雜亂，可設計一個方法，再將這些取得元件的程式碼移到方法中。通常我們會將方法命名為「findViews」，Android Studio提供快速鍵可將程式碼抽出成為方法，請先選取想要抽出的程式碼，如本例在onCreate方法中的的三個findViewById語法，如下圖：



再按下Extract Method抽取程式碼的快速鍵「Ctrl」+「Alt」+「M」，出現對話框後，再輸入方法的名稱「findViews」，如下圖：



最後按下Enter鍵，程式碼很快就完成原本繁複工作，如下：



相關文章:

[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/)

[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/)

[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/)

[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/)

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

Bmi專案功能與事件處理，什麼是onClick屬性?

按鈕的事件處理

使用者與APP中的UI元件互動時，元件會產生各類型的事件(Event)，例如點擊、長按、觸碰等事件，在Android中，這些事件由專門處理的「事件傾聽器（Event Listener）」負責監聽，最常見的是元件被按下事件的「View.ClickListener」，它是View類別內所設計的Java介面，宣告了一個方法：



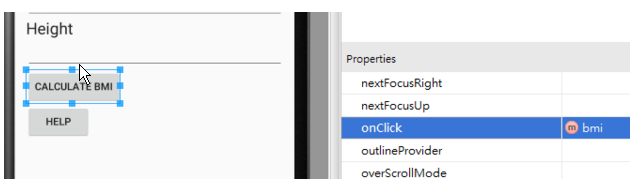
所有實作ClickListener的類別都須一併實作這個方法，先將ClickListener指派到一個特定元件（如Button），未來，當Button被按下時，會自動執行ClickListener內的onClick方法。通常事件傾聽器會以「匿名類別（Anonymouse Inner Class）」方式設計，匿名類別寫法會在本章後續介紹，筆者先談的是使用方便的onClick屬性。

使用onClick屬性

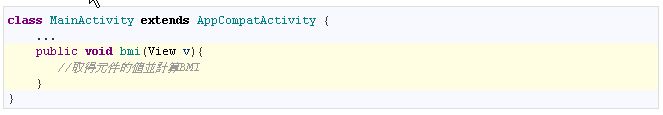
假設一個畫面中有很多個按鈕，每個按鈕都實作一個事件傾聽者去監聽事件是一個很繁複的工作，因此，Android對於Click事件有特殊的對待，可以在元件屬性onClick設定一個方法名稱，再到MainActivity類別中設計一個同名的方法，方法規格如下：



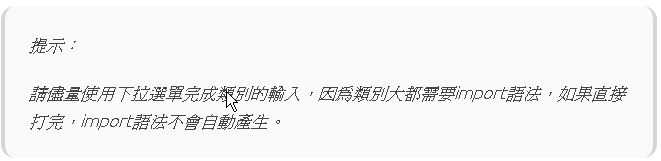
接下來實作範例的Bmi運算按鈕的事件處理，在activity\_main.xml的設計畫面中，選擇「Calculate BMI」按鈕，再到屬性區找到「onclick」屬性，輸入「bmi」，如下圖：



輸入完成後再開啟MainActivity.java，新增一個bmi方法，Java類別的方法應在類別內的第一層新增，如下圖：



在輸入View類別時，未打完時會出現下拉選單，請使用選單完成View類別的輸入，Android Studio會自動加入import語法。未來，當使用者按下此按鈕時，會自動執行bmi方法內的程式碼。



在bmi方法中，設計以下程式碼以計算出BMI值，如下：

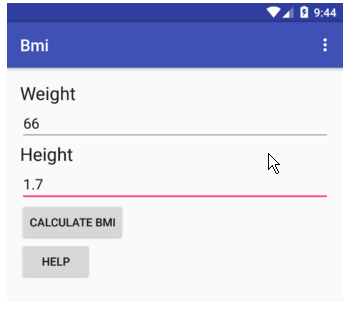


程式碼的2與3行呼叫EditText的getText()方法取得一個Editable的物件，再呼叫其toString方法取得目前輸入方塊中的字串。第4與5行呼叫Float類別的靜態方法parseFloat，將字串轉換為float型態，最後以體重/身高2的公式算出bmi值。

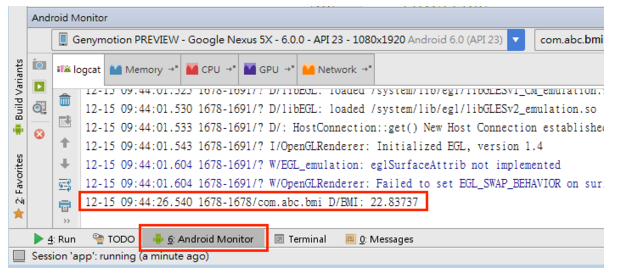
最後的第7行，使用android.util.Log類別，將浮點數bmi轉換為字串後，顯示在系統記錄（LogCat），LogCat是Android開發人員用來除錯的工具，可在執行時顯示資料並驗證程式的正確性。Log類別提供了數個靜態方法，可供產生不同層級的記錄資料，例如代表除錯用的d方法（debug）、資訊的i方法(info)、代表錯誤的e方法(error)等，以英文字的第一個字母為命名方式，可簡化程式碼的長度。這些方法使用上大致相同，以d除錯方法為例，規格如下：



第一個參數代表標籤Tag，為這個訊息設定一個可供辨認的標籤值，方便未來在Log視窗中找到訊息，第二個參數則是想顯示的字串訊息，以上範例的執行結果如下：



按下畫面中的「CALCULATE BMI」按鈕後會自動去執行MainActivity內的bmi方法，計算後得到bmi值並在下方的LogCat視窗中顯示，如下圖：



Tip：

「LogCat」是由Android開發工具所提供，讓開發者能夠觀察系統產生的除錯資訊與記錄。當應用程式出錯拋出例外時，可利用LogCat找到出錯的原因。

使用Log類別可在開發時期提供資料確認、除錯的訊息，但無法將資訊提供給APP使用者，接下來將利用浮動顯示方式告知使用者運算出的BMI值的結果。

相關文章:

[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/)

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

[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/)

[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/)

Bmi專案功能之顯示資訊，包括Toast、AlertDialog

使用浮動顯示－Toast類別

Android提供浮動顯示的元件「android.widget.Toast」類別，使用Toast類別的靜態方法makeText方法可以浮動顯示文字，規格如下：



第一個參數為Context，目前類別MainActivity即是一個Context物件，因為Activity的上二層父類別就是Context，因此本類別MainActivity即是Context的一種，可使用「this」關鍵字代表本類別。第二個參數為「Charsequence」介面，Java的String類別即是其子類別，因此第二個參數即是想要顯示的字串值。最後，第三個參數為浮動顯示的時間長度，1代表有較長的時間停留在畫面上（大概三秒），而0代表較短時間，使用範例如下：



請注意第三個參數，不要直接使用1或0，應使用Toast類別中為其所定義的LENGTH\_LONG或LENGTH\_SHORT代表長與短時間的常數，雖然程式碼會比起直接給１或0來得長，但能大大增加程式在未來的可讀性。最後，別忘了呼叫Toast類別的show方法才會在畫面下方顯示浮動資訊，如下圖：

在本例先將bmi值（float型態）轉換為String後，即可浮動顯示BMI值，程式碼如下：



執行結果如下圖：



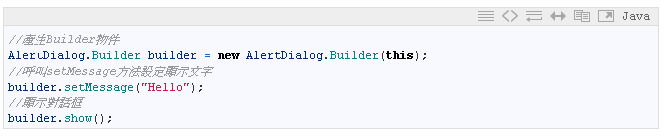
使用浮動顯示主要目的是盡量不影響使用者，重要性不高的資訊可以使用Toast類別顯示，但如果有重要資訊需告知使用者，則可使用AlertDialog對話框。

使用對話框－AlertDialog類別

對話框是很常見的視覺元件，可用來提示、告知使用者必要的訊息，主要由Android中的「android.app.AlertDialog」類別提供相關的功能。由於對話框的種類繁多，有無標題、圖示、訊息、按鈕等各類需求，要提供如此繁複的建構子是有一定難度的，就算有，開發人員光是要記住這麼多的建構子參數也是非常頭痛的，因此，AlertDialog是以「Builder」設計模式提供使用者產生合適的物件。

AlertDialog.Builder類別

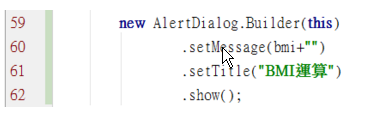
「AlertDialog.Builder」是一個公開的內部類別，它就如同一個「製造機」，先產生AlertDialog.Builder物件後，再呼叫Builder提供的方法配置想要的功能與資訊，完成後再呼叫show()方法產生對話框並顯示在畫面上。產生Builder物件時，建構子參數需要Context類別，可使用this關鍵字將MainActivity傳入，範例程式碼如下：



上述的builder物件是為特定的對話框服務，對話框顯示完成後，其實builder物件不會再被使用。

簡化寫法

在開發時，對於不需要重複使用的物件，可使用匿名方式(Anonymouse)建立Builder物件，設定完成後即顯示，而不需要為物件命名一個物件，使用方式如下：



上述程式碼的第59行產生一個Builder物件，接著馬上在第60、61行呼叫設定對話框資訊的setMessage與setTitle方法設定內文與標題，使用「bmi+ “”」是利用加號其中一個運算為String時，會將另一個運算元也轉換為String的特性，目的是稍微簡化程式碼，最後62行直接將已設定完成的對話框物件顯示在畫面上。

AlertDialog.Builder的方法

常用的方法如下表：

AlertDialog create()：依Builder目前配置產生一個對話框物件。

Builder setIcon(int resId)：設定圖示，使用R.drawable.X資源記錄。

Builder setMessage(Charsequence text)：設定對話框的內文字串。

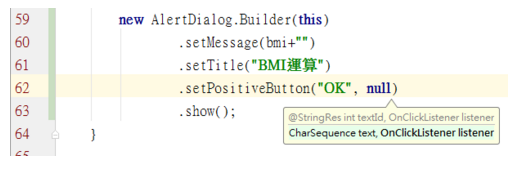
Builder setTitle(Charsequence text)：設定對話框的標題字串。

void show()：產生對話框物件並顯示在畫面中。

對話框按鈕

Android的AlertDialog能夠配置按鈕，預設按鈕有三種，一為「正向按鈕」，例如確定(OK)、是(Yes)，二為「中性按鈕」，例如取消(Cancel)，第三種為「否定按鈕」，例如否(No)。設定這些按鈕的方法如下：

正向按鈕方法為setPositiveButton，使用如下：



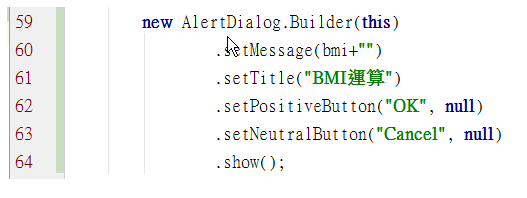
提示：

當游標放在方法的括號內時，按下「Ctrl+P」可顯示方法的規格，可得知傳入參數的資料型態。

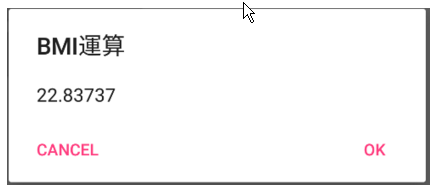
第一個參數為String，代表按鈕中顯示的文字資料，第二個參數是當按鈕被使用者按下時的傾聽器物件，上圖第62行中給的是null值，代表按下OK按鈕後不做任何事，會直接關閉對話框，執行結果對話框如下圖：



中性按鈕方法為setNeuButton，使用如下：



執行結果對話框如下圖：



各個版本的Android系統對於正向、中性、反向按鈕的排列方式會有不同，因此，針對按鈕的意義去使用合適的方法設定，讓Android系統決定按鈕應該的位置即可。

相關文章:

[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/)

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

[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/)

[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/)

[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/)

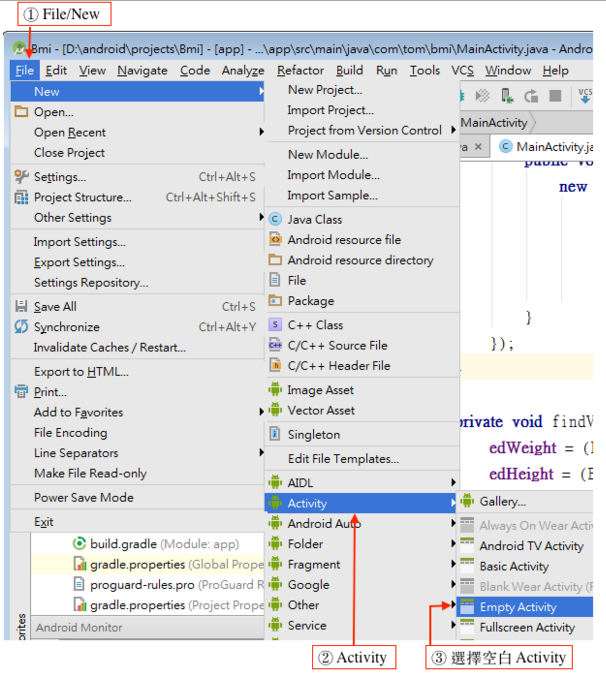
建立第二個Activity，自動產生了那些東西?

第二個Activity

本文將介紹如何建立APP的第二個Activity，並深入瞭解一個Activity的各項要素，介紹如何利用Intent意圖讓畫面從一個Acitivty轉換到第二個Activity，在轉換的過程中使用Intent傳遞簡單與複雜資料，在最後會介紹一個Activity的生命週期。

產生第二個Activity(方法、產生的檔案)

先用滑鼠在左方的專案區點擊選擇專案名稱(app)，再使用功能表中的File/New/Activity/Empty Activity，產生一個空白的Activity，如下圖：



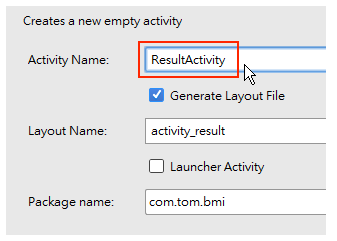
提示：  
記得先點一下專案app，之後在File/New時才會出現Activity的選項。

更換合適的名稱

新的Activity的預設名稱不大符合我們的用途，因此需要修改為符合新Activity功能的名稱，預設如下：



第二個Activity的用途是顯示BMI運算的結果，因此可命名為「ResultActivity」，以後光看類別名稱就能得知其用途或功能，請修改第一個「Activity Name」的值為「ResultActivity」，大小寫請符合Java語言類別的慣例，如下圖：



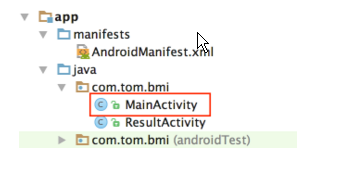
讀者發現了嗎，當我們修改Activity Name之後，以下的Layout檔名自動更改為合適的名稱了，這是Android Studio貼心的地方。

一個Activity的組成要素

新增Activity後，總共會產生三個檔案，說明如下：

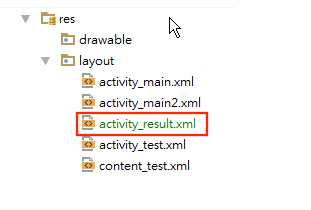
① Activity(Java類別)

自動產生在專案package下的ResultActivity.java類別檔，如下圖：



② Layout(版面配置檔，XML格式)

ResultActivity類別的畫面配置檔，檔名為activity\_result.xml，如下圖：



另外還會在AndroidManifest.xml中為我們加入一個Activity元素，在一個APP中，所有使用到的Activity都必須定義在AndroidManifest.xml中，如下圖：



上圖所框起來的，即是Android Studio自動為我們加入的Activity元素。

提示：  
在幾年前Android剛起步時，開發工具沒那麼方便，這些自動產生的類別、XML等都是要一個個手動產生，可見開發工具對我們的幫助有多重要。但也不要忽略了基礎知識，這些都是一個Activity的組成要件。

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/)

[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/)

[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/)

[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/)

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

使用Intent轉換Activity並傳遞資料，什麼是Intent意圖?

使用Intent轉換Activity

要如何由原本的第一個MainActivity轉換到我們剛產生的ResultActivity呢？答案是使用Intent，也就是「意圖」，Intent代表使用者與應用程式的互動，互動通常會產生變化，例如按下一個圖示後進行撥出電話，或者按下一個按鈕後轉換到另一個畫面(也就是另一個Activity)。使用android.content.Intent類別可以達到這些功能，Intent類別的建構子有很多種，其中常用來轉換Activity的建構子規格如下：



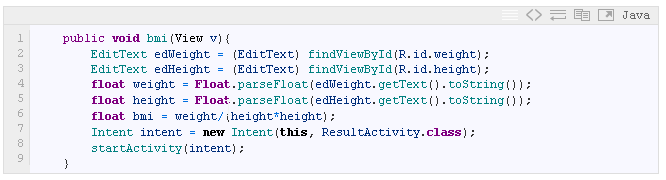
這個建構子的第一個參數是Context，代表從那一個Activity出發，如果在MainActivity中產生Intent，那就是代表MainActivity本身，可使用「this」關鍵字，第二個參數則需給予轉換目的地的類別，本例為ResultActivity，應該使用「ResultActivity.class」，範例程式碼如下：



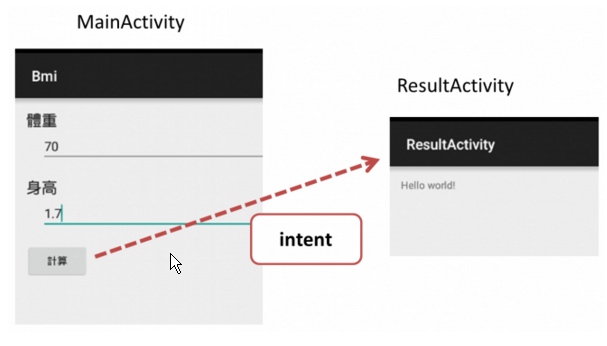
上例程式碼產生一個名稱為小寫「intent」的Intent物件，但此時intent並不會有任何作用，它只是個在記憶體中的物件罷了，在一個Activity中可以使用startActivity方法，將一個intent物件發送至Android系統中，由Android系統判別，判別後由系統將我們的ResultActivity顯示在畫面上，範例程式碼如下：



請在MainActivity的bmi方法中實作以上程式碼，在運算並得到bmi值後，產生Intent物件，並呼叫startActivity方法，部份程式碼如下：



執行結果，當輸入體重與身高並按下計算按鈕後，會轉換至ResultActivity，如下圖：



目前新的ResultActivity仍然未實作BMI顯示的功能，接下來我們將一一完成必須的程式設計工作。

使用Intent傳遞資料

在MainActivity中所計算得到的bmi值要如何在轉換至ResultActivity時，一併將bmi資料傳遞過去呢？答案是使用Intent，可以在Intent物件中夾帶資料，在轉換後再取出資料後使用，這種在Intent中的資料稱為「Extra」額外附帶的資料。

簡單資料

如果欲傳遞的值很簡單，就像本例的浮點數(float)bmi值，可直接使用Intent類別所提供的putExtra方法，在轉換Activity之前，將資料（本例為bmi）放進去Intent物件中，如：



程式碼的第二行，在執行startActivity方法之前，將bmi值放進intent物件中，必須在放入時指定一個名稱字串，也就是為這個資料命名一個標籤如「BMI\_EXTRA」，未來使用這個標籤就能取得資料了。在本例中，放入資料的程式碼實作在MainActivity中，也就是計算bmi且已產生intent物件後。

而在轉換Activity後的目的地（本例為ResultActivity），可在onCreate方法中，取得在前一個Activity己放入的資料。由於資料是放在Intent中，因此必須先呼叫Activity所提供的方法getIntent()得到Intent物件，如下：



再使用Intent類別的getXXExtra方法取得其中所附帶的資料，如本例之前放入的是float型態的資料，在此就應使用getFloatExtra方法，如下：

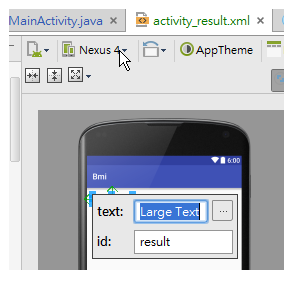


這類取得附檔資料的方法的第一個參數為字串，代表資料的標籤，而第二個參數是使用該標籤而取不到資料時的預設值，本例定義為0。如果資料是整數int時，使用getIntExtra方法，是String資料時，就使用getStringExtra方法。Intent類別提供了以下幾個常用取得資料的方法：

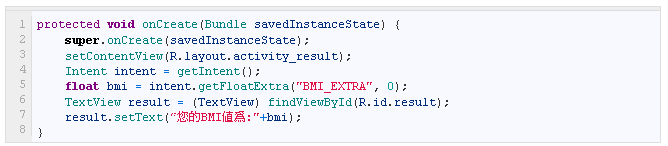


顯示在畫面上

取得前一個Activity傳遞過來的資料後，現在可以將資料送到畫面上，請讀者打開ResultActivity的畫面配置檔「res/layout/activity\_result.xml」，設計畫面中加入一個TextView元件(Large Text)，再設定TextView元件的id，連擊TextView元件，在id欄位中更改為「result」(或點擊元件後，在右邊的屬性視窗)，如下圖：



修改後按下Enter鍵即完成設定，回到ResultActivity中的onCreate方法，加入程式碼，先取得畫面中的TextView元件，再將bmi值設定在TextView元件中，如下程式碼：



執行結果如下圖，在ResultActivity中顯示MainActivity所運算的BMI結果：



複雜的資料－使用Bundle類別

如果有多種類的資料需要傳遞，可使用Bundle類別，Bundle類別類似一個袋子，先將所有要傳遞的資料放在一個袋子中，最後再將袋子放到Intent中，如下：

先產生一個Bundle物件：



再使用Bundle類別的putXX方法，放入不同型態的資料：



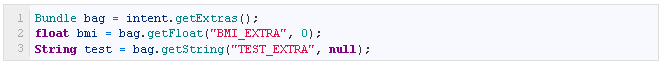
最後，再呼叫Intent類別的putExtras方法，將bag放入：



而在ResultActivity中的onCreate方法中，需先呼叫getIntent()得到Intent物件後，再呼叫getExtras()方法取得Bundle物件，再取得Bundle中的資料，如下：



或是先取得Bundle物件後，再一一取得資料，如下：



提示：  
使用Bundle確實多了一道手續，而在執行的結果上好像沒什麼差別，其實，Bundle如果在程式未來有需求更動時，會帶來較方便的效果。例如，如果之後決定要在傳遞資料前，先將所有資料儲存在資料庫時，就可將Bundle物件傳遞到其他方法去處理，而不用傳遞多個散亂資料。

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/)

[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/)

[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/)

[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/)

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

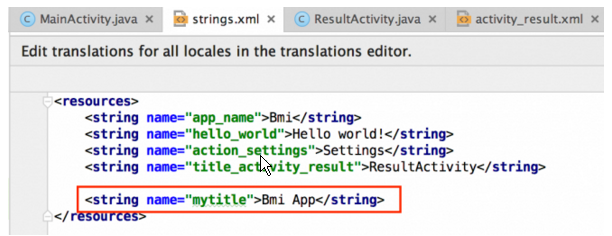
將字串抽取出成為資源，重要的能力

抽取字串成為資源

在畫面中顯示的文字資料經常會在程式中定義為字串，例如上一篇文章傳遞資料時的「BMI\_EXTRA」或對話框按鈕上的文字「OK」或「確認」等字串，未來假如需要更換這些字串，除了耗費時間去找到不同類別內的字串並修改外，在程式設計時也容器發生打錯字而出現bug(臭蟲)。如果可以將字串的定義從Java類別程式碼中抽離，放在資源定義檔中(strings.xml)，日後只需要修改XML資源定義檔「res/values/strings.xml」就好了。

手動加入字串資源

打開專案下的res/values/strings.xml字串資源定義檔，在根元素resources內加入一個string元素，屬性name的值代表該字串的名稱，再將字串值放在string元素中，如下定義一個名稱為「mytitle」的字串值，其字串的內容為「Bmi App」：



程式中使用字串資源

在程式碼中可使用Activity類別所提供的getString方法，快速取得在字串資源檔中的字串，專案中的所有字串資源會歸納在R.string.mytitle，如下程式碼：



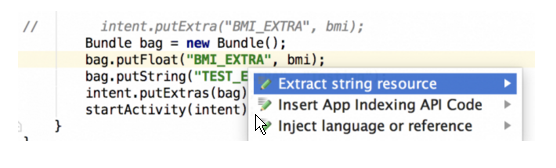
雖然在程式碼中改以字串資源方式取得字串時，程式碼會比較長一點，可是它所帶來的是未來能快速調整與修改的方便性，並可提高程式未來的可維護性。

提示：  
在設計APP的一開始，可以不必每個字串都用資源來表示，因為有些字串在未來不一定會使用，還會經過調整。但在整個APP漸漸成型後，字串使用都較固定了後，就可以開始進行字串定義抽離了。

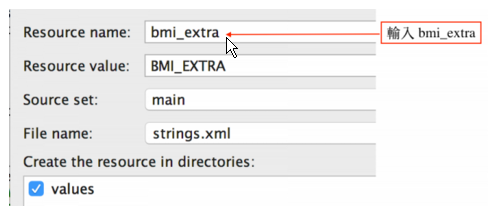
使用Android Studio抽離字串資源

當程式碼中已經使用了許多的字串，一個一個手動抽離成為字串資源不是太耗費時間了嗎? Android Studio能幫你快速的抽離字串。

先將游標停在欲抽離字串中任一個位置(如BMI\_EXTRA字串上)，按下「Alt」+「Enter」組合鍵(Mac為Option+Enter)，會出現「Extract string resource」抽取為字串資源的小對話框，請再按下「Enter」，如下圖：



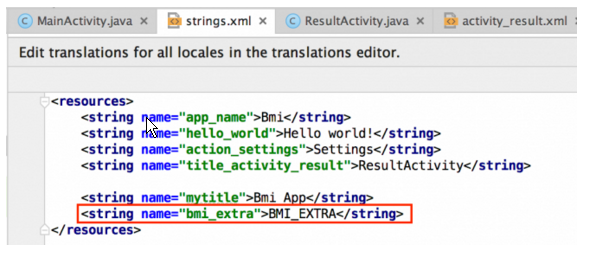
按下Enter後出現「Extract Resource」設定對話框，只需要為這個字串取一個資源名稱，資源名稱的命名慣例是全小寫，若有多個英文字，中間以底線表示，例如「bmi\_extra」，名稱打完後請直接按下Enter完成抽取，如下圖：



原本在程式碼中的字串已自動產生getString(R.string.bmi\_extra)的程式碼，如下圖：



同時字串資源檔中也自動新增一個string元素，如下：



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/)

[清單元件實作－下拉式選單Spinner](https://litotom.com/2016/03/26/%e6%b8%85%e5%96%ae%e5%85%83%e4%bb%b6%e5%af%a6%e4%bd%9c%ef%bc%8d%e4%b8%8b%e6%8b%89%e5%bc%8f%e9%81%b8%e5%96%aespinner/)

[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/)

[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/)

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

Activity的生命週期，它會自動執行那些方法? 方法如何覆寫?

Activity的生命週期(說明、覆寫方法、測試觀察)

Android系統在啟動一個Activity的過程中，有幾個特定的方法會被呼叫，稱為Callback方法，目的是提供設計Activity的人能夠更能靈活控制自己的Activity，本節將介紹幾個情境，與其方法與其被執行的時機。

一般情境

一個Activity啟動後出現在手機螢幕上，之後再由使用者按下返回鍵結束Activity，各個Callback方法執行時機說明如下：

當Activity準備要產生時，先呼叫onCreate方法。

Activity產生後（還未出現在手機螢幕上），呼叫onStart方法。

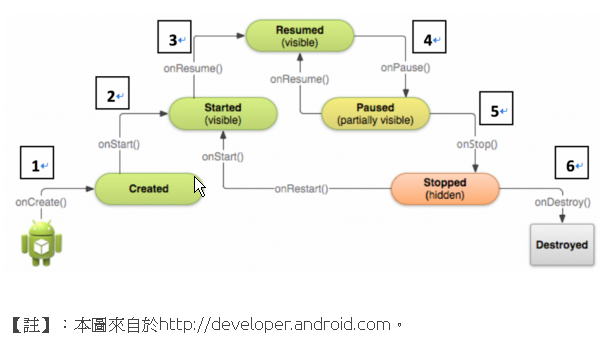
當Activity出現手機上後，呼叫onResume方法。

當使用者按下返回鍵結束Activity時， 先呼叫onPause方法。

當Activity從螢幕上消失時，呼叫onStop方法。

最後完全結束Activity之前，呼叫onDestroy方法。

整個呼叫過程像是金字塔般，從金字塔的左邊Activity啟動開始往右，如下圖：



暫停情境

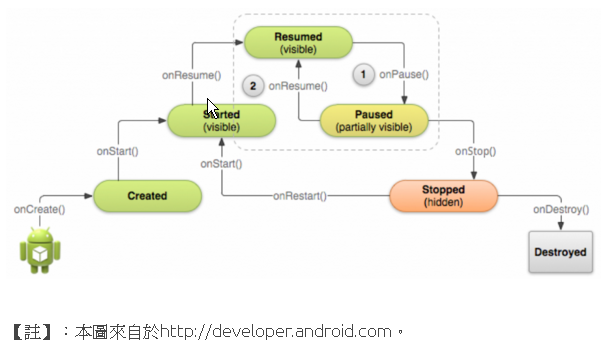
當Activity已經顯示在手機螢幕上了，但這個Activity如果有對話框出現在Activity的前面，此時Activity是無法使用的，稱之為在暫停狀態下，這時候各個Callback方法執行時機說明如下：

（前三個步驟同上小節）

當出現對話框，Activity是部份可見，但無法使用時，呼叫onPause方法。

對話框消失，呼叫onResume方法後，Activity才完全可見。

呼叫過程如下圖標示的1與2：



切換Activity情境

當Activity在手機螢幕時，使用者開啟最近使用的APP清單，並點擊了另一個APP時，在前景的Activity會停止並進入背景，直到使用者再由APP清單中點擊這個Activity後，這個Activity會被重新執行。

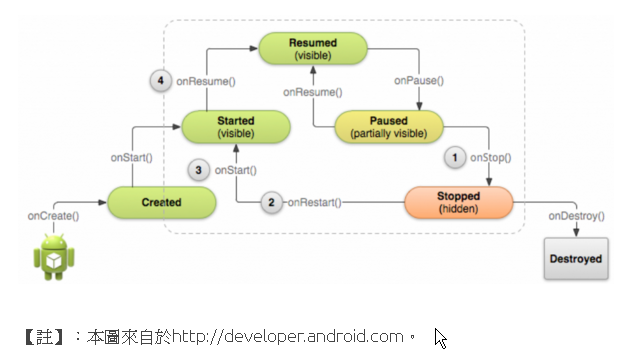
使用者點擊另一個APP執行，讓原本在前景的Activity進入背景前，會先呼叫onPause方法，再呼叫onStop方法後，此時Activity完全進入背景，不在手機畫面上。

當Activity由使用者從最近使用APP清單中被點擊了時，先執行onRestart方法。

之後再執行onStart方法後。

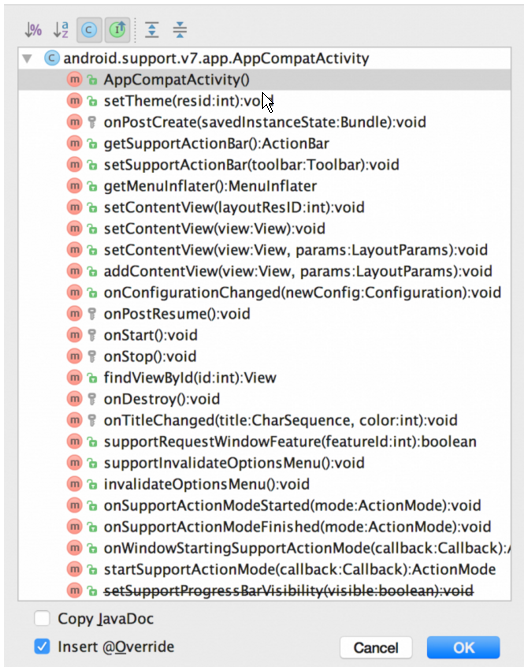
Activity顯示在手機畫面後，再執行onResume方法。

呼叫過程如下圖標示的1到4：

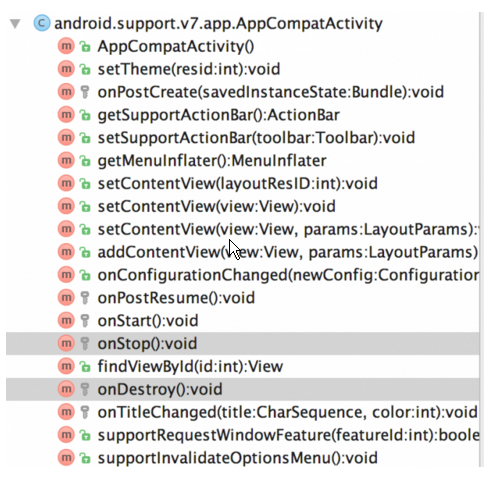
覆寫Callback方法

接下來我們利用MainActivity進行測試並驗證，首先要覆寫(override)父類別的Callback方法並使用Toast類別顯示該方法被執行，所要覆寫的方法有onStart、onResume、onPause、onStop、onRestart與onDestroy。這些方法有些屬於上層父類別，有些屬於更上一層或更多層的類別，因此在選擇時需要找到該方法屬於那層父類別下。

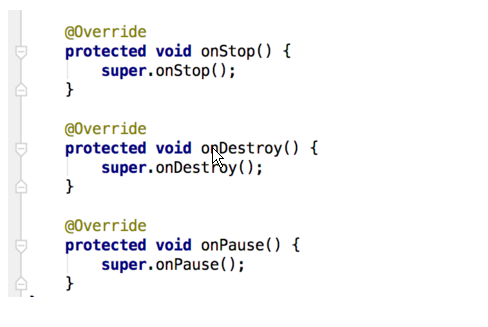
首先將游標移至MainActivity類別內的最後一行，Android Studio覆寫方法的按鍵是「Ctrl」+「O」(不是數字零，而是字母O)，可用Override的第一個字母做為記憶的技巧，按下後會出現MainActivity的所有繼承類別與可覆寫的方法，如下圖：



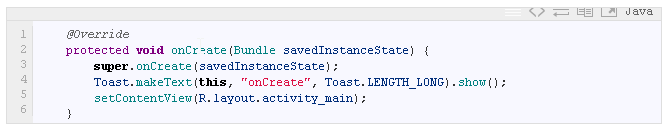
此時請按著「Ctrl」鍵不放，用滑鼠點擊欲覆寫的方法，如onStop與onDestroy方法，只要按著Ctrl鍵去點擊，就可一次選擇多個方法，如下圖：



請再往下拉動捲抽找到onPause、onPause、onStart與onRestart方法都選取，最後按下Enter鍵即可，所選擇的方法都會在程式中覆寫完成，如下圖：



接著在各個方法中使用Toast類別分別顯示該方法的名稱，以便能在執行時得知該方法被執行了，部份程式碼如下：

最後別忙了也在onCreate方法中加入Toast程式碼：

相關文章:

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

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

[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/)

[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/)

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

生命週期的測試情境—使用Bmi專案

測試情境

現在我們利用Bmi專案的執行情境，可以觀察這些Callback方法被執行的順序與時機。

請執行Bmi專案，當MainActivity啟動後，onCreate方法被執行，再執行onStart方法，最後onResume方法被執行。



輸入體重與身高，按下計算按鈕，從MainActivity轉換畫面到ResultActivity時，onPause方法先被執行，之後onStop方法執行了。



在ResultActivity畫面中按下返回鍵，MainActivity被叫回前景時，onRestart方法會被執行，再執行onStart方法，最後再執行onResume方法。



最後在MainActivity畫面中按下返回鍵，結束MainActivity時，onPause方法先被執行，再執行onStop方法，最後執行onDestroy方法。

相關文章:

[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/)

[Android高效入門-Fragment片段的由來與生命週期](https://litotom.com/2016/06/02/android-fragment-lifecycle/)

[使用 Android Studio 開發 Firebase 快速導入 Email and password Authentication 設定並實作登入功能](https://litotom.com/2016/11/21/as_222_firebase_email/)

[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/)

[清單元件實作－清單式功能表](https://litotom.com/2016/03/26/%e6%b8%85%e5%96%ae%e5%85%83%e4%bb%b6%e5%af%a6%e4%bd%9c%ef%bc%8d%e6%b8%85%e5%96%ae%e5%bc%8f%e5%8a%9f%e8%83%bd%e8%a1%a8/)

來證明你會了吧，Atm專案之登入功能實務

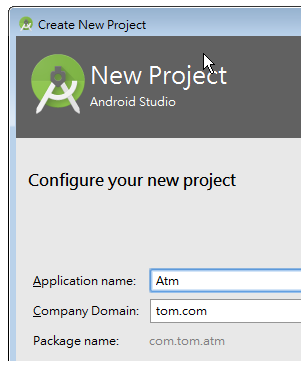
Atm專案之登入功能

利用前面幾篇文章之所學，完成一個網路銀行的「Atm」專案，這個專案將成為之後文章的練習範例。

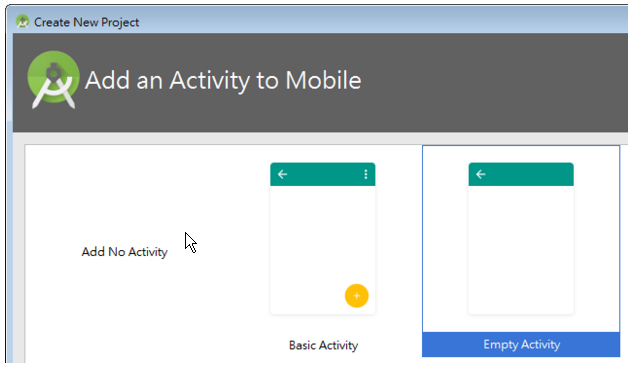
主畫面－MainActivity

一開始Atm專案有兩個Activity，一個為主畫面MainActivity，另一個為登入畫面LoginActivity，一開始設計的是功能主畫面。請新增一個專案(File/New/New Project)，名稱為「Atm」，套件(package)名稱為「com.tom.atm」，使用空白Activity即可(Empty Activity)，如下圖：

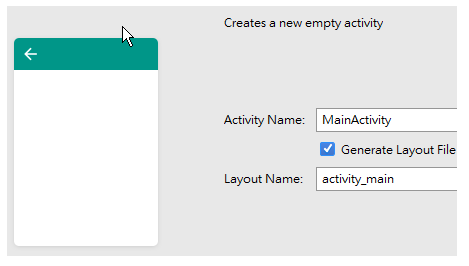
Step1  建立Atm專案



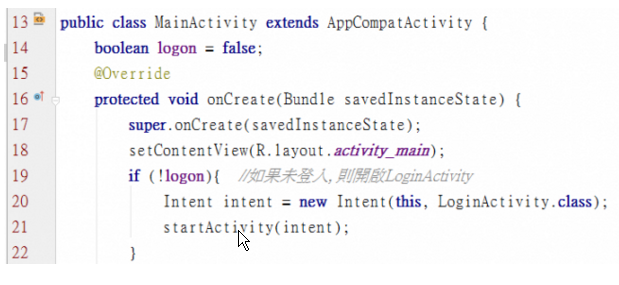
Step2  選擇Empty Activity



Step3  建立活動

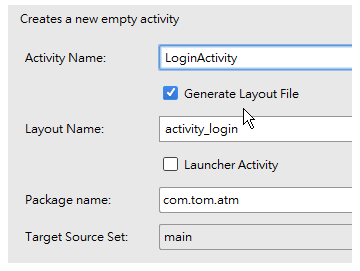


建立完成後，請開啟「MainActivity」，在類別中加入一個布林值屬性「logon」，預設為false假值，在MainActivity的onCreate方法的最後，檢查使用者是否已經登入過，若未登入，則準備開啟LoginActivity，如下圖：

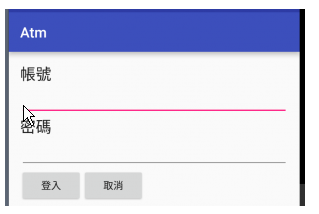


新增登入畫面－LoginActivity

在這個專案中，新增第二個活動，名稱為LoginActivity，目的是提供使用者以帳號、密碼登入，點擊app後，再使用功能表「File/New/Activity/Empty Activity」新增活動的畫面如下：

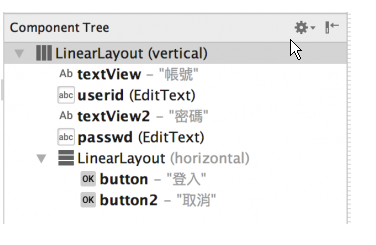


接下來設計登入畫面版面「activity\_login.xml」，設計如下圖的登入畫面，筆者在本例使用LinearLayout，請將LinearLayout的排列方式設定為垂直：



更換為LinearLayout請參考 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/)

接著為兩個輸入方塊(EditText)分別設定id值為「userid」與「passwd」，此畫面的元件結構如下圖：

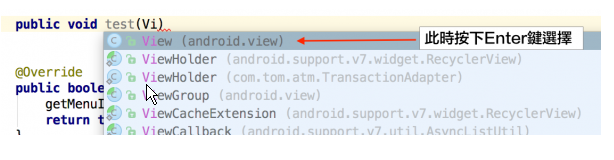


元件的onClick屬性

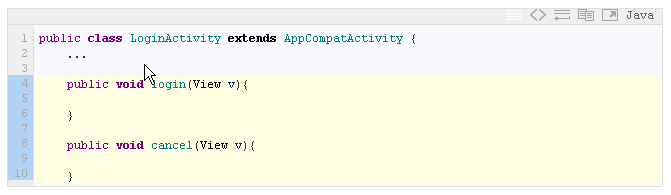
Android的元件有一個「onClick」屬性，只要在onClick中設定方法的名稱，未來執行時只要使用者按下該元件（通常是Button按鈕元件），便會自動執行該方法內的程式碼。因此，設定了onClick屬性後，應馬上到Activity中設計一個同名稱的方法，且方法的傳入值一定要有一個View元件，例如：



View元件的全名為android.view.View，初學者常打完上述方法後仍有錯誤訊息，原因是該Activity未import該類別，最快的解決方式是將游標移到View字上，再按下「Alt+Enter」組合鍵，Android Studio會自動加入import的語法。而最好的方法則是在鍵入指令的當下，在類別名稱還沒打完時出現選單時，按下Enter鍵選擇類別，此時會自動加入import語法，如下圖：



接下來，將進行實作按下兩個按鈕時的程式碼，請為提供使用者輸入帳號與密碼的輸入方塊(EditText)元件設定id，分別為「userid」與「passwd」，最後為登入按鈕的屬性onClick設定方法名稱「login」，再為取消按鈕設定onClick屬性值為「cancel」，再回到LoginActivity類別中設計login與cancel兩個方法，如下圖：



最後完成的activity\_login.xml如下：





登入判斷

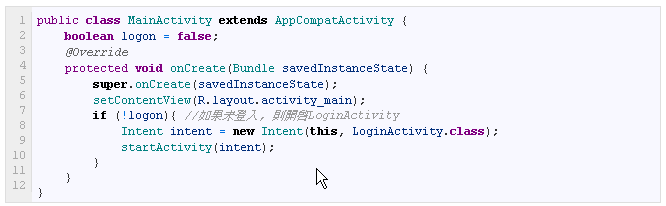
在login方法中實作登入驗證功能，在此時，我們先假裝有一個帳號叫jack，密碼為1234，進行簡單的假驗證，在後續章節介紹到網路程式設計時，再真的將帳號與密碼經過網路伺服器進行驗證。

先以findViewById方法取得兩個EditText元件，再取得字串uid與pw，再判斷帳號與密碼，先以if方式設計：



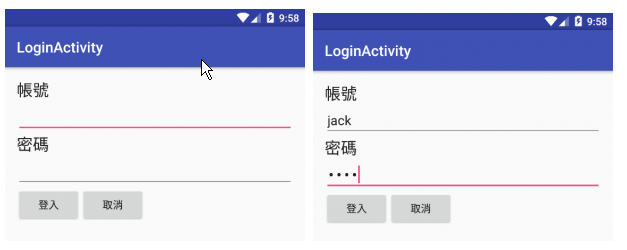
當帳號與密碼都正確時，以Toast顯示浮動文字，並呼叫Activity類別的finish()方法以結束本活動，因為本活動的前一個是MainActivity，因此會回到MainActivity畫面。若登入失敗，以對話框告知使用者。

最後，回到MainActivity的onCreate方法，產生Intent並轉換畫面至LoginActivity，如下：

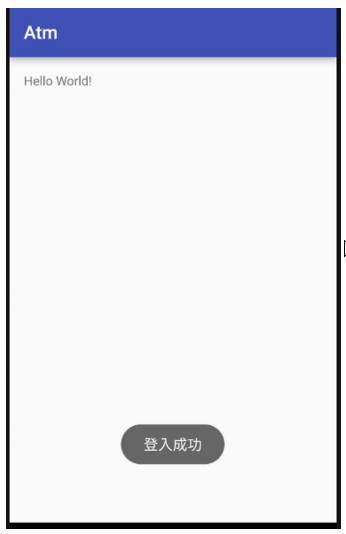


執行結果

一開始因為未登入(logon = false)，開啟LoginActivity如下圖：



輸入測試帳號jack與密碼1234後，再按下登入按鈕，如下圖：



相關文章:

[為你的APP設計漂亮的登入畫面](https://litotom.com/2016/07/19/prettylogin/)

[Android中使用Firebase的E-mail登入功能教學 (三)](https://litotom.com/2016/06/03/firebase-email-signin-p3/)

[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/)

[使用RecyclerView展示Firebase資料庫與雲端儲存的相片](https://litotom.com/2016/08/29/firephoto-recyclerview-storage/)

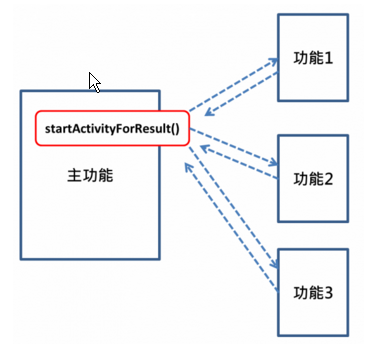
[如何在Android Studio設計並套用自己的Theme](https://litotom.com/2016/07/21/design-theme/)

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

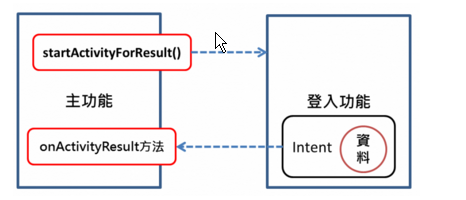
轉換Activity取得結果

其實，上一篇 [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/)  所完成的功能實際上是有問題（bug）的，在登入功能畫面時假如按下返回鍵，也可以正常回到主畫面。這個問題出在從MainActivity中過去LoginActivity時，使用的是startActivity方法，而不是startActivityForResult方法，由方法名稱可看出來，startActivityForResult是到另一個畫面中並取得結果(資料)的，在本例的登入功能應該使用這一個方法才是。

一個主功能的Activity可能會轉換到不同的畫面後，再回到主功能畫面後若需要取得特定的結果，使用startActivityForResult方法，可達成這類的需求，如下圖：



當畫面轉換至登入功能後，在結束的地方，以程式碼撰寫，並把欲傳回主功能的資料放在Intent中，再結束Activity，當回到主功能時，會自動呼叫「onActivityResult方法」，如下圖：



轉換Activity並取得結果的設計方式，有以下幾個步驟：

定義代表該功能的常數

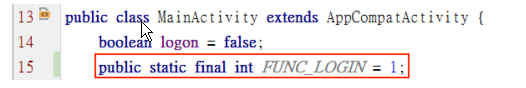
呼叫startActivityForResult

在功能畫面結束之前設定返回值

回到主功能的onActivityResult方法

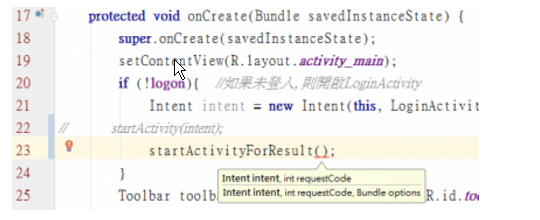
定義功能常數(int)

在MainActivity中新增一個類別層級常數FUNC\_LOGIN，代表登入功能，如下：



呼叫startActivityForResult

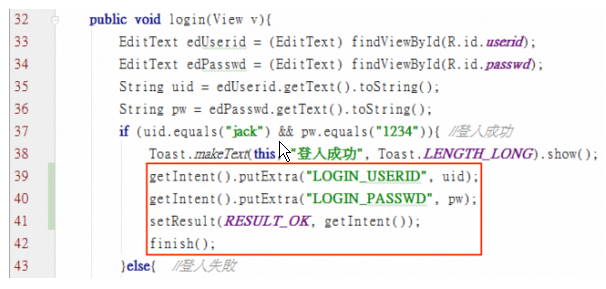
在onCreate方法判斷尚未登入時，改以startActivityForResult方法啟動LoginActivity，如下圖：



上圖方法的第二個參數，使用上一步驟所定義的功能常數FUNC\_LOGIN，如下：

在功能畫面結束之前設定返回值

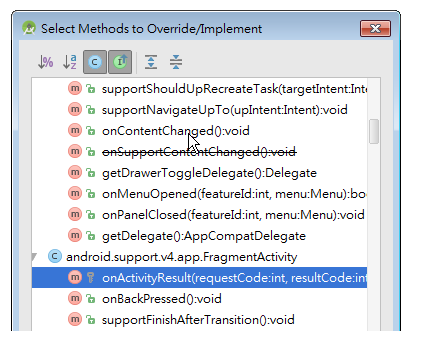
開啟LoginActivity，在login方法中，在原本呼叫finish方法之前，加入程式碼：



上圖程式碼第39與40行將帳號與密碼放入Intent物件中，第41行再呼叫Activity的setResult方法，設定這一個功能畫面的結果為RESULT\_OK，並把Intent物件放入結果中，目的是在功能結束後，回去MainActivity後仍可得到結果。

回到主功能的onActivityResult方法

回到MainActivity中，按下覆寫方法的組合鍵「Ctrl+O」，再選擇覆寫onActivityResult方法，如下圖：

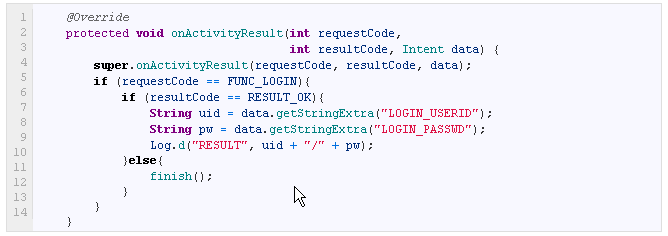


覆寫完成後，onActivityResult方法如下：



方法內傳入參數的requestCode即是第一定義與第二步驟所傳遞的功能常數，第二與第三個參數則是由第三步驟在LoginActivity結束前所設定的RESULT\_OK與Intent物件，參數名稱為resultCode與data。

接下來應該先判斷是否是requestCode是否為FUNC\_LOGIN，是的話再進一步判斷resultCode是否為RESULT\_OK，最後再取得Intent中的資料，程式碼如下：



執行，並輸入帳號jack與密碼1234後，回到MainActivity之後檢視LogCat，將可找到如下圖的Log資料：



相關文章:

[使用者拒絕Android 6.0的危險權限後，APP難道只有被移除一途？Google推出的好物easypermissions](https://litotom.com/2017/09/15/android-6-permission-easypermissions/)

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

[使用 Android Studio 開發 Firebase 快速導入 Email and password Authentication 設定並實作登入功能](https://litotom.com/2016/11/21/as_222_firebase_email/)

[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/)

[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/)

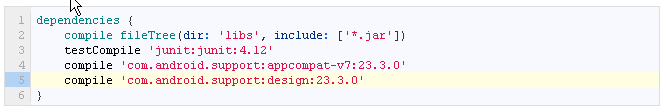
Material Design實感設計，浮動按鈕與它的事件處理

Material Design實感設計

「Material Design」官方稱為「實感設計」是Google所公佈一套適用於Android、iOS、網頁、平板或電視等平台的設計規則，它是一個與以前「仿物化」的設計完全不同的概念，iPhone幾年前的圖示、按鈕等介面以仿真為主要目的，像是按鈕有光影、3D、弧度等效果，盡其所能的設計出擬真的圖示與介面。但近年來，設計的概念不再重視如何說服使用者的眼睛，而是以「扁平化」的設計為主流，由iOS後來整個介面的改變可得知。

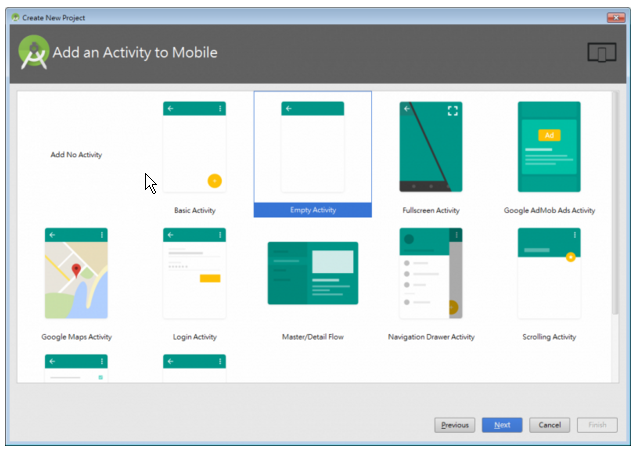
Google推出的跨裝置適用的設計指南就是以實感設計為基礎，並在Android 5.0(Lollipop)中落實，以扁平化的介面設計為基礎，加上材質分層級的概念的一個整體使用經驗的設計。

自從Android 5.0開始，許多新元件就是因應實感設計而推出的，例如「FloatingActionButton」浮動按鈕、「SnackBar」訊息提示等元件，使用這類由Android 5.0開始才有的實感設計類別，需要導入「com.android.support.design」函式庫，在build.gradle(Module:app)中的dependencies區塊加入，如下：



具浮動按鈕的Activity

當我們使用Android Studio建立活動時，可選擇許多種類的Activity，例如「Basic Activity基本活動」、「Empty Activity空白活動」、「Fullscreen Activity全螢幕活動」或「Google Maps Activity地圖活動」等，如下圖：



使用者可依需求選擇要產生的活動類型，以說明幾個重點項目：

| Basic Activity |

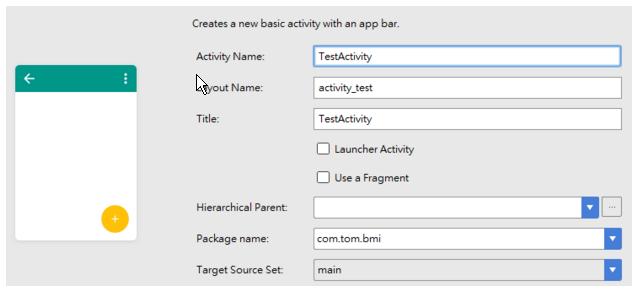
Basic是使用實感設計（Material Design）的基本活動，如預覽圖所示，它使用了ToolBar工具列與FloatingActionButton浮動按鈕，提供使用者最新的互動介面感受，也自動導入「com.android.support.design」函式庫，可使用實感設計帶來的新元件。

| Empty Activity |

產生一個空白的活動，它也是最陽春、不加料的活動，只產生一個繼承Activity的類別、專用的版面配置檔與在AndroidManifest.xml加入活動元素，適合由零開始實作一個完整的活動類別。

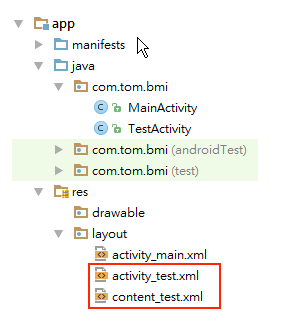
產生活動

筆者將在Bmi專案內新增一個預設就具有浮動按鈕的活動類別，名稱為TestActivity，請點擊專案區中的app，再使用功能表的「File/New/Activity/Basic Activity」，並修改類別名稱為「TestActivty」，如下圖：

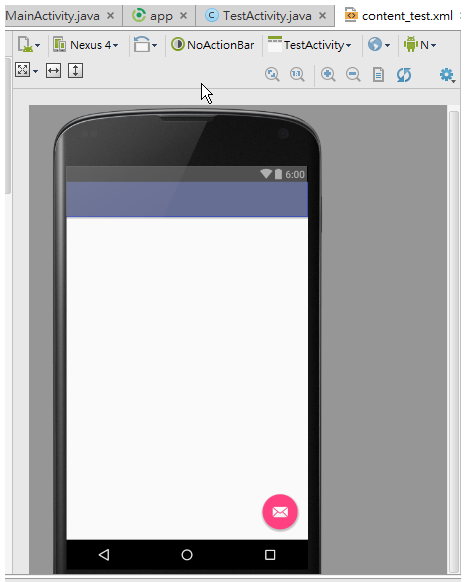


由上圖的左方示意圖可觀察有一個圓形按鈕在活動的右下方，活動的右上方還有選項的三個點圖示。

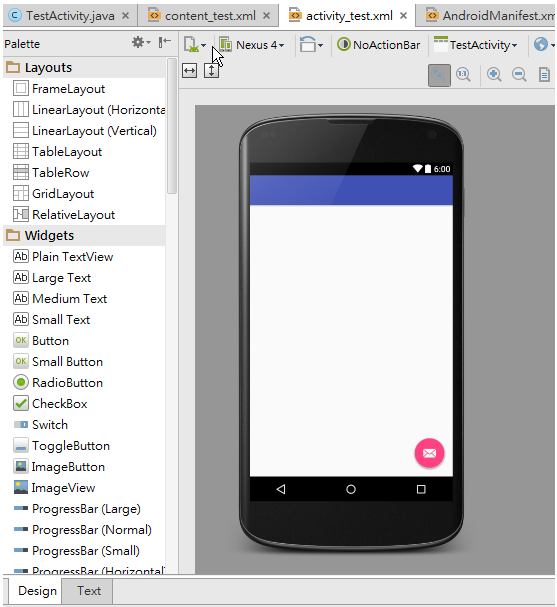
自動產生的版面配置檔有兩個，一個是「activity\_test.xml」，另一個是「content\_test.xml」，如下圖：



完成後預設開啟的檔案是「res/layout/content\_test.xml」，它是整個活動的主畫面區塊，內有浮動按鈕，但它不包括上方的ToolBar元件，如下圖：



請再開啟「activity\_test.xml」，會在編輯區域中開啟檔案，顯示如下圖：

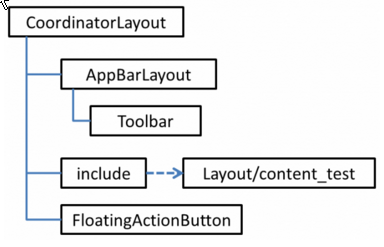


畫面的最上方是一條橫向長條狀的「Toolbar」，在其下方整塊是主要畫面區塊，另外，右下方有一個小圓形的圖示，稱為「浮動式動作鈕(Floating Action Button」，使用者點擊它進行其他的動作，如寄送Email、分享、新增筆記等。

讀者應該也發現content\_test.xml與activity\_test.xml所顯示的預覽圖是一樣的，代表了雖然畫面由activity\_test.xml設計配置，但因為參照(匯入)了content\_test.xml的內容，因此設定畫面中的主要區域是由content\_test.xml所負責。

Layout的配置

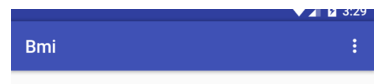
辨認XML的架構對於進階的開發人員是必要的能力，以便日後能夠自行微調畫面配置中的元件資訊，在此，筆者該檔案中的類別名稱與樹狀結構整理如下圖：



整個畫面主要由「CoordinatorLayout」元件所構成，內有「AppBarLayout」、「include」與「FloatingActionButton」三項元素，說明如下：

❶ AppBarLayout元素

畫面中的最上方橫向長條狀區域，在這個元素中放置了一個Toolbar工具列，是Android應用程式畫面中常用的元件，可加入功能項目，如下圖：



❷ include元素

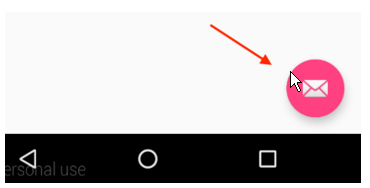
include元件可加入另一個畫面配置XML檔，在本例其屬性layout的值為「@layout/content\_test」，代表這個元素的位置會導入另一個在layout/content\_test.xml檔的內容。在專案中連擊「res/layout/content\_test.xml」開啟檔案並切換至原始碼，下圖是content\_test.xml的原始碼：



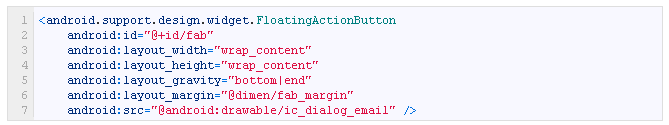
其內容主要由一個「RelativeLayout」容器元件所構成。

❸ FloatingActionButton元素

浮動式動作鈕是顯示在所有元件的上層的一個快速功能按鈕，用途是類似快捷鈕，方便使用者能快速直覺地使用常用功能，如下圖：

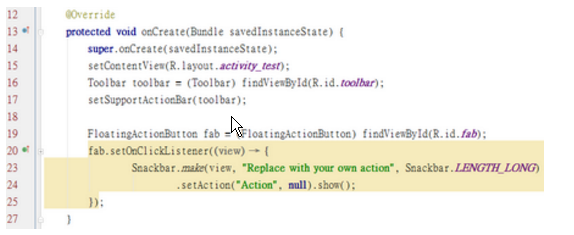


浮動按鈕在版面配置中使用的元素為「com.support.design.widget.FloatingActionButton」類別，預設ID值為「fab」如下圖：



類別程式碼說明

請開啟「TestActivity」類別，其onCreate方法內容如下：



上圖在onCreate方法中的第14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三項工作，說明如下：

❶ 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，使用了「res/layout/activity\_test」版面資源。

❷ 啟用Toolbar

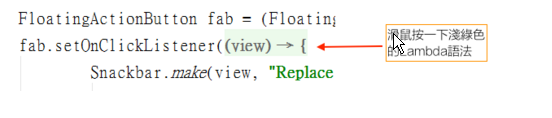


第一行呼叫findViewById方法，findViewById方法亦是由Activity類別設計用來取得畫面配置中的元件，因為findViewById方法回傳的是android.view.View類別，因此需經過轉型而得到Toolbar物件。最後再呼叫setSupportActionBar方法，指定本類別上方橫向長條的元件。

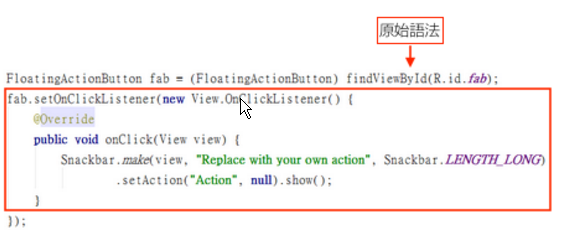
❸ 設定浮動式動作鈕



同樣先呼叫findViewById取得layout中的FloatintActionButton元件－fab，再呼叫其setOnClickListener方法設定事件傾聽者，未來當使用者按下手機畫面中的浮動式動作鈕後，會自動執行傾聽者內實作的onClick方法，上圖使用了Java語言的「匿名類別」與Java 8推出的Lambda新語法。請觀察圖中的淺綠色部份，Android Studio雖然以Lambda方式顯示，但實際上程式碼仍然以舊式的匿名類別方式儲存，請讀者在淺綠色部份按一下滑鼠左鍵，立即會以原始方式顯示，如下圖：



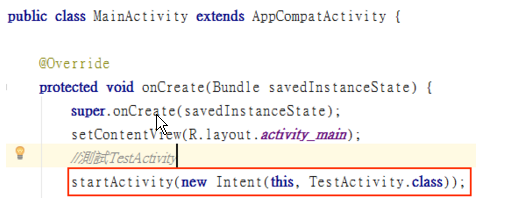
點擊後會展示原始語法，如下圖方塊標示：



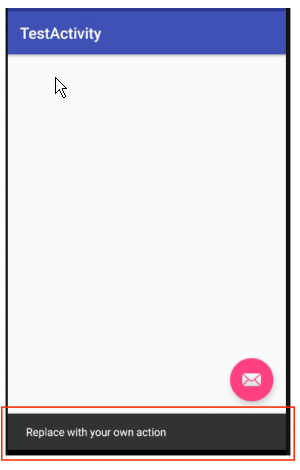
上圖onClick方法內的程式碼使用了Snackbar類別的make方法，在晝面下方顯示一條暫時訊息列，後續再介紹Snackbar類別的使用方式。

執行結果

請在Bmi原本的主活動類別MainActivity內的onCreate方法中加入啟動TestActivity的程式碼，以便測試活動，測試完畢後再將程式碼註解掉，如下：



執行結果如下圖，請試著點擊浮動按鈕：



點擊浮動按鈕後在下方所出現的長條提示即是SnackBar元件的效果，它也是Android 5.0實感設計新推出的元件。

相關文章:

[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/)

[清單元件實作－清單式功能表](https://litotom.com/2016/03/26/%e6%b8%85%e5%96%ae%e5%85%83%e4%bb%b6%e5%af%a6%e4%bd%9c%ef%bc%8d%e6%b8%85%e5%96%ae%e5%bc%8f%e5%8a%9f%e8%83%bd%e8%a1%a8/)

[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/)

[Cloud Firestore 之二部曲，資料新增讓你的 APP 馬上成為雲端應用](https://litotom.com/2017/11/14/firestore-2-add/)

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

Material Design之SnackBar提示訊息元件

SnackBar提示訊息元件

SnackBar是實感設計中的一個新元件，Android中原本有Toast可以提供較輕量的訊息給使用者，但Toast不符實感設計的理念，因此推出另一個提示使用者訊息的元件，也就是「android.support.design.widget.Snackbar」類別，它的使用方式與Toast類別非常類似，如下：



執行時顯示在活動的最下方，如下圖：



設定Action

與Toast不同的是，它不只是短時間顯示就消失，還能在顯示時讓使用者點擊，使用setAction方法可在訊息旁顯示點擊的提示文字，並在點擊時加入事件處理，如下：



setAction方法需要一個字串，如”Action”，與按下動作時的事件處理，如上述的OnClickListener匿名類別，執行效果如下圖：



Action的字體顏色則可透過setActionTextColor方法設定，按下時的設計可寫在匿名類別內的onClick方法，如下：



相關文章:

[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/)

[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/)

[清單元件實作－清單式功能表](https://litotom.com/2016/03/26/%e6%b8%85%e5%96%ae%e5%85%83%e4%bb%b6%e5%af%a6%e4%bd%9c%ef%bc%8d%e6%b8%85%e5%96%ae%e5%bc%8f%e5%8a%9f%e8%83%bd%e8%a1%a8/)

[清單元件實作－下拉式選單Spinner](https://litotom.com/2016/03/26/%e6%b8%85%e5%96%ae%e5%85%83%e4%bb%b6%e5%af%a6%e4%bd%9c%ef%bc%8d%e4%b8%8b%e6%8b%89%e5%bc%8f%e9%81%b8%e5%96%aespinner/)

[使用Firebase的Email註冊功能實作Android APP教學(四)](https://litotom.com/2016/06/23/firebase-email-register/)

使用SharedPreferences存取設定資料

在Android中設計應用程式時，若想要儲存簡單資料，可使用android.content.SharedPreferences類別，儲存如帳號、設定、上一次登入時間、遊戲關卡或電子郵件等，好讓APP在下一次執行時可讀取到這些上一次儲存下來的資料。資料的儲存格式是XML檔，儲存在Android手機中每個APP都會擁有的一個專用目錄下。

偏好設定SharedPreferences

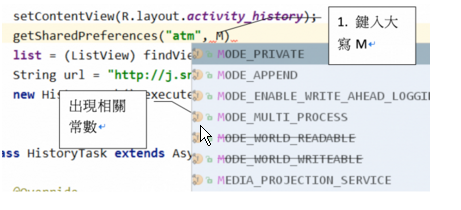
SharedPreferences是Android的一個介面，可在Activity中呼叫getSharedPreferences(String, int)方法得到物件。

｜參數：檔案名稱String｜

第一個參數是定義此設定檔的檔名，不需要指定副檔名，如”atm”即寫入到atm.xml。

｜參數：存取權限int｜

設定這個設定檔的存取權限，有四種選擇 (0,1,2,4)，我們不需要背這些值，使用Activity中已定義的常數MODE\_XXX即可。在設計時鍵入大寫M後可快速選擇這些參數，如下圖：



參數的值與其說明如下：

MODE\_PRIVATE

只允許本應用程式內存取，這是最常用參數。

MODE\_MULTI\_PROCESS

允許多個行程同時存取這個設定檔，這個設定在Android 2.3(含)以前都是預設啟用的，但2.3之後得要指定這個參數才允許多行程同時存取設定檔。

MODE\_WORLD\_READABLE

讓手機中的所有app都能讀取這個設定檔，因為風險性太高，從API 17版開始就不建議使用這個參數了。

MODE\_WORLD\_WRITEABLE

讓手機中的所有app都能存取、寫入這個設定檔，已經不建議使用了，原因與上一個參數相同。

寫入資料

寫資料時應使用SharedPreferences.Editor這個編輯器介面，可由edit()方法取得Editor物件，在Activity中寫入資料的片段程式碼如下：



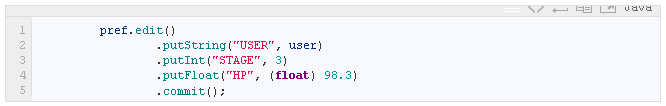
第1行是將被寫入的字串user，內容為jack。

第2行，呼叫getSharedPreferences()方法，產生一個檔名為test.xml的設定儲存檔，並只供本專案(app)可讀取，物件名稱為pref。

第3-4行，呼叫edit()方法取得編輯器物件，此時使用匿名方式呼叫Editor的putString()方法將user字串的內容寫入設定檔，資料標籤為”USER”。

第5行，最後必須呼叫commit()方法，此時資料才真正寫入到設定檔中。

如果有多筆資料要寫入設定檔時，先put資料後，最後再呼叫commit()方法一次寫入，可減少寫入的次數，可兼顧SD卡的效能與降低耗損，如下片段程式：



第1行，呼叫edit()方法取得Editor物件。

第2-4行開始新增三筆資料，分別為字串、整數與浮點數。

第5行，最後才將三筆資料寫入設定檔。

讀取設定檔內的資料

使用SharedPreferences的方法getXX方法讀取先前已寫入的設定值，getString方法為讀取字串設定值，使用getInt方法讀取整數設定值，以此類推，還有getBoolean、getFloat、getLong與getStringSet等方法可取得對應資料型態的設定值。利用getSharedPreferences()方法先取得物件後，再讀取所需的設定值，如下程式片段：



第1行，先取得SharedPreferences物件。

第2行，取得標籤”USER”的設定值，getString方法的第二個參數是預設值(default)，當讀取不到或設定檔內無該設定值時，會傳回這個預設值，本例預設是空字串。

手機內部儲存方式

本節雖然介紹了在程式設計時如何寫入與讀取設定值，但在開發實務過程中，往往需要在除錯時能夠遠端連接至手機或模擬器中，以便進行檢視或手動更改設定值。因此在下一節，筆者將介紹如何使用Android SDK的除錯工具「adb」，登入設備中並檢視設定檔的內容。

相關文章:

[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/)

[清單元件實作－清單式功能表](https://litotom.com/2016/03/26/%e6%b8%85%e5%96%ae%e5%85%83%e4%bb%b6%e5%af%a6%e4%bd%9c%ef%bc%8d%e6%b8%85%e5%96%ae%e5%bc%8f%e5%8a%9f%e8%83%bd%e8%a1%a8/)

[使用RecyclerView展示Firebase資料庫與雲端儲存的相片](https://litotom.com/2016/08/29/firephoto-recyclerview-storage/)

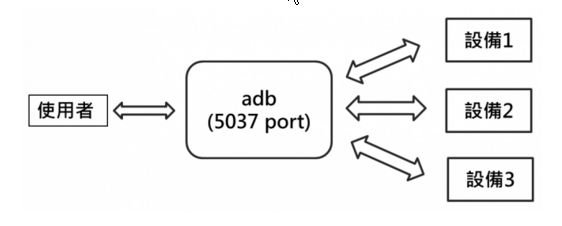
[轉換畫面後，按返回鍵回到前一個Activity時怎麼取得結果? onActivityResult方法](https://litotom.com/2017/06/04/ch5-7-activity-result/)

[Cloud Firestore 之三部曲，設計資料類別(Model)快速新增複雜資料](https://litotom.com/2017/11/20/firestore-3-model-add/)

使用ADB登入模擬器

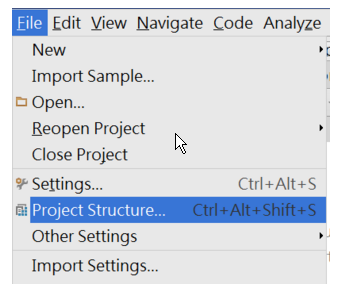
使用ADB登入模擬器

ADB工具(Android Debug Bridge)是Android SDK中所提供的一個除錯工具，使用ADB可連接模擬器與手機，進行除錯、測試、檔案處理等工作。它是個客戶端工具，也是個在使用者端背景執行的伺服器，接收使用者的指令，並轉送到設備端（包括模擬器與實體手機），如下圖。

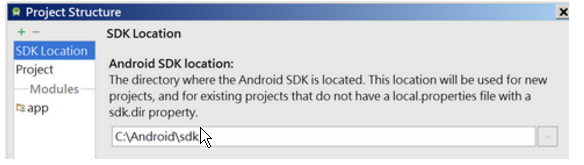


ADB執行檔

ADB執行檔放置於Android SDK所安裝目錄下的「platform-tools」子目錄下，SDK安裝所在目錄可打開專案架構設定得知，請在功能表中選擇「File/Project Structure」，如下圖：



在專案架構(Project Structure)設定中，點擊左方「SDK Location」即可得知SDK安裝目錄，如下圖為自訂的SDK目錄：



若安裝Android Studio時使用預設的SDK安裝位置，Windows的預設路徑為「C:\Users\登入帳號\AppData\Local\Android\sdk\」，Mac則是「/Users/登入帳號/Android-sdks/」。

設定系統環境變數

在開發Android應用程式時常需使用ADB工具，在開發環境中設定環境變數，方便日後很快就能執行ADB執行檔，將ADB執行檔的路徑加入到系統環境中，可加快開發與除錯的速度。

｜在Windows中加入路徑｜

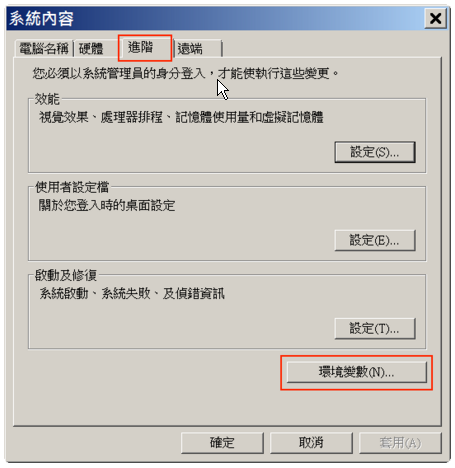
Step 1  在「電腦」上按右鍵，選擇「內容」，如下圖



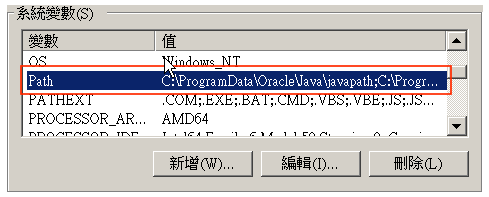
Step 2  左上方點擊「進階系統設定」，如下圖：



Step 3  在「進階」標籤中點擊下方的「環境變數」，如下圖：



Step 4  在「系統變數」下找到「Path」變數後，連擊或按下「編輯」，如下圖：



Step 5  在變數值的最尾端加上「;」後，再輸入「Android SDK安裝路徑\platform-tools\」，如下圖：



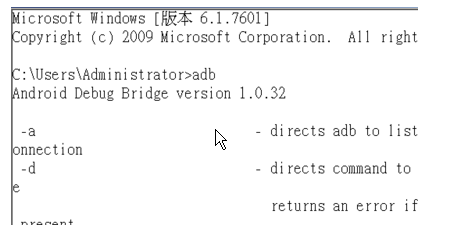
依序按下「確定」完成設定，未來打開「命令提示字元」時，即可直接執行adb.exe執行檔了。

｜在Mac中加入路徑｜

在終端機中使用以下指令加入PATH環境變數設定，替換使用者帳號為讀者環境中的帳號名稱，以下指令不需要跳行：



最後開啟「命令提示字元」或「終端機」，執行「adb」，確認有無出現如以下的回應：



使用adb登入模擬器

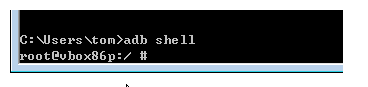
｜列出已連線設備｜

使用「adb devices」指令可列出目前與ADB建立連線的所有設備，每個設備有其辨識用的名稱，例如：



｜連接預設的設備｜

如果目前ADB只連接到一個設備，直接使用「adb shell」指令即可遠端連線至該設備，連線成功後會進入Android系統，並取得命令提示（prompt），如下圖：



圖中的最後的「＃」號，代表目前登入為系統超級使用者root，可以自由複製、刪除、管理Android內的檔案與目錄，模擬器大都能取得root權限。若為連結實體手機時，連接後的提示字元最後為「$」號，代表操作的權限是受到限制的，無法提供管理層面的指令。

最後，使用「exit」指令，即可離開遠端連接的設備，如下圖：



｜指定連接的設備｜

若是同時有一台以上的設備連接至ADB時，遠端登入時就必須指定要連接至那台設備，例如目前有兩個設備，adb devices指令結果如下圖：



每台設備的名稱就是前面所顯示的，如「192.168.56.102:5555」是第二台設備的名稱，要遠端連接至第二台設備需使用「-s」選項指定名稱，最後再加上shell連線指令，如下圖：



相關文章:

[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/)

[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/)

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

[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/)

[[影片] Android Studio與Firebase實作類似Instagram功能，上傳手機照片至雲端(Android 6.0)](https://litotom.com/2016/07/24/%e5%bd%b1%e7%89%87-%e4%bd%bf%e7%94%a8firebase%e5%af%a6%e4%bd%9c%e9%a1%9einstagram%e4%b8%8a%e5%82%b3%e6%89%8b%e6%a9%9f%e7%85%a7%e7%89%87%e8%87%b3%e9%9b%b2%e7%ab%af/)

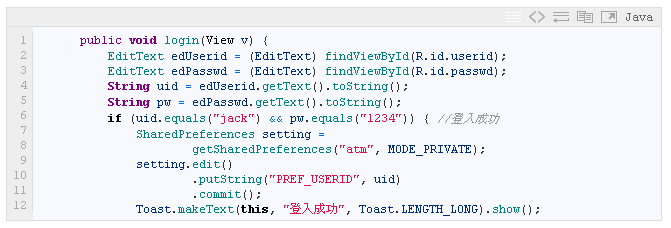
實務練習，怎麼讓APP記住我的登入帳號? Atm專案

實務練習－記住登入帳號

本篇繼續使用本系列文章的Atm專案進行實務練習，在登入畫面(LoginActivity)中設計記住登入成功後的帳號名稱，並在未來再次啟動APP時，能將已記住的帳號預先填入在帳號輸入方塊中。

儲存登入帳號

首先在登入按鈕的事件方法login中判斷帳號與密碼的程式碼中，加入使用SharedPreference儲存帳號的程式碼，如下：



第7-8行先取得儲存設定物件setting

第9行先呼叫其edit()方法取得SharedPreferences.Editor編輯器物件

第10行再呼叫putString方法將uid帳號字串儲存

最後11行再呼叫commit()方法，實際進行儲存資料。在此帳號資料儲存的鍵值為「PREF\_USERID」。

填入已儲存帳號

當LoginActivity開啟前，讀取儲存設定中的「PREF\_USERID」鍵值，並將該值設定至帳號的輸入方塊中，請在LoginActivity的onCreate方法中加入以下程式碼：



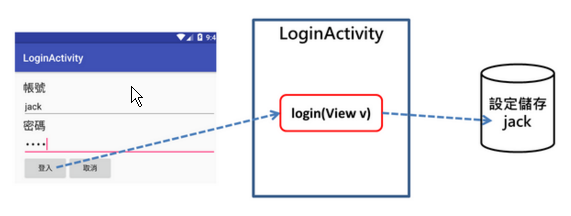
第4行先取得畫面上的帳號EditText元件，名稱為edUserid。

第5-6行取得儲存設定物件setting。

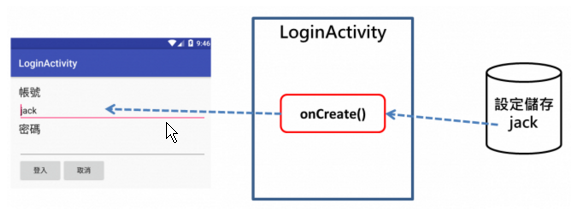
第7行設定edUserid的預設值。

執行結果

先執行一次Atm專案，第一次先輸入正確的帳號與密碼(jack與1234)，讓帳號能先儲存，如下圖：



結束Atm APP，再執行一次，此時會在LoginActivity顯示之前先讀取設定，將帳號預先設定在輸入方塊中，如下圖：



檢視APP執行後產生的設定檔

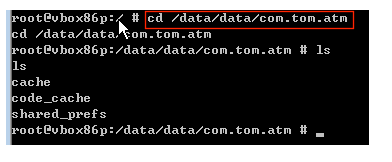
經過上述步驟在模擬器中正確執行後，請使用adb shell登入模擬器中，如下圖：



上圖顯示的模擬器是192.168.56.101:555，請讀者使用自己環境中的模擬器名稱。

Android系統的所有應用程式安裝在「/data/data/」下，在這之下建立應用程式專用資料夾，資料夾的名稱是應用程式所訂定的「package名稱」，本例的Atm專案即放置在「/data/data/com.tom.atm/」下。

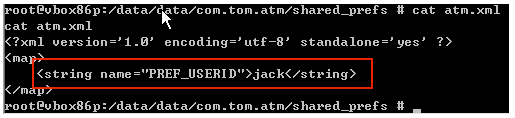
使用cd指令，切換至Atm專案的安裝資料夾，如下：



使用ls指令可列出目前資料夾下的檔案清單，由上圖可觀察到一個叫「shared\_prefs」的資料夾，請cd至該資料夾，並使用ls指令列出檔案清單，如下：



該資料夾已有一個XML檔「atm.xml」，請再使用「cat atm.xml」可印出atm.xml檔案的內容，如下圖：



atm.xml中即是使用SharedPreferences所儲存的資料鍵「PREF\_USERID」與值「jack」。確認後，可使用「exit」指令離開模擬器。

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/)

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

[[影片] Android Studio與Firebase實作類似Instagram功能，上傳手機照片至雲端(Android 6.0)](https://litotom.com/2016/07/24/%e5%bd%b1%e7%89%87-%e4%bd%bf%e7%94%a8firebase%e5%af%a6%e4%bd%9c%e9%a1%9einstagram%e4%b8%8a%e5%82%b3%e6%89%8b%e6%a9%9f%e7%85%a7%e7%89%87%e8%87%b3%e9%9b%b2%e7%ab%af/)

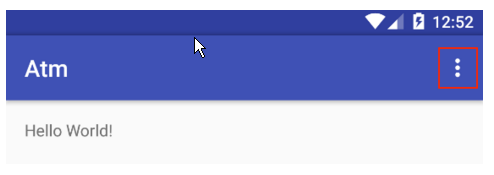
[來證明你會了吧，Atm專案之登入功能實務](https://litotom.com/2017/06/03/ch5-6-atm-login/)

[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/)

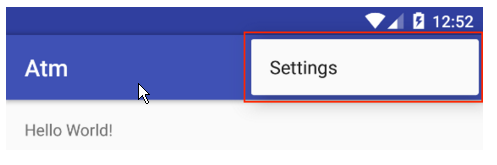
實作Activity上方選單Menu與下拉項目功能

選單Menu

Android 6.0開始，在上方橫條狀區塊使用的是Toolbar元件，在其右方有個選單按鈕，垂直三個小點的圖示即是應用程式的選單鈕，如下圖：



按下選單鈕後會出現選單（Menu），預設只有一個選單項目(Menu item)為Settings，如下圖：

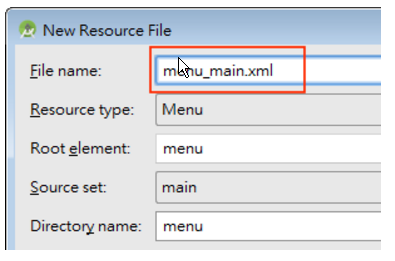


設計選單功能

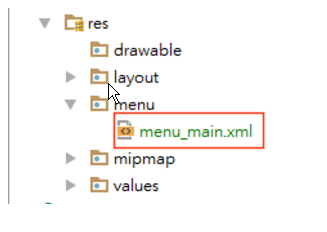
在Activity出現在畫面之前，會自動呼叫Activitiy的「onCreateOptionsMenu」方法，方法的參數帶來一個menu物件，在本節，筆者將在Atm專案中加入選單元件。

｜建立選單配置資源｜

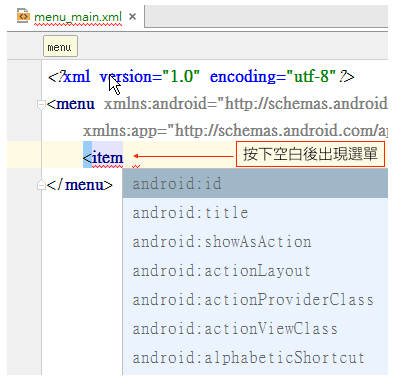
請在左方專案區中點擊app後，使用功能表的「New/Android resource file」，產生一個資源檔，請在畫面中Reource type下拉選單將此資源定義為「Menu」，再輸入File name檔名為「menu\_main.xml」，如下圖：



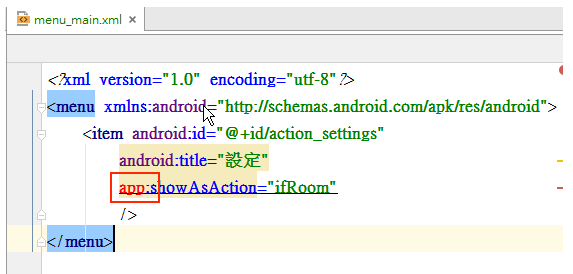
建立的選單配置檔在「res/menu/menu\_main.xml」，請開啟它，如下圖：



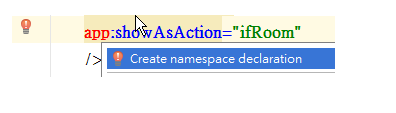
選單配置檔的根元素一律是menu，請在menu\_main.xml中加入一個item元素，手動輸入item元素後，按下空白時會自動出現下拉屬性選單，如下圖：



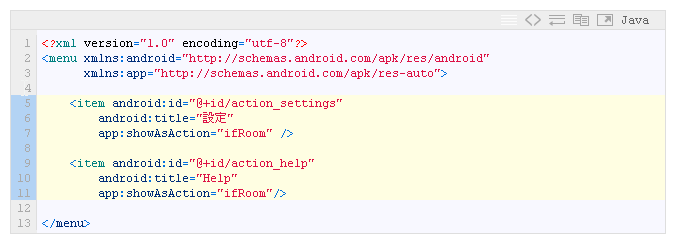
請為其加上「android:id」、「android:title」與「app:showAsAction」三項屬性，如下圖：



在上圖的「app:showAsAction」的app出現紅色錯誤，這是因為這個XML檔並未加入app這個必要的namespace宣告，請將游標停在在紅色的app處，按下快速解決鍵「Alt+Enter」，選擇「create namespace declaration」後Enter加入宣告，如下圖：



請以同樣的方式，再新增一個項目Help，menu\_main.xml檔案原始碼如下：



在選單中宣告二個項目(item)，其資源ID為「action\_settings」、「action\_help」，標題為「設定」，而showAsAction代表的是它在ToolBar上顯示的原則。

在畫面上方選單鈕的左方空白區稱之為「Action」動作按鈕區域，常用的選單功能可固定顯示在這個區域中，如下圖所示：

item元素中可為其設定「app:showAsAction」屬性，可使用以下值：

never：不會顯示在Action區域

ifRoom：若Action區域有空間，就會顯示

always：無論如何都會顯示在Action區域上，但不建議使用此值

withText：除了icon圖示外，也顯示item的標題文字

每個元素應該為其訂定「android:id」屬性，用來辨識每個不同的選單項目，也應有「android:title」屬性，代表在畫面中顯示的標題文字。也可使用「android:icon」為項目指定一個圖示。

｜覆寫方法｜

開啟MainActivity類別，使用覆寫方法快速鍵「Ctrl+O」，覆寫「onCreateOptionsMenu」與「onOptionsItemSelected」兩個方法，並在裏面實作以下程式碼：

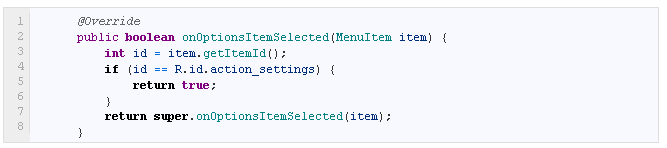
程式碼第5行先呼叫Activity的getMenuInflater方法得到「MenuInflater物件」，它是一個專門用來產生選單的類別，呼叫inflate方法可讀取一個Menu設計圖（XML檔），依照設計圖產生menu物件。就像是充氣機，設計圖就是還未灌氣的氣球。因此，在專案中的「/res/menu/menu\_main.xml」就是MainActivity的選單設計檔。

執行後請先以jack登入，成功登入後的MainActivity如下圖：



選單的事件處理

當按下選單中的任一個項目時，會自動呼叫Activity的onOptionsItemSelected方法，並傳來被按下的項目MenuItem物件，在方法中實作以下程式碼，如下：



第3行先取得選單項目的id值，id值即為設計檔中為每個item所訂定的值。

第4行即是簡單判斷是否按下的是「Settings」項目，目前無任何處理的程式碼，直接回傳true。

如果有多個選單項目，應將if判斷以switch…case改寫，可增加程式的可讀性並提昇日後維護的效率。

相關文章:

[清單元件實作－下拉式選單Spinner](https://litotom.com/2016/03/26/%e6%b8%85%e5%96%ae%e5%85%83%e4%bb%b6%e5%af%a6%e4%bd%9c%ef%bc%8d%e4%b8%8b%e6%8b%89%e5%bc%8f%e9%81%b8%e5%96%aespinner/)

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

[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/)

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

[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/)