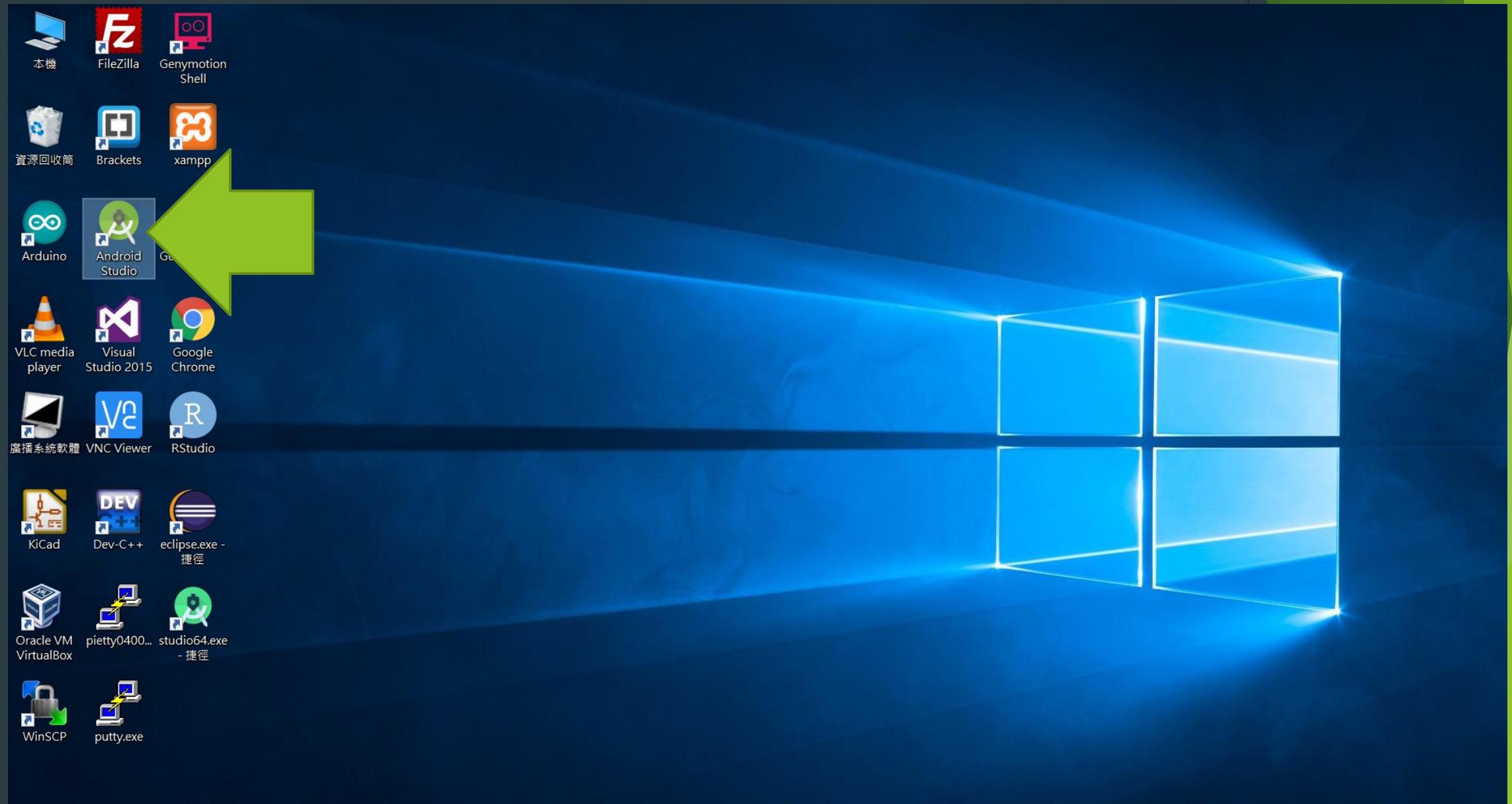


Android APP 程式設計

崑山科技大學 資訊工程系

李宗儒

開啟 Android Studio (I2604電腦教室)



到網大下載Android範例

10901App 程式設計_四資工二

Kun Shan University
崑山網路大學

全校課程 | 我的課程 | 辦公室 | 李宗儒 | 登出 | 正體中文

學習互動區

課程公告

開始上課

課程討論

線上討論

議題討論

分組討論

直播活動列表

同步教室錄影列表

EverCam下載

評量區

作業/報告

測驗/考試

問卷/投票

10901App 程式設計_四資工二 A

Lesson 1

Android Studio 安裝建立新專案

Lesson 2

MyBMI 範例下載

50%

←

目

+

MyBMI_Final.zip (131.27 KB)

下載檔案，開始閱讀

This screenshot shows a learning management system interface for a course titled '10901App 程式設計_四資工二 A'. The main content area displays two lessons: 'Lesson 1' and 'Lesson 2'. In Lesson 2, there is a link labeled 'MyBMI 範例下載' (MyBMI Example Download). To the right of this link, a file named 'MyBMI_Final.zip' is listed with a size of '131.27 KB'. Below the file name is a teal-colored button with the text '下載檔案，開始閱讀' (Download file, start reading) and a download icon. On the far left, a vertical sidebar lists various course modules and links, such as '學習互動區', '開始上課', and '評量區'. The '開始上課' link is highlighted with a teal background. At the top of the page, the university logo 'Kun Shan University 崑山網路大學' is displayed, along with navigation links for '全校課程', '我的課程', '辦公室', '李宗儒', '登出', and '正體中文'.

到網大下載Android範例 儲存位置 D:/AndroidAPP2020

10901App 程式設計_四資工二

學習互動區

課程公告

開始上課

課程討論

線上討論

議題討論

分組討論

直播活動列表

同步教室錄影列表

EverCam 下載

評量區

作業/報告

測驗/考試

Kun Shan University

另存新檔

本機 > 本機磁碟 (D:)

新增資料夾

名稱	修改日期	類型	大小
.metadata	2020/3/6 下午 06...	檔案資料夾	
0328	2018/3/28 下午 0...	檔案資料夾	
20180703	2018/7/3 下午 07...	檔案資料夾	
Adobe	2018/9/5 上午 09...	檔案資料夾	
AndroidAPP2020	2020/9/15 上午 1...	檔案資料夾	
c	2018/9/20 下午 0...	檔案資料夾	
c4080e999	2020/9/18 上午 1...	檔案資料夾	
cpp	2018/3/20 下午 0...	檔案資料夾	
curl	2018/4/24 下午 0...	檔案資料夾	
demo_20200929	2020/9/29 下午 0...	檔案資料夾	
eclipse_jenner	2017/12/13 下午 ...	檔案資料夾	
ffmpeg-20180325-5b31dd1-win64-st...	2018/3/25 上午 0...	檔案資料夾	
Jenner	2018/3/8 下午 06...	檔案資料夾	

131.27 KB)

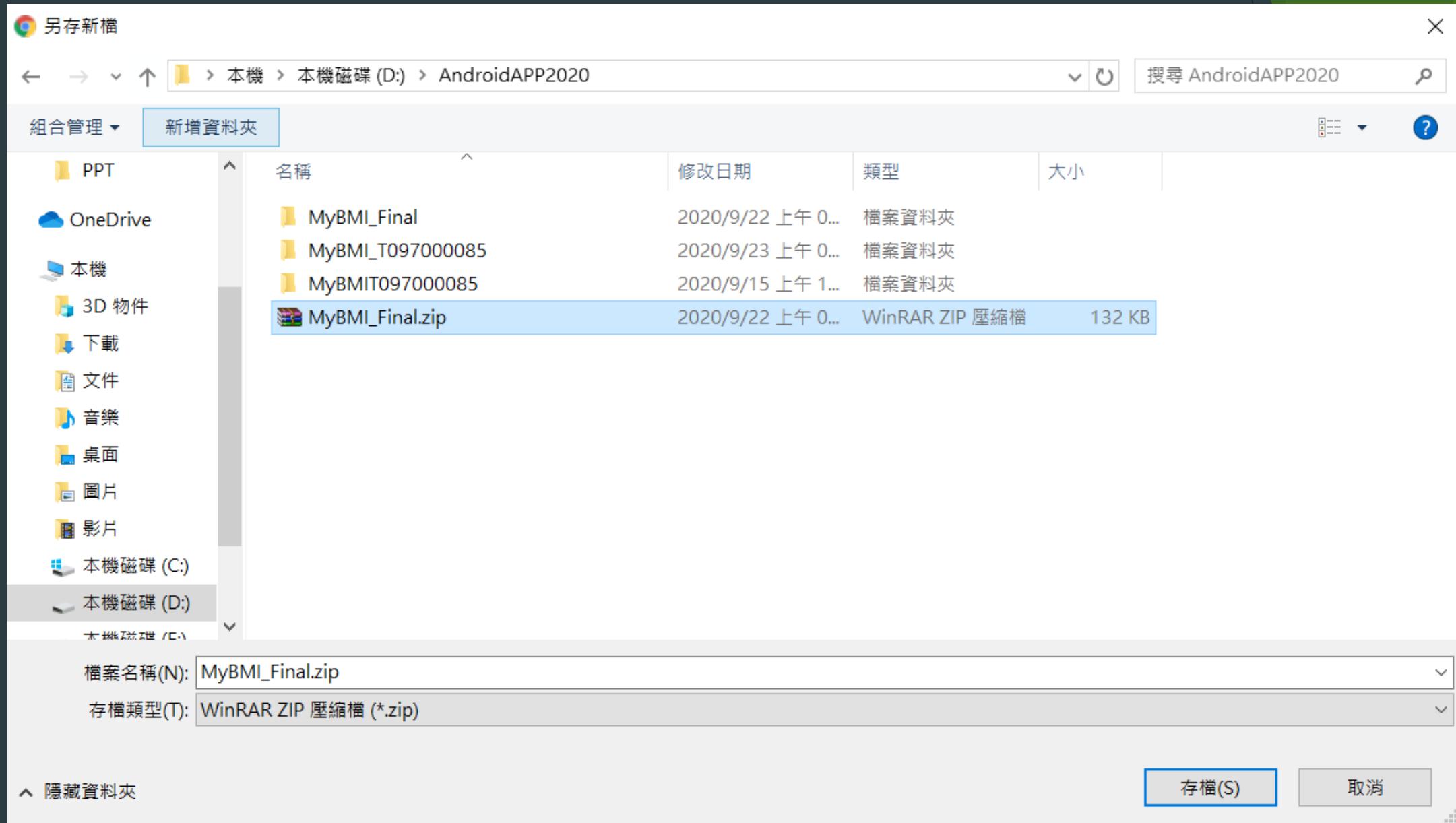
開始閱讀

檔案名稱(N): MyBMI_Final.zip

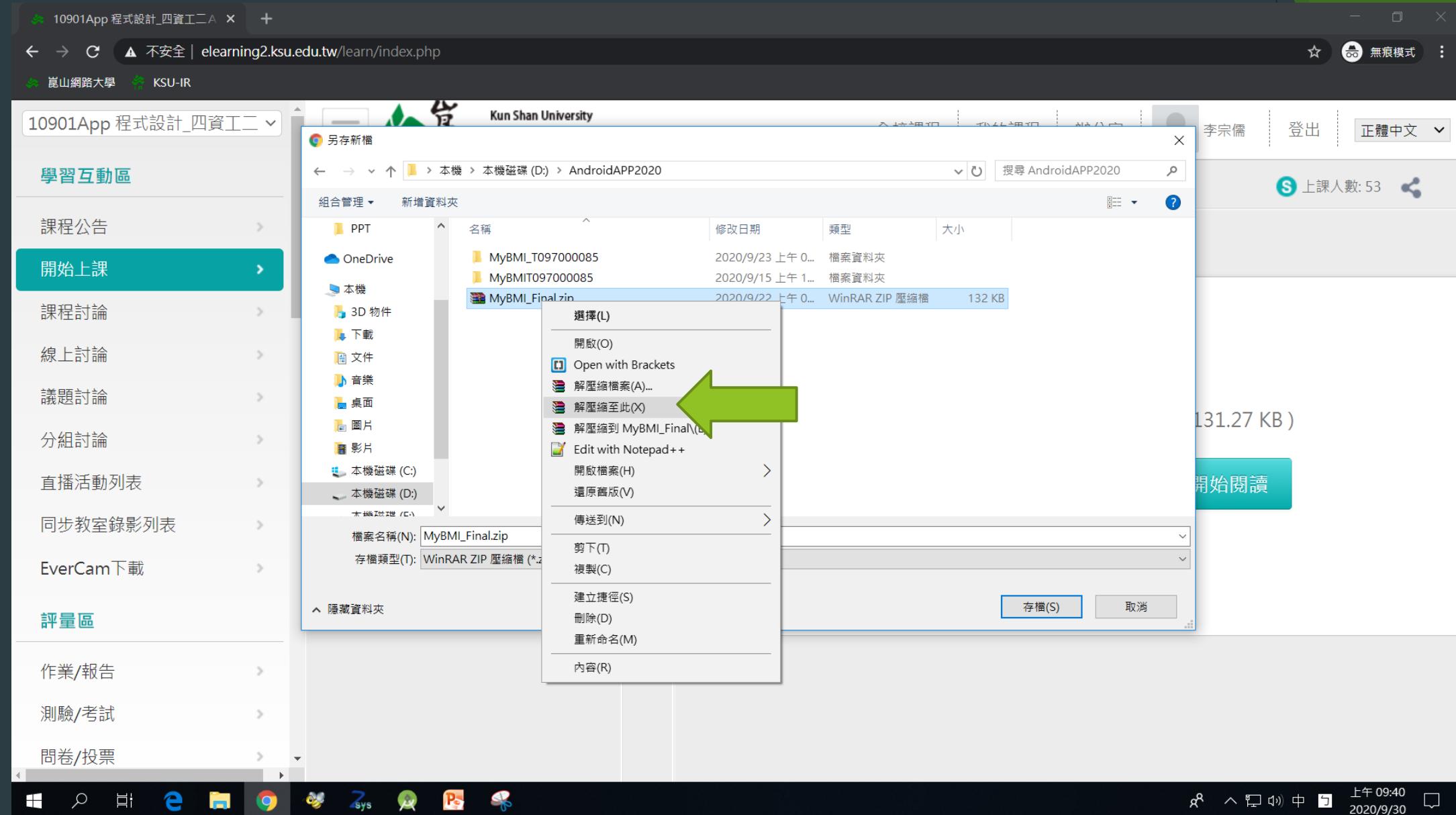
存檔類型(T): WinRAR ZIP 壓縮檔 (*.zip)

存檔(S) 取消

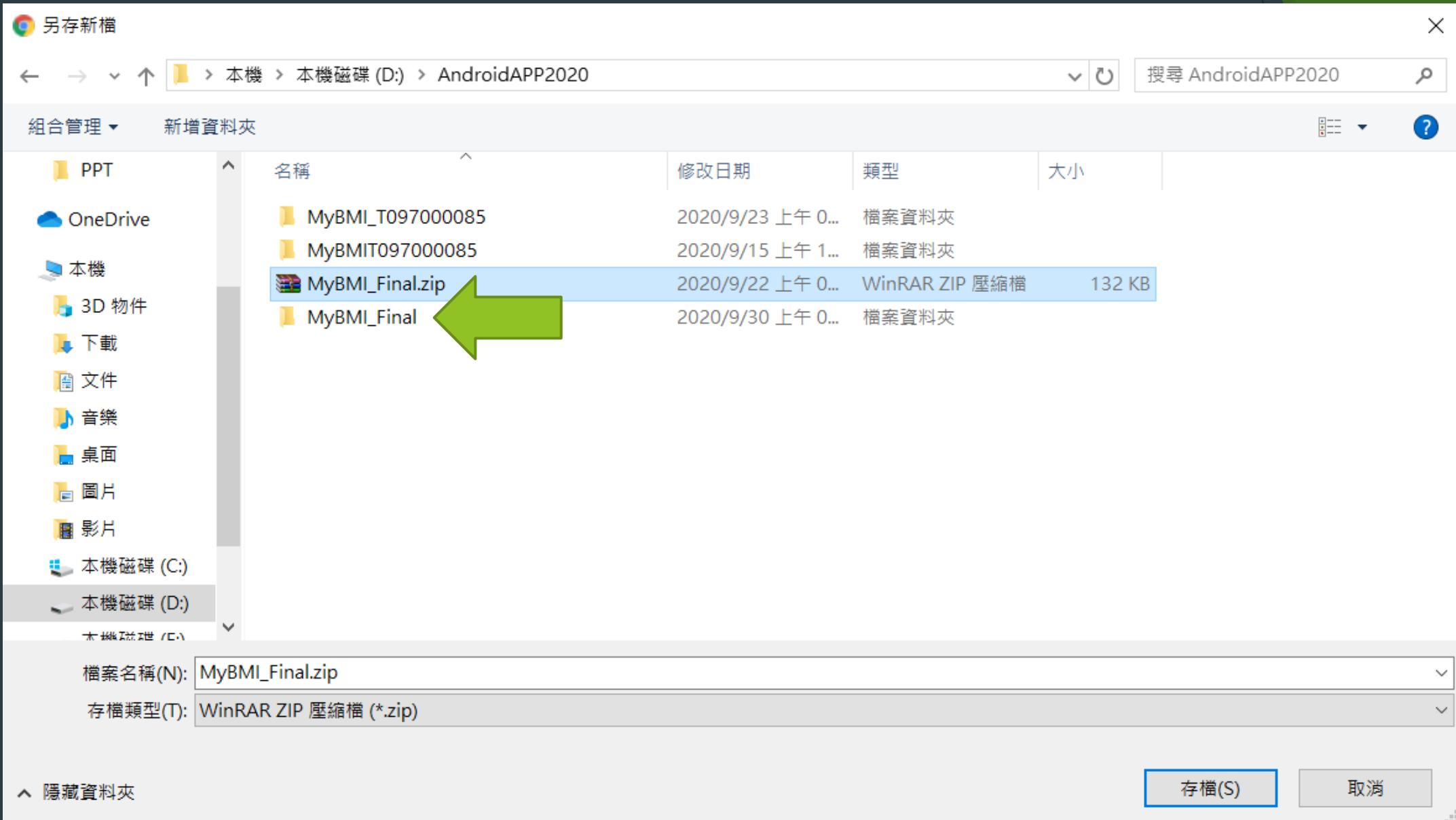
到網大下載Android範例 儲存位置 D:/AndroidAPP2020



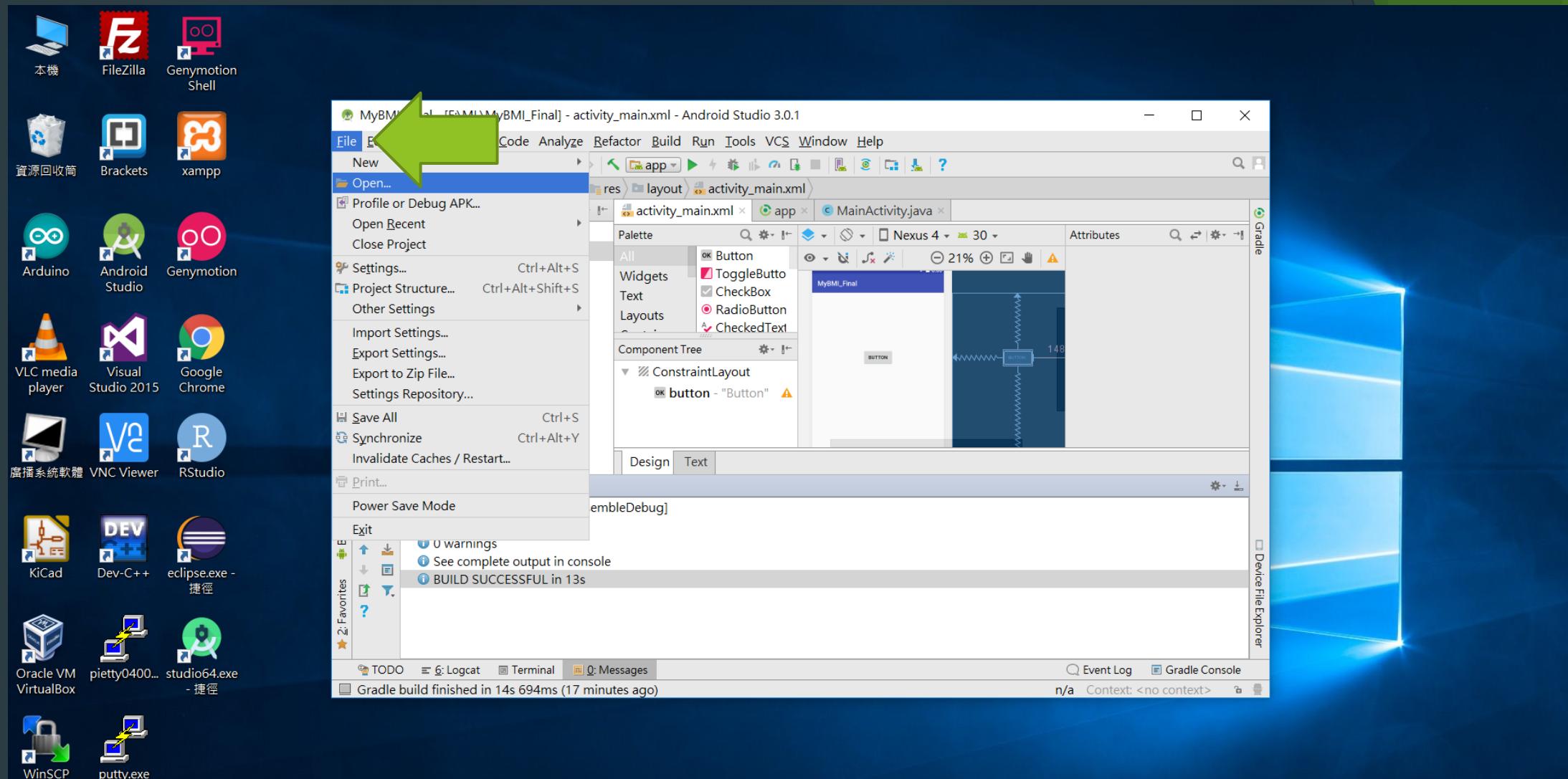
解壓縮 Android範例 D:/AndroidAPP2020



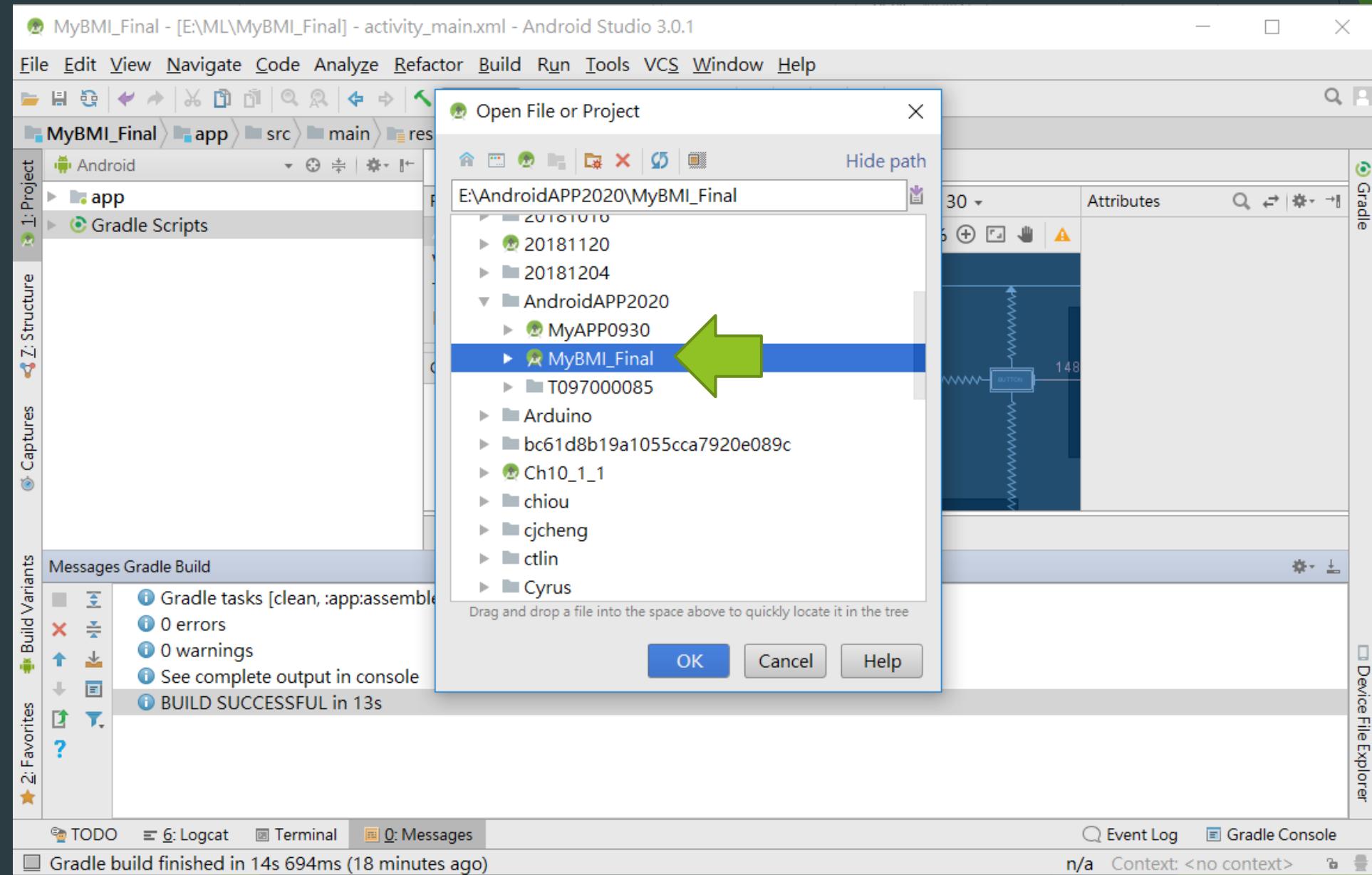
解壓縮 Android範例 D:/AndroidAPP2020



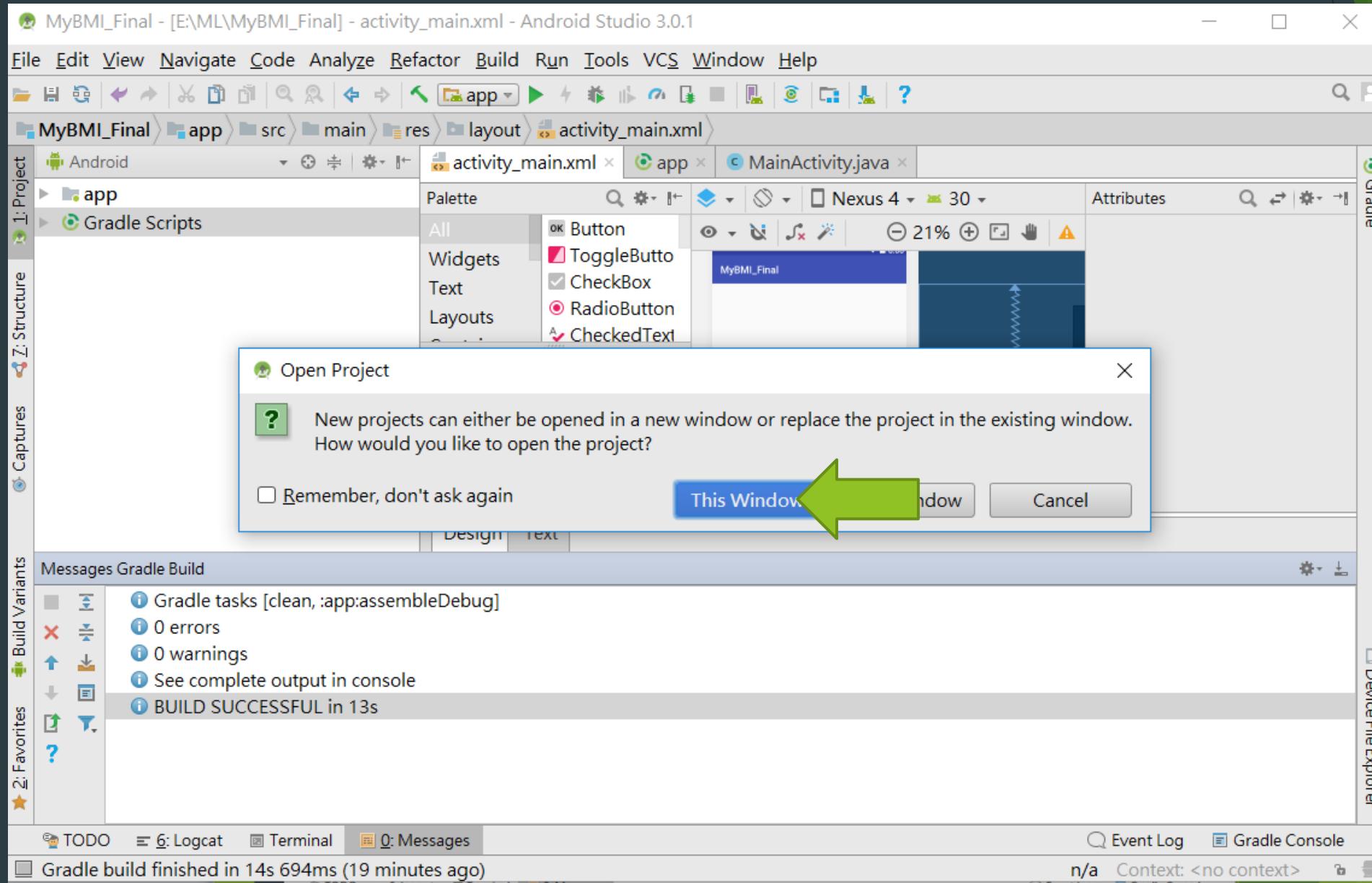
開啟剛剛解壓縮的Android範例 File → Open



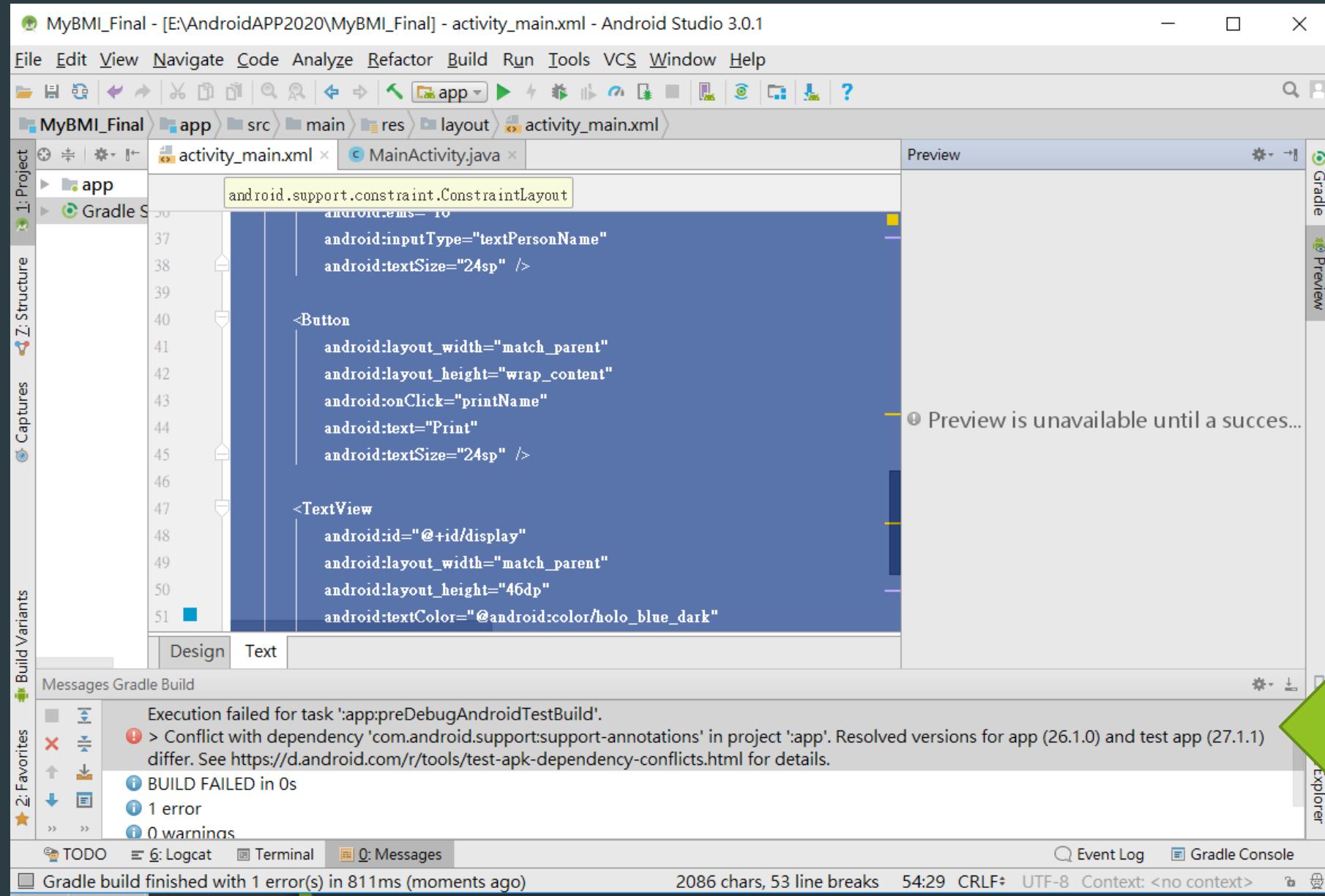
開啟剛剛解壓縮的Android範例 找到剛剛解壓縮的位置



開啟剛剛解壓縮的Android範例 選擇This Window

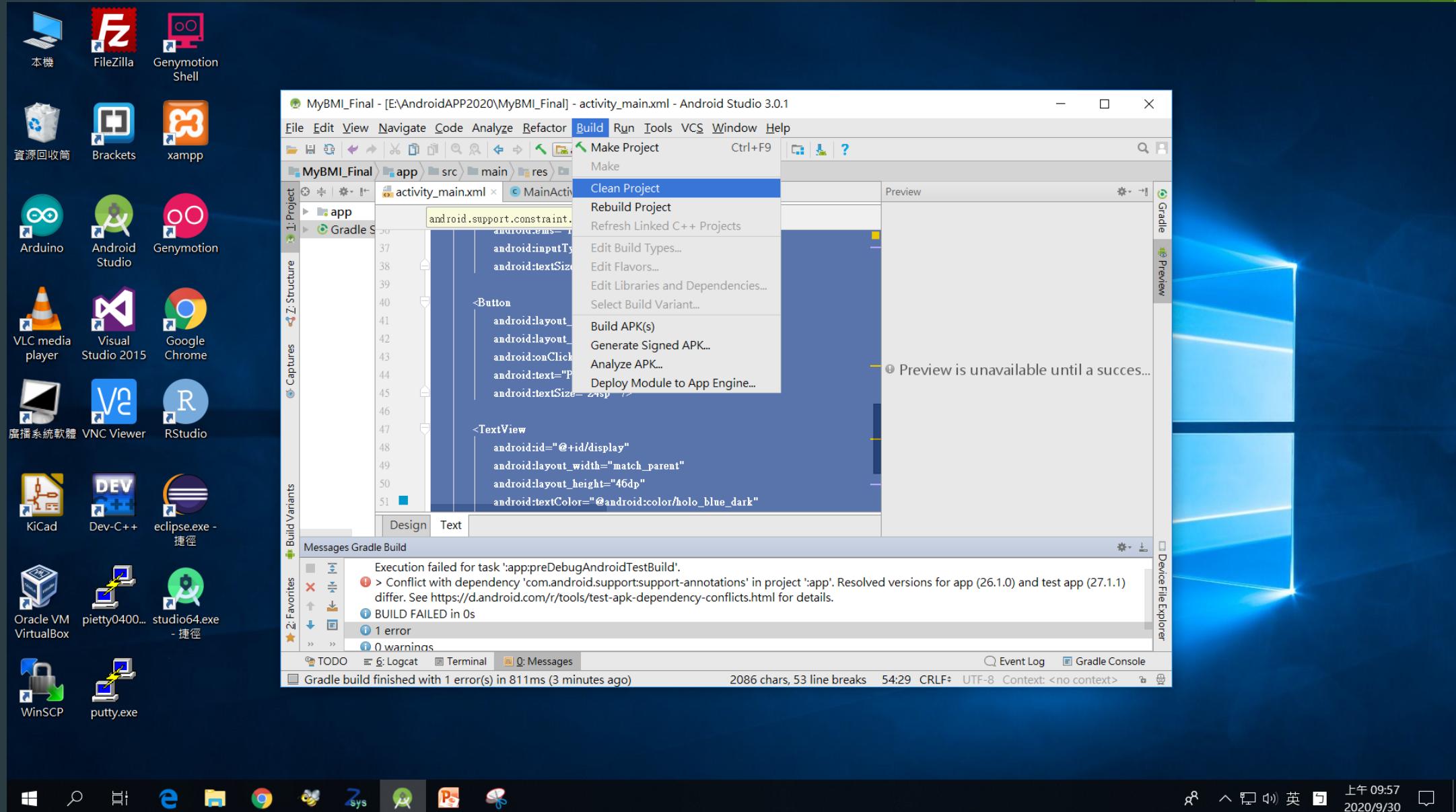


開啟專案後，會有下面顯示的錯誤需要排除

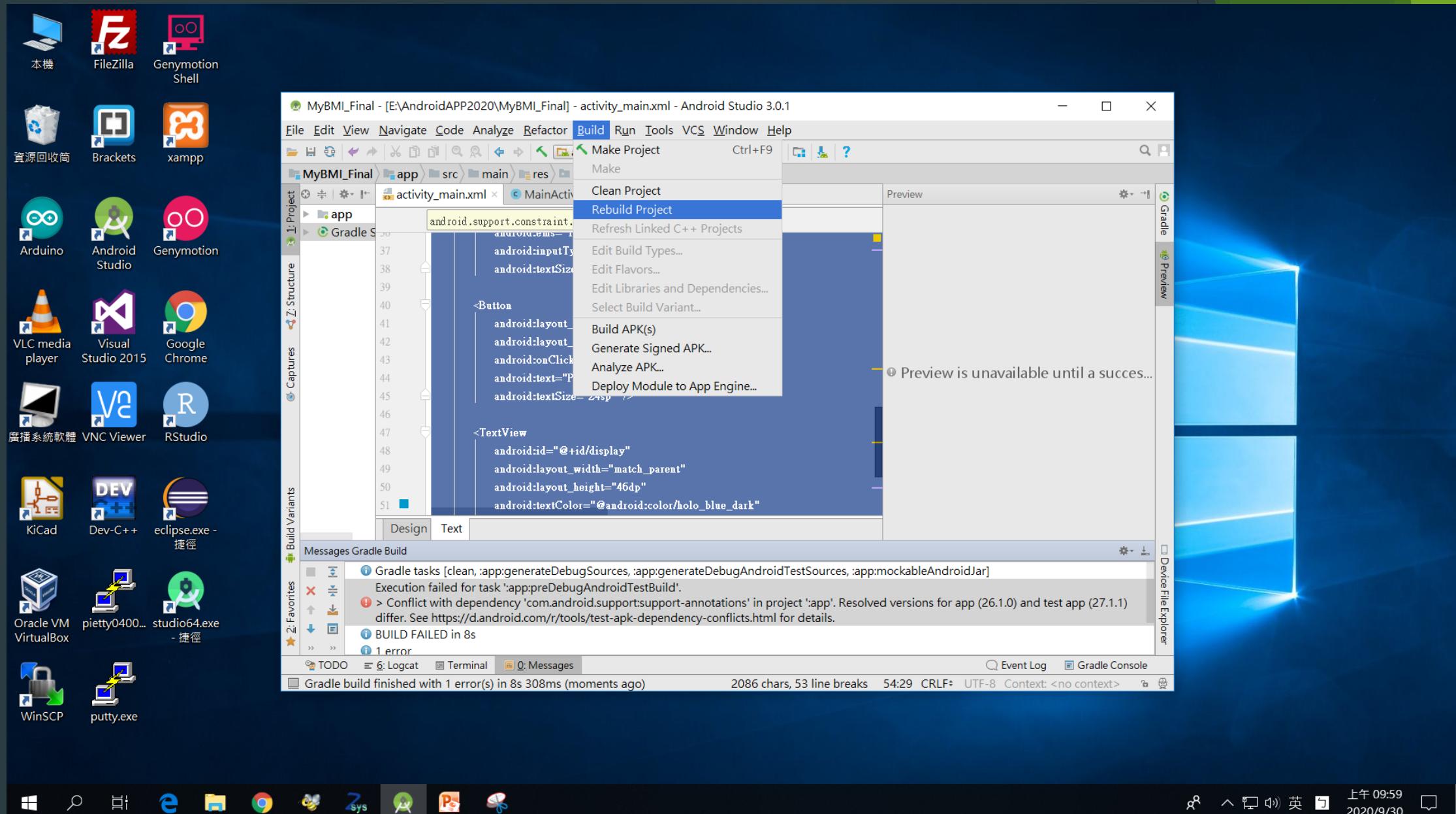


版本
問題

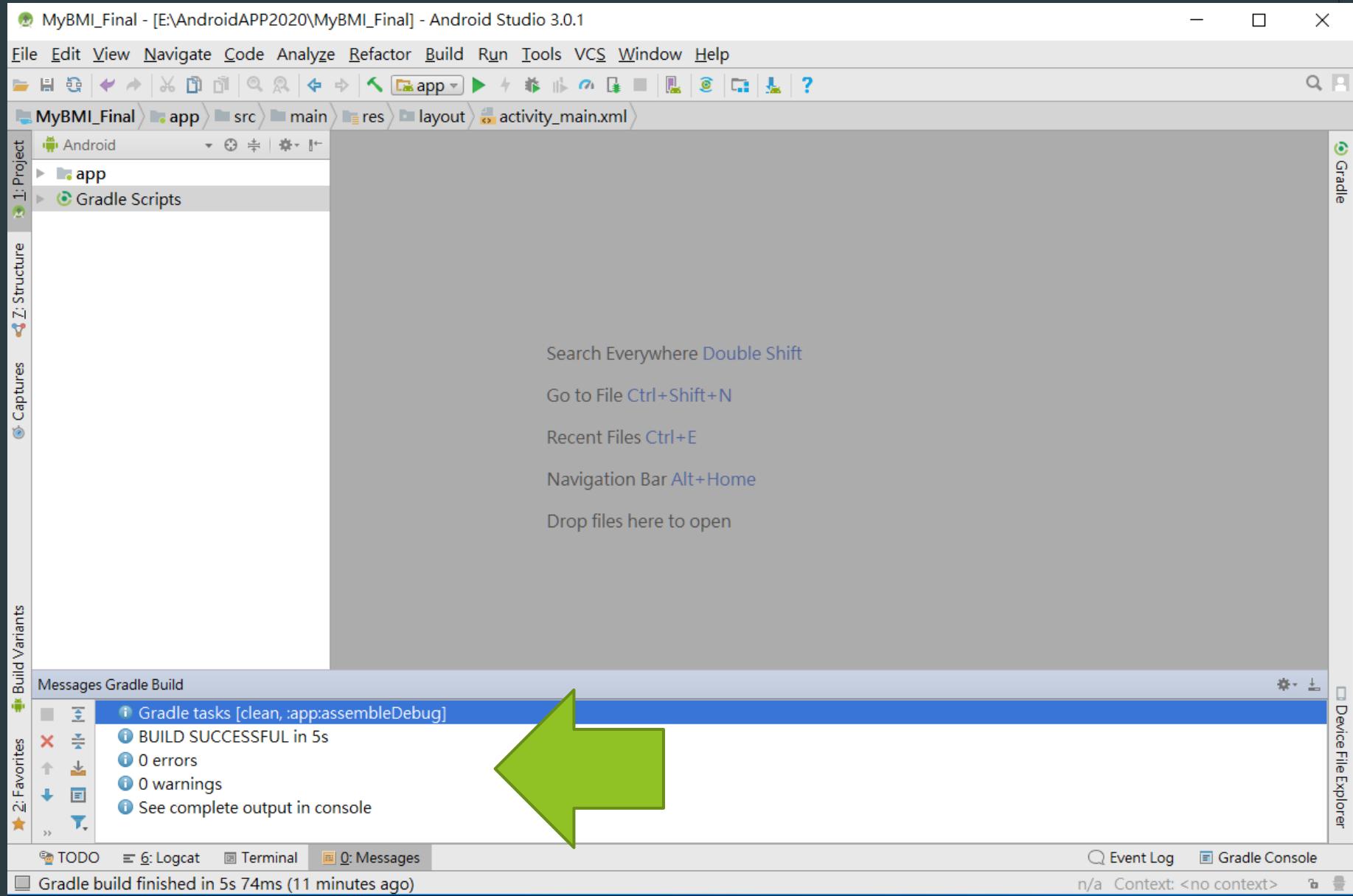
Build → Clean Project



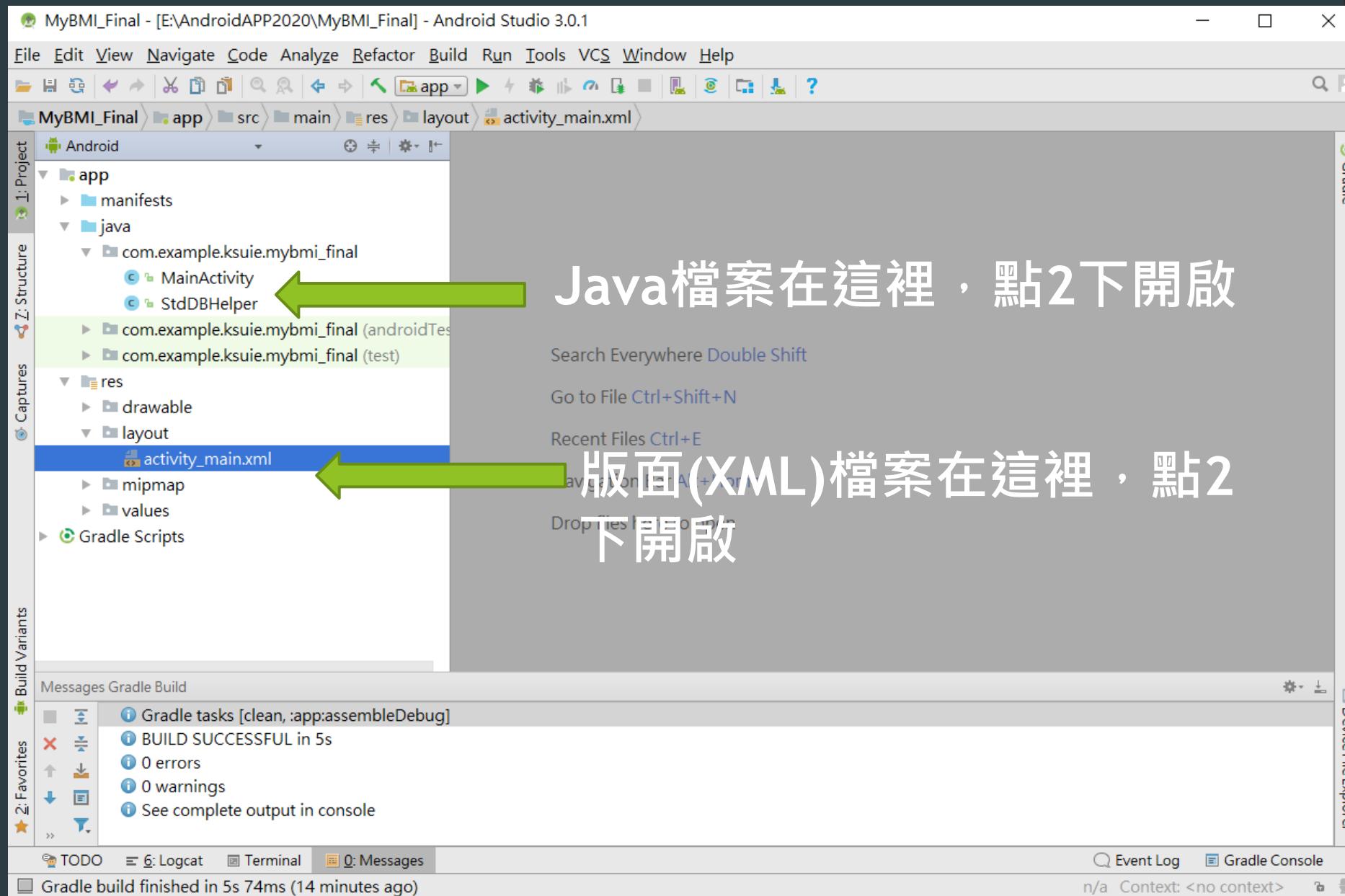
Build → Rebuild Project



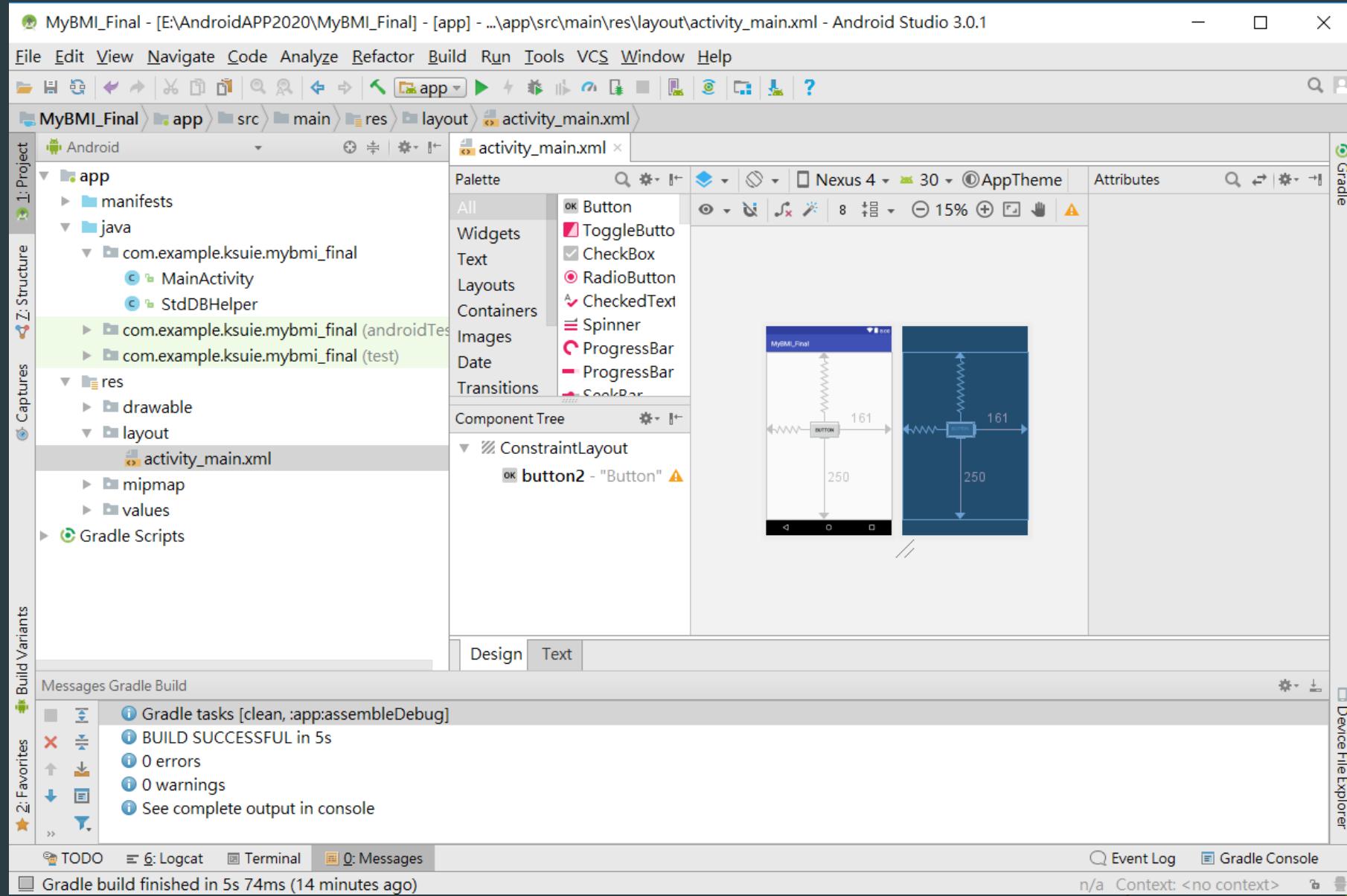
Build → Rebuild Project 錯誤排除



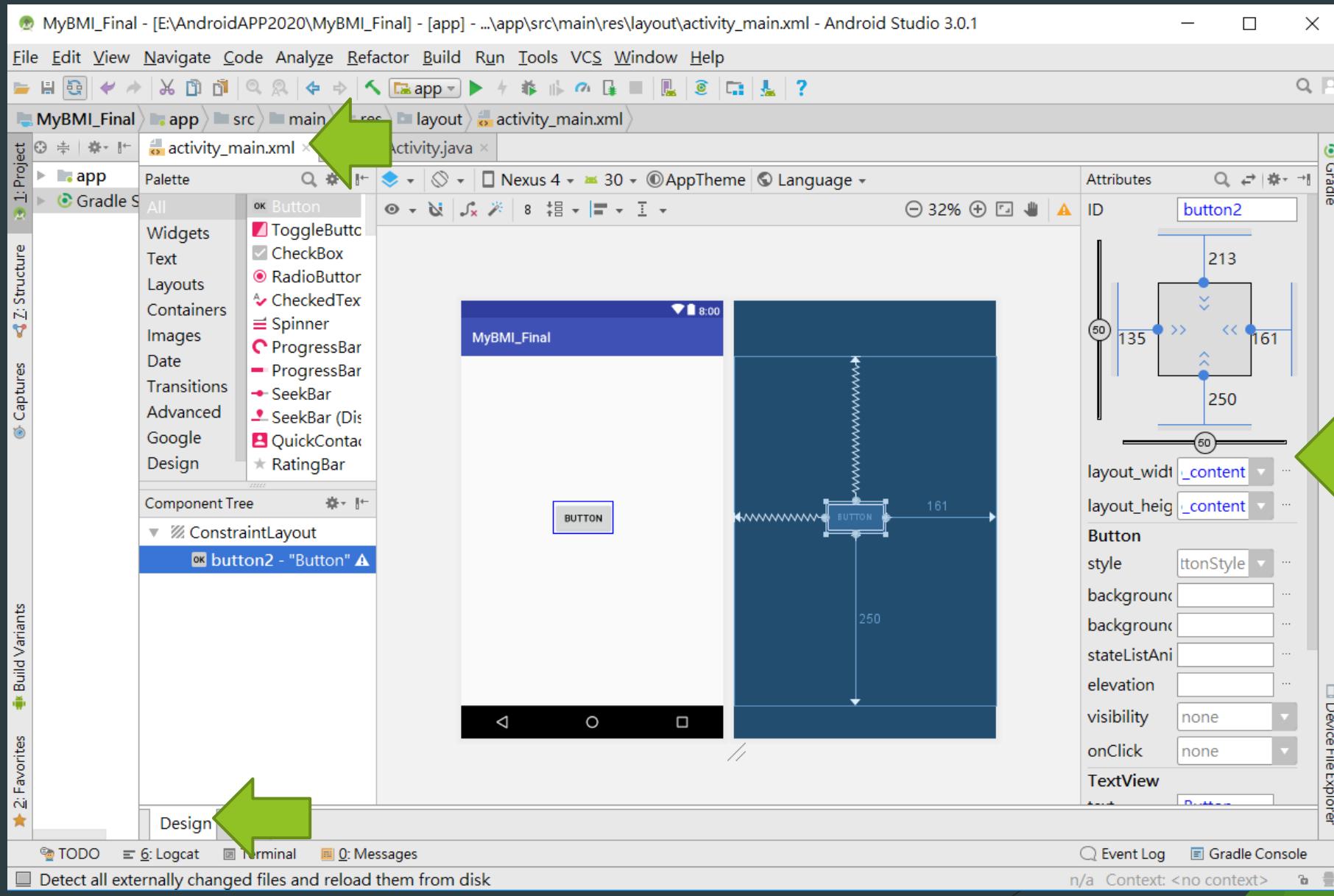
Build → Rebuild Project 錯誤排除



點2下activity_main.xml 開啟版面設計 Design模式



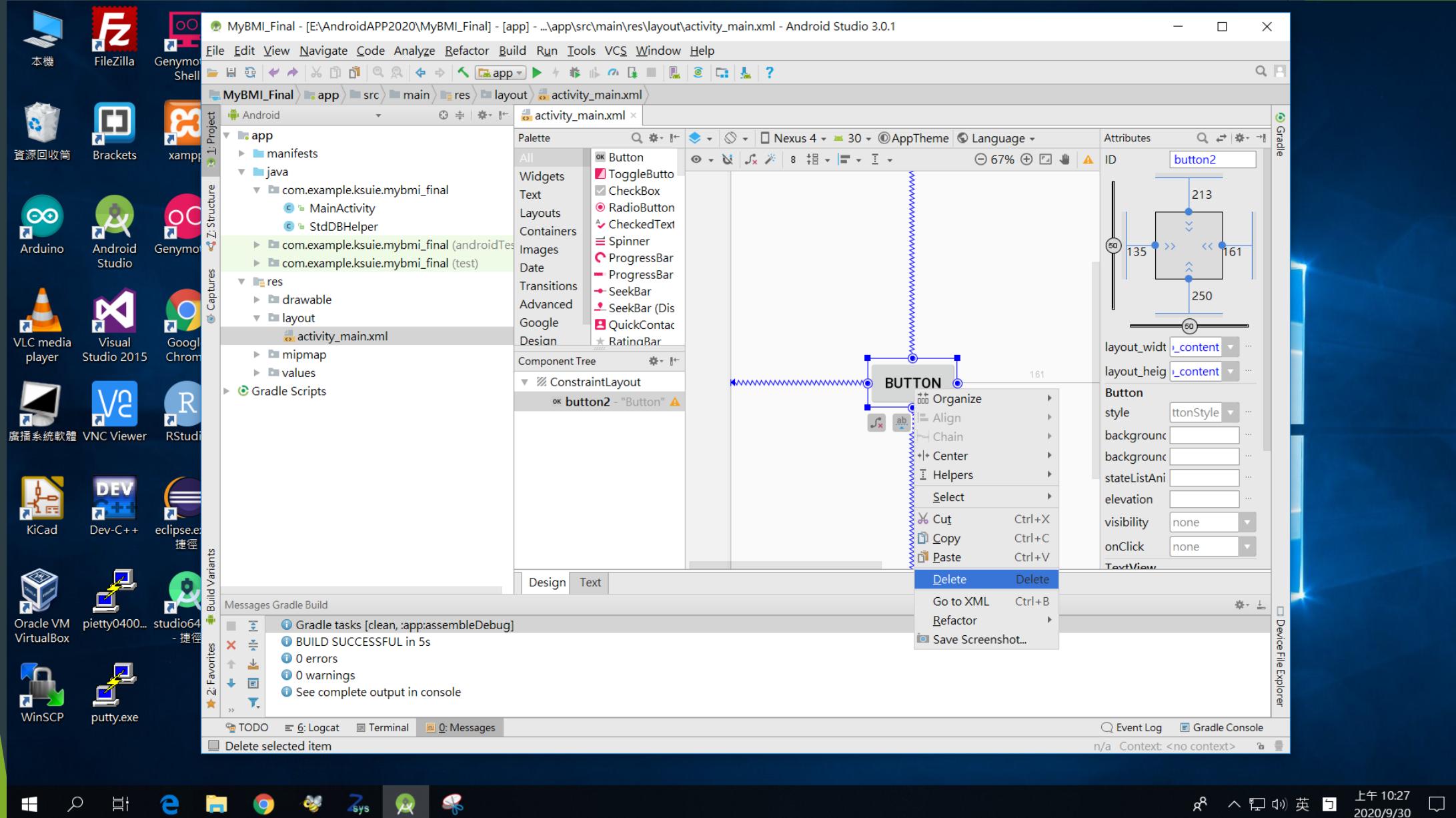
點2下activity_main.xml 開啟版面設計 Design模式



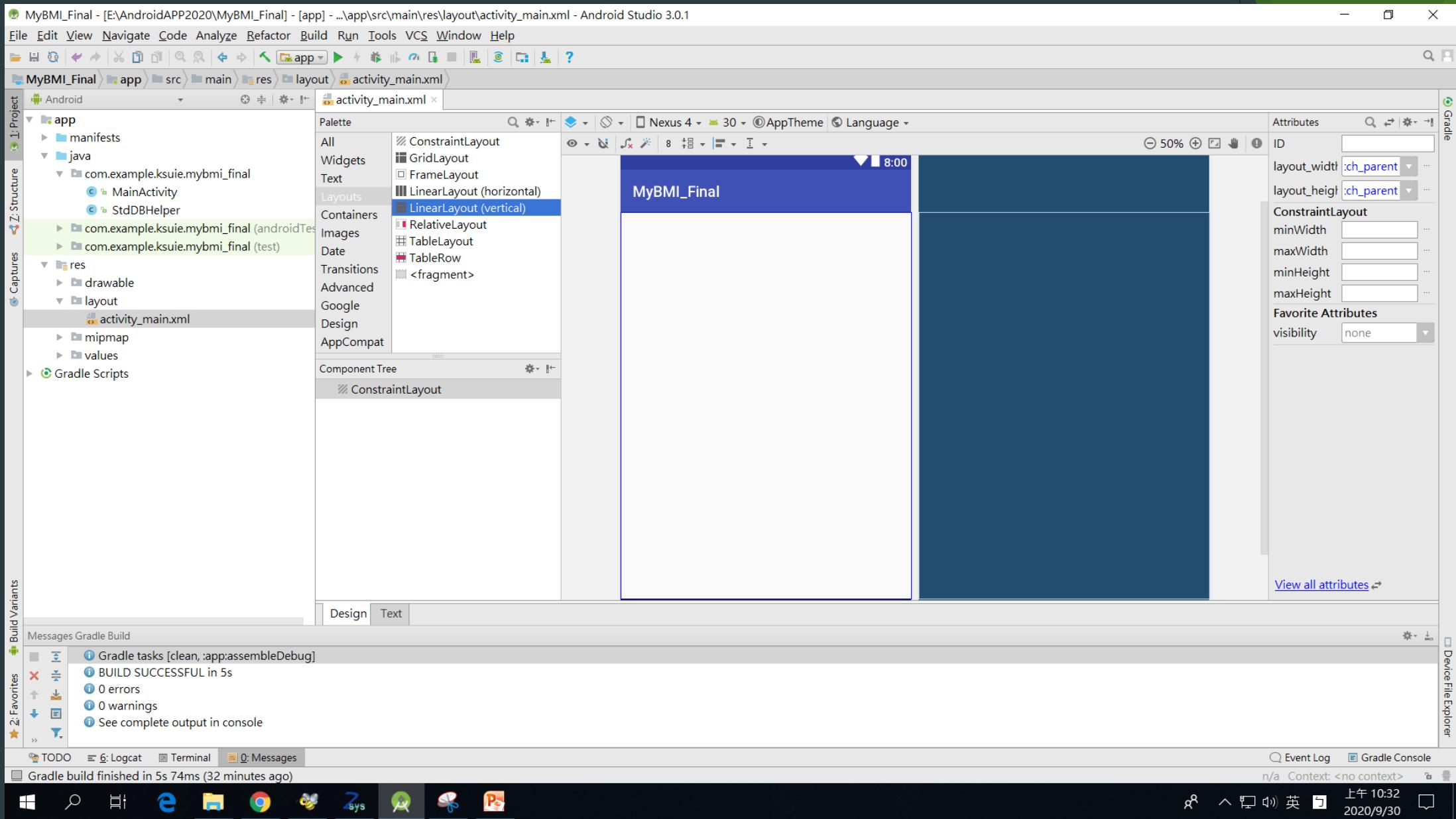
出現手機
的版面，
就是專案
開啟成功

開始小試一下拉元件

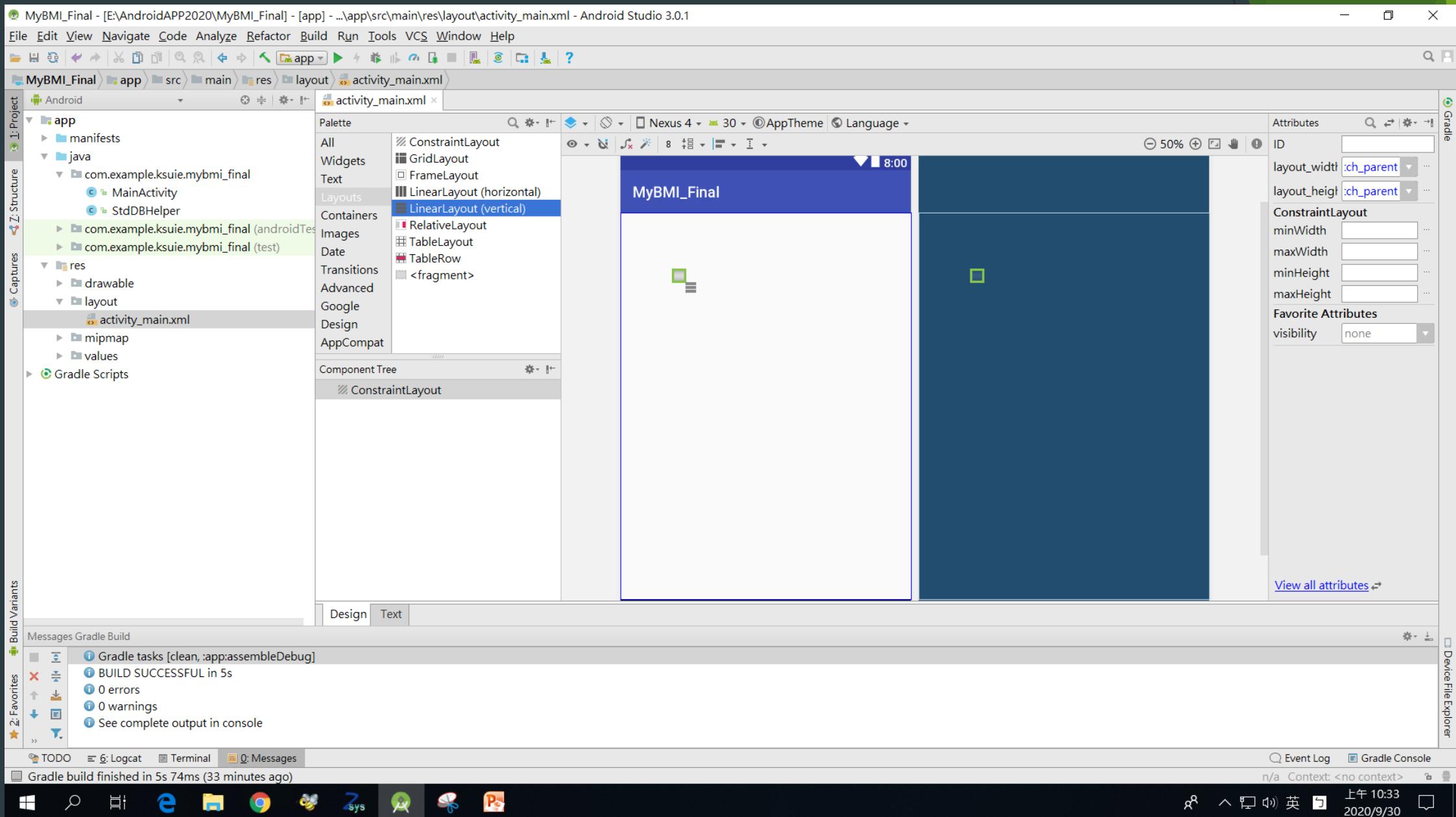
點滑鼠右鍵，點delete 先刪除原本的button



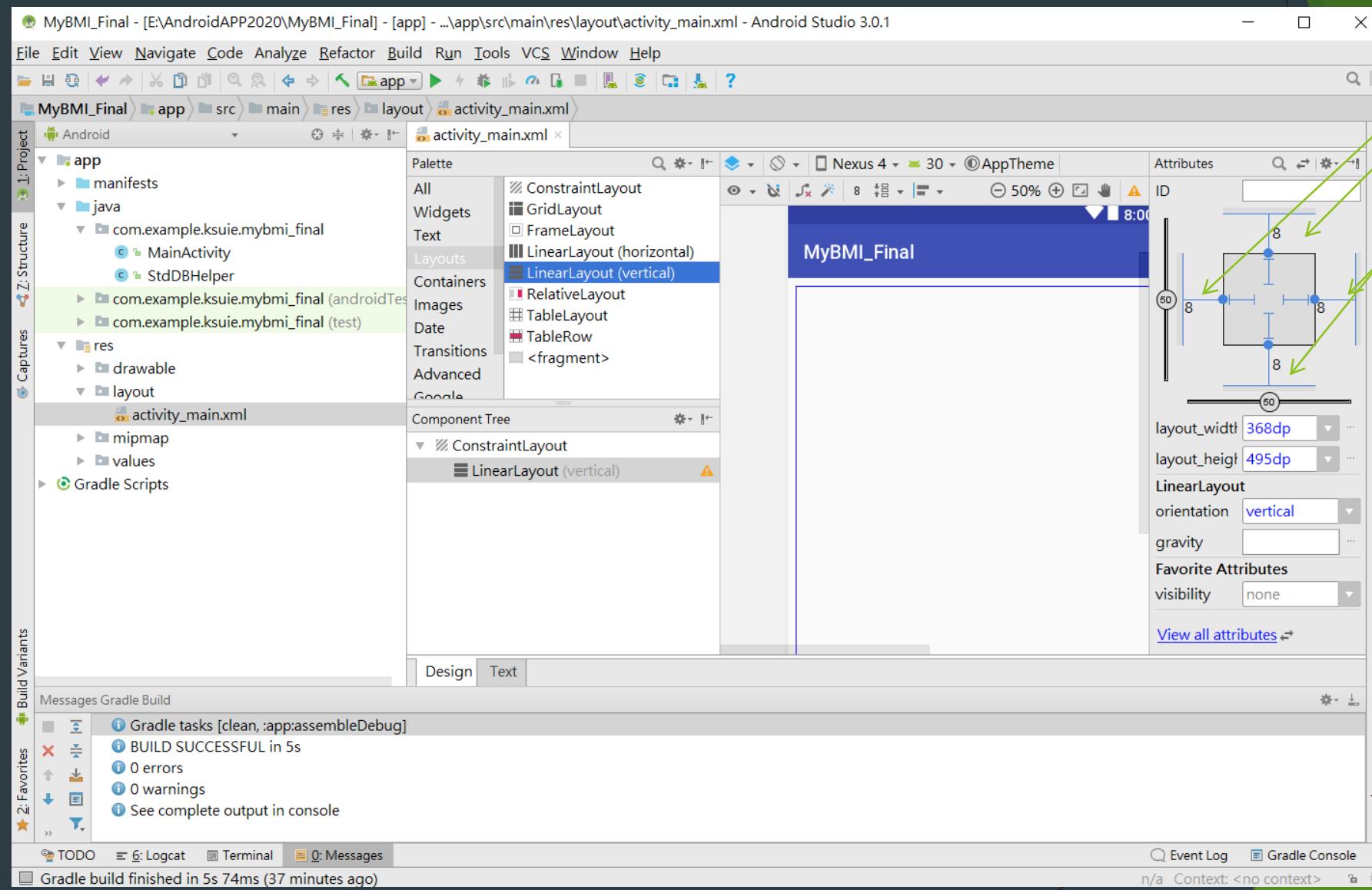
先選擇Layout → LinearLayout



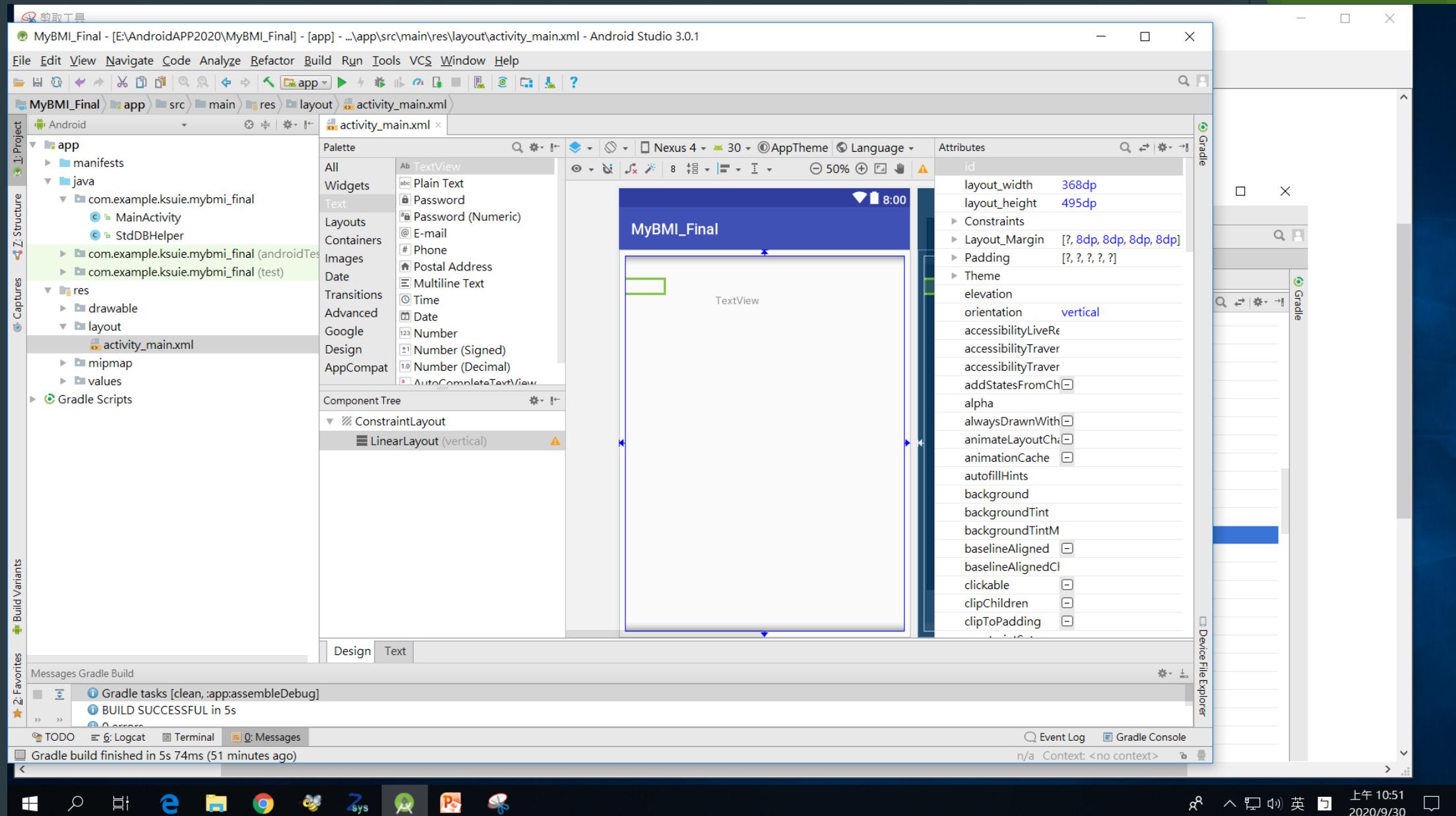
先選擇Layout → LinearLayout



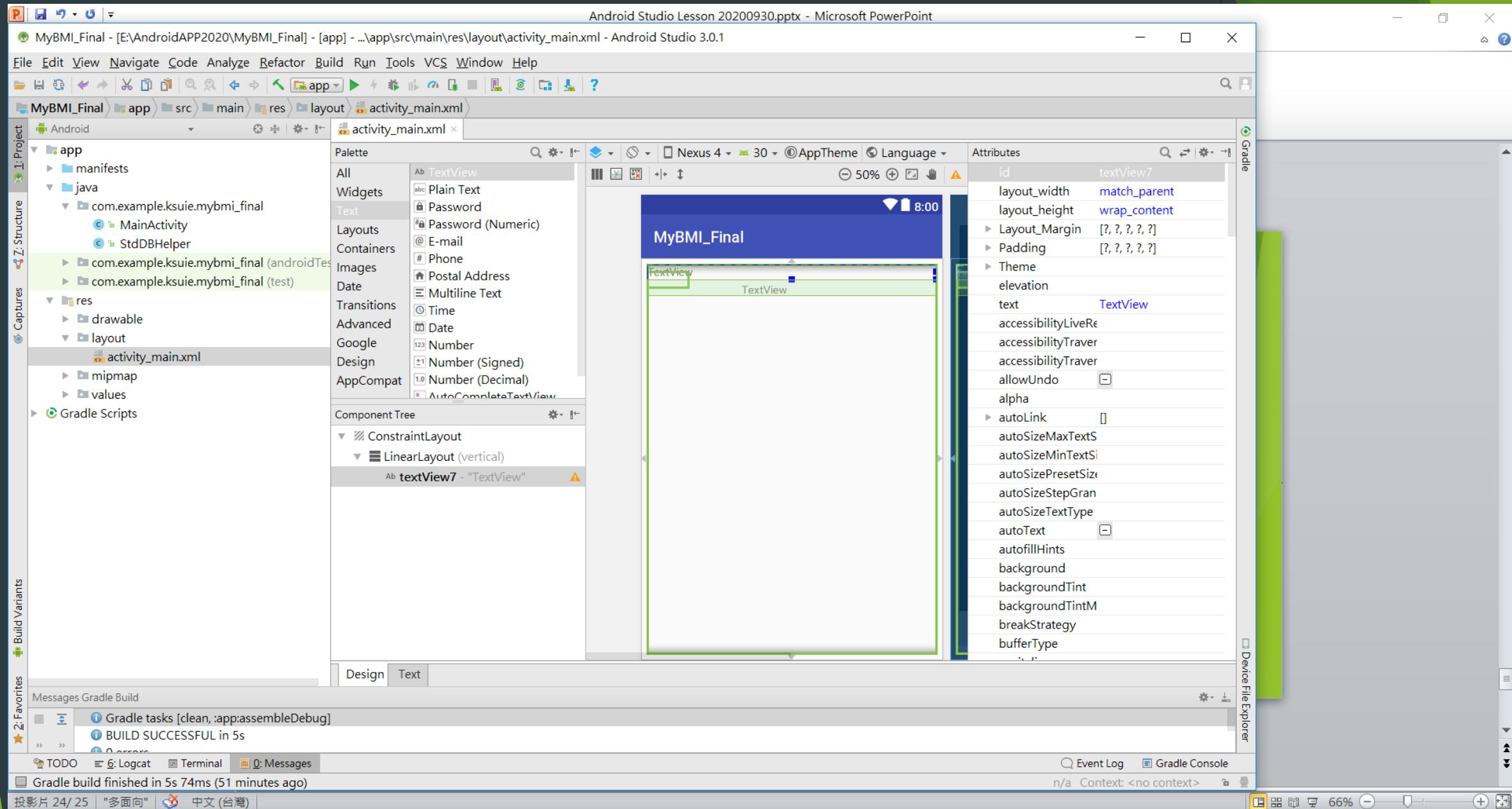
因為LinearLayout 是在ConstraintLayout中，所以要設定邊界



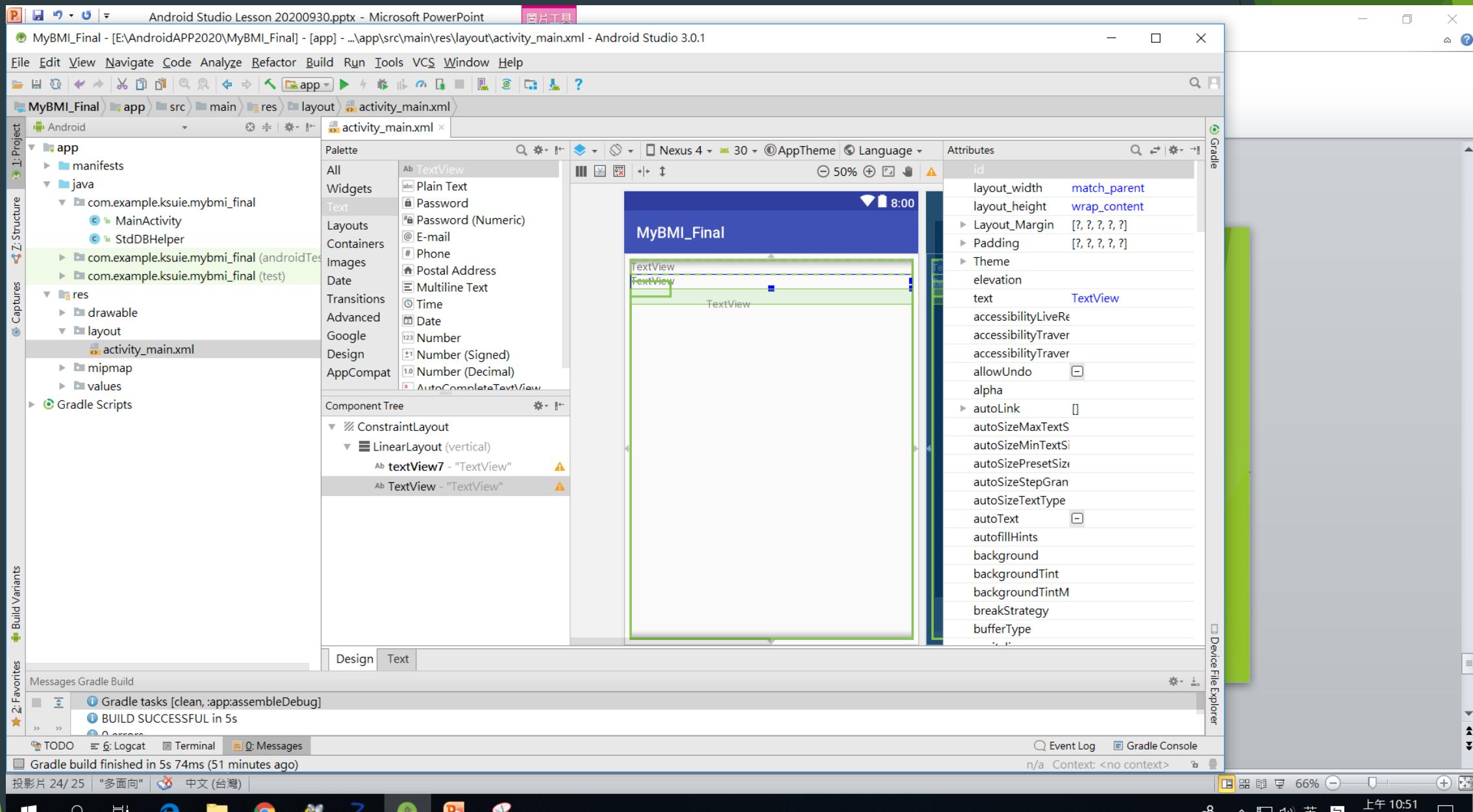
加入5個TextView



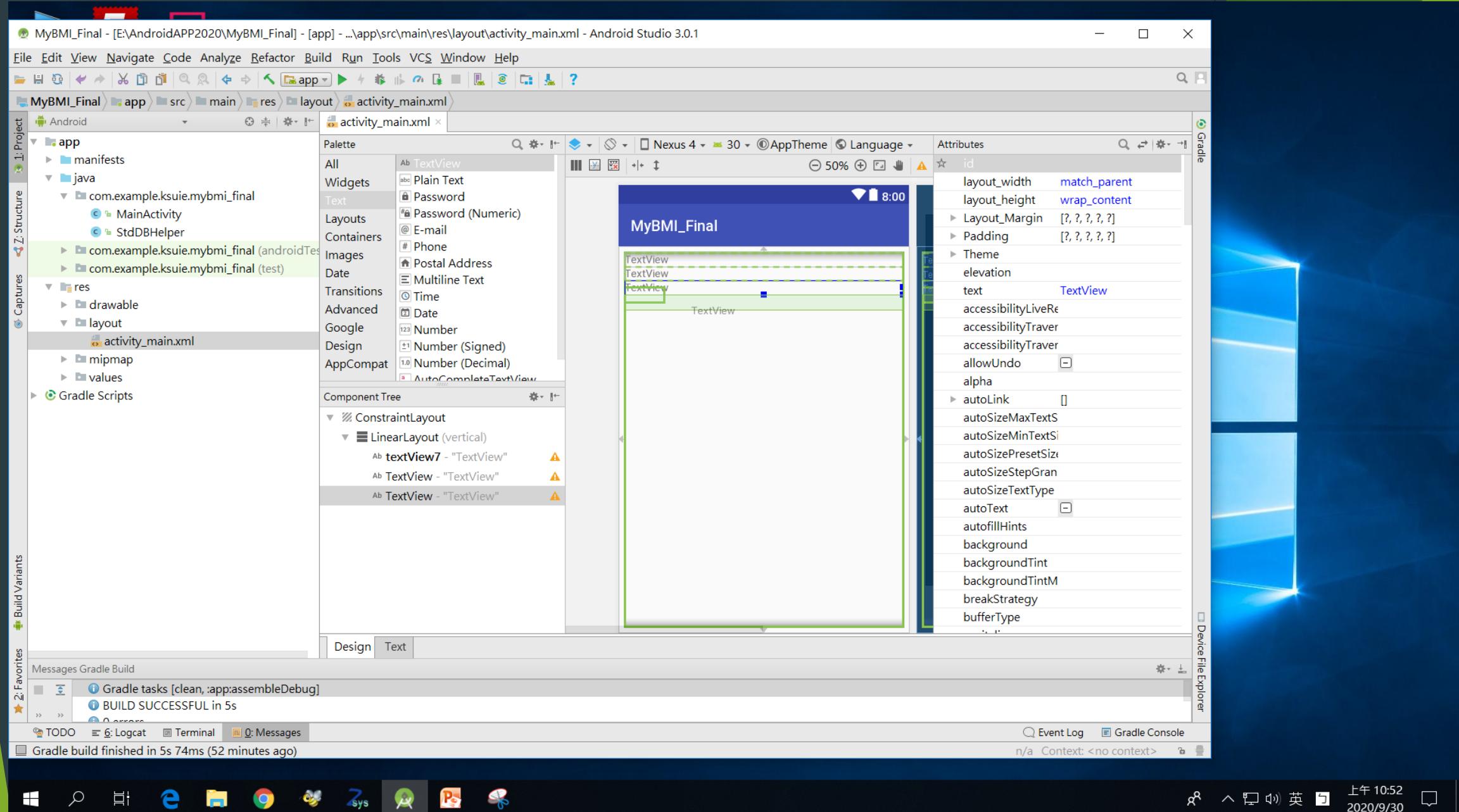
加入5個TextView



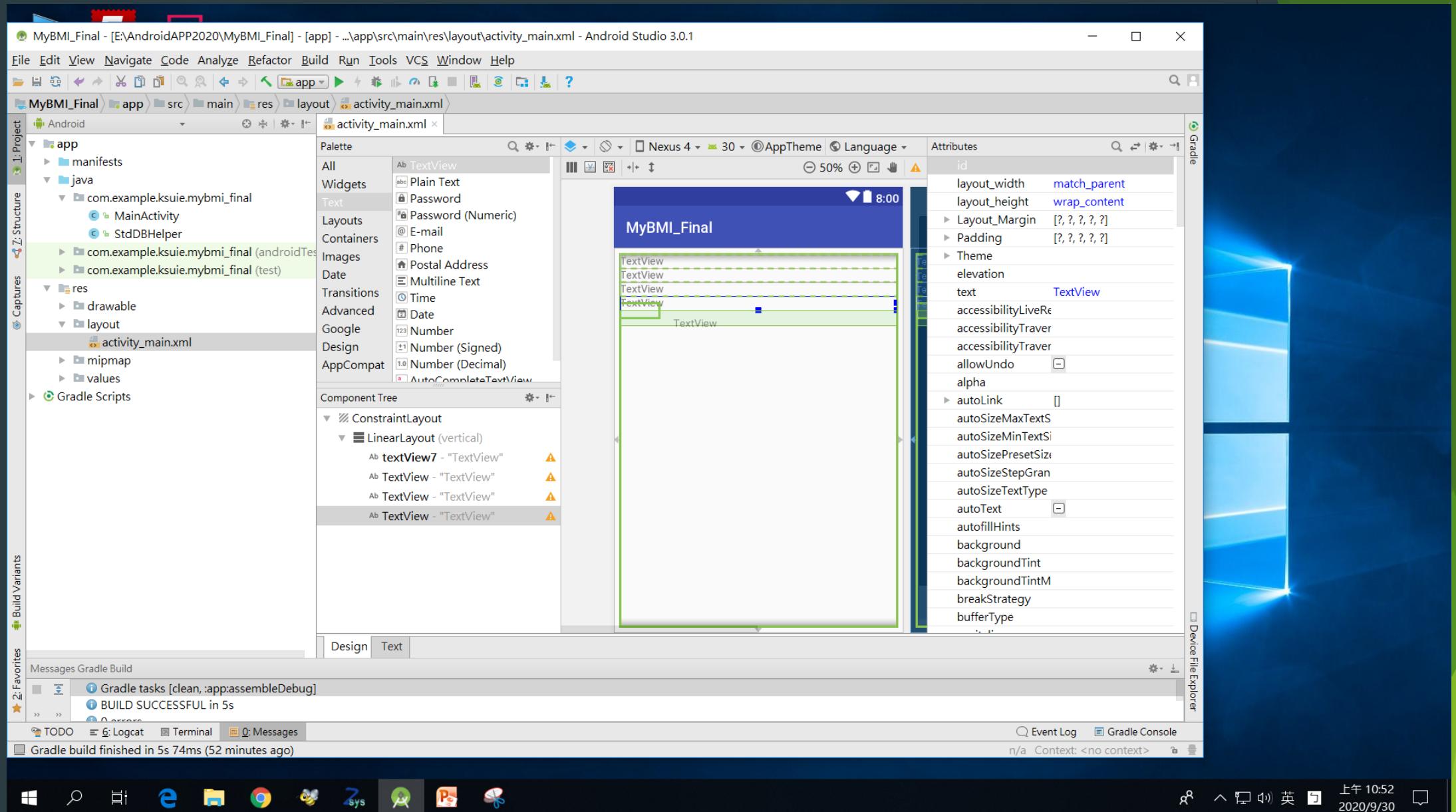
加入5個TextView



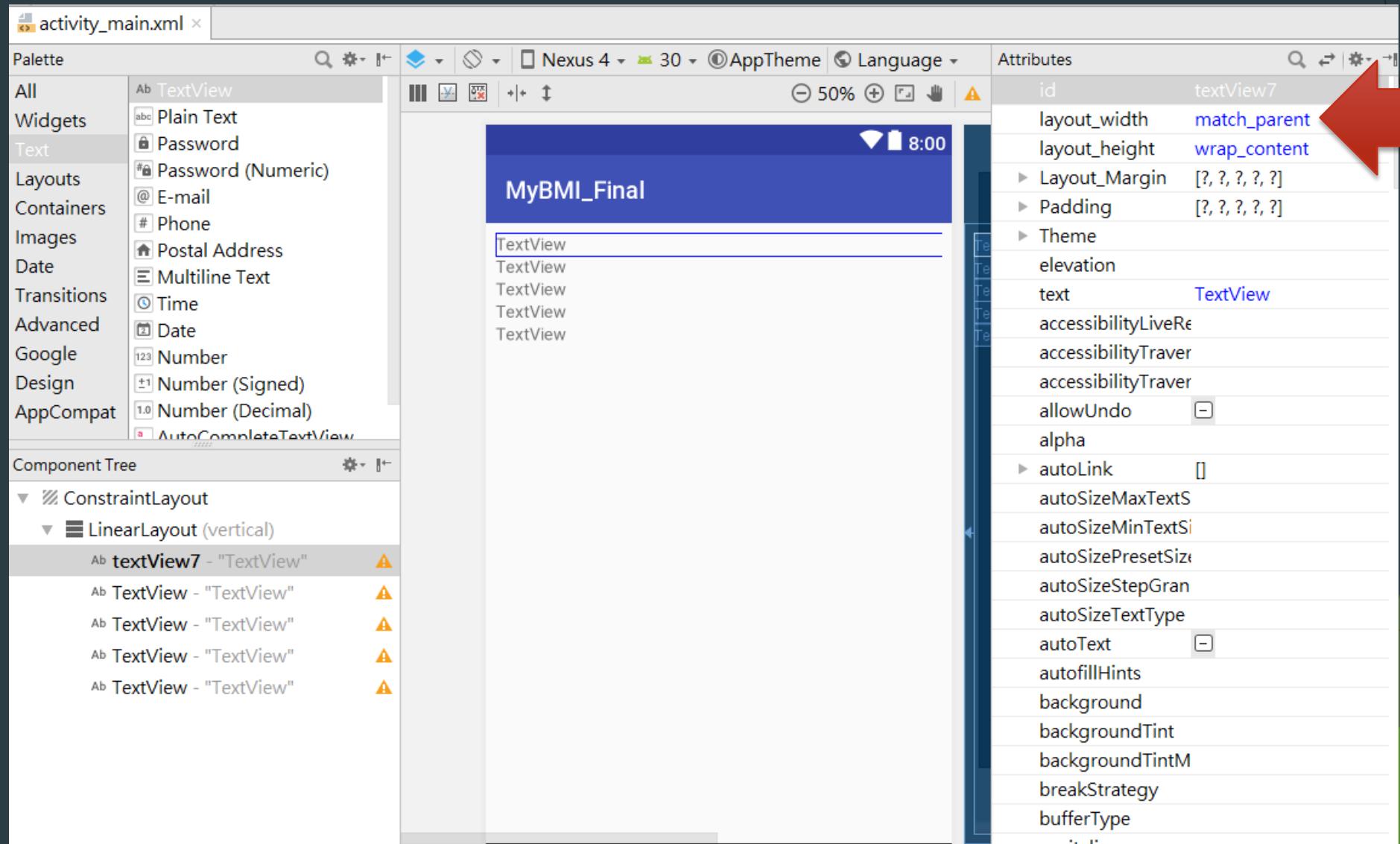
加入5個TextView



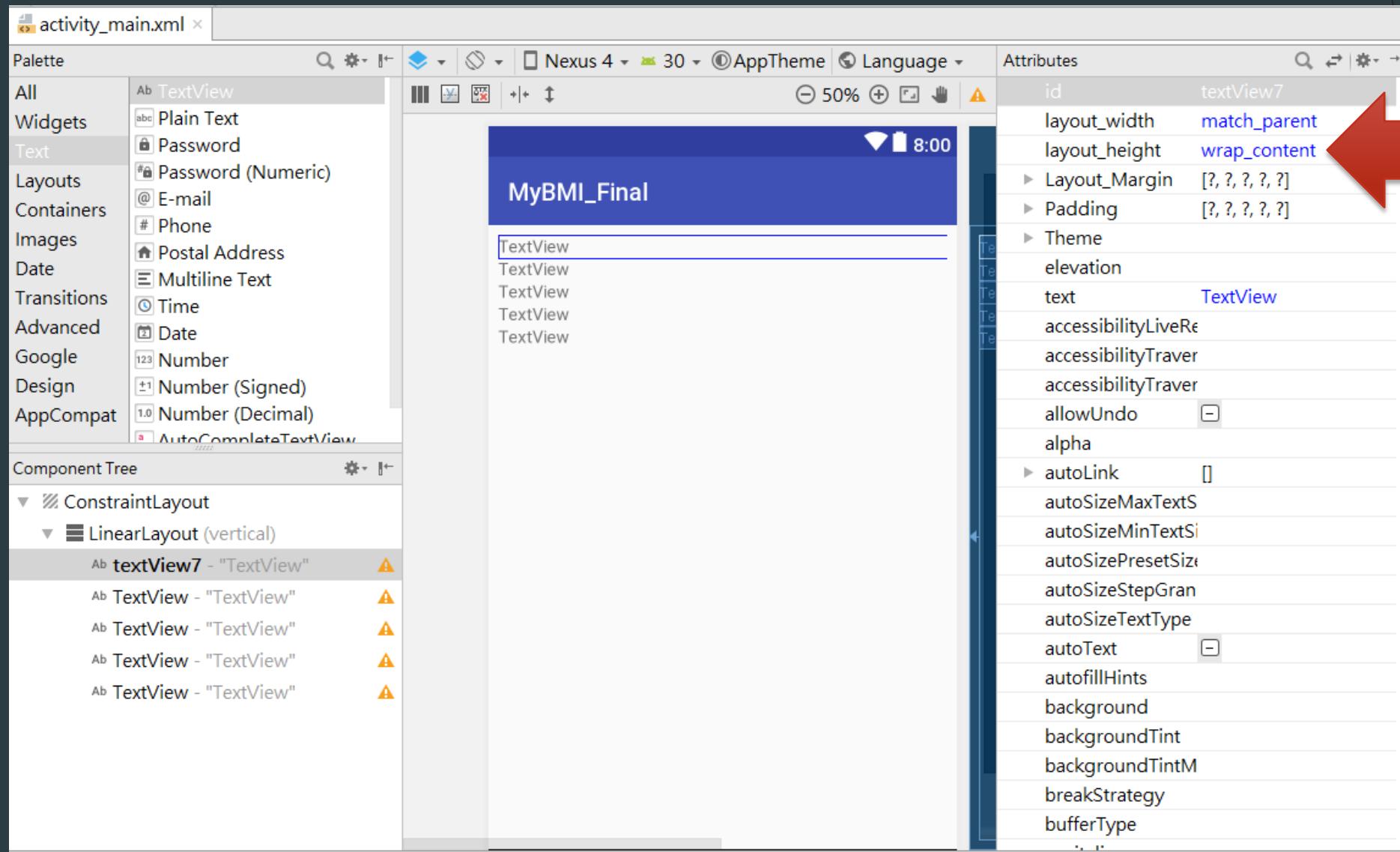
加入5個TextView



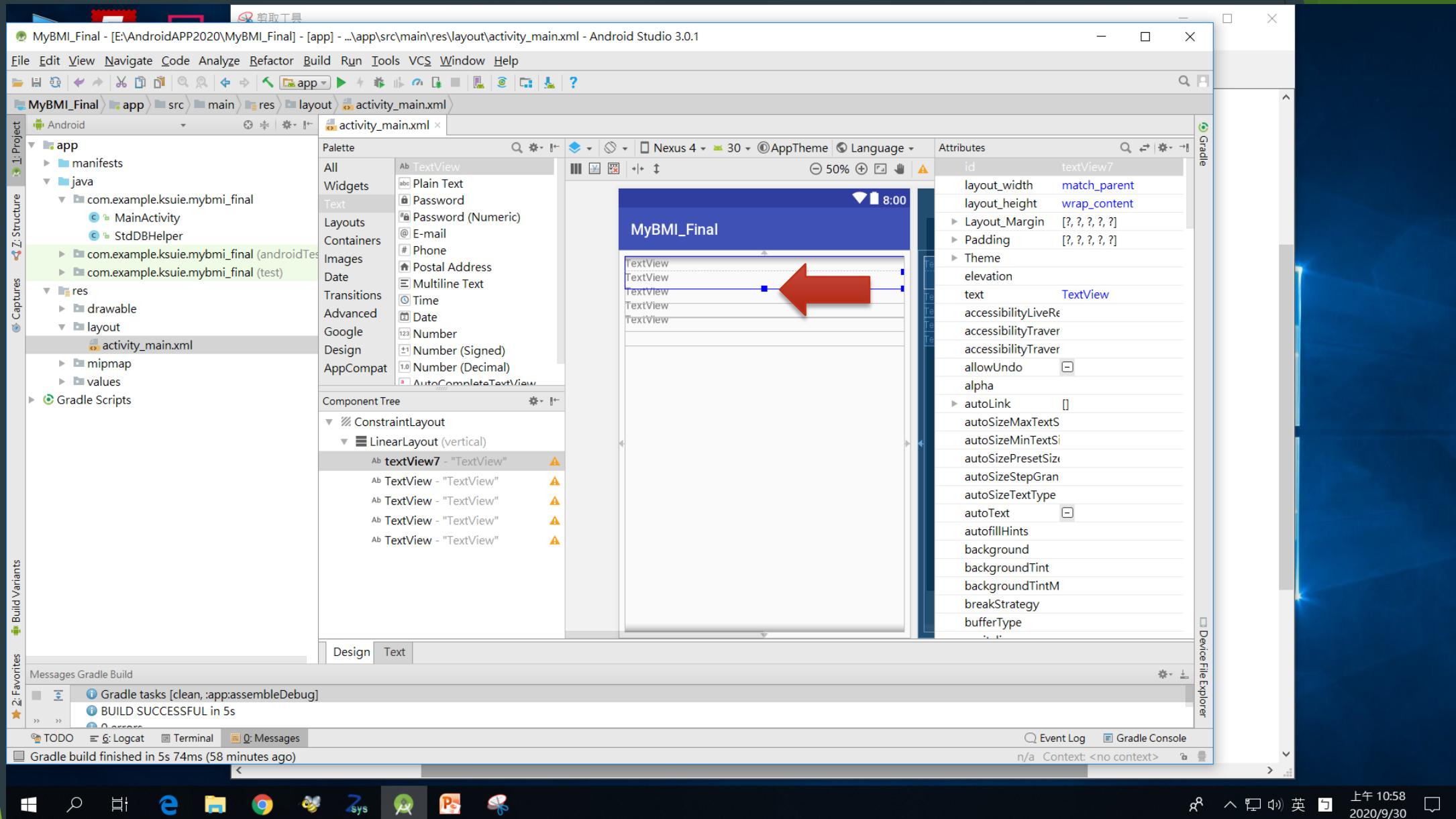
TextView元件的寬度 layout_width =
match_parent 與上一層(parent)的寬度相同



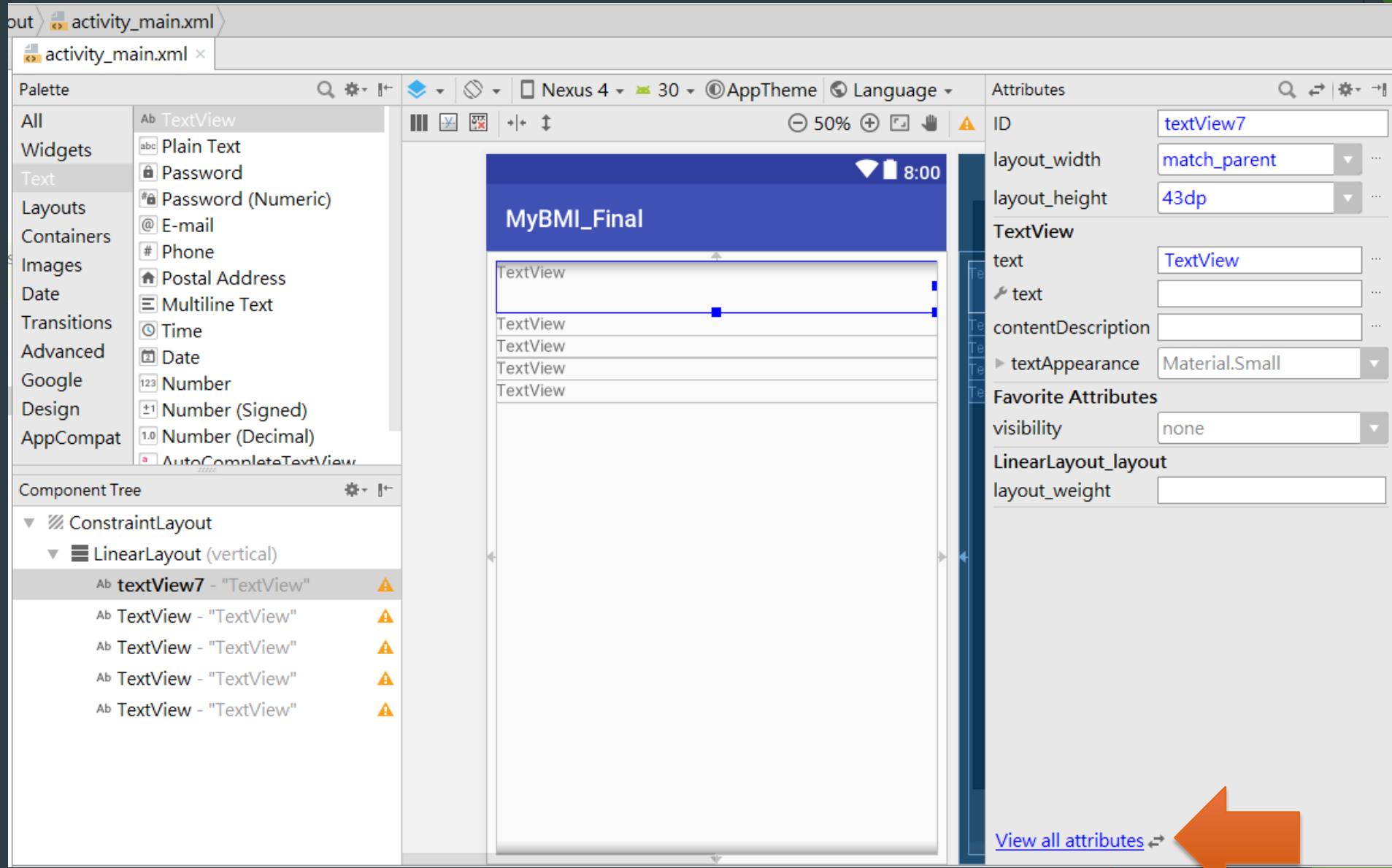
TextView元件的高度 layout_height = wrap_content 自動調整與內容(content的高度)相同



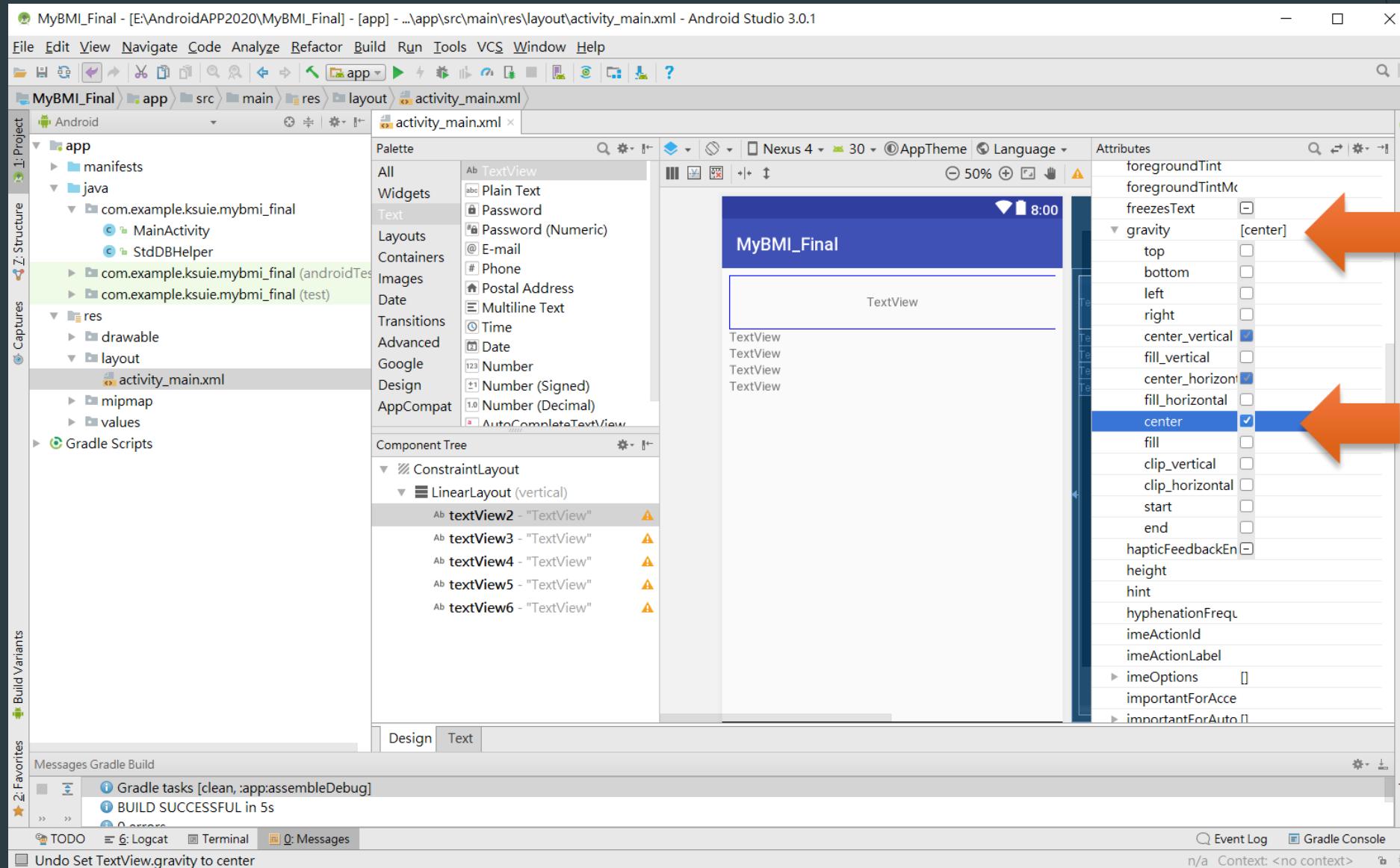
利用元件下方的藍色方塊 手動調整TextView元件的高度



文字置中 View all attribute → gravity → center



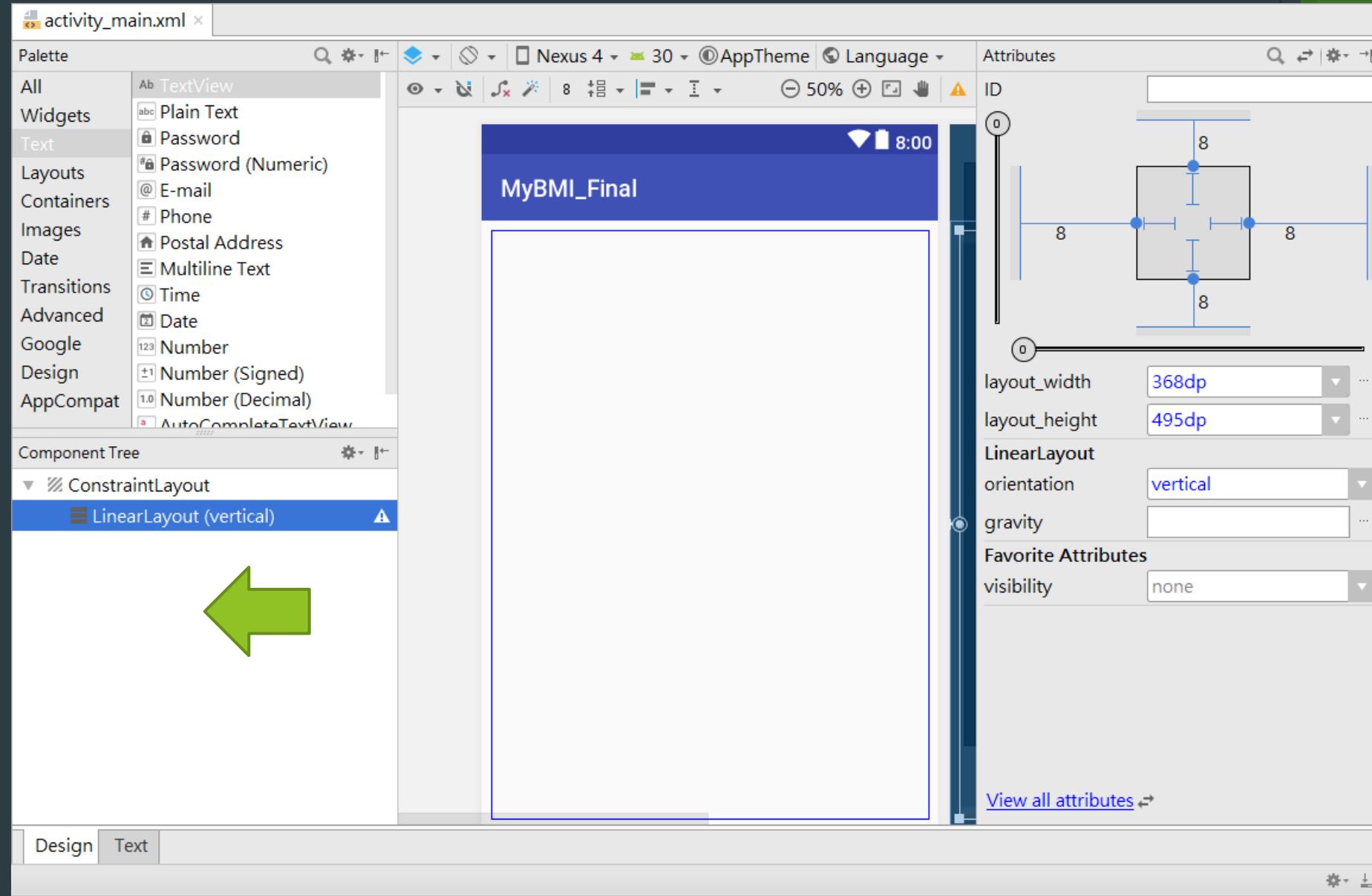
文字置中 View all attribute → gravity → center



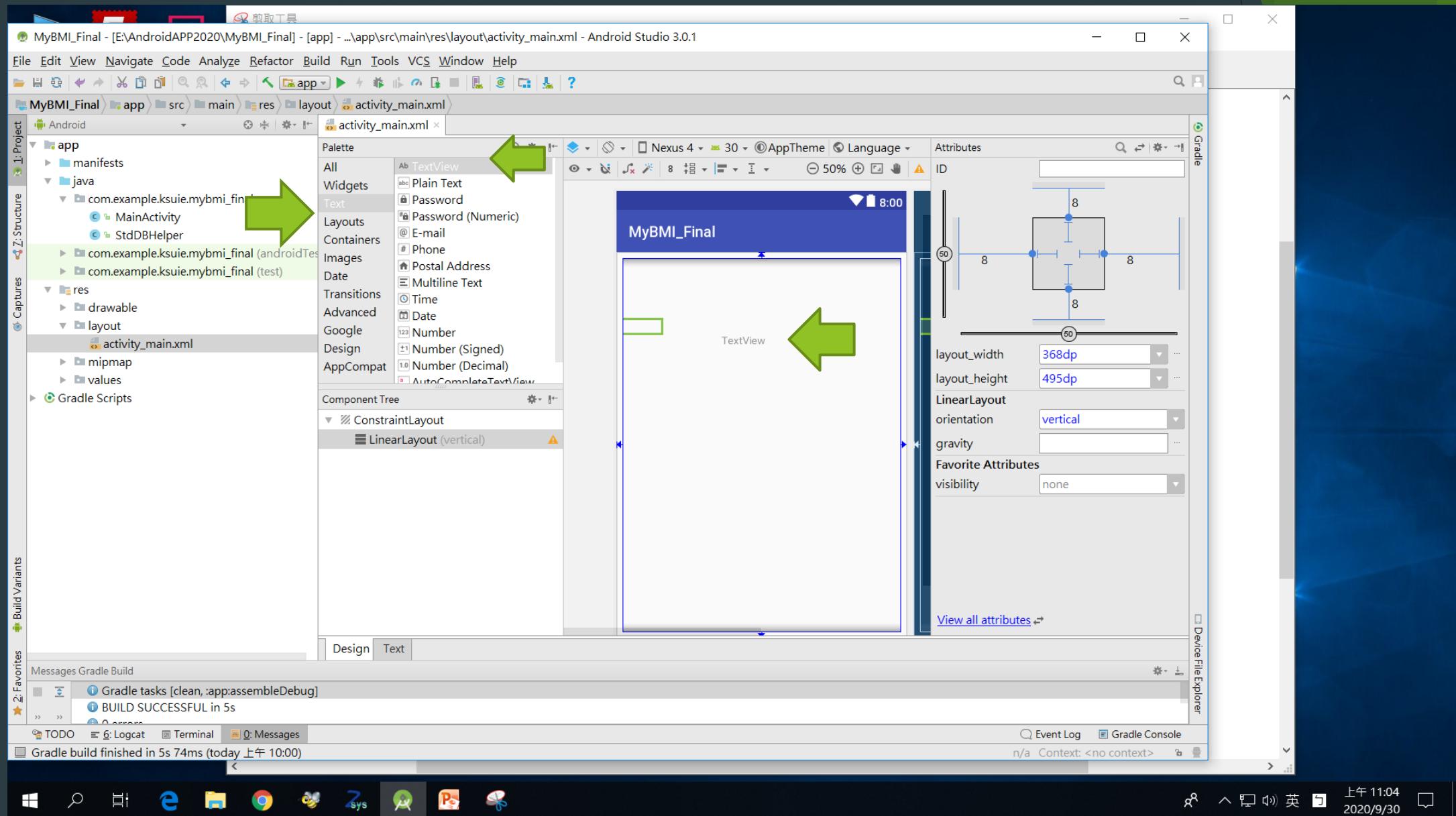
開始今天的APP進度

建立一個APP，有一個文字框，讓使用者可以輸入名字，下方有Button按鈕，按下按鈕後，輸入的名字可以顯示在APP下方的空間。

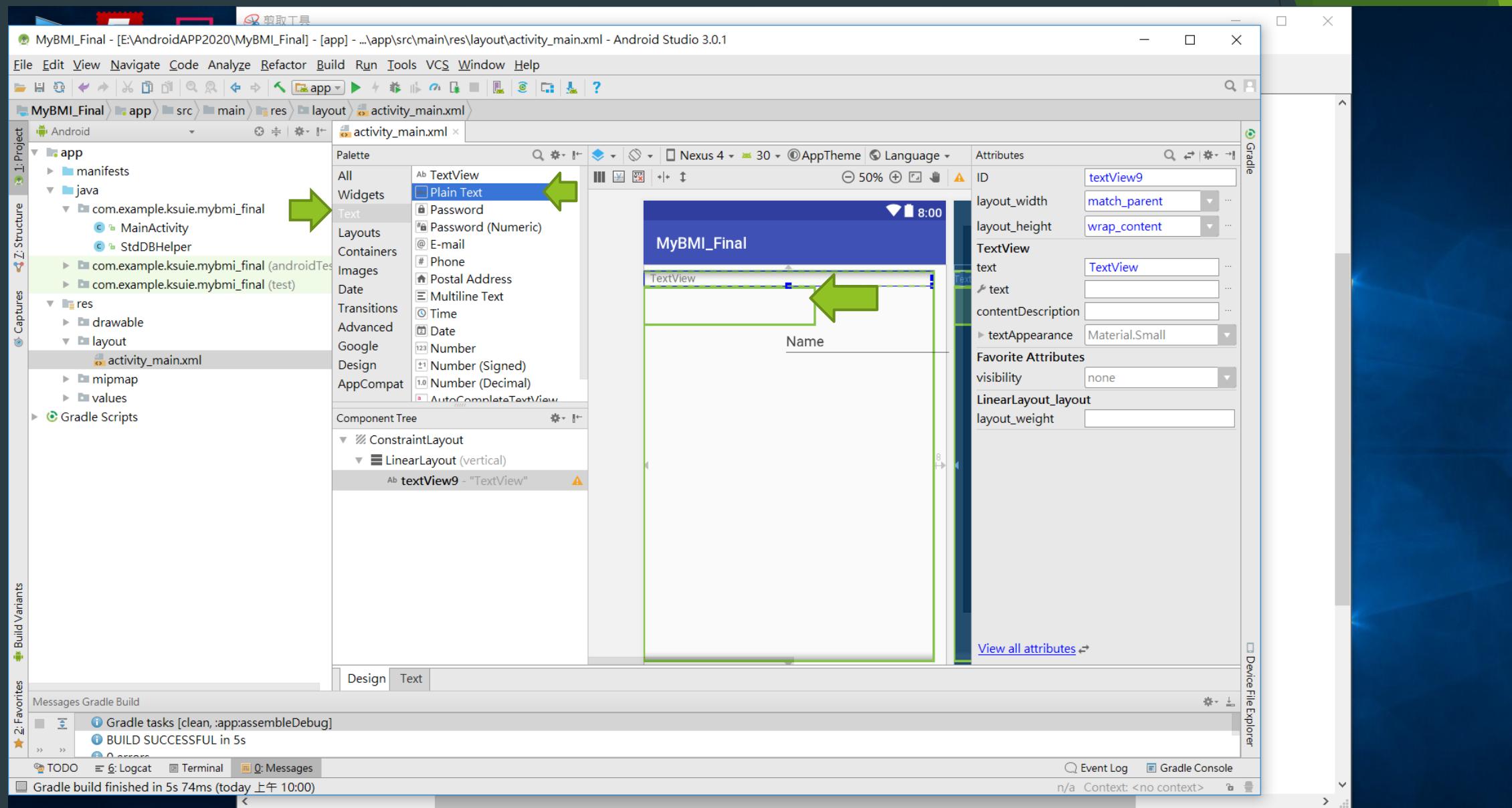
刪除剛剛練習新增的5個TextView 只保留LinearLayout



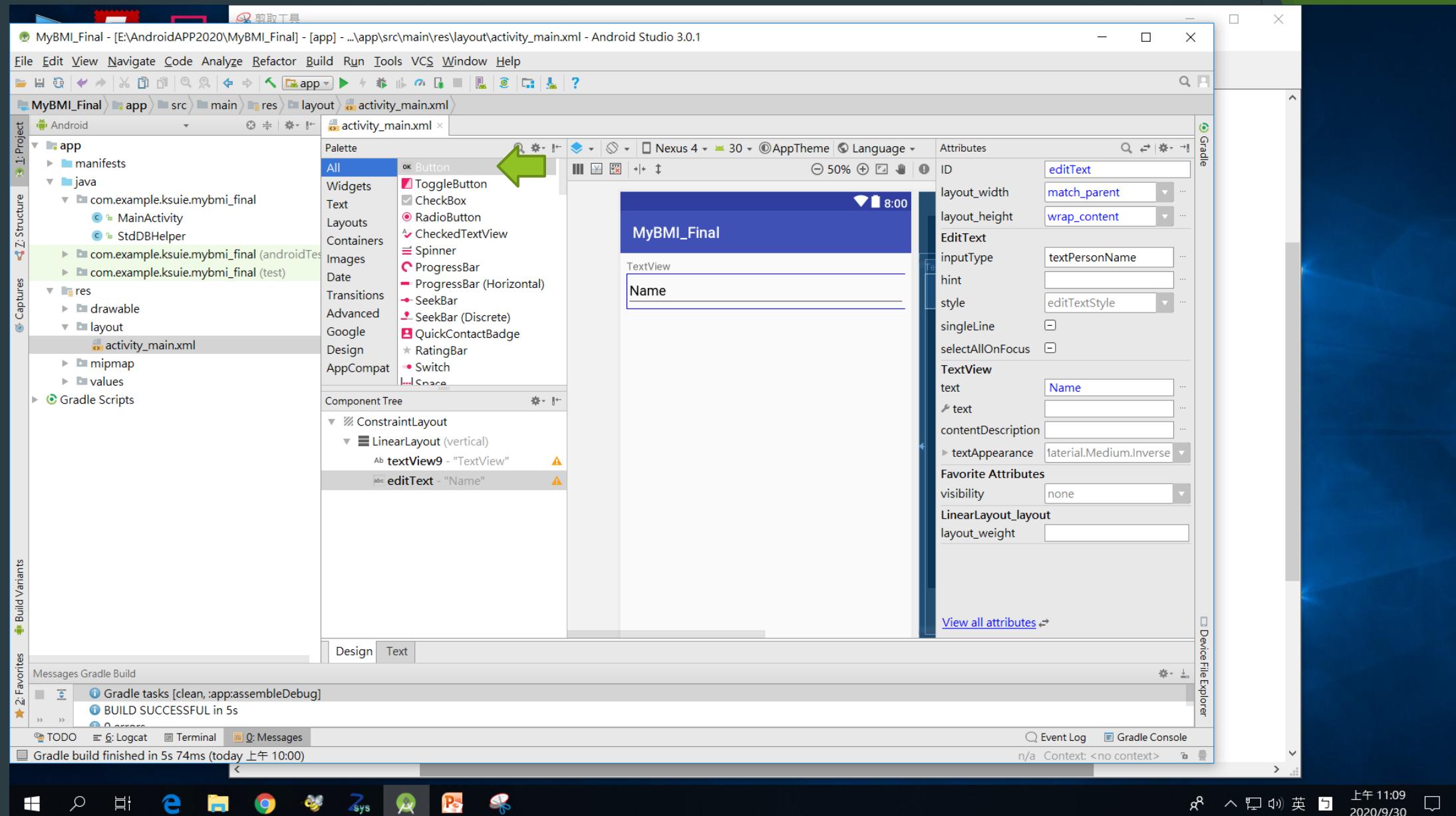
在Text -> 選 TextView (文字標籤) -> 拖拉到指定位置



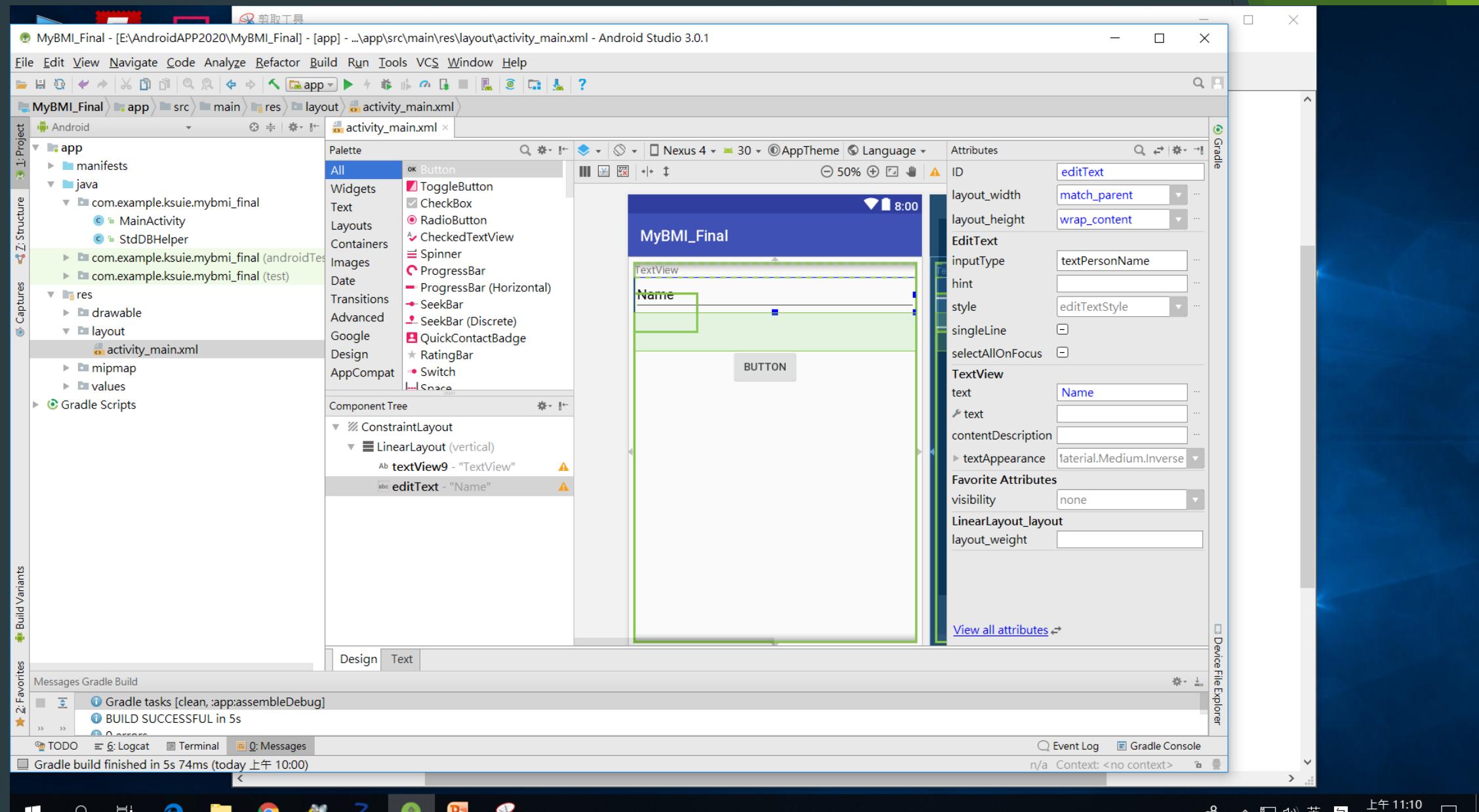
在Text -> 選Plain Text (輸入文字框) -> 拖拉到指定位置



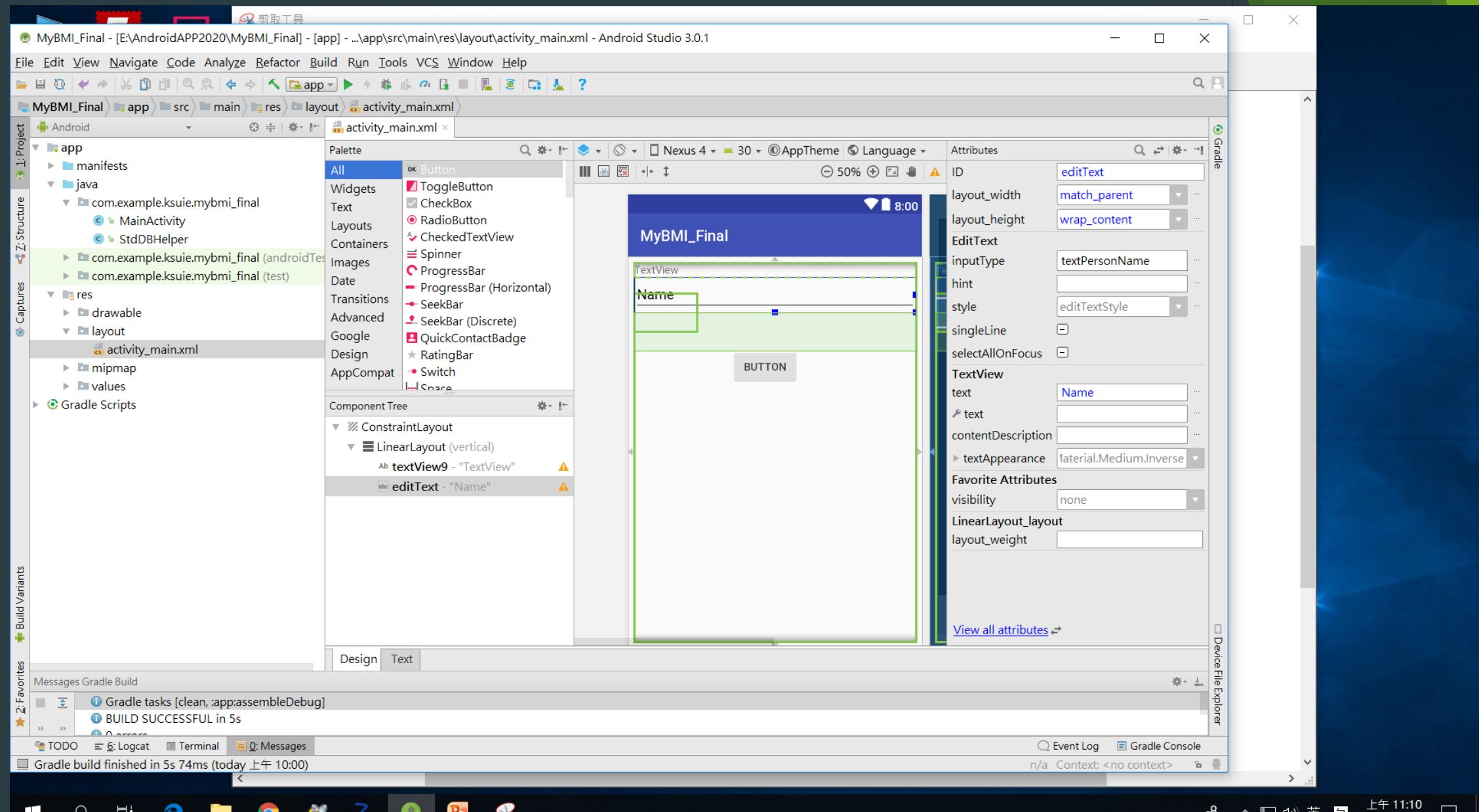
在All分類 -> 選Button元件(按鈕) -> 拖拉到指定位置

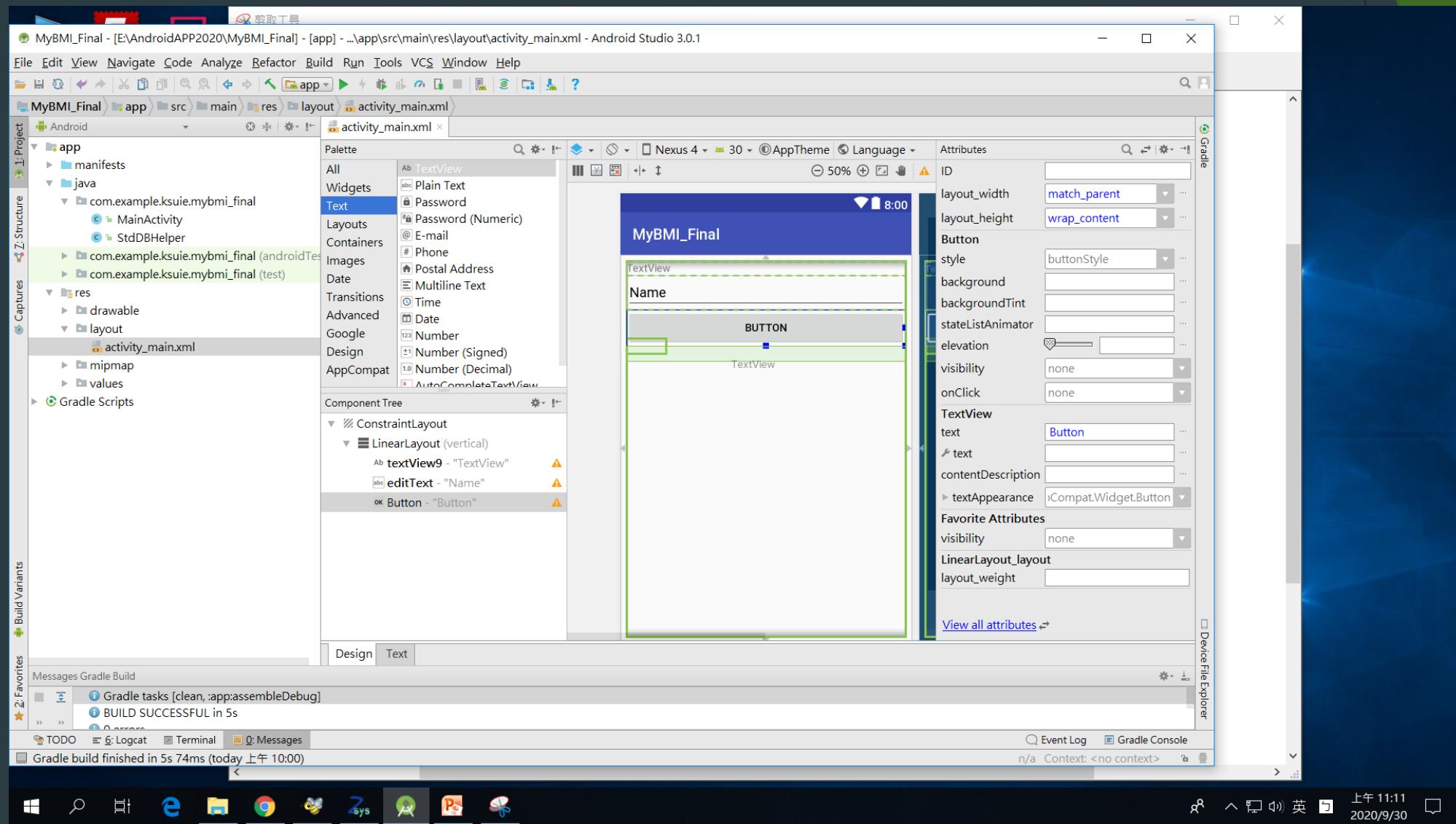


在All分類 -> 選Button元件(按鈕) -> 拖拉到指定位置

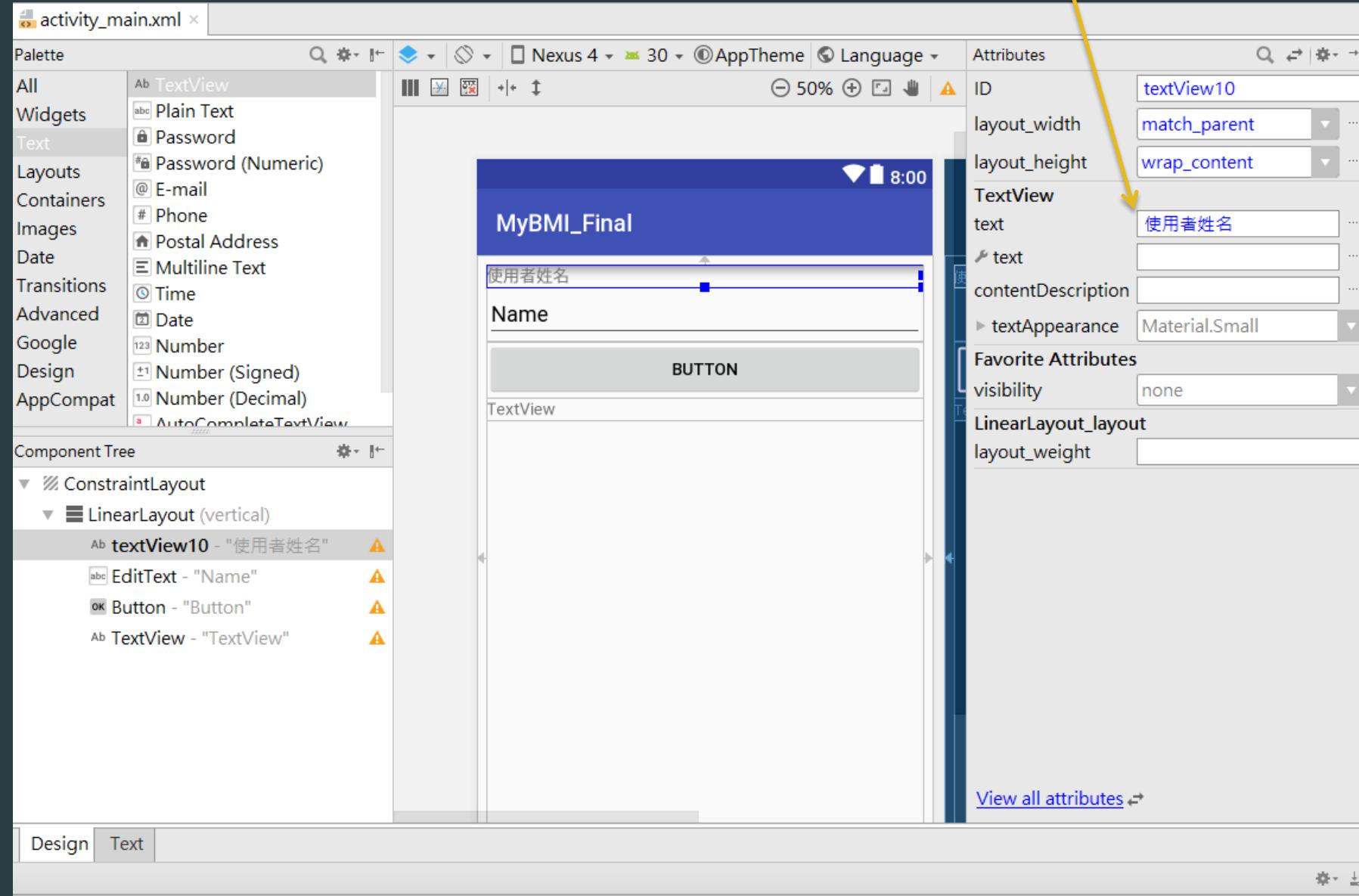


在All分類 -> 選Button元件(按鈕) -> 拖拉到指定位置

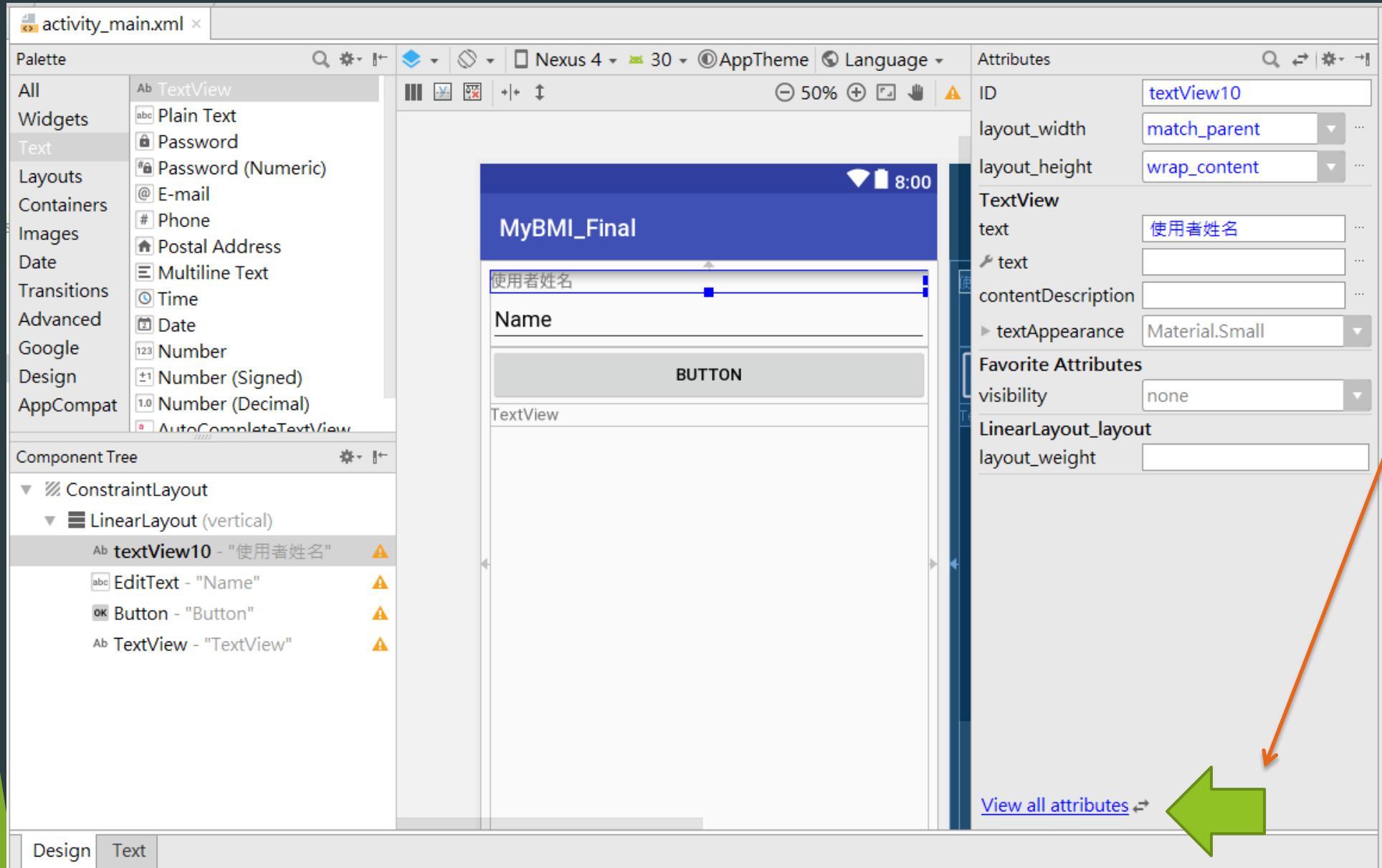




更改TextView 文字標籤的文字內容屬性(text)

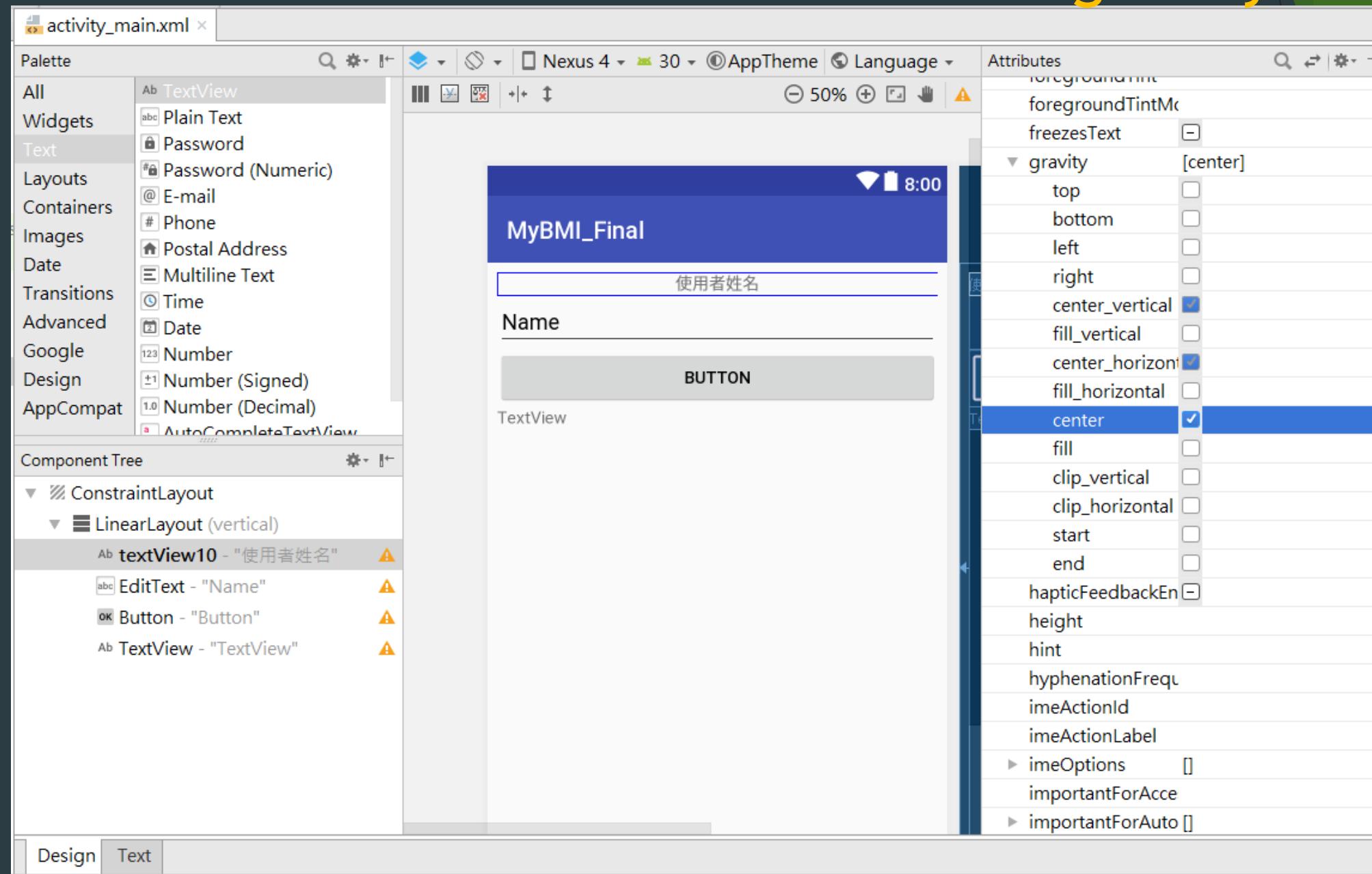


調整字體大小、顏色、置中 需要先點View all attributes

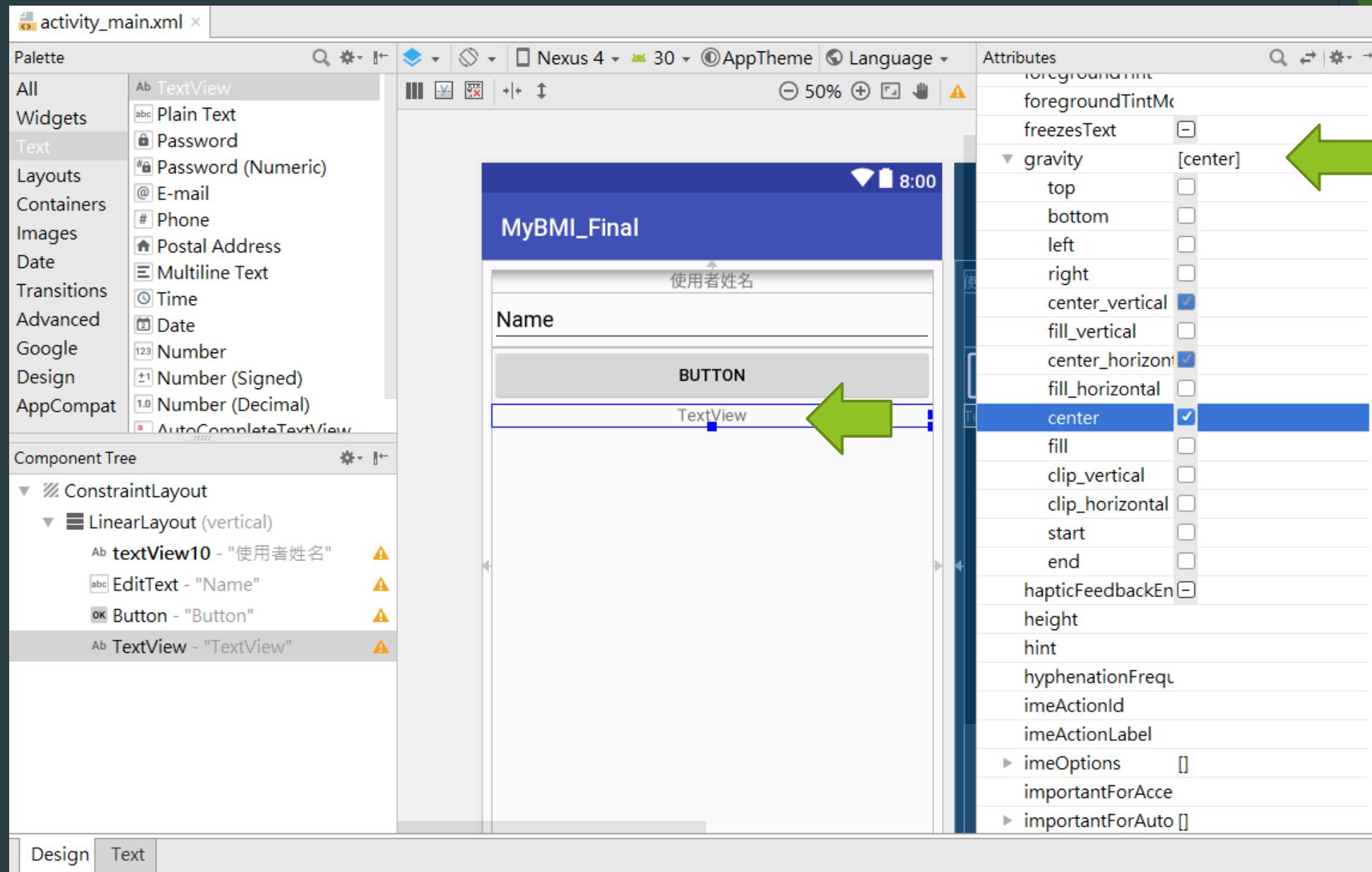


調整字體置中

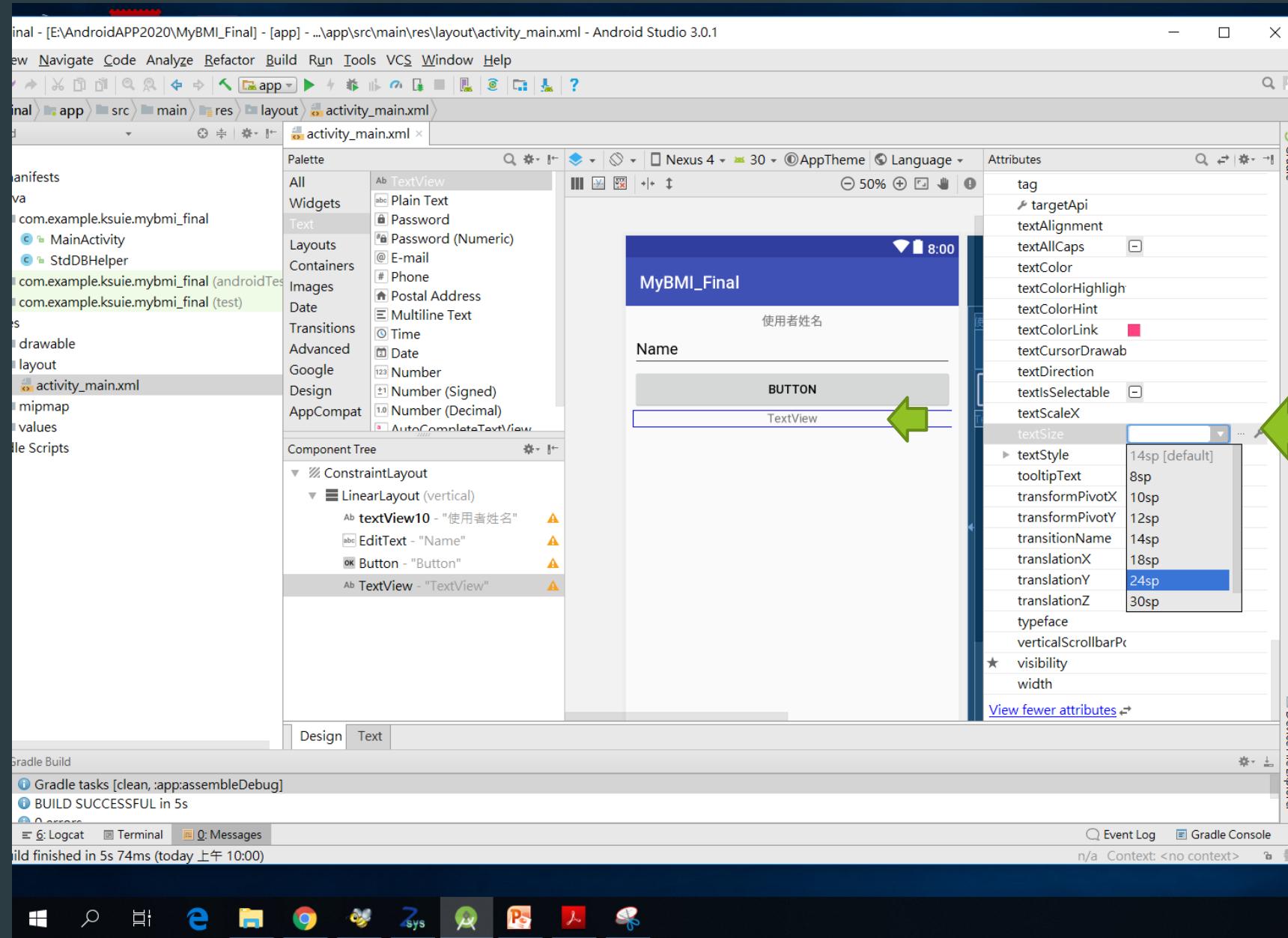
View all attributes -> gravity -> center



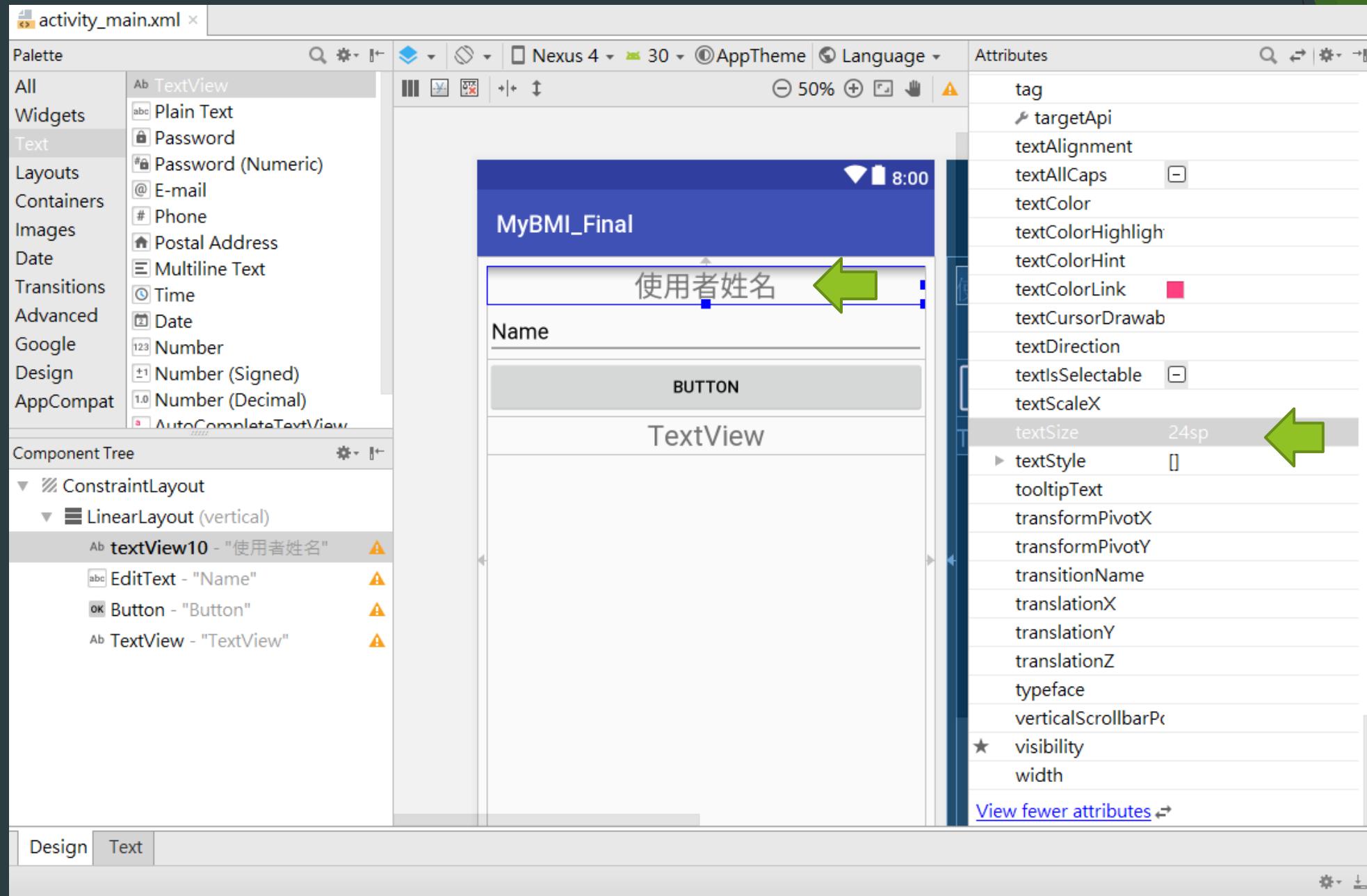
可再點選其他元件調整字體置中 gravity -> center



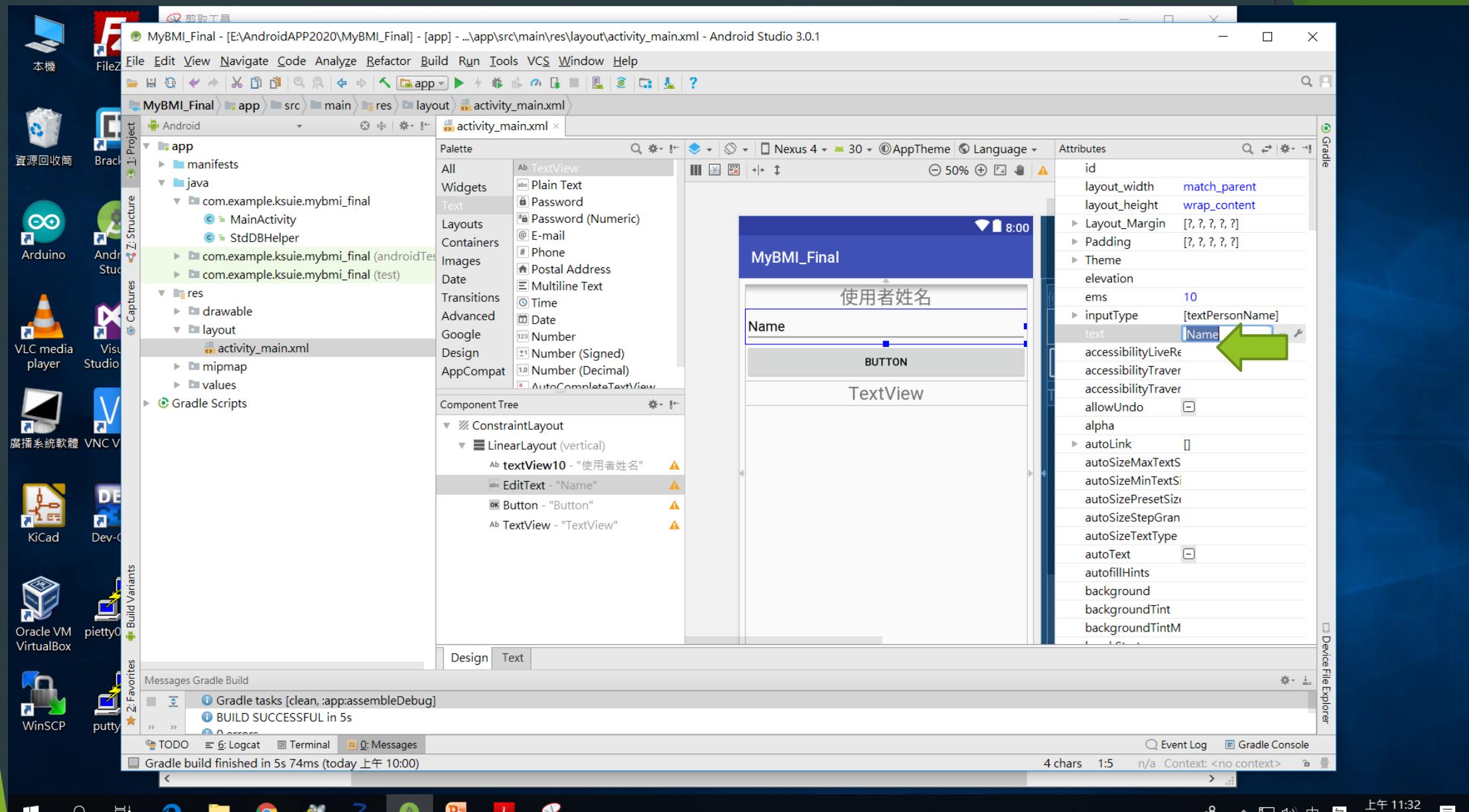
調整字體大小 View all attributes -> textSize



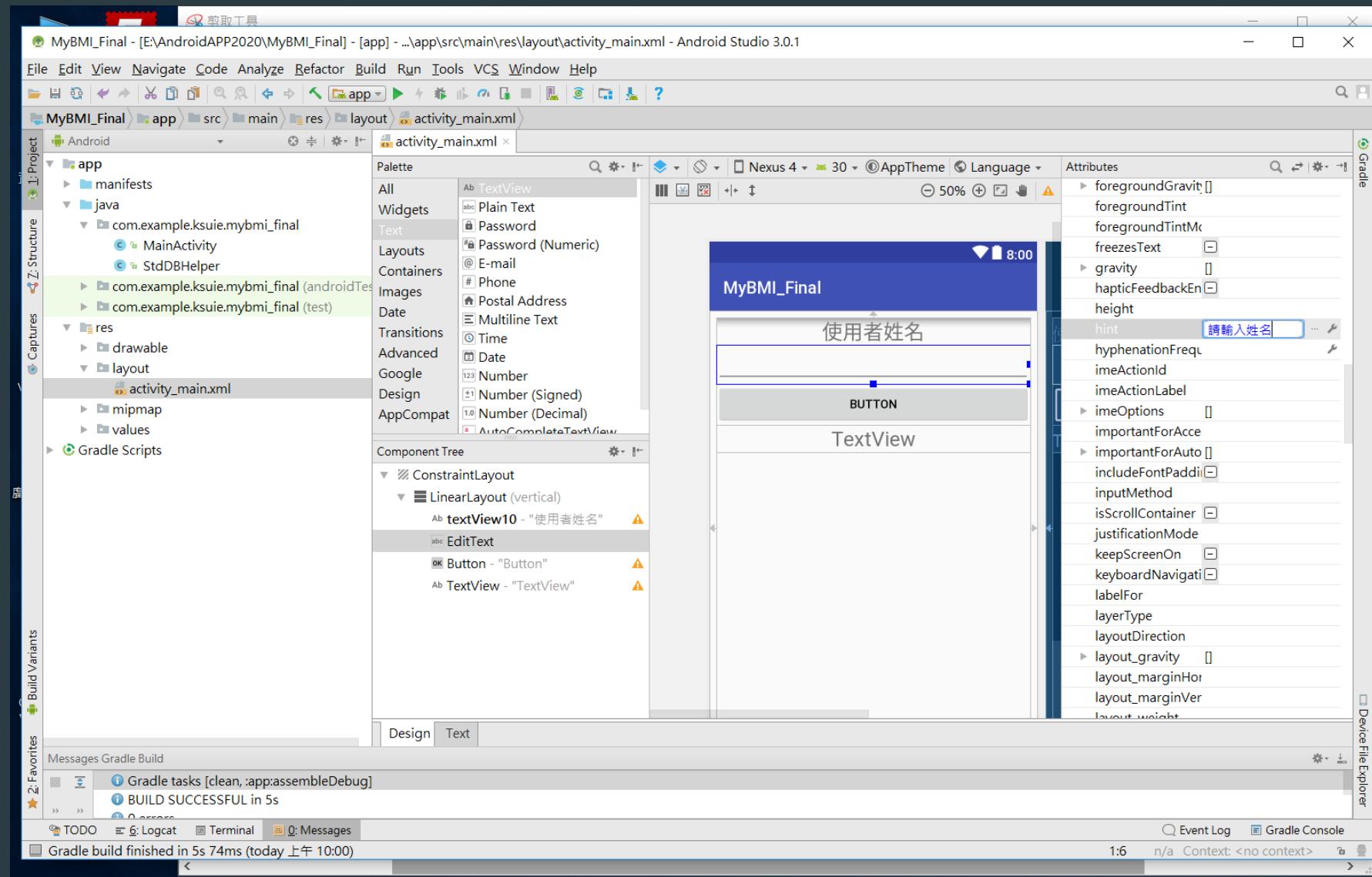
調整字體大小 View all attributes -> textSize



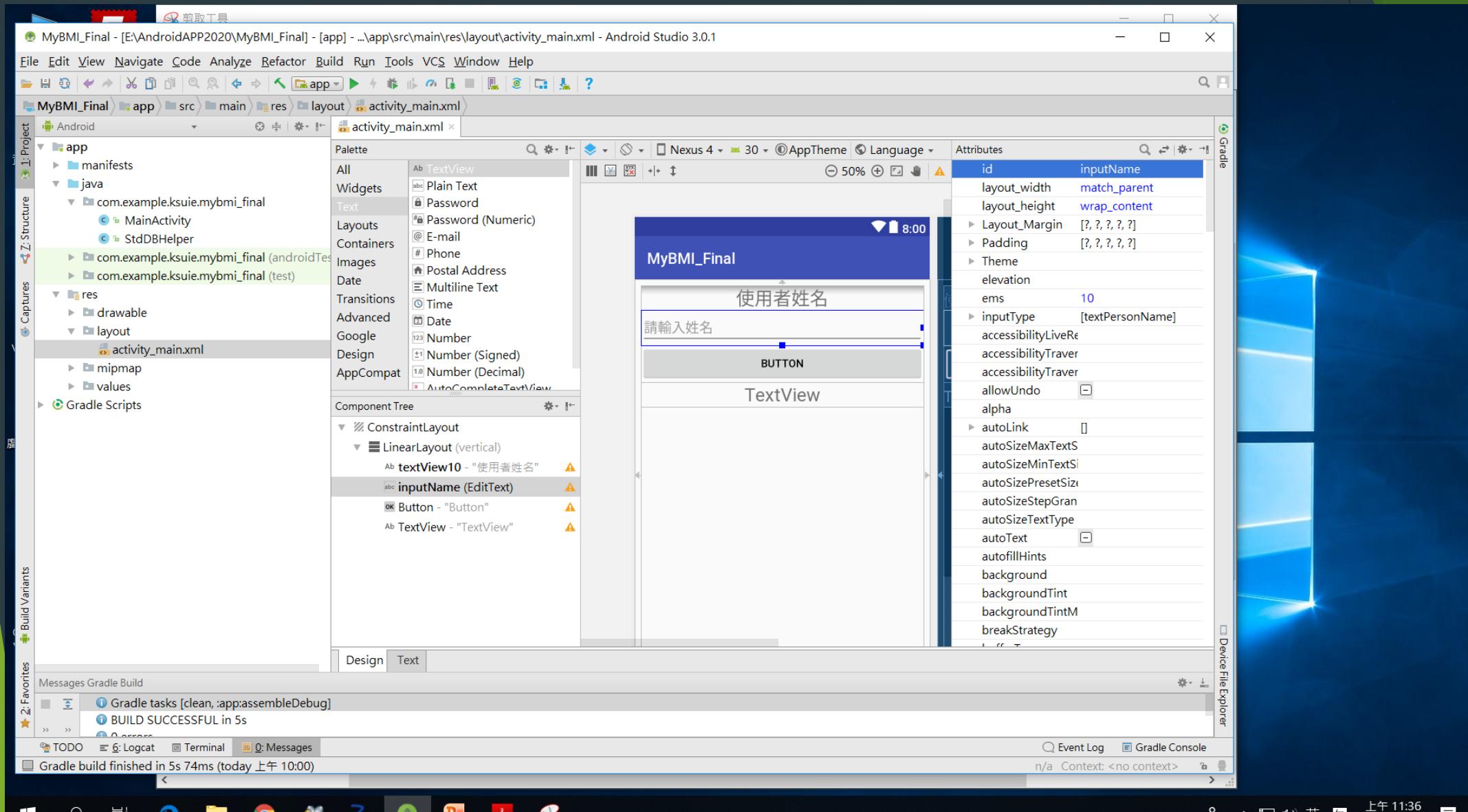
刪除文字框中的text內容



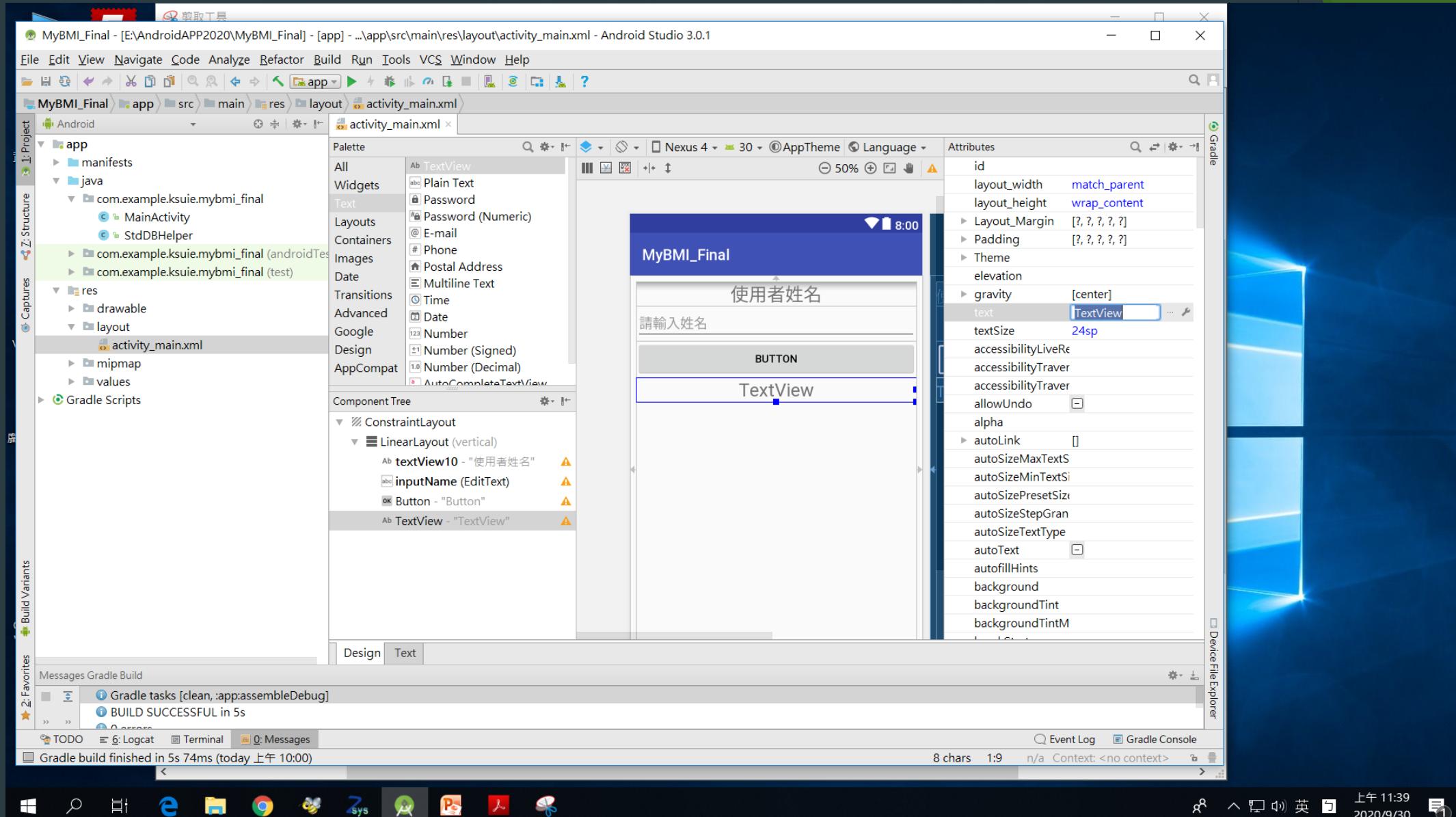
新增文字框的提示(hint)屬性



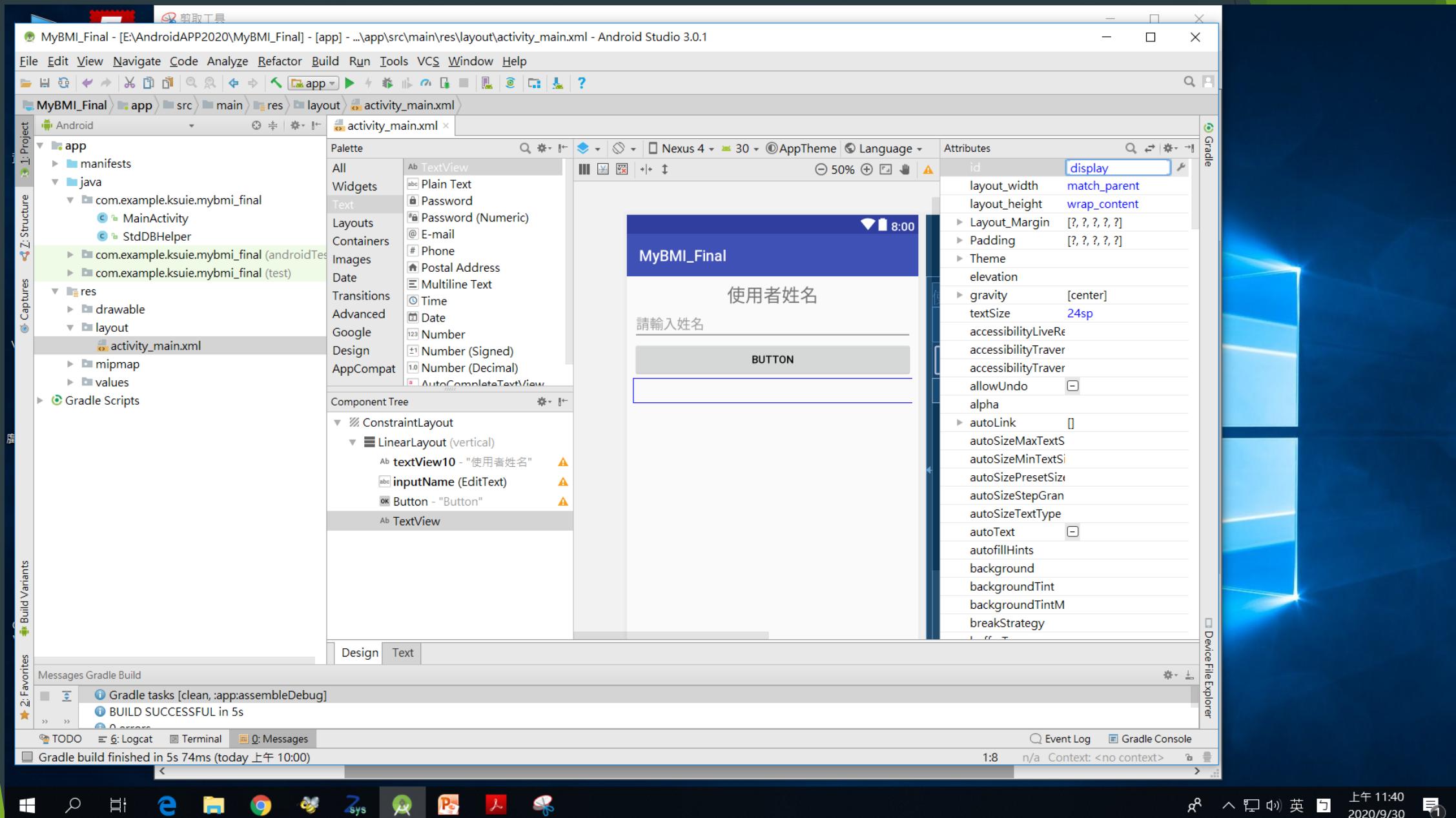
新增文字框的 id屬性 = inputName



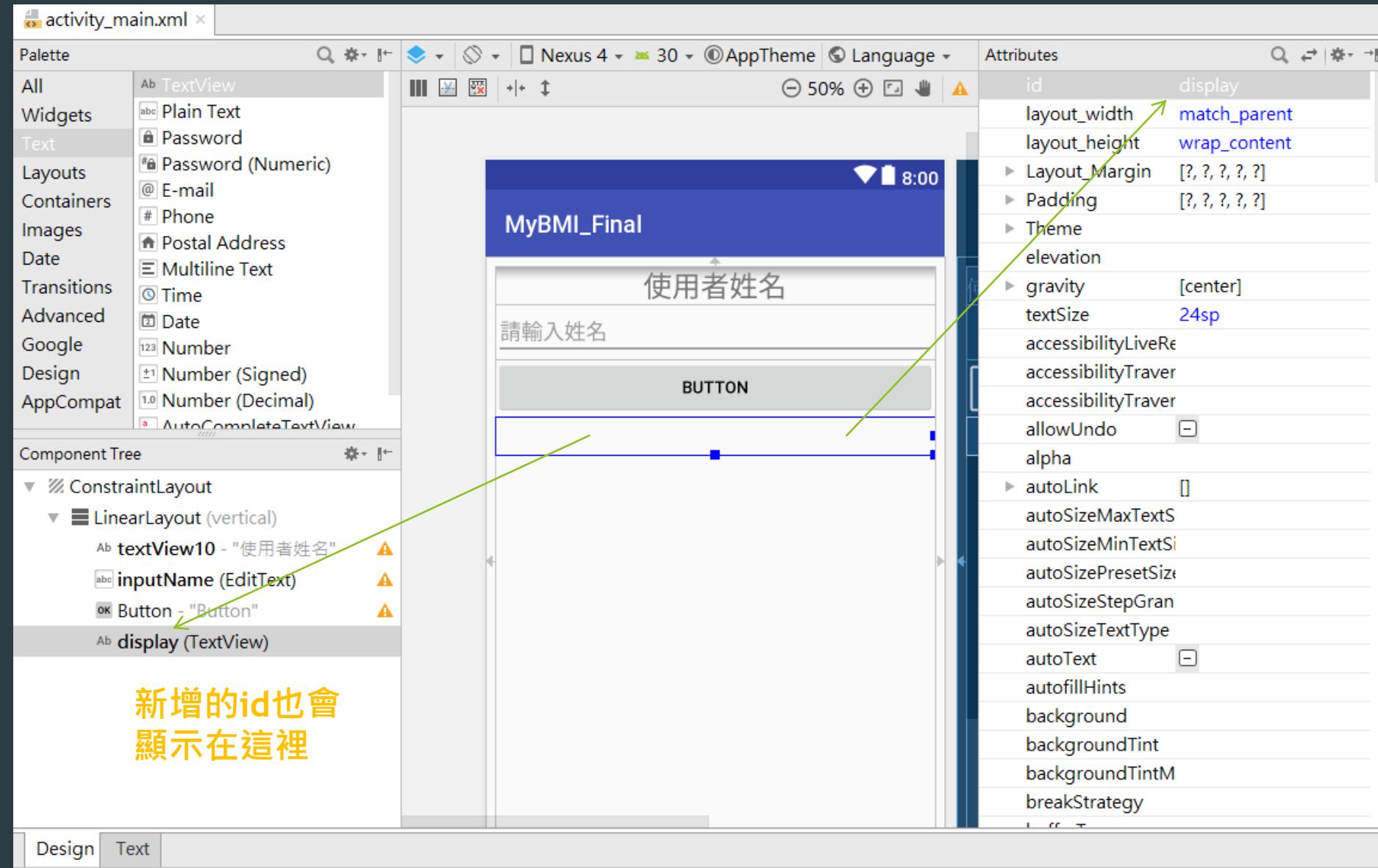
刪除最後的TextView內的text內容



新增最後顯示的TextView id屬性 = display

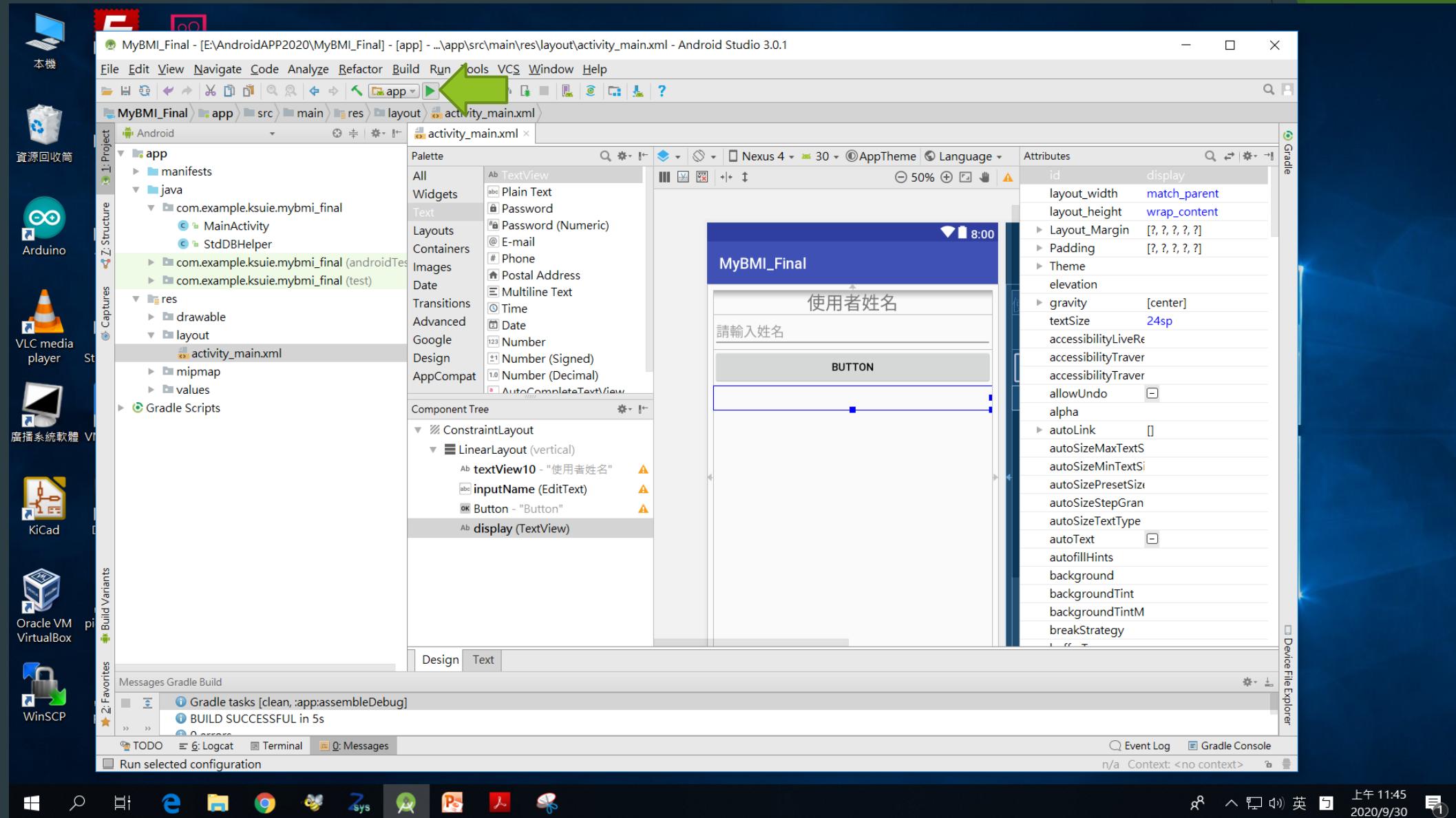


新增最後顯示的TextView id屬性 = display

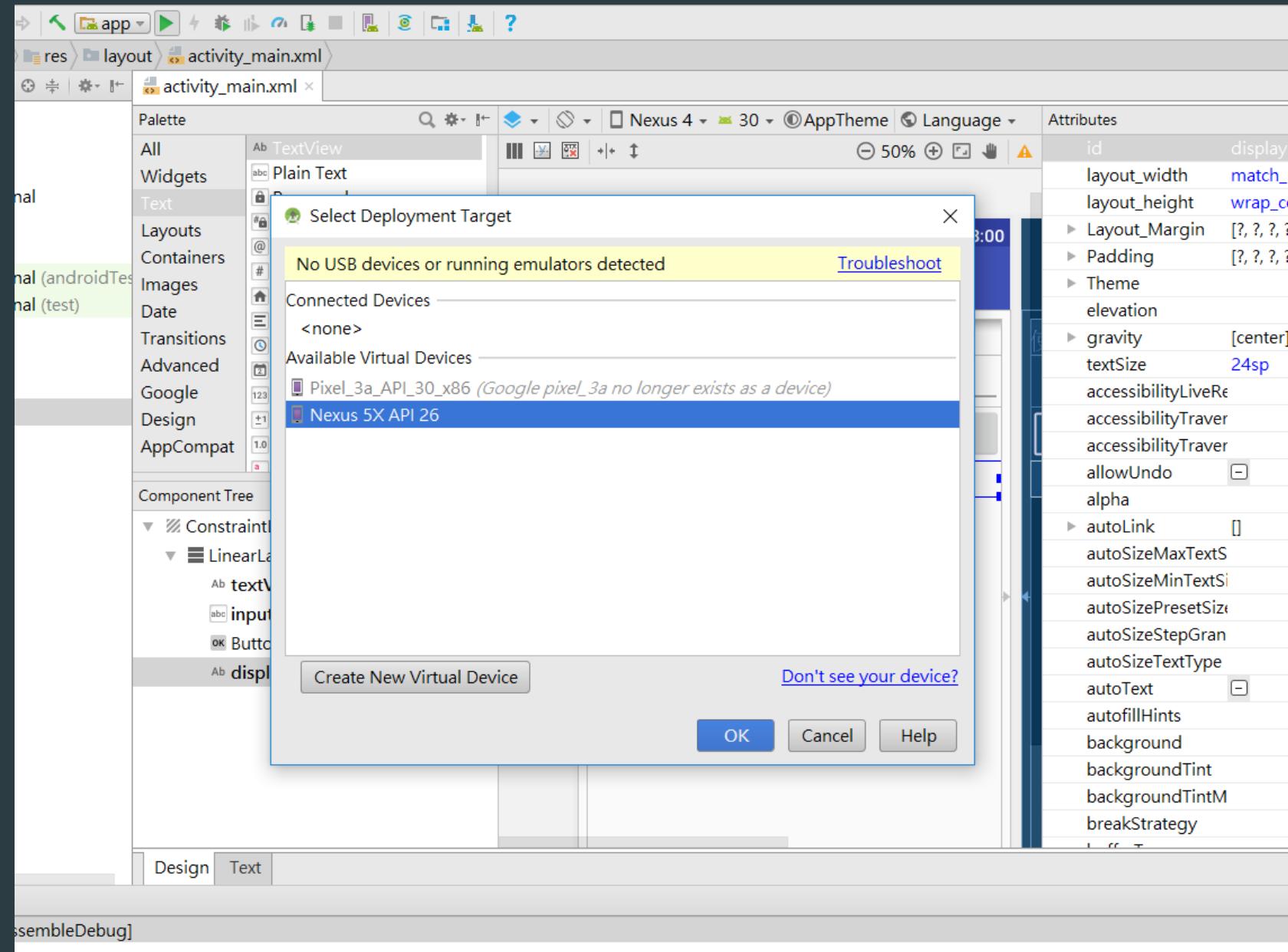


測試看看APP的執行畫面

點綠色箭頭 Run APP

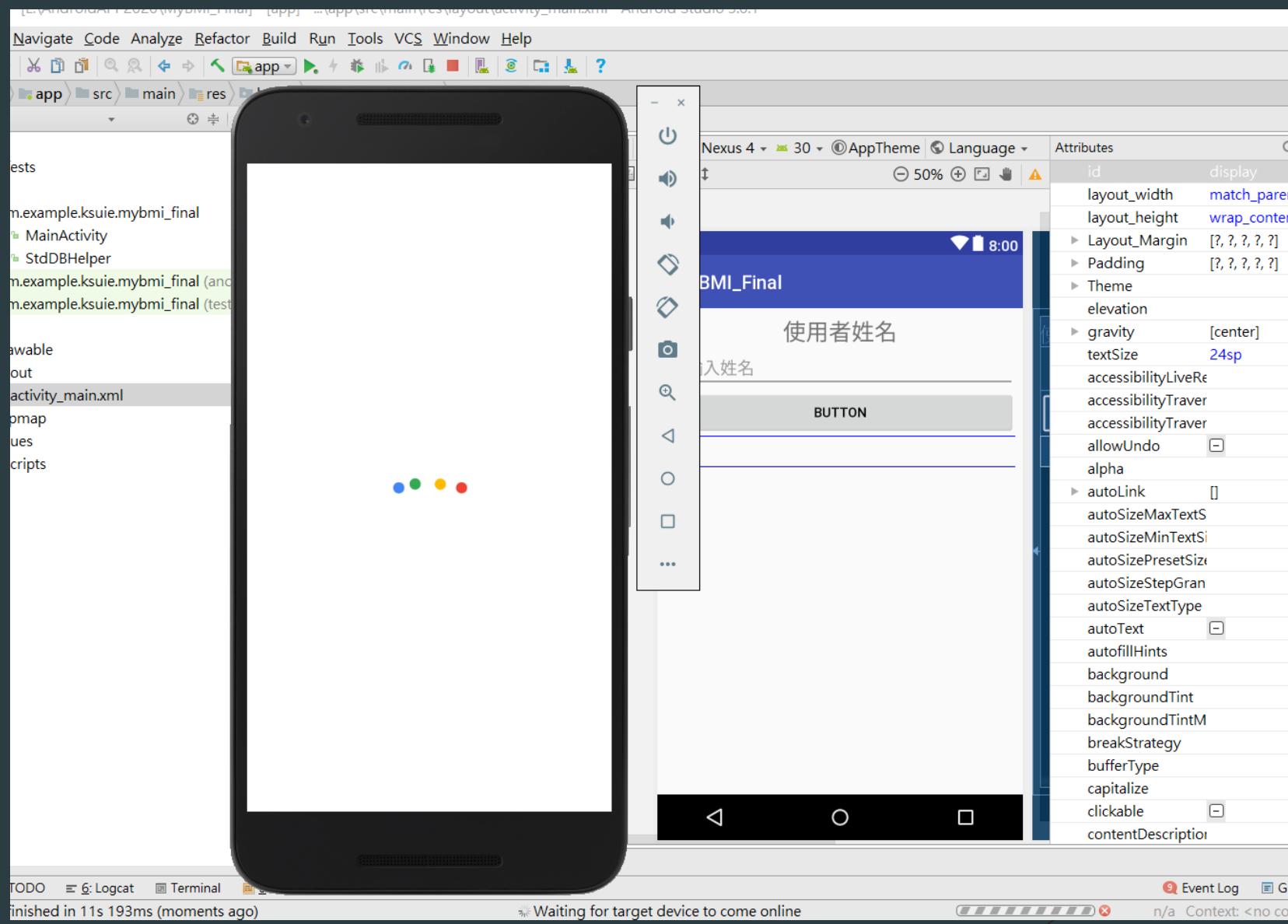


選擇虛擬手機型號

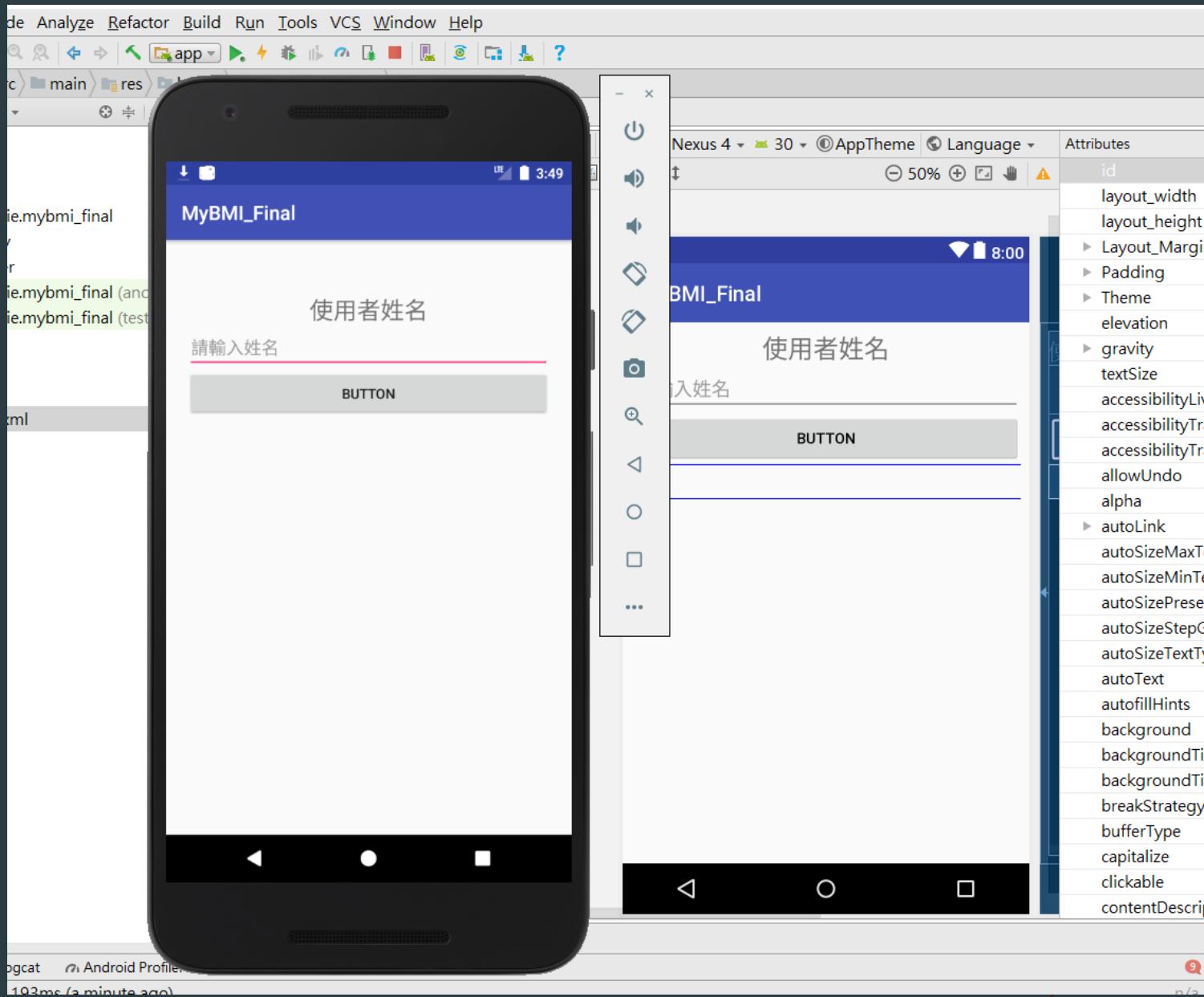


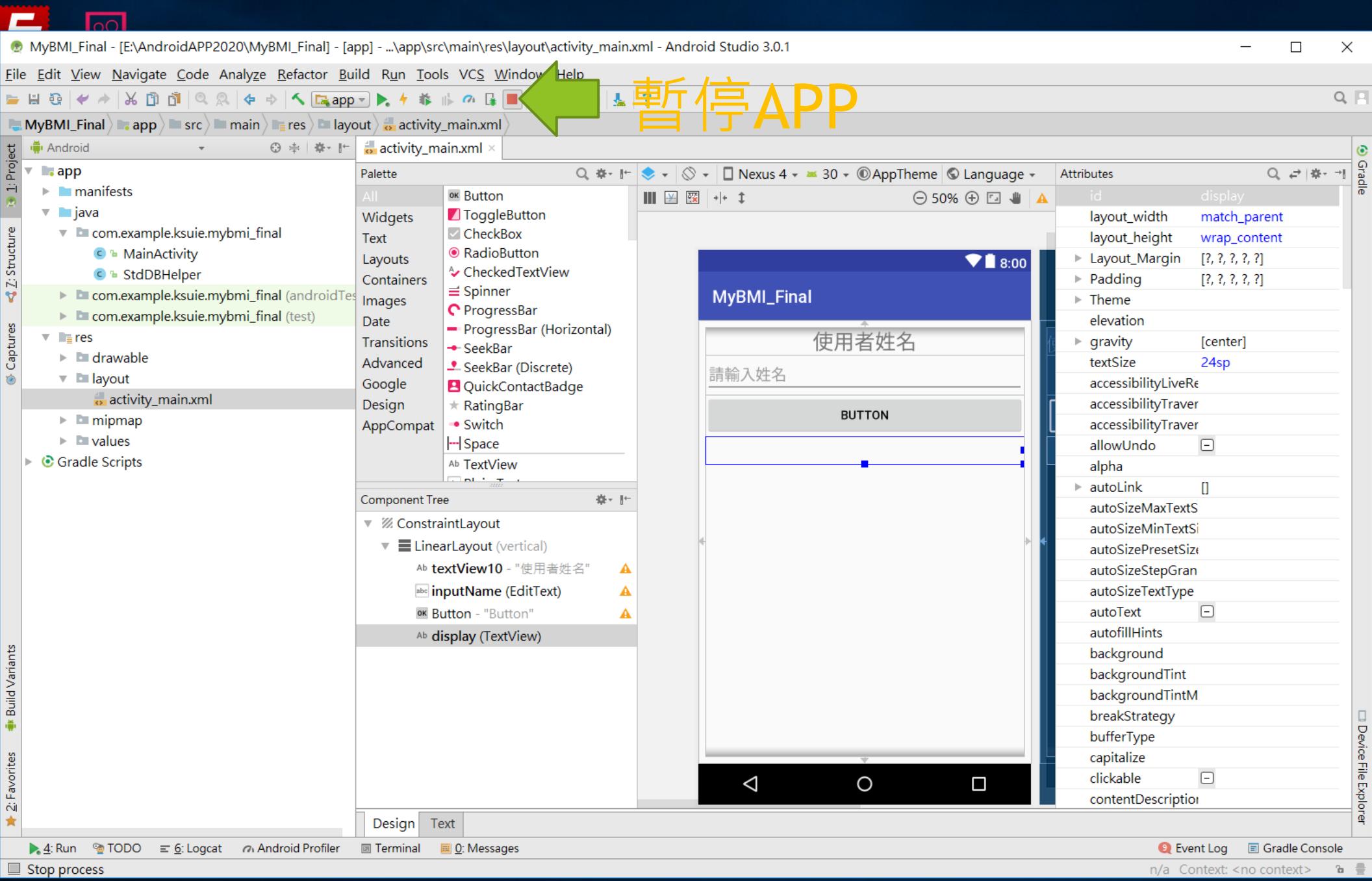
啟動APP中.....

需要一點時間

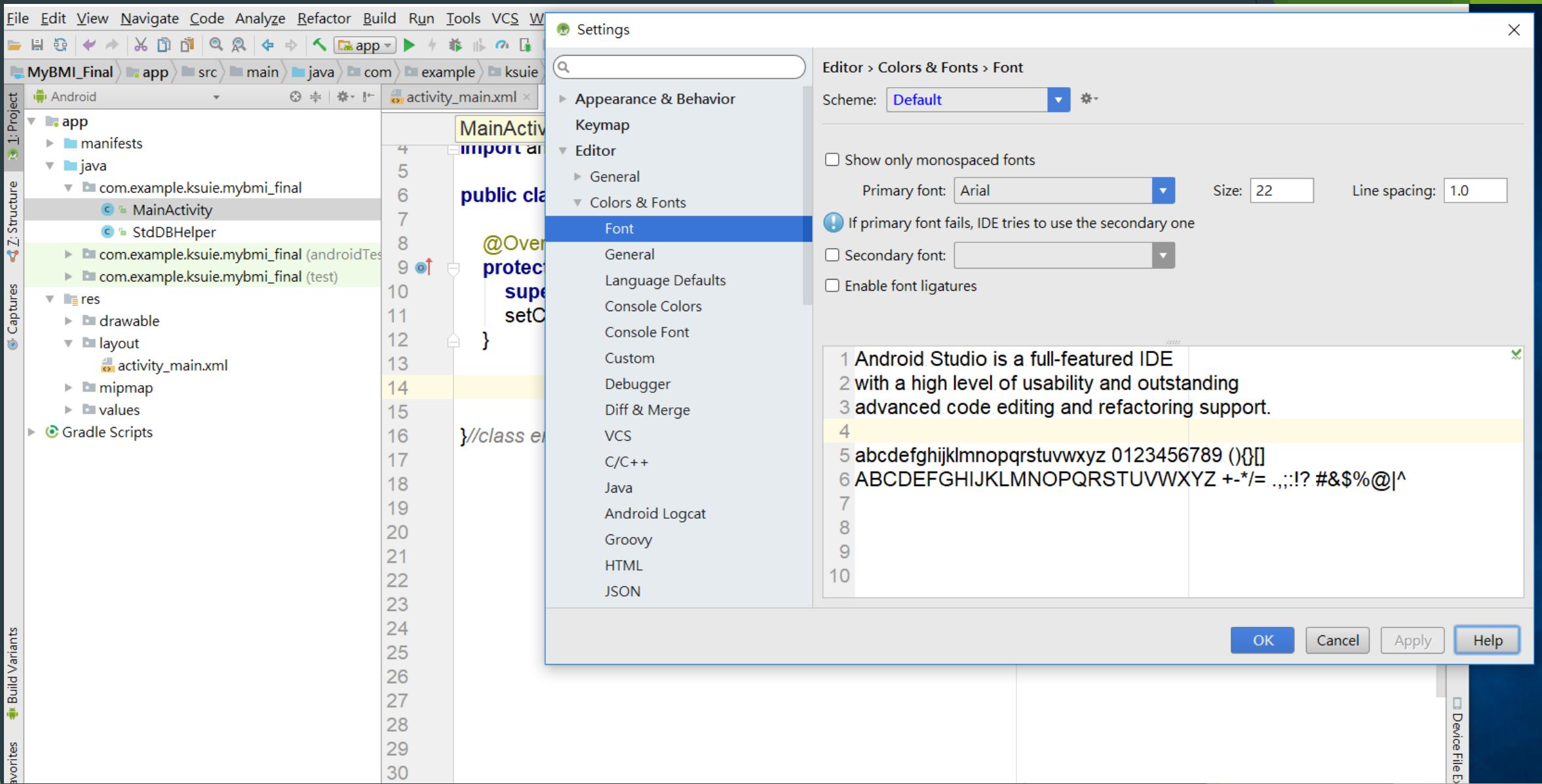


APP開啟成功 YA !





修改編輯區的字體大小 File → Settings



在MainActivity中，建立displayName副函式

The screenshot shows the Android Studio interface with the project 'MyBMI_Final' open. The code editor displays the MainActivity.java file. Two green arrows point to specific parts of the code:

- A green arrow points to the package declaration: `com.example.ksuie.mybmi_final`.
- A green arrow points to the `public void displayName(View view){}` method definition.

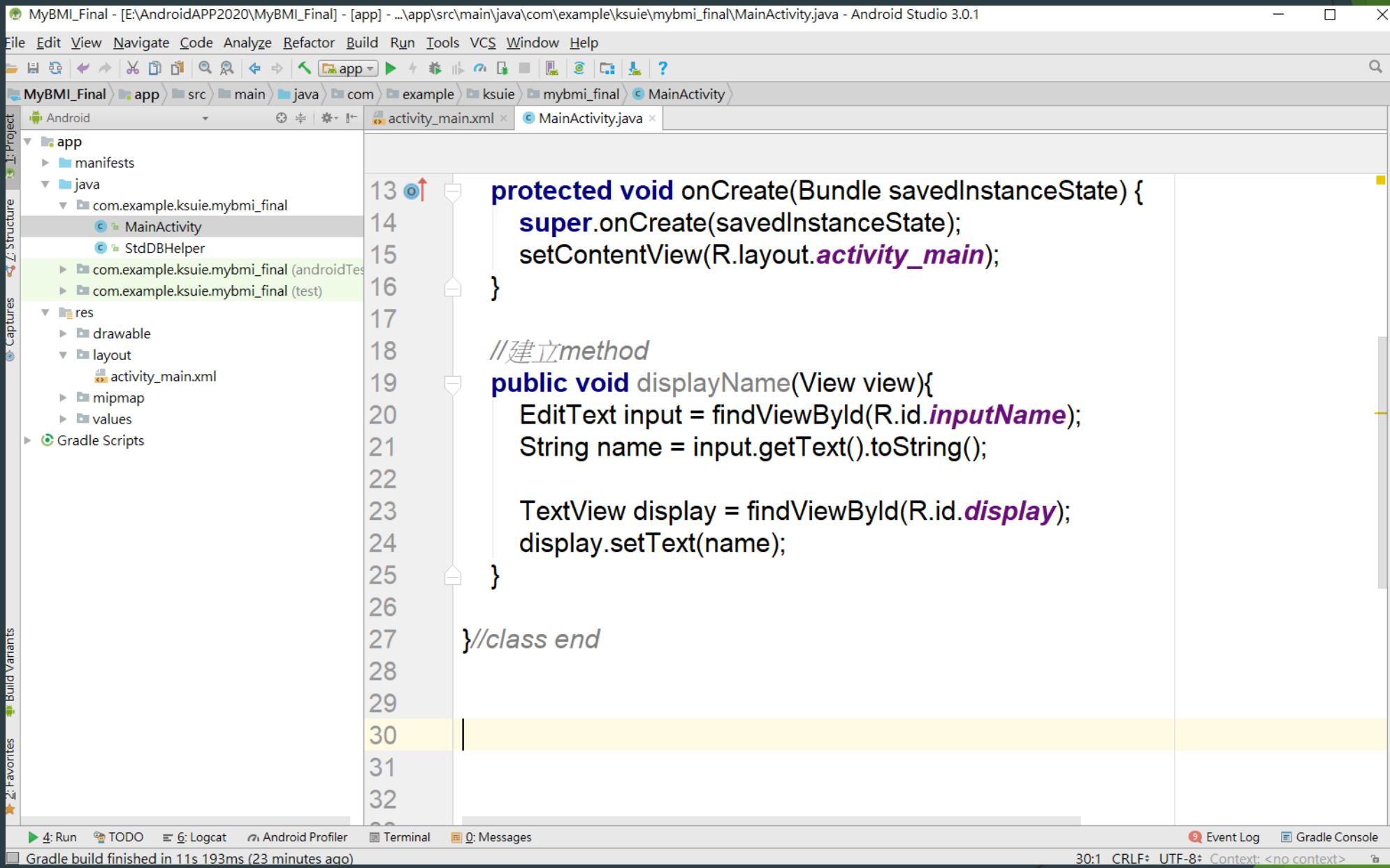
```
MyBMI_Final - [E:\AndroidAPP2020\MyBMI_Final] - [app] - ...\\app\\src\\main\\java\\com\\example\\ksuie\\mybmi_final\\MainActivity.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MyBMI_Final app src main java com example ksuie mybmi_final MainActivity
MainActivity displayName()
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}

//建立method
public void displayName(View view){
}

}//class end

11 ①
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
```

在MainActivity中，建立displayName副函式



The screenshot shows the Android Studio interface with the project 'MyBMI_Final' open. The code editor displays the MainActivity.java file under the com.example.ksuie.mybmi_final package. The code implements the onCreate method and defines a displayName method.

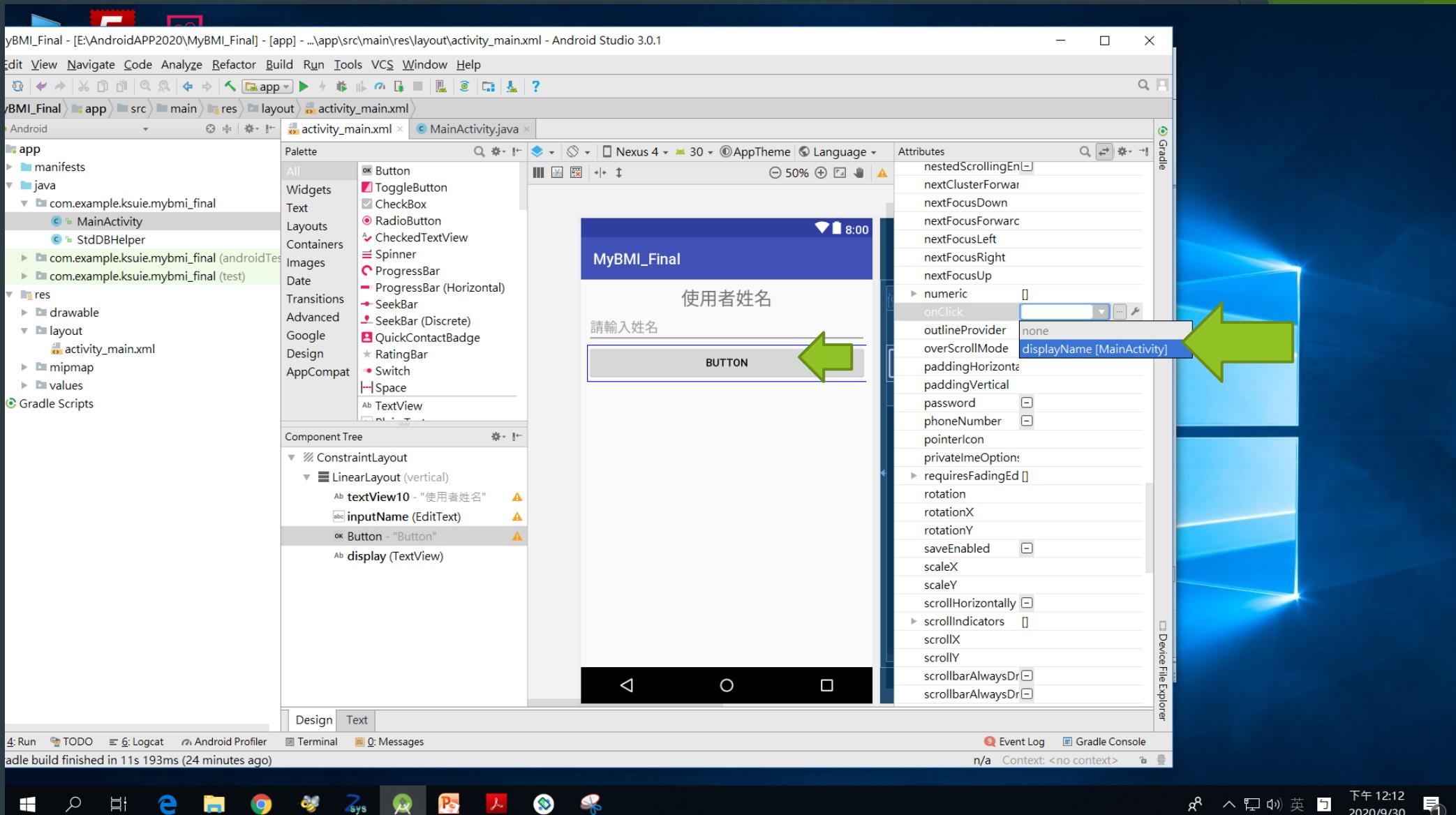
```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //建立method
    public void displayName(View view){
        EditText input = findViewById(R.id.inputName);
        String name = input.getText().toString();

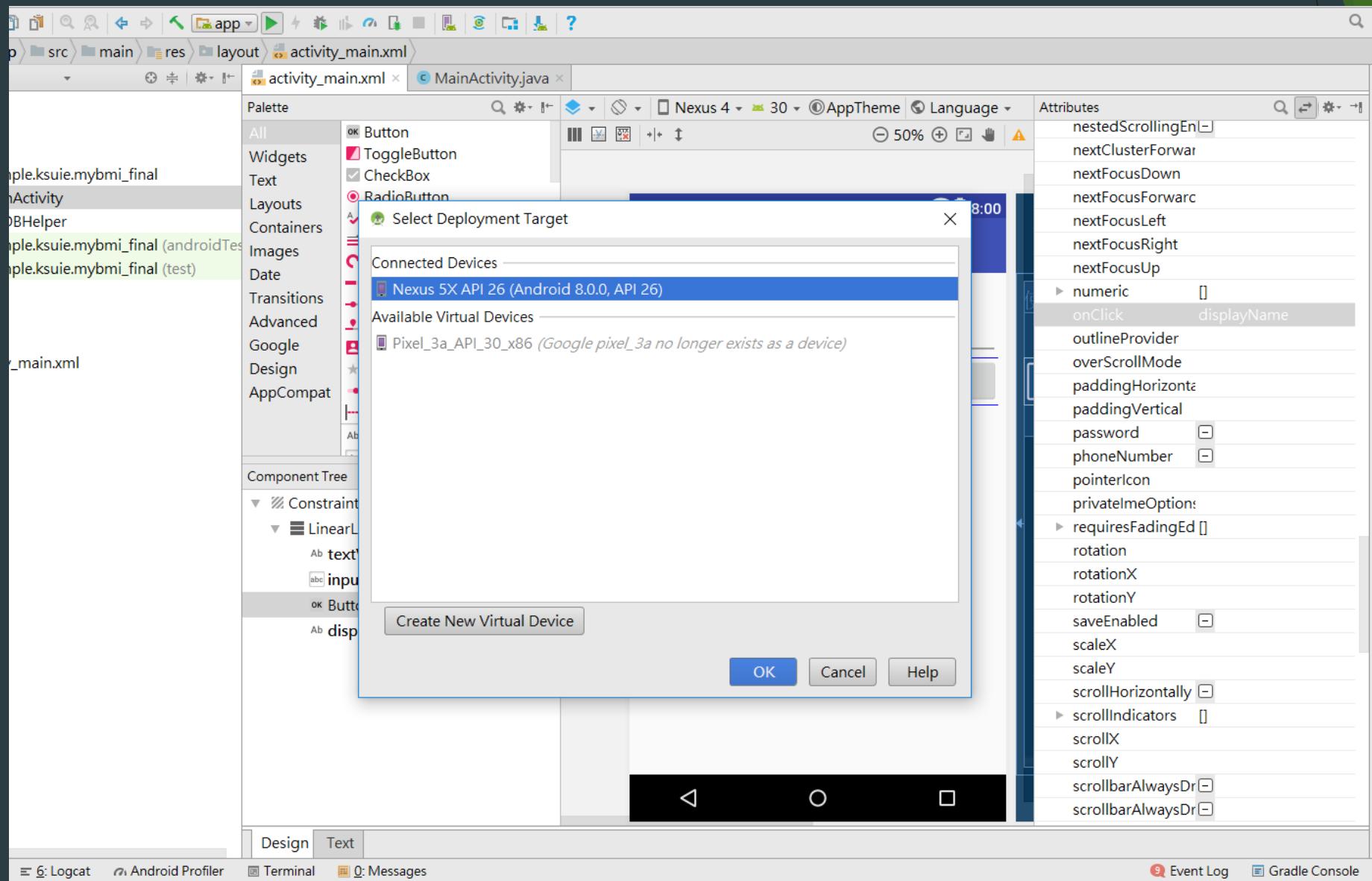
        TextView display = findViewById(R.id.display);
        display.setText(name);
    }
} //class end
```

The code editor highlights several identifiers in purple: `activity_main`, `inputName`, and `display`. The Java code uses standard conventions like camelCase for method names and `R.layout.activity_main` for layout resources.

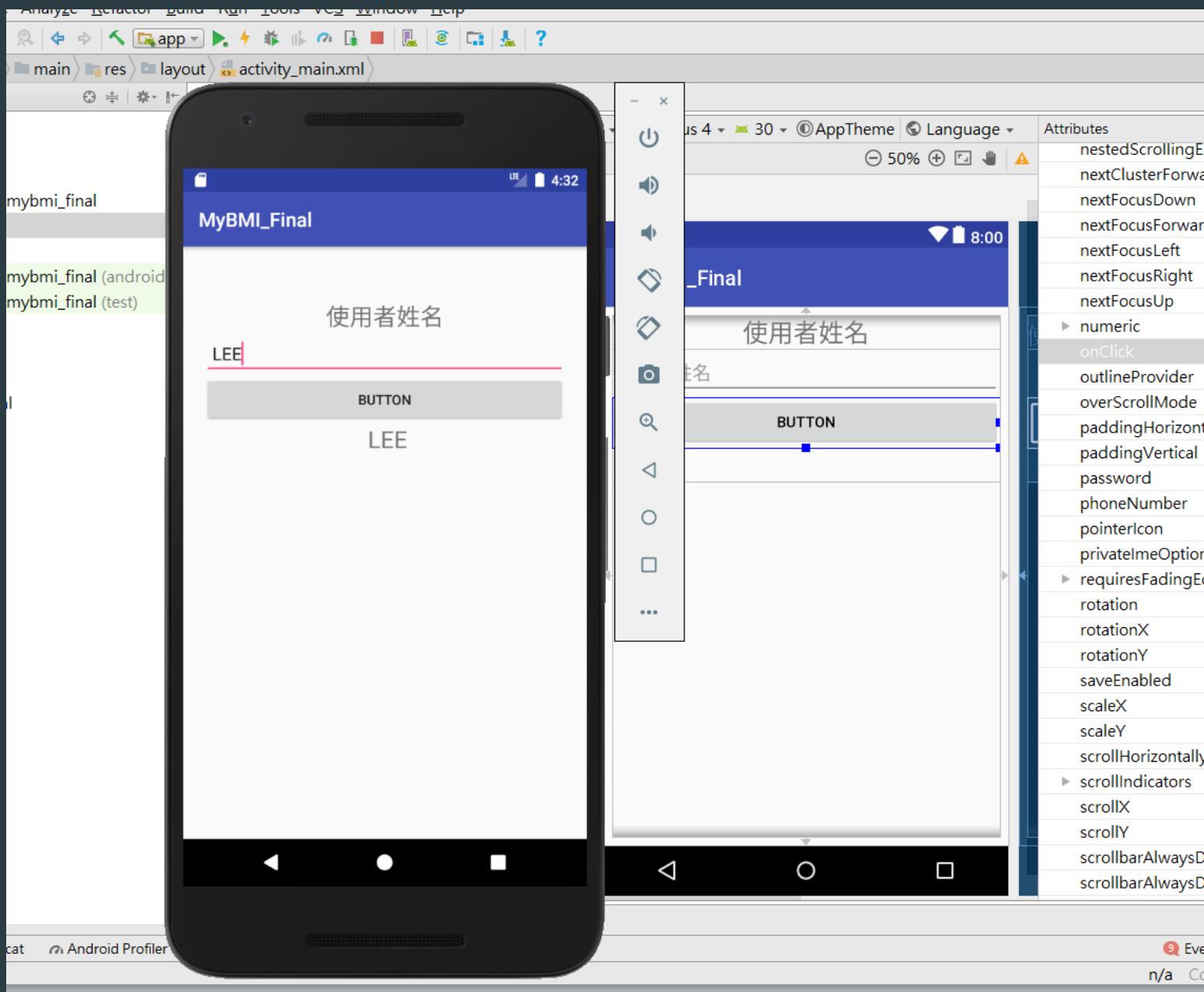
回到activity_main.xml，將button的onClick屬性與displayName副函式連結



Run APP , 選擇虛擬手機型號



APP執行結果





恭喜你，Android APP 完成！