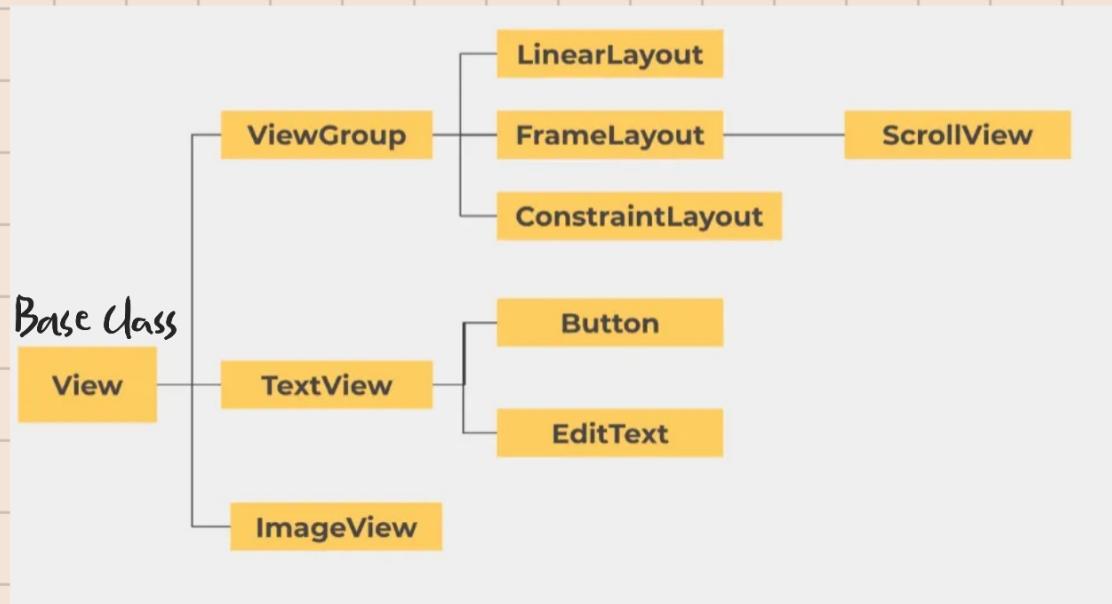


Layouts and ViewClass Hierarchy



Bio App

1º Delete default constrain layout

change it to Linear layout

In activity_main.xml, change

<android.constraintlayout.widget....>

↓ to

<android.appcompat.widget.LinearLayout....>

↑ Type Linear and this option will show up

2º Pull ViewImage to gui

Navigation

3° search "orientation" → vertical

4° Center Image → search "gravity" → layout-gravity
→ center

5° Pull textView → layout-gravity: center → bold, textSize

→ padding  → string name (refactor)

→ right click textView → refactor → extract style

→ give Style name → res/values/styles.xml

6° Pull ScrollView → left click ScrollView

→ delete LinearLayout → Pull textView onto ScrollView

→ go to strings.xml → paste:

`<string name="my-b10">...In <u><i></i></u>...</string>`

斜体

→ design page → text button " " → select "my-b10"

7° Search "style" → style button " " → select "nameStyle"

8° Search "padding" → padding button " " → "+" Dimension value

→ Resource name: text-padding → ok

orient

gravity

Add
style.xml

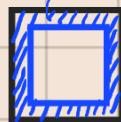
right
xml

design

text

Resource value: 8dp

layout margin



→ res/values/dimens.xml

9° Search "margin" → layout-margin "0"

Resource name: layout_margin → ok

Resource value: 6dp

10° Search "lineSpace" → lineSpacingMultiplier → 1.4

11° Right click ScrollView/TextView → refactor

→ Extract style → name: generalStyle

→ res/values/styles.xml

12° Search "font" → fontFamily scroll down → more

13° Search "style" → style button →

13° Palette → Text tab → Pull Ab Plain Text to gui

Note: Ab, underscore means this is edit text

14° Select PlainText obj → Style button → generalStyle

15 Search "hint" → hint button

Resource name: hobby-hint-text → ok

Resource value: Enter your hobby

→ Delete text Attribute → will see resource value

16^o Drag Button below PlainText obj

17^o Style button → generalStyle

layout_gravity → center

text button → Resource name: done-text → ok

Resource value: Done

18^o Drag TextView below button

Style button → generalStyle

obj寬度會和上方的一樣

✓ Search "layout" → layout_width → "match-parent"

Remove text → Enter

✓ Search "visibility" → select "invisible"

invisible: 方框不顯示

gone: 完全消失

19° See ImageView warning

→ Image without 'content Description'

→ Search "content description"

→ content description button

→ Resource name: image_description → ok

Resource value: Android bot Image

20° EditText Warning "Autofill"

android:autofillHints = "@string/auto_fill_hints"

→ put cursor between string → click red bulb

→ Create value resource →

Resource name: classical-guitar → ok

21° Change editText id → "enter-hobbies" → refactor

Change button id → "done-button"

22°

*. MainActivity.java

```
activity_main.xml x MainActivity.java x
1 package com.logan.bio;
```

```
2
3 import ...
15
16 public class MainActivity extends AppCompatActivity {
17     private EditText enterHobbies;
18     private TextView hobbies;
19
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_main);
25
26         enterHobbies = findViewById(R.id.enter_hobbies);
27         hobbies = findViewById(R.id.hobbies_text);
28     }
29
30     public void addHobbies(View view) {
31         hobbies.setText(String.format("Hobbies: %s",
32             enterHobbies.getText().toString().trim()));
33         hobbies.setVisibility(VISIBLE);
34
35         // Hide keyboard
36         // Make the keyboard disappear once we press done
37         InputMethodManager inputMethodManager =
38             (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
39         inputMethodManager.hideSoftInputFromWindow(view.getWindowToken(), 0);
40     }
41 }
```

1^o 顯示 input 文字
2^o 並設為 visible
3^o 按 button 後，keyboard 自動消失

① "Hobbies: " + enterHobbies.getText().toString().trim()

↳ setText 內容這樣會有 warning，要把 "+" 形式改掉

按燈泡 → Replace '+' with 'String.format()'

② = getSystemService (Context.INPUT_METHOD_SERVICE);

↳ 等號左側只能這樣會有 warning，

→ put cursor between 'getSystemService'

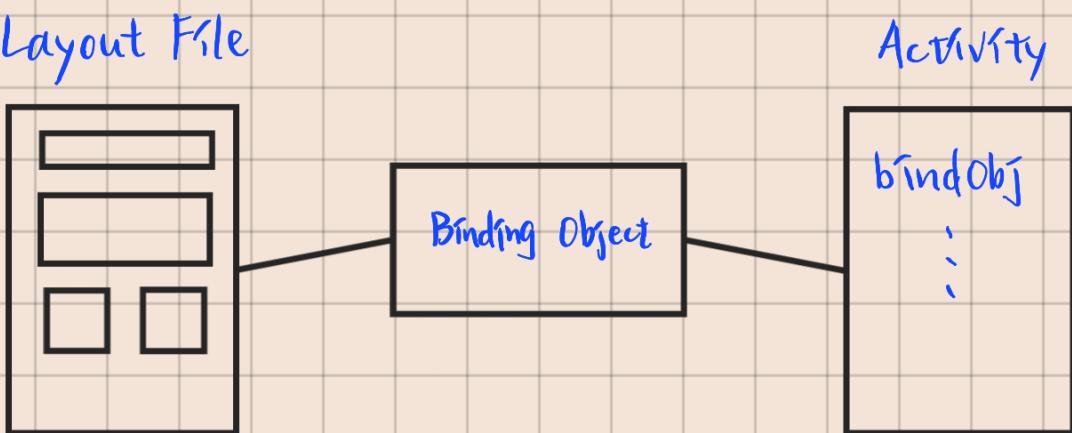
→ Alt + Enter →

Cast to 'android.view.InputMethodManager'

Data Binding

findViewById is an expensive operation

Data Binding 可透过 binding 物件 + Id 操件所有 View obj.



*. Gradle Scripts / build.gradle (Module: Bio.app)

```
    android {  
        compileSdk 32
```

buildFeatures {

dataBinding true

3

7

*. activity_main.xml

Add this
做完記得按右上角
的 Sync !!!

activity_main.xml

```

1  <?xml version="1.0" encoding="utf-8"?>  1° 先將 <layout></layout>
2  <layout xmlns:android="http://schemas.android.com/apk/res/android"  2° 將 xmlns 貼到
3      xmlns:app="http://schemas.android.com/apk/res-auto"  layout 這
4      xmlns:tools="http://schemas.android.com/tools">  3° 將其他全部
5
6      <androidx.appcompat.widget.LinearLayoutCompat>  貼到 <layout>
7          android:layout_width="match_parent"  内
8          android:layout_height="match_parent"  4° 利用
9          android:background="#2196F3"  Ctrl+Alt+L
10         android:orientation="vertical"  排版
11         tools:context=".MainActivity"
12
13         <ImageView>
14             android:id="@+id/imageView"  上述所有步驟，皆可使用燈泡
15             android:layout_width="wrap_content"
16             android:layout_height="wrap_content"
17             android:layout_gravity="center"
18             android:contentDescription="@string/image_description"
19             app:srcCompat="@drawable/ic_launcher_foreground" />
20
21         <TextView>
22             android:id="@+id/textView2"
23             style="@style/nameStyle"
24             android:layout_width="wrap_content"
25             android:layout_height="wrap_content"
26             android:fontFamily="@font/aldrich"
27             android:text="@string/name" />
28
29         <EditText>

```

→ Convert to data binding layout

→ 完成

*. MainActivity.java

activity_main.xml

```

1  package com.logan.bio;
2
3  import ...
4
5  public class MainActivity extends AppCompatActivity {
6      private ActivityMainBinding binding;
7
8      @Override
9      protected void onCreate(Bundle savedInstanceState) {
10          super.onCreate(savedInstanceState);
11          setContentView(R.layout.activity_main);
12          enterHobbies = findViewById(R.id.enter_hobbies);
13          hobbies = findViewById(R.id.hobbies_text);
14          binding = DataBindingUtil.setContentView(activity, R.layout.activity_main);
15
16          binding.doneButton.setOnClickListener(view -> addHobbies(view));
17
18      }
19
20      public void addHobbies(View view) {
21          binding.hobbiesText.setText(String.format("Hobbies: %s",
22              binding.enterHobbies.getText().toString().trim()));
23
24          binding.invalidateAll();
25
26      }
27
28  }
29
30
31
32
33
34
35
36
37
38

```

宣告 binding

將預設的 setContentView Comment

不用再 connect obj 和 Id 以後都直接 binding. Id

↳ On Click 在這宣告即可！

(不須在 activity_main 宣告 onClick !!!)

```
39 binding.hobbiesText.setVisibility(VISIBLE); ↗ 全部是 binding.id
40
41     // Hide keyboard
42     // Make the keyboard disappear once we press done
43     InputMethodManager inputMethodManager = (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
44     inputMethodManager.hideSoftInputFromWindow(view.getWindowToken(), flags: 0);
45
46 }
```

Binding Data

以往都用 strings.xml 設定要取值再呼叫

Data binding 可以建立一个 Data obj, 在 MainActivity 透过 methods 設定取值後, 可以在 activity_main.xml 或 MainActivity 使用

1° 1.java / com.logan.bio / MainActivity

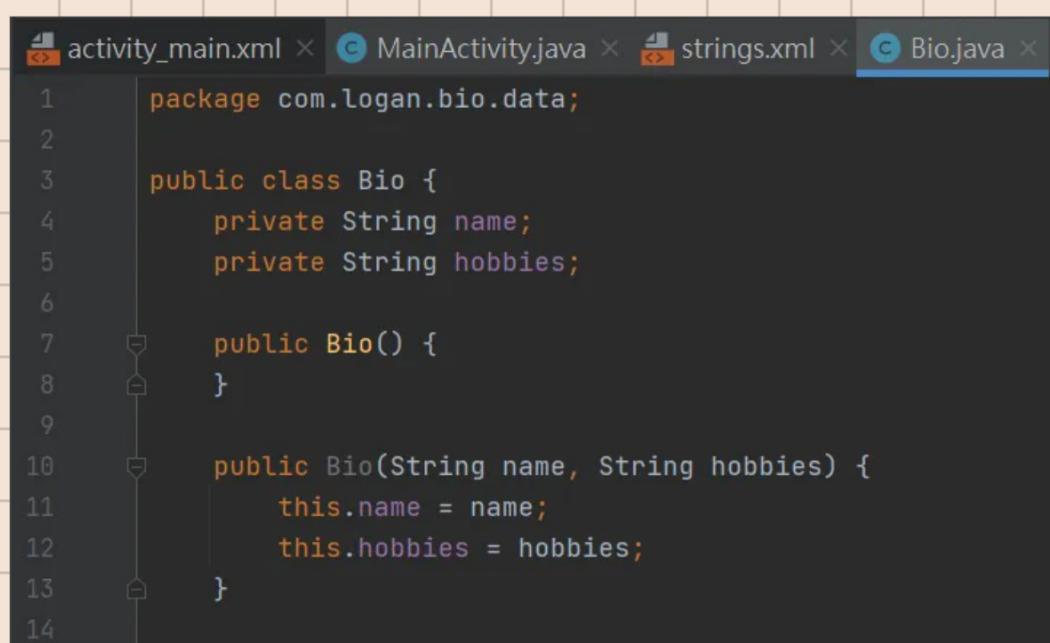
Right Click  → New → package → com.logan.bio.data

→ 產生 1.java / com.logan.bio / data

2° Right Click /data → New → Java Class → Bio

*. Bio.java

*. Right click → generate → / constructor
/ Getter and Setter



```
activity_main.xml × MainActivity.java × strings.xml × Bio.java ×
1 package com.logan.bio.data;
2
3 public class Bio {
4     private String name;
5     private String hobbies;
6
7     public Bio() {
8     }
9
10    public Bio(String name, String hobbies) {
11        this.name = name;
12        this.hobbies = hobbies;
13    }
14}
```

```
15     }
16     }
17     }
18     }
19     }
20     }
21     }
22     }
23     }
24     }
25     }
26     }
27     }
28     }
29     }
30     }
31     }
```

*. activity-main.xml

3° 將創建的 Bio Class 傳到 activity_main 中

```
1      <?xml version="1.0" encoding="utf-8"?>
2      <layout xmlns:android="http://schemas.android.com/apk/res/android"
3              xmlns:app="http://schemas.android.com/apk/res-auto"
4              xmlns:tools="http://schemas.android.com/tools">
5
6          <data>
7              <variable
8                  name="bio"
9                  type="com.logan.bio.data.Bio" />
10         </data>
11
12         <androidx.appcompat.widget.LinearLayoutCompat
13             android:layout_width="match_parent"
14             android:layout_height="match_parent"
15             android:background="#2196F3"
16             android:orientation="vertical"
17             tools:context=".MainActivity">
18
19             <ImageView
```

4° 再將 @string 的部分替換成 @ { bio, name
bio, hobbies }

activity_main.xml × MainActivity.java × Bio.java × build.gradle (:app) ×

```
Q bio
  × Cc W * 6 results ↑ ↓ □ + - ✎
  22
  23
  24
  25
  26
  27
  28
  29
  30
  31
  32
  33
  34

<TextView
    android:id="@+id/textView2"
    style="@style/nameStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/aldrich"
    android:text="@{bio.name}" />
```

*. MainActivity.java

```
activity_main.xml × MainActivity.java × Bio.java × build.gradle (app) ×
20
21 public class MainActivity extends AppCompatActivity {
22     private ActivityMainBinding binding;
23     private final Bio bio = new Bio(); // final makes bio become global 宣告 bio obj
24
25     @Override final 讓 bio 成 global
26     protected void onCreate(Bundle savedInstanceState) {
27         super.onCreate(savedInstanceState);
28         binding = DataBindingUtil.setContentView(activity: this, R.layout.activity_main);
29
30         bio.setName("Logan Zhang"); → Bio class setter
31         binding.setBio(bio);
32         binding.doneButton.setOnClickListener(view -> addHobbies(view));
33
34     }
35
36     @
37     public void addHobbies(View view) {
38         bio.setHobbies(String.format("Hobbies: %s",
39             binding.enterHobbies.getText().toString().trim()));
40         binding.hobbiesText.setText(String.format("Hobbies: %s",
41             binding.enterHobbies.getText().toString().trim()));
42         binding.invalidateAll(); 將 bio obj 在 activity_main 中宣告的 data 連結
43         binding.hobbiesText.setVisibility(VISIBLE);
44
45         // Hide keyboard
46         // Make the keyboard disappear once we press done
47         InputMethodManager inputMethodManager = (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
48         inputMethodManager.hideSoftInputFromWindow(view.getWindowToken(), flags: 0);
49     }
}
```

宣告 bio obj

final 讓 bio 成 global

→ Bio class setter

將 bio obj 在 activity_main 中宣告的 data 連結

別忘了這步驟耶 !!!

更改完 bio obj 後，要用 invalidateAll() 更新值 !!!

