

Lab 1: Basic Android Programming Report

學號：0811562

姓名：何祁恩

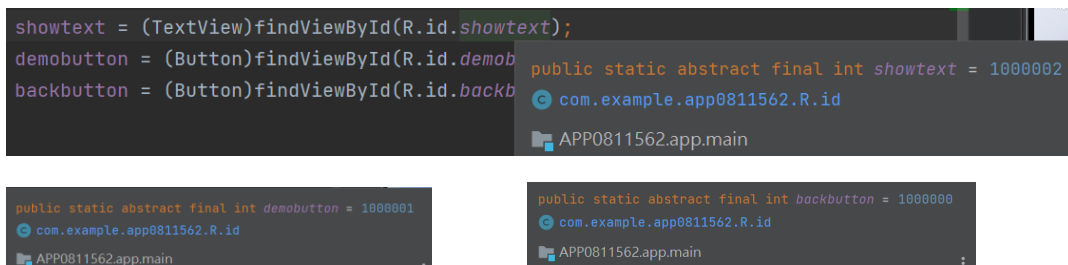
1. 請說明 android 程式如何處理 widget 物件？

把這些 widget 物件(button, TextView)在 layout 上放好之後，透過 findViewById(ID)將 code 與物件連結，因此可以透過 Java code 去對物件進行操作。舉例來說，對於 Q1 Q2 的程式來說，我們可以透過 setter 及 getter 去對物件進行操作。Ex: 透過 showtext.getTextSize()這個 getter 去得到 showtext 裡面這個字串字體的大小。透過 showtext.setTextSize(...)這個 setter 去設置這個字串的字體大小。透過 setter 跟 getter，我們便能夠更動到該 object private 的 member variable。

Android 系統是一種事件驅動(event driven)的程式，載入使用介面之後，便開始等待使用者的操作。那對於按鈕而言，程式是如何運作的呢？按鈕這個物件會持續地監聽他是否有被點擊，在點擊之後，會去執行相對應的 method(OnClick)，接著便能夠完成我們希望使用者按下按鈕後所觸發的事件。

2. 請說明 R.id.button1 是什麼？

在 android 系統中需要透過 id 來將組件編號，舉例來說：將滑鼠放在 R.id.showtext 上便可以看到 android 系統將其編號為 1000002 的整數型別，而後系統便可以透過這個 id 去找到相對應的組件而去執行相對應的動作。



如上圖，對於 demobutton 及 backbutton 也都會有相對應的 id(整數)。因此 R.id.button1 為一個數字，使得程式能夠透過這個 id，去知道是要對哪一個組件進行動作。

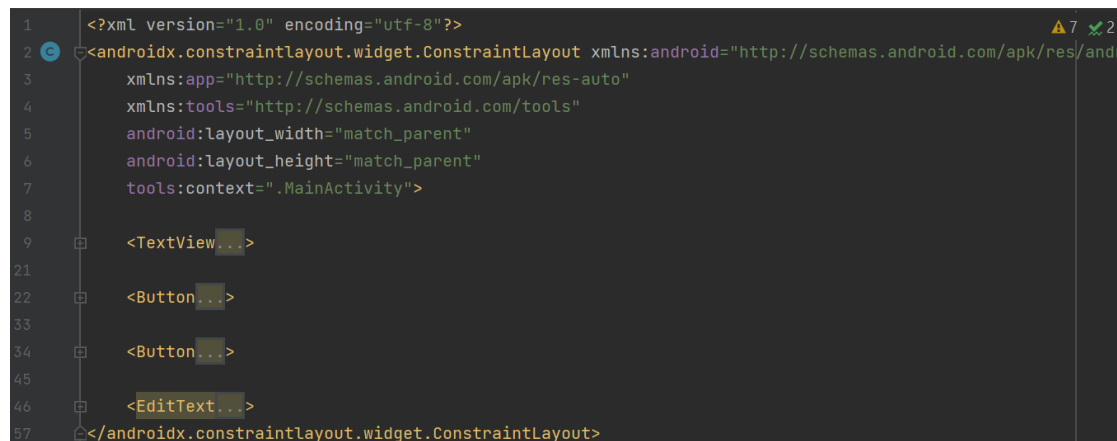
3. 請舉例一個有結尾標籤的元件（可截圖）

下圖中，`androidx.constraintlayout.widget.ConstraintLayout` 便是一個有結尾標籤的元件。可以看到在第 2 行的部分有

`<androidx.constraintlayout.widget.ConstraintLayout`

而後在第 57 行有`</androidx.constraintlayout.widget.ConstraintLayout>`

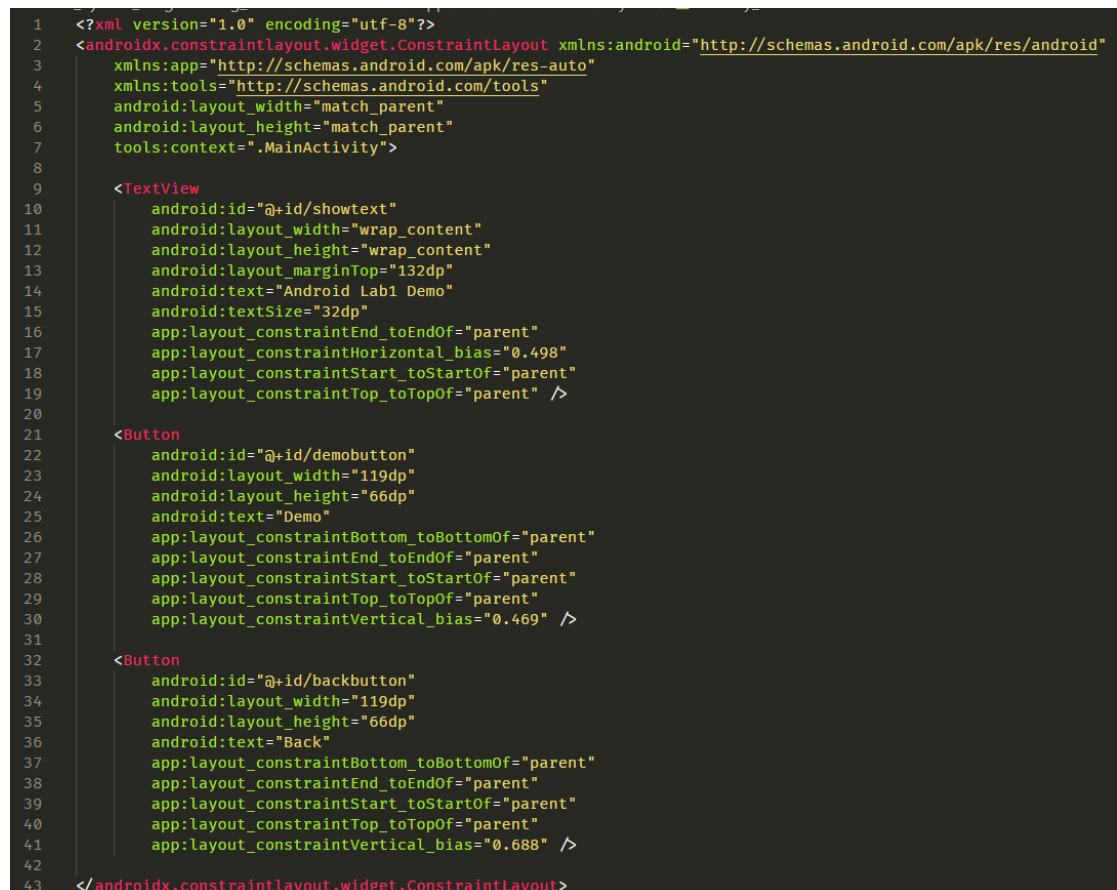
為結尾標籤。



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout 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     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9     <TextView...>
10
21
22     <Button...>
23
33
34     <Button...>
35
45
46     <EditText...>
47
57 </androidx.constraintlayout.widget.ConstraintLayout>
```

4. 請貼上自己的程式碼並附上註解

XML file for Q1 and Q2:



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout 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     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9     <TextView
10         android:id="@+id/showtext"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:layout_marginTop="132dp"
14         android:text="Android Lab1 Demo"
15         android:textSize="32dp"
16         app:layout_constraintEnd_toEndOf="parent"
17         app:layout_constraintHorizontal_bias="0.498"
18         app:layout_constraintStart_toStartOf="parent"
19         app:layout_constraintTop_toTopOf="parent" />
20
21     <Button
22         android:id="@+id/demobutton"
23         android:layout_width="119dp"
24         android:layout_height="66dp"
25         android:text="Demo"
26         app:layout_constraintBottom_toBottomOf="parent"
27         app:layout_constraintEnd_toEndOf="parent"
28         app:layout_constraintStart_toStartOf="parent"
29         app:layout_constraintTop_toTopOf="parent"
30         app:layout_constraintVertical_bias="0.469" />
31
32     <Button
33         android:id="@+id/backbutton"
34         android:layout_width="119dp"
35         android:layout_height="66dp"
36         android:text="Back"
37         app:layout_constraintBottom_toBottomOf="parent"
38         app:layout_constraintEnd_toEndOf="parent"
39         app:layout_constraintStart_toStartOf="parent"
40         app:layout_constraintTop_toTopOf="parent"
41         app:layout_constraintVertical_bias="0.688" />
42
43 </androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java for Q1 and Q2:

```
1  package com.example.app0811562;
2
3  import android.os.Bundle;
4  import android.util.TypedValue;
5  import android.view.View;
6  import android.widget.Button;
7  import android.widget.TextView;
8  import androidx.appcompat.app.AppCompatActivity;
9
10
11 public class MainActivity extends AppCompatActivity {
12
13     //variable declaration
14     TextView showtext;
15     Button demobutton;
16     Button backbutton;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22
23         //connect the object by ID
24         showtext = (TextView)findViewById(R.id.showtext);
25         demobutton = (Button)findViewById(R.id.demobutton);
26         backbutton = (Button)findViewById(R.id.backbutton);
27
28
29         //demo button: increase text size and show "Pass"
30         demobutton.setOnClickListener(new View.OnClickListener(){
31             @Override
32             public void onClick(View v){
33                 int size = (int)showtext.getTextSize();
34                 size = size + 3;
35                 showtext.setTextSize(TypedValue.COMPLEX_UNIT_PX, size);
36                 showtext.setText("Pass");
37             }
38         });
39
40
41         //back button: decrease text size and show "Android Lab1 demo"
42         backbutton.setOnClickListener(new View.OnClickListener(){
43             @Override
44             public void onClick(View v){
45                 int size = (int)showtext.getTextSize();
46                 size = size - 3;
47                 showtext.setTextSize(TypedValue.COMPLEX_UNIT_PX, size);
48                 showtext.setText("Android Lab1 demo");
49             }
50         });
51     }
52 }
```

兩個按鈕都去監聽使用者是否有點擊，因為兩個按鈕都使用相同的函數名稱(setOnClickListener)，因此需要加上 Override，才能使其執行到內部 onClick 的這個 method。當使用者點擊 demo 這個按鈕時，會先透過 showtext 的 getter 得到 showtext 當下的字體大小，而後將字體大小放大三個單位，輸出”Pass”。同樣的方法應用在 back button，只不過始將字體大小縮小三個單位並且輸出”Android Lab1 demo”。

XML file for Q3:


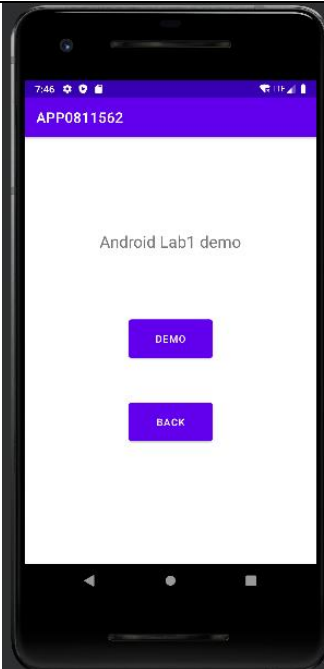
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9     <TextView
10         android:id="@+id/TopText"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="Hello World"
14         android:textSize="28dp"
15         app:layout_constraintBottom_toBottomOf="parent"
16         app:layout_constraintEnd_toEndOf="parent"
17         app:layout_constraintHorizontal_bias="0.498"
18         app:layout_constraintStart_toStartOf="parent"
19         app:layout_constraintTop_toTopOf="parent"
20         app:layout_constraintVertical_bias="0.103" />
21
22     <Button
23         android:id="@+id/setbutton"
24         android:layout_width="wrap_content"
25         android:layout_height="wrap_content"
26         android:text="Set"
27         app:layout_constraintBottom_toBottomOf="parent"
28         app:layout_constraintEnd_toEndOf="parent"
29         app:layout_constraintHorizontal_bias="0.139"
30         app:layout_constraintStart_toStartOf="parent"
31         app:layout_constraintTop_toTopOf="parent"
32         app:layout_constraintVertical_bias="0.256" />
33
34     <Button
35         android:id="@+id/resetbutton"
36         android:layout_width="wrap_content"
37         android:layout_height="wrap_content"
38         android:text="Reset"
39         app:layout_constraintBottom_toBottomOf="parent"
40         app:layout_constraintEnd_toEndOf="parent"
41         app:layout_constraintHorizontal_bias="0.86"
42         app:layout_constraintStart_toStartOf="parent"
43         app:layout_constraintTop_toTopOf="parent"
44         app:layout_constraintVertical_bias="0.256" />
45
46     <EditText
47         android:id="@+id/Name"
48         android:layout_width="wrap_content"
49         android:layout_height="wrap_content"
50         android:ems="10"
51         android:inputType="textPersonName"
52         android:text="Name"
53         app:layout_constraintBottom_toBottomOf="parent"
54         app:layout_constraintEnd_toEndOf="parent"
55         app:layout_constraintStart_toStartOf="parent"
56         app:layout_constraintTop_toTopOf="parent" />
57 </androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java for Q3:

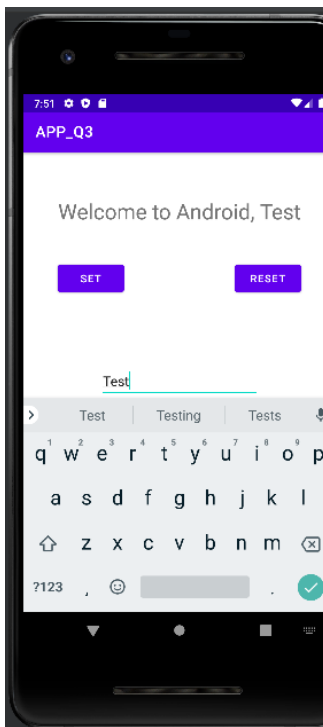
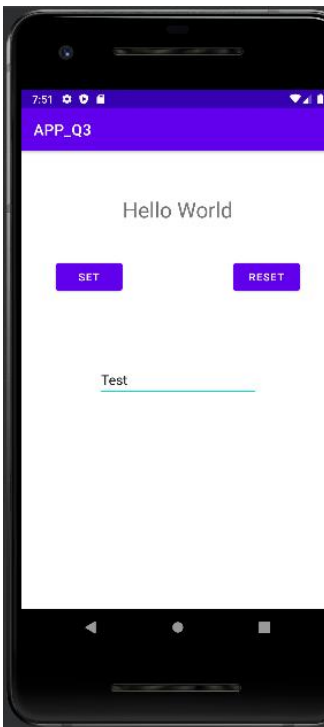
```
1  package com.example.app_q3;
2  import androidx.appcompat.app.AppCompatActivity;
3  import android.os.Bundle;
4  import android.util.TypedValue;
5  import android.view.View;
6  import android.widget.Button;
7  import android.widget.EditText;
8  import android.widget.TextView;
9
10 public class MainActivity extends AppCompatActivity {
11
12     //variable declaration
13     TextView TopText;
14     Button setbutton;
15     Button resetbutton;
16     EditText Name;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22
23         //connect object by ID
24         TopText = (TextView)findViewById(R.id.TopText);
25         setbutton = (Button)findViewById(R.id.setbutton);
26         resetbutton = (Button)findViewById(R.id.resetbutton);
27         Name = (EditText)findViewById(R.id.Name);
28
29
30         //set button: show the "Welcome to Android" with the string on EditText
31         setbutton.setOnClickListener(new View.OnClickListener(){
32             @Override
33             public void onClick(View v){
34                 TopText.setText("Welcome to Android, "+ Name.getText().toString());
35             }
36         });
37
38         //reset button: reset the string in TopText to "Hello World"
39         resetbutton.setOnClickListener(new View.OnClickListener(){
40             @Override
41             public void onClick(View v){
42                 TopText.setText("Hello World");
43             }
44         });
45     }
46 }
47
```

與 Q1 及 Q2 差不多，只不過要顯示的字串改成由使用者經由 EditText 取得，因此只要透過 Name.getText() 這個 getter 取得當前存在 Name 這個物件的資料之後，再將其輸出在上方的 TopText 即可。

Q1/Q2 結果:

初始畫面:	按下 Demo 後字體變大，並且顯示”Pass”:	按下 Back 之後，字體變小，並且顯示”Android Lab1 demo”
		

Q3 結果:

初始畫面:	輸入字串後，上方字串改成打招呼	重置
		

5. 心得

之前就有使用過 Android Studio 來試著做一些 app，大家 Android Studio 的第一個範例一定都是 BMI 計算機，可惜在寫背後的運算式使用的是 JAVA 程式語言，對還不太熟悉物件導向程式設計的我來說不太容易上手。因此當時的我早早就放棄了。到了這次實驗才大致上對 Android Studio 有更進一步的認識，透過拖拉把 app 版面做好，再 Main code 把背後的邏輯寫出來。