

Java 和 Kotlin 語法的差異

Import 物件差異

Kotlin 的 import 寫法 `android.widget.*` 可以減少很多行

Java 語法

```
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6
7 import android.os.Bundle;
8 import android.view.View;
9 import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.RadioButton;
12 import android.widget.TextView;
13
14 import com.example.javaguess.R;
15
16
```

Kotlin 語法

```
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.widget.*
```

類別差異

Java 無須建立類別

Java 語法

```
15
16
17 public class MainActivity extends AppCompatActivity {
18
```

Kotlin 語法

```
17 class MainActivity : AppCompatActivity() {
```

宣告變數差異

Kotlin 用 `val` 宣告變數讓程式碼簡潔

Java 語法

```

19
20     private EditText ed_name;
21     private TextView tv_text, tv_name, tv_winner, tv_mmora, tv_cmora;
22     private RadioButton btn_scissor, btn_stone, btn_paper;
23     private Button btn_mora;
24
25     @Override
26     protected void onCreate(Bundle savedInstanceState) {
27         super.onCreate(savedInstanceState);
28         setContentView(R.layout.activity_main);
29
30         ed_name = findViewById(R.id.ed_name);
31         tv_text = findViewById(R.id.tv_text);
32         tv_name = findViewById(R.id.tv_name);
33         tv_winner = findViewById(R.id.tv_winner);
34         tv_mmora = findViewById(R.id.tv_mmora);
35         tv_cmora = findViewById(R.id.tv_cmora);
36         btn_scissor = findViewById(R.id.btn_scissor);
37         btn_stone = findViewById(R.id.btn_stone);
38         btn_paper = findViewById(R.id.btn_paper);
39         btn_mora = findViewById(R.id.btn_mora);
40         btn_mora.setOnClickListener(new View.OnClickListener() {

```

Kotlin 語法

```

9 override fun onCreate(savedInstanceState: Bundle?) {
10     super.onCreate(savedInstanceState)
11     setContentView(R.layout.activity_main)
12     val ed_name = findViewById<EditText>(R.id.ed_name)
13     val tv_text = findViewById<TextView>(R.id.tv_text)
14     val tv_name = findViewById<TextView>(R.id.tv_name)
15     val tv_winner = findViewById<TextView>(R.id.tv_winner)
16     val tv_mmora = findViewById<TextView>(R.id.tv_mmora)
17     val tv_cmora = findViewById<TextView>(R.id.tv_cmora)
18     val radioGroup = findViewById<RadioGroup>(R.id.radioGroup)
19     val btn_scissor = findViewById<RadioButton>(R.id.btn_scissor)
20     val btn_stone = findViewById<RadioButton>(R.id.btn_stone)
21     val btn_papper = findViewById<RadioButton>(R.id.btn_papper)
22     val btn_mora = findViewById<Button>(R.id.btn_mora)

```

監聽差異

Java 需要 new 來建立物件

Java 語法

```

40 btn_mora.setOnClickListener(new View.OnClickListener() {

```

Kotlin 語法

```

22
23 btn_mora.setOnClickListener{ // View?

```

判斷式用法差異一

Kotlin 的 Ischecked 後面無需打()輸出文字可以

減少 tv_name

Java 語法

```

41         @Override
42         public void onClick(View v) {
43             //判斷字串是否是空白來要求輸入姓名
44             if (ed_name.length() < 1)
45                 tv_text.setText("請輸入玩家姓名");
46             else {
47                 //顯示玩家姓名與我方出拳
48                 tv_name.setText(String.format("名字\n%s",
49                     ed_name.getText().toString()));
50                 if (btn_scissor.isChecked())
51                     tv_mmora.setText("我方出拳\n剪刀");
52                 else if (btn_stone.isChecked())
53                     tv_mmora.setText("我方出拳\n石頭");
54                 else
55                     tv_mmora.setText("我方出拳\n布");
56                 //Random 產生介於0~1間不含1的亂數,乘3產生0~2當作電腦的出拳
57                 int computer = (int) (Math.random() * 3);
58                 if (computer == 0)
59                     tv_cmora.setText("電腦出拳\n剪刀");
60                 else if (computer == 1)
61                     tv_cmora.setText("電腦出拳\n石頭");
62                 else
63                     tv_cmora.setText("電腦出拳\n布");

```

Kotlin 語法

```

24         if(ed_name.length()<1)
25             tv_text.text="請輸入玩家姓名"
26         else{
27             //顯示名字
28             tv_name.text="名字\n${ed_name.text}"
29             //顯示我方出拳
30             tv_mmora.text = "我方出拳\n${if(btn_scissor.isChecked())"剪刀"
31             else if(btn_stone.isChecked())"石頭" else "布"}"
32             //電腦出拳,亂數選擇0、1、2
33             val computer = (Math.random()*3).toInt()
34             //顯示電腦出拳
35             tv_cmora.text = "電腦出拳\n${if(computer==0)"剪刀"
36             else if(computer==1)"石頭" else "布"}"
37             //三個判斷式判斷結果

```

判斷式用法差異二

Kotlin 的 when 使用有點像是 java 的 if else 用法看起來更簡潔

Java 語法

```

64         //用三個判斷式判斷並顯示猜拳結果
65         if ((btn_scissor.isChecked() && computer == 2) ||
66             (btn_stone.isChecked() && computer == 0) ||
67             (btn_paper.isChecked() && computer == 1)) {
68             tv_winner.setText("勝利者\n" +
69                 ed_name.getText().toString());
70             tv_text.setText("恭喜你獲勝了!!!");
71         } else if ((btn_scissor.isChecked() && computer == 1) ||
72             (btn_stone.isChecked() && computer == 2) ||
73             (btn_paper.isChecked() && computer == 0)) {
74             tv_winner.setText("勝利者\n電腦");
75             tv_text.setText("可惜,電腦獲勝了!");
76         } else {
77             tv_winner.setText("勝利者\n平手");
78             tv_text.setText("平局,請再試一次!");
79         }
80     }
81 }
82 });
83 }
84 }

```

Kotlin 語法

```
38         when{
39             btn_scissor.isChecked && computer==2 ||
40             btn_stone.isChecked && computer==0 ||
41             btn_papper.isChecked && computer==1 ->{
42                 tv_winner.text = "勝利者\n${ed_name.text}"
43                 tv_text.text = "你贏了好棒棒!!"
44             }
45             btn_scissor.isChecked && computer==1 ||
46             btn_stone.isChecked && computer==2 ||
47             btn_papper.isChecked && computer==0 ->{
48                 tv_winner.text = "勝利者\n電腦"
49                 tv_text.text = "你輸了幫哭哭"
50             }
51             else ->{
52                 tv_winner.text = "勝利者\n平手"
53                 tv_text.text = "平局，再來一次"
54             }
55         }
56     }
57 }
58 }
59 }
```

心得：

在使用 kotlin 和 java 後，他們兩者的程式碼差異蠻多的，發現 kotlin 的程式碼簡潔整齊了很多，閱讀也比較容易。