

流程控制

TYIC桃高資訊社

流程控制(Flow Control)

流程控制，顧名思義，就是控制程式執行的流程、順序
在 **Java** 中，能控制流程的有：

if...else、**switch**、**for**、**while**、**continue**、**break**

這些是**控制流程陳述式(control flow statement)**

與表達陳述式和宣告陳述式不同的是，流程控制陳述式不用分號

if...else

if...else 是用來處理在特定情況下才執行的程式碼

```
if (boolean(條件)) {  
    陳述式...  
} else {  
    陳述式...  
}
```

java

在 **Java** 中，一對大括號表示一個區塊(**block**)

當條件為真時才會執行 **if** 後方區塊的程式碼

否則就會執行 **else** 後方區塊裡的程式碼

else 部分可以省略，省略時如果條件不為真就不會做任何事


if...else

若 **if**、**else** 後方的區塊內只有一行陳述式
則可以不寫程式碼區塊，直接寫陳述式
所以可以撰寫下方這種 **if...else if...else** 的語法

```
if (boolean(條件)) {  
    陳述式...  
} else if (boolean(條件)) {  
    陳述式...  
} else {  
    陳述式...  
}
```

java

但除非是 **if...else if...else** 的語法
否則強烈建議使用區塊，避免閱讀錯誤

CVE-2014-1266 

if...else

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String input = scanner.next();
        if (input.equals("回憶過去")) {
            System.out.println("痛苦的相思忘不了");
        } else if (input.equals("想念你的笑")) {
            System.out.println("想念你的外套");
        } else if (input.equals("怎麼忍心怪你犯了錯")) {
            System.out.println("是我給你自由過了火");
        } else if (input.equals("我願變成童話裡你愛的那個天使")) {
            System.out.println("張開雙手變成翅膀守護你");
        } else {
            System.out.println("我不知道下一句是什麼");
        }
    }
}
```



java

回憶過去
痛苦的相思忘不了

output

想念你的笑
想念你的外套

output

怎麼忍心怪你犯了錯
是我給你自由過了火

output

我願變成童話裡你愛的那個天使
張開雙手變成翅膀守護你

output

我是真的不能控制我自己
我不知道下一句是什麼

output

switch 流程控制陳述式

switch 的作用是根據不同的傳入值做不同的事

```
switch (傳入值) {  
    case 條件值:  
        陳述式...  
    case 條件值:  
        陳述式...  
    default:  
        陳述式...  
}
```

java

其中 **case** 可以有若干個，而 **default** 可以省略
當傳入值和某個條件值相等時，便會從相等的條件值那行開始
往下執行直到 **switch** 結束，而不管中間的條件值是否相等

switch 流程控制陳述式

在 `switch` 中
可以使用 `break`
讓 `switch` 立刻結束
從而避免 `switch`
一直往下執行的情況

與上個程式碼同樣的功能
在這種情況下寫成 `switch`
比剛剛的 `if...else`
更容易閱讀



```
import java.util.Scanner;

public class Main1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String input = scanner.next();
        switch (input) {
            case "回憶過去":
                System.out.println("痛苦的相思忘不了");
                break;
            case "想念你的笑":
                System.out.println("想念你的外套");
                break;
            case "怎麼忍心怪你犯了錯":
                System.out.println("是我給你自由過了火");
                break;
            case "我願變成童話裡你愛的那個天使":
                System.out.println("張開雙手變成翅膀守護你");
                break;
            default:
                System.out.println("我不知道下一句是什麼");
        }
    }
}
```

java

switch 流程控制陳述式

在 Java 14 中，**switch** 支援了一個 **case** 多個條件值以及可以使用**箭頭** **"->"** 來替代冒號

若使用箭頭，**switch** 只會執行相等條件值箭頭後方的區塊或陳述式而不會像是用冒號時會一直往下執行，也就不需要 **break**

```
import java.util.Scanner;

public class Main2 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String input = scanner.next();
        switch (input) {
            case "回憶過去" -> System.out.println("痛苦的相思忘不了");
            case "想念你的笑" -> System.out.println("想念你的外套");
            case "怎麼忍心怪你犯了錯" -> System.out.println("是我給你自由過了火");
            case "我願變成童話裡你愛的那個天使" -> System.out.println("張開雙手變成翅膀守護你");
            default -> System.out.println("我不知道下一句是什麼");
        }
    }
}
```



java