

程式設計概論

# PROGRAMMING 101

2021/03/26

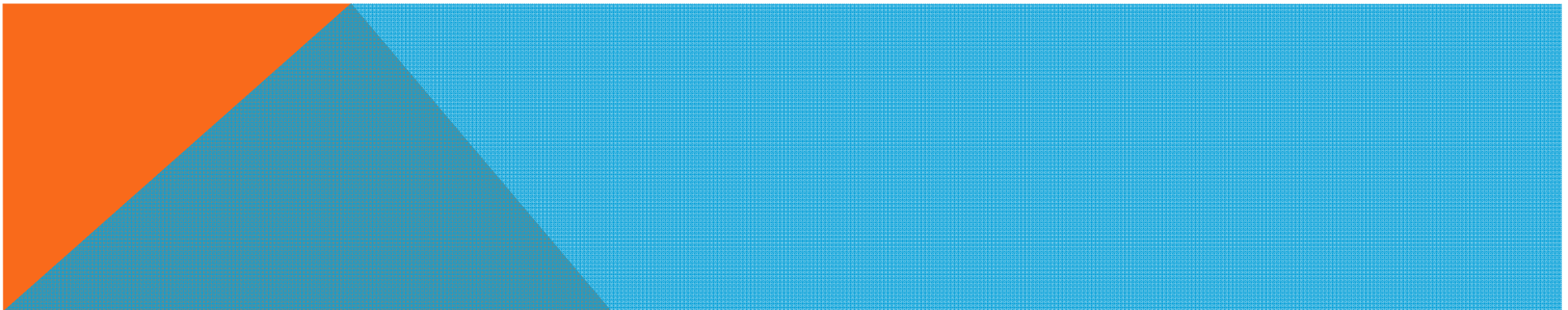
江胡慧










# 流程圖與程式結構

# 流程圖的功用

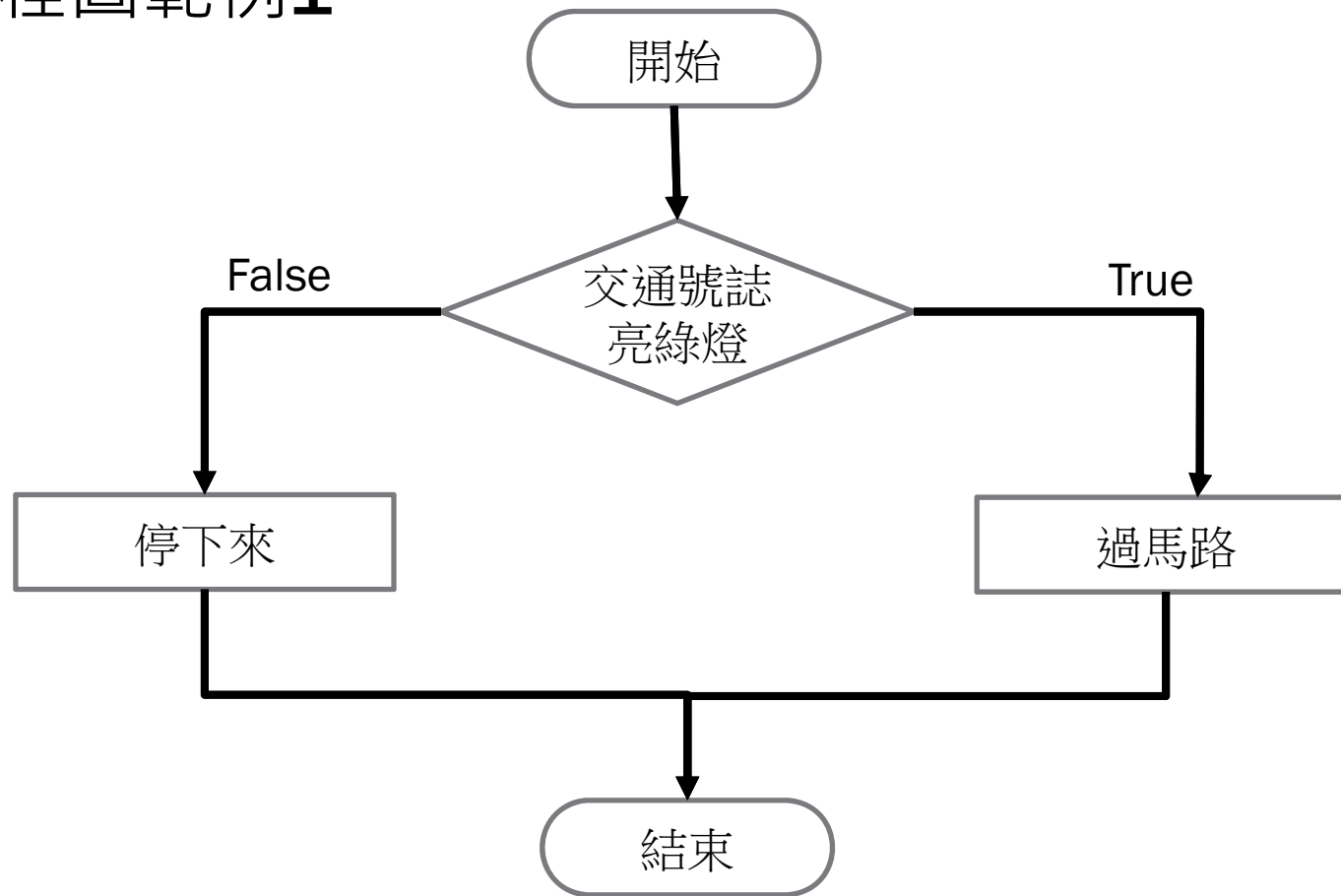
- 以圖形化的呈現方式來摘要程式流程。
- 可在撰寫程式碼前，做為事先規劃的工具。
- 可概覽程式流程的整體結構，做為與他人溝通程式流程的輔助工具。



# 流程圖符號

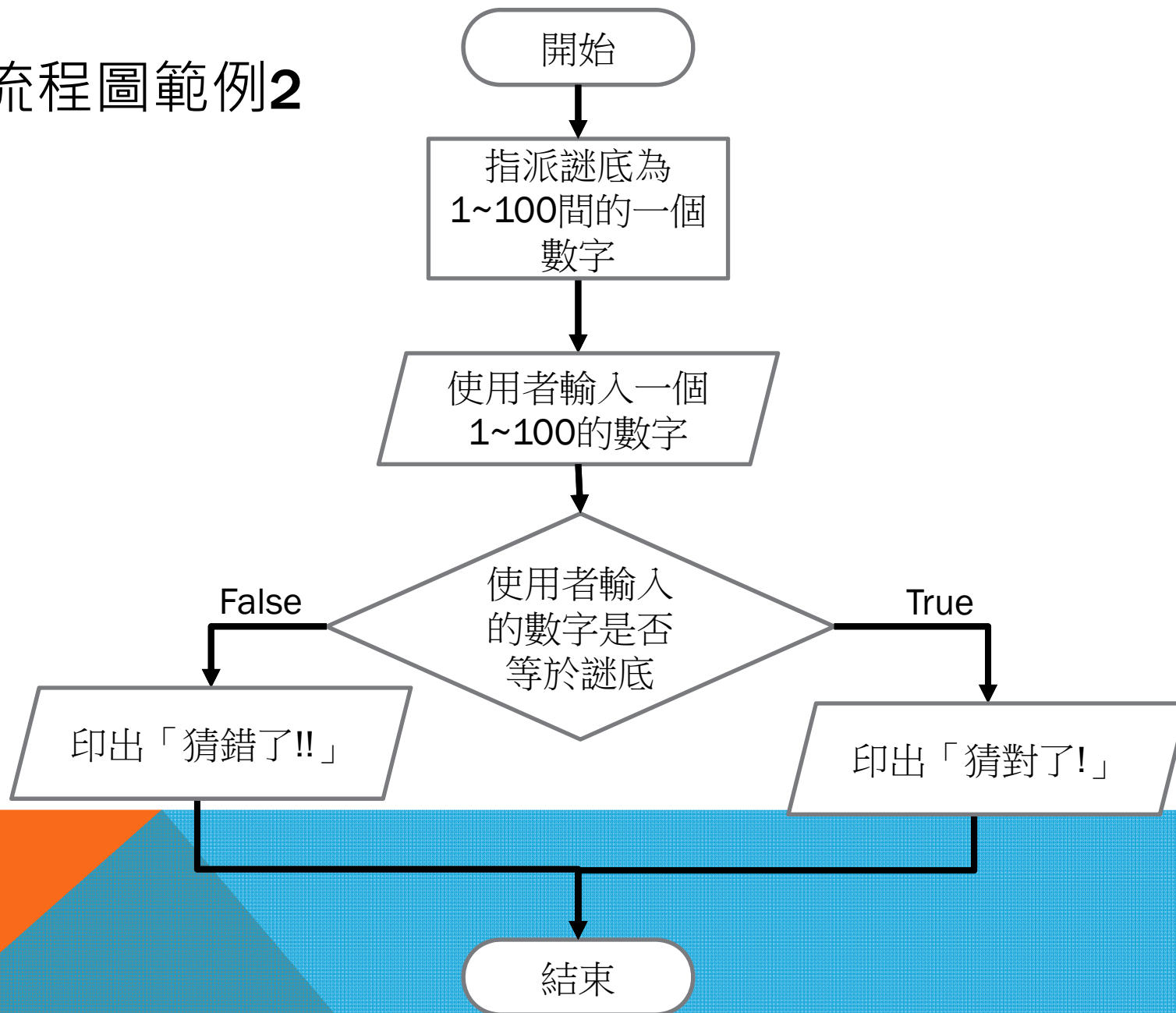
名稱	符號圖形	意義
開始或結束符號		表示程式流程的開始或結束
流程符號		表示程式流程的進行方向
程序處理符號		表示要進行處理的工作
輸入或輸出符號		表示資料輸入或結果輸出
決策判斷符號		表示根據條件式來判斷程式接下來的進行方向
同頁連接點		連接同一頁的流程
跨頁連接點		跨頁連接流程圖

## 流程圖範例1



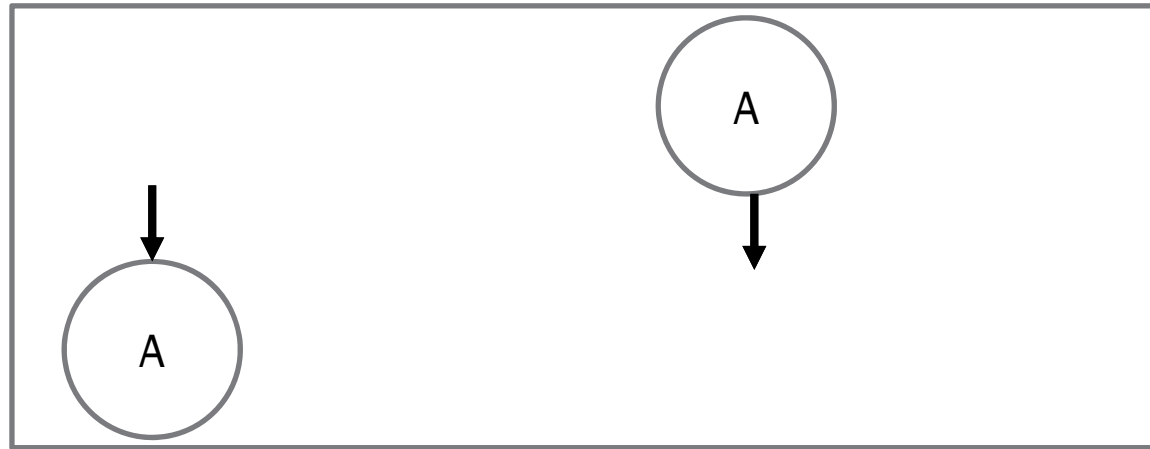


## 流程圖範例2

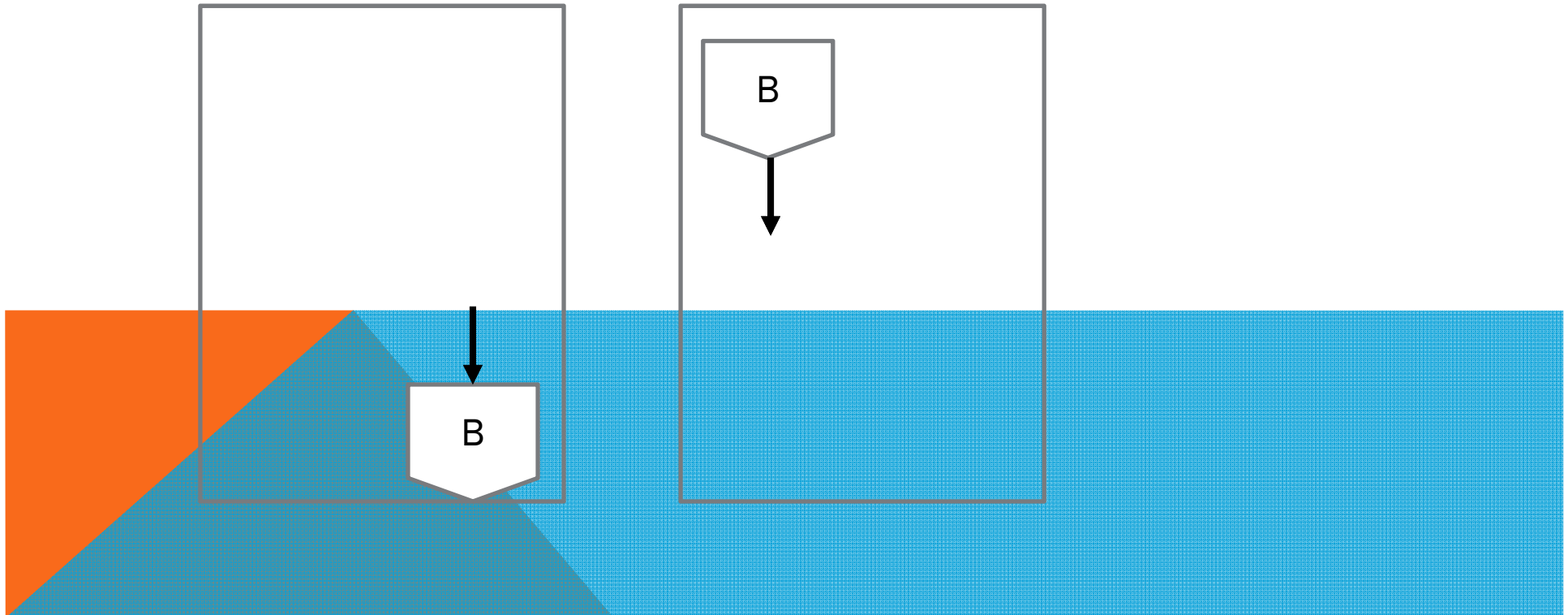


連接點

同頁連接點

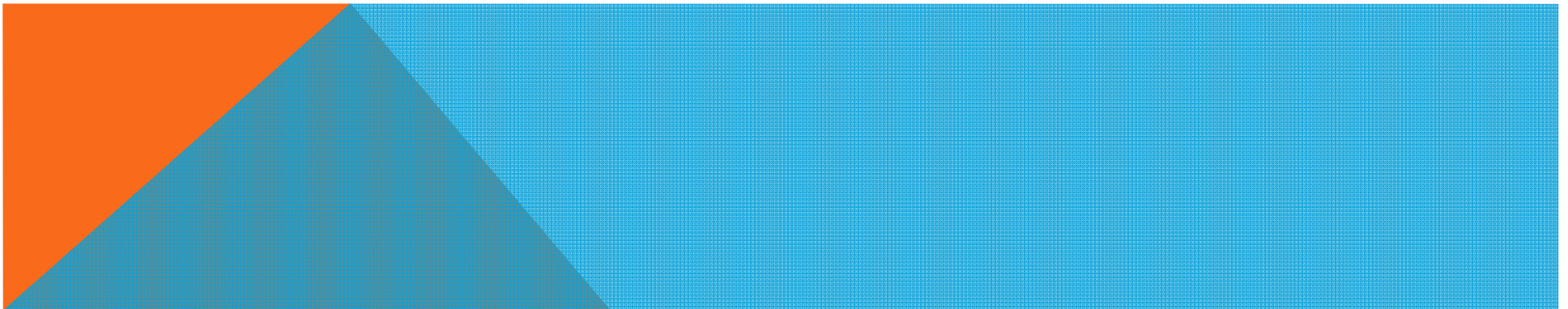


跨頁連接點



# 流程圖軟體

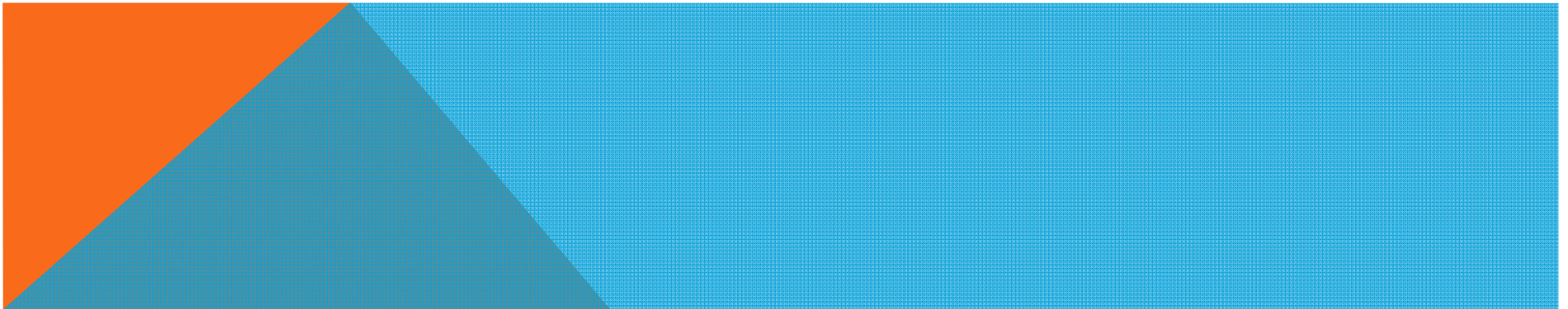
- Google Drawing <https://docs.google.com/drawings>
- 政大雲→文書處理→Visio 2013→基本流程圖



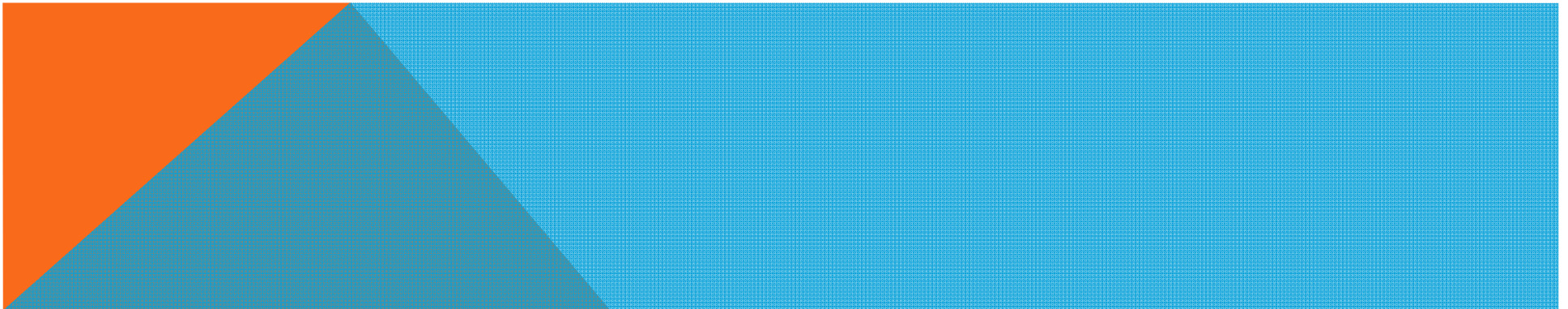
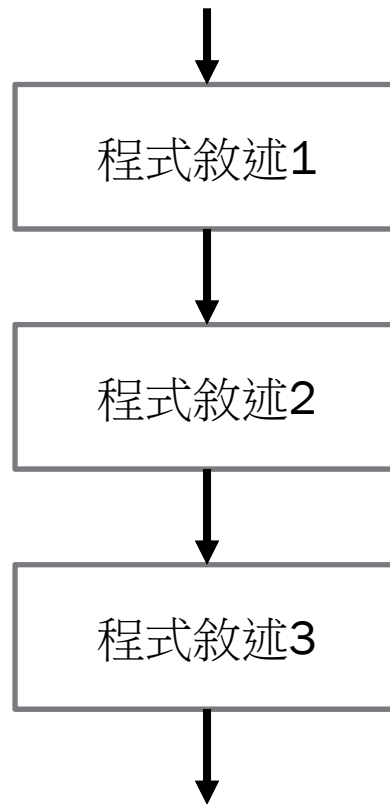


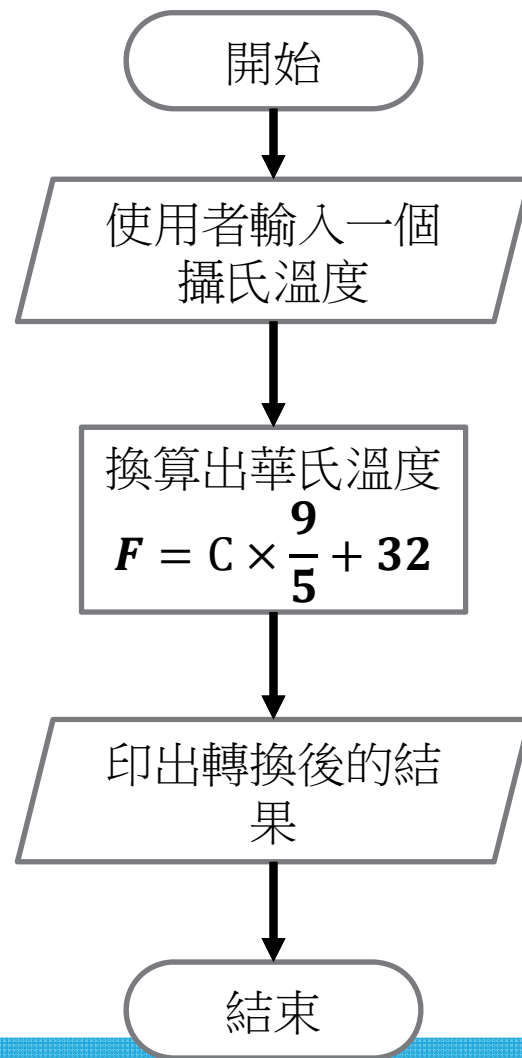
# 三種基本程式結構

- 循序結構
- 選擇結構
- 重複結構

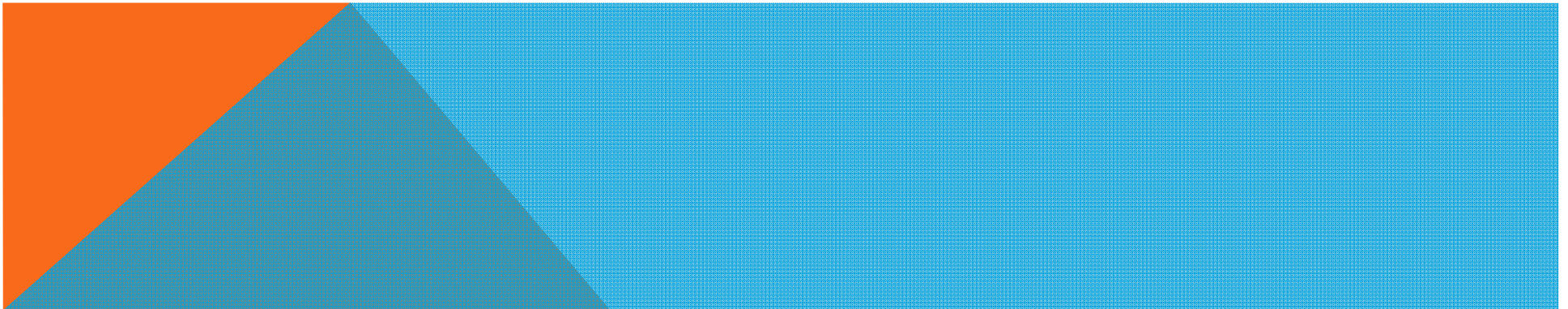
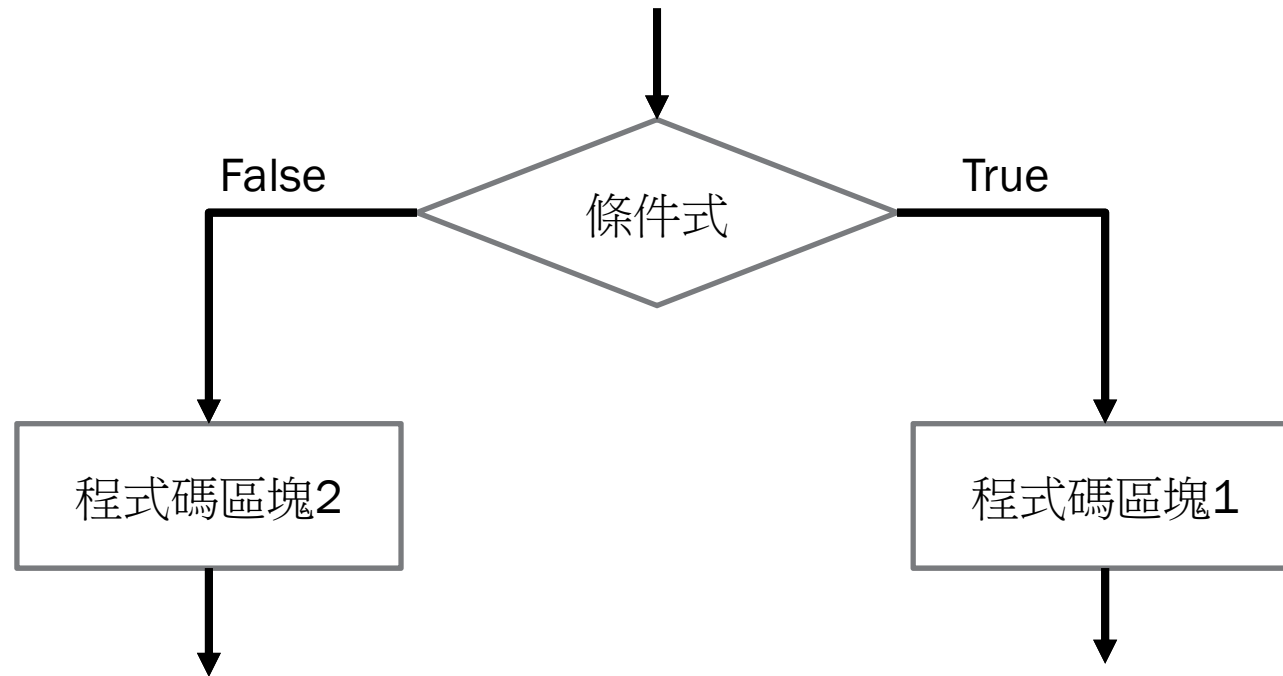


# 循序結構





# 選擇結構



# 選擇結構的程式碼語法

通常為  
比較運算式  
或  
邏輯運算式

if 條件式：

當條件式~~成立~~時 (運算結果為布林值的True) 時，要做的事情 (程式碼區塊1)

else：

當條件式~~不成立~~時 (運算結果為布林值的False) 時，要做的事情 (程式碼區塊2)

```
a=5; b=10
```

```
if a>b:
```

```
    print("變數a的值比變數b的值大")
```

```
    print("a=%4d"%(a))
```

```
    print("b=%4d"%(b))
```

```
else:
```

```
    print("變數a的值沒有比變數b的值大")
```

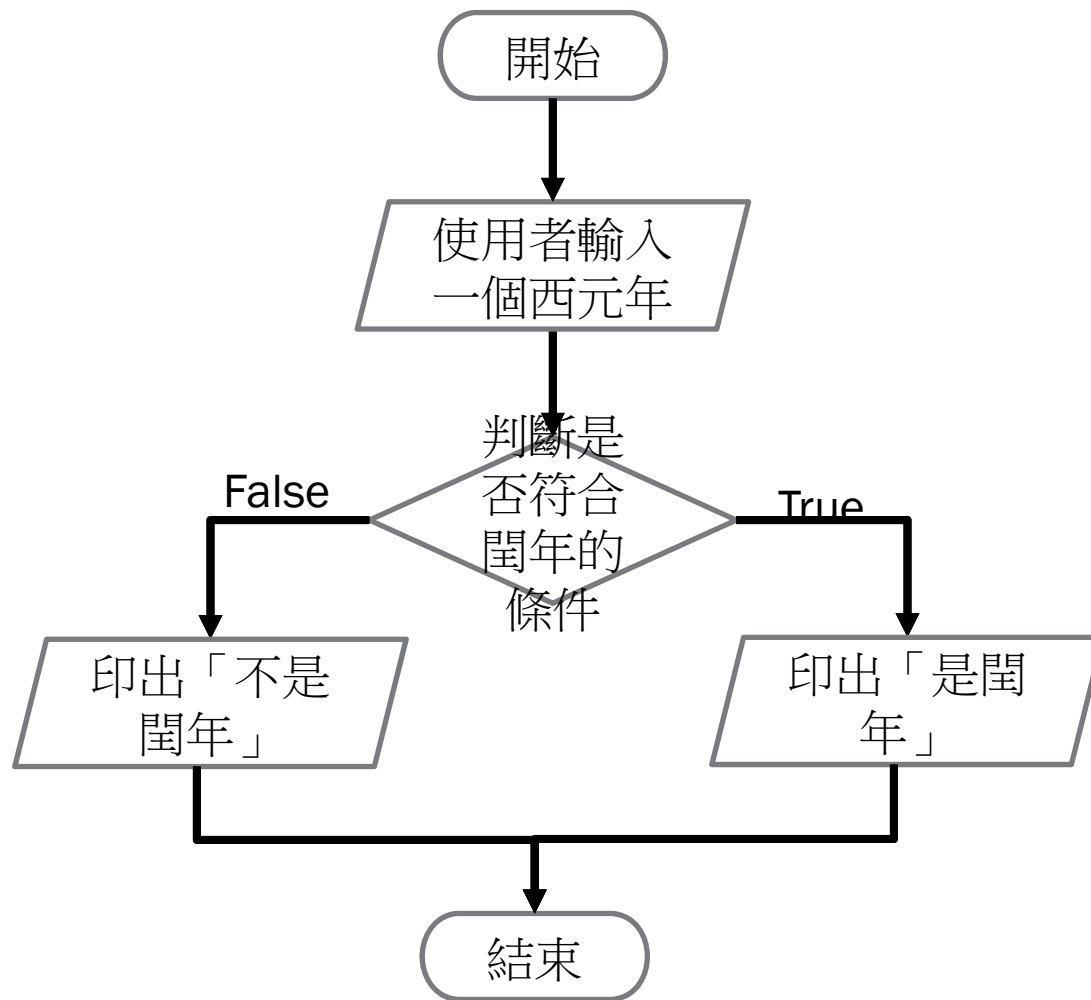
```
    print(f"a={a:10.4f}")
```

```
    print(f"b={b:10.4f}")
```

格式化字串(第一種寫法)  
d: 將資料以十進位整數格式顯示  
4: 保留4個寬度來顯示資料

格式化字串(第二種寫法)  
f: 將資料以浮點數格式顯示  
10.4: 保留10個寬度來顯示資料，  
並顯示到小數點以下第4位





```
years= input("請輸入一個西元年：")
year=int(years)
isLeapYear = (year % 4 == 0 and year % 100 != 0) or (year % 400 ==0)

if isLeapYear:
    print ("西元",year,"年是閏年")
else:
    print ("西元",year,"年不是閏年")
```

# 【實作練習】1

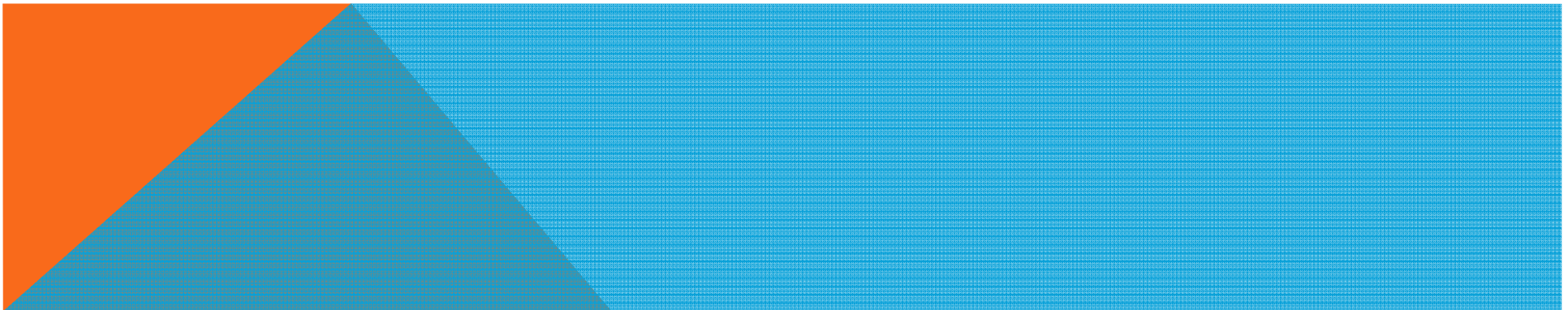
(1) 請畫出以下需求的程式流程圖：

- 讓使用者輸入目前額溫。
- 如果使用者輸入的額溫高於37.5度(不包含37.5度)，則印出「體溫過高!!!」的警訊
- 如果額溫介於36.5~37.5度，則印出「體溫正常」
- 如果額溫低於36.5度(不包含36.5度)，則印出「體溫過低!!!」

將完成的流程圖存成 **week05\_ex1\_flowchart\_你的學號.pdf** 之後上傳到moodle的【實作練習】檔案上傳區

(2) 請依照 (1) 劃出的流程圖，完成程式碼。

將完成的程式碼存成 **week05\_ex1\_你的學號.py** 並壓縮後，上傳到moodle的【實作練習】檔案上傳區

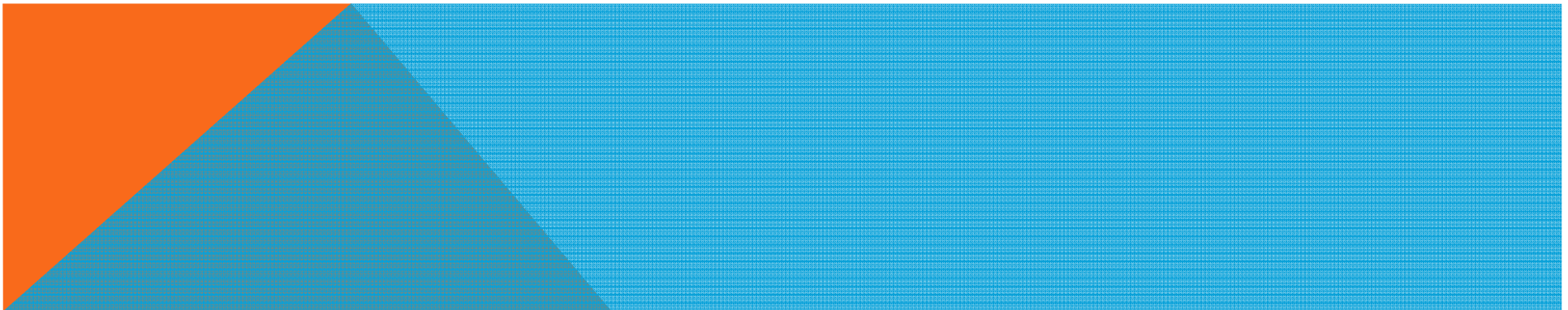


## 測試情況：

Case 1: 體溫過高的情況：輸入38

Case 2: 體溫正常的情况：輸入 36.5

Case 3: 體溫過低的情況：輸入 35

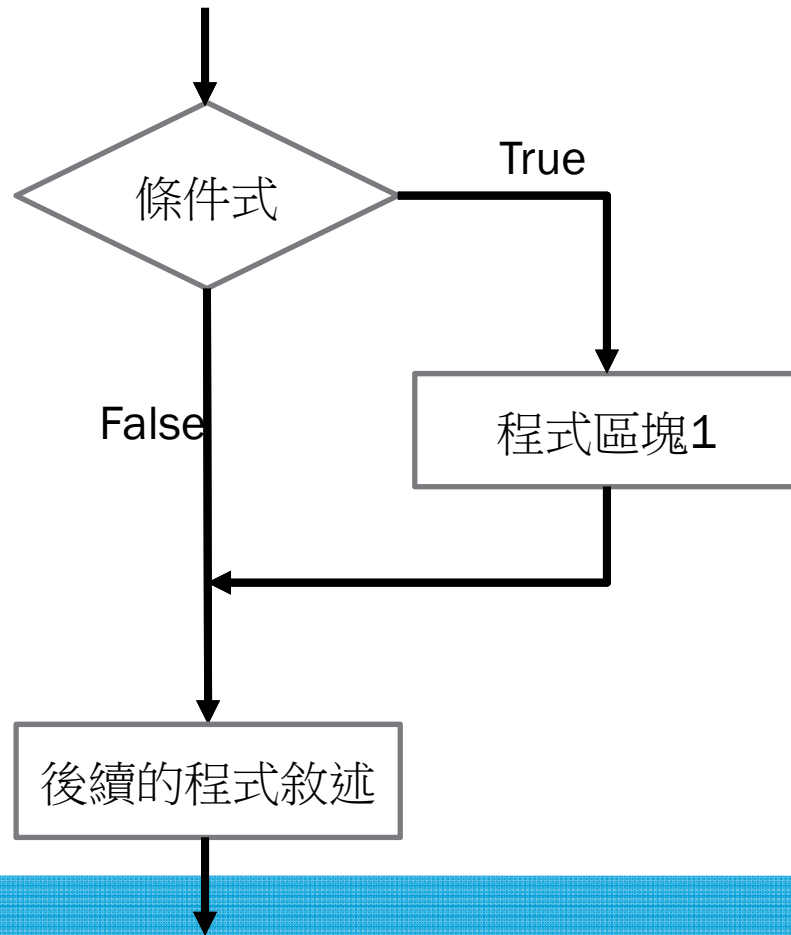


條件判斷



# IF...

if 條件式 :  
    程式區塊1





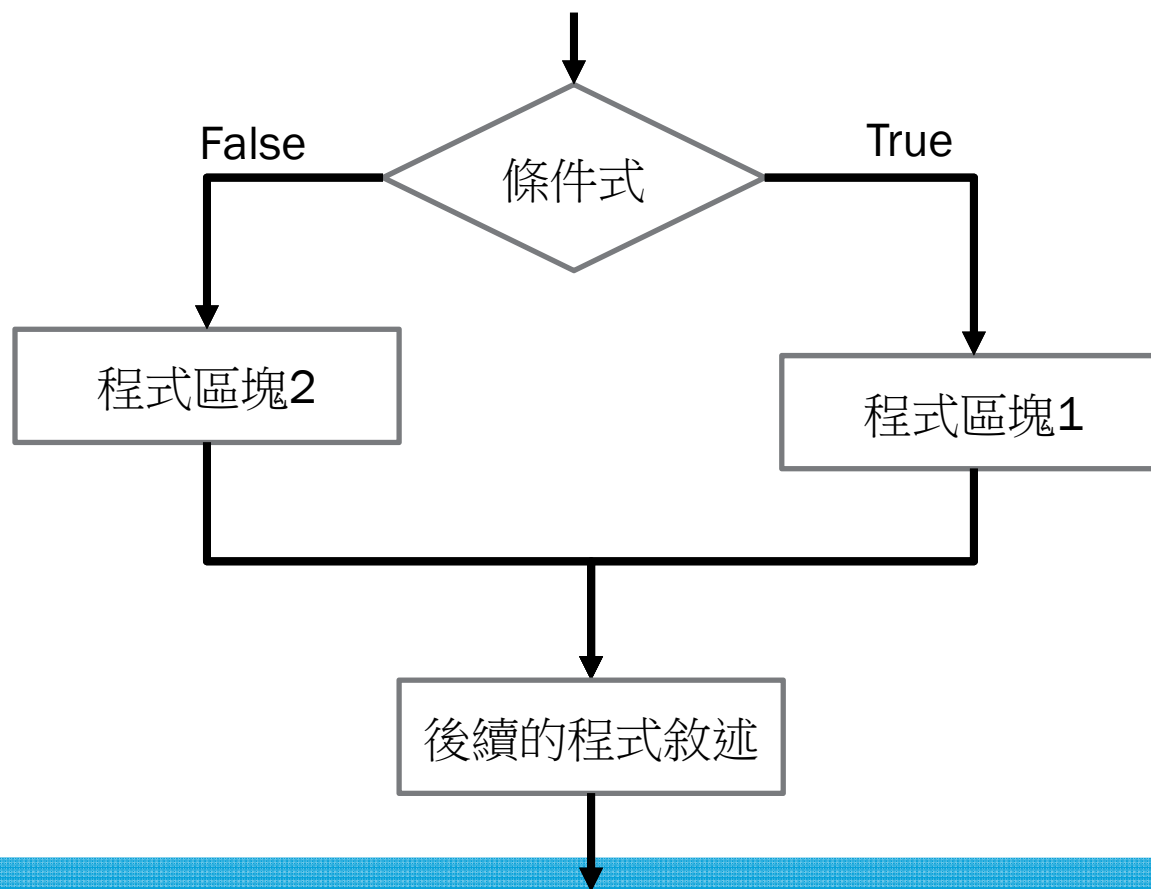
# IF....ELSE....

if 條件式:

    程式區塊1

else:

    程式區塊2



# 巢狀條件判斷

if 條件式1 :

if 條件式2:

程式區塊1

else:

程式區塊2

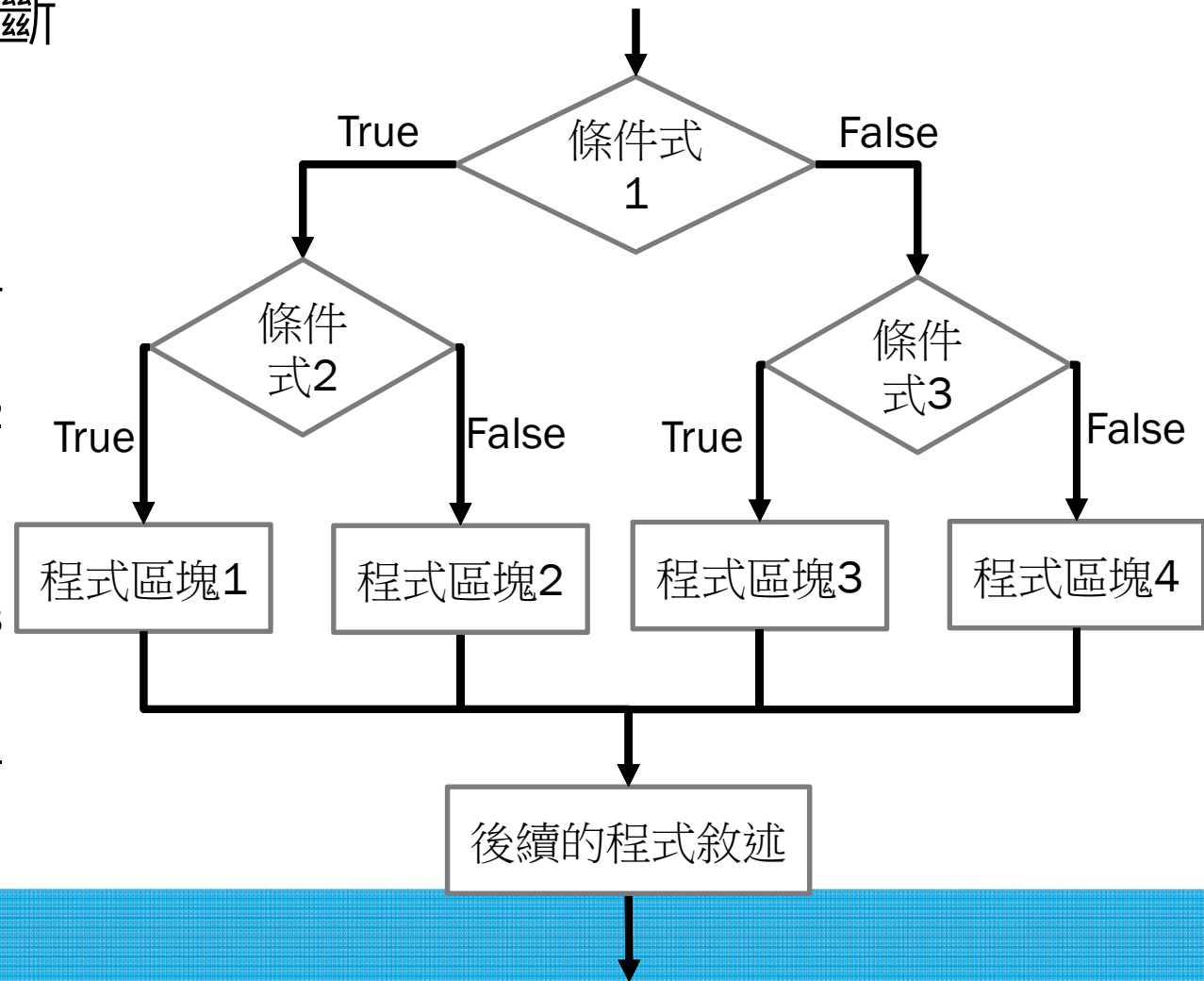
else:

if 條件式3:

程式區塊3

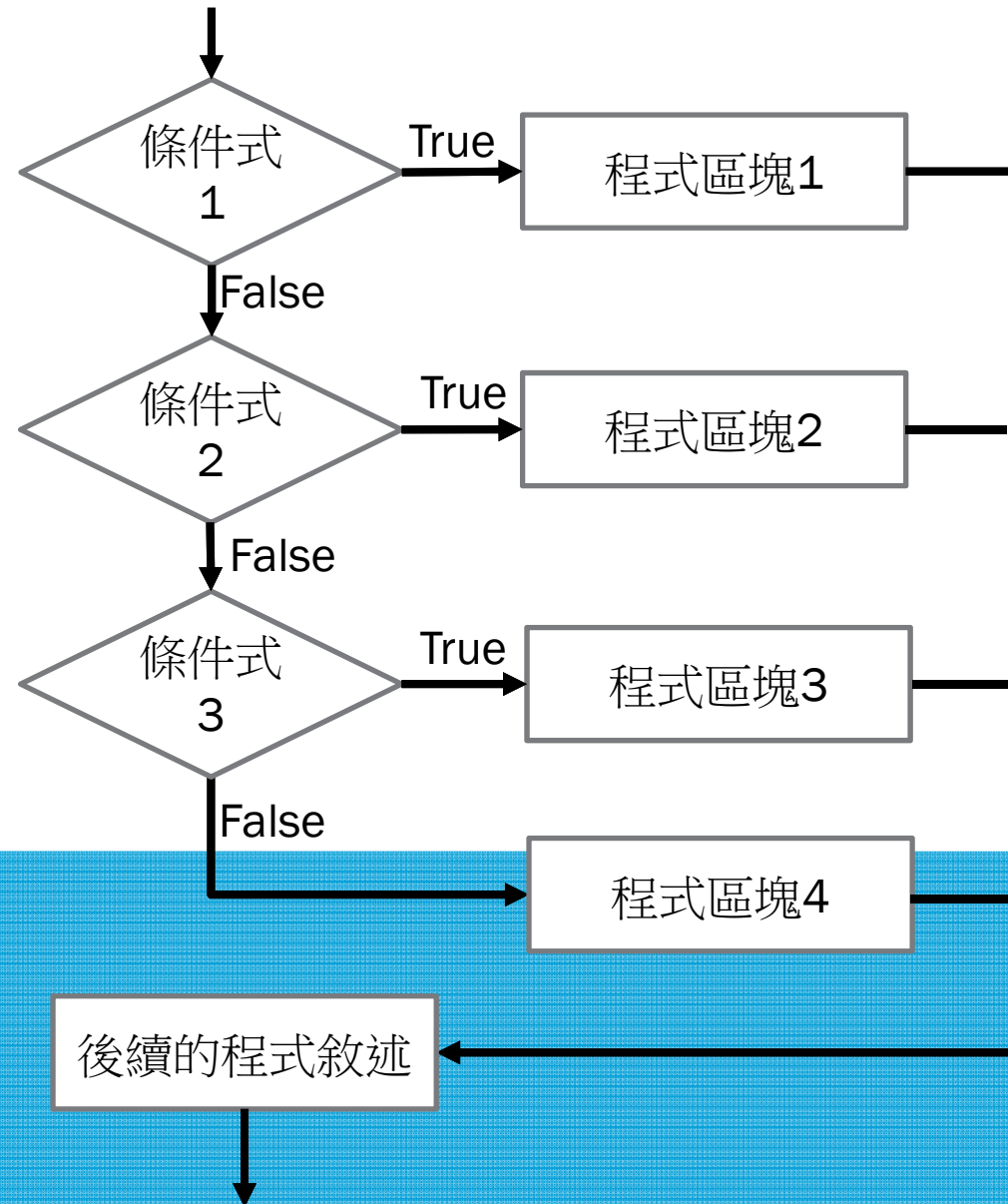
else:

程式區塊4



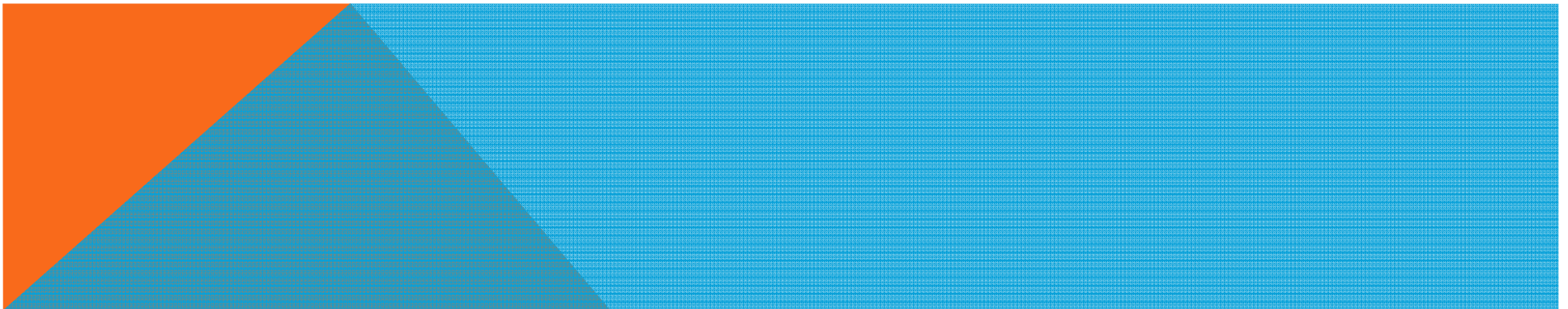
# IF...ELIF..ELSE...

```
if 條件式1:  
    程式區塊1  
elif 條件式2 :  
    程式區塊2  
elif 條件式3 :  
    程式區塊3  
else :  
    程式區塊4
```



# 條件判斷的語法組合

- 一個 `if`
- 零個以上的 `elif`
- 零個或一個 `else`



## 【小組討論】

請與組員討論：

1. 【Code 1】 & 【Code 2】 的差異
2. 【Code 1】 & 【Code 3】 的差異

請將討論結果回應於Moodle的【小組討論】討論區

### 【Code 1】

```
if 條件式1:  
    程式區塊1  
elif 條件式2 :  
    程式區塊2  
elif 條件式3 :  
    程式區塊3  
else :  
    程式區塊4
```

### 【Code 2】

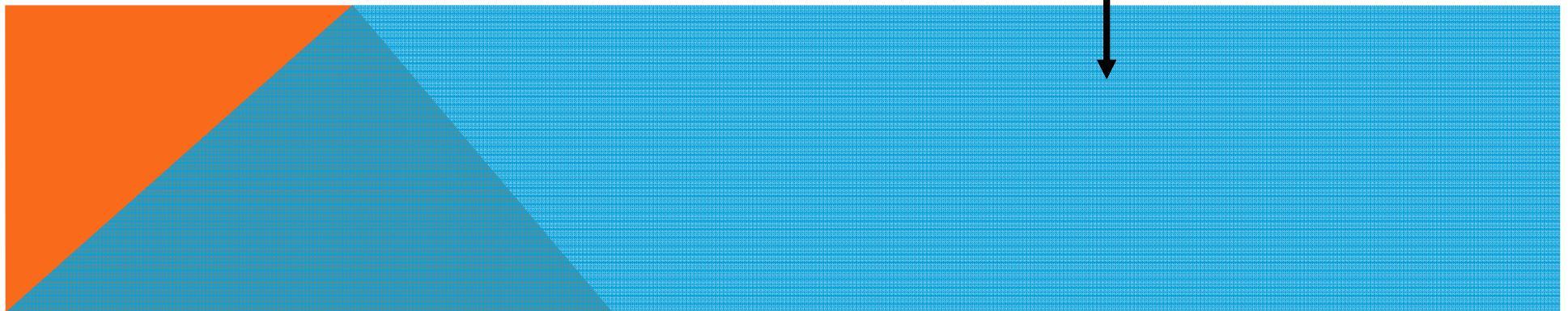
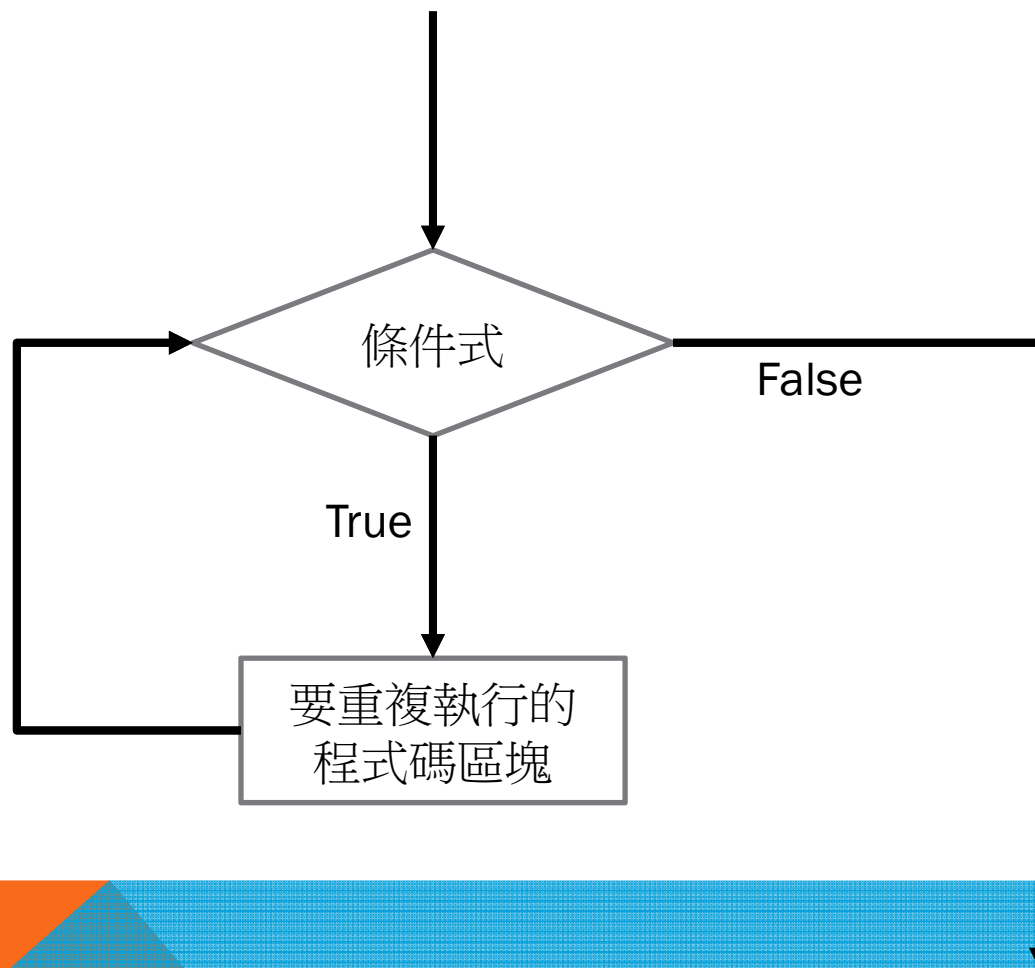
```
if 條件式1:  
    程式區塊1  
else:  
    if 條件式2 :  
        程式區塊2  
    else:  
        if 條件式3 :  
            程式區塊3  
        else :  
            程式區塊4
```

### 【Code 3】

```
if 條件式1:  
    程式區塊1  
if 條件式2 :  
    程式區塊2  
if 條件式3 :  
    程式區塊3  
else :  
    程式區塊4
```



# 重複結構



## 重複結構的程式碼語法：

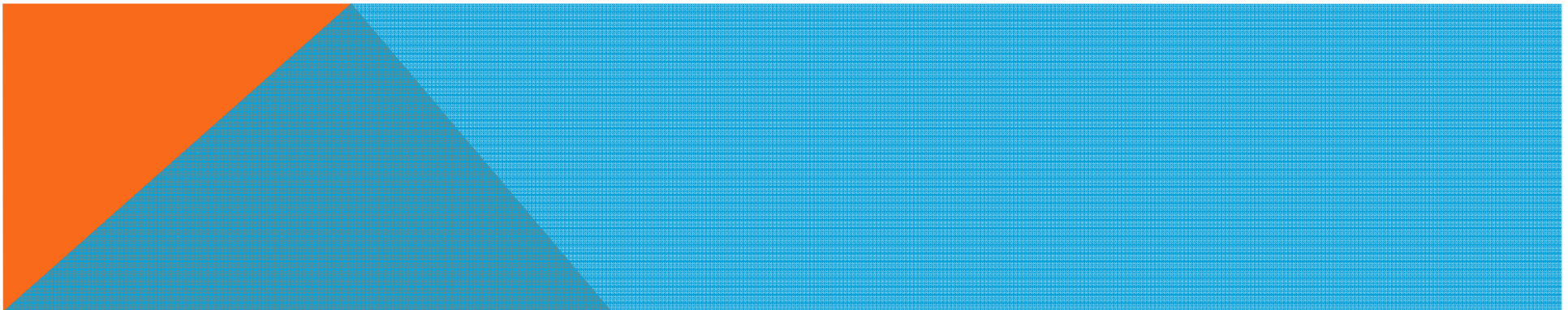
通常為  
比較運算式  
或  
邏輯運算式

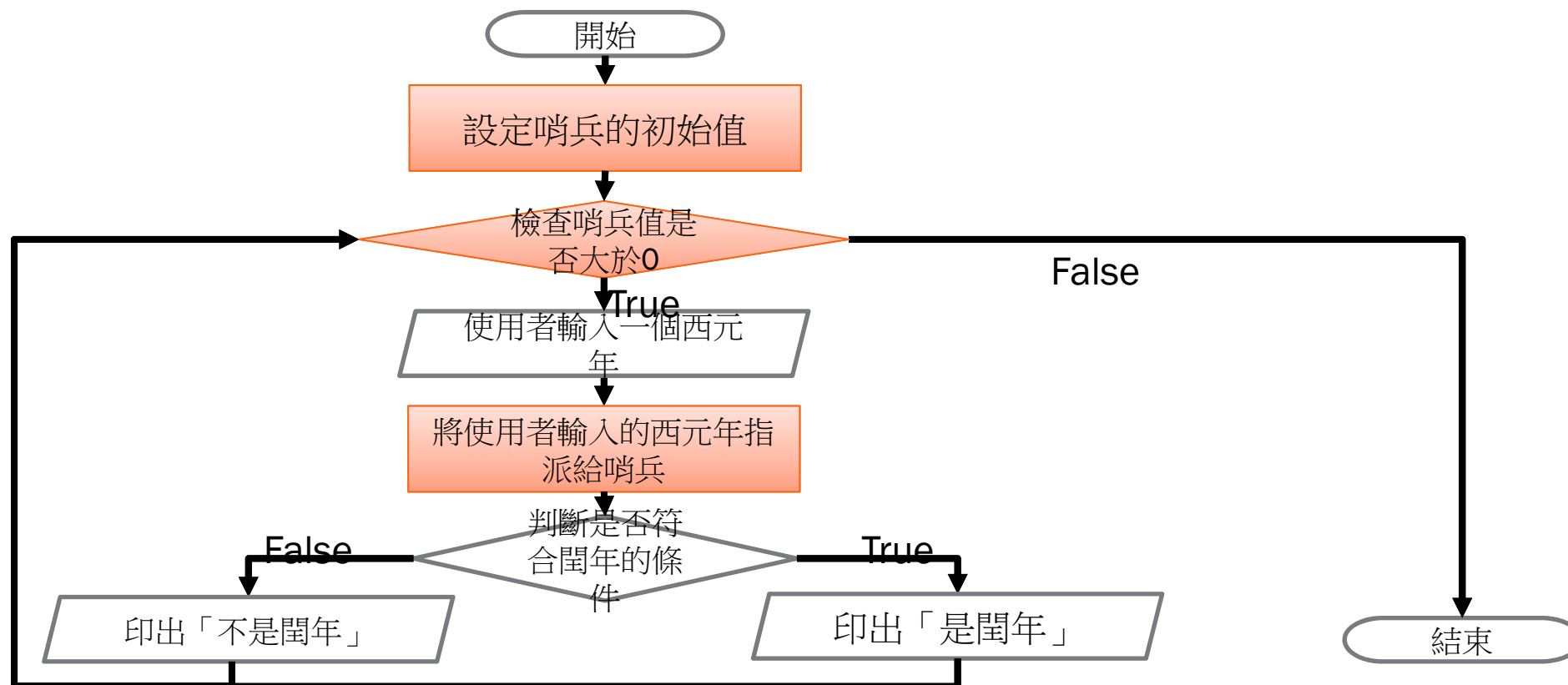
**while** 條件式：

當條件式**成立** (運算結果為布林值的True) 時，要**重複**做的事

哨兵是一個變數，  
它的值可被用來判  
斷是否終止重複結  
構

```
a=1 #哨兵的初始值
print("印出1~100的正整數:")
while a<=100: #當哨兵的值>100時，重複結構結束
    print(a)
    a+=1
```





```
year=1
while year > 0:
    years= input("請輸入一個西元年（輸入0代表結束程式）：")
    year=int(years)
    isLeapYear = (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)
    if year > 0:
        if isLeapYear:
            print ("西元",year,"年是閏年")
        else:
            print ("西元",year,"年不是閏年")
```

## 【實作練習】2

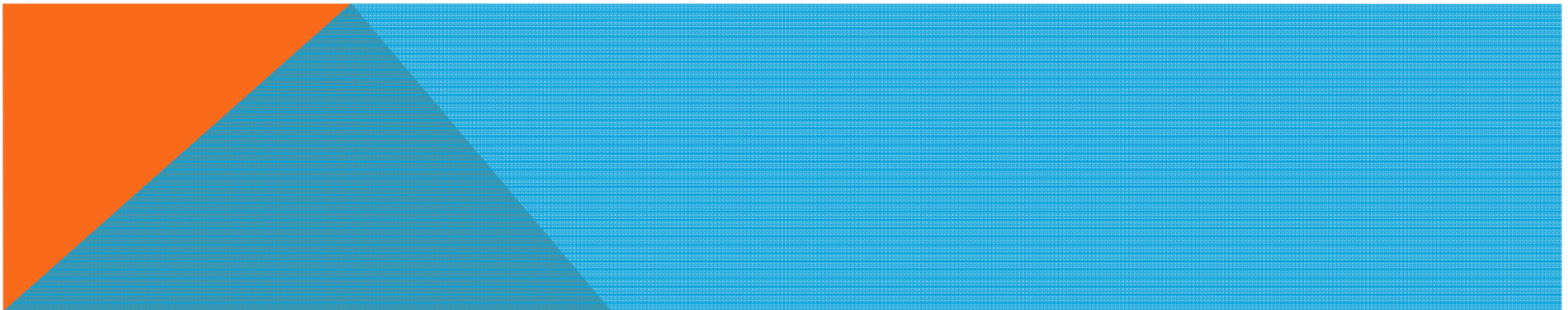
(1) 請修改【實作練習】1的流程圖：

- 讓【實作練習】1的功能可重複執行，直到使用者輸入的額溫小於或等於0，才結束程式。

將完成的流程圖存成 **week05\_ex2\_flowchart\_你的學號.pdf** 之後上傳到moodle的【實作練習】檔案上傳區

(2) 請依照 (1) 劃出的流程圖，完成程式碼。

將完成的程式碼存成 **week05\_ex2\_你的學號.py** 並壓縮後，上傳到moodle的【實作練習】檔案上傳區



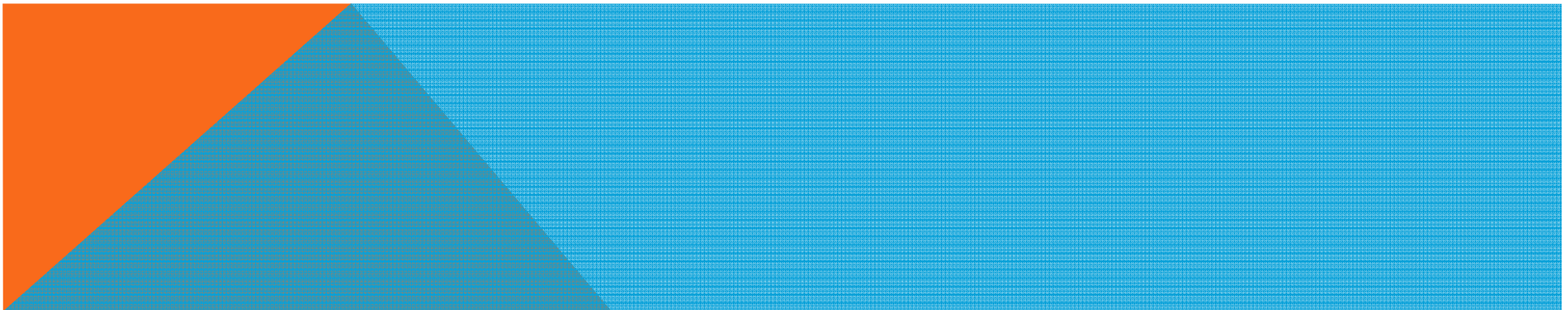
## 測試情況：

Case 1: 體溫過高的情況：輸入38

Case 2: 體溫正常的情况：輸入 36.5

Case 3: 體溫過低的情况：輸入 35

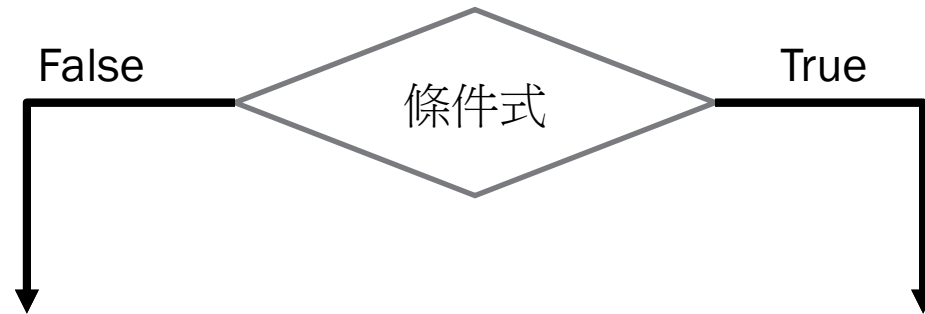
Case 4: 程式可重複執行，直到輸入0 結束程式



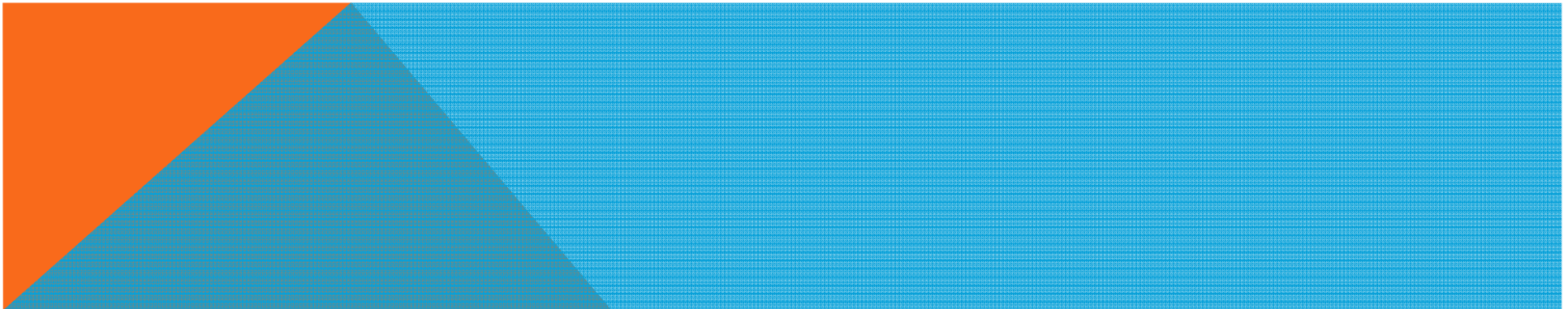


## 【流程圖注意事項】

- 一個流程圖一定要有一個開始和一個結束
- 一個決策判斷的符號只會有兩個分支(一邊為True; 一邊為False)



- 流程圖符號中的說明文字應儘量避免直接使用程式碼



小組作業一

# 先導任務

1. 複習並整理到目前為止課堂上已教過的內容。
2. 依目前上課已經教過的範圍，觀察並思考生活中有那些問題可以透過寫程式來幫忙解決，每人至少發想一個問題，並在下次上課前將想到的問題以文字敘述寫在moodle的【小組作業一】個別筆記整理&題目發想區。





10:55 臺北 – 15:05 臺東

2019/07/22 (一)

全票

普悠瑪 → 臺北 – 臺東

422

7 車 37 號

N90101896286119

台鐵的火車票上面印有車次和座位號碼，但是無法看出這個座位是靠窗或靠走道的，往往要等到搭上火車找到座位後才知道自己是座在靠窗的位子還是靠走道的位子。有人整理出，台鐵座位排列的規則如下表所示：

窗       戶	1	3	中 間 走 道	4	2	窗       戶
	5	7		8	6	
	9	11		12	10	
	13	15		16	14	
	17	19		20	18	
	21	23		24	22	
	25	27		28	26	
	29	31		32	30	
	33	35		36	34	
	37	39		40	38	
	41	43		44	42	
	45	47		48	46	
	49	51		52	50	



【生活中的問題】台鐵的火車票上面印有車次和座位號碼，但是無法看出這個座位是靠窗或靠走道的，往往要等到搭上火車找到座位後才知道自己是座在靠窗的位子還是靠走道的位子。

【思考】此問題是否適合透過寫程式來解決？

【輸入】、【運算處理的規則】、【輸出】

有人整理出台鐵火車座位的排列規則：當座位號碼除4，餘數是1或2時，代表這個座位是靠窗戶的座位，餘數是3或0時，代表這個座位是靠走道的座位。請運用Python來計算寫一支程式能讓使用者輸入座位號碼，然後回傳給使用者這個座位號碼是靠窗戶還是靠走道。

