

電腦的決策： 選擇結構

- 單向/雙向選擇結構
- 關係/邏輯運算子
- 多向/巢狀選擇結構

摘要：選擇結構

- Who(對象)：選擇結構
- Why(目的緣由)：為什麼需要選擇結構
 - 模擬現實生活中根據條件的判斷結果來進行選擇的情境
- When(應用時機)/Where(應用場域)：選擇結構的應用情境
 - 根據條件的判斷結果來選擇後續的處理流程
- What(內容定義)：選擇結構的類型
 - 1.單向/雙向 2.多向 3. 巢狀
- How(操作機制)：1. 選擇結構語法 2. 設定判斷條件式
 - 1. 選擇結構語法：(1) if / if...else (2) if...elif...else (3) if...if
 - 2. 判斷條件式的組成元素：邏輯運算子、關係運算子

摘要：單向/雙向選擇

本章介紹了負責掌控程式選擇流程的if判斷式，
if判斷式就如同在生活中我們需要做出許多抉擇，
不同的抉擇將會進行不同的運算，
if判斷式需要搭配不同的關係/邏輯運算子來完成選擇。

日常生活中，常常會遇到很多的選擇題...



若 吃!

則

否則



變胖了

可憐哪

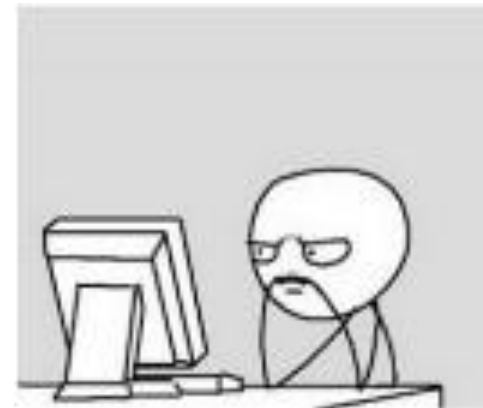
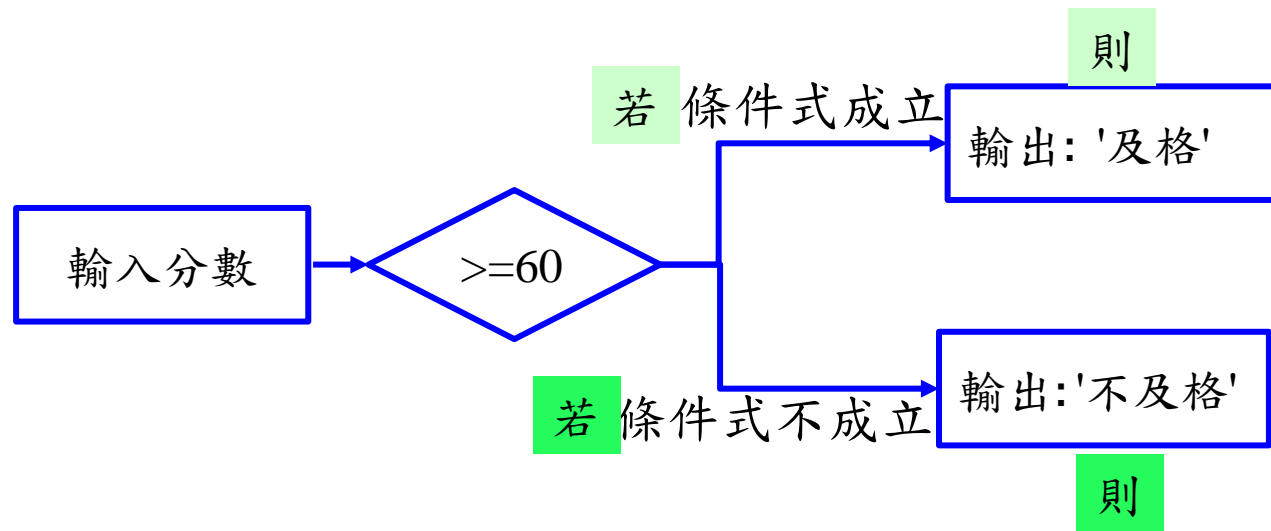


動動腦

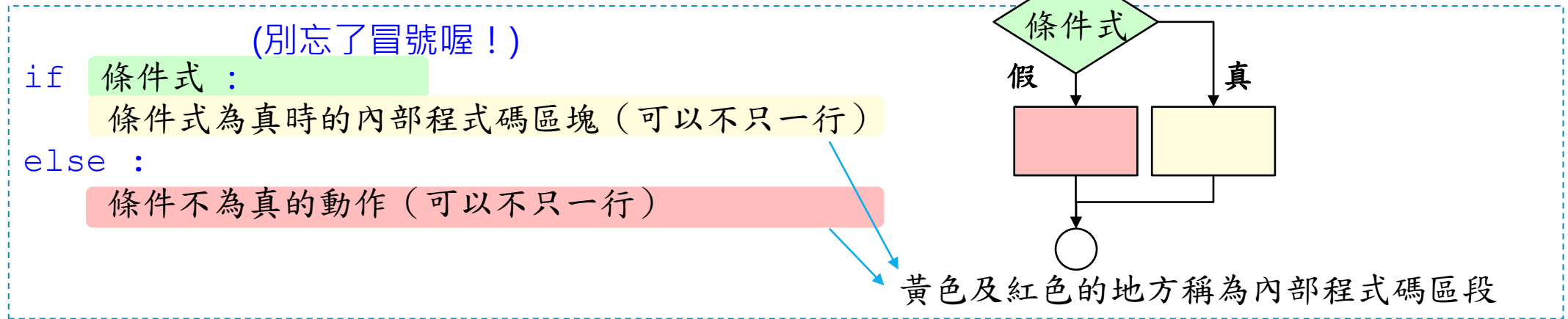
程式是否也能表達出這種『若...則...』的語句呢？

例如：

你今天小考分數多少？



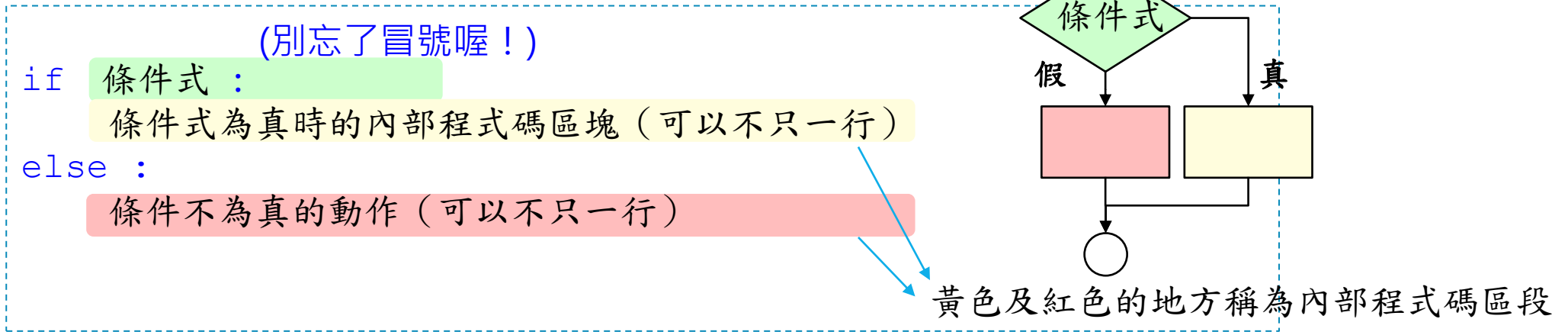
雙向選擇結構乃是根據條件式的判斷結果來選擇程式碼的執行區塊



條件式：設定條件來決定某一段內部程式碼區塊是否要執行。

- 條件式就是透過關係/邏輯運算式來設定判斷情況。
- 選擇結構的內部程式碼區塊是以條件式後面的冒號(:)開始。
- 同一層縮排的程式碼才視為是同一個區塊。
- Python是根據縮排來決定哪些程式碼是屬於同一個區塊。

雙向選擇結構



- 條件式的運算結果為真的時候，執行if條件式後面的程式區塊，接著便離開雙向選擇結構的程式碼區塊。
- 條件式的運算結果為假的時候，執行else後面的程式區塊，接著便離開雙向選擇結構的程式碼區塊。

雙向選擇結構的語法規則

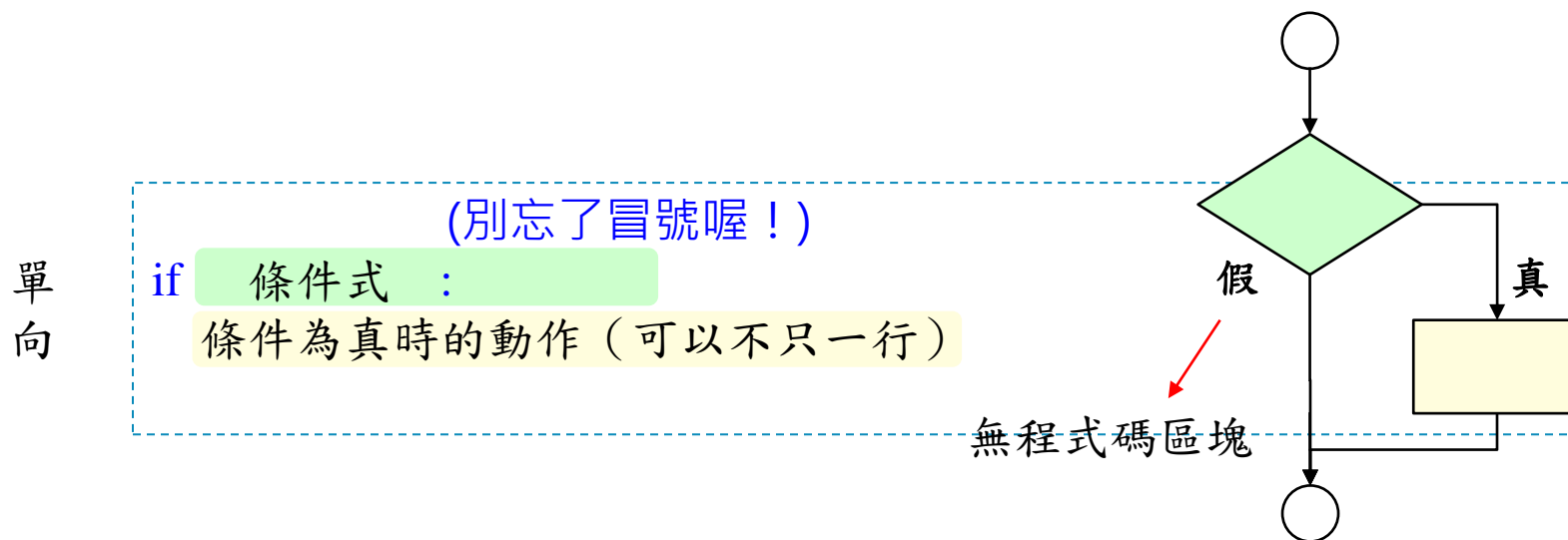
縮排

- 雙向選擇結構(if...else...)語法中的程式碼必須要縮排
- 縮排就是往內縮四個空格或是一個Tab
- 大部份開發工具都會自動排版了
 - 寫完 if ...: 之後按下 ENTER 鍵，會自動跳到下一行內縮四個空白處

句型排版格式規定

- Python 對於選擇結構有嚴格的排版格式規定，若不符合規定會出現 IndentationError: expected an indented block 的語法錯誤。

單向選擇結構



單向if條件判斷式：無else對應需要執行的程式碼區塊

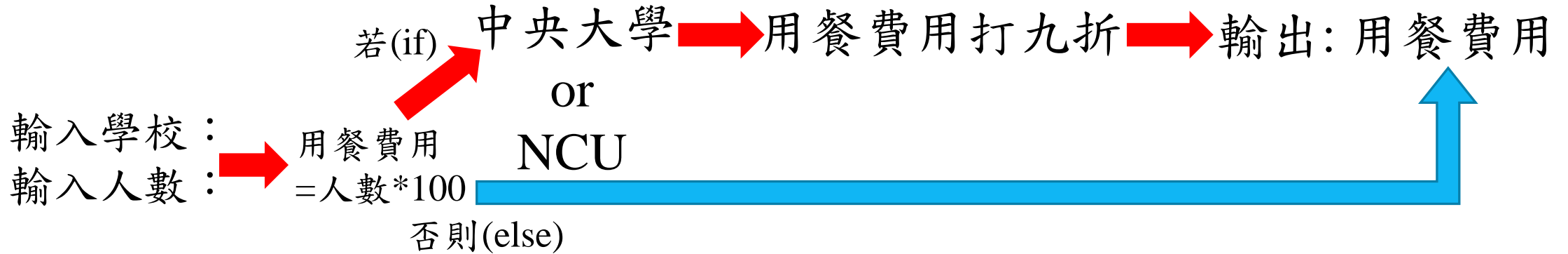
條件判斷式不成立時候，不用執行任何的程式碼

- 可以省略掉 else：以下的部份

單向選擇結構程式範例

條件判斷式：

`school == "NCU"`



程式範例：

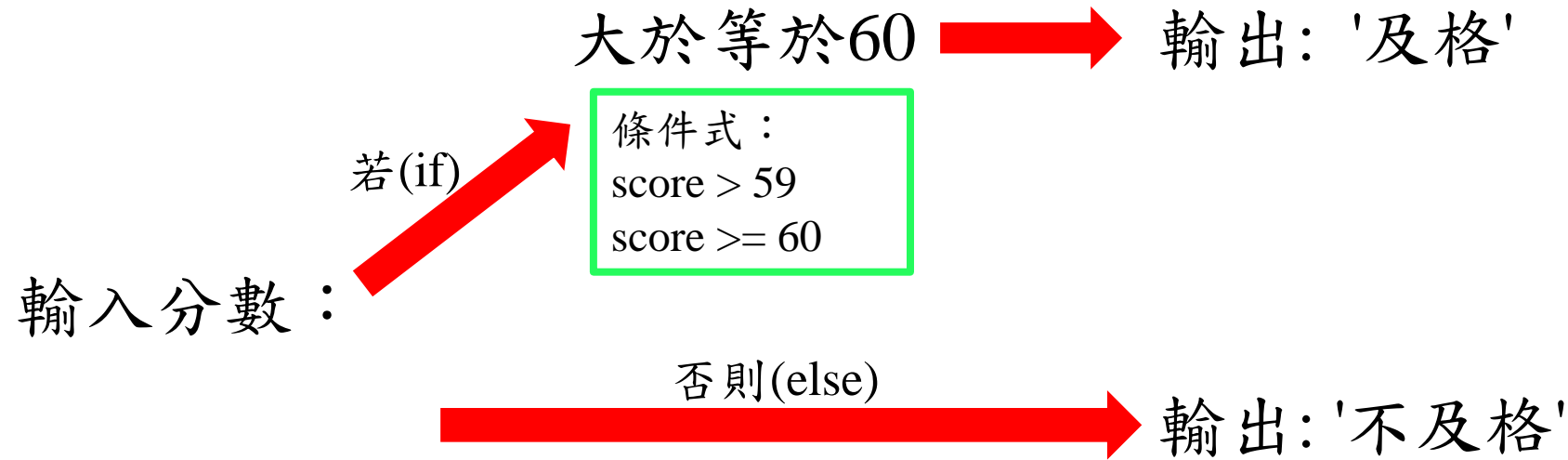
```
number = int(input("Please input number : "))
school = input("Please input school : ")
price = number * 100
if school == "NCU":
    price = price * 0.9
print("用餐費用 :", price)
```

選擇結構語法的規則

選擇結構(if...else...)程式語法規則

- 由 if 區塊和 else 區塊組成
- 一定要有一個 if 區塊，而且只有一個
- else 區塊可以沒有，也可以有一個，但是絕對不可以多個
 - 單向選擇結構：沒有else區塊
 - 雙向選擇結構：有else區塊

雙向選擇結構程式範例：你今天小考分數多少？



```
score = int(input("Please input score"))
```

```
if score > 59:  
    print('及格')  
else:  
    print('不及格')
```

```
if score >= 60:  
    print('及格')  
else:  
    print('不及格')
```

OJ 習題 Problem：單向/雙向選擇結構

C4_P2：預算管理

C4_P3：奇偶數判斷

OJ 習題 Problem：單向/雙向選擇結構

C4_P2：預算管理

1. 輸入：
 1. 當月的預算金額budge
 2. 當日的花費金額expense
2. 判斷是否超出預算
 1. 若未超出預算則輸出"未超出預算"
 2. 反之輸出"超出預算"。

OJ 習題 Problem：單向/雙向選擇結構

C4_P3：奇偶數判斷

1. 輸入：一個整數num
2. 判斷：該數字是奇數還是偶數
 1. 奇數：輸出”奇數”
 2. 偶數：輸出”偶數”

總結小憩：單向/雙向選擇結構



本章關鍵字
本章重點概念簡述



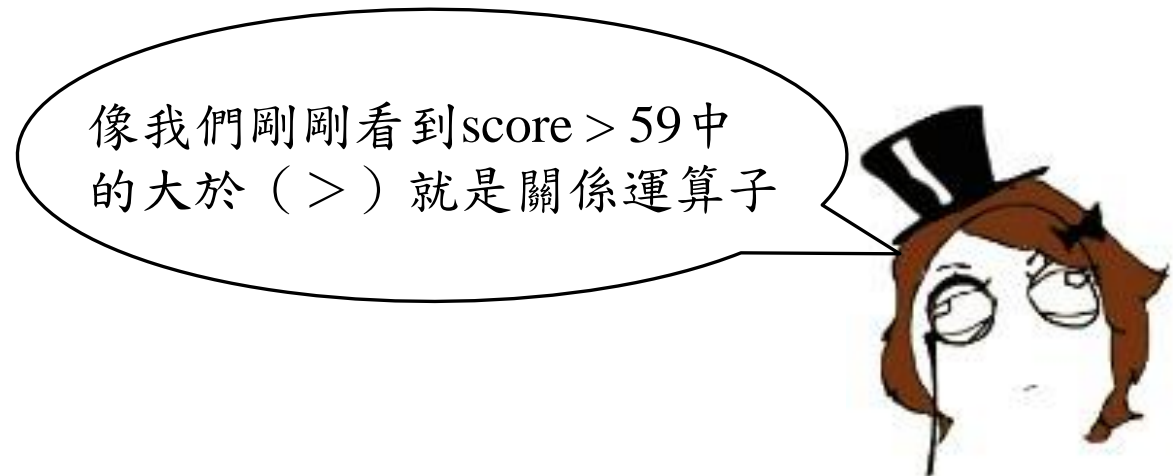
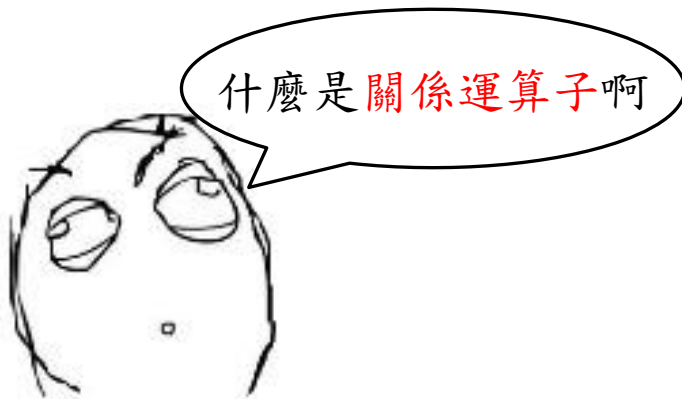
請將你的章節總結寫在最後一頁
(第47頁)的Memo上，Memo的title
請設定為總結

關係運算子

運算子的種類

在前面幾頁我們看到了，if後方的條件式裡會放一句條件式，條件式的『成立』或『不成立』，則會決定要執行的程式碼區塊。

因此，決定條件式要如何設定是相當重要的，並且需要搭配各種關係運算子來完成條件式。



動動腦

關係運算子有幾種

聰明的你能想到還有哪些關係運算子嗎？

那…關係運算子總共有哪
些啊!?



就讓黃老師帶你繼續看下去



關係運算子

關係運算子是二元運算子，其結果經常作為選擇結構語法的條件式。

- 二元運算子：運算時需要兩個運算元。
- 英文時間：運算子(operator)、運算元(operand)。

意義	關係運算子	結果為真 True	結果為假 False
相等	==	3 == 3	3 == 4
不相等	!=	3 != 4	4 != 4
大於	>	4 > 3	3 > 4
大於等於	>=	4 >= 4	3 >= 4
小於	<	3 < 4	4 < 3
小於等於	<=	3 <= 4	5 <= 4

關係運算子

各種運算子次序由高至低

- 次方
- 乘/除/餘
- 加/減
- 關係運算

Python 程式碼	運算次序	運算結果
<code>3*3 == 9</code>	<code>(3*3) == 9</code>	True
<code>2+3 > 1+4</code>	<code>(2+3) > (1+4)</code>	False

關係運算子

小試身手

```
x = 11
y = 20

print('A')
if 100 > 3 :
    print('B')
if 100 < 200 :
    print('C')
if x/2 > 5 :
    print('D')
print('E')
if x+y < 40 :
    print('F')
if x*2 == y :
    print('G')
if x <= y :
    print('H')
if x+3*y >= x*3+y :
    print('I')
if x!= y :
    print('J')
```



A
B
C
D
E
F
H
I
J

測驗來啦!!!



邏輯運算子

- 「邏輯運算子」的功能就是用來「結合多個條件式」
- 範例：當寒流來襲，而且濕度足夠的話，高山就會下雪。
 - 結果：高山會下雪
 - 條件一：「寒流來襲」 而且 條件二：「濕度足夠」 都必須成立
(and)
 - 而且 (and) 邏輯運算子：條件一與條件二都必須成立
- 邏輯運算子的優先順序是：not最優先執行、and其次、or最後

邏輯運算子: and

- 當a與b皆為True，a and b的結果才會是True
- a為False，無論b是True或是False，a and b的結果都是False。
- 非0的數值都視為True

```
a = 3.14
if a:
    print("這行會印出來")
```

- 若a為False，a and b運算並不會評估b的值，此稱為短路(short-circuit)。

```
a = -2
if False and a >= 0:
    print('負數')
```

a	b	a and b
False	False	False
False	True	False
True	False	False
True	True	True

```
if month == 8 and day == 15:
    print('中秋節快樂!')
else:
    print('平常日')
```

邏輯運算子: or

- a與b中只要有一個皆為True， a or b的結果就會是True
- 若a為True，無論b是True或是False，a or b的結果都是True。
- 若a為True，a or b運算並不會評估b的值，此稱為短路(short-circuit)。

```
a = 3  
if True or a>0:  
    print('正數' )
```

a	b	a or b
False	False	False
False	True	True
True	False	True
True	True	True

```
if score >= 60 or times_absence < 3:  
    print('及格了')  
else:  
    print('不及格')
```


邏輯運算子: not

- 用於更改布林值。
- 若a為True，not a為False。
- 若a為False，not a為True。

a	not a
False	True
True	False

```
if not (month == 8 and day == 15):  
    print('平常日')  
else:  
    print('中秋節快樂!')
```

邏輯運算子來簡化選擇結構

```
if month == 8:  
    if day == 15:  
        print('中秋節快樂!')  
    else:  
        print('平常日')  
else:  
    print('平常日')
```

使用邏輯運算子 and



```
if month == 8 and day == 15:  
    print('中秋節快樂')  
else:  
    print('平常日')
```

```
if age <= 12:  
    price = price * 0.9  
    normal = False  
if age >= 65:  
    price = price * 0.9  
    normal = False  
if normal:  
    price = price - 50
```

使用邏輯運算子 or



```
if age <= 12 or age >= 65:  
    price = price * 0.9  
else:  
    price = price - 50
```

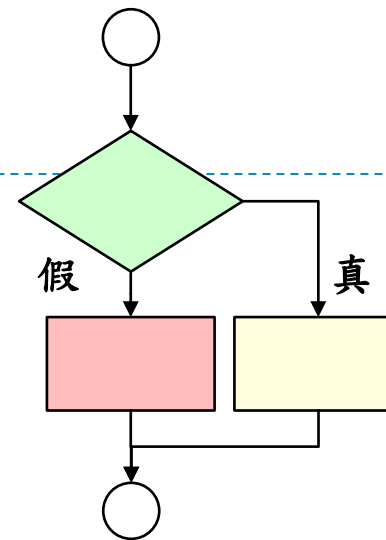
運算子總結

類型	說明	運算子
指派運算子 (賦值)	等號右邊先運算，再將運算結果指定給左邊的變數	=
算術運算子	數學四則運算的運算子 <ul style="list-style-type: none">除了常見的加減乘除（+、-、*、/、//）之外尚有取餘數運算子（%，modulo operator）以及次方運算子（**，power operator）	** *, /, //, % +, -
關係運算子	<ul style="list-style-type: none">透過比較運算來測定其等同與否以及相對大小等等比較兩個運算式，若比較結果成立就傳回True，若比較結果不成立則傳回False	>, >=, <, <=, ==, !=
邏輯運算子	結合多個關係運算式來綜合得到最終的比較結果，大多用於比較複雜的比較條件式。	not and or

比較雙向、單向if判斷式

雙向

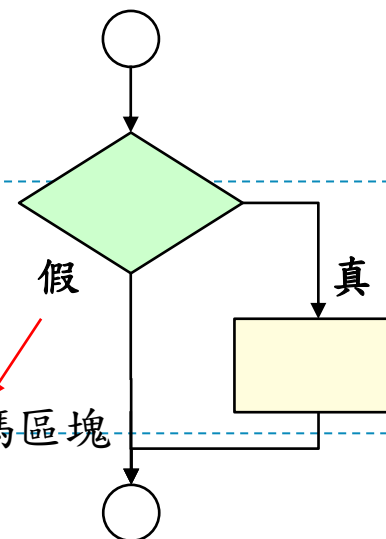
(別忘了冒號喔！)
if 條件式 :
條件為真時的動作 (可以不只一行)
else :
條件不為真的動作 (可以不只一行)



單向

(別忘了冒號喔！)
if 條件式 :
條件為真時的動作 (可以不只一行)

無內部程式碼區塊



OJ 習題 Problem：邏輯運算子的搭配應用

C4_P6：三角形

1. 輸入：輸入三個正整數 a 、 b 、 c
2. 三角形定義法則：
任意兩邊和要大於第三邊
3. 判斷與輸出：是否為三角形
 1. 是三角形：輸出"True"
 2. 不是三角形：輸出"False"

總結小憩：關係/邏輯運算子



本章關鍵字
本章重點概念簡述



請將你的章節總結寫在最後一頁
(第47頁) Memo上，Memo的title請
設定為總結

巢狀選擇結構

- if選擇結構裡頭還有另一組if選擇結構，就形成巢狀選擇架構。
- 我們通常稱這種多於一層的程式為巢狀 (nested) 控制結構。
- 巢狀選擇結構就好像是設定多道關卡，
也就是對多組資料進行各種的條件式判斷，
資料必須經過每一道關卡的檢驗。
- 注意：建立巢狀選擇結構時，要注意同一組的if、elif和else必須對齊。
Python是根據縮排來決定哪些程式碼是屬於同一個區塊。

巢狀選擇結構範例：適婚年齡的判斷

- 假設男生和女生的適婚年齡不同，若要判斷一個人的適婚年齡，必須先知道他/她的性別。

- 男生的適婚年齡：30歲
- 女生的適婚年齡：25歲

- 檢查性別：

- ① 性別判斷式
- ② 年齡判斷式

```
gender = input("請輸入男/女：")
age = int(input("請輸入年齡："))

if gender == '男': #男生
    if age >= 30:
        print('已達適婚年齡')
    else:
        print('還未達適婚年齡')
else: #女生
    if age >= 25:
        print('已達適婚年齡')
    else:
        print('還未達適婚年齡')
```

①

②

②

動動腦

如果...不只有兩種可能呢？


- 判斷個數是不是正值 (只有正值 or 非正值兩種可能)
- 如果我們想判斷正值、零和負值(共三種可能)，分別作不同的動作，該怎麼作呢？




動動腦

利用巢狀的作法來解決三種可能的問題

```
v = int(input('請輸入一個整數值>'))  
if v > 0:  
    print('正值')  
else:  
    if v == 0:  
        print('零')  
    else:  
        print('負值')
```



是要利用巢狀
if的作法嗎..?

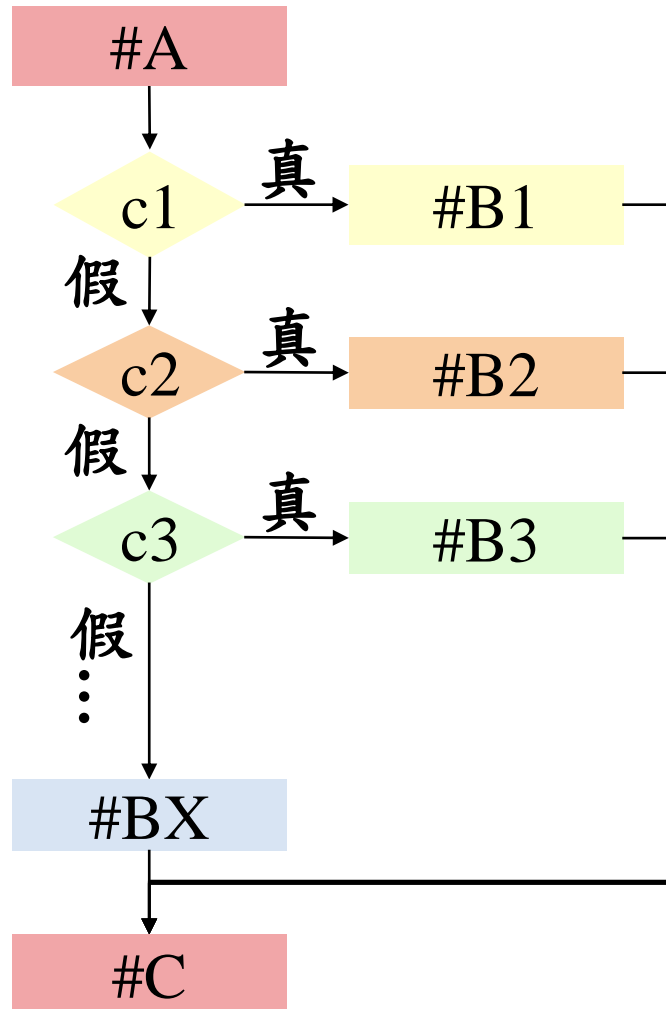


不錯的想法，但其
實還有更簡單的方法！

多向選擇結構：if...elif...else 句型

- 應用時機：資料需要進行多重判斷的時候
- 透過多個條件判斷式來設定各種狀況，以及要執行的程式區塊
- 在多種的可能情況下，僅執行某種情況的程式區塊
 - if...elif：設定多個條件判斷式的各種不同狀況，以及要執行的程式區塊
 - else：不符合所列之各種情況時，要執行的程式區塊
- 當多向選擇結構中的某個程式區塊被執行後，便會離開(結束)多向選擇結構

多向選擇結構：if...elif...else句型



條件情況：C1成立

- 執行#B1程式區塊，並接著執行#C程式區塊

條件情況：C1不成立，C2成立

- 執行#B2程式區塊，並接著執行#C程式區塊

條件情況：C1不成立，C2不成立，C3成立

- 執行#B3程式區塊，並接著執行#C程式區塊

條件情況：所有條件式都不成立

- 執行#BX程式區塊，並接著執行#C程式區塊

多向選擇結構V.S. 巢狀結構

判斷變數的正值、零和負值，並在不同的情況下分別作不同的動作

巢狀：

```
v = int(input('請輸入一個整數值>'))  
  
if v > 0:  
    print('正值')  
else:  
    if v == 0:  
        print('零')  
    else:  
        print('負值')
```

多向：

```
v = int(input('請輸入一個整數值>'))  
  
if v > 0:  
    print('正值')  
elif v == 0:  
    print('零')  
else:  
    print('負值')
```

範例：多向選擇結構

程式範例：

```
grade = int(input('你的年級 >')) #A

if grade == 1: #B
    print('一年級')
elif grade == 2: #C
    print('二年級')
elif grade == 3: #D
    print('三年級')
elif grade == 4: #E
    print('四年級')
else: #F
    print('高年級')
```

雖然多向選擇結構不一定要有 else 的部份，但建議保留此部份作為檢查時的最後一道關卡。

多向選擇結構 V.S. 單向選擇結構

- 延續上頁的範例，用單向選擇結構也可以做到一樣的功能，但兩者之間有什麼差別呢?(以使用者輸入的數值為4為例)

- 單向選擇結構：執行所有的if判斷。
- 多向選擇結構：依序判斷到第三個elif後，執行grade==4的程式區塊，再進行grade>4的判斷。

➤ 減少判斷次數。

```
grade = int(input("你的年級 > "))
if (grade==1):
    print ("一年級")
elif (grade==2):
    print ("二年級")
elif (grade==3):
    print ("三年級")
elif (grade==4):
    print ("四年級")
elif (grade>4):
    print ("高年級")
```

多向

你的年級 > 4
四年級

```
grade = int(input("你的年級 > "))
if (grade==1):
    print ("一年級")
if (grade==2):
    print ("二年級")
if (grade==3):
    print ("三年級")
if (grade==4):
    print ("四年級")
if (grade>4):
    print ("高年級")
```

單向

你的年級 > 4
四年級

範例：多向選擇結構

```
price = 10
number = int(input())
if number >= 12:
    price = price*number*0.9
elif number >= 6:
    price = price*number*0.95
elif number >= 2:
    price = price*number*0.99
else:
    price = price*number

print(number, price)
```

- 12 人以上：number>=12 # condition C1
- 6-11 人：number>=6 # condition C2
- 2-5 人：number>=2 # condition C3
- 1 人

要特別注意「由上而下」的判斷順序

OJ 習題 Problem：巢狀/多向判斷式

- C4_P1：夢想成真的表現評價
- C4_P4：BMI計算
- C4_P5：年齡分類
- C4_P7：電影票收費系統

OJ 習題 Problem : C4_P1 夢想成真的表現評價

1. 輸入：輸入小美在比賽中獲得的分數
2. 判斷與輸出：評分屬於哪種等級
 - 90以上: 優等
 - 80 - 89: 甲等
 - 70 - 79: 乙等
 - 60 - 69: 丙等
 - 60以下: 不及格

OJ 習題 Problem : C4_P1 夢想成真的表現評價

1. 輸入：
 1. 輸入身高（單位：公尺）
 2. 體重（單位：公斤）
2. 計算：BMI 公式（ $BMI = \text{體重} / \text{身高}^2$ ）
求出 BMI 值
3. 判斷與輸出：體重屬於哪種等級
 - $BMI < 18.5$: 過輕
 - $18.5 \leq BMI < 24$: 正常
 - $24 \leq BMI < 27$: 過重
 - $BMI \geq 27$: 肥胖

OJ 習題 Problem : C4_P5 年齡分類

1. 輸入：年齡
2. 判斷與輸出：年齡屬於哪種類別
 - 0 - 12 歲：兒童
 - 13 - 18 歲：青少年
 - 19 - 65 歲：成年人
 - 66 歲以上：老年人

OJ 習題 Problem： C4_P7 電影票收費系統

1. 輸入：
 1. 使用者年齡
 2. 是否為學生
2. 判斷與輸出：電影票費用
3.
 - 4歲(含)以下：免費入場
 - 5歲(含)以上的學生：60元
 - 5歲~17歲但不是學生：120元
 - 18歲(含)以上但不是學生：180元

OJ 習題 Exercise

C4_E1 平年閏年判斷

C4_E2 直角三角形

C4_E3 月底突如其來的加班計算

C4_E4 員工等級

總結



本章關鍵字
本章重點概念簡述



請將你的章節總結寫在Memo上，
Memo的title請設定為**總結**