

02：運算式、判斷式

2018.1.30

主題

- 運算式
 - 算術運算子
 - 關係運算子
 - 邏輯運算子
 - 複合指定運算子
- 判斷式
 - if ...



● 算術運算子

- 用於執行一般數學運算的運算子

運算子	意義	範例	範例結果
+	兩運算元相加	12+3	15
-	兩運算元相減	12-3	9
*	兩運算元相乘	12*3	36
/	兩運算元相除	32/5	6.4
%	取得餘數	32%5	2
//	取得整除的商數	32//5	6
**	(運算元 1) 的 (運算元 2) 次方	7**2	7 ² = 49

「+」運算子的功能

- 「+」可用於數值運算，也可用於字串組合，使用時須特別留意運算元的資料型態。

```
3 print(23 + 45)  
4 print("23" + "45")
```

5

關係運算子

- 關係運算子會比較兩個運算式，
 - 若比較結果正確，就傳回 True
 - 若比較結果錯誤，就傳回 False
- e.g., $3 > 2$ → 傳回 True

6

關係運算子

運算子	意義
==	運算式 1 是否等於運算式 2
!=	運算式 1 是否不等於運算式 2
>	運算式 1 是否大於運算式 2
<	運算式 1 是否小於運算式 2
>=	運算式 1 是否大於或等於運算式 2
<=	運算式 1 是否小於或等於運算式 2

7

邏輯運算子

- 能結合多個比較運算式

運算子	意義	範例	範例結果
not	傳回與原來比較結果相反的值，即比較結果是 True，就傳回 False；比較結果是 False，就傳回 True。	not(3>5) not(5>3)	True False
and	只有兩個運算元的比較結果都是 True 時，才傳回 True，其餘情況皆傳回 False。	(5>3) and (9>6) (5>3) and (9<6) (5<3) and (9>6) (5<3) and (9<6)	True False False False
or	只有兩個運算元的比較結果都是 False 時，才傳回 False，其餘情況皆傳回 True。	(5>3) or (9>6) (5>3) or (9<6) (5<3) or (9>6) (5<3) or (9<6)	True True True False

8

複合指定運算子

- 複合指定運算子同時做「執行運算」及「指定」

• $x += 3 \leftrightarrow x = x + 3$

複合指定運算子

運算子	意義	範例	範例結果
+=	相加後再指定給原變數	i += 5	15
-=	相減後再指定給原變數	i -= 5	5
*=	相乘後再指定給原變數	i *= 5	50
/=	相除後再指定給原變數	i /= 5	2
%=	相除得到餘數後再指定給原變數	i %= 5	0
//=	相除得到整除商數後再指定給原變數	i //= 5	2
**=	做指數運算後再指定給原變數	i **= 3	1000

• Pythonic

- Python 追求簡單、優雅的風格，Pythonic 的意思是「具有 Python 風格」的程式碼，也就是很 Python。
- 例如：如何交換兩個變數 `a, b` 的內容？
- Python 寫法

11

• Pythonic

- 另一個例子：
 - 假設 `a = 3, b = 1`
 - 請寫程式輸出下列判別式的真(True)、假(False)值
 - `(1<=b) and (b<=a) and (a<10)`
- Python 寫法

12

- 判斷式

13

[illegible]

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Import the module
from sklearn.cluster import KMeans

# Create a dataset
X = np.random.rand(100, 2)

# Create a KMeans object
kmeans = KMeans(n_clusters=3)

# Fit the data
kmeans.fit(X)

# Predict the cluster for each data point
y_kmeans = kmeans.predict(X)

# Print the cluster labels for each data point
print(y_kmeans)

```

中間部分
重複執行

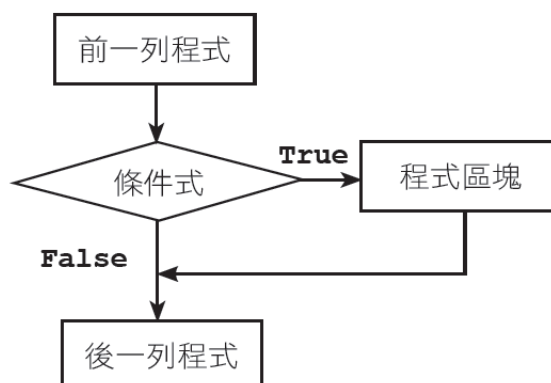
● 流程控制

- Python 流程控制命令分為兩大類：
 - 判斷式：根據關係或邏輯運算的條件式來判斷程式執行的流程，若條件式結果為 **True**，就執行跳躍。
 - if ... elif ... else
 - 迴圈：根據關係或邏輯運算條件式的結果為 **True** 或 **False** 來判斷，以決定是否重複執行指定的程式。
 - for
 - while

15

● 單向判斷式(if ...)

- 「if...」為單向判斷式，是 if 指令中最簡單的型態，語法為：
 - if (條件式):
 - if 條件式:



16

範例

```
3 score = 58
4
5 if score < 60:
6     print("Fail")
```

17

雙向判斷式(if ... else ...)

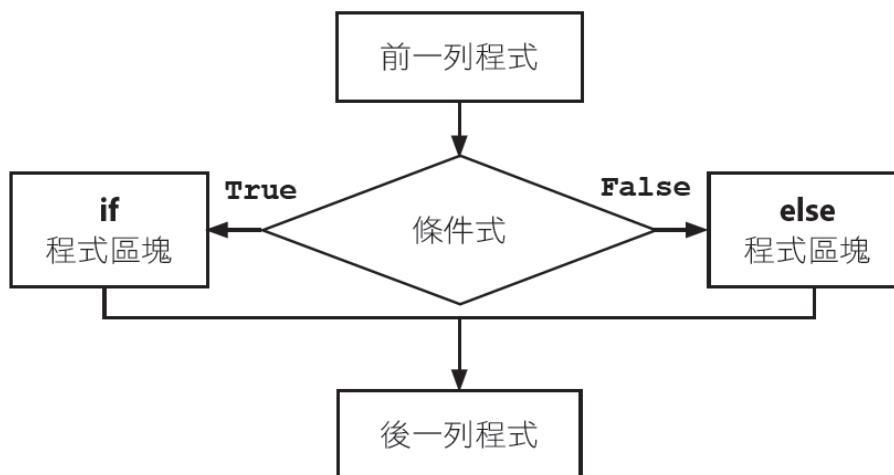
```
3 score = 58
4
5 if score < 60:
6     print("Fail")
```

條件成立
時就執行

那不成立
呢？

18

雙向判斷式(if ... else ...)



19

雙向判斷式(if ... else ...)

```
2 score = int(input("請輸入成績："))
3
4 if 0 <= score < 60:
5     print("Fail")
6 else:
7     print("PASS")
```

注意冒號

注意縮排

20

範例

- 輸入：考試成績 (滿分100)
- 輸出：若成績在
 - 90 分以上顯示「優等」，
 - 80 - 89 分顯示「甲等」，
 - 70 - 79 分顯示「乙等」，
 - 60 - 69 分顯示「丙等」，
 - 60 分以下顯示「丁等」。

```
2 score = int(input("請輸入成績："))
3
4 if 90 < score:
5     print("優等")
6 elif 80 <= score < 90:
7     print("甲等")
8 elif 70 <= score < 80:
9     print("乙等")
10 elif 60 <= score < 70:
11     print("丙等")
12 elif score < 60:
13     print("丁等")
```

這程式
有小錯

21

巢狀(nested)

- 在判斷式 (if ... elif ... else ...) 內可以有判斷式，稱為巢狀判斷式
 - Python 並未限制巢狀判斷式的層數
 - 層數太多會降低可讀性，而且維護困難

22

● PY3-0002

二、算術運算

2. 設計說明：

- (1) 請使用選擇敘述撰寫一程式，讓使用者輸入兩個整數 a 、 b ，然後再輸入一算術運算子（ $+$ 、 $-$ 、 $*$ 、 $/$ 、 $//$ 、 $\%$ ），輸出這兩個數以及其經過運算後的結果。

TO BE CONTINUED...