

書中所有 (MAT 模擬試題) 需更正為 (MTA 模擬試題)

## P1\_2

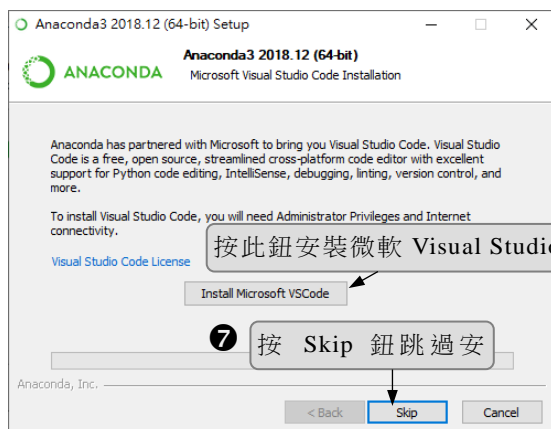
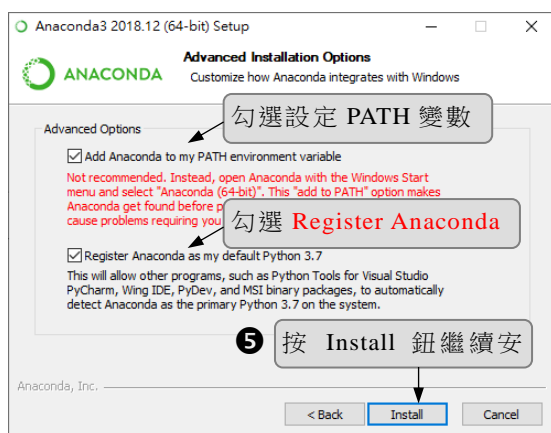
1. 中央處理單元負責電腦的運算，其中包含控制單元(Control Unit，CU)和算術邏輯單元(Arithmetic and Logic Unit，ALU)兩大部分。控制單元會負責程式指令的取得和解釋，並指揮電腦的各部分協調運作。ALU 是算術與邏輯指令的實際執行單元，其中有各種暫存器(Register)可暫存運算中的各類資料，以減少 CPU 到主記憶體存取的頻率。CPU 中的各種單元，是利用匯流排(Bus)方式連接以利資料流通。

## P1\_6

### 一. 直譯

直譯器(Interpreter，或稱解譯器)的功能是將程式語言所編寫的程式碼，依照程式的邏輯順序，將指令逐行轉為機器語言指令，並且立即執行。程式語言中 Python、

## P1\_17



## P2\_1

電腦軟體就是用來處理各類的資料，以解決人類生活上的問題。生活中有各式各樣的資料，例如姓名、身高、年齡、數量、車牌號碼、編號...等。這些資料的內容有些是屬於文字，例如姓名、車牌號碼...等。有些是屬於數值內容，例如身高、年齡...等。在 Python 語言中定義了一些基本資料型別，來處理各類的資料。在程式執行中為了方便快捷存取資料，資料會存在記憶體中。所以在使用相關資料時，如果能根據資料的類別和可能的大小，配置合適的記憶體空間，如此才能順利運作又不浪費記憶體。例如學生人數是沒有小數的數值資料，平均分數是有小數的數值資料，政府總預算是有效位數較多的數值資料、身份證字號則是長度為 10 的字串資料。

## P2\_12

運算子	功能說明	範例	結果(假設 y=5)
+	加法	$x = y + 2$	x 變數值為 7
-	減法	$x = y - 2$	x 變數值為 3
*	乘法	$x = y * 2$	x 變數值為 10
/	浮點數除法	$x = y / 2$	x 變數值為 2.5
//	整數除法	$x = y // 2$	x 變數值為 2
%	取除法的餘數	$x = y \% 2$	x 變數值為 1
**	次方(指數)	$x = y ** 2$	x 變數值為 25(5 <sup>2</sup> )

## P2\_13

使用複合指定運算子時，變數必須先宣告否則會產生錯誤。常用的複合運算子：

運算子	功能	範例	結果(x 變數值原為 3)
+=	相加後再指定	$x += 2 (x = x + 2)$	x 變數值為 5
-=	相減後再指定	$x -= 2 (x = x - 2)$	x 變數值為 1
*=	相乘後再指定	$x *= 2 (x = x * 2)$	x 變數值為 6
/=	浮點數相除後再指定	$x /= 2 (x = x / 2)$	x 變數值為 1.5
//=	整數相除後再指定	$x //= 2 (x = x // 2)$	x 變數值為 1
%=	相除取餘數後再指定	$x \% = 2 (x = x \% 2)$	x 變數值為 1
**=	次方運算後再指定	$x ** = 2 (x = x ** 2)$	x 變數值為 9(3 <sup>2</sup> = 9)

P2\_14

[例] 練習各種關係運算子的運算： (檔名：relational.py)

```
01 a,b=2,3
02 print('a = 2, b = 3')
03 print('a < b = ',a < b)      ⇨ True
04 print('a >= b = ',a >= b)   ⇨ False
05 print('a == b = ', a == b)  ⇨ False
```

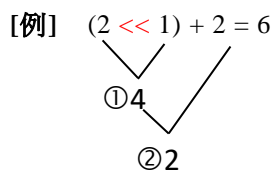
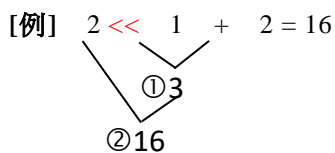
P2\_15

& (And)、| (Or)、^ (Xor) 及 ~ (Not) 為位元運算子(Bitwise Operator)，做法是先將運算元轉換成二進位，接著再做指定的二進位布林運算。位元運算子的運算方式如下表所示：

A	B	A & B	A   B	A ^ B	~A
1	1	1	1	0	0
1	0	0	1	1	0
0	1	0	1	1	1
0	0	0	0	0	1
說明		&(And，且)運算時，兩者皆為1才會是1。	(Or，或) 運算時，要一個為1就會是1。	^ (Xor，互斥) 運算時，只要兩者不同就會是1。	~ (Not，非) 運算時，1變成0；0變成1。

P2-18

例如運算式  $1 + 2 * 3$ ，因為\*運算子的優先順序高於+，所以會先計算  $2*3$  結果為6，然後  $1 + 6$  所以結果為7。運算式使用()左右括號，不但可以減少運算錯誤增加可讀性，而且可以改變運算的順序。例如上面運算式改為  $(1 + 2) * 3$  時，因為  $1 + 2$  用()括住所以要先計算結果為3，然後  $3 * 3$  結果為9。



## P2\_23

[例] 設定寬度來格式化輸出各種資料：(Δ：代表空格) (檔名：print3.py)

```
01 print('%d'%(12345)) # 顯示整數資料          ⇨12345
02 print('%8d'%(12345)) # 設寬度為 8,寬度有剩時補空格 ⇨Δ Δ Δ12345
03 print('%-8d'%(-12345)) # 靠左對齊,寬度有剩時補空格 ⇨-12345Δ Δ
04 print('%08d'%(12345)) # 設寬度為 8,寬度有剩時補 0  ⇨00012345
05 print('%3d'%(-12345)) # 設寬度為 3,寬度不足時全部顯示⇨-12345
06 print('%c'%('A')) # 顯示字元「A」        ⇨ A
07 print('%4c'%('A')) # 寬度為 4 有剩補空格  ⇨Δ Δ ΔA
08 print('%c'%(65)) # 65 的 ASCII 碼為 A    ⇨ A
09 print('%s'%('ABCDE')) # 顯示字串資料     ⇨ ABCDE
10 print('%8s'%('ABCDE')) # 設寬度為 8,寬度有剩時補空格 ⇨Δ Δ ΔABCDE
11 print('%3s'%('ABCDE')) # 設寬度為 3,寬度不足時全部顯示⇨ ABCDE
12 print('%6.2s'%('ABCDE'))# 設寬度為 6 並只顯示 2 字元 ⇨Δ Δ Δ ΔAB
```

## P2\_30

### 一. 選擇題

2. 下列哪個變數宣告方式錯誤？① a      ② a=5      ③ a = b = 5      ④ a, b=5, 1.4

## P3\_7

[例] 門票 300 元，若年齡低於 10 歲或 65 歲以上半價。

```
if (age < 10 or age >= 65) :
    price = 150
else :
    price = 300
```

一. 選擇題

1. Python 比較(關係)運算子所傳回的結果為哪種型別的資料？
2. Python 比較(關係)運算式的結果為真，則會傳回什麼資料？
3. Python 比較(關係)運算式的結果為假，則會傳回什麼資料？
5. 下列條件式的運算式結果，何者為 **False** ？
  - ①  $15 > 13$     ② `'c' < 'k'`    ③ `'A' == 65 + 0`    ④  $(90 / 3 > 100) \text{ or } (30 * 3 != 120)$

P4\_3

3. 多行迴圈程式碼的縮排必須要對齊，如果沒有對齊程式將無法正確執行。縮排時請按 **Tab** 鍵，會縮排固定的距離。如果要取消縮排時，則按 **Shift + Tab** 鍵。

```

02 | for x in range(-3, 3):
03 |     |sum += x
04 | print(sum) # 無縮排對齊，所以印出 -3、-5、-6、-6、-5、-3
    
```



3. 多行迴圈程式碼的縮排必須要對齊，如果沒有對齊程式將無法正確執行，如：

```

02 | for x in range(-3, 3):
03 |     |sum += x
04 | print(sum) # 無縮排對齊，所以無法執行
    
```

縮排時請按 **Tab** 鍵，會縮排固定的距離。若要取消縮排時，則按 **Shift + Tab** 鍵。

p4-11

3. 第 7~9 行：**while** 迴圈負責印出 \* 字元，變數 **k** 的初值為  $6 - 2 * i$ ，**i** 值由 0~2，所以 **k** 值依序為 6、4、2。因為第一行 \* 字元字數為 6、第二行為 4、第三行為 2，所以條件式應該設為 **k < 0**。
4. 答案是 (C)，程式碼請參考 test04\_2.py。

倒數第二行紅圈內文字「**k<0**」更正為「**k>0**」

p4-14

in range(len(p\_list)):」, 所以①的答案是 (3)。串列的詳細說明, 請參考第 6 章。

2. 因為售價高於 1 萬元的家電不需要調整售價, 所以要跳回迴圈開頭, 忽略第 5 行程式。第 4 行程式碼應該為 `continue`, 所以②的答案是 (1)。程式碼請參考 `test04_5.py`。

紅圈 1 內文字「第 6 章」更正為「第 5 章」

P5\_21

1. 第 1~2 行: `a1=[[1,2,3], [4,5,6], [6,7,8], [10,11,12]]` 為旋轉前的 4\*3 矩陣內容, 而 `a2` 串列用來存放 `a1` 矩陣順時針旋轉 90 度轉換後的 3\*4 矩陣內容。`a2` 串列的所有元素內容先預設為 0。
2. 第 10~12 行: `a1` 矩陣順時針旋轉 90 度的轉換運算成 `a2` 矩陣。觀察 `[0][0]⇒[0][3]`、`[0][1]⇒[1][3]`、`[0][2]⇒[2][3]`、`[1][0]⇒[0][2]` ... , 會發現轉換後第一維註標值等於原第二維註標值, 而第二維註標值會由 0~3。

a1 矩陣(4\*3)

[0][0]	1	[0][1]	2	[0][2]	3
[1][0]	4	[1][1]	5	[1][2]	6
[2][0]	6	[2][1]	7	[2][2]	8
[3][0]	10	[3][1]	11	[3][2]	12



a2 矩陣(3\*4)

[0][0]	10	[0][1]	6	[0][2]	4	[0][3]	1
[1][0]	11	[1][1]	7	[1][2]	5	[1][3]	2
[2][0]	12	[2][1]	8	[2][2]	6	[2][3]	3

p7-5

5. 第 17、18 行: 若輸入值為奇數時, 呼叫 `g(n)` 的遞迴函式後執行第 18 行。
6. 第 19、20 行: 若輸入值為偶數時, 呼叫 `g(n)` 的遞迴函式後執行第 20 行。

P7-6



簡例 (檔名: `gcd.py`)

使用輾轉相除法的遞迴運算設計 `GCD(p, q)` 函式, 求出兩整數 60、96 的最大公因數。

刪除「除」字

P7-13

05      `return ListSum(arr, n-1) + arr[n]`

P8-13

rstrip	<p>語法：<code>string.rstrip(sub) → str</code></p> <p>功能：刪除字串開頭的 <code>sub</code> 字串，如果省略 <code>sub</code> 參數會刪除字串開頭元、Tab 字元及 <code>None</code> 元。</p> <p>簡例：<code>print(s2.rstrip())</code></p>
--------	---

刪除  
→str

P8-19

```
06 score = float(input('輸入你投擲距離：(輸入-1,結束程式)'))
07 if (score == -1):
08     break
```

更正為  
-1

P9-10

- 2. 第 9、14 行：新增元素到 `set1`，第 9 行參數是字串，而 14 行是串列，其產生集合也有所差異。
- 3. 第 11、13、16、17 行：皆是由 `set1` 刪除元素，第 13、16 行刪除的元素已不存在，但是使用 `remove` 會發生錯誤訊息；第 17 行會隨機刪除一個元素。

更正為  
不存在

P9-15

 題目 (三)

(APCS 檢測試題)

下列是輸入整數來查詢對應英文單字的程式碼，但執行結果並不正確，請回答

更正為  
MTA

10-5

刪除已存在的空資料夾

#### 四. rmdir() 函式

os 套件的 rmdir() 函式是用來刪除已存在的資料夾 (或稱目錄)，語法如下：

```
os.rmdir(路徑名稱)
```

1. 使用本函式須和 os.path.exists() 函式搭配，要先確定要刪除的資料夾是否存在，才能進一步進行刪除動作。
2. 若要刪除的資料夾不存在，而直接使用 rmdir() 函式時會出現錯誤訊息。

不存在，或是其中有檔案時，而直接

10-11

37 個字元？

簡例 (檔名: read01.py)

讀取前例寫入的資料檔案 stu.txt 的內容。分兩次讀取，第一次先讀取前面 3 個字元；第二次再讀取後面剩餘內容。

11-4

字體將換成

若要解決中文顯示的問題，只要在程式碼開頭撰寫如下敘述，將圖表預設使用的字體將換成標楷體(字型的英文名 DFKai-SB)即可：

```
--font = {'family': 'DFKai-SB'} -- #建立 font 字型物件同時指定為標楷體
```

11-5

方法讓開發人員

#### 11.2.3 如何設定圖表標題、座標標題與座標範圍

圖表另外還提供下表的方法開發人員可以設定圖表標題、座標標題與座標範圍等，讓瀏覽者可能瞭解圖表所代表的意義。



11-11

小數 1 位數

autopct↵	格式語法為「%整數位數.小數位數 f%」。例如：%3.1f%表示整數 3 位數、小 1 位數。↵
----------	--

11-15

基峰科大

## 二. 程式設計↵

1. 繪製如下圖基峰科大 2015~2020 年招生分數的圖表。有商管、設計、資電三群，歷年錄取分數如下：↵

11-15

基峰科大

2. 繪製基峰科大的商管、設計、資電三群招生員額圓餅圖，三群的分配比例為：



12-1

win01.py-> window01.py

立一個視窗應用程式，並指定該視窗標題為「第一個視窗應用程式」，同時指定視窗大小寬 400px，高 200px。(檔名：win01.py)↵

12-2

MessageBox -> **messagebox**

MessageBox↵ 訊息方塊↵	用來顯示應用程式的相關提示訊息。↵
----------------------	-------------------

13-5

6

5. → 第 8~9 行：取得基峰資訊網頁程式並指定給 `htmlContent`，接著使用 `decode()` 方法進行字串解碼。

5. → 若取得網頁程式採如下寫法沒有執行 `decode()` 進行解碼：

13-10

## 13.5 網頁爬蟲應用 - 基峰資訊新書快報

13-12

12、13

```
11 link=data[0].find_all('a')↵  
12 for n in range(0, len(link)):↵  
13     print(link[n].text)↵
```

# MAT Python

## 國際認證模擬試題

### A.2 資料類型和運算子

19. \_\_\_\_\_ ( $2 * (1 + 2) ** 2 - 2 ** 2 * 2$ ) 運算式的執行結果為何？

- ① 4    ② 20    ③ 10    ④ 28

### A.3 流程控制與迴圈

1. 下列為換算成績等級的程式碼，換算的規則如下：

- 90(含)~100 分為「優」。
- 80(含)~89 分為「甲」。
- 70(含)~79 分為「乙」。
- 60(含)~69 分為「丙」。
- 0(含)~59 分為「丁」。

請回答以下問題來完成程式：

```
01 score = int(input("請輸入分數"))
02 _____
03     grade = '優'
04 _____
```

```

05  grade = '甲'
06  _____
07  grade = '乙'
08  _____
09  grade = '丙'
10 else:
11  grade = '丁'
12 print("成績等級為：", grade)

```

A：\_\_\_\_\_ 請問第 2 行應該填入下列哪段敘述？

- ① if score <= 90:    ② if score >= 90:    ③ if score > 90:    ④ if score == 90:

B：\_\_\_\_\_ 請問第 4 行應該填入下列哪段敘述？

- ① if score >= 80:    ② elif score >= 80:    ③ elif score > 80:    ④ else if score >= 80:

C：\_\_\_\_\_ 請問第 6 行應該填入下列哪段敘述？

- ① if score >= 70:    ② elif score >= 70:    ③ elif score > 70:    ④ else if score >= 70:

D：\_\_\_\_\_ 請問第 8 行應該填入下列哪段敘述？

- ① if score >= 60:    ② elif score >= 60:    ③ elif score > 60:    ④ else if score >= 60:

## 附錄B MTA Python 國際認證模擬試題解答

### 一. 資料類型和運算子

1. ③ 、 2. A: ③ 、 2. B: ② 、 3. ② ④ 、 4. A: int 、  
 4. B: bool 、 4. C: str 、 4. D: float 、 4. E: str 、  
 5. A: str 、 5. B: int 、 5. C: str 、 6. A: 是 、 6. B: 是 、  
 6. C: 否 、 7. A: float 、 7. B: float 、 7. C: str 、 7. D: bool 、  
 8. A: ③ 、 8. B: ④ 、 8. C: ② 、 9. A: ① 、 9. B: ② 、  
 9. C: ③ 、 9. D: ④ 、 10. ② 、 11. A: ④ 、 11. B: ④ 、  
 12. ① 、 13. A: ③ 、 13. B: ⑥ 、 13. C: ⑧ 、 13. D: ② 、  
 13. E: ⑧ 、 13. F: ① 、 14. ② ④ 、 15. ③ 、  
 16. ② ① ⑦ ③ ⑤ 、 17. ① 、 18. A: ③ 、 18. B: ② 、  
 18. C: ① 、 18. D: ① 、 19. ③ 、 20. A: ① 、 20. B: ③ 、  
 20. C: ① 、 21. A: ② 、 21. B: ① 、 21. C: ② 、  
 22. ④ ⑤ ③ ② ① ⑥ 、 23. A: ② 、 23. B: ④ 、 24. A: ② 、  
 24. B: ① 、 24. C: ① 、 25. A: ① 、 25. B: ① 。

## 二. 流程控制與迴圈

1. A : ② 、1. B : ② 、1. C : ② 、1. D : ② 、  
 2. A : ④ 、2. B : ① 、2. C : ② 、2. D : ③ 、  
 3. A : ① 、3. B : ③ 、3. C : ② 、3. D : ③ 、  
 4. A : ① 、4. B : ① 、4. C : ③ 、5. A : ① 、5. B : ② 、  
 5. C : ① 、6. A : ① 、6. B : ③ 、6. C : ⑤ 、7. A : ① 、  
 7. B : ③ 、8. ① ⑤ ② ④ 、9. A : ③ 、9. B : ② 、  
 10. A : ② 、10. B : ③ 、11. A : ① 、11. B : ⑦ 、11. C : ③ 、  
 11. D : ⑤ 、12. A : ⑤ 、12. B : ① 、12. C : ② 、12. D : ④ 、  
 12. E : ③ 。

## 三. 輸入與輸出

1. A : ④ 、1. B : ③ 、1. C : ② 、1. D : ③ 、2. A : ④ 、  
 2. B : ① 、2. C : ③ 、2. D : ⑤ 、2. E : ② 、3. ① 、  
 4. ① 、5. A : ② 、5. B : ④ 、5. C : ① 、6. ③ 、  
 7. A : ④ 、7. B : ② 、8. ① ⑦ ⑥ ⑤ 。

## 四. 文件結構

1. ④ 、2. ② 、3. A : 是 、3. B : 否 、3. C : 否 、  
 3. D : 是 、4. ① ⑥ 、5. A : 是 、5. B : 是 、5. C : 否 、  
 5. D : 否 。

## 五. 疑難排解和錯誤處理

1. ① 、2. ③ 、3. A : ④ 、3. B : ① 、4. ⑥ ⑦ ⑧ 、  
 5. A : ① 、5. B : ② 、6. A : 是 、6. B : 是 、6. C : 是 、  
 6. D : 否 。

## 六. 模組與工具的操作

1. ② 、2. ① 、3. ① ③ 、4. ① 、5. ① ③ 。