

## 1. 【CH01-1】資料型態的檢視：

你是運動 App

的程式設計師。你必須製作一個函式為跑者計算步速，所謂步速就是每公里所花的時間。輸出結果必須盡可能精准。要如何完成程式碼？請在回答區中選擇適當的程式碼片段。其中距離轉換為浮點數，分秒的輸入值都要轉換為整數。

```
#步速計算器
distance = _(選項 1)_(input("Enter the distance traveled in meters"))
# convert to kilometers
distance_kms = distance / 1000
#輸入已用時間(分鐘)
time_minute = _(選項 2)_(input("Enter the time elapsed in minutes"))
#輸入已用秒數
time_sec = _(選項 3)_(input("Enter the time elapsed in seconds"))
time = time_minute*60+time_sec
pace = time / distance_kms
#平均速度
print("The average velocity is", str((pace //60)) + " " + str ((pace%60)))
```

以上空格分別要填入的函式名為：

Quiz:

1. (1) int (2) string (3) float
2. (1) int (2) string (3) float
3. (1) int (2) string (3) float

Answer: 1: float (3), 2: int (1), 3: int (1)

Explanation:

- 使用 `float()` 將距離轉為浮點數，因為距離可能有小數（如 1500.5 公尺）。
- 使用 `int()` 將時間（分鐘和秒）轉為整數，因為這些值應該是整數。

## 2. 【CH01-2】你正在編寫一個 Python 程式用來記錄客戶資料並將其儲存在資料庫中。

這個程式處理各種各樣的資料。

以下的變數宣告後它們的資料類別是？

選項 1. age = 12

選項 2. minor = False

選項 3. name = 'David'

選項 4. weight = 64.5

選項 5. zip = '545'

Quiz:

1. (1) int (2) bool (3) str (4) float
2. (1) int (2) bool (3) str (4) float
3. (1) int (2) bool (3) str (4) float
4. (1) int (2) bool (3) str (4) float
5. (1) int (2) bool (3) str (4) float

Answer: 1: int (1), 2: bool (2), 3: str (3), 4: float (4), 5: str (3)

Explanation:

- 使用 `int()` 將年齡轉為整數，因為年齡應該是整數。

- 使用 `bool()` 將未成年轉為布林值，因為這是一個是非題。
- 使用 `str()` 將姓名轉為字串，因為姓名應該是字串。
- 使用 `float()` 將體重轉為浮點數，因為體重可能有小數（如 64.5 公斤）。
- 使用 `str()` 將郵遞區號轉為字串，因為郵遞區號應該是字串。

3. 【CH01-3】你正在編寫一個計算使用者出生年份的程式，該程式詢問使用者的年齡和當前年份，然後輸出使用者的出生年份。

你編寫以下程式碼：

```
age = input("Enter your age: ")
year = input("Enter the four digit year: ")
born = eval(year) - eval(age)
message = "You were born in " + str(born)
```

下列選項正確答案為？

Quiz:

1. 第 1 行中的 `age` 的資料類型是 `str`
2. 第 3 行中的 `born` 的資料類型是 `float`
3. 第 4 行中的 `message` 的資料類型是 `bool`

Answer: 1

Explanation:

- 在第 1 行，`age` 是從使用者輸入獲得的，因此它們的資料類型是字串 (str)。
- 在第 3 行，使用 `eval()` 會進行文字轉數字作業，但如果沒有明確指定型別，結果會是動態產生，因為 `age` 及 `year` 都是整數型態的字串，因此解析下來兩都皆為整數，而整數減掉整數結果仍為整數，因此 `born` 的資料類型是整數 int。
- 在第 4 行，將浮點數轉換為字串後與另一個字串連接，因此 `message` 的資料類型應該是字串 (str)，而不是布林值 (bool)。

4. 【CH01-4】在 Python 資料類型的課程中創建以下三個程式碼片段：

```
# 程式碼片段 1
x1 = "5"
y1 = 4
a = x1 * y1
# 程式碼片段 2
x2 = 10
y2 = 4
b = x2 / y2
# 程式碼片段 3
x3 = 5.5
```

```
y3 = 1  
c = x3 / y3
```

Quiz:

1. 執行程式碼片段 1 後，變數 a 的資料類為 str。
2. 執行程式碼片段 2 後，變數 b 的資料類型是 float
3. 執行程式碼片段 3 後，變數 c 的資料類型為 int

Answer: 1, 2

Explanation:

- 在 `程式碼片段 1` 中，`x1` 是字串類型，`y1` 是整數類型，在 Python 中，當你使用乘法運算子 `\*` 在一個字串和一個整數之間時，它執行的不是數學上的乘法，而是 `字串重複`。因此會將字串 `5` 重複 4 次 = `5555`，因此變數 a 的資料類型仍然是字串 (str)。
- 在 `程式碼片段 2` 中，`x2` 和 `y2` 都是整數類型，除法運算會產生浮點數，因此變數 b 的資料類型是浮點數 (float)。
- 在 `程式碼片段 3` 中，`x3` 是浮點數，除以整數 `y3` 後結果仍然是浮點數，因此變數 c 的資料類型也是浮點數，而不是整數 (int)。

## 5. 【CH01-5】在 Python 程式中我們利用 type() 查詢每個值的資料類別，以下的程式執行後出現的資料類別分別是：

```
type(+1E10)  
type(5.0)  
type("True")  
type(False)
```

Quiz:

1. int, int, bool, bool
2. float, float, str, bool
3. int, float, str, bool
4. float, int, str, str

Answer: 2

Explanation:

- `+1E10` 是科學記數法表示的浮點數，`type(+1E10)` 返回 `float`。
- `5.0` 是浮點數，`type(5.0)` 返回 `float`。
- `True` 是字串，`type("True")` 返回 `str`。
- `False` 是布林值，`type(False)` 返回 `bool`。

因此正確答案是 `float, float, str, bool`。

## 6. 【CH01-6】你加入了電子商務公司成為其程式開發部門的實習生。

你的程式中有一個地方要讓使用者提供一個數值。

即使使用者輸入了小數，該值也必須轉換為整數來進行計算。

你應該使用哪個程式碼片段？？

Quiz:

```
1. totalNums = input("How many items would you like?")
2. totalNums = int(input("How many items would you like?"))
3. totalNums = str(input("How many items would you like?"))
4. totalNums = float(input("How many items would you like?"))
```

Answer: 2

Explanation:

- 使用 `int()` 將年齡轉為整數，因為年齡應該是整數。
- 使用 `bool()` 將未成年轉為布林值，因為這是一個是非題。
- 使用 `str()` 將姓名轉為字串，因為姓名應該是字串。
- 使用 `float()` 將體重轉為浮點數，因為體重可能有小數（如 64.5 公斤）。
- 使用 `str()` 將郵遞區號轉為字串，因為郵遞區號應該是字串。

7. 【CH01-7】你設計了一個 Python 程式用來顯示每個員工每天工作到現在的小時數，你需要計算工作時數並顯示訊息，程式碼如下：

```
start = input("What time do you start work today?")
end = input("what time is it now?")
```

如果要完成這個程式，在 3 行應該使用哪個程式碼？

Quiz:

1. print("You have worked for" + str(int(end) - int(start)) + "hours")
2. print("You have worked for" + (int(end) - int(start)) + "hours")
3. print("You have worked for" + str(end - start) + "hours")
4. print("You have worked for" + int(end - start) + "hours")

Answer: 1

Explanation:

- `input()` 函數功能可輸入值，轉成文字型別的值傳給變數。
- 因此需要使用 `int()` 將 `start` 和 `end` 先將輸入變數值轉換為整數，然後才能計算工作時數。
- 最後使用 `str()` 將結果轉換為字串以便與其他字串連接。

8. 【CH02-1】高年級的老師要製作一份報表來顯示這次考試班上所有學生的平均分數。報表必須去除平均分數的小數部分。

每個正確的答案都提供了一個完整的解決方案。

你應該使用哪兩個程式碼片段？

Quiz:

1. average\_score = float(total\_score // number\_of\_students)
2. average score = int(total\_score / number of students)
3. average\_score = float(total\_score \*\* number\_of\_students)
4. average\_score = total\_score // number\_of\_students

Answer: 2, 4

Explanation:

- 使用 `float()` 將平均分數轉換為浮點數，會保留小數部分，不符合要求。

- `int()` 將平均分數轉換為整數，去除小數部分。
- 使用 `\*\*` (幕運算符)，這不是計算平均分數的正確方法。
- 使用 `//` (整除運算符)，這會直接計算整數平均分數，去除小數部分。
- 因此正確答案是使用 `int()` 及 `//` (整除運算符)。

## 9. 【CH02-2】你編寫了以下的程式碼：

```
a = 24
b = 7
ans = (a % b * 100) // 2.0 ** 3.0 - b
print(ans)
```

執行程式碼的輸出值是？

Quiz:

1. 30.0
2. 30.5
3. 457
4. 語法錯誤

Answer: 1

Explanation:

- 首先計算 `a % b`，即 `24 % 7`，結果是 `3`。
- 然後計算 `3 \* 100`，結果是 `300`。
- 接著計算 `2.0 \*\* 3.0`，即 `2.0` 的立方，結果是 `8.0`。
- 然後計算 `(300 // 8.0)`，整除運算結果是 `37.5`。
- 最後計算 `37.5 - 7`，結果是 `30.5`。

## 10. 【CH02-3】你正在編寫 Python 程式用於計算一個數學公式。

公式內容為  $b = a + 5^2$ ，然後再平方，其中  $a$  是輸入的值， $b$  是結果。

你設計了以下的程式碼片段：

```
a = eval(input("Enter a number for the equation:"))
b =
```

如何完成 2 行的程式碼？

Quiz:

1.  $b = (a + 5)^{**2}$
2.  $b = a + 5^{**2}$
3.  $b = a + 5 * 2$
4.  $b = a + (5^{**2})$

Answer: 1

Explanation:

- 正確是將 `a` 先加上 `5`，然後將結果平方。
- 因此正確答案是 `b = (a + 5)\*\*2`。

11. 【CH02-4】你正在開發一個補習班的 Python 函式來計算折扣，補習班希望鼓勵小朋友和老年人報名，只要是小朋友和老年人報名相關課程就會獲得10%的折扣。你編寫了以下程式碼：

```
def get_discount(kid, senior):
    discount = 0.1
    if not (kid or senior):
        discount = 0
    return discount
```

為了完成這個程式碼，你應該在 3 行加入什麼程式碼？

Quiz:

1. if not (kid or senior):
2. if (not kid) or senior:
3. if not (kid and senior):
4. if (not kid) and senior:

Answer: 1

Explanation:

- 正確地檢查如果 `kid` 和 `senior` 都是 `False`，即都不是小朋友和老年人身份，則不應該給予折扣。
- 因此正確答案是 if not (kid or senior):。

12. 【CH02-5】你開發了一個比較數字的 Python 程式，下列何者的值是 True ? (可複選)

Quiz:

1. 0 or 5
2. bool(0)
3. None is None
4. -5 < 0 < 5

Answer: 3, 4

Explanation:

- `0 or 5` 的結果是 `5`，因為 `or` 運算符會返回第一個真值，因此這個表達式的值是 `5`，而不是 `True`。
- `bool(0)` 的結果是 `False`，因為 `bool(0)` 返回 `False`。
- `None is None` 的結果是 `True`，因為 `None is None` 確實是 `True`。
- `-5 < 0 < 5` 的結果是 `True`，因為這是一個連鎖比較，表示 `-5` 小於 `0` 且 `0` 小於 `5`。

13. 【CH02-6】計算以下的 Python 數學運算式結果為何？

$(3*(1+2)**2-2**2*3)$

Quiz:

1. 3
2. 13
3. 15
4. 69

Answer: 3

Explanation:

- 首先計算括號內的部分: `(1 + 2)` 等於 `3`。
- 然後計算 `3 \*\* 2`，結果是 `9`。
- 接著計算 `2 \*\* 2`，結果是 `4`。
- 現在將這些值代入原式:  
` $(3 * 9 - 4 * 3)$ `  
`= (27 - 12)`  
`= 15`。

14. 【CH02-7】你編寫了以下的程式碼：

```
a = 'Test 1'  
print(a)  
b = 'Test 2'  
a += b  
print(a)  
print(b)
```

根據程式碼片段中提供的資訊選擇每個問題的答案選項。

在第一次 print 後會顯示什麼？\_(選項 1)\_

在第二次 print 後會顯示什麼？\_(選項 2)\_

在第三次 print 後會顯示什麼？\_(選項 3)\_

Quiz:

1. (1) Test 1 (2) Test 1Test 2 (3) Test 2
2. (1) Test 1 (2) Test 1Test 2 (3) Test 2
3. (1) Test 1 (2) Test 1Test 2 (3) Test 2

Answer: 1: Test 1 (1), 2: Test 1Test 2 (2), 3: Test 2 (3)

Explanation:

- 在第一次 print 後，顯示的是變數 a 的值，即'Test 1'。
- 在第二次 print 後，顯示的是變數 a 的值，即'Test 1Test 2'，因為在這一行中，變數 a 被更新為原本的值加上變數 b 的值。
- 在第三次 print 後，顯示的是變數 b 的值，即'Test 2'。

15. 【CH02-8】你為公司開發了一個 Python 應用程式，程式碼如下：

```
def test(a, b, c, d):  
    value=(a+b)*c-d  
    return value
```

根據程式碼片段中提供的資訊選擇每個問題的答案選項。

選項1 :運算式的哪個部份將第一個進行計算?

選項2 :運算式的哪個部份將第二個進行計算?

選項3 :哪個運算式等於題目函式中的運算式?

Quiz:

1. (1)  $a+b$  (2)  $b*c$  (3)  $c-d$
2. (1) 加(+) (2) 減(-) (3) 乘(\*)
3. (1)  $(a+b)*(c-d)$  (2)  $(a+(b*c))-d$  (3)  $((a+b)*c)-d$

Answer: 1:  $a+b$  (1), 2: 乘(\*) (3), 3:  $((a+b)*c)-d$  (3)

Explanation:

- 選項1 的運算式為 ` $a+b$ `，將首先進行計算。
- 選項2 的運算式為 ` $b*c$ `，將其次進行計算。
- 選項3 的運算式為先執行 ' $a + b$ ' 再 ' $* c$ ' 再 ' $- d$ '，將最後進行計算。

16. 【CH02-9】請按先後順序從頭至尾排列這六類運算的正確順序:

1. 加法和減法(+, -)
2. 乘法和除法(\*, /)
3. 正數 (+)、負數 (-) 與反位元 (not)
4. 括弧
5. 指數 (\*\*)
6. 且 (And)

Quiz:

1. 加法和減法 -> 乘法和除法 -> 正數、負數與反位元 -> 括弧 -> 指數 -> 且
2. 括弧 -> 指數 -> 正數、負數與反位元 -> 乘法和除法 -> 加法和減法 -> 且
3. 指數 -> 乘法和除法 -> 正數、負數與反位元 -> 括弧 -> 且 -> 加法和減法
4. 乘法和餘法 -> 括弧 -> 正數、負數與反位元 -> 指數 -> 且 -> 加法和減法

Answer: 2

Explanation:

• 在 Python 中，運算的優先順序如下:

1. 括弧
2. 指數 (\*\*)
3. 正數 (+)、負數 (-) 與反位元 (not)
4. 乘法和除法 (\*, /)
5. 加法和減法 (+, -)
6. 且 (And)

• 因此正確的順序是：括弧 -> 指數 -> 正數、負數與反位元 -> 乘法和除法 -> 加法和減法 -> 且。

17. 【CH02-10】租車公司需要一種方法來決定客戶租用車輛的費用，該費用取決於車輛歸還的時間，然而週四和週日也有特別的費率。費用結構如下所示：

1. 費用每天 100 美元。
2. 如果在晚上 11 點後返還，將被多收取額外一天的費用。
3. 如果是在星期天租的，可享受 10% 的折扣。
4. 如果是在星期四租的，可以享受 20% 的折扣。

你需要撰寫程式碼去符合這個需求，要如何完成這段程式碼？

```

# 車輛出租計算機
ontime = input("Was car returned before 11 pm? y or n").lower()
days_rented = int(input("How many days was car rented?"))
day_rented = input("What day was the car rented?").capitalize()
cost_per_day = 100
if ontime_(選項 1)_ #不準時
    days_rented += 1
if day_rented_(選項 2)_ #星期天租的
    total = (days_rented * cost_per_day) * 0.9
elif day_rented_(選項 3)_ #星期四租的
    total = (days_rented*cost_per_day) * 0.8
else:
    total = days_rented * cost_per_day
print("Cost of the car rental is : $", total)

```

以上空格選項分別要填入的程式為：

Quiz:

1. (1) != "n": (2) == "n": (3) -- "y":
2. (1) == "Sunday": (2) >= "Sunday": (3) is "Sunday":
3. (1) == "Thursday": (2) <= "Thursday": (3) is "Thursday":

Answer: 1: == "n": (2), 2: == "Sunday": (1), 3: == "Thursday": (1)

Explanation:

- 選項1 的程式碼為 `ontime == "n"`，表示如果車輛不準時歸還，需要加費 1 天費用。
- 選項2 的程式碼為 `day\_rented == "Sunday"`，表示如果在星期天租車，則可以享受 10% 折扣。
- 選項3 的程式碼為 `day\_rented == "Thursday"`，表示如果在星期四租車，則可以享受 20% 折扣。

18. 【CH02-11】你設計了一個數學運算的 Python 程式，程式碼如下：

```
a = 11
b = 5
```

選項答案選擇區

```
print(a / b)的輸出結果是?_(選項 1)_
print(a // b)的輸出結果是?_(選項 2)_
print(a % b)的輸出結果是?_(選項 3)_

```

Quiz:

1. (1) 1 (2) 2 (3) 2.2
2. (1) 1 (2) 2 (3) 2.2
3. (1) 1 (2) 2 (3) 2.2

Answer: 1: 2.2 (3), 2: 2 (2), 3: 1 (1)

Explanation:

- `print(a / b)` 的輸出結果是 `2.2`，因為這是浮點數除法，返回除法運算的 `值`。
- `print(a // b)` 的輸出結果是 `2`，因為這是整除運算，返回除法運算的 `商數`。

- `print(a % b)` 的輸出結果是 `1`，因為這是整除運算，返回除法運算的 `餘數`。

### 19. 【CH02-12】你設計了一個比較數字的 Python 程式，內容如下：

```
n1 = eval(input("Please enter the first number: "))
n2 = eval(input("Please enter the second number: "))
if n1 = n2:
    print("The two numbers are equal.")
if n1 < n2:
    print("Number 1 is greater than number 2.")
if n2 <> n1
print("The two numbers are the same.")
```

針對下列每個敘述，如果正確就選擇 Yes，否則請選擇 No。

選項答案選擇區

選項 1. 第 3 行的語法是不正確的比較。

選項 2. 第 6 行的語法只有 n1 小於 n2 時才會列印出來。

選項 3. 第 8 行的語法只有 n1 大於 n2 時才會列印出來。

選項 4. 第 9 行的語法是不正確的比較。

Quiz:

1. (1) Yes (2) No
2. (1) Yes (2) No
3. (1) Yes (2) No
4. (1) Yes (2) No

Answer: 1: Yes (1), 2: No (2), 3: Yes (1), 4: Yes (1)

Explanation:

- 第 3 行的語法是不正確的比較，應該是用 `==` 取代 `=`。
- 第 6 行的語法只有 n1 小於等於 n2 時才會列印出來。
- 第 8 行的語法只有 n1 大於 n2 時才會列印出來。
- 在 9 行的語法是不正確的比較，應該是用 `==` 取代 `<>`。

### 20. 【CH02-13】老闆要求你對以下程式碼除錯，什麼將會輸出列印到螢幕上？：

```
x = 0
while x < 4:
    if x % 4 == 0:
        print("party")
    elif x - 2 < 0:
        print("cake")
    elif x / 3 == 0:
        print("greeting")
    else:
        print("birthday")
```

`x = x + 1`

請選擇正確的輸出結果。

Quiz:

1. party, greeting, birthday, cake
2. party, cake, birthday, birthday
3. birthday, party, greeting, cake
4. birthday, greeting, party, cake

Answer: 2

Explanation:

- #x=0時: party、x=1時: cake、x=2時: birthday、x=3時: birthday

21. 【CH02-14】您正在撰寫一個 Python 程式評估算術公式。此公式描述 a 是先取得一個輸入值，而 b 等於 a 乘以 -1 之後再平方，所以 b 是結果。

您所建立的程式碼片段如下。

```
a = eval(input("Enter a number for the equation: "))
```

b = 選項1 選項2 選項3 選項4 選項5

您需要確保結果正確。第 2 行程式中有 5

個空格選項，請選擇程式碼片段中的內容到正確的位置上，以完成第 2 行的程式碼。每個程式碼片段可能使用一次，也可能使用多次，甚至完全用不到。

Quiz:

1. (1) - (2) ((3)) (4) \*\* (5) \*\*2 (6) 2 (7) a
2. (1) - (2) ((3)) (4) \*\* (5) \*\*2 (6) 2 (7) a
3. (1) - (2) ((3)) (4) \*\* (5) \*\*2 (6) 2 (7) a
4. (1) - (2) ((3)) (4) \*\* (5) \*\*2 (6) 2 (7) a
5. (1) - (2) ((3)) (4) \*\* (5) \*\*2 (6) 2 (7) a

Answer: 1: ( (2), 2: - (1), 3: a (7), 4: ) (3), 5: \*\*2 (5)

Explanation:

- 在第 2 行的程式碼中，`b = -a \*\* 2` 是正確的。
- `-` 表示負號，`a` 是輸入的數字，`\*\*` 是指數運算符，表示平方。
- 因此正確的程式碼片段應該是 `b = (-a)\*\*2`。

22. 【CH03-1】你編寫了以下的程式碼：

```
list_1 = [1, 2, 3]
list_2 = [4, 5, 6]
list_3 = list_1 + list_2
list_4 = list_3 * 2
print(list_4)
```

執行程式碼的輸出值是？

Quiz:

1. [1, 2, 3], [4, 5, 6], [1, 2, 3], [4, 5, 61]
2. [4, 10, 18]

- 3. [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]
- 4. [[1, 2, 3, 4, 5, 6], [1, 2, 3, 4, 5, 6]]

Answer: 3

Explanation:

- list\_4 的值是 [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]，因為 list\_3 包含了 list\_1 和 list\_2 的所有元素，然後再乘以 2，這樣就重複了兩次。

23. 【CH03-2】在下列的程式碼中：

```
aList = [0, 1, 2, 3, 4]
```

```
print(4 in aList)
```

會輸出列印的內容？

Quiz:

- 1. 4
- 2. 5
- 3. True
- 4. False

Answer: 3

Explanation:

- `in` 關鍵字用於檢查某個值是否存在於序列中，這裡是檢查 4 是否在 aList 中。

- 因此，程式碼會輸出 True。

24. 【CH03-3】你為公司開發了一個 Python 應用程式，設計了以下的程式碼：

```
aList = ["a", "b", "c", "d", "e"]
```

```
bList = [1,2,3,4,5]
```

```
print(aList is bList)
```

```
print(aList == bList)
```

```
aList = bList
```

```
print(aList is bList)
```

```
print(aList == bList)
```

根據程式碼片段中提供的資訊選擇每個問題的答案選項。

選項 1. 第一次 print (第 3 行)後會顯示什麼？

選項 2. 第二次 print (第 4 行)後會顯示什麼？

選項 3. 第三次 print (第 6 行)後會顯示什麼？

選項 4. 第四次 print (第 7 行)後會顯示什麼？

Quiz:

- 1. (1) True (2) False
- 2. (1) True (2) False
- 3. (1) True (2) False
- 4. (1) True (2) False

Answer: 1: False (2), 2: False (2), 3: True (1), 4: True (1)

Explanation:

- 在第一次 print 後，`aList is bList` 會顯示 False，因為它們是不同的物件。

- 在第二次 print 後，`aList == bList` 會顯示 False，因為它們的內容不同。
- 在第三次 print 後，`aList is bList` 會顯示 True，因為它們現在指向同一個物件。
- 在第四次 print 後，`aList == bList` 會顯示 True，因為它們的內容相同。

25. 【CH03-4】同事開發一個將產品名稱輸入到資料庫的程式，但是其中發生了錯誤，每個存入的名稱字母順序都顛倒了。

請你開發一個 Python

函式，將每個產品名稱以正確的順序輸出。請選擇適當的程式碼片段來完成程式：

# 函式會反轉字串中的字元，以相反的順序返回新字串。

```
def reverse_pname(backwards_pname):
    forward_pname = ""
    for index in _(選項 1)_:
        forward_pname += _(選項 2)_
    return forward_pname
print(reverse_pname("klim"))
```

選項 1 答案選擇區

答案 A.backwards\_pname

答案 B.len(backwards\_pname)

答案 C.range(0, len(backwards\_pname), -1)

答案 D.range(len(backwards\_pname)-1, -1, -1)

選項 2 答案選擇區

答案 A.backwards\_pname[index-1]

答案 B.backwards\_pname[len(forward\_pname)-1]

答案 C.backwards\_pname[len(backward\_pname)-len(forward\_pname)]

答案 D.backwards\_pname[index]

Quiz:

1. (1) A (2) B (3) C (4) D

2. (1) A (2) B (3) C (4) D

**Answer: 1: D (4), 2: D (4)**

Explanation:

• 在第一個空格中，應該使用 `range(len(backwards\_pname)-1, -1, -1)`，這樣可以從字串的最後一個字元開始迭代到第一個字元。

• 在第二個空格中，應該使用 `backwards\_pname[index]`，這樣可以正確地取得每個字元並將其添加到 `forward\_pname` 中。

26. 【CH03-5】你有以下清單結構：

alph = "abcdefghijklmnopqrstuvwxyz"

以下各個程式碼片段的結果各是如何？

請將回答區項目的選項正確填到程式碼片段的選項中

#程式碼片段

1. alph[3: 15] = (\_選項1\_)

2. alph[3:15:3] = (\_選項2\_)

3. alph[15:3:-3] = (\_選項3\_)

4. `alph[::-3] = (_選項4_)`

答案選項區

- A. zwtqnkheb
- B. pmjg
- C. defghijklmno
- D. ponmlkjihgfe
- E. defghijklmnop
- F. dgjm
- G. olif

Quiz:

- 1. (1) A (2) B (3) C (4) D (5) E (6) F (7) G
- 2. (1) A (2) B (3) C (4) D (5) E (6) F (7) G
- 3. (1) A (2) B (3) C (4) D (5) E (6) F (7) G
- 4. (1) A (2) B (3) C (4) D (5) E (6) F (7) G
- 5. (1) A (2) B (3) C (4) D (5) E (6) F (7) G

Answer: 1: C (3), 2: F (6), 3: B (2), 4: A (1)

Explanation:

- 在第一個程式碼片段中，`alph[3: 15]` 會返回從索引 3 到 14 的子字串，即 `defghijklmno`。
- 在第二個程式碼片段中，`alph[3: 15: 3]` 會返回從索引 3 開始，每隔 3 個字元取一個，即 `dgjm`。
- 在第三個程式碼片段中，`alph[15: 3: -3]` 會返回從索引 15 到 4 的子字串，每隔 3 個字元取一個，即 `ponmlkjihgfe`。
- 在第四個程式碼片段中，`alph[::-3]` 會返回整個字串的反向，每隔 3 個字元取一個，即 `olif`.

27. 【CH03-6】你為學校設計了一個 Python 應用程式，在 classroom 的清單中包含了 60 位同學的姓名，最後 3 名是班上的幹部。

你需要分割清單內容顯示除了幹部以外的所有同學，你可以利用以下哪二個程式碼達成？

可以答成回答 Yes，不能回答回答 No

- 1. `classroom[0: -2] = (_選項 1_)`
- 2. `classroom[0: -3] = (_選項 2_)`
- 3. `classroom[1: -3] = (_選項 3_)`
- 4. `classroom[: -3] = (_選項 4_)`
- 5. `classroom[1: -3] = (_選項 5_)`

Quiz:

- 1. (1) Yes (2) No
- 2. (1) Yes (2) No
- 3. (1) Yes (2) No
- 4. (1) Yes (2) No
- 5. (1) Yes (2) No

Answer: 1: No (2), 2: Yes (1), 3: No (2), 4: Yes (1), 5: No (2)

Explanation:

- 在這個情況下，`classroom[0: -3]` 和 `classroom[: -3]` 都可以正確地取得除了最後 3 名同學以外的所有同學。

- 因此，選項 2 和選項 4 是正確的。

28. 【CH03-7】你開發了一個 Python 應用程式，其中有一個名為 month 的清單單儲存所有月份的英文。  
你要分割這個清單，取得由第二個月份開始，每間隔一個值的月份名稱，你應該使用哪個程式碼？

Quiz:

1. month[2:2]
2. month[:2]
3. month[1::2]
4. month[1:2]

Answer: 3

Explanation:

- Python 陣列元素值由 0 開始算，0 = 第 1 個元素，1 = 第 2 個元素，以此類推。
- month[2:2] 這會建立一個從索引 2 開始到索引 2 (不包含) 的切片，結果會是一個空清單。
- month[:2] 這會從清單的開頭 (索引 0) 開始，每間隔一個值取一個元素。這會包含第一個月份，但題目要求從第二個月份開始。
- month[1::2] 從索引 1 (第二個元素) 開始一直取到清單的末尾，每次取一個元素後，跳過接下來的一個元素 (步長為 2)。
- month[1:2]: 這會建立一個從索引 1 開始到索引 2 (不包含) 的切片，結果只會包含第二個月份的名稱。

29. 【CH03-8】你設計了一個函式來執行除法，因為除法的除數不能為零，所以在函式中必須要針對這個重點進行檢查。

你要如何完成這段程式碼？

請在回答區選擇適當的程式碼片段。

```
def safe_divide(numerator, denominator):  
    _(選項 1)_  
        print("A required value is missing.")  
    _(選項 2)_  
        print("The denominator is zero.")  
    else:  
        return numerator / denominator
```

\_(選項 1)\_ 答案選項

答案 A. if numerator is None or denominator is None:

答案 B. if numerator is None and denominator is None:

答案 C. if numerator = None or denominator = None:

答案 D. if numerator - None and denominator = None:

\_(選項 2)\_ 答案選項

答案 A. elif denominator == 0:

答案 B. elif denominator = 0:

答案 C. elif denominator != 0:

答案 D. elif denominator in 0:

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D

Answer: 1: A (1), 2: A (1)

Explanation:

- 在選項 1 中，正確的程式碼是 `if numerator is None or denominator is None:`, 這樣可以檢查分子或分母是否為 `None`。

- 在選項 2 中，正確的程式碼是 `elif denominator == 0:`, 這樣可以檢查分母是否為零。

### 30. 【CH03-9】您建立了下列程式來找出學生並顯示姓名。

```
students = {1: 'Jone', 2: 'Mary'}  
student = input('Enter the student id: ')  
if student not in students:  
    print('The student does not exist.')  
else:
```

```
    print('The student name is ' + students[student])
```

同事回報指出,這個程式有時候產生的結果不正確, 請回答以下問題:

1. 請問哪兩種資料型態儲存在第 1 行的 students 中?

A. 整數和字串 B. 整數和浮點數 C. 字串和浮點數 D. 字串和布林值

2. 第 2 行的 student 是哪種資料型態?

A. int B. float C. string D. bool

3. 為什麼第 3 行找不到學生?

A. 語法不正確 B. 不符合的資料型態 C. 變數命名錯誤

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D
3. (1) A (2) B (3) C

Answer: 1: A (1), 2: C (3), 3: B (2)

Explanation:

- 在這個程式中, `students` 是一個字典, 包含了整數和字串兩種資料型態。

- `student` 變數是由使用者輸入的, 因此它的資料型態是字串。

- 第 3 行找不到學生的原因是因為 `student` 變數的資料型態是字串, 而 `students` 字典的鍵是整數, 這導致了不符合的資料型態。

### 31. 【CH04-1】老師正在設計一個 Python 程式, 學生可以使用它來記錄他們考試的平均分數。

該程式必須允許使用者輸入他們的名字和當前分數。

該程式將輸出使用者名和使用者的平均分數。

輸出必須符合以下要求:

- 使用者姓名必須是靠左對齊的。
- 如果使用者姓名少於 20 個字元, 則必須在右側添加額外的空間。
- 平均分數在小數點左方是三位數, 小數點右方是二位數。(XXX.XX)。

你要如何完成程式碼? 請在回答區中選擇適當的程式碼片段。

```
name = input('Please enter your name:')
```

```
score = count = 0
```

```

total = 0
while (score != -1):
    score = int(input('Enter your score,-1 for done:'))
    if score == -1:
        break
    total += score
    count += 1
average = total / count
print('_選項 1)_ your average is _選項 2)_ %(name, average))

```

#### 問題回答區

選項 1. 答案 A. %-20i 答案 B. %-20d 答案 C. %-20f 答案 D. %-20s

選項 2. 答案 A. %1.6s 答案 B. %6.2f 答案 C. %6.2d 答案 D. %2.6f

#### Quiz:

- 1. (1) A (2) B (3) C (4) D
- 2. (1) A (2) B (3) C (4) D

**Answer: 1: D (4), 2: B (2)**

#### Explanation:

- `name` 變數的資料型態是字串(使用 s)。
- `average` 變數的資料型態是浮點數(使用 f)，且需要有小數2位，即 `%6.2f` 符合。

### 32. 【CH04-2】你正在設計一個函式以讀取資料檔案並將結果列印為格式化表格。

資料檔案中包含有關蔬菜的資訊。每個記錄都包含蔬菜的名稱、重量和價格。

你需要列印資料，使其看起來像下面的範例：

Potatos 5.4 2.33

Carrots 2.5 1.50

Corns 5.2 5.96

具體地說，列印輸出必須符合以下要求：

- 蔬菜名稱必須印在 10 個空格範圍內並靠左對齊。
- 重量必須印在 5 個空格範圍內並靠右對齊，小數點最多一個位數。
- 價格必須印在 7 個空格範圍內並右對齊，小數點後最多兩位數。

你創建了以下的程式碼。

```

def print_table(file):
    data = open(file,'r')
    for record in data:
        fields = record.split(',')
    ? ? ?

```

你應該如何完成第 5 行的程式碼？請將適合的程式碼片段選項填到正確的位置上。每個程式碼片段都可以使用一次、多次，或者不使用。

#### 問題選項區

(選項1) (選項2) (選項3) (選項4)

.format(fields[0], eval(fields[1]), eval(fields[2])))

#### 選項程式碼片段區

答案 A. print('

答案 B. {10: 0}

答案 C. {5: 1f}

答案 D. {7: 2f}

答案 E. {2: 7.2f}

答案 F. {1: 5.1f}

答案 G. {0: 10}

Quiz:

1. (1) A (2) B (3) C (4) D (5) E (6) F (7) G

2. (1) A (2) B (3) C (4) D (5) E (6) F (7) G

3. (1) A (2) B (3) C (4) D (5) E (6) F (7) G

4. (1) A (2) B (3) C (4) D (5) E (6) F (7) G

Answer: 1: A (1), 2: G (7), 3: F (6), 4: E (5)

Explanation:

- 在第 5 行的程式碼中，應該使用 `print('{0: 10} {1: 5.1f} {2: 7.2f}'.format(fields[0], eval(fields[1]), eval(fields[2])))`

33. 【CH04-3】你正在設計一個電子商務程式，它可以接受來自使用者輸入，並以逗號分隔的格式輸出資料。

你可以編寫以下程式碼接受資料輸入：

```
product = input('Enter product name:')
```

```
qty = input('Input quantity:')
```

輸出必須符合以下要求：

- 字串必須括在雙引號內。
- 數字不得用引號或其他字元括起來。
- 每個產品必須用逗號隔開。

你需要完成程式碼以符合要求。你應該使用哪三個程式碼片段？

Quiz:

1. print("{0}, {1}'.format(product, qty))
2. print('' + product + ', ' + str(qty))
3. print("%s, %s" % (product, qty))
4. print'{0}, {1}'.format(product, qty))
5. print(product + ', ' + qty)

Answer: 1, 2, 3

Explanation:

- 選項 1 使用了 `format` 方法。
- 選項 2 使用了字符串拼接。
- 選項 3 使用了舊式的格式化方法 `%`。
- 選項 4 和 5 沒有將產品名稱括在雙引號內，因此不符合要求。

34. 【CH04-4】請檢視下面的程式碼：

```
x = 'Tiger'
```

```
y = 'Lion'
```

```
z = 'Jaguar'
```

```
animals = '{1} and {0} and {2}'
```

```
print(animals.format(x, y, z))
```

輸出的結果為？

Quiz:

1. Lion and Tiger and Jaguar
2. Lion and Jaguar and Tiger
3. Jaguar and Lion and Tiger
4. Tiger and Lion and Jaguar

Answer: 1

Explanation:

- `animals` 字串使用了格式化方法 `format`，

其中 `{1}` 代表第二個參數 `y`，`{0}` 代表第一個參數 `x`，`{2}` 代表第三個參數 `z`。

因此輸出的結果是 `Lion and Tiger and Jaguar`。

### 35. 【CH04-5】你為公司設計 Python

應用程式，需要接受來自使用者的輸入並將該資訊列印到螢幕上。你的程式碼如下：

```
print('Enter product name: ')
```

?

```
print(product_name)
```

在02行應該編寫哪個程式碼？

Quiz:

1. product\_name = input()
2. input(product\_name)
3. input('product\_name')
4. name = product\_input

Answer: 1

Explanation:

- `product\_name` 變數用來儲存使用者輸入的產品名稱。

- `input()` 函數會顯示提示訊息，並等待使用者輸入。

- 使用者輸入的值會被賦值給 `product\_name` 變數，然後在最後一行被列印出來。

### 36. 【CH04-6】有一個旅行社需要一個簡單的程式用來輸入合作飯店及民宿的調查資料。

- 程式必須接受輸入並返回基於五顆星規模的平均評等。

- 輸出必須四捨五入到小數第二位。

你必須完成這個程式碼以符合需求，你要如何完成這個程式碼？

請在選項答案選擇區選擇適當的程式碼片段。

```
sum = count = done = 0
average = 0.0
while (done != -1):
    rating = _(選項 1)
    if rating == -1:
        break
    sum += rating
```

```
count += 1  
average = float(sum / count)  
_(選項 2)_ + _(選項 3)_
```

#### 選項1答案選擇區

- 答案 A. `print('Enter next rating (1-5), -1 for done')`  
答案 B. `float(input('Enter next rating (1-5), -1 for done'))`  
答案 C. `input('Enter next rating (1-5), -1 for done')`  
答案 D. `input 'Enter next rating (1-5), -1 for done'`

#### 選項2答案選擇區

- 答案 A. `output('The average star rating for this hotel is:')`  
答案 B. `console.input('The average star rating for this hotel is:')`  
答案 C. `printline('The average star rating for this hotel is:')`  
答案 D. `print('The average star rating for this hotel is:')`

#### 選項3答案選擇區

- 答案 A. `format(average, '.2f')`  
答案 B. `format(average, '.2d')`  
答案 C. `(average, '.2f')`  
答案 D. `format.average.(2d)`

#### Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D
3. (1) A (2) B (3) C (4) D

**Answer: 1: B (2), 2: D (4), 3: A (1)**

#### Explanation:

- 選項 1 使用了 `print` 函數來顯示提示訊息。
- 選項 2 使用了 `float` 函數來將輸入的字串轉換為浮點數。
- 選項 3 使用了舊式的格式化方法 `%`。
- 選項 4 和 5 沒有將產品名稱括在雙引號內，因此不符合要求。

### 37. 【CH04-7】你必須開發一個簡單的 Python 檔案程式來執行以下的動作：

- 檢查檔案是否存在。
- 如果該檔案存在，就顯示檔案內容。
- 如果該檔案不存在，就使用指定的名稱新增檔案。

• 在檔案最後加入文字：'End of file'。

你需要完成程式碼以符合要求。你要如何完成這段程式碼？  
請在回答區選擇適當的程式碼片段。

```
import os  
if _(選項 1)_
```

```
file = open('theFile.txt')
_(選項 2)_  
file.close()
file = _(選項 3)_  
_(選項 4)_ ('End of file')
file.close()
```

選項1答案選擇區

- 答案 A. isfile('theFile.txt')
- 答案 B. os.exist('theFile.txt')
- 答案 C. os.find('theFile.txt')
- 答案 D. os.path.isfile('theFile.txt')

選項2答案選擇區

- 答案 A. output(theFile.txt')
- 答案 B. print(file.get('theFile.txt'))
- 答案 C. print(file.read())
- 答案 D. print('theFile.txt')

選項3答案選擇區

- 答案 A. open('theFile.txt', 'a')
- 答案 B. open('theFile.txt', 'at')
- 答案 C. open('theFile.txt', 'w')
- 答案 D. open('theFile.txt', 'wt')

選項3答案選擇區

- 答案 A. Append
- 答案 B. file.add
- 答案 C. file.write
- 答案 D. write

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D
3. (1) A (2) B (3) C (4) D
4. (1) A (2) B (3) C (4) D

Answer: 1: D (4), 2: C (3), 3: A (1), 4: C (3)

Explanation:

- 在選項 1 中, `os.path.isfile('theFile.txt')` 用來檢查檔案是否存在。
- 在選項 2 中, `print(file.read())` 用來顯示檔案內容。
- 在選項 3 中, `open('theFile.txt', 'a')` 用來以附加模式打開檔案, 如果檔案不存在則會創建它。
- 在選項 4 中, `file.write('End of file')` 用來將文字寫入檔案。

38. 【CH04-8】你正在設計一個 Python 處理檔案的函式。

- 如果檔案不存在, 則返回 '檔案不存在'。

- 如果該檔案存在，則該函式返回第一行的內容。

請完成以下程式碼：

```
import os  
def get_file_message(file):  
    _(選項1)_  
    _(選項2)_  
    _(選項3)_  
    _(選項4)_  
    _(選項5)_
```

你應該如何安排這些程式碼片段的順序來完成函式？

請將適合的程式碼片段移動到回答區，並按正確的順序排列。

程式碼片段

- A.with open(file, 'r') as file:
- B.return 'File dose not exist'
- C.return file.readline()
- D.if os.path.isfile(file):
- E.else:

Quiz:

1. (1) A (2) B (3) C (4) D (5) E
2. (1) A (2) B (3) C (4) D (5) E
3. (1) A (2) B (3) C (4) D (5) E
4. (1) A (2) B (3) C (4) D (5) E
5. (1) A (2) B (3) C (4) D (5) E

Answer: 1: D (4), 2: A (1), 3: C (3), 4: E (5), 5: B (2)

Explanation:

- 在這個函式中，首先檢查檔案是否存在(if os.path.isfile(file):)，並以唯讀模式開啟檔案(with open(file, 'r') as file:)，並返回第一行的內容(return file.readline())。

- 如果不存在(else:)在則返回 '檔案不存在'(return 'File dose not exist')。

39. 【CH04-9】你設計一個 Python 應用程式，需要將資料讀寫到文字檔中。

如果檔案不存在，則必須新增它。

如果檔案已有內容，則將文字加到最後。

你應該使用哪個程式碼？

Quiz:

1. open('file\_data', 'a')
2. open('file\_data', 'w')
3. open('file\_data', 'rt')
4. open('file\_data', 'r')

Answer: 1

Explanation:

- 在這個情況下，使用 `open('file\_data', 'a')` 可以以附加模式(append mode)打開檔案，如果檔案不存在則會創建它，如果檔案已存在則會將新內容添加到檔案的末尾。

40. 【CH04-10】你正在設計一個 Python 程式來讀取學生資料的檔案，文件中包含了學生的班級、座號和姓名，下面顯示的是檔案中的資料範例：

'1A',1,'David'

'1A',2,'Mary'

程式碼必須符合以下的需求：

- 檔案的每一行都必須讀取和列印。

- 如果遇到空行，則必須忽略。

- 在完成所有行的讀取後，必須關閉檔案。

你創建了以下的程式碼：

```
students = open('students.txt', 'r')
```

```
eof = False
```

```
while eof == False:
```

```
    line = students.readline()
```

(程式碼1)

(程式碼2)

```
    print(line.strip())
```

```
else:
```

```
    print('End of file')
```

```
    eof = True
```

```
students.close()
```

在(程式碼1)及(程式碼2)中你應該編寫哪些程式碼？

Quiz:

1. (程式碼1).if line != "": (程式碼2).if line != '\n':
2. (程式碼1).if line = 'An': (程式碼2).if line != "":
3. (程式碼1).if line!= != 'An': (程式碼2).if line != None:
4. (程式碼1).if line != ":": (程式碼2).if line != "":

Answer: 1

Explanation:

- 在這個程式碼中，(程式碼1)和(程式碼2)的條件判斷都是用來檢查讀取的行是否為空行，如果不是空行則進行處理。

- 因此，正確的選擇是 ` (程式碼1).if line != "": (程式碼2).if line != '\n':`。

- 這樣可以確保只處理非空行，並且在讀取完所有行後關閉檔案。

#### 41. 【CH04-11】某間公司需要有人協助更新他們的檔案系統。

您必須建立一個執行下列動作的簡易檔案操作程式：

1. 查看檔案是否存在。

2. 如果檔案存在，則顯示其內容。

請選取正確的程式碼片段以完成程式碼。

```
#程式碼片段
```

```
import os
```

```
if _(選項 1)_
```

```
    file = open('myFile.txt')
```

```
    _(選項 2)_
```

```
    file.close()
```

選項1答案選擇區

- A. os.path.isfile('myFile.txt')
- B. os.path.exists('myFile.txt')
- C. os.path.isdir('myFile.txt')
- D. os.path.exists('myFile.txt')

選項2答案選擇區

- A. print(file.read())
- B. print(file.write())
- C. print(file.close())
- D. print(file.open())

Quiz:

- 1. (1) A (2) B (3) C (4) D
- 2. (1) A (2) B (3) C (4) D

Answer: 1: A (1), 2: A (1)

Explanation:

• 在選項 1 中，`os.path.isfile('myFile.txt')` 用來檢查檔案是否存在。

• 在選項 2 中，`print(file.read())` 用來顯示檔案的內容。  
這樣可以確保程式能夠正確地檢查檔案是否存在並顯示其內容。

42. 【CH04-12】請問下列陳述式有何功能？

data = input()

Quiz:

- 1. 允許使用者在主控台中輸入文字
- 2. 建立 HTML 輸入元素
- 3. 顯示電腦上的所有輸入周邊裝置
- 4. 顯示允許使用者輸入的訊息方塊

Answer: 1, 1

Explanation:

• 在這個程式碼中，`data = input()`

允許使用者在主控台中輸入文字，並將用來接收使用者的輸入，並將其存儲在變數 `data` 中。

43. 【CH05-1】你設計了一個程式要依學生的成績來顯示等級，它的規定如下：

Percentage range Letter grade  
90 through 100 A  
80 through 89 B  
70 through 79 C  
60 through 69 D  
0 through 59 F

如果使用者輸入 90，則輸出應該是 '你的成績為 A (甲等)'。

你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。

#程式碼片段

```
# Letter Grade Converter
grade = int(input('Enter a numeric grade:'))
_(選項 1)_  
letter_grade = 'A'  
_(選項 2)_  
letter_grade = 'B'  
_(選項 3)_
```

```
letter_grade = 'C'  
_(選項 4)_  
letter_grade = 'D'  
else:  
letter_grade = 'F'  
print('Your letter grade is :', letter_grade)
```

選項1答案選擇區

答案 A. if grade = 90:

答案 C. elif grade > 90:

答案 D. elif grade >= 90:

選項2答案選擇區

答案 A. if grade > 80:

答案 B. if grade >= 80:

答案 C. elif grade > 80:

答案 D. elif grade >= 80:

選項3答案選擇區

A. if grade > 70:

B. if grade >= 70:

C. elif grade > 70:

D. elif grade >= 70:

選項4答案選擇區

A. if grade > 60:

B. if grade >= 60:

C. elif grade > 60:

D. elif grade >= 60:

Quiz:

1. (1) A (2) B (3) C (4) D

2. (1) A (2) B (3) C (4) D

3. (1) A (2) B (3) C (4) D

4. (1) A (2) B (3) C (4) D

Answer: 1: B (2), 2: D (4), 3: D (4), 4: D (4)

Explanation:

- 

在這個程式碼中，使用了多個條件判斷來確定學生的成績等級，並根據不同的成績範圍給予相應的字母等級。

44. 【CH05-2】你要設計一款以使用者年齡進行電影分級的程式，必須符合以下要求：

- 任何 18 歲以上的人會顯示 '限制級' 的訊息。
- 任何 13 歲以上，但小於 18 歲的人都會顯示 '輔導級' 的訊息。
- 任何 12 歲或更年輕的人都會顯示 '普通級' 的訊息。

● 如果年齡未知，則會顯示 '未知' 的訊息。

你需要完成程式以符合要求，應該要如何完成這段程式碼？

```
#程式碼片段
def get_rating(age):
    rating = ""
    if_(選項 1)_
    elif_(選項 2)_
    elif_(選項 3)_
    else_(選項 4)_
return rating
```

選項答案選擇區

- 答案 A. age<13: rating='普通級'
- 答案 B. age<18: rating='輔導級'
- 答案 C. :rating='限制級'
- 答案 D. age is None: rating='未知'

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D
3. (1) A (2) B (3) C (4) D
4. (1) A (2) B (3) C (4) D

Answer: 1: D (4), 2: A (1), 3: B (2), 4: C (3)

Explanation:

- 在這個程式碼中，使用了多個條件判斷來確定使用者的年齡分級，並根據不同的年齡範圍給予相應的分級訊息。

45. 【CH05-3】你用學生的成績 (grade) 及排名 (rank) 編寫程式碼來決定最後成績：

```
if grade > 80 and rank >= 3:
    grade += 10
if grade > 70 and rank > 3:
    grade += 5
else:
    grade -= 5
當 grade = 76, rank = 3 時，執行程式碼的輸出值是？
```

Quiz:

1. 71
2. 76
3. 81
4. 86

Answer: 1

Explanation:

- 在這個程式碼中， grade = 76 , rank = 3 , 因此跳過兩個判斷來到 else: , 因此最終成績為 76 - 5 = 71。

46. 【CH05-4】你正在編寫一個函式來判別負數與非負數。這個函式必須符合以下要求：

- 如果 a 是負數，則回傳 'The result is a negative number'

- 如果 `a` 不是負數，則為非負數，再繼續判別。
- 如果 `a` 大於 0，則回傳 'The result is a positive number'

• 否則回傳 'The result is a zero'

你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。

```
def reResult(a):
    if _(選項 1)_:
        answer = 'The result is a negative number'
    _(選項 2)_:
        if _(選項 3)_:
            answer = 'The result is a positive number'
        _(選項 4)_:
            answer = 'The result is a zero'
    return answer
```

選項答案選擇區

答案 A. if `a < 0`:

答案 C. else:

答案 D. elif:

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D
3. (1) A (2) B (3) C (4) D
4. (1) A (2) B (3) C (4) D

Answer: 1: A (1), 2: C (3), 3: B (2), 4: C (3)

Explanation:

- 在這個程式碼中，首先檢查 `a` 是否小於 0，如果是，則回傳 'The result is a negative number'。
- 接著，如果 `a` 大於 0，則回傳 'The result is a positive number'。
- 如果 `a` 等於 0，則回傳 'The result is a zero'。

#### 47. 【CH05-5】你設計了一個電影票收費的函式，票價的規則如下：

- 5 歲以下 = 免費入場
- 5 歲及以上的學生 = 60 元
- 5 歲到 17 歲但不是學生 = 120 元

• 17 歲以上但不是學生 = 180 元

你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。

```
def ticket_fee(age, school):
```

```
    fee = 0
    _(選項 1)_:
        fee = 60
    _(選項 2)_:
        _(選項 3)_:
```

```
fee = 120
else:
    fee = 180
return fee
```

選項答案選擇區

答案 A. if age >= 5 and school:

答案 B. elif age >= 5 and not school:

答案 C. if age <= 17:

Quiz:

1. (1) A (2) B (3) C
2. (1) A (2) B (3) C
3. (1) A (2) B (3) C

Answer: 1: A (1), 2: B (2), 3: C (3)

Explanation:

- 在這個程式碼中，首先檢查 `age` 是否大於等於 5 且 `school` 為真，如果是，則票價為 60 元。
- 接著，如果 `age` 大於等於 5 且 `school` 為假，則進一步檢查 `age` 是否小於等於 17，如果是，則票價為 120 元；否則票價為 180

48. 【CH05-6】你設計一個 Python 程式來檢查使用者輸入的數字是 1 位數、2 位數還是 2 位數以上，其中規定輸入的值必須是正整數。你要如何完成這段程式碼？

```
num = int(input('Enter a number with 1 or 2 digits:'))
digits = '0'
if num > 0:
    _(選項 1)_  
    digits = '1'  
    _(選項 2)_  
    digits = '2'  
    _(選項 3)_  
    digits = '>2'  
    print(digits + ' digits.')
elif num == 0:  
    print('The number is 0')
else:  
    print('The number is not a positive number')
```

選項答案選擇區

答案 A. if num < 10:

答案 B. if num < 100:

答案 C. elif num < 100:

答案 D. else:

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D
3. (1) A (2) B (3) C (4) D

Answer: 1: A (1), 2: C (3), 3: D (4)

Explanation:

- 在這個程式碼中，使用了多個條件判斷來檢查使用者輸入的數字是 1 位數、2 位數還是 2

位數以上，並根據不同的數字範圍給予相應的提示訊息。

49. 【CH05-7】你在設計一個 Python 程式遊戲，讓參加的人從 1 到 100 之間猜一個數字，最多有三次機會。

程式碼如下：

```
from random import randint
target = randint(1,100)
chance = 1
print('Guess an integer from 1 to 10. You will have 3 chances.')
_(選項 1)_  
guess = int(input('Guess an integer:'))
if guess > target:
    print('Guess is too high')
elif guess < target:
    print('Guess is too low')
else:
    print('Guess is just right')
_(選項 2)_  
(選項 3)_
```

程式可以讓使用者猜三次，如果猜出正確數字即停止程式。你要如何完成行號 5、13 及 14 的程式碼？

請將回答區項目的選到正確的程式碼片段

選項答案選擇區

答案 A. while chance < 3:

答案 B. break

答案 C. pass

答案 D. chance + 1

答案 E. while chance < 3:

答案 F. chance = 2

Quiz:

1. (1) A (2) B (3) C (4) D (5) E (6) F
2. (1) A (2) B (3) C (4) D (5) E (6) F
3. (1) A (2) B (3) C (4) D (5) E (6) F

Answer: 1: A (1), 2: C (3), 3: E (5)

Explanation:

- 在這個程式碼中，使用了 while 迴圈來控制猜測的次數，最多允許三次猜測。
- 如果猜測正確，則使用 break 來終止迴圈。
- 如果猜測錯誤，則使用 chance + 1 來增加猜測次數，並繼續下一次猜測。

50. 【CH05-8】在以下的程式碼中，要加入哪些程式碼片段讓 print 語法可以正確執行？

你要如何完成程式碼讓 print 語法是正確的？請在回答區中選擇適當的程式碼片段。

```
aList = 1,2,3]
```

```
bList = ['a', 'b', 'c']
```

\_(選項 1)\_

```
print('aList is equal to bList')
```

\_(選項 2)\_

```
print('aList is not equal to bList')
```

選項答案選擇區

- A. if aList == bList:
- B. if alist == blist
- C. else:
- D. else

Quiz:

- 1. (1) A (2) B (3) C (4) D
- 2. (1) A (2) B (3) C (4) D

Answer: 1: A (1), 2: C (3)

Explanation:

- 首先檢查 `aList` 和 `bList` 是否相等，如果相等則印出 'aList is equal to bList'。
- 如果不相等，則印出 'aList is not equal to bList'。

## 51. 【CH05-9】你設計了一個 Python

程式來檢查輸入英文姓名的大小寫，請在下方選擇四個程式碼到回答區中進行順序排列。

第 1 段程式為 \_ (選項 1)

第 2 段程式為 \_ (選項 2)

第 3 段程式為 \_ (選項 3)

第 4 段程式為 \_ (選項 4)

選項答案選擇區

答案 A.

```
name = input('Enter your English name:')
```

答案 B.

```
elif name.lower() == name:  
    print(name, 'is all lower case.')
```

答案 C.

```
else:  
    print(name, 'is upper case.')
```

答案 D.

```
else:  
    print(name, 'is mixed case.')
```

答案 E.

```
if name.upper() == name:  
    print(name, 'is all upper case.')
```

答案 F.

```
else:  
    print(name, 'is lower case.')
```

Quiz:

- 1. (1) A (2) B (3) C (4) D (5) E (6) F
- 2. (1) A (2) B (3) C (4) D (5) E (6) F
- 3. (1) A (2) B (3) C (4) D (5) E (6) F
- 4. (1) A (2) B (3) C (4) D (5) E (6) F

Answer: 1: A (1), 2: E (5), 3: B (2), 4: D (4)

Explanation:

- 首先檢查 `name` 是否為全大寫，如果是，則印出 'is all upper case.'。

- 接著，如果 `name` 為全小寫，則印出 'is all lower case.'。
- 如果 `name` 同時包含大寫和小寫字母，則印出 'is mixed case.'。

52. 【CH05-10】公司決定要幫所有年薪不到 50 萬的員工調升基本工資 5%，並給予獎金 1 萬元，以下是計算公式：

- 新工資 = 目前工資 X 105% + \$10000 獎金。

• 程式會將每個人調整後的年薪資料存入 salaryList 清單中。

你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。

#清單中的每個人的工資都根據增加而更新。

#年薪50萬元以上的員工將不會得到加薪。

# salaryList 是由員工資料庫中取得，程式碼不會顯示。

(選項 1)

if salaryList[index] >= 500000:

(選項 2)

salaryList[index] = int((salaryList[index] \* 1.05) + 10000)

選項 1 答案選擇區

答案 A. for index in range(len(salaryList)+1):

答案 B. for index in range(len(salaryList)-1):

答案 C. for index in range(len(salaryList)):

答案 D. for index in salaryList:

選項 2 答案選擇區

A. exit()

B. continue

C. break

D. end

Quiz:

1. (1) A (2) B (3) C (4) D

2. (1) A (2) B (3) C (4) D

Answer: 1: C (3), 2: B (2)

Explanation:

- 使用了 for 迴圈來遍歷 `salaryList` 中的每個員工的工資。
- 如果員工的工資大於等於 50 萬，則跳過該員工的工資調整。
- 否則，根據公式計算新的工資並更新 `salaryList`。

53. 【CH05-11】你設計了一個函式計算並顯示從 2 到 9 的所有乘法組合的九九乘法表。

你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。

# Displays times tables 2 - 9

def times\_tables():

(選項 1)

(選項 2)

```
print(row * col, end=' ')
print()

# main
times_tables()
```

選項 1 答案選擇區

答案 A. for col in range(9):

答案 B. for col in range(2, 10):

答案 C. for col in range(2, 9, 1):

答案 D. for col in range(10):

選項 2 答案選擇區

答案 A. for row in range(9):

答案 B. for row in range(2, 9, 1):

答案 C. for row in range(2, 10):

答案 D. for row in range(10):

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D

**Answer: 1: B (2), 2: C (3)**

Explanation:

- 使用了兩個 for 迴圈來遍歷從 2 到 9 的所有乘法組合。
- 第一個迴圈遍歷列(col)，第二個迴圈遍歷行(row)。
- range(2, 10) 為產生從 2 到 9 的整數序列（注意 Python 的 range 不包含結束值）

54. 【CH05-12】你設計了一個 Python 程式來顯示 2 到 100 中的所有質數，請將下方的程式碼片段排列到回答區的正確位置。

第 1 段程式為 \_(選項 1)\_

第 2 段程式為 \_(選項 2)\_

第 3 段程式為 \_(選項 3)\_

```
if is_prime:  
    print(n)
```

第 4 段程式為 \_(選項 4)\_

選項答案選擇區

答案 A.

n = 2

is\_prime = True

while n <= 100:

答案 B.

n = 2

while n <= 100:

```
    is_prime = True
```

答案 C.

```
        break
```

答案 D.

Continue

答案 E.

```
n += 1
```

答案 F.

```
for i in range(2, n):
    if n / i == 0:
        is_prime = False
```

答案 G.

```
for i in range(2, n):
    if n % i == 0:
        is_prime = False
```

Quiz:

1. (1) A (2) B (3) C (4) D (5) E (6) F (7) G
2. (1) A (2) B (3) C (4) D (5) E (6) F (7) G
3. (1) A (2) B (3) C (4) D (5) E (6) F (7) G
4. (1) A (2) B (3) C (4) D (5) E (6) F (7) G

Answer: 1: B (2), 2: G (7), 3: C (3), 4: E (5)

Explanation:

- 使用了 while 迴圈來遍歷從 2 到 100 的所有整數。
- 對於每個整數 `n`，使用 for 迴圈來檢查是否為質數，如果 `n` 可以被任何小於 `n` 的整數整除，則不是質數。
- 如果是質數，則印出該質數。

55. 【CH05-13】你用 Python 設計了一個比大小函式，必須符合以下的需求：

- 這個函式有二個參數：一個整數清單和一個整數作為參數。
- 該函式必須在清單中以整數進行比較。
- 如果在清單中的整數有一個比參數整數大，則該函式必須列印一則消息，  
指示已經找到的整數比參數整數大，然後停止查看清單的動作。
- 如果找不到比參數整數大的數，則該函式必須列印一則消息，  
指示在清單中找不到比參數整數大的數。

你應該如何安排這些程式碼片段的順序來開發解決方案？

請將適合的程式碼片段從程式碼片段清單移動到回答區，並按正確的順序排列。

第 1 段程式為\_(選項 1)\_

第 2 段程式為\_(選項 2)\_

第 3 段程式為\_(選項 3)\_

第 4 段程式為\_(選項 4)\_

第 5 段程式為\_(選項 5)\_

選項答案選擇區

答案 A.

```
for i in range(len(nums)):
```

答案 B.

```
    if nums[i] > num:
```

```
    print('A value greater than {0} is found in the list of {1}'.format(num, nums))
答案 C.
else:
    print('A value greater than {0} cannot be found in the list of {1}'.format(num, nums))
答案 D.
break
答案 E.
def search(nums, num):
```

Quiz:

1. (1) A (2) B (3) C (4) D (5) E
2. (1) A (2) B (3) C (4) D (5) E
3. (1) A (2) B (3) C (4) D (5) E
4. (1) A (2) B (3) C (4) D (5) E
5. (1) A (2) B (3) C (4) D (5) E

Answer: 1: E (5), 2: A (1), 3: B (2), 4: D (4), 5: C (3)

Explanation:

- 首先定義了一個函式 `search(nums, num)`，該函式接受一個整數清單和一個整數作為參數。
- 接著使用 for 迴圈遍歷清單中的每個整數，檢查是否有比參數整數大的數字。
- 如果找到比參數整數大的數字，則印出相應的消息並停止迴圈。
- 如果沒有找到比參數整數大的數字，則印出另一條消息。

## 56. 【CH05-14】你正在設計一個 Python 程式驗證產品編號。

- 產品編號的格式必須為 dd-dddd，並且只包含數字和破折號。
  - 如果格式正確則程式必須列印 True，如果格式不正確，則列印 False。
- 你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。

\_(選項 1)\_  
parts = ''  
\_(選項 2)\_  
\_(選項 3)\_  
product\_no = input('Enter product number (dd-dddd): ')  
parts = product\_no.split('-')  
if len(parts) == 2:  
 if len(parts[0]) == 2 and len(parts[1]) == 4:  
 if parts[0].isdigit() and parts[1].isdigit():  
 \_(選項 4)\_  
print(valid)

選項 1 答案選擇區

- A. product\_no = ''  
B. product\_no = 'sentinel'

選項 2 答案選擇區

- A. while product\_no != "":  
B. while product\_no != 'sentinel':

選項 3 答案選擇區

- A. valid = False
- B. valid = True

選項 4 答案選擇區

- A. valid = False
- B. valid = True

Quiz:

- 1. (1) A (2) B
- 2. (1) A (2) B
- 3. (1) A (2) B
- 4. (1) A (2) B

Answer: 1: B (2), 2: A (1), 3: A (1), 4: B (2)

Explanation:

- 首先定義了一個空的 `product\_no` 變數，然後使用 while 迴圈來持續要求使用者輸入產品編號，直到輸入 'sentinel' 為止。
- 接著將輸入的產品編號分割成兩部分，並檢查其格式是否符合要求。
- 如果格式正確，則印出 True；否則印出 False。

57. 【CH05-15】你正在設計 Python 應用程式。該程式將逐一查看數字清單，並在找到 4 時進行跳脫的動作。

你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。

```
nList = [0,1,2,3,4,5,6,6,7,8,9]
index = 0
_(選項 1)_ index < 10:
    print(nList[index])
    if nList[index] == 4:
        _(選項 2)_
    else:
        _(選項 3)_
```

選項 1 答案選擇區

- A. while
- B. for
- C. if
- D. break

選項 2 答案選擇區

- A. while
- B. for
- C. if
- D. break

選項 3 答案選擇區

- A. continue
- B. break

- C. index += 1
- D. index = 1

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D
3. (1) A (2) B (3) C (4) D

Answer: 1: A (1), 2: D (4), 3: C (3)

Explanation:

- 使用了 while 迴圈來遍歷數字清單，並在找到 4 時使用 break 跳出迴圈。

## 58. 【CH05-16】您正在撰寫程式碼以使用星星建立 E 字形。

- 您需要列印五行程式碼。
- 其中第一行第三行和第五行各有四個星星。
- 而第二行和第四行各有一個星。

如下所示：請在下列位置中選擇正確的數字以完成程式碼。

```
result_str = ""  
for row in range(1, _(選項 1)_):  
    for column in range(1, _(選項 2)_):  
        if row == 1 or row == 3 or row == 5:  
            result_str = result_str + '*'  
        elif column == 1:  
            result_str = result_str + '*'  
        result_str += '\n'  
print(result_str)
```

選項答案選擇區

- A. 3
- B. 4
- C. 5
- D. 6

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D

Answer: 1: D (4), 2: C (3)

Explanation:

- 使用了兩個巢狀的 for 迴圈來列印星星，外層迴圈控制行數，內層迴圈控制列印星星的數量。

## 59. 【CH05-17】您正在撰寫程式碼來根據下列準則傳回字母成績：

- 變數 grade 會將數字成績儲存成 0 到 100 之間的整數。
- 90 到 100 的成績 = A。
- 80 到 89 的成績 = B。
- 70 到 79 的成績 = C。

- 其他所有成績都不及格。

請選取正確的程式碼片段以完成程式碼。

#程式碼片段

```
grade = int(input('Please enter your grade: '))
_(選項 1)_ grade = 0:
_(選項 2)_ grade >= 90:
    print('Your grade is A.')
_(選項 3)_ grade >= 80:
    print('Your grade is B.')
_(選項 4)_ grade < 69:
    print('Your grade is C.')
_(選項 5)_ 
    print('Your grade is failing.')
_(選項 6)_ 
    print('Invalid grade entered.'
```

選項答案選擇區

A. if B. else: C. elif D. else if

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D
3. (1) A (2) B (3) C (4) D
4. (1) A (2) B (3) C (4) D
5. (1) A (2) B (3) C (4) D
6. (1) A (2) B (3) C (4) D

Answer: 1: A (1), 2: A (1), 3: C (3), 4: C (3), 5: B (2), 6: B (2)

Explanation:

- 使用了 if、elif 和 else 來根據成績的範圍來判斷並印出對應的字母成績。

60. 【CH05-18】出版社想要檢查出版物中特定字母的數量，你需要設計一個函式以符合要求。你要如何完成這段程式碼？

#程式碼片段

```
# Function accepts list of words from a file
# word_list = ['apple', 'banana', 'peach', 'grape']
# and letter to search for.
# Returns count of a particular letter in that list.
def count_letter(letter, word_list):
    count = 0
    for_(選項 1)_ 
        if _(選項 2)_ 
            count += 1
    return count

# word_list is populated from a file. Code not shown. here we use a sample list.
word_list = ['apple', 'banana', 'peach', 'grape']
letter = input('which letter would you like to count: ')
letter_count = count_letter(letter, word_list)
print('There are:', letter_count, 'instances of', letter)
```

選項 1 答案選擇區

- A. letter\_list in word:
- B. word in letter\_list:
- C. word == letter\_list:
- D. word is letter\_list:

選項 2 答案選擇區

- A. word is letter:
- B. letter is word:
- C. word in letter:
- D. letter in word:

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D

Answer: 1: B (2), 2: D (4)

Explanation:

- 使用了 for 迴圈來遍歷字詞列表，並在找到特定字母時增加計數。

61. 【CH05-19】學校要求你對一些導致成績單出現問題的程式碼除錯。在程式中已經宣告了下列變數：

#程式碼片段

```
scores = [80,90,75,85]
count = 0
sum = 0
# 在以下的程式碼有二個錯誤：
for index in range(_(選項 1)_):
    count += 1
    sum += scores[index]
average = _(選項 2)_
print('The total scords is:', sum)
print('The average scords is:', average)
```

你應使用哪些程式碼來修復錯誤？回答時，請在回答區中選擇適當的程式碼片段。

選項 1 答案選擇區

- A. size(scores)
- B. size(scores) - 1
- C. len(scores) + 1
- D. len(scores)

選項 2 答案選擇區

- A. sum / count
- B. sum\*\*count
- C. sum \* count
- D. sum // count

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D

Answer: 1: C (3), 2: A (1)

### Explanation:

- 使用了 for 迴圈來遍歷分數列表，並計算總分和平均分。

### 62. 【CH05-20】您正在撰寫符合下列需求的程式碼：

- 允許使用者不斷輸入字詞。

- 輸出每個字詞的字元數。

請選取正確的選項以完成程式碼。

```
#程式碼片段
x = 'Hello World'
_(選項 1)_ x != 'QUIT':
    num = 0
    _(選項 2)_ char _(選項 3)_ x:
        num += 1
    print(num)
x = input('Enter a new word or QUIT to exit: ')
```

選項 1 答案選擇區

- A. while B. for C. if D. in

### Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D
3. (1) A (2) B (3) C (4) D

Answer: 1: A (1), 2: B (2), 3: D (4)

### Explanation:

- 使用了 while 迴圈來持續接收使用者輸入，並使用 for 迴圈來計算字詞的字元數。

### 63. 【CH05-21】你在測試以下程式碼時發現錯誤。其中包含的行號只是做為參考。

```
numList = [0,1,2,3,4,5,6,7,8,9]
```

```
i = 0
```

```
while (i < 10)
```

```
    print(numList[i])
```

```
if numList[i] = 6
    break
else:
    i += 1
```

你需要更正第 3 行和 6 行中的程式碼。你要如何更正程式碼？

選項 1，在第 3 行中應使用哪個程式碼片段？

- A. while (i < 10):
- B. while [i < 10]
- C. while (i < 5):
- D. while [i < 5]

選項 2，在第 6 行中應使用哪個程式碼片段？

- A. if numList[i] == 6
- B. if numList[i] = 6:
- C. if numList(i) = 6:
- D. if numList(i) ! 6

Quiz:

- 1. (1) A (2) B (3) C (4) D
- 2. (1) A (2) B (3) C (4) D

Answer: 1: A (1), 2: B (2)

Explanation:

- 首先需要在第 3 行的 while 迴圈後面加上冒號 `:`, 以正確結束條件語句。
- 在第 6 行的 if 條件中，需要使用雙等號 `==` 來比較是否等於 6，並且也需要加上冒號 `:`。

64. 【CH06-1】公司正將過去的進銷存程式碼轉移到 Python，以下哪個是正確的語法？

選項 A.

```
// Return the current revenue
def get_saletotal():
    return saletotal
```

選項 B.

```
/* Return the current revenue */
def get_saletotal():
    return saletotal
```

選項 C.

```
'Return the current revenue
def get_saletotal():
    return saletotal
```

選項 D.

```
# Return the current revenue
def get_saletotal():
    return saletotal
```

Quiz:

- 1. A
- 2. B
- 3. C
- 4. D

Answer: 4

Explanation:

- 使用了正確的註解語法 (#) 來描述函式的功能。

65. 【CH06-2】你設計一個函式，使用 Python 計算矩形的面積。在函式中有加入注釋，程式碼如下：

```
# The area_rectangle function calculates the area of rectangle
```

```

# x is the length
# y is the width
# return the value of x*y
def area_rectangle(x, y):
    comment = '# Retrun the value'
    return x*y # x*y

```

針對下列每個敘述，如果是正確的就選擇 Yes，否則請選擇 No。

選項 A. 第 1 到 4 行在語法檢查時將被忽略。

選項 B. 第 2 和第 3 行中的井字符串 (#) 非必填的。

選項 C. 第 6 行中的字串將被解釋為注釋。

選項 D. 第 7 行包含內嵌注釋。

Quiz:

1. (1) Yes (2) No
2. (1) Yes (2) No
3. (1) Yes (2) No
4. (1) Yes (2) No

Answer: 1: Yes (1), 2: No (2), 3: No (2), 4: Yes (1)

Explanation:

- 第一到第四行的註解將在語法檢查時被忽略。
- 第二和第三行中的井字符串 (#) 是必要的，因為它們用來標記註解。
- 第六行中的字串 `comment` 並不會被解釋為註解，因為它是被引號包圍的字串。
- 第七行中的內嵌註解是正確的，因為它使用了井字符串 (#)。

66. 【CH06-3】一家運動器材公司正在設計一個程式用來記錄客戶跑步時的距離，你設計以下的 Python 程式碼：

```

?
name = input('What is your name?')
return name
?
calories = kms * calories_per_km
return calories
distance = int(input('How many kilometers did you run this week?'))
burn_rate = 80
runner = get_name()
calories_burned = calc_calories(distance, burn_rate)
print(runner, ', you burned about ', calories_burned, 'calories.')

```

在程式中必須要定義二個必要的函式。你將在第 1 及第 4 行中使用哪些程式碼片段？

Quiz:

1. 第 1 行 = def get\_name():
2. 第 1 行 = def get\_name(runner):
3. 第 1 行 = def get\_name(name):
4. 第 4 行 = def calc\_calories():
5. 第 4 行 = def calc\_calories(kms, burn\_rate):
6. 第 4 行 = def calc\_calories(kms, calories\_per\_km):

Answer: 1, 6

### Explanation:

- 使用了正確的函式定義方式，並且參數名稱也符合函式的功能。

67. 【CH06-4】你正在設計一個線上遊戲記分的 Python 應用程式。需要符合以下條件的函式：

- 函式名為 calc\_score
- 函式接收二個參數：目前分數和一個值
- 函式將值增加到目前分數

• 函式返回新分數  
你要如何完成程式碼？請在回答區中還撲適當的程式碼片段。

#程式碼片段

\_ (選項 1) \_ \_ (選項 2)\_  
current += value  
\_ (選項 3) \_

選項 1 答案選擇區

- A. calc\_score
- B. def calc\_score
- C. return calc\_score

選項 2 答案選擇區

- A. (current, value):
- B. ():
- C. (current, value)
- D. ()

選項 3 答案選擇區

- A. pass current
- B. return current
- C. return
- D. pass

Quiz:

1. (1) A (2) B (3) C
2. (1) A (2) B (3) C (4) D
3. (1) A (2) B (3) C (4) D

Answer: 1: B (2), 2: A (1), 3: B (2)

Explanation:

- 首先定義了一個函式 `calc\_score`，該函式接受兩個參數：目前分數和一個值。
- 接著將值增加到目前分數，並返回新的分數。

68. 【CH06-5】你正設計一個函式用來增加遊戲中的玩家得分。該函式有以下的要求：

- 如果沒有為變數 points 指定值，則 points 等於 1。

- 如果變數 plus 是 True , 那麼 points 必須加倍。

程式碼如下：

```
def add_score(score, plus, points):
    if plus == True:
        points = points * 2
    score = score + points
    return score
points = 5
score = 10
new_score = add_score(score, True, points)
```

針對下列每個敘述，如果是正確的就選擇 Yes，否則請選擇 No。

選項答案選擇區

選項 1.為了符合要求必須將 1 行更改為以下內容：

```
def add_score(score, plus, points = 1):
```

選項 2.一旦使用預值定義了任何參數，其右側任何參數也必須使用默認值進行定義。

選項 3.如果只用兩個參數呼叫函式，則第三個參數的值將為 None

選項 4.第 3 行的結果會改變在第 6 行中變數 points 的值。

Quiz:

1. (1) Yes (2) No
2. (1) Yes (2) No
3. (1) Yes (2) No
4. (1) Yes (2) No

Answer: 1: Yes (1), 2: Yes (1), 3: No (2), 4: No (2)

Explanation:

- 為了符合要求，必須將函式的參數 `points` 設定為預設值 1。
- 當使用預設值定義任何參數時，其右側的參數也必須使用預設值進行定義。
- 如果只用兩個參數呼叫函式，則第三個參數的值將為 None。
- 函式內部的計算不會改變在第 6 行中變數 `points` 的值。

## 69. 【CH06-6】你設計了以下的函式來計算薪水：

#程式碼片段

```
def getpay(hours=40, rate=25, qty=0, qtyrate=0, salary=0):
    overtime = 0
    if qty > 0:
        return qty * qtyrate
    if salary > 0:
        return salary
    if hours > 40:
        overtime = (hours - 40) * (1.5 * rate)
        return overtime + (40 * rate)
    else:
        return hours * rate
```

針對下列每個敘述，如果是正確的就選擇 Yes，否則請選擇 No。

#### 選項答案選擇區

- 選項 1. 呼叫 getpay() 函式會發生錯誤。  
選項 2. getpay(salary = 5000) 不會回傳任何值。  
選項 3. getpay(qty = 500, qtyrate=4) 回傳值為 2000

Quiz:

1. (1) Yes (2) No
2. (1) Yes (2) No
3. (1) Yes (2) No

Answer: 1: No (2), 2: No (2), 3: Yes (1)

Explanation:

- 呼叫 `getpay()` 函式不會發生錯誤，因為所有的參數都有預設值。
- `getpay(salary = 5000)` 會回傳 5000，因為它符合 `if salary > 0:` 的條件。
- `getpay(qty = 500, qtyrate=4)` 會回傳 2000，因為它符合 `if qty > 0:` 的條件。

#### 70. 【CH06-7】請從選取正確的選項以回答有關記錄 Python 程式碼的問題：

##### 選項 1 答案選擇區

請問哪些字元代表單行文件字串的開頭和結尾？

- A. 單引號
- B. 雙引號
- C. 二個雙引號
- D. 三個雙引號

##### 選項 2 答案選擇區

在記錄函式時，文件字串的標準位置在哪裡？

- A. 在函式宣告區塊前
- B. 緊接在函式標頭後面
- C. 在函式宣告區塊後
- D. 在程式的最後

##### 選項 3 答案選擇區

請閱下列函式：

```
def cube (n):  
    """ Returns the cube of number n """  
    return n*n*n
```

請問哪個命令可列印文件字串？

- A. print(\_doc\_)
- B. print(cube(doc))
- C. print(cube.\_doc\_)
- D. print(cube(docstring))

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D
3. (1) A (2) B (3) C (4) D

Answer: 1: A (1), 2: A (1), 3: B (2)

Explanation:

- 選項 1 和選項 2 的正確答案都是 A，因為單行文件字串可以用單引號或雙引號來表示。
- 選項 3 的正確答案是 C，因為可以使用 `cube.\_doc\_` 來列印函式的文件字串。
- ITS : A A B 而 ChatGPT : D B C

### 71. 【CH06-8】請檢閱下列程式。

```
def petstore(category, species, breed='none'):
    """Display information about a pet."""
    print(f'You have selected an animal from the {category} category.')
    if breed == 'none':
        print(f'The {category} you selected is a {species}')
    else:
        print(f'The {category} you selected is a {species} {breed}')
    print(f'The {category} would make a great pet!')

category = input('What animal category are you interested in?')
species = input('What species are they from (canine, feline, Scarlet Macaw, Blue and Gold Macaw)?')
if category == 'dog' or category == 'cat':
    breed = input('What breed are you interested in?')
    petstore(category, species, breed)
else:
    petstore(category, species)
    petstore(breed='Maltese', species='Canine', category='dog')
    petstore('bird', species='Scarlet Macaw')
```

針對下列每個敘述，如果是正確的就選擇 Yes，否則請選擇 No。

選項答案選擇區

選項 1. 此函式會傳回一個值。

選項 2. 第 14 行和第 17 行的函式呼叫無效。

選項 3. 第 16 行和第 18 行的函式呼叫會產生錯誤。

Quiz:

1. (1) Yes (2) No
2. (1) Yes (2) No
3. (1) Yes (2) No

Answer: 1: No (2), 2: Yes (1), 3: No (2)

Explanation:

- 函式 `petstore` 沒有寫 `return` 因此並不會傳回任何值，選項 1 是錯誤的。
- 第 14 行和第 17 行的函式呼叫是有效的，因為它們符合函式的參數要求。
- 第 16 行和第 18 行的函式呼叫會產生錯誤，因為它們使用了不正確的參數名稱或順序。
- 第14和17行的函式呼叫，應該是有效呼叫，ITS答案 Yes；ChatGPT: No #有效呼叫

### 72. 【CH06-9】學生們正在參加學校活動之夜。下列函式會告訴學生該前往何處參加活動：

```
def roomAssignment(student, year):
    """Assign rooms to students"""
    if year == 1:
```

```
print(f'{student.title()}, please report to room 115')
elif year == 2:
    print(f'{student.title()}, please report to room 210')
elif year == 3:
    print(f'{student.title()}, please report to room 320')
elif year == 4:
    print(f'{student.title()}, please report to room 405')
elif year == 5:
    print(f'{student.title()}, please report to room 515')
else:
    print(f'{student.title()}, please report to room 625')
```

請問哪兩個函式呼叫是正確的？請選擇 2 個答案。

選項答案選擇區

選項 1.

```
name = input('What is your name?')
grade = 0
while grade not in (1,2,3,4,5,6):
    grade = int(input('What grade are you in (1-6)?'))
    roomAssignment(name, year=grade)
```

選項 2.

```
name = input('What is your name?')
grade = 0
while grade not in (1,2,3,4,5,6):
    grade = int(input('What grade are you in (1-6)?'))
    roomAssignment(student, year)
```

選項 3.

```
roomAssignment('Sherlock Sassafrass',4)
```

選項 4.

```
roomAssignment(year=6, name='Minnie George')
```

Quiz:

1. (1) 正確 (2) 錯誤
2. (1) 正確 (2) 錯誤
3. (1) 正確 (2) 錯誤
4. (1) 正確 (2) 錯誤

Answer: 1: 正確 (1), 2: 錯誤 (2), 3: 正確 (1), 4: 錯誤 (2)

Explanation:

- 選項 1 的函式呼叫是正確的。
- 選項 2 的函式呼是錯誤的， student 參數 應為 name 參數。
- 選項 3 的函式呼叫也是正確的，因為它直接傳入了學生姓名和年級。
- 選項 4 的函式呼叫是錯誤的，因為它使用了不正確的順序及參數名稱。

73. 【CH06-10】您為公司開發了一個 Python 應用程式，您想要在自己的程式碼中加入附註，好讓其他團隊成員能夠了解。請問您應該採取下列哪一項做法？

Quiz:

1. 在任何程式碼片段的 `<! — 和 —>` 之間放置附註。
2. 在任何一行的 `#` 後面放置附註。
3. 在任何一行的 `//` 後面放置附註。
4. 在任何程式碼片段的 `/**/` 之間置附註

Answer: 2

Explanation:

- 在 Python 中，單行註解是以 `#` 開頭的，因此選擇 2 是正確的。
- 選擇 1 和選擇 4 是錯誤的，因為這些註解語法不是 Python 所使用的。
- 選擇 3 是錯誤的，因為 `//` 是 JavaScript 和其他語言的註解語法，而不是 Python 的。

74. 【CH07-1】在程式中要使用 `datetime` 模組中 `datetime` 函式，再設定為替代名稱，在導入時應該使用哪個程式碼片段？

Quiz:

1. `from datetime as dt`
2. `from datetime import datetime as dt`
3. `import datetime from datetime as dt`
4. `import datetime.datetime as dt`

Answer: 2

Explanation:

- 在 Python 中，使用 `'from datetime import datetime as dt'` 可以將 `'datetime'` 模組中的 `'datetime'` 函式導入並設定為替代名稱 `'dt'`。
- 選擇 1 是錯誤的，因為它沒有指定要導入的函式。
- 選擇 3 和選擇 4 的語法都是錯誤的。

75. 【CH07-2】你設計一個讀取檔案後將檔案中的每一行列印出來的函式。程式碼如下：

```
def print_file(filename):
    line = None
    if os.path.isfile(filename):
        data = open(filename, 'r')
        for line in data:
            print(line)
```

當你執行該程式時，你會收到第 3 行上的錯誤。導致錯誤的原因是什麼？

Quiz:

1. 你需導入 `os` 模組。
2. `path` 方法並不存在 `os` 模組中。
3. `path` 物件中不存在 `isfile` 方法。
4. `isfile` 方法不接受一個參數。

Answer: 1

Explanation:

- 錯誤的原因是因為沒有導入 `'os'` 模組。

- 在使用 `os.path.isfile(filename)` 之前，必須先導入 `os` 模組。
- 選擇 2、3 和 4 都是錯誤的，因為 `os.path.isfile` 方法確實存在並且接受一個參數。

76. 【CH07-3】設計程式碼用來生成的隨機整數最小值為 11，最大值為 20。你應該使用哪兩種函式？

Quiz:

1. random.randrange(11, 21, 1)
2. random.randrange(11, 20, 1)
3. random.randint(11, 20)
4. random.randint(11, 21)

Answer: 1, 3

Explanation:

- 選擇 1 和選擇 3 是正確的。
- 選擇 2 是錯誤的，因為 `random.randrange(11, 20, 1)` 的最大值是 20，不包括 20。
- 選擇 4 是錯誤的，因為 `random.randint(11, 21)` 的最大值是 21，包括 21。

77. 【CH07-4】你設計程式碼用來生成的隨機整數，最小值是 0，最大值是 10。你應該使用哪個語法？

Quiz:

1. random.random()
2. random.randrange(0.0, 1.0)
3. random.randrange()
4. random.randint(0, 10)

Answer: 4

Explanation:

- 選擇 4 是正確的，因為 `random.randint(0, 10)` 可以生成 0 到 10 之間的隨機整數，包括 0 和 10。
- 選擇 1 是錯誤的，因為 `random.random()` 生成的是 0 到 1 之間的浮點數。
- 選擇 2 是錯誤的，因為 `random.randrange(0.0, 1.0)` 生成的是 0 到 1 之間的浮點數。
- 選擇 3 是錯誤的，因為 `random.randrange()` 需要指定範圍。

78. 【CH07-5】你設計程式碼來產生一個隨機數數來符合以下要求：

- 數字是 2 的倍數。
- 最低的數字是 2。
- 最高的數字是 50。

哪兩個程式碼片段將符合要求？

選項答案選擇區

選項 1.

```
from random import randint  
print(randint(1,25) * 2)
```

選項 2.

```
from random import randint  
print(randint(1,50))
```

選項 3.

```
from random import randrange  
print(randrange(2,50,2))
```

選項 4.

```
from random import randrange  
print(randrange(2,50,1))
```

Quiz:

1. 選項 1 程式碼片段符合要求
2. 選項 2 程式碼片段符合要求
3. 選項 3 程式碼片段符合要求
4. 選項 4 程式碼片段符合要求

Answer: 1, 3

Explanation:

- 選擇 1 和選擇 3 是正確的，因為它們都能生成符合條件的隨機數字。
- 選擇 2 是錯誤的，因為 `randint(1,50)` 可能生成奇數。
- 選擇 4 是錯誤的，因為 `randrange(2,50,1)` 會生成所有的整數，包括奇數。
- 選擇 3 似乎不對，除非改成 print(randrange(2, 51, 2))

79. 【CH07-6】你正在設計一個處理數字的函式。該函式具有以下要求：

- 將浮點數傳遞到函式中
  - 函式必須取浮點數的絕對值
  - 函式必須無條件進位到整數
- 你應該使用哪兩個數學函式？

Quiz:

1. math.fabs(x)
2. math.floor(x)
3. math.fmod(x)
4. math.ceil(x)
5. math.frexp(x)

Answer: 1, 4

Explanation:

- 選擇 1 和選擇 4 是正確的，因為它們都能滿足函式的要求。
- 選擇 2 是錯誤的，因為 `math.floor(x)` 會將浮點數向下取整，而不是無條件進位。

- 選擇 3 是錯誤的，因為 `math.fmod(x)` 會返回浮點數的模，而不是整數。
- 選擇 5 是錯誤的，因為 `math.frexp(x)` 會返回浮點數的尾數和指數，而不是整數。
- $\text{math.fabs}(x) =$  傳回浮點數的絕對值
- $\text{math.floor}(x) =$  #無條件捨去(向下取整)
- $\text{math.fmod}(x) =$  取餘數(浮點除法後的餘數)
- $\text{math.ceil}(x) =$  無條件進位(向上取整)
- $\text{math.frexp}(x) =$  將浮點數拆成底數和指數(主要用於底層表示)

## 80. 【CH07-7】您正在撰寫一個程式來顯示 My Healthy Eata Delivery 的特價優惠。

```
import datetime
dailySpecials('Spaghetti', 'Macaroni & Cheese', 'Meatloaf', 'Fried Chicken')
weekendSpecials = ('Lobster', 'Prime Rib', 'Parmesan-Crusted Cod')
```

```
print('My Healthy Eats Delivery')
if today == 'Friday' or today == 'Saturday' or today == 'Sunday':
    print('The weekend specials include:')
    for item in weekendSpecials:
        print(item)
else:
    print('The weekday specials include: ')
    for item in dailySpecials:
        print(item)
```

print(f'Pricing specials change in {daysLeft} days')

請選取正確的還項以完成第 4、5 和 15 行的程式碼。

選項答案選擇區

選項 1. 在第 4 行，擷取目前的日期。

- A. now = datetime.datetime.now()
- B. now = datetime.datetime.today()
- C. now = datetime.datetime.day()
- D. now = datetime.datetime()

選項 2. 在第 5 行，擷取工作日。

- A. today = now.strftime('&A')
- B. today = now.strftime('&B')
- C. today = now.strftime('\*')
- D. today = now.strftime('SY')

選項 3. 在第 15 行，計算週剩餘天數。

- A. daysLeft = now.weekday()
- B. daysLeft = 6 - now.weekday()
- C. daysLeft = 6 + now.weekday()
- D. daysLeft = 6 + now.week()

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D
3. (1) A (2) B (3) C (4) D

Answer: 1: A (1), 2: A (1), 3: B (2)

Explanation:

- 選擇 1 的正確答案是 A 和 B，因為它們都可以用來獲取當前日期。
- 選擇 2 的正確答案是 A，因為 `strftime('%A')` 可以獲取當前的工作日。
- 選擇 3 的正確答案是 B，因為 `6 - now.weekday()` 可以計算出週剩餘天數。
- 注意：選項中的字母可能有誤，應該是 `%A` 而不是 `&A;`。

81. 【CH07-8】您正在撰寫一個程式來隨機指派房間(roomnumber)和隊建立組別(group)以進行公司培訓活動。請選取正確的程式碼片段以完成程式碼。

```
import random
```

```
roomsAssigned = [1]
groupList = ['Ropes', 'Rafting', 'Obstacle', 'Wellness']
count = 0
print('Welcome to CompanyPro's Team-Building Weekend!')
name = input('Please enter your name (q to quit)? ')
while name != 'q' and count < 50:
    room_number = random.randint(1,50)
    while room_number in roomsAssigned:
        _(選項 1)_  

        print(f'{name}, your room number is {room_number}')
        roomsAssigned.append(room_number)
        _(選項 2)_  

        print(f'You will meet with the {group} Group this afternoon.')
    count += 1
    name = input('Please enter your name (q to quit)? ')
```

選項答案選擇區

選項 1

- A. room\_number = random.randint(1,50)
- B. room\_number = random.randint(1,51)
- C. room\_number = random.randrange(50)
- D. room\_number = random.randrange(51)

選項 2

- A. group = random.choice(groupList)
- B. group = random.shuffle(groupList)
- C. group = random.random(groupList)
- D. group = random.uniform(groupList)

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D

Answer: 1: A (1), 2: A (1)

#### Explanation:

- 選擇 1 的正確答案是 A，因為 `random.randint(1,50)` 可以生成 1 到 50 之間的隨機整數。
- 選擇 2 的正確答案是 A，因為 `random.choice(groupList)` 可以從 `groupList` 中隨機選擇一個組別。
- 選擇 B、C 和 D 都是不正確的，因為它們不符合要求。

#### 82. 【CH08-1】在下列的語法敘述中，如果是正確的就選擇 Yes，否則請選擇 No。

選項答案選擇區

- A. 在 try 語法中可以有不只一個 except 子句。
- B. 在 try 語法中可以不加 except 子句。
- C. 在 try 語法中可以有一個 finally 子句與 except 子句。
- D. 在 try 語法中可以有不只一個 finally

#### Quiz:

1. (1) Yes (2) No
2. (1) Yes (2) No
3. (1) Yes (2) No
4. (1) Yes (2) No

Answer: 1: Yes (1), 2: Yes (1), 3: Yes (1), 4: No (2)

#### Explanation:

- 在 Python 中，`try` 語法可以有多個 `except` 子句，因此選項 A 是正確的。
- 選項 B 是正確的，因為 `try` 語法可以沒有 `except`，但須搭配 `finally` 子句才合法。
- 選項 C 是正確的，因為 `try` 語法可以有一個 `finally` 子句與多個 `except` 子句。
- 選項 D 是錯誤的，因為 `try` 語法只能有一個 `finally` 子句。

#### 83. 【CH08-2】你製作一個程式詢問使用者家中有多少個小孩，使用者必須輸入整數，如果輸入值不是整數，程式碼必須指出並要求重新輸入。你要如何完成程式碼？請在回答區中選擇適當的程式碼片段。

程式碼片段

```
while True:  
    _(選項 1)_  
    x = int(input('How many children do you have? '))  
    break  
    _(選項 2)_ ValueError:  
        print('Please make sure you entered an integer, please try again...')
```

選項答案選擇區

選項 1

- A. try:
- B. else:
- C. except:
- D. raise:
- E. finally:

選項 2

- A. try
- B. else
- C. except
- D. raise
- E. finally

Quiz:

1. (1) A (2) B (3) C (4) D (5) E
2. (1) A (2) B (3) C (4) D (5) E

Answer: 1: A (1), 2: C (3)

Explanation:

- 選擇 1 的正確答案是 A，因為 `try` 語句用來捕捉可能發生的錯誤。
- 選擇 2 的正確答案是 C，因為 `except` 用來處理 `ValueError` 異常。
- 選擇 B、D 和 E 都是不正確的，因為它們不符合語法要求。

#### 84. 【CH08-3】關於 assert 方法的敘述，請在回答區中選擇還適當的選項。

選項答案選擇區

選項 1. 若要測試變數 x 與變數 y 的值是否相同，可以使用：

- A. assertIs(x, y)
- B. assertIn(x, y)
- C. assertEquals(x, y)
- D. assertIsInstance(x, y)

選項 2. 若要測試物件 x 與物件 y 是否相同，可以使用：

- A. assertIs(x, y)
- B. assertIn(x, y)
- C. assertEquals(x, y)
- D. assertIsInstance(x, y)

選項 3. 若要測試清單中是否存在某個值，可以使用：

- A. assertIs(x, y)
- B. assertIn(x, y)
- C. assertEquals(x, y)
- D. assertIsInstance(x, y)

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D
3. (1) A (2) B (3) C (4) D

Answer: 1: C (3), 2: A (1), 3: B (2)

Explanation:

- 選擇 1 的正確答案是 C，因為 `assertEquals(x, y)` 用來測試變數的值是否相等。
- 選擇 2 的正確答案是 A，因為 `assertIs(x, y)` 用來測試物件是否相同。
- 選擇 3 的正確答案是 B，因為 `assertIn(x, y)` 用來測試某個值是否在清單中。
- `assertEquals(x, y)` 測試的是「x 與 y 的值相等」

- `assertIs(x, y)` 測試的是「`x` 與 `y` 是同一個記憶體中的物件」
- `assertIn(x, y) #` 是否存在於 `y` 裡面(如清單、字串、字典鍵等等)
- `assertIsInstance(x, y) #` 確認 `x` 是否是類型 `y` 的實例，如果不是，則拋出 `AssertionError`

## 85. 【CH08-4】你需要測試某個物件是否為特定類別的執行個體，請問如何進行單元測試？

程式碼片段

```
_(_選項 1)_ unittest
class TestIsInstance(_(_選項 2)_)
def _(_選項 3)_:
    _(_選項 4)_
if __name__ == "__main__":
    unittest.main()
```

選項答案選擇區

選項 1

- A. from
- B. include
- C. import
- D. use

選項 2

- A. TestCase
- B. unittest.TestCase
- C. unittest
- D. TestCase.unittest

選項 3

- A. test isInstance()
- B. isInstance()
- C. test\_isInstance(self)
- D. isInstance(self)

選項 4

- A. assertIsInstance(obj, cls, msg=None)
- B. self.assertEqual(obj, cls, msg=None)
- C. assertIsInstance(obj, cls)
- D. self.assertIsInstance(obj, cls)

Quiz:

1. (1) A (2) B (3) C (4) D
2. (1) A (2) B (3) C (4) D
3. (1) A (2) B (3) C (4) D
4. (1) A (2) B (3) C (4) D

Answer: 1: C (3), 2: B (2), 3: C (3), 4: D (4)

Explanation:

- 選擇 1 的正確答案是 C，因為 `import` 用來導入模組。
- 選擇 2 的正確答案是 B，因為 `unittest.TestCase` 是單元測試的基礎類別。
- 選擇 3 的正確答案是 C，因為 `test\_isinstance(self)` 是測試方法的命名慣例。
- 選擇 4 的正確答案是 D，因為 `self.assertIsInstance(obj, cls)` 用來測試物件是否為特定類別的實例。
- 本題 ITS 答案 C、B、C、B ChatGPT 答案 C、B、C、D

### 86. 【CH08-5】您需要撰寫執行下列工作的程式碼：

1. 呼叫 `process()` 函式。
2. 如果 `process()` 函式回傳錯誤，則呼叫 `logError()` 函式。
3. 呼叫 `process()` 函式之後一律呼叫 `displayResult()` 函式。

你要如何完成程式碼？請在回答區中選擇適當的程式碼片段。

程式碼片段

```
_(_選項 1)_:  
    process()  
_(_選項 2)_:  
    logError()  
_(_選項 3)_:  
    displayResult()
```

選項答案選擇區

- A. assert
- B. raise
- C. except
- D. try
- E. finally

Quiz:

1. (1) A (2) B (3) C (4) D (5) E
2. (1) A (2) B (3) C (4) D (5) E
3. (1) A (2) B (3) C (4) D (5) E

**Answer: 1: D (4), 2: C (3), 3: E (5)**

Explanation:

- 選擇 1 的正確答案是 D，因為 `try` 用來捕捉可能發生的錯誤。
- 選擇 2 的正確答案是 C，因為 `except` 用來處理 `ValueError` 異常。
- 選擇 3 的正確答案是 E，因為 `finally` 用來確保無論是否發生錯誤都會執行的程式碼。

### 87. 【CH08-6】下列函式會計算使用指數之運算式的值。

```
def calc_power(a, b):  
    return a**b  
base = input('Enter the number for the base: ')  
exponent = input('Enter the number for the exponent: ')  
result = calc_power(base, exponent)  
print('The result is ' + result)
```

針對下列每個敘述，如果是正確的就選擇 Yes，否則請選擇 No。

選項答案選擇區

選項 1. 第 2 行會造成執行階段錯誤。

選項 2. 第 6 行會造成執行階段錯誤。

選項 3. eval 函式應該用於第 3 和 4 行。

Quiz:

1. (1) Yes (2) No
2. (1) Yes (2) No
3. (1) Yes (2) No

Answer: 1: Yes (1), 2: Yes (1), 3: Yes (1)

Explanation:

• 選項 1 的正確答案是 Yes，因為 `a` 和 `b` 是字串類型，無法進行指數運算。

• 選項 2 的正確答案是 Yes，因為 `result` 是字串類型，無法與其他字串進行相加運算。

• 選項 3 的正確答案是 Yes，因為 `eval` 函式可以將字串轉換為數值類型。

## 88. 【CH08-7】您撰寫了以下這段程式碼：

```
import sys
try:
    file_in = open('in.txt', 'r')
    file_out = open('out.txt', 'w+')
except IOError:
    print('cannot open', file_name)
else:
    i = i + 1
    for line in file_in:
        print (line.rstrip())
        file_out.write('line ' + str(i) + ':' + line)
        i = i + 1
    file_in.close ()
    file_out.close ()
```

out.txt 檔案不存在執行程式碼。針對下列每個敘述，如果是正確的就選擇 Yes，否則請選擇 No。

選項答案選擇區

選項 1. 此程式碼將正常執行，不會發生錯誤。

選項 2. 此程式碼將執行，但會產生邏輯錯誤。

選項 3. 此程式碼將產生執行階段錯誤。

選項 4. 此程式碼將產生語法錯誤。

Quiz:

1. (1) Yes (2) No
2. (1) Yes (2) No
3. (1) Yes (2) No
4. (1) Yes (2) No

Answer: 1: No (2), 2: Yes (1), 3: Yes (1), 4: No (2)

Explanation:

• 選項 1 的答案是 No — 程式碼中存在未定義變數 i 及 file\_name，將產生 UnDefined Error。

- 選項 2 的答案是 Yes — 它不會正常執行，因為會直接因 `i` 及 `file_name` 未定義而產生執行階段錯誤，也因 `i` 未定義而執行 `i = i + 1` 產生邏輯錯誤。
- 選項 3 的答案是 Yes , `i = i + 1` 在未初始化的情況下會導致 `NameError` , 為執行階段錯誤。
- 選項 4 的答案是 No , 語法是合法的，但執行時會出錯。

89. 【CH04-補充】你編寫了以下程序來處理字符串：

```
text = ' Python Programming '
result = text.strip().upper().split(' ')[0]
```

變量 `result` 的最終值是什麼？

Quiz:

- 'PYTHON'
- 'PYTHON PROGRAMMING'
- ' PYTHON'
- 'PROGRAMMING'

Answer: 1

Explanation:

- `strip()` 會移除字符串前后的空格，變為 'Python Programming'。
- `upper()` 將字符串轉為全大寫，變為 'PYTHON PROGRAMMING'。
- `split(' ')` 以空格切割，產生清單 ['PYTHON', 'PROGRAMMING']。
- [0] 取得清單第一個元素，即 'PYTHON'。

90. 【CH06-補充】關於變數的區域範圍 (Scope)，請檢視以下程式碼：

```
x = 10
def update_x():
    x = 20
update_x()
print(x)
```

執行後的輸出結果為何？

Quiz:

- 10
- 20
- None
- 發生執行階段錯誤

Answer: 1

Explanation:

- 在函數

`update_x`

內部賦值

`x = 20`

是「區域變數」 (Local Variable)。

- 它不會影響外部的「全域變數」 (Global Variable)

`x = 10`

•

- 因此最後 `print(x)` 顯示的仍是全域變數的值 10。

91. 【CH02-補充】在 Python 中執行以下數學運算，結果為何？

```
result = 2 ** 3 ** 2
```

Quiz:

1. 512
2. 64
3. 18
4. 81

Answer: 1

Explanation:

- Python 中的指數運算子 `**` 具有「右結合性」 (Right-to-left associativity)。

- 這意味著

`2 ** 3 ** 2`

等同於

`2 ** (3 ** 2)`

•

- 先計算 3 的平方得 9，再計算 2 的 9 次方，結果為 512。

92. 【CH07-補充】你正在編寫一個通過命令列 (Command Line) 接收參數的 Python 腳本 `script.py`

•

使用者執行命令如下：

```
python script.py data.csv
```

若要取得文件名 'data.csv'，你應使用哪個程式碼？

Quiz:

1. `sys.argv[1]`
2. `sys.argv[0]`
3. `os.args[1]`
4. `sys.input[1]`

Answer: 1

Explanation:

•

`sys.argv`

是一個包含命令列參數的清單。

•

`sys.argv[0]`

永远是脚本本身的名称 ('script.py')。

- sys.argv[1]  
才是第一个传递给脚本的参数 ('data.csv')。

### 93. 【CH04-補充】使用

with open('data.txt', 'r') as f:

語法的主要好處是什麼？

Quiz:

1. 當區塊結束時，檔案會自動關閉，即使發生錯誤也不例外
2. 它會自動將檔案內容讀取到清單中
3. 它允許同時讀取和寫入檔案
4. 它會自動檢查檔案是否存在，不存在則建立之

Answer: 1

Explanation:

- with  
語句 (Context Manager) 確保資源被正確管理。

- 無論  
with  
區塊是正常結束還是因為異常中斷，Python 都會自動呼叫  
f.close()  
, 防止檔案鎖定或資源洩漏。

### 94. 【CH08-補充】檢視以下錯誤處理程式碼：

```
try:  
    num = int('10')  
except ValueError:  
    print('Error')  
else:  
    print('Success')
```

執行後的輸出結果為何？

Quiz:

1. Success
2. Error
3. Error Success
4. 什麼都不會顯示

Answer: 1

Explanation:

- try  
區塊中的  
int('10')  
成功執行，沒有發生錯誤。

- 因此跳過  
except  
區塊。
- 若無錯誤發生，列印出 'Success'。

95. 【CH05-補充】執行以下敘述程式碼，最後會列印出什麼？

```
for i in range(1, 4):
    if i == 2:
        continue
    print(i, end=' ')
```

Quiz:

1. 1 3
2. 1 2 3
3. 1
4. 1 2

Answer: 1

Explanation:

- 當

i  
為 1 時，列印 1。

- 當

i  
為 2 時，觸發  
continue  
, 跳過當次迴圈剩餘程式碼 (不列印)，直接進入下一次迭代。

- 當

i  
為 3 時，列印 3。

- 因此結果為 1 3。

96. 【CH04-補充】使用 f-String (Python 3.6+) 格式化字串，若變數 `name='Alice'` 和 `age=20`，下列哪種寫法可以正確輸出 'Alice is 20'？

Quiz:

1. f'{name} is {age}'
2. f'{name} is {age}'
3. '{name} is {age}'.format()
4. f(name, ' is ', age)

Answer: 1

Explanation:

- f-String 的語法是在字串前加上 `f` 或 `F`，並在字串內部使用 `{}` 包裹變數或運算式。
- 例如 `f'{name} is {age}'` 會自動將變數值填入。

97. 【CH06-補充】如果你想為你的 Python 模組 `my\_module.py` 生成 HTML 格式的說明文件，應該在命令列使用什麼指令？

Quiz:

1. python -m pydoc -w my\_module
2. python -m docgen my\_module
3. pip install documentation
4. python my\_module.py --help

Answer: 1

Explanation:

- `pydoc` 是 Python 的內建文件產生工具。
- `-w` (write) 參數表示將文件寫入 HTML 檔案，而不是僅在控制台顯示。

98. 【CH07-補充】關於 `math.trunc()` 和 `math.floor()` 處理負數時的行為，下列何者正確？(例如處理 -3.5)

Quiz:

1. `math.trunc(-3.5)` 回傳 -3, `math.floor(-3.5)` 回傳 -4
2. 兩者都回傳 -3
3. 兩者都回傳 -4
4. `math.trunc(-3.5)` 回傳 -4, `math.floor(-3.5)` 回傳 -3

Answer: 1

Explanation:

- `math.trunc(x)` (Truncate) 是\*\*無條件捨去小數部分\*\*，向 0 取整。所以 -3.5 變成 -3。
- `math.floor(x)` 是\*\*向下取整\*\* (往負無限大方向)。所以 -3.5 會變成更小的整數 -4。

99. 【CH07-補充】你希望從清單 `[1, 2, 3, 4, 5]` 中\*\*隨機選取 3 個不重複\*\*的元素，應該使用 `random` 模組的哪個方法？

Quiz:

1. random.sample([1, 2, 3, 4, 5], 3)
2. random.choices([1, 2, 3, 4, 5], k=3)
3. random.select([1, 2, 3, 4, 5], 3)
4. random.pick([1, 2, 3, 4, 5], 3)

Answer: 1

Explanation:

- random.sample(population, k)  
用於不重複取樣 (Without replacement)。
- random.choices(population, k=N)  
用於可重複取樣 (With replacement)。

#### 【常用方法對照表】

方法 (Method) 功能描述語法範例參數 k (Count) 取樣性質 sample 取多個 sample(seq, 3)

k=3 (取 3 個)

□□ 必填 (未填報錯) □ 不重複 (Unique) choices 取多個

choices(seq, k=3)

k=3 (取 3 個)

□ 選填 (預設 k=1) □ 可重複 (Repeats) choice 取單個

choice(seq)  
□ 不支援1□□ 單一值 (Single)pick(錯誤)  
pick(...)  
N/A□ 不存在此語法

100. 【CH07-補充】你正在設計一個遊戲的抽獎系統，獎品有三種： - \*\*大獎 (Gold)\*\*：機率 10%  
- \*\*二獎 (Silver)\*\*：機率 30% - \*\*安慰獎 (Stone)\*\*：機率 60% 若希望根據上述機率\*\*隨機抽取 5 次\*\* (可重複中獎)，應該使用下列哪一行程式碼？

Quiz:

1. random.choices(['Gold', 'Silver', 'Stone'], weights=[10, 30, 60], k=5)
2. random.sample(['Gold', 'Silver', 'Stone'], weights=[10, 30, 60], k=5)
3. random.choices(['Gold', 'Silver', 'Stone'], k=5)
4. random.sample(['Gold', 'Silver', 'Stone'], 5)

Answer: 1

Explanation:

- \*\*`random.choices`\*\* 支援 `weights` 參數來設定每個元素的權重 (機率)，且支援可重複取樣 ('k' 可大於清單長度)。
- \*\*`random.sample`\*\* \*\*不支援\*\* `weights` 參數，且無法在母體不足時取出比母體更多的元素 (會報錯)。

#### 【選項解析】

選項語法特徵結果與問題1 (正確)

choices(..., weights=[...], k=5)

□ 成功。

依指定機率 (10:30:60) 隨機選 5 次。2

sample(..., weights=...)

□ 錯誤 (TypeError)。

sample  
不接受  
weights  
參數。3  
choices(..., k=5)  
□ 機率相等。  
未設權重，每個獎品機率皆為 33.3%。4  
sample(..., 5)  
□ 錯誤 (ValueError).  
母體只有 3 個，無法取出 5 個不重複元素。

101. 【CH07-補充】執行以下程式碼後，變數 `nums` 的狀態為何？

```
import random
nums = [1, 2, 3, 4, 5]
random.shuffle(nums)
```

Quiz:

1. nums 的元素順序被隨機打亂，且直接修改了原始清單
2. nums 保持不變，shuffle 回傳了一個新的打亂清單
3. nums 變成了 None
4. 會發生錯誤，因為 list 不支援 shuffle

Answer: 1

Explanation:

- `random.shuffle(x)` 會\*\*原地 (in-place)\*\* 打亂序列 `x` 的順序。
- 它不會回傳新清單 (回傳 None)，而是直接修改傳入的物件。

```
import random
nums = [1, 2, 3, 4, 5]
random.shuffle(nums)
print(nums) # Output could be: [3, 5, 2, 1, 4]
```

102. 【CH08-補充】在函式中，如果你檢測到一個不合法的參數值，並希望主動中斷程式執行並回報錯誤，應該使用哪個關鍵字？

Quiz:

1. raise ValueError('Invalid value')
2. throw ValueError('Invalid value')
3. return ValueError('Invalid value')
4. break ValueError('Invalid value')

Answer: 1

Explanation:

- Python 使用 `raise` 關鍵字來主動引發 (throw) 一個例外。
- `throw` 是 Java/C++ 等語言的關鍵字，Python 不支援。
- `return` 只是回傳值，不會中斷程式流程進入錯誤處理機制。

103. 【CH06-補充】你需要測試一段會輸出文字到控制台 (stdout) 的程式碼，但不希望文字真的印在螢幕上，而是想存在記憶體中以便分析。可以使用哪個模組？

Quiz:

1. io.StringIO
2. sys.stdout\_capture
3. os.memory\_file
4. io.TextWrapper

Answer: 1

Explanation:

- `io.StringIO` 建立一個在記憶體中的檔案物件 (In-memory file-like object)。
- 它常用於單元測試中，用來模擬檔案或標準輸出/輸入流 (stdin/stdout)，將字串寫入記憶體緩衝區。

104. 【CH08-補充】在 `unittest` 模組中，若要驗證 `x` 的值是否為 `True`，應使用哪個斷言方法？

Quiz:

1. self.assertTrue(x)
2. self.assertIsTrue(x)
3. self.assertEqual(x, 'True')
4. self.assert(x)

Answer: 1

Explanation:

- `assertTrue(x)` 是 `unittest.TestCase` 的方法，用於驗證 `x` 是否為真值 (Truthiness)。
- `assertIsTrue` 不是標準方法名稱。
- `assertEquals(x, 'True')` 會檢查 `x` 是否等於字串 'True'，這通常不是我們想要的。

105. 【CH03-補充】你有一個名為 colors 的清單：

colors = ['red', 'green', 'blue', 'yellow']

執行 print(colors[-2:]) 的結果是什麼？

Quiz:

1. ['blue', 'yellow']
2. ['green', 'blue']
3. ['yellow']
4. ['red', 'green']

Answer: 1

Explanation:

Python 的切片中，負數索引表示從末尾開始算。

106. 【CH03-補充】關於 Python 中的元組 (Tuple) 和清單 (List)，下列哪一項敘述是正確的？

Quiz:

1. 元組是不可變的 (Immutable)，而清單是可變的 (Mutable)
2. 元組和清單都可以使用 append() 方法動態增加元素
3. 元組定義時使用中括號 []，清單使用小括號 ()
4. 元組不能包含不同類型的數據

Answer: 1

Explanation:

• 元組 (Tuple) 一旦建立就無法修改其內容，稱為不可變 (Immutable)。

• 清單 (List) 可以隨時增加、刪除或修改元素，稱為可變 (Mutable)。

• 元組使用

()

，清單使用

[]

。

### 【1. 符號與空值設定表】

類型符號一般定義 (範例) 空內容設定清單 (List)

[]

[1, 2]

[]

元組 (Tuple)

()

(1, 2)

()

字典 (Dict)

{}

{'k': 'v'}

{}

集合 (Set)

{}

{1, 2}

set()

\*注意\*注：空集合必須使用

set()  
。  
**【2. 集合运算符号 (Set Operations)】**  
运算符号逻辑 / 用途交集 (Intersection)  
&  
且 (AND)  
找共同点 (两人都有) 联集 (Union)  
|  
或 (OR)  
合并所有 (去除重复) 差集 (Difference)  
-  
扣除  
A有但B没有的

### 107. 【CH03-補充】你有一個字典

student = {'name': 'John', 'age': 18}

- 。你希望取得 'score' 的值，如果 'score' 不存在，則回傳 0，避免程式崩潰。應使用哪種語法？

Quiz:

1. student.get('score', 0)
2. student['score'] or 0
3. student.find('score', 0)
4. student['score'] if exists else 0

Answer: 1

Explanation:

- get(key, default)  
是字典安全取值的標準方法。

- 選項 2 錯誤：

student['score']  
若鍵不存在會直接拋出 KeyError，程式崩潰，無法執行後面的  
or 0

。

- 選項 3 錯誤：字典物件沒有  
find()  
方法 (這是字串的方法)。

- 選項 4 錯誤：語法錯誤，Python 中沒有  
exists

關鍵字，正確寫法應為

student['score'] if 'score' in student else 0

。

### 108. 【CH03-補充】你希望在清單 `fruits = ['apple', 'banana']` 的\*\*開頭\*\*插入一個元素 'orange'。應該使用哪種方法？

Quiz:

1. fruits.insert(0, 'orange')
2. fruits.append('orange')
3. fruits.add('orange')

4. fruits[0] = 'orange'

Answer: 1

Explanation:

- insert(index, element)  
方法可以在指定索引位置插入元素。  
insert(0, 'orange')  
会将 'orange' 插入到清单的最前面。

- append()  
只能加在末尾。

- add()  
是集合 (Set) 的方法。

- fruits[0] = ...  
会取代原本的第一个元素，而不是插入。

### 【数据新增与插入综合比较】

类型方法核心特性清单 (List)

.append(x)

.insert(i, x)  
append: 排在最后  
insert: 插在指定位置 (i)集合 (Set)  
.add(x)  
无序，没有所谓「插入位置」。  
不能用 append。字典 (Dict)  
d[key] = val  
直接指定 Key 赋值。  
若 Key 存在则覆盖，不存在则新增。元组 (Tuple) 为不可变 (Immutable)  
无法新增或插入数据。

### 109. 【CH03-補充】關於 `sort()` 和 `sorted()` 的區別，下列敘述何者正確？

Quiz:

1. `list.sort()` 會直接修改原始清單 (In-place)，而 `sorted(list)` 會回傳一個新的已排序清單
2. `sorted(list)` 會直接修改原始清單，而 `list.sort()` 會回傳新清單
3. 兩者功能完全相同，只是語法不同
4. `sort()` 只能用於數字清單，`sorted()` 只能用於字串清單

Answer: 1

Explanation:

- `list.sort()` 是清單物件的方法，它會\*\*原地 (in-place)\*\* 修改清單順序，回傳 `None`。
- `sorted()` 是內建函式，它會接受一個可迭代物件，並\*\*回傳一個新的\*\*已排序清單，原始清單不會改變。

### 110. 【CH03-综合应用】执行下列 Python 代码后，最终打印出的结果为何？

```
nums = [1, 2, 2, 3]
nums = list(set(nums))
nums.append(4)
print(nums)
```

Quiz:

1. [1, 2, 3, 4]
2. [1, 2, 2, 3, 4]
3. {1, 2, 3, 4}
4. (1, 2, 3, 4)

Answer: 1

Explanation:

此题考查了三个步骤的逻辑变化:

### 【步骤解析】

步骤代码数据变化说明

nums = [1, 2, 2, 3]

[1, 2, 2, 3]

初始清单，包含重复的 2。

set(nums)

{1, 2, 3}

转为集合，自动去除重复值。

list(...)

[1, 2, 3]

再转回清单，准备进行修改。

nums.append(4)

[1, 2, 3, 4]

清单是可变的，加入新元素 4。

### 【错误选项分析】

• 选项 B: 错误，因为转为 set 时重复的 2 已经被移除了。

• 选项 C: 错误，最终已转回 List，符号应为

[]

而非

{}

。

• 选项 D: 错误，这是 Tuple 的符号。

111. 【CH03-集合运算】观察下列代码，print(result) 的输出结果为何？

```
set_a = {1, 2, 3, 4}  
set_b = {3, 4, 5, 6}  
result = set_a & set_b  
print(result)
```

Quiz:

1. {1, 2, 3, 4, 5, 6}
2. {3, 4}
3. [3, 4]
4. {1, 2, 5, 6}

Answer: 2

Explanation:

此题考查集合 (Set) 的交集运算符号

&

。

### 【解析步骤】

符号名称逻辑计算结果

&

交集 (Intersection) 且 (AND)

两个集合都有的元素 set\_a 有 3,4

set\_b 有 3,4

结果:

{3, 4}

### 【错误选项排除】

- 选项 A: 这是联集 (Union, 符号 | ) 的结果。
- 选项 C: 集合运算的结果依然是集合 {}, 不会自动变成清单 []。
- 选项 D: 这是差集的概念 (互有的差异)。

112. 【CH03-字典操作】已知字典 data = {'a': 10, 'b': 20}, 下列哪一种写法可以正确印出 20?

Quiz:

1. print(data[1])
2. print(data['b'])
3. print(data.b)
4. print(data(20))

Answer: 2

Explanation:

字典 (Dictionary) 是透过键 (Key) 来查找值 (Value), 而不是透过数字索引。

### 【选项解析】

选项程式码结果 / 原因A

data[1]

报错 (KeyError)

字典没有顺序, 不能用数字索引。B

data['b']

正确

透过 Key 'b' 找到对应的 Value 20。C

data.b

报错 (AttributeError)

Python 字典不支援点号 (.) 访问属性。D

data(20)

报错 (TypeError)

字典不是函式, 不能用小括号呼叫。

113. 【CH03-不可变性】下列哪一行代码在执行时会产生错误 (Error)?

Quiz:

1. my\_list = [1, 2, 3]; my\_list[0] = 9
2. my\_dict = {'x': 1}; my\_dict['x'] = 9
3. my\_tuple = (1, 2, 3); my\_tuple[0] = 9

4. my\_set = {1, 2, 3}; my\_set.add(9)

Answer: 3

Explanation:

本题考查数据结构的可变性 (Mutability)。

#### 【逐项分析】

A (List): 清单是可变的 (Mutable), 可以修改指定位置的值。 -> PassB (Dict): 字典是可变的, 可以更新 Key 对应的值。 -> PassC (Tuple): 错误 (TypeError)

元组是不可变 (Immutable) 的, 一旦建立, 不能修改内部元素。D (Set): 集合可以透过 add() 方法新增数据。  
-> Pass

114. 【CH03-集合操作】你想要在一个集合 `my\_set = {1, 2, 3}` 中加入数字

4. 下列哪一行代码是正确的?

Quiz:

1. my\_set.append(4)
2. my\_set.add(4)
3. my\_set.insert(3, 4)
4. my\_set[3] = 4

Answer: 2

Explanation:

此题考查集合 (Set) 的专用新增方法。

#### 【分析】

● 选项 B (正确): 集合是无序的, 像是把东西丢进篮子, 所以使用 add()

。

● 选项 A (错误):

append()

是清单 (List) 专用的, 集合没有顺序, 不能「排在最后」。

● 选项 C (错误): 集合没有索引, 无法使用

insert()

指定位置。

● 选项 D (错误): 集合不支援索引赋值 (Item Assignment)。

115. 【CH03-除错】执行下列哪一段代码会导致 `AttributeError` (属性错误)?

Quiz:

1. data = [1, 2]; data.append(3)
2. data = {1, 2}; data.add(3)
3. data = {'a': 1}; data['b'] = 2
4. data = (1, 2); data.append(3)

Answer: 4

Explanation:

本题考查不同资料结构的可用方法 (Method)。

#### 【逐项检查】

选项类型程式码结果A清单 (List)

.append(3)

成功 (加入末尾)B集合 (Set)

.add(3)

成功 (新增元素)C字典 (Dict)

`['b'] = 2`

成功 (新增 Key-Value)D元组 (Tuple)

`.append(3)`

报错

元组是不可变的，且本身没有 `append` 方法。

116. 【CH03-清单逻辑】执行下列代码后，`nums` 的内容为何？

`nums = [10, 30]`

`nums.insert(1, 20)`

Quiz:

1. `[10, 20, 30]`
2. `[20, 10, 30]`
3. `[10, 30, 20]`
4. `[10, 20]`

Answer: 1

Explanation:

此题考查 `insert(index, value)` 的运作逻辑。

【图解步骤】

1. 初始清单:

`[10, 30]`

(索引 0 是 10, 索引 1 是 30)

2. 执行

`insert(1, 20)`

: 意思是「我要插队占领索引 1 的位置，把原本的人往后挤」。

3. 结果:

索引 0: 10 (不动)

索引 1: 20 (新来的)

索引 2: 30 (原本的索引 1 被挤到了 2)

故结果为

`[10, 20, 30]`

。

117. 【CH06-補充】你正在開發一個科學計算程式，需要判斷一個變數 `x` 是否為合法的數值 (不是 `NaN`)。

下列哪一種方式是\*\*正確\*\*的判斷方法？

Quiz:

1. `if x == float('nan'):`
2. `if x == math.nan:`
3. `if math.isnan(x):`
4. `if x is math.nan:`

Answer: 3

Explanation:

• 在 Python (IEEE 754 標準) 中，`NaN` (Not a Number) 有一個特殊性質：\*\*它不等於自己\*\*。

• 因此 `x == math.nan` 永遠會回傳 `False`。

• 必須使用專用函式 `math.isnan(x)` 來檢查。

118. 【CH06-補充】關於 `math.fmod(x, y)` 與模數運算子 `x % y` 的差異，下列敘述何者正確？

Quiz:

1. 兩者功能完全相同，沒有差異
2. `math.fmod` 適合處理整數，`%` 適合處理浮點數
3. 當處理負數時，`math.fmod` 的結果符號會隨被除數 (x)，而 `%` 的結果符號隨除數 (y)
4. `math.fmod` 只回傳整數，`%` 回傳浮點數

Answer: 3

Explanation:

- `math.fmod(-10, 3)` 會得到 `-1.0` (符號跟隨 -10)。
- `-10 % 3` 會得到 `2` (符號跟隨 3)。
- 這是 ITS/Python 考試中關於數學模組的經典細節題。

119. 【CH06-補充】你需要寫一個程式來清理暫存檔。若要\*\*刪除\*\*一個已存在的檔案 `temp.txt`，應該使用 `os` 模組的哪個函式？

Quiz:

1. os.delete('temp.txt')
2. os.remove('temp.txt')
3. os.path.delete('temp.txt')
4. os.files.del('temp.txt')

Answer: 2

Explanation:

- Python 中刪除檔案的標準函式是 `os.remove()` (或 `os.unlink()`)。
- `os.rmdir()` 是用來刪除空資料夾的。
- 沒有 `os.delete` 這個函式。

120. 【CH03-補充】你有一個清單 `scores = [100, 80, 90, 80]`。你想要移除\*\*第一個\*\*出現的 80 分。應該使用哪個方法？

Quiz:

1. scores.remove(80)
2. scores.pop(80)
3. scores.delete(80)
4. del scores[80]

Answer: 1

Explanation:

- `list.remove(x)`：移除清單中\*\*第一個\*\*值為 `x` 的元素。
- `list.pop(i)`：移除\*\*索引\*\*為 `i` 的元素 (若不填則移除最後一個)。
- `pop(80)` 會試圖移除索引 80 的元素，導致 IndexError。

121. 【CH06-補充】你的程式需要計算一個圓的面積，公式為  $\pi r^2$ 。使用 `math` 模組時，下列寫法何者最精確且標準？

Quiz:

1. area = 3.14 \* (r \*\* 2)
2. area = math.pi \* math.pow(r, 2)
3. area = math.PI \* r \* r
4. area = math.pi \* math.sqrt(r)

Answer: 2

Explanation:

- `math.pi` 提供比 3.14 更精確的圓周率數值。
- `math.pow(x, y)` 是計算 x 的 y 次方。
- 雖然 `math.pi \* r\*\*2` 也是對的，但在考題選項中，通常測試對 `math` 模組函式 (`pow`, `pi`) 的認識。

122. 【CH03-補充】你這在編寫一個程式，需要將清單 `list\_b` 的\*\*所有元素\*\*合併到 `list\_a` 的尾端，形成一個長清單。應該使用哪種方法？

```
list_a = [1, 2]
list_b = [3, 4]
# 預期結果: [1, 2, 3, 4]
```

Quiz:

1. list\_a.extend(list\_b)
2. list\_a.append(list\_b)
3. list\_a.insert(list\_b)
4. list\_a.add(list\_b)

Answer: 1

Explanation:

- `extend()` 會將另一個清單的內容\*\*拆解\*\*後加入。
- `append()` 會將整個清單當作\*\*一個元素\*\*加入 (變成 `[1, 2, [3, 4]]` )。
- 這是 ITS 考試中關於 List Merging 的常見考點。

123. 【CH03-補充】你有一個學生成績清單 `grades = [85, 92, 78, 90, 55]`。若要快速找出清單中的\*\*最高分\*\*與\*\*最低分\*\*，應使用哪些內建函式？

Quiz:

1. max(grades), min(grades)
2. grades.max(), grades.min()
3. largest(grades), smallest(grades)
4. high(grades), low(grades)

Answer: 1

Explanation:

- Python 提供內建函式 `max()` 和 `min()` 來處理序列資料。
- List 物件本身沒有 `.max()` 或 `.min()` 方法。

124. 【CH06-補充】關於 `math.sqrt(x)` 與 `math.isqrt(x)` 的差異，下列何者正確？

Quiz:

1. `math.sqrt` 回傳浮點數 (float)，`math.isqrt` 回傳整數 (int) 且無條件捨去小數
2. `math.sqrt` 可以處理負數，`math.isqrt` 不行
3. 兩者功能完全相同，只是新舊版本差異
4. `math.isqrt` 會四捨五入到最接近的整數

Answer: 1

Explanation:

- `math.sqrt(10)` 回傳 `3.16...` (float)。
- `math.isqrt(10)` 回傳 `3` (int)，即整數平方根 (Integer Square Root)。
- 這是 Python 3.8 新增的函式，也在 ITS 考綱範圍內。

125. 【CH06-補充】你需要分析浮點數的底層結構。請問 `math.frexp(x)` 函式的作用是什麼？

Quiz:

1. 回傳 x 的尾數 (mantissa) 與指數 (exponent)
2. 回傳 x 的階乘 (factorial)
3. 回傳 x 的絕對值 (absolute value)
4. 回傳 x 的自然對數 (natural log)

Answer: 1

Explanation:

- `math.frexp(x)` 是 ITS 考綱中列出的較冷門函式。
- 它會將浮點數分解為 `(m, e)`，使得 `x = m \* (2 \*\* e)`。
- 例如 `math.frexp(8.0)` 回傳 `(0.5, 4)`，因為 `0.5 \* 16 = 8`。

126. 【CH07-補充】你使用 `datetime` 模組查詢今天是星期幾：

```
import datetime
today = datetime.datetime.now()
print(today.weekday())
```

如果輸出結果是 `0`，代表今天是星期幾？

Quiz:

1. 星期一 (Monday)
2. 星期日 (Sunday)
3. 星期六 (Saturday)
4. 1 月 1 日

Answer: 1

Explanation:

- 在 Python 的 `datetime.weekday()` 方法中：
- \*\*0 代表星期一\*\*
- \*\*6 代表星期日\*\*

- 注意與某些資料庫或語言 (如 1=Sunday) 的差異。

127. 【CH03-補充】你想要將一個清單 `nums = [1, 2, 3]` 的順序\*\*原地 (in-place) 反轉\*\*，使其變為 `[3, 2, 1]` 且不產生新的清單物件。應該使用哪種語法？

Quiz:

1. `nums.reverse()`
2. `nums = reversed(nums)`
3. `nums = nums[::-1]`
4. `nums.sort(reverse=False)`

Answer: 1

Explanation:

- `list.reverse()` 是清單的方法，會直接修改原始清單，回傳 `None`。
- `reversed(list)` 是內建函式，會回傳一個新的迭代器，不會修改原始清單。
- `[::-1]` 會產生一個新的清單複本。

128. 【CH07-補充】你需要產生一個 \*\*0.0 到 1.0 之間\*\* (包含 0.0 但不包含 1.0) 的隨機\*\*浮點數\*\*。應該使用 `random` 模組的哪個函式？

Quiz:

1. `random.random()`
2. `random.randint(0, 1)`
3. `random.uniform(0, 10)`
4. `random.float()`

Answer: 1

Explanation:

- `random.random()`: 回傳 [0.0, 1.0) 之間的浮點數，這是最基礎的隨機函式。
- `random.randint(a, b)`: 回傳整數。
- `random.uniform(a, b)`: 回傳指定範圍內的浮點數 (不一定是 0-1)。
- 沒有 `random.float()` 這個函式。

129. 【CH08-補充】在單元測試 (Unittest) 中，若要驗證兩個變數是否指向\*\*同一個物件\*\* (記憶體位址相同，而不僅僅是值相等)，應使用哪個斷言方法？

Quiz:

1. `self.assertIs(a, b)`
2. `self.assertEqual(a, b)`
3. `self.assertSame(a, b)`
4. `self.assertTrue(a == b)`

Answer: 1

Explanation:

- `assertIs(a, b)`: 檢查 `a is b` (是否為同一物件)。
- `assertEquals(a, b)`: 檢查 `a == b` (值是否相等)。

- 例如: `a = [1]; b = [1]`，它們的值相等 (`Equal`) 但不是同一個物件 (`Not Is`)。

**【演算法-佇列】**此程式中 q 是用來當作一個先進先出的佇列 (queue)。push 與 pop 兩函數中所缺的指令 x 與 y 應填入下列何者？

```
q = []
head = 0
def push(q, c):
    x
#
def pop(q):
    c = q[head]
    y
    return c
```

Quiz:

1. x: q[head]=c; y: head += 1
2. x: q.append(c); y: head += 1
3. x: q[head]=c; y: head -= 1
4. x: q.append(c); y: head -= 1
5. 以上皆非

Answer: 2

Explanation:

本題考查佇列 (Queue) 的 FIFO (先進先出) 實作邏輯。

**【解析】**

1. Push (入列): 必須將新資料加在隊伍的「最後面」。在 Python List 中，`append(c)` 用於將元素加到尾端。若使用 `q[head]=c` 會覆蓋掉頭部資料，導致錯誤。
2. Pop (出列): 必須取出隊伍「最前面」的資料。程式碼 `c = q[head]` 已取出當前頭部資料。為了讓下一次 Pop 取出下一個元素，頭部指標 `head` 必須往後移動 (索引值 +1)，故 `y` 應為 `head += 1`。

故選 (B)。

**【Python運算子】**執行下列程式碼後，輸出為何？

```
a = 13 % 5
b = -13 % 5
c = -13 // 5
print(a, b, c)
```

Quiz:

1. 3 -3 -3
2. 3 2 -3
3. 3 -3 -2
4. 3 2 -2
5. 執行錯誤

Answer: 2

Explanation:

本題考查 Python 對負數進行除法與取餘數的規則。

**【解析】**

1. `a = 13 % 5`:

13 除以 5，商 2 餘 3。結果為 3。

2. `c = -13 // 5` (地板除法):

Python 的 `//` 是向負無限大方向取整 (Floor Division)。

$-13 \div 5 = -2.6$ , 向負無限大取整後為 -3。

3. `b = -13 % 5` (取餘數):

公式為:

$$a \% b = a - (b * (a // b))$$

。代入數值:

$$-13 - (5 * (-3)) = -13 - (-15) = -13 + 15 = 2$$

。或者記憶口訣: Python 取餘數的符號永遠與除數 (後者) 相同。因為除數 5 是正數，餘數必為正數 2。

故輸出為

3 2 -3

，選 (B)。

【Python迴圈邏輯】執行下列程式碼後，輸出為何？

```
s = [1, 1, -3, 2, 4, -1, 2, 3, -5, 2, -1, 1]
```

```
q = []
```

```
h = 0
```

```
for v in s:
```

```
    if v > 0:
```

```
        q.append(v)
```

```
    else:
```

```
        while v < 0 and h < len(q):
```

```
            v += q[h]
```

```
            h += 1
```

```
print(sum(q[h:]))
```

Quiz:

1. 1

2. 3

3. 4

4. 6

5. 以上皆非或執行錯誤

Answer: 2

Explanation:

此題模擬了一個類似佇列 (Queue) 的消去過程：正數入列，負數則從隊頭開始抵銷正數。

【執行步驟追蹤】

1.

s

為輸入列表，

q

為正數佇列，

h

指向

q

目前有效的開頭索引。

2. 遇到 1, 1:

q=[1, 1]  
◦ 3. 遇到 -3: 抵銷

q[0]

(1) 剩 -2,

h=1

; 再抵銷

q[1]

(1) 剩 -1,

h=2

◦

q

有效部分為空。

4. 遇到 2, 4:

q=[1, 1, 2, 4]

(有效部分: 2, 4),

h=2

◦ 5. 遇到 -1: 抵銷

q[2]

(2) 剩 1,

h=3

◦

v

變正數，停止抵銷。

6. 遇到 2, 3:

q=[..., 4, 2, 3]

(有效部分: 4, 2, 3),

h=3

◦ 7. 遇到 -5: 抵銷

q[3]

(4) 剩 -1,

h=4

; 再抵銷

q[4]

(2) 剩 1,

h=5

◦ 停止。

8. 遇到 2:

q=[..., 3, 2]

(有效部分: 3, 2),

h=5

◦ 9. 遇到 -1: 抵銷

q[5]

(3) 剩 2,

h=6

◦ 停止。

10. 遇到 1:

q=[..., 2, 1]

(有效部分: 2, 1),

h=6

◦

最後

q[h:]

即

q[6:]

, 內容為  
[2, 1]  
, 總和為 3。故選 (B)。

【Python流程控制】執行下列程式碼後，輸出為何？

```
q = []
h = 0
s = ""
for v in [3, 12, -2, 765, 23, -1, -3]:
    if v > 0:
        q.append(str(v))
    else:
        s += q[h]
        h += 1
print(s)
```

Quiz:

1. 312765
2. 780
3. 312
4. 執行錯誤

Answer: 1

Explanation:

此題模擬併列 (Queue) 行為：

1. 遇到正數時，將數字轉為字串並加入清單 `q` (Enqueue)。
2. 遇到負數時，從清單 `q` 的 `h` 位置取出字串並串接到 `s`，同時將 `h` 加 1 (Dequeue)。

【執行過程】

- 入列: '3', '12'
- 出列: '3' (s='3')
- 入列: '765', '23'
- 出列: '12' (s='312')
- 出列: '765' (s='312765')

故結果為 312765。

【Python迴圈與列表】執行下列程式碼後，輸出為何？

```
s = [1, 1, -3, 3, 4, -1, 2, 3, -5, 2, 1, -1]
q = []
for v in s:
    if v > 0:
        q.append(v)
    else:
        while v < 0 and q:
            v += q.pop()
```

```
print(sum(q))
```

Quiz:

- 1. 1
- 2. 3
- 3. 5
- 4. 7
- 5. 以上皆非或執行錯誤

Answer: 3

Explanation:

此題模擬正負數抵銷的過程：正數入列，負數則從隊尾取出正數進行抵銷。

### 【關鍵邏輯】

當遇到負數

v

時，會不斷從

q

尾端

pop()

出正數加到

v

上，直到

v

變成非負數 ( $>=0$ ) 或

q

變空為止。注意：抵銷後若

v

變為正數，該數值會被捨棄，不會加回

q

。

### 【執行追蹤】

1.

1, 1

入列 ->

q=[1, 1]

2.

-3

抵銷 ->

pop

1 (剩-2),

pop

1 (剩-1)。

q=[]

3.

3, 4

入列 ->

q=[3, 4]

4.

-1

抵銷 ->

pop

4 (剩3)。停止 ( $3 >= 0$ )。

q=[3]

5.

2, 3  
入列 ->  
q=[3, 2, 3]

6.  
-5  
抵銷 ->  
pop  
3 (剩-2),  
pop  
2 (剩0)。停止 (0>=0)。  
q=[3]

7.  
2, 1  
入列 ->  
q=[3, 2, 1]

8.  
-1  
抵銷 ->  
pop  
1 (剩0)。停止。  
q=[3, 2]

最後  
sum([3, 2])  
為 5。

【Python變數範圍】執行下列程式碼：

```
s = input()  
a = [s for s in input().split()]  
print(s, a)
```

若輸入為以下內容：

hello py

123 45

則輸出為何？

Quiz:

1. hello py [123, 45]
2. hello 123 45
3. hello ['123', '45']
4. hello py ['123', '45']
5. 以上皆非或執行錯誤

Answer: 4

Explanation:

本題考查 Python 3 中列表生成式 (List Comprehension) 的變數範圍規則。

1.  
s = input()

: 讀取第一行,  
s  
為  
'hello py'  
。  
2.  
a = [s for s in input().split()]  
: 讀取第二行並切割為  
['123', '45']  
。雖然生成式內部使用了變數名稱  
s  
, 但在 Python 3 中, 這被視為區域變數, 不會覆蓋外部原本的變數  
s  
。  
3. 清單內容為字串, 故輸出時會帶有引號。  
4. 最後  
print(s, a)  
輸出外部的  
s  
和清單  
a  
, 即  
hello py ['123', '45']  
。

註: 若是 Python 2, 變數  
s  
會被洩漏並覆蓋, 輸出結果會不同, 但 ITS 考試以 Python 3 為主。

【Python輸入與運算】執行下列程式碼:

```
a, b = [int(x) for x in input().split()]
print(a + b)
```

若輸入為以下內容:

100 45

則輸出為何?

Quiz:

1. 10045
2. [100, 45]
3. 145
4. [10045]
5. 以上皆非或執行錯誤

Answer: 2

Explanation:

本題考查 `input` 讀取、`split` 切割、`int` 型別轉換以及清單解包 (List Unpacking)。

【解析步驟】

1.  
`input().split()`  
將輸入字串 "100 45" 切割成清單  
['100', '45']

。2. 列表生成式將元素轉為整數，得到

[100, 45]

。

3.

a, b = ...

將列表解包，

a

變成整數 100，

b

變成整數 45。

4.

print(a + b)

執行數學加法  $100 + 45 = 145$ 。

若未進行 int 轉換直接相加，結果才會是字串串接的 "10045" (選項 A)。本題有轉 int，故選 (C)。

【Python解包錯誤】執行下列程式碼：

```
a, b = [int(x) for x in input().split()]
print(a, b)
```

若輸入為以下內容：

100 45 25

則輸出為何？

Quiz:

1. 100 45
2. [100, 45], 25
3. [100, [45, 25]]
4. 45 25
5. 以上皆非或執行錯誤

Answer: 5

Explanation:

本題考查 Python 的解包 (Unpacking) 規則。

【解析】

1. 輸入 `100 45 25` 經過處理後會變成包含 3 個元素的列表 `[100, 45, 25]`。

2. 左側只有 `a, b` 兩個變數。

3. 當右側值的數量 (3個) 多於左側變數的數量 (2個) 時，Python 無法自動決定如何分配，因此會拋出 `ValueError: too many values to unpack`。

若要避免錯誤，需使用 `\*b` (Extended Iterable Unpacking) 或確保數量一致。故本題為執行錯誤，選 (E)。

【Python字元編碼】執行下列程式碼：

```
a = "fake"
b = ord(a[3]) - ord(a[1])
c = a + chr(ord('f') + b)
print(c)
```

輸出為何？

Quiz:

1. fakek
2. fakej
3. fakk
4. fakj
5. 執行錯誤 Error

Answer: 2

Explanation:

本題考查

ord()

與

chr()

的轉換。

### 【解析】

1.  
a[3]  
='e',  
a[1]  
='a'。
2.  
 $b = \text{ord}('e') - \text{ord}('a')$   
, 即 'e' 與 'a' 的距離，結果為 4。
3.  
 $\text{ord}('f') + 4$   
, 即 'f' 往後推 4 個字母: g, h, i, j。
4.  
chr()  
轉回字元得到 'j'。
5. 字串串接 "fake" + "j" = "fakej"。

故選 (B)。

【APCS-進階切片】執行下列程式碼後，輸出為何？

```
s = 'abcdefg'  
print(s[::-2])
```

Quiz:

1. geca
2. aceg
3. gec
4. db

Answer: 1

Explanation:

本題考查字串切片的步長 (Step) 與方向。

### 【解析】

語法為

$s[\text{start}:\text{end}:\text{step}]$

。當 step 為負數 (  
-2)

) 時，表示從後往前每隔 2 個字元取一次。

1. 取最後一個 'g' (索引 -1)
2. 往前跳 2 格取 'e' (索引 -3)
3. 往前跳 2 格取 'c' (索引 -5)
4. 往前跳 2 格取 'a' (索引 -7)

結果為

'geca'

。

ITS 較少考這種負數步長，但 APCS 必考。

### 【APCS-遞迴邏輯】執行下列函式

f(3)

的回傳值為何？

```
def f(n):  
    if n == 0:  
        return 0  
    return n + f(n-1)
```

Quiz:

1. 0
2. 3
3. 6
4. 執行錯誤 (RecursionError)

Answer: 3

Explanation:

本題考查遞迴 (Recursion) 的執行流程。

### 【追蹤過程】

1. 呼叫

f(3)

: 回傳

3 + f(2)

2. 呼叫

f(2)

: 回傳

2 + f(1)

3. 呼叫

f(1)

: 回傳

1 + f(0)

4. 呼叫

f(0)

: 遇到終止條件 ( $n==0$ )，回傳

0

### 【回溯計算】

$$f(1) = 1 + 0 = 1$$

$$f(2) = 2 + 1 = 3$$

$$f(3) = 3 + 3 = 6$$

這其實就是在計算  $0+1+2+3$  的總和。

【APCS-迴圈追蹤】執行下列程式碼後，

`cnt`

的值為何？

```
cnt = 0
for i in range(1, 5):
    for j in range(i, 5):
        cnt += 1
print(cnt)
```

Quiz:

- 1. 10
- 2. 16
- 3. 20
- 4. 25

Answer: 1

Explanation:

本題考查巢狀迴圈 (Nested Loop) 的執行次數計算。

【解析】

外層迴圈

i

從 1 跑到 4。

內層迴圈

j

從

i

跑到 4。

- i=1 時: j=1,2,3,4 (共 4 次)
- i=2 時: j=2,3,4 (共 3 次)
- i=3 時: j=3,4 (共 2 次)
- i=4 時: j=4 (共 1 次)

總次數

$$cnt = 4 + 3 + 2 + 1 = 10$$

。

這類題目考驗您是否能冷靜地追蹤變數變化，是 APCS 的核心考題。