

02 您正在建立一個電子商務指令碼來接受使用者輸入並以逗點分隔格式輸出資料。

您撰寫了下列程式碼以接受輸入：

```
item = input("Enter the item name: ")
sales = int(input("Enter the quantity: "))
```

輸出必須符合下列需求：

- 以雙引號括住字串。
- 不以引號或其他字元括住數字。
- 以逗點分隔項目。

您需要完成符合需求的程式碼。

請問您可以使用哪兩個程式碼片段？(選擇兩項)

- ()A. print("{0},{1}".format(item, sales))
- ()B. print(item + ',' + sales)
- ()C. print("{0}, {1}".format(item, sales))
- ()D. print("'" + item + "', " + sales)

正確答案：C、D

解題分析：

1. 需求分析：目標輸出格式為 "ItemName", Quantity。
2. 選項 A : "{0},{1}" 輸出結果為 ItemName,Quantity (字串沒有雙引號，逗號後沒有空格)，不符需求。
3. 選項 B : item + ',' + sales 會引發 **TypeError**，因為 sales 是整數 (int)，不能直接與字串相加。
4. 選項 C : "{0}, {1}" 手動在 {0} 外加上雙引號。輸出結果為 "ItemName", Quantity，符合需求。
5. 選項 D : print 函式使用逗號分隔參數時，預設會加入一個空格。"'" + item + "','" 串接後變成 "ItemName",，接著印出 sales。結果為 "ItemName", Quantity，符合需求。

04 您正在為公司開發一個 Python 應用程式。

您需要接受使用者的輸入並將該項資訊輸出至使用者的畫面。

您從下列程式碼開始著手。

```
01 print("What is your name?")
02 _____
03 print(name)
```

請問您應該在第 02 行撰寫哪個程式碼？

- ()A. input("name")
- ()B. name = input()
- ()C. input(name)
- ()D. name = input

正確答案：B

解題分析：

1. 變數指派：第 03 行需要列印變數 name，因此第 02 行必須將使用者的輸入值 指派 (**Assign**) 給變數 name。
2. 選項 B : name = input() 正確地呼叫了輸入函式，並將回傳值存入 name。
3. 錯誤選項：
 - A：沒有將輸入值存入變數，第 03 行會報錯。
 - C：name 在此時尚未定義，會報 NameError。
 - D：將函式物件本身指派給變數，並未執行輸入動作。

型態轉換與計算 (eval)：第 06 題 對應 CH01 第 3 題。皆為透過 eval() 計算年齡與出生年份的題型。

(重覆) 但問的方式不同

06 您正在撰寫一個程式來計算使用者的約略出生年份。

此程式會要求使用者輸入使用者年齡(age)和目前年份(year)，然後在訊息(message)中輸入使用者的約略出生年份(born)。

您撰寫了以下這段程式碼。加上行號僅為參考之用。

```
01 age = input("Enter your age: ")
02 year = input("Enter the four-digit year: ")
03 born = eval(year) - eval(age)
04 message = "You were born in " + str(born)
05 print(message)
```

請回答下列問題：

(1) 請問第 01 行的 age 是哪種資料類型?	(2) 請問第 03 行的 born 是哪種資料類型?	(3) 請問第 05 行的 message 是哪種資料類型?
()A. int ()B. str ()C. float ()D. bool	()A. int ()B. str ()C. float ()D. bool	()A. int ()B. str ()C. float ()D. bool

正確答案：(1) B (str) (2) A (int) (3) B (str)

解題分析：

1. 第 01 行：input() 函式回傳的永遠是 字串 (str)。
2. 第 03 行：eval() 會計算字串運算式的值。整數減整數的結果仍為 整數 (int)。
3. 第 04-05 行："..." + str(born) 是字串串接運算，結果為 字串 (str)。

字典查詢錯誤：第 07 題 對應 CH03 第 9 題。考驗字典鍵值型態不符（字串與整數比較）導致的查詢失敗，前者情境為找尋「會議室」，後者為找尋「學生」（重覆）學生名改成房間名

07 您建立了下列程式來找出會議室並顯示房間名稱。

同事回報指出，當輸入值 1 或 2 時，程式輸出為 Room does not exist.。您必須對此程式進行疑難排解。

請評估這段程式碼，然後從每個下拉式清單中選取正確的選項以回答問題。

```
01 rooms = {1: 'Foyer', 2: 'Conference Room'}
02 room = input('Enter the room number: ')
03 if not room in rooms:
04     print('Room does not exist.')
05 else:
06     print("The room name is " + rooms[room])
```

請問哪兩種資料類型儲存在第 01 行的 rooms 字典中？	2. 請問第 02 行的 room 是哪種資料類型？	3. 為什麼第 03 行無法尋找房間？
()A. 布林值和字串 ()B. 浮點數和布林值 ()C. 整數和字串 ()D. 浮點數和整數	()A. bool ()B. float ()C. int ()D. String	()A. 無效的語法 ()B. 不符合的資料類型 ()C. 命名錯誤的變數

正確答案：

1. C (整數和字串)
2. D (String)
3. B (不符合的資料類型)

解題分析：

1. 字典內容：1 (Key) 是整數，'Foyer' (Value) 是字串。故選 C。
2. 輸入類型：input() 取得的是字串。故選 D。
3. 錯誤原因：字典的鍵 (Key) 是整數 1，但變數 room 是字串 "1"。在 Python 中 "1" != 1，因此找不到該鍵。這是資料類型不匹配的問題。故選 B。

08 您撰寫了以下這段程式碼：

```
import datetime
d = datetime.datetime(2017, 4, 7)
print('{:%B-%d-%y}'.format(d))
```

執行程式。請問輸出為何？

- ()A. 04-07-17
()B. 2017-APRIL-07
()C. APRIL-07-17
()D. 04-07-2017

正確答案：C

解題分析：

1. %B：完整的月份名稱（如 April，題目顯示為全大寫 APRIL）。
2. %d：兩位數的日期（07）。
3. %y：兩位數的年份（17）。
4. 格式字串：{:%B-%d-%y} 組合為 Month-Day-Year，即 APRIL-07-17。

09 某間食品公司需要一套簡易程式，讓他們的客服中心能夠用來輸入新咖啡種類的問卷調查資料。

此程式必須執行下列工作：

- 接受輸入 (rating)。
傳回五星評比標準的平均評分 (average)。
將輸出四捨五入到兩位小數。

```
sum = count = done = 0
average = 0.0
while done != -1:
    rating = ____(1)____
    if rating == -1:
        break
    sum += rating
    count += 1
average = float(sum / count)
____(2)____ ("The average star rating for the new coffee is: " + ____(3)____)
```

(1) 的程式碼應為	(2) 的程式碼應為	(3) 的程式碼應為
()A. float(input("Enter next rating (1-5, -1 for done)")) ()B. input("Enter next rating (1-5, -1 for done)") ()C. input("Enter next rating (1-5, -1 for done)") ()D. print("Enter next rating (1-5, -1 for done)") input ...	()A. console.input ()B. output ()C. print ()D. printline	()A. {average, '.2f'} ()B. format.average.{2d} ()C. format(average,'.2d')) ()D. format(average, '.2f'))

正確答案：(1) A (2) C (3) D

解題分析：

1. 輸入處理 (1)：輸入必須轉換為數值才能進行加法運算，選項 A 使用 float(input(...)) 是正確的語法。
2. 輸出函式 (2)：Python 3 使用 print 函式輸出資料。
3. 格式化 (3)：題目要求四捨五入到兩位小數。format(average, '.2f') 是正確用法。選項 D 補上了與前方字串串接時對應的右括號（雖然程式碼片段有點亂，但 D 是唯一符合 .2f 格式且語法接近正確的選項）。

10 您為公司開發了一個 Python 應用程式。

您需要完成程式碼，好讓 `print` 陳述式正確。

請從每個下拉式清單中選取正確的程式碼片段以完成程式碼。

作答區：

```
numList = [1, 2, 3, 4, 5]
alphaList = ["a", "b", "c", "d", "e"]
____(1)____
    print("The value in numList are equal to alphaList")
____(2)____
    print("The value in numList are not equal to alphaList")
```

(1) 的程式碼應為	(2) 的程式碼應為
()A. if numList = alphaList:	()A. else:
()B. if numList == alphaList:	()B. elif:
()C. if numList += alphaList:	()C. elseif:

- 相等比較 (1)：在 Python 中，比較兩個物件是否相等使用 `==` 運算子。`if numList == alphaList:` 是正確的語法。
- 條件分支 (2)：當 `if` 條件不成立時，執行的區塊使用 `else`。`elif` 需要接續條件式，`elseif` 不是 Python 關鍵字。

12 您正在建立一個比較數字的 Python 程式。

您撰寫了以下這段程式碼。加上行號僅為參考之用。

```
01 num1 = eval(input("Please enter the first number: "))
02 num2 = eval(input("Please enter the second number: "))
03 if num1 == num2:
04     print("The two numbers are equal.")
05 if num1 <= num2:
06     print("Number 1 is less than number 2.")
07 if num1 > num2:
08     print("Number 1 is greater than number 2.")
09 if num2 = num1:
10     print("The two numbers are the same.")
```

對於下列每一項敘述，請選取【正確】或【錯誤】

作答區：

- 第 04 行的 `print` 陳述式只有在這兩個數字的值相等時才會列印。 (O)
- 第 06 行的 `print` 陳述式只有在 `num1` 小於 `num2` 時才會列印。 (X)
- 第 08 行的 `print` 陳述式只有在 `num1` 大於 `num2` 時才會列印。 (O)
- 第 09 行的陳述式是無效的比較。 (O)

正確答案：

- 正確 (O)
- 錯誤 (X)
- 正確 (O)
- 正確 (O)

解題分析：

- 敘述 1：第 03 行使用 `==` 進行相等比較，邏輯正確。
- 敘述 2：第 05 行使用的是 `<=` (小於或等於)。如果 `num1` 等於 `num2`，此行也會列印，並非「只有在小於時」。
- 敘述 3：第 07 行使用 `>` (大於)，邏輯正確。
- 敘述 4：第 09 行使用的是單一個等號 `=`，這是 賦值 (Assignment) 運算子，不能用在 `if` 條件中做比較，會導致語法錯誤 (Syntax Error)。正確的比較應使用 `==`。

13 您正在建立一個函式來根據下列規則計算入場費 (admission_fee)

未滿 5 歲的任何人 = 免費入場
年滿 5 歲以上的任何在校學生 = 10 美元
5 至 17 歲的任何非在校人士 = 20 美元
超過 17 歲的任何非在校人士 = 50 美元

請從每個下拉式清單中選取正確的程式碼片段以完成程式碼。

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

```
    rate = 0
    if ____(1)____:
        rate = 10
    elif ____(2)____:
        if ____(3)____:
            rate = 20
    else:
        rate = 50
    return rate
```

(1) 程式碼應為	(2) 程式碼應為	(3) 程式碼應為
()A. age >= 5 and school == True: ()B. age >= 5 and age <= 17 ()C. age >= 5 and school == False	()A. elif age >= 5 and school == False: ()B. else age >= 5 and school == False: ()C. elif age >= 5 and school == True:	()A. if age >= 5 and school == True: ()B. if age >= 5 and school == False: ()C. if age <= 17:

正確答案： (1) A (2) A (3) C

解題分析：

- 邏輯分層 (1)：先處理「5 歲以上且是學生」的情況 (費用 10)。對應代碼 `if age >= 5 and school == True:`。
- 邏輯分層 (2)：接著處理「5 歲以上且非學生」的情況。使用 `elif`。對應代碼 `elif age >= 5 and school == False:`。
- 內部邏輯 (3)：在非學生的情況下，區分年齡。因為外層已經確認 `age >= 5`，所以內層只需檢查是否小於等於 17 (`age <= 17`)。若是，則費用 20；否則 (else) 費用 50。

14 您正在撰寫一個 Python 程式來判斷使用者輸入的數字 (num) 是一位數、兩位數，還是超過兩位數 (digits)。

```
num = int(input("Enter a number with 1 or 2 digits: "))
digits = "0"
____(1)____:
    digits = "1"
____(2)____:
    digits = "2"
____(3)____:
    digits = ">2"
print(digits + " digits.")
```

(1) 程式碼應為	(2) 程式碼應為	(3) 程式碼應為
()A. if num > -10 and num < 10: ()B. if num > -100 and num < 100:	()A. if num > -100 and num < 100: ()B. elif num > -100 and num < 100: ()C. if num > -10 and num < 10: ()D. elif if num > -10 and num < 10:	()A. else: ()B. elif:

正確答案： (1) A (2) B (3) A

解題分析：

- 一位數 (1)：範圍是 -9 到 9。條件為 `num > -10 and num < 10`。
- 兩位數 (2)：排除一位數後，檢查是否為兩位數 (-99 到 99)。條件為 `elif num > -100 and num < 100`。
- 超過兩位數 (3)：剩下的所有情況。使用 `else`。

15 您正在設計一個決策結構，以將學生的數字成績 (grade) 轉換成字母成績 (letter_grade)。

此程式必須依照下表的指定內容指派字母成績：

舉例來說，如果使用者輸入 90，輸出就應該是“ Your letter grade is A” 。同樣地，如果使用者輸入 89。輸出就應該是“ Your letter grade is B”

百分比範圍	字母成績
90 到 100	A
80 到 89	B
70 到 79	C
65 到 69	D
0 到 64	F

```
# Letter Grade Converter
grade = int(input("Enter a numeric grade"))
____(1):
    letter_grade = 'A'
____(2):
    letter_grade = 'B'
____(3):
    letter_grade = 'C'
____(4):
    letter_grade = 'D'
else:
    letter_grade = 'F'
print("Your letter grade is", letter_grade)
```

(1) 的程式碼應為	(2) 的程式碼應為	(3) 的程式碼應為	(4) 的程式碼應為
()A. if grade <=90:	()A. if grade <=90:	()A. if grade >70 :	()A. if grade >65 :
()B. if grade >=90:	()B. if grade >=90:	()B. if grade >=70:	()B. if grade >=65:
()C. elif grade >90:	()C. elif grade >90:	()C. elif grade >70:	()C. elif grade >65:
()D. elif grade >=90:	()D. elif grade >=90:	()D. elif grade >= 70:	()D. elif grade >= 65:

正確答案：

(1) B、(2) D、(3) D、(4) D

解題分析：

這是一個標準的階梯式 if-elif-else 結構。

- 首先檢查是否大於等於 90 (if grade >= 90)。
- 若不符合，再檢查是否大於等於 80 (elif grade >= 80)。
- 依此類推，直到檢查是否大於等於 65 (elif grade >= 65)。
- 最後剩下的就是 F (else)。

16 您正在撰寫程式碼來根據下列準則傳回字母成績：

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。

#Letter Grade Converter

```
Grade = int(input("Enter a numeric grade"))
```

```
____(1) grade < 100:
____(2) grade >= 90:
    print("Your grade is A.")
____(3) grade >= 80:
    print("Your grade is B.")
____(4) grade < 80 ____(5) grade > 69:
    print("Your grade is C.")
____(6):
    print("Your grade is failing.")
____(7):
    print("Invalid grade entered.")
```

(1) 程式碼	(2) 程式碼	(3) 程式碼	(4) 程式碼	(5) 程式碼	(6) 程式碼	(7) 程式碼
()A. elif						
()B. if						
()C. for						
()D. else						

正確答案：

(1) B、(2) B、(3) A、(4) A、(5) B、(6) D、(7) D

解題分析：

- 外部驗證 (1, 7)：最外層先檢查成績是否在有效範圍內 (if grade < 100)，否則視為無效輸入 (else)。
- 內部邏輯 (2, 3)：進入範圍後，開始判斷等級。第一個判斷用 if，後續用 elif。
- 複合條件 (4, 5)：針對 C 等級 (70-79)，題目使用了明確的上下限檢查 elif grade < 80 and grade > 69 (意即 70-79)。連接兩個條件需使用 and。
- 不及格 (6)：剩餘的情況使用 else。

17 您正在為公司開發一個 Python 應用程式。您撰寫了以下這段程式碼：

```
numList = [1, 2, 3, 4, 5]
alphaList = ["a", "b", "c", "d", "e"]
print(numList is alphaList)
print(numList == alphaList)
numList = alphaList
print(numList is alphaList)
print(numList == alphaList)
```

請評估這段程式碼，然後從每個下拉式清單中選取正確的選項以回答問題。

請問第一個 `print` 後面顯示的內容為何? `False`

請問第二個 `print` 後面顯示的內容為何? `False`

請問第三個 `print` 後面顯示的內容為何? `True`

請問第四個 `print` 後面顯示的內容為何? `True`

正確答案： `False`、`False`、`True`、`True`

解題分析：

1. 初始狀態：`numList` 和 `alphaList` 分別指向記憶體中兩個不同的清單物件，內容也不同。
 - `is` (物件身分比較) : `False`。
 - `==` (內容值比較) : `False`。
2. 賦值操作：`numList = alphaList`。這將 `numList` 變數指向 `alphaList` 目前所指向的同一個記憶體物件。
3. 賦值後狀態：兩者指向同一物件。

`is` : `True`。

`==` : `True`。

18 您正在撰寫一個函式來執行數字的安全除法。

您需要確保分母和分子會傳遞給函式，而且分母不為零。

請從每個下拉式清單中選取正確的程式碼片段以完成程式碼。

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

(1) 程式碼應為

- ()A. if numerator is None or denominator is None:
- ()B. if numerator is None and denominator is None:
- ()C. if numerator = None or denominator = None:
- ()D. if numerator = None and denominator = None:

(2) 程式碼應為

- ()A. elif denominator == 0:
- ()B. elif denominator = 0:
- ()C. elif denominator != 0:
- ()D. elif denominator in 0:

正確答案： (1) A (2) A

解題分析：

1. 檢查遺漏值 (1)：如果分子「或」分母其中之一為 `None`，則資料不全。使用 `or` 連接條件。語法為 `if numerator is None or denominator is None`。
2. 檢查分母為零 (2)：分母不能為 0。使用 `==` 進行比較。因為是第二個條件，使用 `elif`。語法為 `elif denominator == 0`。

19 您正在撰寫一個程式來計算 DVD 租借金額。費用結構如下所示：

- 預設費用 (`cost_per_night`) 是每晚 1.59 美元。
- 如果客戶在晚上 8 點過後歸還 DVD，則需支付一晚的費用。
- 星期日 (Sunday) 租借的 DVD 可享整個租借期間七折優惠。
- 星期四 (Thursday) 租借的 DVD 可享整個租借期間五折優惠。

```
ontime = input("Was the video returned before 8 PM? Y or N").lower()
days_rented = int(input("How many days was the video rented?"))
weekday = input("What day was the video rented?").capitalize()
cost_per_night = 1.59
if ontime ____(1____:
    days_rented += 1
if weekday ____(2____:
    total = (days_rented * cost_per_night) * 0.7
elif weekday ____(3____:
    total = (days_rented * cost_per_night) * 0.5
else:
    total = days_rented * cost_per_night
print("Cost of the DVD rental is: $", total)
```

(1) 的程式碼應為

- ()A. != "n":
- ()B. == "n":
- ()C. == " y":

(2) 的程式碼應為

- ()A. == "Sunday":
- ()B. >= "Sunday":
- ()C. is "Sunday":

(3) 的程式碼應為

- ()A. == "Thursday":
- ()B. <= "Thursday":
- ()C. is "Thursday":

正確答案：

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

解題分析：

1. 逾時歸還 (1)：ontime 輸入為 'n' 代表未準時 (Not on time)，需加收一天費用。比較運算子為 ==。
2. 星期日折扣 (2)：輸入已透過 capitalize() 轉為首字大寫。檢查是否為 "Sunday"。使用 ==。
3. 星期四折扣 (3)：同理，檢查是否為 "Thursday"。使用 ==。

20 您正在建立一個 Python 指令碼以評估文字輸入並檢查是否有大寫和小寫。

請問您應該依序使用哪四個程式碼片段來開發解決方案？

請將四個程式碼片段移至作答區中，然後按照正確的順序排列。

程式碼片段選項：

```
name = input("Enter your name: ")
```

正確的程式碼順序：

```
    elif name.upper() == name:  
        print(name, "is all upper case.")
```

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

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

```
if name.lower() == name:  
    print(name, "is all lower case.")
```

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

```
name = input("Enter your name: ")
```

```
if name.lower() == name:  
    print(name, "is all lower case.")
```

```
elif name.upper() == name:  
    print(name, "is all upper case.")
```

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

解題分析：

1. 輸入：首先必須取得使用者輸入 name。
2. 全小寫檢查：使用 if name.lower() == name: 檢查是否全為小寫。
3. 全大寫檢查：若非全小寫，使用 elif name.upper() == name: 檢查是否全為大寫。
4. 混合大小寫：若上述皆非，則為混合大小寫，使用 else: 處理。

21 您正在為某間零售商店撰寫 Python 程式，將庫存追蹤自動化。

您的第一項工作是讀取庫存交易的檔案。

該檔案包含前一天的銷售量，包括商品識別碼、價格和數量。 程式碼必須執行下列工作：

- 讀取並列印檔案的每一行。
- 忽略空白行。
- 讀取所有行之後關閉檔案。

您撰寫了以下這段程式碼。

```
01 inventory = open("inventory.txt", 'r')  
02 eof = False  
03 while eof == False:  
04     line = inventory.readline()  
05     ____(1)____  
06     ____(2)____  
07         print(line)  
08     else:  
09         print("End of file")  
10     eof = True  
11 inventory.close()
```

請問您應該針對第 05 行和第 06 行撰寫哪些程式碼？

A	B	C	D
05 if line != '\n': 06 if line != "":	05 if line != "": 06 if line != "":	05 if line != '\n': 06 if line != None:	05 if line != "": 06 if line != "\n":

正確答案：D

解題分析：

1. 讀取機制：readline() 在讀到檔案結尾 (EOF) 時會回傳 空字串 ""；若讀到空白行則回傳包含換行符號的字串 "\n"。
2. 第 05 行 (EOF 判斷)：此處對應第 08 行的 else (設定 eof=True)。因此，第 05 行必須判斷是否 尚未 到達檔案結尾。條件應為 if line != "" :。
3. 第 06 行 (過濾空白行)：題目要求忽略空白行。因此，只有當內容不是換行符號時才列印。條件應為 if line != "\n" :。
4. 綜合：選項 D 符合此邏輯。

22 您正在使用 Python 編寫數學公用程式的程式碼。

您正在撰寫一個函式來計算方根。

此函式必須符合下列需求：

- 如果 a 為非負數，則傳回 $a^{1/b}$
- 如果 a 為負偶數 (題目原意應指開偶數次方根)，則傳回 "Result is an imaginary number"
- 如果 a 為負奇數 (題目原意應指開奇數次方根)，則傳回 $-(-a)^{1/b}$

請從每個下拉式清單中選取正確的程式碼片段以完成程式碼。

```
def safe_root(a, b):  
    ___(1)___  
    answer = a ** (1/b)  
    ___(2)___  
    ___(3)___  
    answer = "Result is an imaginary number"  
    ___(4)___  
    answer = -(-a) ** (1/b)  
    return answer
```

(1) 程式碼應為	(2) 程式碼應為	(3) 程式碼應為	(4) 程式碼應為
()A. if $a \geq 0$:	()A. if $a \geq 0$:	()A. if $a \geq 0$:	()A. if $a \geq 0$:
()B. if $b \% 2 == 0$:	()B. elif $b \% 2 == 0$:	()B. if $b \% 2 == 0$:	()B. if $b \% 2 == 0$:
()C. else:	()C. else:	()C. else:	()C. else:
()D. elif:	()D. elif:	()D. elif:	()D. elif:

正確答案：(1) A (2) C (3) B (4) C

解題分析：

第一層判斷：正數或零

- 邏輯：如果底數 a 是非負數 ($a \geq 0$)，直接計算。
- 代碼位置 (1)：對應選項 A. if $a \geq 0$:

第二層結構：處理負數

- 邏輯：如果上方條件不成立 (即 a 是負數)，則進入此區塊。這是一個互斥的條件。
- 代碼位置 (2)：對應選項 C. else:

內層判斷：負數開偶數次方根 (虛數)

- 邏輯：在 a 為負數的情況下，檢查次方根 b 是否為偶數。如果是偶數 (例如開平方)，結果會是虛數。
- 判斷方式：利用取餘數運算子 $\%$ ，若 $b \% 2 == 0$ 代表是偶數。
- 代碼位置 (3)：對應選項 B. if $b \% 2 == 0$:

內層其餘狀況：負數開奇數次方根 (實數)

- 邏輯：如果 b 不是偶數 (即奇數，例如開立方)，負數是可以開根號的 (例如 $\sqrt[3]{-8} = -2$)。這是內層 if 的對立面。
- 代碼位置 (4)：對應選項 C. else:

24 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 (Comment)，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？

- ()A. 在任何一行的 // 後面放置附註。
- ()B. 在任何程式碼片段的 * 和 */ 之間放置附註。
- ()C. 在任何一行的 # 後面放置附註。
- ()D. 在任何程式碼片段的 `` 之間放置附註。

正確答案：C

解題分析：

- Python 註解：Python 使用 # 符號來標示單行註解。
- 錯誤選項：
 - // : C++, Java 等語言的單行註解 (Python 中是整數除法)。
 - /* */ : C, CSS 等語言的區塊註解。
 - `` : HTML 註解。

25 您建立了下列 Python 函式來計算數字的次方。加上行號僅為參考之用。

```
01 # The calc_power function calculates exponents  
02 # x is the base  
03 # y is the exponent  
04 # The value of x raised to the y power is returned  
05 def calc_power(x, y):  
06     comment = "# Return the value"  
07     return x ** y # raise x to the y power
```

對於下列每一項敘述，請選取 [正確] 或 [錯誤]。

作答區：

- ()Python 不會檢查第 01 到 04 行的語法。
- ()第 02 和 03 行的井字號 (#) 是選擇性的。
- ()第 06 行的字串將解釋為註解。
- ()第 07 行包含內嵌註解。

正確答案：

- 正確 (O) (註解會被直譯器忽略)
- 錯誤 (X) (註解必須以 # 開頭，否則視為程式碼)

3. 錯誤 (X) (引號內的 # 視為字串內容)
4. 正確 (O) (程式碼後方的 # ... 是合法的內嵌註解)

26 您建立了下列 Python 函式來計算數字的次方。加上行號僅為參考之用。 (此題與第 25 題完全相同，僅題目編號不同)

```
01 # The calc_power function calculates exponents
02 # x is the base
03 # y is the exponent
04 # The value of x raised to the y power is returned
05 def calc_power(x, y):
06     comment = "# Return the value"
07     return x ** y # raise x to the y power
```

對於下列每一項敘述，請選取【正確】或【錯誤】。

1. Python 不會檢查第 01 到 04 行的語法。 ()
2. 第 02 和 03 行的井字號 (#) 是選擇性的。 ()
3. 第 06 行的字串將解釋為註解。 ()
4. 第 07 行包含內嵌註解。 ()

正確答案：

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

27 某位朋友要求您重構並記錄下列 Python 程式碼:

```
value1 = 9
value2 = 4
answer = (value1 % value2 * 10) // 2.0 ** 3.0 + value2
```

您執行程式碼。請問結果為何?

- ()A. 顯示值 129。
- ()B. 發生語法錯誤。
- ()C. 顯示值 5.667。
- ()D. 顯示值 5.0。

正確答案：D

解題分析：

1. 運算子優先級：指數 ** > 乘除取餘 *, /, //, % > 加減 +, -。
2. 步驟 1(指數) : $2.0^{**} 3.0 = 8.0$ 。
3. 步驟 2(括號與取餘) : $value1 \% value2 = 9 \% 4 = 1$ 。
4. 步驟 3(乘法) : $1 * 10 = 10$ 。
5. 步驟 4(整數除法) : $10 // 8.0$ 。Python 中浮點數參與整數除法，結果為浮點數，但數值取整。 $\$10 / 8 = 1.25\$$ ，取整為 1.0。
6. 步驟 5(加法) : $1.0 + 4 = 5.0$ 。

28 您正在撰寫一個 Python 程式來評估算數公式。

此公式描述為 b 等於 a 乘以負一之後再平方。其中 a 是即將輸入的成績，而 b 是結果。

您所建立的程式碼片段如下。加上行號僅為參考之用。

```
01 a = eval(input("Enter a number for the question:"))
02 b =
```

您需要確保結果正確。

請將左側清單中適當的程式碼片段搬移至右側的正確位置，以完成 02 行的程式碼。

程式碼片段選項：(,), **, **2, 2, a

正確答案：

b = (-a) ** 2

解題分析：

1. 公式解讀：「a 乘以負一」即 $-a$ 。「再平方」即對 $-a$ 進行平方。
2. 優先順序：指數運算優先於負號 (Unary negation)。
 - 若寫 $-a^{**} 2$ ，會解釋為 $-(a^{**} 2)$ 。
 - 若要計算 $(-a)^2$ ，必須加上括號： $(-a)^{**} 2$ 。

29 某間銀行必須產生一份報表來顯示所有客戶每天的平均餘額。

這份報表必須截斷餘額的小數部分。

請問哪兩個程式碼片段可達成目標？

- ()A. average_balance = int(total_deposits / number_of_customers)
- ()B. average_balance = total_deposits // number_of_customers
- ()C. average_balance = total_deposits ** number_of_customers
- ()D. average_balance = float(total_deposits // number_of_customers)

正確答案：A、B

解題分析：

1. 目標：計算平均值並移除小數部分 (取整數)。
2. 選項 A : total / count 得到浮點數平均值，int() 強制轉型會無條件捨去小數。正確。
3. 選項 B : // 是整數除法 (Floor Division)，直接計算商的整數部分。正確。
4. 選項 C : ** 是指數運算，錯誤。
5. 選項 D : 先做整數除法得到整數 (如 10)，再轉為 float(10.0)。雖數值正確，但型態變為 float，且通常題目要求「截斷」隱含整數型態，A、B 為最佳解。

30 某間銀行要將他們舊有的銀行交易程式碼轉至 Python。該銀行雇用您來記錄已移轉的程式碼。

請問哪個文件記錄語法是正確的？

()A.

```
def get_balance():
    /* Returns the current balance of the bank account */
    return balance
```

()B.

```
// Returns the current balance of the bank account
def get_balance():
    return balance
```

()C.

```
'Returns the current balance of the bank account
def get_balance():
    return balance
```

()D.

```
# Returns the current balance of the bank account
def get_balance():
    return balance
```

正確答案：D

解題分析：

1. 語法檢查：
 - A、B 使用了 C/Java 風格的註解 /* */, ///, 在 Python 中無效。
 - C 使用了單引號開頭但未閉合，語法錯誤。
 - D 使用了 #，這是合法的 Python 單行註解。雖然標準文件字串 (Docstring) 建議使用 """..."""，但在給定選項中，只有 D 是語法正確且能運作的代碼。

30 請從每個下拉式清單中選取正確的選項以回答有關記錄 Python 程式碼的問題。

<p>1. 請問哪些字元代表單行文件字串 (Docstring) 的開頭和結尾？</p> <p>()A. 單引號 (') ()B. 雙引號 (") ()C. 兩個雙引號 (""") ()D. 三個雙引號 ("""")</p>	<p>2. 在記錄函式時，文件字串的標準位置在哪裡？</p> <p>()A. 在程式檔案的開頭 ()B. 在函式定義後的第一行 (縮排內) ()C. 在程式檔案的結尾</p>	<p>3. 請檢閱下列函式：</p> <pre>def cube(n): """ Returns the cube of number n """ return n*n*n</pre>
--	---	---

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))

正確答案：

1. **D**
2. **B** (根據 Python 規範 PEP 257，Docstring 位於函式定義後的第一行)
3. **C**

解題分析：

1. **Docstring 語法**：Python 使用三個雙引號 """...""" 來定義多行字串或文件字串。
2. **位置**：函式的文件字串必須緊接在 def 語句之後，且要有正確的縮排。
3. **屬性存取**：Python 的物件（如函式、模組、類別）都有一個內建屬性 __doc__ (注意前後各兩個底線)，用來儲存該物件的文件字串說明。

32 您撰寫了以下這段程式碼：

```
a = 'Config1'
print(a)
b = a
a += 'Config2'
print(a)
print(b)
```

請評估這段程式碼，然後從每個下拉式清單中選取正確的選項以回答問題。

- (1) 請問第二個 print 後面顯示的內容為何？**
- ()A. Config1
()B. Config1Config2
()C. Config2
()D. Config2Config1

- (2) 請問第三個 print 後面顯示的內容為何？**

- ()A. Config1
()B. Config1Config2
()C. Config2
()D. Config2Config1

正確答案：(1) B (2) A

解題分析：

1. 字串不可變性 (**Immutability**)：Python 的字串是不可變物件。
2. 變數參照：
 - `b = a`：此時 `b` 指向與 `a` 相同的記憶體位置（即字串 '`Config1`'）。
 - `a += 'Config2'`：由於字串不可修改，Python 會建立一個 **新的字串物件** '`Config1Config2`'，並讓 `a` 指向這個新物件。
3. 結果：`b` 仍然指向原始的物件 '`Config1`'，而 `a` 指向了新產生的物件 '`Config1Config2`'。

33 您撰寫了以下這段程式碼：

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

您執行程式碼。請問輸出值為何？

- ()A. [1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]
()B. [[1, 2, 3, 4], [1, 2, 3, 4], [1, 2, 3, 4]]
()C. [3, 6, 9, 12]
()D. [[1, 2], [3, 4], [1, 2], [3, 4], [1, 2], [3, 4]]

正確答案：A

解題分析：

1. 清單加法 (+)：`list_1 + list_2` 會將兩個清單串接成一個新清單 [1, 2, 3, 4]。
2. 清單乘法 (*)：`list * n` 會將清單內容重複 `n` 次並串接。
 - `[1, 2, 3, 4] * 3` 結果為 [1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]。

format() 參數索引：第 34 題 對應 CH04 第 4 題。皆使用 {1} and {0} and {2} 來排列三個單字。

題目不一樣

34 請檢閱下列程式碼：

```
x = "oranges"
y = "apples"
z = "bananas"
data = "{1} and {0} and {2}"
print(data.format(z, y, x))
```

請問 `print` 陳述式的輸出為何？

- ()A. bananas and oranges and apples
()B. apples and bananas and oranges
()C. apples and oranges and bananas
()D. oranges and apples and bananas

正確答案：B

解題分析：

1. format 參數索引： <code>format(z, y, x)</code> 中： 索引 0 是 <code>z ("bananas")</code> 。 索引 1 是 <code>y ("apples")</code> 。 索引 2 是 <code>x ("oranges")</code> 。	2. 格式化字串："{1} and {0} and {2}" 依序填入參數： {1} -> apples {0} -> bananas {2} -> oranges	3. 結果：apples and bananas and oranges。
---	---	---------------------------------------

字串與清單配量 (Slicing)：第 35 題與第 38 題 對應 CH03 第 5 題。都是針對 abcdefghijklmnopqrstuvwxyz 字串進行 [3:15:3] 等切片操作。

35 您需要識別針對下列順序結構執行各種配量 (Slicing)

作業的結果: `alph = "abcdefghijklmnopqrstuvwxyz"`

請將左側清單中適當的結果移至右側正確配量作業。

作答區：

1. `alph[3:6] -> _____`
2. `alph[:6] -> _____`

正確答案：

1. `def`
2. `abcdef`

解題分析：

1. <code>alph[3:6]</code> ： 起始索引 3 是 'd' (a=0, b=1, c=2, d=3)。 結束索引 6 是 'g' (不包含)。 取出索引 3, 4, 5 的字元：def。	2. <code>alph[:6]</code> ： 省略起始索引，預設從 0 開始。 結束索引 6 (不包含)。 取出索引 0 到 5 的字元：abcdef。
---	---

36 某間公司要求您為造成薪資問題的部分程式碼偵錯。

他們要求您找出薪資錯誤的來源。

已經宣告的變數如下：

```
employee_pay = [15000, 120000, 35000, 45000]
count = 0
sum = 0
```

下列程式碼有兩個錯誤：

```
for index in range(0, len(employee_pay)-1):
    count += 1
    sum += employee_pay[index]
average = sum // count
print("The total payroll is:", sum)
```

請問您應該使用哪個程式碼來修正錯誤？

- | | |
|--|--|
| (1) 的程式碼應為 (修正 range 範圍) | (2) 的程式碼應為 (修正平均計算) |
| ()A. (size(employee_pay)):
()B. (size(employee_pay)-1):
()C. (len(employee_pay)+1):
()D. (len(employee_pay)): | ()A. sum / count
()B. sum ** count
()C. sum * count |

正確答案：(1) D (2) A

解題分析：

- 迴圈範圍錯誤：原程式碼 `range(0, len(employee_pay)-1)` 會少跑一次迴圈（因為 `range` 不包含結束值）。要遍歷所有元素，結束值應為清單長度。故修正為 `range(len(employee_pay))` 或 `range(0, len(employee_pay))`。
- 平均計算錯誤：原程式碼 `sum // count` 使用整數除法，會捨去小數部分導致精確度流失。計算平均薪資通常需要保留小數，應改為浮點數除法 `/`。

37 您為公司開發了一個 Python 應用程式。

有一份名為 `employees` 的清單包含 200 個員工姓名，最後五位員工是公司管理階層。

您需要對清單進行篩選以顯示不包含管理階層的所有員工。

請問您應該使用哪兩個程式碼片段？(請選擇 2 個答案)

- ()A. `employees[1:-5]`
()B. `employees[1:-4]`
()C. `employees[0:-4]`
()D. `employees[:-5]`
()E. `employees[0:-5]`

正確答案：D、E

解題分析：

- 需求：從第一位員工取到倒數第五位之前（排除最後 5 位）。
- 索引概念：
起始索引：從頭開始，可以是 0 或省略。
結束索引：倒數第 5 個位置是 -5。切片 `[start:end]` 不包含 `end`，所以寫 -5 正好會排除最後 5 個元素（索引 -5, -4, -3, -2, -1）。
- 選項分析：
D: `employees[:-5]` (省略起始，預設為 0，正確)。
E: `employees[0:-5]` (明確指定起始為 0，正確)。

39 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。

請從每個下拉式清單中選取正確的選項以完成程式碼。

```
x = "Hello World"
____(1)____ x != "QUIT":
    num = 0
    ____(2)____ char ____(3)____ x:
        num += 1
    print(num)
x = input("Enter a new word or QUIT to exit: ")
```

- | | | |
|--------------------------------------|--------------------------------------|-----------------------------------|
| (1) 的程式碼應為 | (2) 的程式碼應為 | (3) 的程式碼應為 |
| ()A. for
()B. if
()C. while | ()A. for
()B. if
()C. while | ()A. and
()B. or
()C. in |

正確答案：(1) C (2) A (3) C

解題分析：

- 外層迴圈 (1)：這是一個條件式迴圈，只要輸入不是 "QUIT" 就繼續執行。應使用 `while`。
- 內層迴圈 (2, 3)：要計算字串 `x` 中的字元數，需要遍歷字串中的每個字元 `char`。這是迭代迴圈，應使用 `for` 迴圈配合 `in` 關鍵字。語法為 `for char in x:`。

40 您正在建置一個 Python 程式來顯示 2 到 100 之間的所有質數(質數是指只有該數字本身和 1 能整除的任何數字)。

請將左側清單中適當的程式碼片段移至右側的正確位置,以完成程式碼。每個程式碼片段可能只使用一次,也可能使用多次,甚至完全用不到。

請注意:每一個放置正確的程式碼都能獲得部分分數。

程式碼片段：

作答區：

break

p = 2
while p <= 100:
 is_prime = True

continue

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

p = p + 1

p = 2
is_prime = True
while p <= 100:

p = 2
while p <= 100:
 is_prime = True

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

break

if is_prime == True:
 print(p)

p = p + 1

解題分析：

第一個空格 (位於 if $p \% i == 0$: 區塊內) :

- 答案 : break
- 解析 : 當 $p \% i == 0$ 成立時, 代表 p 能被 i 整除, 所以 p 不是質數。
 1. 程式將 `is_prime` 設為 False。
 2. 既然已經確定不是質數了, 就不需要再檢查剩下的數字 (例如檢查 10 時, 發現能被 2 整除, 就不需要再檢查是否能被 5 整除了)。
 3. 因此, 使用 `break` 指令直接「跳出」內層的 `for` 迴圈, 可以節省運算資源。

第二個空格 (位於 while 迴圈的最後) :

- 答案 : `p = p + 1`
- 解析 : 這是 while 迴圈的計數器更新步驟。
 1. 程式檢查完數字 p 是否為質數後, 必須繼續檢查下一個數字。
 2. 如果不加這行, p 的值永遠不變, $p \leq 100$ 的條件永遠成立, 程式會陷入無窮迴圈 (死當)。
 3. 因此, 必須加上 `p = p + 1` 讓 p 遞增, 直到超過 100 才會停止程式。

42 請檢閱下列程式碼片段:

```
product = 2
n = 5
while (n != 0):
    product *= n
    print(product)
    n -= 1
    if n == 3: break
```

請問此程式碼會列印幾行輸出?

- () 1
() 2
() 3
() 5

正確答案 : 2

解題分析 :

1. 第一圈 :
 - n 為 5 (不等於 0), 進入迴圈。
 - $product = 2 * 5 = 10$ 。
 - 列印 10 (第 1 行)。
 - n 減 1 變為 4。
 - 檢查 `if n == 3` ($4 \neq 3$), 繼續。
2. 第二圈 :
 - n 為 4 (不等於 0), 繼續。
 - $product = 10 * 4 = 40$ 。
 - 列印 40 (第 2 行)。
 - n 減 1 變為 3。
 - 檢查 `if n == 3` ($3 == 3$), 條件成立, 執行 `break` 跳出迴圈。
3. 結論 : 共列印了 2 行數字。

43 某間遊戲開發公司需要設法找出清單中包含特定字母的單字數。

請從每個下拉式清單中選取正確的程式碼片段以完成程式碼。

```
# Function accepts list of words and letter to search for.  
# Returns count of the number of words that contain that letter.  
def count_letter(letter, word_list):  
    count = 0  
    for ____(1)____  
        if ____(2)____  
            count += 1  
    return count  
  
word_list = []  
# word_list is populated by the readWords() function. Code not shown.  
letter = input("which letter would you like to count")  
letter_count = count_letter(letter, word_list)  
print("There are: ", letter_count, " word that contain ", letter)
```

(1) 的程式碼應為

- ()A. word list in word:
- ()B. word in word_list:
- ()C. word == word_list:
- ()D. word is word_list:

(2) 的程式碼應為

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

正確答案： (1) B (word in word_list:) (2) D (letter in word:)

解題分析：

1. 遍歷清單 (1)：使用 `for` 迴圈逐一取出清單 `word_list` 中的元素，標準語法為 `for 變數 in 清單:`。
2. 檢查包含 (2)：要檢查「字母」是否在「單字」中，使用 `in` 運算子。語法為 `if 子字串 in 字串:`，即 `if letter in word:`。

44 您在評估下列程式碼時發現錯誤。加上行號僅為參考之用。

您必須更正第 03 行和第 06 行的程式碼。

請評估這段程式碼，然後從每個下拉式清單中選取正確的選項以回答問題。

```
01 numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
02 index = 0  
03 ____(1)____  
04     print(numbers[index])  
05     if numbers[index] == 6:  
06         ____(2)____  
07     else:  
08         index += 1
```

(1) 請問您應該在第 03 行使用哪個程式碼片段？

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

(2) 請問您應該在第 06 行使用哪個程式碼片段？

- ()A. if numbers[index] == 6
- ()B. break

正確答案：

(1) A (while (index < 10):)

(2) break (根據題庫邏輯與類似題型 46，這裡應填入 `break` 跳出迴圈)，

依據題意修正與上下文推斷，此處應為跳出指令，雖然題庫選項顯示有誤，但邏輯上應選 `break` 或類似功能。

解題分析：

1. 迴圈條件 (1)：清單長度為 10 (索引 0-9)。要遍歷清單，索引必須小於 10。正確語法為 `while (index < 10):`。
2. 終止條件 (2)：程式邏輯為「找到 6 時停止」。因此在 `if numbers[index] == 6:` 成立時，應執行 `break` 結束搜尋。

巢狀迴圈 (for)：第 45 題 對應 CH05 第 11 題。皆為計算 2 到 12 (或 2 到 9) 的乘法組合程式。

45 您正在建立一套適用於小學孩童的互動式乘法表助手程式。

您需要完成一個名為 `times_tables` 的函式，以計算並顯示所有 2 到 12 的乘法表組合。

```
01  
02 #Displays time tables 2-12  
03 def times_tables():  
04     ____(1)____  
05     ____(2)____  
06         print(row * col, end=" ")  
07     print()  
08  
09 #main  
10 times_tables()
```

51 您正在撰寫一個 Python 程式來驗證員工識別碼。

識別碼的格式必須是 `ddd-dd-dddd`，而且只能包含數字和虛線。

如果格式正確，此程式必須列印 `True`；如果格式不正確，則列印 `False`。

作答區

```
____(1)____  
parts = ""  
____(2)____:  
    ____(3)____  
        employee_number = input("Enter employee number (ddd-dd-dddd): ")  
        parts = employee_number.split('-')  
        if len(parts) == 3:  
            if len(parts[0]) == 3 and len(parts[1]) == 2 and len(parts[2]) == 4:  
                if parts[0].isdigit() and parts[1].isdigit() and parts[2].isdigit():  
                    ____(4)____  
                    print(valid)
```

(1) 的程式碼應為	(2) 的程式碼應為	(3) 的程式碼應為	(4) 的程式碼應為
()A. <code>employee_number = ""</code> ()B. <code>employee_number = "123"</code>	()A. <code>while employee_number != ""</code> ()B. <code>while employee_number != "123"</code>	()A. <code>valid = False</code> ()B. <code>valid = True</code>	()A. <code>valid = False</code> ()B. <code>valid = True</code>

正確答案： (1) A (2) B (3) A (4) B

解題分析：

- 初始化 (1)：為了進入迴圈，必須先初始化變數 `employee_number`。通常設為空字串，或不影響後續邏輯的初始值。
- 迴圈條件 (2)：題目設計了一個迴圈來持續接收輸入，直到輸入特定值（此處題庫選項設定為 "123"）才停止。
- 預設狀態 (3)：在每次驗證開始前，先假設格式是錯誤的 (`valid = False`)。只有當所有檢查都通過時才改為 `True`。
- 驗證成功 (4)：經過 `split` 分割、長度檢查（3 段）、每段長度檢查（3, 2, 4 碼）、以及 `isdigit()` 數字檢查後，如果全數通過，則將狀態改為正確 (`valid = True`)。

52 您正在撰寫一個必須執行下列工作的 Python 函式：

- 接受清單和字串做為參數。
- 在清單中搜尋字串。
- 如果字串存在清單中，則列印一則訊息，指出找到字串，然後停止逐一查看清單。
- 如果字串不在清單中，則列印一則訊息，指出找不到字串。

請問您應該按什麼順序排列程式碼片段以完成此函式？

程式碼片段選項：

1. `def search(items, term):`
2. `2. for i in range(len(items)):`
3. `3. if items[i] == term:`
4. `4. print("{0} was found in the list.".format(term))`
5. `5. break`
6. `6. else:`
7. `7. print("{0} was not found in the list.".format(term))`

正確順序：

```
def search(items, term):  
    for i in range(len(items)):  
        if items[i] == term:  
            print("{0} was found in the list.".format(term))  
            break  
        else:  
            print("{0} was not found in the list.".format(term))
```

解題分析：

這題考查 Python 特有的 `for-else` 語法。

1. 函式定義：`def search(...)` 是函式的開頭。
2. 迴圈搜尋：`for i in range(...)` 用於遍歷清單。
3. 比對：`if items[i] == term:` 檢查是否找到目標。
4. 找到時：列印訊息 (`print`) 並使用 `break` 立即跳出迴圈。
5. Else 區塊：`for` 迴圈的 `else` 子句只有在迴圈 沒有被 `break` 中斷 的情況下（即完整跑完所有項目都沒找到）才會執行。這正好符合「找不到字串」的需求。

例外處理 (try-except) : 第 53 題 對應 CH08 第 1 題 。判斷 try、except 與 finally 子句的使用規則 。

題目相同，作答不同

53 對於下列每一項有關 try 陳述式的敘述，請選取 [Yes] 或 [No] 。

作答區：

- | | | |
|--|--------|-------|
| 1. try 陳述式可以包含一個或多個 except 子句。 | (V)Yes | ()No |
| 2. try 陳述式可以包含一個 finally 子句，但不含 except 子句。 | (V)Yes | ()No |
| 3. try 陳述式可以包含一個 finally 子句以及一個 except 子句。 | (V)Yes | ()No |
| 4. try 陳述式可以包含一個或多個 finally 子句。 | ()Yes | (V)No |

正確答案：

1. 正確 (O) - 用於捕捉不同類型的錯誤 (如 ValueError, TypeError) 。
2. 正確 (O) - try...finally 是合法結構，用於確保無論是否發生錯誤都執行清理工作，即使沒有捕捉錯誤 。
3. 正確 (O) - try...except...finally 是完整的錯誤處理結構 。
4. 錯誤 (X) - 一個 try 區塊只能配對 一個 finally 子句，且必須放在最後 。

54 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:

- 使用指定的名稱建立檔案。
- 將 "End of listing" 這段片語附加至檔案。

您需要完成符合需求的程式碼。

請從每個下拉式清單中選取正確的程式碼片段以完成程式碼。

```
import os
file = ____(1)____
____(2)____ ("End of listing")
file.close()
```

- | | |
|--|--|
| (1) 的程式碼應為
()A. open('myFile.txt', 'a')
()B. open('myFile.txt', 'r')
()C. open('myFile.txt', 'w') | (2) 的程式碼應為
()A. append
()B. file.add
()C. file.write
()D. write |
|--|--|

正確答案： (1) A (2) C

解題分析：

1. 開啟模式 (1)：題目要求「附加 (Append)」內容。
 - 'r'：唯讀。
 - 'w'：寫入（會覆寫/清空原檔案）。
 - 'a'：附加（寫在檔案末端，若檔案不存在則自動建立）。符合題目「建立」與「附加」的需求。
2. 寫入方法 (2)：Python 檔案物件寫入資料的方法是 write()。語法為 file.write("string") 。

55 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:

- 查看檔案是否存在
- 如果檔案存在，則顯示其內容。

請從每個下拉式清單中選取正確的程式碼片段以完成程式碼。

```
import os
if ____(1)____
    file = open('myFile.txt')
    ____(2)____
    file.close()
```

- | |
|--|
| (1) 的程式碼應為
()A. isfile('myFile.txt'):br/>()B. os.exist('myFile.txt'):br/>()C. os.find('myFile.txt'):br/>()D. os.path.isfile('myFile.txt'):br/> (2) 的程式碼應為
()A. output('myFile.txt')
()B. print(file.get('myFile.txt'))
()C. print(file.read())
()D. print('myFile.txt') |
|--|

正確答案： (1) D (2) C

解題分析：

1. 檢查存在 (1)：Python 中檢查檔案是否存在標準函式位於 os.path 模組下，方法為 os.path.isfile(path) 。
2. 讀取內容 (2)：讀取檔案全部內容的方法是 file.read()。配合 print() 將內容顯示在螢幕上。

57 請檢閱下列程式碼片段：

```
f = open("python.txt", "a")
f.write("This is a line of text.")
f.close()
```

對於下列每一項有關程式碼片段的敘述，請選取正確或錯誤。

1. 如果名稱為 python.txt 的檔案不存在，就會建立此檔案。()
2. 此檔案中的資料會被覆寫。()
3. 其他程式碼可以在這段程式碼執行完畢之後，開啟此檔案。()

正確答案：

1. 正確 (O) - 開啟模式 'a' (append) 若檔案不存在會自動建立。
2. 錯誤 (X) - 模式 'a' 是「附加」，新資料會寫在舊資料後面，不會覆寫。若要覆寫需用 'w'。
3. 正確 (O) - 檔案使用 close() 關閉後，釋放了資源鎖定，其他程式碼即可開啟與操作該檔案。

58 您正在建立一個接受使用者輸入的程式。

此程式必須將輸入轉型為整數，並且在無法轉型時正確處理錯誤。

請選取正確的程式碼片段以完成程式碼。

while True:

```
    ____(1)____
    x = int(input("Please enter a number: "))
    break
    ____(2)____ ValueError:
        print("Not a valid number. Try again...")
```

(1) 應填入	(2) 應填入
()A. try:	()A. try:
()B. else:	()B. else:
()C. except:	()C. except:
()D. raise:	()D. raise:
()E. finally:	()E. finally:

正確答案： (1) A (2) C

解題分析：

1. 例外處理結構：使用 try...except 區塊來處理可能發生的錯誤。
2. try 區塊 (1)：包含可能引發異常的程式碼 (int() 轉換)。
3. except 區塊 (2)：包含當特定異常 (ValueError) 發生時的處理邏輯（印出錯誤訊息，因迴圈未 break 而重試）。

59 您需要撰寫執行下列工作的程式碼：

- 呼叫 process() 函式。
- 如果 process() 函式擲回錯誤，則呼叫 logError() 函式。
- 呼叫 process() 函式之後一律呼叫 displayResult() 函式。

請將左側清單中適當的程式碼片段移至右側的正確位置，以完成程式碼。

程式碼片段選項：

assert:

except:

finally:

raise:

try:

 正確的程式碼結構：

```
try:
    process()
except:
    logError()
finally:
    displayResult()
```

解題分析：

1. 嘗試執行：放在 try: 區塊中 -> process()。
2. 錯誤處理：放在 except: 區塊中 -> logError()。
3. 總是執行：放在 finally: 區塊中 -> displayResult()。finally 的特性是無論是否發生異常，最後一定會被執行。

檔案讀寫模式：第 60 題 對應 CH04 第 9 題。測驗開啟文字檔時 w（寫入並覆寫）與 a（附加）模式的差異

60 您正在為學校開發一個 Python 應用程式。此應用程式必須讀取並寫入資料至文字檔案。

如果檔案不存在，此應用程式就必須建立檔案。

如果檔案含有內容，此應用程式就必須刪除內容。

請問您應該使用哪個程式碼片段？

- ()A. open("local_data", "w+")
- ()B. open("local_data", "r")
- ()C. open("local_data", "r+")
- ()D. open("local_data", "w")

正確答案：D

解題分析：

1. 需求分析：

- 建立檔案 (若不存在)。
- 刪除內容/覆寫 (若存在)。
- 寫入資料。

2. 模式 'w' (Write)：

- 若檔案不存在：建立新檔案。
- 若檔案存在：清空 (Truncate) 內容並從頭寫入。
- 這完全符合題目需求。

3. 其他選項：

- 'r'：唯讀，檔案必須存在。
- 'r+'：讀寫，檔案必須存在，不會自動清空。
- 'w+'：讀寫，會清空內容。雖然功能涵蓋需求，但若只需寫入（題目未提及需要讀取剛寫入的資料），'w' 是最精確的標準答案。

61 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 = 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。您執行程式碼。

請問以下哪一項敘述是正確的？

- ()A. 此程式碼將正確執行，不會發生錯誤。
- ()B. 此程式碼將執行，但會產生邏輯錯誤。
- ()C. 此程式碼將產生執行階段錯誤。
- ()D. 此程式碼將產生語法錯誤。

正確答案：A

解題分析：

1. 檔案開啟模式 w+ : open("out.txt", 'w+') 表示以「讀寫模式」開啟檔案。此模式的特性是：

- 如果檔案 存在，會先清空檔案內容 (Truncate)。
- 如果檔案 不存在，Python 會 自動建立 該檔案。

2. 執行流程：

- 即使 out.txt 原本不存在，程式碼也能成功建立並開啟它，因此不會觸發 IOError。
- try 區塊成功執行後，接著執行 else 區塊，進行檔案讀寫操作。

3. 結論：程式碼語法正確，且邏輯上能處理檔案不存在的情況，因此能正確執行。

62 某間自行車公司正在建立一個程式，讓客戶能夠記錄騎乘英里數。此程式會根據客戶所記錄的英里數傳送訊息。

您需要定義兩個必要的函式。

請問您應該針對第 01 行和第 04 行使用哪兩個程式碼片段？(請選擇 2 個答案)

```
01 ____(1)____  
02     name = input("What is your name? ")  
03     return name  
04 ____(2)____  
05     calories = miles * calories_per_mile  
06     return calories  
07 distance = int(input("How many miles did you bike this week?"))  
08 burn_rate = 50  
09 biker = get_name()  
10 calories_burned = calc_calories(distance, burn_rate)  
11 print(biker, ", you burned about", calories_burned, "calories.")
```

- ()A. def get_name():
- ()B. def get_name(biker):
- ()C. def get_name(name):
- ()D. def calc_calories():
- ()E. def calc_calories(miles, burn_rate):
- ()F. def calc_calories(miles, calories_per_mile):

正確答案：A、F

解題分析：

1. 第 01 行 (get_name)：
 - 觀察第 09 行的呼叫方式：biker = get_name()。
 - 呼叫時 沒有傳遞任何引數 (Argument)，因此函式定義時括號內不應有參數。
 - 故選 A。
2. 第 04 行 (calc_calories)：
 - 觀察第 10 行的呼叫方式：calc_calories(distance, burn_rate)，傳遞了兩個引數。
 - 觀察第 05 行的函式內容：使用了變數 miles 和 calories_per_mile 進行計算。
 - 因此函式定義的參數名稱必須與內部使用的變數名稱一致，分別對應傳入的 distance 和 burn_rate。
 - 故選 F。

63 某位同事撰寫了一個程式來輸入名稱至資料庫中。然而，此程式卻將每個名稱中的字母前後顛倒。

您需要撰寫一個 Python 函式來按照正確順序輸出名稱中的字元。

請從每個下拉式清單中選取正確的程式碼片段以完成程式碼。

```
def reverse_name(backward_name):  
    forward_name = ""  
    length = ____(1)____  
    while length >= 0:  
        forward_name += ____(2)____  
        length = length - 1  
    return forward_name  
print(reverse_name("nohtyp"))
```

(1) 的程式碼應為

- ()A. backward_name
- ()B. len(backward_name) - 1
- ()C. range(0, len(backward_name), -1)
- ()D. len(backward_name)

(2) 的程式碼應為

- ()A. backward_name[index]
- ()B. backward_name[length]
- ()C. backward_name[length+1]
- ()D. backward_name

正確答案：

(1) B

(2) B

解題分析：

1. 初始化索引 (1)：我們要從字串的最後一個字元開始往前讀取。
 - Python 的字串索引從 0 開始。
 - 長度為 N 的字串，最後一個字元的索引是 N - 1。
 - 因此起始索引應設為 len(backward_name) - 1。
2. 取得字元 (2)：在 while 迴圈中，變數 length 被用作當前的索引指標 (Pointer)。
 - 要取出當前指向的字元，語法為 字串[索引]，即 backward_name[length]。
 - 取出後串接到 forward_name，然後索引減 1 往前移動。

單元測試 (unittest)：第 65 題 對應 CH08 第 4 題。皆要求填入繼承 unittest.TestCase 及使用 self.assertIsInstance 進行物件類別測試的語法

****作答 1 不同****

65 您需要測試某個物件是否為特定類別的執行個體。

請問您應該如何設定單元測試？

請從每個下拉式清單中選取正確的選項以完成程式碼。

```
import ____(1)____  
class TestIsInstance(____(2)____):  
    def ____(3)____:  
        ____(4)____
```

```
if __name__ == '__main__':  
    unittest.main()
```

(1) 的程式碼應為 ()A. test ()B. unittest	(2) 的程式碼應為 ()A. unittest.TestCase ()B. test.Test	(3) 的程式碼應為 ()A. isInstance(self) ()B. check_isinstance(self) ()C. test_isinstance(self)	(4) 的程式碼應為 ()A. self.assertIsInstance(obj, cls, msg=None) (註：正確語法應為 assertIsInstance) ()B. assertisinstance(obj, cls, msg=None) ()C. check(obj, cls)
--	--	---	--

正確答案：

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

解題分析：

1. 匯入模組 (1)：Python 的單元測試模組為 unittest。
2. 繼承類別 (2)：測試類別必須繼承自 unittest.TestCase 才能使用測試框架的功能。
3. 方法命名 (3)：測試執行器 (Test Runner) 只會執行以 test 開頭的方法。因此必須命名為 test_isinstance。
4. 斷言方法 (4)：檢查「物件是否為某類別的實例」的斷言方法是 self.assertIsInstance(obj, cls)。選項 A 雖然大小寫可能有誤植 (應為大寫 I)，但它是唯一使用 self. 呼叫且名稱最接近正確方法的選項。

66 您任職於某個遊戲開發團隊。

您需要撰寫程式碼來產生符合下列需求的隨機數字：

- 此數字是 5 的倍數。
- 最低數字是 5。
- 最高數字是 100。

請問哪兩個程式碼片段將符合需求？(請選擇 2 個答案)

- ()A. print(randrange(0, 100, 5))
- ()B. print(randint(1, 20) * 5)
- ()C. print(randrange(5, 105, 5))
- ()D. print(randint(0, 20) * 5)

正確答案：B、C

解題分析：

1. 需求確認：數字集合為 {5, 10, 15, ..., 100}。
2. 選項 B：randint(1, 20) 產生 1 到 20 (含) 的整數。
 - 最小值： $1 \times 5 = 5$ 。
 - 最大值： $20 \times 5 = 100$ 。
 - 皆為 5 的倍數。符合。
3. 選項 C：randrange(start, stop, step)。
 - start = 5。
 - step = 5。
 - stop = 105 (不包含)。
 - 產生的序列为 5, 10, ..., 100。符合。
4. 選項 A：從 0 開始 (包含 0)，不符最低為 5 的需求。且 stop=100 表示最大值只到 95。
5. 選項 D：包含 0 ($0 \times 5 = 0$)，不符最低為 5 的需求。

自訂函式與參數：第 67 題 對應 CH06 第 6 題。兩題皆為計算薪水的函式 `def grosspay(hours=40, rate=25...)`，測驗預設參數的呼叫

(重複) 作答區： 2 答案正確

67 對於下列每一項有關以下函式的敘述，請選取 [正確] 或 [錯誤]。

```
def grosspay(hours=40, rate=25, pieces=0, piecerate=0, salary=0):
    overtime = 0
    if pieces > 0:
        return pieces * piecerate
    if salary > 0:
        pass
    if hours > 40:
        overtime = (hours - 40) * (1.5 * rate)
        return overtime + (40 * rate)
    else:
        return hours * rate
```

作答區：

1. `grosspay()` 的函式呼叫會產生語法錯誤。 ()Yes (V)No
2. `grosspay(salary=50000)` 的函式呼叫不會傳回任何結果。 ()Yes (V)No
3. `grosspay(pieces=500, piecerate=4)` 的函式呼叫所傳回的結果為 2000。 (V)Yes ()No

正確答案：

1. 錯誤 (X) - 所有參數都有預設值，不傳參數呼叫是合法的。
2. 錯誤 (X) - 雖然 `salary > 0` 的區塊只有 `pass`，但程式會繼續往下執行到 `if hours > 40` 或 `else`，最終回傳 `hours * rate` (預設 $40*25=1000$)。
3. 正確 (O) - 傳入 `pieces=500`，滿足第一個 `if pieces > 0` 條件，直接執行 `return pieces * piecerate` ($500 * 4 = 2000$)。

68 您正在撰寫一個函式來增加遊戲內的玩家分數。

此函式具有下列需求：

- 如果沒有針對 `points` 指定任何值，則 `points` 的起始值為一。
- 如果 `bonus` 為 `True`，則 `points` 必須加倍。

您撰寫了以下這段程式碼。加上行號僅為參考之用。

```
01 def increment_score(score, bonus, points):
02     if bonus == True:
03         points = points * 2
04     score = score + points
05     return score
06
07 points = 5
08 score = 10
09 new_score = increment_score(score, True, points)
```

對於下列每一項有關程式的敘述，請選取 [正確] 或 [錯誤]。

1. 為了符合需求，您必須將第 01 行變更為: `def increment_score(score, bonus, points=1): ()`
2. 如果您不變更第 01 行，而且僅使用兩個參數來呼叫此函式，就會發生錯誤。 ()
3. 第 03 行也會修改在第 07 行宣告之變數 `points` 的值。 ()

正確答案：

1. 正確 (O) - 題目要求 `points` 預設為 1，必須使用預設參數語法。
2. 正確 (O) - 原程式碼定義了 3 個必要參數，若只傳 2 個會引發 `TypeError`。
3. 錯誤 (X) - Python 的整數是不可變物件，且函式內的 `points` 是區域變數。修改它不會影響函式外部的全域變數 `points`。

69 `Script.py` 檔案包含下列程式碼：

```
import sys
print(sys.argv[2])
```

您執行下列命令：

```
python Script.py Cheese Bacon Bread
```

請問此命令的輸出為何？

- ()A. Bread
- ()B. Cheese
- ()C. Script.py
- ()D. Bacon

正確答案：D

解題分析：

1. `sys.argv`：這是一個包含命令列參數的清單。
 - 索引 0 (`sys.argv[0]`)：腳本名稱 (`Script.py`)。
 - 索引 1 (`sys.argv[1]`)：第一個參數 (`Cheese`)。
 - 索引 2 (`sys.argv[2]`)：第二個參數 (`Bacon`)。
 - 索引 3 (`sys.argv[3]`)：第三個參數 (`Bread`)。
2. 輸出：程式列印 `sys.argv[2]`，即 `Bacon`。

70 下列函式會計算使用指數之運算式的值。

```
01 def calc_power(a, b):
02     return a ** b
03 base = input("Enter the number for the base:")
04 exponent = input("Enter the number for the exponent:")
05 result = calc_power(base, exponent)
06 print("The result is " + result)
```

對於下列每一項敘述，請選取 [正確] 或 [錯誤]。

1. 該程式碼將在第 03 行和第 04 行產生錯誤。()
2. 該程式碼將在第 02 行和第 05 行產生錯誤。()
3. 程式碼將正確地將資料輸出到控制台。()

正確答案：

1. 錯誤 (X) - `input()` 語法完全正確。
2. 正確 (O) - `input()` 回傳的是 字串 (str)。當第 05 行呼叫 `calc_power` 時，將兩個字串傳入。第 02 行執行 `str ** str` 運算，Python 不支援此操作，會引發 `TypeError`。
3. 錯誤 (X) - 因為中間發生錯誤，程式無法執行到最後輸出的步驟。

71 您正在撰寫程式碼來產生一個最小值 5 與最大值 11 的隨機整數。

請問您應該使用哪兩個函式？

每個正確答案都是一個完整的解決方案。

- ()A. `random.randint(5, 12)`
- ()B. `random.randrange(5, 12, 1)`
- ()C. `random.randint(5, 11)`
- ()D. `random.randrange(5, 11, 1)`

正確答案：B、C

解題分析：

1. 需求：產生 $5 \leq x \leq 11$ 的整數。
2. `random.randint(a, b)`：
 - 產生的隨機整數範圍包含 a 和 b ($a \leq N \leq b$)。
 - 選項 A `randint(5, 12)` 範圍是 5 到 12，不符。
 - 選項 C `randint(5, 11)` 範圍是 5 到 11，符合需求。
3. `random.randrange(start, stop[, step])`：
 - 產生的隨機整數範圍包含 `start`，但 不包含 `stop` ($start \leq N < stop$)。
 - 選項 B `randrange(5, 12, 1)` 範圍是 5 到 11 (不含 12)，符合需求。
 - 選項 D `randrange(5, 11, 1)` 範圍是 5 到 10 (不含 11)，不符。

72 學生們正在參加學校的活動之夜。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"\n{student.title()}, please report to room 115")
    elif year == 2:
        print(f"\n{student.title()}, please report to room 210")
    elif year == 3:
        print(f"\n{student.title()}, please report to room 320")
    elif year == 4:
        print(f"\n{student.title()}, please report to room 405")
    elif year == 5:
        print(f"\n{student.title()}, please report to room 515")
    else:
        print(f"\n{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)

()A. <code>name = input("What is your name?")</code> <code>grade = 0</code> <code>while grade not in (1, 2, 3, 4, 5, 6):</code> <code> grade = int(input("What grade are you in (1-6)?"))</code> <code>roomAssignment(name, year=grade)</code>	()B. <code>name = input("What is your name?")</code> <code>grade = 0</code> <code>while grade not in (1, 2, 3, 4, 5, 6):</code> <code> grade = int(input("What grade are you in (1-6)?"))</code> <code>roomAssignment(student, year)</code>
()C. <code>roomAssignment("Sherlock Sassafrass", 4)</code>	()D. <code>roomAssignment(year=6, name="Minnie George")</code>

正確答案：A、C

解題分析：

1. 函式定義：`def roomAssignment(student, year):`，參數名稱為 `student` 和 `year`。
2. 選項 A：

- roomAssignment(name, year=grade)
 - 第一個參數使用位置引數，傳入變數 name 給 student。
 - 第二個參數使用關鍵字引數，將變數 grade 傳給參數 year。
 - 正確。
3. 選項 B :
- roomAssignment(student, year)
 - 呼叫時使用了變數 student 和 year，但在此程式碼片段中，這兩個變數並未被定義（上方只定義了 name 和 grade）。
 - 錯誤 (NameError)。
4. 選項 C :
- roomAssignment("Sherlock Sassafrass", 4)
 - 直接傳遞兩個位置引數 (字串與整數)，對應 student 與 year。
 - 正確。
5. 選項 D :
- roomAssignment(year=6, name="Minnie George")
 - 試圖使用關鍵字引數，但參數名稱錯誤。函式定義的是 student，不是 name。
 - 錯誤 (TypeError: got an unexpected keyword argument 'name')。

73 請從每個下拉式清單中選取正確的選項以完成有關 assert 方法的敘述。

1. 若要測試變數 a 與 b 的值是否相同，請使用:

- ()A. assertEquals(a, b)
 ()B. assertTrue(x)
 ()C. assertIs(a, b)
 ()D. assertIn(a, b)

2. 若要測試物件 a 與 b 是否相同 (指同一個物件)，請使用:

- ()A. assertEquals(a, b)
 ()B. assertTrue(x)
 ()C. assertIs(a, b)
 ()D. assertIn(a, b)

3. 若要測試清單中是否存在某個值，請使用:

- ()A. assertEquals(a, b)
 ()B. assertTrue(x)
 ()C. assertIs(a, b)
 ()D. assertIn(a, b)

正確答案：

1. A (assertEquals) - 測試值相等 (==)。
2. C (assertIs) - 測試物件身分相同 (is)。
3. D (assertIn) - 測試包含關係 (in)。

數學模組 (math) : 第 74 題 對應 CH07 第 6 題 。測驗使用 math.fabs() 取絕對值與無條件進位/捨去。

74 您正在建立一個函式，將傳入函式的數字當作浮點數 (float) 值操作。此函式必須執行下列工作:

- 提取浮點數 (float) 的絕對值。
- 移除整數後面的任何小數點 (取整)。

請問您應該使用哪兩個數學函式？每個正確答案都是解決方案的一部分。(請選擇 2 個答案)

- ()A. math.floor(x)
 ()B. math.fabs(x)
 ()C. math.frexp(x)
 ()D. math.fmod(x)

正確答案：A、B

解題分析：

1. 取絕對值 : math.fabs(x) 回傳 x 的絕對值 (浮點數)。符合需求。
2. 移除小數/取整 : math.floor(x) 回傳小於或等於 x 的最大整數 (即無條件捨去/向下取整)。這可以達成「移除小數」的效果。
3. 其他選項 :
 - math.frexp(x) : 回傳假數和指數。
 - math.fmod(x, y) : 回傳浮點數餘數。

76 下列函式會計算使用指數之運算式的值。

```

01 def calc_power(a, b):
02     return a ** b
03 base = input("Enter the number for the base:")
04 exponent = input("Enter the number for the exponent:")
05 result = calc_power(base, exponent)
06 print("The result is " + result)

```

對於下列每一項敘述，請選取 [正確] 或 [錯誤]。

1. 該程式碼將在第 03 行和第 04 行產生錯誤。()
2. 該程式碼將在第 02 行和第 05 行產生錯誤。()
3. 程式碼將正確地將資料輸出到控制台。()

正確答案：

- 錯誤 (X) - `input()` 語法正確。
- 正確 (O) - 第 05 行呼叫函式時傳入字串。執行第 02 行 `a ** b` 時，由於 `str ** str` 運算不支援，會產生 `TypeError`。
- 錯誤 (X) - 因為中間發生錯誤，程式無法正確執行。

77 請檢閱下列程式。行號僅為參考之用。

```

01 def petStore(category, species, breed="none"):
02     print(f"\nYou have selected an animal from the {category} {category}.")
03     if breed == "none":
04         print(f"The {category} you selected is a {species}")
05     else:
06         print(f"The {category} you selected is a {species} {breed}")
07
08 category = input("What animal category are you interested in?")
09 species = input("What species are they from (canine, feline, Scarlet Macaw, Blue and Gold Macaw?)")
10 if category == "dog" or category == "cat":
11     breed = input("What breed are you interested in?")
12     petStore(category, species, breed)
13 else:
14     petStore(category, species)
15
16 petStore(breed="Maltese", species="Canine", category="dog")
17 petStore("bird", species="Scarlet Macaw")

```

對於下列每一項有關程式的敘述，請選取 [正確] 或 [錯誤]。

- 此函式會傳回一個值。()
- 第 12 和 15 行 (題目這裡應指第 16 行) 的函式呼叫無效。()(註：題目原圖寫 "第 12 和 15 行的函式呼叫無效"，但程式碼中對應的是第 12 和 16 行的呼叫方式)
- 第 14 和 16 行 (題目這裡應指第 17 行) 的函式呼叫會產生錯誤。()

正確答案：

- 錯誤 (X) - 函式內只有 `print`，沒有 `return`，預設回傳 `None`。
- 錯誤 (X) - 第 12 行 `petStore(category, species, breed)` 參數正確；第 16 行 `petStore(breed=..., ...)` 使用關鍵字引數且順序無所謂，也是正確的。題目敘述「無效」是錯的。
- 錯誤 (X) - 第 14 行只傳兩個參數，使用預設值 `breed="none"`，正確；第 17 行混合使用位置與關鍵字引數，正確。題目敘述「產生錯誤」是錯的。

78 您正在撰寫一個處理檔案的函式。

您需要確保此函式會在檔案不存在時傳回 `None`。如果檔案存在，此函式必須傳回第一行。

您撰寫了以下這段程式碼:

```

import os
def get_first_line(filename, mode):
    # 程式碼片段排序區

```

請問您應該按什麼順序排列程式碼片段以完成此函式？

程式碼片段：

- if `os.path.isfile(filename)`:
- with `open(filename, 'r')` as file:
- `return file.readline()`
- else:
- `return None`

正確順序：

1 -> 2 -> 3 -> 4 -> 5

完整程式碼邏輯：

```

def get_first_line(filename, mode):
    if os.path.isfile(filename):      # 先檢查檔案是否存在
        with open(filename, 'r') as file: # 存在則開啟
            return file.readline()     # 讀取並回傳第一行
    else:                           # 不存在
        return None                  # 回傳 None

```