Preface

序

Python 是目前最熱門的程式語言，其特性爲功能強大、語法簡潔優雅，容易學習、沒有複雜的結構，而且容易維護。其應用範圍很廣，無論是資料收集、資料分析、機器學習、自然語言處理、視窗應用程式、電腦視覺辨視、數據圖表設計、網站建置、雲端服務開發甚至是遊戲開發，都能看到 Python 的身影。

本書教學部分以考取 ITS Python 國際認證爲主，培養初學者程式設計的基本能力，內容涵蓋 Python 開發環境建置、變數與資料型別、輸出入函式、選擇結構、重複結構、串列與集合、排序與搜尋、函式、遞迴、字典、套件使用、檔案與例外處理，數據圖表設計、視窗應用程式開發，網頁爬蟲實戰…等。爲 Python 初學者奠定前進大數據、機器學習與人工智慧的基礎。

各章內容講解融入 ITS Python 解題技巧，同時**書末整理兩組「ITS Python 國際認證模擬試題」試卷**供教師教學與讀者練習。除此之外，本書更加上視窗應用程式、數據圖表設計與網路爬蟲的章節，讓初學者不僅能瞭解 Python 的應用，更讓初學者程式設計訓練更加札實，同時也是教師訓練學生考取 ITS Python 國際認證的最佳教材。

爲方便教學，本書另提供教學投影片、各章習題與解答，歡迎採用本書的授課教師向碁峰業務索取。有關本書的任何問題可來信至 itPCBook@gmail.com，我們會盡快答覆。本書雖經多次精心校對，難免百密一疏，尙祈讀者先進不吝指正，以期再版時能更趨紮實。感謝周家旬與蔡文真小姐細心校稿與提供寶貴的意見，以及碁峰同仁的鼓勵與協助，使得本書得以順利出書。在此聲明，書中所提及相關產品名稱皆爲各所屬公司之註冊商標。

微軟最有價值專家、僑光科技大學多媒體與遊戲設計系 助理教授 蔡文龍
何嘉益、張志成、張力元 編著
2021.11.30 於台中

CHAPTER

4

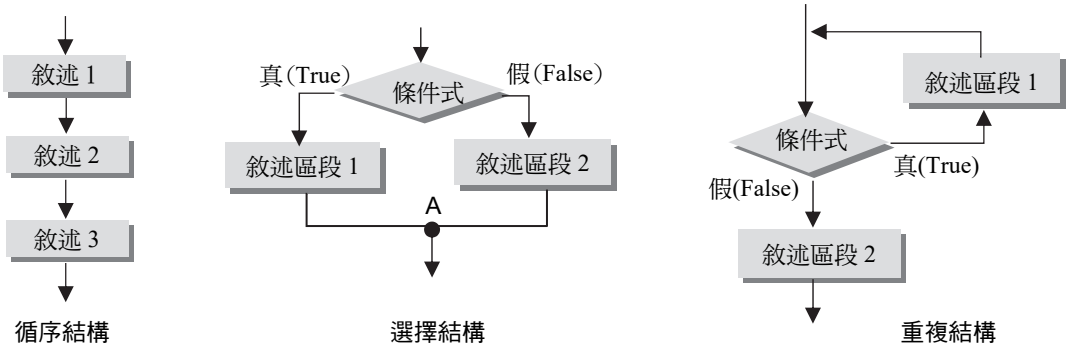
選擇結構

- 結構化程式設計
- 關係運算式
- 邏輯運算式
- 選擇結構
- 檢測模擬試題解析

4.1 結構化程式設計

Python 是一種高階程式語言，可以同時支援多種撰寫方式，例如：物件導向、命令、函式與程序的編寫方式；Python 也是一種「結構化程式設計」的程式語言。「結構化程式設計」技術是透過程式的模組化和程式的結構化，來簡化程式設計的流程，降低邏輯錯誤發生的機率。這種程式設計的觀念，是由上而下的程式設計，將程式中有獨立功能的程式區段分割出來使成為「模組」(Module)，這些模組最後再組合成一個大的完整程式軟體。

「結構化程式設計」採用「循序結構」、「選擇結構」、「重複結構」這三個基本流程架構來設計程式。在前面章節所撰寫的程式，架構是採用由上而下一行接著一行執行的「循序結構」。本章所要介紹的程式流程，是會因條件的不同而執行不同的程式區塊，這種有選擇性的流程架構稱為「選擇結構」。下一章再來介紹「重複結構」，這種流程會在條件式成立的情況下重複執行相同的程式區段。

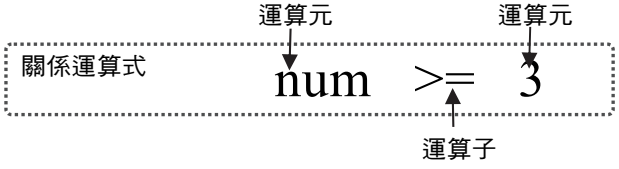


在程式語言中的條件是透過運算式來設定，Python 中能產生條件的運算式有「關係運算式」和「邏輯運算式」。運算式的結果，有條件成立與條件不成立兩種情況，由布林值來記錄運算結果。當條件成立時，運算結果的布林值為「True」(稱為「真」)，；當條件不成立時，運算結果的布林值為「False」(稱為「假」)。

NOTE Python 布林(bool)資料型別所提供的值為 True 和 False，但是當布林值進行整數運算時，True 會轉成「1」，False 會轉成「0」，此處要特別注意。

4.2 關係運算子

程式中要將兩個相同型別資料做比較時，這個比較資料的運算式子就稱為「關係運算式」。在一個最簡單的關係運算式中，必須含有兩個用來被比較的相同型別資料稱為「運算元」，也含有用來比較兩個運算元的「關係運算子」。



關係運算式比較後會傳回布林值，就是有 True 和 False 兩種結果，或者稱為「真」和「假」。關係運算式的條件若成立時，則結果會為 True(真)；條件若不成立，則結果會為 False(假)。在程式的選擇結構敘述中，可以使用關係運算式做為條件式，來決定程式的流程。

在 Python 中關係運算子是由「>」(大於)、「<」(小於)或「=」(等於)三個運算子組合成六種狀態，以供設計程式時使用。關係運算子說明如下表：

運算子	說明	簡例
== (相等)	判斷此運算子左右兩邊資料值是否相等。	10 == 10 ⇨ True(真) 15 == 3 ⇨ False(假) 4 + 2 == 1 + 5 ⇨ True(真)
!= (不相等)	判斷此運算子左右兩邊資料值是否不相等。	7 != 9 ⇨ True(真) 16 != 16 ⇨ False(假) 2 * 3 != 3 * 2 ⇨ False(假)
< (小於)	判斷此運算子左邊的資料值是否小於右邊的資料值。	8 < 9 ⇨ True(真) 15 < 10 ⇨ False(假) 2 < 9 - 6 ⇨ True(真)

運算子	說明	簡例
> (大於)	判斷此運算子左邊的資料值是否大於右邊的資料值。	12 > 10 ⇨ True(真) 2 > 3 ⇨ False(假) 6 * 2 > 4 * 2 ⇨ True(真)
<= (小於等於)	判斷此運算子左邊的資料值是否小於或等於右邊的資料值。	12 <= 13 ⇨ True(真) 10 <= 10 ⇨ True(真) 10+4 <= 13 ⇨ False(假)
>= (大於等於)	判斷此運算子左邊的資料值是否大於或等於右邊的資料值。	12 >= 13 ⇨ False(假) 10 >= 10 ⇨ True(真) 10 + 4 >= 13 ⇨ True(真)

[例] 使用關係運算子並將結果以字串和整數格式顯示： (檔名：operator1.py)

```
01 k=5;
02 i = 'A'>'B';
03 j = ((5+i) == k);
04 k = 5+(100<50)*3+(-20!=20)*2;
05 print("以字串顯示：")
06 print('i=%s, j=%s, k=%s '%(i,j,k))      #顯示 i=False, j=True, k=7
07 print("以整數顯示：")
08 print('i=%d, j=%d, k=%d '%(i,j,k))      #顯示 i=0, j=1, k=7
```

1. 第 2 行： A 字元的 ASCII 碼為 65，B 字元的 ASCII 碼為 66。'A' 與 'B' 比大小時會以 ASCII 碼值做比較，'A' > 'B' 等於 65 > 66，比較結果為 False(假)，所以 i = False。
2. 第 3 行：5+i 因為 i 是 False 會轉型為 0，所以運算式為 5+0 == 5，比較結果為 True(真)，所以 j =True。
3. 第 4 行：因(100 < 50)結果 False(假)，(-20 != 20) 結果為 True(真)；整個運算式為 k = 5+0×3+1×2 = 7，所以 k = 7。
4. 第 6 行：以字串方式顯示 i、j、k 的值，分別顯示為 False、True 和 7。
5. 第 8 行：以整數方式顯示 i、j、k 的值，分別顯示為 0、1 和 7。

4.3 邏輯運算式

當要把多個關係運算式一起做判斷時，便需要使用「邏輯運算子」來連結運算，這種運算式稱為「邏輯運算式」。邏輯運算式運算後會傳回布林值，就是有 True 和 False 兩種結果，或者稱為「真」和「假」。在程式的選擇結構敘述中，也可以使用邏輯運算式來做為條件式，來決定程式的流程。Python 提供的邏輯運算子如下：

1. **and (且)** 邏輯運算子

   當 and 運算子左右兩邊有 <條件式 A> 和 <條件式 B>，若 <條件 A > 和 <條件 B> 的運算結果皆不為 False(假)，則「A and B」這個邏輯運算式的結果為 True(真)；否則為 False(假)。and 邏輯運算子的運算結果如下表：

A	B	A and B
True	True	True
True	False	False
False	True	False
False	False	False

[例] 溫度(temper)高於 30 而且不超過 38 的條件式寫法：

```
(temper > 30) and (temper <= 38)
```

2. **or (或)** 邏輯運算子

   當 or 運算子左右兩邊的 <條件 A > 和 <條件 B>，只要其中一個條件運算結果為 True(真) 時，則「A or B」這個邏輯運算式的結果為 True(真) ，若兩個條件皆為 False(假)時，「A or B」這個條件式的結果才會為 False(假)。or 邏輯運算子的運算結果如下表：

A	B	A or B
True	True	True
True	False	True
False	True	True
False	False	False

[例] 分數(score)必須介於 0~100 之間，無效分數的條件式寫法：

```
(score < 0) or (score > 100)
```

3. **not (相反)** 邏輯運算子

not 運算子是單一的條件式運算，主要是把條件式的結果運算成相反結果，即 True⇨False，False⇨True。not 邏輯運算子的運算結果如下表：

A	not A
True	False
False	True

[例] 練習使用邏輯運算子，並將運算的結果顯示出來： (檔名：operator2.py)

```
01 x=8
02 y=3
03 i = (y==3) and (x<3)
04 j = (x+y==11) or (y>8)
05 k = not((x<=y) and (x>y) or ('A'>'C'))
06 print(f'i={i}, j={j}, k={k}')          #顯示 i=False, j=True, k=True
```

1. 第 3 行：i = (y==3) and (x<3)
   → i = (3==3) and (8<3)
   → i = (True) and (False)
   → i = False
2. 第 4 行：j = (x+y==11) or (3>8)
   → j = (8+3==11) or (3>8)
   → j = (True) or (False)
   → j = True
3. 第 5 行：k = not ((x<=y) and (x>y) or ('A'>'C'))
   → k = not ((8<=3) and (8>3) or False)
   → k = not ((False and True) or False)
   → k = not (False or False)
   → k = not (False)
   → k = True
4. 第 6 行：顯示 i、j、k 的值，會分別顯示為 False、True 和 True。

4.4 選擇結構

由上而下一行接一行逐行執行，即使再執行一次其流程仍不會改變，此種程式架構稱為「循序結構」。但是較複雜的程式會應程式的需求，依照條件式的不同而進行不同的執行流程，而得到不同的結果，這種架構就是「選擇結構」。舉一個日常生活的例子：如果 (if) 今天天氣好就去郊遊，否則 (else) 就待在家裡看電視，就是一種「選擇結構」。Python 提供的 if 選擇結構敘述如下：

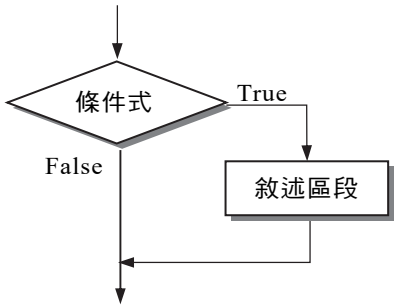
1. 單向選擇： if …
2. 雙向選擇： if … else …
3. 巢狀選擇： if … else …
4. 多向選擇： if … elif … else

選擇結構中會有條件式，依照條件式的不同會執行不同的程式流程。條件式可以使用前面介紹的關係和邏輯運算式來編寫。

4.4.1 單向選擇 if...

所謂「單向選擇」是指當 if 的條件式成立為 True 時，即會執行 if 條件式「:」冒號後面縮排的敘述區段；若 if 的條件式不成立為 False 時，則跳過 if 單向選擇結構，執行縮排敘述區段後的敘述。語法及流程圖如下：

```
if (條件式) :
    敘述區段
```



if 後面條件式的左右括號()亦可省略，但條件式加上左右括號較容易閱讀程式。

[例] 求 num 的絕對值。

```
if (num < 0) :        #省略左右括號()亦可寫為 if num < 0 :
    num = -num
```

[例] 成績在 55 分以上未達 60 分者，以 60 分計分。

```
if ((score >= 55) and (score < 60)) :
    score = 60
    print('差強人意,通融過關')
```

NOTE Python3.8 中新增「:=」指定表式(Assignment Expressions 或稱指派表示式)，因為符號很像海象臉部的眼睛和長牙，因此被暱稱為海象運算子。指定表式可縮短程式碼加快執行速度，但會降低可讀性，可自行視需求使用。例如：

```
score = 65
if (score >= 60):
    print('及格')
```

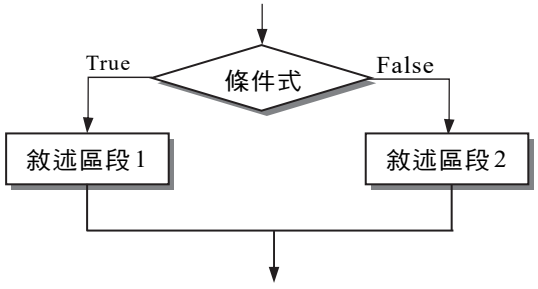
使用海象運算子時，可以改寫為：

```
if ((score:= 65) >= 60):
    print('及格')
```

4.4.2 雙向選擇 if...else...

所謂「雙向選擇」是指當條件式成立時，即執行 if 條件式「:」冒號後面的敘述區段 1；若條件式不成立時，則執行 else「:」冒號後面的敘述區段 2。語法及流程圖如下：

```
if (條件式) :
    敘述區段 1
else :
    敘述區段 2
```



[例] 門票 300 元，若年齡低於 10 歲或 65 歲以上半價。

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

[例] 根據成績及格或不及格，給予不同回饋。

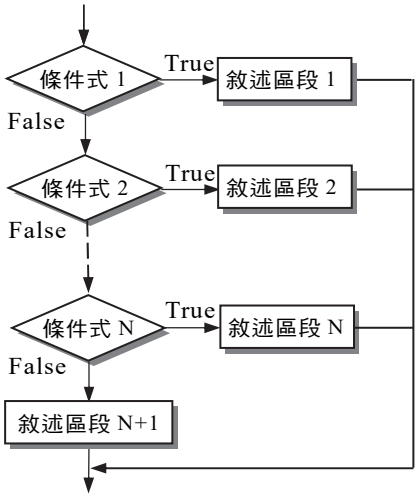
```
if (score > 60) :
    print('及格');
    print('恭喜過關！')
else :
    print('不及格')
    print('明年再會！')
```

NOTE 注意選擇結構中同一敘述區段的敘述縮排長度要相同，不可以長短不一，執行時會造成錯誤。所以建議使用 Tab 鍵來做縮排，長度一致才不易出錯。要增加縮排可再按一次 Tab 鍵，減少縮排則可按 Backspace 退位鍵。

4.4.4 多向選擇 if...elif...else

當選擇的項目超過兩個，除了可以使用巢狀選擇結構外，也可以使用「多向選擇」結構來處理。其使用方式就是除了在第一個條件使用 if 判斷外，其他條件都使用 elif 來判斷，最後再以 else 來處理剩下的可能性。語法及流程圖如下：

```
if (條件式 1) :
    敘述區段 1
elif (條件式 2) :
    敘述區段 2
……
elif (條件式 N) :
    敘述區段 N
else :
    敘述區段 N+1
```



簡例 (檔名：movie.py)

根據最低年齡制定電影分級，電影分級的規則如下：

① 普遍級：0 歲以上皆可欣賞。

② 保護級：6 歲(含)以上皆可欣賞。

③ 輔導級：12 歲(含)以上皆可欣賞。

④ 限制級：18 歲(含)以上皆可欣賞。

⑤ 預設為普遍級。

結果

```
請輸入年齡(正整數)：24 Enter
年齡 24 歲最高可看限制級電影
```

```
請輸入年齡(正整數)：6 Enter
年齡 6 歲最高可看保護級電影
```

程式碼

檔名： \ex03\movie.py

```
01 age=int(input('請輸入年齡(正整數)：'))
02 rating = "普遍級"
03 if age >= 18:
04     rating = "限制級"
05 elif age >= 12:
```

```
06      rating = "輔導級"
07 elif age >= 6:
08      rating = "保護級"
09 else:
10      rating = "普遍級"
11 print(f'年齡{age}歲最高可看{rating}電影')
```

說明

1. 第 1 行：使用 input()函式取得輸入的字串，再使用 int()函式轉成整數，接著將整數指定給 age 變數，此變數代表使用者所輸入的年齡。
2. 第 2 行：設 rating = "普遍級"，來預設電影為"普遍級"。
3. 第 3,4 行：若 age >= 18 條件成立，則改設 rating = "限制級"。
4. 第 5,6 行：若 age >= 12 條件成立，則改設 rating = "輔導級"。
5. 第 7,8 行：若 age >= 6 條件成立，則改設 rating = "保護級"。
6. 第 9,10 行：其餘就設 rating = "普遍級"。

4.5 檢測模擬試題解析

題目 (一)

下列為兩數大小比較的程式碼(行號僅供參考)，請回答下列問題。

```
01 n1 = eval(input("請輸入第一個數字："))
02 n2 = eval(input("請輸入第二個數字："))
03 if n1 < n2:
04       print("第一個數字較小")
05 if n1 >= n2:
06       print("第一個數字較大")
07 if n1 == n2:
08       print("二個數字相等")
09 if n2 = n1:
10       print("二個數字相等")
```

1. 第 04 行的敘述只有當 n1 小於 n2 時才會執行。(A) 正確　(B) 錯誤
2. 第 06 行的敘述只有當 n1 大於 n2 時才會執行。(A) 正確　(B) 錯誤
3. 第 08 行的敘述只有當 n1 等於 n2 時才會執行。(A) 正確　(B) 錯誤
4. 第 09 行的敘述無法正確執行。(A) 正確　(B) 錯誤

說明

1. 第 1 題：第 04 行的敘述只有當第 03 行的條件成立才會執行，所以題目正確因此答案為 (A)。
2. 第 2 題：第 05 行的條件式是 n1 大於等於 n2，所以題目錯誤因此答案為 (B)。
3. 第 3 題：第 07 行的條件式是 n1 等於 n2，所以題目正確因此答案為 (A)。
4. 第 4 題：因為兩數相等的運算子為==而非=，所以第 09 行的敘述無法正確執行，因此答案為(A)。題目相關的程式碼請參考 test04_01.py。

題 目 (二)

下列程式碼是根據玩家目前的點數(points)和等級(grade)，來決定玩家的最終點數，請問程式執行後輸出值為何(行號僅供參考)？

```
01 points = 1876
02 grade = 5
03 if points > 2000 and grade >=5:
04        points += 100
05 elif points > 1000 and grade >5:
06        points += 50
07 else:
08        points -=50
09 print(points)
```

(A) 1826　(B) 1876　(C) 1926　(D) 1976

說明

因為 points = 1876 和 grade = 5，所以不符合第 03 行 points > 2000 and grade >=5 的條件，也不符合第 05 行 points > 1000 and grade >5 的條件，所以會執行第 08 行的敘述，所以 points-50 變數值為 1826，所以答案為(A)。題目相關的程式碼請參考 test04_02.py。

題 目 (三)

下列為檢查使用者輸入的整數位數，是一位、二位還是超過二位數的程式碼(行號僅供參考)，請回答以下問題來完成程式：

```
01 n = int(input("請輸入一~二位數的整數："))
02 
03     digit = "一"
04 
```

```
05      digit = "二"
06
07      digit = "大於二"
08 print("%d 是%s 位數"%(n, digit))
```

1. 請問第 02 行應該填入下列哪段敘述？

(A) if n >-10 and n <10:　　(B) if n >-100 and n < 100:

(C) if n > -10 or n < 10:　　(D) if n > -100 or n < 100:

2 請問第 04 行應該填入下列哪段敘述？

(A) if n > -100 and n < 100:　　(B) elif n > -100 and n < 100:

(C) if n > -10 and num < 10:　　(D) elif n > -10 and n < 10:

3. 請問第 06 行應該填入下列哪段敘述？

(A) else:　(B) elif:　(C) elseif:　(D) if:

說明

1. 第 01 行用 input() 函式接受輸入，並用 int() 函式轉成整數存在變數 n 中。
2. 第 1 題：因為第 03 行設 digit = "一"，表示要篩選出一位數整數，敘述應為：if n >-10 and n <10:，所以答案為(A)。
3. 第 2 題：因為第 05 行設 digit = "二"，表示要篩選出二位數整數，敘述應為：elif n > -100 and n < 100:，所以答案為(B)。
4. 第 3 題：因為是 if 選擇結構剩餘的部分，敘述應為：else:，所以答案為(A)。題目相關的程式碼請參考 test04_03.py。

題目 (四)

下列為換算成績等級的程式碼，換算的規則如下：

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

請回答以下問題來完成程式(行號僅供參考)：

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

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

1. 請問第 02 行應該填入下列哪段敘述？

(A) if score <= 90:　(B) if score >= 90:　(C) if score > 90:　(D) if score == 90:

2. 請問第 04 行應該填入下列哪段敘述？

(A) if score >= 80:　(B) elif score == 80:　(C) elif score > 80:　(D) elif score >= 80:

3. 請問第 06 行應該填入下列哪段敘述？

(A) if score >= 70:　(B) elif score == 70:　(C) elif score > 70:　(D) elif score >= 70:

4. 請問第 08 行應該填入下列哪段敘述？

(A) if score >= 60:　(B) elif score == 60:　(C) elif score >60:　(D) elif score >= 60:

說明

1. 第 1 題：因為分數大於等於 90 就屬於優級，第 02 行敘述應為 if score >= 90:，所以答案為(B)。
2. 第 2 題：因為分數 80(含)~89 分為甲級，第 04 行敘述應為 elif score >= 80:，所以答案為(D)。第 3、4 題寫法和第 2 題類似，所以答案都為(D)。題目相關的程式碼請參考 test04_04.py。

題目 (五)

下列為計算單車出租費用的程式碼，費用計算規則如下：

- 出租一天費用是每天 100 元。
- 如果在晚上 10 點後返還，將加收一天額外的費用。
- 如果是在星期天(Sunday)租借，可享八折優惠。
- 如果是在星期三(Wednesday)租借，可享六折優惠。

請回答以下問題來完成程式：(行號僅供參考)：

```
01 onTime = input("單車是否在晚上 10 點前返還？(請填 y 或 n)").lower()
02 days = int(input("請輸入單車出租天數？"))
03 weekday = input("單車是在星期幾出租?(請用英文)").capitalize()
04 money = 100
05 if onTime 
06     days += 1
```

```
07 if weekday
08     total = (days * money) * 0.8
09 elif weekday
10     total = (days * money) * 0.6
11 else:
12     total = (days * money)
13 print("單車的出租費用總計為 : ", int(total), "元")
```

1. 請問第 05 行應該填入下列哪個指令？

(A) != "n":　(B) == "n":　(C) == "y":　(D) = "y":

2. 請問第 07 行應該填入下列哪個指令？

(A) == "Sunday":　(B) >= "Sunday":　(C) is "Sunday":　(D) ="Wednesday":

3. 請問第 09 行應該填入下列哪個指令？

(A) == "Wednesday":　(B) >= "Wednesday":　(C) is "Wednesday":　(D) ="Sunday":

說明

1. 第 1 題：因為晚上 10 點後才返還會加收一天費用，第 05 行敘述應為 if onTime== "n":，所以答案為(B)。另外，第 01 行的 lower()函式會將字母轉成小寫。
2. 第 2 題：因為星期天(Sunday)租借將享八折，第 07 行敘述應為 if weekday== "Sunday":，所以答案為(A)。另外，第 03 行的 capitalize ()函式會將第一個字母大寫其餘字母小寫。
3. 第 3 題：因為星期三(Wednesday)租借將享六折，第 09 行敘述應為 if weekday== "Wednesday":，所以答案為(A)。題目相關的程式碼請參考 test04_05.py。

題目 (六)

請由下列敘述區段中選出四個，並依序組合成檢查輸入字串中是否有大小寫字母的程式。

(A) word = input("請輸入英文單字：")
(B) elif word.upper() == word:
　　print(word, "全部大寫字母")
(C) else:
　　print(word, "是小寫字母")
(D) else:
　　print(word, "有大和小寫字母")
(E) if word.lower() == word:
　　print(word, "全部小寫字母")
(F) else:
　　print(word, "是大寫字母")

說明

1. 要檢查輸入字串中是否有大小寫字母，首先要有輸入的字串，所以第一個敘述為(A) word = input("請輸入英文單字：")
2. 要判斷大小寫字母可使用 if 選擇結構，所以第 2 個敘述區段為(E)。敘述區段 E 用 lower()函式將輸入字串轉成小寫，若和原字串相同表示全部為小寫字母。
3. 判斷出全部小寫字母，接著判斷是否是全部為大寫字母，所以第 3 個敘述區段為(B)。敘述區段 B 以 elif 開頭，用 upper ()函式將輸入字串轉成大寫，若和原字串相同表示全部為大寫字母。
4. 判斷出全部小寫和全部大寫字母後，剩下的就是混合有大和小寫字母，所以第 4 個敘述區段為(D)。敘述區段 D 以 else 開頭，處理選擇結構剩餘的部分。因此答案依序為(A)、(E)、(B)、(D)，題目相關的程式碼請參考 test04_06.py。

題 目 (七)

下列為換算成績(0~100)等級的程式碼，換算的規則如下：

- 90(含)~100 分為「優」。
- 80(含)~89 分為「甲」。
- 70(含)~79 分為「乙」。
- 其它分數為「不及格」。

請在以下空格填入適當指令來完成程式(行號僅供參考)，指令的代碼如下：
(A) elif (B) if (C) for (D) else (E) or (F) and (G) not

```
01   ___①___ score <=100:
02           ___②___  score >=90:
03                   print('成績等級為優')
04           ___③___  score >=80:
05                   print('成績等級為甲')
06           ___④___  score <80  ___⑤___  scrre > 69:
07                   print('成績等級為乙')
08           ___⑥___
09                   print('成績不及格')
10   ___⑦___
11           print('輸入的成績不正確')
```

說明

1. 本題目為巢狀選擇結構，外層選擇結構先過濾 score 分數必須小於等於 100，所以第①題答案為(B) if，第⑦題答案為(D) else。

2. 第 02~09 行為內層選擇結構，第②題：因為分數 80(含)~89 分為甲級，第 04 行敘述應為 elif score >= 80:，所以答案為(D)。第③、④題寫法和第②題類似，所以答案都為(D)。題目相關的程式碼請參考 test04_07.py。

題目 (八)

下列為遊樂場門票費用的函式 price()，門票費用的規則如下：

- 未滿 6 歲免費入場。
- 年滿 6 歲以上的當地居民門票為 100 元。
- 6 ~ 17 歲的非當地居民門票為 200 元。
- 超過 17 歲的非當地居民門票為 500 元。

請回答以下問題來完成程式(行號僅供參考)：

```
01 def price(age, locate):
02     money = 0
03     
04         money = 100
05     
06         
07             money = 200
08         else:
09             money = 500
10     return money
```

1. 請問第 3 行應該填入下列哪段敘述？

(A) if age >= 6 and locate == True:　　(B) if age >= 6 and age <=17:

(C) if age >= 6 and locate == False:　　(D) if age > 6 and locate == True:

2. 請問第 5 行應該填入下列哪段敘述？

(A) elif age >= 6 and locate == False:　　(B) else age >= 6 and locate == False:

(C) elif age >= 6 or locate == True:　　(D) else age >= 6 or locate == True:

3. 請問第 6 行應該填入下列哪段敘述？

(A) if age >= 6 and locate == True:　　(B) if age >= 6 and locate == False:

(C) if age <= 17:　　(D) if age > 17:

說明

1. 第 01 行敘述是當呼叫 price()自定函式時，會傳入 age(年齡)和 locate(居民)兩個變數值，運算後由第 10 行敘述將 money(門票金額)傳回。函式的相關用法，將在第 7 章再做詳細的說明。