課後習題解答

第一章【本章課後習題】

1. 解答：程式碼簡潔易讀、跨平台、物件導向、容易擴充、自由／開放原始碼等。
2. 解答：直譯式語言則是利用解譯器（interpreter）來對高階語言的原始程式碼做逐行解譯，每解譯完一行程式碼後，才會再解譯下一行。解譯的過程中如果發生錯誤，則解譯動作會立刻停止。由於使用解譯器翻譯的程式每次執行時都必須再解譯一次，所以執行速度較慢，不過因爲僅需存取原始程式，不需要再轉換爲其它型態檔案，因此所占用記憶體較少。例如Python、Basic、LISP、Prolog等語言皆使用解譯的方法。
3. 解答：所謂整合開發環境（Integrated Development Environment, IDE），就是把有關程式的編輯（edit）、編譯（compile）、執行（execute）與除錯（debug）等功能於同一操作環境下，讓使用者只需透過此單一整合的環境，即可輕鬆撰寫程式。
4. 解答：編譯（compile）使用編譯器（compiler）來將程式碼翻譯爲目的程式（object code），編譯必須原始程式碼完全正確，編譯的動作才會成功。直譯（interpret）是使用直譯器（interpreter）來對原始程式碼做逐行解釋的方法，每解釋完一行程式碼後，才會再解釋下一行。若解釋的過程中發生錯誤，則直譯的動作會停止。
5. 解答：
   - 輸入資料：0個或多個輸入。
   - 輸出結果：至少1個以上的輸出結果。

- 明確性：描述的處理程序必須是明確，不能模稜兩可。
- 有限性：必須在有限的步驟內完成工作，不可以有無窮迴圈。
- 正確性：可以正確地解決問題。

6. 解答：
- 編譯：編譯器會先檢查整個程式，確認完全沒有語法錯誤之後，再連結相關資源輸出可執行檔（executable file）。編譯完成的可執行檔是可以直接執行的檔案，每一次執行時，不需再翻譯，所以執行速度較快。缺點是編譯過程中如發生錯誤時，則必須回到程式碼找出錯誤的地方加以更正，再重新編譯、連結、產生可執行檔，開發過程會比較不便。編譯式的語言例如C、Fortran、COBOL等等。
- 直譯：Python就是屬於直譯式的語言，直譯顧名思義就是一邊解讀原始碼，一邊執行，當錯誤發生時會停止執行並顯示錯誤行數與原因，對程式開發來說會比較方便，也因爲它不產生執行檔，每一次執行都必須經過直譯才能執行，因此執行效率會比編譯式稍差。直譯式的語言如HTML、JavaScript、Python等等。

7. 解答：步驟1：指定i=1、sum=0

步驟2：sum的值+i (sum=sum+i)

步驟3：i的值+1 (i=i+1)

步驟3：如果i大於5，演算法結束，否則，返回重新執行步驟2。

第二章【本章課後習題】

一、填充題

1. 保留字
2. help()

3. 靜態型別　動態型別

4. True　False

5. %　format

二、問答與實作題

1. 解答：fileName01 (ans:有效)

   $result (Ans:無效,不能有$)

   2_result (Ans:無效,第一個字元不能是數字)

   number_item (Ans:有效)

2. 解答：整數（int）：100

   浮點數（float）：25.3

   布林值（bool）：True

3. 解答：字串的索引值具有順序性，如果要取得單一字元或子字串，就可以使用[]運算子，而從字串中擷取子字串的動作就稱爲「切片」（Slicing）運算。例如：

   msg = 'Sunday is fun!'

   則msg[2:5]結果值爲'nda'。

4. 解答：

跳脫字元	說明
\t	水平跳格字元（horizontal tab）
\n	換行字元（new line）
\'	顯示單引號（single quote）
\\	顯示反斜線（backslash）

5. 解答：

格式化符號	說明
%s	字串

格式化符號	說明
%d	整數
%f	浮點數
%x	十六進位整數

6. 解答：● 不需要理會引數資料型態，一律用{}表示
   ● 可使用多個引數，同一個引數可以多次輸出，位置可以不同
7. 解答：● int()：強制轉換爲整數資料型態
   ● float()：強制轉換爲浮點數資料型態
   ● str()：強制轉換爲字串資料型態
8. 解答：● 變數名稱第一個字元必須是英文字母或是底線或是中文。
   ● 其餘字元可以搭配其他的大小寫英文字母、數字、_或中文。
   ● 不能使用Python內建的保留字。
   ● 區分大小寫字母。
9. 解答：

   ● 7_up

   錯誤原因：變數名稱第一個字元必須是英文字母或是底線或是中文，但不能是數字。

   ● for

   錯誤原因：不能使用Python內建的保留字，for是保留字。

   ● $$$999

   錯誤原因：變數名稱第一個字元必須是英文字母或是底線或是中文，但不能是特殊符號。

   ● happy new year

   錯誤原因：變數名稱不能包含空白。

第三章【本章課後習題】

一、填充題

1. 運算子　運算元
2. 單一指派　複合指派
3. and　or　not
4. 快捷運算
5. 由左到右

二、問答與實作

1. 解答：8
2. 解答：16
3. 解答：54.0
4. 解答：False
5. 解答：17
   a=8
   a=2
6. 解答：因爲15的二進位表示法爲1111，10的二進位表示法爲1010，兩者執行AND運算後，結果爲$(1010)_2$也就是$(10)_{10}$。
7. 解答：NOT作用是取1的補數（Complement），也就是0與1互換。例如a=12，二進位表示法爲1100，取1的補數後，由於所有位元都會進行0與1互換。
8. 解答：在等號關係是「==」運算子，至於「=」則是指定運算子，這種差距很容易造成程式碼撰寫時的疏忽，請多加留意。
9. 解答：指定運算子(=)右側可以是常數、變數或運算式，最終都會將值指定給左側的變數；而運算子左側也僅能是變數，不能是數值、函數或運算式等。例如運算式X-Y=Z就是不合法的。

10. 解答：

① 當遇到一個運算式時，先區分運算子與運算元。

② 依照運算子的優先順序作整理的動作。

③ 將各運算子根據其結合順序進行運算。

11. 解答：200.0

-59.2

-1.0

第四章【本章課後習題】

一、填充題

1. for　while
2. range()
3. break
4. continue
5. 變數初始值　迴圈條件式　調整變數增減值

二、問答與實作

1. 解答：1

4

7

10

13

2. 解答：

```
sum=0
index=0
```

```
while index <= 50:
    sum=sum+index
    index += 2
print ('1～50偶數總和', sum)
```

3. 解答：

```
N = int(input("請輸入一個數值："))
print('False' if N%7 else 'True')
```

4. 解答：

1. 迴圈的執行主體，由程式敘述或複合敘述組成。
2. 迴圈的條件判斷，決定迴圈何時停止執行。

5. 解答：13579
6. 解答：Love
   Happy
   Money
7. 解答：1 3
8. 解答：280
9. 解答：97135
10. 解答：Tall
11. 解答：5的倍數

第五章【本章課後習題】

一、填充題

1. tuple
2. del

3. 「+」　「*」

4. key　value

5. get()

二、問答與實作

1. 解答：{'a': 1, 'b': 9, 'c': 4, 'd': 25, 'e': 16}

2. 解答：

資料型態	tuple	list	dict	set
中文名稱	序對	串列	字典	集合
使用符號	()	[]	{}	{}
具順序性	有序	有序	無序	無序
可變／不可變	不可	可	可	可
舉例	(1, 2, 3)	[1,2,3]	{'word1':'apple'}	{1, 2, 3}

3. 解答：[4, 5, 6, 7, 8, 9, 10]

4. 解答：{'name': 'Tom', 'age': 18, 'city': '高雄', 'hobby': '籃球'}

5. 解答：['1', '3']

6. 解答：(1, 2, 6, 1, 2, 6, 1, 2, 6)

7. 解答：{'name': 'Python程式設計第二版', 'author': '許志峰'}

8. 解答：{'Andy', 'Axel'}

9. 解答：[15, 16, 17, 18, 19]

10. 解答：①[9, 7, 5, 3]
②[3, 1]

11. 解答：[1, 8, 77]
1

12. 解答：[51, 82, 48]

第六章【本章課後習題】

一、填充題

1. 內建函數　自訂函數
2. 形式參數　實際引數
3. 位置　關鍵字
4. 傳值
5. global

二、問答與實作題

1. 解答：

```
def func(a,b,c):
    x = a + b +c
    return x
```

2. 解答：12
   None
3. 解答：25 1
4. 解答：27
5. 解答：64
6. 解答：14
   14
7. 解答：11
8. 解答：
   - 自訂函數的函數名稱，可作爲呼叫lambda()函數的變數名稱。
   - 定義函數時，函數主體有多行指令；但是lambda()函數只能有一行運算式。

- 自訂函數有名稱，但lambda()函數無名稱，lambda()函數必須指定一個變數來儲存運算結果。
- 自訂函數以return指令回傳；lambda()函數由變數指定變數儲存。
- lambda()函數必須以變數名稱來呼叫lambda()函數，並依其定義傳入參數。

9. 解答：全域變數和區域變數。

10. 解答：80
    170
    270
    270

11. 解答：sorted函數與sort ()方法都是排序，兩者功能大同小異，都有reverse與key參數，差別在於sort()方法只支援list串列資料進行排序，要注意的是sort()方法沒有回傳值，會直接排序序列的內容。

12. 解答：Python的引數傳遞以可變和不可變物件來運作：
    - 不可變物件（immutable object）（如數值、字串）傳遞引數時，接近於「傳值」。
    - 可變物件（mutable object）（如串列），傳遞引數時以「傳址」處理。簡單來說，如果可變物件被修改內容值，因爲占用同一位址，會連動影響函數外部的值。

13. 解答：變數依其有效範圍分爲全域變數與區域變數：
    - 全域（global）變數：定義在函數外的變數，其有效範圍適用於整個檔案（*.py）。
    - 區域（local）變數：適用於所宣告的函數或流程控制的程式區塊，離開此範圍就會結束其生命週期。

第七章【本章課後習題】

一、填充題

1. 套件
2. __init__.py
3. import
4. 別名.函數名稱
5. __name__ __main__

二、問答與實作題

1. 解答：math模組提供許多浮點數運算的函數；time模組定義了一些與時間和日期相關的函數；datetime模組有許多操作日期以及時間的函數；os模組是作業系統相關模組。
2. 解答：我們可以將自己所寫的函數或類別放在.py文件，儲存之後就可以當作模組被匯入。將寫好的.py文件儲存在與主文件相同資料夾就可以當成模組來使用了。
3. 解答：15
   21
   32
4. 解答：localtime()函數傳回的元組資料型態中，各名稱的意義如下：
   - tm_year：元組資料索引值0，代表西元年。
   - tm_mon：元組資料索引值1，代表1-12月分。
   - tm_mday：元組資料索引值2，代表1-31日數。
   - tm_hour：元組資料索引值3，代表0-23小時。
   - tm_min：元組資料索引值4，代表0-59分。
   - tm_sec：元組資料索引值5，代表0-60的秒數，有可能閏秒。

- tm_wday：元組資料索引值6，代表星期幾，數值0-6。
- tm_yday：元組資料索引值7，代表一年中第幾天，數值爲1-366，有可能潤年。
- tm_isdst：元組資料索引值8，代表時光節約時間，0爲無時光節約時間，1爲時光節約時間。

5. 解答：16

6. 解答：

```
import random as r
for i in range(20):
    print ( r.randrange(2, 1000, 2))
```

7. 解答：

```
import 套件名稱 as 別名
```

8. 解答：以逗點「,」隔開不同的套件名稱，語法如下：

```
import 套件名稱1, 套件名稱2, ..., 套件名稱n
```

9. 解答：

```
01 import datetime
02 def isVaildDate(yy,mm,dd):
03     try:
04         return datetime.date(yy,mm,dd)
05     except:
06         return "日期錯誤"
07
08 print(isVaildDate(2017,2,30))
```

10. 解答：Python的標準函數庫裡面有非常多好用的模組，可以讓我們省下不少程式開發的時間。當程式裡會同時匯入多個模組時，這時函數名稱就有可能會重複，好在Python提供命名空間

（namespace）機制，它就像是一個容器，將模組資源限定在模組的命名空間內，避免不同模組之間同名衝突的問題。

第八章【本章課後習題】

一、填充題

1. 二進位檔
2. 檔案系統
3. 檔案物件
4. r
5. readline()
6. 「b」

二、問答與實作題

1. 解答：

例外類型	說明
FileNotFoundError	找不到檔案的錯誤
ZeroDivisionError	除零錯誤
TypeError	型別不符錯誤

2. 解答：所謂檔案物件就是一個提供存取的介面，它並非實際的檔案，當開啓檔案之後，就必須透過「檔案物件」做讀（read）或寫（write）的動作。
3. 解答：open()函數語法如下：

```
open(file, mode, encode)
```

- file：以字串來指定想要開啓檔案的路徑和檔案名稱。

- mode：以字串指定開啓檔案的存取模式，預設值爲讀取模式。
- encode：檔案的編碼模式，通常可以設定成cp950或UTF-8兩種，其中cp950就是Big-5的中文編碼模式。

4. 解答：

mode	說明
"r"	讀取模式（預設值）
"w"	寫入模式，建立新檔或覆蓋舊檔（覆蓋舊有資料）
"a"	附加（寫入）模式，建立新檔或附加於舊檔尾端
"b"	二進位模式

5. 解答：所謂「檔案指標」是記錄目前檔案寫入或讀取到哪一個位置。

6. 解答：如果在絕對路徑前面加r，來告知編譯器系統接著所使用路徑的字串是原始字串，如此一來，原先用\\來表示\就可以簡化如下：

```
file1=open(r"C:\ex\test.txt"."r")
```

7. 解答：使用open()開啓檔案後，最後必須用close()關閉檔案，但如果使用with...as語法搭配open()函數開啓檔案，當with指令結束後，檔案會自動關閉所有已開啓的檔案。

8. 解答：檔案處理時，如果文件設定的編碼和檔案讀取指定的編碼不同，會造成檔案判讀上的錯誤。例如假設test_encode.txt檔案是以UTF-8的編碼格式存檔，如果我們使用open()函數開啓檔案時，指定了cp950的編碼模式就會造成錯誤。

9. 解答：

方法	功能說明
read(n)	從檔案指標讀取指定個數的字元
readlines()	會讀取所有行再以串列回傳所有行

方法	功能說明
flush()	強制將緩衝區資料寫入檔案，並清空緩衝區
tell()	傳回目前文件的指標位置
write(str)	將指定參數的字串寫入檔案

10. 解答：例外是指程式執行時，產生了「不可預期」的特殊情形，這時Python直譯器會接手管理，發出錯誤訊息，並將程式終止。
11. 解答：檔案如果依儲存方式來分類，可以分文字檔（text file）與二進位檔（binary file）兩種。
12. 解答：「絕對路徑」指的是一個絕對的位置，並不會隨著現在目錄的改變而改變。「相對路徑」就是相對於現在目錄的路徑表示法，因此「相對路徑」所指到的檔案或目錄，會隨著現在目錄的不同而改變。
13. 解答：首先必須以open()方法開啓指定的檔案，接著使用檔案物件所提供的read()、readline()或readlilnes()方法從檔案讀取資料，最後再以close()方法關閉檔案。

第九章【本章課後習題】

一、填充題

1. 由上而下　模組化
2. 類別
3. 封裝　繼承　多型
4. public
5. 實體化
6. self
7. 兩個下底線_ _

8. 衍生
9. 覆寫
10. has_a

二、問答與實作題

1. 解答：通常宣告類別後，會將類別實體化爲物件，並將物件指派給變數，再透過這個變數來存取物件。在Python程式語言有一項重要特性，那就是每個東西都是物件，所以寫程式時，也可以在不需要將物件指派給變數的情況下去使用物件，這就是一種稱爲匿名物件（anonymous object）的程式設計技巧。
2. 解答：__init__()方法是類似其他語言中的建構子（constructor），可以做爲物件初始化的工作，也就是如果在宣告物件後，希望能指定物件中資料成員的初始值，可以使用__init__()方法來宣告。
3. 解答：①物件必須有一個可以依據的原型（prototype），而這個原型就是一般在物件導向程式設計中所稱的「類別」（class）。
   ②方法（method）是物件的動作與行爲，或稱爲成員函數（member function）。
   ③產生類別之後，還要具體化物件，稱爲「實體化」（instantiation），經由實體化後的物件，才可以透過該類別實作出該物件的應有的功能。
4. 解答：①子類別要呼叫父類別所定義的方法需使用內建函式super()
   ②isinstance()功能是判斷第一個參數的物件是否屬於第二個參數類別的一種。
   ③issubclass()功能是如果類別一是類別二所指定的子類別。
5. 解答：在類別中定義方法的第一個參數必須self敘述，如果未加self敘述，當以物件呼叫此方法時會發生TypeError。應該將程式修正

如下：

```
class Book:
    #定義方法：取得書籍名稱和價格
    def setInfo(self, title, price):
        self.title = title
        self.price = price
```

6. 解答：

```
01 class Date:
02     def setDate(self,birthday): #第一種方法
03         self.birthday =birthday
04     def showDate(self): #第二種方法
05         print("出生年月日:",self.birthday)
06 d1 = Date()
07 d1.setDate("民國67年7月3日")#呼叫方法時傳入字串
08 d1.showDate()
```

7. 解答：讓物件內的資料只能由物件本身的方法來存取，其他物件內方法不可以直接存取資料，這樣的功能稱爲「資訊隱藏」（information hiding），就是表示在此區塊中的屬性與方法是私有（private）的。當類別外部想要存取這些私有的屬性資料時，並不能直接由類別外部進行存取，必須透過該類別所提供的公用方法。

8. 解答：所謂封裝（encapsulation）是利用「類別」來實作「抽象化資料型態」（ADT）。所謂「抽象化」，是讓使用者只能接觸到這些方法（函數），而無法直接使用資料。資料抽象化的目的是方便於日後的維護，當應用程式的複雜性越高，資料抽象化做得越好，越能提高程式的再利用性和閱讀性。另外，抽象化也符合了資訊隱藏的意義，這就是「封裝」的主要作用。

9. 解答：簡單的說，多型最直接的定義就是讓具有繼承關係的不同類別物件，可以呼叫相同名稱的成員函數，並產生不同的反應結果。

第十章【本章課後習題】

一、填充題

1. mainloop
2. 平台獨立性　客製化　保存設定
3. GUI
4. Entry
5. Text

二、問答與實作題

1. 解答：在視窗模式下，使用者的操作是經由事件（event）的觸發與視窗程式溝通，使用圖形方式顯示使用者操作介面。
2. 解答：所謂事件（event）是指：「使用者執行視窗程式時，對視窗元件所採取的動作」。在視窗模式下，程式必須在元件上加入事件處理的程式，當使用者利用滑鼠或鍵盤輸入資訊時，這時特定的事件將會被觸發來處理使用者的需求。
3. 解答：grid方法是利用表格配置的方式來安排元件的位置，所有的內容會被放在這些規律的表格中，也就是用表格的形式定位。
4. 解答：(A) Label　④用來顯示唯讀的文字敘述
   (B) Button　①主要被使用於指令
   (C) Entry　②在單行的文字方塊中輸入簡單的資料
   (D) Text　③用來顯示或編輯多行文字

5. 解答：

```
01 # -*- coding: utf-8 -*-
02
03 import tkinter as tk
04 win = tk.Tk()
05 win.geometry("400x400")
06 win.title("這是我的第一支用Python寫的視窗程式")
07 win.mainloop()
```

6. 解答：

【範例：grid_ex.py】

```
01 # -*- coding: utf-8 -*-
02
03 import tkinter as tk
04 win = tk.Tk()
05 win.geometry("400x100")
06 win.title("grid版面布局的示範")
07
08 plus=tk.Button(win, width=20, text="加法範例")
09 plus.grid(column=0,row=0)
10 minus=tk.Button(win, width=20, text="減法範例")
11 minus.grid(column=0,row=1)
12 multiply=tk.Button(win, width=20, text="乘法範例")
13 multiply.grid(column=1,row=0)
14 divide=tk.Button(win, width=20, text="除法範例")
15 divide.grid(column=1,row=1)
16
17 win.mainloop()
```

7. 解答：

【範例：**Radiobutton_ex.py**】

```
01 from tkinter import *
02 wnd = Tk()
03 wnd.title('運動類型調查表')
04 def select():
05     print('你的選項是 :', var.get())
06 ft = ('標楷體', 14)
07 Label(wnd,
08     text = "請選擇喜愛的運動: ", font = ft,
09     justify = RIGHT, padx = 20).pack()
10 place = [('籃球', 1), ('桌球', 2),
11         ('游泳', 3)]
12 var = IntVar()
13 var.set(3)
14 for item, val in place:
15     Radiobutton(wnd, text = item, value = val,
16         font = ft, variable = var, padx = 20,
17         command = select).pack(anchor = NE)
```

8. 解答：在使用tkinter套件所提供的元件功能之前，必須先匯入tkinter套件。匯入的寫法有以下三種：

- import tkinter
- from tkinter import *
- import tkinter as tk

9. 解答：①不能

②需在.Tk()、.Button()、.Label()前加上tkinter或者在import tkinter後面加上as名稱（例：import tkinter as tk）

10. 解答：tkinter提供了3種布局方法：pack、grid以及place。

11. 解答：透過textvariable參數指定文字變數、透過config方法更改文字內容或屬性值。
12. 解答：Entry元件可以讓使用者輸入資料，它是單行模式，想要輸入多行就要使用Text元件。
13. 解答：選項元件有兩種：Checkbutton（核取方塊）和Radiobutton（單選按鈕）。Checkbutton提供多選的功能，而Radiobutton只能從多個項目中擇一，無法多選。
14. 解答：捲軸（Scrollbar）通常被使用在文字區域（Text）、清單方塊（Listbox）或是畫布（Canvas）等。

第十一章【本章課後習題】

一、填充題

1. linewidth
2. 圓形圖
3. subplot
4. pyplot
5. matplotlibrc

二、問答與實作題

1. 解答：pyplot模組繪製基本的圖形非常快速而且簡單，使用步驟與語法如下：
   ①設定x軸與y軸要放置的資料串列：plt.plot(x,y)
   ②設定圖表參數：例如x軸標籤名稱plt.xlabel()、y軸標籤名稱plt.ylabel()、圖表標題plt.title()
   ③輸出圖表：plt.show()
2. 解答：指定色彩的方法Matplotlib幾乎都可以使用，不管是使用色彩的

英文全名、HEX（十六進位碼）、RGB或RGBA都可以。

3. 解答：圓形圖（又稱爲餅圖或派圖，pie chart）主要的特色是能夠清楚顯示各類別數量相對於整體所占的比重，經常使用於商業統計圖表，譬如每季的銷售量、產品年度銷售量等等。

4. 解答：長條圖（bar chart）算是較常使用的圖表，是一種以視覺化長方形的長度爲變量的統計圖表。橫條圖是水平方向的長條圖，語法與bar()大致相同，差別在於width是定義數值而height是設定橫條圖的粗細，圖表的起始值從底部（bottom）改爲左邊（left）。

第十二章【本章課後習題】

一、選擇題

1. A
2. D
3. D
4. C
5. D

二、問答與實作題

1. 解答：二分搜尋法是將資料分割成兩等分，再比較鍵值與中間值的大小，如果鍵值小於中間值，可確定要找的資料在前半段的元素，否則在後半部。如此分割數次直到找到或確定不存在爲止。

2. 解答：使用分治法（divide and conquer）的方式，主要會先在資料中找到一個隨機自行設定的虛擬中間值，並依此中間值將所有打算排序的資料分爲兩部分。其中小於中間值的資料放在左邊，

而大於中間值的資料放在右邊，再以同樣的方式分別處理左右兩邊的資料，直到排序完爲止。

3. 解答：在計算機領域中可以把演算法定義成：「爲了解決某一個工作或問題，所需要有限數目的機械性或重複性指令與計算步驟。」

4. 解答：分治法（divide and conquer）是一種很重要的演算法，其核心精神在於將一個難以直接解決的大問題依照不同的概念，分割成兩個或更多的子問題，以便各個擊破，分而治之。

5. 解答：包括一個可以反覆執行的遞迴過程，與一個跳出執行過程的出口。

6. 解答：費伯那序列就是一序列的第零項是0、第一項是1，其他每一個序列中項目的值是由其本身前面兩項的值相加所得。例如：
0,1,1,2,3,5,8,13,21,34,55,….

7. 解答：144

8. 解答：1
2
6
24