Python 講義 I

第一章 基本觀念

Ch1_1:

```
print("Hello! Python")
```

Ch1_2:

```
print("Hello! Python")      # 列印字串
```

Ch1_3:

```
'''
使用三個單引號當作註解
'''
print("Hello! Python")      # 列印字串
```

Ch1_4:

```
"""
使用三個雙引號當作註解
"""
print("Hello! Python")      # 列印字串
```

章節練習

第二章 變數與基本運算

Ch2_1:

```
hourly_salary = 125
annual_salary = hourly_salary * 8 * 300
monthly_fee = 9000
annual_fee = monthly_fee * 12
annual_savings = annual_salary - annual_fee
print(annual_savings)
```

Ch2_2:

```
hourly_salary = 125                               # 設定時薪
annual_salary = hourly_salary * 8 * 300           # 計算年薪
monthly_fee = 9000                                # 設定每月花費
annual_fee = monthly_fee * 12                     # 計算每年花費
annual_savings = annual_salary - annual_fee       # 計算每年儲存金額
print(annual_savings)                             # 列出每年儲存金額
```

Ch2_3:

```
x = 10
print(x)
y = 20;print(y)     # 一行有 2 個敘述不過不鼓勵這種寫法
```

Ch2_4:

```
a = b = c = 10
x = a + b + c + 12
print(x)
# 續行方法 1
y = a +\
    b +\
    c +\
    12
print(y)
# 續行方法 2
z = ( a +      # 此處可以加上註解
      b +
      c +
      12 )
print(z)
```

Ch2_5:

```
money = 50000 * ( 1 + 0.015 ) ** 5
print("本金和是")
print(money)
```

Ch2_6:

```
PI = 3.14159
r = 5
print("圓面積:單位是平方公分")
area = PI * r * r
print(area)
circurference = 2 * PI * r
print("圓周長:單位是公分")
print(circurference)
```

章節練習

第三章　基本資料型態

Ch3_1:

```
x = 10
y = x / 3
print(x)
print(type(x))
print(y)
print(type(y))
```

Ch3_2:

```
x = 10.0
print(x)
print(type(x))
```

Ch3_3:

```
x = 10
y = x + 5.5
print(x)
print(type(x))
print(y)
print(type(y))
```

Ch3_4:

```
x = 10
print(x)
print(type(x))          #  加法前列出 x 資料型態
x = x + 5.5
print(x)
print(type(x))          #  加法後列出 x 資料型態
```

Ch3_5:

```
x = 0b1101                #  這是 2 進為整數
print(x)                 #  列出 10 進位的結果
y = 13                    #  這是 10 進為整數
print(bin(y))           #  列出轉換成 2 進位的結果
```

Ch3_6:

```
x = 0o57               #  這是 8 進為整數
print(x)              #  列出 10 進位的結果
y = 47                 #  這是 10 進為整數
print(oct(y))        #  列出轉換成 8 進位的結果
```

Ch3_7:

```
x = 0x5D               #  這是 16 進為整數
print(x)              #  列出 10 進位的結果
y = 93                 #  這是 10 進為整數
print(hex(y))        #  列出轉換成 16 進位的結果
```

Ch3_8:

```
x = 10.5
print(x)
print(type(x))       #  加法前列出 x 資料型態
y = int(x) + 5
print(y)
print(type(y))       #  加法後列出 y 資料型態
```

Ch3_9:

```
x = 10
print(x)
print(type(x))       #  加法前列出 x 資料型態
y = float(x) + 10
print(y)
print(type(y))       #  加法後列出 y 資料型態
```

Ch3_10:

```
x = -10
print("以下輸出 abs( )函數的應用")
print(x)                  #  輸出 x 變數
print(abs(x))             #  輸出 abs(x)
x = 5
y = 3
print("以下輸出 pow( )函數的應用")
print(pow(x, y))          #  輸出 pow(x,y)
x = 47.5
print("以下輸出 round(x)函數的應用")
print(x)                  #  輸出 x 變數
print(round(x))           #  輸出 round(x)
x = 48.5
print(x)                  #  輸出 x 變數
print(round(x))           #  輸出 round(x)
x = 49.5
print(x)                  #  輸出 x 變數
print(round(x))           #  輸出 round(x)
print("以下輸出 round(x,n)函數的應用")
x = 2.15
print(x)                  #  輸出 x 變數
print(round(x,1))         #  輸出 round(x,1)
x = 2.25
print(x)                  #  輸出 x 變數
print(round(x,1))         #  輸出 round(x,1)
x = 2.151
print(x)                  #  輸出 x 變數
print(round(x,1))         #  輸出 round(x,1)
x = 2.251
print(x)                  #  輸出 x 變數
print(round(x,1))         #  輸出 round(x,1)
```

Ch3_11:

```
x = True
print(x)
print(type(x))          #  列出 x 資料型態
y = False
print(y)
print(type(y))          #  列出 y 資料型態
```

Ch3_12:

```
x = True
print(int(x))
print(type(x))          #  列出 x 資料型態
y = False
print(int(y))
print(type(y))          #  列出 y 資料型態
```

Ch3_13:

```
xt = True
x = 1 + xt
print(x)
print(type(x))          #  列出 x 資料型態

yt = False
y = 1 + yt
print(y)
print(type(y))          #  列出 y 資料型態
```

Ch3_14:

```
x = "Deep Learning"       #  雙引號設定字串
print(x)
print(type(x))                                      #  列出 x 字串資料型態
y = '滴水穿石'          #  單引號設定字串
print(y)
print(type(y))                                      #  列出 y 字串資料型態
```

Ch3_15:

```
num1 = 222
num2 = 333
num3 = num1 + num2
print("以下是數值相加")
print(num3)
numstr1 = "222"
numstr2 = "333"
numstr3 = numstr1 + numstr2
print("以下是由數值組成的字串相加")
print(numstr3)
numstr4 = numstr1 + " " + numstr2
print("以下是由數值組成的字串相加，同時中間加上一空格")
print(numstr4)
str1 = "DeepStone "
str2 = "Deep Learning"
str3 = str1 + str2
print("以下是一般字串相加")
print(str3)
```

Ch3_16:

```
str1 = '''Hello ... '''
print(str1)
```

Ch3_17:

```
#以下輸出使用單引號設定的字串，需使用\'
str1 = 'I can\'t stop loving you.'
print(str1)
#以下輸出使用雙引號設定的字串，不需使用\'
str2 = "I can't stop loving you."
print(str2)
#以下輸出有\t 和\n 字元
str3 = "I \tcan't stop \nloving you."
print(str3)
```

Ch3_18:

```
num1 = 222
num2 = 333
num3 = num1 + num2
print("這是數值相加")
print(num3)
str1 = str(num1) + str(num2)
print("強制轉換為字串相加")
print(str1)
```

Ch3_19:

```
x1 = "22"
x2 = "33"
x3 = x1 + x2
print(x3)                #  列印字串相加
x4 = int(x1) + int(x2)
print(x4)                #  列印整數相加
```

Ch3_20:

```
x1 = "A"
x2 = x1 * 10
print(x2)                #  列印字串乘以整數
x3 = "ABC"
x4 = x3 * 5
print(x4)                #  列印字串乘以整數
```

Ch3_21:

```
str1 = "賴政良"
str2 = "課程名稱"
str3 = "Python 程式語言"
str4 = str1 + "\n" + str2 + "\n" + str3
print(str4)
```

Ch3_22:

```
str1 = "Hello!\nPython"
print("不含 r 字元的輸出")
print(str1)
str2 = r"Hello!\nPython"
print("含 r 字元的輸出")
print(str2)
```

Ch3_23:

```
x1 = 97
x2 = chr(x1)
print(x2)                    # 輸出數值 97 的字元
x3 = ord(x2)
print(x3)                    # 輸出字元 x3 的 Unicode(10 進位)碼值
x4 = '良'
print(hex(ord(x4)))        # 輸出字元'魁'的 Unicode(16 進位)碼值
```

Ch3_24:

```
dist = 384400                          # 地球到月亮距離
speed = 1225                           # 馬赫速度每小時 1225 公里
total_hours = dist // speed            # 計算小時數
days = total_hours // 24               # 商 = 計算天數
hours = total_hours % 24               # 餘數 = 計算小時數
print("總共需要天數")
print(days)
print("小時數")
print(hours)
```

Ch3_25:

```
dist = 384400                              # 地球到月亮距離
speed = 1225                               # 馬赫速度每小時 1225 公里
total_hours = dist // speed                # 計算小時數
days, hours = divmod(total_hours, 24)      # 商和餘數
print("總共需要天數")
print(days)
print("小時數")
print(hours)
```

Ch3_26:

```
x1 = 1
y1 = 8
x2 = 3
y2 = 10
dist = ((x1 - x2) ** 2 + ((y1 - y2) ** 2)) ** 0.5
print("2 點的距離是")
print(dist)
```

章節練習

第四章 基本輸入與輸出

Ch4_1:

```
num1 = 222
num2 = 333
num3 = num1 + num2
print("這是數值相加", num3)
str1 = str(num1) + str(num2)
print("強制轉換為字串相加", str1, sep=" $$$ ")
```

Ch4_2:

```
num1 = 222
num2 = 333
num3 = num1 + num2
print("這是數值相加", num3, end="\t")    # 以Tab鍵值位置分隔2筆資料輸出
str1 = str(num1) + str(num2)
print("強制轉換為字串相加", str1, sep=" $$$ ")
```

Ch4_3:

```
score = 90
name = "賴政良"
count = 1
print("%s 你的第 %d 次物理考試成績是 %d" % (name, count, score))
```

Ch4_4:

```
score = 90
name = "賴政良"
count = 1
formatstr = "%s 你的第 %d 次物理考試成績是 %d"
print(formatstr % (name, count, score))
```

Ch4_5:

```
x = 100
print("100 的 16 進位 = %x\n100 的 8 進位 = %o" % (x, x))
```

Ch4_6:

```
x = 10
print("整數%d \n 浮點數%f \n 字串%s" % (x, x, x))
y = 9.9
print("整數%d \n 浮點數%f \n 字串%s" % (y, y, y))
```

Ch4_7:

```
x = 100
print("x=/%6d/" % x)
y = 10.5
print("y=/%6.2f/" % y)
s = "Deep"
print("s=/%6s/" % s)
print("以下是保留格數空間不足的實例")
print("x=/%2d/" % x)
print("y=/%3.2f/" % y)
print("s=/%2s/" % s)
```

Ch4_8:

```
x = 100
print("x=/%-6d/" % x)
y = 10.5
print("y=/%-6.2f/" % y)
s = "Deep"
print("s=/%-6s/" % s)
```

Ch4_9:

```
x = 10
print("x=/%+6d/" % x)
y = 10.5
print("y=/%+6.2f/" % y)
```

Ch4_10:

```
print("  姓名      國文      英文      總分")
print("%3s    %4d      %4d      %4d" % ("洪冰儒", 98, 90, 188))
print("%3s    %4d      %4d      %4d" % ("洪雨星", 96, 95, 191))
print("%3s    %4d      %4d      %4d" % ("洪冰雨", 92, 88, 180))
print("%3s    %4d      %4d      %4d" % ("洪星宇", 93, 97, 190))
```

Ch4_11:

```
score = 90
name = "賴政良"
count = 1
print("{}你的第 {} 次物理考試成績是 {}".format(name, count, score))
```

Ch4_12:

```
score = 90
name = "賴政良"
count = 1
string = "{}你的第 {} 次物理考試成績是 {}"
print(string.format(name, count, score))
```

Ch4_12_1:

```
score = 90
name = "賴政良"
count = 1
# 以下鼓勵使用
print("{0}你的第 {1} 次物理考試成績是
{2}".format(name,count,score))

# 以下語法對但不鼓勵使用
print("{2}你的第 {1} 次物理考試成績是
{0}".format(score,count,name))
```

Ch4_12_2:

```
print("{n}你的第 {c} 次物理考試成績是 {s}".format(n="姓名
",c=1,s=90))
```

Ch4_12_3:

```
r = 5
PI = 3.14159
area = PI * r ** 2
print("/半徑{0:3d}圓面積是{1:10.2f}/".format(r,area))
```

Ch4_12_4:

```
r = 5
PI = 3.14159
area = PI * r ** 2
print("/半徑{0:3d}圓面積是{1:10.2f}/".format(r,area))
print("/半徑{0:>3d}圓面積是{1:>10.2f}/".format(r,area))
print("/半徑{0:<3d}圓面積是{1:<10.2f}/".format(r,area))
print("/半徑{0:^3d}圓面積是{1:^10.2f}/".format(r,area))
```

Ch4_12_5:

```
title = "南極旅遊講座"
print("/{0:*^20s}/".format(title))
```

Ch4_12_6:

```
sp = " " * 40
print("%s      1231 Delta Rd" % sp)
print("%s      Oxford, Mississippi" % sp)
print("%s      USA\n\n\n" % sp)
print("Dear Ivan")
print("I am pleased to inform you that your application for fall 2020 has")
print("been favorably reviewed by the Electrical and Computer Engineering")
print("Office.\n\n")
print("Best Regards")
print("Peter Malong")
```

Ch4_13:

```
fstream1 = open("d:\python\ch4\out1.txt", mode="w") #  取代先前資料
print("Testing for output", file=fstream1)
fstream1.close( )
fstream2 = open("d:\python\ch4\out2.txt", mode="a") #  附加資料後面
print("Testing for output", file=fstream2)
fstream2.close( )
```

Ch4_14:

```
name = input("請輸入姓名：")
engh = input("請輸入成績：")
print("name 資料類型是", type(name))
print("engh 資料類型是", type(engh))
```

Ch4_15:

```
print("歡迎使用成績輸入系統")
name = input("請輸入姓名：")
engh = input("請輸入英文成績：")
math = input("請輸入數學成績：")
total = int(engh) + int(math)
print("%s  你的總分是  %d" % (name, total))
```

Ch4_16:

```
clastname = input("請輸入中文姓氏：")
cfirstname = input("請輸入中文名字：")
cfullname = clastname + cfirstname
print("%s 歡迎使用本系統" % cfullname)
lastname = input("請輸入英文 Last Name：")
firstname = input("請輸入英文 First Name：")
fullname = firstname + " " + lastname
print("%s Welcome to SSE System" % fullname)
```

Ch4_17:

```
numberStr = input("請輸入數值公式 ：")
number = eval(numberStr)
print("計算結果 ：%5.2f" % number)
```

Ch4_18:

```
print("歡迎使用成績輸入系統")
name = input("請輸入姓名：")
engh = eval(input("請輸入英文成績："))
math = eval(input("請輸入數學成績："))
total = engh + math
print("%s 你的總分是 %d" % (name, total))
```

Ch4_19:

```
n1, n2, n3 = eval(input("請輸入 3 個數字："))
average = (n1 + n2 + n3) / 3
print("3 個數字平均是 %6.2f" % average)
```

Ch4_20:

```
f = input("請輸入華氏溫度：")
c = ( int(f) - 32 ) * 5 / 9
print("華氏 %s 等於攝氏 %4.1f" % (f, c))
```

Ch4_21:

```
loan = eval(input("請輸入貸款金額："))
year = eval(input("請輸入年限："))
rate = eval(input("請輸入年利率："))
monthrate = rate / (12*100)                # 改成百分比以及月利率

# 計算每月還款金額
```

```
molecules = loan * monthrate
denominator = 1 - (1 / (1 + monthrate) ** (year * 12))
monthlyPay = molecules / denominator        #  每月還款金額
totalPay = monthlyPay * year * 12           #  總共還款金額

print("每月還款金額  %d" % int(monthlyPay))
print("總共還款金額  %d" % int(totalPay))
```

Ch4_22:

```
import math

s = eval(input("請輸入正五角形邊長 ："))
area = (5 * s ** 2) / (4 * math.tan(math.pi / 5))
print("area = ", area)
```

Ch4_23:

```
import math

r = 6371                               #  地球半徑
x1, y1 = 22.2838, 114.1731          #  香港紅磡車站經緯度
x2, y2 = 25.0452, 121.5168          #  台北車站經緯度

d =
6371*math.acos(math.sin(math.radians(x1))*math.sin(math.radians(x2))
+math.cos(math.radians(x1))*math.cos(math.radians(x2))*
math.cos(math.radians(y1-y2)))

print("distance = ", d)
```

out1.txt:

```
Testing for output
```

out2.txt:

```
Testing for output
```

第五章 程式的流程控制

Ch5_1:

```
age = input("請輸入年齡: ")
if (int(age) < 20):
    print("你年齡太小")
    print("需年滿 20 歲才可以購買菸酒")
```

Ch5_2:

```
print("輸出絕對值")
num = input("請輸入任意整數值: ")
x = int(num)
if (int(x) < 0):
    x = abs(x)
print("絕對值是 %d" % int(x))
```

Ch5_2_1:

```
print("輸出絕對值")
num = input("請輸入任意整數值: ")
x = int(num)
if (int(x) < 0): x = abs(x)
print("絕對值是 %d" % int(x))
```

Ch5_3:

```
age = input("請輸入年齡: ")
if (int(age) < 20):
    print("你年齡太小")
    print("需年滿 20 歲才可以購買菸酒")
else:
    print("歡迎購買菸酒")
```

Ch5_4:

```
print("奇數偶數判斷")
num = input("請輸入任意整值: ")
rem = int(num) % 2
if (rem == 0):
    print("%d 是偶數" % int(num))
else:
    print("%d 是奇數" % int(num))
```

Ch5_5:

```
print("計算最終成績")
score = input("請輸入分數: ")
sc = int(score)
if (sc >= 90):
    print(" A")
elif (sc >= 80):
    print(" B")
elif (sc >= 70):
    print(" C")
elif (sc >= 60):
    print(" D")
else:
    print(" F")
```

Ch5_6:

```
print("計算票價")
age = input("請輸入年齡: ")
age = int(age)
ticket = 100
if age >= 80 or age <= 6:
    ticket = ticket * 0.2
    print("票價是: %d" % ticket)
elif age >= 60 or age <= 12:
    ticket = ticket * 0.5
    print("票價是: %d" % ticket)
else:
    print("票價是: %d" % ticket)
```

Ch5_6_1:

```
print("計算票價")
age = input("請輸入年齡: ")
age = int(age)
ticket = 100
if (age >= 80) or (age <= 6):
    ticket = ticket * 0.2
    print("票價是: %d" % ticket)
elif (age >= 60) or (age <= 12):
    ticket = ticket * 0.5
    print("票價是: %d" % ticket)
```

```
else:
    print("票價是: %d" % ticket)
```

Ch5_7:

```
print("判斷輸入字元類別")
ch = input("請輸入字元: ")
if ord(ch) >= ord("A") and ord(ch) <= ord("Z"):
    print("這是大寫字元")
elif ord(ch) >= ord("a") and ord(ch) <= ord("z"):
    print("這是小寫字元")
elif ord(ch) >= ord("0") and ord(ch) <= ord("9"):
    print("這是數字")
else:
    print("這是特殊字元")
```

Ch5_8:

```
print("判斷輸入年份是否潤年")
year = input("請輸入年分: ")
rem4 = int(year) % 4
rem100 = int(year) % 100
rem400 = int(year) % 400
if rem4 == 0:
    if rem100 != 0 or rem400 == 0:
        print("%s  是潤年" % year)
    else:
        print("%s  不是潤年" % year)
else:
    print("%s  不是潤年" % year)
```

Ch5_8_1:

```
flag = None
if flag == None:
    print("尚未定義 flag")

if flag:
    print("有定義")
else:
    print("尚未定義 flag")
```

Ch5_9:

```
height = input("請輸入身高(公分)：")
weight = input("請輸入體重(公斤)：")
bmi = int(weight) / ( (float(height) / 100) ** 2 )
if bmi >= 18.5 and bmi < 24:
    print("體重正常")

else:
    print("體重不正常")
```

Ch5_10:

```
ans = 0                                         #  讀者心中的數字
print("猜數字遊戲,請心中想一個  0 - 7 之間的數字，然後回答問題")

truefalse = "輸入 y 或 Y 代表有，其它代表無  : "
#  檢測 2 進位的第 1 位是否含 1
q1 = "有沒有看到心中的數字  : \n" + \
      "1, 3, 5, 7 \n"
num = input(q1 + truefalse)
print(num)
if num == "y" or num == "Y":
    ans += 1
#  檢測 2 進位的第 2 位是否含 1
truefalse = "輸入 y 或 Y 代表有，其它代表無  : "
q2 = "有沒有看到心中的數字  : \n" + \
      "2, 3, 6, 7 \n"
num = input(q2 + truefalse)
if num == "y" or num == "Y":
    ans += 2
#  檢測 2 進位的第 3 位是否含 1
truefalse = "輸入 y 或 Y 代表有，其它代表無  : "
q3 = "有沒有看到心中的數字  : \n" + \
      "4, 5, 6, 7 \n"
num = input(q3 + truefalse)
if num == "y" or num == "Y":
    ans += 4

print("讀者心中所想的數字是  : ", ans)
```

Ch5_11:

```
ans = 0                                         # 讀者心中的數字
print("猜生日日期遊戲,請回答下列 5 個問題,這個程式即可列出你的
生日")
truefalse = "輸入 y 或 Y 代表有，其它代表無 ："
# 檢測 2 進位的第 1 位是否含 1
q1 = "有沒有看到自己的生日日期 ：\n" + \
         "1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31 \n"
num = input(q1 + truefalse)
print(num)
if num == "y" or num == "Y":
        ans += 1
# 檢測 2 進位的第 2 位是否含 1
truefalse = "輸入 y 或 Y 代表有，其它代表無 ："
q2 = "有沒有看到自己的生日日期 ：\n" + \
         "2, 3, 6, 7, 10, 11, 14, 15, 18, 19, 22, 23, 26, 27, 30, 31 \n"
num = input(q2 + truefalse)
if num == "y" or num == "Y":
        ans += 2
# 檢測 2 進位的第 3 位是否含 1
truefalse = "輸入 y 或 Y 代表有，其它代表無 ："
q3 = "有沒有看到自己的生日日期 ：\n" + \
         "4, 5, 6, 7, 12, 13, 14, 15, 20, 21, 22, 23, 28, 29, 30, 31 \n"
num = input(q3 + truefalse)
if num == "y" or num == "Y":
        ans += 4      # 檢測 2 進位的第 4 位是否含 1
truefalse = "輸入 y 或 Y 代表有，其它代表無 ："
q4 = "有沒有看到自己的生日日期 ：\n" + \
         "8, 9, 10, 11, 12, 13, 14, 15, 24, 25, 26, 27, 28, 29, 30, 31 \n"
num = input(q4 + truefalse)
if num == "y" or num == "Y":
        ans += 8      # 檢測 2 進位的第 5 位是否含 1
truefalse = "輸入 y 或 Y 代表有，其它代表無 ："
q5 = "有沒有看到自己的生日日期 ：\n" + \
         "16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 \n"
num = input(q5 + truefalse)
if num == "y" or num == "Y":
        ans += 16
print("讀者的生日日期是 ：", ans)
```

Ch5_12:

```
year = eval(input("請輸入西元出生年 ："))
year -= 1900
zodiac = year % 12
if zodiac == 0:
	print("你是生肖是：鼠")
elif zodiac == 1:
	print("你是生肖是：牛")
elif zodiac == 2:
	print("你是生肖是：虎")
elif zodiac == 3:
	print("你是生肖是：兔")
elif zodiac == 4:
	print("你是生肖是：龍")
elif zodiac == 5:
	print("你是生肖是：蛇")
elif zodiac == 6:
	print("你是生肖是：馬")
elif zodiac == 7:
	print("你是生肖是：羊")
elif zodiac == 8:
	print("你是生肖是：猴")
elif zodiac == 9:
	print("你是生肖是：雞")
elif zodiac == 10:
	print("你是生肖是：狗")
else:
	print("你是生肖是：豬")
```

Ch5_13:

```
a = 3
b = 5
c = 1

r1 = (-b + (b**2-4*a*c)**0.5)/(2*a)
r2 = (-b - (b**2-4*a*c)**0.5)/(2*a)
print("r1 = %6.4f,	r2 = %6.4f" % (r1, r2))
```

Ch5_14:

```
a = 2
b = 3
c = 1
d = -2
e = 13
f = -4

x = (e*d - b*f) / (a*d - b*c)
y = (a*f - e*c) / (a*d - b*c)
print("x = %6.4f,        y = %6.4f" % (x, y))
```

第六章 串列 (List)

Ch6_1:

```
james = [23, 19, 22, 31, 18]                          #  定義 james 串列
print("列印 james 串列", james)
James = ['Lebron James',23, 19, 22, 31, 18]       #  定義 James 串列
print("列印 James 串列", James)
fruits = ['apple', 'banana', 'orange']                #  定義 fruits 串列
print("列印 fruits 串列", fruits)
cfruits = ['蘋果', '香蕉', '橘子']                      #  定義 cfruits 串列
print("列印 cfruits 串列", cfruits)
ielts = [5.5, 6.0, 6.5]                               #  定義 IELTS 成績串列
print("列印 IELTS 成績", ielts)
                                                      #  列出串列資料型態
print("串列 james 資料型態是: ",type(james))
```

Ch6_2:

```
james = [23, 19, 22, 31, 18]                          #  定義 james 串列
print("列印 james 第 1 場得分", james[0])
print("列印 james 第 2 場得分", james[1])
print("列印 james 第 3 場得分", james[2])
print("列印 james 第 4 場得分", james[3])
print("列印 james 第 5 場得分", james[4])
```

Ch6_3:

```
james = [23, 19, 22, 31, 18]                          #  定義 james 串列
#  傳統設計方式
game1 = james[0]
game2 = james[1]
game3 = james[2]
game4 = james[3]
game5 = james[4]
print("列印 james 各場次得分", game1, game2, game3, game4, game5)
# Python 高手好的設計方式
game1, game2, game3, game4, game5 = james
print("列印 james 各場次得分", game1, game2, game3, game4, game5)
```

Ch6_4:

```
james = [23, 19, 22, 31, 18]                    #  定義 james 串列
print("列印 james 第 1-3 場得分", james[0:3])
print("列印 james 第 2-4 場得分", james[1:4])
print("列印 james 第 1,3,5 場得分", james[0:6:2])
```

Ch6_5:

```
warriors = ['Curry', 'Durant', 'Iquodala', 'Bell', 'Thompson']
first3 = warriors[:3]
print("前 3 名球員",first3)
n_to_last = warriors[1:]
print("球員索引 1 到最後",n_to_last)
last3 = warriors[-3:]
print("後 3 名球員",last3)
```

Ch6_6:

```
warriors = ['Curry', 'Durant', 'Iquodala', 'Bell', 'Thompson']
print("最後一名球員",warriors[-1])
james = [23, 19, 22, 31, 18]
print("最後一場得分",james[-1])
mixs = [9, 20.5, 'DeepStone']
print("最後一筆元素",mixs[-1])
```

Ch6_7:

```
warriors = ['Curry', 'Durant', 'Iquodala', 'Bell', 'Thompson']
print(warriors[-1],warriors[-2],warriors[-3],warriors[-4],warriors[-5])
```

Ch6_8:

```
james = [23, 19, 22, 31, 18]                 #  定義 james 的 5 場比賽得分
print("最高得分  = ", max(james))
print("最低得分  = ", min(james))
print("得分總計  = ", sum(james))
```

Ch6_9:

```
James = ['Lebron James', 23, 19, 22, 31, 18] #定義 james 的 5 場比賽得分
print("最高得分  = ", max(James[1:6]))
print("最低得分  = ", min(James[1:6]))
print("得分總計  = ", sum(James[1:6]))
```

Ch6_10:

```
james = [23, 19, 22, 31, 18]              # 定義 james 的 5 場比賽得分
games = len(james)                            # 獲得場次數據
print("經過 %d 比賽最高得分 = " % games, max(james))
print("經過 %d 比賽最低得分 = " % games, min(james))
print("經過 %d 比賽得分總計 = " % games, sum(james))
```

Ch6_11:

```
james = [23, 19, 22, 31, 18]              # 定義 james 的 5 場比賽得分
print("舊的 James 比賽分數", james)
james[4] = 28
print("新的 James 比賽分數", james)
```

Ch6_12:

```
cars = ['Toyota', 'Nissan', 'Honda']
print("舊汽車銷售品牌", cars)
cars[1] = 'Ford'                    # 更改第二筆元素內容
print("新汽車銷售品牌", cars)
```

Ch6_13:

```
cars1 = ['Toyota', 'Nissan', 'Honda']
print("舊汽車銷售品牌", cars1)
cars2 = ['Audi', 'BMW']
cars1 += cars2
print("新汽車銷售品牌", cars1)
```

Ch6_14:

```
num1 = [1, 3, 5]
num2 = [2, 4, 6]
num3 = num1 + num2                 # 字串為主的串列相加
print(num3)
```

Ch6_15:

```
cars = ['toyota', 'nissan', 'honda']
nums = [1, 3, 5]
carslist = cars * 3               # 串列乘以數字
print(carslist)
numslist = nums * 5                 # 串列乘以數字
print(numslist)
```

Ch6_16:

```
James = ['Lebron James',23, 19, 22, 31, 18] #  定義 James 串列
Love = ['Kevin Love',20, 18, 30, 22, 15]       #  定義 Love 串列
game3 = James[4] + Love[4]
LKgame = James[0] + '  和  ' +   Love[0] + '第四場總得分  = '
print(LKgame, game3)
```

Ch6_17:

```
warriors = ['Curry', 'Durant', 'Iquodala', 'Bell', 'Thompson']
print("2018 年初 NBA 勇士隊主將陣容", warriors)
del warriors[3]                              #  不明原因離隊
print("2018 年末 NBA 勇士隊主將陣容", warriors)
```

Ch6_18:

```
nums1 = [1, 3, 5]
print("刪除 nums1 串列索引 1 元素前      = ",nums1)
del nums1[1]
print("刪除 nums1 串列索引 1 元素後      = ",nums1)
nums2 = [1, 2, 3, 4, 5, 6]
print("刪除 nums2 串列索引[0:2]前      = ",nums2)
del nums2[0:2]
print("刪除 nums2 串列索引[0:2]後      = ",nums2)
nums3 = [1, 2, 3, 4, 5, 6]
print("刪除 nums3 串列索引[0:6:2]前  = ",nums3)
del nums3[0:6:2]
print("刪除 nums3 串列索引[0:6:2]後  = ",nums3)
```

Ch6_19:

```
cars = ['Toyota', 'Nissan', 'Honda']
print("cars 串列長度是  = %d" %    len(cars))
if len(cars) != 0:
        del cars[0]
        print("刪除 cars 串列元素成功")
        print("cars 串列長度是  = %d" % len(cars))
else:
        print("cars 串列內沒有元素資料")
nums = []
print("nums 串列長度是  = %d" % len(nums))
if len(nums) != 0:
        del nums[0]
        print("刪除 nums 串列元素成功")
else:
        print("nums 串列內沒有元素資料")
```

Ch6_20:

```
cars = ['bmw', 'benz', 'audi']
carF = "我開的第一部車是 " + cars[1].title( )
carN = "我現在開的車子是 " + cars[0].upper( )
print(carF)
print(carN)
```

Ch6_21:

```
strN = " DeepStone "
strL = strN.lstrip( )          #  刪除字串左邊多餘空白
strR = strN.rstrip( )          #  刪除字串右邊多餘空白
strB = strN.lstrip( )          #  先刪除字串左邊多餘空白
strB = strB.rstrip( )          #  再刪除字串右邊多餘空白
strO = strN.strip( )           #  一次刪除頭尾端多餘空白
print("/%s/" % strN)
print("/%s/" % strL)
print("/%s/" % strR)
print("/%s/" % strB)
print("/%s/" % strO)
```

Ch6_22:

```
string = input("請輸入名字 ：")
print("/%s/" % string)
string = input("請輸入名字 ：")
print("/%s/" % string.strip())
```

Ch6_23:

```
title = "Ming-Chi Institute of Technology"
print("/%s/" % title.center(50))
dt = "Department of ME"
print("/%s/" % dt.ljust(50))
site = "JK Hung"
print("/%s/" % site.rjust(50))
```

Ch6_24:

```
cars = []
print("目前串列內容 = ",cars)
cars.append('Honda')
print("目前串列內容 = ",cars)
cars.append('Toyota')
print("目前串列內容 = ",cars)
cars.append('Ford')
print("目前串列內容 = ",cars)
```

Ch6_25:

```
cars = ['Honda','Toyota','Ford']
print("目前串列內容 = ",cars)
print("在索引1位置插入Nissan")
cars.insert(1,'Nissan')
print("新的串列內容 = ",cars)
print("在索引0位置插入BMW")
cars.insert(0,'BMW')
print("最新串列內容 = ",cars)
```

Ch6_26:

```
cars = ['Honda','Toyota','Ford','BMW']
print("目前串列內容 = ",cars)
print("使用pop( )刪除串列元素")
popped_car = cars.pop( )                # 刪除串列末端值
print("所刪除的串列內容是 : ", popped_car)
print("新的串列內容 = ",cars)
print("使用pop(1)刪除串列元素")
popped_car = cars.pop(1)                # 刪除串列索引為1的值
print("所刪除的串列內容是 : ", popped_car)
print("新的串列內容 = ",cars)
```

Ch6_27:

```
cars = ['Honda','bmw','Toyota','Ford','bmw']
print("目前串列內容 = ",cars)
print("使用remove( )刪除串列元素")
expensive = 'bmw'
cars.remove(expensive)                  # 刪除第一次出現的元素bmw
print("所刪除的內容是: " + expensive.upper( ) + " 因為太貴了" )
print("新的串列內容",cars)
```

Ch6_28:

```
cars = ['Honda','bmw','Toyota','Ford','bmw']
print("目前串列內容  = ",cars)
#  直接列印 cars[::-1]顛倒排序,不更改串列內容
print("列印使用[::-1]顛倒排序\n", cars[::-1])
#  更改串列內容
print("使用 reverse( )顛倒排序串列元素")
cars.reverse( )                    #  顛倒排序串列
print("新的串列內容  = ",cars)
```

Ch6_29:

```
cars = ['honda','bmw','toyota','ford']
print("目前串列內容  = ",cars)
print("使用 sort( )由小排到大")
cars.sort( )
print("排序串列結果  = ",cars)
nums = [5, 3, 9, 2]
print("目前串列內容  = ",nums)
print("使用 sort( )由小排到大")
nums.sort( )
print("排序串列結果  = ",nums)
```

Ch6_30:

```
cars = ['honda','bmw','toyota','ford']
print("目前串列內容  = ",cars)
print("使用 sort( )由大排到小")
cars.sort(reverse=True)
print("排序串列結果  = ",cars)
nums = [5, 3, 9, 2]
print("目前串列內容  = ",nums)
print("使用 sort( )由大排到小")
nums.sort(reverse=True)
print("排序串列結果  = ",nums)
```

Ch6_31:

```
cars = ['honda','bmw','toyota','ford']
print("目前串列 car 內容  = ",cars)
print("使用 sorted( )由小排到大")
cars_sorted = sorted(cars)
print("排序串列結果  = ",cars_sorted)
print("原先串列 car 內容  = ",cars)
nums = [5, 3, 9, 2]
print("目前串列 num 內容  = ",nums)
print("使用 sorted( )由小排到大")
nums_sorted = sorted(nums)
print("排序串列結果  = ",nums_sorted)
print("原先串列 num 內容  = ",nums)
```

Ch6_32:

```
cars = ['honda','bmw','toyota','ford']
print("目前串列 car 內容  = ",cars)
print("使用 sorted( )由大排到小")
cars_sorted = sorted(cars,reverse=True)
print("排序串列結果        = ",cars_sorted)
print("原先串列 car 內容  = ",cars)
nums = [5, 3, 9, 2]
print("目前串列 num 內容  = ",nums)
print("使用 sorted( )由大排到小")
nums_sorted = sorted(nums,reverse=True)
print("排序串列結果        = ",nums_sorted)
print("原先串列 num 內容  = ",nums)
```

Ch6_33:

```
cars = ['toyota', 'nissan', 'honda']
search_str = 'nissan'
i = cars.index(search_str)
print("所搜尋元素  %s  第一次出現位置索引是  %d" % (search_str, i))
nums = [7, 12, 30, 12, 30, 9, 8]
search_val = 30
j = nums.index(search_val)
print("所搜尋元素  %s  第一次出現位置索引是  %d" % (search_val, j))
```

Ch6_34:

```
James = ['Lebron James',23, 19, 22, 31, 18] #  定義 James 串列
games = len(James)                                     #  求元素數量
score_Max = max(James[1:games])                       #  最高得分
i = James.index(score_Max)                            #  場次
print(James[0], "在第  %d  場得最高分  %d" % (i, score_Max))
```

Ch6_35:

```
cars = ['toyota', 'nissan', 'honda']
search_str = 'nissan'
num1 = cars.count(search_str)
print("所搜尋元素 %s 出現 %d 次" % (search_str, num1))
nums = [7, 12, 30, 12, 30, 9, 8]
search_val = 30
num2 = nums.count(search_val)
print("所搜尋元素 %s 出現 %d 次" % (search_val, num2))
```

Ch6_36:

```
James = [['Lebron James','SF','12/30/84'],23,19,22,31,18] # 定義 James 串列
games = len(James)                                  # 求元素數量
score_Max = max(James[1:games])                     # 最高得分
i = James.index(score_Max)                          # 場次
name = James[0][0]
position = James[0][1]
born = James[0][2]
print("姓名        : ", name)
print("位置        : ", position)
print("出生日期 : ", born)
print("在第 %d 場得最高分 %d" % (i, score_Max))
```

Ch6_37:

```
James = [['Lebron James','SF','12/30/84'],23,19,22,31,18] # 定義 James 串列
games = len(James)                                          # 求元素數量
score_Max = max(James[1:games])                             # 最高得分
i = James.index(score_Max)                                  # 場次
name, position, born = James[0]
print("姓名        : ", name)
print("位置        : ", position)
print("出生日期 : ", born)
print("在第 %d 場得最高分 %d" % (i, score_Max))
```

Ch6_38:

```
cars1 = ['toyota', 'nissan', 'honda']
cars2 = ['ford', 'audi']
print("原先cars1串列內容 = ", cars1)
print("原先cars2串列內容 = ", cars2)
cars1.append(cars2)
print("執行append( )後串列cars1內容 = ", cars1)
print("執行append( )後串列cars2內容 = ", cars2)
```

Ch6_39:

```
cars1 = ['toyota', 'nissan', 'honda']
cars2 = ['ford', 'audi']
print("原先cars1串列內容 = ", cars1)
print("原先cars2串列內容 = ", cars2)
cars1.extend(cars2)
print("執行extend( )後串列cars1內容 = ", cars1)
print("執行extend( )後串列cars2內容 = ", cars2)
```

Ch6_40:

```
sc = [['王大明', 80, 95, 88, 0],
          ['林小明', 98, 97, 96, 0],
        ]
sc[0][4] = sum(sc[0][1:4])
sc[1][4] = sum(sc[1][1:4])
print(sc[0])
print(sc[1])
```

Ch6_41:

```
mysports = ['basketball', 'baseball']
friendsports = mysports
print("我喜歡的運動     = ", mysports)
print("我朋友喜歡的運動 = ", friendsports)
```

Ch6_42:

```
mysports = ['basketball', 'baseball']
friendsports = mysports
print("我喜歡的運動     = ", mysports)
print("我朋友喜歡的運動 = ", friendsports)
mysports.append('football')
friendsports.append('soccer')
```

```
print("我喜歡的最新運動        = ", mysports)
print("我朋友喜歡的最新運動  = ", friendsports)
```

Ch6_43:

```
mysports = ['basketball', 'baseball']
friendsports = mysports
print("列出 mysports 位址         = ", id(mysports))
print("列出 friendsports 位址  = ", id(friendsports))
print("我喜歡的運動          = ", mysports)
print("我朋友喜歡的運動  = ", friendsports)
mysports.append('football')
friendsports.append('soccer')
print(" --  新增運動項目後  -- ")
print("列出 mysports 位址         = ", id(mysports))
print("列出 friendsports 位址  = ", id(friendsports))
print("我喜歡的最新運動        = ", mysports)
print("我朋友喜歡的最新運動  = ", friendsports)
```

Ch6_44:

```
mysports = ['basketball', 'baseball']
friendsports = mysports[:]
print("列出 mysports 位址         = ", id(mysports))
print("列出 friendsports 位址  = ", id(friendsports))
print("我喜歡的運動          = ", mysports)
print("我朋友喜歡的運動  = ", friendsports)
mysports.append('football')
friendsports.append('soccer')
print(" --  新增運動項目後  -- ")
print("列出 mysports 位址         = ", id(mysports))
print("列出 friendsports 位址  = ", id(friendsports))
print("我喜歡的最新運動        = ", mysports)
print("我朋友喜歡的最新運動  = ", friendsports)
```

Ch6_45:

```
string = "Python"
#  正值索引
print(" string[0] = ", string[0],
          "\n string[1] = ", string[1],
          "\n string[2] = ", string[2],
          "\n string[3] = ", string[3],
          "\n string[4] = ", string[4],
          "\n string[5] = ", string[5])
# 負值索引
print(" string[-1] = ", string[-1],
          "\n string[-2] = ", string[-2],
          "\n string[-3] = ", string[-3],
          "\n string[-4] = ", string[-4],
          "\n string[-5] = ", string[-5],
          "\n string[-6] = ", string[-6])
# 多重指定觀念
s1, s2, s3, s4, s5, s6 = string
print("多重指定觀念的輸出測試 = ",s1,s2,s3,s4,s5,s6)
```

Ch6_46:

```
string = "Deep Learning"                        # 定義字串
print("列印 string 第 0-2 元素        = ", string[0:3])
print("列印 string 第 1-3 元素        = ", string[1:4])
print("列印 string 第 1,3,5 元素      = ", string[1:6:2])
print("列印 string 第 1 到最後元素  = ", string[1:])
print("列印 string 前 3 元素         = ", string[0:3])
print("列印 string 後 3 元素         = ", string[-3:])
```

Ch6_47:

```
string = "Deep Learning"                        # 定義字串
strlen = len(string)
print("字串長度", strlen)
maxstr = max(string)
print("字串最大的 unicode 碼值和字元", ord(maxstr), maxstr)
minstr = min(string)
print("字串最小的 unicode 碼值和字元", ord(minstr), minstr)
```

Ch6_48:

```
str1 = "Silicon Stone Education"
str2 = "D:\Python\ch6"
sList1 = str1.split()                           # 字串轉成串列
sList2 = str2.split("\\")                       # 字串轉成串列
print(str1, " 串列內容是 ", sList1)             # 列印串列
print(str1, " 串列字數是 ", len(sList1))        # 列印字數
print(str2, " 串列內容是 ", sList2)             # 列印串列
print(str2, " 串列字數是 ", len(sList2))        # 列印字數
```

Ch6_49:

```
path = ['D:','ch6','ch6_49.py']
connect = '\\'                          #  路徑分隔字元
print(connect.join(path))
connect = '*'                           #  普通字元
print(connect.join(path))
```

Ch6_50:

```
msg = '''CIA Mark told CIA Linda that the secret USB had given to CIA Peter'''
print("字串開頭是 CIA: ", msg.startswith("CIA"))
print("字串結尾是 CIA: ", msg.endswith("CIA"))
print("CIA 出現的次數: ",msg.count("CIA"))
msg = msg.replace('Linda','Lxx')
print("新的 msg 內容  : ", msg)
```

Ch6_51:

```
password = 'deepstone'
ch = input("請輸入字元  = ")
print("in 運算式")
if ch in password:
    print("輸入字元在密碼中")
else:
    print("輸入字元不在密碼中")

print("not in 運算式")
if ch not in password:
    print("輸入字元不在密碼中")
else:
    print("輸入字元在密碼中")
```

Ch6_52:

```
fruits = ['apple', 'banana', 'watermelon']
fruit = input("請輸入水果  = ")
if fruit in fruits:
    print("這個水果已經有了")
else:
    fruits.append(fruit)
    print("謝謝提醒已經加入水果清單: ", fruits)
```

Ch6_53:

```
x = 10
y = 10
z = 15
r = 20
print("x = %d, y = %d, z = %d, r = %d" % (x, y, z, r))
print("x 位址  = %d, y 位址  = %d, z 位址  = %d, r 位址  = %d"
          % (id(x), id(y), id(z), id(r)))
r = x                                                   # r 的值將變為 10
print("x = %d, y = %d, z = %d, r = %d" % (x, y, z, r))
print("x 位址  = %d, y 位址  = %d, z 位址  = %d, r 位址  = %d"
          % (id(x), id(y), id(z), id(r)))
```

Ch6_54:

```
x = 10
y = 10
z = 15
r = z - 5
boolean_value = x is y
print("x 位址  = %d, y 位址  = %d" % (id(x), id(y)))
print("x = %d, y = %d, " % (x, y), boolean_value)

boolean_value = x is z
print("x 位址  = %d, z 位址  = %d" % (id(x), id(z)))
print("x = %d, z = %d, " % (x, z), boolean_value)

boolean_value = x is r
print("x 位址  = %d, r 位址  = %d" % (id(x), id(r)))
print("x = %d, r = %d, " % (x, r), boolean_value)

boolean_value = x is not y
print("x 位址  = %d, y 位址  = %d" % (id(x), id(y)))
print("x = %d, y = %d, " % (x, y), boolean_value)

boolean_value = x is not z
print("x 位址  = %d, z 位址  = %d" % (id(x), id(z)))
print("x = %d, z = %d, " % (x, z), boolean_value)

boolean_value = x is not r
print("x 位址  = %d, r 位址  = %d" % (id(x), id(r)))
print("x = %d, r = %d, " % (x, r), boolean_value)
```

Python 講義 II

1. 迴圈設計
2. 元組(Tuple)
3. 字典(Dict)
4. 集合(Set)
5. 函數設計
6. 類別-物件導向程式設計
7. 設計與應用模組

迴圈設計

```
# ch7_1.py
sum = 1+2+3+4+5+6+7+8+9+10
print("總和  = ", sum)

# ch7_3.py
players = ['Curry', 'Jordan', 'James', 'Durant', 'Obama']
for player in players:
        print(player)

# ch7_4.py
players = ['Curry', 'Jordan', 'James', 'Durant', 'Obama']
for player in players:print(player)

# ch7_5.py
players = ['curry', 'jordan', 'james', 'durant', 'obama']
for player in players:
        print(player.title( ) + ", it was a great game.")
        print("我迫不及待想看下一場比賽, " + player.title( ))

# ch7_6.py
players = ['Curry', 'Jordan', 'James', 'Durant', 'Obama']
print("列印前 3 位球員")
for player in players[:3]:
        print(player)
print("列印後 3 位球員")
for player in players[-3:]:
        print(player)

# ch7_7.py
files = ['da1.c','da2.py','da3.py','da4.java']
py = []
for file in files:
        if file.endswith('.py'):        #  以.py 為副檔名
                py.append(file)                 #  加入串列
print(py)
```

```
# ch7_8.py
names = ['林大一','洪大二','東霞','大成']
lastname = []
for name in names:
    if name.startswith('洪'):        # 是否姓氏洪開頭
        lastname.append(name)       # 加入串列
print(lastname)
```

```
# ch7_9.py
fruits = ['蘋果', '香蕉', '西瓜', '水蜜桃', '百香果']
print("目前fruits串列：", fruits)
i = 1
for fruit in fruits[:]:
    fruits.remove(fruit)
    print("刪除%s " % fruit)
    print("目前fruits串列：", fruits)
```

```
# ch7_10.py
n = int(input("請輸入星號數量：")) # 定義星號的數量
for number in range(n):                  # for迴圈
    print("*",end="")                     # 列印星號
```

```
# ch7_11.py
money = 50000
rate = 0.015
n = 5
for i in range(n):
    money *= (1 + rate)
    print("第%d年本金和：%d" % ((i+1),int(money)))
```

```
# ch7_12.py
n = int(input("請輸入n值："))
sum = 0
for num in range(1,n+1):
    sum += num
print("總和 = ", sum)
```

```
# ch7_13.py
n = int(input("請輸入整數:"))
total = sum(range(n + 1))
print("從 1 到%d 的總和是  = " % n, total)

# ch7_14.py
squares = []                                # 建立空串列
n = int(input("請輸入整數:"))
if n > 10 : n = 10                          # 最大值是 10
for num in range(1, n+1):
        value = num * num                   # 元素平方
        squares.append(value)               # 加入串列
print(squares)

# ch7_15.py
squares = []                                # 建立空串列
n = int(input("請輸入整數:"))
if n > 10 : n = 10                          # 最大值是 10
for num in range(1, n+1):
        squares.append(num ** 2)            # 加入串列
print(squares)

# ch7_15_1.py
fruits = ['蘋果', '香蕉', '西瓜', '水蜜桃', '百香果']
print("目前 fruits 串列  : ", fruits)
i = 1
for fruit in fruits[:]:
        fruits.remove(fruit)
        print("刪除  %s " % fruit)
        print("目前 fruits 串列  : ", fruits)
```

```
# ch7_15_2.py
xlst = []
xlst.append(0)
xlst.append(1)
xlst.append(2)
xlst.append(3)
xlst.append(4)
xlst.append(5)
print(xlst)

# ch7_15_3.py
xlst = []
for n in range(6):
        xlst.append(n)
print(xlst)

# ch7_15_4.py
xlst = list(range(6))
print(xlst)

# ch7_15_5.py
xlst = [ n for n in range(6)]
print(xlst)

# ch7_15_5.py
xlst = [ n for n in range(6)]
print(xlst)

# ch7_16.py
n = int(input("請輸入整數:"))
if n > 10 : n = 10                          #  最大值是 10
squares = [num ** 2 for num in range(1, n+1)]
print(squares)

# ch7_17.py
celsius = [21, 25, 29]
fahrenheit = [(x * 9 / 5 + 32) for x in celsius]
print(fahrenheit)
```

```
# ch7_18.py
x = [[a, b, c] for a in range(1,20) for b in range(a,20) for c in range(b,20)
         if a ** 2 + b ** 2 == c **2]
print(x)

# ch7_19.py
colors = ["Red","Green","Blue"]
shapes = ["Circle","Square","Line"]
result = [[color,shape] for color in colors for shape in shapes]
print(result)

# ch7_20.py
colors = ["Red", "Green", "Blue"]
shapes = ["Circle", "Square"]
result = [[color, shape] for color in colors for shape in shapes]
for color, shape in result:
        print(color, shape)

# ch7_21.py
for i in range(1, 10):
        for j in range(1, 10):
                result = i * j
                print("%d*%d=%-3d" % (i, j, result), end=" ")
        print()               #  換行輸出

# ch7_22.py
for i in range(1, 10):
        for j in range(1, 10):
                if j <= i:
                        print("aa", end="")
        print()                              #  換行輸出
```

```
# ch7_23.py
print("測試 1")
for digit in range(1, 11):
    if digit == 5:
        break
    print(digit, end=', ')
print( )
print("測試 2")
for digit in range(0, 11, 2):
    if digit == 5:
        break
    print(digit, end=', ')
```

```
# ch7_24.py
players = ['Curry', 'Jordan', 'James', 'Durant', 'Obama', 'Kevin', 'Lin']
n = int(input("請輸入人數  = "))
if n > len(players) : n = len(players)    #  列出人數不大於串列元素數
index = 0                                         #  索引
for player in players:
    if index == n:
        break
    print(player, end=" ")
    index += 1                                    #  索引加 1
```

```
# ch7_25.py
scores = [94, 82, 60, 91, 88, 79, 61, 93, 99, 77]
scores.sort(reverse = True)               #  從大到小排列
count = 0
for sc in scores:
    count += 1
    print(sc, end=" ")
    if count == 5:                        #  取前 5 名成績
        break                               #  離開 for 迴圈
```

```
# ch7_26.py
scores = [33, 22, 41, 25, 39, 43, 27, 38, 40]
games = 0
for score in scores:
    if score < 30:                        #  小於 30 則不往下執行
        continue
    games += 1                              #  場次加 1
print("有%d 場得分超過 30 分" % games)

# ch7_27.py
players = [['James', 202],
           ['Curry', 193],
           ['Durant', 205],
           ['Jordan', 199],
           ['David', 211]]
for player in players:
    if player[1] < 200:
        continue
    print(player)

# ch7_28.py
num = int(input("請輸入大於 1 的整數做質數測試  = "))
if num == 2:                                        # 2 是質數所以直接
輸出
    print("%d 是質數" % num)
else:
    for n in range(2, num):                     #  用 2 .. num-1 當除數
測試
        if num % n == 0:                        #  如果整除則不是質
數
            print("%d 不是質數" % num)
            break                                   #  離開迴圈

    else:                                           #  否則是質數
        print("%d 是質數" % num)
```

```
# ch7_29.py
msg1 = '人機對話專欄,告訴我心事吧,我會重複你告訴我的心事!'
msg2 = '輸入 q 可以結束對話'
msg = msg1 + '\n' + msg2 + '\n' + '= '
input_msg = ''                      # 預設為空字串
while input_msg != 'q':
    input_msg = input(msg)
    print(input_msg)
```

```
# ch7_30.py
msg1 = '人機對話專欄,告訴我心事吧,我會重複你告訴我的心事!'
msg2 = '輸入 q 可以結束對話'
msg = msg1 + '\n' + msg2 + '\n' + '= '
input_msg = ''                        # 預設為空字串
while input_msg != 'q':
    input_msg = input(msg)
    if input_msg != 'q':            # 如果輸入不是 q 才輸出訊息
        print(input_msg)
```

```
# ch7_31.py
msg1 = '人機對話專欄,告訴我心事吧,我會重複你告訴我的心事!'
msg2 = '輸入 q 可以結束對話'
msg = msg1 + '\n' + msg2 + '\n' + '= '
active = True
while active:                         # 迴圈進行直到 active 是 False
    input_msg = input(msg)
    if input_msg != 'q':            # 如果輸入不是 q 才輸出訊息
        print(input_msg)
    else:
        active = False              # 輸入是 q 所以將 active 設為 False
```

```
# ch7_32.py
answer = 30                     # 正確數字
guess = 0                       # 設定所猜數字的初始值
while guess != answer:
    guess = int(input("請猜 1-100 間的數字 = "))
    if guess > answer:
        print("請猜小一點")
    elif guess < answer:
        print("請猜大一點")
    else:
        print("恭喜答對了")
```

```
# ch7_33.py
n = int(input("請輸入一個值 : "))
sum = 0
while n != 0:
    sum += n
    n = int(input("請輸入一個值 : "))
print("輸入總和 = ", sum)
```

```
# ch7_34.py
tuition = 50000
year = 0
while tuition < 60000:
    tuition = int(tuition * 1.05)
    year += 1
print("經過 %d 年後學費會達到或超過 60000 元 " % year)
```

```
# ch7_35.py
i = 1                       # 設定 i 初始值
while i <= 9:               # 當 i 大於 9 跳出外層迴圈
    j = 1                   # 設定 j 初始值
    while j <= 9:           # 當 j 大於 9 跳出內層迴圈
        result = i * j
        print("%d*%d=%-3d" % (i, j, result), end=" ")
        j += 1              # 內層迴圈加 1
    print()                 # 換行輸出
    i += 1                  # 外層迴圈加 1
```

```
# ch7_36.py
```

```
msg1 = '人機對話專欄,請告訴我妳喜歡吃的水果!'
msg2 = '輸入  q  可以結束對話'
msg = msg1 + '\n' + msg2 + '\n' + '= '
while True:                              #  這是 while 無限迴圈
    input_msg = input(msg)
    if input_msg == 'q':               #  輸入 q 可用 break 跳出迴圈
        break
    else:
        print("我也喜歡吃  %s " % input_msg.title( ))

# ch7_37.py
players = ['Curry', 'Jordan', 'James', 'Durant', 'Obama', 'Kevin', 'Lin']
n = int(input("請輸入人數  = "))
if n > len(players) : n = len(players)    #  列出人數不大於串列元素數
index = 0                                          #  索引 index
while index < len(players):                   #  是否 index 在串列長度範圍
    if index == n:                               #  是否達到想列出的人數
        break
    print(players[index], end=" ")
    index += 1                                     #  索引 index 加 1

# ch7_38.py
index = 0
while index <= 10:
    index += 1
    if ( index % 2 != 0 ):    #  測試是否奇數
        continue                  #  不往下執行
    print(index)                #  輸出偶數
```

元組(Tuple)

```
# ch8_1.py
numbers1 = (1, 2, 3, 4, 5)          #  定義元組元素是整數
fruits = ('apple', 'orange')       #  定義元組元素是字串
mixed = ('James', 50)                #  定義元組元素是不同型態資料
val_tuple = (10,)                    #  只有一個元素的元祖
print(numbers1)
print(fruits)
print(mixed)
print(val_tuple)
#  列出元組資料型態
print("元組 mixed 資料型態是: ",type(mixed))
```

```
# ch8_2.py
numbers1 = (1, 2, 3, 4, 5)          #  定義元組元素是整數
fruits = ('apple', 'orange')       #  定義元組元素是字串
val_tuple = (10,)                    #  只有一個元素的元祖
print(numbers1[0])                     #  以中括號索引值讀取元素內容
print(numbers1[4])
print(fruits[0],fruits[1])
print(val_tuple[0])
x, y = ('apple', 'orange')          #  有趣的應用也可以用 x,y=fruits
print(x,y)
```

```
# ch8_3.py
keys = ('magic', 'xaab', 9099)           #  定義元組元素是字串與數字
for key in keys:
        print(key)
```

```
# ch8_4.py
fruits = ('apple', 'orange')               #  定義元組元素是字串
print(fruits[0])                                #  列印元組 fruits[0]
fruits[0] = 'watermelon'                       #  將元素內容改為 watermelon
print(fruits[0])                                #  列印元組 fruits[0]
```

```
# ch8_5.py
fruits = ('apple', 'orange')              #  定義元組元素是水果
print("原始 fruits 元組元素")
for fruit in fruits:
        print(fruit)

fruits = ('watermelon', 'grape')          #  定義新的元組元素
print("\n 新的 fruits 元組元素")
for fruit in fruits:
        print(fruit)

# ch8_6.py
fruits = ('apple', 'orange', 'banana', 'watermelon', 'grape')
print(fruits[1:3])
print(fruits[:2])
print(fruits[1:])
print(fruits[-2:])
print(fruits[0:5:2])

# ch8_7.py
keys = ('magic', 'xaab', 9099)              #  定義元組元素是字串與數字
print("keys 元組長度是  %d " % len(keys))

# ch8_8.py
keys = ('magic', 'xaab', 9099)              #  定義元組元素是字串與數字
key = keys.pop( )                         #  錯誤

# ch8_9.py
keys = ('magic', 'xaab', 9099)              #  定義元組元素是字串與數字
keys.append('secret')                       #  錯誤

# ch8_10.py
keys = ('magic', 'xaab', 9099)              #  定義元組元素是字串與數字
list_keys = list(keys)                        #  將元組改為串列
list_keys.append('secret')                     #  增加元素
print("列印元組", keys)
print("列印串列", list_keys)
```

```
# ch8_11.py
keys = ['magic', 'xaab', 9099]          #  定義串列元素是字串與數字
tuple_keys = tuple(keys)                   #  將串列改為元組
print("列印串列", keys)
print("列印元組", tuple_keys)
tuple_keys.append('secret')                #  增加元素  ---  錯誤錯誤

# ch8_12.py
tup = (1, 3, 5, 7, 9)
print("tup 最大值是", max(tup))
print("tup 最小值是", min(tup))
```

字典(Dict)

```
# ch9_1.py
fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25}
noodles = {'牛肉麵':100, '肉絲麵':80, '陽春麵':60}
print(fruits)
print(noodles)
#  列出字典資料型態
print("字典 fruits 資料型態是: ",type(fruits))

# ch9_2.py
soldier0 = {'tag':'red', 'score':3}
print(soldier0)

# ch9_3.py
fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25}
noodles = {'牛肉麵':100, '肉絲麵':80, '陽春麵':60}
print("水蜜桃一斤  = ", fruits['水蜜桃'], "元")
print("牛肉麵一碗  = ", noodles['牛肉麵'], "元")

# ch9_4.py
soldier0 = {'tag':'red', 'score':3}
print("你剛打死標記  %s  小兵" % soldier0['tag'])
print("可以得到  ", soldier0['score'], "  分")

# ch9_4_1.py
fruits = {0:'西瓜', 1:'香蕉', 2:'水蜜桃'}
print(fruits[0], fruits[1], fruits[2])

# ch9_5.py
fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25}
fruits['橘子'] = 18
print(fruits)
print("橘子一斤  = ", fruits['橘子'], "元")
```

```
# ch9_6.py
soldier0 = {'tag':'red', 'score':3}
soldier0['xpos'] = 100
soldier0['ypos'] = 30
soldier0['speed'] = 'slow'
print("小兵的 x 座標    = ", soldier0['xpos'])
print("小兵的 y 座標    = ", soldier0['ypos'])
print("小兵的移動速度  = ", soldier0['speed'])
```

```
# ch9_7.py
fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25}
print("舊價格香蕉一斤  = ", fruits['香蕉'], "元")
fruits['香蕉'] = 12
print("新價格香蕉一斤  = ", fruits['香蕉'], "元")
```

```
# ch9_8.py
soldier0 = {'tag':'red', 'score':3, 'xpos':100,
                    'ypos':30, 'speed':'slow' }
print("小兵的 x,y 舊座標    = ", soldier0['xpos'], ",", soldier0['ypos'] )
if soldier0['speed'] == 'slow':               #  慢
      x_move = 1
elif soldier0['speed'] == 'medium':           #  中
      x_move = 3
else:
      x_move = 5                                         #  快
soldier0['xpos'] += x_move
print("小兵的 x,y 新座標    = ", soldier0['xpos'], ",", soldier0['ypos'] )
```

```
# ch9_9.py
fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25}
print("舊 fruits 字典內容:", fruits)
del fruits['西瓜']
print("新 fruits 字典內容:", fruits)
```

```
# ch9_9_1.py
fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25}
print("舊 fruits 字典內容:", fruits)
objKey = '西瓜'
value = fruits.pop(objKey)
print("新 fruits 字典內容:", fruits)
print("刪除內容:", objKey + ":" + str(value))

# ch9_9_2.py
fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25}
print("舊 fruits 字典內容:", fruits)
valueTup = fruits.popitem()
print("新 fruits 字典內容:", fruits)
print("刪除內容:", valueTup)

# ch9_10.py
fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25}
print("舊 fruits 字典內容:", fruits)
fruits.clear( )
print("新 fruits 字典內容:", fruits)

# ch9_11.py
fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25}
print("舊 fruits 字典內容:", fruits)
del fruits
print("新 fruits 字典內容:", fruits)          #  錯誤!  錯誤!

# ch9_12
soldier0 = {}                    #  建立空字典
print("空小兵字典", soldier0)
soldier0['tag'] = 'red'
soldier0['score'] = 3
print("新小兵字典", soldier0)
```

```
# ch9_13.py
fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25, '蘋果':18}
cfruits = fruits.copy( )
print("位址  = ", id(fruits), "    fruits 元素  = ", fruits)
print("位址  = ", id(cfruits), "    fruits 元素  = ", cfruits)

# ch9_14.py
fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25, '蘋果':18}
noodles = {'牛肉麵':100, '肉絲麵':80, '陽春麵':60}
empty_dict = {}
print("fruits 字典元素數量          = ", len(fruits))
print("noodles 字典元素數量        = ", len(noodles))
print("empty_dict 字典元素數量  = ", len(empty_dict))

# ch9_15.py
fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25}
key = input("請輸入鍵(key) = ")
value = input("請輸入值(value) = ")
if key in fruits:
        print("%s 已經在字典了" % key)
else:
        fruits[key] = value
        print("新的 fruits 字典內容  = ", fruits)

# ch9_16.py
players = {'Stephen Curry':'Golden State Warriors',
                      'Kevin Durant':'Golden State Warriors',
                      'Lebron James':'Cleveland Cavaliers',
                      'James Harden':'Houston Rockets',
                      'Paul Gasol':'San Antonio Spurs',
                    }
print("Stephen Curry 是  %s  的球員" % players['Stephen Curry'])
print("Kevin Durant 是  %s  的球員" % players['Kevin Durant'])
print("Paul Gasol 是  %s  的球員" % players['Paul Gasol'])
```

集合(Set)

```
# ch10_1.py
langs = {'Python', 'C', 'Java'}
print("列印集合  = ", langs)
print("列印類別  = ", type(langs))

# ch10_2.py
langs = {'Python', 'C', 'Java', 'Python', 'C'}
print(langs)

# ch10_3.py
#  集合由整數所組成
integer_set = {1, 2, 3, 4, 5}
print(integer_set)
#  集合由不同資料型態所組成
mixed_set = {1, 'Python', (2, 5, 10)}
print(mixed_set)
#  集合的元素是不可變的所以程式第 6 行所設定的元組元素改成
#  第 10 行串列的寫法將會產生錯誤
# mixed_set = { 1, 'Python', [2, 5, 10]}

# ch10_4.py
x = {}                              #  這是建立空字典非空集合
print("列印         = ", x)
print("列印類別  = ", type(x))

# ch10_5.py
empty_dict = {}                              #  這是建立空字典
print("列印類別  = ", type(empty_dict))
empty_set = set()                            #  這是建立空集合
print("列印類別  = ", type(empty_set))

# ch10_6.py
x = set('DeepStone mean Deep Learning')
print(x)
print(type(x))
```

```
# ch10_7.py
#  表達方式 1
fruits = ['apple', 'orange', 'apple', 'banana', 'orange']
x = set(fruits)
print(x)
#  表達方式 2
y = set(['apple', 'orange', 'apple', 'banana', 'orange'])
print(y)

# ch10_8.py
cities = set(('Beijing', 'Tokyo', 'Beijing', 'Taipei', 'Tokyo'))
print(cities)

# ch10_8_1.py
asia = {'China':'Beijing', 'Japan':'Tokyo', 'Thailand':'Bangkok'}
asiaSet = set(asia)
print(asiaSet)

# ch10_9.py
fruits1 = ['apple', 'orange', 'apple', 'banana', 'orange']
x = set(fruits1)                    #  將串列轉成集合
fruits2 = list(x)                   #  將集合轉成串列
print("原先串列資料 fruits1 = ", fruits1)
print("新的串列資料 fruits2 = ", fruits2)

# ch10_10.py
math = {'Kevin', 'Peter', 'Eric'}             #  設定參加數學夏令營成員
physics = {'Peter', 'Nelson', 'Tom'}          #  設定參加物理夏令營成員
both = math & physics
print("同時參加數學與物理夏令營的成員  ",both)

# ch10_11.py
A = {1, 2, 3, 4, 5}              #  定義集合 A
B = {3, 4, 5, 6, 7}              #  定義集合 B
#  將 intersection( )應用在 A 集合
AB = A.intersection(B)           # A 和 B 的交集
print("A 和 B 的交集是  ", AB)
#  將 intersection( )應用在 B 集合
BA = B.intersection(A)           # B 和 A 的交集
print("B 和 A 的交集是  ", BA)
```

```
# ch10_12.py
math = {'Kevin', 'Peter', 'Eric'}            #  設定參加數學夏令營成員
physics = {'Peter', 'Nelson', 'Tom'}        #  設定參加物理夏令營成員
allmember = math | physics
print("同時參加數學與物理夏令營的成員  ",allmember)
```

```
# ch10_13.py
A = {1, 2, 3, 4, 5}                        #  定義集合 A
B = {3, 4, 5, 6, 7}                        #  定義集合 B
#  將 union( )應用在 A 集合
AorB = A.union(B)                          # A 和 B 的聯集
print("A 和 B 的聯集是  ", AorB)
#  將 union( )應用在 B 集合
BorA = B.union(A)                          # B 和 A 的聯集
print("B 和 A 的聯集是  ", BorA)
```

```
# ch10_14.py
math = {'Kevin', 'Peter', 'Eric'}            #  設定參加數學夏令營成員
physics = {'Peter', 'Nelson', 'Tom'}        #  設定參加物理夏令營成員
math_only = math - physics
print("參加數學夏令營同時沒有參加物理夏令營的成員  ",math_only)
physics_only = physics - math
print("參加數學夏令營同時沒有參加物理夏令營的成員
",physics_only)
```

```
# ch10_15.py
A = {1, 2, 3, 4, 5}                        #  定義集合 A
B = {3, 4, 5, 6, 7}                        #  定義集合 B
#  將 difference( )應用在 A 集合
A_B = A.difference(B)                      # A-B 的差集
print("A-B 的差集是  ", A_B)
#  將 difference( )應用在 B 集合
B_A = B.difference(A)                      # B-A 的差集
print("B-A 的差集是  ", B_A)
```

```
# ch10_16.py
math = {'Kevin', 'Peter', 'Eric'}              #  設定參加數學夏令營成員
physics = {'Peter', 'Nelson', 'Tom'}          #  設定參加物理夏令營成員
math_sydi_physics = math ^ physics
print("沒有同時參加數學和物理夏令營的成員  ",math_sydi_physics)
```

```
# ch10_17.py
A = {1, 2, 3, 4, 5}                              #  定義集合 A
B = {3, 4, 5, 6, 7}                              #  定義集合 B
#  將 symmetric_difference( )應用在 A 集合
A_sydi_B = A.symmetric_difference(B)        # A 和 B 的對稱差集
print("A 和 B 的對稱差集是  ", A_sydi_B)
#  將 symmetric_difference( )應用在 B 集合
B_sydi_A = B.symmetric_difference(A)        # B 和 A 的對稱差集
print("B 和 A 的對稱差集是  ", B_sydi_A)
```

```
# ch10_18.py
A = {1, 2, 3, 4, 5}                              #  定義集合 A
B = {3, 4, 5, 6, 7}                              #  定義集合 B
C = {1, 2, 3, 4, 5}                              #  定義集合 C
#  列出 A 與 B 集合是否相等
print("A 與 B 集合相等", A == B)
#  列出 A 與 C 集合是否相等
print("A 與 C 集合相等", A == C)
```

```
# ch10_19.py
A = {1, 2, 3, 4, 5}                              #  定義集合 A
B = {3, 4, 5, 6, 7}                              #  定義集合 B
C = {1, 2, 3, 4, 5}                              #  定義集合 C
#  列出 A 與 B 集合是否相等
print("A 與 B 集合不相等", A != B)
#  列出 A 與 C 集合是否不相等
print("A 與 C 集合不相等", A != C)
```

```
# ch10_20.py
#  方法 1
fruits = set("orange")
print("字元 a 是屬於 fruits 集合?", 'a' in fruits)
print("字元 d 是屬於 fruits 集合?", 'd' in fruits)
#  方法 2
cars = {"Nissan", "Toyota", "Ford"}
boolean = "Ford" in cars
print("Ford in cars", boolean)
boolean = "Audi" in cars
print("Audi in cars", boolean)
```

```
# ch10_21.py
math = {'Kevin', 'Peter', 'Eric'}            #  設定參加數學夏令營成員
print("列印參加數學夏令營的成員")
for name in math:
    print(name)
```

```
# ch10_22.py
#  方法 1
fruits = set("orange")
print("字元 a 是不屬於 fruits 集合?", 'a' not in fruits)
print("字元 d 是不屬於 fruits 集合?", 'd' not in fruits)
#  方法 2
cars = {"Nissan", "Toyota", "Ford"}
boolean = "Ford" not in cars
print("Ford not in cars", boolean)
boolean = "Audi" not in cars
print("Audi not in cars", boolean)
```

```
# ch10_22_1.py
cities = { 'Taipei', 'Beijing', 'Tokyo'}
# 增加一般元素
cities.add('Chicago')
print('cities 集合內容 ', cities)
# 增加已有元素並觀察執行結果
cities.add('Beijing')
print('cities 集合內容 ', cities)
# 增加元組元素並觀察執行結果
tup = (1, 2, 3)
cities.add(tup)
print('cities 集合內容 ', cities)

# ch10_23.py
# 賦值
numset = {1, 2, 3}
deep_numset = numset
deep_numset.add(10)
print("賦值    - 觀察 numset          ", numset)
print("賦值    - 觀察 deep_numset     ", deep_numset)

# 淺拷貝 shallow copy
shallow_numset = numset.copy( )
shallow_numset.add(100)
print("淺拷貝 - 觀察 numset          ", numset)
print("淺拷貝 - 觀察 shallow_numset", shallow_numset)

# ch10_24.py
countries = {'Japan', 'China', 'France'}
print("刪除前的 countries 集合 ", countries)
countries.remove('Japan')
print("刪除後的 countries 集合 ", countries)

# ch10_25.py
animals = {'dog', 'cat', 'bird'}
print("刪除前的 animals 集合 ", animals)
animals.remove('fish')          # 刪除不存在的元素產生錯誤
print("刪除後的 animals 集合 ", animals)

# ch10_26.py
```

```
animals = {'dog', 'cat', 'bird'}
print("刪除前的 animals 集合        ", animals)
#  欲刪除元素有在集合內
animals.discard('cat')
print("刪除後的 animals 集合        ", animals)
#  欲刪除元素沒有在集合內
animals.discard('pig')
print("刪除後的 animals 集合        ", animals)
#  列印傳回值
print("刪除資料存在的傳回值      ", animals.discard('dog'))
print("刪除資料不存在的傳回值  ", animals.discard('pig'))

# ch10_27.py
animals = {'dog', 'cat', 'bird'}
print("刪除前的 animals 集合  ", animals)
ret_element = animals.pop( )
print("刪除後的 animals 集合  ", animals)
print("所刪除的元素是            ", ret_element)

# ch10_28.py
states = {'Mississippi', 'Idoho', 'Florida'}
print("刪除前的 states 集合        ", states)
states.clear( )
print("刪除前的 states 集合        ", states)

#  測試刪除空集合
empty_set = set( )
print("刪除前的 empty_set 集合  ", empty_set)
states.clear( )
print("刪除前的 empty_set 集合  ", empty_set)
```

函數設計

```
# ch11_1.py
def greeting( ):
        """我的第一個 Python 函數設計"""
        print("Python 歡迎你")
        print("祝福學習順利")
        print("謝謝")
#  以下的程式碼也可稱主程式
greeting( )
greeting( )
greeting( )
greeting( )
greeting( )
```

```
# ch11_2.py
print("Python 歡迎你")
print("祝福學習順利")
print("謝謝")
print("Python 歡迎你")
print("祝福學習順利")
print("謝謝")
print("Python 歡迎你")
print("祝福學習順利")
print("謝謝")
print("Python 歡迎你")
print("祝福學習順利")
print("謝謝")
print("Python 歡迎你")
print("祝福學習順利")
print("謝謝")
```

```
# ch11_3.py
def greeting(name):
        """Python 函數需傳遞名字 name"""
        print("Hi,", name, "Good Morning!")
greeting('Nelson')
```

```
# ch11_3_1.py
```

```
def greeting(name):
    """Python 函數需傳遞名字 name"""
    print("Hi, " + name + " Good Morning!")
greeting('Nelson')

# ch11_4.py
def greeting(name):
    """Python 函數需傳遞名字 name"""
    print("Hi, " + name + " Good Morning!")

# ch11_5.py
def subtract(x1, x2):
    """  減法設計  """
    result = x1 - x2
    print(result)                    #  輸出減法結果
print("本程式會執行  a - b  的運算")
a = int(input("a = "))
b = int(input("b = "))
print("a - b = ", end="")            #  輸出 a-b 字串,接下來輸出不跳行
subtract(a, b)

# ch11_6.py
def interest(interest_type, subject):
    """  顯示興趣和主題  """
    print("我的興趣是  " + interest_type )
    print("在  " + interest_type + "  中，最喜歡的是  " + subject)
    print( )
interest('旅遊', '敦煌')
interest('程式設計', 'Python')

# ch11_7.py
def interest(interest_type, subject):
    """  顯示興趣和主題  """
    print("我的興趣是  " + interest_type )
    print("在  " + interest_type + "  中，最喜歡的是  " + subject)
    print( )
interest(interest_type = '旅遊', subject = '敦煌')    #  位置正確
interest(subject = '敦煌', interest_type = '旅遊')    #  位置更動
```

```
# ch11_8.py
def interest(interest_type, subject = '敦煌'):
    """ 顯示興趣和主題 """
    print("我的興趣是 " + interest_type )
    print("在 " + interest_type + " 中，最喜歡的是 " + subject)
    print( )

interest('旅遊')                                                # 傳遞一個參數
interest(interest_type = '旅遊')                                # 傳遞一個參數
interest('旅遊', '張家界')                                      # 傳遞二個參數
interest(interest_type = '旅遊', subject = '張家界') # 傳遞二個參數
interest(subject = '張家界', interest_type = '旅遊') # 傳遞二個參數
interest('閱讀', '旅遊類')                          # 傳遞二個參數,不同的主題

# ch11_9.py
def greeting(name):
    """Python 函數需傳遞名字 name"""
    print("Hi, ", name, " Good Morning!")
ret_value = greeting('Nelson')
print("greeting( )傳回值 = ", ret_value)
print(ret_value, " 的 type  = ", type(ret_value))

# ch11_10.py
def greeting(name):
    """Python 函數需傳遞名字 name"""
    print("Hi, ", name, " Good Morning!")
    return                          # Python 將自動回傳 None
ret_value = greeting('Nelson')
print("greeting( )傳回值 = ", ret_value)
print(ret_value, " 的 type  = ", type(ret_value))

# ch11_10_1.py
val = None
if val:
    print("I love Java")
else:
```

```
print("I love Python")
```

```
# ch11_10_2.py
val = None
print("I love Java" if val else "I love Python")

# ch11_10_3.py
def is_None(string, x):
        if x is None:
                print("%s = None" % string)
        elif x:
                print("%s = True" % string)
        else:
                print("%s = False" % string)
is_None("空串列", [])                        #  空串列
is_None("空元組", ())                        #  空元組
is_None("空字典", {})                        #  空字典
is_None("空集合", set())                    #  空集合
is_None("None    ", None)
is_None("True    ", True)
is_None("False ", False)

# ch11_11.py
def subtract(x1, x2):
        """  減法設計  """
        result = x1 - x2
        return result                          #  回傳減法結果
print("本程式會執行  a - b  的運算")
a = int(input("a = "))
b = int(input("b = "))
print("a - b = ", subtract(a, b))      #  輸出 a-b 字串和結果
```

```
# ch11_12.py
def subtract(x1, x2):
    """  減法設計  """
    return x1 - x2                          #  回傳減法結果
def addition(x1, x2):
    """  加法設計  """
    return x1 + x2                          #  回傳加法結果

#  使用者輸入
print("請輸入運算")
print("1:加法")
print("2:減法")
op = int(input("輸入 1/2: "))
a = int(input("a = "))
b = int(input("b = "))

#  程式運算
if op == 1:
    print("a + b = ", addition(a, b))     #  輸出 a-b 字串和結果
elif op == 2:
    print("a - b = ", subtract(a, b))     #  輸出 a-b 字串和結果
else:
    print("運算方法輸入錯誤")

# ch11_13.py
def mutifunction(x1, x2):
    """  加,  減,  乘,  除四則運算  """
    addresult = x1 + x2
    subresult = x1 - x2
    mulresult = x1 * x2
    divresult = x1 / x2
    return addresult, subresult, mulresult, divresult
x1 = x2 = 10
add, sub, mul, div = mutifunction(x1, x2)
print("加法結果  = ", add)
print("減法結果  = ", sub)
print("乘法結果  = ", mul)
print("除法結果  = ", div)
```

```
# ch11_14.py
def guest_info(firstname, middlename, lastname, gender):
    """ 整合客戶名字資料 """
    if gender == "M":
        welcome = lastname + middlename + firstname + '先生歡迎你'
    else:
        welcome = lastname + middlename + firstname + '小姐歡迎妳'
    return welcome
info1 = guest_info('宇', '星', '洪', 'M')
info2 = guest_info('雨', '冰', '洪', 'F')
print(info1)
print(info2)
```

```
# ch11_15.py
def guest_info(firstname, lastname, gender, middlename = ''):
    """ 整合客戶名字資料 """
    if gender == "M":
        welcome = lastname + middlename + firstname + '先生歡迎你'
    else:
        welcome = lastname + middlename + firstname + '小姐歡迎妳'
    return welcome
info1 = guest_info('濤', '劉', 'M')
info2 = guest_info('雨', '洪', 'F', '冰')
print(info1)
print(info2)
```

```
# ch11_16.py
def build_vip(id, name):
    """ 建立 VIP 資訊 """
    vip_dict = {'VIP_ID':id, 'Name':name}
    return vip_dict
member = build_vip('101', 'Nelson')
print(member)
```

```
# ch11_17.py
def build_vip(id, name, tel = ''):
    """  建立 VIP 資訊  """
    vip_dict = {'VIP_ID':id, 'Name':name}
    if tel:
        vip_dict['Tel'] = tel
    return vip_dict
member1 = build_vip('101', 'Nelson')
member2 = build_vip('102', 'Henry', '0952222333')
print(member1)
print(member2)

# ch11_18.py
def build_vip(id, name, tel = ''):
    """  建立 VIP 資訊  """
    vip_dict = {'VIP_ID':id, 'Name':name}
    if tel:
        vip_dict['Tel'] = tel
    return vip_dict
while True:
    print("建立 VIP 資訊系統")
    idnum = input("請輸入 ID: ")
    name = input("請輸入姓名: ")
    tel = input("請輸入電話號碼: ")             #  如果直接按 Enter 可不建立此欄位
    member = build_vip(idnum, name, tel)      #  建立字典
    print(member, '\n')
    repeat = input("是否繼續(y/n)?  輸入非 y 字元可結束系統: ")
    if repeat != 'y':
        break
print("歡迎下次再使用")
```

```
# ch11_19
def product_msg(customers):
        str1 = '親愛的: '
        str2 = '本公司將在 2020 年 12 月 20 日舉行產品發表會'
        str3 = '總經理:TOM 敬上'
        for customer in customers:
                msg = str1 + customer + '\n' + str2 + '\n' + str3
                print(msg, '\n')
members = ['Damon', 'Peter', 'Mary']
product_msg(members)
```

類別-物件導向程式設計

```
# ch12_1.py
class Banks():
    ''' 定義銀行類別 '''
    bankname = 'Taipei Bank'            # 定義屬性
    def motto(self):                    # 定義方法
        return "以客為尊"

# ch12_2.py
class Banks():
    ''' 定義銀行類別 '''
    bankname = 'Taipei Bank'       # 定義屬性
    def motto(self):               # 定義方法
        return "以客為尊"
userbank = Banks()                    # 定義物件 userbank
print("目前服務銀行是 ", userbank.bankname)
print("銀行服務理念是 ", userbank.motto())

# ch12_3.py
class Banks():
    ''' 定義銀行類別 '''
    bankname = 'Taipei Bank'                          # 定義屬性
    def __init__(self, uname, money):              # 初始化方法
        self.name = uname                                # 設定存款者名字
        self.balance = money                           # 設定所存的錢

    def get_balance(self):                           # 獲得存款餘額
        return self.balance

hungbank = Banks('hung', 100)                        # 定義物件 hungbank
print(hungbank.name.title( ), " 存款餘額是 ", hungbank.get_balance())
```

```
# ch12_4.py
class Banks():
    ''' 定義銀行類別 '''
    bankname = 'Taipei Bank'                    # 定義屬性
    def __init__(self, uname, money):          # 初始化方法
        self.name = uname                          # 設定存款者名字
        self.balance = money                     # 設定所存的錢

    def save_money(self, money):                # 設計存款方法
        self.balance += money                    # 執行存款
        print("存款 ", money, " 完成")          # 列印存款完成

    def withdraw_money(self, money):            # 設計提款方法
        self.balance -= money                    # 執行提款
        print("提款 ", money, " 完成")          # 列印提款完成

    def get_balance(self):                      # 獲得存款餘額
        print(self.name.title(), " 目前餘額: ", self.balance)

hungbank = Banks('hung', 100)                   # 定義物件 hungbank
hungbank.get_balance()                             # 獲得存款餘額
hungbank.save_money(300)                            # 存款 300 元
hungbank.get_balance()                             # 獲得存款餘額
hungbank.withdraw_money(200)                        # 提款 200 元
hungbank.get_balance()                             # 獲得存款餘額
```

```
# ch12_5.py
class Banks():
    '''  定義銀行類別  '''
    bankname = 'Taipei Bank'                    #  定義屬性
    def __init__(self, uname, money):          #  初始化方法
        self.name = uname                        #  設定存款者名字
        self.balance = money                    #  設定所存的錢

    def save_money(self, money):                #  設計存款方法
        self.balance += money                   #  執行存款
        print("存款  ", money, "  完成")        #  列印存款完成

    def withdraw_money(self, money):            #  設計提款方法
        self.balance -= money                   #  執行提款
        print("提款  ", money, "  完成")        #  列印提款完成

    def get_balance(self):                      #  獲得存款餘額
        print(self.name.title(), "  目前餘額: ", self.balance)

hungbank = Banks('hung', 100)                   #  定義物件 hungbank
johnbank = Banks('john', 300)                  #  定義物件 johnbank
hungbank.get_balance()                           #  獲得 hung 存款餘
額
johnbank.get_balance()                          #  獲得 john 存款餘
額
hungbank.save_money(100)                          # hung 存款 100
johnbank.withdraw_money(150)                     # john 提款 150
hungbank.get_balance()                           #  獲得 hung 存款餘
額
johnbank.get_balance()                          #  獲得 john 存款餘
額
```

```
# ch12_6.py
class Banks():
    '''  定義銀行類別  '''

    def __init__(self, uname):                      #  初始化方法
        self.name = uname                              #  設定存款者名字
        self.balance = 0                             #  設定開戶金額是 0
        self.bankname = "Taipei Bank"               #  設定銀行名稱

    def save_money(self, money):                     #  設計存款方法
        self.balance += money                        #  執行存款
        print("存款  ", money, "  完成")              #  列印存款完成

    def withdraw_money(self, money):                 #  設計提款方法
        self.balance -= money                        #  執行提款
        print("提款  ", money, "  完成")              #  列印提款完成

    def get_balance(self):                           #  獲得存款餘額
        print(self.name.title(), "  目前餘額: ", self.balance)

hungbank = Banks('hung')                             #  定義物件 hungbank
print("目前開戶銀行  ", hungbank.bankname)            #  列出目前開戶銀行
hungbank.get_balance()                                #  獲得 hung 存款餘
額
hungbank.save_money(100)                                # hung 存款 100
hungbank.get_balance()                                #  獲得 hung 存款餘
額
```

設計與應用模組

```
# ch13_1.py
def make_icecream(*toppings):
    #  列出製作冰淇淋的配料
    print("這個冰淇淋所加配料如下")
    for topping in toppings:
        print("--- ", topping)

def make_drink(size, drink):
    #  輸入飲料規格與種類,然後輸出飲料
    print("所點飲料如下")
    print("--- ", size.title())
    print("--- ", drink.title())
make_icecream('草莓醬')
make_icecream('草莓醬', '葡萄乾', '巧克力碎片')
make_drink('large', 'coke')
```

```
# ch13_2.py
import makefood                  #  導入模組 makefood.py

makefood.make_icecream('草莓醬')
makefood.make_icecream('草莓醬', '葡萄乾', '巧克力碎片')
makefood.make_drink('large', 'coke')
```

```
# ch13_3.py
from makefood import make_icecream    #  導入模組 makefood.py 的函數 make_icecream

make_icecream('草莓醬')
make_icecream('草莓醬', '葡萄乾', '巧克力碎片')
make_drink('large', 'coke')                #  因為沒有導入此函數所以會產生錯誤
```

```
# ch13_4.py
#  導入模組 makefood.py 的 make_icecream 和 make_drink 函數
from makefood import make_icecream, make_drink
make_icecream('草莓醬')
make_icecream('草莓醬', '葡萄乾', '巧克力碎片')
make_drink('large', 'coke')
```

```
# ch13_5.py
from makefood import *          #  導入模組 makefood.py 所有函數

make_icecream('草莓醬')
make_icecream('草莓醬', '葡萄乾', '巧克力碎片 ')
make_drink('large', 'coke')

# ch13_6.py
#  使用 icecream 替代 make_icecream 函數名稱
from makefood import make_icecream   as icecream

icecream('草莓醬')
icecream('草莓醬', '葡萄乾', '巧克力碎片 ')

# ch13_7.py
import makefood as m            #  導入模組 makefood.py 的替代名稱 m

m.make_icecream('草莓醬')
m.make_icecream('草莓醬', '葡萄乾', '巧克力碎片 ')
m.make_drink('large', 'coke')
```

```
# ch13_8.py
class Banks():
    '''  定義銀行類別  '''

    def __init__(self, uname):                      #  初始化方法
        self.__name = uname                             #  設定私有存款者
名字
        self.__balance = 0                              #  設定私有開戶金額
是 0
        self.__title = "Taipei Bank"            #  設定私有銀行名稱

    def save_money(self, money):                    #  設計存款方法
        self.__balance += money                     #  執行存款
        print("存款  ", money, "  完成")          #  列印存款完成

    def withdraw_money(self, money):                #  設計提款方法
        self.__balance -= money                     #  執行提款
        print("提款  ", money, "  完成")          #  列印提款完成

    def get_balance(self):                          #  獲得存款餘額
        print(self.__name.title(), "  目前餘額: ", self.__balance)

    def bank_title(self):                           #  獲得銀行名稱
        return self.__title

class Shilin_Banks(Banks):
    '''  定義士林分行  '''
    def __init__(self, uname):
        self.title = "Taipei Bank - Shilin Branch"    #  定義分行名稱
    def bank_title(self):                           #  獲得銀行名稱
        return self.title

jamesbank = Banks('James')                          #  定義 Banks 類別物
件
print("James's banks = ", jamesbank.bank_title())   #  列印銀行名稱
jamesbank.save_money(500)                               #  存錢
jamesbank.get_balance()                               #  列出存款金額
hungbank = Shilin_Banks('Hung')                     #  定義 Shilin_Banks 類
別物件
```

```
print("Hung's banks    = ", hungbank.bank_title())      #  列印銀行名稱
```

```
# ch13_9.py
from banks import Banks                              # 導入 banks 模組的 Banks 類別

jamesbank = Banks('James')                           # 定義 Banks 類別物件
print("James's banks = ", jamesbank.bank_title())    # 列印銀行名稱
jamesbank.save_money(500)                            # 存錢
jamesbank.get_balance()                              # 列出存款金額

# ch13_10.py
# 導入 banks 模組的 Banks 和 Shilin_Banks 類別
from banks import Banks, Shilin_Banks

jamesbank = Banks('James')                           # 定義 Banks 類別物件
print("James's banks = ", jamesbank.bank_title())    # 列印銀行名稱
jamesbank.save_money(500)                            # 存錢
jamesbank.get_balance()                              # 列出存款金額
hungbank = Shilin_Banks('Hung')                      # 定義 Shilin_Banks 類別物件
print("Hung's banks    = ", hungbank.bank_title())   # 列印銀行名稱

# ch13_11.py
from banks import *                                  # 導入 banks 模組所有類別

jamesbank = Banks('James')                           # 定義 Banks 類別物件
print("James's banks = ", jamesbank.bank_title())    # 列印銀行名稱
jamesbank.save_money(500)                            # 存錢
jamesbank.get_balance()                              # 列出存款金額
hungbank = Shilin_Banks('Hung')                      # 定義 Shilin_Banks 類別物件
print("Hung's banks    = ", hungbank.bank_title())   # 列印銀行名稱

# ch13_12.py
import banks                                         # 導入 banks 模組
```

```
jamesbank = banks.Banks('James')                    #  定義 Banks 類別物件
print("James's banks = ", jamesbank.bank_title())    #  列印銀行名稱
jamesbank.save_money(500)                             #  存錢
jamesbank.get_balance()                              #  列出存款金額
hungbank = banks.Shilin_Banks('Hung')             #  定義 Shilin_Banks 類
別物件
print("Hung's banks    = ", hungbank.bank_title())      #  列印銀行名稱

# ch13_13.py
from banks1 import Banks                          #  導入 banks 模組的
Banks 類別
from shilin_Banks import Shilin_Banks      #  導入 Shilin_Banks 模組的
Shilin_Banks 類別

jamesbank = Banks('James')                          #  定義 Banks 類別物件
print("James's banks = ", jamesbank.bank_title())    #  列印銀行名稱
jamesbank.save_money(500)                          #  存錢
jamesbank.get_balance()                          #  列出存款金額
hungbank = Shilin_Banks('Hung')               #  定義 Shilin_Banks 類別物
件
print("Hung's banks    = ", hungbank.bank_title())      #  列印銀行名稱

# ch13_14.py
import random                       #  導入模組 random

n = 3
for i in range(n):
      print("1-100         : ", random.randint(1, 100))
for i in range(n):
      print("500-1000    : ", random.randint(500, 1000))
for i in range(n):
      print("2000-3000 : ", random.randint(2000, 3000))

# ch13_15.py
import random                                        #  導入模組 random

min, max = 1, 10
ans = random.randint(min, max)          #  隨機數產生答案
while True:
      yourNum = int(input("請猜 1-10 之間數字: "))
```

```
if yourNum == ans:
        print("恭喜!答對了")
        break
elif yourNum < ans:
        print("請猜大一些")
else:
        print("請猜小一些")
```

```
# ch13_16.py
import random                               #  導入模組 random

min, max = 1, 100                           #  隨機數最小與最大值設定
winPercent = int(input("請輸入莊家贏的比率(0-100)之間 :"))

while True:
    print("猜大小遊戲: L 或 l 表示大,   S 或 s 表示小, Q 或 q 則程式結
束")
    customerNum = input("= ")               #  讀取玩家輸入
    if customerNum == 'Q' or customerNum == 'q':      #  若輸入 Q 或 q
        break                                               #  程式
結束
    num = random.randint(min, max)    #  產生是否讓玩家答對的隨機
數
    if num > winPercent:                    #  隨機數在 81-100 間回應玩家
猜對
        print("恭喜!答對了\n")
    else:                                    #  隨機數在 1-80 間回應玩家
猜錯
        print("答錯了!請再試一次\n")

# ch13_17.py
import random                               #  導入模組 random

fruits = ['蘋果', '香蕉', '西瓜', '水蜜桃', '百香果']
print(random.choice(fruits))

# ch13_17_1.py
import random                               #  導入模組 random

for i in range(10):
    print(random.choice([1,2,3,4,5,6]), end=",")

# ch13_18.py
import random                               #  導入模組 random

porker = ['2', '3', '4', '5', '6', '7', '8',
          '9', '10', 'J', 'Q', 'K', 'A']
for i in range(3):
```

```
random.shuffle(porker)                  # 將次序打亂重新排列
print(porker)
```

```
# ch13_18_1.py
import random                                          #  導入模組
random

lotterys = random.sample(range(1,50), 7)        # 7 組號碼
specialNum = lotterys.pop()                          #  特別號

print("第 xxx 期大樂透號碼  ", end="")
for lottery in sorted(lotterys):                  #  排序列印大樂透號碼
    print(lottery, end=" ")
print("\n 特別號:%d" % specialNum)                   #  列印特別號

# ch13_18_2.py
import random                                          #  導入模組
random

for i in range(5):
    print("uniform(1,10) : ", random.uniform(1, 10))

# ch13_18_3.py
import random

for i in range(10):
    print(random.random())

# ch13_19.py
import time                                      #  導入模組 time

print("計算 1970 年 1 月 1 日 00:00:00 至今的秒數  = ", int(time.time()))
```

```
# ch13_20.py
import random                                # 導入模組 random
import time                                  # 導入模組 time

min, max = 1, 10
ans = random.randint(min, max)          # 隨機數產生答案
yourNum = int(input("請猜 1-10 之間數字: "))
starttime = int(time.time())       # 起始秒數
while True:
    if yourNum == ans:
        print("恭喜!答對了")
        endtime = int(time.time())    # 結束秒數
        print("所花時間: ", endtime - starttime, "  秒")
        break
    elif yourNum < ans:
        print("請猜大一些")
    else:
        print("請猜小一些")
    yourNum = int(input("請猜 1-10 之間數字: "))

# ch13_21.py
import time                                  # 導入模組 time

fruits = ['蘋果', '香蕉', '西瓜', '水蜜桃', '百香果']
for fruit in fruits:
    print(fruit)
    time.sleep(1)                            # 暫停 1 秒

# ch13_22.py
import time                                  # 導入模組 time

print(time.asctime())                      # 列出目前系統時間
```

```
# ch13_23.py
import time                                    # 導入模組 time

xtime = time.localtime()
print(xtime)                                   # 列出目前系統時間
print("年  ", xtime[0])
print("月  ", xtime[1])
print("日  ", xtime[2])
print("時  ", xtime[3])
print("分  ", xtime[4])
print("秒  ", xtime[5])
print("星期幾    ", xtime[6])
print("第幾天    ", xtime[7])
print("夏令時間  ", xtime[8])

# ch13_24.py
import sys

print("目前 Python 版本是: ", sys.version)
print("目前 Python 版本是: ", sys.version_info)

# ch13_25.py
import sys
print("請輸入字串，輸入完按 Enter = ", end = "")
msg = sys.stdin.readline()
print(msg)

# ch13_26.py
import sys
print("請輸入字串，輸入完按 Enter = ", end = "")
msg = sys.stdin.readline(8)          # 讀 8 個字
print(msg)

# ch13_27.py
import sys

sys.stdout.write("I like Python")
```

Ch6_55:

```
mysports = ['basketball', 'baseball']
```

```
sports1 = mysports                    # 賦值
sports2 = mysports[:]                 # 切片拷貝新串列
print("我喜歡的運動 = ", mysports, "位址是 = ", id(mysports))
print("運動 1          = ", sports1,   "位址是 = ", id(sports1))
print("運動 2          = ", sports2,   "位址是 = ", id(sports2))
boolean_value = mysports is sports1
print("我喜歡的運動 is 運動 1        = ", boolean_value)

boolean_value = mysports is sports2
print("我喜歡的運動 is 運動 2        = ", boolean_value)

boolean_value = mysports is not sports1
print("我喜歡的運動 is not 運動 1 = ", boolean_value)

boolean_value = mysports is not sports2
print("我喜歡的運動 is not 運動 2 = ", boolean_value)
```