

國立北斗家商 109 學年度

程式語言與設計

Python

資一 1 | 24 | 黃敏甄

目 錄

主題	頁數
擲骰子	1
印星號	2
印出數字三角形	3
整除不整除	4
印星號	7
單位矩陣	8
擲骰子直到 6 為止	9
求平方和	10
韓信點兵	11
階乘	12
是否為 3 的倍數	13
階乘函式	14
兩數的最大公因數	15
華氏轉攝氏	16
身分證字號判斷男女	17
字串	18
取出詩中的每一個句子	22
計算英文字母個數	23
顯示環境變數	24
計算程式執行時間	25
計算程式執行時間	26
計算三角函數	27
求兩點的距離	28

擲骰子

題目說明:

擲一個骰子，當出現三次點數為 1 時程式停止，否則繼續擲骰子，印出每次的點數，請使用 `break` 跳出迴圈。

程式碼:

```
1 import random #匯入亂碼
2 cnt=0 #骰子般的次數
3 while True: #當為「真」時
4     n=random.randint(1,6) #n為 骰子般的點數 點數範圍為1到6
5     print(n) #印出 骰子般的點數
6     if n==1: #如果 骰子般的點數為1
7         cnt=cnt+1 #般的次數+1
8         if cnt==3: #般的次數為3
9             break #中斷
```

* 以下為執行結果

輸出:

```
5
1
6
4
1
1
```

印星號

題目說明:

寫一程式印出下列結果

```
*  
**  
***  
****  
***  
**  
*
```

程式碼:

```
1  ▢ for i in range(1,5): #i在1到4的範圍  
2  ▢     for j in range(1,i+1): #i迴圈中的j迴圈，範圍為1到i  
3  ▢         print('*',end='') #列印出* 但不自動換行  
4  ▢     print() #換行  
5  ▢ for i in range(3,0,-1): #i在3到0之間，每次迴圈經過就-1  
6  ▢     for j in range(1,i+1): #i迴圈中的j迴圈，範圍為1到i  
7  ▢         print('*',end='') #列印出* 但不自動換行  
8  ▢     print() #換行  
9
```

* 以下為執行結果

輸出:

```
*  
**  
***  
****  
***  
**  
*
```

印出數字三角形

題目說明:

請使用巢狀迴圈印出以下數字三角形。

```
1
12
123
1234
12345
123456
```

程式碼:

```
1 for i in range(1,7): #i迴圈的範圍為1到6
2     for j in range(1,i+1): #在i迴圈的j迴圈範圍為1到i
3         print(j,end='') #印出j，並不自動換行
4     print() #換行
```

* 以下為執行結果

輸出:

```
1
12
123
1234
12345
123456
```

整除不整除

題目說明:

請列出 1 到 1000 中可以被 7 整除但無法被 21 整除的所有數字，並計算總和。
提示：使用 `continue`。

程式碼:

```
1 sum=0 #累加數
2 for i in range(1,1001): #i迴圈的範圍為1到1000
3     if i%7==0: #如果i可以被7整除
4         if i%21==0: #如果i也可以被21整除
5             continue #就繼續回到for迴圈
6         else: #如果不可以被21整除就跳到else
7             print(i) #印出i(無法被21整除的所有數字)
8             sum=sum+i #加總被7整除但無法被21整除的總和
9 print(sum) #印出被7整除但無法被21整除的總和
```

* 以下為執行結果

輸出:

```
7
14
28
35
49
56
70
77
91
98
112
119
133
140
154
161
175
182
196
203
```

輸出:

```
217
224
238
245
259
266
280
287
301
308
322
329
343
350
364
371
385
392
406
413
427
```

輸出:

```
427
434
448
455
469
476
490
497
511
518
532
539
553
560
574
581
595
602
616
623
```

輸出:

```
637
644
658
665
679
686
700
707
721
728
742
749
763
770
784
791
805
812
826
833
847
```

```
854
868
875
889
896
910
917
931
938
952
959
973
980
994
47383
```


印星號

題目說明:

請寫一個程式印出以下星號排列，第一行印出四個空白與一個星號，第二行印出三個空白與兩個星號，第三行印出兩個空白與三個星號，第四行印出一個空白與四個星號，第五行印出五個星號。

程式碼:

```
1 i=1 #迴圈次數
2 while i<=5: #當i小於等於5
3     for j in range(5-i,0,-1): #「空白數量」範圍為5-i到1 每經過一次-1
4         print(' ',end='') #印出空白 不自動換行
5     for j in range(1,i+1): #「*數量」範圍為1到i
6         print('*',end='') #印出* 不自動換行
7     print() #換行
8     i=i+1 #迴圈次數+1
```

* 以下為執行結果

輸出:

```
    *
   **
  ***
 ****
*****
```

單位矩陣

題目說明:

請寫一個程式印出單位矩陣，單位矩陣為方陣，左上到右下對角線都為 1，其他都為 0，請利用程式控制印出 4x4 的單位矩陣，

1000

0100

0010

0001

程式碼:

```
1  i=1 #迴圈次數
2  while i <=4: #當i小於等於4
3      j=1 #直排排數
4      while j<=4: #當直排排數小於等於4
5          if i ==j: #如果回圈次數等於直排
6              print('1',end='') #印出1 不自動換行
7          else: #如果第五行不符合，就跳到else
8              print('0',end='') #印出0 不自動換行
9          j=j+1 #直排排數+1
10     print() #換行
11     i=i+1 #迴圈次數+1
```

* 以下為執行結果

輸出:

```
1000
0100
0010
0001
```

擲骰子直到 6 為止

題目說明:

擲一個骰子，當點數為 6 時程式停止，否則繼續擲骰子

程式碼:

```
1 import random #匯入亂碼
2 a=random.randint(1,6) #「骰子點數」範圍為1到6
3 print(a) #印出 骰子點數
4 while a!=6: #當「骰子點數」不等於6
5     a=random.randint(1,6) #「骰子點數」範圍為1到6
6     print(a) #印出 骰子點數
```

※第四行，當骰子點數不等於 6 就繼續執行第五行，否則中斷

* 以下為執行結果

第一次測試

輸出:

```
2
6
```

第二次測試

輸出:

```
3
3
5
4
3
2
6
```

求平方和

題目說明:

使用者輸入正整數 n ，求該正整數的平方和，求 $1^{**}2+2^{**}2+3^{**}2+...+n^{**}2$

程式碼:

```
1 n=int(input('請輸入n值')) #輸入n值
2 平方和=0
3 for i in range(1,n+1): #i迴圈範圍為1到n
4     平方和=平方和+i*i #計算平方和
5 print('平方和為',平方和) #印出 平方和為(平方和的數)
6
```

* 以下為執行結果

輸入:

8

輸出:

請輸入n值

輸出:

請輸入n值[8]
平方和為 204

韓信點兵

題目說明:

找出 1 到 1000 的所有數字滿足三個三個一數餘 2，五個五個一數餘 1，七個七個一數餘 4。

程式碼:

```
1 for i in range (1,1001): #i的範圍為1到1000
2     if(i%3==2)and(i%5==1)and(i%7==4): #如果i除以3餘2、除以5餘1，且除以7餘4
3         print(i) #印出i
```

* 以下為執行結果

輸出:

```
11
116
221
326
431
536
641
746
851
956
```

階乘

題目說明:

使用者輸入正整數，求該正整數的階乘，N 階乘等於 $1*2*3*...*(N-1)*N$ 。

程式碼:

```
1 n=int(input('請輸入n值?')) #輸入階乘
2 階乘值=1
3 for i in range(1,n+1): #i的範圍為1到n
4     階乘值=階乘值*i #階乘值計算
5 print(n,'階乘為',階乘值) #印出第n階乘，階乘值為 階乘值
```

* 以下為執行結果

輸入:

輸出:

請輸入n值?

輸出:

請輸入n值?[20]
20 階乘為 2432902008176640000

是否為 3 的倍數

題目說明:

請寫一個程式判斷一個正整數是否為 3 的倍數。

提示：若輸入的數字是 3 的倍數，則每個位數的數字相加總和是 3 的倍數。可以將輸入的數字當成字串進行處理。

程式碼:

```
dp02_Ch8_習題_是否為3的倍數
1 num =input('請輸入一個數字') #使用者輸入一個數
2 sum=0
3 for i in range(len(num)): #i的範圍為 num的長度
4     sum=sum+int(num[i]) #字串轉為數字並跟sum相加=sum
5 if sum%3==0: #如果sum的餘數為0
6     print(num,'是3的倍數') #如果第五行符合的話，就印出num是3的倍數
7 else: #不符合 第五行
8     print(num,'不是3的倍數') #印出num不是3的倍數
```

* 以下為執行結果

執行中，請輸入資料



輸入

4353545345

輸出結果

請輸入一個數字[4353545345]
4353545345 不是3的倍數

階乘函式

題目說明:

設計一個程式使用者輸入 n 值，求 n 階乘，所輸入的 n 值介於 1 到 100 的正整數。請自訂一個階乘函式，輸入 n 值，回傳 n 階乘的值。

程式碼:

```
dp02_Ch8_習題_求n階乘
1  def f(x): #函數定義
2      result=1 #結果=1結果=1
3      for i in range(1,x+1): #i的範圍為1到x
4          result=result*i #結果為1*i
5      return result #回傳到result
6  n=int(input('請輸入n')) #輸入
7  print('n階乘為',f(n)) #印出 n階成為
```

* 以下為執行結果

輸入

輸出結果

```
請輸入n[10]
n階乘為 3628800
```


兩數的最大公因數

題目說明:

兩數的最大公因數定義為整除這兩數的最大數。假設兩數為 A 與 B，求最大公因數可以使用 A 與 B 的最大公因數等於 B 與「A 除以 B 餘數」的最大公因數。請自訂一個最大公因數的函式，輸入兩個數字，回傳這兩個數字的最大公因數。

程式碼:

```
dp02_Ch8_習題_求兩數的最大公因數

1 def gcd(a,b): #函數定義
2     while b!=0: #當b不等於0
3         r=a%b #a%b的餘數
4         a=b #b取代掉a
5         b=r #r取代掉b
6     return a #while不符合，回傳到a
7 a=int(input('請輸入一個整數A')) #輸入
8 b=int(input('請輸入一個整數B')) #輸入
9 print('A與B的最大公因數為',gcd(a,b)) #印出 gcd(a,b)函數
```

* 以下為執行結果

輸入

輸入

輸出結果

```
請輸入一個整數A[13]
請輸入一個整數B[52]
A與B的最大公因數為 13
```

華氏轉攝氏

題目說明:

將華氏溫度轉攝氏溫度寫成函式，輸入華氏溫度，回傳攝氏溫度。

程式碼:

```
dp02_Ch8_習題_華氏轉攝氏
1 def f(x): #函數定義
2     result=(a-32)*5/9 #計算結果
3     return result #回傳到結果
4 a=float(input('請輸入華氏溫度')) #輸入
5 c=f(a) #f(a) 會傳到 f(x)
6 print('攝氏溫度為',c) #印出 設氏溫度為c
```

* 以下為執行結果

輸入

輸出結果

```
請輸入華氏溫度[212]
攝氏溫度為 100.0
```

身分證字號判斷男女

題目說明:

可以依據身分證字號的第 2 個字元判斷是男生還是女生，即身分證字號的英文字母的下一位數值。若是 1 表示男生，若是 2 表示女生。使用者可以輸入身分證字號，判斷是男生還是女生。

程式碼:

```
dp02_Ch8_習題_身分證字號判斷男女
1 a=input('請輸入一個身分證字號') #輸入
2 if a[1]=='1': #如果字串的第一個數為1
3     print('你是男生') #印出 你是男生
4 else: #第二行不符合就到else
5     print('你是女生') #印出 你是女生
```

* 以下為執行結果

輸入

A111222333

輸出結果

請輸入一個身分證字號[A111222333]
你是男生

字串

題目說明:

計算字串長度

程式碼:

```
1 a=input() #輸入字串
2 print(len(a)) #計算字串長度
```

* 以下為執行結果

輸入

abcd1234我是天才

輸出結果

```
[abcd1234我是天才]
12
```

題目說明:

分割字串

程式碼:

```
1 a=input('輸入要分割字串') #輸入字串
2 print(a.split(',')) #分割字串
```

* 以下為執行結果

輸入

abc,123,ABC

輸出結果

```
輸入要分割字串[abc,123,ABC]
['abc', '123', 'ABC']
```

題目說明:

取代字串

程式碼:

```
1 a=input('請輸入字串')
2 b=input('尋找取代的')
3 c=input('要取代為')
4 print(a.replace(b,c)) #印出 已被取代的字串
```

* 以下為執行結果

輸入

我是誰

輸入

誰

輸入

人

輸出結果

```
請輸入字串[我是誰]
尋找取代的[誰]
要取代為[人]
我是人
```

題目說明:

尋找某字的位置值、出現次數

程式碼:

```
1 a=input('輸入字串')
2 b=input('尋找某字的位置值')
3 c=input('尋找尋找某字的出現次數')
4 print(a.find(b)) #找出a字串的b位置值
5 print(a.count(c)) #數出a字串的c出現次數
```

* 以下為執行結果

輸入

我是天才我是天才我是天才我是天才我是天才

輸入

才

輸入

我

輸出結果

```
輸入字串[我是天才我是天才我是天才我是天才我是天才]
尋找某字的位置值[才]
尋找尋找某字的出現次數[我]
3
5
```

題目說明:

把英文字串大寫改小寫、小寫改大寫

程式碼:

```
1 a=input('輸入字串') #輸入英文字串
2 print(a.upper()) #把小寫改大寫
3 print(a.lower()) #把大寫改小寫
```

* 以下為執行結果

輸入

abcdABCDefgEFG

輸出結果

```
輸入字串[abcdABCDefgEFG]
ABCDABCDEFGEFG
abcdabcdefgefg
```

取出詩中的每一個句子

題目說明:

取出詩中的每一個句子

程式碼:

```
1 poem='床前明月光，疑是地上霜。舉頭望明月，低頭思故鄉。'  
2 poem=poem.replace('。','，') #把poem字串的。改為，  
3 print(poem) #印出poem的字串  
4 result=poem[:-1].split('，') #結果為[:-1]是字串的倒數一個的前面所有字串切割  
5 print(result) #印出結果
```

* 以下為執行結果

輸出結果

```
床前明月光，疑是地上霜，舉頭望明月，低頭思故鄉，  
['床前明月光', '疑是地上霜', '舉頭望明月', '低頭思故鄉']
```


計算英文字母個數

題目說明:

計算英文字母個數

程式碼:

```
1 s='An apple a day keep doctor away' #字串
2 s=s.lower() #將s字串大寫的字母改小寫
3 char='abcdefghijklmnopqrstuvwxyz' #字串
4 for i in range(len(char)): #i的範圍為char的字串長度
5     if s.count(char[i])>0: #char的字串出現次數範圍是否大於0 如果符合就
6         print(char[i],s.count(char[i])) #印出char的第x字母、char的字串出現次數
```

* 以下為執行結果

輸出結果

```
a 6
c 1
d 2
e 3
k 1
l 1
n 1
o 2
p 3
r 1
t 1
w 1
y 2
```

顯示環境變數

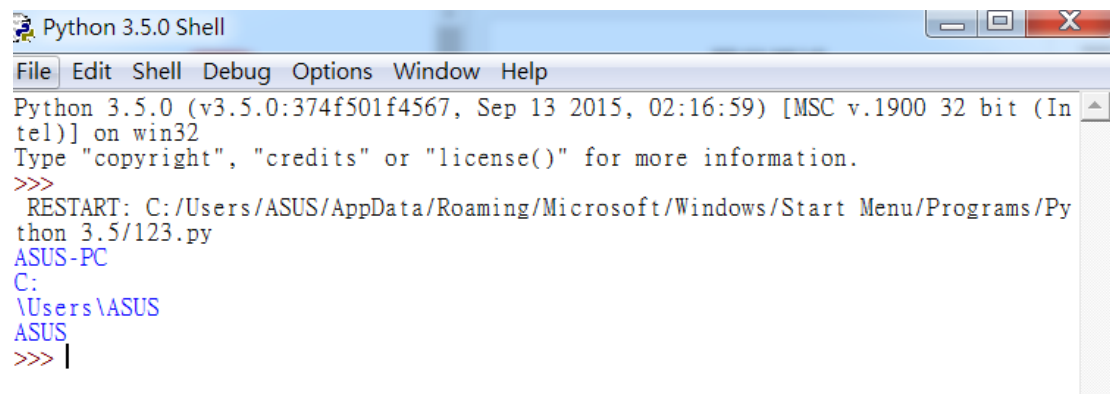
題目說明:

顯示環境變數

程式碼:

```
1 import os #匯入模組os
2 print(os.getenv('COMPUTERNAME')) #印出電腦名稱到螢幕上
3 print(os.getenv('HOMEDRIVE')) #印出主目錄所在的磁碟機到螢幕上
4 print(os.getenv('HOMEPATH')) #印出主目錄所在路徑到螢幕上
5 print(os.getenv('USERNAME')) #印出使用者名稱到螢幕上
```

* 以下為執行結果



```
Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:16:59) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/ASUS/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Python 3.5/123.py
ASUS-PC
C:
/Users\ASUS
ASUS
>>> |
```

計算程式執行時間

題目說明:

顯示目前時間

程式碼:

```
1 import datetime #匯入模組datetime
2 today=datetime.date.today() #使用模組datetime內的模組date的函式today查詢今天日期
3 print(today) #印出變數
4 print(today.year,today.month,today.day) #印出年月日
5 now=datetime.datetime.now() #使用模組datetime內的模組datetime的函式now查詢今天日期與時間
6 print(now) #印出變數
7 print(now.year,now.month,now.day) #印出年月日
8 print(now.hour,now.minute,now.second,now.microsecond) #印出變數 hour minute second microsecond
```

* 以下為執行結果

執行完畢



輸出結果

```
2021-06-11
2021 6 11
2021-06-11 03:44:41.833097
2021 6 11
3 44 41 833097
```

計算程式執行時間

題目說明:

計算程式執行時間

程式碼:

```
1 import time #匯入模組time
2 st=time.time() #使用模組time的還是time產生目前時間的物件
3 print(2**100) #印出2的100次方
4 et=time.time() #使用模組time的還是time產生目前時間的物件
5 print('執行所需時間為',et-st,'秒') #印出 值型所需時間為 et-st 秒
```

* 以下為執行結果

執行完畢



輸出結果

```
1267650600228229401496703205376
執行所需時間為 3.361701965332031e-05 秒
```

計算三角函數

題目說明:

計算三角函數

題目說明:

```
dp02_Ch8-5-7_計算三角函數
1 import math #匯入函數
2 deg=float(input('請輸入角度')) #輸入角度
3 r=deg/180*math.pi #計算半徑
4 print('Sin值為',math.sin(r)) #印出sin值
5 print('Cos值為',math.cos(r)) #印出cos值
6 print('Tan值為',math.tan(r)) #印出tan值
```

* 以下為執行結果

輸入

輸出結果

```
請輸入角度[30]
Sin值為 0.49999999999999994
Cos值為 0.8660254037844387
Tan值為 0.5773502691896257
```

求兩點的距離

題目說明:

計算三角函數

題目說明:

dp02_Ch8-5-8_求兩點的距離

```
1 import math #匯入函數
2 x1=float(input('請輸入x1')) #輸入x1座標
3 y1=float(input('請輸入y1')) #輸入y1座標
4 x2=float(input('請輸入x2')) #輸入x2座標
5 y2=float(input('請輸入y2')) #輸入y2座標
6 dis=math.sqrt(math.pow(x1-x2,2)+math.pow(y1-y2,2)) #計算兩點距離
7 print('兩點距離為',dis) #印出 兩點的距離
```

* 以下為執行結果

輸入

2

輸入

3

輸入

5

輸入

7

輸出結果

```
請輸入x1[2]
請輸入y1[3]
請輸入x2[5]
請輸入y2[7]
兩點距離為 5.0
```