

# python 程式設計

## 第七講

### 檔案

# 檔案 (一)

## ■ 檔案輸出入

### ➤ 使用 `open` 開啟檔案

# 設定 `infile` 物件開啟 `fname` 檔案準備由之讀取資料

```
>>> infile = open("fname", "r")
```

# 設定 `outfile` 物件開啟 `fname` 檔案準備輸出資料到此檔

```
>>> outfile = open("fname", "w")
```

# 設定 `outfile` 物件開啟 `fname` 檔案準備輸出資料到此檔末尾

```
>>> outfile = open("fname", "a")
```

❖ 若省略 `open` 第二個參數 ("`r`"、"`w`"、"`a`") 則代表是讀檔

### ➤ 使用 `close` 關閉檔案連結

```
>>> infile.close()
```

# 關閉 `infile` 物件與檔案的連結

```
>>> outfile.close()
```

# 關閉 `outfile` 物件與檔案的連結

# 檔案 (二)

## ■ 檔案讀取

### ➤ 使用 `open` 回傳設定輸入物件

```
infile = open("fname","r")           # 設定 infile 物件讀取 fname 檔案

for line in infile :                  # 迴圈每次由 infile 讀入一行存於 line
    print( line.strip() )             # 去除每一行的前後空格後印出

infile.close()                       # 關檔
```

❖ 檔案預設是以 utf-8 編碼，若是以 big5 編碼，需使用 `encoding` 參數

```
infile = open( "fname", encoding='big5' )
```

### ➤ 使用 `with open()` as 設定輸入物件

```
with open("fname") as infile :        # 設定 infile 物件讀取 fname 檔案

    for line in infile :               # 每次由 infile 讀入一行存於 line
        print( line.strip() )         # 去除每一行的前後空格後印出
```

❖ `infile` 離開 `with ... as` 後，檔案自動關閉

# 檔案 (三)

## ➤ 讀取資料

- `readline()` : 一次讀取一行資料
- `readlines()` : 以行為單位讀取剩餘資料
- `read(n)` : 讀取 `n` 個位元組
- `read()` : 將剩餘資料串成一個字串

```
with open("fname") as infile :  
    print( infile.readline().rstrip() )  
  
    lines = infile.readlines()           # 也可寫成 lines = list(infile)  
  
    for line in lines :  
        print( line[:12] )
```

❖ 讀到末尾時, `readline()` 回傳空字串

# 檔案 (四)

## ■ 檔案寫入

- 使用 `write` 將字串寫入檔案
- 使用 `open` 設定輸出物件以下為輸出九九乘法表於 `fname` 檔案

```
# 設定 outfile 物件將資料寫入 fname 檔案
outfile = open("fname", "w")

for x in range(1,10) :
    line = ""
    for y in range(1,10) :
        s = "{0:3} x {1:1} = {2:>2}".format(x,y,x*y)
        line += s

    # 將 line 字串透過 outfile 寫到 fname 檔案內
    outfile.write(line+"\n")

outfile.close()
```

❖若寫入的檔案已經存在，則此檔會被移除後寫入

# 檔案 (五)

- 使用 `with open as` 設定輸出物件

```
with open("fname", "w") as outfile :  
  
    for x in range(1,10) :  
        for y in range(1,10) :  
            s = "{0:3} x {1:1} = {2:>2}".format(x,y,x*y)  
            line += s  
  
    # 將 line 字串透過 outfile 寫到 fname 檔  
    outfile.write(line+"\n")
```

# 檔案 (六)

➤ **writelines**: 將字串序列合併寫入檔案

```
with open("fname", "w") as outfile :
```

```
# 型式一: 寫入 1234
```

```
outfile.writelines( [ "1" , "2" , "3" , "4" ] )
```

```
# 型式二: 寫入 1234
```

```
outfile.writelines( [ str(n) for n in range(1,5) ] )
```

```
# 型式三: 寫入 1 2 3 4 共四列
```

```
outfile.writelines( [ str(n)+"\n" for n in range(1,5) ] )
```

❖ **writelines** 僅能處理字串串列

# 檔案 (七)：混合讀檔、寫檔

## ➤ 讀檔並編行號

- 使用不同的 `open` 式子設定輸出/輸入物件

```
infile = open("data")
outfile = open("data2", "w")

for n , line in enumerate(infile) :
    line2 = "{:0>3}".format(str(n+1)) + ": " + line
    outfile.write(line2)

infile.close()
outfile.close()
```

- `with open as` 同時設定輸出/輸入物件

```
with open("data") as infile , open("data2", "w") as outfile :

    for n , line in enumerate(infile) :
        line2 = "{:0>3}".format(str(n+1)) + ": " + line
        outfile.write(line2)
```

以上也可改用 `writelines`:

```
with open("data") as infile , open("data2", "w") as outfile :

    outfile.writelines( [ "{:0>3}".format(str(n+1)) + ": " + line
                          for n , line in enumerate(infile) ] )
```



# 檔案處理自動化 (一)

- 檔案處理自動化：一個程式可同時讀入許多檔，經過處理再寫到不同檔案。
  - 檔名若有規則，則可建構檔名後開檔：
    - 將同乘數但不同被乘數的九九乘法公式輸出到 9 個不同檔案：

```
for y in range(1,10) :  
  
# 輸出的檔名依次為: mtable1、mtable2、mtable3、...  
fname = "mtable" + str(y)  
  
with open(fname,"w") as outfile :  
    for x in range(1,10) :  
        line = "{:1} x {:1} = {:>2}"  
                .format(x,y,x*y)  
        outfile.write(line+"\n")
```

例如: mtable2 檔內存

```
1 x 2 = 2  
2 x 2 = 4  
3 x 2 = 6  
4 x 2 = 8  
5 x 2 = 10  
6 x 2 = 12  
7 x 2 = 14  
8 x 2 = 16  
9 x 2 = 18
```

# 檔案處理自動化 (二)

- 合併以上九個檔案成一個完整九九乘法表

```
lines = []
for y in range(1,10) :
    fname = "mtable" + str(y)
    with open(fname) as infile :
        for i , line in enumerate(infile) :
            if y == 1 :
                lines.append( line.rstrip() )
            else :
                lines[i] += " | " + line.rstrip()

print( "\n".join(lines) )
```

輸出為:

```
1 x 1 = 1 | 1 x 2 = 2 | 1 x 3 = 3 ... 1 x 8 = 8 | 1 x 9 = 9
2 x 1 = 2 | 2 x 2 = 4 | 2 x 3 = 6 ... 2 x 8 = 16 | 2 x 9 = 18
3 x 1 = 3 | 3 x 2 = 6 | 3 x 3 = 9 ... 3 x 8 = 24 | 3 x 9 = 27
4 x 1 = 4 | 4 x 2 = 8 | 4 x 3 = 12 ... 4 x 8 = 32 | 4 x 9 = 36
5 x 1 = 5 | 5 x 2 = 10 | 5 x 3 = 15 ... 5 x 8 = 40 | 5 x 9 = 45
6 x 1 = 6 | 6 x 2 = 12 | 6 x 3 = 18 ... 6 x 8 = 48 | 6 x 9 = 54
7 x 1 = 7 | 7 x 2 = 14 | 7 x 3 = 21 ... 7 x 8 = 56 | 7 x 9 = 63
8 x 1 = 8 | 8 x 2 = 16 | 8 x 3 = 24 ... 8 x 8 = 64 | 8 x 9 = 72
9 x 1 = 9 | 9 x 2 = 18 | 9 x 3 = 27 ... 9 x 8 = 72 | 9 x 9 = 81
```

# 時雨量檔 → 多小時累積雨量檔 (一)

- 有個檔案內存某地區十個測站一天所記錄到的時雨量 (mm)，檔案資料如下：

```
4.32 17.03 9.29 7.31 10.02 9.34 11.52 12.74 14.46 3.97
2017-03-31:01 0 0 0 0 0 0 0 0 0 0
2017-03-31:02 0 0 0 0 0 0 0 0 0 0
2017-03-31:03 0 0 0 0 0 0 0 0 0 0
2017-03-31:04 0 0 0 0 0 0 0 0 0 0
2017-03-31:05 0 0 0 0 0 0 0 0 0 0
2017-03-31:06 0 0 0 0 0 0 0 0 0 0
2017-03-31:07 0 0 0 0 0 0 0 0 0 0
2017-03-31:08 0 0 0 0 0 0 0 0 0 0
2017-03-31:09 0 0 0 0 0 0 0 0 0 0
2017-03-31:10 0 0 0 0 0 0 0 0 0 0
2017-03-31:11 0 0 0 0 0 0 0 0 0 0
2017-03-31:12 0 0 0 0 0 0 0 0 0 0
2017-03-31:13 0 5 0 0 0 0 0 0 0 0
2017-03-31:14 1 1 3 0 0 1 0 0 0 0
2017-03-31:15 1 2 0 0 0 0 0 0 0 0
2017-03-31:16 2 0 0 0 0 0 0 0 0 0
2017-03-31:17 0 0 0 0 0 0 0 0 0 1
2017-03-31:18 0 0 0 0 0 0 0 0 0 1
2017-03-31:19 9 4 3 2 1 3 1 0 0 1
2017-03-31:20 6 6 5 4 4 5 5 4 3 3
2017-03-31:21 2 2 3 4 4 4 6 3 1 3
2017-03-31:22 2 1 2 3 3 2 4 3 1 2
2017-03-31:23 3 3 2 2 1 2 3 2 0 2
2017-03-31:24 5 3 3 4 4 4 5 5 4 2
```

# 時雨量檔 → 多小時累積雨量檔 (二)

- 檔案的第一列為各個測站的雨量百分加權數，整個地區的平均雨量為各個測站的雨量乘上對應的加權數之和。某生要讀入此時雨量檔 ("**rain.dat**") 造出指定的多個小時累積雨量檔。例如：四小時累積雨量為以每四個小時為累積單位的雨量數據 ("**rain4.dat**")，造出的資料檔如下：

4.32	17.03	9.29	7.31	10.02	9.34	11.52	12.74	14.46	3.97
2017-03-31:04	0	0	0	0	0	0	0	0	0
2017-03-31:08	0	0	0	0	0	0	0	0	0
2017-03-31:12	0	0	0	0	0	0	0	0	0
2017-03-31:16	4	8	3	0	0	1	0	0	0
2017-03-31:20	15	10	8	6	5	8	6	4	3
2017-03-31:24	12	9	10	13	12	12	18	13	6

# 時雨量檔 → 多小時累積雨量檔 (三)

```
n = int( input( "> " ) )
ofile = "rain" + str(n) + ".dat"

with open("rain.dat") as infile , open(ofile,"w") as outfile :

    # 讀取第一列
    weight = infile.readline()
    outfile.write(weight)

    # 第二列以後
    for i , line in enumerate(infile) :

        dhr , *rain = line.split()

        rain = list( map( int , rain ) )

        if i%n == 0 :
            sum = rain
        else :

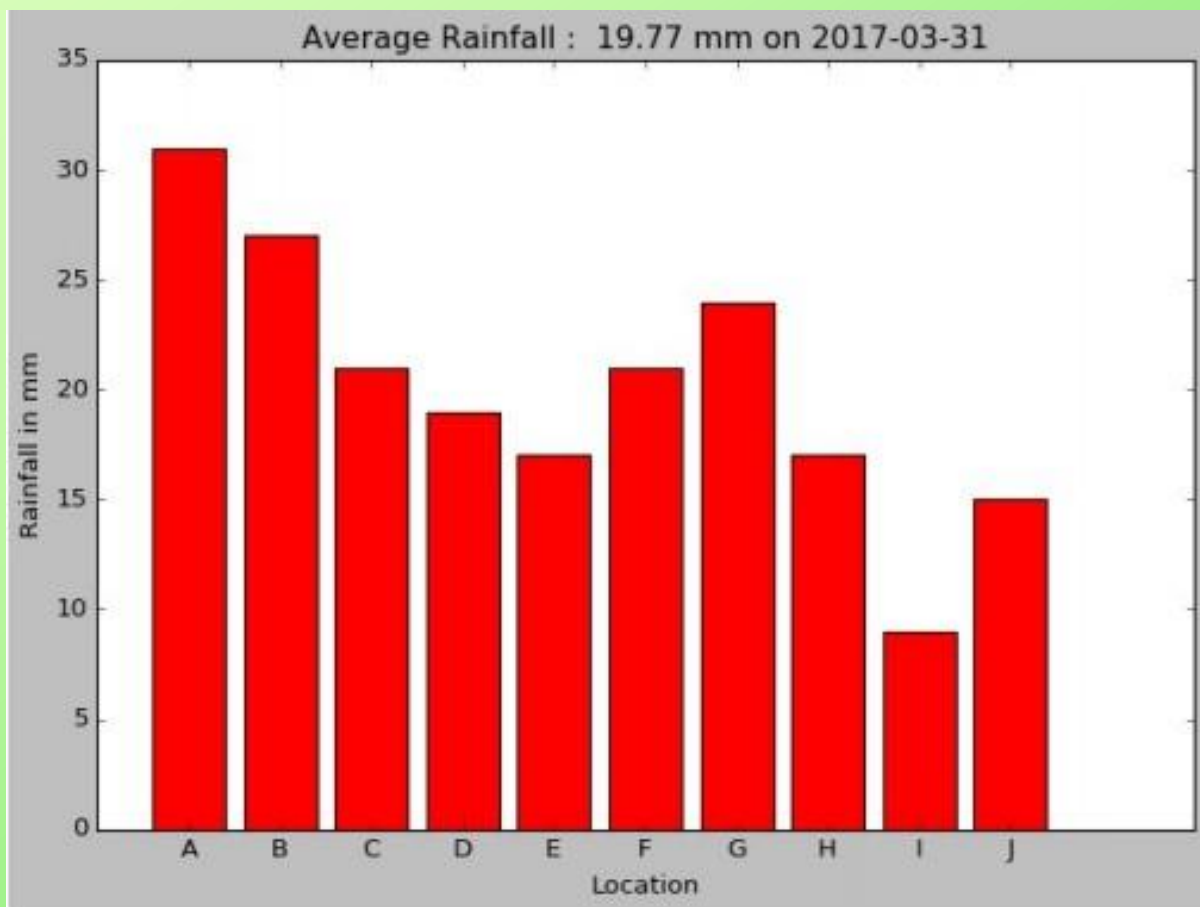
            for k , r in enumerate(rain) : sum[k] += r

            oline = dhr + "".join( map( lambda x : "{:>3}".format(x) , sum ) )

    if (i+1)%n == 0 :
        outfile.write(oline+"\n")
```

# 雨量直條圖 (一)

- 讀入上題的時雨量檔案，畫出各測站當天所量測到的雨量直條圖。



# 雨量直條圖 (二)

```
import pylab

pylab.figure(facecolor='white')

with open( "rain.dat" ) as infile :

    # 百分加權數
    weight = infile.readline().split()

    sumloc = [0] * len(weight)

    for line in infile :

        dhr , *srainfall = line.strip().split()

        rainfall = [ int(x) for x in srainfall ]

        # 各個測站一天的雨量總和
        sumloc = list( map( lambda x,y:x+y , sumloc , rainfall ) )
```



# 雨量直條圖 (三)

# 一天的平均雨量

```
total = sum( map( lambda x , y : x*float(y)/100 , sumloc , weight ) )  
date , foo = dhr.split(':')
```

```
xs = [ x+1 for x in range(len(weight)) ]
```

# 給 xs 與 sumloc 畫直條圖

```
pylab.bar(xs,sumloc,align='center',color='red')
```

# 設定各直條圖的刻度文字

```
pylab.xticks(xs,[ chr(ord('A')+i) for i in range(len(weight))])
```

# 設定 x 軸與 y 軸文字

```
pylab.xlabel('Location')
```

```
pylab.ylabel('Rainfall in mm')
```

# 設定圖案文字

```
pylab.title( 'Average Rainfall : ' + "{:6.2f}".format(avgrain) +  
            ' mm on ' + date, color='red' )
```

# 儲存圖形

```
pylab.savefig('rain_vbar.jpg')
```

```
pylab.show()
```



# 世界各國教育支出比例排名 (一)

- 百科網頁中，有著各個國家在教育支出上佔該國 GDP 百分比的統計表，資料如下：

```
Afghanistan 3.1 2012 [2]
Albania 3.3 2012 [1]
Algeria 4.3 2012 [1]
Andorra 3 2012 [1]
Angola 3.5 2012 [1]
Antigua and Barbuda 2.5 2012 [1]
Argentina 5.8 2012 [1]
Armenia 3.1 2012 [1]
Australia 5.1 2012 [1]
Austria 6 2012 [1]
Azerbaijan 2.8 2012 [1]
Bahamas 2.8 2000 [1]
Bahrain 2.9 2012 [1]
Bangladesh 2.2 2012 [1]
Barbados 7.5 2012 [1]
Belarus 5.2 2012 [1]
Belgium 6.6 2012 [1]
Belize 6.6 2012 [1]
Benin 5.3 2012 [1]
Bhutan 4.7 2012 [1]
Bolivia 7.6 2012 [1]
Bosnia and Herzegovina n.a. n.a. [1]
Botswana 7.8 2012 [1]
Brazil 5.8 2012 [1]
...
```

# 世界各國教育支出比例排名 (二)

以上每一列資料依次為國名、**GDP** 百分比、統計年度、註解。資料中有些國家的名稱可能不只為一個字。若該國家沒有 **GDP** 百分比時則以 **n.a.** 替代。

現在我們要撰寫程式重新以 **GDP** 的百分比由大到小排列，沒有 **GDP** 百分比的國家排列在最後，同時將資料以橫線替代。此外如果 **GDP** 百分比一樣的國家，則按照國家名稱的字母順列排列。以下程式要輸出的內容：

```
1: 14.6 [2000] Marshall Islands
2: 13.0 [2012] Lesotho
3: 12.9 [2012] Cuba
4: 11.0 [2000] Kiribati
5: 10.1 [2012] Timor-Leste
6: 9.8 [2000] Palau
7: 8.7 [2012] Denmark
8: 8.6 [2012] Moldova
9: 8.4 [2012] Djibouti
10: 8.4 [2012] Namibia
11: 8.2 [2012] Ghana
12: 7.8 [2012] Botswana
13: 7.8 [2012] Iceland
```

```
...
177: 1.3 [2012] Zambia
178: 1.2 [2012] Central African Republic
179: 0.8 [2012] Myanmar
180: 0.7 [2000] Equatorial Guinea
181: --- [n.a.] Bosnia and Herzegovina
182: --- [n.a.] Grenada
183: --- [n.a.] Haiti
184: --- [n.a.] Macedonia
...
```

# 世界各國教育支出比例排名 (三)

```
nations , gdps , years = [] , [] , []

with open("edu.dat") as infile :

    for line in infile :

        if line.isspace() : continue

        # 星號將多字串的國家名稱儲存到串列
        *nation , gdp , year , continent = line.split()

        # 儲存國名與年度資料於串列
        nations.append( " ".join(nation) )
        years.append( year )

        # 沒有資料時, gdp 以 -1 替代
        if year == "n.a." :
            gdps.append( -1 )
        else :
            gdps.append( float(gdp) )
```

# 世界各國教育支出比例排名 (四)

```
i = 1

# 依 gdp 由大到小，國名有小到大排列
for nation , gdp , yr in sorted( zip(nations,gdps,years) ,
                                key=lambda p : ( -p[1] , p[0] ) ) :

    if gdp == -1 :
        line = "{:>3}:{:>5} [{:>4}] {:}" .format( i , "---" ,
                                                yr , nation )

    else :
        line = "{:>3}:{:>5.1F} [{:>4}] {:}" .format( i , gdp ,
                                                yr , nation )

    print( line )
    i += 1
```