

CSV 檔案讀取

Data Processing - CSV

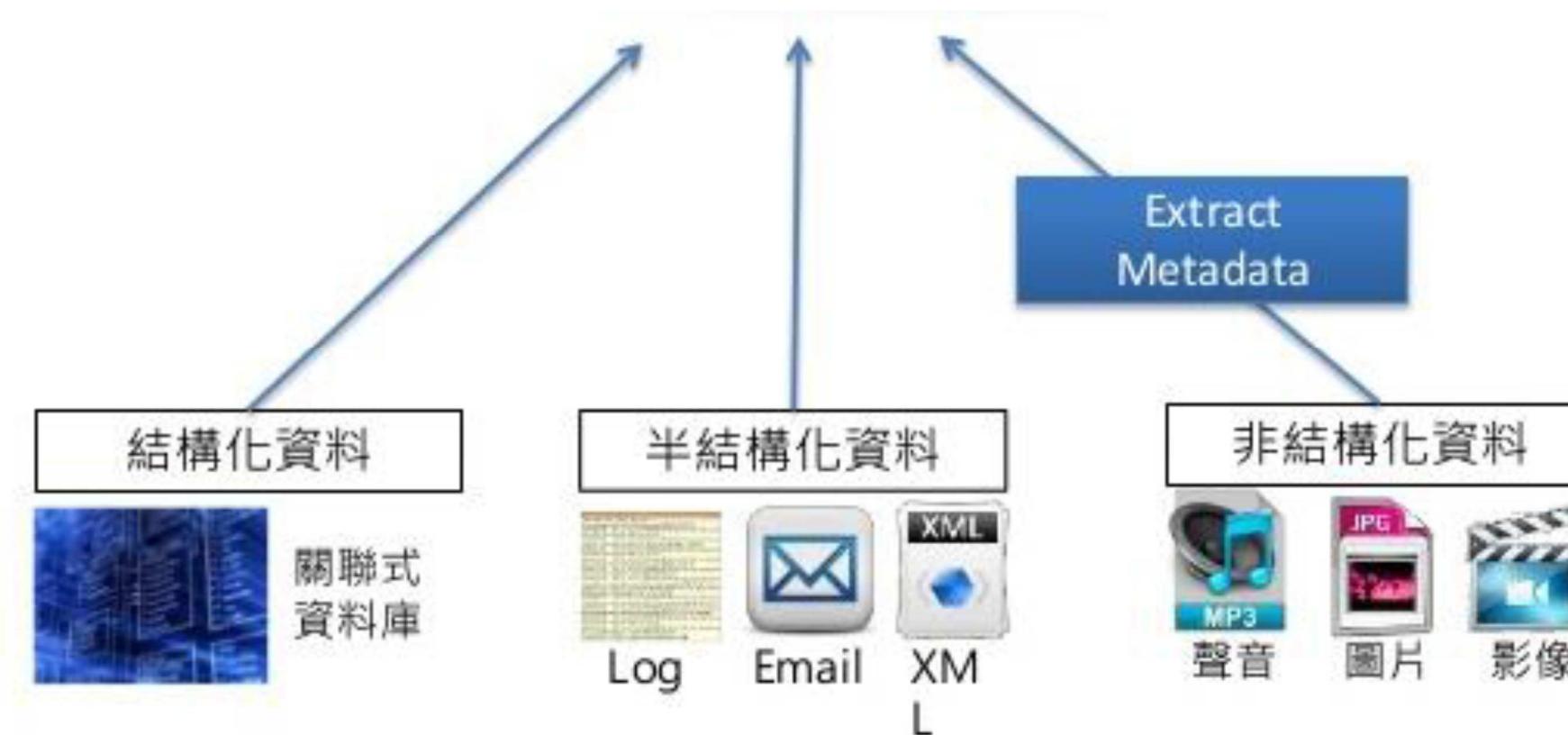


Python Data Analysis

資料蒐集

◆ 資料分成結構化資料、半結構化資料、非結構化資料三種。

- 結構化資料：CSV、Excel、資料庫
- 半結構化資料：JSON、XML
- 非結構化資料：沒有格式的文字、網頁資料



資料蒐集

◆ 資料分成結構化資料、半結構化資料、非結構化資料三種。

- 結構化資料：CSV、Excel、資料庫
- 半結構化資料：JSON、XML
- 非結構化資料：沒有格式的文字、網頁資料

CSV

A	B	C	D
1 ID	Gender	City	Monthly_I
2 ID000002C	Female	Delhi	20000
3 ID000004E	Male	Mumbai	35000
4 ID000007H	Male	Panchkula	22500
5 ID000008I	Male	Saharsa	35000
6 ID000009J	Male	Bengaluru	100000
7 ID000010K	Male	Bengaluru	45000
8 ID000011L	Female	Sindhudun	70000
9 ID000012M	Male	Bengaluru	20000
10 ID000013N	Male	Kochi	75000
11 ID000014O	Female	Mumbai	30000
12 ID000016C	Male	Mumbai	25000
13 ID000018S	Female	Surat	25000
14 ID000019T	Female	Pune	24000
15 ID000021V	Male	Bhubanes	27000
16 ID000022V	Female	Howrah	28000

JSON

```
1 "Employee": [  
    {  
        "id": "1",  
        "Name": "Ankit",  
        "Sal": "1000",  
    },  
    {  
        "id": "2",  
        "Name": "Faizy",  
    }  
]
```

XML

```
<?xml version="1.0"?>  
  
<contact-info>  
  
<name>Ankit</name>  
  
<company>Analytics Vidhya</company>  
  
<phone>+9187654321</phone>  
  
</contact-info>
```

資料蒐集

◆ 使用封閉規格的私有檔案格式問題資料：

- 資訊需要特定的軟體才能讀取
- 軟體售價昂貴，而且可能有期限的問題，過期將不再支援
- 可能會造成對第三方軟體或檔案格式授權持有者的依賴



資料蒐集

◆ Open Data Handbook 建議開放檔案格式

- CSV
- JSON
- XML
- 試算表
- 文書文件 (Word、ODF、OOXML、PDF 格式)
- 純文字檔 (txt)



☰ OPEN DATA HANDBOOK

GUIDE

檔案格式

Languages: de el en es fr he hr id is it ja ko lt lv my ne nl_BE pt_BR ro ru zh_CN
zh_TW

檔案格式綜覽

JSON

JSON 是一種簡單的檔案格式，任何程式語言都可輕鬆讀取。其簡易的特性代表它和其他格式 - 例如 XML 相較，可更容易地透過電腦來處理。

XML

XML 廣泛地使用於資料交換，因為其可以完好地保存資料中的結構，還有檔案建立的方式，並且能讓程式員將部份說明文件寫進同檔案中，而不會干擾資料的可讀性。

RDF

RDF，一種 W3C 所建議的格式，他呈現資料的方式讓其與其他格式相比，叫能結合多種不同的資料來源。RDF 資料可以 XML 和 JSON 儲存。RDF 建議使用 URLs 當作識別方式，提供一種方便的方式可直接相互連結於網頁上現存的 open data 草案。RDF 目前使用還不廣泛，但已逐漸開始隨著開放政府草案有變多的趨勢，像是英國和西班牙政府的「連結開放資料計劃」。還有全球資訊網的發明者，Tim Berners-Lee，最近也提出了一個五星等級方案，內容包含將連結的 RDF 資料成為開放資料草案追求的目標。





CSV 檔案

- ◆ CSV 全名為 Comma-Separated Values，是指以逗點作為區隔的文字格式資料。
- ◆ CSV 簡潔易讀，且格式簡單，多半是肉眼即可判讀，因此被廣泛的接受作為簡單格式的應用。
- ◆ CSV 是試算表和資料庫之間最常用的資料格式，我們可以先將所搜集的各式檔案轉成CSV，之後可以使用Python讀取所有的csv檔案，再擷取需要的資料做後續的分析；或是利用csv檔案，將它當作不同資料庫間的或資料庫與試算表之間的橋段。
- ◆ CSV 模組常用函式。

函式	功能說明
<code>csv.reader (csvfile, **fmtparams)</code>	從 <code>csvfile</code> 讀取的每行都作為字串串列回傳給一個可迭代的閱讀器物件
<code>csv.writer (csvfile, **fmtparams)</code>	傳回一個寫入器物件， <code>dialect</code> 參數用法同上
<code>writerow(row)</code>	將 <code>row</code> 參數傳給寫入器物件，寫入CSV檔案

讀取CSV 檔案

◆ Python 內建csv模組，只要匯入模組後，就可以讀取CSV檔案，方便之後程式的操作。其流程如下：

1. 汇入csv模組：

import csv

2. 使用open()函式開啟CSV檔案：

with open (檔案名稱, encoding='utf8') as xxx (可自行命名)

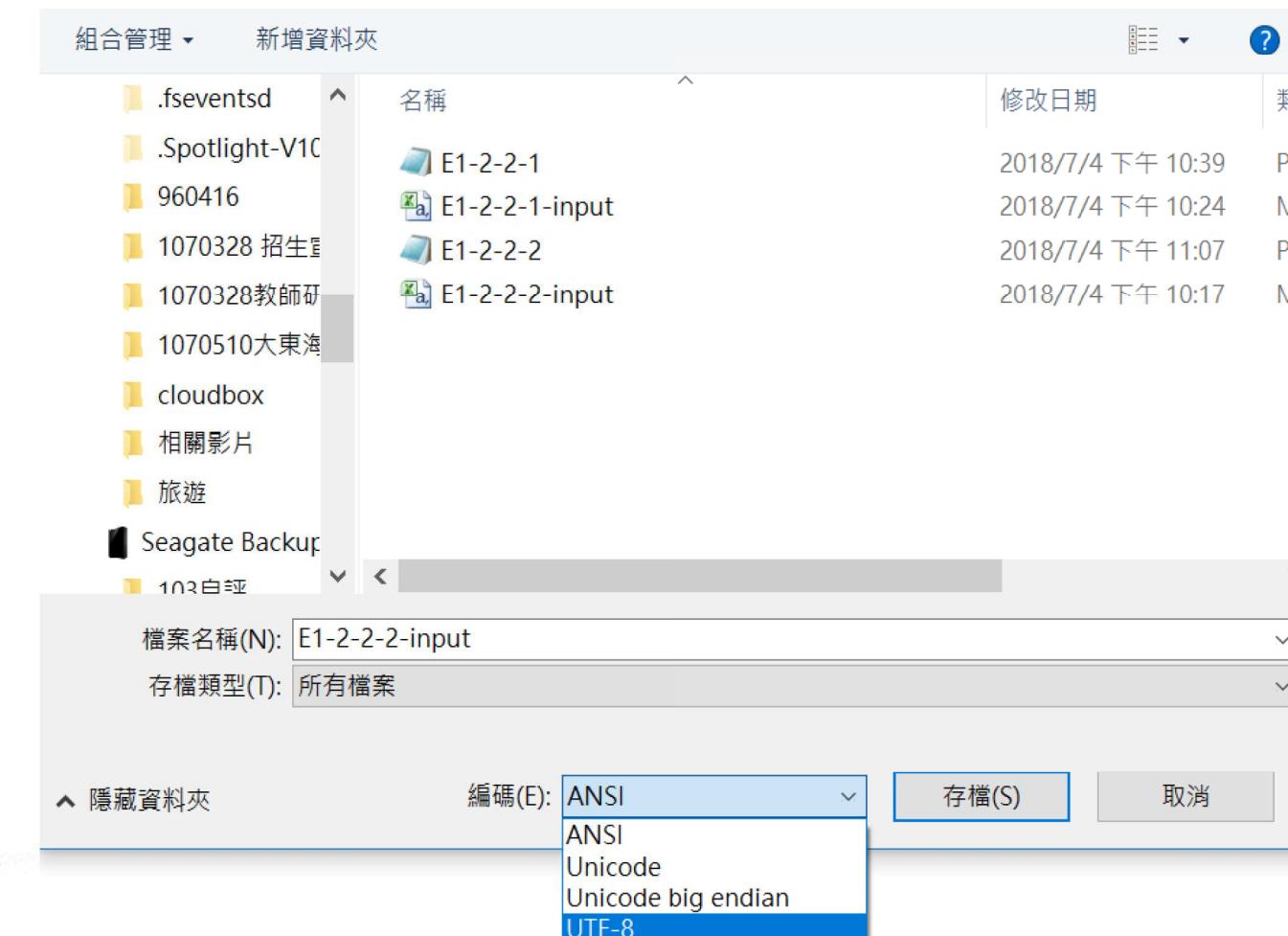
3. 呼叫csv模組的reader()函式，建立讀取器物件(csvReader)，之後就可進行CSV檔案的讀取：

csvReader = csv.reader(xxx)

讀取CSV 檔案常見錯誤(訊息)

◆ 編碼格式錯誤：

- 有些csv資料，其編碼格式並非以UTF-8編碼，而是用ANSI編碼，在Python 預設的UTF-8編碼下就會產生以下錯誤：
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xba in position 0: invalid start byte
- 解決方式：使用記事本開啟檔案，再另存新檔時，設定編碼方式為UTF-8，如下圖：



讀取CSV 檔案_1

◆ 建立Reader物件，再轉成串列後直接輸出。

1. csv模組中的reader()函式可建立閱讀器(Reader)物件。
2. 之後使用list()函式將Reader物件轉換成串列(list)，即可將串列輸出。

```
1 import csv
2
3 fn = 'csvReport.csv'
4 # 開啟csv檔案
5 with open(fn,encoding = 'utf8') as csvFile:
6     # 讀取檔案物件建立Reader物件
7     csvReader = csv.reader(csvFile)
8     # 將資料轉成串列
9     listReport = list(csvReader)
10    print(listReport)
```

```
[[ 'Name', 'Year', 'Product', 'Price', 'Quantity', 'Revenue', 'Location'], ['Diana', '2015', 'Black Tea', '10', '600', '6000', 'New York'], ['Diana', '2015', 'Green Tea', '7', '660', '4620', 'New York'], ['Diana', '2016', 'Black Tea', '10', '750', '7500', 'New York'], ['Diana', '2016', 'Green Tea', '7', '900', '6300', 'New York'], ['Julia', '2015', 'Black Tea', '10', '1200', '12000', 'New York'], ['Julia', '2016', 'Black Tea', '10', '1260', '12600', 'New York'], ['Steve', '2015', 'Black Tea', '10', '1170', '11700', 'Chicago'], ['Steve', '2015', 'Green Tea', '7', '1260', '8820', 'Chicago'], ['Steve', '2016', 'Black Tea', '10', '1350', '13500', 'Chicago']]
```

讀取CSV 檔案_2

- ◆ 建立Reader物件，使用迴圈逐筆輸出Reader物件內容。

```
1 import csv  
2  
3 fn = 'csvReport.csv'  
4 with open(fn,encoding = 'utf8') as csvFile: # 開啟csv檔案  
5     csvReader = csv.reader(csvFile)           # 讀取檔案建立Reader物件csvReader  
6     for row in csvReader:                   # 用迴圈列出csvReader物件內容  
7         print("Row {:d} = ".format(csvReader.line_num), row)
```

```
Row 1 = ['Name', 'Year', 'Product', 'Price', 'Quantity', 'Revenue', 'Location']  
Row 2 = ['Diana', '2015', 'Black Tea', '10', '600', '6000', 'New York']  
Row 3 = ['Diana', '2015', 'Green Tea', '7', '660', '4620', 'New York']  
Row 4 = ['Diana', '2016', 'Black Tea', '10', '750', '7500', 'New York']  
Row 5 = ['Diana', '2016', 'Green Tea', '7', '900', '6300', 'New York']  
Row 6 = ['Julia', '2015', 'Black Tea', '10', '1200', '12000', 'New York']  
Row 7 = ['Julia', '2016', 'Black Tea', '10', '1260', '12600', 'New York']  
Row 8 = ['Steve', '2015', 'Black Tea', '10', '1170', '11700', 'Chicago']  
Row 9 = ['Steve', '2015', 'Green Tea', '7', '1260', '8820', 'Chicago']  
Row 10 = ['Steve', '2016', 'Black Tea', '10', '1350', '13500', 'Chicago']  
Row 11 = ['Steve', '2016', 'Green Tea', '7', '1440', '10080', 'Chicago']
```

讀取CSV 檔案_3

◆ 使用索引列出特定串列內容

```
1 import csv
2
3 fn = 'csvReport.csv'
4 with open(fn,encoding = 'utf8') as csvFile: # 開啟csv檔案
5     csvReader = csv.reader(csvFile)           # 讀取檔案建立Reader物件
6     listReport = list(csvReader)            # 將資料轉成串列
7
8 print(listReport[0][1], listReport[0][2])
9 print(listReport[1][1], listReport[1][2])
10 print(listReport[3][1], listReport[3][2])
```

Year	Product
2015	Black Tea
2016	Black Tea

讀取CSV 檔案

◆ 使用DictReader()讀取：

- 傳回值是排序的字典(OrderedDict)。
- 可以用欄位名稱當索引方式取得資料。

```
import csv
fn = 'csvReport.csv'
with open(fn,encoding = 'utf8') as csvFile:
    csvDictReader = csv.DictReader(csvFile)
    for row in csvDictReader:
        print(row)
```

```
OrderedDict([('Name', 'Diana'), ('Year', '2015'), ('Product', 'Black Tea'), ('Price', '10'),
('Quantity', '600'), ('Revenue', '6000'), ('Location', 'New York')])
OrderedDict([('Name', 'Diana'), ('Year', '2015'), ('Product', 'Green Tea'), ('Price', '7'),
('Quantity', '660'), ('Revenue', '4620'), ('Location', 'New York')])
OrderedDict([('Name', 'Diana'), ('Year', '2016'), ('Product', 'Black Tea'), ('Price', '10'),
('Quantity', '750'), ('Revenue', '7500'), ('Location', 'New York')])
OrderedDict([('Name', 'Diana'), ('Year', '2016'), ('Product', 'Green Tea'), ('Price', '7'),
('Quantity', '900'), ('Revenue', '6300'), ('Location', 'New York')])
OrderedDict([('Name', 'Julia'), ('Year', '2015'), ('Product', 'Black Tea'), ('Price', '10'),
('Quantity', '1200'), ('Revenue', '12000'), ('Location', 'New York')])
OrderedDict([('Name', 'Julia'), ('Year', '2016'), ('Product', 'Black Tea'), ('Price', '10'),
('Quantity', '1260'), ('Revenue', '12600'), ('Location', 'New York')])
```

讀取CSV 檔案

- ◆ 抓取Dict中特定欄位的資料。

```
import csv

fn = 'csvReport.csv'
with open(fn,encoding = 'utf8') as csvFile: # 開啟csv檔案
    csvDictReader = csv.DictReader(csvFile) # 讀檔案建立DictReader物件
    for row in csvDictReader:          # 使用迴圈列出字典內容
        print(row['Product'], row['Price'])
```

```
Black Tea 10
Green Tea 7
Black Tea 10
Green Tea 7
Black Tea 10
Black Tea 10
Black Tea 10
Green Tea 7
Black Tea 10
Green Tea 7
```

寫入CSV 檔案

- ◆ 呼叫csv模組的writer方法，建立寫入器物件(outWriter)，就可進行csv檔的寫入，語法如下：

with open('檔案名稱', 'w', newline=' ', encoding='utf8') as csvFile :

```
outWriter = csv.writer(csvFile)
```

- ◆ 參數newline=' '，可以避免輸出時每個行之間多空一行。

寫入CSV 檔案_1

◆ writerow() 函式寫入串列資料

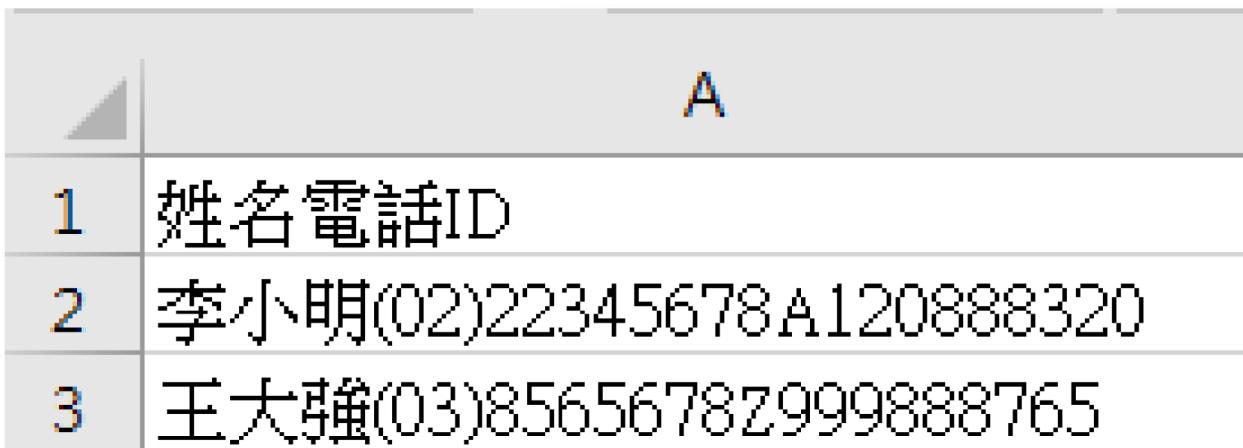
```
import csv
fn = 'csvoutput.csv'
with open(fn, 'w', newline='', encoding ='utf-8') as csvFile: # 開啟csv檔案
    csvWriter = csv.writer(csvFile) # 建立writer物件
    csvWriter.writerow(['姓名', '電話', 'ID', '費用','是否前往'])
    csvWriter.writerow(['李小明', '(02)22345678', 'A120888320',100,True])
    csvWriter.writerow(['王大強', '(03)8565678', 'Z999888765',200,False])
```

姓名	電話	ID	費用	是否前往
李小明	(02)22345678	A120888320	100	TRUE
王大強	(03)8565678	Z999888765	200	FALSE

寫入CSV 檔案_1

- ◆ 當我們將資料寫入CSV檔案時，預設是各欄間是以逗號做區隔。
- ◆ 我們可以用 `delimiter` 分隔符號更改各欄間的逗號設定 (用在`writer()`方法)。
- ◆ 當用 '`\t`' 字元取代逗號後，若用Excel開啟會將每行資料放在一起。建議用記事本開啟這類的csv檔案。

```
1 import csv
2
3 fn = 'csvoutput2.csv'
4 with open(fn, 'w', newline = '') as csvFile:          # 開啟csv檔案
5     csvWriter = csv.writer(csvFile, delimiter='\t')        # 建立Writer物件
6     csvWriter.writerow(['姓名', '電話', 'ID'])
7     csvWriter.writerow(['李小明', '(02)22345678', 'A120888320'])
8     csvWriter.writerow(['王大強', '(03)8565678', 'Z999888765'])
```



A screenshot of a Microsoft Notepad window titled "csvoutput.csv - 記事本". The window displays a table with three rows of data. The first row contains column headers: "姓名", "電話", and "ID". The second row contains the data for "李小明": "(02)22345678" and "A120888320". The third row contains the data for "王大強": "(03)8565678" and "Z999888765".

	A	csvoutput.csv - 記事本		
1	姓名 電話 ID	檔案(F)	編輯(E)	格式(O)
2	李小明 (02)22345678 A120888320	李小明	(02)22345678	A120888320
3	王大強 (03)8565678 Z999888765	王大強	(03)8565678	Z999888765

寫入CSV 檔案_2

- ◆ DictWriter() 可以寫入字典資料，語法如下：

dictWriter =csv.DictWriter(csvFile, fieldnames=fields)

- ◆ dictWriter是字典的Writer 物件，在上述指令前我們需要先設定 fields串列，這個串列將包含未來字典內容的鍵(key)。

```
import csv
fn = 'csvoutput.csv'
with open(fn, 'w', newline = '') as csvFile:          # 開啟csv檔案
    fields = ['姓名', '電話', 'ID']
    dictWriter = csv.DictWriter(csvFile, fieldnames=fields) # 建立Writer物件
    dictWriter.writeheader()                                # 寫入標題
    dictWriter.writerow({'姓名': '李小明', '電話': '(02)22345678', 'ID': 'A120888320'})
    dictWriter.writerow({'姓名': '王大強', '電話': '(03)8565678', 'ID': 'Z999888765'})
```

	A	B	C
1	姓名	電話	ID
2	李小明	(02)22345678	A120888320
3	王大強	(03)8565678	Z999888765

寫入CSV 檔案_3

- ◆ 改寫上述範例，將欲寫入CSV檔案的資料改成串列資料，此串列資料的元素是字典。

```
import csv

# 定義串列，元素是字典
dictList = [{‘姓名’:‘李小明’, ‘電話’:‘(02)22345678’, ‘ID’:‘A120888320’},
            {‘姓名’:‘王大強’, ‘電話’:‘(03)8565678’, ‘ID’:‘Z999888765’}]

# 開啟csv檔案
fn = ‘csvoutput.csv’
with open(fn, ‘w’, newline = ‘’) as csvFile:
    fields = [‘姓名’, ‘電話’, ‘ID’]
    dictWriter = csv.DictWriter(csvFile, fieldnames=fields) # 建立Writer物件
    dictWriter.writeheader() # 寫入標題
    for row in dictList:
        dictWriter.writerow(row)
```

	A	B	C
1	姓名	電話	ID
2	李小明	(02)22345678	A120888320
3	王大強	(03)8565678	Z999888765

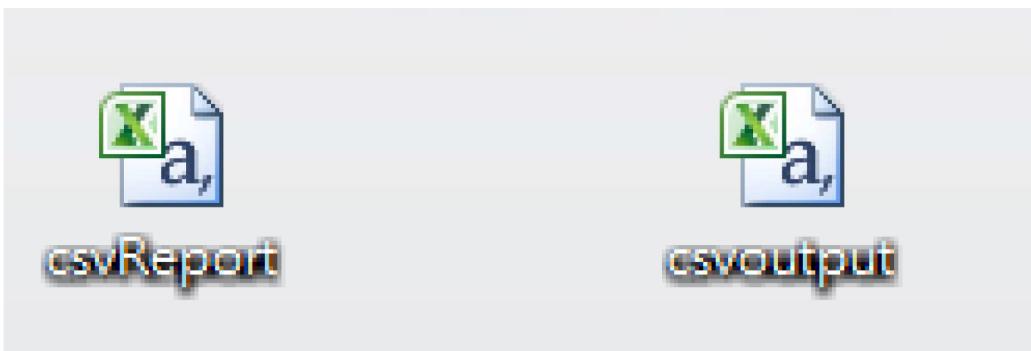
讀寫CSV 檔案

- ◆ 先讀CSV檔案，再將檔案寫入另一個檔案，達成拷貝的目的。

```
import csv

infn = 'csvReport.csv'                                # 來源檔案
outfn = 'csvoutput.csv'                               # 目的檔案
with open(infn,encoding = 'utf8') as csvRFile:       # 開啟csv檔案供讀取
    csvReader = csv.reader(csvRFile)                  # 讀取檔案建立Reader物件
    listReport = list(csvReader)                      # 將資料轉成串列

with open(outfn, 'w', newline = '',encoding = 'utf8') as csvOFile: # 開啟csv檔案供寫入
    csvWriter = csv.writer(csvOFile)                 # 建立Writer物件
    for row in listReport:                           # 將串列寫入
        csvWriter.writerow(row)
```



Q & A