

勞動部產業新尖兵計畫

人工智慧金融應用與實務培訓班



課程模組： AI 金融科技課程 - 網路爬蟲技術

3. Python 與 CSV

葉建華 (Yeh, Jian-hua)

tdi.jhyeh@tdi.edu.tw
au4290@gmail.com

講次內容

- 認識 CSV 格式
- Python 的 CSV 模組：讀取 CSV
- Python 的 CSV 模組：寫出 CSV
- Python 與 Excel 試算表

CSV 格式

- 以逗號分隔取值的格式
- 所有 CSV 檔都是純文字檔
- 通常用於大量資料情況下不同的應用程式之間交換資料

CSV 格式

- 以逗號分隔取值的格式

"110年06月 發行量加權股價指數歷史資料"

- 所有

"日期", "開盤指數", "最高指數", "最低指數", "收盤指數",

- 通過

換

"110/06/01", "17,098.49", "17,184.30", "17,060.45", "17,162.38",

"110/06/02", "17,192.91", "17,274.66", "17,056.42", "17,165.04",

"110/06/03", "17,201.30", "17,311.28", "17,198.89", "17,246.16",

"110/06/04", "17,225.06", "17,225.06", "17,084.49", "17,147.41",

"110/06/07", "17,176.09", "17,189.23", "16,775.85", "17,083.91",

"110/06/08", "17,094.34", "17,181.52", "17,056.53", "17,076.21",

"110/06/09", "17,082.52", "17,082.52", "16,907.43", "16,966.22",

"110/06/10", "17,024.54", "17,159.22", "16,978.01", "17,159.22",

"110/06/11", "17,200.45", "17,278.55", "17,193.61", "17,213.52",

"110/06/15", "17,279.70", "17,371.29", "17,279.70", "17,371.29",

月交

講次內容

- 認識 CSV 格式
- Python 的 CSV 模組：讀取 CSV
- Python 的 CSV 模組：寫出 CSV
- Python 與 Excel 試算表

Python 讀取 CSV : List

- import csv 模組
- 使用 reader() 函數建立 reader 物件
- 轉換成 Python list

Python 讀取 CSV : List

- import csv 模組
- 使用 reader() 函數建立 reader 物件

• 轉換

```
import csv
infile = 'twii.202106.csv'
# 開啟csv檔案
with open(infile, "r", encoding="big5") as csvf:
    csvReader = csv.reader(csvf)      # 讀檔案建立Reader物件
    list1 = list(csvReader)          # 將資料轉成串列

for rec in list1:
    print(rec)
```


Python 讀取 CSV：List

- import
- 使用
- 轉換

```
import csv
```

```
infilename = 'twii.202106.csv'
```

```
# 開啟csv檔案
```

```
with open(infilename, "r", encoding="big5") as csvf:|
```

```
    csvReader = csv.reader(csvf)          # 讀檔案建立Reader物件
```

```
    list1 = list(csvReader)              # 將資料轉成串列
```

```
for
```

```
    ['110年06月 發行量加權股價指數歷史資料']
```

```
    ['日期', '開盤指數', '最高指數', '最低指數', '收盤指數', '']
```

```
    ['110/06/01', '17,098.49', '17,184.30', '17,060.45', '17,162.38', '']
```

```
    ['110/06/02', '17,192.91', '17,274.66', '17,056.42', '17,165.04', '']
```

```
    ['110/06/03', '17,201.30', '17,311.28', '17,198.89', '17,246.16', '']
```

```
    ['110/06/04', '17,225.06', '17,225.06', '17,084.49', '17,147.41', '']
```

```
    ['110/06/07', '17,176.09', '17,189.23', '16,775.85', '17,083.91', '']
```

```
    ['110/06/08', '17,094.34', '17,181.52', '17,056.53', '17,076.21', '']
```

```
    ['110/06/09', '17,082.52', '17,082.52', '16,907.43', '16,966.22', '']
```

```
    ['110/06/10', '17,024.54', '17,159.22', '16,978.01', '17,159.22', '']
```

```
    ['110/06/11', '17,200.45', '17,278.55', '17,193.61', '17,213.52', '']
```

```
    ['110/06/15', '17,279.70', '17,371.29', '17,279.70', '17,371.29', '']
```


Python 讀取 CSV : List

- import csv 模組
- 使用 reader() 函數建立 reader 物件
- 轉換成 Python list
- 既然是 list , 就可以逐個元素操作!

Python 讀取 CSV : List

- import csv 模組
- 使用 reader() 函數建立 reader 物件

• 轉換成

• 既然是

```
import csv

infilename = 'twii.202106.csv'
# 開啟csv檔案
with open(infilename, "r", encoding="big5") as csvf:
    csvReader = csv.reader(csvf)      # 讀檔案建立Reader物件
    list1 = list(csvReader)          # 將資料轉成串列

    for rec in list1:
        for field in rec:
            print(field+' ', end='')
        print('')
```

Python 讀取 CSV：List

- import csv 模組
- 使用 reader() 函數建立 reader 物件

• 轉換成

• 既然是

```
import csv
```

```
infile = 'twii.2021
```

```
# 開啟csv檔案
```

```
with open(infile, "
```

```
    csvReader = csv.
```

```
    list1 = list(csv
```

```
for rec in list1:
```

```
    for field in rec
```

```
        print(field)
```

```
    print('')
```

110年06月 發行量加權股價指數歷史資料

日期 開盤指數 最高指數 最低指數 收盤指數

110/06/01 17,098.49 17,184.30 17,060.45 17,162.38

110/06/02 17,192.91 17,274.66 17,056.42 17,165.04

110/06/03 17,201.30 17,311.28 17,198.89 17,246.16

110/06/04 17,225.06 17,225.06 17,084.49 17,147.41

110/06/07 17,176.09 17,189.23 16,775.85 17,083.91

110/06/08 17,094.34 17,181.52 17,056.53 17,076.21

110/06/09 17,082.52 17,082.52 16,907.43 16,966.22

110/06/10 17,024.54 17,159.22 16,978.01 17,159.22

110/06/11 17,200.45 17,278.55 17,193.61 17,213.52

110/06/15 17,279.70 17,371.29 17,279.70 17,371.29

辭典結構的 CSV 內容

- 鍵值：日期、開高低收指數
- 資料值：第二列起相對應直欄資料

"日期", "開高低收指數"

"110/06/01", "17098.49,17184.30,17060.45,17162.38"

"110/06/02", "17192.91,17274.66,17056.42,17165.04"

"110/06/03", "17201.30,17311.28,17198.89,17246.16"

"110/06/04", "17225.06,17225.06,17084.49,17147.41"

"110/06/07", "17176.09,17189.23,16775.85,17083.91"

"110/06/08", "17094.34,17181.52,17056.53,17076.21"

"110/06/09", "17082.52,17082.52,16907.43,16966.22"

"110/06/10", "17024.54,17159.22,16978.01,17159.22"

"110/06/11", "17200.45,17278.55,17193.61,17213.52"

"110/06/15", "17279.70,17371.29,17279.70,17371.29"

Python 讀取 CSV : Dict

- import csv
- 使用 DictReader() 建立 reader 物件
- 轉換成 Python dict

Python 讀取 CSV : Dict

- import csv
- 使用 DictReader() 建立 reader 物件
- 轉換成 Python dict

```
import csv

infilename = 'twii.202106-2.csv'
# 開啟csv檔案
with open(infilename, "r", encoding="big5") as csvf:
    csvReader = csv.DictReader(csvf) # 讀檔案建立Reader物件
    for dic1 in csvReader:
        print(dic1)
```


Python 讀取 CSV : Dict

- import csv

- 使用 DictReader

- 轉換成 Dict

```
{ '日期': '110/06/01', '開高低收指數': '17098.49,17184.30,17060.45,17162.38' }  
{ '日期': '110/06/02', '開高低收指數': '17192.91,17274.66,17056.42,17165.04' }  
{ '日期': '110/06/03', '開高低收指數': '17201.30,17311.28,17198.89,17246.16' }  
{ '日期': '110/06/04', '開高低收指數': '17225.06,17225.06,17084.49,17147.41' }  
{ '日期': '110/06/07', '開高低收指數': '17176.09,17189.23,16775.85,17083.91' }  
{ '日期': '110/06/08', '開高低收指數': '17094.34,17181.52,17056.53,17076.21' }  
{ '日期': '110/06/09', '開高低收指數': '17082.52,17082.52,16907.43,16966.22' }  
in { '日期': '110/06/10', '開高低收指數': '17024.54,17159.22,16978.01,17159.22' }  
  { '日期': '110/06/11', '開高低收指數': '17200.45,17278.55,17193.61,17213.52' }  
in { '日期': '110/06/15', '開高低收指數': '17279.70,17371.29,17279.70,17371.29' }  
# 開啟csv檔案  
with open(infile, "r", encoding="big5") as csvf:  
    csvReader = csv.DictReader(csvf) # 讀檔案建立Reader物件  
    for dic1 in csvReader:  
        print(dic1)
```


Python 讀取 CSV : Dict

- import csv
- 使用 DictReader() 建立 reader 物件
- 轉換成 Python dict
- dict 可以操作 key-value 的存取!

Python 讀取 CSV : Dict

- import csv

- 使用 DictReader

- 轉換成 dict

- dict 可以操作 key-value 的存取!

```
import csv

infile = 'twii.202106-2.csv'
# 開啟csv檔案
with open(infile, "r", encoding="big5") as csvf:
    csvReader = csv.DictReader(csvf) # 讀檔案建立Reader物件
    for dic1 in csvReader:
        print(dic1['日期'])
```

Python 讀取 CSV : Dict

- import csv

- 使用 [

- 轉換成

- dict 可以操作 key-value 的存取!

```
import csv
```

```
infilename = 'twii.202106-2.csv'
```

```
# 開啟csv檔案
```

```
with open(infilename, "r", encoding="big5") as csvf:
```

```
    csvReader = csv.DictReader(csvf) # 讀檔案建立Reader
```

```
    for dic1 in csvReader:
```

```
        print(dic1['日期'])
```

110/06/01

110/06/02

110/06/03

110/06/04

110/06/07

110/06/08

110/06/09

110/06/10

110/06/11

110/06/15

講次內容

- 認識 CSV 格式
- Python 的 CSV 模組：讀取 CSV
- Python 的 CSV 模組：寫出 CSV
- Python 與 Excel 試算表

Python 寫出 CSV : List

- import csv
- 使用 writer() 建立 csv 的 writer 物件
- writer 物件有 writerow() 函數來對應 list 資料

Python 寫出 CSV : List

- import csv
- 使用 writer() 建立 csv 的 writer 物件
- writer 物件有 writerow() 函數來對應 list 資料

```
import csv

outfname = '2330.csv'
with open(outfname, 'w', newline='') as outf:      # 開啟csv檔案
    csvWriter = csv.writer(outf)                  # 建立Writer物件
    csvWriter.writerow(['Date', 'Open', 'High', 'Low', 'Close'])
    csvWriter.writerow(['2021-06-10', 591, 599, 587, 599])
    csvWriter.writerow(['2021-06-11', 602, 603, 600, 602])
    csvWriter.writerow(['2021-06-15', 607, 609, 606, 609])
```


Python 寫出 CSV : List

- import csv
- 使用 writer() 建立 csv 的 writer 物件
- writer 物件有 writerow() 函數

Date	Open	High	Low	Close
2021-06-10	591	599	587	599
2021-06-11	602	603	600	602
2021-06-15	607	609	606	609

```
import csv

outfile = '2330.csv'
with open(outfile, 'w', newline='') as outf:
    csvWriter = csv.writer(outf)
    csvWriter.writerow(['Date', 'Open', 'High', 'Low', 'Close'])
    csvWriter.writerow(['2021-06-10', 591, 599, 587, 599])
    csvWriter.writerow(['2021-06-11', 602, 603, 600, 602])
    csvWriter.writerow(['2021-06-15', 607, 609, 606, 609])
```


Python 寫出 CSV : List

- 請注意開啟寫入檔案的設定
 - `open(outfname, 'w', newline="")`
 - 模式 'w' 沒有問題
 - 可是 `newline=""` ? ?
 - 因為 `writerow()` 會自行輸出換行字元!!

Python 寫出 CSV：List

- 可以使用 delimiter 參數來指定欄位分隔方式
 - 就是設定分隔符號

Python 寫出 CSV : List

- 可以使用 delimiter 參數來指定欄位分隔方式
 - 就是設定分隔符號

delimiter 參數設定 TAB 字元 '\t'

```
import csv

outfname = '2330-2.csv'
with open(outfname, 'w', newline='') as outf:      # 開啟csv檔案
    csvWriter = csv.writer(outf, delimiter='\t')    # 建立Writer物件
    csvWriter.writerow(['Date', 'Open', 'High', 'Low', 'Close'])
    csvWriter.writerow(['2021-06-10', 591, 599, 587, 599])
    csvWriter.writerow(['2021-06-11', 602, 603, 600, 602])
    csvWriter.writerow(['2021-06-15', 607, 609, 606, 609])
```

Python 寫出 CSV : List

- 可以使用 delimiter 參數來指定欄位分隔方式
 - 就是設定分隔符號

```
import csv
```

```
outfname = '2330-2.csv'
```

```
with open(outfname, 'w', newline='') as outf: # 開啟csv檔案
```

```
    csvWriter = csv.writer(outf, delimiter='\t') # 建立Writer物件
```

```
    csvWriter.writerow(['Date', 'Open', 'High', 'Low', 'Close'])
```

```
    csvWriter.writerow(['Date', 'Open', 'High', 'Low', 'Close'])
```

```
    csvWriter.writerow(['2021-06-10', '591', '599', '587', '599'])
```

```
    csvWriter.writerow(['2021-06-11', '602', '603', '600', '602'])
```

```
    csvWriter.writerow(['2021-06-15', '607', '609', '606', '609'])
```

TAB 字元 '\t' 有「定位」的功能!

Python 寫出 CSV : Dict

- import csv
- 使用 DictWriter() 建立對應 dict 的 writer 物件
- 寫出 Python dict

Python 寫出 CSV : Dict

- import csv
- 使用 DictWriter() 建立對應 dict 的 writer 物件
- 寫出 Python dict

```
import csv

outfname = '2330-2.csv'
with open(outfname, 'w', newline='') as outf:           # 開啟csv檔案
    keys = ['Date', 'Open', 'Close']
    dictWriter = csv.DictWriter(outf, fieldnames=keys) # 建立Writer物件
    dictWriter.writeheader()                          # 先把keys寫入作為標題列
    dictWriter.writerow({'Date': '2021-06-10', 'Open': 591, 'Close': 599})
    dictWriter.writerow({'Date': '2021-06-11', 'Open': 602, 'Close': 602})
    dictWriter.writerow({'Date': '2021-06-15', 'Open': 607, 'Close': 609})
```


Python 寫出 CSV : Dict

- import csv
- 使用 DictWriter() 建立對應 dict 的 writer 物件
- 寫出 Python dict

```
import csv

outfname = '2330-2.csv'
with open(outfname, 'w', newline='') as outf:
    keys = ['Date', 'Open', 'Close']
    dictWriter = csv.DictWriter(outf, fieldnames=keys) # 建立Writer物件
    dictWriter.writeheader() # 先把keys寫入作為標題列
    dictWriter.writerow({'Date': '2021-06-10', 'Open': 591, 'Close': 599})
    dictWriter.writerow({'Date': '2021-06-11', 'Open': 602, 'Close': 602})
    dictWriter.writerow({'Date': '2021-06-15', 'Open': 607, 'Close': 609})
```

Date,Open,Close
2021-06-10,591,599
2021-06-11,602,602
2021-06-15,607,609

開啟csv檔案

建立Writer物件

先把keys寫入作為標題列

Python 寫出 CSV : Dict

- 常常你的多筆 dict 資料是包在 list 裡的 ...
 - 所以需要使用迴圈寫出 dict 結構的資料
 - 關鍵： list iterator, dict csv writer

Python 寫出 CSV : Dict

- 常常你的多筆 dict 資料是包在 list 裡的 ...
 - 所以需要使用迴圈寫出 dict 結構的資料

```
import csv

listData = [{'Date': '2021-06-10', 'Open': 591, 'Close': 599},
            {'Date': '2021-06-11', 'Open': 602, 'Close': 602},
            {'Date': '2021-06-15', 'Open': 607, 'Close': 609}]

outfname = '2330-2.csv'
with open(outfname, 'w', newline='') as outf:           # 開啟csv檔案
    keys = ['Date', 'Open', 'Close']
    dictWriter = csv.DictWriter(outf, fieldnames=keys)   # 建立Writer物件
    dictWriter.writeheader()                             # 先把keys寫入作為標題列
    for dictRec in listData:
        dictWriter.writerow(dictRec)
```

Python 寫出 CSV : Dict

- 常常你的多筆 dict 資料是包在 list 裡的 ...
 - 所以需要使用迴圈寫出 dict 結構的資料

```
import csv
```

```
listData = [{'Date': '2021-06-10', 'Open': 591, 'Close': 599},  
             {'Date': '2021-06-11', 'Open': 602, 'Close': 602},  
             {'Date': '2021-06-15', 'Open': 607, 'Close': 609}]
```

```
outfname = '2330-2.csv'
```

```
with open(outfname, 'w', newline='') as outf:
```

```
    keys = ['Date', 'Open', 'Close']
```

```
    dictWriter = csv.DictWriter(outf, fieldnames=keys)
```

```
    dictWriter.writeheader()
```

```
    for dictRec in listData:
```

```
        dictWriter.writerow(dictRec)
```

```
Date,Open,Close
```

```
2021-06-10,591,599
```

```
2021-06-11,602,602
```

```
2021-06-15,607,609
```

```
# 開啟csv檔案
```

```
# 建立Writer物件
```

```
# 先把keys寫入作為標題列
```

講次內容

- 認識 CSV 格式
- Python 的 CSV 模組：讀取 CSV
- Python 的 CSV 模組：寫出 CSV
- Python 與 Excel 試算表

淺談 Excel...

- Excel 與 CSV 格式的比較
 - 同樣是表格式的資料結構
 - Excel 對欄位資料的屬性與操作比較豐富
 - Python 的 Excel 處理模組：xlrd, xlwt

安裝 Python Excel 模組

- 第三方開發套件，所以理論上應該要安裝
- `pip install xlrd xlwt`
 - `xlrd`：Excel 讀取模組
 - `xlwt`：Excel 寫出模組
- 就這樣而已！
 - 其實 Anaconda 已經幫你裝好了

Python 讀取 Excel 檔案

- 使用 xlrd
 - 存取層次：試算表 -> 工作表 -> 資料列
 - 使用 open_workbook() 取得試算表物件
 - 試算表物件使用 sheets() 取得所有工作表
 - 工作表物件使用 row_values() 取得資料列
 - 以 list 物件表示

Python 讀取 Excel 檔案

- 使用 xlrd

- 存取層次：試算表 -> 工作表 -> 資料列
- 使用 open_workbook() 取得試算表物件

- 試算表

- 工作表

- 以 list 方式

```
import xlrd
infilename = '2330.xls'
wb2330 = xlrd.open_workbook(infilename) # 開啟Excel檔案
sheet0 = wb2330.sheets()[0] # 取出第一個工作表
rows = sheet0.nrows # 工作表的資料列數
for i in range(rows):
    print(sheet0.row_values(i)) # 取出每一列資料
```

Python 讀取 Excel 檔案

- 使用 xlrd

- 存取層次：試算表 -> 工作表 -> 資料列
- 使用 open_workbook() 取得試算表物件

- 試算表

- 工作表

- 以 list 方式

```
import xlrd
```

```
infilename = '2330.xls'
```

```
wb2330 = xlrd.open_workbook(infilename) # 開啟Excel檔案
```

```
sheet0 = wb2330.sheets()[0] # 取出第一個工作表
```

```
rows = sheet0.nrows
```

```
for i in range(rows):
```

```
    print(sheet0.row_values(i))
```

```
['Date', 'Open', 'High', 'Low', 'Close']  
['2021-06-10', 591.0, 599.0, 587.0, 599.0]  
['2021-06-11', 602.0, 603.0, 600.0, 602.0]  
['2021-06-15', 607.0, 609.0, 606.0, 609.0]
```

Python 寫出 Excel 檔案

- 使用 xlwt
 - 存取層次：試算表 -> 工作表 -> 資料列
 - 使用 Workbook() 建立試算表物件
 - 試算表物件使用 add_sheet() 加入新的工作表
 - 工作表物件使用 write() 指定列、欄，寫出資料

Python 寫出 Excel 檔案

- 使用 xlwt
 - 存取層次
 - 使用 Workbook
 - 試算表物件
 - 工作表物件

```
import xlwt

outfname = '2330-2.xls'
cols = ['Date', 'Open', 'High', 'Low', 'Close']
# list of list
listData = [['2021-06-10', 591, 599, 587, 599],
             ['2021-06-11', 602, 603, 600, 602],
             ['2021-06-15', 607, 609, 606, 609]]

wb = xlwt.Workbook()
sh = wb.add_sheet('sheet1', cell_overwrite_ok=True)

for i in range(len(cols)):
    sh.write(0, i, cols[i])                # 寫入column list

row = 1                                   # 第"0"列是標題列
for rec in listData:
    for j in range(len(rec)):
        sh.write(row, j, rec[j])          # 寫入日期與開高低收價格
    row += 1

wb.save(outfname)
```

Python 寫出 Excel 檔案

- 使用 xlwt

- 存取層次

- 使用 Workbook

- 試算表物件

- 工作表物件

```
import xlwt
```

```
outfname = '2330-2.xls'
```

```
cols = ['Date', 'Open', 'High', 'Low', 'Close']
```

```
# list of list
```

```
listData = [['2021-06-10', 591, 599, 587, 599],
```

```
['2021-06-11', 602, 603, 600, 602],
```

```
['2021-06-15', 607, 609, 606, 609]]
```

```
wb = xlwt.Workbook()
```

```
sh = wb.add_sheet('2330-2')
```

```
for i in range(len(listData)):
```

```
    sh.write(0, i, cols[i])
```

```
for i in range(1, len(listData)+1):
```

```
    row = 1
```

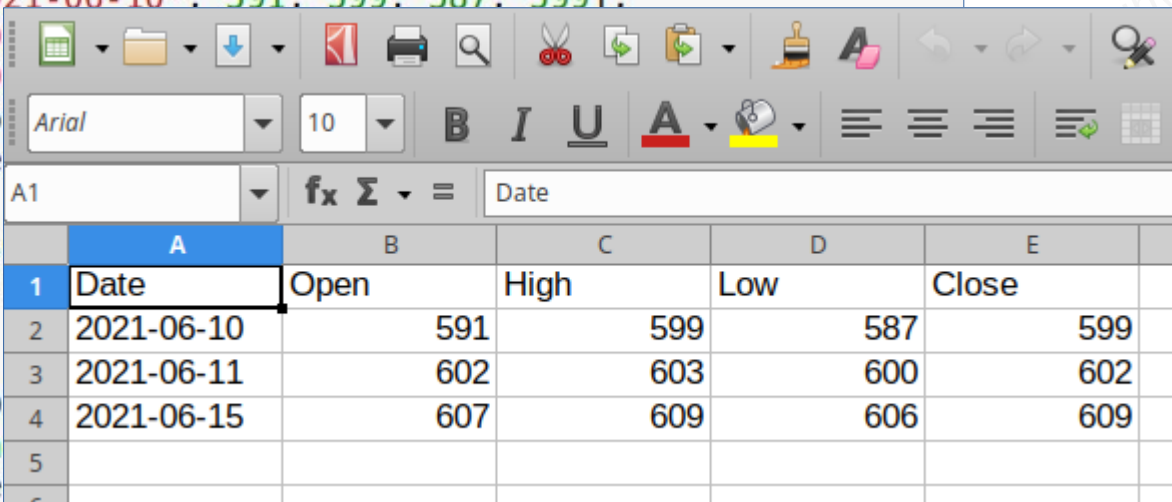
```
    for rec in listData[i-1]:
```

```
        for j in range(len(cols)):
```

```
            sh.write(row, j, rec[j])
```

```
        row += 1
```

```
wb.save(outfname)
```



	A	B	C	D	E
1	Date	Open	High	Low	Close
2	2021-06-10	591	599	587	599
3	2021-06-11	602	603	600	602
4	2021-06-15	607	609	606	609
5					
6					

這個講次中，你應該學到了 ...

- 了解 CSV 格式
- Python list 資料對 CSV 檔案的讀寫
- Python dict 資料對 CSV 檔案的讀寫
- Python 如何讀寫 Excel 檔案