

## 2.6 檔案讀寫

# 你將學習到

1. 檔案讀寫函數 `open()`
2. `with-as` 語法
3. `csv.writer()` 用法
4. `csv.reader()` 用法
5. `csv.DictWriter()` 用法
6. `csv.DictReader()` 用法
7. `f.tell()` 用法

# 檔案讀寫函數 open()

`open( 檔案路徑, 開啟模式, encoding="utf8" )`

開啟模式為:

"rt": 讀取文字模式。

"wt": 寫入文字模式, 覆蓋舊檔案。

"at": 新增文字模式, 舊檔案內最後一行開始寫入, 如果無則新建檔案

另外還有 "rb", "wb", "ab" 的二進制模式。

如果你需要讀寫中文字元, 你可加上 `encoding="utf8"` 參數, 指定utf8為檔案儲存編碼, 確保中文字元在不同作業系統上都能正確支援不會亂碼。

## 寫入檔案 f.write()

當代碼運行到 with block 以外時，會自動關閉檔案，這樣就保障檔案正確關閉。

```
# "wt" 以寫入文字模式開啟
```

```
with open("result3.txt", "wt", encoding="utf8") as f:
```

```
    f.write("00001.hk,13.0,2018-01-01 13:00")
```

## 讀取檔案 `f.read()` 和 `f.read().split('\n')`

```
with open("result4.txt", "rt", encoding="utf8") as f:  
    print( f.read() )
```

```
with open("result4.txt", "rt", encoding="utf8") as f:  
    print( f.read().split('\n') )
```

# "rt" 以讀取文字模式開啟

# `f.read()` 會返回檔案中的  
所有字串內容

# 在 `f.read()` 運用 `.split()` 函  
數後，會以 `\n` 作為分隔定  
位，然後返回一個 `list`。

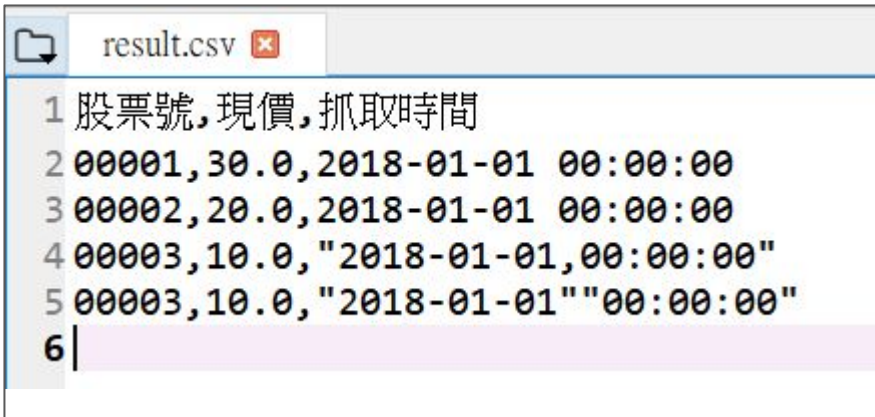
## 讀取檔案 for line in f

這個方法可以順序一行行去對檔案的字串進行操作，不會一次將整個檔案讀入記憶體。當檔案比較大時，例如 2GB 的文件，for line in f 的用法將會更節省記憶體。

```
with open("result4.txt", "rt", encoding= "utf8") as f:  
    for line in f:  
        print(line)
```

內容輸出：  
hello world  
  
hello handsome!

# CSV 檔案 (Comma Separated Values)



```
1 股票號,現價,抓取時間
2 00001,30.0,2018-01-01 00:00:00
3 00002,20.0,2018-01-01 00:00:00
4 00003,10.0,"2018-01-01,00:00:00"
5 00003,10.0,"2018-01-01""00:00:00"
6 |
```

Rule 1:

每筆資料用一行。

Rule 2:

每個欄位之間用逗號分隔。

Rule 3:

檔案第一行為欄位名稱定義。

Rule 4:

如果資料包含逗號, 則前後用" 包著。

Rule 5:

如果資料包含", 則寫成兩個"。

# CSV 模組

Python 提供 csv 模組以讀寫CSV檔案。

把每行以 List 形式讀寫:

`csv.writer()`

`csv.reader()`

把每行以 Dictionary 形式讀寫:

`csv.DictWriter()`

`csv.DictReader()`

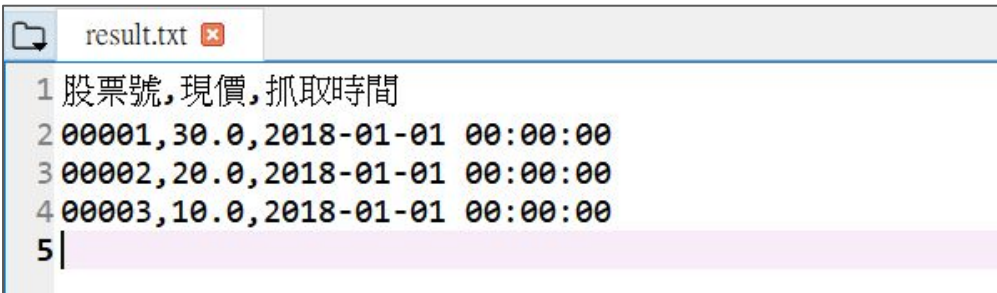


# CSV 寫入檔案 csv.writer()

```
import csv
```

```
with open("result.csv", "wt", newline="", encoding="utf8") as f:  
    writer = csv.writer(f)  
    writer.writerow( ["股票號", "現價", "抓取時間"] )  
    writer.writerow( ["00001", "30.0", "2018-01-01 00:00:00"] )  
    writer.writerow( ["00002", "20.0", "2018-01-01 00:00:00"] )  
    writer.writerow( ["00003", "10.0", "2018-01-01 00:00:00"] )
```

注意用 csv 模組時，這裡要加 `newline=""`



```
result.txt  
1 股票號,現價,抓取時間  
2 00001,30.0,2018-01-01 00:00:00  
3 00002,20.0,2018-01-01 00:00:00  
4 00003,10.0,2018-01-01 00:00:00  
5 |
```

## CSV 讀取檔案 csv.reader()

```
import csv
```

```
with open(file, "rt", newline="", encoding="utf8") as f:
```

```
# 以CSV格式讀取
```

```
reader = csv.reader(f)
```

```
# 把資料每行讀出
```

```
for row in reader:  
    print( row )
```

```
In [27]: with open("result.txt", "rt", newline="") as f:  
...:     reader = csv.reader(f)  
...:     for row in reader:  
...:         print( row )  
['股票號', '現價', '抓取時間']  
['00001', '30.0', '2018-01-01 00:00:00']  
['00002', '20.0', '2018-01-01 00:00:00']  
['00003', '10.0', '2018-01-01 00:00:00']
```

# CSV 寫入檔案 csv.DictWriter()

```
import csv
```

```
with open("result.txt", "wt", newline="", encoding="utf8") as f:
```

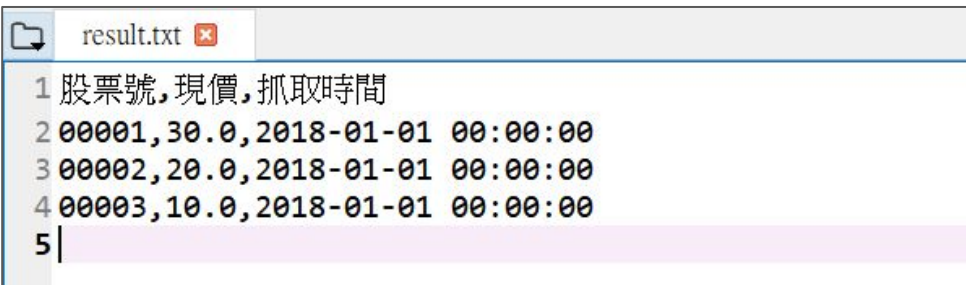
```
    writer = csv.DictWriter(f, ["股票號", "現價", "抓取時間"])
```

```
    writer.writeheader()
```

```
    writer.writerow( {"股票號": "00001", "現價": "30.0", "抓取時間": "2018-01-01 00:00:00"} )
```

```
    writer.writerow( {"股票號": "00002", "現價": "20.0", "抓取時間": "2018-01-01 00:00:00"} )
```

```
    writer.writerow( {"股票號": "00003", "現價": "10.0", "抓取時間": "2018-01-01 00:00:00"} )
```



```
result.txt x
1 股票號,現價,抓取時間
2 00001,30.0,2018-01-01 00:00:00
3 00002,20.0,2018-01-01 00:00:00
4 00003,10.0,2018-01-01 00:00:00
5 |
```

## CSV 讀取檔案 csv.DictReader()

```
import csv
```

```
with open("result.txt", "rt", newline="", encoding="utf8") as f:  
    reader = csv.DictReader(f)  
    for row in reader:  
        print( row["現價"] )
```

```
In [26]: with open("result.txt", "rt", newline="") as f:  
...:     reader = csv.DictReader(f)  
...:     for row in reader:  
...:         print( "%s %s %s" % (row["股票號"], row["現價"], row["抓取時間"]) )  
00001 30.0 2018-01-01 00:00:00  
00002 20.0 2018-01-01 00:00:00  
00003 10.0 2018-01-01 00:00:00
```

## 取得檔案指針位置 f.tell()

檔案用 append-mode 開啟時，檔案指針 (cursor) 預設在檔案尾部。  
我們可利用這項功能來檢測檔案是否為空檔案。因為空檔案會返回 0。

```
with open(file, "at", newline="", encoding="utf8") as f:  
    print( f.tell() )
```

# 學習回顧

1. 檔案讀寫函數 `open()`
2. `with-as` 語法
3. `csv.writer()` 用法
4. `csv.reader()` 用法
5. `csv.DictWriter()` 用法
6. `csv.DictReader()` 用法
7. `f.tell()` 用法