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 - 金融大數據分析

標的 1. 台灣證券交易所

葉建華 (Yeh, Jian-hua)

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

講次內容

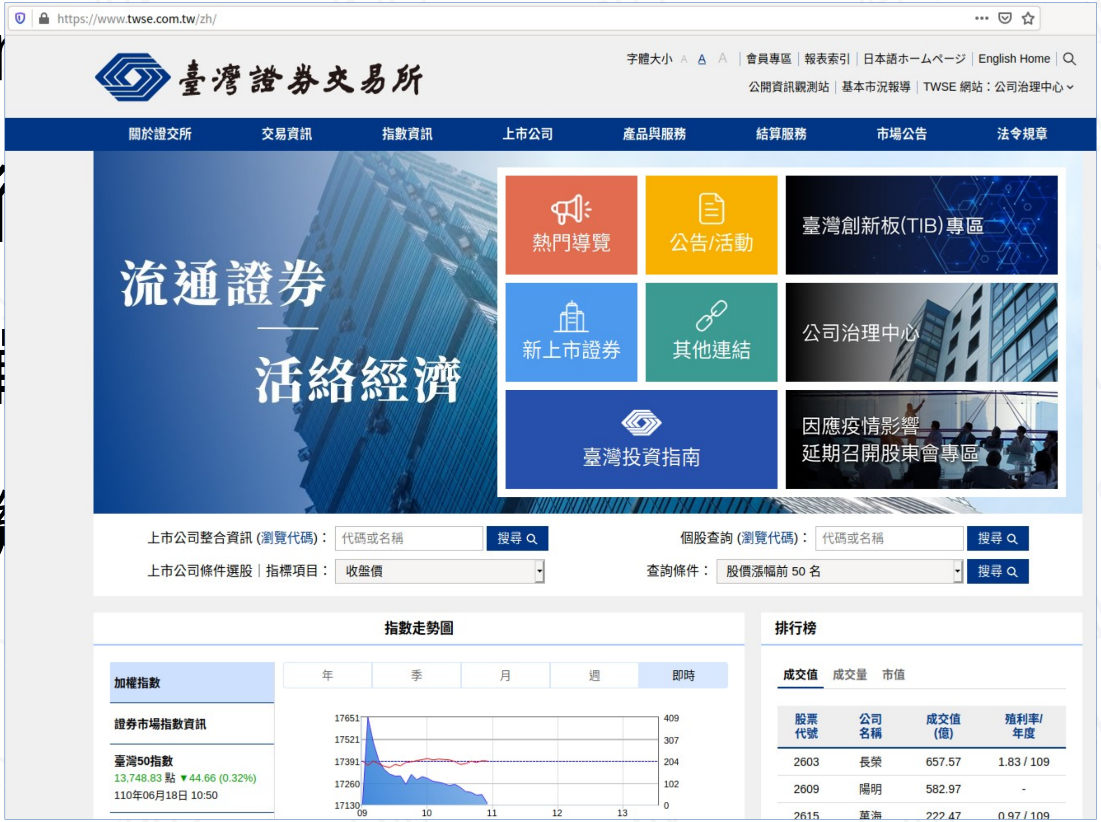
- TWSE 每日收盤資訊介紹
- 網路爬蟲與資料庫的規劃與操作
- SQL 收盤價查詢
- 專題：上市公司收盤價走勢

台灣證券交易所官網

- https://www.twse.com.tw
- 是 https 哦！（還記得嗎？）
- .com 代表商業公司單位
- .tw 代表台灣地區的機構

台灣證券交易所官網

- https://www.twse.co
- 是 https 哦！ （還記
- .com 代表商業公司單
- .tw 代表台灣地區的



https://www.twse.com.tw/zh/

臺灣證券交易所

字體大小 A A A | 會員專區 | 報表索引 | 日本語ホームページ | English Home

公開資訊觀測站 | 基本市況報導 | TWSE 網站：公司治理中心

關於證交所 交易資訊 指數資訊 上市公司 產品與服務 結算服務 市場公告 法令規章

流通證券

活絡經濟

熱門導覽 公告/活動 臺灣創新板(TIB)專區

新上市證券 其他連結 公司治理中心

臺灣投資指南 因應疫情影響延期召開股東會專區

上市公司整合資訊 (瀏覽代碼)：代碼或名稱 搜尋

個股查詢 (瀏覽代碼)：代碼或名稱 搜尋

上市公司條件選股 | 指標項目：收盤價

查詢條件：股價漲幅前 50 名 搜尋

指數走勢圖

加權指數

證券市場指數資訊

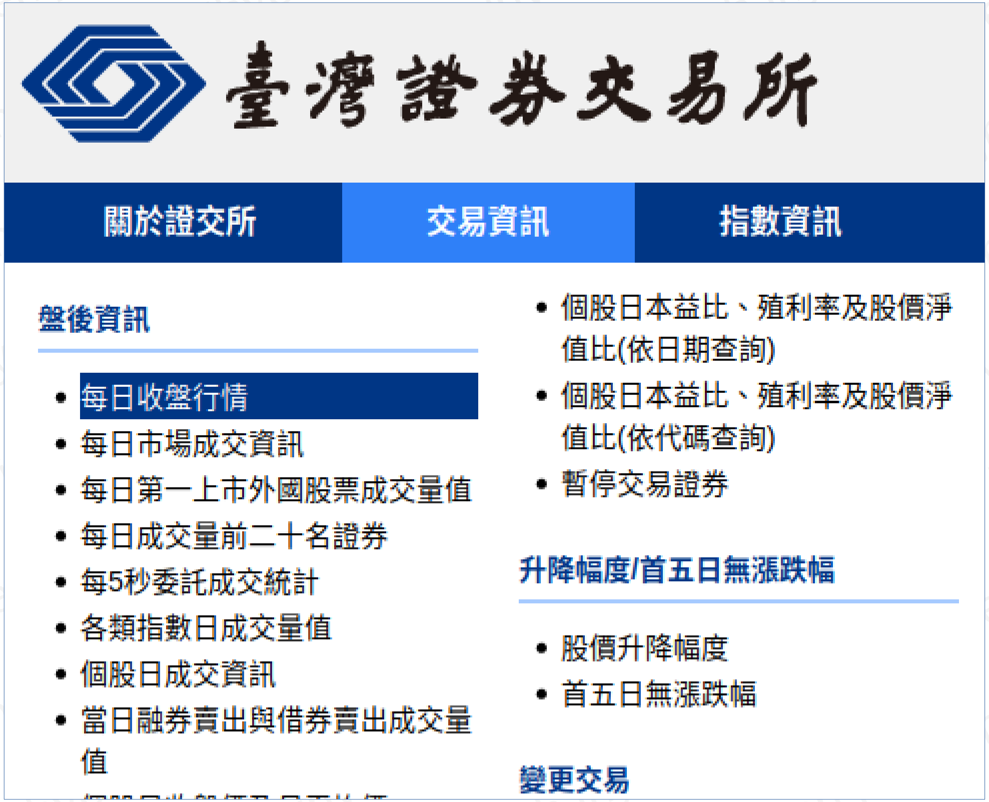
臺灣50指數
13,748.83 點 ▼44.66 (0.32%)
110年06月18日 10:50

排行榜

成交值 成交量 市值

股票代號	公司名稱	成交值(億)	殖利率/年度
2603	長榮	657.57	1.83 / 109
2609	陽明	582.97	-
2615	萬海	222.47	0.97 / 109

TWSE 每日收盤資訊介紹



臺灣證券交易所

盤後資訊

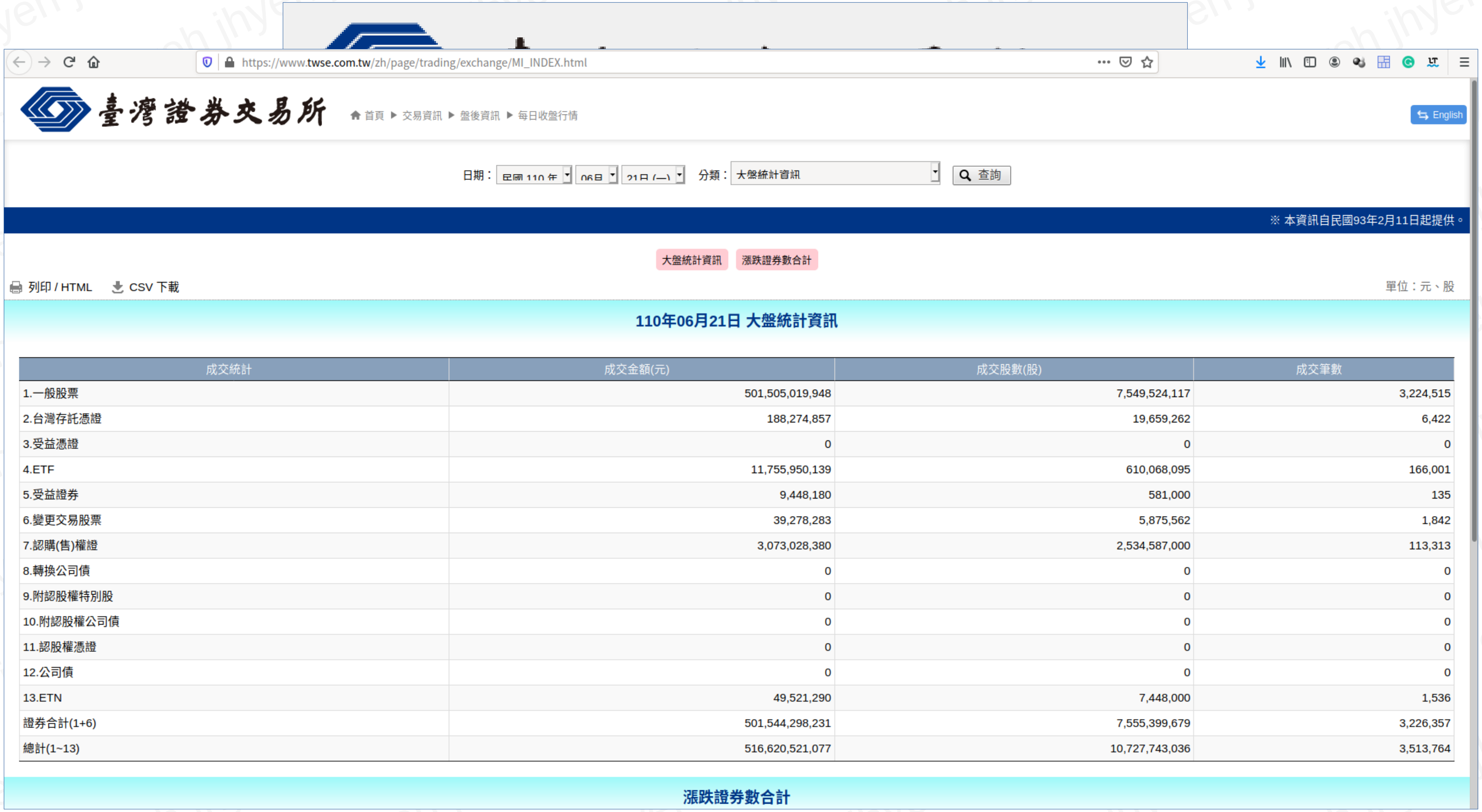
- 每日收盤行情
- 每日市場成交資訊
- 每日第一上市外國股票成交量值
- 每日成交量前二十名證券
- 每5秒委託成交統計
- 各類指數日成交量值
- 個股日成交資訊
- 當日融券賣出與借券賣出成交量值
- 個股日本益比、殖利率及股價淨值比(依日期查詢)
- 個股日本益比、殖利率及股價淨值比(依代碼查詢)
- 暫停交易證券

升降幅度/首五日無漲跌幅

- 股價升降幅度
- 首五日無漲跌幅

變更交易

TWSE 每日收盤資訊介紹



https://www.twse.com.tw/zh/page/trading/exchange/MI_INDEX.html
臺灣證券交易所
首頁 ▶ 交易資訊 ▶ 盤後資訊 ▶ 每日收盤行情
English
查詢
大盤統計資訊
漲跌證券數合計
列印 / HTML
CSV 下載

日期：民國 110 年 06月 21日 (一) 分類：大盤統計資訊

※ 本資訊自民國93年2月11日起提供。

單位：元、股

110年06月21日 大盤統計資訊

成交統計	成交金額(元)	成交股數(股)	成交筆數
1.一般股票	501,505,019,948	7,549,524,117	3,224,515
2.台灣存託憑證	188,274,857	19,659,262	6,422
3.受益憑證	0	0	0
4.ETF	11,755,950,139	610,068,095	166,001
5.受益證券	9,448,180	581,000	135
6.變更交易股票	39,278,283	5,875,562	1,842
7.認購(售)權證	3,073,028,380	2,534,587,000	113,313
8.轉換公司債	0	0	0
9.附認股權特別股	0	0	0
10.附認股權公司債	0	0	0
11.認股權憑證	0	0	0
12.公司債	0	0	0
13.ETN	49,521,290	7,448,000	1,536
證券合計(1+6)	501,544,298,231	7,555,399,679	3,226,357
總計(1~13)	516,620,521,077	10,727,743,036	3,513,764

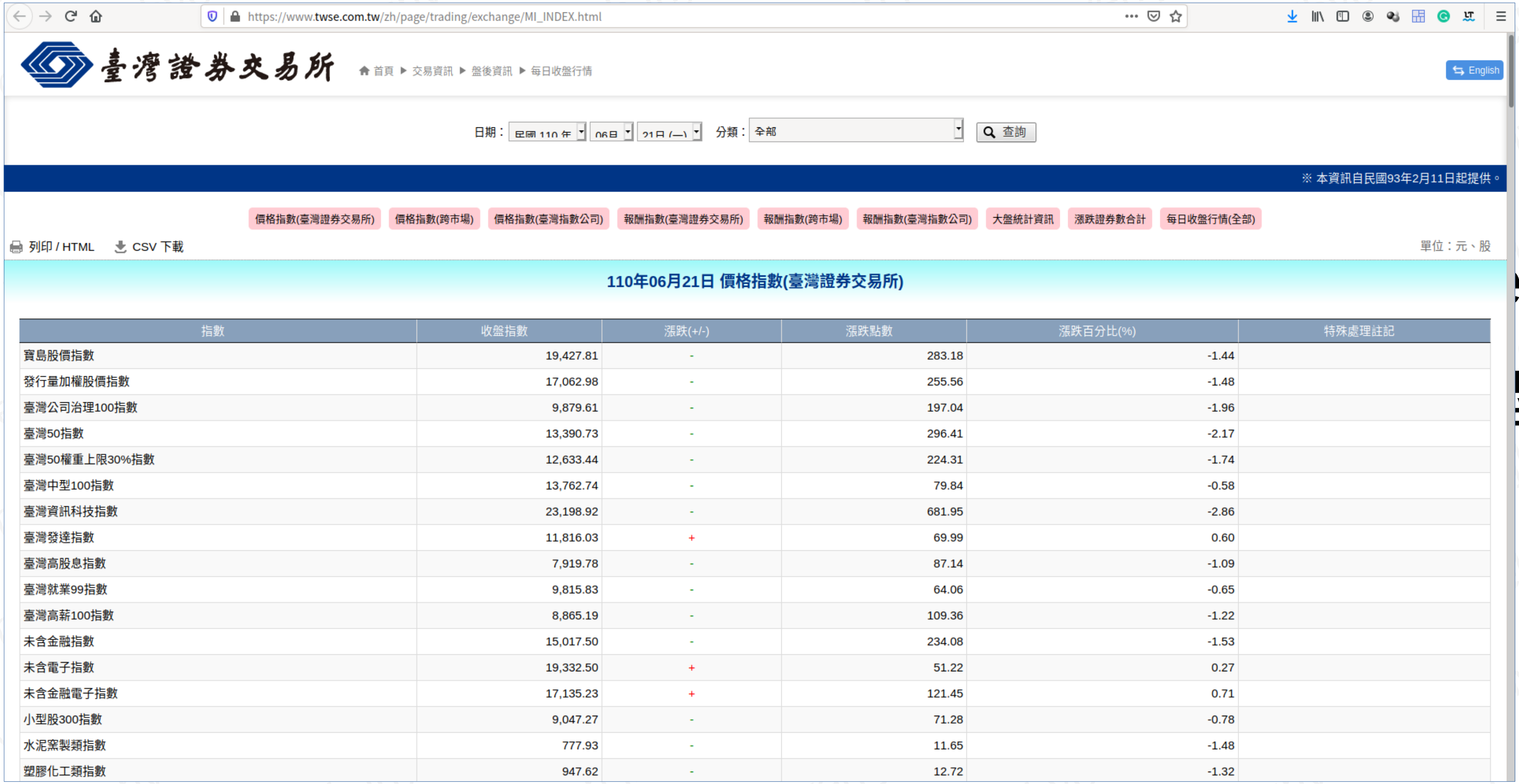
漲跌證券數合計

TWSE 每日收盤資訊介紹

- 觀察一下左上角「CSV 下載」 ...
  - https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&date=20210621&type=MS
  - 「csv」格式、日期「20210621」、「MS」型態 (Market Summary)
  - 分類如果選「全部」呢?

TWSE 每日收盤資訊介紹

-



https://www.twse.com.tw/zh/page/trading/exchange/MI_INDEX.html

臺灣證券交易所

首頁 ▶ 交易資訊 ▶ 盤後資訊 ▶ 每日收盤行情

English

日期：民國 110 年 06月 21日 (一) 分類：全部 查詢

※ 本資訊自民國93年2月11日起提供。

價格指數(臺灣證券交易所) 價格指數(跨市場) 價格指數(臺灣指數公司) 報酬指數(臺灣證券交易所) 報酬指數(跨市場) 報酬指數(臺灣指數公司) 大盤統計資訊 漲跌證券數合計 每日收盤行情(全部)

列印 / HTML CSV 下載

單位：元、股

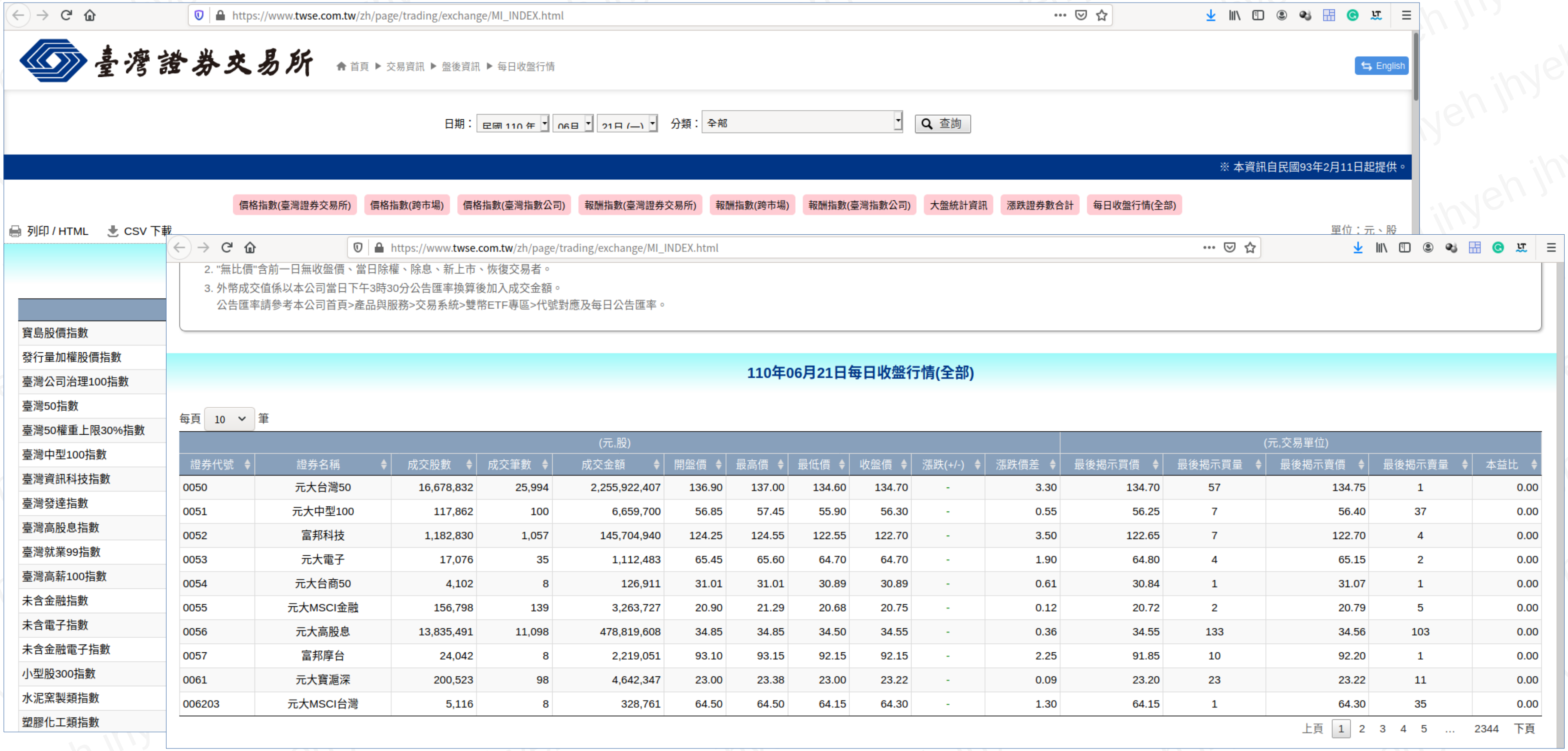
110年06月21日 價格指數(臺灣證券交易所)

指數	收盤指數	漲跌(+/-)	漲跌點數	漲跌百分比(%)	特殊處理註記
寶島股價指數	19,427.81	-	283.18	-1.44	
發行量加權股價指數	17,062.98	-	255.56	-1.48	
臺灣公司治理100指數	9,879.61	-	197.04	-1.96	
臺灣50指數	13,390.73	-	296.41	-2.17	
臺灣50權重上限30%指數	12,633.44	-	224.31	-1.74	
臺灣中型100指數	13,762.74	-	79.84	-0.58	
臺灣資訊科技指數	23,198.92	-	681.95	-2.86	
臺灣發達指數	11,816.03	+	69.99	0.60	
臺灣高股息指數	7,919.78	-	87.14	-1.09	
臺灣就業99指數	9,815.83	-	64.06	-0.65	
臺灣高薪100指數	8,865.19	-	109.36	-1.22	
未含金融指數	15,017.50	-	234.08	-1.53	
未含電子指數	19,332.50	+	51.22	0.27	
未含金融電子指數	17,135.23	+	121.45	0.71	
小型股300指數	9,047.27	-	71.28	-0.78	
水泥窯製類指數	777.93	-	11.65	-1.48	
塑膠化工類指數	947.62	-	12.72	-1.32	

=MS

態

TWSE 每日收盤資訊介紹

- 

https://www.twse.com.tw/zh/page/trading/exchange/MI_INDEX.html

臺灣證券交易所

首頁 ▶ 交易資訊 ▶ 盤後資訊 ▶ 每日收盤行情

English

日期：民國 110 年 06月 21日 (一) 分類：全部 查詢

※ 本資訊自民國93年2月11日起提供。

價格指數(臺灣證券交易所) 價格指數(跨市場) 價格指數(臺灣指數公司) 報酬指數(臺灣證券交易所) 報酬指數(跨市場) 報酬指數(臺灣指數公司) 大盤統計資訊 漲跌證券數合計 每日收盤行情(全部)

列印 / HTML CSV 下載

單位：元、股

寶島股價指數
發行量加權股價指數
臺灣公司治理100指數
臺灣50指數
臺灣50權重上限30%指數
臺灣中型100指數
臺灣資訊科技指數
臺灣發達指數
臺灣高股息指數
臺灣就業99指數
臺灣高薪100指數
未含金融指數
未含電子指數
未含金融電子指數
小型股300指數
水泥窯製類指數
塑膠化工類指數

https://www.twse.com.tw/zh/page/trading/exchange/MI_INDEX.html

2. "無比價"含前一日無收盤價、當日除權、除息、新上市、恢復交易者。
3. 外幣成交值係以本公司當日下午3時30分公告匯率換算後加入成交金額。
公告匯率請參考本公司首頁>產品與服務>交易系統>雙幣ETF專區>代號對應及每日公告匯率。

110年06月21日每日收盤行情(全部)

每頁 10 筆

(元,股)											(元,交易單位)				
證券代號	證券名稱	成交股數	成交筆數	成交金額	開盤價	最高價	最低價	收盤價	漲跌(+/-)	漲跌價差	最後揭示買價	最後揭示買量	最後揭示賣價	最後揭示賣量	本益比
0050	元大台灣50	16,678,832	25,994	2,255,922,407	136.90	137.00	134.60	134.70	-	3.30	134.70	57	134.75	1	0.00
0051	元大中型100	117,862	100	6,659,700	56.85	57.45	55.90	56.30	-	0.55	56.25	7	56.40	37	0.00
0052	富邦科技	1,182,830	1,057	145,704,940	124.25	124.55	122.55	122.70	-	3.50	122.65	7	122.70	4	0.00
0053	元大電子	17,076	35	1,112,483	65.45	65.60	64.70	64.70	-	1.90	64.80	4	65.15	2	0.00
0054	元大台商50	4,102	8	126,911	31.01	31.01	30.89	30.89	-	0.61	30.84	1	31.07	1	0.00
0055	元大MSCI金融	156,798	139	3,263,727	20.90	21.29	20.68	20.75	-	0.12	20.72	2	20.79	5	0.00
0056	元大高股息	13,835,491	11,098	478,819,608	34.85	34.85	34.50	34.55	-	0.36	34.55	133	34.56	103	0.00
0057	富邦摩台	24,042	8	2,219,051	93.10	93.15	92.15	92.15	-	2.25	91.85	10	92.20	1	0.00
0061	元大寶滬深	200,523	98	4,642,347	23.00	23.38	23.00	23.22	-	0.09	23.20	23	23.22	11	0.00
006203	元大MSCI台灣	5,116	8	328,761	64.50	64.50	64.15	64.30	-	1.30	64.15	1	64.30	35	0.00

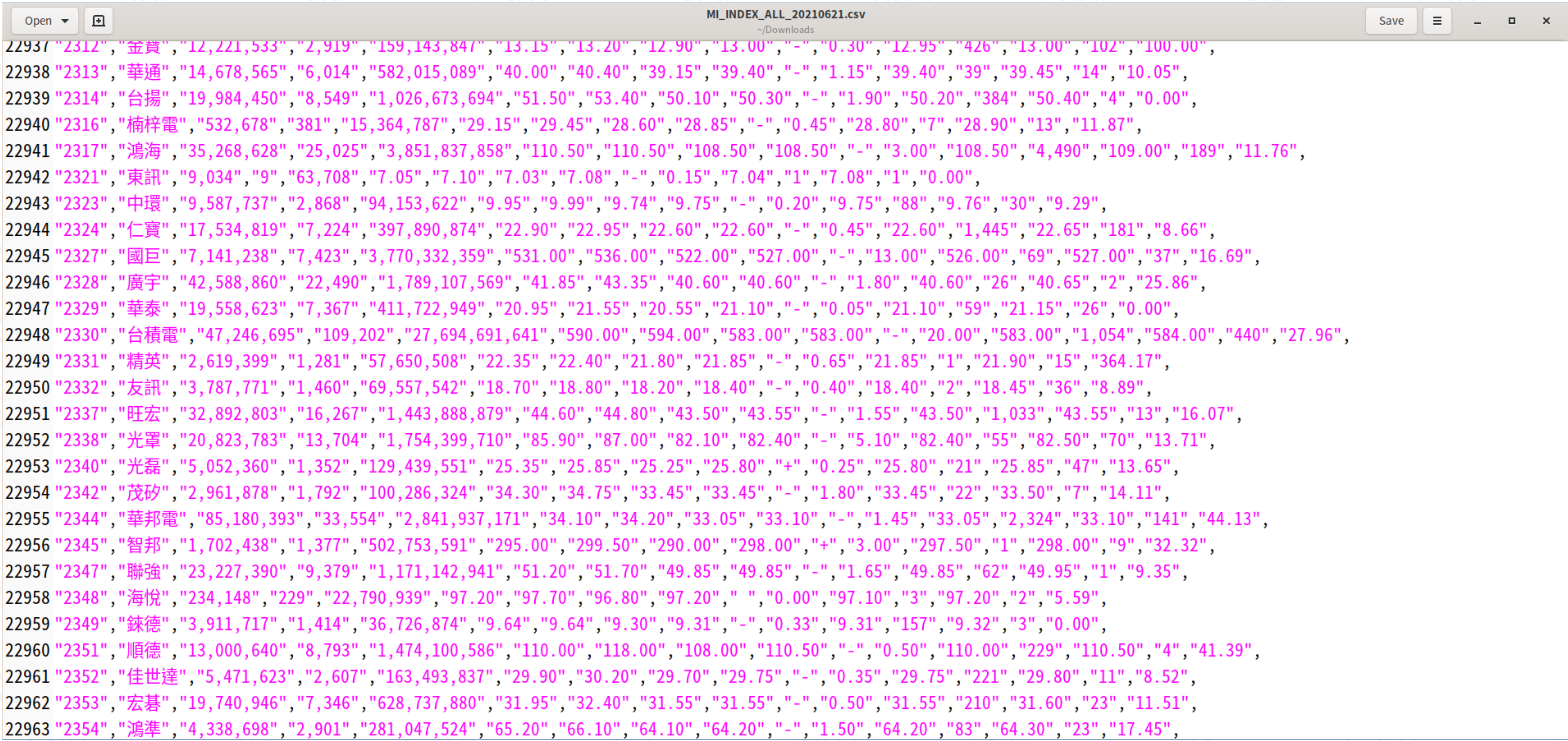
上頁 1 2 3 4 5 … 2344 下頁

拉到最下方，出現個股收盤資訊了！

TWSE 每日收盤資訊介紹

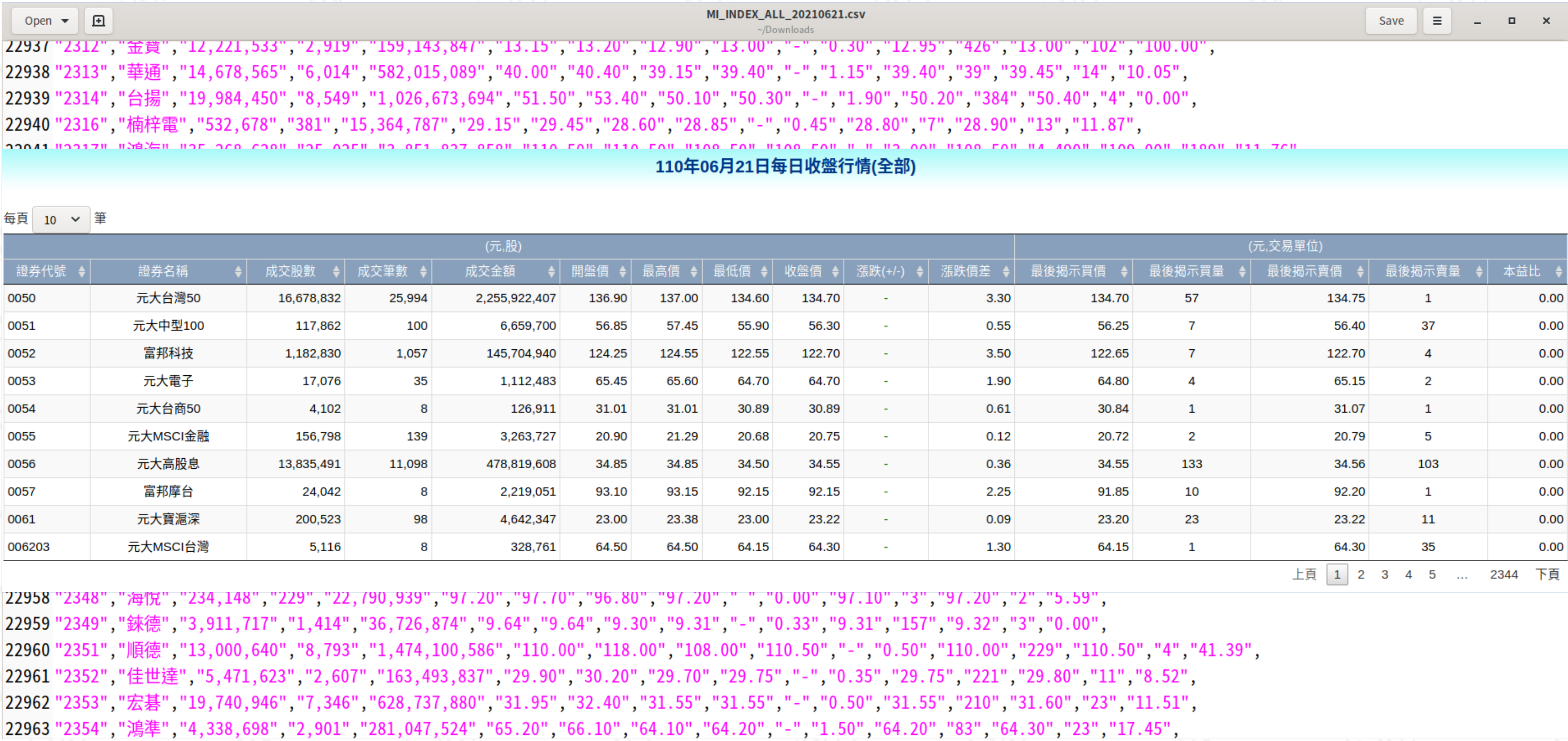
- 左上角還是有「 CSV 下載」 ...
  - https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&date=20210621&type=ALL
  - 「 ALL 」型態 !
  - 下載 CSV 來看看吧 !

是哪些欄位啊？！



```
"2312","金寶","12,221,533","2,919","159,143,847","13.15","13.20","12.90","13.00","-","0.30","12.95","426","13.00","102","100.00",
"2313","華通","14,678,565","6,014","582,015,089","40.00","40.40","39.15","39.40","-","1.15","39.40","39","39.45","14","10.05",
"2314","台揚","19,984,450","8,549","1,026,673,694","51.50","53.40","50.10","50.30","-","1.90","50.20","384","50.40","4","0.00",
"2316","楠梓電","532,678","381","15,364,787","29.15","29.45","28.60","28.85","-","0.45","28.80","7","28.90","13","11.87",
"2317","鴻海","35,268,628","25,025","3,851,837,858","110.50","110.50","108.50","108.50","-","3.00","108.50","4,490","109.00","189","11.76",
"2321","東訊","9,034","9","63,708","7.05","7.10","7.03","7.08","-","0.15","7.04","1","7.08","1","0.00",
"2323","中環","9,587,737","2,868","94,153,622","9.95","9.99","9.74","9.75","-","0.20","9.75","88","9.76","30","9.29",
"2324","仁寶","17,534,819","7,224","397,890,874","22.90","22.95","22.60","22.60","-","0.45","22.60","1,445","22.65","181","8.66",
"2327","國巨","7,141,238","7,423","3,770,332,359","531.00","536.00","522.00","527.00","-","13.00","526.00","69","527.00","37","16.69",
"2328","廣宇","42,588,860","22,490","1,789,107,569","41.85","43.35","40.60","40.60","-","1.80","40.60","26","40.65","2","25.86",
"2329","華泰","19,558,623","7,367","411,722,949","20.95","21.55","20.55","21.10","-","0.05","21.10","59","21.15","26","0.00",
"2330","台積電","47,246,695","109,202","27,694,691,641","590.00","594.00","583.00","583.00","-","20.00","583.00","1,054","584.00","440","27.96",
"2331","精英","2,619,399","1,281","57,650,508","22.35","22.40","21.80","21.85","-","0.65","21.85","1","21.90","15","364.17",
"2332","友訊","3,787,771","1,460","69,557,542","18.70","18.80","18.20","18.40","-","0.40","18.40","2","18.45","36","8.89",
"2337","旺宏","32,892,803","16,267","1,443,888,879","44.60","44.80","43.50","43.55","-","1.55","43.50","1,033","43.55","13","16.07",
"2338","光罩","20,823,783","13,704","1,754,399,710","85.90","87.00","82.10","82.40","-","5.10","82.40","55","82.50","70","13.71",
"2340","光磊","5,052,360","1,352","129,439,551","25.35","25.85","25.25","25.80","+","0.25","25.80","21","25.85","47","13.65",
"2342","茂矽","2,961,878","1,792","100,286,324","34.30","34.75","33.45","33.45","-","1.80","33.45","22","33.50","7","14.11",
"2344","華邦電","85,180,393","33,554","2,841,937,171","34.10","34.20","33.05","33.10","-","1.45","33.05","2,324","33.10","141","44.13",
"2345","智邦","1,702,438","1,377","502,753,591","295.00","299.50","290.00","298.00","+","3.00","297.50","1","298.00","9","32.32",
"2347","聯強","23,227,390","9,379","1,171,142,941","51.20","51.70","49.85","49.85","-","1.65","49.85","62","49.95","1","9.35",
"2348","海悅","234,148","229","22,790,939","97.20","97.70","96.80","97.20"," ","0.00","97.10","3","97.20","2","5.59",
"2349","錸德","3,911,717","1,414","36,726,874","9.64","9.64","9.30","9.31","-","0.33","9.31","157","9.32","3","0.00",
"2351","順德","13,000,640","8,793","1,474,100,586","110.00","118.00","108.00","110.50","-","0.50","110.00","229","110.50","4","41.39",
"2352","佳世達","5,471,623","2,607","163,493,837","29.90","30.20","29.70","29.75","-","0.35","29.75","221","29.80","11","8.52",
"2353","宏碁","19,740,946","7,346","628,737,880","31.95","32.40","31.55","31.55","-","0.50","31.55","210","31.60","23","11.51",
"2354","鴻準","4,338,698","2,901","281,047,524","65.20","66.10","64.10","64.20","-","1.50","64.20","83","64.30","23","17.45",
```

是哪些欄位啊？！



```
"2312","金寶","12,221,533","2,919","159,143,847","13.15","13.20","12.90","13.00","-","0.30","12.95","426","13.00","102","100.00",
"2313","華通","14,678,565","6,014","582,015,089","40.00","40.40","39.15","39.40","-","1.15","39.40","39","39.45","14","10.05",
"2314","台揚","19,984,450","8,549","1,026,673,694","51.50","53.40","50.10","50.30","-","1.90","50.20","384","50.40","4","0.00",
"2316","楠梓電","532,678","381","15,364,787","29.15","29.45","28.60","28.85","-","0.45","28.80","7","28.90","13","11.87",
"2317","鴻海","35,268,628","25,025","3,851,837,858","110.50","110.50","108.50","108.50"," ","2.00","108.50","4,490","109.00","180","11.76"
```

110年06月21日每日收盤行情(全部)

每頁 10 筆

(元,股)											(元,交易單位)				
證券代號	證券名稱	成交股數	成交筆數	成交金額	開盤價	最高價	最低價	收盤價	漲跌(+/-)	漲跌價差	最後揭示買價	最後揭示買量	最後揭示賣價	最後揭示賣量	本益比
0050	元大台灣50	16,678,832	25,994	2,255,922,407	136.90	137.00	134.60	134.70	-	3.30	134.70	57	134.75	1	0.00
0051	元大中型100	117,862	100	6,659,700	56.85	57.45	55.90	56.30	-	0.55	56.25	7	56.40	37	0.00
0052	富邦科技	1,182,830	1,057	145,704,940	124.25	124.55	122.55	122.70	-	3.50	122.65	7	122.70	4	0.00
0053	元大電子	17,076	35	1,112,483	65.45	65.60	64.70	64.70	-	1.90	64.80	4	65.15	2	0.00
0054	元大台商50	4,102	8	126,911	31.01	31.01	30.89	30.89	-	0.61	30.84	1	31.07	1	0.00
0055	元大MSCI金融	156,798	139	3,263,727	20.90	21.29	20.68	20.75	-	0.12	20.72	2	20.79	5	0.00
0056	元大高股息	13,835,491	11,098	478,819,608	34.85	34.85	34.50	34.55	-	0.36	34.55	133	34.56	103	0.00
0057	富邦摩台	24,042	8	2,219,051	93.10	93.15	92.15	92.15	-	2.25	91.85	10	92.20	1	0.00
0061	元大寶滬深	200,523	98	4,642,347	23.00	23.38	23.00	23.22	-	0.09	23.20	23	23.22	11	0.00
006203	元大MSCI台灣	5,116	8	328,761	64.50	64.50	64.15	64.30	-	1.30	64.15	1	64.30	35	0.00

上頁 1 2 3 4 5 … 2344 下頁

```
"2348","海悅","234,148","229","22,790,939","97.20","97.70","96.80","97.20"," ","0.00","97.10","3","97.20","2","5.59",
"2349","鋐德","3,911,717","1,414","36,726,874","9.64","9.64","9.30","9.31","-","0.33","9.31","157","9.32","3","0.00",
"2351","順德","13,000,640","8,793","1,474,100,586","110.00","118.00","108.00","110.50","-","0.50","110.00","229","110.50","4","41.39",
"2352","佳世達","5,471,623","2,607","163,493,837","29.90","30.20","29.70","29.75","-","0.35","29.75","221","29.80","11","8.52",
"2353","宏碁","19,740,946","7,346","628,737,880","31.95","32.40","31.55","31.55","-","0.50","31.55","210","31.60","23","11.51",
"2354","鴻準","4,338,698","2,901","281,047,524","65.20","66.10","64.10","64.20","-","1.50","64.20","83","64.30","23","17.45",
```

CSV 格式

- 第 215 行:
  - "證券代號","證券名稱","成交股數","成交筆數","成交金額","開盤價","最高價","最低價","收盤價","漲跌(+/-)","漲跌價差","最後揭示買價","最後揭示買量","最後揭示賣價","最後揭示賣量","本益比"
- 從 CSV 找到這行：「"證券代號","證券名稱"」
  - 從這以下，都是這種個股收盤資料
- 所以呢?

講次內容

- TWSE 每日收盤資訊介紹
- 網路爬蟲與資料庫的規劃與操作
- SQL 收盤價查詢
- 專題：上市公司收盤價走勢

從網路爬蟲到資料庫

- 分工！

我們設定 2020 全年

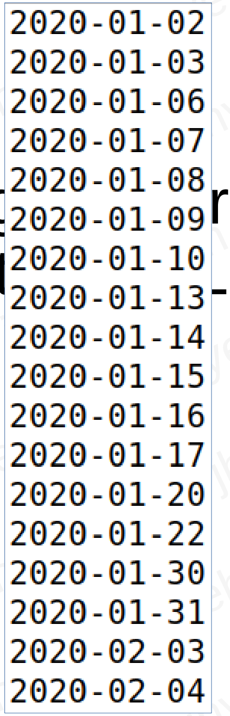
  - 網路爬蟲：把每日全市場交易資訊抓回來，CSV！
  - CSV 解析：把 CSV 依據欄位進行拆解
  - 資料庫：把拆解完的欄位對應資料儲存到資料表中

Part 1. 網路爬蟲

- 目標網址
  - https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&date=20210621&type=ALL
- 只要換日期就好了？！
  - 沒錯，但是哪些日期是交易日期?
  - 1. 抓回來，儲存成 CSV 檔案
  - 2. 抓回來，直接拆解 CSV 存成 Python 內部資料結構

Part 1. 網路爬蟲

- 目標網址
  - https://www.twse.com.tw/exchang rt/MI_INDEX?response=csv&date=20210621&t LL
- 只要換日期就好了？！
  - 沒錯，但是哪些日期是交易日期?
  - 1. 抓回來，儲存成 CSV 檔案
  - 2. 抓回來，直接拆解 CSV 存成 Python 內部資料結構



```
2020-01-02
2020-01-03
2020-01-06
2020-01-07
2020-01-08
2020-01-09
2020-01-10
2020-01-13
2020-01-14
2020-01-15
2020-01-16
2020-01-17
2020-01-20
2020-01-22
2020-01-30
2020-01-31
2020-02-03
2020-02-04
```

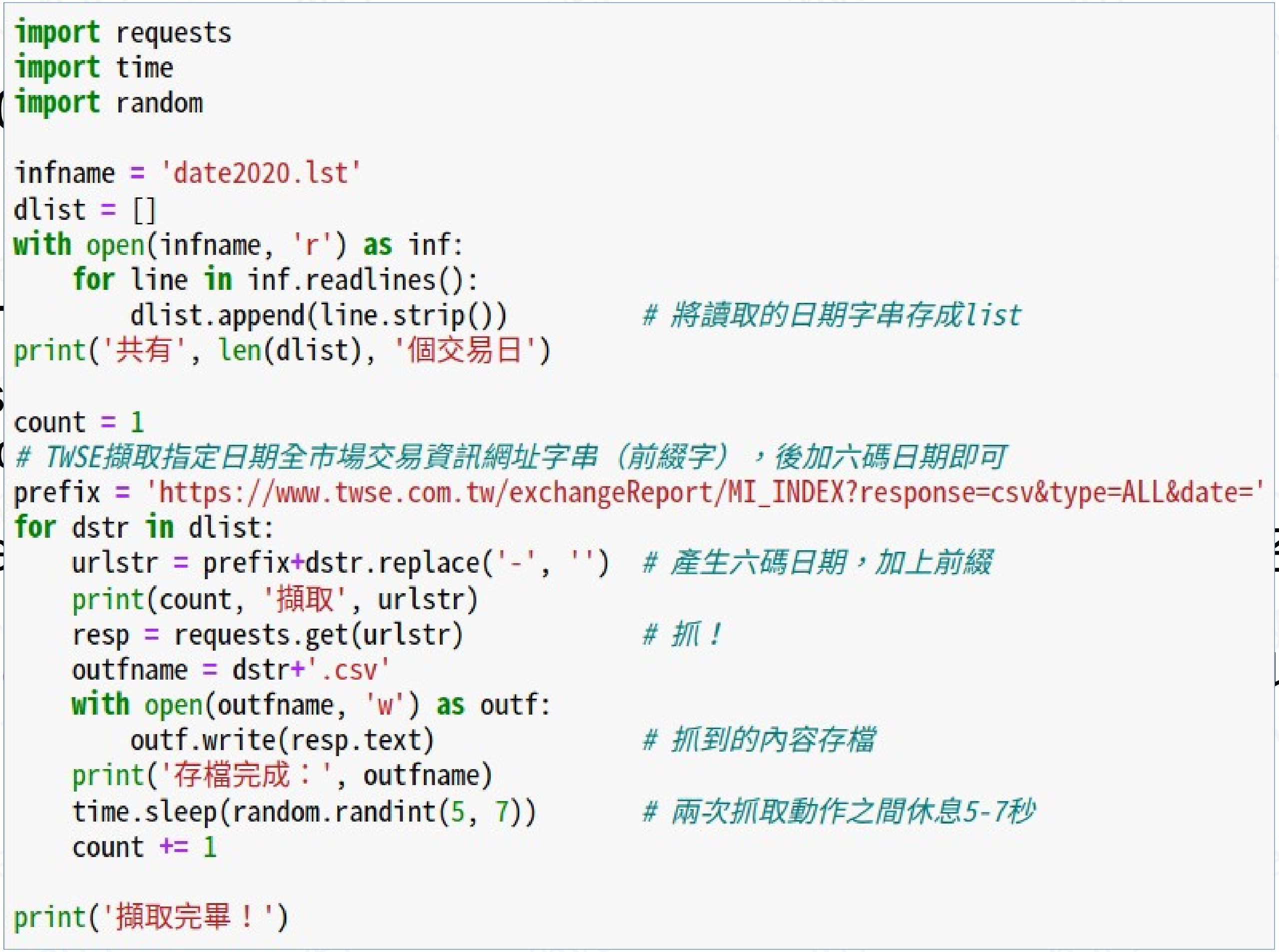
請下載 date2020.lst

爬蟲邏輯框架

- 載入 2020 年的所有交易日期
  - 讀取 date2020.lst
- 對每一個交易日期，製作擷取網頁的網址
  - https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&date=yyyymmdd&type=ALL
- 使用 requests 抓回 CSV 格式資料，存成檔案（以日期命名）
  - 用 requests.get() 方法，但是要記得台灣證券交易所**會擋**，所以要 time.sleep()

爬蟲邏輯框架

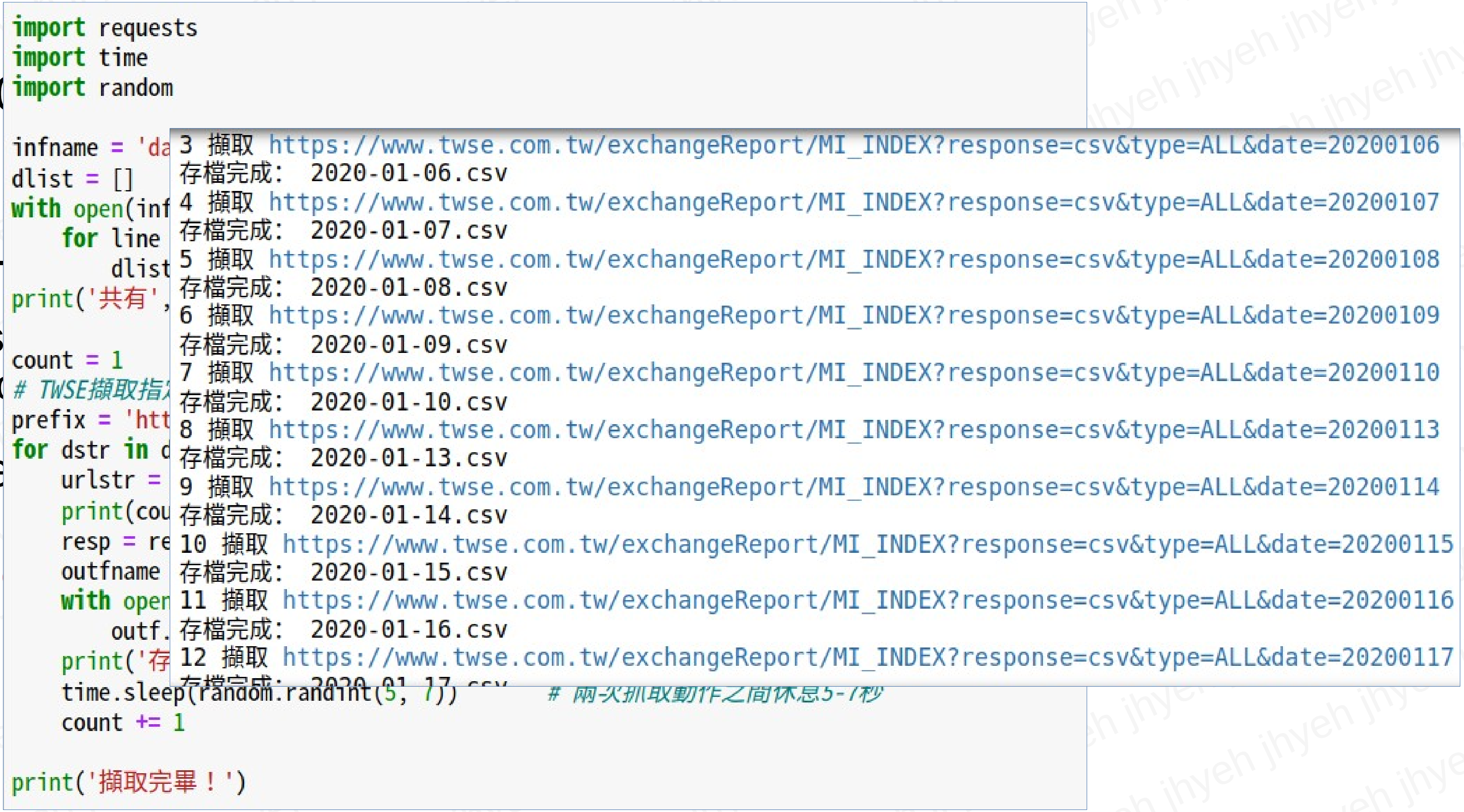
- 載入 20
  - 讀取
- 對每一
  - https
    resp
- 使用 re 名）
  - 用 re 以要 time.sleep()



```python
import requests
import time
import random

infname = 'date2020.lst'
dlist = []
with open(infname, 'r') as inf:
    for line in inf.readlines():
        dlist.append(line.strip())          # 將讀取的日期字串存成list
print('共有', len(dlist), '個交易日')

count = 1
# TWSE擷取指定日期全市場交易資訊網址字串（前綴字），後加六碼日期即可
prefix = 'https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&type=ALL&date='
for dstr in dlist:
    urlstr = prefix+dstr.replace('-', '')  # 產生六碼日期，加上前綴
    print(count, '擷取', urlstr)
    resp = requests.get(urlstr)            # 抓！
    outfname = dstr+'.csv'
    with open(outfname, 'w') as outf:
        outf.write(resp.text)              # 抓到的內容存檔
    print('存檔完成：', outfname)
    time.sleep(random.randint(5, 7))       # 兩次抓取動作之間休息5-7秒
    count += 1

print('擷取完畢！')
```

爬蟲邏輯框架

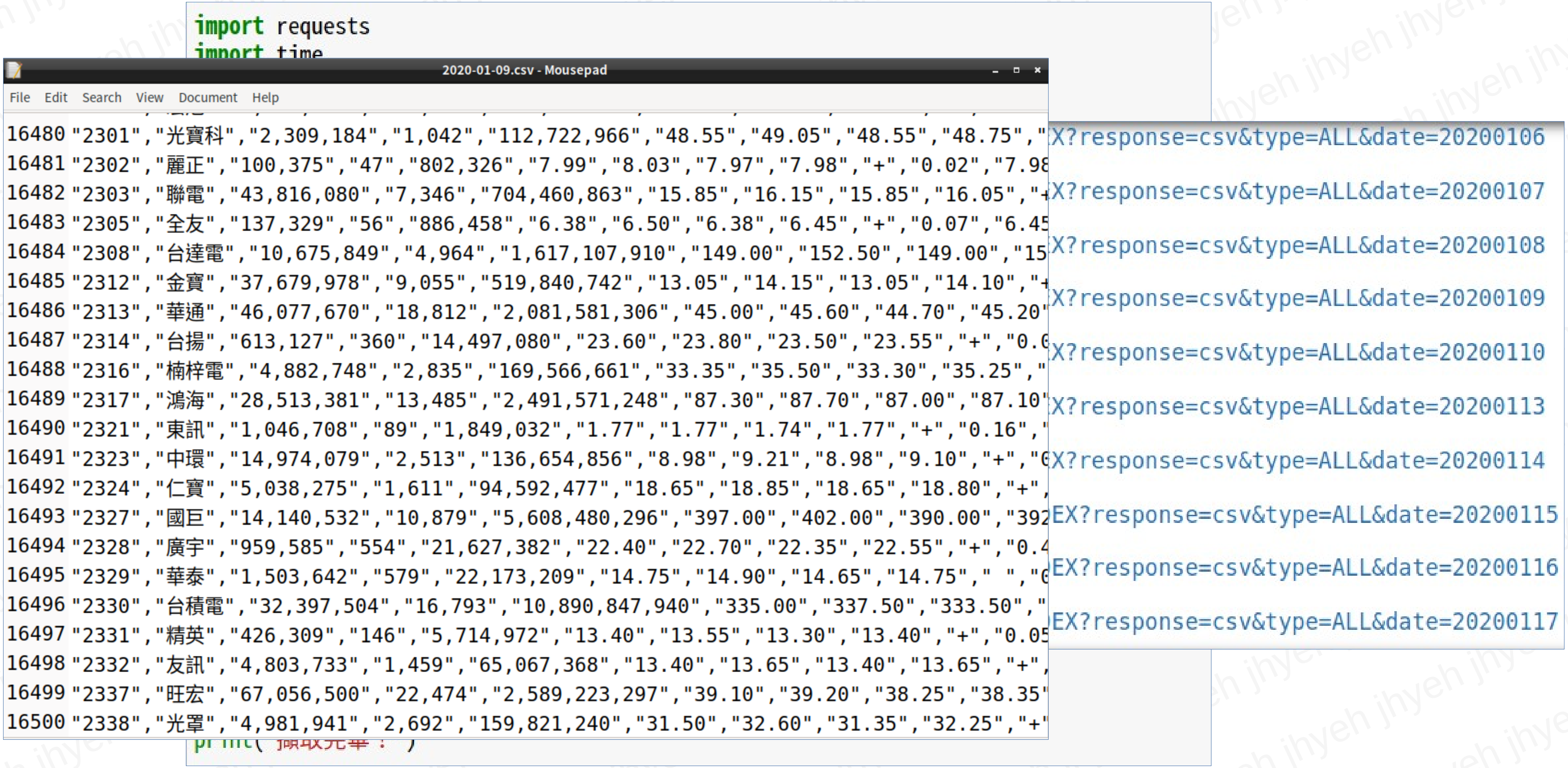
- 載入 20
  - 讀取
- 對每一
  - https
    resp
- 使用 re
  - 用 re



```
import requests
import time
import random

infname = 'da
dlist = []
with open(inf
    for line
        dlist
print('共有',

count = 1
# TWSE擷取指
prefix = 'htt
for dstr in d
    urlstr =
    print(cou
    resp = re
    outfname
    with open
        outf.
    print('存
    time.sleep(random.randint(5, 7))       # 兩次抓取動作之間休息5-7秒
    count += 1

print('擷取完畢！')
```

```
3 擷取 https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&type=ALL&date=20200106
存檔完成:  2020-01-06.csv
4 擷取 https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&type=ALL&date=20200107
存檔完成:  2020-01-07.csv
5 擷取 https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&type=ALL&date=20200108
存檔完成:  2020-01-08.csv
6 擷取 https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&type=ALL&date=20200109
存檔完成:  2020-01-09.csv
7 擷取 https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&type=ALL&date=20200110
存檔完成:  2020-01-10.csv
8 擷取 https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&type=ALL&date=20200113
存檔完成:  2020-01-13.csv
9 擷取 https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&type=ALL&date=20200114
存檔完成:  2020-01-14.csv
10 擷取 https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&type=ALL&date=20200115
存檔完成:  2020-01-15.csv
11 擷取 https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&type=ALL&date=20200116
存檔完成:  2020-01-16.csv
12 擷取 https://www.twse.com.tw/exchangeReport/MI_INDEX?response=csv&type=ALL&date=20200117
存檔完成:  2020-01-17.csv
```

爬蟲邏輯框架



```python
import requests
import time
```

2020-01-09.csv - Mousepad
File Edit Search View Document Help

```
16480 "2301","光寶科","2,309,184","1,042","112,722,966","48.55","49.05","48.55","48.75","
16481 "2302","麗正","100,375","47","802,326","7.99","8.03","7.97","7.98","+","0.02","7.98
16482 "2303","聯電","43,816,080","7,346","704,460,863","15.85","16.15","15.85","16.05","+
16483 "2305","全友","137,329","56","886,458","6.38","6.50","6.38","6.45","+","0.07","6.45
16484 "2308","台達電","10,675,849","4,964","1,617,107,910","149.00","152.50","149.00","15
16485 "2312","金寶","37,679,978","9,055","519,840,742","13.05","14.15","13.05","14.10","+
16486 "2313","華通","46,077,670","18,812","2,081,581,306","45.00","45.60","44.70","45.20"
16487 "2314","台揚","613,127","360","14,497,080","23.60","23.80","23.50","23.55","+","0.0
16488 "2316","楠梓電","4,882,748","2,835","169,566,661","33.35","35.50","33.30","35.25","
16489 "2317","鴻海","28,513,381","13,485","2,491,571,248","87.30","87.70","87.00","87.10"
16490 "2321","東訊","1,046,708","89","1,849,032","1.77","1.77","1.74","1.77","+","0.16","
16491 "2323","中環","14,974,079","2,513","136,654,856","8.98","9.21","8.98","9.10","+","0
16492 "2324","仁寶","5,038,275","1,611","94,592,477","18.65","18.85","18.65","18.80","+",
16493 "2327","國巨","14,140,532","10,879","5,608,480,296","397.00","402.00","390.00","392
16494 "2328","廣宇","959,585","554","21,627,382","22.40","22.70","22.35","22.55","+","0.4
16495 "2329","華泰","1,503,642","579","22,173,209","14.75","14.90","14.65","14.75"," ","0
16496 "2330","台積電","32,397,504","16,793","10,890,847,940","335.00","337.50","333.50","
16497 "2331","精英","426,309","146","5,714,972","13.40","13.55","13.30","13.40","+","0.05
16498 "2332","友訊","4,803,733","1,459","65,067,368","13.40","13.65","13.40","13.65","+",
16499 "2337","旺宏","67,056,500","22,474","2,589,223,297","39.10","39.20","38.25","38.35"
16500 "2338","光罩","4,981,941","2,692","159,821,240","31.50","32.60","31.35","32.25","+"
```

```
X?response=csv&type=ALL&date=20200106
X?response=csv&type=ALL&date=20200107
X?response=csv&type=ALL&date=20200108
X?response=csv&type=ALL&date=20200109
X?response=csv&type=ALL&date=20200110
X?response=csv&type=ALL&date=20200113
X?response=csv&type=ALL&date=20200114
EX?response=csv&type=ALL&date=20200115
EX?response=csv&type=ALL&date=20200116
EX?response=csv&type=ALL&date=20200117
```

```python
print("擷取完畢：")
```

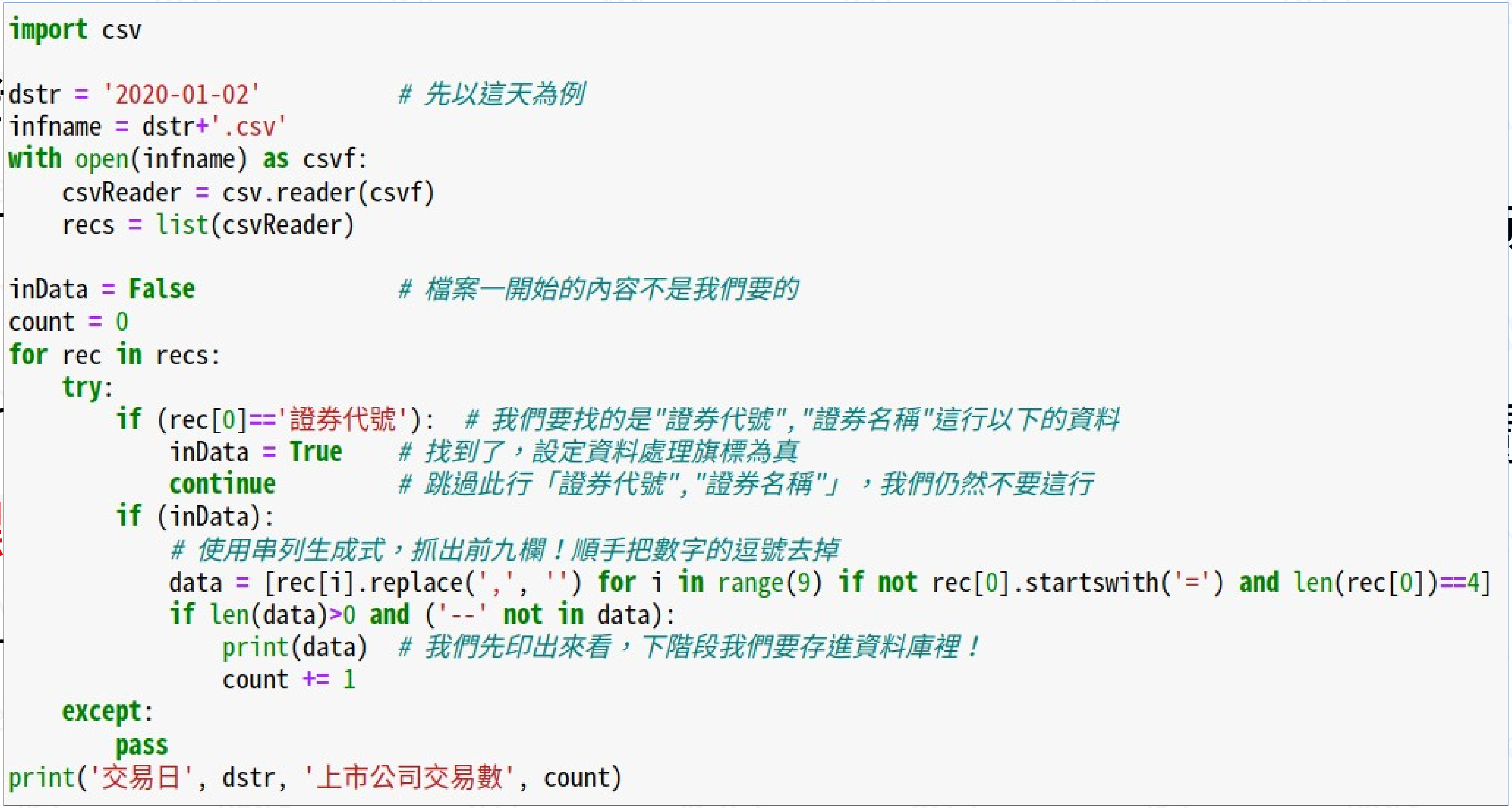
Part 2. 解析 CSV

- 如何對應?
  - "證券代號","證券名稱","成交股數","成交筆數","成交金額","開盤價","最高價","最低價","收盤價","漲跌(+/-)","漲跌價差","最後揭示買價","最後揭示買量","最後揭示賣價","最後揭示賣量","本益比",
  - "2330","台積電","47,246,695","109,202","27,694,691,641","590.00", "594.00","583.00","583.00","-","20.00","583.00","1,054","584.00", "440","27.96",
  - 目標先設定前九欄，到收盤價為止

資料解析框架

- 針對 CSV 格式檔案進行後處理
  - 找到「"證券代號","證券名稱"」，以下各行開始解析成 Python 資料結構
- 抓取前九欄資料，前 2 欄文字、中 3 欄整數、後 3 欄浮點數
  - 我稱它們為 id, name, vol, tra, val, op, hi, lo, cl

資料解析框架



```
import csv

dstr = '2020-01-02'          # 先以這天為例
infname = dstr+'.csv'
with open(infname) as csvf:
    csvReader = csv.reader(csvf)
    recs = list(csvReader)

inData = False               # 檔案一開始的內容不是我們要的
count = 0
for rec in recs:
    try:
        if (rec[0]=='證券代號'):  # 我們要找的是"證券代號","證券名稱"這行以下的資料
            inData = True     # 找到了，設定資料處理旗標為真
            continue          # 跳過此行「證券代號","證券名稱"」，我們仍然不要這行
        if (inData):
            # 使用串列生成式，抓出前九欄！順手把數字的逗號去掉
            data = [rec[i].replace(',', '') for i in range(9) if not rec[0].startswith('=') and len(rec[0])==4]
            if len(data)>0 and ('--' not in data):
                print(data)   # 我們先印出來看，下階段我們要存進資料庫裡！
                count += 1
    except:
        pass
print('交易日', dstr, '上市公司交易數', count)
```

- 金

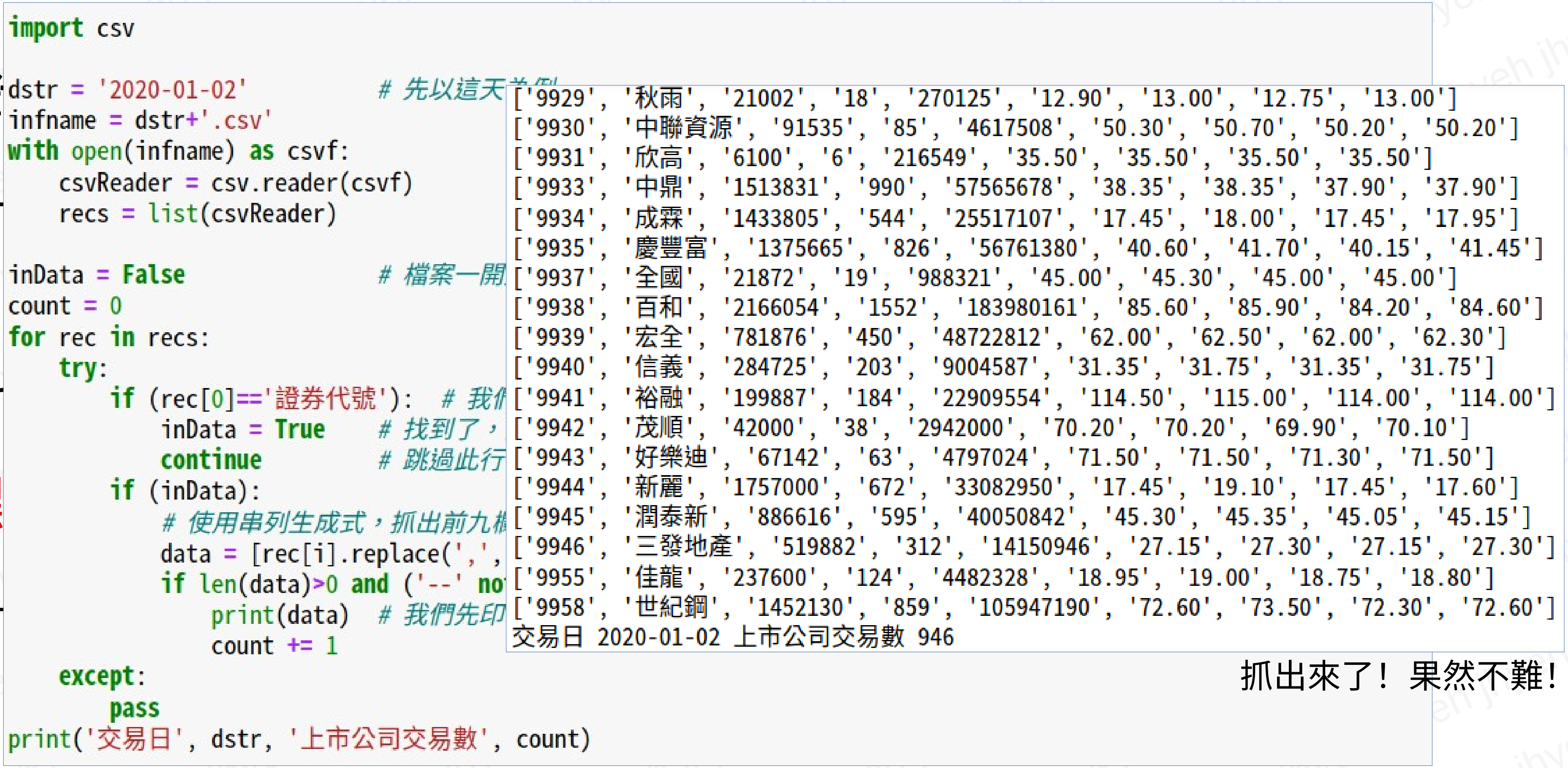
  – 成

- 扌 閘浮

  點

  –

資料解析框架



```
import csv

dstr = '2020-01-02'          # 先以這天為例
infname = dstr+'.csv'
with open(infname) as csvf:
    csvReader = csv.reader(csvf)
    recs = list(csvReader)

inData = False               # 檔案一開
count = 0
for rec in recs:
    try:
        if (rec[0]=='證券代號'):  # 我們
            inData = True     # 找到了，
            continue          # 跳過此行
        if (inData):
            # 使用串列生成式，抓出前九欄
            data = [rec[i].replace(',',
            if len(data)>0 and ('--' not
                print(data)   # 我們先印
                count += 1
    except:
        pass
print('交易日', dstr, '上市公司交易數', count)
```

```
['9929', '秋雨', '21002', '18', '270125', '12.90', '13.00', '12.75', '13.00']
['9930', '中聯資源', '91535', '85', '4617508', '50.30', '50.70', '50.20', '50.20']
['9931', '欣高', '6100', '6', '216549', '35.50', '35.50', '35.50', '35.50']
['9933', '中鼎', '1513831', '990', '57565678', '38.35', '38.35', '37.90', '37.90']
['9934', '成霖', '1433805', '544', '25517107', '17.45', '18.00', '17.45', '17.95']
['9935', '慶豐富', '1375665', '826', '56761380', '40.60', '41.70', '40.15', '41.45']
['9937', '全國', '21872', '19', '988321', '45.00', '45.30', '45.00', '45.00']
['9938', '百和', '2166054', '1552', '183980161', '85.60', '85.90', '84.20', '84.60']
['9939', '宏全', '781876', '450', '48722812', '62.00', '62.50', '62.00', '62.30']
['9940', '信義', '284725', '203', '9004587', '31.35', '31.75', '31.35', '31.75']
['9941', '裕融', '199887', '184', '22909554', '114.50', '115.00', '114.00', '114.00']
['9942', '茂順', '42000', '38', '2942000', '70.20', '70.20', '69.90', '70.10']
['9943', '好樂迪', '67142', '63', '4797024', '71.50', '71.50', '71.30', '71.50']
['9944', '新麗', '1757000', '672', '33082950', '17.45', '19.10', '17.45', '17.60']
['9945', '潤泰新', '886616', '595', '40050842', '45.30', '45.35', '45.05', '45.15']
['9946', '三發地產', '519882', '312', '14150946', '27.15', '27.30', '27.15', '27.30']
['9955', '佳龍', '237600', '124', '4482328', '18.95', '19.00', '18.75', '18.80']
['9958', '世紀鋼', '1452130', '859', '105947190', '72.60', '73.50', '72.30', '72.60']
交易日 2020-01-02 上市公司交易數 946
```

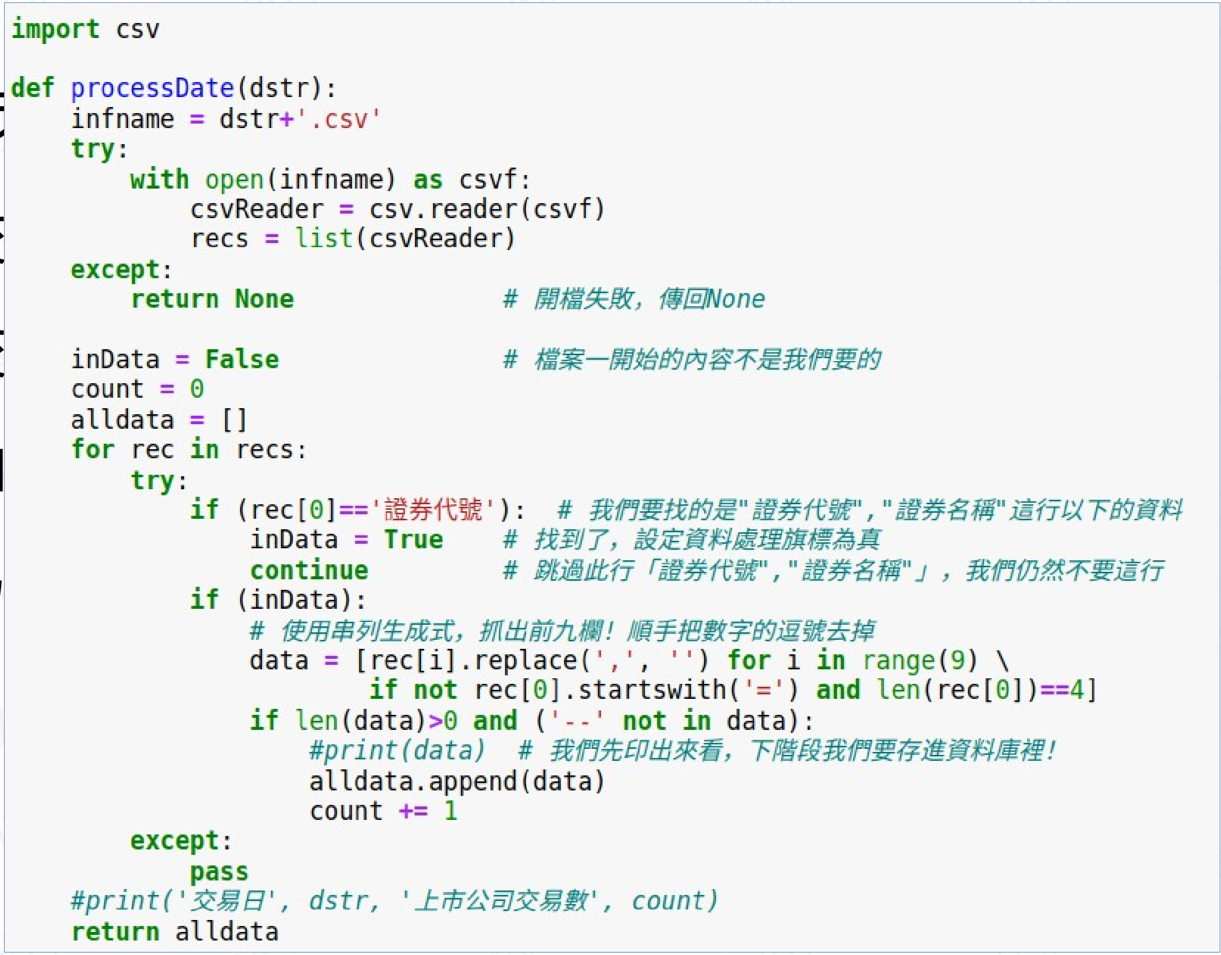
抓出來了！ 果然不難！

資料解析框架

- 再進一步!
  - 解析交易日內所有個股資料，每一個股存成串列
  - 同一交易日形成大的串列
  - 寫成函數，傳入交易日進行處理，回傳大的串列
  - 將交易日字串作為 key，回傳值作為 value，存入 dict
- 怎麼做?

資料解析框架

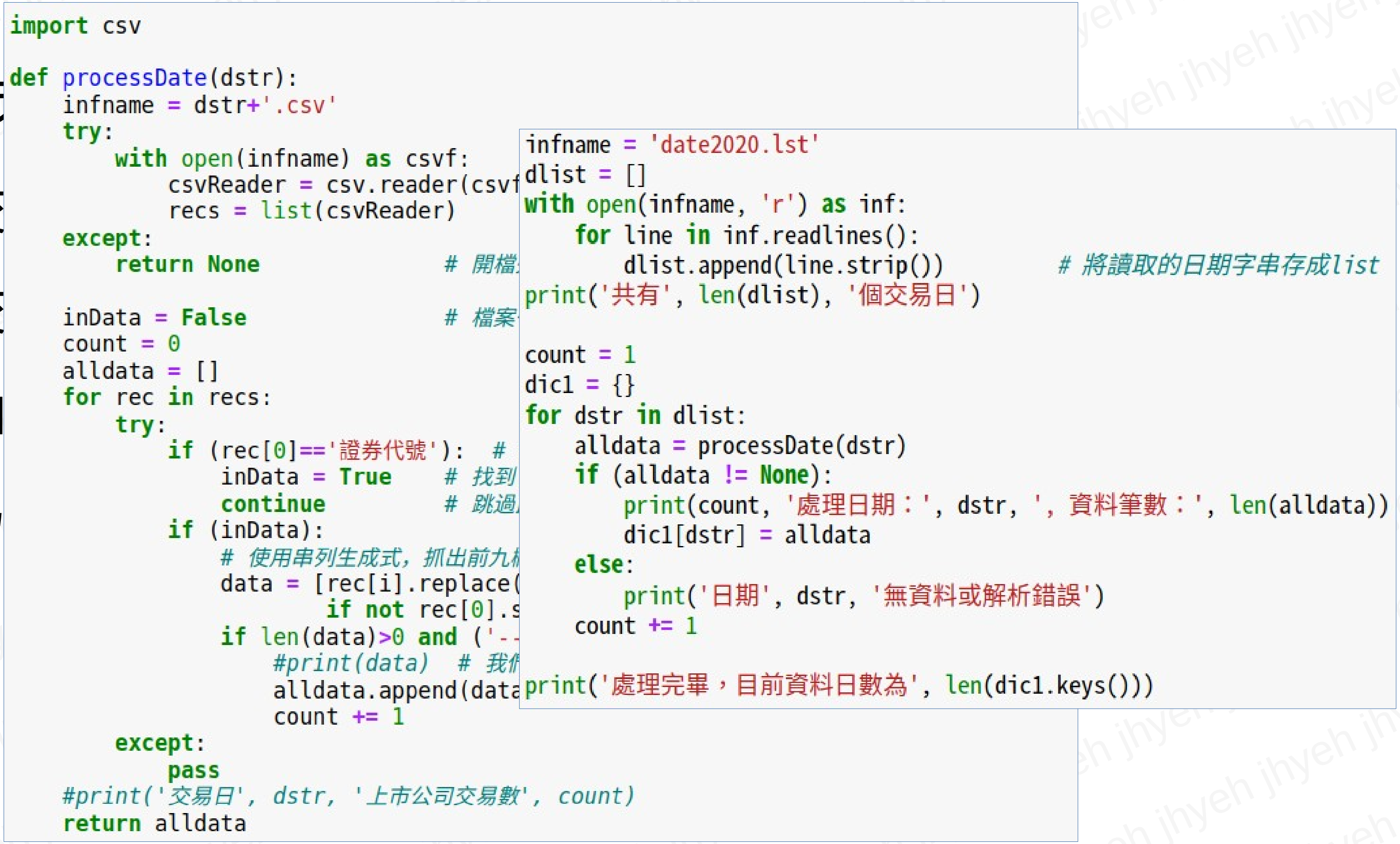
- 再進一步
  - 解析交
  - 同一交
  - 寫成函
  - 將交易 ct
- 怎麼做?



```python
import csv

def processDate(dstr):
    infname = dstr+'.csv'
    try:
        with open(infname) as csvf:
            csvReader = csv.reader(csvf)
            recs = list(csvReader)
    except:
        return None              # 開檔失敗, 傳回None

    inData = False               # 檔案一開始的內容不是我們要的
    count = 0
    alldata = []
    for rec in recs:
        try:
            if (rec[0]=='證券代號'):   # 我們要找的是"證券代號","證券名稱"這行以下的資料
                inData = True    # 找到了, 設定資料處理旗標為真
                continue         # 跳過此行「證券代號","證券名稱"」, 我們仍然不要這行
            if (inData):
                # 使用串列生成式, 抓出前九欄! 順手把數字的逗號去掉
                data = [rec[i].replace(',', '') for i in range(9) \
                        if not rec[0].startswith('=') and len(rec[0])==4]
                if len(data)>0 and ('--' not in data):
                    #print(data)   # 我們先印出來看, 下階段我們要存進資料庫裡!
                    alldata.append(data)
                    count += 1
        except:
            pass
    #print('交易日', dstr, '上市公司交易數', count)
    return alldata
```

資料解析框架

- 再進一步
  - 解析交
  - 同一交
  - 寫成函
  - 將交易
- 怎麼做?

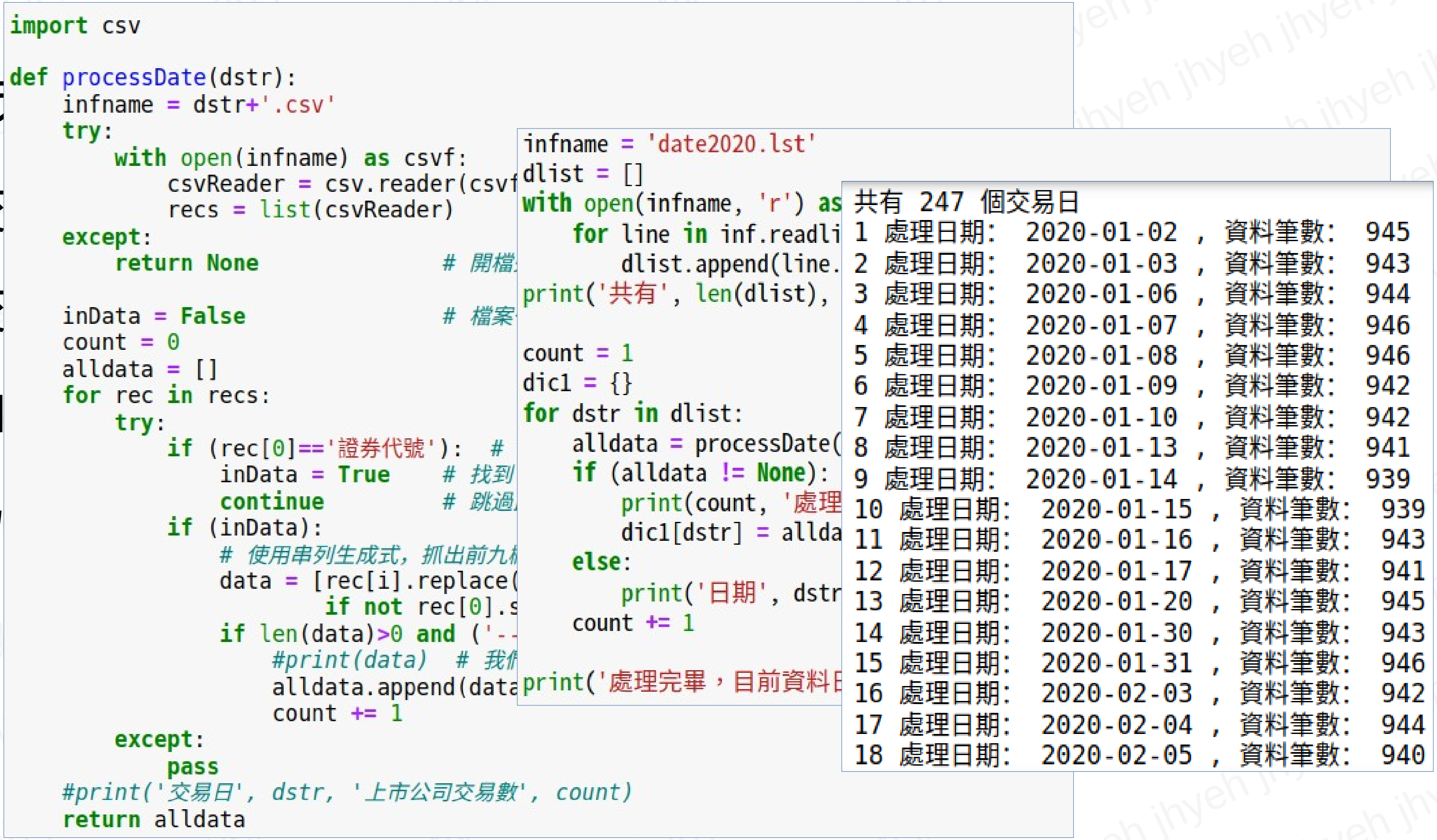


```
import csv

def processDate(dstr):
    infname = dstr+'.csv'
    try:
        with open(infname) as csvf:
            csvReader = csv.reader(csvf
            recs = list(csvReader)
    except:
        return None                 # 開檔

    inData = False                  # 檔案
    count = 0
    alldata = []
    for rec in recs:
        try:
            if (rec[0]=='證券代號'):  #
                inData = True     # 找到
                continue          # 跳過
            if (inData):
                # 使用串列生成式，抓出前九
                data = [rec[i].replace(
                        if not rec[0].s
                if len(data)>0 and ('--
                    #print(data)   # 我
                    alldata.append(data
                    count += 1
        except:
            pass
    #print('交易日', dstr, '上市公司交易數', count)
    return alldata
```

```
infname = 'date2020.lst'
dlist = []
with open(infname, 'r') as inf:
    for line in inf.readlines():
        dlist.append(line.strip())          # 將讀取的日期字串存成list
print('共有', len(dlist), '個交易日')

count = 1
dic1 = {}
for dstr in dlist:
    alldata = processDate(dstr)
    if (alldata != None):
        print(count, '處理日期：', dstr, ', 資料筆數：', len(alldata))
        dic1[dstr] = alldata
    else:
        print('日期', dstr, '無資料或解析錯誤')
    count += 1

print('處理完畢，目前資料日數為', len(dic1.keys()))
```

資料解析框架

- 再進一步
  - 解析交
  - 同一交
  - 寫成函
  - 將交易
- 怎麼做?



```
import csv

def processDate(dstr):
    infname = dstr+'.csv'
    try:
        with open(infname) as csvf:
            csvReader = csv.reader(csvf
            recs = list(csvReader)
    except:
        return None                    # 開檔

    inData = False                     # 檔案
    count = 0
    alldata = []
    for rec in recs:
        try:
            if (rec[0]=='證券代號'):  #
                inData = True      # 找到
                continue           # 跳過
            if (inData):
                # 使用串列生成式, 抓出前九
                data = [rec[i].replace(
                        if not rec[0].s
                if len(data)>0 and ('--
                    #print(data)   # 我們
                    alldata.append(data
                    count += 1
        except:
            pass
    #print('交易日', dstr, '上市公司交易數', count)
    return alldata
```

```
infname = 'date2020.lst'
dlist = []
with open(infname, 'r') as
    for line in inf.readli
        dlist.append(line.
print('共有', len(dlist),

count = 1
dic1 = {}
for dstr in dlist:
    alldata = processDate(
    if (alldata != None):
        print(count, '處理
        dic1[dstr] = allda
    else:
        print('日期', dstr
    count += 1

print('處理完畢，目前資料日
```

```
共有 247 個交易日
1 處理日期: 2020-01-02 , 資料筆數: 945
2 處理日期: 2020-01-03 , 資料筆數: 943
3 處理日期: 2020-01-06 , 資料筆數: 944
4 處理日期: 2020-01-07 , 資料筆數: 946
5 處理日期: 2020-01-08 , 資料筆數: 946
6 處理日期: 2020-01-09 , 資料筆數: 942
7 處理日期: 2020-01-10 , 資料筆數: 942
8 處理日期: 2020-01-13 , 資料筆數: 941
9 處理日期: 2020-01-14 , 資料筆數: 939
10 處理日期: 2020-01-15 , 資料筆數: 939
11 處理日期: 2020-01-16 , 資料筆數: 943
12 處理日期: 2020-01-17 , 資料筆數: 941
13 處理日期: 2020-01-20 , 資料筆數: 945
14 處理日期: 2020-01-30 , 資料筆數: 943
15 處理日期: 2020-01-31 , 資料筆數: 946
16 處理日期: 2020-02-03 , 資料筆數: 942
17 處理日期: 2020-02-04 , 資料筆數: 944
18 處理日期: 2020-02-05 , 資料筆數: 940
```

Part 3. 資料庫

- 規劃資料表，建立資料表 如何 CREATE TABLE?
- Python 資料結構 (dict) 對應資料表
  - 產出 SQL 字串 如何 INSERT INTO?
- 執行 SQL 字串，資料新增到資料表內

資料表處理框架

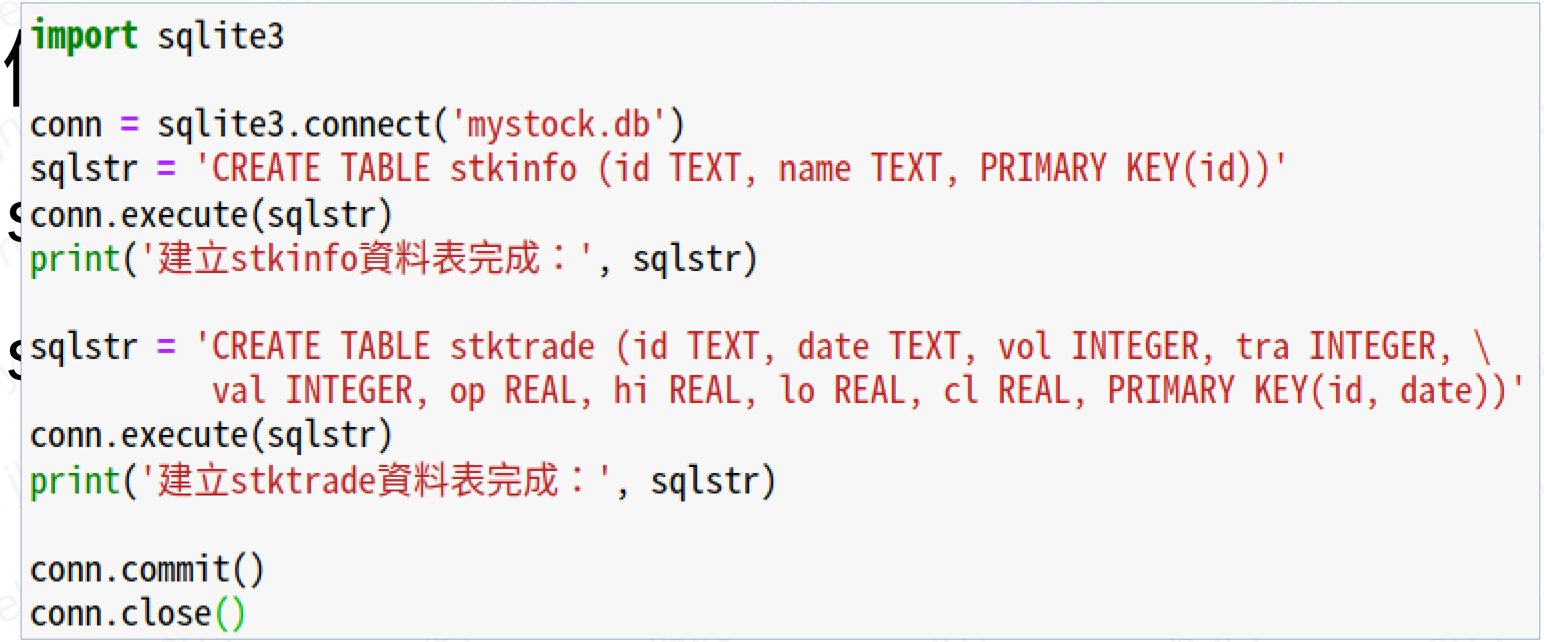
- 建立資料表
  - SQLite, CREATE TABLE...
- 將 Python 資料結構 dict 轉換成 SQL 字串
  - SQLite, INSERT INTO...
- 做基本的檢查，看看有多少筆資料進資料表了
  - SQLite, SELECT COUNT(*) FROM...

建立資料表

- 本例建立兩個資料表
  - stkinfo：存放上市公司名稱資料 (id, name)
  - stktrade：存放個股交易資料
    - id, date, vol, tra, val, op, hi, lo, cl
    - 名稱放在 stkinfo 作為對照，節省空間，避免不一致

建立資料表

- 本
  - s
  - s



```
import sqlite3

conn = sqlite3.connect('mystock.db')
sqlstr = 'CREATE TABLE stkinfo (id TEXT, name TEXT, PRIMARY KEY(id))'
conn.execute(sqlstr)
print('建立stkinfo資料表完成：', sqlstr)

sqlstr = 'CREATE TABLE stktrade (id TEXT, date TEXT, vol INTEGER, tra INTEGER, \
          val INTEGER, op REAL, hi REAL, lo REAL, cl REAL, PRIMARY KEY(id, date))'
conn.execute(sqlstr)
print('建立stktrade資料表完成：', sqlstr)

conn.commit()
conn.close()
```

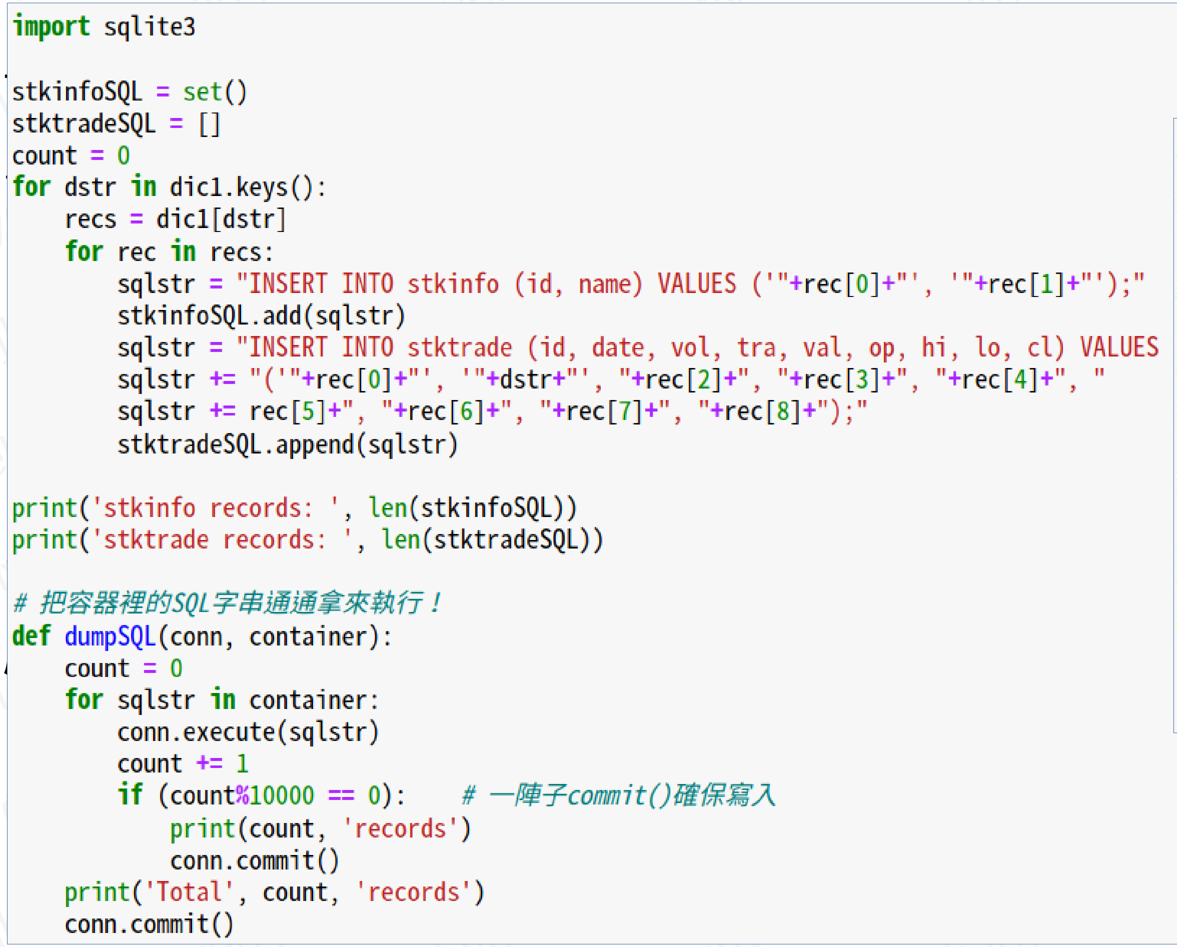
Dict 轉換成 SQL 存入

- 目前的 dict 結構
  - {日期：當日全市場交易資訊 list}
- 當日全市場交易資訊 list 結構
  - [[個股當日交易九欄資料],
    [個股當日交易九欄資料],
    [個股當日交易九欄資料], …]
  - “List of List”
- 個股當日交易九欄資料
  - id, date, vol, tra, val, op, hi, lo, cl
  - 全部是字串型別

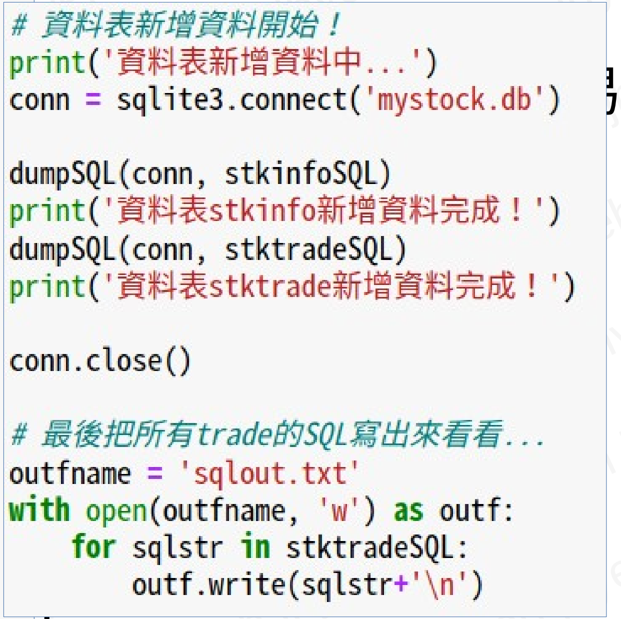
- 拆解結構
  - 用 keys 解出 dict 當日全市場交易資訊
  - 用 list iterator 拆解個股當日交易資訊
  - 轉成單一 SQL INSERT INTO 字串
  - 執行 SQL

Dict 轉換成 SQL 存入

- 目
  –
- 當
  –
  –
- 個
  –
  –



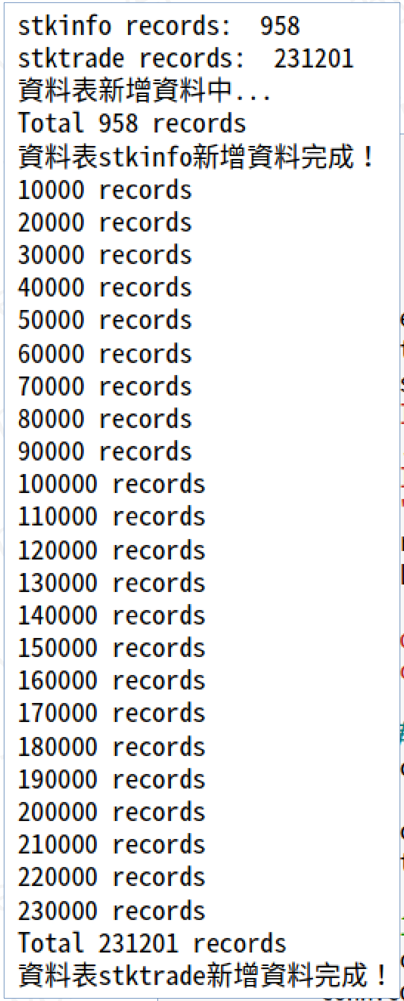
```python
import sqlite3

stkinfoSQL = set()
stktradeSQL = []
count = 0
for dstr in dic1.keys():
    recs = dic1[dstr]
    for rec in recs:
        sqlstr = "INSERT INTO stkinfo (id, name) VALUES ('"+rec[0]+"', '"+rec[1]+"');"
        stkinfoSQL.add(sqlstr)
        sqlstr = "INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES
        sqlstr += "('"+rec[0]+"', '"+dstr+"', "+rec[2]+", "+rec[3]+", "+rec[4]+", "
        sqlstr += rec[5]+", "+rec[6]+", "+rec[7]+", "+rec[8]+");"
        stktradeSQL.append(sqlstr)

print('stkinfo records: ', len(stkinfoSQL))
print('stktrade records: ', len(stktradeSQL))

# 把容器裡的SQL字串通通拿來執行！
def dumpSQL(conn, container):
    count = 0
    for sqlstr in container:
        conn.execute(sqlstr)
        count += 1
        if (count%10000 == 0):    # 一陣子commit()確保寫入
            print(count, 'records')
            conn.commit()
    print('Total', count, 'records')
    conn.commit()
```



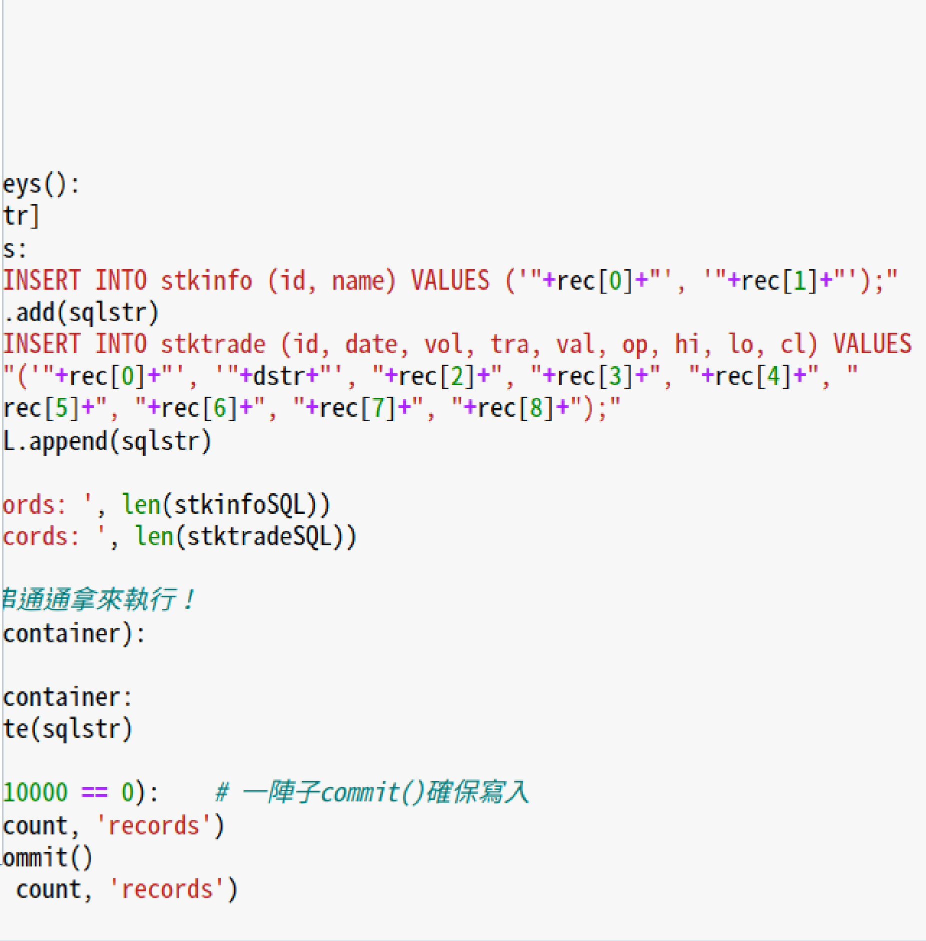
```python
# 資料表新增資料開始！
print('資料表新增資料中...')
conn = sqlite3.connect('mystock.db')

dumpSQL(conn, stkinfoSQL)
print('資料表stkinfo新增資料完成！')
dumpSQL(conn, stktradeSQL)
print('資料表stktrade新增資料完成！')

conn.close()

# 最後把所有trade的SQL寫出來看看...
outfname = 'sqlout.txt'
with open(outfname, 'w') as outf:
    for sqlstr in stktradeSQL:
        outf.write(sqlstr+'\n')
```

L

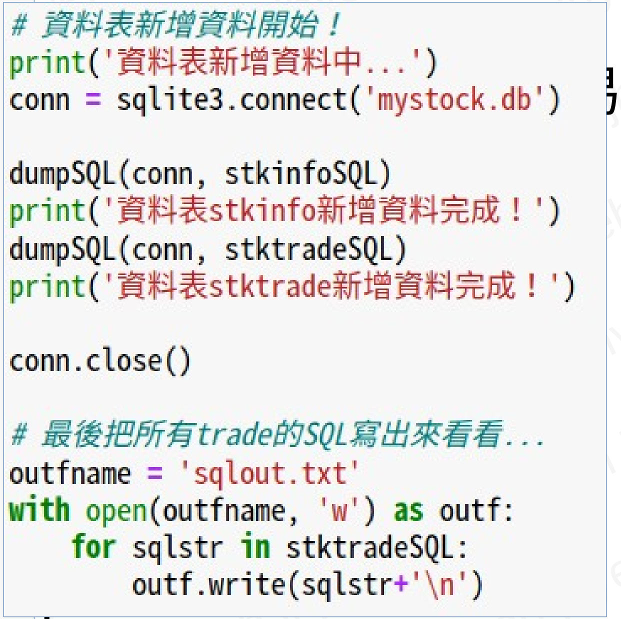
Dict 轉換成 SQL 存入



```
stkinfo records:  958
stktrade records:  231201
資料表新增資料中...
Total 958 records
資料表stkinfo新增資料完成！
10000 records
20000 records
30000 records
40000 records
50000 records
60000 records
70000 records
80000 records
90000 records
100000 records
110000 records
120000 records
130000 records
140000 records
150000 records
160000 records
170000 records
180000 records
190000 records
200000 records
210000 records
220000 records
230000 records
Total 231201 records
資料表stktrade新增資料完成！
```

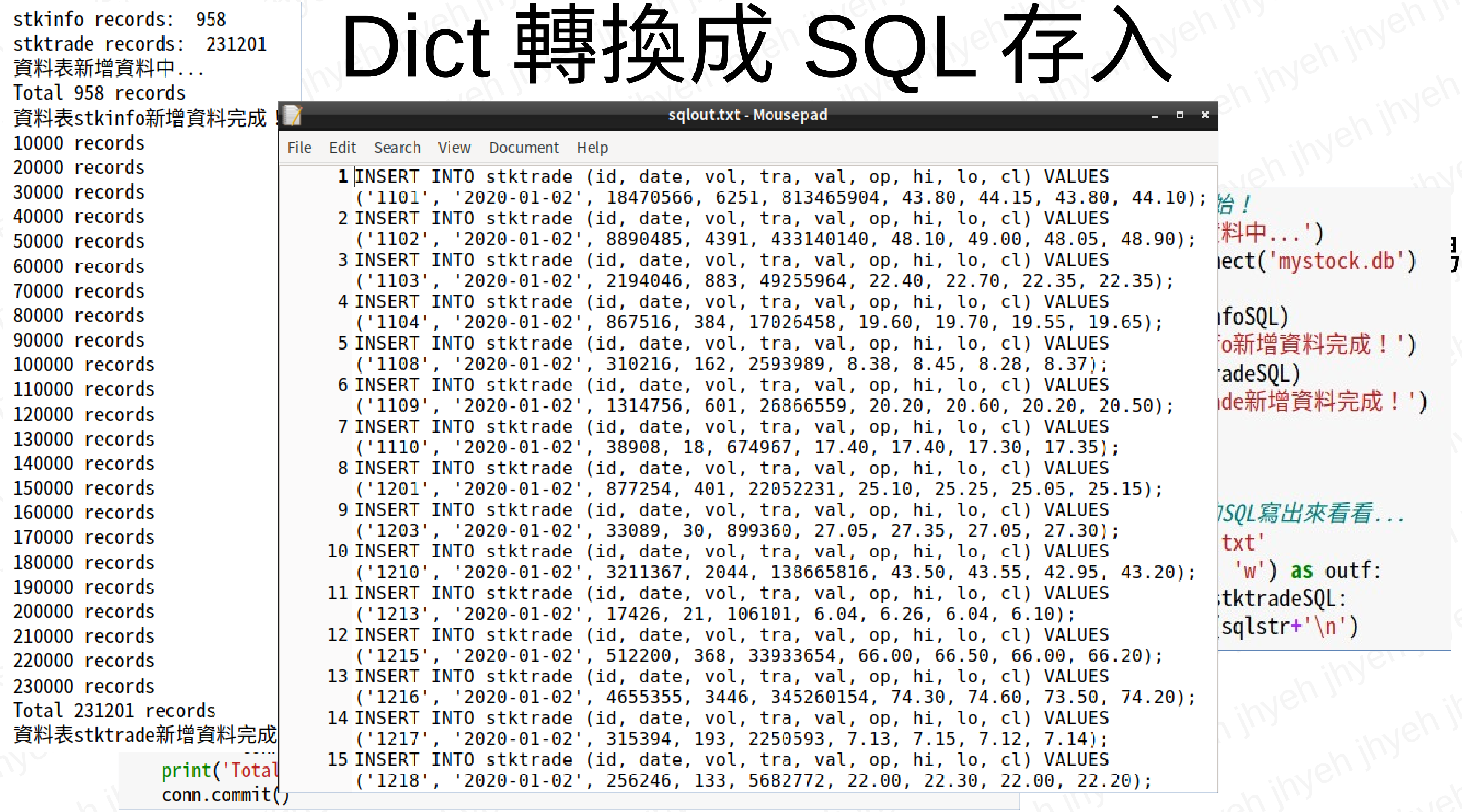


```
eys():
tr]
s:
INSERT INTO stkinfo (id, name) VALUES ('"+rec[0]+"', '"+rec[1]+"');"
.add(sqlstr)
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES
"('"+rec[0]+"', '"+dstr+"', "+rec[2]+", "+rec[3]+", "+rec[4]+", "
rec[5]+", "+rec[6]+", "+rec[7]+", "+rec[8]+");"
L.append(sqlstr)

ords: ', len(stkinfoSQL))
cords: ', len(stktradeSQL))

串通通拿來執行！
container):

container:
te(sqlstr)

10000 == 0):    # 一陣子commit()確保寫入
count, 'records')
ommit()
    print('Total', count, 'records')
    conn.commit()
```



```python
# 資料表新增資料開始！
print('資料表新增資料中...')
conn = sqlite3.connect('mystock.db')

dumpSQL(conn, stkinfoSQL)
print('資料表stkinfo新增資料完成！')
dumpSQL(conn, stktradeSQL)
print('資料表stktrade新增資料完成！')

conn.close()

# 最後把所有trade的SQL寫出來看看...
outfname = 'sqlout.txt'
with open(outfname, 'w') as outf:
    for sqlstr in stktradeSQL:
        outf.write(sqlstr+'\n')
```

Dict 轉換成 SQL 存入



```
stkinfo records:  958
stktrade records:  231201
資料表新增資料中...
Total 958 records
資料表stkinfo新增資料完成！
10000 records
20000 records
30000 records
40000 records
50000 records
60000 records
70000 records
80000 records
90000 records
100000 records
110000 records
120000 records
130000 records
140000 records
150000 records
160000 records
170000 records
180000 records
190000 records
200000 records
210000 records
220000 records
230000 records
Total 231201 records
資料表stktrade新增資料完成
```

```
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1101', '2020-01-02', 18470566, 6251, 813465904, 43.80, 44.15, 43.80, 44.10);
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1102', '2020-01-02', 8890485, 4391, 433140140, 48.10, 49.00, 48.05, 48.90);
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1103', '2020-01-02', 2194046, 883, 49255964, 22.40, 22.70, 22.35, 22.35);
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1104', '2020-01-02', 867516, 384, 17026458, 19.60, 19.70, 19.55, 19.65);
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1108', '2020-01-02', 310216, 162, 2593989, 8.38, 8.45, 8.28, 8.37);
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1109', '2020-01-02', 1314756, 601, 26866559, 20.20, 20.60, 20.20, 20.50);
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1110', '2020-01-02', 38908, 18, 674967, 17.40, 17.40, 17.30, 17.35);
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1201', '2020-01-02', 877254, 401, 22052231, 25.10, 25.25, 25.05, 25.15);
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1203', '2020-01-02', 33089, 30, 899360, 27.05, 27.35, 27.05, 27.30);
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1210', '2020-01-02', 3211367, 2044, 138665816, 43.50, 43.55, 42.95, 43.20);
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1213', '2020-01-02', 17426, 21, 106101, 6.04, 6.26, 6.04, 6.10);
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1215', '2020-01-02', 512200, 368, 33933654, 66.00, 66.50, 66.00, 66.20);
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1216', '2020-01-02', 4655355, 3446, 345260154, 74.30, 74.60, 73.50, 74.20);
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1217', '2020-01-02', 315394, 193, 2250593, 7.13, 7.15, 7.12, 7.14);
INSERT INTO stktrade (id, date, vol, tra, val, op, hi, lo, cl) VALUES ('1218', '2020-01-02', 256246, 133, 5682772, 22.00, 22.30, 22.00, 22.20);
```

```
始！
料中...')
ect('mystock.db')

foSQL)
o新增資料完成！')
adeSQL)
de新增資料完成！')



SQL寫出來看看...
txt'
 'w') as outf:
tktradeSQL:
sqlstr+'\n')
```

```
print('Total
conn.commit()
```

講次內容

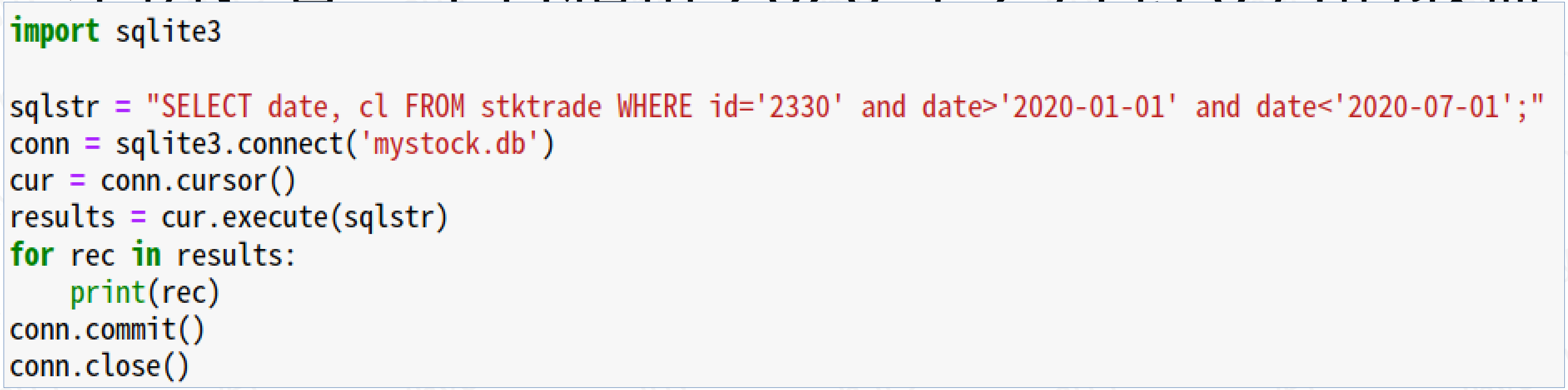
- TWSE 每日收盤資訊介紹
- 網路爬蟲與資料庫的規劃與操作
- SQL 收盤價查詢
- 專題：上市公司收盤價走勢

SQL 收盤價查詢

- 舉例，查一下台積電 2020 年 2 月到 6 月的收盤價為何?
- 怎麼做? 如何 SELECT-FROM-WHERE?

SQL 收盤價查詢

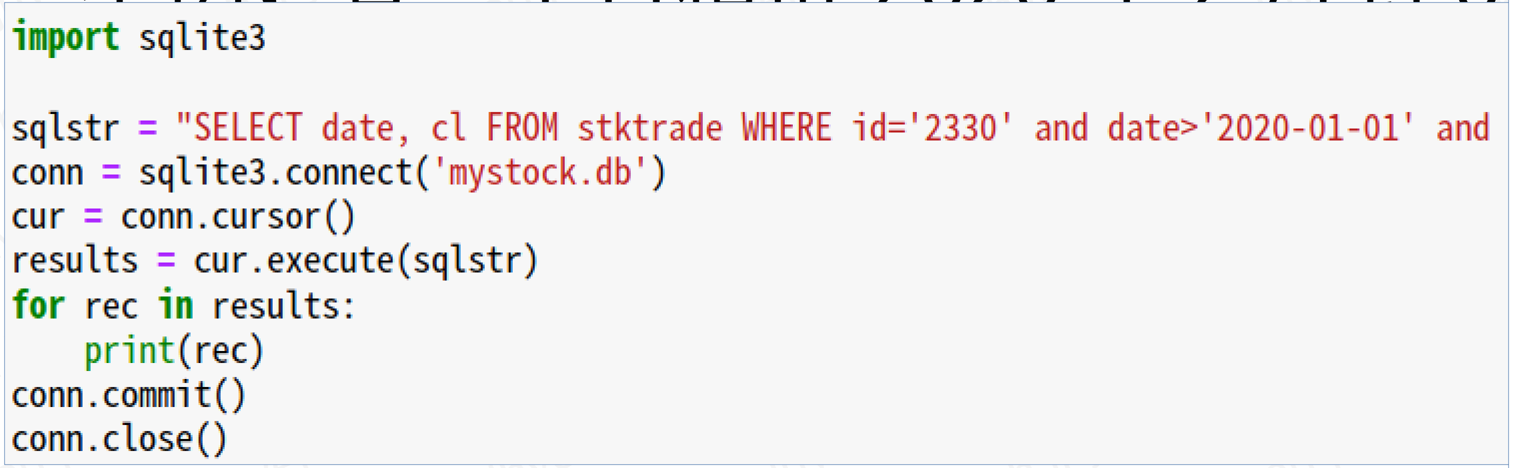
- 舉例，查一下台積電 2020 年 2 月到 6 月的收盤



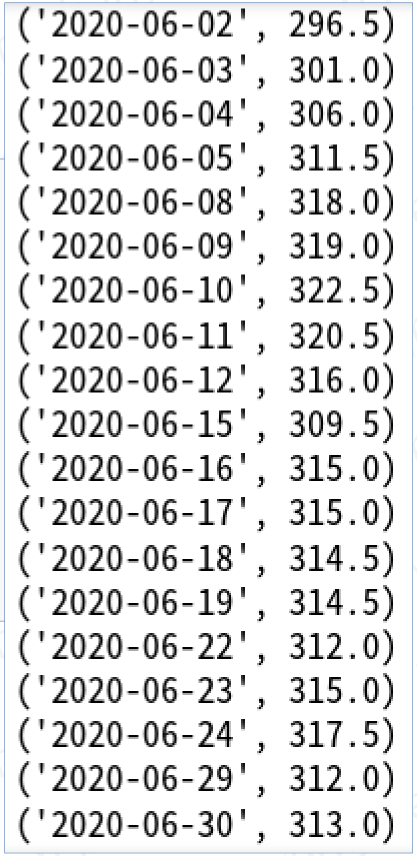
```
import sqlite3

sqlstr = "SELECT date, cl FROM stktrade WHERE id='2330' and date>'2020-01-01' and date<'2020-07-01';"
conn = sqlite3.connect('mystock.db')
cur = conn.cursor()
results = cur.execute(sqlstr)
for rec in results:
    print(rec)
conn.commit()
conn.close()
```

SQL 收盤價查詢

- 舉例：查一下台積電 2020 年 2 月到 6



```python
import sqlite3

sqlstr = "SELECT date, cl FROM stktrade WHERE id='2330' and date>'2020-01-01' and
conn = sqlite3.connect('mystock.db')
cur = conn.cursor()
results = cur.execute(sqlstr)
for rec in results:
    print(rec)
conn.commit()
conn.close()
```



```
('2020-06-02', 296.5)
('2020-06-03', 301.0)
('2020-06-04', 306.0)
('2020-06-05', 311.5)
('2020-06-08', 318.0)
('2020-06-09', 319.0)
('2020-06-10', 322.5)
('2020-06-11', 320.5)
('2020-06-12', 316.0)
('2020-06-15', 309.5)
('2020-06-16', 315.0)
('2020-06-17', 315.0)
('2020-06-18', 314.5)
('2020-06-19', 314.5)
('2020-06-22', 312.0)
('2020-06-23', 315.0)
('2020-06-24', 317.5)
('2020-06-29', 312.0)
('2020-06-30', 313.0)
```

講次內容

- TWSE 每日收盤資訊介紹
- 網路爬蟲與資料庫的規劃與操作
- SQL 收盤價查詢
- 專題：上市公司收盤價走勢

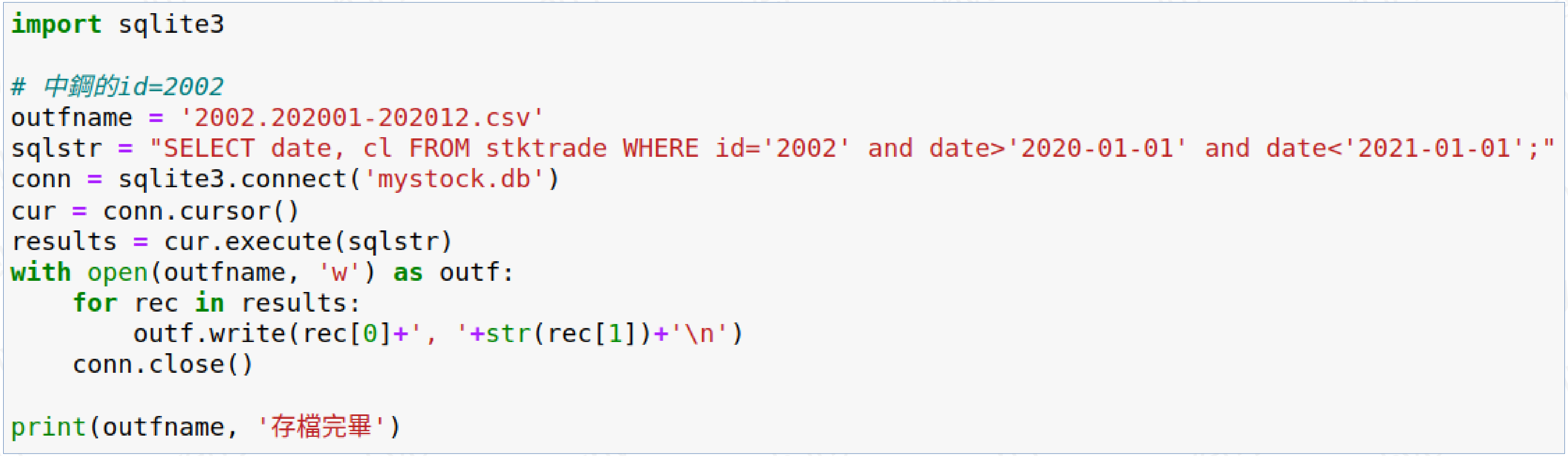
專題：上市公司收盤價走勢

- 請製作中鋼公司
  - 在 2020 年全年的收盤價格走勢
- 請製作台泥公司
  - 在 2020 年上半年的成交量走勢
- 使用試算表作為輔助
  - 你要有基本的試算表操作基礎哦!

Google： " 試算表教學 "，趕快看一看!

中鋼 (2002)

- 中鋼公司 2020 年全年的收盤價格走勢

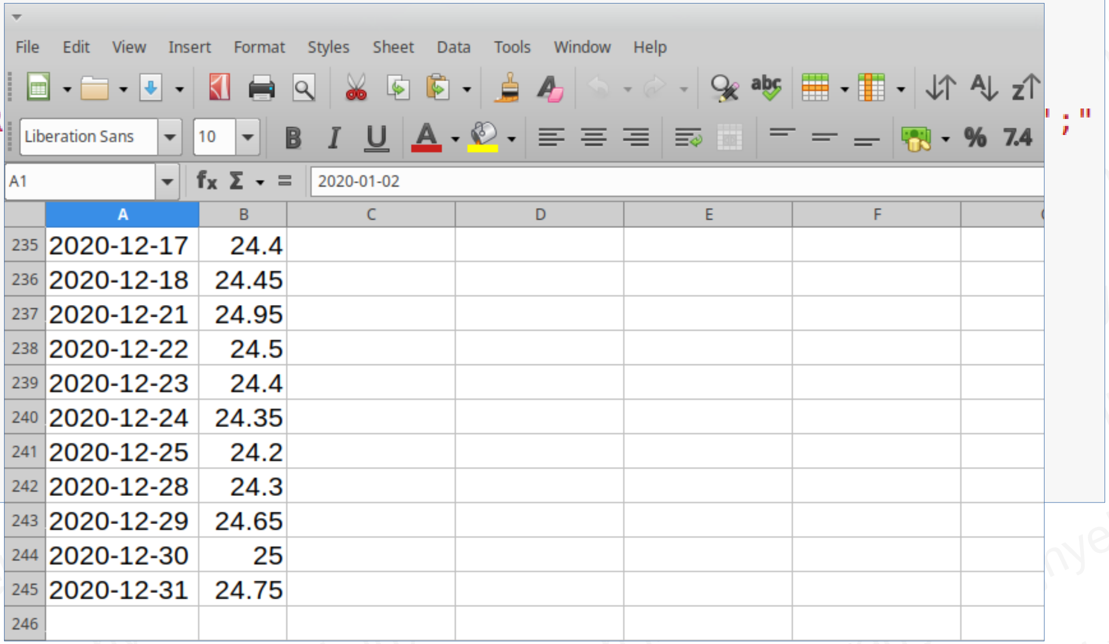


```
import sqlite3

# 中鋼的id=2002
outfname = '2002.202001-202012.csv'
sqlstr = "SELECT date, cl FROM stktrade WHERE id='2002' and date>'2020-01-01' and date<'2021-01-01';"
conn = sqlite3.connect('mystock.db')
cur = conn.cursor()
results = cur.execute(sqlstr)
with open(outfname, 'w') as outf:
    for rec in results:
        outf.write(rec[0]+', '+str(rec[1])+'\n')
    conn.close()

print(outfname, '存檔完畢')
```

中鋼 (2002)

- 中鋼公司 2020 年全年的收盤價格走勢

```
import sqlite3

# 中鋼的id=2002
outfname = '2002.202001-202012.csv'
sqlstr = "SELECT date, cl FROM stktrade WHER ';"
conn = sqlite3.connect('mystock.db')
cur = conn.cursor()
results = cur.execute(sqlstr)
with open(outfname, 'w') as outf:
    for rec in results:
        outf.write(rec[0]+', '+str(rec[1])+'
    conn.close()

print(outfname, '存檔完畢')
```

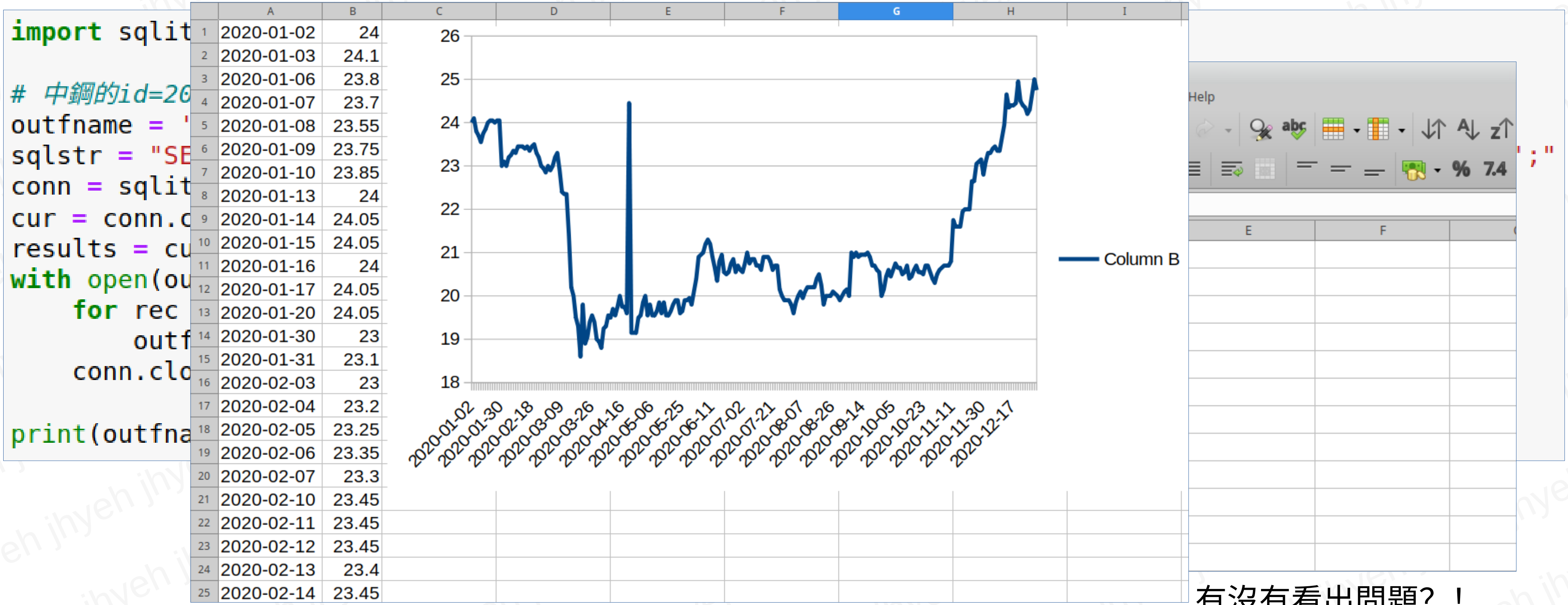


2020-01-02

	A	B	C	D	E	F
235	2020-12-17	24.4				
236	2020-12-18	24.45				
237	2020-12-21	24.95				
238	2020-12-22	24.5				
239	2020-12-23	24.4				
240	2020-12-24	24.35				
241	2020-12-25	24.2				
242	2020-12-28	24.3				
243	2020-12-29	24.65				
244	2020-12-30	25				
245	2020-12-31	24.75				
246						

中鋼 (2002)

- 中鋼公司 2020 年全年的收盤價格走勢



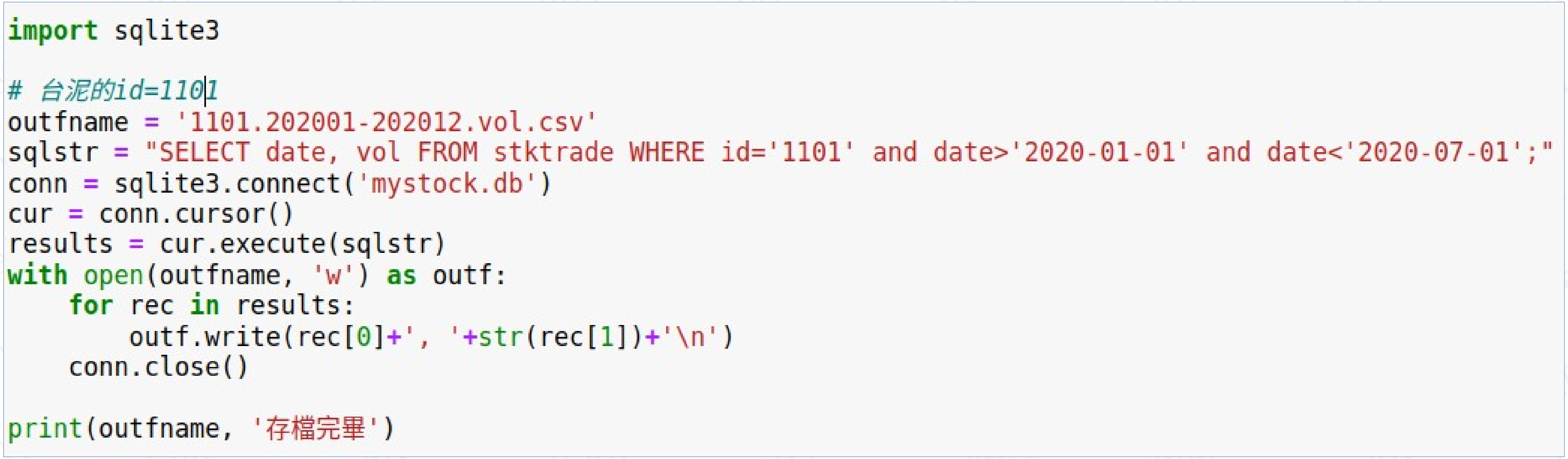
```
import sqlit

# 中鋼的id=20
outfname = '
sqlstr = "SE
conn = sqlit
cur = conn.c
results = cu
with open(ou
    for rec
        outf
    conn.clo

print(outfna
```

	A	B
1	2020-01-02	24
2	2020-01-03	24.1
3	2020-01-06	23.8
4	2020-01-07	23.7
5	2020-01-08	23.55
6	2020-01-09	23.75
7	2020-01-10	23.85
8	2020-01-13	24
9	2020-01-14	24.05
10	2020-01-15	24.05
11	2020-01-16	24
12	2020-01-17	24.05
13	2020-01-20	24.05
14	2020-01-30	23
15	2020-01-31	23.1
16	2020-02-03	23
17	2020-02-04	23.2
18	2020-02-05	23.25
19	2020-02-06	23.35
20	2020-02-07	23.3
21	2020-02-10	23.45
22	2020-02-11	23.45
23	2020-02-12	23.45
24	2020-02-13	23.4
25	2020-02-14	23.45

有沒有看出問題? ！

台泥 (1101)

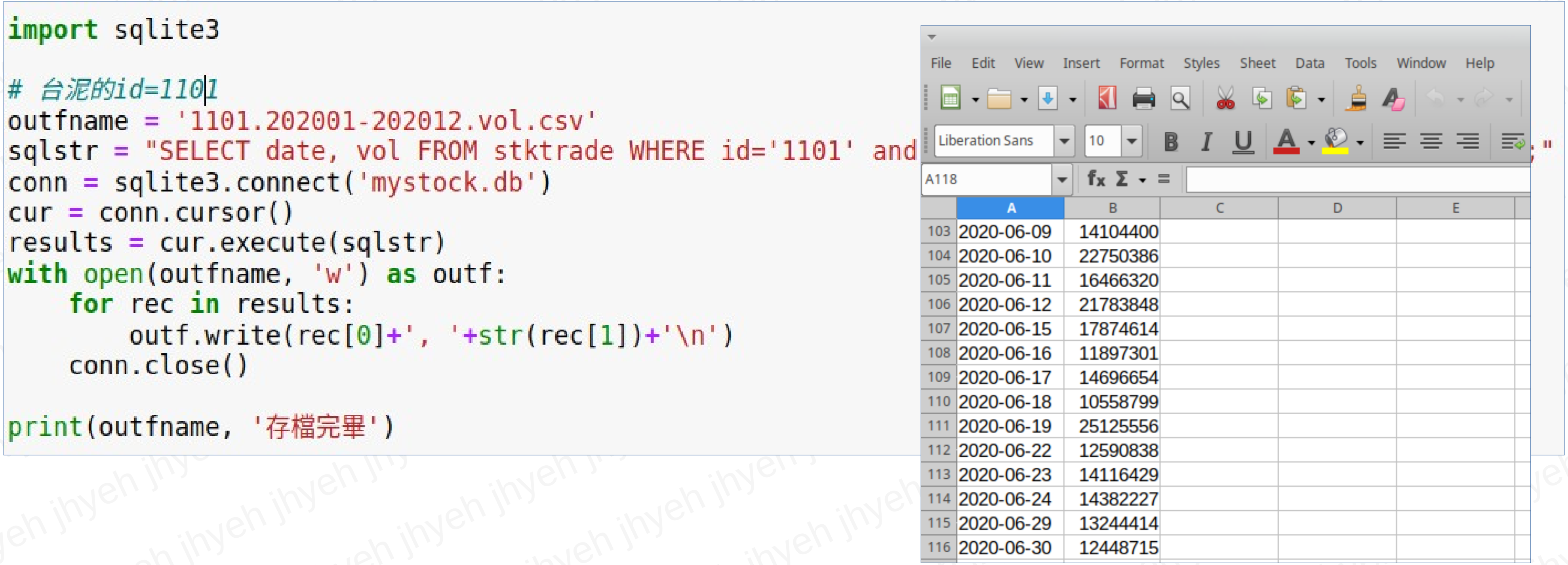
- 台泥公司 2020 年上半年的成交量走勢



```python
import sqlite3

# 台泥的id=1101
outfname = '1101.202001-202012.vol.csv'
sqlstr = "SELECT date, vol FROM stktrade WHERE id='1101' and date>'2020-01-01' and date<'2020-07-01';"
conn = sqlite3.connect('mystock.db')
cur = conn.cursor()
results = cur.execute(sqlstr)
with open(outfname, 'w') as outf:
    for rec in results:
        outf.write(rec[0]+', '+str(rec[1])+'\n')
    conn.close()

print(outfname, '存檔完畢')
```

台泥 (1101)

- 台泥公司 2020 年上半年的成交量走勢

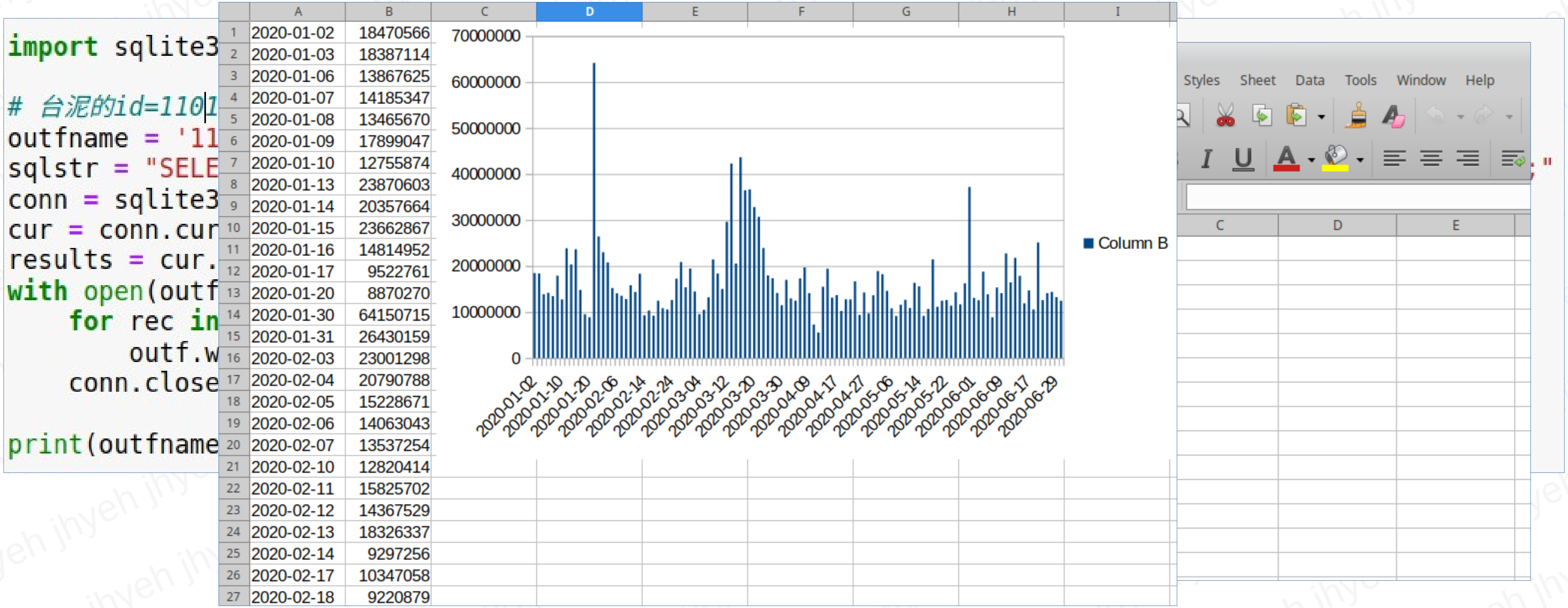


```
import sqlite3

# 台泥的id=1101
outfname = '1101.202001-202012.vol.csv'
sqlstr = "SELECT date, vol FROM stktrade WHERE id='1101' and                    ;"
conn = sqlite3.connect('mystock.db')
cur = conn.cursor()
results = cur.execute(sqlstr)
with open(outfname, 'w') as outf:
    for rec in results:
        outf.write(rec[0]+', '+str(rec[1])+'\n')
    conn.close()

print(outfname, '存檔完畢')
```

	A	B	C	D	E
103	2020-06-09	14104400			
104	2020-06-10	22750386			
105	2020-06-11	16466320			
106	2020-06-12	21783848			
107	2020-06-15	17874614			
108	2020-06-16	11897301			
109	2020-06-17	14696654			
110	2020-06-18	10558799			
111	2020-06-19	25125556			
112	2020-06-22	12590838			
113	2020-06-23	14116429			
114	2020-06-24	14382227			
115	2020-06-29	13244414			
116	2020-06-30	12448715			

台泥 (1101)

- 台泥公司 2020 年上半年的成交量走勢



```
import sqlite3

# 台泥的id=1101
outfname = '11
sqlstr = "SELE
conn = sqlite3
cur = conn.cur
results = cur.
with open(outf
    for rec in
        outf.w
    conn.close

print(outfname
```

	A	B
1	2020-01-02	18470566
2	2020-01-03	18387114
3	2020-01-06	13867625
4	2020-01-07	14185347
5	2020-01-08	13465670
6	2020-01-09	17899047
7	2020-01-10	12755874
8	2020-01-13	23870603
9	2020-01-14	20357664
10	2020-01-15	23662867
11	2020-01-16	14814952
12	2020-01-17	9522761
13	2020-01-20	8870270
14	2020-01-30	64150715
15	2020-01-31	26430159
16	2020-02-03	23001298
17	2020-02-04	20790788
18	2020-02-05	15228671
19	2020-02-06	14063043
20	2020-02-07	13537254
21	2020-02-10	12820414
22	2020-02-11	15825702
23	2020-02-12	14367529
24	2020-02-13	18326337
25	2020-02-14	9297256
26	2020-02-17	10347058
27	2020-02-18	9220879

這個講次中，你應該學到了 ...

- TWSE 每日收盤資訊介紹
- 網路爬蟲與資料庫的規劃與操作
- SQL 收盤價查詢
- 製作上市公司各類數據走勢