合併表格化的文字檔

- 利用別人已經發展成熟的程式庫來實現檔案合併
  - 參考Introduction to Python for Science and Engineering by David J. Pine (Chapter_10 Data Manipulation and Analysis: Pandas)
    - 只要先將各式檔案(txt, csv, json, xlsx,...)轉成Pandas的資料結構，後續對資料的處理行為都會一致
    - 合併表格化的文字檔只是Pandas模組的一項「小」功能
- 和主程式分開，另外使用yaml格式的檔案來簡化檔案路徑的存取
  - 參考Network Programmability and Automation by Jason Edelman, Scott S. Lowe & Matt Oswalt (ch_5 Data Formats and Data Models)
    - 將檔案的路徑存放在另一個檔案內，yaml格式可以方便轉成python容易處理的資料結構
- 測試：
  - 中央氣象局海象監測資料(劉奕的版本)：txt檔案的讀和寫
    - 合併04/09 ~ 05/06與04/23 ~ 05/21兩項不同日期範圍的資料，最後產生04/09 ~ /05/21新資料
  - 港灣環境資訊網的定點歷線圖：json檔案的讀和寫。xlsx檔案的寫。
    - 使用Pandas前：114行程式碼。使用Pandas後：50行。使用Pandas模組後的新版程式減少64行，最後程式為原先總行數的44%。

合併兩個 txt 檔

頭

date	time	waterlevel(m)	Hs(m)	wave_dir	Ts(s)	wind(m/s)	wind_rank	wind_dir	gust(m/s)
s)	current_spd_knot								
0410	0000	-0.38	nan	nan	nan	4.4	3	NNE	6.5	4	20.0	18.2	nan	nan	nan
0410	0100	-0.40	nan	nan	nan	3.3	2	NW	5.1	3	19.9	17.7	nan	nan	nan
0410	0200	-0.31	nan	nan	nan	3.2	2	N	5.3	3	19.9	17.5	nan	nan	nan
...															
0506	2100	-0.16	nan	nan	nan	1.5	1	SW	2.6	2	23.4	23.0	nan	nan	nan
0506	2200	-0.25	nan	nan	nan	1.5	1	SSW	2.5	2	23.4	22.8	nan	nan	nan
0506	2300	-0.26	nan	nan	nan	0.9	1	SW	1.6	2	23.4	22.5	nan	nan	nan

Input 1

+

尾

date	time	waterlevel(m)	Hs(m)	wave_dir	Ts(s)	wind(m/s)	wind_rank	wind_dir	gust(m/s)
s)	current_spd_knot								
0424	0000	-0.29	nan	nan	nan	2.5	2	NNW	4.2	3	21.1	22.9	nan	nan	nan
0424	0100	-0.14	nan	nan	nan	3.3	2	N	6.1	4	21.1	22.8	nan	nan	nan
0424	0200	0.03	nan	nan	nan	2.7	2	NW	4.5	3	21.1	22.8	nan	nan	nan
...															
0521	2100	-0.18	nan	nan	nan	1.7	2	SW	2.5	2	28.0	28.0	nan	nan	nan
0521	2200	-0.20	nan	nan	nan	1.3	1	WSW	2.8	2	28.0	27.8	nan	nan	nan
0521	2300	-0.09	nan	nan	nan	0.6	1	WSW	1.2	1	27.9	27.7	nan	nan	nan

Input 2

=

頭

尾

date	time	waterlevel(m)	Hs(m)	wave_dir	Ts(s)	wind(m/s)	wind_rank	wind_dir	gust(m/s)
s)	current_spd_knot								
0410	0000	-0.38	nan	nan	nan	4.4	3	NNE	6.5	4	20.0	18.2	nan	nan	nan
0410	0100	-0.40	nan	nan	nan	3.3	2	NW	5.1	3	19.9	17.7	nan	nan	nan
0410	0200	-0.31	nan	nan	nan	3.2	2	N	5.3	3	19.9	17.5	nan	nan	nan
...															
0521	2100	-0.18	nan	nan	nan	1.7	2	SW	2.5	2	28.0	28.0	nan	nan	nan
0521	2200	-0.20	nan	nan	nan	1.3	1	WSW	2.8	2	28.0	27.8	nan	nan	nan
0521	2300	-0.09	nan	nan	nan	0.6	1	WSW	1.2	1	27.9	27.7	nan	nan	nan

output

簡潔的 txt 合併程式

```
1  import yaml
2  import glob
3  import os
4
5  import pandas as pd
6
7
8  with open('path.yml', 'rt') as f_r:
9      data = yaml.load(f_r, Loader=yaml.FullLoader)
10     FOLDER_TXT = data['new']
11     FOLDER_MERGE = data['old']
12
13 #  merge txt files and discard additional duplicates
14
15 files_path = FOLDER_TXT + '*.txt'
16 filenames_new = glob.glob(files_path)
17
18 for filename_new in filenames_new:
19
20     filename_no_path = os.path.basename(filename_new)
21     filename_merge = f'{FOLDER_MERGE}{filename_no_path}'
22
23     df_new = pd.read_table(filename_new, sep='\t', dtype=str)
24
25     try:
26         df_old = pd.read_table(filename_merge, sep='\t', dtype=str)
27     except:
28         df_old = None
29
30     full_df = pd.concat([df_old, df_new])
31     unique_df = full_df.drop_duplicates(subset=['date', 'time'])
32     unique_df.to_csv(filename_merge, index=False, sep='\t', na_rep='nan')
33
34     #print(f'merge {filename_no_path} (complete)')
35 print(f'merge txt files(complete)')
36
```

```
pydoc@pydoc:~/slides/merge_cwb_mmc_txt/code$ tree -L 2
.
├── data
│   ├── merge
│   └── txt
├── merge_cwb_mmc_txt_0.py
└── path.yml

3 directories, 2 files
```

讀取 yaml 檔案
來獲得路徑

開啟(O)　path.yml
~/slides/merge_cwb_mmc_txt/code

```
---
old: './data/merge/' # folder of old, merged files
new: './data/txt/' # folder of new files
```

收集待處理的 txt 檔案 (最新下載到的)

txt 檔被合併前 (後) 的路徑和檔名

將 txt 檔 (最新下載) 轉成 Pandas 的資料結構

將 txt 檔 (待合併的) 轉成 Pandas 的資料結構

合併檔案並根據 date 和 time 兩個欄位剔除重複的部份

將合併後的資料轉成 txt 檔 (覆蓋原先待合併的檔案)

合併兩個 json 檔

頭


```
[
  {
    "datetime": "2021-03-24 11:00:00",
    "year": "2021",
    "MONTH_": "3",
    "day": "24",
    "HOUR": "11",
    "minute": "0",
    "tpv": "23.9"
  },
  {
    "datetime": "2021-03-24 12:00:00",
    "year": "2021",
    "MONTH_": "3",
    "day": "24",
    "HOUR": "12",
    "minute": "0",
    "tpv": "24"
  },
  {
    "datetime": "2021-03-24 13:00:00",
    "tpv": "26.7"
  },
  ...
  {
    "datetime": "2021-04-24 09:00:00",
    "year": "2021",
    "MONTH_": "4",
    "day": "24",
    "HOUR": "9",
    "minute": "0",
    "tpv": "26.8"
  },
  {
    "datetime": "2021-04-24 10:00:00",
    "year": "2021",
    "MONTH_": "4",
    "day": "24",
    "HOUR": "10",
    "minute": "0",
    "tpv": "26.8"
  }
]
```


Input 1

+

尾


```
[
  {
    "datetime": "2021-04-22 11:00:00",
    "year": "2021",
    "MONTH_": "4",
    "day": "22",
    "HOUR": "11",
    "minute": "0",
    "tpv": "26.5"
  },
  {
    "datetime": "2021-04-22 12:00:00",
    "year": "2021",
    "MONTH_": "4",
    "day": "22",
    "HOUR": "12",
    "minute": "0",
    "tpv": "26.6"
  },
  {
    "datetime": "2021-04-22 13:00:00",
    "tpv": "30.1"
  },
  ...
  {
    "datetime": "2021-05-22 09:00:00",
    "year": "2021",
    "MONTH_": "5",
    "day": "22",
    "HOUR": "9",
    "minute": "0",
    "tpv": "30.2"
  },
  {
    "datetime": "2021-05-22 10:00:00",
    "year": "2021",
    "MONTH_": "5",
    "day": "22",
    "HOUR": "10",
    "minute": "0",
    "tpv": "30.2"
  }
]
```


Input 2

=

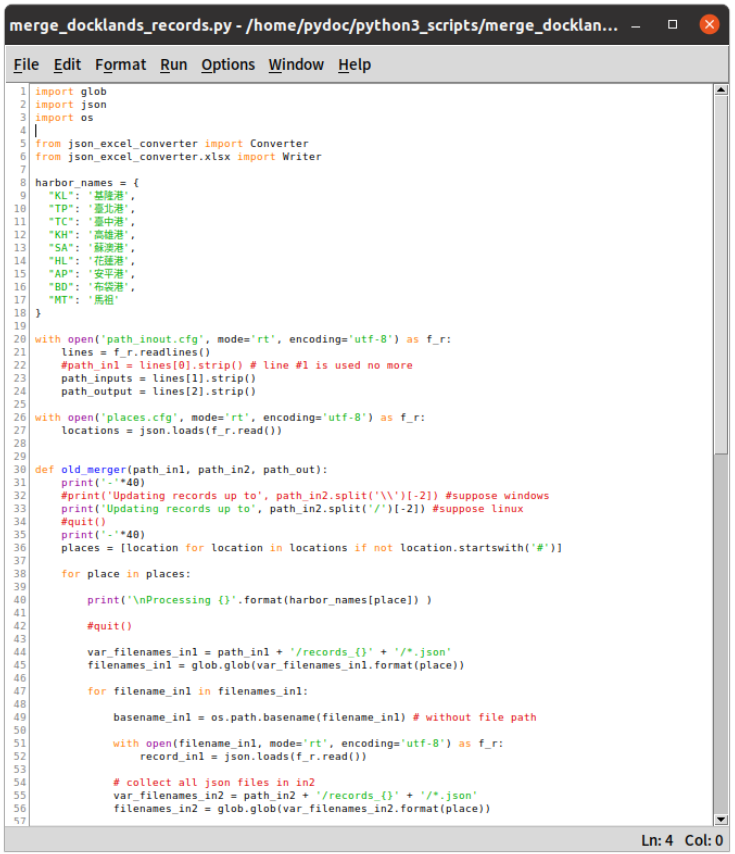
頭

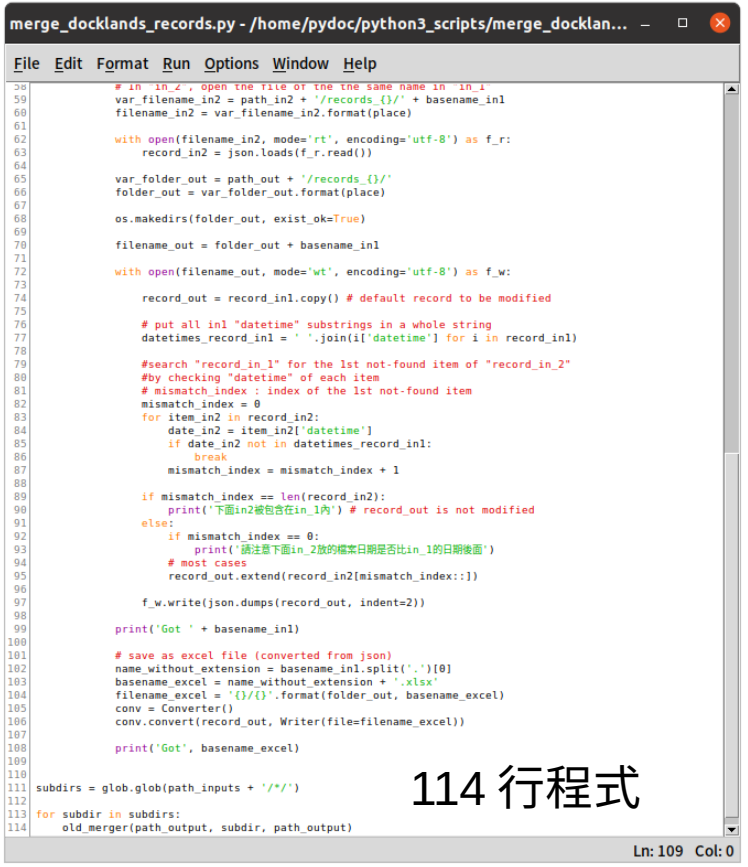
尾


```
[
  {
    "datetime":"2021-03-24 11:00:00",
    "year":"2021",
    "MONTH_":"3",
    "day":"24",
    "HOUR":"11",
    "minute":"0",
    "tpv":"23.9"
  },
  {
    "datetime":"2021-03-24 12:00:00",
    "year":"2021",
    "MONTH_":"3",
    "day":"24",
    "HOUR":"12",
    "minute":"0",
    "tpv":"24"
  },
  {
    "datetime":"2021-03-24 13:00:00",
    "tpv":"30.1"
  },
  ...
  {
    "datetime":"2021-05-22 09:00:00",
    "year":"2021",
    "MONTH_":"5",
    "day":"22",
    "HOUR":"9",
    "minute":"0",
    "tpv":"30.2"
  },
  {
    "datetime":"2021-05-22 10:00:00",
    "year":"2021",
    "MONTH_":"5",
    "day":"22",
    "HOUR":"10",
    "minute":"0",
    "tpv":"30.2"
  }
]
```


output

使用 Pandas 前（定點歷線圖程式）





114 行程式

使用 Pandas 後（定點歷線圖程式）

```python
import yaml
import glob
import os

import pandas as pd


with open('path_docklands.yml', 'rt') as f_r:
    data = yaml.load(f_r, Loader=yaml.FullLoader)
    FOLDER_JSON = data['new']
    FOLDER_MERGE = data['old']

#  merge json files and discard additional duplicates

files_path = FOLDER_JSON + '*.json'
# specify recursive to walk all parent folders of json files
filenames_new = glob.glob(files_path, recursive=True)


for filename_new in filenames_new:

    path_file = os.path.split(filename_new)
    # -> ('./inputs/20210522_103004/records_BD', 'BD_info_markers_tempZ1.json')

    filename_no_path = path_file[-1] # get 'BD_info_markers_tempZ1.json'
    path_only = os.path.split(path_file[0])
    # -> ('./inputs/20210522_103004', 'records_BD')
    last_subpath = path_only[-1] # get 'records_BD'

    filename_merge = os.path.join(FOLDER_MERGE, last_subpath, filename_no_path)

    filename_for_xlsx = filename_no_path.split('.')[0] + '.xlsx'
    filename_xlsx = os.path.join(FOLDER_MERGE, last_subpath, filename_for_xlsx)

    df_new = pd.read_json(filename_new, dtype=str) #all columns as srings

    try:
        df_old = pd.read_json(filename_merge, dtype=str) #all columns as srings
    except:
        df_old = None

    full_df = pd.concat([df_old, df_new])
    unique_df = full_df.drop_duplicates(subset='datetime')
    unique_df.to_excel(filename_xlsx, index=False)
    # specify indent for text editor to work well
    unique_df.to_json(filename_merge, orient='records', indent=2)

    print(f'merge {filename_merge} (complete)')

#print(f'merge json (& xlsx) files(complete)')
```

使用 Pandas 後縮減剩 50 行程式