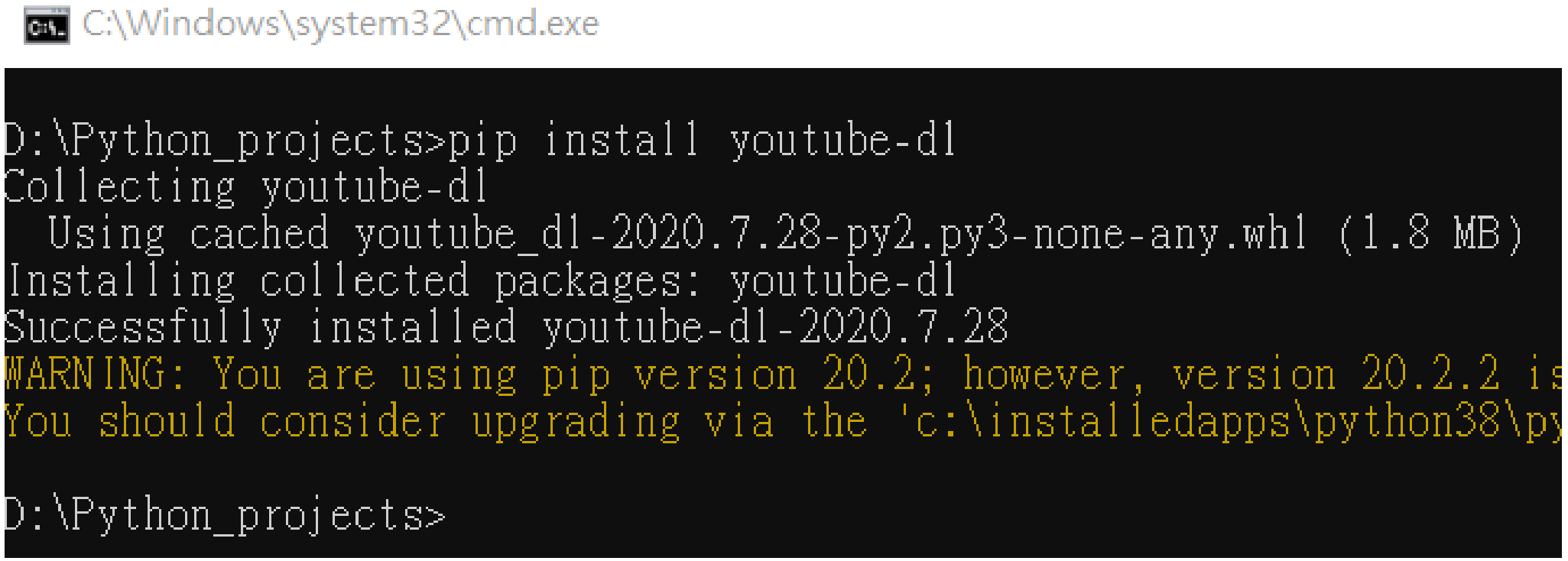
爬「港灣環境資訊網」

布袋港的網路爬蟲

如何網路爬蟲 (webscraping)

- 先安裝好學易懂的 python 程式語言解釋器 (其它語言也可以做網路爬蟲)
  - https://www.python.org/downloads/
- 依需求安裝第三方的模組 (module, package, library)
  - 這是 python 的最大優勢，站在巨人的肩膀上做開發，不需要從無到有慢慢刻
  - pip install xxx
    - 在 windows command prompt 視窗下敲以上指令
    - xxx 代表某個模組，比如 youtube-dl
- 判斷網站 (有點偵探的感覺)，原則上有 3 種方式
  - 尋找 web api (例如「中央氣象局」就有提供，這種站很少)
  - 分析網頁 (html)，土法煉鋼慢慢 try(這次爬「港灣環境資訊網」網站的資料就是)
  - 模擬瀏覽器 (靠第三方模組 selenium)

安裝第三方模組


C:\Windows\system32\cmd.exe
D:\Python_projects>pip install youtube-dl
Collecting youtube-dl
Using cached youtube_dl-2020.7.28-py2.py3-none-any.whl (1.8 MB)
Installing collected packages: youtube-dl
Successfully installed youtube-dl-2020.7.28
WARNING: You are using pip version 20.2; however, version 20.2.2 is
You should consider upgrading via the 'c:\installedapps\python38\py
D:\Python_projects>

港灣環境資訊網

https://isohe.ihmt.gov.tw/docklands/Budai.aspx#in-content

港灣環境資訊網
Harbor Environment Information Website

字體大小： A A+ 線上人數：11

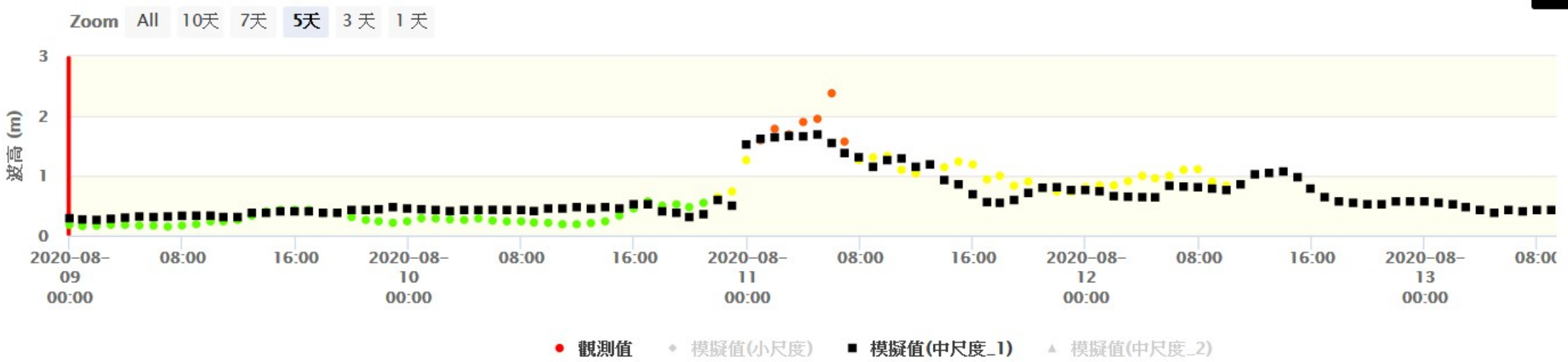
港區海象 全國海象 藍色公路 港區影像 港區地震 港區海嘯 港區腐蝕

風力歷線圖 | 波浪歷線圖 | 海流歷線圖 | 潮汐/水溫歷線圖

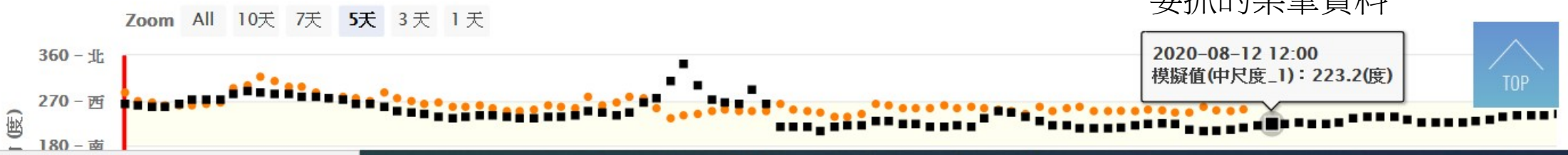
測站名稱: 選擇日期: 2020/08/12 00 時 查詢

<=0.6 | >0.6 <=1.5 | >1.5 <=3 | >3

波高單位：公尺(m)



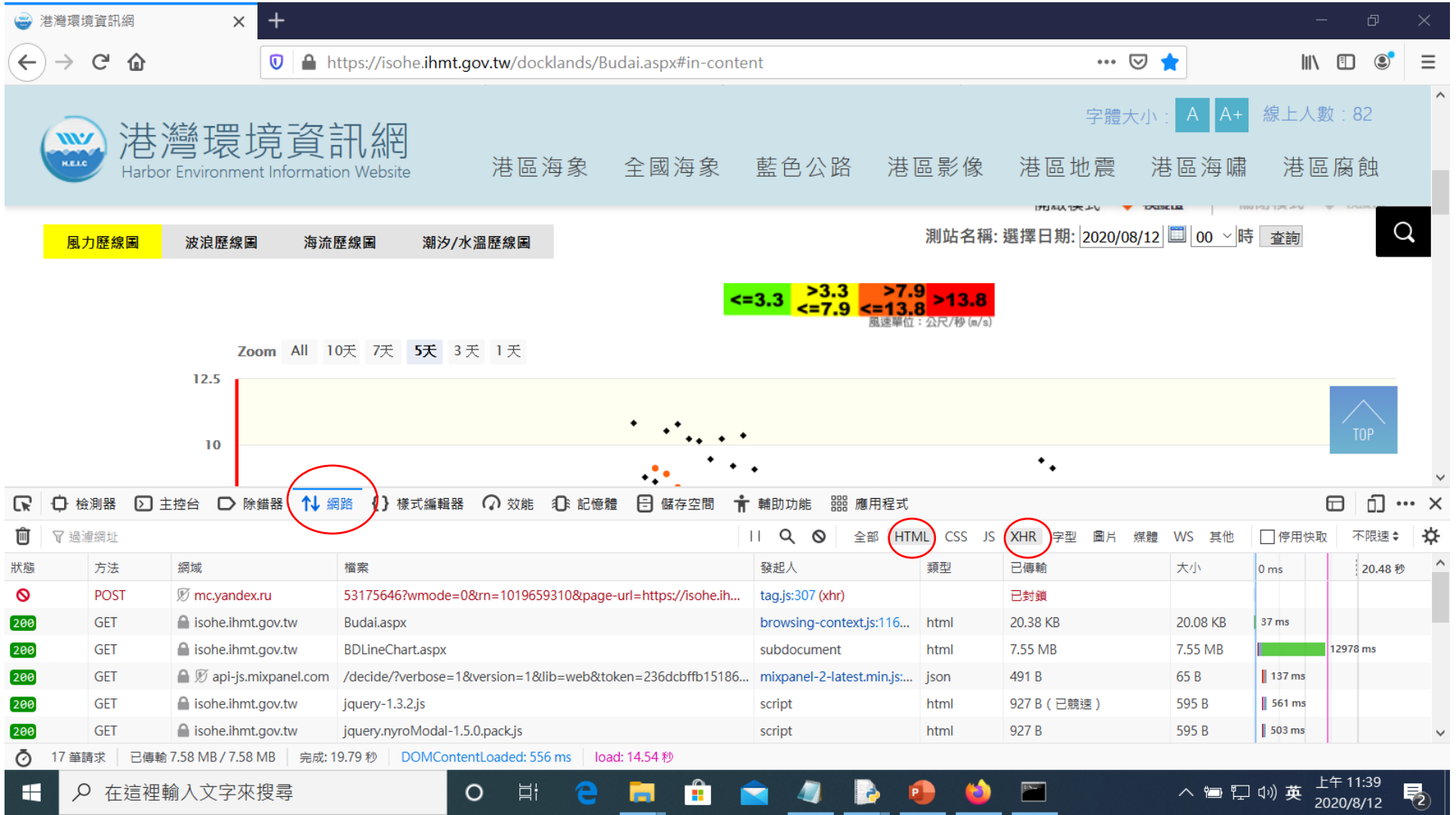
要抓的某筆資料



在這裡輸入文字來搜尋

英 上午 11:35 2020/8/12

開啟瀏覽器內附的開發者模式



分析網站的 source html(只需處理文字)

source.html

```
<script type="text/javascript">

//波浪觀測資料

var markers_waveZ1 = [

{

"datetime": '2020-07-12 14:00:00',

"year": '2020',

"MONTH_": '7',

"day": '12',

"HOUR": '14',

⋮

"minute": '0',

"hs" : '0.96',

"mdir" : '244',

"tp" : '7.1'

}

];

//波浪模擬資料

var markers_waveZ2 = [
```

1 (開始前的標記)

2 (中間廣大的區塊，這是我們要的)

3 (結束後的標記)

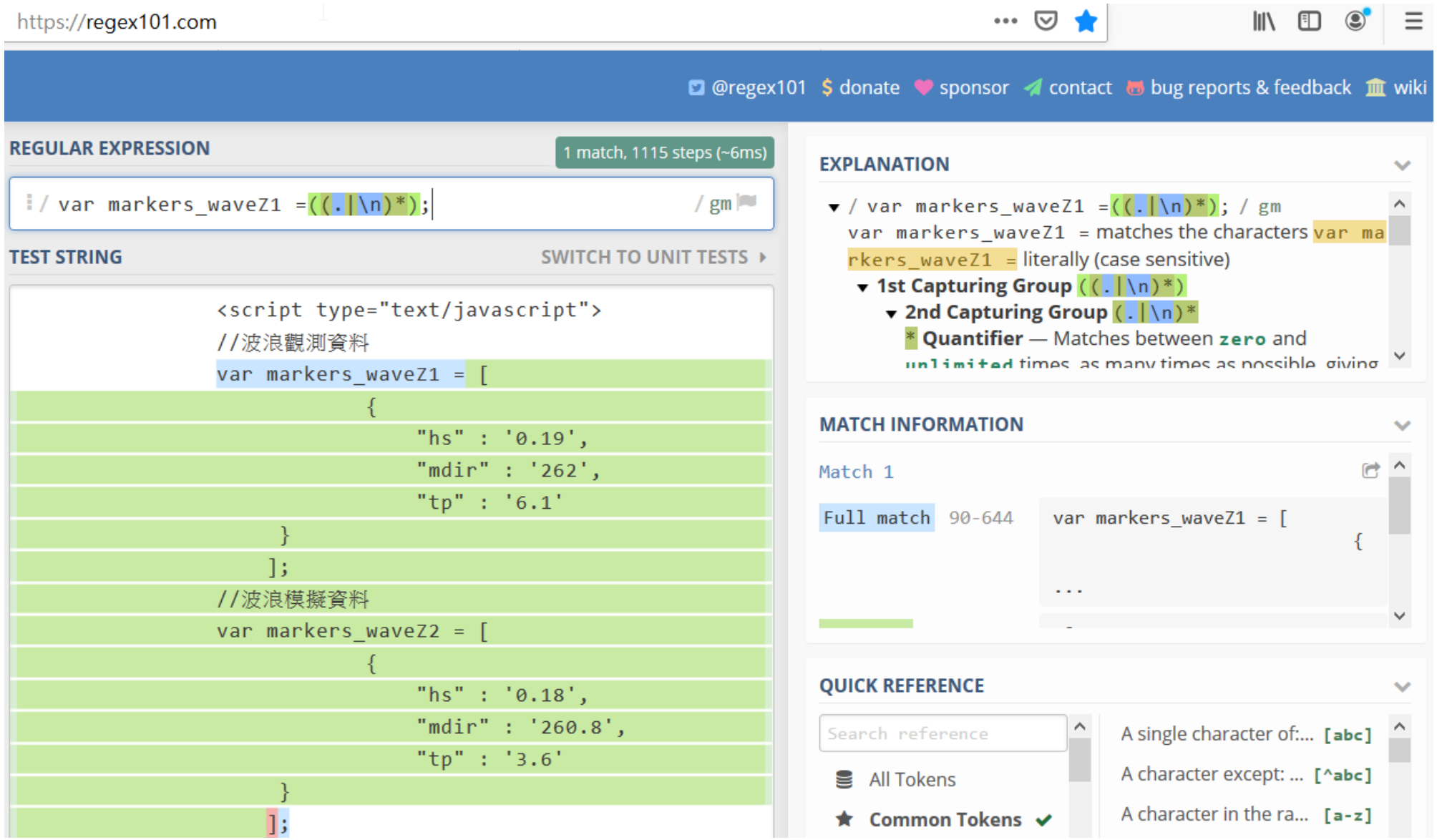
用正規表示式 (Regular Expression) 過濾文字

```
pattern = re.compile(r'var markers_waveZ1 =((.|\n)*);')
primitive = pattern.search(src_file)
```

. 任意字元 (不含換行控制符號)
\n 換行
(.|\n) 任意字元或換行
(.|\n)* “ 任意字元或換行” 從 0 到無限多
((.|\n)*) 欲擷取的資料範圍框成一群組

上面寫法會抓太多，下面是其補救並修正成 json 格式

```
raw = primitive.group(1).partition(';')
tmp2 = raw[0].replace('\'','"')
```


https://regex101.com
@regex101 $ donate sponsor contact bug reports & feedback wiki
REGULAR EXPRESSION
1 match, 1115 steps (~6ms)
/ var markers_waveZ1 =((.|\n)*); / gm
TEST STRING
SWITCH TO UNIT TESTS
<script type="text/javascript">
//波浪觀測資料
var markers_waveZ1 = [
{
"hs" : '0.19',
"mdir" : '262',
"tp" : '6.1'
}
];
//波浪模擬資料
var markers_waveZ2 = [
{
"hs" : '0.18',
"mdir" : '260.8',
"tp" : '3.6'
}
];
EXPLANATION
/ var markers_waveZ1 =((.|\n)*); / gm
var markers_waveZ1 = matches the characters var markers_waveZ1 = literally (case sensitive)
1st Capturing Group ((.|\n)*)
2nd Capturing Group (.|\n)*
* Quantifier — Matches between zero and unlimited times, as many times as possible, giving
MATCH INFORMATION
Match 1
Full match 90-644
var markers_waveZ1 = [
{
...
QUICK REFERENCE
Search reference
All Tokens
Common Tokens
A single character of:... [abc]
A character except: ... [^abc]
A character in the ra... [a-z]

更精準地指定正規表示式條件

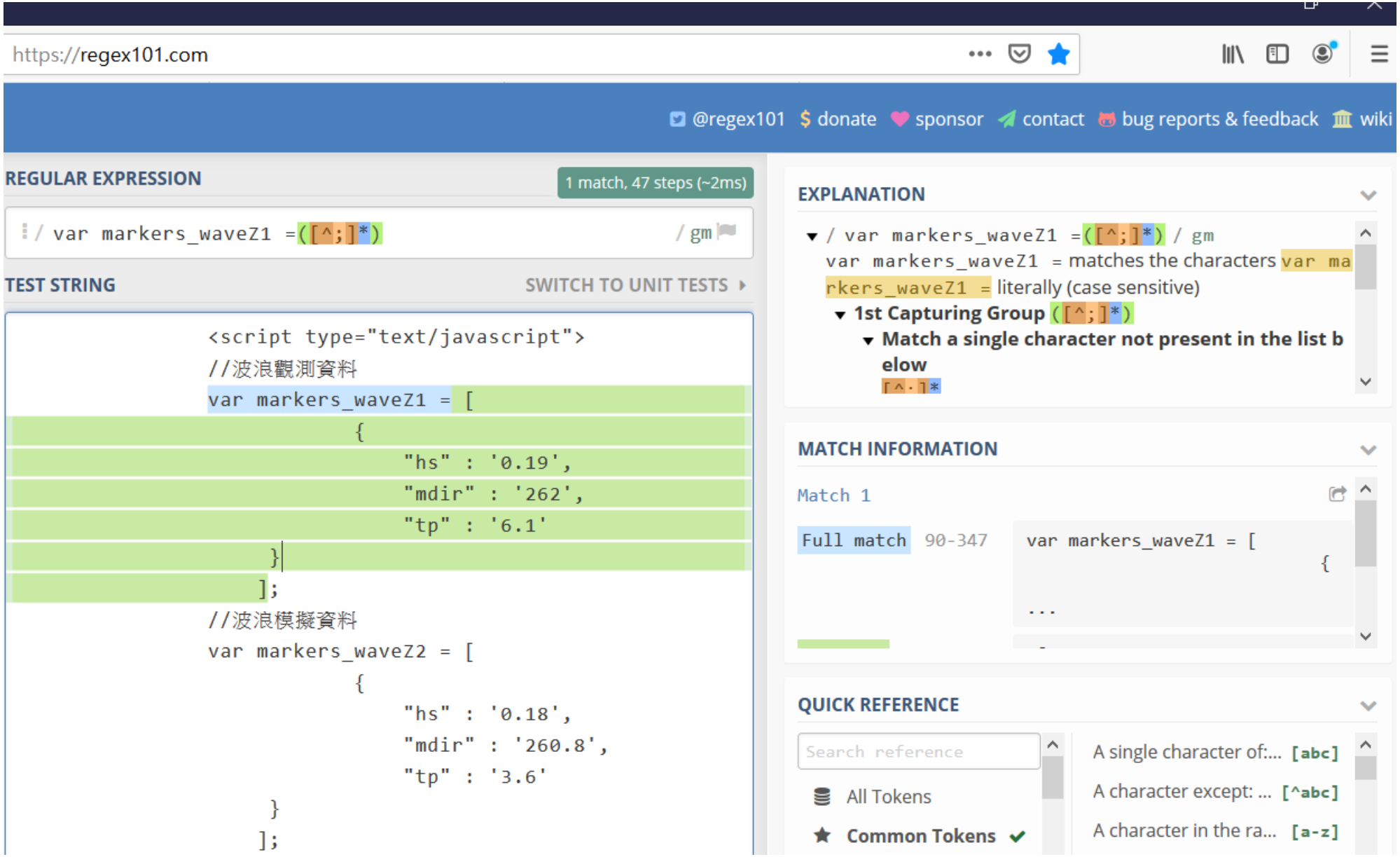
```
pattern = re.compile(r'var markers_waveZ1 =([^;]*)’)
primitive = pattern.search(src_file)
```

```
raw = primitive.group(1)
tmp2 = raw.replace('\'','"')
```

[^;] 不含；的任意字元

[^;]* 上述不含；的字元可以有 0 至無窮多

([^;]*) 欲擷取的資料範圍框成一群組


https://regex101.com
@regex101
donate
sponsor
contact
bug reports & feedback
wiki
REGULAR EXPRESSION
1 match, 47 steps (~2ms)
/ var markers_waveZ1 =([^;]*)
/ gm
TEST STRING
SWITCH TO UNIT TESTS
<script type="text/javascript">
//波浪觀測資料
var markers_waveZ1 = [
{
"hs" : '0.19',
"mdir" : '262',
"tp" : '6.1'
}
];
//波浪模擬資料
var markers_waveZ2 = [
{
"hs" : '0.18',
"mdir" : '260.8',
"tp" : '3.6'
}
];
EXPLANATION
/ var markers_waveZ1 =([^;]*) / gm
var markers_waveZ1 = matches the characters var markers_waveZ1 = literally (case sensitive)
1st Capturing Group ([^;]*)
Match a single character not present in the list below
[^;]*
MATCH INFORMATION
Match 1
Full match 90-347
var markers_waveZ1 = [
{
...
QUICK REFERENCE
Search reference
All Tokens
Common Tokens
A single character of:... [abc]
A character except: ... [^abc]
A character in the ra... [a-z]

程式輪廓 1(準備)

harbor_budai_Abstract.py - D:\Python_projects\harbor\harbor_budai_Abstract.py (3.8.5)

File Edit Format Run Options Window Help

```
import requests
import re
import json
import os

from json_excel_converter import Converter
from json_excel_converter.xlsx import Writer

headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.102 Safari/537.36'
}

url = "https://isohe.ihmt.gov.tw/docklands/chart/BDLineChart.aspx"

res = requests.get(url, headers=headers)
res.raise_for_status()

with open('source.html', mode='wt', encoding='utf-8') as f_w:
    src_file = res.text
    f_w.write(src_file)

vars = [
        # 風力歷線圖
        'markers_windZ1', # 風力觀測資料
        'markers_windZ2', # 風力模擬資料
        'markers_windZ3', # 風力模擬資料

        # 波浪歷線圖
        'markers_waveZ1', # 波浪觀測資料
        'markers_waveZ2', # 波浪模擬資料
        'markers_waveZ3', # 波浪模擬資料
        'markers_waveZ4', # 波浪模擬資料 ?
        'markers_waveZ5', # 波浪模擬資料

        # 海流歷線圖
        'markers_currentZ1', # 海流觀測資料
        'markers_currentZ2', # 海流模擬資料 空
        'markers_currentZ3', # 海流模擬資料

        # 潮汐歷線圖
        'markers_tideZ1', # 潮位觀測資料
        'markers_tideZZ', # 潮位觀測資料
        'markers_tideZ2', # 潮位模擬資料
        'markers_tideZ3', # 潮位模擬資料
        'markers_tideZ4', # 潮位模擬資料 空
        'markers_tideZ5', # 潮位模擬資料

        # 水溫歷線圖
        'markers_tempZ1', # 水溫觀測資料

        # 能見度歷線圖
        'markers_visZ1', # 能見度觀測資料
        ]
```

把裝備先選好

選好目標

啟動開關

抓到最原始的資料了

選定接下來要過濾的細項

程式輪廓 2
(處理核心)

不想存較原始的文字檔了 (debug 用)

存成網路世界通用的 json 格式檔案
並在螢幕上顯示訊息

存 excel 格式檔案，有第三方模組用真好
，大家快來學 python 吧，老闆要以身作則！

```
folderName = 'records'
os.makedirs(folderName, exist_ok=True)

for var in vars:
    # use regular expression to aid in filtering raw dta
    re_filter = r'var {} =([^;]*)'

    pattern = re.compile(re_filter.format(var) )

    primitive = pattern.search(src_file)

    '''
    # save filtered-raw-data file
    filename_filteredRawData = '{}/raw_{}.txt'.format(folderName,var)
    with open(filename_filteredRawData, mode='wt', encoding='utf-8') as f:
        tmp1 = primitive.group(1)
        f.write(tmp1)

    print('Got', filename_filteredRawData)

    tmp2 = tmp1.replace('\'','"') # replace ' with "
    '''

    # handle re object without saving filtered-raw-data
    raw = primitive.group(1)
    tmp2 = raw.replace('\'','"') # replace ' with "

    info = json.loads(tmp2) # got python object(list with dictionaries)


    # save modified info as json file
    filename_modifiedInfo = '{}/info_{}.json'.format(folderName,var)
    with open(filename_modifiedInfo, mode='wt', encoding='utf-8') as f:
        #f.write(json.dumps(info))
        f.write(json.dumps(info, indent=2))

    print('Got', filename_modifiedInfo)

    # save as excel file (converted from json)
    filename_excel = '{}/info_{}.xlsx'.format(folderName,var)
    conv = Converter()
    conv.convert(info, Writer(file=filename_excel)) # info is python object

    print('Got', filename_excel)
```

設好子目錄準備放過濾好的資料

一次處理一種港灣環境資料

趕快把強大的正規表示式學起來吧！
大陸那邊叫正則表示式

原始的文字資料需要將單引號換成
雙引號後才能滿足 json 的格式

程式輪廓 3(資料註解)

```
###################
#   風力歷線圖    #
###################

# <windZ1>
#   wd: 風向觀測值
#   ws: 風速觀測值

# <windZ2>
#   wd: 風向模擬值(小尺度)
#   ws: 風速模擬值(小尺度)

# <windZ3>
#   wd: 風向模擬值(中尺度)
#   ws: 風速模擬值(中尺度)

###################
#   波浪歷線圖    #
###################

# <waveZ1>
#   hs: 波高觀測值
# mdir: 波向觀測值
#   tp: 週期觀測值

# <waveZ2>
#   hs: 波高模擬值(小尺度)
# mdir: 波向模擬值(小尺度)
#   tp: 週期模擬值(小尺度)

# <waveZ3>
#   hs: 波高模擬值(中尺度_1)
# mdir: 波向模擬值(中尺度_1)
#   tp: 週期模擬值(中尺度_1)

# <waveZ4>
# ?

# <waveZ5>
#   hs: 波高模擬值(中尺度_2)
# mdir: 波向模擬值(中尺度_2)
#   tp: 週期模擬值(中尺度_2)
```

純粹給人看的，
和 python 沒啥關

```
###################
#   海流歷線圖    #
###################

# <currentZ1>
# vc: 流速觀測值
# vd: 流向觀測值

# <currentZ3>
# vc: 流速模擬值(中尺度)
# vd: 流向模擬值(中尺度)

###################
#   潮汐歷線圖    #
###################

# <tideZ1>
# tva: 潮位觀測值(1)

# <tideZZ>
# tva: 潮位觀測值(2)

# <tideZ2>
# tva: 潮位模擬值(中尺度_1)

# <tideZ3>
# tva: 潮位模擬值(中尺度_2)

# <tideZ4>
# 空

# <tideZ5>
# tva: 潮位模擬值(調和分析)

###################
#   水溫歷線圖    #
###################

# <tempZ1>
# tpv: 水溫觀測值

####################
#   能見度歷線圖    #
####################

# <visZ1>
# vis: 能見度觀測值
```

改進 1 （下載所有港口和所有項目）

- 歷線圖港口的網址移出主程式（另存一個檔案）

```
#url = "https://isohe.ihmt.gov.tw/docklands/chart/BDLineChart.aspx"
with open('places.cfg', mode='rt', encoding='utf-8') as f_r:
    locations = json.loads(f_r.read())

places = [location for location in locations if not location.startswith('#')]

for place in places:

    print('\nProcessing {}'.format(harbor_names[place]) )
    #place = 'BD'#'KH'#'BD'
    where = "https://isohe.ihmt.gov.tw/docklands/chart/{}LineChart.aspx"
    url = where.format(place)
```

*places.cfg - 記事本

檔案(F) 編輯(E) 格式(

```
[
  "KL",
  "#TP",
  "TC",
  "#KH",
  "SA",
  "HL",
  "AP",
  "#BD",
  "MT"
]
```

台北港、高雄港和布袋港不下載
(雙英文字母代號前加 # 符號)

- 歷線圖港口的項目移出主程式外（另存一個檔案）

```
with open('js_vars.cfg', mode='rt', encoding='utf-8') as f_r:
    items = json.loads(f_r.read())

vars = [item for item in items if not item.startswith('#')] #

'''
vars = [
        # 風力歷線圖
        'markers_windZ1', # 風力觀測資料
        'markers_windZ2', # 風力模擬資料
```

*js_vars.cfg - 記事本

檔案(F) 編輯(E) 格式(O)

```
[
  "#markers_windZ1",
  "markers_windZ2",
  "markers_windZ3",
  "markers_waveZ1",
  "markers_waveZ2",
  "markers_waveZ3",
  "markers_waveZ4",
```

←風力觀測資料不下載
(markers?? 前加 # 符號)

改進 2（提供圖形化介面）

- 將之前用 command line 寫法的程式主體改成 function 後，將之套在 GUI 的世界裡。

```
def download_history_lines():

    print('\n請等幾秒鐘下載網頁資料\n')
    window.Refresh()

    where = "https://isohe.ihmt.gov.tw/d
    url = where.format(place)
```

⋮

```
import PySimpleGUI as sg

sg.theme('BluePurple')

layout = [[sg.Text('港口：'), sg.Text(size=(15,1), key='-OUTPUT-')],
          [sg.Text(hint)],
          [sg.Output(size=(80,20))],
          #[sg.Multiline(size=(80,20))], # output messages do not ap
          [sg.Button('Harbor'), sg.Input(key='-IN-')],
          [sg.Button('Run'), sg.Button('Exit')]]

window = sg.Window('港灣環境資訊網(下載單一港口的歷線圖)', layout)


while True:  # Event Loop
    event, values = window.read()
    #print(event, values)
    if event == sg.WIN_CLOSED or event == 'Exit':
        break
    if event == 'Harbor':
        # Update the "output" text element to be the va
        place = values['-IN-'].upper()
        if place in harbor_names.keys():
            harbor_name = harbor_names[place]
            window['-OUTPUT-'].update(harbor_name)
        else:
            print('港口代號輸入錯誤，請參考以下列表。')
            for symbol, name in harbor_names.items():
                print(symbol, name)

    if event == 'Run':
        download_history_lines()

window.close()
```

GUI 前置作業

GUI 主體迴圈