

將 windy.com 某 GPS 座標的資料存成 CSV

# 存取 windy.com 上的即時資料

- 擷取某 GPS 座標的**海洋溫度**、**洋流**和**波浪** 3 項資訊
  - 存成 csv 檔案並放在 [/WTGroup\\_Data/information/Windy\\_PowerPlants](#)
    - 用一般帳號登入後即可存取，不需要用到特殊的 administrator 權限
    - **不要在群暉主機上直接開啟檔案**，可能會造成**檔案寫入失敗** (程式剛好跑到時撞到)
  - **11 個電廠**，共 **56 處 GPS 座標點**，在群暉主機上**每小時下載一次**
    - 順利的話，約 **15 分鐘內**抓完所有的資料
    - 不是每次都能成功下載，若失敗，則放棄這次位置，改換成存取下個座標點
      - **關閉** windy.com 網頁的「**粒子動畫**」可大幅提升成功機率 (目前看到是九成九以上)
      - 拉長「**按鍵**」之間的時間間隔也可提升成功機率，不過沒關閉粒子動畫後的效果明顯
- 似乎不需要登入收費帳號就可以擷取每小時的即時資料

# 檔案在群暉主機上的存放路徑

File Station

WTGroup\_Data > information > Windy\_PowerPlants > 塔山

上傳 新增 操作 工具 設定

Windy\_PowerPlants

- 協和
- 協和馬祖珠山
- 南部
- 台中
- 塔山**
- 大林
- 大潭
- 尖山
- 林口
- 興達
- 通霄

12 個電廠, 共 56 處 gps 座標點

CSV 檔案的存放路徑

- 1A\_24-42242\_118-27787.csv
- 1B\_24-41955\_118-28245.csv
- 2A\_24-4131\_118-2714.csv
- 2B\_24-409767\_118-276883.csv
- 背景站\_24-423286\_118-267192.csv

CSV 檔名由名稱和 gps 座標所組成

背景站\_24-423286\_118-267192-1.csv (唯讀) - LibreOffice Calc

檔案(F) 編輯(E) 檢視(V) 插入(I) 格式(O) 樣式(Y) 工作表(S) 資料(D) 工具(T) 視

A1 fx Σ = time

此文件以唯讀模式開啟。

	A	B	C	D	E	F	G	H
1	time	Sea temperature	Currents	Waves				
2	2020/11/02 10:42:53	22°C	E 0.2 kt	NE 0.8m Period 4 s.				
3	2020/11/02 11:06:15	22°C	E 0.2 kt	NE 0.8m Period 4 s.				
4	2020/11/02 12:06:13	22°C	E 0.2 kt	NE 0.8m Period 4 s.				
5	2020/11/02 13:06:30	22°C	E 0.2 kt	E 1m Period 4 s.				
6	2020/11/02 14:06:16	22°C	E 0.2 kt	E 1m Period 4 s.				

檔案內的資訊

# 加入任務排程表

新增 ▾ 編輯 執行 動作 ▾ 儲存 設定

已啟動	擁有者	應用程式	任務名稱	動作	下次執行時間
<input checked="" type="checkbox"/>	root	使用者定義指令碼	Windy crawler - Power Plants	執行: docker container start win...	2020-10-30 16:00
<input type="checkbox"/>	root			執行: docker container start win...	2020-10-30 16:00
<input checked="" type="checkbox"/>	root			執行: docker container start sel_...	2020-10-30 19:00
<input checked="" type="checkbox"/>	root			執行: docker container start pyt...	2020-10-31 00:00
<input checked="" type="checkbox"/>	root			發送 DSM 更新通知	2020-10-31 00:20
<input checked="" type="checkbox"/>	Administrator			執行: export PYTHONIOENCODIN...	2020-10-31 10:30
<input checked="" type="checkbox"/>	root			對所有支援快速檢測功能的硬碟執行...	2020-11-18 00:00

### 編輯任務

一般 排程 任務設定

#### 通知設定

透過電子郵件傳送執行細節 ⓘ

電子郵件:

僅在指令碼異常終止時傳送執行細節

#### 執行命令

使用者定義指令碼

```
docker container start windy_powerplants
docker container exec windy_powerplants sh -c "cd /root ; python3
windy_chrome_PowerPlants_docker.py ; exit"
docker container stop windy_powerplants
```

← 三道 docker 指令，每小時執行一次

確定 取消

# 程式原始碼的存放路徑

The image shows a web-based file station interface. The address bar at the top displays `https://60.249.144.60:5001`. The main area is titled "File Station" and shows a breadcrumb path: `home > python3_scripts > windy_powerplants`. Below the path are navigation buttons: "上傳", "新增", "操作", "工具", and "設定". A search bar with a magnifying glass icon and the text "搜尋" is also present. The left sidebar shows a tree view of the file system, with "windy\_powerplants" selected. The right pane displays the contents of the "windy\_powerplants" directory, including folders "output", ".local", ".pki", and "input", and files ".bash\_history" and "windy\_chrome\_PowerPlants\_docker.py". A red box highlights the breadcrumb path, and a red text overlay reads "需要 administrator 帳號才看得到".

需要 administrator 帳號才看得到

```
windy_chrome_PowerPlants_docker.py - 文字編輯器
```

檔案 編輯 搜尋 偏好設定

```
71 # 關閉粒子動畫
72 PARTICLES_CSS = '#particles'
73 particles_mode = browser.find_element_by_css_selector(PARTICLES_CSS)
74 if 'off' not in particles_mode.get_attribute('class'):
75     particles_mode.click()
76     time.sleep(AFTER_CLICK)
77
78 #---- 定義 主要函數 ----#
79 #for gps_pos in gps_positions:
80 def gps2csv(gps_pos, gps_name):
81
```

部分程式碼

# 檢查要觀測的資料是否打開 (例如，海洋溫度)



1. 瀏覽器上顯示的待觀察資料

```
> <a data-do="set,cloudtop" data-overlay="cloudtop">...</a>  
> <a data-do="set,cbase" data-overlay="cbase">...</a>  
> <a data-do="set,visibility" data-overlay="visibility">...</a>  
> <a data-do="set,cape" data-overlay="cape">...</a>  
> <a data-do="set,waves" data-overlay="waves">...</a>  
> <a data-do="set,swell1" data-overlay="swell1">...</a>  
> <a data-do="set,swell2" data-overlay="swell2">...</a>  
> <a data-do="set,swell3" data-overlay="swell3">...</a>  
> <a data-do="set,wwaves" data-overlay="wwaves">...</a>  
▼ <a data-do="set,sst" data-overlay="sst"> == $0  
  > <div class="checkbox off" data-do="toggleFav,sst">...</div>  
  > <span class="noselect inlined" data-icon-after="☐">...</span>  
</a>  
> <a data-do="set,currents" data-overlay="currents">...</a>  
> <a data-do="set,no2" data-overlay="no2">...</a>
```

2. Google chrome 開發者模式內對應的 html5 程式碼

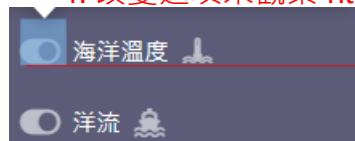
3. CSS selector 顯示的內容

5. python 程式碼檢查是否開啟海洋溫度選項

```
SST_TOGGLE_CSS = '#plugin-overlays a[data-do="set,sst"] div'  
sst_toggle = browser.find_element_by_css_selector(SST_TOGGLE_CSS)  
if 'off' in sst_toggle.get_attribute('class'):  
    sst_toggle.click()  
    time.sleep(AFTER_CLICK)
```

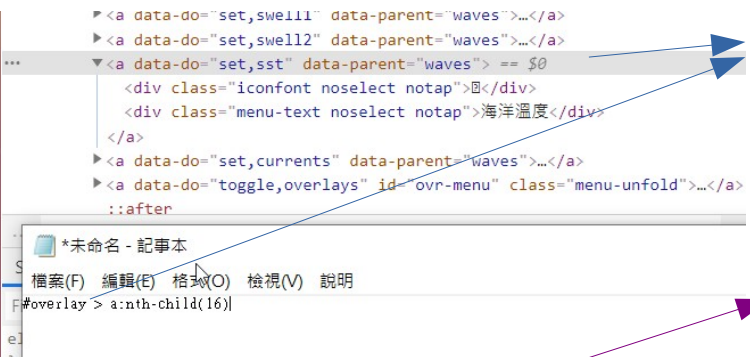
4. 改變選項來觀察 html5 的變化

其他選項以此類推

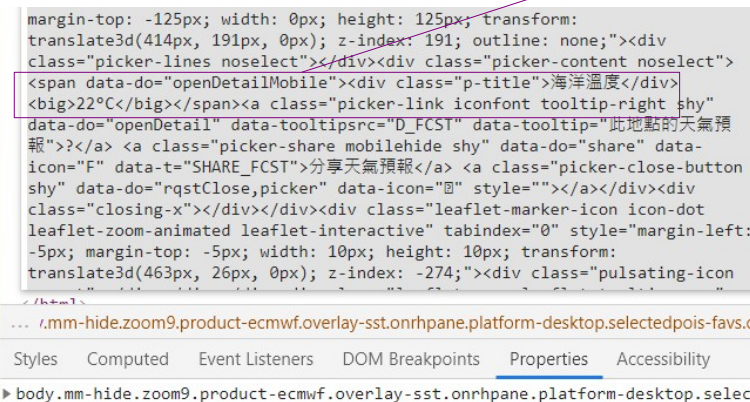
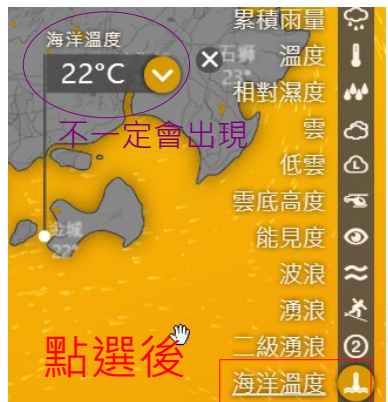


```
> <a data-do="set,sst" data-overlay="sst"> == $0  
  > <div class="checkbox" data-do="toggleFav,sst">...</div>  
  > <span class="noselect inlined" data-icon-after="☐">...</span>  
</a>
```

# 顯示和記錄擷取到的資料



```
# 顯示海洋溫度  
SST_SHOW_CSS = '#overlay > a[data-do="set,sst"]'  
try:  
    #if_sst = EC.visibility_of_element_located((By.CSS_SELECTOR,SST_SHOW_CSS)  
    if_sst = EC.element_to_be_clickable((By.CSS_SELECTOR,SST_SHOW_CSS))  
    WebDriverWait(browser, 10).until(if_sst) # 10,20  
except TimeoutException:  
    print('Timed out: [sst show] window')  
    sst_show = browser.find_element_by_css_selector(SST_SHOW_CSS)  
    sst_show.click()  
    time.sleep(AFTER_CLICK) # 3,5 may fail # 這個delay無法拿掉  
SHOW_BOX_CSS = 'span[data-do="openDetailMobile"]'
```



```
try:  
    #if_show = EC.visibility_of_element_located((By.CSS_SELECTOR,SHOW_BOX_CS  
    if_show = EC.element_to_be_clickable((By.CSS_SELECTOR,SHOW_BOX_CSS))  
    WebDriverWait(browser, 3).until(if_show) # 10,20 failed  
except TimeoutException:  
    print('Timed out: [show box] window')  
    # 這裡放出錯時，試圖補救的程式碼
```

```
show_box = browser.find_element_by_css_selector(SHOW_BOX_CSS)  
print(show_box.text)  
# show_box 第一個字串是屬性，當作key.其餘當作內容  
# show_box的內容似乎會自動根據所選的屬性而變，不需要每次都呼叫  
info = show_box.text.split('\n')  
record[info[0]] = '.'.join(info[1:])
```

穩定出現「海洋溫度視窗」是目前最大瓶頸

# 最外圍的控制流程 (top down)

```
#---- 主要程式 ----#  
path_input = './input'  
var_filenames_input = path_input + '/*.cfg'  
filenames_input = glob.glob(var_filenames_input)
```

收集所有電廠的設定

處理單一電廠

```
for filename_input in filenames_input:
```

```
    with open(filename_input, mode='rt', encoding='utf-8') as f_r:  
        gps_raw = f_r.readlines()
```

```
        gps_es = gps_raw[1:] # discard the 1st line  
        # simple filter of gps positions  
        gps_positions = [g for g in gps_es if ',' in g and (not g.startswith('#'))]
```

收集單一電廠內的 gps 座標點

```
        filename_nopath = os.path.basename(filename_input) # discard additional path  
        place = filename_nopath.partition('-')[0] # e.g. 塔山-gps_pos.cfg  
        show_process = place + ' : 處理中'  
        print('') # 和前一筆資料隔開  
        print('□' * len(show_process))  
        print(show_process)  
        print('□' * len(show_process))  
        for pos in gps_positions:
```

一旦處理目前的 gps 座標點出現  
錯誤，則改成處理下一座標點

```
            try:  
                gps2csv(pos, place) # 處理單一 gps 座標點並將擷取的資料寫入 csv 檔案  
            #except NoSuchElementException:  
            except: # 攔截所有exception，因為出錯狀況比預期多樣  
                print('抓取失敗:', place, pos, start_fetching_time)  
                continue
```

```
#browser.close() # docker
```