# 巨集程式與資料分析應用
# Macro programming and data analytics

## Section 3
## 網路爬蟲與資料分析
## Lecture 3

陳彥佑

# VD資料讀取並輸出–多個檔案

定義相關參數

宣告儲存VD資料的矩陣

選擇要讀取的檔案

打開XML文件

讀取XML檔案

輸出檔案

```
import xml.etree.ElementTree as ET
import numpy
import os
import csv

year_start = 2022
year_end = 2022
start_month = 1
end_month = 1
start_day = 1
end_day = 1
start_hour = 8
end_hour = 9
start_min = 0
end_min = 59

#宣告檔案存放路徑
xml_file_path = "D:/Program/VD files unzip/“

# 定義命名空間(因為xml檔具有命名空間)
namespace = {"ns": "http://traffic.transportdata.tw/standard/traffic/schema/"}
```

定義相關參數

宣告儲存VD資料的矩陣

選擇

打

讀取XML檔案

輸出檔案

可否改為變數?
目的為何?

```
#宣告陣列來存放資料
speed_matrix = numpy.zeros((60*2, 4)) #[time_index][lane_index]
occ_matrix =  numpy.zeros((60*2, 4)) #[time_index][lane_index]
vol_matrix = numpy.zeros((60*2, 4, 3)) #[time_index][lane_index][vehicle_type_index]
speed_matrix_by_veh_typ = numpy.zeros((60*2, 4, 3))
```

- 定義相關參數
- 宣告儲存VD資料的矩陣
- 選擇要讀取的檔案
- 打開XML文件
- 讀取XML檔案
- 輸出檔案

```
time_index = 0
for year_index in range(year_start, year_end + 1):
    year_str = str(year_index)

    for month_index in range(start_month, end_month + 1):
        if month_index <= 9:
            month_str = "0" + str(month_index)
        else:
            month_str = str(month_index)

        for day_index in range(start_day, end_day + 1):
            if day_index <= 9:
                day_str = "0" + str(day_index)
            else:
                day_str = str(day_index)

            for hour_index in range(start_hour, end_hour + 1):
                if hour_index <= 9:
                    hour_str = "0" + str(hour_index)
                else:
                    hour_str = str(hour_index)

                for min_index in range(start_min, end_min + 1):
                    if min_index <= 9:
                        min_str = "0" + str(min_index)
                    else:
                        min_str = str(min_index)
```

```
xml_file_name = "VDLive_" + year_str + month_str + day_str + hour_str + min_str + ".xml"
```



```
            # 打開XML文件
            tree = ET.parse(os.path.join(xml_file_path, xml_file_name))
            vd_lives = tree.findall("./ns:VDLives/ns:VDLive", namespaces=namespace)

            # 掃描所有VDLive元素，並提取每個Lane下的Speed
            for vd_live in vd_lives:
                vd_id = vd_live.find("./ns:VDID", namespaces=namespace).text
                if vd_id == "VD-N1-N-92.900-M-LOOP":

                    lanes = vd_live.findall("./ns:LinkFlows/ns:LinkFlow/ns:Lanes/ns:Lane", namespaces=namespace)

                    for lane in lanes:

                        lane_id = int(lane.find("ns:LaneID", namespaces=namespace).text)
                        speed_by_lane = int(lane.find("ns:Speed", namespaces=namespace).text)
                        occupancy = int(lane.find("ns:Occupancy", namespaces=namespace).text)
                        speed_matrix[time_index][lane_id] = speed_by_lane #speed_matrix[time_index][車道] [車道:0最內側, 1中間, 2外側, 3路肩]
                        occ_matrix[time_index][lane_id] = occupancy
                        vehicles = lane.findall(".//ns:Vehicle", namespaces=namespace)

                        for vehicle in vehicles:
                            vehicle_type = vehicle.find("ns:VehicleType", namespaces=namespace).text
                            speed_by_veh = int(vehicle.find("ns:Speed", namespaces=namespace).text)
                            volume = int(vehicle.find("ns:Volume", namespaces=namespace).text)

                            #將vehicle type對應到矩陣
                            if vehicle_type == "S":
                                vehicle_type_index = 0
                            elif vehicle_type == "L":
                                vehicle_type_index = 1
                            elif vehicle_type == "T":
                                vehicle_type_index = 2

                            speed_matrix_by_veh_typ[time_index][lane_id][vehicle_type_index] = speed_by_veh
                            vol_matrix[time_index][lane_id][vehicle_type_index] = volume

            time_index += 1
```

```
#輸出標頭
output_file_name = 'VD_output_spd_vol_txt_mul_VD.txt'

with open(os.path.join(xml_file_path, output_file_name), 'w') as outputfile:
    outputfile.write('VDID, time_index, ')
    for lane_index in range(0,4):
        for veh_index in range(0,3):
            if veh_index == 0:
                veh_str = 'S'
            elif veh_index == 1:
                veh_str = 'L'
            elif veh_index == 2:
                veh_str = 'T'

            if lane_index == 3 and veh_index == 2:
                outputfile.write('lane_' + str(lane_index) +'_'+ veh_str + '_spd,')
                outputfile.write('lane_' + str(lane_index) +'_'+ veh_str + '_vol\n')
            else:
                outputfile.write('lane_' + str(lane_index) +'_'+ veh_str + '_spd,')
                outputfile.write('lane_' + str(lane_index) +'_'+ veh_str + '_vol,')
```

```
#輸出矩陣內容
for t_index in range(0, time_index):

    outputfile.write('VD-N1-N-92.900-M-LOOP,' + str(t_index) + ',')

    for lane_index in range(0,4):
        for veh_index in range(0,3):

            if lane_index == 3 and veh_index == 2:
                outputfile.write(str(speed_matrix_by_veh_typ[t_index][lane_index][veh_index]) + ',')
                outputfile.write(str(vol_matrix[t_index][lane_index][veh_index]) + '\n')

            else:
                outputfile.write(str(speed_matrix_by_veh_typ[t_index][lane_index][veh_index]) + ',')
                outputfile.write(str(vol_matrix[t_index][lane_index][veh_index]) + ',')
```

讀取XML檔案

輸出檔案

- 練習：讀取多顆VD資料，存到相對應的.csv檔