

台北市空氣檢測

科二甲 D10619134 李宗儒

說明：

共使用六個檔案，分別是：

臺北市環境空氣品質監測結果-一氧化碳.csv

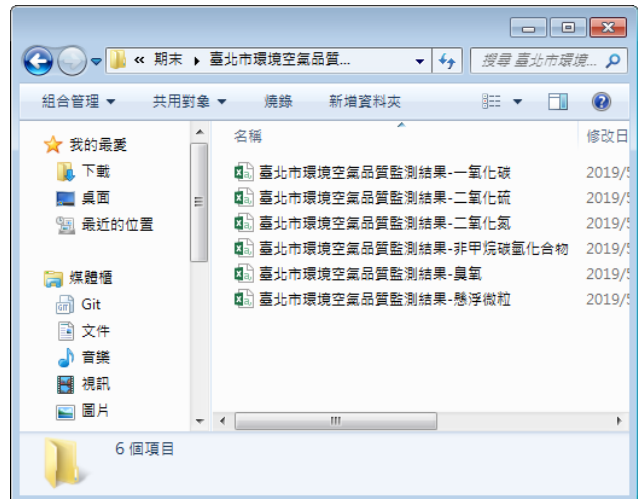
臺北市環境空氣品質監測結果-二氧化硫.csv

臺北市環境空氣品質監測結果-二氧化氮.csv

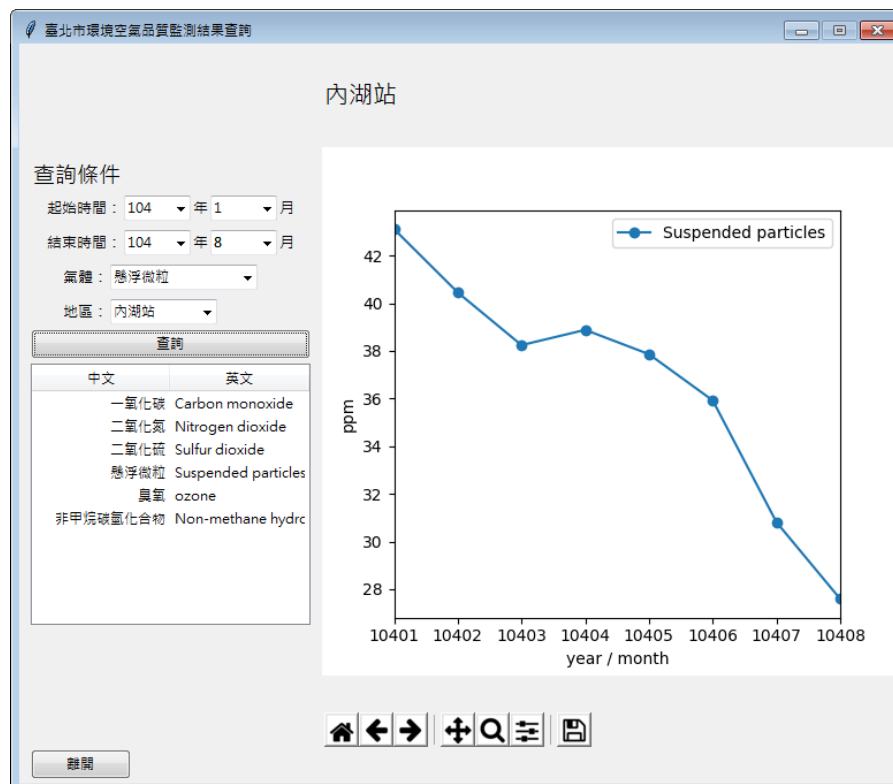
臺北市環境空氣品質監測結果-非甲烷碳氫化合物.csv

臺北市環境空氣品質監測結果-臭氧.csv

臺北市環境空氣品質監測結果-懸浮微粒.csv



可以用表單查詢某一站點的氣體在一段時間內的每用平均值為多少，並且用折線圖顯示出來，在查詢表下方有氣體的中英文對照表。



資料庫設計：

建立六個資料表：

一氧化碳
二氧化硫
二氧化氮
非甲烷碳氫化合物
臭氧
懸浮微粒

欄位分別有：

編號（主鍵） - 整數
年月 - 整數
監測項目 - 字串
監測站 - 字串
監測站編號 - 字串
月平均 - 浮點數

```
SQLQuery1.sql - lo...dministrator (52))* x
create database air
on
primary (
name = 'air',
filename = 'd:\db\air.mdf')
log on (
name = 'air_log',
filename = 'd:\db\air_log.ldf');

use air;

create table 一氧化碳
(編號 int not null,
年月 int not null,
監測項目 varchar(20) null,
監測站 varchar(20) null,
監測站編號 varchar(20) null,
月平均 float null,
primary key (編號));

create table 二氧化硫
(編號 int not null,
年月 int not null,
監測項目 varchar(20) null,
監測站 varchar(20) null,
監測站編號 varchar(20) null,
月平均 float null,
primary key (編號));
```

```
create table 二氧化氮
(編號 int not null,
年月 int not null,
監測項目 varchar(20) null,
監測站 varchar(20) null,
監測站編號 varchar(20) null,
月平均 float null,
primary key (編號));

create table 非甲烷碳氫化合物
(編號 int not null,
年月 int not null,
監測項目 varchar(20) null,
監測站 varchar(20) null,
監測站編號 varchar(20) null,
月平均 float null,
primary key (編號));
```

```
create table 臭氧
(編號 int not null,
年月 int not null,
監測項目 varchar(20) null,
監測站 varchar(20) null,
監測站編號 varchar(20) null,
月平均 float null,
primary key (編號));

create table 懸浮微粒
(編號 int not null,
年月 int not null,
監測項目 varchar(20) null,
監測站 varchar(20) null,
監測站編號 varchar(20) null,
月平均 float null,
primary key (編號));
```

原始程式碼：

插入資料：

右圖的 python 檔能將前面說的六個 CSV 檔寫入到 SQL SERVER 裡。

```
air_插入資料表.py - C:\Users\phil\Downloads\德明\資料庫應用\期末\...
File Edit Format Run Options Window Help
import pyodbc
import csv

def GetData(filename):
    data = []
    with open(filename, newline='') as csvfile:
        rows = csv.reader(csvfile)
        count = 0
        for row in rows:
            data.append(row)
            count += 1
        print('共', count, '筆資料')
    return data

def InsertScore(data, gas):
    connStr = 'DRIVER={ODBC Driver 11 for SQL Server};SERVER=localhost;DATABASE=
    conn = pyodbc.connect(connStr)
    cursor = conn.cursor()
    count = 0
    for row in data:
        #print(row)

        if count != 0:
            sql = 'insert into ' + gas + ' values('
            sql += str(count) + ', '
            sql += row[0][0:3]
            if row[0][5] == '月':
                sql += '0' + row[0][4:5] + ', \''
            else:
                sql += row[0][4:6] + ', \''
            sql += row[1] + ', \''
            sql += row[2] + ', \''
            sql += row[3] + ', \''
            if gas == '非甲烷碳氫化合物':
                sql += row[5] + '))'
            else:
                sql += row[2] + ', \''
                sql += row[3] + ', \''
                if gas == '非甲烷碳氫化合物':
                    sql += row[5] + '))'
                else:
                    sql += row[6] + '))'
            cursor.execute(sql)
            cursor.commit()
            count += 1

    conn.close()
    print('一共轉入', count, '筆資料')

def main():
    data = GetData('臺北市環境空氣品質監測結果/臺北市環境空氣品質監測結果-一氧化
    InsertScore(data, '一氧化碳')

    data = GetData('臺北市環境空氣品質監測結果/臺北市環境空氣品質監測結果-二氧化
    InsertScore(data, '二氧化硫')

    data = GetData('臺北市環境空氣品質監測結果/臺北市環境空氣品質監測結果-二氧化
    InsertScore(data, '二氧化氮')

    data = GetData('臺北市環境空氣品質監測結果/臺北市環境空氣品質監測結果-非甲烷
    InsertScore(data, '非甲烷碳氫化合物')

    data = GetData('臺北市環境空氣品質監測結果/臺北市環境空氣品質監測結果-臭氧.c
    InsertScore(data, '臭氧')

    data = GetData('臺北市環境空氣品質監測結果/臺北市環境空氣品質監測結果-懸浮微
    InsertScore(data, '懸浮微粒')

try:
    main()
except KeyboardInterrupt:
    print('關閉程式')
```

Ln: 35 Col: 33

```
Python 3.6.6 Shell
File Edit Shell Debug Options
Python 3.6.6 (v3.6.6:4c1f54eb7
4) on win32
Type "copyright", "credits" or
>>>
===== RESTART: F
=====
共 177 筆資料
一共轉入 177 筆資料
共 177 筆資料
一共轉入 177 筆資料
共 177 筆資料
一共轉入 177 筆資料
共 67 筆資料
一共轉入 67 筆資料
共 177 筆資料
一共轉入 177 筆資料
共 177 筆資料
一共轉入 177 筆資料
>>>
```

```
air_插入資料表.py - C:\Users\phil\Downloads\德明\資料庫應用\期末\...
File Edit Format Run Options Window Help
        sql += row[2] + ', \''
        sql += row[3] + ', \''
        if gas == '非甲烷碳氫化合物':
            sql += row[5] + '))'
        else:
            sql += row[6] + '))'
        cursor.execute(sql)
        cursor.commit()
        count += 1

    conn.close()
    print('一共轉入', count, '筆資料')

def main():
    data = GetData('臺北市環境空氣品質監測結果/臺北市環境空氣品質監測結果-一氧化
    InsertScore(data, '一氧化碳')

    data = GetData('臺北市環境空氣品質監測結果/臺北市環境空氣品質監測結果-二氧化
    InsertScore(data, '二氧化硫')

    data = GetData('臺北市環境空氣品質監測結果/臺北市環境空氣品質監測結果-二氧化
    InsertScore(data, '二氧化氮')

    data = GetData('臺北市環境空氣品質監測結果/臺北市環境空氣品質監測結果-非甲烷
    InsertScore(data, '非甲烷碳氫化合物')

    data = GetData('臺北市環境空氣品質監測結果/臺北市環境空氣品質監測結果-臭氧.c
    InsertScore(data, '臭氧')

    data = GetData('臺北市環境空氣品質監測結果/臺北市環境空氣品質監測結果-懸浮微
    InsertScore(data, '懸浮微粒')

try:
    main()
except KeyboardInterrupt:
    print('關閉程式')
```

Ln: 35 Col: 33

查詢：

```
air_查詢.py - C:\Users\phill\Downloads\德明\資料庫應用\期末\air_查詢.py (3.7.2)
File Edit Format Run Options Window Help

import math
import csv
import tkinter as tk
from tkinter import ttk
import matplotlib.pyplot as plt #pip install matplotlib

from matplotlib.backends.backend_tkagg import (
    FigureCanvasTkAgg, NavigationToolbar2Tk)
# Implement the default Matplotlib key bindings.
from matplotlib.backend_bases import key_press_handler
from matplotlib.figure import Figure
import numpy as np
import pyodbc

def on_key_press(event):
    print("you pressed {}".format(event.key))
    key_press_handler(event, canvas, toolbar)

def query():
    #取得各選項內容
    start_year = combobox_start_date_year.get()
    start_month = combobox_start_date_month.get()
    end_year = combobox_end_date_year.get()
    end_month = combobox_end_date_month.get()
    gas = combobox_gas.get()
    area = combobox_area.get()

    if len(start_month) < 2:
        start_month = '0' + start_month
    if len(end_month) < 2:
        end_month = '0' + end_month

    print('起始時間：' + start_year + '年' + start_month + '月')
    print('結束時間：' + end_year + '年' + end_month + '月')
    print('氣體：' + gas)
    print('站點：' + area)
    label_area_pic.config(text = area)

    #連線字串
    connStr = 'DRIVER={ODBC Driver 11 for SQL Server};SERVER=localhost;DATABASE=air;Trusted_Connection=yes'

    # 連線
    conn = pyodbc.connect(connStr)

    #成功後，後面的程式都使用conn 這個物件代表資料庫連線
    # 建立cursor，這是資料查詢用的機制
    cursor = conn.cursor()

    sql = 'SELECT * from ' + gas + ' as A ' #要加空白
    sql += 'WHERE A.年月 BETWEEN ' + start_year + start_month + ' and ' + end_year + end_month + ' '
    sql += 'and A.監測站 = \'' + area + '\', '
    sql += 'order by A.編號'

    # 將SQL 命令送去執行，查詢結果放入cursor內
    cursor.execute(sql)

    #將傳出來的結果取出 並逐筆處理
    rows = cursor.fetchall()
    cnt = 0
    year_month = []

    year_month = []
    month_avg = []
    for row in rows:
        print(row.編號, row.年月, row.監測項目, row.監測站, row.監測站編號, row.月平均)
        year_month.append(str(row.年月)[0:3] + str(row.年月)[3:])
        month_avg.append(str(row.月平均))
        cnt += 1
    print('共', cnt, '筆資料')
    print()
    #將時間及濃度加入陣列 (時間，濃度)
    data_1 = []
    for i in range(len(year_month)):
        data_1.append((year_month[i], float(month_avg[i])))

    # x為月份 y為濃度
    x = [p[0] for p in data_1]
    y = [p[1] for p in data_1]

    #畫線
    plt.clf()
    l1 = plt.plot(x, y, '-o')
    #l2 = plt.plot(x2, y2, color='red', linewidth=1.0, linestyle='--', label='square line')
    plt.legend(handles=[l1], labels=[gas_eng[gas]], loc='best')

    #set x limits x軸最小最大值
    plt.xlim((0, year_month[len(year_month)-1]))
    #plt.ylim((month_avg[0], month_avg[len(month_avg)-1]))
    #x y 軸名稱
    plt.xlabel('year / month')
    plt.ylabel('ppm')
Ln: 89 Col: 12

air_查詢.py - C:\Users\phill\Downloads\德明\資料庫應用\期末\air_查詢.py (3.7.2)
File Edit Format Run Options Window Help

if len(end_month) < 2:
    end_month = '0' + end_month

print('起始時間：' + start_year + '年' + start_month + '月')
print('結束時間：' + end_year + '年' + end_month + '月')
print('氣體：' + gas)
print('站點：' + area)
label_area_pic.config(text = area)

#連線字串
connStr = 'DRIVER={ODBC Driver 11 for SQL Server};SERVER=localhost;DATABASE=air;Trusted_Connection=yes'

# 連線
conn = pyodbc.connect(connStr)

#成功後，後面的程式都使用conn 這個物件代表資料庫連線
# 建立cursor，這是資料查詢用的機制
cursor = conn.cursor()

sql = 'SELECT * from ' + gas + ' as A ' #要加空白
sql += 'WHERE A.年月 BETWEEN ' + start_year + start_month + ' and ' + end_year + end_month + ' '
sql += 'and A.監測站 = \'' + area + '\', '
sql += 'order by A.編號'

# 將SQL 命令送去執行，查詢結果放入cursor內
cursor.execute(sql)

#將傳出來的結果取出 並逐筆處理
rows = cursor.fetchall()
cnt = 0
year_month = []

year_month = []
month_avg = []
for row in rows:
    print(row.編號, row.年月, row.監測項目, row.監測站, row.監測站編號, row.月平均)
    year_month.append(str(row.年月)[0:3] + str(row.年月)[3:])
    month_avg.append(str(row.月平均))
    cnt += 1
print('共', cnt, '筆資料')
print()
#將時間及濃度加入陣列 (時間，濃度)
data_1 = []
for i in range(len(year_month)):
    data_1.append((year_month[i], float(month_avg[i])))

# x為月份 y為濃度
x = [p[0] for p in data_1]
y = [p[1] for p in data_1]

#畫線
plt.clf()
l1 = plt.plot(x, y, '-o')
#l2 = plt.plot(x2, y2, color='red', linewidth=1.0, linestyle='--', label='square line')
plt.legend(handles=[l1], labels=[gas_eng[gas]], loc='best')

#set x limits x軸最小最大值
plt.xlim((0, year_month[len(year_month)-1]))
#plt.ylim((month_avg[0], month_avg[len(month_avg)-1]))
#x y 軸名稱
plt.xlabel('year / month')
plt.ylabel('ppm')
Ln: 31 Col: 26

air_查詢.py - C:\Users\phill\Downloads\德明\資料庫應用\期末\air_查詢.py (3.7.2)
File Edit Format Run Options Window Help

year_month = []
month_avg = []
for row in rows:
    print(row.編號, row.年月, row.監測項目, row.監測站, row.監測站編號, row.月平均)
    year_month.append(str(row.年月)[0:3] + str(row.年月)[3:])
    month_avg.append(str(row.月平均))
    cnt += 1
print('共', cnt, '筆資料')
print()
#將時間及濃度加入陣列 (時間，濃度)
data_1 = []
for i in range(len(year_month)):
    data_1.append((year_month[i], float(month_avg[i])))

# x為月份 y為濃度
x = [p[0] for p in data_1]
y = [p[1] for p in data_1]

#畫線
plt.clf()
l1 = plt.plot(x, y, '-o')
#l2 = plt.plot(x2, y2, color='red', linewidth=1.0, linestyle='--', label='square line')
plt.legend(handles=[l1], labels=[gas_eng[gas]], loc='best')

#set x limits x軸最小最大值
plt.xlim((0, year_month[len(year_month)-1]))
#plt.ylim((month_avg[0], month_avg[len(month_avg)-1]))
#x y 軸名稱
plt.xlabel('year / month')
plt.ylabel('ppm')
Ln: 91 Col: 21
```

```
air_查詢.py - C:\Users\phil\Downloads\德明\資料庫應用\期末\air_查詢.py (3.7.2)
File Edit Format Run Options Window Help
plt.ylabel('ppm')

plt.gcf().canvas.draw()

conn.close()

def _quit():
    root.quit() # stops mainloop
    root.destroy() # this is necessary on Windows to prevent
                  # Fatal Python Error: PyBval_RestoreThread: NULL tstate

try:
    print('Press Ctrl-C to exit.')

    data = []
    data.append([104, 105])
    data.append([i for i in range(1, 13)])
    data.append(['二氧化硫', '一氧化碳', '二氧化氮', '非甲烷碳氫化合物', '臭氧', '懸浮微粒'])
    data.append(['中正站', '大直站', '信義站', '南港站', '內湖站', '木柵站', '承德站'])

    #creatWindow
    root = tk.Tk()
    root.title('臺北市環境空氣品質監測結果查詢')
    #root.width = 500
    #root.height = 200
    root.resizable(0,0) #lock size

    #search area 搜尋區
    frm_form = tk.Frame(root)
    frm_form.pack(side=tk.LEFT, expand=1)

Ln: 91 Col: 21

air_查詢.py - C:\Users\phil\Downloads\德明\資料庫應用\期末\air_查詢.py (3.7.2)
File Edit Format Run Options Window Help
frm_form.pack(side=tk.LEFT, expand=1)

frm_search = tk.Frame(frm_form)
frm_search.pack(side=tk.TOP, padx=10, pady=100, expand=1)

label_title = ttk.Label(frm_search, text="查詢條件", font=("微軟正黑體", 14))
label_title.pack(side=tk.TOP, fill=tk.X, expand=1)

#date area 時間選擇
#start time 開始時間
frm_date_from = tk.Frame(frm_search)
frm_date_from.pack(side=tk.TOP, pady=4, expand=1)
label_date_from = ttk.Label(frm_date_from, text="起始時間:", font=("微軟正黑體", 10))
label_date_from.pack(side=tk.LEFT, fill=tk.X, expand=1)

#choice 下拉式選擇
date = sorted(set(i for i in data[0]))
month = sorted(set(i for i in data[1]))
combobox_start_date_year = ttk.Combobox(frm_date_from, width=5, value=date, state='readonly')
combobox_start_date_year.set(date[0])
combobox_start_date_year.pack(side=tk.LEFT, fill=tk.X, expand=1)
label_date_year = ttk.Label(frm_date_from, text="年", font=("微軟正黑體", 10))
label_date_year.pack(side=tk.LEFT, fill=tk.X, expand=1)

combobox_start_date_month = ttk.Combobox(frm_date_from, width=5, value=month, state='readonly')
combobox_start_date_month.set(month[0])
combobox_start_date_month.pack(side=tk.LEFT, fill=tk.X, expand=1)
label_date_month = ttk.Label(frm_date_from, text="月", font=("微軟正黑體", 10))
label_date_month.pack(side=tk.LEFT, fill=tk.X, expand=1)

#end time 結束時間
frm_date_to = tk.Frame(frm_search)

Ln: 121 Col: 41

air_查詢.py - C:\Users\phil\Downloads\德明\資料庫應用\期末\air_查詢.py (3.7.2)
File Edit Format Run Options Window Help
frm_date_to = tk.Frame(frm_search)
frm_date_to.pack(side=tk.TOP, pady=4, expand=1)
label_date_to = ttk.Label(frm_date_to, text="結束時間:", font=("微軟正黑體", 10))
label_date_to.pack(side=tk.LEFT, fill=tk.X, expand=1)

#choice 下拉式選擇
combobox_end_date_year = ttk.Combobox(frm_date_to, width=5, value=date, state='readonly')
combobox_end_date_year.set(date[0])
combobox_end_date_year.pack(side=tk.LEFT, fill=tk.X, expand=1)
label_date_year = ttk.Label(frm_date_to, text="年", font=("微軟正黑體", 10))
label_date_year.pack(side=tk.LEFT, fill=tk.X, expand=1)

combobox_end_date_month = ttk.Combobox(frm_date_to, width=5, value=month, state='readonly')
combobox_end_date_month.set(month[0])
combobox_end_date_month.pack(side=tk.LEFT, fill=tk.X, expand=1)
label_date_month = ttk.Label(frm_date_to, text="月", font=("微軟正黑體", 10))
label_date_month.pack(side=tk.LEFT, fill=tk.X, expand=1)

#gas choice 氣體選擇
frm_gas = tk.Frame(frm_search)
frm_gas.pack(side=tk.TOP, anchor="w", pady=4, expand=1)
label_gas = ttk.Label(frm_gas, text="氣體:", font=("微軟正黑體", 10))
label_gas.pack(side=tk.LEFT, fill=tk.X, expand=1)

gas = sorted(set(i for i in data[2]))
combobox_gas = ttk.Combobox(frm_gas, width=15, value=gas, state='readonly')
combobox_gas.set(gas[0])
combobox_gas.pack(side=tk.LEFT, fill=tk.X, expand=1)

#area choice 地區選擇
frm_area = tk.Frame(frm_search)
frm_area.pack(side=tk.TOP, anchor="w", pady=4, expand=1)

Ln: 151 Col: 38
```

```
air_查詢.py - C:\Users\phill\Downloads\傳明資料庫應用\期末\air_查詢.py (3.7.2)
File Edit Format Run Options Window Help
frm_area.pack(side=tk.TOP, anchor="w", pady=4, expand=1)
label_area = ttk.Label(frm_area, text="地區:", font=("微軟正黑體", 10))
label_area.pack(side=tk.LEFT, fill=tk.X, expand=1)

area = sorted(set(i for i in data[3]))
combobox_area = ttk.Combobox(frm_area, width=10, value=area, state='readonly')
combobox_area.set(area[0])
combobox_area.pack(side=tk.LEFT, fill=tk.X, expand=1)

#submit button 查詢按鈕
button_query = ttk.Button(master=frm_search, text="查詢", command = _query)
button_query.pack(side=tk.TOP, fill=tk.X, expand=1)

#gas table 氣體中英文表
frm_gas_chi_eng = tk.Frame(frm_search)
frm_gas_chi_eng.pack(side=tk.TOP, anchor="w", pady=4, expand=1)

gas_eng = {
    '一氧化碳': 'Carbon monoxide',
    '二氧化硫': 'Sulfur dioxide',
    '二氧化氮': 'Nitrogen dioxide',
    '非甲烷碳氫化合物': 'Non-methane hydrocarbon',
    '臭氧': 'ozone',
    '懸浮微粒': 'Suspended particles'
}

chi_eng = ['中文', '英文']
tree = ttk.Treeview(frm_gas_chi_eng, columns=('c1', 'c2', 'c3', 'c4', 'c5', 'c6'),
                    show="headings")#表格
tree["columns"]=(('中文', '英文'))
tree.column('中文', width=120, anchor='e') #表示列,不顯示
tree.column('英文', width=120, anchor='w')
tree.heading('中文',text='中文') #顯示表頭

Ln: 181 Col: 60

air_查詢.py - C:\Users\phill\Downloads\傳明資料庫應用\期末\air_查詢.py (3.7.2)
File Edit Format Run Options Window Help
tree.heading('中文',text='中文') #顯示表頭
tree.heading('英文',text='英文')

for i in range(len(gas)):
    tree.insert("", i, values=(gas[i], gas_eng[gas[i]])) #插入數據

tree.pack(side=tk.TOP, expand=1)

#quit button 離開按鈕
frm_quit = tk.Frame(frm_form)
frm_quit.pack(side=tk.TOP, anchor="w", pady=4, expand=1)
button_quit = ttk.Button(master=frm_quit, text="離開", command = _quit)
button_quit.pack(side=tk.BOTTOM, fill=tk.X, padx=10, expand=1)

#pic area 圖表
frm_pic = tk.Frame(root)
frm_pic.pack(side=tk.LEFT, fill=tk.Y, expand=1)

label_area_pic = ttk.Label(frm_pic, text=combobox_area.get(), font=("微軟正黑體", 16))
label_area_pic.pack(side=tk.TOP, fill=tk.BOTH, expand=1)

fig = plt.figure(figsize=(5, 4), dpi=100)

canvas = FigureCanvasTkAgg(fig, master=frm_pic) # A tk.DrawingArea.
canvas.draw()
canvas.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH, expand=1)

toolbar = NavigationToolbar2Tk(canvas, frm_pic)
toolbar.update()
toolbar.pack(side=tk.TOP, fill=tk.BOTH, expand=1)

Ln: 211 Col: 39

air_查詢.py - C:\Users\phill\Downloads\傳明資料庫應用\期末\air_查詢.py (3.7.2)
File Edit Format Run Options Window Help
frm_quit.pack(side=tk.TOP, anchor="w", pady=4, expand=1)
button_quit = ttk.Button(master=frm_quit, text="離開", command = _quit)
button_quit.pack(side=tk.BOTTOM, fill=tk.X, padx=10, expand=1)

#pic area 圖表
frm_pic = tk.Frame(root)
frm_pic.pack(side=tk.LEFT, fill=tk.Y, expand=1)

label_area_pic = ttk.Label(frm_pic, text=combobox_area.get(), font=("微軟正黑體", 16))
label_area_pic.pack(side=tk.TOP, fill=tk.BOTH, expand=1)

fig = plt.figure(figsize=(5, 4), dpi=100)

canvas = FigureCanvasTkAgg(fig, master=frm_pic) # A tk.DrawingArea.
canvas.draw()
canvas.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH, expand=1)

toolbar = NavigationToolbar2Tk(canvas, frm_pic)
toolbar.update()
toolbar.pack(side=tk.TOP, fill=tk.BOTH, expand=1)

canvas.mpl_connect("key_press_event", on_key_press)

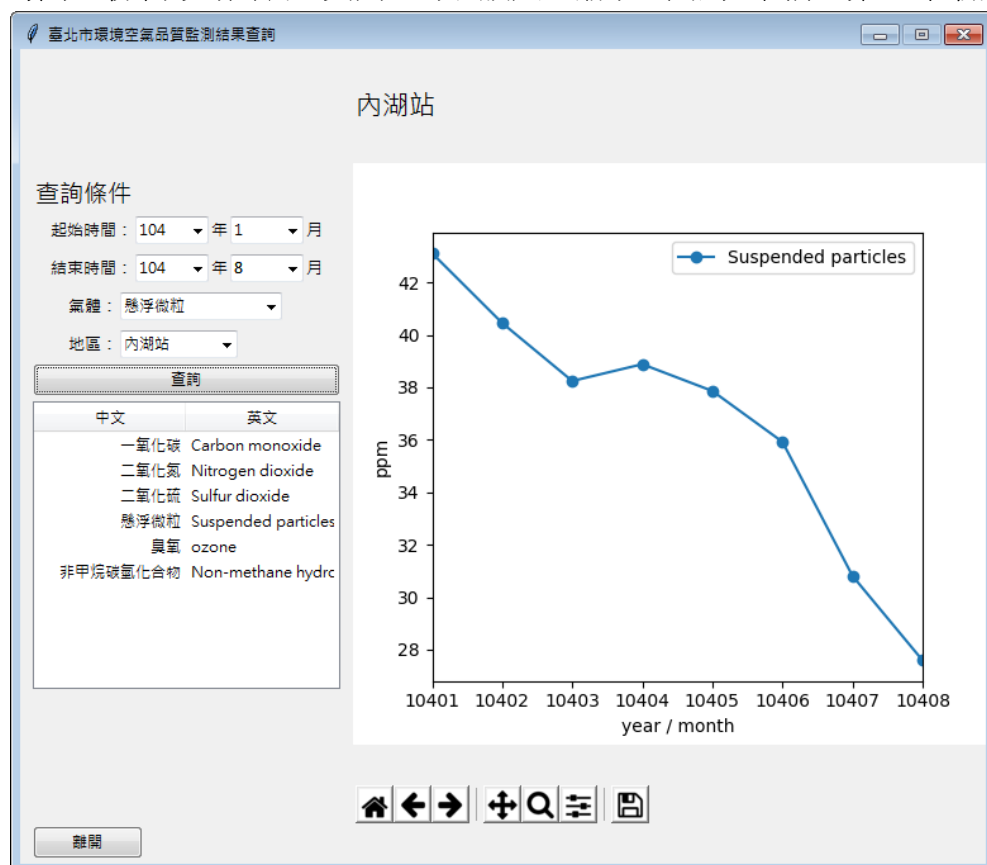
#show window
root.mainloop()

except:
    print('Program is stop.')
```

執行畫面：

下圖將 104 年 1 月到 104 年 8 月在內湖站測到的「懸浮微粒」濃度畫出來，並且會在 python shell 中顯示查詢條件及查詢結果。

左側為查詢區，可選時間區段、監測氣體、監測站和氣體中英文對照表，右側為圖表區，圖表最上方會顯示站名；圖表左側 y 軸表示濃度，下方 x 軸表示時間；最下方為圖表工具列，可以放大、縮小、回到一開始的位置和儲存圖檔。

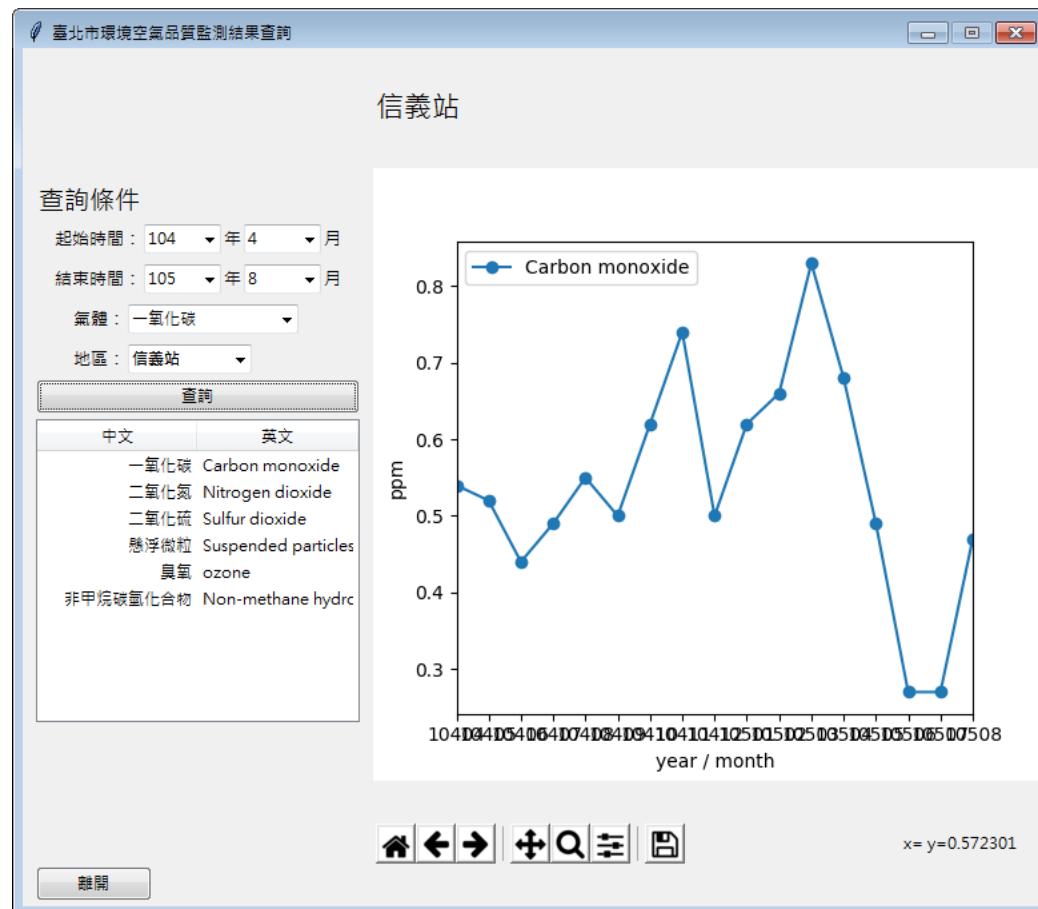


```
*Python 3.6.6 Shell*
File Edit Shell Debug Options Window Help
141 10506 懸浮微粒 內湖站 2534216A0011 28.4
149 10507 懸浮微粒 內湖站 2534216A0011 31.3
157 10508 懸浮微粒 內湖站 2534216A0011 28.94
165 10509 懸浮微粒 內湖站 2534216A0011 29.92
173 10510 懸浮微粒 內湖站 2534216A0011 26.68
共 10 筆資料

起始時間：104年01月
結束時間：104年08月
氣體：懸浮微粒
站點：內湖站
5 10401 懸浮微粒 內湖站 2534216A0011 43.11
13 10402 懸浮微粒 內湖站 2534216A0011 40.45
21 10403 懸浮微粒 內湖站 2534216A0011 38.24
29 10404 懸浮微粒 內湖站 2534216A0011 38.88
37 10405 懸浮微粒 內湖站 2534216A0011 37.86
45 10406 懸浮微粒 內湖站 2534216A0011 35.92
53 10407 懸浮微粒 內湖站 2534216A0011 30.81
61 10408 懸浮微粒 內湖站 2534216A0011 27.59
共 8 筆資料

Ln: 91 Col: 35
```

下圖為104年4月到105年8月在信義站測到的「一氧化碳」濃度，查詢可以跨年份，但是月份太多導致軸上的字擠在一起。



Python 3.6.6 Shell

File Edit Shell Debug Options Window Help

共 17 筆資料

起始時間：104年04月
 結束時間：105年08月
 氣體：一氧化碳
 站點：信義站

27	10404	一氧化碳	信義站	2534516A0004	0.54
35	10405	一氧化碳	信義站	2534516A0004	0.52
43	10406	一氧化碳	信義站	2534516A0004	0.44
51	10407	一氧化碳	信義站	2534516A0004	0.49
59	10408	一氧化碳	信義站	2534516A0004	0.55
67	10409	一氧化碳	信義站	2534516A0004	0.5
75	10410	一氧化碳	信義站	2534516A0004	0.62
83	10411	一氧化碳	信義站	2534516A0004	0.74
91	10412	一氧化碳	信義站	2534516A0004	0.5
99	10501	一氧化碳	信義站	2534516A0004	0.62
107	10502	一氧化碳	信義站	2534516A0004	0.66
115	10503	一氧化碳	信義站	2534516A0004	0.83
123	10504	一氧化碳	信義站	2534516A0004	0.68
131	10505	一氧化碳	信義站	2534516A0004	0.49
139	10506	一氧化碳	信義站	2534516A0004	0.27
147	10507	一氧化碳	信義站	2534516A0004	0.27
155	10508	一氧化碳	信義站	2534516A0004	0.47

共 17 筆資料

Ln: 91 Col: 35

總結：

一開始做這個小專題時，就想用圖形介面來做，然後看 `tkinter` 是 `python` 內建的圖形介面，而且容易上手使用，所以就決定用 `tkinter`，但是用 `pack` 布局是照順序丟上視窗裡，所以為了排版要分成一個一個小區域來放，但是缺點是某個東西建失敗或有錯誤，後面的東西也不會出來，而且也不會顯示哪裡錯誤，除厝上不太方便。

然後圖表要用 `matplotlib` 來做，這個圖表有很多種圖表類型可以選，而折線圖較符合濃度變化的比較，使用上也很簡單，只要將需要的 `x y` 軸放入就可以，放入陣列可以直接畫出折線圖，需要的程式碼非常少，方法的功能都包好了，非常方便。

在做完看到圖表可以成功畫出來，沒有出現錯誤，非常有成就感；而且畫出來的成果也很好看。