修平科技大學

資訊網路技術系實務專題

Python 爬蟲程式應用

指導老師：沈良澤

學生：陳膺任 BN107007

張浩瑜 BN107046

中華民國 111 年 6 月 14 日

資訊網路技術系實務專題

Python 爬蟲程式應用

學生：陳膺任 BN107007

張浩瑜 BN107046

指導老師：_____老師

評審老師:_____老師

_____老師

_____老師

中華民國 111 年 6 月 14 日

摘要

在這個網際網路發展快速的時代，資訊採集是很重要的工作，人們已經不用像以前一樣在成堆的書中尋找資料，但隨著網際網路的興起，網路中的資料也變得多而複雜，如何高效且快速的獲取網際網路中我們需要的資訊成為了一個重要的問題，在這當中就顯現出網路爬蟲在這個大資訊時代對人們來說是多麼的重要，這也是我們製作此程式的原因之一。

網路爬蟲（web crawler），也叫網路蜘蛛。簡單來說網路爬蟲就是利用爬蟲軟體自動抓取網站上的資料，搜尋引擎通過爬蟲軟體更新自身的網站內容或其對其他網站的索引，以便於搜尋引擎使用者搜尋。

本專題以網路爬蟲之技術為基礎，以 requests 及 Beautiful Soup 等擴充包來訪問網站獲取當中的 HTML 資料，並解析原始的 HTML 程式碼，以便提取網站中的資料， 再將這些資料整合為類似目錄形式的小程式，使其在使用方面更為直覺，讓使用者查找公告時能快速掌握資訊，節省時間成本。

目錄

第一章 前言

1-1 動機與目的

動機

日常生活中，人們時常使用總統府的網站去查詢總統府的資料，但當我們點開總統府網站時，卻因為各種訊息的分類、種類繁多，不知道該從哪裡下手。在這個資訊繁雜的時代，想要在網上查到我們真正所需的資料，可能就要耗費你大量的時間，所以我們希望能建立一個較為便利的程式，讓使用者能快速且精確的找到想要查詢的資料或公告。

目的

日常生活中，人們時常使用總統府的網站去查詢各行政機關的資料，但當我們點開總統府網站時，卻因為各種訊息的分類、種類繁多，不知道該從哪裡下手。在這個資訊繁雜的時代，想要在網上查到我們真正所需的資料，可能就要耗費你大量的時間，所以我們希望能建立一個較為便利的程式，讓使用者能快速且精確的找到想要查詢的資料或公告。

1-2 軟硬體需求

我們需要用到的軟體是 Python idle 編輯器，這是一個 Python 的整合開發環境。我們需要先到 www.python.org 中找到 Downloads 區點選進去，然後找到適合的版本安裝包進行下載。

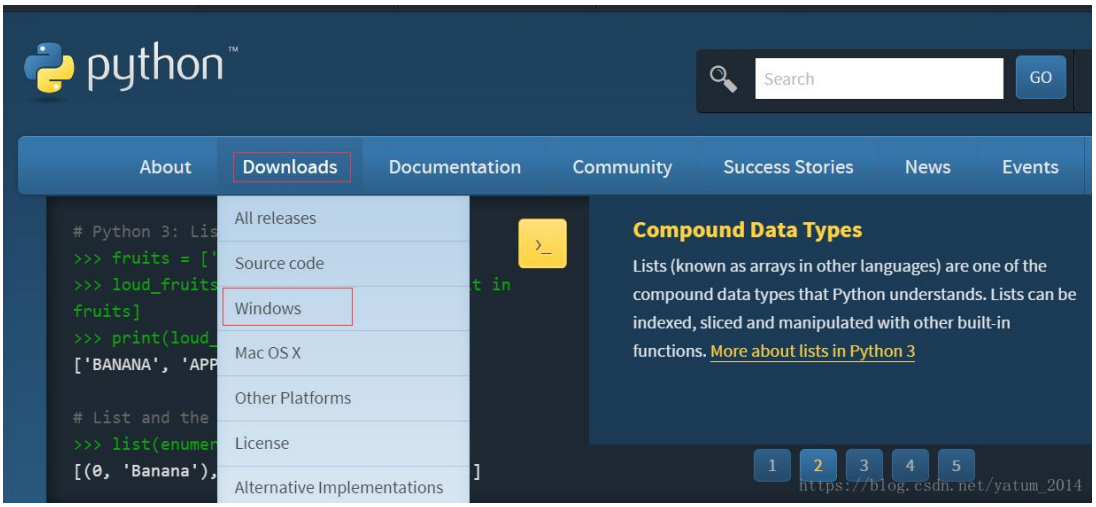

圖 1-1 www.python.org 網站

當下載完後點擊剛下載 Python idle 安裝包，執行安裝程式。並不用做過多的設定，直接點選安裝程式和下一步直到安裝成功便可。

而且我們此程式需要用到不少的 Python 擴充包，其中包含了：bs4, certifi, chardet, idna, requests, soupsieve, urllib3。要安裝這些擴充包我們需要先用系統管理員開啟命令提示字元

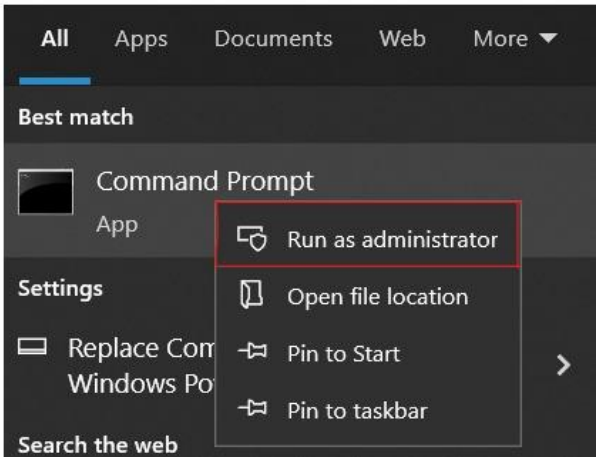

圖 1-2 開啟命令提示字元

開啟後輸入 pip install _____ 來安裝指定擴充包，如下圖：

圖 1-3 安裝指定擴充包

有些擴充包需要指定版本，而要安裝之定的版本要輸入以下指令

圖 1-4 安裝指定擴充包版本

以下擴充包是需要這些指定版本的：

1. beautifulsoup4 4.9.3　　2. Certifi 2021.5.30

3. Idna 3.2　　4. Requests 2.25.1

5. Soupsieve 2.2.1　　6. Urllib3 1.26.5

如要知道是否有安裝這些擴充包，可在命令提示字元輸入 pip list,

這樣被可以看到是否有沒有安裝到這些擴充包了

第二章 文獻探討

2－1 Python

Python 是一種被廣泛使用且易於學習的直譯式、通用和進階的程式語言。(由 Guido van Rossum 所創)，由於當時其他的編成語言都各有缺點，Guido van Rossum 就想自己創建編成語言，當時對於 Python 的設計哲學也很簡單，Python 設計哲學強調程式碼的可讀性與簡潔的語法，讓其能利用簡單但有效方法去完成物件導向的程式設計，由於其優雅的語法和動態型別，與其直譯的特殊性質，使其成為眾多平台上許多領域程式編寫和快速應用程序開發時的理想程式語言。也因為 Python 的功能強大及應用範圍廣，讓全球許多知名企業、和各領域的專家，紛紛投入開發，下圖簡單列出一些生活當中常見的應用及著名企業，都是利用 Python 來開發和建構。

使用 Python 編寫的著名應用

NETFLIX-網路隨選串流影片的 OTT 服務公司

Reddit-娛樂、社交及新聞網站

Dropbox-檔案儲存服務

YouTube-影音社群網站

Facebook-社交軟體

Amazon-電子商務平台

Spotify-音樂串流服務平台

Yahoo-網際網路服務提供商

Trivago-飯店比價搜尋引擎

NOKIA-生產行動通訊裝置和服務的公司

Instagram--一款免費提供線上圖片及影片分享的社群應用軟體

圖 2- 1 使用 Python 編寫應用

Python 優點

1. 語法簡單

2. 工法完整

3. 應用廣泛

4. 豐富的函式庫

2—2 網路爬蟲

網路爬蟲（web crawler），也叫網路蜘蛛，簡單來說是一種網路機器人它能代替人們在這繁雜的網際網路中進行資訊的採集及資料的整理。其目的一般為編寫網路索引。

網路爬蟲的起源最早可以追溯到Google搜索引擎的誕生，各搜索引擎基本離不開爬蟲，例如百度的爬蟲「百度蜘蛛」，其爬蟲每天會在大量的網際網路資料中進行爬取並收錄，當使用者搜索關鍵字時，百度會對關鍵字進行分析處理，從收錄的網頁中推送給使用者相關網頁並按照一定的排序規則展示出來。

2—3 網路爬蟲合法性探討

因網路爬蟲的興起，說起爬內容爬數據，大家或多或少都能明白這個動作的意義，雖然這已經是許多人賴以為生的技術了，但近幾年因網路爬蟲所引發的法律爭議卻越來越多，你是否有想過，網路爬蟲的行為真的合法嗎？

網路爬蟲領域目前還屬於拓荒階段，雖然網際網路世界已經通過自己的遊戲規則建立起一定的道德規範(Robots 協議，全稱是“網路爬蟲排除標準”)，但法律部分仍在進一步建立和完善中，也就是說，現在這個領域暫時還是灰色地帶。

以美國為例，法院規制數據爬蟲的法律途徑主要有四種：

- 非法入侵私人財產（trespass to chattels）；
- 合約違約（breach of contract）；
- 違反著作權（copyright violations）；
- 違反《電腦欺詐和濫用法》（CFAA violations）。

其中，CFAA 在實務中被廣泛援用是近年來的一大趨勢。

這一法案對「故意未經授權或超越授權存取電腦訊息系統並因此從任何受保護的電腦獲取訊息」的行為創設了民事和刑事責任。

美國最高法院解釋，CFAA 規定了兩類非法存取受保護的電腦訊息系統進而構成犯罪的行為：

- 未經授權存取。
- 雖獲得授權存取但不當使用。

因此，雖然爬蟲本身在法律上並不被禁止，但是利用爬蟲技術獲取數據的行為還是具有違法風險的，就如同菜刀在法律上並不禁止使用，但是你如果用來砍人，那就不被法律所容忍了。

2－4 Tkinter

Tkinter 是 TK GUI 整合到 Python 裡的 GUI 開發套件，是一個 Python 模組，用於製作 Python 專案的操作介面，Tkinter 雖然功能簡單，但因為是 Python 自帶的 GUI 套件，效能相對於整合度高的套件 pyqt 更好，如果需求是簡單的介面設計更好的效率那 Tkinter 會是一個不錯的選擇

2－5 Beautiful Soup

Beautiful Soup 是一個 Python 的函式庫模組可使開發者僅需撰寫少量的程式碼就可快速的解析網頁 HTML 碼從中提取處對使用者有用的資料大大增加了撰寫爬蟲程式的速度，也降低了網路爬蟲程式的開發門檻。

第三章 系統功能

3-1 工作進度

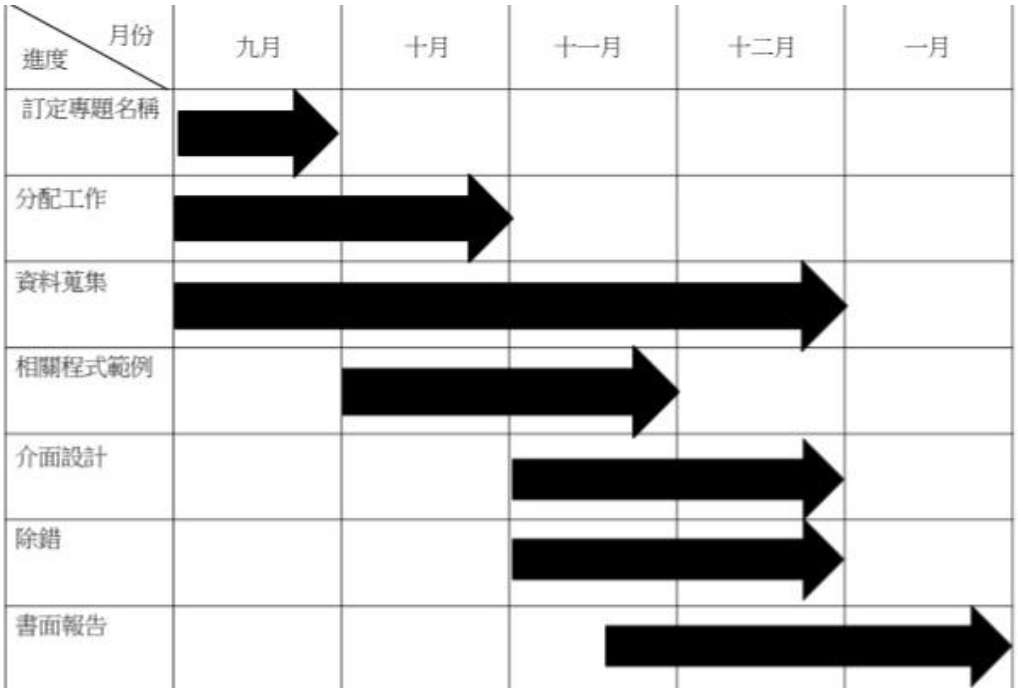

圖 3-1 專題工作進度

3-2 系統功能圖

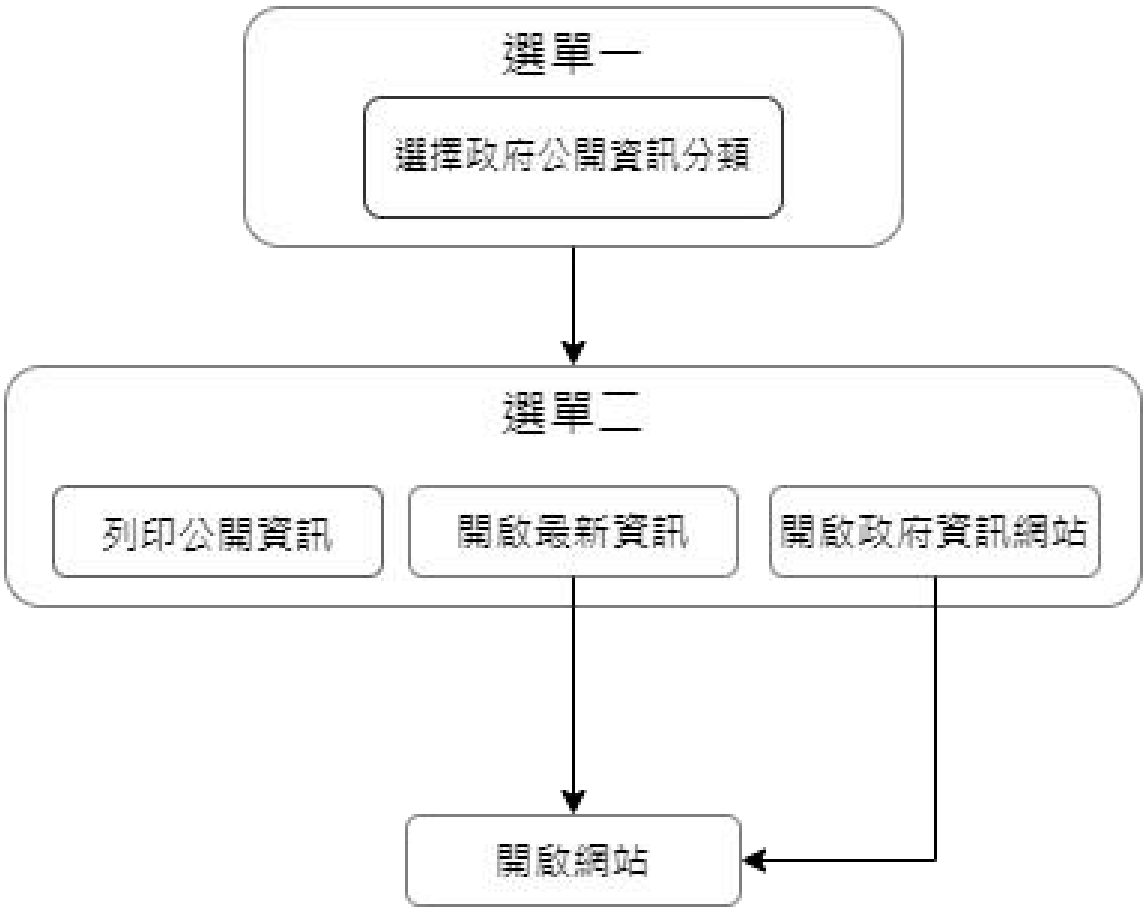

圖 3-2 系統功能圖

此程式會先出來一個視窗選單以讓你選擇政府的資訊分類列表。

當你選擇了某一分類後,它便會出現第二個視窗並簡單的列印出所有此分類中的公開資訊，以讓你尋找是否有你所需要的資訊。

如果有你需要的資訊在此分類中，你可以選擇“開啟政府資訊網站”以快速的進入此分類的網站,而如果你只是想觀看此分類中的最新資訊，你可以點選“開啟最新資訊”來開啟最新資訊的網站直接觀看。

3-3 系統架構圖

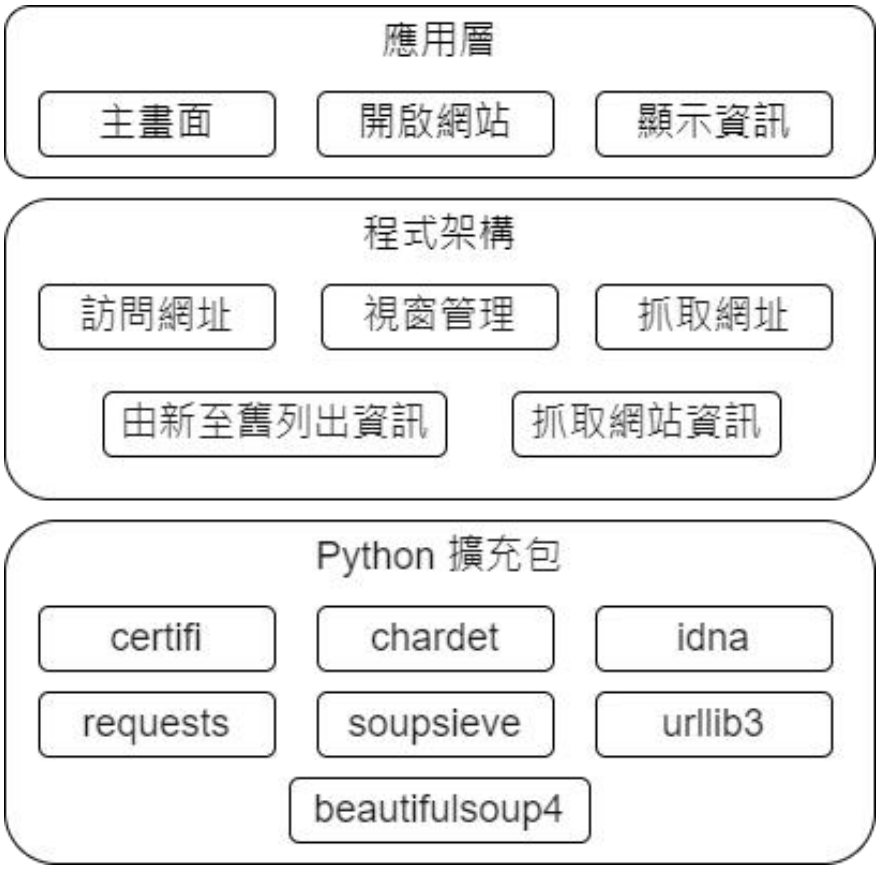

圖 3-3 系統架構圖

此程式的系統架構分為三個部分，分別為 Python 擴充包，程式架構和應用層這三個部分。

其中最為核心部分是 Python 的擴充包，因為這些擴充包的功能是構成此程式可以運作的關鍵；程式架構的部分是用來說明在所有的程式中分別都負責什麼功能；而應用層則是我們希望此程式所能呈現出來並順利運行的功能。

3-4 系統流程圖

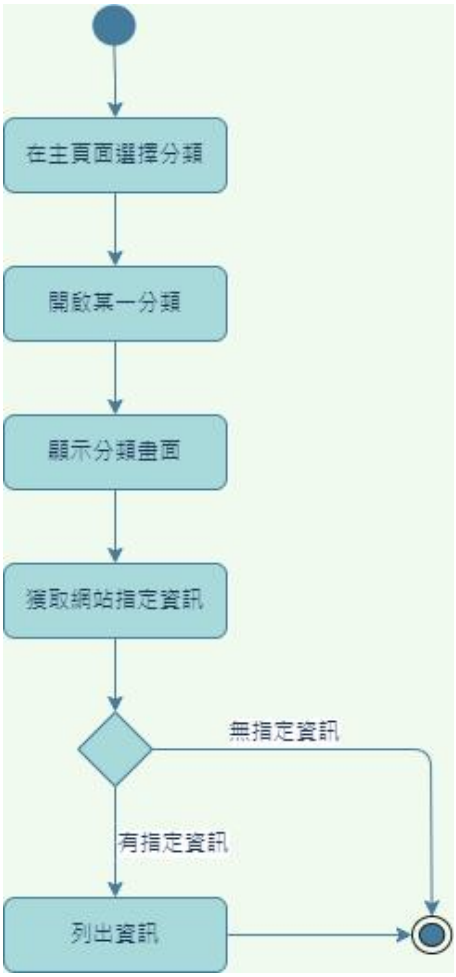

圖 3-4 系統流程圖

此系統的運作流程在開始時會先開啟主畫面讓你選擇你所想要的資訊分類，當你選擇分類後就會開啟指定分類畫面並開始抓取指定網站中的資訊。如果網站中沒能抓取到指定資訊的話就會直接顯示畫面且不會列印任何的資訊，而有抓取到指定資訊的話就會在畫面中簡單的列印出所有抓取到的資訊。

第四章 程式說明

4-1 主畫面程式

```
import tkinter as tk
import os
from layout import define_layout

#以下為每一個按鈕所對應的運行結果

def B1():
    os.popen('python A1.py')

def B2():
    os.popen('python A2.py')


def B3():
    os.popen('python A3.py')
```

圖 4-1 主畫面程式一

我們在主畫面用了 tkinter 擴充包來做圖形界面，並用 OS 擴充包來開啟其他的 python 程式。

```
def define_layout(obj, cols=1, rows=1):
    def method(trg, col, row):

        for c in range(cols):
            trg.columnconfigure(c, weight=1)
        for r in range(rows):
            trg.rowconfigure(r, weight=1)

    if type(obj) == list:
        [method(trg, cols, rows) for trg in obj]
    else:
        trg = obj
        method(trg, cols, rows)
```

圖 4-2 主畫面程式二

副函式 define_layout 是用來定義 grid 並計算需要用到多少欄多少列，col 為欄，row 為列。

下圖是 tkinter 的程式內容，主要設定界面的配置。

```
win= tk.Tk()                                #定義一個視窗名叫 win
win.title('請選擇以下政府公開資料')          #設定標題名
win.geometry('400x500')                     #設定視窗大小
win.resizable(0, 0)                         #固定視窗的大小
align_mode = 'nesw'                         #來統一其對其方式，字串'nswe'是置中的意思
pad = 3                                     #pad是邊與邊之間的距離

div_side = 500
divl = tk.Frame(win, width=div_side, height=div_side)

win.update()
win_size = min(win.winfo_width(), win.winfo_height())

divl.grid(row=0, padx=pad, pady=pad, sticky=align_mode)

define_layout(win)          #呼叫副函式 ( define_layout )
```

圖 4-3 tkinter 程式一

```
bt1 = tk.Button(divl, text='訊息公告', bg='white', command=B1, font=('Helvetica', '12'))
bt2 = tk.Button(divl, text='人事公告', bg='Gainsboro', command=B2, font=('Helvetica', '12'))
bt3 = tk.Button(divl, text='業務統計及研究報告', bg='white', command=B3, font=('Helvetica', '12'))
bt4 = tk.Button(divl, text='預算、決算書', bg='Gainsboro', command=B4, font=('Helvetica', '12'))
bt5 = tk.Button(divl, text='接收及支付補助金', bg='white', command=B5, font=('Helvetica', '12'))
bt6 = tk.Button(divl, text='總統府採購資訊', bg='Gainsboro', command=B6, font=('Helvetica', '12'))
bt7 = tk.Button(divl, text='個人資料保護', bg='white', command=B7, font=('Helvetica', '12'))
bt8 = tk.Button(divl, text='訴願決定書', bg='Gainsboro', command=B8, font=('Helvetica', '12'))
bt9 = tk.Button(divl, text='政策宣導相關廣告', bg='white', command=B9, font=('Helvetica', '12'))
bt10 = tk.Button(divl, text='總統府節能減碳', bg='Gainsboro', command=B10, font=('Helvetica', '12'))
bt11 = tk.Button(divl, text='資訊安全', bg='white', command=B11, font=('Helvetica', '12'))
```

圖 4-4 tkinter 程式二

```
bt1.grid(column=0, row=0, sticky=align_mode)
bt2.grid(column=0, row=1, sticky=align_mode)
bt3.grid(column=0, row=2, sticky=align_mode)
bt4.grid(column=0, row=3, sticky=align_mode)
bt5.grid(column=0, row=4, sticky=align_mode)
bt6.grid(column=0, row=5, sticky=align_mode)
bt7.grid(column=0, row=6, sticky=align_mode)
bt8.grid(column=0, row=7, sticky=align_mode)
bt9.grid(column=0, row=8, sticky=align_mode)
bt10.grid(column=0, row=9, sticky=align_mode)
bt11.grid(column=0, row=10, sticky=align_mode)
```

圖 4-5 tkinter 程式三

```
define_layout(divl, rows=11) #用來告知define_layout函式需要運算的東西

win.mainloop()
```

圖 4-6 tkinter 程式四

我們一共用了 tkinter 中的 11 個按鈕配件來做這些資訊的分類，分別為：訊息公告，人事公告，業務統計及研究報告等。

mainloop 是負責用來檢測事件，一旦有新的事件發生就會刷新組件。

下圖為是主畫面的執行結果。

圖 4-7 主畫面的執行結果

4-2 爬蟲程式

```
import tkinter as tk
from layout import define_layout
import requests
from bs4 import BeautifulSoup
import webbrowser

BC = []  #用來做全域變數，因為會用在複數的函式裡
```

圖 4-8 爬蟲程式擴充包

在做爬蟲程式的時候我們主要用到 requests 和 Beautiful Soup。

requests 主要用來訪問網站並獲取當中的 HTML 資料，而 Beautiful Soup 主要用來解析原始的 HTML 程式碼，以便提取網站中的資料，web browser 則是負責用來開啟網站的一個擴充包。

```
def oURL1():
    webbrowser.open_new(BC[0])    #開啟全域變數BC第一個網站

def oURL2():
    webB = 'https://www.president.gov.tw/Page/19?DeteailNo=1&tagNo=12'
    webbrowser.open_new(webB)     #開啟變數webB 的指定網站
```

圖 4-9 爬蟲程式一

變數 BC 是負責用來存取網站中所有獲得的其他網址資料，這使得我們可以開啟最新資訊的網站。

```
def web():
    Tl.tag_config('tagl',justify='center')              #設定字體置中
    Tl.insert('insert','所有訊息公告\n\n',('tagl'))     #輸入的第一行字（開頭）
    X = 0       #變數X 負責用來記錄有幾個資料
```

圖 4-10 爬蟲程式二

這一串程序是用來設定開頭顯示所需要呈現的的資料，而 tagl 是用來做字體的設定。

```
for page in range(1, 10):   #重覆執行此動作(從網頁中的第1頁到第10頁)

    web = requests.get('https://www.president.gov.tw/Page/19?DeteailNo=' + str(page) + '&tagNo=12')  #連結網站
    soup = BeautifulSoup(web.text, "html.parser")       #HTML原始碼解析
    titles = soup.find_all("td", class_="selSearch")    #取得所有class為selSearch的<td>
```

圖 4-11 爬蟲程式三（抓取網站資訊）

這一串程序是利用了迴圈的方式運行 10 次，並將每一次的數字存進變數 page 中，並在抓取網站的時候將變數 page 放在網址中表示頁數的地方，以此方式達成抓取網站中不同頁數的資訊。並在每次抓取一頁後直接解析原始的 HTML 程式碼，並在解析完後找尋所有的指定資料，其中包含了資訊的網址和標題。

```
for title in titles:
    BC.append('https://www.president.gov.tw' + title.select_one("a").get('href'))   #將找到的網頁存入全域變數BC

    B = title.select_one("a").getText()     #獲取資料的名字
    X += 1
    Y = str(X)      #將變數X 轉為字體

    Tl.insert('insert', Y + '.' + B.strip('-') + '\n\n')   #輸入變數Y 和變數B 後進行換行的動作
```

圖 4-12 爬蟲程式四（獲取特定資訊）

在獲取了指定資料後，我們利用迴圈的方式執行以下步驟直至換下一頁，首先將其獲取的網址存進變數 BC 中，再獲取資料中的標題資料，X 和 Y 變數負責用來表示有多少資訊。最後則將獲取到的標題資料和 Y 變數放進顯示畫面上。

```
Tl.insert('end','以上為目前所有訊息公告',('tag1'))        #輸入的最後一行字（結尾）
Tl.configure(state='disabled')        #設定使用者無法在顯示頁面無法進行輸入的動作
```

圖 4-13 爬蟲程式五

這是畫面最後會顯示的資訊，也是表示結尾的部分。

下圖為爬蟲程式的執行結果：

圖 4-14 爬蟲程式六（執行結果）

第五章 操作說明

圖 5-1 主畫面

通過主頁面點擊你想要知道的公告，就會開啟一個新的頁面並獲取官方網站上的全部相關公告，頁面中會以新到舊的方式排列並讓你知道一共有多少個公告

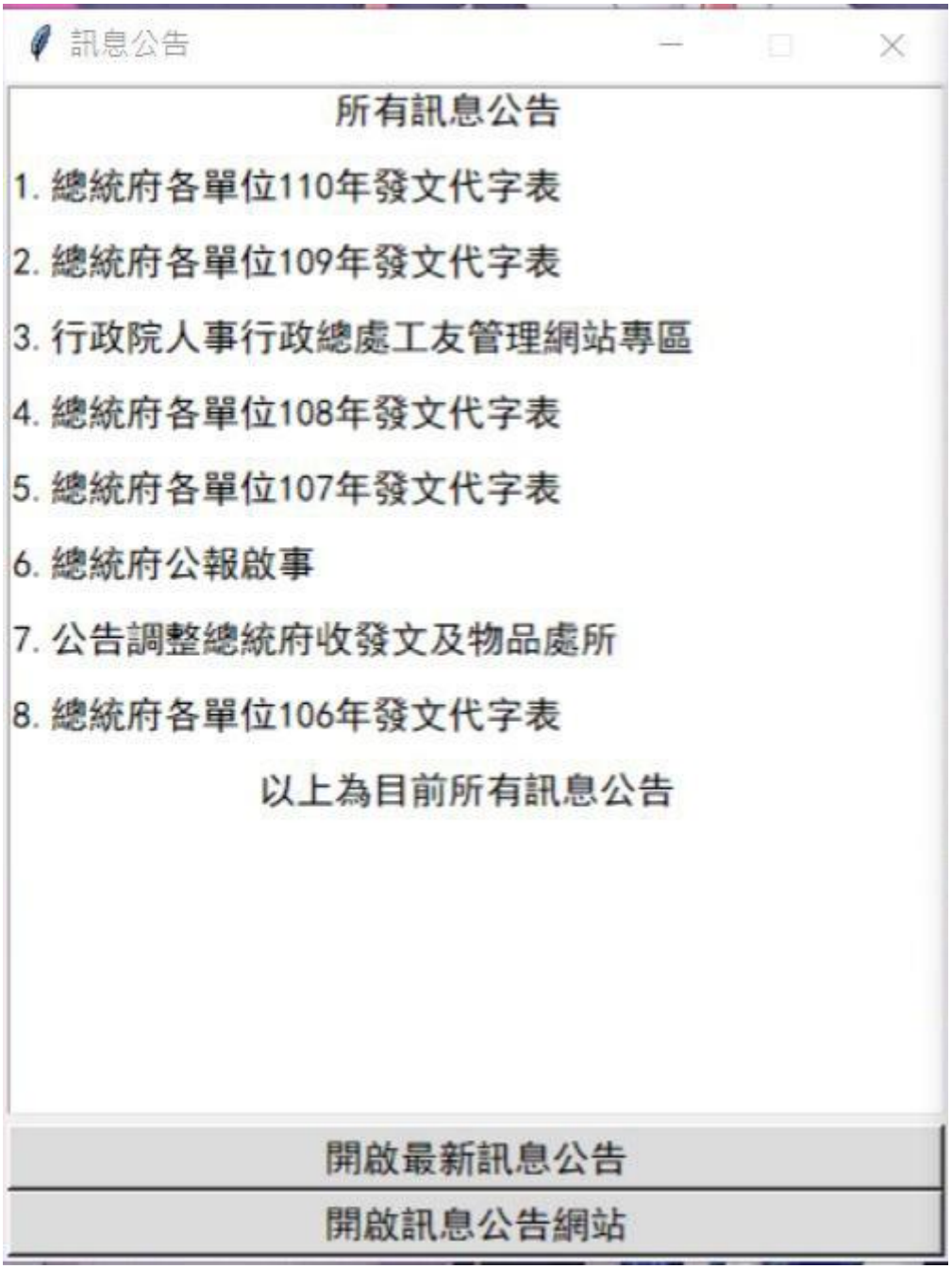

圖 5-2 爬蟲程式畫面

如果你想開啟最新的公告，你可以點擊“開啟最新公告”的按鈕來進入最新的網站，而如果你有想要看的相關公告則可以點擊第二個按鈕進入特定網站來自行找尋你想要的公告

圖 5-3 開啟最新訊息公告

開啟最新訊息公告

圖 5-4 開啟訊息公告網站

開啟訊息公告網站

第六章 問題討論

1.如何把爬蟲程序和 tkinter 擴充包做結合?

由於是第一次用 python，所以當開始的時候不知道如何把 tkinter 和爬蟲程式做結合，但在不斷的嘗試和上網找資料後，終於找到了解決辦法。我們決定先將 tkinter 直接運行並把圖形界面顯示出來，而後爬蟲程式以函式的方式另外運行，將其放在 tkinter 後面做運行的動作

2.如何抓取網站中不同頁數的資訊?

```
for page in range(1, 10):    #重覆執行此動作(從網頁中的第1頁到第10頁)
    web = requests.get('https://www.president.gov.tw/Page/19?DeteailNo=' + str(page) + '&tagNo=12')  #連結網站
```

圖 6-1 抓取網站資訊的程式

如上圖所示，我們利用了迴圈的方式運行 10 次，並將每一次的數字存進變數 page 中，並在抓取網站的時候將變數 page 放在網址中表示頁數的地方，以此方式達成抓取網站中不同頁數的資訊

第七章 結論

經過幾個月的研究我們覺得,要寫好一個完整的爬蟲軟體真的不容易，在其中我們也了解了爬蟲技術不僅僅是使用於搜尋引引擎上,還能用於分析股市資料或是利用爬蟲技術建立一個大規模語料庫,用來做機器翻譯等，這無不體現網路爬蟲在這個大資訊時代的重要性

我們目前只是以總統府網站為範例來做應該便與查詢公告的程式,嚴格上來說這並不能算是網路爬蟲,只能算是結合小部份爬蟲技術特點的小程式,由於是第一次接觸網路爬蟲之技術,所以難免有些不成熟，例如它不能直接的套用在其他的網站上,還有一些功能方面的問題都是我們能繼續改進或增強的地方，讓使用的體驗更加方便順手。

很感謝因這次專題的機會，能讓我們更深入了解網路爬蟲這項技術，如果之後還有機會繼續研究，我們希望能製作出更完善的爬蟲系統，使其能直接套用在其他網站。

參考文獻

1. https://www.rs-online.com/designspark/python-tkinter-cn

   為應用程式設計圖形化介面，使用 Python Tkinter 模組

2. https://www.gushiciku.cn/pl/p5zj/zh-tw

   Python 爬蟲+tkinter 介面來實現歷史天氣查詢

3. https://zhidao.baidu.com/question/137156858.html

   python py 文件中执行另一个 py 文件

4. https://www.learncodewithmike.com/2020/06/how-to-scrape-different-pages-using-python-scraper.html

   [Python 爬蟲教學]Python 網頁爬蟲動態翻頁的實作技巧