## PYTHON 自主學習計畫

基礎程式設計概念 LineBot+爬蟲實作 建國中學 林欣毅

# 目錄

學期目標 ————	3
課程計畫表 ———	3
課程心得與反思 ————	4
Line機器人專案 ————	5
成果展示 ————	6
開發流程 ———	7
建置環境 ———	7
爬蟲函式 ———	11
結合 ————	13
未來拓展方向 ———	14
專案心得與反思 ———	15
程式碼 ————	16
網站資源連結 ————	20

計畫時長:12週,每週2.5小時

### 學期目標

建構基礎設計程式的邏輯,並應用所學及課程提供的資源完成屬於自己、具實用性與拓展性的程式

### 課程資源

Udemy平臺-Python - 輕鬆學會寫程式

講師:劉曣全 時長:17.5小時

### 自主學習計畫表

<b></b> 周次	課程內容
1	下載Python、Git、Sublime,註冊Github帳號,介紹'宣告'
2	學習: '註解'、比較符號、 'if迴圈'
3	學習: '上傳至Github'、'While迴圈'、'import'
4	學習: 'For迴圈'、'清單 (list)'、'讀取檔案'、'range'
5	學習: '二維清單'、'寫入資料'、'分割資料 '、Continue
6	學習:'函式(funtion)' 介紹常見文法錯誤
7	學習:字典清單 (word)、載入套件
8	學習寫入Excel,Word時修正輸入法、處理簡訊、圖片
9	介紹:CMD, sublime操作技巧, 學習:路徑(path)
10	練習:嘗試設計Line機器人 認識class,self
11	Line機器人
12	Line機器人

圖表來源:excel自製

# 課程心得與反思

#### 因為興趣相同而聚集的小團體

由於我們高一班有一群對程式有興趣但都沒接觸過正式教學的人,因此我們合買了一堂線上課程並申請了圖書館的小教室一同學習Python的基礎,即使彼此剛認識不到一個月,我們仍舊因為相同目標聚在一起。

#### 九人行必有我師

一群人比起一個人學習有許多好處,特別是在遇到不懂的概念時可以互助合作,<u>不懂的人可以藉由不同角度的解說理解觀念,懂的人也能藉由講解的過程梳理自己的邏輯</u>,對大家都有所助益。

#### 從失敗中收集經驗

很遺憾的,因為高一時對程式的理解還不足,我們最後在架設伺服器的階段停止了我們最後三週設計Line機器人的計畫,不過<u>挫折亦是養分,留下來的資料跟經驗也幫助我在上完高二的大數據應用專題後能與之結合做出自己的Line機器人。</u>



### 專案目標

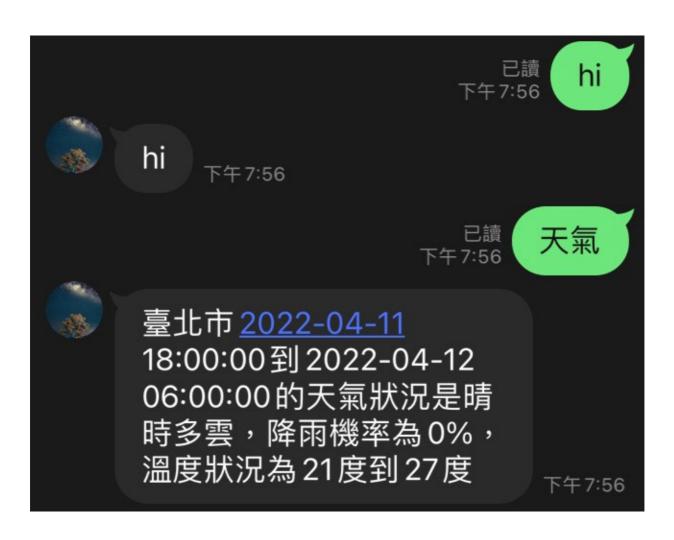


做出Line機器人並可以根據關鍵字爬天氣預報

### 發想動機

起初我原本打算做公車站況,但後來考慮到我手機中的 "台北等公車"app已經成為我每天搭車的習慣,做一個 bot實用性不高,所以後來打算做臺北地區的天氣預報

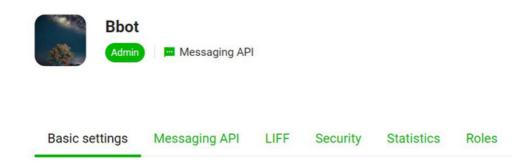
#### 成果展示



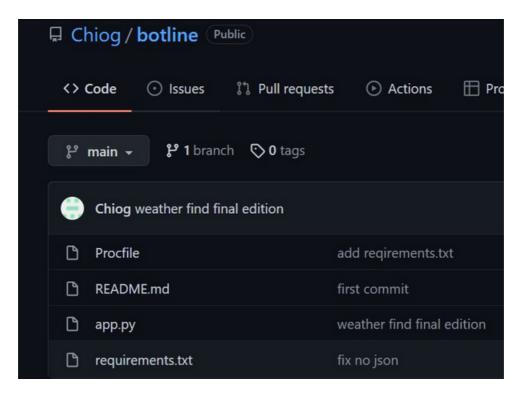
# 開發流程

## 建立環境

#### 建立Line Developer帳號



#### 建立GitHub專案



由於高一已經上過怎麼開啓專案, 這部分我沒有遇到很大的問題

安裝Line SDK(放在GitHub)

\$ pip install line-bot-sdk

軟體開發套件(SDK)對我來講是一個很新的概念, 目前也只會運用,未來會再更深入瞭解。

使用範例程式碼,把權杖(token)跟密碼(secret)放入其中

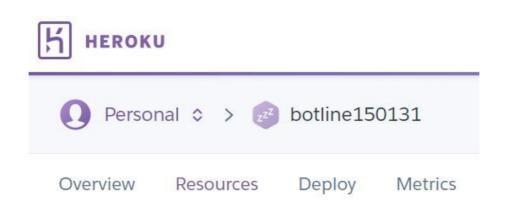
Channel secret ② 2f924288068b86649d

Channel access token (long-lived) ③

m0v5zVz8LTLNgnT8q40sTaqW8QnNF

權杖跟密碼的部分擷圖

註冊Heroku帳號、安裝Heroku CLI 建立Heroku專案



這邊是我們高一停止的地方,由於網路教學年分較早,Heroku的官網介面有巨大更動,除此之外 CLI啓動方式也有所不同,而我們當時的英文能力並不足以解決,高三後英文有所進步,我才解決了這些問題。



教學裡有要創立這個檔案,但目前仍 不清楚目的,之後會再瞭解

```
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明
line-bot-sdk
gunicorn
flask
requests
bs4
```

可以用requirements.txt告訴heroku要import甚麼

#### 將Heroku網址放到Line developer

Your app can be found at <a href="https://botline150131.herokuapp.com/">https://botline150131.herokuapp.com/</a>

Webhook URL ② https://botline150131.herokuapp.com/callback

Verify Edit

這邊讓我卡了很久,因為line需要的網址最後不能加反斜但又不報錯,導致我花了很久才找到問題點,

不過這也讓我重新審視了一遍我其他地方有沒有出現問題

連接完之後就已經能發訊息了, 接下來就是要爬資料



## **爬蟲函式**

我一開始是想要用高二學到的beautifulsoup去爬中央氣象局的網站,但是無論怎麼樣它的輸出都是NONE有懷疑過是不是自己抓錯東西,但後來用測試網站測試沒問題,懷疑可能是有特別反爬蟲,但不確定,最終只好轉方向去搜尋有沒有API可以用

中央氣象局的網站程式碼節錄



後面也會放

## **爬蟲函式**

幸運的是,後來有找到中央氣象局開放資料平臺之資料擷取API,只要申請就可以用了。至於方法則是JSON 找到資料放在哪即可使用,這部分由於高二也有學過, 沒有遇到太大問題。

臺北市2022-04-12 00:00:00到2022-04-12 06:00:00的天氣狀況是睛時多雲,降雨機率為0%,溫度狀況為21度到24度

```
main.py ×
     import requests
     import json
  3
  4
  5 #print("hi")
  6 ▼ def get_data():
  8
          url = "https://opendata.cwb.gov.tw/api/v1/rest/datastore/F-C0032-001"
  9 ₹
          params = {
 10
              "Authorization": "CWB-BC84844D-BF61-4B5E-9DB2-8CAFF5C4DF3F",
 11
              "locationName": "臺北市",
 12
 13
 14
          response = requests.get(url, params=params)
 15
          #print(response.status code) 確認狀態
 16
 17 ▼
          if response.status_code == 200:
 18
 19
              #print(response.text)
 20
              data = json.loads(response.text)
 21
 22
             location = data["records"]["location"][0]["locationName"]
 23
             weather_elements = data["records"]["location"][0]["weatherElement"]
 25
             start_time = weather_elements[0]["time"][0]["startTime"]
             end_time = weather_elements[0]["time"][0]["endTime"]
 26
 27
             weather_state = weather_elements[0]["time"][0]["parameter"]["parameterName"]
 28
             rain_prob = weather_elements[1]["time"][0]["parameter"]["parameterName"]
 29
             min_tem = weather_elements[2]["time"][0]["parameter"]["parameterName"]
 30
             max_tem = weather_elements[4]["time"][0]["parameter"]["parameterName"]
 31
             r = (location + start_time + "到" + end_time + "的天氣狀況是" + weather_state + ",降雨機率
      為"+ rain_prob + "%, 溫度狀況為" + min_tem + "度到" + max_tem + "度")
 33
             #print(r)
 34
           return r
```

接著只差最後一步 - 把兩者結合就好

## 結合

原本想說會很順利,但因為Line的報錯很不明顯, 又因為需要上傳到Heroku才能執行,因此這邊花了比 預期還要多很多的時間,結果後來發現只需將輸出的 元組(tuple)換成字串(str)就好

```
2022-04-08T14:25:53.847573+00:00 app[web.1]: File "/app/.heroku/python/lib/python3.10/site-packages/linebot/api.py", line 1875, in __check_error
2022-04-08T14:25:53.847573+00:00 app[web.1]: raise LineBotApiError(
2022-04-08T14:25:53.847575+00:00 app[web.1]: linebot.exceptions.LineBotApiError: LineBotApiError: status_code=400, request_id=2f90c18e-7936-4260-acf9-083486191a30,
error_response={"details": [], "message": "The property, 'messages[0].text', in the request body is invalid (line: 1, column: 90)"}, headers={'Content-Type': 'application/json',
'Server': 'envoy', 'x-content-type-options': 'nosniff', 'x-frame-options': 'DENY', 'x-line-request-id': '2f90c18e-7936-4260-acf9-083486191a30', 'x-xss-protection': '1;
mode*block', 'Content-Length': '100', 'Expires': 'Fri, 08 Apr 2022 14:25:53 GMT', 'Cache-Control': 'max-age=0, no-cache, no-store', 'Pragma': 'no-cache', 'Date': 'Fri, 08 Apr
2022 14:25:53 GMT', 'Connection': 'close'}
```

#### Heroku logs的執行畫面(類似簡化版CMD)

這一步有尋求同學的幫助,我也在他身上學到了 更多debug技巧,以及如何分析長長的報錯訊息中 到底隱含了哪些重要訊息。



# 未來展望

- 1. 加入可以選擇小區域的功能
- 2. 加入更多對話内容
- 3. 加入傳送課表功能
- 4. 加入待辦事項紀錄
- 5. 其他功能



來自同學的回饋
(Bbot是機器人名)
證明它具有
可拓展性
實用性

## 心得與反思

### 在這次的專案裡,我學到了:

- 1. 觀察報錯的能力
- 2. 用英文找問題解決方法 (中文圈資料不夠時)
- 3. 所學統整(高一自主、高二 大數據應用專題)結合
- 4.用GitHub進行版本控制
- 5.借用伺服器(Heroku)

### 我發現自己還需要加強:

- 1.更快的找到偵錯
- 2.英文能力
- 3.基礎程式觀念
- 4. 減少開發時長
- 5. 自學能力
- 6.熟悉其他GitHub介面

# 程式碼一主程式

#### 網址:

https://replit.com/@BrueLin/linebotmain#main.py

```
app.py 6 X
C: > Users > user > Desktop > botline > @ app.py > 🕅 handle_message
       from flask import Flask, request, abort
  2
       import requests
      from bs4 import BeautifulSoup
       import ison
      from linebot import (
           LineBotApi, WebhookHandler
       from linebot.exceptions import (
  8
           InvalidSignatureError
 10
       from linebot.models import (
 11
 12
           MessageEvent, TextMessage, TextSendMessage,
 13
 14
       def get_weather_data():
 15
 16
 17
           url = "https://opendata.cwb.gov.tw/api/v1/rest/datastore/F-C0032-001"
           params = {
 18
               "Authorization": "CWB-BC84844D-BF61-4B5E-9DB2-8CAFF5C4DF3F",
 19
               "locationName": "臺北市",
 20
 21
           }
 22
           response = requests.get(url, params=params)
 23
           #print(response.status_code) 確認狀態
 24
 25
 26
           if response.status_code == 200:
 27
               #print(response.text)
 28
               data = json.loads(response.text)
 29
```

```
location = data["records"]["location"][0]["locationName"]
            weather elements = data["records"]["location"][0]["weatherElement"]
            start_time = weather_elements[0]["time"][0]["startTime"]
34
            end_time = weather_elements[0]["time"][0]["endTime"]
            weather_state = weather_elements[0]["time"][0]["parameter"]["parameterName"]
            rain_prob = weather_elements[1]["time"][0]["parameter"]["parameterName"]
            min_tem = weather_elements[2]["time"][0]["parameter"]["parameterName"]
            max_tem = weather_elements[4]["time"][0]["parameter"]["parameterName"]
            w = (location, start_time, "到", end_time, "的天氣狀況是", weather_state, ",降兩機率為"
            ,rain_prob ,"%<sup>,</sup>溫度狀況為" ,min_tem ,"度到" ,max_tem ,"度")
41
     app = Flask(__name__)
     line_bot_api = LineBotApi('m0v5zVz8LTLNgnT8q40sTaqW8QnNPk5r/HPucZJ9lZ2BLt9B2Um/pFNsRSNJx6Mqhz1Sh3OYc/9
     handler = WebhookHandler('2f924288068b86649d555531365bde00')
     #網址後callback 這邊會被執行
     @app.route("/callback", methods=['POST'])
     def callback():
        # get X-Line-Signature header value
         signature = request.headers['X-Line-Signature']
54
        # get request body as text
        body = request.get_data(as_text=True)
57
         app.logger.info("Request body: " + body)
60
          try:
              handler.handle(body, signature)
          except InvalidSignatureError:
              print("Invalid signature. Please check your channel access token/channel secret.")
64
              abort(400)
          return 'OK'
67
      @handler.add(MessageEvent, message=TextMessage)
      def handle_message(event):
70
71
72
          msg = event.message.text
          r = '無法回覆的內容
73
          if msg == "hi":
              r = "hi"
75
              line_bot_api.reply_message(
              event.reply_token, #要給token才能執行
78
              TextSendMessage(text= r))
          elif msg == "天氣":
80
81
              line_bot_api.reply_message(
              event.reply_token, #要給token才能執行
82
              TextSendMessage(text= str(get_weather_data())))
83
              #r = str(location, start_time, "到", end_time, "的天氣狀況是", weather_state, ",
84
85
      if __name__ == "__main__":
86
87
          app.run()
```

# 程式碼一爬蟲(失敗)

網址:https://replit.com/@BrueLin/det#main.py

```
main.py ×
                                                             Console Shell
  1
     import requests
                                                              資料接收中...請稍等
  2
     from bs4 import BeautifulSoup
                                                              wow None
     print('資料接收中...請稱等')
  3
                                                              » [
  4
  5
     requests.get('https://www.cwb.gov.tw/V8/C/W/Town/Tow
     n.html?TID=6300500')
     soup = BeautifulSoup(res.text , 'html.parser')
  6
     #print(soup.prettify())
     monday = soup.find("span", class_ = 'tem-C')
  8
     print('wow' ,monday)
```

# 程式碼一爬蟲測試

網址:https://replit.com/@BrueLin/scrapeTest#main.py

```
main.py ×
                                                              Console Shell
      import requests
                                                               資料接收中...請稍等
      from bs4 import BeautifulSoup
  2
                                                               WOW
  3
      print('資料接收中...請稍等')
  4
  5
     res = requests.get('https://quotes.toscrape.com')
                                                               Quotes to Scrape
     soup = BeautifulSoup(res.text , 'html.parser')
  6
  7
     #print(soup.prettify())
     monday = soup.find("div", class_ = 'header-box')
  8
     print('wow' ,monday.text)
  9
                                                               Login
```

# 網站資源連結

Python - 輕鬆學會寫程式 | Udemy

https://www.udemy.com/course/python-

learn/#instructor-1

Line Developer

https://developers.line.biz/en/

Line SDK

https://github.com/line/line-bot-sdk-python

Heroku

https://dashboard.heroku.com/

IT邦

https://ithelp.ithome.com.tw/articles/10276375

中央氣象局

https://www.cwb.gov.tw/V8/C/W/Town/Town.ht

ml?TID=6300500

爬蟲測試網站

https://quotes.toscrape.com