1. 執行環境

   Visual Studio Code
2. 程式語言

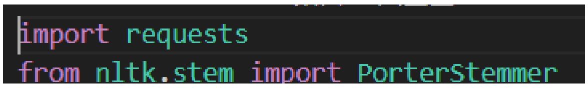
   Python 3.10.6
3. 執行方式

   確認 python 已經安裝並可由 vscode 執行

   在 cmd 中使用 pip install requests, pip install nltk 來安裝套件

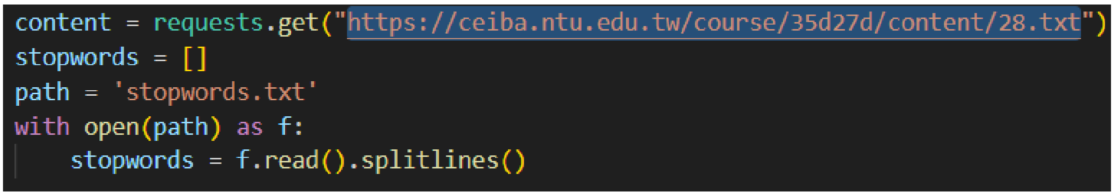
   確認 stopwords.txt 與 pa1.py 在同個資料夾中，在 vscode 打開該資料夾點擊 run 即可執行(必須打開整個資料夾避免找不到 stopwords.txt)
4. 邏輯說明

   a. Package import: import requests and import PorterStemmer from nltk.stem



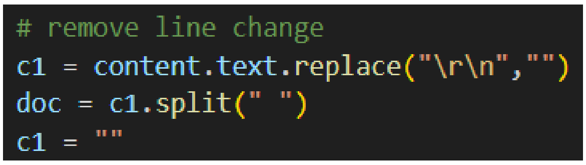
```
import requests
from nltk.stem import PorterStemmer
```

   b. Read data: use requests.get() method to get the text file I need to process at https://ceiba.ntu.edu.tw/course/35d27d/content/28.txt and read the stopwords.txt file for stop words, the stop words I use are based on this website, https://www.ranks.nl/stopwords .



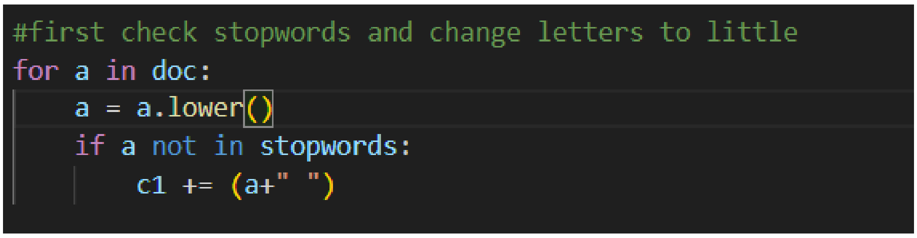
```
content = requests.get("https://ceiba.ntu.edu.tw/course/35d27d/content/28.txt")
stopwords = []
path = 'stopwords.txt'
with open(path) as f:
    stopwords = f.read().splitlines()
```

   c. Data preprocessing: the data originally contained change line symbol, represented by \r\n, I first delete them and we split the sentence by white space.



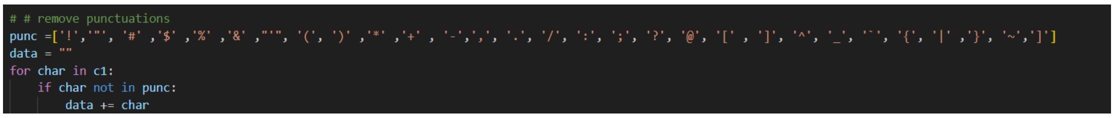
```
# remove line change
c1 = content.text.replace("\r\n","")
doc = c1.split(" ")
c1 = ""
```

   d. Lowercase and remove stopwords: use lower() method to change all letters to lowercase and check if they are in the stopwords. If they are, remove them, otherwise keep them. The reason I'd like to do stopwords removal first is that the following steps are punctuations removal and stemming. If I don't do removal first, words like "ill" and "I'll" will become non-distinguishable and words like" you've" may become "youv" after stemming, so I chose to do removal first



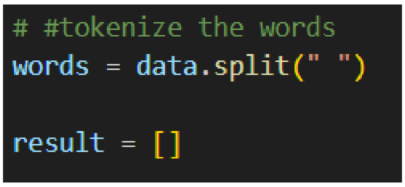
```python
#first check stopwords and change letters to little
for a in doc:
    a = a.lower()
    if a not in stopwords:
        c1 += (a+" ")
```

e. Delete punctuations: if we find punctuations, replace it with empty strings. String.puntuation is a good fit, but since I can't import String, I just list them in a list to do the task.



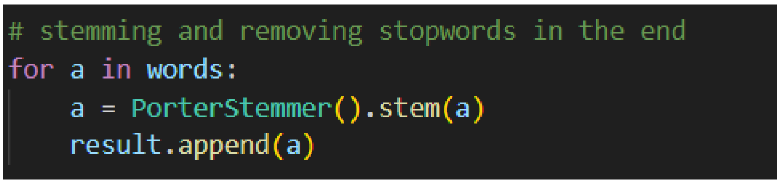
```python
# # remove punctuations
punc =['!','"', '#' ,'$' ,'%' ,'&' ,"'", '(', ')' ,'*' ,'+' , '-',',', '.', '/', ':', ';', '?', '@', '[' , ']', '^', '_', '`', '{', '|' ,'}', '~',']']
data = ""
for char in c1:
    if char not in punc:
        data += char
```

f. Tokenization: since I now throw away all the punctuations and stop words, I can easily tokenize the words in the sentence by split with the white space.



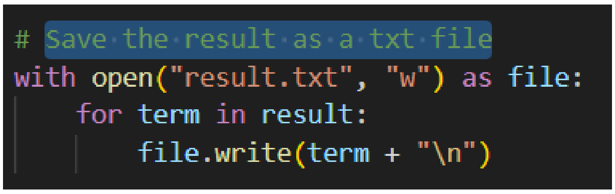
```python
# #tokenize the words
words = data.split(" ")

result = []
```

g. Stemming using Porter's algorithm: for every token, I stem it using Porter's algorithm (by the imported function PorterStemmer).



```python
# stemming and removing stopwords in the end
for a in words:
    a = PorterStemmer().stem(a)
    result.append(a)
```

h. Save the result as a txt file:



```python
# Save the result as a txt file
with open("result.txt", "w") as file:
    for term in result:
        file.write(term + "\n")
```