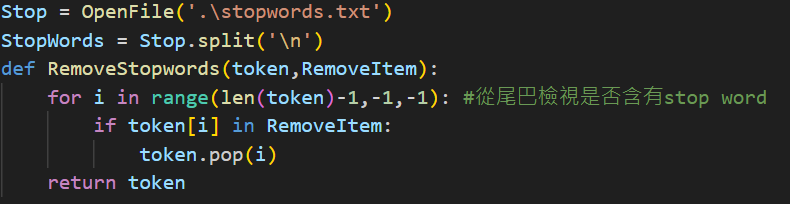
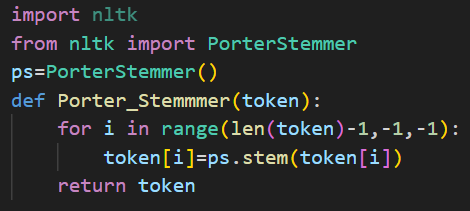
1. 執行環境: VS code
2. 程式語言: Python 3.11.5
3. 執行方式:
   1. 將資料來源放在data資料夾中stopwords.txt放在最外層的資料夾
   2. 建立Output資料夾放置輸出結果
   3. 在外層建立兩個txt文檔，分別為dictionary.txt以及1.txt
4. 作業處理邏輯
   1. 建立讀檔案函式與將文件切割成token的兩個函式



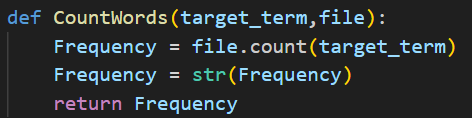
* 1. 讀入stopwords.text文件，此文件中包含常見的stop word，並且設計移除stop word的函式



* 1. 導入PorterStemmer已進行Porter演算法，並將此演算法寫成函式



* 1. 建立CountWords函式，可以計算資料中特定term出現的次數

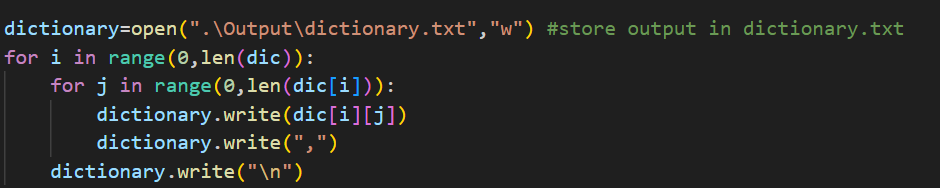


* 1. 建立TermFrequency函式，可以讀進特定範圍內的文件數，並且這次的資料來源命名方式都是按照數序的方式命名。此函式可以計算出DF

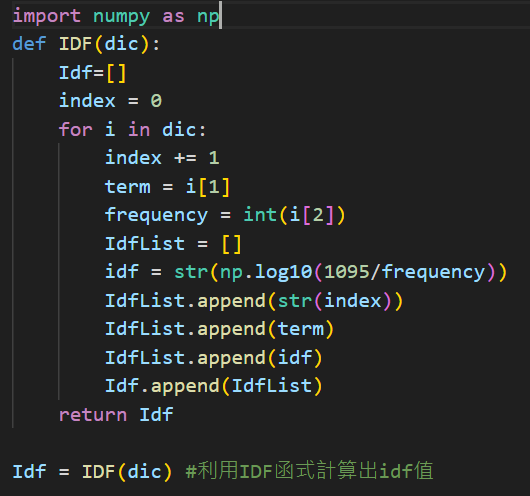


最後計算出的結果存在dic變數中

* 1. 將dic變數中的結果寫入名為dictionary.txt的文件中

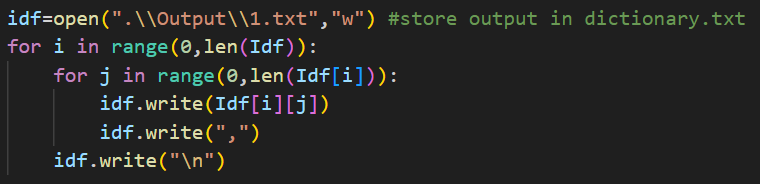


* 1. 導入numpy套件(np)，撰寫可以計算IDF的函式

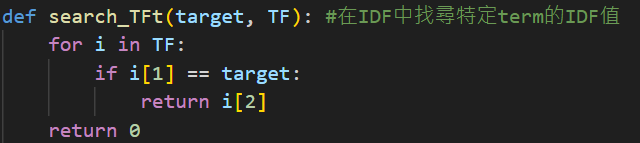


並將結果保存在Idf的變數當中

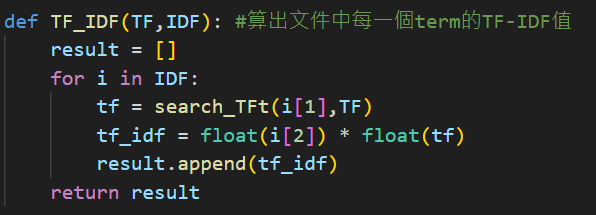
* 1. 將Idf的結果寫入名為1.txt的文件中



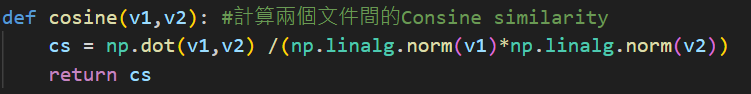
* 1. 建立search\_TFt函式，可以在先前儲存的結果中找尋特定term的TF值



* 1. 建立TF-IDF函式，可以計算出TF-IDF值



* 1. 建立cosine函式，可以計算cosine similarity值



* 1. 最終試著計算文件1與文件2之間的cosine similarity值

