

File IO 檔案讀寫

這邊介紹python 內建的開啟檔案與檔案讀寫指令，在處理文字為主的檔案時比較方便

open()打開檔案

Before you read or write a file using the python built-in read/write function, you need to "open" the file first.

```
fo = open('filename', 'open_mode_option')
```

(fo可以是自己決定的變數名稱，指向被開啟的檔案，稱為fileObject，後續讀取檔案的時候會使用到)

針對文字檔，幾種比較常用到的open mode options:

- **r**: 只讀檔，檔案要先存在，從頭開始讀
- **rt**: 可讀可寫，檔案要先存在，從頭開始讀。
- **w**: 只可寫出，從頭覆蓋既有檔案，檔案如果不存在就新增。
- **wt**: 可讀可寫，從頭覆蓋既有檔案，檔案不存在就新增。
- **a**: 接續既有檔案寫出，檔案不存在就新增。
- **at**: 接續既有檔案寫出與讀取，檔案不存在就新增。

[其他的open mode options可以在這邊找到](#)

Example of opening a file, and checking its status

```
In [7]: # Open a file
fo = open("foo.txt", "w")           # Open for writing only. Overwrites
existing file.
print("Name of the file: ", fo.name) # Let's check if the file name is
correct
print("Opening mode : ", fo.mode)    # Check the read/write/binary mode
status
print("Closed or not : ", fo.closed) # Check the open/close status

Name of the file:  foo.txt
Opening mode :  w
Closed or not :  False
```

close() 關閉檔案

```
In [8]: # Open a file
fo = open("foo.txt", "w")
print("Closed or not : ", fo.closed) # Check the open/close status

# Close opened file
```

```
fo.close()
print("Closed or not : ", fo.closed)    # Check the open/close status again
```

```
Closed or not : False
Closed or not : True
```

write() 輸出到檔案

```
In [9]: # Open a file
fo = open("foo.txt", "w")    # open for writing only
fo.write("This is the first output.") # output string to the file.
# As no \n is included at the end of the string ,
# the next __write().__ will continue to output to the same line.

fo.write("This is the second output.\n") # Continue the first line.
# A \n is included at the end. The next __write().__ will start a new line

fo.write("Python is a great language.\nYeah its great!!\n") #write two new lines to the file

# Close opened file
fo.close()
```

read() 讀取檔案

```
In [10]: fo = open("foo.txt", "r+") # open for readingly only; read from the beginning
A=fo.read(7)
print(A)
B=fo.read(10)
print(B)
```

```
This is
the first
```

readline and readlines 單列或多列讀取

```
In [11]: fo = open("foo.txt", "r+") # open for readingly only; read from the beginning
A=fo.readline() # read only the first line in the file
print(A)

B=fo.readlines() # read the rest of the file (2nd~3rd lines) as a python list
print(len(B))    # checking the size of the list
print(B)         # the elements are separated by '\n' in the string
print(B[1])      # the second element in the list
```

```
This is the first output.This is the second output.
```

```
['Python is a great language.\n', 'Yeah its great!!\n']
Yeah its great!!
```

readline搭配split分割字串

```
In [14]: fo = open("foo.txt", "r+") # open for readingly only; read from the begin
ning
A=fo.readline() # read only the first line in the file
print(A)
As=A.split(' ')
print(As)

B=fo.readlines()
print(B[1])
Bs=B[1].split(' ')
print(Bs)
print(Bs[0])
```

This is the first output.This is the second output.

```
['This', 'is', 'the', 'first', 'output.This', 'is', 'the', 'second', 'outpu
t.\n']
Yeah its great!!
```

```
['Yeah', 'its', 'great!!\n']
Yeah
```

其他讀寫檔案方式 請自行學習

- 利用pandas讀寫檔案(csv)，讀檔的話讀進來就變成DataFrame格式
- 利用np.loadtxt, np.savetxt
- 利用csv模組讀寫檔案
- 使用pickle直接將資料結構儲存起來

小練習

資料夾中有一個**words.txt**的檔案大家可以嘗試去讀檔案並將檔案中的英文字計算出現次數後做排序(先出現在文章中的排在前面)，之後輸出到另一個**.txt**檔中，格式如下：

詞，index，出現次數

```
Singaporeans 0 2
were 1 1
meant 2 1
to 3 18
go 4 1
the 5 29
polls 6 1
at 7 3
```

```
end 8 1
of 9 14
next 10 1
week 11 1
vote 12 1
for 13 8
a 14 13
new 15 6
president 16 4
but 17 3
theyll 18 1
no 19 2
longer 20 1
...
```

小練習2

讀取201409220900_ROBS.txt檔案，並將////改為**-999.**，注意前面幾行要略過，還有最後一行也要拿掉

In []: