COM738 – Data Science Foundations – Week 3 Practical

This practical will give you the experience of file handling in Python.

Read Text File

Task 1- Reading the entire content of the file. Open sample.txt file in read mode. Read all the contents of the text file and print the text to the console.

Solution:

```python
fileObject = open("sample.txt", "r")
data = fileObject.read()
print(data)
```

```
Hi User!
Welcome to Python Examples.
Continue Exploring.
```

Task 2- Read Text File Line by Line. Open sample.txt file in read mode. Read text line by line from the file and print the text to the console. Use File.readline() function.

Solution:

```python
#get file object
f = open("sample.txt", "r")

while(True):
	#read next line
	line = f.readline()
	#if line is empty, you are done with all lines in the file
	if not line:
		break
	#you can access the line
	print(line.strip())

#close file
f.close
```

```
Hi User!
Welcome to Python Examples.
Continue Exploring.
```

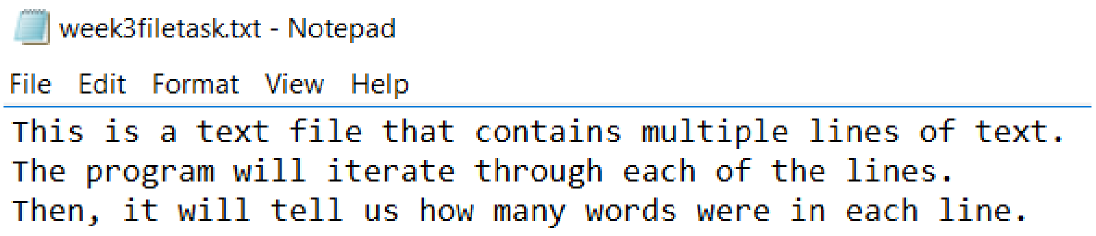
Task 3- Create a plaintext file using notepad, containing a few sentences on separate lines. Save this file in the same directory as your Jupyter Notebook. Create a Python function that will output something similar to the following:

```
11 words in: This is a text file that contains multiple lines of text.

9 words in: The program will iterate through each of the lines.

12 words in: Then, it will tell us how many words were in each line.
```

Solution:



```
This is a text file that contains multiple lines of text.
The program will iterate through each of the lines.
Then, it will tell us how many words were in each line.
```

```
with open('week3filetask.txt','r') as file:
    for line in file:
        words = line.split(' ')
        print("{0} words in: {1}".format(len(words),line))
```

Note: The split function splits based on the separation criteria. In this case space ‘   ‘ given as criteria.

Read CSV File

Task 4- Use the Belfast Food Premise Hygiene Ratings dataset→ foodhygienedata.csv, provided in week 3 material. Print a list containing all field names in the header.

Solution:

```
#Print a list containing all field names in the header
import csv
with open('foodhygienedata.csv', 'r') as file:
    input_file = csv.DictReader(file,delimiter=',')
    print(input_file.fieldnames)
```

```
['establishmentname', 'establishmentaddressline1', 'establishmentaddressline2', 'establishmentaddressline3', 'establishmentaddr
essline4', 'postcode', 'rating', 'latitude', 'longitude', 'inspectiondate']
```

Task 5- Print a list of all postcodes.

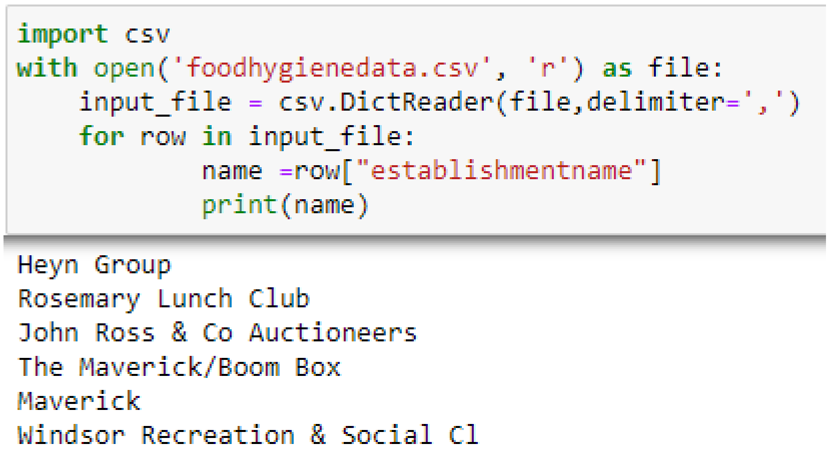
Solution:

```
#Print a list of all  postcodes
import csv
with open('foodhygienedata.csv', 'r') as file:
    input_file = csv.DictReader(file,delimiter=',')
    for row in input_file:
            name =row["postcode"]
            print(name)
```

```
BT3 9AH
BT15 5HD
BT1 4NX
BT1 2GX
BT1 2JF
BT9 7DS
```

Task 6- Print a list of all establishment names.

Solution:

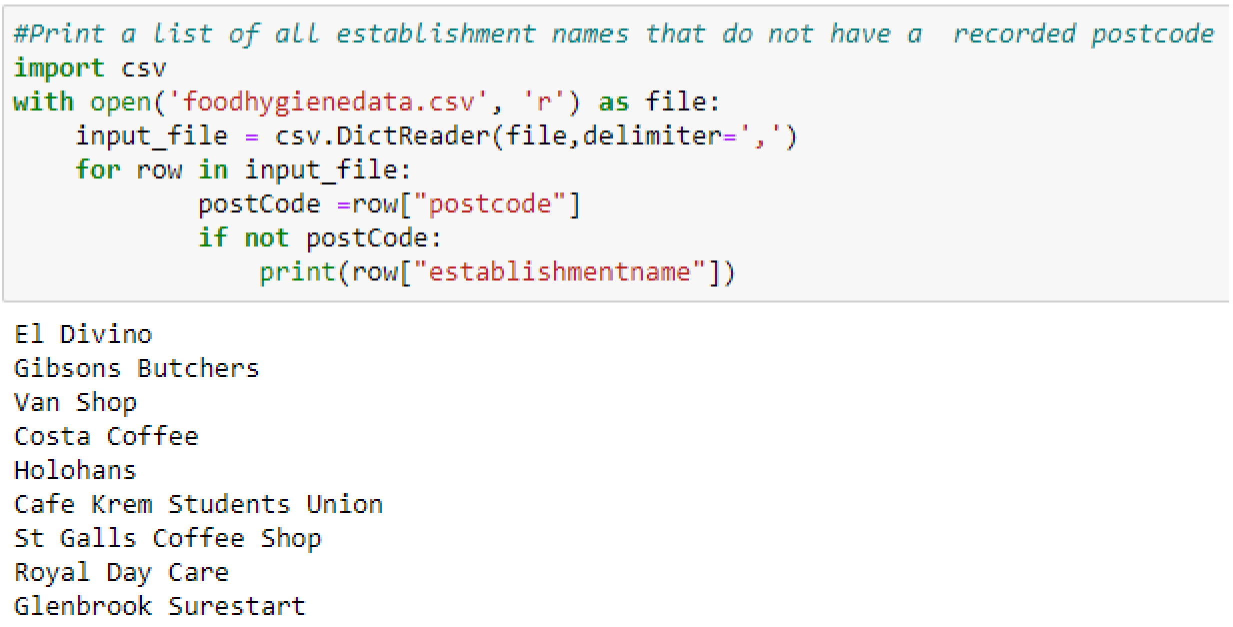


```
import csv
with open('foodhygienedata.csv', 'r') as file:
    input_file = csv.DictReader(file,delimiter=',')
    for row in input_file:
            name =row["establishmentname"]
            print(name)
```

```
Heyn Group
Rosemary Lunch Club
John Ross & Co Auctioneers
The Maverick/Boom Box
Maverick
Windsor Recreation & Social Cl
```

Task 7- Print a list of all establishment names that do not have a recorded postcode.

Solution



```
#Print a list of all establishment names that do not have a  recorded postcode
import csv
with open('foodhygienedata.csv', 'r') as file:
    input_file = csv.DictReader(file,delimiter=',')
    for row in input_file:
            postCode =row["postcode"]
            if not postCode:
                print(row["establishmentname"])
```

```
El Divino
Gibsons Butchers
Van Shop
Costa Coffee
Holohans
Cafe Krem Students Union
St Galls Coffee Shop
Royal Day Care
Glenbrook Surestart
```

The “**any**” function:

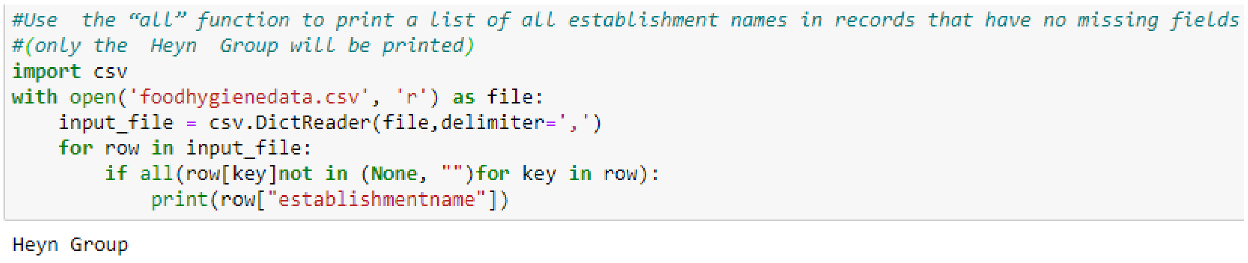
- Returns **True** on the first encounter of a condition evaluating to True, else returns False.

The “**all**” function:

- Returns **False** on the first encounter of a condition evaluating to False, else returns True.

Task 8- Print a list of all establishment names in records that have no missing fields (only the Heyn Group will be printed).

Solution:



```
#Use  the “all” function to print a list of all establishment names in records that have no missing fields
#(only the  Heyn  Group will be printed)
import csv
with open('foodhygienedata.csv', 'r') as file:
    input_file = csv.DictReader(file,delimiter=',')
    for row in input_file:
        if all(row[key]not in (None, "")for key in row):
            print(row["establishmentname"])
```

```
Heyn Group
```

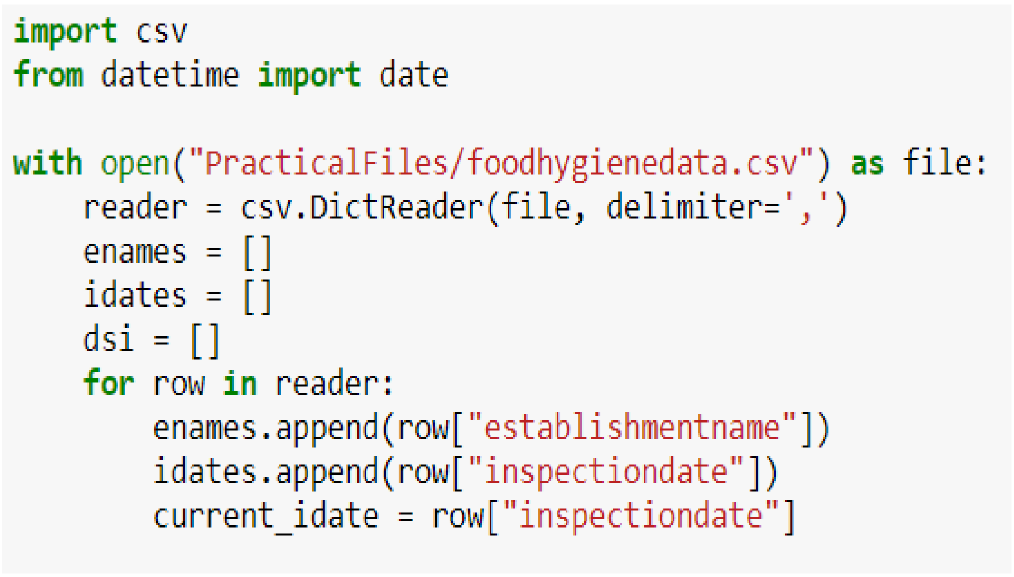
<u>Note</u>: Use the “all” function to print a list of all establishment names in records that have no missing fields (only the Heyn Group will be printed).

Task 9-Read in the establishment names and inspection dates listed in the Food Hygiene Dataset. For each record, calculate the value “DaysSinceInspection”. This value contains the number of days since the date specified in the “inspectiondate” field.

Note: You will need to use the string “**split**” method on each date from the CSV file, and then create a new **Date** object from these string components. You will then need to subtract this new Date object from today’s date to find the difference. Do some Googling around this area! You will also need to deal appropriately with records that have no date value (use an if statement)

Solution:

Step 1 - Read in the establishment names and inspection dates listed in the Food Hygiene Dataset



```
import csv
from datetime import date

with open("PracticalFiles/foodhygienedata.csv") as file:
    reader = csv.DictReader(file, delimiter=',')
    enames = []
    idates = []
    dsi = []
    for row in reader:
        enames.append(row["establishmentname"])
        idates.append(row["inspectiondate"])
        current_idate = row["inspectiondate"]
```

Step 2 – Check if there is a date for each row. If there is:

(a) Split the date string into separate components (day, month, year)

(b) Calculate the number of days since the inspection date.

```python
if current_idate:
    dtcomponents = current_idate.split("/")
    dt = date(int(dtcomponents[2]), int(dtcomponents[1]),int(dtcomponents[0]))
    dtdifference = date.today() - dt

    dsi.append(dtdifference.days)
else:
    dsi.append("")
```

Task 10- Create a new CSV file which contains a collection of establishment names, inspection dates, and days since inspection.

Solution:

```python
with open("PracticalFiles/foodhygienedata_DAYS.csv", "w", newline="") as file:
    writer = csv.writer(file,delimiter=",")
    headers = ["establishmentname","inspectiondate","DaysSinceInspection"]
    writer.writerow(headers)

    for ename,idate,numdays in zip(enames,idates,dsi):
        writer.writerow([ename,idate,numdays])
```

Complete solution Task 9 and Task 10

```python
import csv
from datetime import date

with open("PracticalFiles/foodhygienedata.csv") as file:
    reader = csv.DictReader(file, delimiter=',')
    enames = []
    idates = []
    dsi = []
    for row in reader:
        enames.append(row["establishmentname"])
        idates.append(row["inspectiondate"])
        current_idate = row["inspectiondate"]

        if current_idate:
            dtcomponents = current_idate.split("/")
            dt = date(int(dtcomponents[2]), int(dtcomponents[1]),int(dtcomponents[0]))
            dtdifference = date.today() - dt

            dsi.append(dtdifference.days)
        else:
            dsi.append("")

with open("PracticalFiles/foodhygienedata_DAYS.csv", "w", newline="") as file:
    writer = csv.writer(file,delimiter=",")
    headers = ["establishmentname","inspectiondate","DaysSinceInspection"]
    writer.writerow(headers)

    for ename,idate,numdays in zip(enames,idates,dsi):
        writer.writerow([ename,idate,numdays])
```

Task 11: Generate the following metrics from the dataset detailing-A count of how many establishments received each rating: 1, 2, 3, 4 and 5, and how many ratings were omitted. Output should look like the following:

```
Amount scoring 1:  28
Amount scoring 2:  61
Amount scoring 3:  391
Amount scoring 4:  820
Amount scoring 5:  1786
Amount missing:  82
```

Solution:

```python
import csv

with open("PracticalFiles/foodhygienedata.csv") as file:
    reader = csv.DictReader(file,delimiter=',')
    ratings = [0,0,0,0,0]
    omittedRatings = 0
    for row in reader:
        if row["rating"]=="5":
            ratings[4]+=1
        if row["rating"]=="2":
            ratings[1]+=1
        if row["rating"]=="3":
            ratings[2]+=1
        if row["rating"]=="4":
            ratings[3]+=1
        if row["rating"]=="1":
            ratings[0]+=1
        if row["rating"] in (None,""):
            omittedRatings+=1
    print("Amount scoring 1: ", ratings[0])
    print("Amount scoring 2: ", ratings[1])
    print("Amount scoring 3: ", ratings[2])
    print("Amount scoring 4: ", ratings[3])
    print("Amount scoring 5: ", ratings[4])
    print("Amount missing: ", omittedRatings)
```