csvsplitter.py

Purpose:

To support data_gen_mk2.py in separating the merged csv file, with the six years of data, into item specific files. This is needed as the Machine Learning model requires individual files to run against to predict order quantities.

Libraries used:

```
import pandas as pd
import os
```

Only two libraries were used in this program, pandas and os.

- import pandas as pd – A powerful data analysis tool. Used to open the csv files and separate based on column.
  https://pandas.pydata.org/docs/user_guide/10min.html
- Import os – Allows for the use of operating system dependent functionality.
  https://docs.python.org/3/library/os.html

Classes:

None used here.

Global variables:

None.

Operation:

```
def seperate(fileDescriptor):
    pos = os.getcwd()
    data = pd.read_csv(pos+"\\"+ fileDescriptor + ".csv",encoding = "ISO-8859-1")

    data_category_range = data['Item'].unique()
    data_category_range = data_category_range.tolist()
    folder = file_check("individual_csv")
    for value in enumerate(data_category_range):
        placement = pos+"\\"+folder+'\\Item_'+str(value)+'_.csv'
        data[data['Item'] == value].to_csv(placement,index = False, na_rep = 'N/A')
    get_files(pos+"\\"+folder)
```

The operation of this program is very small in scope, separate a csv into many csv. Separate() takes in the folder location and extracts all csv files. Next, it sets which column to separate the items with, for this we are separating by 'Item'. After, a folder is created to store the new separated files. File_check() works in the same fashion as file_check() from data_gen_mk2.py. The for loop then separates the csv into item specific csv's and saves them to the folder created. Get_files() is just a check for the correct placement of the files and offers no other functionality at this time.

```
def get_files(path):
    mylist = os.listdir(path)
    print(mylist)
```