file handling:it is used to handle different format of files like excel,json,csv,sql,txt modes of files: r---read mode w---write mode a---append mode r+----read and write mode b---binary mode(images,audios and videos)

```
In [ ]: CRUD operations(create,read,update and delete)
```

```
In [ ]: open()----use to open the file
        close()---use to close the file
        read()----use to read the file
        write()---use to write the file
```

```
In [ ]: x=open("filename","mode")
        x.close()
        x.read()
        x.write()
```

```
In [24]: y=open("demo__file.txt","r")
```

```
In [26]: y.read()
```

```
Out[26]: 'The different modes are read,write,append etc'
```

```
In [28]: y=open("demo__file.txt","a")
         y.write("The different modes are read,write,append etc")
```

```
Out[28]: 45
```

```
In [30]: y=open("demo__file.txt","r")
```

```
In [32]: y.read()
```

```
Out[32]: 'The different modes are read,write,append etcThe different modes are read,write,append etc'
```

```
In [34]: y=open("demo__file.txt","w")
         y.write("The different modes are read,write,append etc")
```

```
Out[34]: 45
```

```
In [36]: y=open("demo__file.txt","r")
```

```
In [38]: y.read()
```

```
Out[38]: 'The different modes are read,write,append etc'
```

```
In [40]: with open("demo__file.txt","r") as x:
             print(x.read())
```

```
The different modes are read,write,append etc
```

```
In [42]: pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\pc\new folder\python\lib\site-packages (2.2.3)
Requirement already satisfied: numpy>=1.26.0 in c:\users\pc\new folder\python\lib\site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\pc\new folder\python\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\pc\new folder\python\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\pc\new folder\python\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\pc\new folder\python\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

```
WARNING: Ignoring invalid distribution ~umpy (C:\Users\pc\New folder\python\Lib\site-packages)
WARNING: Ignoring invalid distribution ~umpy (C:\Users\pc\New folder\python\Lib\site-packages)
WARNING: Ignoring invalid distribution ~umpy (C:\Users\pc\New folder\python\Lib\site-packages)
```

In [43]:
```
import pandas as pd
```

In [44]:
```
data=pd.read_csv("advertising.csv")
```

In [45]:
```
data
```

Out[45]:

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Timestamp	Clicked on Ad
0	68.95	35	61833.90	256.09	Cloned 5thgeneration orchestration	Wrightburgh	0	Tunisia	2016-03-27 00:53:11	0
1	80.23	31	68441.85	193.77	Monitored national standardization	West Jodi	1	Nauru	2016-04-04 01:39:02	0
2	69.47	26	59785.94	236.50	Organic bottom-line service-desk	Davidton	0	San Marino	2016-03-13 20:35:42	0
3	74.15	29	54806.18	245.89	Triple-buffered reciprocal time-frame	West Terrifurt	1	Italy	2016-01-10 02:31:19	0
4	68.37	35	73889.99	225.58	Robust logistical utilization	South Manuel	0	Iceland	2016-06-03 03:36:18	0
...	...	...	...	...	...	...	...	...	...	...
995	72.97	30	71384.57	208.58	Fundamental modular algorithm	Duffystad	1	Lebanon	2016-02-11 21:49:00	1
996	51.30	45	67782.17	134.42	Grass-roots cohesive monitoring	New Darlene	1	Bosnia and Herzegovina	2016-04-22 02:07:01	1
997	51.63	51	42415.72	120.37	Expanded intangible solution	South Jessica	1	Mongolia	2016-02-01 17:24:57	1
998	55.55	19	41920.79	187.95	Proactive bandwidth-monitored policy	West Steven	0	Guatemala	2016-03-24 02:35:54	0
999	45.01	26	29875.80	178.35	Virtual 5thgeneration emulation	Ronniemouth	0	Brazil	2016-06-03 21:43:21	1

1000 rows × 10 columns

In [46]:
```
df=pd.read_excel("Sample_Superstore_Short.xlsx")
```

In [47]:
```
df
```

Out[47]:

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Region	Category	Sub-Category	Product Name	Sales	Quantity	Discount	Profit
0	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	South	Furniture	Bookcases	Bush Somerset Collection Bookcase	261.96	2	0	41.91
1	CA-2016-152157	2016-11-08	2016-11-11	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	West	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs	731.94	3	0	219.58
2	CA-2016-138688	2016-06-12	2016-06-16	Second Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	South	Office Supplies	Labels	Self-Adhesive Address Labels for Typewriters b...	14.62	2	0	6.87

In [66]:

```
import csv

with open("iris.csv","r") as file:
    reader=csv.reader(file)
    header=next(reader)
    for row in reader:
        if float(row[2])>=1.4:
            print(row)
```

```
['5.1', '3.5', '1.4', '0.2', 'Iris-setosa']
['4.9', '3', '1.4', '0.2', 'Iris-setosa']
['4.6', '3.1', '1.5', '0.2', 'Iris-setosa']
['5', '3.6', '1.4', '0.2', 'Iris-setosa']
['5.4', '3.9', '1.7', '0.4', 'Iris-setosa']
['4.6', '3.4', '1.4', '0.3', 'Iris-setosa']
['5', '3.4', '1.5', '0.2', 'Iris-setosa']
['4.4', '2.9', '1.4', '0.2', 'Iris-setosa']
['4.9', '3.1', '1.5', '0.1', 'Iris-setosa']
['5.4', '3.7', '1.5', '0.2', 'Iris-setosa']
['4.8', '3.4', '1.6', '0.2', 'Iris-setosa']
['4.8', '3', '1.4', '0.1', 'Iris-setosa']
['5.7', '4.4', '1.5', '0.4', 'Iris-setosa']
['5.1', '3.5', '1.4', '0.3', 'Iris-setosa']
['5.7', '3.8', '1.7', '0.3', 'Iris-setosa']
['5.1', '3.8', '1.5', '0.3', 'Iris-setosa']
['5.4', '3.4', '1.7', '0.2', 'Iris-setosa']
['5.1', '3.7', '1.5', '0.4', 'Iris-setosa']
['5.1', '3.3', '1.7', '0.5', 'Iris-setosa']
['4.8', '3.4', '1.9', '0.2', 'Iris-setosa']
['5', '3', '1.6', '0.2', 'Iris-setosa']
['5', '3.4', '1.6', '0.4', 'Iris-setosa']
['5.2', '3.5', '1.5', '0.2', 'Iris-setosa']
['5.2', '3.4', '1.4', '0.2', 'Iris-setosa']
['4.7', '3.2', '1.6', '0.2', 'Iris-setosa']
['4.8', '3.1', '1.6', '0.2', 'Iris-setosa']
['5.4', '3.4', '1.5', '0.4', 'Iris-setosa']
['5.2', '4.1', '1.5', '0.1', 'Iris-setosa']
['5.5', '4.2', '1.4', '0.2', 'Iris-setosa']
['4.9', '3.1', '1.5', '0.1', 'Iris-setosa']
['4.9', '3.1', '1.5', '0.1', 'Iris-setosa']
['5.1', '3.4', '1.5', '0.2', 'Iris-setosa']
['5', '3.5', '1.6', '0.6', 'Iris-setosa']
['5.1', '3.8', '1.9', '0.4', 'Iris-setosa']
['4.8', '3', '1.4', '0.3', 'Iris-setosa']
['5.1', '3.8', '1.6', '0.2', 'Iris-setosa']
['4.6', '3.2', '1.4', '0.2', 'Iris-setosa']
['5.3', '3.7', '1.5', '0.2', 'Iris-setosa']
['5', '3.3', '1.4', '0.2', 'Iris-setosa']
['7', '3.2', '4.7', '1.4', 'Iris-versicolor']
['6.4', '3.2', '4.5', '1.5', 'Iris-versicolor']
['6.9', '3.1', '4.9', '1.5', 'Iris-versicolor']
['5.5', '2.3', '4', '1.3', 'Iris-versicolor']
['6.5', '2.8', '4.6', '1.5', 'Iris-versicolor']
['5.7', '2.8', '4.5', '1.3', 'Iris-versicolor']
['6.3', '3.3', '4.7', '1.6', 'Iris-versicolor']
['4.9', '2.4', '3.3', '1', 'Iris-versicolor']
['6.6', '2.9', '4.6', '1.3', 'Iris-versicolor']
['5.2', '2.7', '3.9', '1.4', 'Iris-versicolor']
['5', '2', '3.5', '1', 'Iris-versicolor']
['5.9', '3', '4.2', '1.5', 'Iris-versicolor']
['6', '2.2', '4', '1', 'Iris-versicolor']
['6.1', '2.9', '4.7', '1.4', 'Iris-versicolor']
['5.6', '2.9', '3.6', '1.3', 'Iris-versicolor']
['6.7', '3.1', '4.4', '1.4', 'Iris-versicolor']
['5.6', '3', '4.5', '1.5', 'Iris-versicolor']
['5.8', '2.7', '4.1', '1', 'Iris-versicolor']
['6.2', '2.2', '4.5', '1.5', 'Iris-versicolor']
['5.6', '2.5', '3.9', '1.1', 'Iris-versicolor']
['5.9', '3.2', '4.8', '1.8', 'Iris-versicolor']
```

```
['6.1', '2.8', '4', '1.3', 'Iris-versicolor']
['6.3', '2.5', '4.9', '1.5', 'Iris-versicolor']
['6.1', '2.8', '4.7', '1.2', 'Iris-versicolor']
['6.4', '2.9', '4.3', '1.3', 'Iris-versicolor']
['6.6', '3', '4.4', '1.4', 'Iris-versicolor']
['6.8', '2.8', '4.8', '1.4', 'Iris-versicolor']
['6.7', '3', '5', '1.7', 'Iris-versicolor']
['6', '2.9', '4.5', '1.5', 'Iris-versicolor']
['5.7', '2.6', '3.5', '1', 'Iris-versicolor']
['5.5', '2.4', '3.8', '1.1', 'Iris-versicolor']
['5.5', '2.4', '3.7', '1', 'Iris-versicolor']
['5.8', '2.7', '3.9', '1.2', 'Iris-versicolor']
['6', '2.7', '5.1', '1.6', 'Iris-versicolor']
['5.4', '3', '4.5', '1.5', 'Iris-versicolor']
['6', '3.4', '4.5', '1.6', 'Iris-versicolor']
['6.7', '3.1', '4.7', '1.5', 'Iris-versicolor']
['6.3', '2.3', '4.4', '1.3', 'Iris-versicolor']
['5.6', '3', '4.1', '1.3', 'Iris-versicolor']
['5.5', '2.5', '4', '1.3', 'Iris-versicolor']
['5.5', '2.6', '4.4', '1.2', 'Iris-versicolor']
['6.1', '3', '4.6', '1.4', 'Iris-versicolor']
['5.8', '2.6', '4', '1.2', 'Iris-versicolor']
['5', '2.3', '3.3', '1', 'Iris-versicolor']
['5.6', '2.7', '4.2', '1.3', 'Iris-versicolor']
['5.7', '3', '4.2', '1.2', 'Iris-versicolor']
['5.7', '2.9', '4.2', '1.3', 'Iris-versicolor']
['6.2', '2.9', '4.3', '1.3', 'Iris-versicolor']
['5.1', '2.5', '3', '1.1', 'Iris-versicolor']
['5.7', '2.8', '4.1', '1.3', 'Iris-versicolor']
['6.3', '3.3', '6', '2.5', 'Iris-virginica']
['5.8', '2.7', '5.1', '1.9', 'Iris-virginica']
['7.1', '3', '5.9', '2.1', 'Iris-virginica']
['6.3', '2.9', '5.6', '1.8', 'Iris-virginica']
['6.5', '3', '5.8', '2.2', 'Iris-virginica']
['7.6', '3', '6.6', '2.1', 'Iris-virginica']
['4.9', '2.5', '4.5', '1.7', 'Iris-virginica']
['7.3', '2.9', '6.3', '1.8', 'Iris-virginica']
['6.7', '2.5', '5.8', '1.8', 'Iris-virginica']
['7.2', '3.6', '6.1', '2.5', 'Iris-virginica']
['6.5', '3.2', '5.1', '2', 'Iris-virginica']
['6.4', '2.7', '5.3', '1.9', 'Iris-virginica']
['6.8', '3', '5.5', '2.1', 'Iris-virginica']
['5.7', '2.5', '5', '2', 'Iris-virginica']
['5.8', '2.8', '5.1', '2.4', 'Iris-virginica']
['6.4', '3.2', '5.3', '2.3', 'Iris-virginica']
['6.5', '3', '5.5', '1.8', 'Iris-virginica']
['7.7', '3.8', '6.7', '2.2', 'Iris-virginica']
['7.7', '2.6', '6.9', '2.3', 'Iris-virginica']
['6', '2.2', '5', '1.5', 'Iris-virginica']
['6.9', '3.2', '5.7', '2.3', 'Iris-virginica']
['5.6', '2.8', '4.9', '2', 'Iris-virginica']
['7.7', '2.8', '6.7', '2', 'Iris-virginica']
['6.3', '2.7', '4.9', '1.8', 'Iris-virginica']
['6.7', '3.3', '5.7', '2.1', 'Iris-virginica']
['7.2', '3.2', '6', '1.8', 'Iris-virginica']
['6.2', '2.8', '4.8', '1.8', 'Iris-virginica']
['6.1', '3', '4.9', '1.8', 'Iris-virginica']
['6.4', '2.8', '5.6', '2.1', 'Iris-virginica']
['7.2', '3', '5.8', '1.6', 'Iris-virginica']
['7.4', '2.8', '6.1', '1.9', 'Iris-virginica']
```

```
['7.9', '3.8', '6.4', '2', 'Iris-virginica']
['6.4', '2.8', '5.6', '2.2', 'Iris-virginica']
['6.3', '2.8', '5.1', '1.5', 'Iris-virginica']
['6.1', '2.6', '5.6', '1.4', 'Iris-virginica']
['7.7', '3', '6.1', '2.3', 'Iris-virginica']
['6.3', '3.4', '5.6', '2.4', 'Iris-virginica']
['6.4', '3.1', '5.5', '1.8', 'Iris-virginica']
['6', '3', '4.8', '1.8', 'Iris-virginica']
['6.9', '3.1', '5.4', '2.1', 'Iris-virginica']
['6.7', '3.1', '5.6', '2.4', 'Iris-virginica']
['6.9', '3.1', '5.1', '2.3', 'Iris-virginica']
['5.8', '2.7', '5.1', '1.9', 'Iris-virginica']
['6.8', '3.2', '5.9', '2.3', 'Iris-virginica']
['6.7', '3.3', '5.7', '2.5', 'Iris-virginica']
['6.7', '3', '5.2', '2.3', 'Iris-virginica']
['6.3', '2.5', '5', '1.9', 'Iris-virginica']
['6.5', '3', '5.2', '2', 'Iris-virginica']
['6.2', '3.4', '5.4', '2.3', 'Iris-virginica']
['5.9', '3', '5.1', '1.8', 'Iris-virginica']
```

In [60]:

```
df=pd.read_csv("iris.csv")
```

In [62]:

```
df
```

Out[62]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [ ]:

```
Domain:                      usage
data science                 dataset loading
Ml                           training data,model saving
Automation                    reports,backups
```

In [ ]:

```
task 01:
count the number of errors
store the error lines separately
```

In [ ]:

```
task 02:
read the csv data (employees)
filter high salary employees
write the result to new file
```

In [ ]:

```
task 03:
generate daily report automatically
```

In [ ]:

```
Exception handling:
it is used to handle the run time errors

1.syntax errors
2.Exceptions
```

In [ ]:

```
1.syntax error:missed the colon,mispelled keyword,missed paranthesis
2.name error:when the name is not defined
3.index error:index is out of range
4.zero division error:we are trying to divide with 0
5.key error:when key is not found in dictionary
6.value error:whenever value is not found in the dictionary
7.attribute error:arguments/parameters are not defined
8.type error:whenever we are trying to add different data types
```

In [68]:

```
if 10>5
print("hi")
```

```
  Cell In[68], line 1
    if 10>5
           ^
SyntaxError: expected ':'
```

In [70]:

```
name="Shreya"
print(Name)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[70], line 2
      1 name="Shreya"
----> 2 print(Name)

NameError: name 'Name' is not defined
```

In [94]:

```
x=[1,2,3]
print(x[5])
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[94], line 2
      1 x=[1,2,3]
----> 2 print(x[5])

IndexError: list index out of range
```

In [74]:

```
x=0
y=10/x
print(y)
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[74], line 2
      1 x=0
----> 2 y=10/x
      3 print(y)

ZeroDivisionError: division by zero
```

```
In [100…  student={"name":"rahul","marks":45}
```

```
In [102…  student['age']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[102], line 1
----> 1 student['age']

KeyError: 'age'
```

```
In [80]:  x=int("abc")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[80], line 1
----> 1 x=int("abc")

ValueError: invalid literal for int() with base 10: 'abc'
```

```
In [84]:  name="python"
          print(name.append("5"))
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[84], line 2
      1 name="python"
----> 2 print(name.append("5"))

AttributeError: 'str' object has no attribute 'append'
```

```
In [86]:  x="shreya"
          y=34
          print(x+y)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[86], line 3
      1 x="shreya"
      2 y=34
----> 3 print(x+y)

TypeError: can only concatenate str (not "int") to str
```

```
In [90]:  marks=90
          try:
              y=marks/0
          except ZeroDivisionError:
              print("cannot divide by zero")
```

```
cannot divide by zero
```

In [92]:

```
try:
    x+y
except TypeError:
    print("cannot add int and string")
```

```
cannot add int and string
```

In [96]:

```
try:
    x[5]
except IndexError:
    print("index out of range")
```

```
index out of range
```

In [110…

```
student = {"name": "rahul", "marks": 45}

try:
    print(student['age'])
except KeyError:
    print("key does not exist in dictionary")
```

```
key does not exist in dictionary
```

In [112…

```
try:
    x=int("abc")
except ValueError:
    print("Invalid value for interger conversion")
```

```
Invalid value for interger conversion
```

In [ ]:

```
try:
    name="python"
    name.append("3")
except AttributeError:
    print("Invalid attribute or method for objects")
```

In [120…

```
data={"name":"asha","salary":"50000"}
try:
    salary=int(data["salary"])
    print("Bonus:",salary*0.1)

except KeyError:
    print("Missing salary key")
except ValueError:
    print("Salary should be numeric")
except TypeError:
    print("Invalid operation")
except AttributeError:
    print("Attribute error occured")
except Exception as e:
    print("unexpected error:",e)
```

```
Bonus: 5000.0
```

In [ ]:

```
Apply file handling+exception handling write any one use case code
```

In [124…

```
import csv
try:
    with open("demo.csv","r") as file:
        reader=csv.reader(file)
        header=next(reader)
        print("rows where petal_length>4.5\n")
        for row in reader:
            try:
                if float(row[2])>4.5:
                    print(row)
            except ValueError:
                print("invalid numeric data",row)

except FileNotFoundError:
    print("file you are looking for is not available")
except Exception as e:
    print("Unexpected error:",e)
```

```
file you are looking for is not available
```

In [ ]:

```
mini hands on questions:
handle division error
handle missing file and log error
validate user input(age,salary)
create custom exception for atm withdrawal
```

In [ ]:

```
Project 1: student record management

filename:students.txt

add the students
view the students
search the students
```

In [ ]:

```
project 2: word counter tool

use file handling

file name:summary.txt

give the exceptions
lines---len(lines)
words---len(words)
characters--len(text)
```

In [ ]:

```
project 3: employee salary analyzer(csv)
filename:employees.csv
highest salaries
lowest salaries
average salaries

file not found error if not exists---exception handling
```

In [ ]:

```
project 4: Atm simulation

functions
class(oops)
exceptions
conditional statements
assignment operator
```

In [ ]:

```
project 5:user login system

file names:user.txt
```

```
exceptions
file handling
function
conditional statements
```

In [ ]:

```
imagine i am having use case automated report generator
```

In [126…

```python
try:
    total_sales=0
    with open("sales.txt","r") as file:
        for line in file:
            total_sales+=int(line.strip())

    with open("report.txt","w") as report:
        report.write("Daily sales report\n")
        report.write(f"Total sales:{total_sales}")

    print("report generated")
except FileNotFoundError:
    print("sales file is missing")
except ValueError:
    print("invalid sales data")
```

```
sales file is missing
```

In [ ]: