Reading text file
In order to read content of the file, the file must be opened in “r” mode
File object provides the following methods read

1. read()
2. readline()

read(size=-1)
Read and return at most size characters from the stream as a single str. If size is negative or None, reads until EOF.

Example:
```
def main():
    try:
        f=open("d:\\file1.txt","r")
        s=f.read()
        print(s)
    except OSError as o:
        print(o)
    finally:
        f.close()
main()
```

Output:
```
Python is a programming language
>>>
```

Example:
```
# write a program to count how many alphabets exists inside file1.txt

def main():
    try:
        f=open("d:\\file1.txt","r")
        c=0
        while True:
            ch=f.read(1)
            if ch=='':
                break
            if (ch>='a' and ch<='z') or (ch>='A' and ch<='Z'):
                c=c+1
        print(f'count of alphabets {c}')
    except OSError as x:
        print(x)
    finally:
        f.close()
main()
```

Output:

```
count of alphabets 28
>>>
```

Example:

```
# write a program to copy the content of one file to another file
def main():
    try:
        f1=open("d:\\file1.txt","r")
        f2=open("d:\\file1_copy.txt","w")
        s=f1.read()
        f2.write(s.upper())
        print("file copied")
    except OSError as x:
        print(x)

main()
```

Output:

```
file copied
```

readline(size=-1)

Read until newline or EOF and return a single str. If the stream is already at EOF, an empty string is returned.
If size is specified, at most size characters will be read.

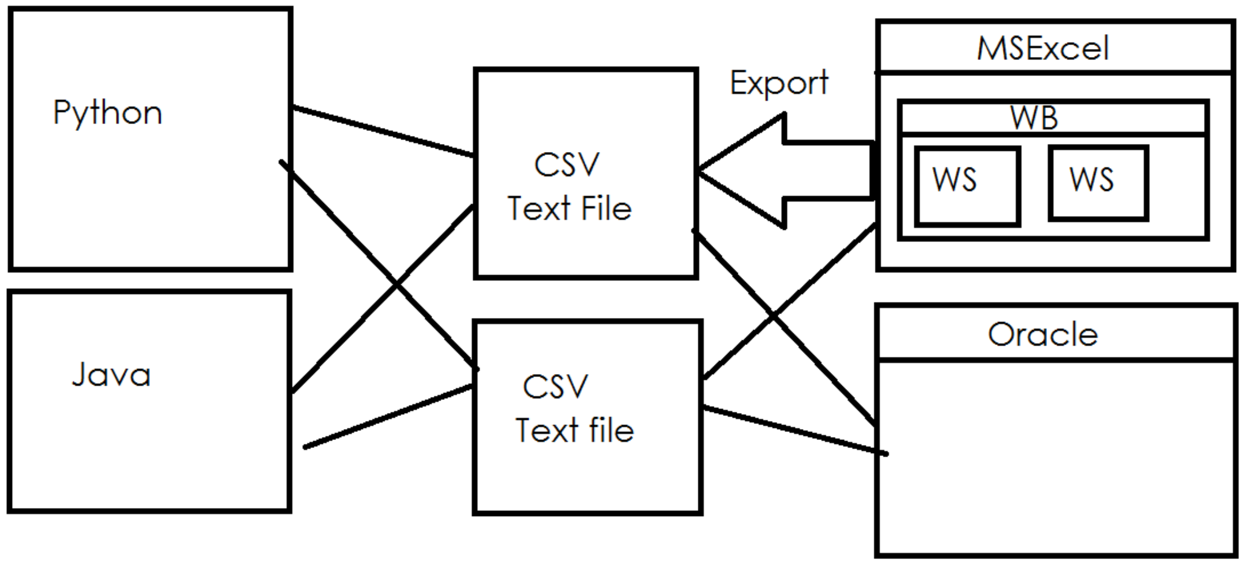
```
def main():
    try:
        f=open("d:\\file2.txt","r")
        str1=f.readline()
        print(str1)
        str2=f.readline()
        print(str2)
        str3=f.readline()
        print(str3)
        f.close()
        f=open("d:\\file2.txt","r")
        while True:
            line=f.readline()
            if line=='':
                break
            print(line)

    except OSError as x:
        print(x)
```

Output:

```
java

oracle

python

java

oracle

python

mysql
```

CSV file (OR) CSV module

The so-called CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases.



The csv module implements classes to read and write tabular data in CSV format. It allows programmers to say, “write this data in the format preferred by Excel,” or “read data from this file which was generated by Excel,” without knowing the precise details of the CSV format used by Excel. Programmers can also describe the CSV formats understood by other applications or define their own special-purpose CSV formats.

The csv module’s reader and writer objects read and write sequences. Programmers can also read and write data in dictionary form using the DictReader and DictWriter classes.

reader
csv.reader(*csvfile)*

return a reader object which will iterate over lines in the given *csvfile*. *csvfile* can be any object which supports the iterator protocol and returns a string each time its __next__() method is called — file objects and list objects are both suitable

```
import csv
def main():
    f=open("employees.csv","r")
    r=csv.reader(f)
    i=iter(r)
    for row in i:
        print(row)
    f.close()
    f=open("employees.csv","r")
    r=csv.reader(f)
    i=iter(r)
    employees_list=list(i)
    print(employees_list)
    total=0
    for  i in range(1,len(employees_list)):
        total=total+float(employees_list[i][2])
    print(f'Total salary {total}')

main()
```

Output:

```
========== RESTART: C:/Users/user/Desktop/python6pm/py182.py
==========
['empno', 'ename', 'salary']
['1', 'aaa', '50000']
['2', 'bbb', '45000']
['3', 'ccc', '45001']
['4', 'ddd', '45002']
['5', 'eee', '65000']
['6', 'fff', '55000']
['7', 'ddd', '45000']
['8', 'sss', '25000']
['9', 'ddd', '15000']
['10', 'ddd', '12000']
['11', 'ccc', '11000']
['12', 'xyz', '10000']
['13', 'sss', '19000']
['14', 'ggg', '22000']
['15', 'eee', '45000']
['16', 'yy', '65000']
['17', 'ttt', '90000']
['18', 'hhh', '34000']
['19', 'iii', '24000']
['20', 'kkk', '23000']
['21', 'ppp', '43000']
['22', 'ooo', '32000']
```

```
['23', 'ttt', '34000']
['24', 'rrrr', '33000']
[['empno', 'ename', 'salary'], ['1', 'aaa', '50000'], ['2', 'bbb', '45000'], ['3', 'ccc',
'45001'], ['4', 'ddd', '45002'], ['5', 'eee', '65000'], ['6', 'fff', '55000'], ['7', 'ddd',
'45000'], ['8', 'sss', '25000'], ['9', 'ddd', '15000'], ['10', 'ddd', '12000'], ['11',
'ccc', '11000'], ['12', 'xyz', '10000'], ['13', 'sss', '19000'], ['14', 'ggg', '22000'],
['15', 'eee', '45000'], ['16', 'yy', '65000'], ['17', 'ttt', '90000'], ['18', 'hhh',
'34000'], ['19', 'iii', '24000'], ['20', 'kkk', '23000'], ['21', 'ppp', '43000'], ['22',
'ooo', '32000'], ['23', 'ttt', '34000'], ['24', 'rrrr', '33000']]
Total salary 887003.0
['empno', 'ename', 'salary']
['1', 'aaa', '50000']
['2', 'bbb', '45000']
['3', 'ccc', '45001']
['4', 'ddd', '45002']
['5', 'eee', '65000']
['6', 'fff', '55000']
['7', 'ddd', '45000']
['8', 'sss', '25000']
['9', 'ddd', '15000']
['10', 'ddd', '12000']
['11', 'ccc', '11000']
['12', 'xyz', '10000']
['13', 'sss', '19000']
['14', 'ggg', '22000']
['15', 'eee', '45000']
['16', 'yy', '65000']
['17', 'ttt', '90000']
['18', 'hhh', '34000']
['19', 'iii', '24000']
['20', 'kkk', '23000']
['21', 'ppp', '43000']
['22', 'ooo', '32000']
['23', 'ttt', '34000']
['24', 'rrrr', '33000']
[['empno', 'ename', 'salary'], ['1', 'aaa', '50000'], ['2', 'bbb', '45000'], ['3', 'ccc',
'45001'], ['4', 'ddd', '45002'], ['5', 'eee', '65000'], ['6', 'fff', '55000'], ['7', 'ddd',
'45000'], ['8', 'sss', '25000'], ['9', 'ddd', '15000'], ['10', 'ddd', '12000'], ['11',
'ccc', '11000'], ['12', 'xyz', '10000'], ['13', 'sss', '19000'], ['14', 'ggg', '22000'],
['15', 'eee', '45000'], ['16', 'yy', '65000'], ['17', 'ttt', '90000'], ['18', 'hhh',
'34000'], ['19', 'iii', '24000'], ['20', 'kkk', '23000'], ['21', 'ppp', '43000'], ['22',
'ooo', '32000'], ['23', 'ttt', '34000'], ['24', 'rrrr', '33000']]
Total salary 887003.0
```

DictReader