File Handling

- As the part of programming requirement, we have to store our data permanently for future purpose. For this requirement we should go for files.
- Files are very common permanent storage areas to store our data.

Types of Files:

There are 2 types of files

1. Text Files: Usually we can use text files to store character data eg: abc.txt
2. Binary Files: Usually we can use binary files to store binary data like images,video files, audio files etc...

Opening a File:

Before performing any operation (like read or write) on the file,first we have to open that file.For this we should use Python's inbuilt function open() But at the time of open, we have to specify mode,which represents the purpose of opening file.

f = open(filename, mode)

The allowed modes in Python are:

1. r -> open an existing file for read operation. The file pointer is positioned at the beginning of the file.If the specified file does not exist then we will get FileNotFoundError.This is default mode.
2. w -> open an existing file for write operation. If the file already contains some data then it will be overridden. If the specified file is not already avaialble then this mode will create that file.
3. a -> open an existing file for append operation. It won't override existing data.If the specified file is not already avaialble then this mode will create a new file.
4. r+ -> To read and write data into the file. The previous data in the file will not be deleted.The file pointer is placed at the beginning of the file.
5. w+ -> To write and read data. It will override existing data.
6. a+ -> To append and read data from the file.It wont override existing data.
7. x -> To open a file in exclusive creation mode for write operation. If the file already exists then we will get FileExistsError.

Note: All the above modes are applicable for text files. If the above modes suffixed with 'b' then these represents for binary files.

Eg: rb,wb,ab,r+b,w+b,a+b,xb

f = open("abc.txt","w")

We are opening abc.txt file for writing data.

Closing a File:

After completing our operations on the file,it is highly recommended to close the file. For this we have to use close() function.

f.close()

Various properties of File Object:

Once we opend a file and we got file object,we can get various details related to that file by using its properties.

- name -> Name of opened file
- mode -> Mode in which the file is opened
- closed -> Returns boolean value indicates that file is closed or not
- readable() -> Retruns boolean value indicates that whether file is readable or not
- writable() -> Returns boolean value indicates that whether file is writable or not.

In [1]:

```
f=open("abc.txt",'w')
print("File Name: ",f.name)
print("File Mode: ",f.mode)
print("Is File Readable: ",f.readable())
print("Is File Writable: ",f.writable())
print("Is File Closed : ",f.closed)
f.close()
print("Is File Closed : ",f.closed)
```

```
File Name:  abc.txt
File Mode:  w
Is File Readable:  False
Is File Writable:  True
Is File Closed :  False
Is File Closed :  True
```

Writing data to text files:

We can write character data to the text files by using the following 2 methods.

- write(str)
- writelines(list of lines)

In [2]:

```
f=open("abcd.txt",'w')
f.write("Parth\n")
f.write("Sinroza\n")
f.write("CSE\n")
print("Data written to the file successfully")
f.close()
```

```
Data written to the file successfully
```

Note: In the above program, data present in the file will be overridden everytime if we run the program. Instead of overriding if we want append operation then we should open the file as follows.

In [3]:

```
f=open("abcd.txt",'w')
list=["sunny\n","bunny\n","vinny\n","chinny"]
f.writelines(list)
print("List of lines written to the file successfully")
f.close()
f=open("abcd.txt")
print(f.read())
f.close()
```

```
List of lines written to the file successfully
sunny
bunny
vinny
chinny
```

Note: while writing data by using write() methods, compulsory we have to provide line seperator(\n),otherwise total data should be written to a single line.

Reading Character Data from text files:

We can read character data from text file by using the following read methods.

- read()-> To read total data from the file
- read(n) -> To read 'n' characters from the file
- readline() -> To read only one line
- readlines() -> To read all lines into a list

Eg 1: To read total data from the file

In [4]:

```
f=open("abcd.txt",'r')
data=f.read()
print(data)
f.close()
```

```
sunny
bunny
vinny
chinny
```

Eg 2: To read only first 10 characters:

In [5]:

```
f=open("abcd.txt",'r')
data=f.read(10)
print(data)
f.close()
```

```
sunny
bunn
```

Eg 3: To read data line by line:

In [7]:

```
f=open("abcd.txt",'r')
line1=f.readline()
print(line1,end='')
line2=f.readline()
print(line2,end='')
line3=f.readline()
print(line3,end='')
f.close()
```

```
sunny
bunny
vinny
```

Eg 4: To read all lines into list:

In [8]:

```
f=open("abcd.txt",'r')
lines=f.readlines()
for line in lines:
    print(line,end='')
f.close()
```

```
sunny
bunny
vinny
chinny
```

Eg 5:

In [9]:

```
f=open("abcd.txt","r")
print(f.read(3))
print(f.readline())
print(f.read(4))
print("Remaining data")
print(f.read())
```

```
sun
ny

bunn
Remaining data
y
vinny
chinny
```

The with statement:

The with statement can be used while opening a file.We can use this to group file operation statements within a block.

The advantage of with statement is it will take care closing of file,after completing all operations automatically even in the case of exceptions also, and we are not required to close explicitly.

Eg:

In [10]:

```
with open("abc.txt","w") as f:
    f.write("Parth\n")
    f.write("Sinroza\n")
    f.write("LJU\n")
    print("Is File Closed: ",f.closed)
print("Is File Closed: ",f.closed)
```

```
Is File Closed:  False
Is File Closed:  True
```

The seek() and tell() methods:

tell():

- We can use tell() method to return current position of the cursor(file pointer) from beginning of the file.
- The position(index) of first character in files is zero just like string index.

Eg:

In [11]:

```
f=open("abc.txt","r")
print(f.tell())
print(f.read(2))
print(f.tell())
print(f.read(3))
print(f.tell())
```

```
0
Pa
2
rth
5
```

seek():

We can use seek() method to move cursor(file pointer) to specified location.

f.seek(offset, fromwhere)

offset represents the number of positions

The allowed values for second attribute(from where) are

0---->From beginning of file(default value)

1---->From current position

2--->From end of the file

Note: Python 2 supports all 3 values but Python 3 supports only zero.

Eg:

In [12]:

```
data="All Humans are GOOD"
f=open("abc.txt","w")
f.write(data)
with open("abc.txt","r+") as f:
    text=f.read()
    print(text)
    print("The Current Cursor Position: ",f.tell())
    f.seek(15)
    print("The Current Cursor Position: ",f.tell())
    f.write("GEMS!!!")
    f.seek(0)
    text=f.read()
    print("Data After Modification:")
    print(text)
```

```
All Humans are GOOD
The Current Cursor Position:  19
The Current Cursor Position:  15
Data After Modification:
All Humans are GEMS!!!
```

WAP to read the file using readline() and print the entire content of the file.

In [13]:

```
f=open('file1.txt','r')
while True:
    data=f.readline()
    if data=='':
        break
    else:
        print(data,end="")
f.close()
```

```
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
```

WAP to create a file having 'Hello World' printed in 1000 lines.

In [15]:

```
big_L=['hello world\n' for i in range(1000)]
with open('big.txt','w') as f:
    f.writelines(big_L)
```

WAP to count no of lines in a text file.

In [16]:

```
with open('big.txt') as f:
    count=0
    for i in f:
        count+=1
print(count)
```

```
1000
```

WAP to count words, characters and spaces in a text file

In [2]:

```
space=0
ch=0
f=open('file1.txt')
data=f.read()
for count in data:
    ch+=1
    if count==" ":
        space+=1
words=data.split()
len_words=len(words)
f.close()
print('Total Characters in the file:',ch)
print('Total spaces in the file:',space)
print('Total words in the file',len_words)
print(data)
```

```
Total Characters in the file: 74
Total spaces in the file: 15
Total words in the file 20
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
```

WAP to create a list that contains the first word of every line in text file

In [3]:

```
with open('file1.txt') as f:
    e=[]
    c=f.readlines()
    for i in c:
        a=i.split()
        e.append(a[0])
print(e)
```

```
['This', 'This', 'This', 'This', 'This']
```

WAP to add all the words that has character 'e' to a list from a file

In [5]:

```
with open('file1.txt', 'r') as f:
    p=[]
    x=f.read().split()
    for i in x:
        if 'e' in i:
            p.append(i)
print(p)
```

```
['line', 'line', 'line', 'line', 'line']
```

WAP that reads data from a file and calculates the percentage of vowels and consonants in the file

In [12]:

```
with open ('file1.txt') as file:
    text=file.read()
    count_vowels=0
    count_consonants=0
    for char in text:
        if char in 'aeiouAEIOU':
            count_vowels+=1
        elif char.isalpha():
            count_consonants+=1
        else:
            pass
print('No. of vowels:',count_vowels)
print('No. of consonants:',count_consonants)
print('Total characters:',len(text))
print(f'% of vowels: {((count_vowels*100)/len(text)):0.2f}')
print(f'% of consonants: {((count_consonants*100)/len(text)):0.2f}')
```

```
No. of vowels: 20
No. of consonants: 30
Total characters: 74
% of vowels: 27.03
% of consonants: 40.54
```

WAP that reads a file line by line. Each line read from a file is copied to another file with line nos specified as prefix of line

In [14]:

```
file1=open('file1.txt')
file2=open('file2.txt','w')
num=1
for line in file1:
    file2.write(str(num)+": "+line)
    num+=1
file1.close()
file2.close()
with open('file2.txt') as f:
    print(f.read())
```

```
1: This is line 1
2: This is line 2
3: This is line 3
4: This is line 4
5: This is line 5
```

WAP to create a dictionary having name, roll no and branch as keys. Copy the dictionary to a file database.txt

In [15]:

```
details={'Name:':'Parth','Roll No:':'001','Branch:':'CSE'}
with open('database.txt','w') as file:
    for key,values in details.items():
        file.write(key+' '+values+'\n')
with open('database.txt') as f:
    print(f.read())
```

```
Name: Parth
Roll No: 001
Branch: CSE
```

WAP to copy all content of one file to another file in uppercase.

In [16]:

```
f1=open('file1.txt')
f2=open('file2.txt','w')
for line in f1:
    f2.write(line.upper())
f1.close()
f2.close()
with open('file2.txt') as f:
    print(f.read())
```

```
THIS IS LINE 1
THIS IS LINE 2
THIS IS LINE 3
THIS IS LINE 4
THIS IS LINE 5
```

WAP to copy all content of one file to another file in reverse order.

In [17]:

```
f1=open('file1.txt')
f2=open('file2.txt','w')
data=f1.read()
data1=data[::-1]
f2.write(data1)
f1.close()
f2.close()
with open('file2.txt') as f:
    print(f.read())
```

```
5 enil si sihT
4 enil si sihT
3 enil si sihT
2 enil si sihT
1 enil si sihT
```

WAP to read specific lines from a file.

In [18]:

```
f=open('file1.txt')
content=f.readlines()
print(content[2])
print(content[0:2])
for i in content[0:2]:
    print(i,end="")
```

```
This is line 3

['This is line 1\n', 'This is line 2\n']
This is line 1
This is line 2
```

WAP to copy odd lines of one file to another.

In [19]:

```
f1=open('file1.txt')
f2=open('copy.txt','w')
data=f1.readlines()
for i in range (len(data)):
    if i%2==0:
        f2.write(data[i])
    else:
        pass
f1.close()
f2.close()
with open('copy.txt') as f:
    print(f.read())
```

```
This is line 1
This is line 3
This is line 5
```

WAP to find the longest word from a file.

In [20]:

```
with open('copy.txt', 'r') as infile:
    words = infile.read().split()
    max_len = len(max(words, key=len))
    print([word for word in words if len(word) == max_len])
```

```
['pythonprogramming']
```

WAP to read the file and replace the letter 'j' with 'i'. The file is miswritten and has content as 'Well thjs js a word by jtself. You can strech thjs to be a sentence'.|

In [22]:

```
with open('copy.txt') as f:
    data=f.read()
    for letter in data:
        if letter=='j':
            print('i',end='')
        else:
            print(letter,end='')
```

```
Well this is a word by itself. You can strech this to be a sentence
```

WAP such that your solution should promt for a file name and display the text file's 25 lines at a time, pausing each time to ask the user to 'press any key to continue or q to quit and exit'.

In [26]:

```
file=open(input('Enter file name: '))
num_lines=int(input('How many lines to print: '))
for index,line in enumerate(file,1):
    print(index,':',line)
    if index%num_lines==0:
        x=input('Hit any key to continue, press q to quit')
        if x.lower()=='q':
            break
file.close()
```

```
Enter file name: big.txt
How many lines to print: 25
1 : hello world

2 : hello world

3 : hello world

4 : hello world

5 : hello world

6 : hello world

7 : hello world

8 : hello world

9 : hello world

10 : hello world

11 : hello world

12 : hello world

13 : hello world

14 : hello world

15 : hello world

16 : hello world

17 : hello world

18 : hello world

19 : hello world

20 : hello world

21 : hello world

22 : hello world

23 : hello world

24 : hello world

25 : hello world

Hit any key to continue, press q to quita
26 : hello world

27 : hello world

28 : hello world

29 : hello world

30 : hello world
```

```
31 : hello world

32 : hello world

33 : hello world

34 : hello world

35 : hello world

36 : hello world

37 : hello world

38 : hello world

39 : hello world

40 : hello world

41 : hello world

42 : hello world

43 : hello world

44 : hello world

45 : hello world

46 : hello world

47 : hello world

48 : hello world

49 : hello world

50 : hello world

Hit any key to continue, press q to quitq
```

WAP to copy a python code from a text file to another file. The copied content should not print the comments present in the python code.

Method 1:

In [31]:

```
f1=open('copy.txt')
f2=open('filter.txt','w')
while True:
    x=f1.readline()
    if len(x)!=0:
        if x[0]=='#':
            continue
        else:
            if '#' in x:
                for index in range (1,len(x)):
                    if x[index]=='#':
                        f2.write(x[0:index]+'\n')
            else:
                f2.write(x)
    else:
        break
f1.close()
f2.close()
with open('copy.txt') as f1:
    print(f1.read())
print()
print('New File:')
with open('filter.txt') as f2:
    print(f2.read())
```

```
#This is line 1
This is line 2
This is line 3
This is line #4
This is #line 5

New File:
This is line 2
This is line 3
This is line
This is
```

Method 2:

In [32]:

```
with open('copy.txt') as f:
    tmp=[]
    lst=f.readlines()
    for i in lst:
        tmp.append(i[:i.find('#')])
with open('filter2.txt','w') as f:
    for i in tmp:
        if i!= '':
            f.writelines(i+'\n')
f=open('filter2.txt')
print(f.read())
f.close()
```

```
This is line 2
This is line 3
This is line
This is
```

WAP to compare two text files, if they are different, give the row and column numbers in the files where the first difference occurs.

In [34]:

```
f1=open('file1.txt')
f2=open('copy.txt')
fone=f1.readlines()
ftwo=f2.readlines()
f1.close()
f2.close()
for index,line in enumerate(fone):
    if line!=ftwo[index]:
        for ind,char in enumerate(line):
            if char!=ftwo[index][ind]:
                print('Row number:',index+1,'column:',ind+1)
                break
        break
```

```
Row number: 2 column: 6
```