



Python Programming - 2301CS404

Lab - 9

223 | Vishal Baraiya |
23010101014

File I/O

01) WAP to read and display the contents of a text file. (also try to open the file in some other directory)

- in the form of a string
- line by line
- in the form of a list

```
In [14]: # Read by File Pointer in another directory
# fp = open("D:\\Media\\demo.txt", "r")
# for i in fp:
#     print(i, end="")
# fp.close()

# Read by File Pointer
fp = open("Demo.txt", "r")
for i in fp:
    print(i, end="")
fp.close()

# Read by Read()
print("\n")
fp = open("Demo.txt", "r")
print(fp.read())
fp.close()
```

```

fp = open("Demo.txt","r")
while True:
    line = fp.readline() # Read a single line
    if not line:         # Break the loop if EOF is reached
        break
    print(line,end="")
fp.close()

# Read by Readlines()
print("\n")
fp = open("Demo.txt","r")
l = fp.readlines()
print(l)
fp.close()

```

I am Python.
This is Demo Text File.

I am Python.
This is Demo Text File.

I am Python.
This is Demo Text File.

```
['I am Python.\n', 'This is Demo Text File.']
```

02) WAP to create file named "new.txt" only if it doesn't exist.

```

In [17]: with open("new.txt","x") as fp:
          fp.write("Hello World!")

          with open("new.txt","r") as fp:
              print(fp.read())

```

Hello World!

03) WAP to read first 5 lines from the text file.

```

In [19]: # Open the file
          with open('demo.txt', 'r') as fp:
              for i in range(5):
                  line = file.readline()
                  if not line: # Break if the file has less than 5 lines
                      break
                  print(line.strip()) # Print the line without extra whitespace

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor in
cididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliqui
p ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu f
ugiat nulla pariatur.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserun
t mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor in
cididunt ut labore et dolore magna aliqua.

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

04) WAP to find the longest word(s) in a file

```
In [33]: with open("demo.txt","r") as fp:
          l = fp.read().split()
          maxLengthS = " "
          for i in l:
              if (len(i) > len(maxLengthS)):
                  maxLengthS = i
          print("Highest Length String :",maxLengthS)
```

Highest Length String : reprehenderit

05) WAP to count the no. of lines, words and characters in a given text file.

```
In [45]: with open("demo.txt","r") as fp:
          print("No. Of Lines : ",len(fp.readlines()))
          fp.seek(0,0)
          print("No. Of Words : ",len(fp.read().split()))
          fp.seek(0,0)
          print("No. Of Charcters : ",len(fp.read().replace("\n","")))
```

No. Of Lines : 8
No. Of Words : 138
No. Of Charcters : 884

06) WAP to copy the content of a file to the another file.

```
In [46]: s = " "
          with open("demo.txt","r") as fp:
              s = fp.read()
          with open("demo-copy.txt","w") as fp:
              fp.write(s)
```

07) WAP to find the size of the text file.

```
In [47]: import os

          # Specify the file path
          file_path = "demo.txt"

          # Check if the file exists
          if os.path.exists(file_path):
              # Get the size of the file in bytes
              file_size = os.path.getsize(file_path)
              print(f"The size of the file '{file_path}' is {file_size} bytes.")
          else:
              print(f"The file '{file_path}' does not exist.")
```

The size of the file 'demo.txt' is 891 bytes.

08) WAP to create an UDF named frequency to count occurances of the specific word in a given text file.

```
In [52]: def frequency(file_path,word):
    with open(file_path,'r') as fp:
        l = fp.read().split()
        count = 0
        for i in l:
            if word.lower() == i.lower():
                count += 1
        print(f"Count Of {word} is = {count}")

frequency("demo.txt","Lorem")
```

Count Of Lorem is = 2

09) WAP to get the score of five subjects from the user, store them in a file. Fetch those marks and find the highest score.

```
In [7]: with open("scores.txt","w") as fp:
    scores = []
    n = int(input("Enter the Length of List : "))
    for i in range(0,n,1):
        temp = int(input(f"Enter the Score of {i}th subject : "))
        scores.append(temp)
    for i in scores:
        fp.write(str(i))
        fp.write("\n")

    with open("scores.txt","r") as fp:
        l = fp.read().split()
        print(f"Max Score : {max(l)}")
```

Max Score : 89

10) WAP to write first 100 prime numbers to a file named primenumbers.txt

(Note: each number should be in new line)

```
In [11]: def isPrime(n = int):
    for i in range(2,n//2+2):
        if(n % i == 0):
            return False
    else:
        return True

    with open("primenumbers.txt","w") as fp:
        i,count = 2,0
        while(count != 100):
            if isPrime(i):
                fp.write(str(i))
                fp.write("\n")
                count += 1
            i += 1
        print("File Write Successfully!")
```

File Write Successfully!

```
In [13]: fp1 = open("scores.txt", "r")
fp2 = open("primenumbers.txt")
fp3 = open("merged.txt", "w")
fp3.write(fp1.read())
fp3.write(fp2.read())
fp1.close()
fp2.close()
fp3.close()
print("File Merged Successfully!")
```

File Merged Successfully!

12) WAP to replace word1 by word2 of a text file. Write the updated data to new file.

```
In [27]: word1 = input("Enter th word1 : ")
word2 = input("Enter th word2 : ")
s = " "
fp1 = open("demo.txt", "r")
fp2 = open("new.txt", "w")
s = fp1.read()
fp2.write(s.replace(word1, word2))
fp1.close()
fp2.close()
print("Successfully complated!")
```

Successfully complated!

13) Demonstrate tell() and seek() for all the cases (seek from beginning-end-current position) taking a suitable example of your choice.

```
In [31]: with open("demo.txt", "rb") as fp:
print(fp.read(10))
print("Current Position : ", fp.tell())
fp.seek(3, 0) # 0 is use for from begining to 3 offset
print("Current Position : ", fp.tell())
fp.seek(5, 1) # 1 is use for from Current Position to 5 offset
print("Current Position : ", fp.tell())
fp.seek(-3, 2) # 2 is use for from ending to -3 offset
print("Current Position : ", fp.tell())

# seek() second args 1,2 use in only binary files
```

```
b'Lorem ipsu'
Current Position : 10
Current Position : 3
Current Position : 8
Current Position : 918
```

In []: