Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

Lab - 9

Charmi Bhalodiya

23010101020

4B 448 8th batch

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 [13]:
```
#in the form of a string
fp = open("Rency.txt","r")
str1 = fp.read()
print(str1)
print(type(str1))
fp.close()

#line by line
fp = open("Rency.txt","r")
li = fp.readlines()
print(li)
print(type(li))
fp.close()

#in the form of a list
fp = open("Rency.txt","r")
line1 = fp.readline()
print(line1, end="")
# print(type(line1))
line2 = fp.readline()
print(line2,end="")
line3 = fp.readline()
print(line3,end="")
print(fp.read())
fp.close()
```

```
Hello World
hi every one
hiii
<class 'str'>
['Hello World\n', 'hi every one\n', 'hiii']
<class 'list'>
Hello World
hi every one
hiii
```

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

In [1]:
```
fp =open("new.txt","x")
fp.close()
```

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

In [19]:
```
with open("Rency.txt", 'r') as file:
          for i in range(5):
           line = file.readline()
           if line:
               print(line.strip())
           else:
               break
```

```
Hello World
hi every one
hiii
abc
def
```

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

In [23]:
```
fp = open("Rency.txt","r")
wl = fp.read().split()
w_len = list(map(len,wl))
max_len = max(w_len)
ans = [i for i in wl if len(i) == max_len]
print(ans)
fp.close()
```

```
['Hello', 'World', 'every']
```

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

In [42]:
```
fp = open("Rency.txt","r")
wl = fp.read().split()
print("No. of words:",len(wl))
fp.seek(0,0)
lines = fp.readlines()
print("No. of lines:",len(lines))
w_len = list(map(len,wl))
print("no. of char:",sum(w_len))
fp.close()
```

```
No. of words: 9
No. of lines: 6
no. of char: 33
```

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

In [52]:
```
fp = open("Rency.txt")
li = fp.readline()
fp1 = open("Rency.txt","w")
fp1.writelines(li)
fp.close()
fp1.close()
```

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

In [60]:
```
fp = open("Rency.txt")
print("the size of the text file:",fp.seek(0,2))
fp.close()
```

```
the size of the text file: 55
```

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

In [2]:
```
fp = open("Rency.txt")
word = "every"
count = 0
with open("Rency.txt", 'r') as f:
        for line in f:
                words = line.split()
                for i in words:
                        if(i==word):
                                count=count+1
print("Occurrences of the word", word, ":", count)
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[2], line 1
----> 1 fp = open("Rency.txt")
      2 word = "every"
      3 count = 0

File ~\anaconda3\Lib\site-packages\IPython\core\interactiveshell.py:324, in _modified_open(file, *args, **kwargs
)
    317 if file in {0, 1, 2}:
    318     raise ValueError(
    319         f"IPython won't let you open fd={file} by default "
    320         "as it is likely to crash IPython. If you know what you are doing, "
    321         "you can use builtins' open."
    322     )
--> 324 return io_open(file, *args, **kwargs)

FileNotFoundError: [Errno 2] No such file or directory: 'Rency.txt'
```

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 [95]:
```
 marks = input("enter 5 sub mrks space separated:").split()
marks = [i+'\n' for i in marks]
fp = open("Rency.txt", 'w+')
fp.writelines(marks)
fp.seek(0,0)
print(fp.read())
fp.close()
```

```
20
10
20
30
40
```

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

(Note: each number should be in new line)

In [ ]:
```
def is_prime(n):
    """Check if a number is prime."""
    if n < 2:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

def generate_primes(count):
    """Generate the first 'count' prime numbers."""
    primes = []
    num = 2
    while len(primes) < count:
        if is_prime(num):
            primes.append(num)
        num += 1
    return primes

filename = "primenumbers.txt"

prime_numbers = generate_primes(100)

with open(filename, "w") as file:
    for prime in prime_numbers:
        file.write(str(prime) + "\n")

print(f"First 100 prime numbers written to {filename}")
```

11) WAP to merge two files and write it in a new file.

In [ ]:
```
file1 = r"C:\Users\YourUsername\Documents\file1.txt"
file2 = r"C:\Users\YourUsername\Documents\file2.txt"
output_file = r"C:\Users\YourUsername\Documents\merged_file.txt"

try:

    with open(file1, 'r', encoding='utf-8') as f1, open(file2, 'r', encoding='utf-8') as f2, open(output_file,
        out.write(f1.read())
        out.write("\n")
        out.write(f2.read())

    print(f"Files '{file1}' and '{file2}' merged successfully into '{output_file}'.")
except FileNotFoundError:
    print("One or both files not found. Please check the file paths.")
```

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

In [ ]:
```
def replace_word(input_file, output_file, word1, word2):
    """Replaces word1 with word2 in input_file and writes to output_file."""
    try:
        with open(input_file, "r") as infile:
            content = infile.read()  # Read the entire file content

        updated_content = content.replace(word1, word2)  # Replace word1 with word2

        with open(output_file, "w") as outfile:
            outfile.write(updated_content)  # Write the modified content to a new file

        print(f"Replaced '{word1}' with '{word2}' and saved in {output_file}")

    except FileNotFoundError:
        print(f"Error: {input_file} not found.")

# Example usage
input_file = "palak.txt"
output_file = "palak.txt"
word1 = "hello"
word2 = "hy"

replace_word(input_file, output_file, word1, word2)
```

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

In [ ]:
```

```