In [156…

```
#File handling
```

"r" - Read - Default value. Opens a file for reading, error if the file does not exist

"a" - Append - Opens a file for appending, creates the file if it does not exist

"w" - Write - Opens a file for writing, creates the file if it does not exist

"x" - Create - Creates the specified file, returns an error if the file exists

In [20]:

```
#open the file
f=open("num.txt")
print("successfully")
```

```
successfully
```

In [21]:

```
#open the file in curent directorie
f=open("num.txt","r")
print(f.read())
```

```
123
456
789
111
222
333
444
555
666
777
888
999
```

In [23]:

```
#open the file at specified location
f=open("C:/Users/Hp/Desktop/file2.txt","r")
print(f.read())
```

```
Hi i am here at desktop location
File handling is an important part of any web application.
Python has several functions for creating, reading, updating, and deleting files.
"r" - Read - Default value. Opens a file for reading, error if the file does not exist
"a" - Append - Opens a file for appending, creates the file if it does not exist
"w" - Write - Opens a file for writing, creates the file if it does not exist
"x" - Create - Creates the specified file, returns an error if the file exists
```

In [25]:

```
#read specific number of characters
f=open("C:/Users/Hp/Desktop/file2.txt","r")
print(f.read(12))
```

```
Hi i am here
```

In [26]:

```
#read complete line
f=open("C:/Users/Hp/Desktop/file2.txt","r")
print(f.readline())
print(f.readline())
```

```
Hi i am here at desktop location

File handling is an important part of any web application.
```

In [28]:

```
#reading a single line
f=open("num.txt","r")
print(f.readline())
```

```
123
```

In [29]:

```
#reading all lines and output is in form of list
f=open("num.txt","r")
print(f.readlines())
```

```
['123\n', '456\n', '789\n', '111\n', '222\n', '333\n', '444\n', '555\n', '666\n', '777\n', '888\n', '999\n']
```

In [139…

```
#observation is once read again file need to open two times reading of same file is not possible
f=open("num.txt","r")
print("f.readlines output is: ",f.readlines())
line=f.read()
print("the f.read output is : ", line) #it is empty
```

```
f.readlines output is:  ['123\n', '456\n', '789\n', '111\n', '222\n', '333\n', '444\n', '555\n', '666\n', '777\n', '888\n', '999 1000 1111\n']
the f.read output is :
```

In [43]:

```
f=open("num.txt","r")
line=f.read()
print("f.readlines output is: ",f.readlines())
print("the f.read output is : ",line) #it is empty
```

```
f.readlines output is:  []
the f.read output is :  123
456
789
111
222
333
444
555
666
777
888
999
```

In [45]:

```
f=open("num.txt","r")
print(f.readlines())
f.close()
f=open("num.txt","r")
line=f.read()
print(line)
f.close()
```

```
['123\n', '456\n', '789\n', '111\n', '222\n', '333\n', '444\n', '555\n', '666\n', '777\n', '888\n', '999\n']
123
456
789
111
222
333
444
555
666
777
888
999
```

In [47]:

```
#reading a lines with for loop
f=open("C:/Users/Hp/Desktop/file2.txt","r")
for x in f:
    print(x)
```

```
Hi i am here at desktop location

File handling is an important part of any web application.

Python has several functions for creating, reading, updating, and deleting files.

"r" - Read - Default value. Opens a file for reading, error if the file does not exist

"a" - Append - Opens a file for appending, creates the file if it does not exist

"w" - Write - Opens a file for writing, creates the file if it does not exist

"x" - Create - Creates the specified file, returns an error if the file exists
```

In [48]:

```
f=open("C:/Users/Hp/Desktop/file2.txt","r")
for x in f:
    print(x,end="")
f.close() #You should always close your files, in some cases, due to buffering, changes made to a file may not show until you close the file.
```

```
Hi i am here at desktop location
File handling is an important part of any web application.
Python has several functions for creating, reading, updating, and deleting files.
"r" - Read - Default value. Opens a file for reading, error if the file does not exist
"a" - Append - Opens a file for appending, creates the file if it does not exist
"w" - Write - Opens a file for writing, creates the file if it does not exist
"x" - Create - Creates the specified file, returns an error if the file exists
```

In [49]:

```
#automatically closes the file, so we don't have to use the close() function.
with open("C:/Users/Hp/Desktop/file2.txt", "r") as file1:
    read_content = file1.read()
    print(read_content)
```

```
Hi i am here at desktop location
File handling is an important part of any web application.
Python has several functions for creating, reading, updating, and deleting files.
"r" - Read - Default value. Opens a file for reading, error if the file does not exist
"a" - Append - Opens a file for appending, creates the file if it does not exist
"w" - Write - Opens a file for writing, creates the file if it does not exist
"x" - Create - Creates the specified file, returns an error if the file exists
```

In [141…

```
with open("C:/Users/Hp/Desktop/file2.txt", "r") as file1:
    read_content=file1.readlines()
    for line in read_content:
        word=line.split()
        print(word)
```

```
['Hi', 'i', 'am', 'here', 'at', 'desktop', 'location']
['File', 'handling', 'is', 'an', 'important', 'part', 'of', 'any', 'web', 'application.']
['Python', 'has', 'several', 'functions', 'for', 'creating,', 'reading,', 'updating,', 'and', 'deleting', 'files.']
['"r"', '-', 'Read', '-', 'Default', 'value.', 'Opens', 'a', 'file', 'for', 'reading,', 'error', 'if', 'the', 'file', 'does', 'not', 'exist']
['"a"', '-', 'Append', '-', 'Opens', 'a', 'file', 'for', 'appending,', 'creates', 'the', 'file', 'if', 'it', 'does', 'not', 'exist']
['"w"', '-', 'Write', '-', 'Opens', 'a', 'file', 'for', 'writing,', 'creates', 'the', 'file', 'if', 'it', 'does', 'not', 'exist']
['"x"', '-', 'Create', '-', 'Creates', 'the', 'specified', 'file,', 'returns', 'an', 'error', 'if', 'the', 'file', 'exists']
```

In [157…

```
with open("C:/Users/Hp/Desktop/file2.txt", 'r') as fp:
    # read line 2
    x = fp.readlines()[2]
    print(x)
```

```
Python has several functions for creating, reading, updating, and deleting files.
```

In [136…

```
#over_write the content in existing file
f=open("C:/Users/Hp/Desktop/file2.txt","w")
f.write("hello i am overwriten " )
f.close()
f=open("C:/Users/Hp/Desktop/file2.txt","r")
print(f.read())
```

```
hello i am overwriten
```

In [54]:

```
f=open("C:/Users/Hp/Desktop/file10.txt","x")
print("successfully created the file")
```

```
successfully created the file
```

In [55]:

```
#if again i try to create a file it will through error, see in below
f=open("C:/Users/Hp/Desktop/file10.txt","x") #FileExistsError
print("successfully created the file")
```

```
---------------------------------------------------------------------------
FileExistsError                           Traceback (most recent call last)
Cell In[55], line 2
      1 #if again i try to create a file it will through error, see in below
----> 2 f=open("C:/Users/Hp/Desktop/file10.txt","x")
      3 print("successfully created the file")

File ~\anaconda3\Lib\site-packages\IPython\core\interactiveshell.py:310, in _modified_open(file, *args, **kwargs)
    303 if file in {0, 1, 2}:
    304     raise ValueError(
    305         f"IPython won't let you open fd={file} by default "
    306         "as it is likely to crash IPython. If you know what you are doing, "
    307         "you can use builtins' open."
    308     )
--> 310 return io_open(file, *args, **kwargs)

FileExistsError: [Errno 17] File exists: 'C:/Users/Hp/Desktop/file10.txt'
```

In [63]:

```
f=open("C:/Users/Hp/Desktop/file10.txt","w")
f.write("hey i am here in file10.txt file" )
f.close()
f=open("C:/Users/Hp/Desktop/file10.txt","r")
print(f.read())
f.close()
```

```
hey i am here in file10.txt file
```

In [160…

```
#Another way to write the data into the file, here by passing the list we are providing the data to file and \n will take next line
file = open("myfile.txt","w")
L = ["This is Python \n","Programm \n","class day5 \n"]

file.write("Hello There \n")
file.writelines(L)
#file.writelines("hii") #it will take both str and list
#file.write(L) #write() argument must be str, not list
file.close()
file = open("myfile.txt","r")
print(file.read())
file.close()
file = open("myfile.txt","a")
file.write("today class about file handling")
file.close()
print("after appending : ")
file = open("myfile.txt","r")
print(file.read())
file.close()
```

```
Hello There
This is Python
Programm
class day5

after appending :
Hello There
This is Python
Programm
class day5
today class about file handling
```

In [143…

```
with open("C:/Users/Hp/Desktop/file2.txt", "r") as file1:
    read_content=file1.readlines()
read_content1=read_content[::-1]
print(read_content1)
```

```
['"x" - Create - Creates the specified file, returns an error if the file exists', '"w" - Write - Opens a file for writing, creates the file if it does not exist\n', '"a" - Append - Opens a file for appending, creates the file if it do
es not exist\n', '"r" - Read - Default value. Opens a file for reading, error if the file does not exist\n', 'Python has several functions for creating, reading, updating, and deleting files.\n', 'File handling is an important part of
any web application.\n', 'Hi i am here at desktop location\n']
```

Delete

To delete a file, you must import the OS module, and run its os.remove() function:

In [64]:

```
import os
os.remove("C:/Users/Hp/Desktop/file10.txt")
print("Sucessfully deleted")
# After perform delete operation, if you try to access means it will trough a error like FileNotFoundError
#f=open("C:/Users/Hp/Desktop/file10.txt","r") #FileNotFoundError
#print(f.read())
#f.close()
```

```
Sucessfully deleted
```

In [66]:

```
#creating a file
f=open("C:/Users/Hp/Desktop/file4.txt","w")
f.write("i am in 4rth file")
f.close()
f=open("C:/Users/Hp/Desktop/file4.txt","r")
print(f.read())
f.close()
```

```
i am in 4rth file
```

In [68]:

```
#if path exists then remove
import os
if os.path.exists("C:/Users/Hp/Desktop/file4.txt"):
    os.remove("C:/Users/Hp/Desktop/file4.txt")
    print("successfully deleted")

else:
  print("The file does not exist")
```

```
successfully deleted
```

In [70]:

```
#if path exists then remove
import os
if os.path.exists("C:/Users/Hp/Desktop/file4.txt"):
    os.remove("C:/Users/Hp/Desktop/file4.txt")
    print("successfully deleted")

else:
  print("The file does not exist") #becoz it is alredy deleted above, that
```

```
The file does not exist
```

File operations:

In [163…

```
#count the number of lines in the file
#method-1
with open("Number_file.txt", 'r') as fp:
    lines=fp.readlines() #it will return a list
    print(lines)
print("Total lines : " ,len(lines))
```

```
['1\n', '2\n', '3\n', '4\n', '5\n', '6\n', '7\n', '8\n', '9\n', '10\n', '\n', '11\n', '12\n', '13\n', '\n', '14\n', '15\n', '16\n', '17\n', '18\n', '19\n', '20 21 22\n']
Total lines :  22
```

In [110…

```
#method-2 : it will exclude the empty lines
with open("Number_file.txt", 'r') as fp:
    count = 0
    for line in fp:
        if line != "\n":
            count += 1
print("Total Lines :", count)
```

```
Total Lines : 20
```

In [111…

```
#method -3 :
#The strip() method removes any leading, and trailing whitespaces.
count = 0
with open("Number_file.txt","r") as fp:
    for line in fp:
        #print(line)
        x=line.strip()
        #print(x)
        if(x):
            count += 1

print('number of non-blank lines', count)
```

```
number of non-blank lines 20
```

In [95]:

```
#size of the file
import os
file_path="Number_file.txt"
a=os.path.getsize(file_path)
#print(a)
print(f'The {file_path} size is',a, 'bytes')
```

```
The Number_file.txt size is 79 bytes
```

In [98]:

```
#write data into the file
import random
fp1=open("Number_file.txt","w")
for i in range(1,20):
    #fp1.write(str(random.randrange(1,100))) #write random numbers into the file
    fp1.write((str(i)))
    #fp1.write(input("enter value")) #from user we can take
    fp1.write("\n") #to take next line
fp1.close()
print("data written into the file successfully")
fp1=open("Number_file.txt","r")
#print(fp1.readlines())
print(fp1.read())
fp1.close()
```

```
data written into the file successfully
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

In [102…

```
#checking whether particular string is exist or not
file = open("myfile.txt","r")
data=file.read()
word="Python"
print(type(data))
if(word in data): #in string we can easily search whether it is exits or not with the help of in operator
    print("yes")
else:
    print("no")
```

```
<class 'str'>
yes
```

In [164…

```
#Search file for a string and Print its line and line number
word = 'class' #you can take search word from the user using input()
with open("myfile.txt", 'r') as fp:
    # read all lines in a list
    lines = fp.readlines()
    print(lines)
    for line in lines:
        # check if string present on a current line
        if line.find(word) != -1:
            print(word, 'string exists in file')
            print('Line Number:', lines.index(line))
            print('Line:', line)
```

```
['Hello There \n', 'This is Python \n', 'Programm \n', 'class day5 \n', 'today class about file handling']
class string exists in file
Line Number: 3
Line: class day5

class string exists in file
Line Number: 4
Line: today class about file handling
```

In [103…

```
#Use for loop with enumerate() function to get a line and its number.
#The enumerate() function adds a counter to an iterable and returns it in enumerate object.
#Pass the file pointer returned by the open() function to the enumerate().
#The enumerate(file_pointer) doesn't load the entire file in memory, so this is an efficient solution.

with open("myfile.txt", 'r') as fp:
    for l_no, line in enumerate(fp):
        if 'class' in line:
            print('string found in a file')
            print('Line Number:', l_no)
            print('Line:', line)
```

```
string found in a file
Line Number: 3
Line: class day5

string found in a file
Line Number: 4
Line: today class about file handling
```

In [144…

```
#Search file for a list of strings
words = ['Python', 'class']
with open("myfile.txt", 'r') as fp:
    content = fp.read()
# Iterate list to find each word
for word in words:
    if word in content:
        print(f"string {word} exist in this file")
```

```
string Python exist in this file
string class exist in this file
```

In [147…

```
import os

dir_path = "C:/Users/Hp/Desktop/File_handling_files"
# iterate each file in a directory
for file in os.listdir(dir_path):
    cur_path = os.path.join(dir_path, file)
    # check if it is a file
    if os.path.isfile(cur_path):
        with open(cur_path, 'r') as file:
            # read all content of a file and search string
            if 'day' in file.read():
                print('string found')
```

```
string found
string found
string found
```

In [125…

```
import os

dir_path = "C:/Users/Hp/Desktop/File_handling_files"
# iterate each file in a directory
word="day"
for file in os.listdir(dir_path):
    cur_path = os.path.join(dir_path, file)
    # check if it is a file
    if os.path.isfile(cur_path):
        with open(cur_path, 'r') as file:
            # read all content of a file and search string
            file_list=file.readlines()
            print(file_list)
            for line in file_list:
                 if word in line:
                     print(line, "string exists in", cur_path,"file")
```

```
['Python\n', 'class\n', 'day-1']
day-1 string exists in C:/Users/Hp/Desktop/File_handling_files\file1.txt file
['Python\n', 'class\n', 'day-2']
day-2 string exists in C:/Users/Hp/Desktop/File_handling_files\file2.txt file
['Python\n', 'class\n', 'day-3']
day-3 string exists in C:/Users/Hp/Desktop/File_handling_files\file3.txt file
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: