|  |  |
| --- | --- |
| **Python File Handling:**  **File Operations**  Files are named locations on our storage device for recording data. Python provides numerous built in functions to work with these files.  There are three steps we need to follow to work with files:  Open a file  Perform Operation (Read or Write)  Close the file | |
| 1. **Opening a File**   In Python, we use the builtin open() function to open files.  Suppose we have a file called message.txt with the following content:  f = open('message.txt')  # read mode  #f = open('message.txt', 'r')  # write mode  #f = open('message.txt', 'w')  f = open('message.txt', 'r')  content = f.read()  print(content)  f.close() | Hello there  this is message txt file |
| read only the first 6 characters:  f = open('message.txt', 'r')  content = f.read(6)  print(content)  f.close() | Hello |
| f = open('message.txt', 'r')  content = f.read(6)  print(content)  more\_content = f.read(12)  print(more\_content)  f.close() | Hello  there  this i |
| **Writing to files in Python**  To write content to a file, we must first open it in write mode. Then, we can start writing content to it using the write() method.  There are two things you need to remember while writing to a file:  If you try to open a file that doesn't exist, a new file is automatically created.  If a file already exists, its contents are removed, and our new content is added to it.  with open('python.txt', 'w') as f:  f.write("Python is awesome")  f.write("I love Python") | A python.txt file is created in the same directory with the following content:  Python is awesomeI love Python |
| To add a new line, we can use \n:  with open('python.txt', 'w') as f:  f.write("Python is awesome\n")  f.write("I love Python") | Python is awesome  I love python |
| #Demonstrated Python Program  # to read file character by character      file = open('file.txt', 'r')    while 1:    # read by character  char = file.read(1)  if not char:  break    print(char)    file.close() | m  y  n  a  m  e  i  s  b  r  i  j  a  l |
| **Appending to Files in Python**  We use this mode if we want to add additional data to the end of the file without erasing our previous data.  Let's add an additional line to the previous python.txt file.  with open('python.txt', 'a') as f:  f.write("\nPython is my first programming language.") | Python is awesome  I love Python  Python is my first programming language. |
| **Python readlines() and writelines()**  The readlines() method returns a list containing each line of the file.  Let's open the same python.txt file we have been working on in read mode and use readlines():  with open('python.txt', 'r') as f:  lines = f.readlines()  print(lines) | ['Python is awesome\n', 'I love Python\n', 'Python is my first programming language.'] |
| there is also a writelines() method to write multiple items into a file. It writes the items of a list to the file.  with open('javascript.txt', 'w') as f:  lines = ['JS is also awesome', '\nJS is my second programming language.']  f.writelines(lines) | JS is also awesome  JS is my second programming language. |
| 1. WAP to read data from keyboard & write it to the file. After writing is completed, the file is closed. The program again opens the same file and reads it. | text\_file = open("input.txt",'w')  text\_file.write("My name is Mohit")  text\_file.write("\nI am a C|EH")  text\_file.write("\nI am also E|CSA")  text\_file.close()  file\_input = open("input.txt ",'r')  print(file\_input.readline())  print(file\_input.readline())  file\_input.close() |
| 1. Count number of lines in a text file in Python | # Opening a file  file = open("gfg.txt","r")  Counter = 0    # Reading from file  Content = file.read()  CoList = Content.split("\n")    for i in CoList:  if i:  Counter += 1    print("This is the number of lines in the file")  print(Counter)  **output**  This is the number of lines in the file  4 |
|  |  |