1. **Python Program to obtain the line number in which given word is present**

# READ FILE

df = open("geeks.txt")

# read file

read = df.read()

# return cursor to

# the beginning

# of the file.

df.seek(0)

read

**Output:** 'Hello, I am Romy\nI am a content writer at GfG\nNice to meet you\nHello, hii all fine'

1. **Count number of lines in a text file in Python**

with open(r"myfile.txt", 'r') as fp:

  lines = len(fp.readlines())

  print('Total Number of lines:', lines)

1. **Python Program to remove lines starting with any prefix**

# defining object file1 to

# open GeeksforGeeks file in

# read mode

file1 = open('GeeksforGeeks.txt','r')

# defining object file2 to

# open GeeksforGeeksUpdated file

# in write mode

file2 = open('GeeksforGeeksUpdated.txt','w')

# reading each line from original

# text file

for line in file1.readlines():

  # reading all lines that do not

  # begin with "TextGenerator"

  if not (line.startswith('TextGenerator')):

    # printing those lines

    print(line)

    # storing only those lines that

    # do not begin with "TextGenerator"

    file2.write(line)

# close and save the files

file2.close()

file1.close()

1. **Python Program to Eliminate repeated lines from a file**

# creating the output file

outputFile = open('C:/Users/user/Desktop/Lorem\_output.txt', "w")

# reading the input file

inputFile = open('C:/Users/user/Desktop/Lorem\_input.txt', "r")

# holds lines already seen

lines\_seen\_so\_far = set()

# iterating each line in the file

for line in inputFile:

  # checking if line is unique

  if line not in lines\_seen\_so\_far:

    # write unique lines in output file

    outputFile.write(line)

    # adds unique lines to lines\_seen\_so\_far

    lines\_seen\_so\_far.add(line)

# closing the file

inputFile.close()

outputFile.close()

1. **Python Program to read List of Dictionaries from File**

def parse(d):

  dictionary = dict()

  # Removes curly braces and splits the pairs into a list

  pairs = d.strip('{}').split(', ')

  for i in pairs:

    pair = i.split(': ')

    # Other symbols from the key-value pair should be stripped.

    dictionary[pair[0].strip('\'\'\"\"')] = pair[1].strip('\'\'\"\"')

  return dictionary

try:

  geeky\_file = open('geeky\_file.txt', 'rt')

  lines = geeky\_file.read().split('\n')

  for l in lines:

    if l != '':

      dictionary = parse(l)

      print(dictionary)

  geeky\_file.close()

except:

  print("Something unexpected occurred!")

1. **Python – Append content of one text file to another**

# entering the file names

firstfile = input("Enter the name of first file ")

secondfile = input("Enter the name of second file ")

# opening both files in read only mode to read initial contents

f1 = open(firstfile, 'r')

f2 = open(secondfile, 'r')

# printing the contents of the file before appending

print('content of first file before appending -', f1.read())

print('content of second file before appending -', f2.read())

# closing the files

f1.close()

f2.close()

# opening first file in append mode and second file in read mode

f1 = open(firstfile, 'a+')

f2 = open(secondfile, 'r')

# appending the contents of the second file to the first file

f1.write(f2.read())

# relocating the cursor of the files at the beginning

f1.seek(0)

f2.seek(0)

# printing the contents of the files after appendng

print('content of first file after appending -', f1.read())

print('content of second file after appending -', f2.read())

# closing the files

f1.close()

f2.close()

1. **Python program to copy odd lines of one file to other**

# open file in read mode

fn = open('bcd.txt', 'r')

# open other file in write mode

fn1 = open('nfile.txt', 'w')

# read the content of the file line by line

cont = fn.readlines()

type(cont)

for i in range(0, len(cont)):

  if(i % 2 ! = 0):

    fn1.write(cont[i])

  else:

    pass

# close the file

fn1.close()

# open file in read mode

fn1 = open('nfile.txt', 'r')

# read the content of the file

cont1 = fn1.read()

# print the content of the file

print(cont1)

# close all files

fn.close()

fn1.close()

1. **Python Program to merge two files into a third file**

# Python program to

# demonstrate merging

# of two files

data = data2 = ""

# Reading data from file1

with open('file1.txt') as fp:

  data = fp.read()

# Reading data from file2

with open('file2.txt') as fp:

  data2 = fp.read()

# Merging 2 files

# To add the data of file2

# from next line

data += "\n"

data += data2

with open ('file3.txt', 'w') as fp:

  fp.write(data)

1. **Python program to Reverse a single line of a text file**

# Open file in read mode

f = open('GFG.txt', 'r')

# Read the content of the

# file and store it in a list

lines = f.readlines()

# Close file

f.close()

# User's choice

choice = 1

# Split the line into words

line = lines[choice].split()

# line is reversed

Reversed = " ".join(line[::-1])

# Updating the content of the

# file

lines.pop(choice)

lines.insert(choice, Reversed)

# Open file in write mode

u = open('GFG.txt', 'w')

# Write the new content in file

# and note, it is overwritten

u.writelines(lines)

u.close()

1. **Python program to reverse the content of a file and store it in another file**

# Open the file in write mode

f1 = open("output1.txt", "w")

# Open the input file and get

# the content into a variable data

with open("file.txt", "r") as myfile:

  data = myfile.read()

# For Full Reversing we will store the

# value of data into new variable data\_1

# in a reverse order using [start: end: step],

# where step when passed -1 will reverse

# the string

data\_1 = data[::-1]

# Now we will write the fully reverse

# data in the output1 file using

# following command

f1.write(data\_1)

f1.close()

1. **Python Program to Reverse the Content of a File using Stack**

# Python3 code to reverse the lines

# of a file using Stack.

# Creating Stack class (LIFO rule)

class Stack:

  def \_\_init\_\_(self):

    # Creating an empty stack

    self.\_arr = []

  # Creating push() method.

  def push(self, val):

    self.\_arr.append(val)

  def is\_empty(self):

    # Returns True if empty

    return len(self.\_arr) == 0

  # Creating Pop method.

  def pop(self):

    if self.is\_empty():

      print("Stack is empty")

      return

    return self.\_arr.pop()

# Creating a function which will reverse

# the lines of a file and Overwrites the

# given file with its contents line-by-line

# reversed

def reverse\_file(filename):

  S = Stack()

  original = open(filename)

  for line in original:

    S.push(line.rstrip("\n"))

  original.close()

  output = open(filename, 'w')

  while not S.is\_empty():

    output.write(S.pop()+"\n")

  output.close()

# Driver Code

filename = "GFG.txt"

# Calling the reverse\_file function

reverse\_file(filename)

# Now reading the content of the file

with open(filename) as file:

    for f in file.readlines():

      print(f, end ="")