1. **Python Program to Check if email address valid or not**

import re

# Make a regular expression

# for validating an Email

regex = r'\b[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'

# Define a function for

# for validating an Email

def check(email):

  # pass the regular expression

  # and the string into the fullmatch() method

  if(re.fullmatch(regex, email)):

    print("Valid Email")

  else:

    print("Invalid Email")

# Driver Code

if \_\_name\_\_ == '\_\_main\_\_':

  # Enter the email

  email = "ankitrai326@gmail.com"

  # calling run function

  check(email)

  email = "my.ownsite@our-earth.org"

  check(email)

  email = "ankitrai326.com"

  check(email)

**Output:** Valid Email

Valid Email

Invalid Email

1. **Python program to find files having a particular extension using RegEx**

# import library

import re

# list of different types of file

filenames = ["gfg.html", "geeks.xml",

      "computer.txt", "geeksforgeeks.jpg"]

for file in filenames:

  # search given pattern in the line

  match = re.search("\.xml$", file)

  # if match is found

  if match:

    print("The file ending with .xml is:",file)

**Output:** The file ending with .xml is: geeks.xml

1. **Python program to extract IP address from file**

# importing the module

import re

# opening and reading the file

with open('C:/Users/user/Desktop/New Text Document.txt') as fh:

fstring = fh.readlines()

# declaring the regex pattern for IP addresses

pattern = re.compile(r'(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})')

# initializing the list object

lst=[]

# extracting the IP addresses

for line in fstring:

lst.append(pattern.search(line)[0])

# displaying the extracted IP addresses

print(lst)

**Output:** The file ending with .xml is: geeks.xml

1. **Python program to check the validity of a Password**

# Python program to check validation of password

# Module of regular expression is used with search()

import re

password = "R@m@\_f0rtu9e$"

flag = 0

while True:

  if (len(password)<=8):

    flag = -1

    break

  elif not re.search("[a-z]", password):

    flag = -1

    break

  elif not re.search("[A-Z]", password):

    flag = -1

    break

  elif not re.search("[0-9]", password):

    flag = -1

    break

  elif not re.search("[\_@$]" , password):

    flag = -1

    break

  elif re.search("\s" , password):

    flag = -1

    break

  else:

    flag = 0

    print("Valid Password")

    break

if flag == -1:

  print("Not a Valid Password ")

**Output:** Valid Password

1. **Categorize Password as Strong or Weak using Regex in Python**

# Categorizing password as Strong or

# Weak in Python using Regex

import re

# Function to categorize password

def password(v):

  # the password should not be a

  # newline or space

  if v == "\n" or v == " ":

    return "Password cannot be a newline or space!"

  # the password length should be in

  # between 9 and 20

  if 9 <= len(v) <= 20:

    # checks for occurrence of a character

    # three or more times in a row

    if re.search(r'(.)\1\1', v):

      return "Weak Password: Same character repeats three or more times in a row"

    # checks for occurrence of same string

    # pattern( minimum of two character length)

    # repeating

    if re.search(r'(..)(.\*?)\1', v):

      return "Weak password: Same string pattern repetition"

    else:

      return "Strong Password!"

  else:

    return "Password length must be 9-20 characters!"

# Main method

def main():

  # Driver code

  print(password("Qggf!@ghf3"))

  print(password("Gggksforgeeks"))

  print(password("aaabnil1gu"))

  print(password("Aasd!feasn"))

  print(password("772\*hd897"))

  print(password(" "))

# Driver Code

if \_\_name\_\_ == '\_\_main\_\_':

  main()

**Output:** Strong Password!

Weak password: Same string pattern repetition

Weak Password: Same character repeats three or more times in a row

Weak password: Same string pattern repetition

Strong Password!

Password cannot be a newline or space!

1. **Python program to read file word by word**

# Python program to read

# file word by word

# opening the text file

with open('GFG.txt','r') as file:

  # reading each line

  for line in file:

    # reading each word

    for word in line.split():

      # displaying the words

      print(word)

**Output:** Geeks

4

geeks

1. **Python program to read character by character from a file**

# Demonstrated Python Program

# to read file character by character

file = open('file.txt', 'r')

while 1:

  # read by character

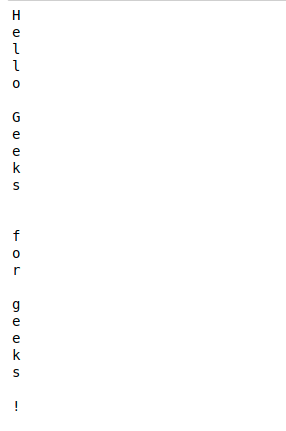
  char = file.read(1)

  if not char:

    break

  print(char)

file.close()



1. **Python – Get number of characters, words, spaces and lines in a file**

# Python implementation to compute

# number of characters, words, spaces

# and lines in a file

# Function to count number

# of characters, words, spaces

# and lines in a file

def counter(fname):

  # variable to store total word count

  num\_words = 0

  # variable to store total line count

  num\_lines = 0

  # variable to store total character count

  num\_charc = 0

  # variable to store total space count

  num\_spaces = 0

  # opening file using with() method

  # so that file gets closed

  # after completion of work

  with open(fname, 'r') as f:

    # loop to iterate file

    # line by line

    for line in f:

      # incrementing value of

      # num\_lines with each

      # iteration of loop to

      # store total line count

      num\_lines += 1

      # declaring a variable word

      # and assigning its value as Y

      # because every file is

      # supposed to start with

      # a word or a character

      word = 'Y'

      # loop to iterate every

      # line letter by letter

      for letter in line:

        # condition to check

        # that the encountered character

        # is not white space and a word

        if (letter != ' ' and word == 'Y'):

          # incrementing the word

          # count by 1

          num\_words += 1

          # assigning value N to

          # variable word because until

          # space will not encounter

          # a word can not be completed

          word = 'N'

        # condition to check

        # that the encountered character

        # is a white space

        elif (letter == ' '):

          # incrementing the space

          # count by 1

          num\_spaces += 1

          # assigning value Y to

          # variable word because after

          # white space a word

          # is supposed to occur

          word = 'Y'

        # loop to iterate every

        # letter character by

        # character

        for i in letter:

          # condition to check

          # that the encountered character

          # is not white space and not

          # a newline character

          if(i !=" " and i !="\n"):

            # incrementing character

            # count by 1

            num\_charc += 1

  # printing total word count

  print("Number of words in text file: ",

    num\_words)

  # printing total line count

  print("Number of lines in text file: ",

    num\_lines)

  # printing total character count

  print('Number of characters in text file: ',

    num\_charc)

  # printing total space count

  print('Number of spaces in text file: ',

    num\_spaces)

# Driver Code:

if \_\_name\_\_ == '\_\_main\_\_':

  fname = 'File1.txt'

  try:

    counter(fname)

  except:

    print('File not found')

**Output:**

Number of words in text file: 25

Number of lines in text file: 4

Number of characters in text file: 91

Number of spaces in text file: 21

1. **Python program to Count the Number of occurrences of a key-value pair in a text file**

# Python program to count the

# occurrences of key-value pair

# in the text file

# opening text file

f = open("file.txt", "r")

d = dict()

for res in f:

  # removing new line and extra

  # space characters

  res = res.strip()

  # changing ase to prevent matching

  # errors

  res = res.lower()

  # separating key-value pairs

  lines = res.split()

  for line in lines:

    if line in d:

      # If the key-value pair

      # is present in d then

      # increment its value by one

      d[line] = d[line]+1

    else:

      # Insert the key-value pair

      # in the dictionary and sets

      # its value to one

      d[line] = 1

f.close()

# Printing Result

for key in list(d.keys()):

  print("The count of {} is {}".format(key,d[key]))

**Output:** Lightbox

1. **Python | Finding ‘n’ Character Words in a Text File**

count = 1

chrw = ""

# text file

file = open('textfile.txt', 'r')

while 1:

  sp = file.read(1)

  if count<= 3:

    chrw = chrw + sp

  if count>3:

    if sp ==" ":

      count = 0

      if len(chrw)>0:

        print(chrw)

        chrw =""

    elif sp !=" ":

      chrw =""

  count = count + 1

  if not sp:

    break

file.close()

**Output:**

**how**

**are**

**you**