1. **Python – Check whether a string starts and ends with the same character or not**

# Python program to check if a string starts

# and ends with the same character

# import re module as it provides

# support for regular expressions

import re

# the regular expression

regex = r'^[a-z]$|^([a-z]).\*\1$'

# function for checking the string

def check(string):

  # pass the regular expression

  # and the string in the search() method

  if(re.search(regex, string)):

    print("Valid")

  else:

    print("Invalid")

if \_\_name\_\_ == '\_\_main\_\_' :

  sample1 = "abba"

  sample2 = "a"

  sample3 = "abcd"

  check(sample1)

  check(sample2)

  check(sample3)

**Output:** Valid

Valid

Invalid

1. **Python regex to find sequences of one upper case letter followed by lower case letters**

# Python3 code to find sequences of one upper

# case letter followed by lower case letters

import re

# Function to match the string

def match(text):

    # regex

    pattern = '[A-Z]+[a-z]+$'

    # searching pattern

    if re.search(pattern, text):

        return('Yes')

    else:

        return('No')

# Driver Function

print(match("Geeks"))

print(match("geeksforGeeks"))

print(match("geeks"))

**Output:** Yes

Yes

No

1. **Python Program to Remove duplicate words from Sentence**

from collections import Counter

def remov\_duplicates(input):

  # split input string separated by space

  input = input.split(" ")

  # now create dictionary using counter method

  # which will have strings as key and their

  # frequencies as value

  UniqW = Counter(input)

  # joins two adjacent elements in iterable way

  s = " ".join(UniqW.keys())

  print (s)

# Driver program

if \_\_name\_\_ == "\_\_main\_\_":

  input = 'Python is great and Java is also great'

  remov\_duplicates(input)

**Output:** Python is great and Java also

1. **Python | Remove all characters except letters and numbers**

import re

# initialising string

ini\_string = "abcjw:, .@! eiw"

# printing initial string

print ("initial string : ", ini\_string)

result = re.sub('[\W\_]+', '', ini\_string)

# printing final string

print ("final string", result)

**Output:** initial string : abcjw:, .@! eiw

final string abcjweiw

1. **Python Regex | Program to accept string ending with alphanumeric character**

# Python program to accept string ending

# with only alphanumeric character.

# import re module

# re module provides support

# for regular expressions

import re

# Make a regular expression to accept string

# ending with alphanumeric character

regex = '[a-zA-z0-9]$'

# Define a function for accepting string

# ending with alphanumeric character

def check(string):

  # pass the regular expression

  # and the string in search() method

  if(re.search(regex, string)):

    print("Accept")

  else:

    print("Discard")

# Driver Code

if \_\_name\_\_ == '\_\_main\_\_' :

  # Enter the string

  string = "ankirai@"

  # calling run function

  check(string)

  string = "ankitrai326"

  check(string)

  string = "ankit."

  check(string)

  string = "geeksforgeeks"

  check(string)

**Output:** Discard

Accept

Discard

Accept

1. **Python Regex – Program to accept string starting with vowel**

# Python program to accept string starting with a vowel

# import re module

# re module provides support

# for regular expressions

import re

# Make a regular expression

# to accept string starting with vowel

regex = '^[aeiouAEIOU][A-Za-z0-9\_]\*'

# Define a function for

# accepting string start with vowel

def check(string):

  # pass the regular expression

  # and the string in search() method

  if(re.search(regex, string)):

    print("Valid")

  else:

    print("Invalid")

# Driver Code

if \_\_name\_\_ == '\_\_main\_\_' :

  # Enter the string

  string = "ankit"

  # calling run function

  check(string)

  string = "geeks"

  check(string)

  string = "sandeep"

  check(string)

**Output:** Valid

Invalid

Invalid

1. **Python Program to check if a string starts with a substring using regex**

# import library

import re

# define a function

def find(string, sample):

# check substring present

# in a string or not

  if(sample in string):

    y = "^" + sample

  # check if string starts

  # with the substring

    x = re.search(y, string)

    if x:

      print("string starts with the given substring")

    else:

      print("string doesn't start with the given substring")

    else:

      print("entered string isn't a substring")

# Driver code

    string = "geeks for geeks makes learning fun"

    sample = "geeks"

# function call

    find(string, sample)

    sample = "makes"

# function call

    find(string, sample)

**Output:** string starts with the given substring

string doesn't start with the given substring

1. **Python Program to Check if an URL is valid or not using Regular Expression**

# Python3 program to check

# URL is valid or not

# using regular expression

import re

# Function to validate URL

# using regular expression

def isValidURL(str):

  # Regex to check valid URL

  regex = ("((http|https)://)(www.)?" +

      "[a-zA-Z0-9@:%.\_\\+~#?&//=]" +

      "{2,256}\\.[a-z]" +

      "{2,6}\\b([-a-zA-Z0-9@:%" +

      ".\_\\+~#?&//=]\*)")

  # Compile the ReGex

  p = re.compile(regex)

  # If the string is empty

  # return false

  if (str == None):

    return False

  # Return if the string

  # matched the ReGex

  if(re.search(p, str)):

    return True

  else:

    return False

# Driver code

# Test Case 1:

url = "https://www.geeksforgeeks.org"

if(isValidURL(url) == True):

  print("Yes")

else:

  print("No")

**Output:** Yes

1. **Parsing and Processing URL using Python – Regex**

# import library

import re

# url link

s = 'https://www.geeksforgeeks.org/'

# finding the protocol

obj1 = re.findall('(\w+)://',

        s)

print(obj1)

# finding the hostname which may

# contain dash or dots

obj2 = re.findall('://www.([\w\-\.]+)',

        s)

print(obj2)

**Output:** ['https']

['geeksforgeeks.org']

1. **Python Program to validate an IP address using ReGex**

# Python program to validate an Ip address

# re module provides support

# for regular expressions

import re

# Make a regular expression

# for validating an Ip-address

regex = "^((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\.){3}(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])$"

# Define a function for

# validate an Ip address

def check(Ip):

  # pass the regular expression

  # and the string in search() method

  if(re.search(regex, Ip)):

    print("Valid Ip address")

  else:

    print("Invalid Ip address")

# Driver Code

if \_\_name\_\_ == '\_\_main\_\_' :

  # Enter the Ip address

  Ip = "192.168.0.1"

  # calling run function

  check(Ip)

  Ip = "110.234.52.124"

  check(Ip)

  Ip = "366.1.2.2"

  check(Ip)

**Output:** Valid Ip address

Valid Ip address

Invalid Ip address