

```
In [1]: #1. Write a Python program to demonstrate the zero division error and overflow error
import math
data = 50
try:
    data = data / 0 # data = data / 5
except ZeroDivisionError:
    print("Zero Division Error")
else:
    print("Division successful :", data) #Division successful : 10
try:
    a = math.exp(1000) #math.exp(2)
    print(a) #7.38905609893065

except OverflowError:
    print("Overflow Error")
```

Zero Division Error
Overflow Error

```
In [2]: #2. Write a Python program to find sequences of lowercase letters joined with a underscore
import re
def match(text):
    pattern = '[a-z]+_[a-z]+$'
    if re.search(pattern, text):
        return 'Yes'
    else:
        return 'No'
print(match(input("Enter Text :")))
```

Enter Text :hi
No

```
In [3]: #3. Write a python program to Check if String Contain Only Defined Characters using Regex
import re
def check(str, pattern):

    if re.search(pattern, str):
        print("Valid String")
    else:
        print("Invalid String")

pattern = re.compile('^[179]+$')
check('179', pattern)
check('157', pattern)
```

Valid String
Invalid String

```
In [4]: #1. Write a Python program to match a string that contains only upper and lowercase letters,
#numbers, and underscores. Write a Python program to raised the attribute error, if attribute
#class object has no attribute with the name attribute.
import re
def text_match(text):
    patterns = '[a-zA-Z0-9_]*$'
    if re.search(patterns, text):
        return 'Found a match!'
    else:
        return 'Not matched!'

print(text_match("The quick brown fox jumps over the lazy dog."))
print(text_match("Python_is_1_Programming_language"))
```

Not matched!
Found a match!

```
In [5]: #2. Write a python Program to Remove duplicate words from Sentence
string = "Python is good Python is for beginners beginners"

print(' '.join(dict.fromkeys(string.split())))
```

Python is good for beginners

In [6]: *#3. Write a python to| Remove all characters except letters and numbers*

```
import re
my_string = "python123:,.@! abc"
print ("The string is : ")
print(my_string)
result = re.sub('[\W_]+', '', my_string)
print ("The String after Removal is :")
print(result)
```

The string is :
python123:,.@! abc
The String after Removal is :
python123abc

In [7]: *#1. Write a python program to Count Uppercase, Lowercase, special character and numeric values using Regex*

```
def Count(str):
    upper, lower, number, special = 0, 0, 0, 0
    for i in range(len(str)):
        if str[i].isupper():
            upper += 1
        elif str[i].islower():
            lower += 1
        elif str[i].isdigit():
            number += 1
        else:
            special += 1
    print('Upper case letters:', upper)
    print('Lower case letters:', lower)
    print('Number:', number)
    print('Special characters:', special)
str = "@@helloTybcs1904WELcome###"
Count(str)
```

Upper case letters: 4
Lower case letters: 13
Number: 4
Special characters: 5

In [8]: *#2. Write a python program to find the most occurring number in a string using Regex*

```
import re
from collections import Counter
def most_occ_element(word):
    arr = re.findall(r'[0-9]+', word)
    maxm = 0
    max_elem = 0
    c = Counter(arr)
    for x in list(c.keys()):
        if c[x] >= maxm:
            maxm = c[x]
            max_elem = int(x)
    return max_elem
if __name__ == "__main__":
    word = 'abc58abc52abd12abcdefg12ab58ac58'
    print(most_occ_element(word))
```

58

In [9]: *#3. Write a python Regex to extract maximum numeric value from a string*

```
def extractMaximum(ss):
    num, res = 0, 0
    for i in range(len(ss)):
        if ss[i] >= "0" and ss[i] <= "9":
            num = num * 10 + int(int(ss[i]) - 0)
        else:
            res = max(res, num)
            num = 0
    return max(res, num)
ss = "100klh564abc365bg100005abcd"
print(extractMaximum(ss))
```

100005

In [10]: *#4. Write a python program to put spaces between words starting with capital letters using Regex*

```
import re
```

```
def capital_words_spaces(str1):
    return re.sub(r"(\w)([A-Z])", r"\1 \2", str1)
print(capital_words_spaces("Python"))
print(capital_words_spaces("PythonPractical"))
print(capital_words_spaces("PythonTybcsPracticeSolution"))
```

Python
Python Practical
Python Tybcs Practice Solution

In [11]: *#5. Write a python to Check whether a string starts and ends with the same character or not*

```
import re
regex = r'^[a-z]$|^([a-z]).*\1$'
def check(string):
    if re.search(regex, string):
        print("Valid")
    else:
        print("Invalid")
if __name__ == '__main__':
    sample1 = "abca"
    sample2 = "pythonp"
    sample3 = "abc"

    check(sample1)
    check(sample2)
    check(sample3)
```

Valid
Valid
Invalid

In [12]: *#6. Write a python regex to find sequences of one upper case letter followed by lower case letters*

```
import re
def match(text):
    pattern = '[A-Z][a-z]+'
    if re.search(pattern, text):
        return('Yes')
    else:
        return('No')
print(match("Python"))
print(match("Pythonpractical"))
print(match("python"))
```

Yes
Yes
No

In [13]: *#7. Write a python Regex program to accept string ending with alphanumeric character*

```
import re
regex = '[a-zA-z0-9]$\n'
def check(string):
    if re.search(regex, string):
        print("Accept")
    else:
        print("Discard")
if __name__ == '__main__':
    string = "pratiksha@"
    check(string)
    string = "pratiksha326"
    check(string)
    string = "pratiksha."
    check(string)
    string = "pratikshadalvi"
    check(string)
```

Discard
Accept
Discard
Accept

In [14]: *#8. Write a python Regex program to accept string starting with vowel*
initializing list

```
test_list = ["red", "is", "dark", "color", "and", "in", "signal", "it", "stops", "vechicals"]
print("The original list is : ")
print(test_list)
```

```

res = []
vow = "aeiou"
for sub in test_list:
    flag = False
    for ele in vow:
        if sub.startswith(ele):
            flag = True
            break
    if flag:
        res.append(sub)
print("The extracted words : " )
print(res)

```

The original list is :
['red', 'is', 'dark', 'color', 'and', 'in', 'signal', 'it', 'stops', 'vechicals']
The extracted words :
['is', 'and', 'in', 'it']

In [15]: *#9. Write a python Program to check if a string starts with a substring using regex*

```

import re
def check_string(my_string, sub_string) :
    if (sub_string in my_string):
        concat_string = "^" + sub_string
        result = re.search(concat_string, my_string)
        if result :
            print("The string starts with the given substring")
        else :
            print("The string doesnot start with the given substring")
    else :
        print("It is not a substring")
my_string = "Python coding is fun to learn"
sub_string = "Python"
print("The string is :")
print(my_string)
print("The sub-string is :")
print(sub_string)
check_string(my_string, sub_string)

```

The string is :
Python coding is fun to learn
The sub-string is :
Python
The string starts with the given substring

In [16]: *#10. Write a python Program to Check if an URL is valid or not using Regular Expression*

```

import re
def isValidURL(str):
    regex = ("((http|https)://)(www.)?" +
        "[a-zA-Z0-9@:%._\\+~#?&/=]" +
        "{2,256}\\.[a-z]" +
        "{2,6}\\b([-a-zA-Z0-9@:%" +
        "._\\+~#?&/=]*)")
    p = re.compile(regex)
    if (str == None):
        return False
    if(re.search(p, str)):
        return True
    else:
        return False
url = "@https://www.pranjaldeo.org#"
if(isValidURL(url) == True):
    print("Yes")
else:
    print("No")

```

No

In [17]: *#11. Write a python Program to Parsing and Processing URL using Python – Regex*

```

import re
s = 'https://www.pratikshadalvi.org/'
obj1 = re.findall('(\\w+)://',s)
print(obj1)
obj2 = re.findall('/://www.([\\w\\-\\.]+)',s)
print(obj2)

```

['https']
['pratikshadalvi.org']

```
In [18]: #12. Write a python Program to validate an IP address using ReGex
import re
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])$"
def check(Ip):
    if re.search(regex, Ip):
        print("Valid Ip address")
    else:
        print("Invalid Ip address")
if __name__ == '__main__':
    Ip = "192.168.0.1"
    check(Ip)
    Ip = "110.234.52.124"
    check(Ip)
    Ip = "366.1.2.2"
    check(Ip)
```

Valid Ip address
Valid Ip address
Invalid Ip address

```
In [19]: #13. Write a python Program to Check if email address valid or not
import re
def isValid(email):
    if re.match("^[a-zA-Z0-9_+&*-. ]+(?:\\.[a-zA-Z0-9_+&*-. ]+)*@(?:[a-zA-Z0-9-]+\\.)+[a-zA-Z]{2,7}$", email) != None):
        return True
    return False
if(isValid("hello#gmail.com") == True):
    print("This is a valid email address")
else:
    print("This is not a valid email address")
```

This is not a valid email address

```
In [20]: #14. Write a python program to find files having a particular extension using RegEx
import re
filenames = ["gfg.html", "hello.xml",
             "computer.txt", "welcome.jpg"]
for file in filenames:
    match = re.search("\.xml$", file)
    if match:
        print("The file ending with .xml is:",file)
```

The file ending with .xml is: hello.xml

```
In [21]: #16. Write a python program to check the validity of a Password
import re
password = "pranjal-deo$"
flag = 0
while True:
    if (len(password)<8):
        flag = -1
        break
    elif not re.search("[a-z]", password):
        flag = -1
        break
    elif not re.search("[A-Z]", password):
        flag = -1
        break
    elif not re.search("[0-9]", password):
        flag = -1
        break
    elif not re.search("[_@$]", 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")
```

Not a Valid Password

In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js