**REGULAR EXPRESSIONS**

Regular Expressions are also know as REG X, which are a bunch of characters or a pattern that will create to search for a string within a given string or to validate the given string to see if tit follows the given pattern.

A close-up of words

Description automatically generated

**Example:**

Validate a string for email

Password Validation

Search for a substring using the pattern using some special characters

In Python, we have re the regular expression module, which has several useful functions.

1. Match
2. Search
3. Findall
4. Split
5. Sub

The regular expression syntax define some special characters called sequence characters that we can to match a single character in a given string.

1. \d 🡪 stand for any digit from 0 to 9
2. \D 🡪 any non digit character
3. \s 🡪 white space in a given string
4. \S 🡪 Opposite of it, which is non-white space
5. \w 🡪 for any Alpha Numeric Values
6. \W 🡪 Non Alpha Numeric Values
7. \b 🡪 Space around words
8. \A 🡪 Matches only at the start of the string it will do a search from right to the beginning of the string
9. \Z 🡪 matches only at the end of the string.

**Search Method (re.search)**

Search Method looks for the entire string based on the regular expression for the given string and given us the value, that it has number of times.

re.search(*r'o\w\w'*,strValue)

**FindAll Method (re.findall)**

The findall method takes a regular expression and it searches the given string for the pattern right from the beginning all the way to the end.

It will return all the sub-strings that match the regular expression as a list, which contain every substring in the given string that matches the given regular expression.

If none of them found, it will return a empty list.

re.findall(*r'd\w\w'*,strValue)

**Match Method (re.match)**

The match method takes the regular expression and it searches for the pattern in a given string right at the beginning of string.

re.match(*r'u\w\w'*,strValue)

**Substitute Method (re.sub)**

The sub method does a replace all that is it will substitute with the given string with the new value.

re.sub(*r'time'*,*'time only'*,strValue)

**Split Method (re.split)**

The Split method split the given string into a list of strings using the regular expression that we pass in as the delimiter.

**Code for all the above methods:**

import re

strValue=*"Take up one idea. One idea at a time."*

#CAN INVOKE THE GROUP METHOD WHEN ITS NOT NONE

print(*"STRING PROVIDED"*)

print(*"---------------"*)

print(strValue)

print()

print(*"re.search"*)

print(*"========="*)

searchResult1 = re.search(*r'o\w\w'*,strValue) # START THE SEARCH FROM 'o'

print(*"Search Result for (o): "*,searchResult1)

print(*"Search Result Group Output for (o): "*,searchResult1.group())

print()

searchResult1a = re.search(*r'z\w\w'*,strValue) # START THE SEARCH FROM 'z'

print(*"Search Result for(z): "*,searchResult1a)

print()

print(*"re.findall"*)

print(*"=========="*)

searchResult2 = re.findall(*r'd\w\w'*,strValue) # START THE SEARCH FROM 'd'. SEARCHES ONLY AT THE BEGINNING OF THE GIVNE STRING

print(*"FindAll Result Output for (d): "*, searchResult2)

print()

print(*"re.match"*)

print(*"========"*)

searchResult3a = re.match(*r'u\w\w'*,strValue) # START THE SEARCH FROM 'u'

print(*"FindAll Result Output for (u): "*, searchResult3a)

print()

searchResult3b = re.findall(*r'T\w\w'*,strValue) # START THE SEARCH FROM 'T'

print(*"FindAll Result Output for (T): "*, searchResult3b)

print()

print(*"re.sub"*)

print(*"=========="*)

searchResult4 = re.sub(*r'time'*,*'time only'*,strValue) # REPLACE 'time' WITH 'time only'

print(*"FindAll Result Output for (u): "*, searchResult4)

print()

strValue1=*"Take 1 up 23 one 345 idea. One 52 idea 6789 at a time."*

print(*"STRING PROVIDED FOR SPLIT"*)

print(*"-------------------------"*)

print(strValue1)

print()

print(*"re.sub"*)

print(*"======"*)

searchResult5 = re.split(*r'\d+'*, strValue1)

print(*"Split Result Output for the string : "*, searchResult5)

print()

**Output for all the above methods:**

STRING PROVIDED

---------------

Take up one idea. One idea at a time.

re.search

=========

Search Result for (o): <re.Match object; span=(8, 11), match='one'>

Search Result Group Output for (o): one

Search Result for(z): None

re.findall

==========

FindAll Result Output for (d): ['dea', 'dea']

re.match

========

FindAll Result Output for (u): None

FindAll Result Output for (T): ['Tak']

re.sub

==========

FindAll Result Output for (u): Take up one idea. One idea at a time only.

STRING PROVIDED FOR SPLIT

-------------------------

Take 1 up 23 one 345 idea. One 52 idea 6789 at a time.

re.sub

======

Split Result Output for the string : ['Take ', ' up ', ' one ', ' idea. One ', ' idea ', ' at a time.']

**QUANTIFIERS**

We can use Quantifies to match multiple characters.

If we use + to specify one or more repetitions of the preceding regular expression. \d+ 🡪 🡪 means one or more digits.

If we use \* quantifier, it means zero or more repetitions of the preceding regular expression.

If we use ? means zero or one repetition of the preceding regular expression

If we use {m} , this means exactly the m occurrences of the preceding regular expression

If we use {m,n}, this means m is the minimum number of occurrences and n is the maximum number of occurrences. By default, m will be zero and n will be infinity.

**Code for all Quantifiers:**

import re

strValue=*"Take up One idea. One idea at a time. Only on a time"*

print(*"STRING PROVIDED"*)

print(*"---------------"*)

print(strValue)

print()

print(*"re.findall using Quantifiers"*)

print(*"============================"*)

searchResult1 = re.findall(*r'O\w+'*,strValue) # START THE SEARCH FROM 'O'. SEARCHES ONLY AT THE BEGINNING OF THE GIVNE STRING

print(*"FindAll Result Output using Quantifiers (+) for value (O): "*, searchResult1)

print()

searchResult1 = re.findall(*r'O\w\*'*,strValue) # START THE SEARCH FROM 'O'. SEARCHES ONLY AT THE BEGINNING OF THE GIVNE STRING

print(*"FindAll Result Output using Quantifiers (\*) for value (O): "*, searchResult1)

print()

searchResult1 = re.findall(*r'O\w?'*,strValue) # START THE SEARCH FROM 'O'. SEARCHES ONLY AT THE BEGINNING OF THE GIVNE STRING

print(*"FindAll Result Output using Quantifiers (?) for value (O): "*, searchResult1)

print()

searchResult1 = re.findall(*r'O\w{1}'*,strValue) # START THE SEARCH FROM 'O'. SEARCHES ONLY AT THE BEGINNING OF THE GIVNE STRING

print(*"FindAll Result Output using Quantifiers ({1}) for value (O) followed by 1 letter : "*, searchResult1)

print()

searchResult1 = re.findall(*r'O\w{2}'*,strValue) # START THE SEARCH FROM 'O'. SEARCHES ONLY AT THE BEGINNING OF THE GIVNE STRING

print(*"FindAll Result Output using Quantifiers ({2}) for value (O) followed by 2 letter : "*, searchResult1)

print()

searchResult1 = re.findall(*r'O\w{3}'*,strValue) # START THE SEARCH FROM 'O'. SEARCHES ONLY AT THE BEGINNING OF THE GIVNE STRING

print(*"FindAll Result Output using Quantifiers ({3}) for value (O) followed by 3 letter : "*, searchResult1)

print()

searchResult1 = re.findall(*r'O\w{1,3}'*,strValue) # START THE SEARCH FROM 'O'. SEARCHES ONLY AT THE BEGINNING OF THE GIVNE STRING

print(*"FindAll Result Output using Quantifiers ({1,3}) for value (O): followed by Min Value is 1 and Max Value is 3 "*, searchResult1)

print()

**Output for all Quantifiers:**

STRING PROVIDED

---------------

Take up One idea. One idea at a time. Only on a time

re.findall using Quantifiers

============================

FindAll Result Output using Quantifiers (+) for value (O): ['One', 'One', 'Only']

FindAll Result Output using Quantifiers (\*) for value (O): ['One', 'One', 'Only']

FindAll Result Output using Quantifiers (?) for value (O): ['On', 'On', 'On']

FindAll Result Output using Quantifiers ({1}) for value (O) followed by 1 letter : ['On', 'On', 'On']

FindAll Result Output using Quantifiers ({2}) for value (O) followed by 2 letter : ['One', 'One', 'Onl']

FindAll Result Output using Quantifiers ({3}) for value (O) followed by 3 letter : ['Only']

FindAll Result Output using Quantifiers ({1,3}) for value (O): followed by Min Value is 1 and Max Value is 3 ['One', 'One', 'Only']

**DATES**

We will create regular expression that will return all the matching dates in a given string.

Usually date can start with either days or months, based on the country format.

**Code:**

import re

strValue=*"Take up 31-03-2024 One idea. One idea at a time. Only on a time 12-04-2024"*

print(*"STRING PROVIDED"*)

print(*"---------------"*)

print(strValue)

print()

searchResult1 = re.findall(*r'\d{1,2}-\d{1,2}-\d{4}'*,strValue) # START THE SEARCH FROM 'O'. SEARCHES ONLY AT THE BEGINNING OF THE GIVNE STRING

#r'\d{1,2}-\d{1,2}-\d{4}

print(*"FindAll Result Output using date format above string : "*, searchResult1)

print()

**Output:**

STRING PROVIDED

---------------

Take up 31-03-2024 One idea. One idea at a time. Only on a time 12-04-2024

FindAll Result Output using date format above string : ['31-03-2024', '12-04-2024']

**SPECIAL CHARACHERS**

With the sequence characters and the qualifiers that we have used, we can use certain special characters in our regular expressions as well.

\ 🡪 escape character

. 🡪 matches any character except new line

^ 🡪 match a given regular expression at the beginning of a string

$ 🡪 match a given regular expression at the end of a string

[…] 🡪 specify a range within the square brackets to match

[^…] 🡪 It matches every characters except the ones in the square brackets, so we can specify a range

(…) 🡪 Regular expressions which will match.

(R|S) 🡪 pipe symbol within () and specify multiple regular expressions. Both R and S are regular expressions

**Code:**

import re

strValue=*"Take up One idea. One idea at a time. Only on a time"*

print(*"STRING PROVIDED"*)

print(*"---------------"*)

print(strValue)

print()

searchResult1 = re.search(*r'^O\w'*,strValue)

#r'^O\w'

print(*"Special Character Result Output using (O) on above string using re.search : "*, searchResult1)

print()

searchResult2 = re.search(*r'^T\w\*'*,strValue)

#r'^O\w'

print(*"Special Character Result Output using (T) on above string using re.search: "*, searchResult2)

print(*"Special Character Result Output using (T) on above string using re.search (Using Group): "*, searchResult2.group())

print()

**Output:**

STRING PROVIDED

---------------

Take up One idea. One idea at a time. Only on a time

Special Character Result Output using (O) on above string using re.search : None

Special Character Result Output using (T) on above string using re.search: <re.Match object; span=(0, 4), match='Take'>

Special Character Result Output using (T) on above string using re.search (Using Group): Take

**REGULAR EXPRESSION USING WEB SCRAPING**

Web Scrapping is the process of collecting information from web pages.

**Code:**

import re

import urllib.request #IS THE PACAKGE THAT HAS CLASSES WHICH WILL ALLOW US TO HIT THE URL AND GET THE HTML CONTENTS OF IT.

sites=[*"google.com"*, *"espncricinfo.com"*]

for s in sites:

print(*"Searching : "*, s)

response=urllib.request.urlopen(*"http://"*+s) #OPEN UP THE URL AND READE THE RESPONSE WHICH RETURNS RAW

text = response.read()

# FIND ALL METHOD TO SEARCH THE TITLE IN THE HTML TEXT THAT COMES BACK IN RESPONSE

# RAW TEXT IS CONVERTED INTO A STRING, WHICH IS PASSING IT TO THE FIND ALL METHOD ALING WITH THE REGULAR EXPRESSION YOU WANT TO MATCH OR FIND.

# ALSO DOING A CASE INSENSITIVE TEST USING re.I

title= re.findall(*"<title>.\*</title>"*, str(text), re.I)

print(title[0])

**Output:**

Searching : google.com

<title>Google</title>

Searching : espncricinfo.com

<title>Today&#x27;s Cricket Match | Cricket Update | Cricket News | ESPNcricinfo</title>

**QUIZ**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated