Regular Expression Introduction

- Need to extract information from given data
- Examples
  - need number of people eho contacted us last month through gmail
  - need phone number of employees whose names starts with 'A'
  - DoB of patients in Hospitals taking treatment for diabetes
- Search and extract that data for further use:

Regular Expression

Regular Expression

- A regular expression is a string that contains special symbols and characters to find and extract the information needed by us from the given data.
- A regular expression helps us to search and information, match, find, and split information as per our requirements.
- A regular expression is simply called as regex.
- Available in Java, perl etc

're' module

- Python provides ' re ' module
- This module contains methods like compile(), search(), match(), findall(), split() etc which are used in finding the information available in the available data.
- import re
- we write regular expressions as raw string in single quotes

Regular Expression

- Example:

  reg=r'm\w\w'
- word with three characters starting with 'm' followed by 2 alphanumeric letters
- r- rawstring - deactivates the escape sequences
- can be also written as reg='m\\w\\w'

compile()

- Next step is to compile the expression using compile()method of 're' module as:

prog= re.compile(r'm\w\w')

Run the expression and print

- After compiling, run the expression using search() method or match() method as:

  str='cat mat bat rat' # str on which re will act

  result=prog.search(str) #searching for re in str

- The result is stored in 'result' object and can display it by calling the group()method on the object as:

  print(result.group()) #mat

Complete Example

```
import re
prog= re.compile(r'm\w\w')
str='cat mat bat rat'
result=prog.search(str)
print(result.group())
```

Another Example which uses same 're'

```
str1='Operating system format'
result=prog.search(str1)
print(result.group()) #mat
```

compile and run

str1=”Operating system format'

result=re.search(r'm\w\w',str1)

equivalent to:

prog= re.compile(r'm\w\w')

result=prog.search(str)

Example- search()

create a reg expression to search for strings starting with m and having total 3 characters using the search() method

```
import re
str='man sun mop run'
result=re.search(r'm\w\w',str) #first occurence only
if result: #returns 'None' if pattern is not matching
    print(result.group()) #o/p: man
```

Example- findall()

create a reg expression to search for strings starting with m and having total 3 characters using the findall() method

```
import re
str='man sun mop run'
result=re.findall(r'm\w\w',str) #returns all the occurences
                                #return as a list
print(result) #['man', 'mop']
#for i in result: print(i)
```

Example- match()

create a reg expression to search for strings starting with m and having total 3 characters using the match() method

```
import re
Eg1:
str='man sun mop run'
result=re.match(r'm\w\w',str) #string is found in the begining
print(result.group()) #man
Eg2:
str='sun man mop run'
result=re.match(r'm\w\w',str)
print(result.group()) #None
```

split() method

- splits te given string into pieces according to the regular expression and returns the pieces as elements as a list.
- re.split(r'\W+',str)
- W- non alpha numeric character
- +- 1 or more occurences

Example- split() method

- create a reg expression to split a string into pieces where one or more nonalpha numeric characters are found

```
import re
str='This; is the: “core” Python\'s book'
result=re.split(r'\W+',str)
print(result)
```

output: ['This', 'is', 'the', 'core', 'Python', 's', 'book']

sub() method

- find the string and then replaces it with a new string
- use sub() method of 're' module
- re.sub(regex,new str, str)

Example

```
import re
str='Amphisoft is located in coimbatore'
res=re.sub(r'coimbatore','Nellore',str)
print(res)
```

Summary:Operations of Regular Expression

- Matching strings
- Searching for strings
- Finding all strings
- Splitting a string into pieces
- Replacing strings

Summary:Operations of Regular Expression

Operation	Method name	Function description	Output	To display the result
Matching strings	match()	searches in the begining of the string	string or None	group()
Searching for strings	search()	searches the entire string	returns the first occurence of the matching string or None	group()
Finding all strings	findall()	searches the entire string	returns the list of matching strings or empty list	for loop
Splitting a string into pieces	split()	splits the strings according to the reg exp	returns the list of strings or empty list	for loop
Replacing strings	sub()	substitues(or replaces) new strings in the place of existing strings	main string is returned by this method	-

Sequence Characters in regex

Character	Its Description
\d	represents any digit (0-9)
\D	represents any non-digit
\s	represents white space (\n\t\r\f\v)
\S	represents non white space char
\w	represents any alphanumeric(A-Z,a-z,0-9)
\W	represents non -alphanumeric
\b	represents a space around words
\A	Matches only at start of the string
\Z	Matches only at end of the string

Examples

- create a reg expression to retrieve all words starting with 'a' in a given string

```
import re
str='an apple a day keeps the doctor away'
result=re.findall(r'a[\w]*',str)
for word in result:
        print(word,end=” “)
```

output:

an apple a ay away #ay from day- wrong

Examples- revised

- create a reg expression to retrieve all words starting with 'a' in a given string

```
import re
str='an apple a day keeps the doctor away'
result=re.findall(r'\ba[\w]*\b',str)
for word in result:
        print(word,end=” “)
```

output:

an apple a away

Examples- beg with digits

- create a reg expression to retrieve all words starting with a numeric digit

```
import re
str='The meeting will be conducted on 1st and 20th of every month'
result=re.findall(r'\d[\w]*',str)
for word in result:
        print(word,end=” “)
```

output:

1st 20th

Examples - 'm' occurences

- create a reg expression to retrieve all words having 5 chars length

```
import re
str='one two three four five six seven 8 9 10'
result=re.findall(r'\b\w{5}\b',str)
print(result)
```

output:

```
['three', 'seven']
```

Examples - 'm' occurences

- create a reg expression to retrieve all words having atleast 4 chars length

```
import re
str='one two three four five six seven 8 9 10'
result=re.findall(r'\b\w{4,}\b',str)
print(result)
```

output:

```
['three', 'four', 'five', 'seven']
```

Examples - 'm' occurences

- create a reg expression to retrieve all words having with 3 or 4 or 5 chars length

```
import re
str='one two three four five six seven 8 9 10'
result=re.findall(r'\b\w{3,5}\b',str)
print(result)
```

output:

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

Examples - 'm' occurences

- create a reg expression to retrieve only single digits from a string

```
import re
str='one two three four five six seven 8 9 10'
result=re.findall(r'\b\d\b',str)
print(result)
```

output:

```
['8', '9']
```

Examples - 'm' occurences

- create a reg expression to retrieve only double digits from a string

```
import re
str='one two three four five six seven 8 9 10'
result=re.findall(r'\b\d\d\b',str) #or re.findall(r'\b\d{2}\b',str)
print(result)
```

output:

```
['10']
```

Examples - 'm' occurences

- create a reg expression to retrieve the last word of a string , if it starts with 't'

```
import re
str='one two three one two three'
result=re.findall(r't[\w]* \Z',str)
print(result)
```

output:

```
['three']
```

Quantifiers in regex

Character	Its Description
+	1 or more repititions of the preceding regex
*	0 or more repititions of the preceding regex
?	0 or 1 repititions of the preceding regex
{m}	Exactly m occurences
{m,n}	from m to n, m defaults to 0 and n to infinity
[ab]	either a or b
[A-Z] [a-z]	range - any letter between A to Z range - any letter between a to z

Examples - quantifiers

- create a reg expression to retrieve the phone number of a person

```
import re
str='ebox: 98940 12234'
result=re.search(r'\d+',str)
print(result.group())
```

output:

98940 12234

Examples -quantifiers

- create a reg expression to retrieve only the name of a person

```
import re
str='ebox: 98940 12234'
result=re.search(r'\D+',str)
print(result.group())
```

output:

ebox:

Examples -quantifiers

- create a reg expression to find all words starting with 'an' or 'ak'

```
import re
str='anil akhil anant arun arati arundati abhijit ankur'
result=re.findall(r'a[nk][\w]*',str)
print(result)
```

output:

['anil', 'akhil', 'anant', 'ankur']

Examples -quantifiers

- create a reg expression to find all words starting with a capital letter followed by nay number of lowercase letters eg.Rahul

```
import re
str='Anil Akhil Anant Arun Rahul chitra arundati Abhijit Ankur'
result=re.findall(r'[A-Z][a-z]*',str)
print(result)
```

output:

['Anil', 'Akhil', 'Anant', 'Arun', 'Rahul', 'Abhijit', 'Ankur']

Examples -quantifiers

- create a reg expression to retrieve DOB from a string

```
import re
str='abi 20 1-5-2001, rohit 21 22-10-1990, sita 22  15-09-2000'
result=re.findall(r'\d{2}-\d{2}-\d{4}',str)
print(result)
```

output:

['22-10-1990', '15-09-2000']

Examples -quantifiers-revised

- create a reg expression to retrieve DOB from a string

```
import re
str='abi 20 1-5-2001, rohit 21 22-10-1990, sita 22  15-09-2000'
result=re.findall(r'\d{1,2}-\d{1,2}-\d{4}',str)
print(result)
```

output:

['1-5-2001', '22-10-1990', '15-09-2000']

Special Characters in regex

Character	Its Description
\	escape spl char nature
.	Matches any character except newline
^	Matches begining of a string
$	Matches ending of a string
[...]	denotes a set of possible characters
[^...]	matches every char except the ones inside []
(...)	matches the regex inside the () and the result can be captured
R|S	matches either regex R or regex S

Examples

- create a reg expression to search whether a given string is starting with 'He' or not

```
import re
str='Hello World!'
res=re.search(r'^He',str)
if res: print(“String starts with 'He'”)
else: print(“String doesnot starts with 'He'”)
```

output:

```
String starts with 'He'
```

Examples

- create a reg expression to search for a word at the end of the string

```
import re
str='Hello World!'
res=re.search(r'World$',str) #case sensitive
if res: print(“String ends with 'World'”)
else: print(“String doesnot ends with 'World'”)
```

output:

```
String ends with 'World'
```

Examples

- create a reg expression to search for a word at the end of the string ignoring the case

```
import re
str='Hello World!'
res=re.search(r'world$',str, re.IGNORECASE) #case sensitive
if res: print("String ends with 'world'")
else: print("String doesnot ends with 'world'")
```

output:

```
String ends with 'world'
```

Examples

- create a reg expression to retrieve marks and names from a given string

```
import re
str='Rahul got 75 marks, vijay got 55 marks whereas subbu got 98 marks'
#extracting only numbers/digits
marks=re.findall(r'\d{2}',str);print(marks)
#extracting names starting with Capital Letters
names=re.findall(r'[A-Z][a-z]*',str);print(names)
```

Output: ['75', '55', '98']

['Rahul']

Examples

- create a reg expression to retrieve the timings either 'am' or 'pm'

```
import re
str='The meeting may be at 8am or 9am or 4pm or 5pm'
times=re.findall(r'\dam|\dpm',str)
print(times)
```

Output: ['8am', '9am', '4pm', '5pm']

Simple Email verfication

```
import re
email = "jeeva@gmail.com"
p = re.match(r'\w+@\w+\.\w{2,3}' ,email)
print(p.group())
#p = re.match(r'\w+@\w+[.][a-z]{2,3}' ,email)
```

Exercise

```
import re
s="This is a python programming class"
x=re.findall('[abc]',s)
print(x)
```

Output??

Exercise

```
import re
s="This is a python 90 programming class"
x=re.findall('\d',s)#\D
print(x)
```

Output??

Exercise

```
import re
s="This is a python 90 programming class"
x=re.findall('[0-9]',s)
print(x)
```

Output??

Exercise

```
import re
s="This is a python 90 programming class T"
x=re.findall('Th.....',s)
print(x)
```

Output??

Exercise

```
import re
x=re.findall('cla.*',s)
print(x)
x=re.findall('cla.+',s)
print(x)
x=re.findall('cla.?',s)
print(x)
```

Output??

Exercise

```
import re
s="This is a python 90 programming class"
x=re.findall('^This',s) #\A
print(x)
x=re.findall('s$',s) #\Z
print(x)
```

Output??

Exercise

```
import re
s="This is a python 90 programming classs"
x=re.findall('clas{3}',s)
print(x)
```

Output??

Any Queries?

Thank You All!!