Regular Expressions

The Regular Expression is a special sequence of charecters,which is used to find or match a pattern into the string or group of strings.

use-cases of Regular Expressions:

1).to get the required data from the Given data

2).to perform the data validations

3).to generate the url patterns in a web application

....
....etc.,

what are Special Charecter's:

1) * --> it match a pattern 0 or more occurencess of preceeding charecter.

ex: ab*c

ac,abc,abbc,abbbc,.... --> valid statements

2) + --> it match a pattern 1 or more occurencess of preceeding charecter.

ex: ab+c

ac --> invalid statement

abc,abbc,abbbc,..... --> valid statement

3) . --> it matches any one charecter

a.c

abc,aqc,aac,aAc,a4c,a@c,... --> valid statements

abbc,aabc,a!Ac,.... --> invalid statements

4) ? --> it matches 0 or 1 occurencess of preceeding charecter.

ex: ab?c

ac,abc --> valid statements

abbc,abbbc,abbbbc,.... --> invalid statements

5) [ ] --> it matches any one charecter from the given list of charecters.

ex: x[aeiou]y

xay,xey,xiy,xoy,xuy --> valid statements

xby,xAy,xpy,x@y,x6y,.... --> invalid statements

6) [^] --> it matches any one charecter otherthan given list of charecters.

ex: x[^aeiou]y

xay,xey,xiy,xoy,xuy --> invalid statements

xby,xAy,xpy,x@y,x6y,.... --> valid statements

7) [-] --> it matches any one charecter from the given range of charecters.

ex: x[a-d]y

xay,xby,xcy,xdy --> valid statements

xAy,xey,x@y,xpy,x9y,.... --> invalid statements

[0-9] --> it matches any one digit

[^0-9] --> it matches any one non-digit

[a-z] --> it matches any one lowercase alphabet

[^a-z] --> it matches any one non lowercase alphabets

[A-Z] --> it matches any one Uppercase alphabet

[^A-Z] --> it matches any one non uppercase alphabet

[a-zA-Z] --> it matches any one alphabet

[^a-zA-Z] --> it matches any one non alphabet

[a-zA-Z0-9] --> it matches any one alphanumeric

[^a-zA-Z0-9] --> it matches any one non alphanumeric

8) | --> it matches any one string from the given list of

strings.

ex: a(python|django|flask)b

apythonb,aflaskb,adjnagob --> valid statements

ajavab,aphpb,arubyb,aperlb,... --> invalid statements

9) {m} --> it matches exactly 'm' occurencess of preceeding charecter.

ab{3}c

abbbc --> valid statement

ac,abc,abbc,abbbbc,abbbbbc,... --> invalid statement

10) {m,n} --> it matches minimum 'm' occurencess and maximum 'n' occurencess of preceeding charecter.

ab{3,5}c

abbbc,abbbbc,abbbbbc --> valid statements

ac,abc,abbc,abbbbbbc,abbbbbbbc,.. Invalid statements

11) {m,} --> it matches minimum 'm' occurencess and maximum no limit of preceeding charecter.

ab{3,}c

abbbc,abbbbc,abbbbbc,abbbbbbc,... --> valid statements

ac,abc,abbc --> invalid statements

12) ^ --> it match a pattern at the begening of the string

13) $ --> it match a pattern at the ending of the string

14) \d --> it matches any one digit

15) \D --> it matches any one non-digit

16) \w --> it matches any one alphanumeric

17) \W --> it matches any one non-alphanumeric

18) \s --> it matches space(' '),tab char('\t'), newline('\n').

if any confusion with regular expression patterns and strings,in that case we can represent the regular expression patterns by using raw string to avoid that confusion.

```
        r'expression'

             (or)

        r"expression"
```

if we want to working with regular expression patterns to required 're' module.

the 're' module provides some functions to perform the operations on the strings,they are

```
        1).match()
        2).search()
        3).findall()
        4).finditer()
        5).sub()
        6).split()
```

match()
---
the match() is used to match the pattern at the begening of the string only.

if matching pattern is find/match at the begening of the string,that match() to return matching object otherwise the match() to return None.

```
        match(pattern,string,flags=0)
```

ex1:

```
import re
string=input("enter your string: ")
pattern=input("enter your matching pattern: ")
match_obj=re.match(pattern,string)
print(match_obj)
```

output1:

```
enter your string: hai siva krishna good afternoon
enter your matching pattern: hai
<re.Match object; span=(0, 3), match='hai'>
```

output2:

```
enter your string: hai siva krishna good afternoon
enter your matching pattern: siva
None
```

```
output3:
-------
enter your string: hai siva krishna good afternoon
enter your matching pattern: rama
None

output4:
-------
enter your string: hai siva krishna good afternoon
enter your matching pattern: Hai
None

note:
----
if we want to ignore the cases in that case we are using re.I flag

ex2:
---
import re
string=input("enter your string: ")
pattern=input("enter your matching pattern: ")
match_obj=re.match(pattern,string,re.I)
print(match_obj)

output:
------
enter your string: hai siva krishna good afternoon
enter your matching pattern: Hai
<re.Match object; span=(0, 3), match='hai'>

ex3:
---
import re
string=input("enter your string: ")
pattern=input("enter your matching pattern: ")
match_obj=re.match(pattern,string,re.I)
if match_obj:
    print("Input string:",match_obj.string)
    print("Regular Expression:",match_obj.re)
    print("Matching Pattern:",match_obj.group())
    print("Span:",match_obj.span())
    print("Starting at:",match_obj.start())
    print("Ending at:",match_obj.end())
else:
    print("No Match!")

output1:
-------
enter your string: hai siva krishna good afternoon
```

```
enter your matching pattern: Hai
Input string: hai siva krishna good afternoon
Regular Expression: re.compile('Hai', re.IGNORECASE)
Matching Pattern: hai
Span: (0, 3)
Starting at: 0
Ending at: 3

output2:
------
enter your string: hai siva krishna good afternoon
enter your matching pattern: siva
No Match!

search()
-------
to match the pattern at any where in the string.

if matching pattern is find/match,the search() to return that matching object
otherwise the search() to return None.

in our string object multiple matches are there,in that case the search() to return
only first occurence matching pattern object.

        search(pattern,string,flags=0)

ex4:
---
import re
string=input("enter your string: ")
pattern=input("enter your matching pattern: ")
match_obj=re.search(pattern,string)
print(match_obj)

output1:
-------
enter your string: hai siva krishna good afternoon
enter your matching pattern: hai
<re.Match object; span=(0, 3), match='hai'>

output2:
--------
enter your string: hai siva krishna good afternoon
enter your matching pattern: krishna
<re.Match object; span=(9, 16), match='krishna'>

output3:
-------
enter your string: afternoon
enter your matching pattern: t
```

```
<re.Match object; span=(2, 3), match='t'>

output4:
--------
enter your string: hai siva krishna good afternoon
enter your matching pattern: afternoon
<re.Match object; span=(22, 31), match='afternoon'>

output5:
-------
enter your string: hai siva krishna good afternoon
enter your matching pattern: rama
None

output6:
------
enter your string: hai siva krishna good afternoon
enter your matching pattern: Hai
None

ex5:
----
import re
string=input("enter your string: ")
pattern=input("enter your matching pattern: ")
match_obj=re.search(pattern,string,re.I)
print(match_obj)

output1:
--------
enter your string: hai siva krishna good afternoon
enter your matching pattern: Hai
<re.Match object; span=(0, 3), match='hai'>

output2:
--------
enter your string: hai siva Hai rama hai krishna
enter your matching pattern: Hai
<re.Match object; span=(0, 3), match='hai'>

findall()
--------
the findall() to return the list of all the matching patterns.

        findall(pattern,string,flags=0)

if there's no match,the findall() to return empty list object.

ex6:
---
```

```
import re
string=input("enter your string: ")
pattern=input("enter your matching pattern: ")
matches=re.findall(pattern,string)
print(matches)

output1:
-------
enter your string: hai siva Hai rama hai krishna
enter your matching pattern: Hai
['Hai']

output2:
--------
enter your string: hai siva Hai rama hai krishna
enter your matching pattern: hello
[]

ex7:
----
import re
string=input("enter your string: ")
pattern=input("enter your matching pattern: ")
matches=re.findall(pattern,string,re.I)
print(matches)

output:
------
enter your string: hai siva Hai rama hai krishna
enter your matching pattern: Hai
['hai', 'Hai', 'hai']

finditer()
----------
to return the matching object for each match match.

if there's no match,in that case the finditer() to return the empty match_object.

        finditer(pattern,string)

ex8:
----
import re
string=input("enter your string: ")
pattern=input("enter your matching pattern: ")
match_objects=re.finditer(pattern,string)
for match_obj in match_objects:
    print(match_obj)

output1:
```

```
-------
enter your string: hai siva Hai rama hai krishna
enter your matching pattern: Hai
<re.Match object; span=(9, 12), match='Hai'>

output2:
--------
enter your string: hai siva Hai rama hai krishna
enter your matching pattern: hai
<re.Match object; span=(0, 3), match='hai'>
<re.Match object; span=(18, 21), match='hai'>

output3:
--------
enter your string: hai siva Hai rama hai krishna
enter your matching pattern: hello

ex9:
---
import re
string=input("enter your string: ")
pattern=input("enter your matching pattern: ")
match_objects=re.finditer(pattern,string,re.I)
for match_obj in match_objects:
    print(match_obj)

output:
------
enter your string: hai siva Hai rama hai krishna
enter your matching pattern: Hai
<re.Match object; span=(0, 3), match='hai'>
<re.Match object; span=(9, 12), match='Hai'>
<re.Match object; span=(18, 21), match='hai'>

sub()
-----
sub means substitute

to match the pattern and replace with another pattern

        sub(pattern,replacedpattern,string)

ex10:
----
import re
string=input("enter your string: ")
pattern=input("enter your matching pattern: ")
replaced_pattern=input("enter your replaced pattern: ")
print(re.sub(pattern,replaced_pattern,string))
```

```
output1:
------
enter your string: hai siva Hai rama hai krishna
enter your matching pattern: hai
enter your replaced pattern: hello
hello siva Hai rama hello krishna

output2:
-------
enter your string: hai siva Hai rama hai krishna
enter your matching pattern: hey
enter your replaced pattern: hello
hai siva Hai rama hai krishna

split()
-------
the split(),to split the string into words based on given delimiters.

the split() to return the output as list object.

        split(pattern,string)

ex11:
----
import re
string=input("enter your string: ")
pattern=input("enter your matching pattern: ")
print(re.split(pattern,string))

output:
------
enter your string: hai siva,krishna;good:morning,how are:you
enter your matching pattern: [\s,;:]+
['hai', 'siva', 'krishna', 'good', 'morning', 'how', 'are', 'you']

ex12:
----
wap to print matching dates and months from the given string?

import re
string=input("enter your string: ")
pattern=input("enter your matching pattern: ")
matches=re.findall(pattern,string)
for match in matches:
    print(match)

output1:
------
enter your string: Jan 14 JUN 10 sep dec 5
enter your matching pattern: [a-zA-Z]+ [0-9]+
```

```
Jan 14
JUN 10
dec 5

output2:
-------
enter your string: Jan 14 JUN 10 sep dec 5
enter your matching pattern: [a-zA-Z]+ \d+
Jan 14
JUN 10
dec 5

output3:
------
enter your string: Jan 14 JUN 10 sep dec 5
enter your matching pattern: ([a-zA-Z]+) \d+
Jan
JUN
dec

output4:
-------
enter your string: Jan 14 JUN 10 sep dec 5
enter your matching pattern: [a-zA-Z]+ (\d+)
14
10
5

ex13:
----
        emp.txt
        -------
101 siva 3000
RAMA 4000 102
2000 103 KrIsHnA

import re
with open('emp.txt') as f1:
    matches=re.findall(r'[a-zA-Z]+',f1.read())
    for match in matches:
        print(match)

output:
-------
siva
RAMA
KrIsHnA

ex14:
----
```

```
        emp.txt
        ------
101 siva 3000 d.siva143@GMAIL.COM
RAMA 4000 rama4u@Yahoo.in 102
d_RAMA4u@outlook.live 2000 103 KrIsHnA


import re
with open('emp.txt') as f1:
    matches=re.findall(r'[\w._]+@[a-zA-Z]+.[a-zA-Z]+',f1.read())
    for match in matches:
        print(match)

output:
------
d.siva143@GMAIL.COM
rama4u@Yahoo.in
d_RAMA4u@outlook.live

ex15:
-----
        emp.txt
        -------
8886501571
786578658
6756474
9098989795
7867574
679808754
7908769690
67586979
909867657
8019658644
97857575
6756586962

import re
with open('emp.txt') as f1:
    matches=re.findall(r'\d{10}',f1.read())
    for match in matches:
        print(match)

output:
-------
8886501571
9098989795
7908769690
8019658644
6756586962
```

ex16:

```
import re
with open('emp.txt') as f1:
    matches=re.findall(r'8\d{9}',f1.read())
    for match in matches:
        print(match)
```

output:

```
8886501571
8019658644
```

ex17:

```
import re
with open('emp.txt') as f1:
    matches=re.findall(r'[86]\d{9}',f1.read())
    for match in matches:
        print(match)
```

output:

```
8886501571
8019658644
6756586962
```

ex18:

```
import re
with open('emp.txt') as f1:
    matches=re.findall(r'[6-8]\d{9}',f1.read())
    for match in matches:
        print(match)
```

output:

```
8886501571
7908769690
8019658644
6756586962
```

ex19:

```
        emp.txt
        -------
ap07tf6765
ts03hj7876
ka06jk8987
tn09hj7865
```

```
AP09FD6752
TS02HL8971
ap06gf3421
mp09gh4326
KA12jk8987

import re
with open('emp.txt') as f1:
    matches=re.findall(r'ap\w+|ts\w+',f1.read(),re.I)
    for match in matches:
        print(match)

output:
------
ap07tf6765
ts03hj7876
AP09FD6752
TS02HL8971
ap06gf3421

ex20:
----
        emp.txt
        -------
hai siva
hello krishna
Hai rama how are you
hello guido where you
hai rossum hai

import re
with open('emp.txt') as f1:
    matches=re.findall(r'^hai',f1.read())
    for match in matches:
        print(match)

output:
------
hai

ex21:
----
import re
with open('emp.txt') as f1:
    matches=re.findall(r'^hai',f1.read(),re.M)
    for match in matches:
        print(match)

output:
------
```

```
hai
hai
```

ex22:

```
----
import re
with open('emp.txt') as f1:
    matches=re.findall(r'^hai',f1.read(),re.M|re.I)
    for match in matches:
        print(match)
```

output:

```
------
hai
Hai
hai
```

ex23:

```
---
import re
with open('emp.txt') as f1:
    matches=re.findall(r'^hai.*',f1.read(),re.M|re.I)
    for match in matches:
        print(match)
```

output:

```
------
hai siva
Hai rama how are you
hai rossum hai
```

ex24:

```
----
import re
with open('emp.txt') as f1:
    matches=re.findall(r'^hai.*hai$',f1.read(),re.M|re.I)
    for match in matches:
        print(match)
```

output:

```
-------
hai rossum hai
```