Python Regular Expression

Prepared by,

Dr. K. Vallidevi

Regular Expression – Package - re

- A RegEx, or Regular Expression, is a sequence of characters that forms a search pattern.
- RegEx can be used to check if a string contains the specified search pattern.
- Python has a built-in package called re, which can be used to work with Regular Expressions.
- used frequently for web page “Scraping” (extract large amount of data from websites)

Example Program

```
import re

#Check if the string starts with "The" and ends with "Spain":

txt = "The rain in Spain"
x = re.search("^The.*Spain$", txt)

if x:
  print("YES! We have a match!")
else:
  print("No match")
```

Use of findall()

```
import re

#Return a list containing every occurrence of "ai":

txt = "The rain in Spain"
x = re.findall("ai", txt)
print(x)
```

Output:
['ai', 'ai']

If no matches are found, an empty list is returned

- The list contains the matches in the order they are found.

Example :

```
import re
txt = "The rain in Spain"
#Check if "Portugal" is in the string:
x = re.findall("Portugal", txt)
print(x)
if (x):
  print("Yes, there is at least one match!")
else:
  print("No match")
```

Output:

```
[]
No match
```

The search() returns the first occurrence alone

```
import re

txt = "The rain in Spain"
x = re.search("\s", txt)

print("The first white-space character is located in
position:", x.start())
```

```
Output:
The first white-space character is located in
position: 3
```

If no matches are found the value returns None

```
import re

txt = "The rain in Spain"
x = re.search("Portugal", txt)
print(x)
```

```
Output: None
```

The split() in re acts the same as that of string split()

```
import re
#Split the string at every white-space character:
txt = "The rain in Spain"
x = re.split("\s", txt)
print(x)
```

Output:

['The', 'rain', 'in', 'Spain']

Replace with the sub()

```
import re
#Replace all white-space characters with the digit "9":
txt = "The     rain in Spain"
x = re.sub("\t", "9", txt)
y = re.sub("\s", "##", x, 1)
print(y)
```

Output:

The9rain##in Spain

Use of [] in findall()

```
import re
txt = "The rain in Spain"
#Find all lower case characters alphabetically between "a" and "m":
x = re.findall("[a-m]", txt)
print(x)
```

Output:

```
['h', 'e', 'a', 'i', 'i', 'a', 'i']
```

Use of \d in findall()

```
import re
txt = "That will be 59 dollars"
#Find all digit characters:
x = re.findall("\d", txt)
print(x)
```

Output:

```
['5', '9']
```

Zero or more occurrences with a *

```
import re

txt = "hello heoplanet"

#Search for a sequence that starts with "he", followed by 0 or more (any) characters, and an "o":

x = re.findall("he.*o", txt)

print(x)
```

Output:

```
['hello heo']
```

One or more occurrences with a +

```
import re

txt = "hello heplanet"

#Search for a sequence that starts with "he", followed by 1 or more (any) characters, and an "o":

x = re.findall("he.+o", txt)

print(x)
```

Output:

```
['hello']
```

Zero or one occurrences with a “?”

```
import re

txt = "hello planet"

#Search for a sequence that starts with "he", followed by 0 or 1  (any) character, and an "o":

x = re.findall("he.?o", txt)

print(x)

#This time we got no match, because there were not zero, not one, but two characters between "he" and the "o“
```

Output: []

Note: Remove one character “l” from the word “hello” .

Output: ['helo']

Use of {} in regular expressions

```
import re
txt = "The sequence is helo planet"
#Search for a sequence that starts with "he", followed excactly 2 (any) characters, and an "o":
x = re.findall("he.{2}o", txt)
print(x)
```

Output:

[]

Note: If the spelling for the word hello(corrected spelling) is changed in txt, then the output will be ['hello']

Use of OR operator (|) in regular expressions

```
import re
txt = "The rain falls in Spain falls mainly in stays the plain!"
#Check if the string contains either "falls" or "stays":
x = re.findall("falls|stays", txt)
print(x)
if x:
  print("Yes, there is at least one match!")
else:
  print("No match")
```

Output:

```
['falls', 'falls', 'stays']
Yes, there is at least one match!
```

Return a match for every non word character

```
import re
txt = "The rain in#Spain!"
#Return a match at every NON word character (characters NOT between a and Z. Like "!", "?" white-space etc.):
x = re.findall("\W|!|#", txt)
print(x)
if x:
  print("Yes, there is at least one match!")
else:
  print("No match")
```

Output:

```
[' ', ' ', '#', '!']
Yes, there is at least one match!
```

Regular Expression with a list of strings

```
import re
str1=["AaBbGg", "Python", "python", "PYTHON", "aA", "Aa"]
patterns = '[A-Z]+[a-z]+$'
for i in str1:
        if re.search(patterns, i):
                print('Found a match!')
        else:
                print('Not matched!’)
```

Output:

```
Found a match!
Found a match!
Not matched!
Not matched!
Not matched!
Found a match!
```

Split the string at the first white-space character

```
import re
txt = "The rain in Spain"
x = re.split("\s", txt, 1)
print(x)
```

Output:

```
['The', 'rain in Spain']
```

Search for an upper case "S" character in the beginning of a word, and print its position

```
import re
txt = "The rain in Spain"
x = re.search(r"\bS\w+", txt)
print(x.span())
```

Output:

(12, 17)

Use of search() in regular expression with list

```
import re
str1=["The quick brown fox jumps over the lazy dog.", " The quick brown fox jumps over the lazy dog.", "!12The quick brown fox jumps over the lazy dog."]
patterns = '^\w+'
for i in str1:
        if re.search(patterns, i):
                print('Found a match!')
        else:
                print('Not matched!’)
```

Output:

Found a match!

Not matched!

Not matched!

Use of [] in Regular Expression

```
import re
txt = "T2he r1ain3 in S4pain"
#Check if the string has any 0, 1, 2, or 3 digits:
x = re.findall("[0123]", txt)
print(x)
if x:
  print("Yes, there is at least one match!")
else:
  print("No match")
```

Output:

```
['2', '1', '3']
Yes, there is at least one match!
```

Use of [][] in regular expressions

```
import re
txt = "08 times before 11:45 AM"
#Check if the string has any two-digit numbers, from 00 to 59:
x = re.findall("[0-5][0-9]", txt)
print(x)
if x:
  print("Yes, there is at least one match!")
else:
  print("No match")
```

[] being used for alphabets and numbericals

```
import re
txt = "At 8 times before 11:45 AM at 5"
#Check if the string has any characters from a to z lower case, and A to Z upper case:
x = re.findall("[0-9a-zA-Z]", txt)
print(x)
if x:
  print("Yes, there is at least one match!")
else:
  print("No match")
```

Output:

```
['A', 't', '8', 't', 'i', 'm', 'e', 's', 'b', 'e', 'f', 'o', 'r', 'e', '1', '1', '4', '5', 'A', 'M', 'a', 't', '5']
Yes, there is at least one match!
```

[+] being searched

```
import re
txt = "8 times before + 11:45 AM"
#Check if the string has any + characters:
x = re.findall("[+]", txt)
print(x)
if x:
  print("Yes, there is at least one match!")
else:
  print("No match")
```

Output:

```
['+']
Yes, there is at least one match!
```

Shuffle the elements in the list

```
import random
nums = [1, 2, 3, 4, 5]
print("Original list:")
print(nums)
random.shuffle(nums)
print("Shuffle list:")
print(nums)
words = ['red', 'black', 'green', 'blue’]
print("\nOriginal list:")
print(words)
random.shuffle(words)
print("Shuffle list:")
print(words)
```

- **“^”:** This expression matches the start of a string
- **“w+**“: This expression matches the alphanumeric character in the string
- [a-e] is the same as [abcde].
- [1-4] is the same as [1234].
- [0-39] is the same as [01239].
- [^abc] means any character except a or b or c.
- [^0-9] means any non-digit character.

Take aways

. - Period

A period matches any single character (except newline '\n’).

The dollar symbol $ is used to check if a string ends with a certain character.

Consider this code: {n,m}. This means at least n, and at most m repetitions of the pattern left to it.