Regular Expression in Python with Examples?

Regular expressions is a kind of programming language which is used to identify whether a pattern exists in a given sequence of characters (string) or not.

Regular expression or Regex is a sequence of characters that is used to check if a string contains the specified search pattern.

RegEx Module

To use RegEx module, python comes with built-in package called re, which we need to work with Regular expression. To use RegEx module, just import re module.

```
import re
```

Example

```
import re
txt = "Use of python in Machine Learning"
x = re.search("^Use.*Learning$", txt)
if (x):
   print("YES! We have a match!")
else:
   print("No match")
```

Output

```
YES! We have a match!
```

RegEx Functions

The re module offers couples of functions that allows us to search a string for a match.

Function	Description
findall	Returns a list containing all matches
search	Returns a Match object, if the match found anywhere in the string
split	Returns a list, where the string has been split at each math
sub	Replaces one or many matches with a string

Metacharacters

Metacharacters in RegEx are characters with a special meaning.

Characters	Description	Example
[]	A set of characters	“[a-m]”
\	Signals a special sequence, also used to escape special characters	“\d”
.	Any character except newline character	“he..o”
^	Starts with	“^Hello”
$	Ends with	“World$”
*	Zero or more occurences	“aix*”
+	One or more occurrences	“aix+”
{}	Exactly the specified number of occurences	“a|{2}”
|	Either or	“short|long”
()	Capture and group	

Special Sequences

Special sequences in RegEx is a \ followed by one of the characters listed below and has a special meaning -

Character	Description	Example
\A	Returns a match if the specified characters are at the beginning of the string	“\APyt”
\b	Returns a match if the specified characters are at the start or at the end of a word	r”\bPython” r”world\b”
\B	Returns a match if the specified characters are present, but NOT at the start(or at the end) of a word	r”\BPython” r”World\B”
\d	Returns a match if the string contains digits	“\d”
\D	Returns a match if the string DOES NOT contain digits	“\D”
\s	Returns a match where the string contains a white space character	“\s”
\S	Returns a match where the string DOES NOT contain a white space character	“\S”
\w	Returns a match if the string contains any word characters (characters may be letter from a to Z, digits from 0-9, and the underscore _ character	“\w”
\W	Returns a match where the string DOES NOT contain any word characters	“\W”
\Z	Returns a match if the specified characters are at the end of the string	“world\Z”

Sets

A set in RegEx is a set of characters inside a pair of square brackets [] having some special meaning.

Set	Description
[raj]	Returns a match if one of the specified characters (a, r or n) are present
[a-r]	Returns a match for any lower case letter, alphabetically between a and r
[^raj]	Returns a match for any character Except r, a and j
[0123]	Returns a match where any of the spe
[0-9]	Returns a match for any digit between 0 and 9
[0-3][0-8]	Returns a match for any two-digit numbers between 00 and 38
[a-zA-Z]	Returns a match for any character alphabetically between a to z or A to Z
[+]	Return a match for any + character in the string

Example - findall()

The findall() function returns a list containg all matches.

```
#Print a list of all matches (“in”) from a text
import re
txt = "Use of python in Machine Learning"
x = re.findall("in", txt)
print(x)
```

Output

```
['in', 'in', 'in']
```

Above output display list contains all the matches in the order they are found. However, if no match found, an empty list is displayed.

Just change the below line in your above program, “pattern” which is not there in the text or string.

```
x = re.findall("Hello", txt)
```

Output

```
[]
```

Example - search() function

The search() function searches the string and returns a match object if match is found.

However, if there are more than one match, only the first occurrence of the match will be returned.

```
import re
txt = "Python is one of the most popular languages around the world"
```

```
searchObj = re.search("\s", txt)
print("The first white-space character is located in position: ", searchObj.start(
```

Output

```
The first white-space character is located in position: 6
```

However, if no match found then None is returned.

Example - split() function

The split() function in RegEx returns a list where the string has been split at each match -

```
# Split at each white-space character
import re
string = "Python is one of the most popular languages around the world"
searchObj = re.split("\s", string)
print(searchObj)
```

Result

```
['Python', 'is', 'one', 'of', 'the', 'most', 'popular', 'languages', 'around', 't
```

Example - sub() function

The sub() function in RegEx is to replace the match with the text of your choice.

```
#Replace every white-space in the string with _:
import re
string = "Python is one of the most popular language around the world"
searchObj = re.sub("\s", "_", string)
print(searchObj)
```

Result

```
Python_is_one_of_the_most_popular_language_around_the_world
```

Match Object

A match object in RegEx is an object containing information about the search and the result. In no match found, None is returned.

Example - Search a string and returned match object.

```
import re
string = "Python is one of the most popular language around the world"
searchObj = re.search("on", string)
print(searchObj)
```

Result

```
<_sre.SRE_Match object; span=(4, 6), match='on'>
```

The match object has properties and methods used to retrieve information about the search, and the Result.

- **.span()** – returns a tuple containing the start and end position of the match found.
- **.string** – returns the string passed into the function.
- **.group()** – returns the part of the string where there was a match.

Example - Print the part of the string where there was a match.

```
#Looks for any words that starts with the an upper case “P”:
import re
string = "Python is one of the most popular language around the world"
searchObj = re.search(r"\bP\w+", string)
print(searchObj)
```

Result

```
<_sre.SRE_Match object; span=(0, 6), match='Python'>
```