**RegEx in Python** – A Brief Research Paper by Ali A Nathani – Data Science & Application Course – Metro College of Technology – Instructor: Vijay Kumar

RegEx, or Regular Expression is a set of functions / programming language found in the ‘re’ module or package in python. It is a function that uses a sequence of characters to form a search pattern and can be used to search for the pattern within a string.  
  
The RegEx functions are called by first importing the ‘re’ package by using the following code:

import re

Within this package, the following table of functions1 are used to search for a string match and are considered as RegEx Functions1:

|  |  |
| --- | --- |
| **Function** | **Description** |
| findall | Returns a list containing all matches |
| search | Returns a Match object if there is a match anywhere in the string |
| split | Returns a list where the string has been split at each match |
| sub | Replaces one or many matches with a string |

The RegEx functions use Metacharacters, Special Sequences and Sets in order to assist with searching within strings .

The following characters are Metacharacters1:

|  |  |  |
| --- | --- | --- |
| **Character** | **Description** | **Example** |
| [] | A set of characters | "[a-m]" |
| \ | Signals a special sequence (can also be used to escape special characters) | "\d" |
| . | Any character (except newline character) | "he..o" |
| ^ | Starts with | "^hello" |
| $ | Ends with | "world$" |
| \* | Zero or more occurrences | "aix\*" |
| + | One or more occurrences | "aix+" |
| {} | Exactly the specified number of occurrences | "al{2}" |
| | | Either or | "falls|stays" |
| () | Capture and group |  |

The following are Special Sequences1:

|  |  |  |
| --- | --- | --- |
| **Character** | **Description** | **Example** |
| \A | Returns a match if the specified characters are at the beginning of the string | "\AThe" |
| \b | Returns a match where the specified characters are at the beginning or at the end of a word | r"\bain" r"ain\b" |
| \B | Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word | r"\Bain" r"ain\B" |
| \d | Returns a match where the string contains digits (numbers from 0-9) | "\d" |
| \D | Returns a match where 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 where the string contains any word characters (characters 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 | "Spain\Z" |

**The findall() Function1**

The findall() function returns a list containing all matches.  
Example

Print a list of all matches:

import re

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

print(x)

**The search() Function1**

The search() function searches the string for a match, and returns a Match object if there is a match.

If there is more than one match, only the first occurrence of the match will be returned:

Example

Search for the first white-space character in the string:

import re

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

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

If no matches are found, the value None is returned:

Example

Make a search that returns no match:

import re

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

print(x)

**The split() Function1**

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

Example

Split at each white-space character:

import re

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

print(x)

You can control the number of occurrences by specifying the maxsplit parameter:

Example

Split the string only at the first occurrence:

import re

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

**The sub() Function1**

The sub() function replaces the matches with the text of your choice:

Example

Replace every white-space character with the number 9:

import re

txt = "The rain in Spain"  
x = re.sub("\s", "9", txt)

print(x)  
The output of the above is: The9rain9in9Spain

You can control the number of replacements by specifying the count parameter:

Example

Replace the first 2 occurrences:

import re

txt = "The rain in Spain"  
x = re.sub("\s", "9", txt, 2)

print(x)  
The output of the above is: The9rain9in Spain

**Match Object1**

A Match Object is an object containing information about the search and the result.

Note: If there is no match, the value None will be returned, instead of the Match Object.

Example

Do a search that will return a Match Object:

import re

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

print(x) #this will print an object  
The output of the above is : <re.Match object; span=(5, 7), match='ai'>

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 positions of the match.  
.string returns the string passed into the function  
.group() returns the part of the string where there was a match

Example

Print the position (start- and end-position) of the first match occurrence.  
The regular expression looks for any words that starts with an upper case "S":

import re

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

print(x.span())  
The output of the above is: (12, 17)

Example

Print the string passed into the function:

import re

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

print(x.string)

The output of the above is: The rain in Spain

Example  
Print the part of the string where there was a match.  
The regular expression looks for any words that starts with an upper case "S":

import re

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

print(x.group())

The output of the above is: Spain

Note: If there is no match, the value None will be returned, instead of the Match Object.

**The following example uses a RegEx function to extract an email address from a string:**

import re

def findemail(str):  
 match = re.search(r'[\w\.-]+@[\w\.-]+', str)  
 print(match.group(0))  
 return;

findemail("Python teacher's email is VijayKumar13@gmail.com, You can email me anytime.")  
findemail("sasteacher@yahoo.com is for SAS Teacher")

The output of the above program is as follows:

VijayKumar13@gmail.com   
[sasteacher@yahoo.com](mailto:sasteacher@yahoo.com)

**Other commonly used RegEx functions or phrases:**

RegEx is also used with other functions such as re.compile() and re.match() along with a combination of Metacharacters, backslashes and parenthesis within the brackets. These combinations vary, depending on requirements of each search.

It is best to have a goal or objective in mind when studying or researching the RegEx function and combinations that should be used whether to search and replace, group or modify strings (eg. Split apart).