Regular Expression

Introduction

- A RegEx, or Regular expressions is a sequence of characters that forms a search pattern.
- It can be used to check if a string contains the specified search pattern or not.
- Python provides a built in module **re** which can be used to work with regular expression. import re
- match=re.method_name(pattern, string)
- If the search is successful, search() returns a match object or None object otherwise.

The re module offers a set 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 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 findall() Function

```
import re

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

Output:

['ai', 'ai']

Note: Return an empty list if no match was found

The search() Function

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:

```
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

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

The split() Function

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

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

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

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

```
import re
#Split the string only at the first occurrence:
txt = "The rain in Spain"
x = re.split("\s", txt, 1)
print(x)
```

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

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

```
import re
#Replace every white-space character with the number 9
txt = "The rain in Spain"
x = re.sub("\s", "9", txt)
print(x)
```

Output: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)
```

Metacharacters

- The real power of regular expression matching in Python emerges when regular expression contains special characters called metacharacters.
- These have a unique meaning to the regular expression matching engine and vastly enhance the capability of the search.
- In a regular expression, a set of characters specified in square brackets ([]) makes up a **character class**. This metacharacter sequence matches any single character that is in the class.
- r=re.search('[0-9][0-9][0-9]', ‘upes123python')
- print(r)

Meta Characters in regular expression

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	"planet$"	
*	Zero or more occurrences	"he.*o"	
+	One or more occurrences	"he.+o"	
?	Zero or one occurrences	"he.?o"	
{}	Exactly the specified number of occurrences	"he.{2}o"	
|	Either or	"falls|stays"	
()	Capture and group		

```
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']

import re
txt = "That will be 59 dollars"
#Find all digit characters:
x = re.findall("\d", txt)
print(x)
Output: ['5', '9']
```

```
import re

txt = "hello planet"

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

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

Output: ['hello']

Special Sequences

A special sequence is a \ followed by one of the characters in the list below, and has a special meaning:

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	

\B Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word

\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"

```
import re
txt = "The rain in Spain"
#Check if the string starts with "The":
x = re.findall("\AThe", txt)
print(x)
Output: ['The']
import re

txt = "The rain in Spain"
#Check if "ain" is present at the end of a WORD:
x = re.findall("ain\b", txt)
print(x)
Output:['ain', 'ain']
```

```
import re
txt = "The rain in Spain"
#Check if the string contains any digits (numbers from 0-9):
x = re.findall("\d", txt)
print(x)
Output:[]

import re
txt = "The rain in Spain0"
#Return a match at every no-digit character:
x = re.findall("\D", txt)
print(x)
Output:['T', 'h', 'e', ' ', 'r', 'a', 'i', 'n', ' ', 'i', 'n', ' ', 'S', 'p', 'a', 'i', 'n']
```

Set	Description	
[arn]	Returns a match where one of the specified characters (a, r, or n) are present	
[a-n]	Returns a match for any lower case character, alphabetically between a and n	
[^arn]	Returns a match for any character EXCEPT a, r, and n	
[0123]	Returns a match where any of the specified digits (0, 1, 2, or 3) are present	
[0-9]	Returns a match for any digit between 0 and 9	
[0-5][0-9]	Returns a match for any two-digit numbers from 00 and 59	
[a-zA-Z]	Returns a match for any character alphabetically between a and z, lower case OR upper case	
[+]	In sets, +, *, ., |, (), $,{} has no special meaning, so [+] means: return a match for any + character in the string	

```
import re
txt = "The rain in Spain"
#Check if the string has any a, r, or n characters:
x = re.findall("[arn]", txt)
print(x)
Output:['r', 'a', 'n', 'n', 'a', 'n']

import re
txt = "8 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)
Output:['11', '45']
```

Program exercise

```
r=re.search('1.3','upes1A3python')

#To check whether a given string is starting with ‘He’ or not
result=re.search(‘^He’,str)

#To search a word at the ending of the string
result=re.search(’world$’,str)
r=re.search('[0-9]', '12345foo')
r=re.search('[^0-9]', '12345foo')
```