Regular Expression

A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern.

The module **re** provides full support for Perl-like regular expressions in Python. The re module raises the exception re.error if an error occurs while compiling or using a regular expression.

Characters Sequence

Characters are characters with a special meaning.

Character	Description
[]	A set of characters
\	Signals a special sequence (can also be used to escape special characters)
.	Any character (except newline character)
^	Starts with
$	Ends with
*	Zero or more occurrences
+	One or more occurrences
{}	Excactly the specified number of occurrences
|	Either or
()	Capture and group

Special Sequences

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

Character	Description
\A	Returns a match if the specified characters are at the beginning of the string
\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	Returns a match where the string DOES NOT contain digits
\s	Returns a match where the string contains a white space character
\S	Returns a match where the string DOES NOT contain a white space character
\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	Returns a match where the string DOES NOT contain any word characters
\Z	Returns a match if the specified characters are at the end of the string

Sets

A set is a set of characters inside a pair of square brackets [] with a special meaning.

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

Match Object

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

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	**Output**

import re
str = "This is python programming"
x = re.search("python", str)
print(x.string)
print(x.span())
print(x.group())
	This is python programming
	(8, 14)
	python

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.

Example	**Output**

import re
str = "This is python Programming"
x = re.search("python", str)
	python

print(x.group())	

split Function

The split() function returns a list where the string has been split at each match. we can control the number of occurrences by specifying the maxsplit parameter.

Example	Output

import re
str = "This is python programming"
x = re.split("\s", str)
print(x)
x = re.split("\s", str,1)
print(x)
	['This', 'is', 'python', 'programming']
	['This', 'is python programming']

sub Function

The sub() function replaces the matches with the text of your choice. we can control the number of replacements by specifying the count parameter.

Example	Output

import re
str = "This is python programming"
x = re.sub("is", "IS", str)
print(x)
	ThIS IS python programming
	ThIS is python programming

x = re.sub("is", "IS", str,1)
print(x)	

findall Function

The findall() function returns a list containing all matches. The list contains the matches in the order they are found. If no matches are found, an empty list is returned.

Example	Output

import re
str = "This is python programming"
x = re.findall("i", str)
print(x)
x = re.findall("x", str)
print(x)
	['i', 'i', 'i']
	[]