Regular_Expression Created By github.com/farzadafi

Description

- The term Regex stands for Regular expression.
- The regex or regexp or regular expression is a sequence of different characters which describe the particular search pattern.
- It is also referred/called as a Rational expression.
- It is mainly used for searching and manipulating text strings
- In simple words, you can easily search the pattern and replace them with the matching pattern with the help of regular expression.
- This concept or tool is used in almost all the programming or scripting languages
- The pattern defined by the regular expression is applied to the given string or a text from left to right.
- https://regex101.com/ — for test regex and practice
- https://regexone.com/lesson/ — learn regex with interactive way

Regex Characters

There are following different type of characters of a regular expression:

1. Metacharacters
2. Quantifier
3. Groups and Ranges
4. Escape Characters or character classes

1 Metacharacters

Metacharacters	Description	Example
1 ^	This character is used to match an expression to its right at the start of a string.	^a is an expression match to the string which starts with 'a' such as "aab", "a9c", "apr", "aaaaab", etc. "(T|t)he" => The car is parked in the garage. "^(T|t)he" => The car is parked in the garage.
2 $	The $sign is used to match an expression to its left at the end of a string.	r$ is an expression match to a string which ends with r such as "aaabr", "ar", "r", "aannn9r", etc. "(at\.)" => The fat cat. sat. on the mat. "(at\.)$" => The fat cat. sat. on the mat.
3 .	This character is used to match any single character in a string except the line terminator, i.e. /n.	b.x is an expression that match strings such as "bax", "b9x", "bar". ".ar" => The car parked in the garage.
4 |	It is used to match a particular character or a group of characters on either side. If the character on the left side is matched, then the right side's character is ignored.	A|b is an expression which gives various strings, but each string contains either a or b. farzad|farhad|fardad
5 \	It is used to escape a special character after this sign in a string.	^a.com == abcom, accom, adcom :(^a\.com == a.com just :) ^tar^zip != tar^zip :(^tar\^zip == tar^zip :) "(f|c|m)at\.?" => The fat cat sat on the mat.
6 A	It is used to match the character 'A' in the string.	This expression matches those strings in which at least one-time A is present. Such strings are "Amcx", "mnAr", "mnopAx4". "the" => The fat cat sat on the mat. "The" => The fat cat sat on the mat.
7 Ab	It is used to match the substring 'Ab' in the string.	This expression matches those strings in which 'Ab' is present at least one time. Such strings are "Abcx", "mnAb", "mnopAbx4".

2 Quantifiers

Characters	Description	Example
1 +	This character specifies an expression to its left for one or more times.	s+ is an expression which gives "s", "ss", "sss", and so on. s+regex = sregex, ssregex, sssregex moham+ad = mohamad, mohammad "c.+t" => The fat cat sat on the mat.
2 ?	This character specifies an expression to its left for 0 (Zero) or 1 (one)times.	aS? is an expression which gives either "a" or "as", but not "ass". afshari? = afshar, afshari "[T]?he" => The car is parked in the garage.
3 *	This character specifies an expression to its left for 0 or more times	Br* is an expression which gives "B", "Br", "Brr", "Brrr", and so on...
4 {x}	It specifies an expression to its left for only x times.	Mab{5} is an expression which gives the following string which contains 5 b's: "Mabbbbb" moham{2}ad = mohammad "[0-9]{3}" => The number was 9.9997 but we rounded it off to 10.0.
5 {x, }	It specifies an expression to its left for x or more times.	Xb{3, } is an expression which gives various strings containing at least 3 b's. Such strings are "Xbbb", "Xbbbb", and so on.
6 {x,y}	It specifies an expression to its left, at least x times but less than y times.	Pr{3,6} a is an expression which provides two strings. Both strings are as follows: "Prrrr" and "Prrrrr" "[0-9]{2,3}" => The number was 9.9997 but we rounded it off to 10.0.

3 Groups and Ranges

Characters	Description	Example
( )	It is used to match everything which is in the simple bracket.	A(xy) is an expression which matches with the following string: "Axy" far(zad|had|dad) == farzad == farhad == fardad "(c|g|p)ar" => The car is parked in the garage. "(T|t)he|car" => The car is parked in the garage.
{ }	It is used to match a particular number of occurrences defined in the curly bracket for its left string.	xz{4,6} is an expression which matches with the following string: "xzzzzz"
[ ]	It is used to match any character from a range of characters defined in the square bracket.	xz[atp]r is an expression which matches with the following strings: "xzar", "xztr", and "xzpr" "[Tt]he" => The car parked in the garage. "ar[.]" => A garage is a good place to park a car.
[pqr]	It matches p, q, or r individually.	Following strings are matched with this expression: "p", "q", and "r".
[pqr][xy]	It matches p, q, or r, followed by either x or y.	Following strings are matched with this expression: "px", "qx", and "rx", "py", "qy", and "ry".
[^....]	It matches a character which is not defined in the square bracket.	"[^c]ar" => The car parked in the garage.
[a-z]	It matches letters of a small case from a to z.	This expression matches the strings such as: "a", "python", "good". "[a-z]*" => The car parked in the garage
[A-Z]	It matches letters of an upper case from A to Z.	This expression matches the strings such as: "EXCELLENT", "NATURE".
^[a-zA-Z]	It is used to match the string, which is either starts with a small case or upper-case letter.	This expression matches the strings such as: "A854xb", "pv4fv", "cdux".
[0-9]	It matches a digit from 0 to 9.	This expression matches the strings such as: "9845", "54455"
[aeiou]	This square bracket only matches the small case vowels.	-
[AEIOU]	This square bracket only matches the upper-case vowels.	-
ab[^4-9]	It matches those digits or characters which are not defined in the square bracket.	This expression matches those strings which do not contain 5, 6, 7, and 8.

4 Escape Characters or Character Classes

Characters	Description	Example
\s	It is used to match a one white space character.	hello\sworld == hello world != helloworld "\s*cat\s*" => The fat cat sat on the concatenation.
\S	It is used to match one non-white space character.	^\S+$ == farzad_afi != farzad afi != " farzad_afi"
\d	It is used to match one decimal digit, which means from 0 to 9.	\d == 45345 != farzad
\D	It is used to match any non-decimal digit.	\D == farzad != 5454
\n	It helps a user to match a new line.	\n = 45345(new line here ->) farzad
\w	It is used to match the alphanumeric [0-9a-zA-Z] characters.	\w == 45345 == farzad != #$%$#
\W	It is used to match one non-word character	
\b	It is used to match a word boundary.	\b = 45345 farzad $%$#%asdklfj$$$%234$ 4 matches :)