RegEx - Regular Expressions

- Regular expression is termed as a rational expression.
- It is a combination of characters to outline a pattern.
- The pattern can be used with strings for "find and replace" operations.
- The character present in a regular expression is either a
  - Meta character – with special meaning, or
  - Regular character – with literal meaning.
  - Both meta and regular characters helps to identify pattern string.
  - Pattern-matches can either be exact or quite comparable.

RegEx Theories:

Elements list and/or members are used to specify a finite set of strings by the way of theories defined below:

- Boolean "or": Vertical bar splits options as A|a.
- Grouping: Parentheses outlines scope and operator precedence as C(A|a)P which means CAP or CaP.
- Quantification: Quantifier following a single or group of characters informs the occurrence count of it preceding element.

RegEx Symbols and Meanings:

S.No	Symbol	Meaning
1.	*	Zero or more occurrence
2.	?	Zero or one occurrence
3.	+	One or more occurrence
4.	n	Exactly n occurences
5.	min	Occurs min or more times
6.	min, max	Occurs between min and max times
7.	.	Single character
8.	[…]	A list of characters
9.	[^…]	Any single character that is not in the list that follows ^.
10.	\d	Any single digit
11.	\s	Any single whitespace character
12.	\S	Any non-whitespace character
13.	\w	Any single "word" character. Equal to [a-zA-Z0-9_]
14.	\W	Matches any non-word character
15.	Circumflex (^)	Present at the beginning of a pattern
16.	$	Present at the end of a pattern
17.	\b	Word boundary

18.	|	Separates two or more alternatives
19.	(...)	Mention the order of Execution
20.	\t	tab
21.	\r	carriage return
22.	\R	Any single newline of any type
23.	\n	Linefeed
24.	\p{xx}	A Unicode character with xx property
25.	\P{xx}	Any Unicode character without xx property
26.	\X	Any number of Unicode characters

RegEx Samples:

- a|b* expands as {ε, "a", "b", "bb", "bbb", ...}
- (a|b)* expands as {ε, "a", "b", "aa", "ab", "ba", "bb", "aaa", ...}
- ab*(c|ε) expands as {"a", "ac", "ab", "abc", "abb", "abbc", ...}
- (0|(1(01*0)*1))* expands in multiples of 3 as { ε, "0", "00", "11", "000", "011", "110", "0000", "0011", "0110", "1001", "1100", "1111", "00000", ... }
- .at expands as "hat", "cat", and "bat".
- [hc] at expands as "hat" and "cat".

- [^b] at expands as strings matched by .at except "bat".
- [^hc] at expands as strings matched by .at other than "hat" and "cat".
- ^[hc] at expands as "hat" and "cat", but only at the beginning of the string or line.
- [hc] at$ expands as "hat" and "cat", but only at the end of the string or line.
- \[.\] expands as any single character surrounded by "[" and "]" since the brackets are escaped, for example: "[a]" and "[b]".
- s.* expands as s followed by zero or more characters, "s", "saw", "seed".
- [hc]+at expands as "hat", "cat", "hhat", "chat", "hcat", "cchchat", but not "at".
- [hc]?at expands as "hat", "cat" and "at".
- [hc]*at expands as "hat", "cat", "hhat", "chat", "hcat", "cchchat", "at”.
- cat|dog expands as "cat" or "dog".