Regular Expression Quick Reference

- All regular expression pattern are written within the syntax **‘ / / ’.** This is sometimes called as the regular expression literal.
- Following is a table explaining the expression patterns:

Pattern	**Meaning**
^	Matches beginning of the string
$	Matches end of the string
*	Repeat zero or more times
?	Repeat zero or 1 time
+	Repeat one or more times
{n}	Repeat exactly n times
{n,}	Repeat atleast n times
{n,m}	Repeats atleast n but at most m times
.	Matches any character
[chars]	Matches any characters within []
[^chars]	Does not match the characters giving within []
[char1-charN]	Matches characters within range 1-N
[^char1-charN]	Does not match characters within range 1-N
[a-z]	Matches only lower case characters
[A-Z]	Matches all upper case characters
[a-zA-Z]	Matches letters
[0-9] or \d	Matches all digits
\D	Matches non-digit characters
[0-9a-zA-Z] or \w	Matches all alphanumeric characters
\W	Matches non-word characters like %
\s	Matches white spaces
\S	Matches non-white spaces
\b	Presence for word boundaries i.e. checks for patterns starting or ending the string but not in the middle of the string
\B	Absence of word boundaries
Pattern1 | Pattern2	Matches either pattern 1 or pattern2
(pattern)	Matches as a single group and allows back referencing
\n	Back reference to group n. eg: /(\s)\1/ matches consecutive occurrences of white spaces
(?pattern)	Treats as a single group not allowing back referencing

- **Escape sequence**: Some characters needs to be prepended with an escape character in order to get its presence in the literal sense. In order to escape any character we use backslash ‘\’ character. Some escape sequences are as follows:
  \/, \\, \., *, \+, \?, \|, \(. \), \{, \}, \^, \$, \n, \r, \t

- **Some useful examples**:

Expression	Explanation
/^es/	Matches strings like “**est**ablished” but not “p**es**ticidies”
/ng$	Matches string like “danci**ng**” but not “e**ng**ineers”
/^abc$/	Matches exactly the string “abc”
/chars/	Matches first occurrence of the characters in string
/ab*/	Matches a followed by zero or more b’s
/ab+/	Matches a followed by one or more b’s
/ab?/	Matches a followed by zero or one b
/a?b+$/	Matches zero or one a followed by one or more b
/ab{2}/	Matches a followed by exactly 2 b’s
/ab{2,}/	Matches a followed by atleast 2 b’s
/ab{3,5}/	Matches a followed by minimum 3 but maximum 5 b’s
/a(bc)*/	Matches a string that has an a followed by zero or more copies of the sequence "bc"
/a(bc){1,3}/	Matches a followed by one through three copies of "bc."
/hey|hi/	Matches a string that has either "hey" or "hi" in it
/(b|cd)ef/	Matches a string that has either "bef" or "cdef"
/(a|b)*c/	Matches a string that has a sequence of alternating a's and b's ending in a c
/a.[0-9]/	Matches a string that has an a followed by one character and a digit
/^.{3}$/	Matches a string with exactly 3 characters
/[ab]/	matches a string that has either an a or b
/[a-d]/	Matches a string that has lowercase letters 'a' through 'd'
/^[a-zA-Z]/	Matches a string that starts with a letter
/[0-9]%/	Matches a string that has a single digit before a percent sign
/,[a-zA-Z0- 9]$/	Matches a string that ends in a comma followed by an alphanumeric character

Reference for examples: http://www.icewarp.com/support/online_help/203030104.htm