

What are Regular Expressions?

A Regular Expression is a text string that describes patterns to find a text or positions of a text within a body of text. For example, 'cool' and '[a-z]+'. The first instance would locate 'cool' and '[a-z]+' would find the first instance of the lower case alphabet, and then end once a digit or whitespace has been encountered. Typically, Regex is to accomplish four main tasks: to find text inside a large body of text; validate that a string follows the format; replace text at positions of your choosing, and to split strings. While using Regular Expressions, the user must be mindful of what they are searching for.

Characters in Regular Expressions

- Literal Characters
 - The most basic Regular Expression. Consists of single character. For example: 'a', 'boy', '92'
- Non-Printable Characters
 - Characters used within text that are not shown to a user. For example: \t to match a tab character, \n for line feed
- Character Classes and Sets
 - Character Classes are used when a user wants to find a range of items. For example: [0-9] would find any numeric character. It is also important to mention that there are two alphabets, the lower case and the upper case.
- Shorthand Character Classes
 - \d – matches a single character that is a digit
 - \w – matches with alphanumeric characters Ex: [a-zA-Z0-9]
 - \s – matches a whitespace character (includes tabs and line breaks)
- The Dot
 - The dot matches a single character, except line break characters but, most applications have a 'single line' mode that makes the dot match any single character, even line breaks.
- Metacharacters
 - There are twelve characters that have special meaning: '\', '^', '.', '|', '?', '*', '+', '(', ')', '[', '{'. When used alone, most cause errors.