Регулярни изрази

Регулярен израз – последователност от символи, които формират някаква шаблон, по който се търси

[character\_group] – matches any single character in character\_group

[^] - negation

[first-last] – character range: matches any single character in the range from first to last (according to the ASCII table)

\w – matches any word character (a-z, A-Z, 0-9, \_)

\W - matches any non-word character – the opposite to \w

\s – matches any whitespace character (\r\n, \t, ‘ ‘)

\S – matches any non-whitespace character

\d – matches any digit character

\D – matches any non-digit character

\* - matches the previous element zero or more times (**as many times as possible**)

+ - matches the previous element one or more times (**as many times as possible**)

? - matches the previous element zero or one time

+? – matches between one and unlimited times (**as few times as possible) –** lazy

\*? – matches between zero and unlimited times (**as few times as possible) –** lazy

{n} - matches the previous element exactly n times

{n,} - matches the previous element at least n times

{n,m}- matches the previous element at least n times, but no more than m times

^ - the match must start at the beginning of the string or line

$ - the match must occur at the end of the string or before \n

\b – the match must occur on a boundary between a \w(alphanumeric) and a \W(non-alphanumeric) character

\B – the match must not occur on a boundary between a \w(alphanumeric) and a \W(non-alphanumeric) character

(subexpression) – captures the matched subexpression and assigns it a number

(?<name>subexpression) – captures the matched subexpression into a named group

(?:subexpression) – defines a non-capturing group

. – matches any character except new line

var regex = new Regex(string pattern);

regex.Match(string text) – returns the first match that corresponds to the pattern

regex.Matches(string text) – returns a collection of matching strings that correspond to the pattern

regex.Replace(string text, string replacement) – replaces all strings that match the pattern with the provided replacement

!! look-behinds are more limited than look-aheads, because they do not support quantifiers of varying size such as “\*”, “?”, “+”

(не работи в C#) The escape sequence \K is similar to a look-behind assertion because it causes any previously-matched characters to be omitted from the final matched string. For example, **foo\Kbar** matches "foobar" but reports that it has matched "bar".