Anchors

^ Beginning of a paragraph

$ End of a paragraph. Line terminating characters are \u000a, \u000b, \u000c, \u000d, \u0085, \u2028, \u2029 and the sequence \u000d \u000a

$ $ on its own matches end of paragraph chars for search & replace of paragraph breaks

\b Word boundary

\B Not a word boundary

\A Beginning of the input

\E Terminates a \Q ... \E quoted sequence

\G The current position is at the end of the previous match

\Q Quotes all following characters until \E

\z End of input

\Z End of input, but before the final line terminator, if one exists

Alternation

a|b Finds either the terms to the left or to the right of the |

Characters

All characters are literal except for * ? + [( ) { } ^ $ | \ .

. Any single character except for a line break or paragraph break

\ Escapes the following character, except for \n, \t, \>, and \<

\n Line break made with Shift-Enter

\t Horizontal Tab \u0009

\d Number or decimal digit

\D Not a decimal digit

\n Line Feed \u000A

\r Carriage Return \u000D.

\R New line character, or the sequence CR LF. The new line characters are \u000a, \u000b, \u000c, \u000d, \u0085, \u2028, \u2029

\s White space character. [\t\n\f\r\p{Z}]

\S Non-white space character

\v New line character \u000a, \u000b, \u000c, \u000d, \u0085, \u2028, \u2029. Does not match the CR LF

\V Non-new line character

\w Word character

\W Non-word character.

Characters-Extended

\a Bell \u0007

\cX Control-X

\e Escape \u001B

\f Form Feed \u000C

\h Horizontal White Space character

\H Non-Horizontal White Space character

\N{CHAR NAME} Named Unicode character

\p{PROP NAME} Any character with the specified Unicode Property

\P{PROP NAME} Any character not having the specified Unicode Property

\uhhhh A four-digit hexadecimal Unicode code

\Uhhhhhhhh An 8-digit hexidecimal Unicode code

\xhh Character with two digit hex value hh

\x{hhhh} Character with hex value hhhh (1 to 6 hex digits)

\X Grapheme Cluster

\0ooo Octal character (1-3 digits). The leading zero is required

Character Sets/Classes

[abc123] Any of the characters

[a-e] Any one the characters that from a to e, inclusive

[a-eh-x] Any one the characters that from a to e, or h-x

[[a-e][h-x]] Any one the characters that from a to e, or h-x

[^a] Any character that is not a

[^a-s] Any character that is not between a and s

\ Escapes the following character in a class. Needed for [ ] \ and sometimes - &

[\u0000-\U0010ffff] Unicode Range

[\p{Letter}] Characters with Unicode Category = Letter

[\P{Letter}] Match everything except Letters.

[\p{numeric_value=9}] Match all numbers with a numeric value of 9. Any Unicode Property may be used.

[\p{Letter}&&\p{script=cyrillic}] Logical AND or intersection. Match the set of all Cyrillic letters

[\p{Letter}--\p{script=latin}] Subtraction. Match all non-Latin letters.

Look-Around

Finds a match to a group without advancing the input position

(?= …) Look-ahead

(?!…) Negative look-ahead

(?<= …) Look-behind assertion, with the last character of the match being the input character just before the current position (no * or + operators)

(?<! …) Negative Look-behind assertion (no * or + operators)

Quantifiers

Finds the preceding char, class, or reference a certain number of times within a paragraph. Unless a quantifier mode is appended, the mode is greedy.

? 0 or 1

* 0 or more

+ 1 or more

{x} Exactly x times

{x,y} From x to y times

{x,} A minimum of x times

Quantifier Modes

Greedy operators match as many times as possible. Lazy operators match as few as possible. Possessive operators eliminate backtracking to find a better match.

? Makes the preceding quantifier lazy, like ?? *? +? {x}? etc.

+ Makes the preceding quantifier possessive like ?+ *+ ++ {x}+ etc.

References and Groups

(…) Chars inside the parentheses are a group available after the match

(?: …) Non-capturing group

(?> …) Atomic-match parentheses. First match of group is the only one tried; if it does not lead to an overall pattern match, back up the search for a match to a position before the "(?>"

(?# …) Free-format comment

(?<name> …) Named capture group. The <angle brackets> are literal - they appear in the pattern.

\k<name> Named Capture Back Reference

\n Back Reference. Match whatever the nth capturing group matched

Replace Syntax

All characters are literal except the following:

& or $0 Entire found string

$x The text from reference x (starting at 1)

\n Paragraph break

Flags

(?i) Case insensitive

(?s) DotAll mode. Dot (.) matches newline characters

(?m) Multiline mode. ^ and $ match the beginning and end of every line

(?w) Use Unicode UAX 29 word boundary definitions

(?x) Allow use of white space and #comments within patterns. To match a space character enter it as [ ]

Multiple flags may be combined. Use - to disable flags.

(?ismwx-ismwx) Change the flag settings. Changes apply to the portion of the pattern following the setting

(?ismwx-ismwx: …) Evaluate the parenthesized expression with the specified flags enabled or disabled