

Operators

Operator	Description
	Alternation. <code>A B</code> matches either <code>A</code> or <code>B</code> .
*	Match 0 or more times. Match as many times as possible.
+	Match 1 or more times. Match as many times as possible.
?	Match zero or one times. Prefer one.
{ <i>n</i> }	Match exactly <i>n</i> times.
{ <i>n</i> , }	Match at least <i>n</i> times. Match as many times as possible.
{ <i>n</i> , <i>m</i> }	Match between <i>n</i> and <i>m</i> times. Match as many times as possible, but not more than <i>m</i> .
*?	Match 0 or more times. Match as few times as possible.
+?	Match 1 or more times. Match as few times as possible.
??	Match zero or one times. Prefer zero.
{ <i>n</i> }?	Match exactly <i>n</i> times.
{ <i>n</i> , }?	Match at least <i>n</i> times, but no more than required for an overall pattern match.
{ <i>n</i> , <i>m</i> }?	Match between <i>n</i> and <i>m</i> times. Match as few times as possible, but not less than <i>n</i> .
*+	Match 0 or more times. Match as many times as possible when first encountered, do not retry with fewer even if overall match fails (Possessive Match).
++	Match 1 or more times. Possessive match.
?+	Match zero or one times. Possessive match.
{ <i>n</i> }+	Match exactly <i>n</i> times.
{ <i>n</i> , }+	Match at least <i>n</i> times. Possessive Match.
{ <i>n</i> , <i>m</i> }+	Match between <i>n</i> and <i>m</i> times. Possessive Match.
(...)	Capturing parentheses. Range of input that matched the parenthesized subexpression is available after the match.
(?: ...)	Non-capturing parentheses. Groups the included pattern, but does not provide capturing of matching text. Somewhat more efficient than capturing parentheses.
(?> ...)	Atomic-match parentheses. First match of the parenthesized subexpression 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 (?# comment).
(?= ...)	Look-ahead assertion. True if the parenthesized pattern matches at the current input position, but does not advance the input position.
(?! ...)	Negative look-ahead assertion. True if the parenthesized pattern does not match at the current input position. Does not advance the input position.
(?<= ...)	Look-behind assertion. True if the parenthesized pattern matches text preceding the current input position, with the last character of the match being the input character just before the current position. Does not alter the input position.
(?<! ...)	Negative Look-behind assertion. True if the parenthesized pattern does not match text preceding the current input position, with the last character of the match being the input character just before the current position. Does not alter the input position.
(?ismx-ismx: ...)	Flag settings. Evaluate the parenthesized expression with the specified flags enabled or -disabled.
(?ismx-ismx)	Flag settings. Change the flag settings. Changes apply to the portion of the pattern following the setting. For example, (?i) changes to a case insensitive match.