Description	Token
Newline	\n
Carriage return	\r
Tab	\t
Null character	\0
A single character of: a, b or c	[abc]
A character except: a, b or c	[^abc]
A character in the range: a-z	[a-z]
A character not in the range: a-z	[^a-z]
A character in the range: a-z or A-Z	[a-zA-Z]
Letters and digits	[[:alnum:]]
Letters	[[:alpha:]]
ASCII codes 0-127	[[:ascii:]]
Space or tab only	[[:blank:]]
Control characters	[[:cntrl:]]
Decimal digits	[[:digit:]]
Visible characters (not space)	[[:graph:]]
Lowercase letters	[[:lower:]]
Visible characters	[[:print:]]
Visible punctuation characters	[[:punct:]]
Whitespace	[[:space:]]
Uppercase letters	[[:upper:]]
Word characters	[[:word:]]
Hexadecimal digits	[[:xdigit:]]
Start of word	[[:<:]]
End of word	[[:>:]]
Any single character	.
Alternate - match either a or b	a|b
Any whitespace character	\s
Any non-whitespace character	\S
Any digit	\d
Any non-digit	\D
Any word character	\w
Any non-word character	\W
Any Unicode sequences, linebreaks included	\X
Match one data unit	\C
Unicode newlines	\R
Match anything but a newline	\N
Vertical whitespace character	\v
Negation of \v	\V
Horizontal whitespace character	\h
Negation of \h	\H
Reset match	\K
Match subpattern number #	\#
Unicode property X	\pX
Unicode property or script category	\p{...}
Negation of \pX	\PX

Negation of \p	\P{...}
Quote; treat as literals	\Q...\E
Match subpattern `name`	\k{name}
Match subpattern `name`	\k<name>
Match subpattern `name`	\k'name'
Match nth subpattern	\gn
Match nth subpattern	\g{n}
Match text the nth relative previous subpattern matched	\g{-n}
Match expression defined in the nth capture group	\g<n>
Match expression defined in the nth relative upcoming capture group.	\g<+n>
Match expression defined in the nth capture group.	\g'n'
Match expression defined in the nth relative upcoming subpattern	\g'+n'
Match previously-named capture group `letter`	\g{letter}
Match expression defined in the capture group named "letter"	\g<letter>
Match expression defined in the named capture group `letter`	\g'letter'
Hex character YY	\xYY
Hex character YYYY	\x{YYYY}
Octal character ddd	\ddd
Control character Y	\cY
Backspace character	[\b]
Makes any character literal	\
Match everything enclosed	(?:...)
Capture everything enclosed	(...)
Atomic group (non-capturing)	(?>...)
Duplicate/reset subpattern group number	(?|...)
Comment group	(?#...)
Named Capturing Group	(?'name'...)
Named Capturing Group	(?<name>...)
Named Capturing Group	(?P<name>...)
Inline modifiers	(?imsxUJnxx)
Localized inline modifiers	(?imsxUJnxx:...)
Conditional statement	(?(1)yes|no)
Conditional statement	(?(R)yes|no)
Recursive Conditional statement	(?(R#)yes|no)
Conditional statement	(?(R&name)yes|no)
Lookahead conditional	(?(?=...)yes|no)
Lookbehind conditional	(?(?<=...)yes|no)
Recurse entire pattern	(?R)
Match expression defined in capture group 1	(?1)
Match expression defined in the first relative capture group	(?+1)
Match expression defined in capture group `name`	(?&name)
Match subpattern `name`	(?P=name)
Match expression defined in the capture group "{name}"	(?P>name)
Pre-define patterns before using them	(?(DEFINE)...)
Positive Lookahead	(?=...)
Negative Lookahead	(?!...)
Positive Lookbehind	(?<=...)

Negative Lookbehind	(?<!...)
Control verb	(*ACCEPT)
Control verb	(*FAIL)
Control verb	(*MARK:NAME)
Control verb	(*COMMIT)
Control verb	(*PRUNE)
Control verb	(*SKIP)
Control verb	(*THEN)
Pattern modifier	(*UTF)
Pattern modifier	(*UTF8)
Pattern modifier	(*UTF16)
Pattern modifier	(*UTF32)
Pattern modifier	(*UCP)
Line break modifier	(*CR)
Line break modifier	(*LF)
Line break modifier	(*CRLF)
Line break modifier	(*ANYCRLF)
Line break modifier	(*ANY)
Empty match modifier	(*NOTEMPTY)
Empty match modifier	(*NOTEMPTY_ATSTART)
JIT Modifier	(*NO_JIT)
Line break modifier	\R
Line break modifier	(*BSR_ANYCRLF)
Line break modifier	(*BSR_UNICODE)
Regex engine modifier	(*LIMIT_MATCH=x)
Regex engine modifier	(*LIMIT_RECURSION=d)
Regex engine modifier	(*NO_AUTO_POSSESS)
Regex engine modifier	(*NO_START_OPT)
Zero or one of a	a?
Zero or more of a	a*
One or more of a	a+
Exactly 3 of a	a{3}
3 or more of a	a{3,}
Between 3 and 6 of a	a{3,6}
Greedy quantifier	a*
Lazy quantifier	a*?
Possessive quantifier	a*+
Start of match	\G
Start of string	^
End of string	$
Start of string	\A
End of string	\Z
Absolute end of string	\z
A word boundary	\b
Non-word boundary	\B
Global	g
Multiline	m

Case insensitive	i
Ignore whitespace / verbose	x
Single line	s
Unicode	u
eXtra	X
Ungreedy	U
Anchor	A
Duplicate group names	J
Contents in capture group 1	$1
Contents in capture group \`foo\`	${foo}
Hexadecimal replacement values	\x20
Hexadecimal replacement values	\x{06fa}
Insert a tab	\t
Insert a carriage return	\r
Insert a newline	\n
Insert a form-feed	\f
Uppercase Transformation	\U
Lowercase Transformation	\L
Terminate any Transformation	\E