

ANCHORS	ASSERTIONS	GROUPS AND RANGES	
^ Start of string, or start of line in multi-line pattern	?= Lookahead assertion	. Any character except new line (\n)	
\A Start of string	?!= Negative lookahead	(a b) a or b	
\$ End of string, or end of line in multi-line pattern	?<= Lookbehind assertion	(...) Group	
\Z End of string	?!= or ?<! Negative lookbehind	(?:...) Passive (non-capturing) group	
\b Word boundary	?> Once-only Subexpression	[abc] Range (a or b or c)	
\B Not word boundary	?() Condition [if then]	[^abc] Not (a or b or c)	
\< Start of word	?() Condition [if then else]	[a-q] Lower case letter from a to q	
\> End of word	?# Comment	[A-Q] Upper case letter from A to Q	
CHARACTER CLASSES	QUANTIFIERS	[0-7] Digit from 0 to 7	
\c Control character	*	\x Group/subpattern number "x"	
\s White space	+ 1 or more	Ranges are inclusive.	
\S Not white space	? 0 or 1	{3}	Exactly 3
\d Digit	{3,}	3 or more	
\D Not digit	{3,5}	3, 4 or 5	
\w Word	Add a ? to a quantifier to make it ungreedy.		
\W Not word	ESCAPE SEQUENCES	PATTERN MODIFIERS	
\x Hexadecimal digit	\ Escape following character	g Global match	
\O Octal digit	\Q Begin literal sequence	i * Case-insensitive	
POSIX	\E End literal sequence	m * Multiple lines	
[:upper:] Upper case letters	"Escaping" is a way of treating characters which have a special meaning in regular expressions literally, rather than as special characters.	s * Treat string as single line	
[:lower:] Lower case letters	COMMON METACHARACTERS	x * Allow comments and whitespace in pattern	
[:alpha:] All letters	^ [. \$	e * Evaluate replacement	
[:alnum:] Digits and letters	{ * (\	U * Ungreedy pattern	
[:digit:] Digits	+) ?	* PCRE modifier	
[:xdigit:] Hexadecimal digits	< >	STRING REPLACEMENT	
[:punct:] Punctuation	The escape character is usually \	\$n nth non-passive group	
[:blank:] Space and tab	SPECIAL CHARACTERS	\$2 "xyz" in /^(abc(xyz))\$/	
[:space:] Blank characters	\n New line	\$1 "xyz" in /^(?:abc)(xyz)\$/	
[:cntrl:] Control characters	\r Carriage return	\$` Before matched string	
[:graph:] Printed characters	\t Tab	\$' After matched string	
[:print:] Printed characters and spaces	\v Vertical tab	\$+ Last matched string	
[:word:] Digits, letters and underscore	\f Form feed	\$& Entire matched string	
	\xxx Octal character xxx	Some regex implementations use \ instead of \$.	
	\xhh Hex character hh		



By Dave Child (DaveChild)
cheatography.com/davechild/
aloneonahill.com

Published 19th October, 2011.
 Last updated 12th March, 2020.
 Page 1 of 1.

Sponsored by [CrosswordCheats.com](http://crosswordcheats.com)
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>