

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	
\d Digit	{3}	
\D Not digit	{3,}	
\w Word	{3,5}	
\W Not word		
\x Hexadecimal digit		
\O Octal digit		
POSIX	ESCAPE SEQUENCES	
[:upper:] Upper case letters	\ Escape following character	PATTERN MODIFIERS
[:lower:] Lower case letters	\Q Begin literal sequence	g Global match
[:alpha:] All letters	\E End literal sequence	i Case-insensitive
[:alnum:] Digits and letters	"Escaping" is a way of treating characters which have a special meaning in regular expressions literally, rather than as special characters.	m Multiple lines
[:digit:] Digits		s Treat string as single line
[:xdigit:] Hexadecimal digits		x Allow comments and whitespace in pattern
[:punct:] Punctuation		e Evaluate replacement
[:blank:] Space and tab		U Ungreedy pattern
[:space:] Blank characters		* PCRE modifier
[:cntrl:] Control characters		
[:graph:] Printed characters		
[:print:] Printed characters and spaces		
[:word:] Digits, letters and underscore		
COMMON METACHARACTERS	STRING REPLACEMENT	
	^ [ . \$	\$n nth non-passive group
	{ * ( \	\$2 "xyz" in /^(abc(xyz))\$/
	+ )   ?	\$1 "xyz" in /^(?:abc)(xyz)\$/
	< >	\$` Before matched string
	The escape character is usually \	\$' After matched string
		\$+ Last matched string
		\$& Entire matched string
SPECIAL CHARACTERS		Some regex implementations use \ instead of \$.
	\n New line	
	\r Carriage return	
	\t Tab	
	\v Vertical tab	
	\f Form feed	
	\xxx Octal character xxx	
	\xhh Hex character hh	



By **Dave Child** (DaveChild)  
[cheatography.com/davechild/aloneonahill.com](https://cheatography.com/davechild/aloneonahill.com)

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

Sponsored by **Readable.com**  
 Measure your website readability!  
<https://readable.com>