

Anchors		Quantifiers		Groups and Ranges	
^	Start of string	*	0 or more	.	Any character except new line (\n)
\A	Start of string	+	1 or more	(a b)	a or b
\$	End of string	?	0 or 1	(...)	Group
\Z	End of string	{3}	Exactly 3	(?:...)	Passive Group
\b	Word boundary	{3,}	3 or more	[abc]	Range (a or b or c)
\B	Not word boundary	{3,5}	3, 4 or 5	[^abc]	Not a or b or c
\<	Start of word	Quantifier Modifiers		[a-q]	Letter between a and q
\>	End of word	"x" below represents a quantifier		[A-Q]	Upper case letter between A and Q
Character Classes		x?		[0-7]	Digit between 0 and 7
\c	Control character	Ungreedy version of "x"		\n	nth group/subpattern
\s	White space	Escape Character		<i>Note: Ranges are inclusive.</i>	
\S	Not white space	\		Pattern Modifiers	
\d	Digit	Escape Character		g	Global match
\D	Not digit	Metacharacters (must be escaped)		i	Case-insensitive
\w	Word	^ [.		m	Multiple lines
\W	Not word	\$ { *		s	Treat string as single line
\x	Hexadecimal digit	(\ +		x	Allow comments and white space in pattern
\O	Octal digit) ?		e	Evaluate replacement
POSIX		< >		U	Ungreedy pattern
[:upper:]	Upper case letters	Special Characters		String Replacement (Backreferences)	
[:lower:]	Lower case letters	\n		\$n	nth non-passive group
[:alpha:]	All letters	\r		\$2	"xyz" in /^(abc(xyz))\$/
[:alnum:]	Digits and letters	\t		\$1	"xyz" in /^(?:abc)(xyz)\$/
[:digit:]	Digits	\v		\$`	Before matched string
[:xdigit:]	Hexadecimal digits	\f		\$'	After matched string
[:punct:]	Punctuation	\xxx		\$+	Last matched string
[:blank:]	Space and tab	\xhh		\$&	Entire matched string
[:space:]	Blank characters	Sample Patterns			
[:cntrl:]	Control characters	<i>Pattern</i>		<i>Will Match</i>	
[:graph:]	Printed characters	([A-Za-z0-9-]+)		Letters, numbers and hyphens	
[:print:]	Printed characters and spaces	(\d{1,2}\d{1,2}\d{4})		Date (e.g. 21/3/2006)	
[:word:]	Digits, letters and underscore	([\s]+(?:=\.(jpg gif png))\.\2)		jpg, gif or png image	
Assertions		(^[1-9]{1}\$ ^[1-4]{1}[0-9]{1}\$ ^50\$)		Any number from 1 to 50 inclusive	
?=	Lookahead assertion	#?([A-Fa-f0-9]){3}([A-Fa-f0-9]){3}?)		Valid hexadecimal colour code	
?!	Negative lookahead	((?=.*\d)(?=. *[a-z])(?=. *[A-Z]).{8,15})		String with at least one upper case letter, one lower case letter, and one digit (useful for passwords).	
?<=	Lookbehind assertion	(\w+@[a-zA-Z_]+?\.[a-zA-Z]{2,6})		Email addresses	
?!= or ?<!	Negative lookbehind	(<(/?[^\>]+)\>)		HTML Tags	
?>	Once-only Subexpression	<i>Note: These patterns are intended for reference purposes and have not been extensively tested. Please use with caution and test thoroughly before use.</i>			
?()	Condition [if then]				
?()	Condition [if then else]				
?#	Comment				