Regular Expressions Quick-Reference/Comparison Chart, v2.0

	vim	BREs	EREs	perlexpr	Ruby
Universal Conventions	**Matches:**				
.	*Any character, except the newline character*[1]				
*	*Zero or more of preceding character*				
^	*Beginning of any line, but not the newline itself*[2,3]				
$	*End of any line, but not the newline itself*[2,3]				
[abc4-9]	*Any character listed*				
[^abc4-9]	*A character not listed*				
Variable Notations	**Is matched by:**				
Stream/File Begin, End	\%^, \%$	—	—	\A, \Z	\A, \Z[8]
0 or 1 of Preceding Atom	\? or \=	—	\?	?	?
1+ of Preceding Atom	\+	\+	+	+	+
Repetition Exactly	\{n}	\{n\}	{n}	{n}	{n}
Repetition Minimum	\{n,}	\{n,\}	{n,}	{n,}	{n,}
Repetition Range	\{n,m}	\{n,m\}	{n,m}	{n,m}	{n,m}
Repetition Maximum	\{,n}	—	—	—	—
Lazy Quantifiers	—	—	—	*?, +?, ??, {}?	*?, +?, ??, {}?[9]
Alternation	|	—	|	|	|
Grouping, Recall n^{th}	\(…\), $\backslash n$	\(…\), $\backslash n$	(…), $\backslash n$	(…), $n	(…), $\backslash n$
Non-Capture Grouping	\%(…\)	—	—	(?:…)	(?:…)
Recall Entire Match	&	—	—	\&	\&
Recall Pre-/Post-Match	—	—	—	—	\`, \'
Word Boundary	\<\>	\<\>, \b	\<\>, \b	\b	\b
Non-Word Boundary	—	\B	\B	\B	\B
Part-of-word[4]	\w	—	—	\w	\w
Non-word[5]	\W	—	—	\W	\W
Whitespace	\s	[[:space:]]	[[:space:]]	\s	\s
Non-whitespace	\S	—	—	\S	\S
Alpha/Numerics	—	[[:alnum:]]	[[:alnum:]]	\{IsAlnum}	[[:alnum:]]
Alphabeticals	\a	[[:alpha:]]	[[:alpha:]]	\{IsAlpha}	[[:alpha:]]
Digits (Octal)	\o	—	—	—	—
Digits (Decimal)	\d	[[:digit:]]	[[:digit:]]	\d	\d
Non-Digit (Decimal)	\D	—	—	\D	\D
Digits (Hex)	\x	[[:xdigit:]]	[[:xdigit:]]	\{IsXDigit}	[[:xdigit:]]
Punctuation	—	[[:punct:]]	[[:punct:]]	\{IsPunct}	[[:punct:]]
Lower-Case	\l	[[:lower:]]	[[:lower:]]	\{IsLower}	[[:lower:]]
Upper-Case	\u	[[:upper:]]	[[:upper:]]	\{IsUpper}	[[:upper:]]
All Derivatives of *"a"*	[[= a =]][6]	[[= a =]]	[[= a =]]	—	—
Tab	\t	—	\t	\t	\t
Newline	\n	\n[7]	\n[7]	\n	\n
Carriage Return	\r	\r	\r	\r	\r
Lookahead Match	\(…\)\@=	—	—	(?=…)	(?=…)
Lookahead Nonmatch	\(…\)\@!	—	—	(?!…)	(?!…)
Lookbehind Match	\(…\)\@<=	—	—	(?<=…)	—
Lookbehind Nonmatch	\(…\)\@<!	—	—	(?<!…)	—
Behaviour Switches	**Is provided by the switch:**				
Replace every match	/g	/g	/g	/g	—
Case-insensitivity	\c	-i or /i	-i or /i	/i	/i
. Matches Newlines	$\backslash . \leftarrow (Atom)$	—	—	/s	/m
$ and ^ Match Internal \n	*(Always True)*	*(Always True)*	*(Always True)*	/m	*(Always True)*

[1] In Perl, this will also match newlines if /s is specified.

[2] In BREs, "^" is only has this meaning at the beginning of the regex, and "$" only at the end.

[3] In Perl, "^" and "$" only match the very beginning or end of the search string unless the /m is specified.

[4] Generally, [0-9a-zA-Z_], but may be locale-dependent.

[5] Generally, [ô-9a-zA-Z_], but may be locale-dependent.

[6] This doesn't match uppercase or several diacritics.

[7] The tools `grep` and `sed` don't pass \n to regexes, so this won't match anything.

[8] \Z can match either the end of a string or a newline preceding it, while \z strictly matches the end of the string.

[9] Ruby's implementation of lazy brackets expressions {0,n}? is not fully lazy. If it can match its atom, it is treated like {1,n}?, even if this blocks the overall match. Only if it cannot match anything in a position will it match a zero-length string there.