

# 10stripe's Perl-Centric Regular Expression Cheat-Sheet

## Basic Metacharacters

.	Match any single character except <code>\n</code> (unless <code>/s</code> )
	OR; <code>(ab cd)</code> matches <code>ab</code> or <code>cd</code>
[abc]	Match one out of a set of characters
[^abc]	Match one character not in set
[a-z]	Match one character from range, often <code>[a-zA-Z]</code>
\	Escape next character, such as <code>\ </code> or <code>\(</code> or <code>\)</code>

## Quantifiers

*	Match 0 or more of previous char/subexpression
+	Match 1 or more of previous char/subexpression
?	Match 0 or 1 of previous char/subexpression
{m,n}	Match m to n (inclusive) of previous char/subex.
{n,}	Match n or more of previous char/subexpression
{n}	Match exactly n of previous char/subexpression
*?, ??	Lazy version of same (works for any quantifier)
*+, ?+	Possessive version (works for any quantifier)

## Specific Characters

\w	Word character (alphanumeric, underscore)
\W	Opposite of \w
\s	Whitespace character (space, tab, etc.)
\S	Opposite of \s
\d	Digit
\D	Opposite of \d
[b]	Backspace (any use of <code>\b</code> in a character set)
\n	Newline
\c	Control character
\f	Form feed
\r	Carriage return
\t	Tab
\v	Vertical tab
\x	Hexadecimal number; <code>\xf0</code> matches hex <code>f0</code>
\0	Octal number; <code>\021</code> matches octal <code>21</code>

## Anchors

^	Start of string (equivalent: <code>\$A</code> unless <code>/m</code> is used)
\$	End of string (equivalent: <code>\$Z</code> unless <code>/m</code> is used)
\b	Word boundary, similar to: <code>(\w\W \W\w)</code>
\B	Anything but a word boundary

## Subexpressions

()	Define a subexpression
\$a	a <sup>th</sup> subexpression (use <b>in or after</b> substitution)
\a	a <sup>th</sup> subexpression (use <b>inside</b> match, eg <code>s/(.)\1/a/</code> )
(?:a)	Non-capturing parentheses (match a)

## Case Conversion

\l	Make next character lowercase
\u	Make next character uppercase
\L	Make entire string (up to <code>\E</code> ) lowercase
\U	Make entire string (up to <code>\E</code> ) uppercase
\E	End <code>\L</code> or <code>\U</code> (so they only apply before <code>\E</code> )
\u\L	Capitalize first char, lowercase rest (sentence)

## Look-around

(?=)	Look-ahead; <code>m/a(?=b)/</code> matches <code>ab</code> , "eats" <code>a</code>
(?<=)	Look-behind; <code>m/(?&lt;=a)b/</code> matches <code>ab</code> , "eats" <code>b</code>
(?!)	Negative look-ahead
(?<!)	Negative look-behind
?(a)b	Conditional; if <code>a</code> then <code>b</code>
?(a)b c	Conditional; if <code>a</code> then <code>b</code> else <code>c</code>

## Modifiers

/i	Case-insensitive
/g	Global match (return array of matches)
/m	Multiline mode ( <code>^</code> and <code>\$</code> match internal <code>\n</code> )
/s	Line-agnostic ( <code>.</code> matches <code>\n</code> )
/x	Allow arbitrary whitespace and comments



[10stripe.com/go/regex](http://10stripe.com/go/regex)