Cheatography

perl regexp Cheat Sheet

by Nikolay Mishin (mishin) via cheatography.com/1008/cs/418/

20. Regular expression special variable

$1, $2, $3	hold the backreferences
$+	holds the last (highest-numbered) backreference
$&	(dollar ampersand) holds the entire regex match
$’	(dollar followed by an apostrophe or single quote) holds the part of the string after (to the right of) the regex matc
$`	(dollar backtick) holds the part of the string before (to the left of) the regex match

Using these variables is not recommended in perl scripts when performance matters, as it causes Perl to slow down all regex matches in your entire perl script.

All these variables are read-only, and persist until the next regex match is attempted.

```
$string = "This is the geek stuff article for perl learner";
$string =~ /the (g.) stuff(.) /;
print "Matched String=>$&\nBefore Match=>$`\nAfter Match=>$'\nLast Paren=>$+\nFirst Paren=>$1\n";
```

Debugging regexp

```
use re 'taint';
# Contents of $match are tainted if $dirty was also tainted.
($match) = ($dirty =~ /^(.*)$/s);
# Allow code interpolation:
use re 'eval';
$pat = '(?{ $var = 1 })'; # embedded code execution
/alpha${pat}omega/; # won't fail unless under -T
# and $pat is tainted
use re 'debug'; # like "perl -Dr"
/^(.*)$/s; # output debugging info during
```

Debugging regexp (cont)

```
# compile time and run time
use re 'debugcolor'; # same as 'debug',
# but with colored output
```

6 Regular Expressions

($var =~ /re/), ($var !~ /re/)	matches / does not match
m/pattern/ig-msoxc	matching pattern
qr/patter-n/imsox	store regex in variable
s/pattern/repla-cement/ig-msoxe	search and replace
Modifiers:	
i case-inse-nsitive	o compile once
g global	x extended
s as single line (. matches \n)	e evaluate replacement
Syntax:	
\	escape
.	any single char
^	start of line
$	end of line
, ?	0 or more times (greedy / nongreedy)
+, +?	1 or more times (greedy / nongreedy)
?, ??	0 or 1 times (greedy / nongreedy)
\b, \B	word boundary (\w - \W) / match except at w.b.
\A	string start (with /m)
\Z	string end (before \n)
\z	absolute string end
\G	continue from previous m//g
[...]	character set
(...)	group, capture to $1, $2
(?:...)	group without capturing

6 Regular Expressions (cont)

{n,m} , {n,m}?	at least n times, at most m times
{n,} , {n,}?	at least n times
{n} , {n}?	exactly n times
|	or
\1, \2	text from nth group ($1, ...)
Escape Sequences:	
\a alarm (beep)	\e escape
\f formfeed	\n newline
\r carriage return	\t tab
\cx control-x	\l lowercase next char
\L lowercase until \E	\U uppercase until \E
\Q diable metachars until \E	\E end case modifications
Character Classes:	
[amy]	'a', 'm', or 'y'
[f-j.-]	range f-j, dot, and dash
[^f-j]	everything except range f-j
\d, \D	digit [0-9] / non-digit
\w, \W	word char [a-zA-Z0-9_] / non-word char
\s, \S	whitepace [\t\n\r\f] / non-space
\C	match a byte
\pP, \PP	match p-named unicode / non-p-named-unicode
\p{...}, \P{...}	match long-named unicode / non-named-unicode
\X	match extended unicode
Posix:	
[:alnum:]	alphanumeric
[:alpha:]	alphabetic
[:ascii:]	any ASCII char
[:blank:]	whitespace [\t]
[:cntrl:]	control characters

By **Nikolay Mishin** (mishin)
cheatography.com/mishin/
mishin.narod.ru

Not published yet.
Last updated 12th May, 2016.

perl regexp Cheat Sheet

by Nikolay Mishin (mishin) via cheatography.com/1008/cs/418/

6 Regular Expressions (cont)

[:digit:]	digits
[:graph:]	alphanum + punctuation
[:lower:]	lowercase chars
[:print:]	alphanum, punct, space
[:punct:]	punctuation
[:space:]	whitespace [\s\ck]
[:upper:]	uppercase chars
[:word:]	alphanum + '_'
[:xdigit:]	hex digit
[:^digit:]	non-digit
Extended Constructs	
(?#text)	comment
(?imxs-imsx:...)	enable or disable option
(?=...), (?!...)	positive / negative look-ahead
(?<=..), (?<!..)	positive / negative look-behind
(?>...)	prohibit backtracking
(?{ code })	embedded code
(??{ code })	dynamic regex
(?(cond)yes|no)	condition corresponding to captured parentheses
(?(cond)yes)	condition corresponding to look-around
Variables	
$&	entire matched string
$`	everything prior to matched string
$'	everything after matched string
$1, $2 ...	n-th captured expression
$+	last parenthesis pattern match
$^N	most recently closed capt.
$^R	result of last (?{...})

6 Regular Expressions (cont)

@-, @+	offsets of starts / ends of groups

http://perldoc.perl.org/perlrequick.html

http://habrahabr.ru/post/17126/

REGEX METACHARS

^	string begin
$	str. end (before \n)
+	one or more
*	zero or more
?	zero or one
{3,7}	repeat in range
()	capture
(?:)	no capture
[]	character class
|	alternation
\b	word boundary
\z	string end

http://perldoc.perl.org/perlre.html

REGEX MODIFIERS

/i	case insens.
/m	line based ^$
/s	. includes \n
/x	ign. wh.space
/g	global
\Q	quote (disable) pattern metacharacters till \E
\E	end either case modification or quoted section, think vi

REGEX CHARCLASSES

.	[^\n]
\s	[\x20\f\t\r\n]
\w	[A-Za-z0-9_]
\d	[0-9]
\S, \W and \D	negate

By **Nikolay Mishin** (mishin)
cheatography.com/mishin/
mishin.narod.ru

Not published yet.
Last updated 12th May, 2016.