

About Perlcheat

AUTHOR	Juere Waalboer, with the help of many Perl Monks.
the original PM post	http://perlmonks.org/?node_id=216602
Damian Conway's Perl 6 version	http://perlmonks.org/?node_id=238031
home of the Perl Cheat Sheet	http://juerd.nl/site.plp/perlcheat

CONTEXTS

void
 scalar
 list

SIGILS

\$scalar
 @array
 %hash
 &sub
 *glob

ARRAYS, HASHES

	ARRAYS	HASHES
whole:	@array	%hash
slice:	@array[0, 2]	@hash{'a', 'b'}
element:	\$array[0]	\$hash{'a'}

SCALAR VALUES

number, string, reference, glob, undef

SYNTAX

for (LIST) {}, for (a,b;c) {}
 while () {}, until () {}
 if () {} elsif () {} else {}
 unless () {} elsif () {} else {}
 for equals foreach (ALWAYS)

LINKS

Search the CPAN - metacpan.org	https://metacpan.org
Perl Weekly	http://perlweekly.com/latest.html
PerlMonks - The Monastery Gates	http://www.perlmonks.org/
What's happening in the world of Perl programming, including Perl 5, Perl 6, the CPAN and Parrot?	http://perlbuzz.com/
The Comprehensive Perl Archive Network - www.cpan.org	http://cpan.org
News and views of the Perl programming language	http://www.perl.com/
Perl programming documentation - perl.doc.perl.org	http://perl.doc.perl.org/

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>

Perl image



REGEX CHARCLASSES

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

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

FUNCTION RETURN LISTS

stat	localtime	caller
0 dev	0 second	0 package
1 ino	1 minute	1 filename
2	2 hour	2 line
mode		

Escape sequences

\t	tab (HT, TAB)
\n	newline (LF, NL)
\r	return (CR)
\f	form feed (FF)
\a	alarm (bell) (BEL)
\e	escape (think troff) (ESC)
\cK	control char (example: VT)
\x{}, \x00	character whose ordinal is the given hexadecimal number
\N{name}	named Unicode character or character sequence
\N{U+263D}	Unicode character (example: FIRST QUARTER MOON)
\o{}, \o00	character whose ordinal is the given octal number
\l	lowercase next char (think vi)
\u	uppercase next char (think vi)
\L	lowercase till \E (think vi)
\U	uppercase till \E (think vi)
\Q	quote (disable) pattern metacharacters till \E
\E	end either case modification or quoted section, think vi

REFERENCES

\	\$\$foo[1] ==
references	\$foo->[1]
\$_%&*	\$\$foo{bar}==
dereference	\$foo->{bar}
{}	\$\$foo[1][2]==
anon. href	\$foo->[1]->[2]
\()	\$\$foo[1][2]==
list of refs	\$foo->[1][2]

SPECIAL VARIABLES

\$_	default variable
\$0	program name
\$/	input separator
\$\	output separator
\$	autoflush
#!	sys/libcall error
\$_@	eval error
\$\$	process ID
\$.	line number
@ARGV	command line args
@INC	include paths
@_	subroutine args
%ENV	environment

NUMBERS vs STRINGS

NUMBERS	STRINGS
=	=

Perl Conferences, YAPCs, Workshops, and Hackathons	http://www.yapc.org/
Perl 6	http://perl6.org/
Perl 6 Tablet 2 - Basic Syntax	http://tablets.perl6.org/tablet-2-basic-syntax.html
Learn Perl - learn.perl.org	http://learn.perl.org/
Perl - Download	http://www.perl.org/get.html
The Perl.org Online Library	http://www.perl.org/books/library.html
Books and More (Videos)	http://www.perl.org/media.html
Padre, the Perl IDE	http://padre.perlide.org/
perlcheat	http://perldoc.perl.org/perlcheat.html
Gabor Szabo Perl Trainer and developer	http://szabgab.com/
Perl, Perl Mongers, Mongers, User Group, Meeting	http://pm.org
Facebook vs Google+ for Perl projects	http://szabgab.com/facebook-vs-google-plus-for-perl-projects.html
Lectures about Perl from Gabor Szabo	http://www.youtube.com/results?search_query=gabor+perl
mod_perl world	http://perl.apache.org
Interpreted Languages: PHP, Perl, Python, Ruby (Sheet One)	http://hyperpolyglot.org/scripting
Moose - A postmodern object system for Perl	http://moose.iinteractive.com/
Perl Training Australia - Perl Tips	http://perltraining.com.au/tips/
Basic Perl (Ynon Perek An Amazing Learning Experience) , super	http://ynonperek.com/courses/basic-perl.html
Perl Questions - Stack Overflow	http://stackoverflow.com/questions/tagged/perl?sort=faq&pagesize=15
Perl - Cheat sheets Collection	http://perl4everything.blogspot.com/2010/12/perl-cheat-sheets-collection.html
complete, working examples of good Perl code	http://perlmeme.org/start_here/index.html
Perl Programming Tutorials	http://www.devshed.com/c/b/Perl/
Video from UniverTV	http://univertv.ru/search?txt=perl&x=0&y=0
FAQ по perl и программированию для web	http://www.cyberguru.ru/web/web-programming/perl-web-programming-faq-page2.html
FAQ PERL Agava	http://hosting.agava.ru/faq/tech/docs/perl.shtml

Modern::Perl

```
Modern::Perl
| enable all of the features of Modern Perl with one
| import
```

3 nlink	3 day	3 subroutine
4 uid	4 month-1	4 hasargs
5 gid	5 year-1900	5 wantarray
6 rdev	6 weekday	6 evaltext
7 size	7 yearday	7 is_require
8 atime	8 is_dst	8 hints
9 mtime		9 bitmask
10 ctime	just use	
11 blksize	POSIX::	3..9 only
12 blksize	strftime! http://docstore.mik.ua/oreilly/perl/prog3/ch32_36.htm	with EXPR

opendir & grep:

```
opendir(DIR,"$SCANDATA");
@files = sort(grep(/[0-9][0-9]-/,readdir(DIR)));
closedir(DIR);
```

hash iteration:

```
while (($key,$val) = each (%hash)) { }
```

getopt

```
use Getopt::Std;
%options=();
getopts("he:f:s:SD:u:F:",\%options);
if ($options{h}) { &usage(); }
```

http://www.perlhowto.com/parsing_command_line_parameters_with_getopt_long
http://www.perlmonks.org/?node_id=455158
<http://aplawrence.com/Unix/perlgetopts.html>
<http://www.cs.mcgill.ca/~abatko/computers/programming/perl/howto/getopts/>
<http://perl.find-info.ru/perl/015/interface/1.htm>

Perl IDE

source <http://www.perlide.org/>

Padre <http://padre.perlide.org/> The Perl Application Development and Refactoring Environment (written in Perl 5)

Eclipse with the EPIC plugin <http://www.epic-ide.org/> or with Perlipse <http://code.google.com/p/perlipse/> using dltk

ActiveState proprietary Komodo IDE http://www.activestate.com/Products/komodo_ide/ or free Komodo Edit http://www.activestate.com/Products/komodo_ide/komodo_edit.mhtml

Kephra <http://kephra.sourceforge.net> Programmers Editor along perl-like Paradigms, written in Perl

Gvim www.vim.org with Perl support http://www.vim.org/scripts/script.php?script_id=556

Emacs www.gnu.org/software/emacs/ with CPerl Mode <http://www.emacswiki.org/cgi-bin/wiki/CPerlMode>

Perl Development Tools <http://www.perlmonks.org/?node=Perl%20Development%20Tools>

== !=	eq ne
< > <= >=	lt gt le ge
<=>	cmp

DO

```
use Modern::Perl;

use strict;

use warnings;

my $var;

open() or die $!;

use Modules;
```

DON'T

```
"$foo"
$$variable_name
`$userinput`
/$userinput/
```

capitalizing:

```
word: $foo = ucfirst lc $foo;
each word: $foo =~ s/(w+)\uL$1/g;
```

referencing a subroutine:

```
$SUBNAME = "junk";
&${SUBNAME}(@PARAMETERS);
```

foreach:

```
foreach $FILE (@FILES) { }
```

OPERATOR PRECEDENCE

```
->
++ --
**
! ~ \u+ u-
=~ !~
* / % x
+ - .
<< >>
named uops
< > <= >= lt gt le ge
== != <=> eq ne cmp
&
|^
&&
||
... ..
?:
= += -= *= etc.
, =>
list ops
not
and
or xor
```

Link to perl cheat

perlcheat

Toolkit

use Toolkit; and put all your favorite modules in a file: > cat \$PERL5LIB/Toolkit/Macros/ALWAYS/Modules use strict; use warnings; use Carp; use Smart::Comments;

use common::sense;

supposed to be the same, with much lower memory usage, as: # use strict qw(vars subs); # use feature qw(say state switch); # no warnings; # use warnings qw(FATAL closed threads internal debugging pack substr malloc # unopened portable prototype inplace io pipe unpack regexp # deprecated exiting glob digit printf utf8 layer # reserved parenthesis taint closure semicolon); # no warnings qw(exec newline);

latest

There is latest, which is no longer the latest. Demonstrates part of the naming problem.

uni::perl

Also, uni::perl which adds enabling unicode part of the mix.

ToolSet

ToolSet offers a subset of Toolkit's abilities, but without source filters.

Moose

I'll include Moose here, since it automatically adds strict and warnings to the calling package.

Acme::Very::Modern::Perl

Using this module, you'll get Modern::Perl, Toolkit, latest, uni::perl, perl5i, and common::sense with just one simple line.

Source

<http://stackoverflow.com/questions/1625839/should-i-use-commonsense-or-just-stick-with-use-strict-and-use-warnings>

Cheatographer



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

Development/Use Tools

Development/Use Tools

perl/cheat

<http://www.cheatography.com/mishin/cheat-sheets/perl/cheat/>

perl-reference-card

<http://www.cheatography.com/mishin/cheat-sheets/perl-reference-card/>

20-killer-perl-programming-tips

<http://www.cheatography.com/mishin/cheat-sheets/20-killer-perl-programming-tips-for-beginners/>

Cheat Sheet

This cheat sheet was published on 1st June, 2012 and was last updated on 2nd July, 2012.

Sponsor

FeedbackFair, increase your conversion rate today!
Try it free!
<http://www.FeedbackFair.com>