

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

News and views of the Perl programming language <http://www.perl.com/>

Perl programming documentation - perldoc.perl.org/

LINKS (cont)

Perl Conferences, YAPCs, Workshops, and Hackathons

Perl 6 <http://perl6.org/>

Perl 6 Tablet 2 - Basic Syntax <http://tablets.perl6.org/tablet-2-basic-syntax.html>

Learn Perl - <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 <http://pm.org>

Mongers,Mongers,User Group,Meeting

Facebook vs Google+ for Perl projects <http://szabgab.com/facebook-vs-google-plus-for-perl-projects.html>



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

Published 1st June, 2012.
Last updated 5th June, 2014.
Page 1 of 5.

Sponsored by **Readability-Score.com**
Measure your website readability!
<https://readability-score.com>

LINKS (cont)

Lections 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

LINKS (cont)

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

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;

Modern::Perl (cont)

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.



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

Published 1st June, 2012.

Last updated 5th June, 2014.

Page 2 of 5.

Sponsored by **Readability-Score.com**

Measure your website readability!

<https://readability-score.com>

Modern::Perl (cont)

Source

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

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	[x20\t\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 mode	2 hour	2 line
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

FUNCTION RETURN LISTS (cont)

9		9
mtime		bitmask
10	just use	
ctime		
11	POSIX::	3..9
blksz		only
12	strftime!	with
blcks	http://docstore.mik.ua/oreilly/p/erl/prog3/ch32_36.htm	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
```



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

Published 1st June, 2012.
 Last updated 5th June, 2014.
 Page 3 of 5.

Sponsored by **Readability-Score.com**
 Measure your website readability!
<https://readability-score.com>

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
Atom	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

Perl IDE (cont)

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

Escape sequences

<code>\t</code>	tab (HT, TAB)
<code>\n</code>	newline (LF, NL)
<code>\r</code>	return (CR)
<code>\f</code>	form feed (FF)
<code>\a</code>	alarm (bell) (BEL)
<code>\e</code>	escape (think troff) (ESC)
<code>\cK</code>	control char (example: VT)
<code>\x{}</code> , <code>\x00</code>	character whose ordinal is the given hexadecimal number
<code>\N{name}</code>	named Unicode character or character sequence
<code>\N{U+263D}</code>	Unicode character (example: FIRST QUARTER MOON)
<code>\o{}</code> , <code>\000</code>	character whose ordinal is the given octal number
<code>\l</code>	lowercase next char (think vi)
<code>\u</code>	uppercase next char (think vi)
<code>\L</code>	lowercase till \E (think vi)
<code>\U</code>	uppercase till \E (think vi)

Escape sequences (cont)

<code>\Q</code>	quote (disable) pattern metacharacters till <code>\E</code>
<code>\E</code>	end either case modification or quoted section, think vi

REFERENCES

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

SPECIAL VARIABLES

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



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

Published 1st June, 2012.
Last updated 5th June, 2014.
Page 4 of 5.

Sponsored by **Readability-Score.com**
Measure your website readability!
<https://readability-score.com>

NUMBERS vs STRINGS

NUMBERS	STRINGS
=	=
.	.
== !=	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+)/\u\L$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

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

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/>



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

Published 1st June, 2012.
Last updated 5th June, 2014.
Page 5 of 5.

Sponsored by **Readability-Score.com**
Measure your website readability!
<https://readability-score.com>