See also: Perl @ Wikipedia perl.org	Perl Guidelines Tools:		rn Perl Best Practices (course) o		
	Learning Perl o: links to O'Reilly Books.	Laurian Dad et Internaciinte Dad et Mantarian Dad et		 Online Perl Interpreter Online PerlTidy option info. 	
• C-c C-h F	Topic • peridoc : abou peritoc : table perisyn : Peri s perifunc : Peri i	of content: names of all pages syntax			
CPAN	CPAN @ Wikipedia The Zen of Comprehensive Archive Networks CPAN Search CPAN — meta::cpan PAUSE - Perl Authors Upload Server		Command line tools interacting cpan : install o cpanplus cpanminus: cpanm: install	on some Linux with: sudo	dnf install perl-CPAN

Perl scripts

Writing Perl scripts		
files.	<pre>use strict; use warnings;</pre>	 The first line of an executable script should be a valid <u>shebang line</u> identifying the appropriate location of the Perl interpreter. Most Perl code should also activate the strict Perl rules and warnings to detect warnings. See: <u>Barewords in Perl</u>
peridiag @ peridoc 1	<pre>use diagnostics;</pre>	 If you want to produce more diagnostics for detected warning or errors then add the use diagnostics; line.
use version/features	<u>use</u> v5.36;	This can be used to enable both the strict and warning pramas as well as several <u>named features</u> . • See the <u>table listing the feature bundles per Perl versions</u> .

Perl 5 Operators

```
Perl has a large number of operators, listed below with their precedence and associativity.

• <u>C Operators missing from Perl</u>: unary &, unary * and (type)

• <u>Quote and Quote-like operators</u>: in Perl quotes are operators and they provide various kind of interpolating and pattern matching capabilities.
Perl 5 Operators
                    Note:
Associativity: one of:
                            left
                                        terms and list operators (leftward)
  right
                             left.
                                        Arrow Operator:
                                        Auto-increment and Auto-decrement: ++ --
                             NA
  NA: not associative:
                             right
                                        Exponentiation:
  cannot use more than
                                        Symbolic Unary Operators:
                            right
                                                                                     ! - -. \ and unary + and -
                                                                                                                                           Note: The operator \ <u>creates a reference</u>. See <u>example</u>.
  one of these
                             left
                                        Binding operators:
                                                                                         !~
  operators in
                             left
                                        Multiplicative Operators:
                                                                                             % x
  sequence.
                             left
                                        Additive Operators:
· CH: chained
                             left
                                        Shift Operators:
                                                                                     <<
                                                                                            >>
                             NA
                                        named unary operators
                            NA
                                        Class instance Operator:
To get this information,
                                                                                     isa
                             CH
                                        Relational Operators:
                                                                                    as numbers: < >
                                                                                                                <= >=
                                                                                                                                as strings: 1t
                                                                                                                                                      gt
                                                                                                                                                              le
                                                                                                                                                                       ge
perldoc perlop
                             CH/NA
                                        Equality Operators:
                                                                                     as numbers: == !=
                                                                                                                                as strings: eq
                                                                                                                                                      ne
                                                                                                                                                              cmp
                             left.
                                        Bitwise And:
                             left
                                        Bitwise Or and Exclusive Or:
                                                                                         1.
                             left
                                        C-style Logical And:
                                                                                    &&
                            left.
                                        Logical Defined-Or:
                                                                                    П
                            NA
                                        Range Operators:
                             right
                                                                                    ?:
                                        Conditional Operator:
                            right
                                        Assignment Operators:
                                                                                                          %=
                                                                                                          x=
                                                                                    goto last next redo dump
                            left
                                        Comma, fat-comma Operators:
                                                                                     =>
                            NA
                                        <u>list operators (rightward)</u>
                                       Logical Not:
                             right
                                        Logical And:
                             left.
                                                                                  and
                            left
                                        Logical or and Exclusive or:
                                                                                  or xor
trick operators
These are not real Per
                                                                                                                                                        -+- is essentially - + - or - - but a + to allow placing them together. The 0+ does the same as -+-, but
                                        Converts a string that starts with digits into a number.
                                                                                                            print -+- '22les poulets!';
                                                                                                            # prints 22
                            0+
operators, but look like
                                                                                                                                                        the second has higher precedence.
operators: they are
                                        Called the 'goatse' operator. It causes the right side
                                                                                                                str = "A 22 before 33 does not make 9, it is 44!";
                             =()=
concatenation of other
                                                                                                            my $digit_count =()= $str =~ /\d/g;
print "$digit_count"; # prints
operators that achieve a
                                        expression to be evaluated in array context. Used to assign
                                        the array/list size to a scalar.
                                                                                                                                                      # prints '7',the number of digits in $str
specific effect.
See the link for others.
                            0{[]}
                                        Useful to interpolate an array inside a string.
                                                                                                            print "these people @{[get_names()]} get promoted"
Understanding these
                                       Note that: "@{[something]}" is the same as join $", something
operators helps
understand Perl. They
should not be used in production code.
                                                                                In scalar context localtime() returns human readable time,
                                                                                                                                                        $ perl -le 'print ~~localtime'
Mon Nov 30 09:06:13 2009
                                        Force scalar context.
                                                                                but in list context it returns a 9-tuple with various date
                                                                                elements.
                             • False in a boolean

    Negation of a true value by "!" or "not"

Truth and falsehood
                                                                                                           So the following scalar values are
                                                                                                                                                         All other scalar values, including the following are
                                                               returns a special false value.
When evaluated as a string it is
                                                                                                            considered false:

undef - the undefined value
                               context:
                                                                                                                                                           1 any non-0 number
' ' the string with a space in it
'00' two or more 0 characters in a string
                                  the number 0,
 Remember that the
                               • the strings '0' and '',
                                                               treated as ", but as a number, it is treated as 0.
                                                                                                            • 0 the number 0, even if you write it
strings '0' and " mean
                                                                                                              as 000 or 0.0
                               • the empty list (),
false. The output of
                                                                                                              " the empty string.
                                                                                                                                                           "0\n" a 0 followed by a newline
glob() may return a file
named '0'!
                                   "undef
                                                                                                            • '0' the string that contains a single
                                                                                                                                                           'true'
                             · All other values are true.
                                                                                                                                                        • 'false' yes, even the string 'false' evaluates to true.
                                                                                                              0 digit.
                                                                                                                                                      (-e $fname && -f _ && -r _ ){
print("$fname exists and is readable\n");
File test operators
                            It is possible to combine the file test operator with the AND operator as in the following example:
The most important
                                        is readable
                                                                                                                                                        is a block special file.
                                        is writable
operators are shown
                                                                                                                                                        is a character special file.
                                        is executable
                                                                                       has nonzero size (returns size in bytes).
                                                                                                                                                        handle is opened to a tty.
They check if the file...
                                        is owned by effective uid.
                                                                                -f
                                                                                       is a plain file.
                                                                                                                                                  -u
                                                                                                                                                        has setuid bit set.
                             -0
                             -R
                                        is readable
                                                                                -d
                                                                                       is a directory.
                                                                                                                                                        has setgid bit set.
                                                                                                                                                  -g
                             -W
                                                                                       is a symbolic link.
                                                                                                                                                        has sticky bit set.
                                        is writable
                                                                                -1
                                                                                                                                                  -k
                                                                                       is a named pipe (FIFO) or Filehandle is a pipe.
                             -X
                                                                                                                                                 -T
                                                                                                                                                        is an ASCII text file (heuristic guess).
                                        is executable
                                                                                -p
-S
```

is a "binary" file (opposite of -T).

is a socket.

-0

file is owned by real uid.

Perl 5 Constants and Variables

Committee Name Content				Peri 5	Constants and Variat	Dies	
d prompts College Col	Perl Variables Names	Scalar Naming Conver	ntions		Array Naming Conventions	All: underscore or letter	of the first character.
Solder (1995) For comment of the property of	all names. ASCII by default, UTF-8 if the	Global variables: Constants:	\$Title_Case \$UPPER_CASE		array names should be plural . • @locals • @Global_Arrays	Constants are UPPEIPackage wide vars areFunctions/methods are	RCASE_WITH_UNDERSCORES re Mixed_Case_With_Underscores are lowercase_with_underscores
Security Control Con	Perl types	Sigil Examples	Meaning			Extra Info	
* discase * Libos and \$1,4,73] Amy data contraining (slabuys)1, 3,340x(1), 4,340x(1)). * discase and the state to secure milips desired state. * Anonomous arrays. * Must are the analysis of complete state. * Anonomous arrays. * What are the analysis of complete state. * Anonomous arrays. * What are the analysis of complete state. * Anonomous arrays. * What are the analysis of complete state. * Anonomous arrays. * What are the analysis of complete state. * Anonomous arrays. * What are the analysis of complete state. * Anonomous arrays. * What are the analysis of complete state. * Anonomous arrays. * What are the analysis of complete state. * Anonomous arrays. * What are the analysis of complete state. * Anonomous arrays. * What are the analysis of complete state. * Anonomous arrays. * Anonomous	Scalar	\$days[28] \$days{'Feb'} \${days} \$Dog::days \$Dog'days \$#days \$days->[28] \$days[0][2] \$d{99}{'Feb'}	29th element of ar Value associated Same as \$days, The \$days varial Same as above. It Last index of arra 29th element of ar Multi-dimensional Multi-dimensional	29th element of array @days Value associated with the Feb key of hash %days Same as \$days, but unambiguous before alphanumerics. Useful inside strings for interpolation of variables followed by other letters. The \$days variable inside the Dog package. Same as above. However this is an archaic use of the single quote. Last index of array @days. 29th element of array pointed to by reference \$days. Multi-dimensional array Multi-dimensional hash			ariables followed by other letters.
- Continues and extension to the continue of the register of boards of the continues of th	Array	@days[3,4,5]	Array slice containing (\$days[3], \$d		ys[4], \$days[5]).		
** Index of Peatons, Definition of Peatons ** Index of the case of the array on a state of the ment of the district data structure.	• <u>slices</u>	Don't use a slice when yo	u know you need e:	xactly one element.			
## Subject (clan = 3.1, Fath or \$1 supple 20 ; 28,) ## Edeyar(17, FP) That side containing (Sobyar (17)), Sabyar (17)), Sabyar (17), Sabyar (17	Anonymous arrays				Array reference allows Perl to to	treat the array as a single	item.
Secondary Seco		% %days	• %days = (Ja	nn => 31, Feb => \$1	leap? 29 : 28,)		
Typesido - +foo Sev. Advanced Pert Programming. Ist. Edition Section 3.2 Trinds of package variables		@days{'J',F'}					
Table of productions Table of productions Section variables	Subroutine	& &foo	& is needed to cre	eate reference to subrou	itine.		
Scalar values Seath values * numeric * integer: using the system's ratio at format * better titlerate sumption * numeric * integer: using the system's ratio at format * better titlerate sumption * better titlerate sumption * integer: using the system's ratio at format * bissual - transparent tog number according * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and values tog * double-coulded strings: perform tog * system * double-coulded strings: perform backstalast and values tog * double-coulded strings: perform tog * system * strings * double-coulded strings: perform backstalast and values tog * system * double-coulded strings: perform backstalast and		100				_	ogramming, 1st Edition Section 3.2
** numeric: ** integer: using the system's native format. ** bigint: ** transparent big integer support. ** bigint: ** tran	variables or variable-	2. array variables		format namehow to form	es nat output in Perl?, Perl-Formats		
Pulmeric: Finger: using the system's ratio format biginar interpolation for the control	Scalar values				•	to-number conversions	Useful related <u>builtin functions</u>
* single-quote adrouble quote and double quoted strings an spread multiple limes it embeds the newline characters; no substitution is done (as explained above). **Unicode support** **See: Pert Unicode Tutorial, Pert Unicode Introduction, Pert Unicode Support** © perfood **See: Pert Unicode Tutorial, Pert Unicode Introduction, Pert Unicode Support** © perfood **See: Pert Unicode Tutorial, Pert Unicode Introduction, Pert Unicode Support** © perfood **See: Pert Unicode Tutorial, Pert Unicode Introduction, Pert Unicode Support** © Perfood **See: Pert Unicode Tutorial, Pert Unicode Introduction, Pert Unicode Support** © Perfood **See: Pert Unicode Tutorial, Pert Unicode Introduction, Pert Unicode Support** © Perfood **See: Pert Unicode Tutorial, Pert Unicode Introduction, Pert Unicode Support** © Perfood **See: Pert Unicode Tutorial, Pert Unicode Support** © Perfood **See: Pert Unicode Tutorial, Pert Unicode Support** © Perfood **Note: See: See: See: See: See: See: See: S		 bignum - transparent big number support. floating-point: using the system's native format. bigrat - transparent big rational number support. my \$x = 6.02e23; # scientific notation my \$x = 0x1f.0p3; # power² exponent: Perl >= v5.22 my \$x = 4.294_967_296; # underline for legibility my \$x = 0x1234_5678; # underline in hex is also OK my \$x = 0377; # octal my \$x = 00377; # octal also perl >= v5.34 my \$x = 0xffff; # hexadecimal 				POSIX::ceil POSIX::floor	
Quote constructs Commany Generic Meaning Interpolates? Notes	• string	 double-quoted strings: perform backslash and variable interpolation of expression that begin with \$ (a scalar) or @ (an array). Hashes cannot be interpolated. single-quote strings: only perform \' and \\ substitution (to ' and \ respectively), nothing else. Single quote and double quote strings can spread multiple lines: it embeds the newline character on each new line. 					
Outote constructs See:	Unicode support						
See: - Strings in Participated (1) qq// Under string qq// Literal string qq// Under	Quote constructs	•	Meaning	Interpolates?	Notes		
escapes (only inside double quoted strings) • translation escapes (only inside double quoted strings) • translation escapes (only inside double quoted strings) • translation escapes (only inside double quoted strings) • bareword In Perl, a bareword refers to a sequence of characters suitable for an identifier. It's not quoted. By default Perl allows barewords to behave like strings. • here documents • Here docs @ Perl maven • Perl here docs @ Perl maven • Perl here doc @ Wikipedia • Perl Regexp info, cheatsheets regexp testers • Perl Pragma to declare constants. • Perl Pragma to declare	 Strings in Perl: quoted, interpolated 	"" qq// `` qx// () qw// // m// s/// s///	Literal string Command execut World list Pattern match Pattern substitution Character translati	yes Yes No Yes On Yes Yes No Yes No No	 You can use whitespace between my \$chuck_of_code = if (\$condition print "Salut"); It's also possible to write: s<int (a-f)="" [a-f];<="" li=""> Array variables are interpola </int>	een the quote specifier are q {) { t!"; foo>(bar) and tr(a-	nd its initial bracketing character:
Force next character to lowercase (only inside double quoted strings) Larrow of the part of the pa	escapes (only inside double quoted	\b Backspace \e ESC character \f Form feed \n Newline (usually L \r Carriage return (U	F)	\033 \0{33} \x7f \x{263a}	ESC in octal ESC in octal DEL in hexadecimal Character number 0x263A	Latin small LE	TTER E WITH ACUTE} é
 This is not allowed when any of use strict; or use strict "subs"; or use v5.12; is specified. Here documents Here docs @ Perl maven Perl here doc @Wikipedia Perl here doc @Wikipedia Double quotes: <"EOF"; Supports variable interpolation. Can also be written with whitespace as in <"EOF"; Single quotes: <"EOF"; Does not support interpolation. Can also be written with whitespace as in <"EOF"; Sackticks: <eof; <<<"eof",="" <<<<"eof",="" <<<<"eof",<="" <<~"eof",="" <<~\eof",="" allows="" also="" can="" forms:="" here-doc="" indenting="" other="" string.="" th="" the="" use="" with="" ~=""><th>escapes (only inside double quoted</th><th></th><th>ter to lowercase</th><th>\L Force all following \F Force all following</th><th>g characters to lowercase. Ends at g characters to Unicode fold case.</th><th>t \E . Ends at \E</th><th>\E Ends \U, \L, \F or \Q</th></eof;>	escapes (only inside double quoted		ter to lowercase	\L Force all following \F Force all following	g characters to lowercase. Ends at g characters to Unicode fold case.	t \E . Ends at \E	\E Ends \U, \L, \F or \Q
* Here docs @ Perl maven * Perl here doc @Wikipedia * Default : < <eof; "eof";="" <<="" <<"eof",="" <<"eof";="" <<*eof;="" <<<"e<="" <<<"eof",="" <<eof;="" <<~"eof",="" <<~eof,="" allows="" also="" as="" be="" can="" dackticks:="" does="" forms:="" here-doc="" in="" indenting="" interpolation.="" not="" other="" quotes:="" single="" string.="" support="" supports="" td="" the="" use="" variable="" whitespace="" with="" written="" ~=""><td>• bareword</td><td colspan="4"></td></eof;>	• bareword						
info, cheatsheets & regexp testers • Learn PCRE in X minutes • regex 101 • RegEx Pal • Perl pragma to declare constants. • Perl pragma to declare constants. • But be aware that these are still not read-only, that they inject sub-routines and have several limitations. Read the doc!!	Here docs @ Perl mavenPerl here doc	must be placed at the beginning of the terminating line: • <u>Default</u> :					
ren pragma to declare constants.	info, cheatsheets &	Regexp Tutorial				• <u>regex101</u>	ster
	Perl Constants						

• Perl Variables • Perl Variables	To get information about a Perl special variation of the special variation about \$< use: perldoc		ne use the peridoc -v command.			
General variables		· · · · · · · · · · · · · · · · · · ·				
default input and pattern searching space	• \$ARG • \$_		subroutine parameters	• @ARG • @_		
list separator	• \$LIST_SEPARATOR • \$"		Subscript separator for multidimensional array emulation	\$SUBSCRIPT_SEPARATOR\$SUBSEP\$;		
Name of executed program	• \$PROGRAM_NAME • \$0		Name used to execute the current copy of Perl	• \$EXECUTABLE_1 • \$^X	NAME	
Perl process ID	• \$PROCESS_ID • \$PID • \$\$					
Process real GID	• \$REAL_GROUP_ID • \$GID • \$(Process effective GID	\$EFFECTIVE_GROUP_ID\$EGID\$)		
Process real UID	• \$REAL_USER_ID • \$UIG • \$<		Process effective UID	 \$EFFECTIVE_USER_ID\$ \$EUID \$>		
Special variables in sort	• \$a • \$b	Example: by default Perforce numerical compari	rl sort function sorts strings. Pass isons: @sorted = sort {	a sorting function that us \$a <=> \$b } @unsor		
Current environment	%ENV		cessed as an associative array (a hess shell environment variables the		ays.	
Perl interpreter revision, version and subversion	• \$OLD_PERL_VERSION • \$]		Perl interpreter revision, version and subversion	• \$PERL_VERSION • \$^V	ſ	
Maximum file descriptor	• \$SYSTEM_FD_MAX • \$^F					
Fields of each line when auto-split mode is on.	@F					
Include Directories	@INC	Included filenames	%INC	Hook localization (?)	\$INC	
inplace-edit extension value	• \$INPLACE_EDIT • \$^I					
Package's class parent classes	@ISA					
Emergency memory pool	\$^M					
Maximum block nesting	\${^MAX_NESTED_EVAL_BEGIN_BLC	OCKS}				
Name of OS where this Perl was built	• \$OSNAME • \$^O					
Signal handlers	%SIG					
Coderefs for various perl keywords	%{^HOOK}					
Time when program began running	• \$BASETIME • \$^T					
Variables related to regular expressions						
captured sub-patterns	\$ <digit>(\$1,\$2,)</digit>					
Capture buffer content	@{^CAPTURE}					
String matched	• \$MATCH • \$&		String matched (compiled regexp)	\${^MATCH}		
String preceding match	• \$PREMATCH • \$		String preceding match (compiled regexp)	\${^PREMATCH}		
String following match	• \$POSTMATCH • \$'		String following match (compiled regexp)	{^POSTMATCH}		
Last capture group	• \$LAST_PAREN_MATCH • \$+ Most recently closed capture group • \$LAST_SUBMATCH_RESULT • \$^N				CH_RESULT	
Match capture key values	• %{^CAPTURE} • %LAST_PAREN_MATCH • %+					
Match start offsets	• @LAST_MATCH_START • @-	Match ends offsets	• @LAST_MATCH_END • @+	Named captured groups	• %{^CAPTURE_ALL} • %-	
Last successful pattern	\${^LAST_SUCESSFUL_PATTERN}					
Result of last successful regexp assertion	• \$LAST_REGEXP_CODE_RESULT • \$^R					
Maximum regexp nested group	\${^RE_COMPILE_RECURSION_LIMIT}					
regexp debug flag	\${^RE_DEBUG_FLAG}					
regexp internal optimization/memory	\${^RE_TRIE_MAXBUF}					
Variables related to file handles	See also: Perl File Handles					
Name of current file read from <> • diamond operator <>	\$ARGV	Command line arguments of the script	@ARGV	Number of arguments minus one	\$#ARGV	

Special file handle that iterates over command-line filenames in @ARGV	ARGV	Special file handle that points to currently open output file when doing edit-in-place processing	ARGVOUT		
Output field separator for the print operator	• IO::Handle->output_field_separa • \$OUTPUT_FIELD_SEPARATO • \$OFS • \$,		Current line number for the last file handled accessed	 HANDLE->input_line_number(EXPR) \$INPUT_LINE_NUMBER \$NR \$.	
Input record separator (newline by default)	• IO::Handle->input_record_separ • \$INPUT_RECORD_SEPARATO • \$RS • \$/		Output record separator	 IO::Handle->output_record_separator(EXPR) \$OUTPUT_RECORD_SEPARATOR \$ORS \$\ 	
Auto-flush control • See: order of output @ Perl Maven	• HANDLE->autoflush(EXPR) • \$OUTPUT_AUTOFLUSH • \$I		Last read file handle	\${^LAST_FH}	
Variables related to format					
Current value of the write() accumulator for format() lines.	• \$ACCUMULATOR • \$^A				
Form feed format. defaults to \f	• IO::Handle->format_formfeed(F • \$FORMAT_FORMFEED • \$^L	XPR)	Set of characters after which a string may be broken to fill continuation fields	• IO::Handle->format_line_break_characters EXPR • \$FORMAT_LINE_BREAK_CHARACTERS • \$:	
Number of lines left on the page on currently selected output channel	HANDLE->format_lines_left(Extraction of the state of	XPR)	Current page length of current output channel	HANDLE->format_lines_per_page(EXPR)\$FORMAT_LINES_PER_PAGE\$=	
Name of current top- page format of output channel	HANDLE->format_top_name(E \$FORMAT_TOP_NAME \$^	XPR)	Report format name of output channel	HANDLE->format_name(EXPR)\$FORMAT_NAME\$~	
Error Variables	The variables \$@, \$!, \$^E, and \$? contain information about different types of error conditions that may appear during execution of a Perl program. They correspond to errors detected by the Perl interpreter, C library, operating system, or an external program, respectively.				
Perl error from the last eval operator	• \$EVAL_ERROR • \$@ Current state of interpreter			• \$EXCEPTIONS_BEING_CAUGHT • \$^S	
Current value of C errno integer variable	• \$ERRNO when us returns t	ns the system variable <u>errno</u> ed in a numeric context, but he string from <u>perror()</u> when string context.	Hash of error names to 0 or 1, set to 1 if current error is this error.	• %OS_ERROR • %ERRNO • %!	
OS detected error	• \$EXTENDED_OS_ERROR • \$^E				
Status returned by last pipe close, backtick command, wait, waited, or system() call.	• \$CHILD_ERROR • \$?		native status returned by last pipe close , backtick command, wait() or waitpid() or system() call	\${^CHILD_ERROR_NATIVE}	
Current value of warning switch	• \$WARNING • \$^W		Current set of warning checks enabled by the use warnings pragma	\${^WARNING_BITS}	
Variables related to the interpreter state	These variables provide information	about the current interpreter	state.		
Flag associated with the -c switch	• \$COMPILING • \$^C		The current value of the debugging flags	• \$DEBUGGING • \$^D	
Current phase of the perl interpreter	\${^GLOBAL_PHASE}				
Compile-time hints for the perl interpreter. Internal use only	\$^H Values of compiled statements %^H				
Input/Output Layers. Internal use by PerilO only.	\${^OPEN}				
Debugging support. Internal variable.	• \$PERLDB • \$^P				
Taint mode	\${^TAINT}		Safe locale operations availability	\${^SAFE_LOCALES}	
Unicode Settings of Perl	\${^UNICODE}				
Internal UTF-8 offset caching code state	\${^UTF8CACHE}		State of UTF-8 locale detected by perl at startup.	\${^UTF8LOCALE}	
Deprecated and removed variables:	\$# \$* \$[\${^ENCOL	SING \${^WIN32_SLO	PPY_STAT}		

Perl 5 open file

References	open @ peridoc browser Writing to files with Perl @ Perl Maven open file in-memory @ stackOverflow	Stupid open() tricks @Perl.com: No explicit filename create an anonymous temporary file print to a string read lines from a string

Perl 5 Statements

Conditional statements				
Loop control	• while (condition) { } • until (condition) { }		loop control keywords: • next : starts the next iteration of the loop. • last : exits the loop. • redo : restarts the loop block without evaluating the condition again.	loop control keywords: • continue block: executed before evaluating condition again.
Statement modifiers	• if EXPR • unless EXPR • while EXPR • until EXPR • for LIST • foreach LIST • when EXPR	processed. Therefore a loop like the following trying to stop on a line that		The while statement imposes a scalar context; it takes one line at a time from <stdin> and the following code works properly: while (<stdin>) { last if /_END/; ; }</stdin></stdin>

Perl 5 Functions

Perl Functions Perl syntax	To get information about a Perl function from the command line use the perldoc -f command. To get information about print use: perldoc -f print
! Cautionary notes	
 each keyword is broken Use Var::Pairs instead. 	Do NOT use the built-in each. It is broken, as described by <u>Damian Conway</u> in his <u>Modern Perl Best Practice O'Reilly course</u> , section control structure. • each is not re-entrant: • nested loops of each over the same hash does not work as expected and will create infinite loop since the nested loop each juts iterates from where the first loop each left it. • Exiting the loop leaves the state of the each internal pointer at the current location. • If you use each on the same hash later it will resume from where it left, it will not start form the beginning.
print functions	• <u>print</u> • say use feature gw(say); or use v5.10; (or higher). Like print, but implicitly appends a newline at the end of the list.

PerlTidy formatting control

perItidy option	Option	Impact
indentation style	bl,opening-brace-on-new-linebrace-left	 Without this option (the default) the code indentation style selected is <u>K&R style</u>. With this option, the indentation style is <u>Allman/BSD style</u>.