**Perl Functions** 

broken

instead.

Cautionary notes each keyword is

Use Var::Pairs

· each is not re-entrant:

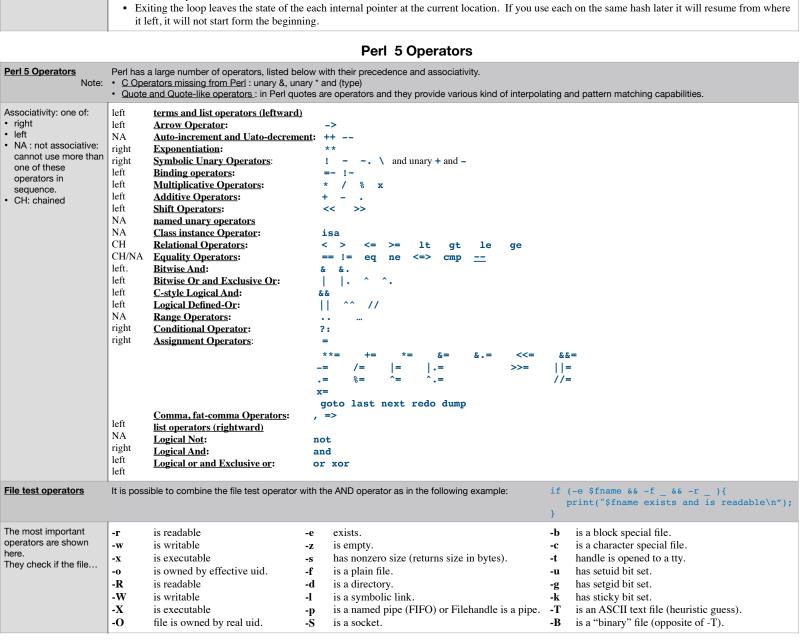
the first loop each left it.

See also: <u>\$\mathbb{P}\tau\$ - Perl</u>	Perl Tools	Perl Style Guide. perlcritic script uses Perl::Critic to scan Perl code. The perltidy application reformats Perl code.			
Perl @ Wikipedia     perl.org     perldoc browser	Learning Perl	Perl Intro - a quick introduction to Perl     Online Perl books     Beginning Perl		perl , Perl command line options     perlivp , perldoc , perlbug / perlthanks     perlsec - Perl security	
CPAN	CPAN @ Wikipedia		Command line tools interacting with CPAN:		
	The Zen of Compre	hensive Archive Networks	• cpan : install on some Linux with: sudo dnf install perl-CPAN		
	• CPAN	• cpanplus			
	Search CPAN — meta::cj	<u>pan</u>	• cpanminus: cpanm: install on some Linux with: sudo dnf install perl-App-		
	<ul> <li>PAUSE - Perl Authors Up</li> </ul>	rs Upload Server cpanminus			

## Perl 5 Keywords ###

Do NOT use the built-in each. It is broken, as described by Damian Conway in his Modern Perl Best Practice O'Reilly course, section control structure.

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



	Perl 5 Constants and Variables				
Perl Constants	<ul> <li>Perl pragma to declare constants.</li> <li>But be aware that these are still not read-only, that they inject sub-routines and have several limitations. Read the doc!!</li> <li>CPAN modules for defining constants by Neil Bowers.</li> <li>Of particular interest: Const::Fast and Attribute::Constant for efficient read-only constants.</li> </ul>				
Perl Special Variables Perl Variables	To get information about a Perl special variable from the command line use the perldoc -v command.  • To get information about \$< use: perldoc -v '\$<'				
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_NAME • \$^X		
Perl process ID	• \$PROCESS_ID • \$PID • \$\$				
Process real GID	<ul><li>\$REAL_GROUP_ID</li><li>\$GID</li><li>\$(</li></ul>	Process effective GID	<ul><li> \$EFFECTIVE_GROUP_ID</li><li> \$EGID</li><li> \$)</li></ul>		

Process real UID	• \$REAL_USER_ID • \$UIG • \$<		Process effective UID	• \$EFFECTIVE_USER_ID\$ • \$EUID • \$>		
Special variables in sort	• \$a • \$b					
Current environment	%ENV Environment variable accessed as an associative array (a hash).  • See: Perl: How to access shell environment variables through Perl associative arrays.					
Perl interpreter revision, version and subversion	• \$OLD_PERL_VERSION • \$1		Perl interpreter revision, version and subversion	• \$PERL_VERSION • \$^V	1	
Maximum file descriptor						
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_BLO	OCKS}				
Name of OS where this	• \$OSNAME					
Perl was built	• \$^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 • \$'			{^POSTMATCH}		
Last capture group	• \$LAST_PAREN_MATCH • \$+		Most recently closed capture group	• \$LAST_SUBMAT • \$^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 <>	\$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	<ul> <li>IO::Handle-&gt;output_field_separator(EX)</li> <li>\$OUTPUT_FIELD_SEPARATOR</li> <li>\$OFS</li> <li>\$,</li> </ul>	KPR)	Current line number for the last file handled accessed	• HANDLE->input_ • \$INPUT_LINE_N • \$NR • \$.	line_number( EXPR ) UMBER	
Input record separator (newline by default)	<ul> <li>IO::Handle-&gt;input_record_separator( EXPR )</li> <li>\$INPUT_RECORD_SEPARATOR</li> <li>\$RS</li> <li>\$/</li> </ul>		Output record separator	<ul> <li>IO::Handle-&gt;output_record_separator( EXPR )</li> <li>\$OUTPUT_RECORD_SEPARATOR</li> <li>\$ORS</li> <li>\$\( \)</li> </ul>		
Auto-flush control	HANDLE->autoflush( EXPR )     SOUTPUT_AUTOFLUSH     \$		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(EXPR)		Set of characters after which a string may be broken to fill		at_line_break_characters EXPR _BREAK_CHARACTERS	
defaults to 1	• \$FORMAT_FORMFEED • \$^L		continuation fields	• \$:		

Number of lines left on the page on currently selected output channel	<ul><li> HANDLE-&gt;format_lines_left(EXPR)</li><li> \$FORMAT_LINES_LEFT</li><li> \$-</li></ul>	Current page length of current output channel	<ul><li>HANDLE-&gt;format_lines_per_page(EXPR)</li><li>\$FORMAT_LINES_PER_PAGE</li><li>\$=</li></ul>		
Name of current top- page format of output channel	<ul><li>HANDLE-&gt;format_top_name(EXPR)</li><li>\$FORMAT_TOP_NAME</li><li>\$^</li></ul>	Report format name of output channel	<ul><li>HANDLE-&gt;format_name(EXPR)</li><li>\$FORMAT_NAME</li><li>\$~</li></ul>		
• 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	• \$OS_ERROR • \$ERRNO • \$!	Hash of error names to 0 or 1, set to 1 if current error is this error.	<ul><li>%OS_ERROR</li><li>%ERRNO</li><li>%!</li></ul>		
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 wiatpid() 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:	\$# \$* \$[ \${^ENCODING} \${^WIN32 SLOP	PPY_STAT}			