↖ PEL Topic Index

Perl 5 🚧

See also: 𝔓𝔈 - Perl • **Perl @ Wikipedia** • **perl.org** • **perldoc browser**	**Perl Tools**	**Perl Style Guide**. **perlcritic script** uses **Perl::Critic** to scan Perl code. The **perltidy** application reformats Perl code.
	Learning Perl	• Perl Intro - a quick introduction to Perl • Online Perl books • Beginning Perl

Perl 5 Syntax 🚧

Perl 5 Operators	Perl has a large number of operators, listed below with their precedence and associativity. Note: • C Operators missing from Perl : unary &, unary * and (type) • Quote and Quote-like operators : in Perl quotes are operators and they provide various kind of interpolating and pattern matching capabilities.		
Associativity: one of: • right • left • NA : not associative: cannot use more than one of these operators in sequence. • CH: chained	left	**terms and list operators (leftward)**	
	left	**Arrow Operator**:	`->`
	NA	**Auto-increment and Uato-decrement**:	`++ --`
	right	**Exponentiation**:	`**`
	right	**Symbolic Unary Operators**:	`! ~ ~. \` and unary `+` and `-`
	left	**Binding operators**:	`=~ !~`
	left	**Multiplicative Operators**:	`* / % x`
	left	**Additive Operators**:	`+ - .`
	left	**Shift Operators**:	`<< >>`
	NA	**named unary operators**	
	NA	**Class instance Operator**:	`isa`
	CH	**Relational Operators**:	`< > <= >= lt gt le ge`
	CH/NA	**Equality Operators**:	`== != eq ne <=> cmp ~~`
	left.	**Bitwise And**:	`& &.`
	left	**Bitwise Or and Exclusive Or**:	`| |. ^ ^.`
	left	**C-style Logical And**:	`&&`
	left	**Logical Defined-Or**:	`|| ^^ //`
	NA	**Range Operators**:	`.. …`
	right	**Conditional Operator**:	`?:`

	right	**Assignment Operators**:	`=`
			`**= += *= &= &.= <<= &&=`
			`-= /= |= |.= >>= ||=`
			`.= %= ^= ^.= //=`
			`x=`
			`goto last next redo dump`

		Comma, fat-comma Operators:	`, =>`
	left	**list operators (rightward)**	
	NA	**Logical Not**:	`not`
	right	**Logical And**:	`and`
	left	**Logical or and Exclusive or**:	`or xor`
	left		

File test operators	It is possible to combine the file test operator with the AND operator as in the following example:		`if (-e $fname && -f _ && -r _ ){`
			`  print("$fname exists and is readable\n");`
			`}`

The most important operators are shown here. They check if the file…	**-r**	is readable	**-e**	exists.	**-b**	is a block special file.
	-w	is writable	**-z**	is empty.	**-c**	is a character special file.
	-x	is executable	**-s**	has nonzero size (returns size in bytes).	**-t**	handle is opened to a tty.
	-o	is owned by effective uid.	**-f**	is a plain file.	**-u**	has setuid bit set.
	-R	is readable	**-d**	is a directory.	**-g**	has setgid bit set.
	-W	is writable	**-l**	is a symbolic link.	**-k**	has sticky bit set.
	-X	is executable	**-p**	is a named pipe (FIFO) or Filehandle is a pipe.	**-T**	is an ASCII text file (heuristic guess).
	-O	file is owned by real uid.	**-S**	is a socket.	**-B**	is a “binary” file (opposite of -T).