Vim-Plugin

perl-support.vim

Version 4.2

Hot keys

Key mappings for Vim and gVim.
Plugin: http://vim.sourceforge.net
Fritz Mehner (mehner@fh-swf.de)

(i) insert mode, (n) normal mode, (v) visual mode

		Load / Unload Perl Support
`\lps`	load menues	(n)
`\ups`	unload menues	(n)

		Help
`\hp`	help (plugin)	(n,i)

		Comments
`\cl`	end-of-line comment	(n, v, i)
`\cj`	adjust end-of-line comments	(n, v, i)
`\cs`	set end-of-line comment col.	(n)
`\cfr`	frame comment	(n, i)
`\cfu`	function description	(n, i)
`\cm`	module description	(n, i)
`\ch`	file header (.pl)	(n)
`\ce`	file header (.pm)	(n)
`\ca`	file header (.t)	(n)
`\cp`	file header (.pod)	(n)
`\ckb`	keyword comm. `BUG`	(n, i)
`\ckt`	keyword comm. `TODO`	(n, i)
`\ckr`	keyword comm. `TRICKY`	(n, i)
`\ckw`	keyword comm. `WARNING`	(n, i)
`\cko`	keyword comm. `WORKAROUND`	(n, i)
`\ckn`	keyword comm. `new keyword`	(n, i)
`\cc`	code ↔ comment	(n, v)
`\cb`	code block → comment	(n, v)
`\cn`	uncomment code block	(n)
`\cd`	date	(n, i)
`\ct`	date & time	(n, i)
`\cv`	vim modeline	(n, i)

		Statements
`\sd`	`do { } while`	(n, v, i)
`\sf`	`for { }`	(n, v, i)
`\sfe`	`foreach { }`	(n, v, i)
`\si`	`if { }`	(n, v, i)
`\sie`	`if { } else { }`	(n, v, i)
`\se`	`else { }`	(n, v, i)
`\sei`	`elsif { }`	(n, v, i)
`\su`	`unless { }`	(n, v, i)
`\sue`	`unless { } else { }`	(n, v, i)
`\st`	`until { }`	(n, v, i)
`\sw`	`while { }`	(n, v, i)
`\s{`	`{ }`	(n, v, i)

		Idioms
`\$`	`my $;`	(n, i)
`\$=`	`my $ = ;`	(n, i)
`\$$`	`my ( $, $ );`	(n, i)
`\@`	`my @;`	(n, i)
`\@=`	`my @ = (,,);`	(n, i)
`\%`	`my %;`	(n, i)
`\%=`	`my % = (=>,=>,);`	(n, i)
`\ir`	`my $rgx_ = q//;`	(n, i)
`\im`	`$ =~ m//xm`	(n, i)
`\is`	`$ =~ s///xm`	(n, i)
`\it`	`$ =~ tr///xm`	(n, i)
`\isu` `\ifu`	`subroutine`	(n, v, i) (n, v, i)
`\ip`	`print "...\n";`	(n ,i)
`\ii`	open input file	(n, v, i)
`\io`	open output file	(n, v, i)
`\ipi`	open pipe	(n, v, i)

		Snippet
`\nr`	read code snippet	(n)
`\nw`	write code snippet	(n, v)
`\nw`	edit code snippet	(n)
`\ntl`	edit local templates	(n)
`\ntg`	edit global templates	(n)
`\ntr`	reread the templates	(n)

		Regular Expressions
`\xr`	pick up Regex	(n, v)
`\xs`	pick up string	(n, v)
`\xf`	pick up flag(s)	(n, v)
`\xm`	match	(n)
`\xmm`	match multiple (Regex/target)	(n)
`\xe`	explain Regex	(n, v)

		POSIX Character Classes
`\pa`	`[:alnum:]`	(n, i)
`\ph`	`[:alpha:]`	(n, i)
`\pi`	`[:ascii:]`	(n, i)
`\pb`	`[:blank:]`	(n, i)
`\pc`	`[:cntrl:]`	(n, i)
`\pd`	`[:digit:]`	(n, i)
`\pg`	`[:graph:]`	(n, i)
`\pl`	`[:lower:]`	(n, i)
`\pp`	`[:print:]`	(n, i)
`\pn`	`[:punct:]`	(n, i)
`\ps`	`[:space:]`	(n, i)
`\pu`	`[:upper:]`	(n, i)
`\pw`	`[:word:]`	(n, i)
`\px`	`[:xdigit:]`	(n, i)

		Run
`\rr`	update file, run script	(n)
`\rs`	update file, check syntax	(n)
`\ra`	set command line arguments	(n)
`\rw`	set Perl cmd. line switches	(n)
`\rd`	start debugger	(n)
`\re`	make script executable	(n)
`\rp \h`	read perldoc for word under cursor	(n)
`\ri`	show installed Perl modules	(n)
`\rg`	generate Perl module list	(n)
`\ry`	run `perltidy`	(n, v)
`\rps`	run `Devel::SmallProf`	(n)
`\rpf`	run `Devel::FastProf`	(n)
`\rpn`	run `Devel::NYTProf`	(n)
`\rc`	run `perlcritic`	(n)
`\rt`	save buffer with timestamp	(n)
`\rh`	hardcopy buffer	(n, v)
`\rk`	settings and hotkeys	(n)
`\rx`	set xterm size	(n, GUI only)
`\ro`	change output destination	(n)

perlcritic

Ex commands for `perlcritic` (version 1.01+)
Use tab expansion to choose the severity or the verbosity.

`:CriticSeverity`	`1`	`2`	`3`	`4`	`5`
	`brutal`	`cruel`	`harsh`	`stern`	`gentle`
`:CriticVerbosity`	1 . . . 11				
`:CriticOptions`	option(s), see `perlcritic(1)`				

Profiling

The following ex commands can be used to sort a profiler report
in the quickfix window.
Use tab expansion to choose the sort criterion or the file name.

For `Devel::SmallProf`

`:SmallProfSort`	`file-name|line-number|line-count|time|ctime`

For `Devel::FastProf`

`:FastProfSort`	`file-name|line-number|time|line-count`

For `Devel::NYTProf`

`:NYTProfCSV`	Read a CSV-file.
`:NYTProfSort`	`file-name|line-number|time|calls|time-call`