Vim-Plug-in

perl-support.vim

Version 5.2

Hot keys

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

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

		***P**erl*
`\ft`	file tests	(n,i)

		***H**elp*
`\h`	read perldoc for word under cursor	(n,i)
`\hp`	help (plug-in)	(n,i)

		***C**omments*
[n]`\cl`	end-of-line comment	(n, v, i)
[n]`\cj`	adjust end-of-line comments	(n, v, i)
`\cs`	set end-of-line comment col.	(n)
[n]`\cc`	code ↔ comment	(n, v)
`\cb`	code block → comment	(n, v)
`\cub`	uncomment code block	(n)
`\cfr`	frame comment	(n, i)
`\cfu`	function description	(n, i)
`\cme`	method description	(n, i)
`\chpl`	file header (.pl)	(n)
`\chpm`	file header (.pm)	(n)
`\cht`	file header (.t)	(n)
`\chpo`	file header (.pod)	(n)
`\cd`	date	(n, i)
`\ct`	date & time	(n, i)
`\ck`	keyword comments	(n, i)
`\cma`	plug-in macros	(n, i)

		***S**tatements*
`\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)
`\sg`	`given { }`	(n, v, i)
`\swh`	`when { }`	(n, v, i)

		***I**dioms*
`\id`	`my $;`	(n, i)
`\ida`	`my $ = ;`	(n, i)
`\idd`	`my ( $, $ );`	(n, i)
`\ia`	`my @;`	(n, i)
`\iaa`	`my @ = (,,);`	(n, i)
`\ih`	`my %;`	(n, i)
`\iha`	`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`	`subroutine`	(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)

		***S**nippet*
`\nr`	read code snippet	(n, i)
`\nv`	view code snippet	(n, i)
`\nw`	write code snippet	(n, v, i)
`\ne`	edit code snippet	(n, i)
`\ntl`	edit local templates	(n, i)
`\ntr`	reread the templates	(n, i)
`\nts`	choose template style	(n, i)
`\njt`	insert jump tag	(n, i)
`\nxs`	regex snippet template	(n, i)

		*Regular **E**xpressions*
`xpc`	POSIX classes	(n, i)
`xup`	Unicode properties	(n, i)
`xex`	extended Regex	(n, i)
`xms`	metasymbols	(n, i)
[n]`\xr`	pick up Regex	(n, v)
[n]`\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)

		***S**pecial Variables*
`\vb`	basics	(n, i)
`\ve`	errors	(n, i)
`\vf`	files	(n, i)
`\vid`	IDs	(n, i)
`\vio`	IO	(n, i)
`\vr`	regexp	(n, i)
`\vs`	POSIX signals	(n, i)
`\vue`	use English	(n, i)

		***P**OD*
`\ppc`	`pod, cut`	(n, i)
`\pfc`	`for, cut`	(n, i)
`\pbh`	`begin html, end`	(n, i)
`\pbm`	`begin man, end`	(n, i)
`\pbt`	`begin text, end`	(n, i)
`\ph1`	`head1`	(n, i)
`\ph2`	`head2`	(n, i)
`\ph3`	`head3`	(n, i)
`\pob`	`over, back`	(n, i)
`\pi`	`item`	(n, i)
`\pod`	run `podchecker`	(n, i)
`\podh`	convert POD data to .html file	(n, i)
`\podm`	Convert POD data to *roff input	(n, i)
`\podt`	Convert POD data to ASCII text	(n, i)
`\pm`	markup sequences	(n, i)

		***P**rofiling*
`\rps`	run `SmallProf`	(n, i)
`\rpss`	sort `SmallProf` report	(n, i)
`\rpso`	open existing `SmallProf` results	(n, i)
`\rpf`	run `FastProf`	(n, i)
`\rpfs`	sort `FastProf` report	(n, i)
`\rpfo`	open existing `FastProf` results	(n, i)
`\rpn`	run `NYTProf`	(n, i)
`\rpns`	sort `NYTProf` report	(n, i)
`\rpno`	open existing `NYTProf` results	(n, i)
`\rpnh`	browse HTML files (`NYTProf`)	(n, i)

		***R**un*
`\rr`	update file, run script	(n, i)
`\rs`	update file, check syntax	(n, i)
`\ra`	set command line arguments	(n, i)
`\rw`	set Perl cmd. line switches	(n, i)
`\re`	make script executable	(n, i)
`\rm`	run `make`	(n, i)
`\rcm`	choose makefile	(n, i)
`\rm`	`make clean`	(n, i)
`\rma`	command line arguments for `make`	(n, i)
`\rd`	start debugger	(n, i)
`\ri`	show installed Perl modules	(n, i)
`\rg`	generate Perl module list	(n, i)
`\ry`	run `perltidy`	(n, v, i)
`\rpc`	run `perlcritic`	(n, i)
`\rpcs`	set `perlcritic` severity	(n, i)
`\rpcv`	set `perlcritic` verbosity	(n, i)
`\rpco`	set `perlcritic` options	(n, i)
`\rt`	save buffer with timestamp	(n, i)
`\rh`	hardcopy buffer	(n, v, i)
`\rk`	settings and hotkeys	(n, i)
`\rx`	set xterm size	(n, i GUI only)
`\ro`	change output destination	(n, i)

Run

Specify command line arguments for the script in the current buffer.
Use tab expansion to choose a file or a directory.

```
:PerlScriptArguments
```

Specify command line switches for the Perl interpreter.

```
:PerlSwitches
```

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)`				

Regular Expression Tester

Ex command for the regular expression tester. Set control character replacements for newline and tabulator used to display the results of a match, e.g.:

```
:RegexSubstitutions  '$~'
```

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.
`:NYTProfHTML`	Read the HTML-reports with an external viewer (GUI only).
`:NYTProfSort`	`file-name|line-number|time|calls|time-call`