vim-support.vim VERSION 2.4 HOT KEYS

Key mappings for Vim and gVim

http://www.vim.org — Wolfgang Mehner, wolfgang-mehner@web.de

(i) insert mode, (n) normal mode, (v) visual mode, (T) tab-completion

		$\underline{H}elp$
\he	English dictionary	(n,i)
\hk	help (Vim functions)	(n,i)
<s-f1></s-f1>		
\hp	help (vim-support)	(n,i)
	<u>C</u>	omments
[n]\ c 1	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	comment code	(n, i, v)
[n]\ CO	uncomment code	(n, i, v)
\ca	function description (auto)	(n, i, v)
\cfr	frame comment	(n, i)
\cfu	${ m function\ description}$	(n, i)
\ch	file description	(n, i)
\cd	date	(n, i)
\ct	date & time	(n, i)
\ck	keyword comments	(n, i, T)
\cma	plugin macros	(n, i, T)

Statements			
\sl let list		<u>S</u>	tatements
\sd let dictionary (n, i) \sf for (n, v, i) \sif ifendif (n, v, i) \sie ifelseendif (n, v, i) \sei elseif (n, i) \sei elseif (n, i) \sw while (n, v, i) \st trycatch (n, v, i) \tag{dioms} \ib builtin functions (n, i, T) \ii iterators (n, v, i, T) \if function (n, v, i) \tag{Regular Expressions} \tag{xc} capture (n, i, v) \tag{xbc} branch (n, i, v) \tag{xbc} branch (n, i, v) \tag{xb} word (n, i, v) \tag{xw} word (n, i, T) \tag{xs} switches (n, i, T) \tag{PerI} \tag{PerI} \tag{PerI} \tag{PerI} \tag{PerI} \tag{PerI} \tag{PerI} \tag{pm} Vim::Msg("", "Comment") (n, i) \tag{n, i} \tag{pmc} Vim::Msg("", "Comment") (n, i) \tag{n, i} \ta	\sv	let variable	(n, i)
\sf for	\sl	let list	(n, i)
\sif ifendif (n, v, i) \sie ifelseendif (n, v, i) \sei elseif (n, i) \sel else (n, i) \sw while (n, v, i) \st trycatch (n, v, i) \tag{dioms} \ib builtin functions (n, i, T) \ii iterators (n, v, i, T) \if function (n, v, i) \tag{Regular Expressions} \tag{xc} capture (n, i, v) \tag{xbc} branch (n, i, v) \tag{xb} word (n, i, v) \tag{xw} word (n, i, v) \tag{xw} word (n, i, T) \tag{xc} character classes (n, i, T) \tag{xs} switches (n, i, T) \tag{perl}	\sd	let dictionary	(n, i)
\sie	\sf	for	(n, v, i)
\sei elseif	\sif	ifendif	(n, v, i)
\sel else	\sie	ifelseendif	(n, v, i)
\sw while	∖sei	elseif	(n, i)
\st trycatch (n, v, i) \[\textit{Idioms} \\ \textit{ib} \text{builtin functions} (n, i, T) \\ \text{ii} \text{iterators} (n, v, i, T) \\ \text{if} \text{function} (n, v, i) \\ \text{Regular Expressions} \\ \text{xc} \text{capture} (n, i, v) \\ \text{xbc} \text{branch} (n, i, v) \\ \text{xm} \text{word} (n, i, v) \\ \text{xm} \text{word} (n, i, v) \\ \text{xs} \text{switches} (n, i, T) \\ \text{xs} \text{switches} (n, i, T) \\ \text{ps} \text{Perl snippet} (n, i) \\ \text{pd} \text{Vim::DoCommand()} (n, i) \\ \text{pm} \text{Vim::Msg("")} (n, i) \\ \text{pmc} \text{Vim::Msg("", "Comment")} (n, i) \\ \text{pmc} \text{Vim::Msg("", "Warning")} (n, i) \\ \end{array}	\sel	else	(n, i)
Idioms	\sw	while	(n, v, i)
\ib	\st	trycatch	(n, v, i)
$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$			<u>I</u> dioms
$\begin{tabular}{ c c c c c } \hline \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	\ib	builtin functions	(n, i, T)
Lif function (n, v, i) Regular Expressions XC capture (n, i, v) Xbc branch (n, i, v) Xbn branch, no capture (n, i, v) Xw word (n, i, T) Xcc character classes (n, i, T) Xs switches (n, i, T) Perl Perl snippet (n, i) Vpd Vim::DoCommand() (n, i) Lype Vim::Eval() (n, i) Lype Vim::Msg("") (n, i) Lype Vim::Msg("", "Comment") (n, i) Lype Vim::Msg("", "Comment") (n, i)	\ii	iterators (
\xc	\if		
\xbc branch (n, i, v) \xbn branch, no capture (n, i, v) \xw word (n, i) \xcc character classes (n, i, T) \xs switches (n, i, T) \ps Perl snippet (n, i) \pd Vim::DoCommand() (n, i) \pd Vim::Eval() (n, i) \pm Vim::Msg("") (n, i) \pmc Vim::Msg("", "Comment") (n, i) \pme Vim::Msg("", "Warning") (n, i)		Regular E	xpressions
\xbn branch, no capture (n, i, v) \xw word (n, i) \xcc character classes (n, i, T) \xs switches (n, i, T) \proceedar \frac{Perl}{Perl} \text{Ps} Perl snippet (n, i) \pd Vim::DoCommand() (n, i) \pd Vim::Eval() (n, i) \pm Vim::Msg("") (n, i) \pmc Vim::Msg("", "Comment") (n, i) \pmc Vim::Msg("", "Warning") (n, i)	\xc	capture	(n, i, v)
$\begin{tabular}{ c c c c c c c c c c c c c c c c c c c$	\xbc	branch	(n, i, v)
$\begin{tabular}{lllllllllllllllllllllllllllllllllll$	\xbn	branch, no capture	(n, i, v)
\xs switches \((n, i, T) \) \tag{Perl} \tag{Perl snippet} \((n, i) \) \tag{Vim::DoCommand()} \((n, i) \) \tag{Vim::Eval()} \((n, i) \) \tag{Vim::Msg("")} \((n, i) \) \tag{Vim::Msg("", "Comment")} \((n, i) \) \tag{Vim::Msg("", "Warning")} \((n, i) \) \tag{Vim::Msg("", "Warning")} \((n, i) \)	\xw		(n, i)
Perl snippet (n, i) \pd \Vim::DoCommand() (n, i) \pe \Vim::Eval() (n, i) \pm \Vim::Msg("") (n, i) \pmc \Vim::Msg("", "Comment") (n, i) \pme \Vim::Msg("", "Warning") (n, i)	\xcc	character classes	(n, i, T)
\ps Perl snippet	\xs	switches	(n, i, T)
\pd			\underline{P} erl
\pe	\ps	Perl snippet	(n, i)
\pm	\pd	Vim::DoCommand()	(n, i)
\pmc \ \Vim::Msg() \ (n, i) \\pmc \ \Vim::Msg("", "Comment") \ (n, i) \\pmc \ \Vim::Msg("", "Warning") \ (n, i)	\pe	Vim::Eval()	(n, i)
\pme Vim::Msg("", "Warning") (n, i)	\pm	Vim::Msg("")	(n, i)
\pme $Vim::Msg("", "Warning")$ (n, i)		Vim::Msg("", "Comment"	
\pmw Vim::Msg("", "ErrorMsg") (n, i)		vimMsg(, waining	
	\pmw	Vim::Msg("", "ErrorMsg	

	<u>D</u> ocun	nentation
\dcc	table-of-contents, chapter	(n, i)
\dcs	table-of-contents, section	(n, i)
\dcu	table-of-contents, subsection	(n, i)
\dtc	text, chapter	(n, i)
\dts	text, section	(n, i)
\dtu	text, subsection	(n, i)
\df	function description	(n, i)
\de	example	(n, v, i)
		$S\underline{n}ippet$
\nr	read code snippet	(n, i)
\nw	write code snippet	(n, v, i)
\ne	edit code snippet	(n, i)
\ntl	edit local templates	(n, i)
\ntc	edit custom templates	(n, i)
\ntp	edit personal templates	(n, i)
\ntr	reread the templates	(n, i)
\ntw	template setup wizard	(n, i)
\nts	choose template style	(n, i, T)
		$\underline{R}un$
\rh	hardcopy buffer to postscript	(n, v, i)
\rs	plugin settings	(n,i)
	•	