

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

<i>Bash</i>		
<code>\bps</code>	parameter substitution (list)	(n, i)
<code>\bsp</code>	special parameters (list)	(n, i)
<code>\ben</code>	environment (list)	(n, i)
<code>\bbu</code>	builtins (list)	(n, i)
<code>\bse</code>	set options (list)	(n, i)
<code>\bso</code>	shopts (list)	(n, i)
<i>Comments</i>		
<code>[n]\cl</code>	end-of-line comment	(n, i, v)
<code>[n]\cj</code>	adjust end-of-line comments	(n, i, v)
<code>\cs</code>	set end-of-line comment col.	(n)
<code>[n]\cc</code>	code → comment	(n, i, v)
<code>[n]\cu</code>	uncomment code	(n, i, v)
<code>\cfr</code>	frame comment	(n, i)
<code>\cfu</code>	function description	(n, i)
<code>\ch</code>	file header	(n, i)
<code>\cd</code>	date	(n, i)
<code>\ct</code>	date & time	(n, i)
<code>\css</code>	script sections	(n, i)
<code>\ckc</code>	keyword comments	(n, i)
<code>\cma</code>	plug-in macros	(n, i)
<code>\ce</code>	echo "actual line"	(n, i)
<code>\cr</code>	remove echo from actual line	(n, i)

<i>Statements</i>		
<code>\sc</code>	case in ... esac	(n, i)
<code>\sei</code>	elif then	(n, i)
<code>\sf</code>	for in do done	(n, i, v)
<code>\sfo</code>	for ((...)) do done	(n, i, v)
<code>\si</code>	if then fi	(n, i, v)
<code>\sie</code>	if then else fi	(n, i, v)
<code>\ss</code>	select in do done	(n, i, v)
<code>\su</code>	until do done	(n, i, v)
<code>\sw</code>	while do done	(n, i, v)
<code>\sfu</code>	function	(n, i, v)
<code>\se</code>	echo -e ""	(n, i, v)
<code>\sp</code>	printf "%s"	(n, i, v)
<code>\sae</code>	array element \${.[.]}	(n, i, v)
<code>\saa</code>	arr. elem.s (all) \${.[@]}	(n, i, v)
<code>\sas</code>	arr. elem.s (1 word) \${.[*]}	(n, i, v)
<code>\ssa</code>	subarray \${.[@]::}	(n, i, v)
<code>\san</code>	no. of arr. elem.s \${.[@]}	(n, i, v)
<code>\sai</code>	list of indices \${.[*]}	(n, i, v)
<i>Tests</i>		
<code>\ta</code>	arithmetic tests	(n, i)
<code>\tfp</code>	file permissions	(n, i)
<code>\tft</code>	file types	(n, i)
<code>\tfc</code>	file characteristics	(n, i)
<code>\ts</code>	string comparisons	(n, i)
<code>\toe</code>	option is enabled	(n, i)
<code>\tvs</code>	variables has been set	(n, i)
<code>\tfd</code>	file descr. refers to a terminal	(n, i)
<code>\tm</code>	string matches regexp	(n, i)
<i>IO-Redirection</i>		
<code>\ior</code>	IO-redirections (list)	(n, i)
<code>\ioh</code>	here-document	(n, i)

<i>Pattern Matching</i>		
<code>pzo</code>	zero or one, ?()	(n, i)
<code>pzm</code>	zero or more, *()	(n, i)
<code>pom</code>	one or more, +()	(n, i)
<code>peo</code>	exactly one, @()	(n, i)
<code>paе</code>	anything except, !()	(n, i)
<code>ppc</code>	POSIX classes	(n, i)
<code>pbr</code>	\${BASH_REMATCH[0..3]}	(n, i)
<i>Snippets</i>		
<code>\nr</code>	read code snippet	(n, i)
<code>\nv</code>	view code snippet	(n, i)
<code>\nw</code>	write code snippet	(n, i, v)
<code>\ne</code>	edit code snippet	(n, i)
<code>\ntl</code>	edit templates	(n, i)
<code>\ntr</code>	reread templates	(n, i)
<code>\nts</code>	choose style	(n, i)
<i>Run</i>		
<code>\rr</code>	update file, run script	(n, i, v ¹)
<code>\ra</code>	set script cmd. line arguments	(n, i)
<code>\rba</code>	set Bash cmd. line arguments	(n, i)
<code>\rc</code>	update file, check syntax	(n, i)
<code>\rco</code>	syntax check options	(n, i)
<code>\rd</code>	start debugger ¹	(n, i)
<code>\re</code>	make script executable/not exec. ¹	(n, i)
<code>\rh</code>	hardcopy buffer	(n, i, v)
<code>\rs</code>	plug-in settings	(n, i)
<code>\rx</code>	set xterm size ^{1,2}	(n, i)
<code>\ro</code>	change output destination	(n, i)
<i>Help</i>		
<code>\hb</code>	display the Bash manual	(n, i)
<code>\hh</code>	help (Bash builtins)	(n, i)
<code>\hm</code>	show manual (cmd. line utilities)	(n, i)
<code>\hp</code>	help (plug-in)	(n, i)

¹ Linux/U**x only

² GUI only