1 Zsh Line Editor (ZLE) Bindings

These key bindings are active when using ZLE's default `emacs` input mode. Refer to `man zshzle` when using `vicmd` and `viins` input modes. `^` denotes the Ctrl key; `△`denotes the Alt or Meta key.

History Bindings

`^p`	up-line-or-history
`^n`	down-line-or-history
`△p`	history-search-backward
`△n`	history-search-forward
`^r`	history-incremental-search-forward
`^s`	history-incremental-search-backward
`△.`	insert-last-word

Cursor Movement Bindings

`△b`	backward-word
`△f`	forward-word
`^B`	backward-char
`^F`	forward-char
`^A`	beginning-of-line
`^E`	end-of-line

Copy/Paste Bindings

`△d`	kill-word
`^y`	yank
`^_`	undo
`^K`	kill-line
`^U`	kill-whole-line
`^W`	backward-kill-word

Text Modification Bindings

`^v`	quoted-insert
`^t`	transpose-chars
`△t`	transpose-words
`△c`	capitalize-word
`△l`	down-case-word
`△u`	up-case-word
`△^E`	decrease-char
`△^O`	increase-char
`△^P`	increase-number
`△^_`	decrease-number
`△'`	quote-line

Miscellaneous Bindings

`^l`	clear-screen
`△q`	push-line
`△h`	run-help
`△?`	which-command

2 Parameter expansion

`$`*name*
`${`*name*`}`
Basic parameter substitution
`${+`*name*`}`
1 if *name* set, 0 otherwise
`${`*name*`:-`*word*`}`
`$`*name* if non-null, else *word*
`${`*name*`-`*word*`}`
`$`*name* if set, else *word*
(Similar for others with/without colon.)
`${`*name*`:=`*word*`}`
`$`*name* if non-null, else use *word* and set *name* to that
`${`*name*`:=`*word*`}`
Unconditional assignment `${`*name*`:?`*word*`}`
`$`*name* if non-null, else print *word* and exit
`${`*name*`:+`*word*`}`
word if `$`*name* non-null, else nothing
`${`*name*`#`*pattern*`}`
`${`*name*`##`*pattern*`}`
`$`*name* with shortest (longest) match of *pattern* removed from head. Patterns as globbing; original parameter unchanged
`${`*name*`%`*pattern*`}`
`${`*name*`%%`*pattern*`}`
As for `#`, but remove from tail of match
`${`*name*`/`*pattern*`/`*repl*`}`
Substitute longest match of *pattern* by *repl*
`${(S)`*name*`/`*pattern*`/`*repl*`}`
Substitute shortest match
`${`*name*`//`*pattern*`/`*repl*`}`
Substitute all non-overlapping longest matches
`${`*name*`/#`*pattern*`/`*repl*`}`
Subst if *pattern* at start of string
`${`*name*`/%`*pattern*`/`*repl*`}`
Subst if *pattern* at end of string
`${`*name*`:/`*pattern*`/`*repl*`}`
Subst if *pattern* matches entire string
`${#`*spec*`}`
Count length of scalar or words of array
`${^`*spec*`}`
`${^^`*spec*`}`
Turn on (off) **RC_EXPAND_PARAM**
`${=`*spec*`}`
`${==`*spec*`}`
Turn on (off) **SH_WORD_SPLIT**
`${~`*spec*`}`
`${~~`*spec*`}`
Turn on (off) **GLOB_SUBST**