DEVHINTS.IO

Edit

Vim cheatsheet

Proudly sponsored by

Segment Send data to any tool without having to implement a new API every time.

ethical ad by CodeFund

Vim is a very efficient text editor. This reference was made for Vim 8.0.
For shortcut notation, see `:help key-notation`.

Exiting

`:qa`
`:qa!`
`:w`
`:wq / :x`
`:q`
`:q!`
`ZZ`
`ZQ`

Navigating

`h j k l`
`<C-U> / <C-D>`
Words
`b / w`
`e / ge`
Line
`0` (zero)
`^`
`$`

Editing

`a`
`i`
`o`
`O`
`s`
`S`
`C`
`r`

Exiting insert mode

`Esc / <C-[>`	Exit insert mode
`<C-C>`	Exit insert mode, and abort current command

Visual mode

`v`	Enter visual mode
`V`	Enter visual line mode
`<C-V>`	Enter visual block mode
In visual mode	
`d / x`	Delete selection
`s`	Replace selection
`y`	Yank selection (Copy)

See Operators for other things you can do.

Character

Document

`gg`
`G`

`:tabedit [file]`
`:tabfind [file]`
`:tabclose`
`:tabs`
`:tabfirst`

`R`

Clipboard

`x`

`:tablast`

`:tabn`

`:tabp`

Operators

Usage

Operators let you operate in a range of text (defined by mo

d

Operator

Operators list

`d`

`y`

`c`

`>`

`<`

`g~`

`gU`

`gu`

`!`

See `:help operator`

Examples

Combine operators

dd

dw

db

2dd

dip

(in visual mode) d

See: `:help motion.`

Text objects

Usage

Text objects let you operate (with an operator) in or around

v	i
Operator	[i]nside or [a]round

Diff

`gvimdiff file1 file2 [file3]`	See differencies between files, in HMI

Text objects

p
w
s
[({ <
' " `
b
B
t

Examples

vip
vipipipip
yip
yap
dip
cip

See Operators for c

Misc

Folds

zo / zO
zc / zC
za / zA

Navigation

[([{ [<
])
[m

`zv`	
`zM`	Close all
`zR`	
`zm`	
`zr`	
`zx`	Update folds

Uppercase ones are recursive (eg, zO is open recursively).

Windows

`z{height}<Cr>`	Resize pane to {height} lines tall

Case

`~`	Toggle case (Case => cASE)
`gU`	Uppercase
`gu`	Lowercase
`gUU`	Uppercase current line (also gUgU)
`guu`	Lowercase current line (also gugu)

Do these in visual or normal mode.

`[M`

Jumping

`<C-O>`
`<C-I>`
`gf`

Counters

`<C-A>`
`<C-X>`

Tags

`:tag Classname`

Misc

`.`	Repeat last command
`]p`	Paste under the current indentation level

Command line

`<C-R><C-W>`	Insert current word into the command line
`<C-R>"`	Paste from " register
`<C-X><C-F>`	Auto-completion of path in insert mode

Calculator

<C-R>=128/2	Shows the result of the division : '64'

Do this in insert mode.

Exiting with an error

```
:cq
:cquit
```

Works like `:qa`, but throws an error. Great for aborting Git commands.

Marks

` `` `

Text alignment

```
:center [width]
```

See :help formatting

Spell checking

`:set spell spelllang=en_us`
`]s`
`[s`
`z=`

`zg`
`zw`
`zu / C-X (Insert Mode)`

See `:help spell`

Also see

- Vim cheatsheet (vim.rotrr.com)
- Vim documentation (vimdoc.sourceforge.net)
- Interactive Vim tutorial (openvim.com)

▶ **6 Comments** for this cheatsheet. Write yours!

Search 381+ cheatsheets

Over 381 curated cheatsheets, by developers for developers.

Devhints home

Other Vim cheatsheets

Vimdiff
cheatsheet

Vim scripting
cheatsheet

Tabular
cheatsheet

Projectionist
cheatsheet

Vim digraphs
cheatsheet

Vim Easyalign
cheatsheet

Top cheatsheets

Elixir
cheatsheet

ES2015+
cheatsheet

React.js
cheatsheet

Vimdiff
cheatsheet

Vim scripting
cheatsheet

Capybara
cheatsheet