

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

Navigating

Editing

| | | |
|----------|---------------|---|
| :qa | h j k l | a |
| :qa! | <C-U> / <C-D> | i |
| :w | Words | o |
| :wq / :x | b / w | O |
| :q | e / ge | s |
| :q! | Line | S |
| ZZ | 0 (zero) | C |
| ZQ | ^ | r |
| | \$ | |

Exiting insert mode

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

Character

R

Clipboard

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 <u>Operators</u> for other things you can do. | |

gg

G

x

:tabedit [file]

:tabfind [file]

:tabclose

:tabs

:tabfirst

`:tablast``:tabn``:tabp`

Operators

Usage

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

`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

Examples

| | | | | |
|--|-----------------------|---------------------------------------|---------------------|--|
| Text objects let you operate (with an operator) in or around | | p | vip | |
| v | i | w | vipipipip | |
| | | s | yip | |
| | | [({ < | yap | |
| Operator | [i]inside or [a]round | ' " ` | dip | |
| | | b | cin | |
| | | P | See Operators for c | |
| Diff | | t | | |
| gvimdiff file1 file2 [file3] | | See differences between files, in HMI | | |

Misc

Folds

Navigation

| | |
|---------|-------------|
| zo / zO | [([{ [< |
| zc / zC |]) |
| za / zA | [m |

| | |
|--|--------------|
| zV | [M |
| zM | Jumping |
| zR | |
| zm | <C-O> |
| zr | <C-I> |
| zx | gf |
| | Update folds |
| Uppercase ones are recursive (eg, z0 is open recursively). | |
| Counters | |
| | |
| | |

Windows

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

Tags

Case

| | |
|------------------------------------|------------------------------------|
| | :tag Classname |
| ~ | 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. | |

Misc

Marks

`.`

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

Text alignment

Calculator

`<C-R>=128/2``:center {width}`

Shows the result of the division : '64'

Do this in insert mode.

See help formatting

Exiting with an error

Spell checking

`:cq
:cquit`

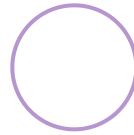
Works like :qa, but throws an error. Great for aborting Git commands.

`: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!](#)



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