Vim cheatsheet

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

Exiting

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

Navigating

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

Exiting insert mode

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

Character

`fc`	
`Fc`	

Clipboard

`x`	Delete character
`dd`	Delete line (Cut)
`yy`	Yank line (Copy)
`p`	Paste
`P`	Paste before
`"*p / "+p`	Paste from system clipboard
`"*y / "+y`	Paste to system clipboard

Window

Find & Replace

| :%s/foo/bar/g | Replace foo with bar in whole document |

| z t |
| z b |
| H |
| M |
| L |

Search

| n |
| N |
| * |
| # |

Tab pages

| :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`

♯ Text objects

Usage

Text objects let you operate (with an operator) in or around tex	
v	i
Operator	[i]nside or [a]round

Diff

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

Text objects

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

♯ Misc

Folds

`zo / zO`
`zc / zC`
`za / zA`
`zv`
`zM`
`zR`
`zm`
`zr`
`zx`

Navigation

%
`[(   [{   [<`
`])`
`[m`
`[M`

Open all

Jumping

Fold more (foldlevel += 1)

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

Uppercase ones are recursive (eg, z0 is open recursively).

`gf`

Counters

`<C-A>`	
Resize pane to {height} lines tall	

Windows

`z{height}<Cr>`	

Case

`~`	
`gU`	
`gu`	
`gUU`	
`guu`	
Do these in visual or normal m	

Tags

`:tag Classname`	Jump to fi
`<C-]>`	
`g]`	
`<C-T>`	
`<C-O> <C-I>`	
`:tselect Classname`	Fin
`:tjump Classname`	Find definitions of C

Marks

Misc

`.`	Repeat last command
`]p`	Paste under the current indentation level
`:set ff=unix`	Convert Windows line endings to Unix line endings

`` `0 ``	
`''`	

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

```
:center [width]
:right [width]
:left
```

See `:help formatting`

`d'a`	
`` d`a ``	

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
```

Spell checking

```
:set spell spellla
```

```
]s
```

```
[s
```

```
z=
```

```
zg
```

```
zw
```

```
zu / C-X (Insert M
```

See `:help spell`

♯ Also see

Vim cheatsheet (vim.rotrr.com)

Vim documentation (vimdoc.sourceforge.net)

Interactive Vim tutorial (openvim.com)

▶   **14 Comments** for this cheatsheet. Write yours!
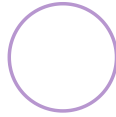
Over 358 curated cheatsheets, by developers for developers.

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

Vue.js
cheatsheet