

my VIM Cheatsheet

Version 0.2

These shortcuts and descriptions are inspired by the awesome guide: "*Onramp to Vim*"[5], an austrian talk script "*Vim für Fortgeschrittene*"[1] and at least the "*Vim Reference Manual*"[3]. Another good ressource for starting with Vim is the online Version of the vimtutor[2]. To consolidate and improve your VIM knowledge, VimGolf [6] is a very good game in the form of a competition. While this cheat sheet is a selection of commands for one page, there are many more online.[4]

Motions (nouns)

Motions are in the sense of movement.

See Vim's help page for motions for a full listing: `:h motion`

Moving within a line

`h, l` move left/right by character
`w` move forward one (w)ord
`b` move to (b)eginning of a word
`e` move forward to the (e)nd of a word

Jumping within a line

`f<char>` (f)ind a character forward in a line and move to it
`t<char>` find a character forward in a line and move un(t)il it
(one character before)
`F<char>` (f)ind a character backward in a line and move to it
`T<char>` find a character backward in a line and move un(t)il it
; repeat last `f`, `t`, `F`, or `T` command
, repeat last `f`, `t`, `F`, or `T` command, but in opposite direction

Moving between lines

`j, k` move up/down one line
`H, M, L` move (H)igh, (M)iddle, or (L)ow within the viewport
/ search
`n` repeat last search - (n)ext
`N` repeat last search in opposite direction
`<ctrl>u, <ctrl>d` moves (u)p or (d)own half of a page
`<NN>G` (G)o to line number *NN*
`<NN>gg` (G)o to line number *NN*
% move cursor to matching character
(default supported pairs: '()', '{}', '[]';
use `:h matchpairs` in vim for more info)

Inserting text

`a` (a)ppend text after the cursor
`A` (a)ppend text at the end of the line
`i` (i)nsert text before the cursor
`I` (I)nsert text before the first non-blank in the line
`o` (o)pen a new line below the current line, append text
`O` (O)pen a new line above the current line, append text

Text Objects (nouns)

Text objects allow you to run the command from anywhere inside the text object

For a full listing, see `:h text-objects`

word

`aw` (a) (w)ord, includes surrounding white space
`iw` (i)nn(er) (w)ord, does not include surrounding white space

Others: `[a|i]` +

`s` (s)entence
`p` (p)aragraph
" "double quoted string"
) (parenthesized block)
`t` <h1>single (t)ag</h1>

Commands (verbs)

After pressing the key-mapping for a given command Vim will wait for you to identify the text you want the command to operate on.

The simplest commands are made by simply repeating the operator a second time to act on the current line.

For example, where `d` is the operator for "delete", `dd` will delete the whole line. Each of `yy`, `cc`, `>>`, `==` behave similarly.

`d` (d)ele(e)te
`c` (c)hange
`y` (y)ank (copy)
`v` (v)isually select
>, < indent, dedent
= reformat (reindent, break long lines, etc.)

The Structure of an Editing Command (sentence)

<number><command><text object or motion>

Examples: Command + Motion

The following table shows some of the many variations of the delete operation you can build by combining the `d` operator mapping with a motion:

`dw` (d)ele(e)te from current position to the next (w)ord
`de` (d)ele(e)te to to the (e)nd of the current word
`d2e` (d)ele(e)te to to the (e)nd of the next word
`dj` (d)ele(e)te down a line (current and one below)
`dt)` (d)ele(e)te forward un(t)il closing paranthesis
`d/world` (d)ele(e)te up until the first search match for *world*
`d$` (d)ele(e)te to (\$) end of line

Foo

bar 1

`"*y{motion}` yank into the system clipboard register
`"*p` paste from the system clipboard register
`<ctrl>r` redo
.
..... repeat last command
`<ctrl>6` toggle last two Buffers
`:messages` startup errors
`:changes` list of changes

Search and replace

`/<pattern>` search for *pattern*
`?<pattern>` search backward for *pattern*
`n` repeat search in same direction
`N` repeat search in opposite direction
`:%s/<old>/<new>/g` replace all *old* with *new* throughout file
`:%s/<old>/<new>/gc` replace all *old* with *new* throughout file
with confirmations
`:noh` nohighlight, removes the highlighting of search results

(Book-)Marks

`m<char>` . set current position for (m)ark *char* within the file/buffer
`m<CHAR>` set current position for (m)ark *CHAR* global
'<char>' jump to position of mark *char* within the file/buffer
`m<CHAR>` jump to buffer and position of mark *CHAR*
:jump list of jumps (:jumps?)
`<ctrl>i` go to newer position in jump list
`<ctrl>o` go to older position in jump list

References

- [1] Guckes. *Vim für Fortgeschrittene*. URL: http://www.guckes.net/talks/vim/vim_advanced.html.
- [2] *Interactive Vim tutorial*. URL: <https://openvim.com/>.
- [3] Bram Moolenaar. *Vim Reference Manual*. URL: <https://vimhelp.org/quickref.txt.html#quickref>.
- [4] Vim Cheat Sheet. *Vim Cheat Sheet*. URL: <https://vim.rtorr.com/>.
- [5] Ben Orenstein & Chris Toomey. *Onramp to Vim*. URL: <https://thoughtbot.com/upcase/onramp-to-vim>.
- [6] *VimGolf*. URL: <https://www.vimgolf.com/>.