Cheatography

Vim Cheat Sheet

by Martinsos via cheatography.com/116342/cs/21764/

Motions

h, j, k, l	one char (left, down, up, right)
0, ^	line start (column 0, first char)
$	line end
f<char>	to next <char> in line
t<char>	f<char>h
T<char>	t<char> but backwards
F<char>	f<char> but backwards
w, e	next word (start, end)
b	prev word (start)
W, E, B	stronger "w", "e", "b"
(,)	start/end of sentence
{, }	start/end of paragraph
%	first or matching parenth
gg, G	file start, file end
Ctrl-u/d	half screen up/down
:<num>	jump to line
<num>G	:<num>
Ctrl-o	undo movement
/<str>	search forward in file
*	/<word_at_position>
?<str>	search backward in file

Motions (cont)

H, M, L	first/middle/last line on screen

There are more motions, but these are the basic ones.

When used alone, motion moves the cursor.
It can also be used with operators.

When it makes sense, you can prefix motion with a count: e.g. "3tg" moves to third occurrence of g.

";" after "t <char>" or "f <char>" does one more search.
"n" is next and "N" is previous when doing search with "/" or "?".

Commands (Substitute)

`:<range>s/<old>/<new>/[flags]`
`range` -> nothing (in line), `#,#` (between lines), `%` (whole file).
`flags` -> `g` (all occurrences), `c` (ask for confirmation), ...
`:s/thee/the` -> in line, first occurrence
`:1,10s/thee/the/g` -> from lines 1 to 10, all occurrences
`:%s/thee/the/gc` -> whole file, all occurrences, ask for confirmation

Concept: extending

Listed here are motions, operators, text-objects and commands that come with vim. However, plugins (or you) can define new ones, bringing even more options!

Operators

c	change (delete and enter insert mode)
d	delete
y	yank (copy)
gu	make lowercase
gU	make uppercase
g~	toggle case
>	shift right
<	shift left
gq	text formatting

There are few other operators, but they are not used much.

Operators can be used as:
[count]operator([count](motion | text-object))
e.g. d, 3d, d$, dw, d3w, 2d3w .

Combining operator with motion executes operator from current point to point where motion takes us.
Combining operator with text object executes operator on that text object.
Numbers (count) execute operator multiple times.

Typing an operator twice, e.g. dd or <<, usually executes the operator on current line.

Commands (Change/Insert)

i, I	insert at point / line start
a, A	append after point / line end
o, O	open line below / above
r, R	rewrite one / many char(s)
C	c$
cc	c_

All these commands enter insert mode.

By **Martinsos**
cheatography.com/martinsos/

Published 15th February, 2020.
Last updated 17th February, 2020.

Cheatography

Vim Cheat Sheet

by Martinsos via cheatography.com/116342/cs/21764/

Commands (:)

:!<cmd>	executes in external shell
:wqa	save all files and quit
:h<smth>	help

Concept: composing

One of the best things in vim is composing its operators with motions and text-objects! Commands can't be composed.
Here we will list some examples (check "-Operators" for exact definition of composition):
di(-> delete all inside ().
3tp or tp;; -> move to third occurence of p in line.
cip -> delete current paragraph and enter insert mode.
<2ib -> shift current block two times left.

Text objects

w	word
s	sentence
p	paragraph
', ", `	quotes
(, [, {, <	parentheses
t	tag
_	current line (*)

Text objects are not to be used alone, but in combination with e.g. operators.

Text objects can be combined with modifiers "a" (around) and "i" (inside) to create new text objects, e.g. iw (word we are inside of), or aw (word and surrounding spaces).

(*) _ is not really a text object, but it behaves mostly like one.

Commands (Deletion)

x, X	delete one char (after, before)
D	d$
dd	d_
J	join current with next line

Commands (Visual)

v	select from point
V	select from line
Ctrl-v	select from point as block

All these commands enter visual mode.

Commands (Other)

yy	y_
p	paste
.	repeat last command
u	undo
Ctrl-r	redo
zz	center screen on point
za, zm, zr	code folding

Concept: modes

While in normal mode, we can do all the fun stuff: commands, operators, movements, and entering other modes.
While in insert mode, we are inserting text.
While in visual mode, we are selecting text.

Credits

These resources helped a lot in creating this cheat sheet:
- http://ismail.badawi.io/blog/2014/04/23/the-compositional-nature-of-vim/
- https://takac.github.io/2013/01/30/vim-grammar/

By **Martinsos**
cheatography.com/martinsos/

Published 15th February, 2020.
Last updated 17th February, 2020.