

Cursor movement and text motions	
h/j/k/l	motion left/up/down/right
f / F char	motion to char in current line
t / T char	motion 1 before/after char in current line
w / B	motion to start of next / prev word
e / ge	motion to end of next / prev word
/[regex]	search
0	motion to start of line
^ / ¢	motion to start of line text
\$	motion to end of line
[number]	go to line at [number]
G / gg	
gg	motion to start of buffer
G	motion to end of buffer
o (visual mode)	move cursor to start/end of selection
(/)	start / end of phrase
{ / }	start / end of paragraph
[{]/[})	motion to next / prev { } ()
<c-o> / <c-i>	navigate cursor location history
:goto [number]	go to byte at position <i>number</i> in buffer
* / #	search next / prev occurrence of word under cursor
<ldr>ta	jump to tag <i>user</i>
<c-j> / <c-k>	pagedown / pageup <i>user</i>
<ldr>e	shortcut jump to word <i>user/plugin:easy motion</i>
Cursor movement and text motions (cont)	
<ldr><-	shortcut jump to <i>char</i> bidirectional <i>plugin:easy motion</i>
ldr>s [char]	
<ldr><-	shortcut jump to end of word forward <i>plugin:easy motion</i>
ldr>e	
<ldr><-	shortcut jump to end of word backward <i>plugin:easy motion</i>
ldr>ge	
[m /]m	jump to prev / next function
Yank/Paste	
register: "*" system clipboard (Windows)	
register: "+" system clipboard (Linux)	
register: "%" buffer name	
register: "= output of system()	
:reg list of registers	
[register]y[motion]	yank (yy for current line)
p / P	paste at the / before current position
:pu / <ldr>, line <i>user</i>	paste below current line
gp	paste and move the cursor to the end
<ldr>yy	show yankring <i>user/plugin:yankring</i>
<c-p> (after pasting)	cycle between recent yanks <i>plugin:yankring</i>
<ldr>V	select last pasted text <i>user</i>
<c-rp> register	yank from register in insert mode
My Plugins	
Pathogen	to manage plugins
Yankring	yank between different instances of Vim. Cycle when pasting
Project	simple project management. \C to create a new project, \R to update files
NERD_C-ommenter	easily comment/uncomment code
Matchparen	shows matching delimiter
Csapprox	use GVim colorschemes in console Vim
Airline	awesome status bar
Tagbar	jumpable tag list
Vimwiki	personal Wiki
Syntastic	Validate syntax for several languages
EasyMotion	jump to any word/line/character with few keystrokes
Unite	Fuzzy search for buffers/files/etc
Matchit	improves Vim's % command to understand more delimiters
MatchT-agAlways	highlight matching HTML tags
Tabular	align things
Emmet	Quickly edit HTML and move between tags
YouCompleteMe	excellent completion
VimColorSchemes	lot of colorschemes
Surround	operations on delimiters



By **Juanjo Álvarez** (juanjux)
cheatography.com/juanjux/
juanjoalvarez.net

Published 17th July, 2014.
 Last updated 12th May, 2016.
 Page 1 of 4.

Sponsored by **ApolloPad.com**
 Everyone has a novel in them. Finish
 Yours!
<https://apollopadd.com>

My Plugins (cont)

Jdaddy	add a JSON text object
GTFO	shortcuts to open GUI file explorer or terminal in buffer CWD
Ack	:Ack search (but use ag instead of Ack!)
Repeat	the repeat operator (dot) will be able to repeat some plugin actions
LustyJ-ugger	Show buffers as a list, switch with a number
Rename	:Rename current buffer and the associated file

Operators: [repeat]operator[motion]

d	delete, dd for line
c	change (delete and enter insert), cc for line
y	yank (yy for line)
gu / gU	to lowercase / uppercase
!	filter through external program
=	auto indent
gw	justify
> / <	indent / dedent

[repeat] [operator] [motion or text object]
 Examples:
 10dj (delete 10 lines down)
 v3w (select 3 words ahead)
 yiw (yank the inner-word object)
 gwip (justify the inner-paragraph object)
 3diw (delete the next 3 inner-words)

Text edition (see also operators & motions)

i	enter insert mode
a	enter insert mode after cursor position (append)
x	delete char under cursor
r[newchar]	replace char under cursor
I	move to start of line and insert
A	move to end of line and append
D	delete from cursor to end of line
C	change from cursor to end of line
o / O	create new line and insert below / above cursor position
<ldr>o / <ld-r>O	create new line above / below without going to insert mode
:tab	tabularize from [char]left
/[char]l1	align 1 space <i>plugin:tabularize</i>
ys[motion][delimiter]	surround motion/object with delimiter <i>plugin:surround</i>
S[visual selection][delimiter]	surround visual selection with delimiter
cs[delimiter][newdelim]	change surrounding delimiter for newdelim <i>plugin:surround</i>

Text edition (see also operators & motions) (cont)

ds[delimiter]	delete surrounding delimiter <i>plugin:surround</i>
<ldr>c<s-pace>	comment / uncomment
J	join lines
[operator]<ldr>e	shortcut operate by lines with <i>plugin:easymotion</i>
<c-w>	delete one word back (insert mode)

Ex Commandline

range: %	full buffer
range: . (dot)	current line
range: \$	last line
range: +-N	+N lines from current position
selector: g	global command (selects lines matching regexp)
selector: v	inverse (select lines NOT matching regexp)
operator: d	delete
operator: ![program]	replaces with output from external program
operator: s/old/new	replaces old with new
operator: y	yank
operator: normal [commands]	run commands over every line in the range or selection



By **Juanjo Álvarez** (juanjux)
cheatography.com/juanjux/
juanjoalvarez.net

Published 17th July, 2014.
 Last updated 12th May, 2016.
 Page 2 of 4.

Sponsored by **ApolloPad.com**
 Everyone has a novel in them. Finish Yours!
<https://apollopadd.com>

Ex Commandline (cont)

operator: put put the register content
[register] below the current line

The format is one of:

:**[range][selector][regexp][operation]**

:**[range][operation]**

:**[operation]**

Example:

:20,30g/FIXME/d => delete lines from 20 to 30 with the string FIXME

.:,\$v/DONTDELETE/d => delete lines without the string DONTDELETE from current position to end of buffer.

:%s/old/new => replace all instances of old with new

:.!ls => insert output of ls replacing current line

Emmet plugin for HTML

<c-y> expand HTML abbreviation
(insert mode) like: `Div#page>div.logo+-ul>li5*>a`

<c-y> , wrap selection inside tag (asks for tag)
(visual mode)

<c-y> n/N jump to next / prev editable point

<c-y> k remove current tag

<c-y> / comment tag

<c-y> a converts URL to <A> tag

<c-y> A converts URL to <A> tag autoloading site description

<c-y> c prettify code with CSS classes by type

<c-y> d select tag surrounding the cursor

<c-y> D select next inside tag or next tag

<c-y> i update image size

<c-y> j join separate block: `<p></p> => <p>`

<c-y> , with "lorem" in insert mode, writes lorem ipsum

Spellcheck

<ldr>ss spanish spell check *user*

<ld> english spell check

r>se

<ld-> disable spell check

r>sn

z= suggestions for word under cursor

js / [s] next / prev misspelled word

zg add word to user dictionary

Text Objects i/a[object]

l char

w/W word / WORD

s sentence

(/) or (block)

b

{/} or {block}

B

j JSON block, use gw to prettify
plugin:jdaddy

p paragraph

Windows / Buffers / Tabs / Splits

:w save buffer

:q:x exit / save and exit

:e [path] open file at path, use "." for netrw browser

:bd delete buffer (doesn't delete file)

<ldr>v create vertical split and jump to it *user*

<ldr>tn new tab *user*

<c-h> / move to left / right tab *user*

<c-l>

[number]gt go to tab [number]

:sp create horizontal split

Windows / Buffers / Tabs / Splits (cont)

<ldr>cv close current split *user*

<ldr>bt open all buffers in tabs *user*

:tabm [pos] move tab to pos (can be +/- pos)

<space><s-pace> Unite buffers/MRU *user/plugin:unite*

<space>b Unite buffers *plugin:unite*

<space>m Unite most recently used *plugin:unite*

<space>f Unite search files recursively *plugin:unite*

<c-w>T Convert split to tab

<c-w>r Rotate split positions

:tabo close all tabs except current

:w!! save buffer as root with sudo *user*

Marks

m[mark] create mark [mark]

'[mark] jump to start of line with [mark]

`[mark] | i jump to [mark]

mark: 0 last file edited (zero)

mark: ' before last jump

mark: , last edition point

Undo / Redo

u undo

<c-r> redo

:earlier [time spec] s:second / m:minute / h:hour / d:days

:later [time spec] s:second / m:minute / h:hour / d:days



By **Juanjo Álvarez** (juanjux)
cheatography.com/juanjux/
juanjoalvarez.net

Published 17th July, 2014.
Last updated 12th May, 2016.
Page 3 of 4.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
<https://apollopadd.com>

Misc	
<code><c-o></code>	run a command in insert mode <i>(insert mode)</i>
<code><ldr>cdt</code>	update D tags <i>user</i>
<code><ldr>ct</code>	clear trailing whitespace <i>user</i>
<code><ldr>sv</code>	reload .vimrc <i>user</i>
<code><ldr>rr</code>	syntax on and off (to reset colors) <i>user</i>
<code>za / zA</code>	toggle fold / class level
<code>gof</code>	open graphical file browser in buffer cwd <i>plugin:gtfo</i>
<code>got</code>	open command terminal in buffer cwd <i>plugin:gtfo</i>
<code><ldr><space></code>	clear search result highlights <i>user</i>
<code><ldr>gs</code>	save vim session <i>user</i>
<code><ldr>css</code>	load vim session <i>user</i>
<code>v / V</code>	enter visual mode / by line
<code><c-v></code>	visual block selection (I to insert before, A to append after)
<code><ldr>tb</code>	open tag bar split <i>user/plugin:-tagbar</i>
<code><ldr>P</code>	open project split <i>user/plugin:-project</i>
<code><ldr>sy</code>	syntactic check with syntactic <i>user/plugin:syntactic</i>
<code>:Ack</code>	search with Ack/ag

Misc (cont)	
<code>:windo diffthis</code>	mark split for windiff (do twice)
<code>. (dot)</code>	repeat last change
<code>:sort [selection]</code>	sorts the selection
<code>:Rename</code>	rename current buffer and file <i>plugin:rename</i>
<code><ldr>lj</code>	show buffer list, switch with a number <i>plugin:lusty-juggler</i>
<code>@:</code>	repeat last ex mode command
<code>[Bash Shell]</code>	put bash commandline in vi mode
<code>set -o vi</code>	edit shell command in Vim
<code>[Bash shell in vi mode] <esc>c>v</code>	

Tricks	
<code>xp</code>	in normal mode to exchange two chars
<code>ddp</code>	to exchange two lines (or lower the current one one down)
In the Bash shell, use <code>set -o vi</code> to enable Vi mode. Then use <code><esc></code> and edit your command like in Vi or use <code>v</code> to edit your command line in Vim	



By **Juanjo Álvarez** (juanjux)
cheatography.com/juanjux/
juanjoalvarez.net

Published 17th July, 2014.
 Last updated 12th May, 2016.
 Page 4 of 4.

Sponsored by **ApolloPad.com**
 Everyone has a novel in them. Finish Yours!
<https://apollopadd.com>