Vim Visual Cheat Sheet

Serge Y. Stroobandt

> **This document is still under construction.**

Introduction

Vim (and `vi` for that matter) is renowned for its notoriously steep learning curve.

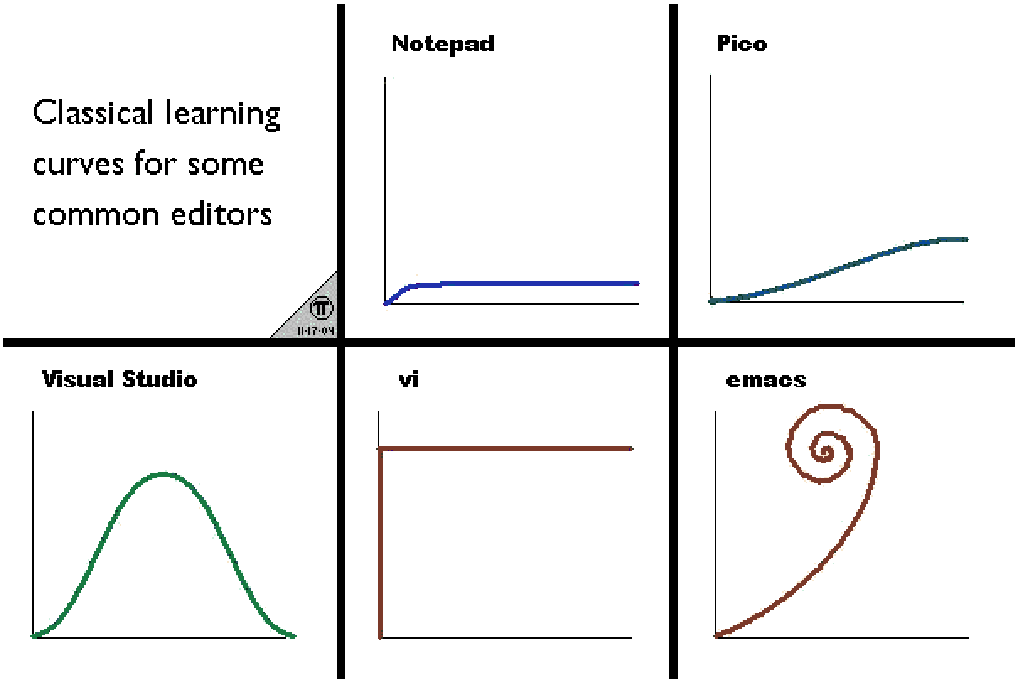


Figure 1: Comparison of editor learning curves *Source:*

This should not scare you off. If you have not done so yet, first read *"Why I Use Vim"* If you cannot bother reading it, here is my personal break down.

No, this is not an extreme case of fanboyhood. Vim is a way of life, or rather, a means towards extreme levels of productivity with the Vim–Markdown–Pandoc–make input combo with any of the LaTeX/ConTeXt/XHTML/Prince/DZSlides back-ends.

Among other things, Vim will teach you to use a keyboard professionally; keeping your index fingers stuck to the home row. Vim allows you to edit texts lightning fast without resorting to any computer mouse or other pointing device. Imagine the space you will gain on the folding table on your next transatlantic business flight! You will no longer have to use the Home, End or arrow keys, if you chose so.

vi gang sign

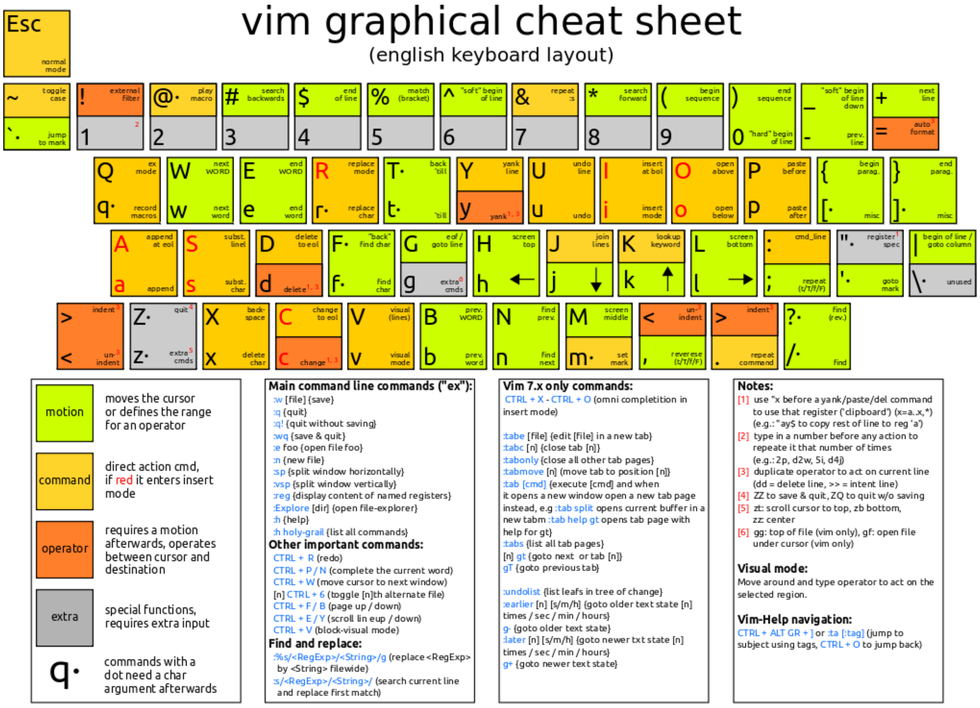
Vim's predecessor `vi` has been around since 1976 and has been continuously improved ever since. If computer application has been around for such an extended period in time, for sure, there must be a good reason to it! Are you not curious towards experiencing for yourself why?

My interest in Vim grew out of my annoyance with other text editors not recognising words, parenthesis, brackets, etc. I got fed up with manually tapping the cursor to the desired editing position. Vim is one of the few text editors which is a word processor in the true sense of the word. It is a word processor which effectively recognises word, bracket and parenthesis boundaries, counts words, etc.

Vim continuously records a small macro of your last command combination, which is easily accessible through the dot . key. Don't be deceived; this functionality is far more powerful than the usual repeat function!

Visual Cheat Sheet

For English keyboards



motion — moves the cursor or defines the range for an operator

command — direct action cmd, if red it enters insert mode

operator — requires a motion afterwards, operates between cursor and destination

extra — special functions, requires extra input

q· — commands with a dot need a char argument afterwards

Main command line commands ("ex"):

:w [file] {save}
:q {quit}
:q! {quit without saving}
:wq {save & quit}
:e foo {open file foo}
:n {new file}
:sp {split window horizontally}
:vsp {split window vertically}
:reg {display content of named registers}
:Explore [dir] {open file-explorer}
:h {help}
:h holy-grail {list all commands}

Other important commands:

CTRL + R (redo)
CTRL + P / N (complete the current word)
CTRL + W (move cursor to next window)
[n] CTRL + 6 (toggle [n]th alternate file)
CTRL + F / B (page up / down)
CTRL + E / Y (scroll lin eup / down)
CTRL + V (block-visual mode)

Find and replace:

:%s/<RegExp>/<String>/g (replace <RegExp> by <String> filewide)
:s/<RegExp>/<String>/ (search current line and replace first match)

Vim 7.x only commands:

CTRL + X - CTRL + O (omni completition in insert mode)

:tabe [file] {edit [file] in a new tab}
:tabc [n] {close tab [n]}
:tabonly {close all other tab pages}
:tabmove [n] (move tab to position [n]}
:tab [cmd] {execute [cmd] and when it opens a new window open a new tab page instead, e.g :tab split opens current buffer in a new tabm :tab help gt opens tab page with help for gt}
:tabs {list all tab pages}
[n] gt {goto next or tab [n]}
gT {goto previous tab}

:undolist {list leafs in tree of change}
:earlier [n] [s/m/h] {goto older text state [n] times / sec / min / hours}
g- {goto older text state}
:later [n] [s/m/h] {goto newer txt state [n] times / sec / min / hours}
g+ {goto newer text state}

Notes:

[1] use "x before a yank/paste/del command to use that register ('clipboard') (x=a..x,*) (e.g.: "ay$ to copy rest of line to reg 'a')
[2] type in a number before any action to repeate it that number of times (e.g.: 2p, d2w, 5i, d4j)
[3] duplicate operator to act on current line (dd = delete line, >> = intent line)
[4] ZZ to save & quit, ZQ to quit w/o saving
[5] zt: scroll cursor to top, zb bottom, zz: center
[6] gg: top of file (vim only), gf: open file under cursor (vim only)

Visual mode:

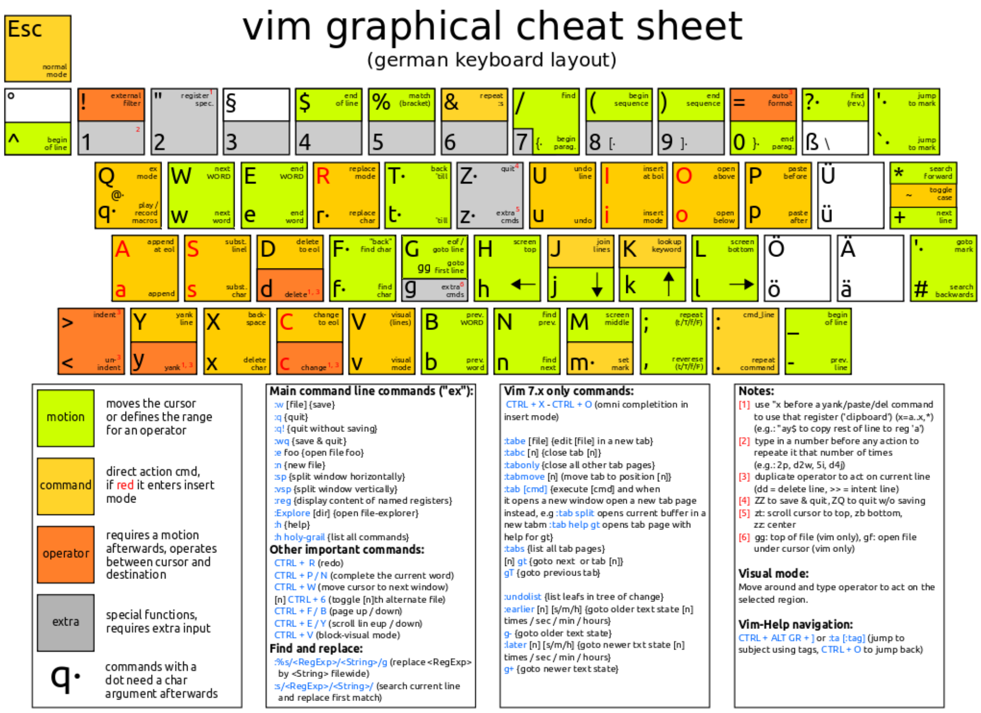
Move around and type operator to act on the selected region.

Vim-Help navigation:

CTRL + ALT GR +] or :ta [:tag] (jump to subject using tags, CTRL + O to jump back)

Figure 2: Vim visual cheat sheet for English keyboards

For German keyboards


vim graphical cheat sheet
(german keyboard layout)
Esc normal mode
° ^ begin of line
! external filter 1 2
" register spec. 1 2
§ 3
$ end of line 4
% match (bracket) 5
& repeat :s 6
/ find 7 {· begin parag.
(begin sequence 8 [·
) end sequence 9]·
= auto format 3 0 }· end parag.
?· find (rev.) ß \
'· jump to mark `· jump to mark
Q ex mode @· q· play / record macros
W next WORD w next word
E end WORD e end word
R replace mode r· replace char
T· back 'till t· 'till
Z· quit 4 z· extra cmds 5
U undo line u undo
I insert at bol i insert mode
O open above o open below
P paste before p paste after
Ü ü
* search forward ~ toggle case + next line
A append at eol a append
S subst. linel s subst. char
D delete to eol d delete 1, 3
F· "back" find char f· find char
G eof / goto line gg goto first line g extra cmds 6
H screen top h ←
J join lines j ↓
K lookup keyword k ↑
L screen bottom l →
Ö ö
Ä ä
'· goto mark # search backwards
> indent 3 < un-indent 3
Y yank line y yank 1, 3
X back-space x delete char
C change to eol c change 1, 3
V visual (lines) v visual mode
B prev. WORD b prev. word
N find prev. n find next
M screen middle m· set mark
; repeat (t/T/f/F) , reverese (t/T/f/F)
: cmd_line . repeat command
_ begin of line - prev. line
motion
command
operator
extra
q·

motion — moves the cursor or defines the range for an operator

command — direct action cmd, if red it enters insert mode

operator — requires a motion afterwards, operates between cursor and destination

extra — special functions, requires extra input

q· — commands with a dot need a char argument afterwards

Main command line commands ("ex"):

:w [file] {save}
:q {quit}
:q! {quit without saving}
:wq {save & quit}
:e foo {open file foo}
:n {new file}
:sp {split window horizontally}
:vsp {split window vertically}
:reg {display content of named registers}
:Explore [dir] {open file-explorer}
:h {help}
:h holy-grail {list all commands}

Other important commands:

CTRL + R (redo)
CTRL + P / N (complete the current word)
CTRL + W (move cursor to next window)
[n] CTRL + 6 (toggle [n]th alternate file)
CTRL + F / B (page up / down)
CTRL + E / Y (scroll lin eup / down)
CTRL + V (block-visual mode)

Find and replace:

:%s/<RegExp>/<String>/g (replace <RegExp> by <String> filewide)
:s/<RegExp>/<String>/ (search current line and replace first match)

Vim 7.x only commands:

CTRL + X - CTRL + O (omni completition in insert mode)

:tabe [file] {edit [file] in a new tab}
:tabc [n] {close tab [n]}
:tabonly {close all other tab pages}
:tabmove [n] (move tab to position [n]}
:tab [cmd] {execute [cmd] and when it opens a new window open a new tab page instead, e.g :tab split opens current buffer in a new tabm :tab help gt opens tab page with help for gt}
:tabs {list all tab pages}
[n] gt {goto next or tab [n]}
gT {goto previous tab}

:undolist {list leafs in tree of change}
:earlier [n] [s/m/h] {goto older text state [n] times / sec / min / hours}
g- {goto older text state}
:later [n] [s/m/h] {goto newer txt state [n] times / sec / min / hours}
g+ {goto newer text state}

Notes:

[1] use "x before a yank/paste/del command to use that register ('clipboard') (x=a..x,*) (e.g.: "ay$ to copy rest of line to reg 'a')
[2] type in a number before any action to repeate it that number of times (e.g.: 2p, d2w, 5i, d4j)
[3] duplicate operator to act on current line (dd = delete line, >> = intent line)
[4] ZZ to save & quit, ZQ to quit w/o saving
[5] zt: scroll cursor to top, zb bottom, zz: center
[6] gg: top of file (vim only), gf: open file under cursor (vim only)

Visual mode:

Move around and type operator to act on the selected region.

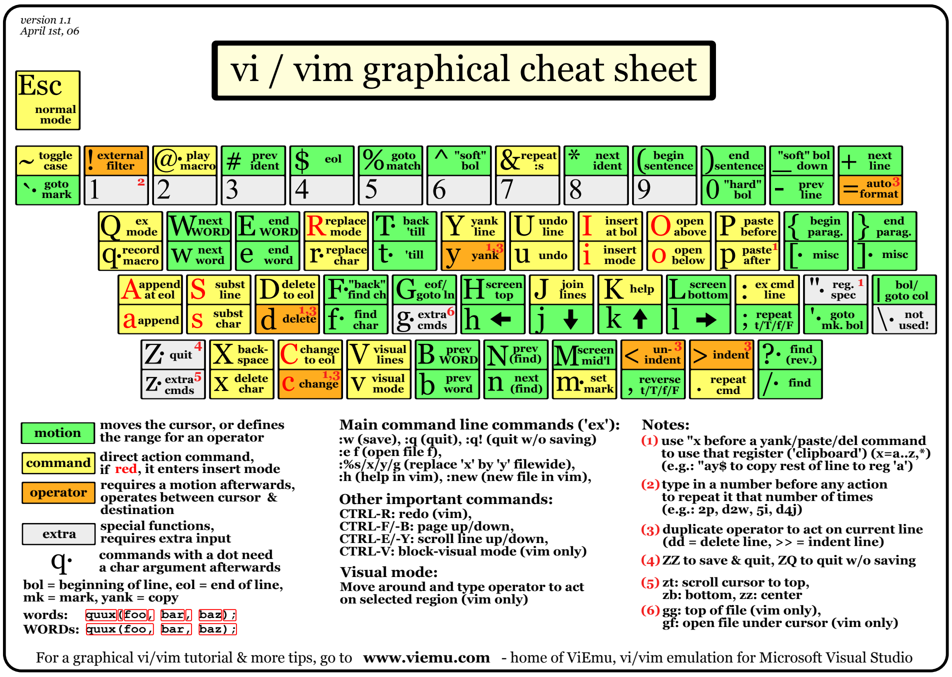
Vim-Help navigation:

CTRL + ALT GR +] or :ta [:tag] (jump to subject using tags, CTRL + O to jump back)

Figure 3: Vim visual cheat sheet for German keyboards

Tutorial

vi/vim graphical cheat sheet tutorial



motion: moves the cursor, or defines the range for an operator

command: direct action command, if red, it enters insert mode

operator: requires a motion afterwards, operates between cursor & destination

extra: special functions, requires extra input

q· commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: quux(foo, bar, baz);
WORDs: quux(foo, bar, baz);

Main command line commands ('ex'):
:w (save), :q (quit), :q! (quit w/o saving)
:e f (open file f),
:%s/x/y/g (replace 'x' by 'y' filewide),
:h (help in vim), :new (new file in vim),

Other important commands:
CTRL-R: redo (vim),
CTRL-F/-B: page up/down,
CTRL-E/-Y: scroll line up/down,
CTRL-V: block-visual mode (vim only)

Visual mode:
Move around and type operator to act on selected region (vim only)

Notes:
(1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*) (e.g.: "ay$ to copy rest of line to reg 'a')
(2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)
(3) duplicate operator to act on current line (dd = delete line, >> = indent line)
(4) ZZ to save & quit, ZQ to quit w/o saving
(5) zt: scroll cursor to top, zb: bottom, zz: center
(6) gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to **www.viemu.com** - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Figure 4: Vi/vim graphical cheat sheet tutorial

Vim game

VIM Adventures

From here on

By now, you might have Vim is actually a word processor which you can easily modify and/or extend yourself.

In other applications

Unattended CSS typesetting with Prince.

This work is published at https://hamwaves.com/vim.tutorial/en/.

Last update: Monday, March 1, 2021.