VI/VIM-Befehlsübersicht

Version 1.23 — 19.2.2016

© 2005–2016 T. Birnthaler, OSTC GmbH

Die Informationen in diesem Skript wurden mit größter Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden. Der Autor übernimmt keine juristische Verantwortung oder irgendeine Haftung für eventuell verbliebene fehlerhafte Angaben und deren Folgen.

Alle Rechte vorbehalten einschließlich Vervielfältigung, Übersetzung, Mikroverfilmung sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Für Dokumente und Programme unter dem Copyright der OSTC GmbH gilt:

- Dürfen heruntergeladen und im privaten Bereich frei verwendet werden.
- Kommerzielle Nutzung bedarf der vorherigen Zustimmung durch die OSTC GmbH.
- Titelseite und Copyright-Hinweise darin dürfen nicht verändert werden.

Hinweise auf inhaltliche Fehler, Schreibfehler und unklare Formulierungen sowie Ergänzungen, Kommentare, Wünsche und Fragen können Sie gerne an den Autor richten:

OSTC Open Source Training and Consulting GmbH
Thomas Birnthaler
E-Mail: tb@ostc.de
Web: www.ostc.de

Inhaltsverzeichnis

1 Vi **3**
1.1 Die wichtigsten Vi-Befehle . . . 3
1.2 Weitere wichtige Vi-Befehle . . . 4
1.3 Wichtige allgemeine Vi-Optionen . . . 4
1.4 Wichtige Vi-Optionen für Programmierer . . . 4
1.5 Wichtige Vim-Optionen . . . 5
1.6 Nützliche Vi-Makros . . . 5

© 2005–2016 T. Birnthaler, OSTC GmbH

1 Vi

1.1 Die wichtigsten Vi-Befehle

~/.exrc ~/.vimrc	Vi/Vim-Konfigurationsdatei [**ex**/**vim resource**]
ESC	Kommando-Modus beenden (2x → piepst + ignoriert)
ESC :q! CR	Vi sicher verlassen (ohne Änderung an Datei)
vi [FILE] CR	Datei FILE bzw. leere Datei (:w FILE *nötig!*) editieren
:wq CR :x CR ZZ	Editierte Datei schreiben + Vi verlassen [**write+quit**/**exit**]
:q CR :q! CR	Editierte Datei nicht schr. + Vi verlassen (!*=erzwingen!*) [**quit**]
:w CR :w FILE CR	Editierten Text (auf Datei FILE) schreiben [**write**]
:e[!] FILE CR	Datei FILE statt aktueller editieren (!*=erzwingen!*) [**edit**]
:f FILE CR	Aktuelle Datei mit dem Namen FILE benennen [**file**]
:n CR	Nächste Datei editieren (falls mehrere angegeben) [**next**]
:prev :N VIM!	Vorherige Datei editieren [**previous**/**Next**]
:args CR	Alle beim Aufruf des Vi angeg. Dateien anzeigen [**arguments**]
:rew CR	Zur 1. beim Aufruf angegeb. Datei zurückspringen [**rewind**]
a TEXT... ESC	Nach Cursorposition TEXT... einfügen [**append**]
A TEXT... ESC	Am Zeilenende TEXT... einfügen [**Append**]
i TEXT... ESC	Vor Cursorposition TEXT... einfügen [**insert**]
I TEXT... ESC	Am Zeilenanfang TEXT... einfügen [**Insert**]
o TEXT... ESC	Neue Zeile TEXT... unter aktueller Zeile einfügen [**open**]
O TEXT... ESC	Neue Zeile TEXT... über aktueller Zeile einfügen [**Open**]
x X	Ein Zeichen unter/vor Cursor löschen [**crossout**]
dd dw dMOVE	Zeile/Wort/gemäß Bewegung MOVE löschen [**delete**]
rX	Ein Zeichen unter Cursor mit X übertippen (*kein* ESC*!*) [**replace**]
R TEXT... ESC	Ab Cursorposition mit TEXT... übertippen [**Replace**]
s TEXT... ESC	Ein Zeichent unter Cursor mit TEXT... übertippen [**substitute**]
cc cw cMOVE ESC	Zeile/Wort/gemäß Bewegung MOVE ändern [**change**]
yy yw yMOVE ESC	Zeile/Wort/gemäß Bew. MOVE in Zw.speicher kopieren [**yank**/**copy**]
p P	Zwischensp. nach/vor akt. Zeichen/Zeile einfügen [**put**]
h l	Zeichen links/rechts (auch <BACKSPACE>/<SPACE>)
j k	Zeile ab/auf (auch +/−/<RETURN>)
w b e	Wort vor/zurück/Wortende [**word**/**backword**/**endword**]
W B E	WORT vor/zurück/WORTende (*inkl. Ziffern, Satzzeichen!*)
0 ^ $ NN|	Zum Zeilenanfang/1.Buchstaben/Zl.ende/Spalte NN springen
<Strg-F> <Strg-B>	Bildschirmseite auf/abblättern [**forward**/**backward**]
G 1G NNG	Zum Dateiende/Dateianfang/Zeile NN springen [**Go**]
mX 'X `X	Marke X (a-z) setzen, Zeile/Zeichen mit M. X anspringen [**mark**]
/SUCH CR ?SUCH CR	Text SUCH vorwärts/rückwärts suchen
:%s/SUCH/TEXT/gc CR	Text SUCH überall in Datei durch TEXT ersetzen (%=1,$=alle Zeilen, s=substitute, g=global, c=confirm)
n N	Letzte Suche vor/zurück wiederholen [**next**]
u U	Letzte Änderung/alle Änd. einer Zeile zurücknehmen [**undo**]
<Strg-R> VIM!	Letzte Änderung wiederherstellen (2x u im Vi) [**redo**]
.	Letztes Edit-Kommando (Änderung) wiederholen [**Punkt**]

Mit VIM! markierte Befehle sind nur im Editor Vim möglich.

© 2005–2016 T. Birnthaler, OSTC GmbH

1.2 Weitere wichtige Vi-Befehle

%	Passende Klammer zu { [(< >)] } suchen
~	Groß/Kleinschreibung von aktuellem Zeichen vertauschen
J	Nachfolgende Zeile an aktuelle anhängen [**join**]
<< >>	Aktuelle Zeile einrücken/ausrücken (Option shiftwidth)
xp dwwP ddp	Zwei Zeichen/Worte/Zeilen vertauschen
v V <Strg-V> VIM!	Block zeichen/zeilen/spaltenorientiert markieren [**visual**]
:e# <Strg-6>	Vorherige Datei editieren
"XyMOVE	Von MOVE ausgewählten Text in Puffer X (a-z) kopieren [**yank**]
"XdMOVE	Von MOVE ausgewählten Text in Puffer X (a-z) verschieben [**del**]
"XP	Inhalt von Puffer X (a-z) vor Cursor/aktueller Zeile einfügen [**put**]
"Xp	Inhalt von Puffer X (a-z) nach Cursor/aktueller Zeile einfügen
:map MACRO TEXT	MACRO mit Ersatztext TEXT für Kommando-Modus definieren
:unmap MACRO	MACRO mit Ersatztext TEXT für Kommando-Modus löschen
:map	Alle Makros für Kommando-Modus anzeigen
:map! MACRO TEX	MACRO mit Ersatztext TEXT für Eingabe-Modus definieren
:unmap! MACRO	MACRO mit Ersatztext TEXT für Eingabe-Modus löschen
:map!	Alle Makros für Eingabe-Modus anzeigen
:ab ABBREV TEXT	Abkürzung ABBREV für Eingabe-Modus definieren (*Leerzeichen!*)
:unab ABBREV	Abkürzung ABBREV für Eingabe-Modus löschen
:ab	Alle Abkürzung für Eingabe-Modus anzeigen

1.3 Wichtige allgemeine Vi-Optionen

errorbells	Bei Eingabefehler piepsen
ignorecase	Groß/Kleinschreibung bei der Suche ignorieren
list	Zeilenenden durch $ und Tabulatoren durch ^I anzeigen
number	Zeilen numeriert anzeigen
report=NN	Änderungen ab NN Zeilen in Statuszeile anz. (*1=ab 1 Zeile, 0 geht n.!*)
showmode	Vi-Modus in Statuszeile anzeigen (INSERT, APPEND, ...)
wrapmargin=NN	Automatischer Umbruch NN Zeichen vor Zeilenende (*0=kein Umbruch!*)
wrapscan	Suche über Dateiende/anfang hinweg fortsetzen

Optionen mit :set OPTION setzen und mit :set noOPTION (*zusammengeschrieben!*) löschen. In Konfigurations-Datei .exrc/.vimrc den Doppelpunkt vor set weglassen. Kommentare in .exrc/.vimrc durch " einleiten (*keine Leerzeilen erlaubt!*).

1.4 Wichtige Vi-Optionen für Programmierer

autoindent	Automatisch einrücken (mit <Strg-D/T> ein/ausrücken)
shiftwidth=NN	Breite NN für nachträgliches Einrücken setzen (<< >>)
showmatch	Passende öffnende Klammer bei Eingabe der schließenden anzeigen
tabstop=NN	Tabulatorbreite auf NN Zeichen setzen

© 2005–2016 T. Birnthaler, OSTC GmbH

1.5 Wichtige Vim-Optionen

`nocompatible`	Nicht im Vi-Kompatibilitätsmodus arbeiten
`ruler`	Koordinatenanzeige (Zeile + Spalte) in Statuszeile [**Lineal**]
`:syntax on`	Syntaxcoloring an (abhängig von Datei-Extension)
`wrap`	Zu lange Zeilen umgebrochen darstellen
`incsearch`	Inkrementell suche [**incremental**]
`hlsearch`	Zur Suche passende Textteile markieren [**highlight**]
`showcmd`	Unvollständige Kommandos in Statuszeile anzeigen [**command**]
`ff=unix`	Dateiformat (`dos`, `unix`, `mac`) [**fileformat**]

1.6 Nützliche Vi-Makros

`K`	`!}fmt -80 -uˆM`	Absatz (bis Leerzeile) auf 80 Zeichen Breite formatieren
`D`	`:%s/ *$//ˆM`	Alle Leerzeichen am Zeilenende entfernen
`Strg-W`	`/\<`	Suche nach Wort einleiten (danach Wort eintippen)
`Strg-W`	`\>/ˆM`	Suche nach Wort durchführen (*mit* `:map!` *definieren!*)
`+`	`:nˆM`	Zur nächsten Datei springen (ohne Abspeichern)
`#`	`:wˆM:nˆM`	Zur nächsten Datei springen (aktuelle vorher speichern)
`-`	`:prevˆM VIM!`	Zur vorherigen Datei springen (ohne Abspeichern)

Mit `:map NAME MACRO` definieren, `ˆM` ist einzugeben durch `<Strg-V> <RETURN>` oder im Vim durch `<CR>`.

© 2005–2016 T. Birnthaler, OSTC GmbH