

# VI/VIM-Befehlsübersicht

Version 1.23 — 19.2.2016

© 2005–2016 T. Birnthal, 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 Birnthal  
E-Mail: [tb@ostc.de](mailto:tb@ostc.de)  
Web: [www.ostc.de](http://www.ostc.de)

## Inhaltsverzeichnis

<b>1</b>	<b>Vi</b>	<b>3</b>
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 . . . . .	5
1.5	Wichtige Vim-Optionen . . . . .	5
1.6	Nützliche Vi-Makros . . . . .	5

# 1 Vi

## 1.1 Die wichtigsten Vi-Befehle

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

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

## 1.2 Weitere wichtige Vi-Befehle

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

## 1.3 Wichtige allgemeine Vi-Optionen

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

## 1.5 Wichtige Vim-Optionen

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

## 1.6 Nützliche Vi-Makros

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

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