

vi Schnellreferenz

V 1.0 Feb-2003 Jörg Hartmann © megalearn
<http://www.megalearn.de>

vi Modi (Betriebsarten)	
visual mode	Standardbetriebsart, direkt nach dem Start aktiv; Eingaben werden als Kommandos interpretiert
input mode	normale Texteingabe möglich wird durch a,A,i,I,c,C,o,O oder R eingeleitet wird mittels ESC wieder verlassen
ex mode	durch die Eingabe von : eingeleitet Ausführung komplexer Kommandos oder Macros

Starten des vi	
nach dem Start ist der vi im visual mode	
vi	Start mit leerem Textpuffer
vi <i>Datei</i>	<i>Datei</i> zum Bearbeiten öffnen
vi +n <i>Datei</i>	<i>Datei</i> zum Bearbeiten öffnen und Cursor auf Zeile n setzen
vi +/ <i>String Datei</i>	<i>Datei</i> zum Bearbeiten öffnen und Cursor beim erstem Vorkommen von <i>String</i> positionieren
vi -R <i>Datei</i>	<i>Datei</i> schreibgeschützt öffnen
:r <i>Datei</i>	<i>Datei</i> in den Textpuffer einlesen

Text eingeben	
i	Eingabe vor dem Cursor (<i>insert</i>)
I	Eingabe am Beginn der Zeile (<i>insert</i>)
a	Eingabe hinter dem Cursor (<i>append</i>)
A	Eingabe am Zeilenende (<i>append</i>)
o	neue Eingabe-Zeile unter dem Cursor einfügen (<i>open</i>)
O	neue Eingabe-Zeile über dem Cursor einfügen (<i>open</i>)
R	Überschreibmodus ab dem aktuellem Zeichen (<i>replace</i>)
CTRL-v	Präfix zur Eingabe eines Steuerzeichens Bsp: CTRL-v ESC m24
Verlassen des input mode durch ESC	

Text ändern und löschen	
Gelöschter Text wird in die Zwischenablage kopiert	
x,X	Zeichen unter bzw. vor dem Cursor löschen
dw	aktuelles Wort löschen
dd	aktuelle Zeile löschen
D	von Cursor bis Zeilenende löschen
J	nächste Zeile an die akt. Zeile anhängen
yy	aktuelle Zeile in Puffer kopieren
yw	aktuelle Wort in Puffer kopieren
9yy	ab aktuelle Zeile 9 Zeilen in Puffer kopieren
p,P	gelöschten Text nach bzw. vor dem Cursor wieder einfügen (aus Puffer)

Speichern bzw. Beenden des vi	
:w	aktuelle Datei speichern
:w <i>Datei</i>	unter dem Namen <i>Datei</i> speichern
:w >> <i>Datei</i>	Textpuffer an <i>Datei</i> anhängen
:q	vi beenden ohne zu speichern wird nach Änderungen verweigert
:wq	aktuelle Datei sichern und vi beenden
:q!	vi beenden, Änderungen ohne Speichern verwerfen
:40,70w <i>Datei</i>	Zeile 40-70 in <i>Datei</i> speichern

Suchen und Ersetzen	
/Muster	Suchen vorwärts nach <i>Muster</i>
?Muster	Suchen rückwärts nach <i>Muster</i>
n,N	Suchkommando in gleicher bzw. anderer Richtung wiederholen
:%s/ <i>foo/bar</i> /	Ersetzt <i>foo</i> im Text <i>einmal</i> durch <i>bar</i>
:%s/ <i>foo/bar</i> /g	Ersetzt alle Vorkommen von <i>foo</i> im Text durch <i>bar</i>
:%s/ <i>foo/bar</i> /c	Nachfrage vor dem Ersetzen von <i>foo</i> durch <i>bar</i>

Muster (reguläre Ausdrücke)	
.	steht für ein beliebiges Zeichen, außer Newline
*	steht für null oder beliebig viele der ihm unmittelbar vorangehenden Zeichen
^	steht am Anfang eines Ausdrucks für den Zeilenanfang
\$	steht am Schluß eines Ausdrucks für das Zeilenende
\	behandelt das folgende Zeichen als gewöhnliches Zeichen, \\ bezeichnet einen normalen \
[]	<i>eines</i> der in Klammern angegebenen Zeichen (Liste)
\< \>	Wortanfang und Wortende

Cursorsteuerung	
h	ein Zeichen nach links
j	eine Zeile nach unten
k	eine Zeile nach oben
l	ein Zeichen nach rechts
G	Cursor zum Dateiende
nG	Cursor zur Zeile <i>n</i>
1G	Cursor zum Dateianfang
CTRL-f	Page Down, eine Bildschirmseite vorwärts
CTRL-b	Page Up, eine Bildschirmseite zurück
CTRL-l	Redraw Screen (Bildschirm neu aufbauen)
\$	Cursor zum Zeilenende
^	Cursor zum Zeilenanfang

Konfiguration	
:set [no]showmode	Anzeige der Betriebsart an- bzw. abschalten
:set [no]ruler	Anzeige der akt. Cursor-Zeile/-Spalte an- bzw. abschalten
:set [no]number	Anzeige von Zeilennummern an- bzw. abschalten
:set [no]ignorecase	Beachten von Groß- bzw. Kleinschreibung beim Suchen an- bzw. abschalten
:set [no]autoindent	automatischen Zeileneinzug an- bzw. abschalten
:set [no]verbose	ausführliche Fehlermeldungen an- bzw. abschalten
:set backup=%bak	Backup <i>Datei.bak</i> anlegen
:set all	Anzeige aller Einstellungen
Konfig-Optionen können in die Datei ~/.exrc eingetragen werden	

Nützliches	
[zähler]Befehl	vielen Befehlen kann ein Zähler vorangestellt werden, der Befehl wird dann mehrfach ausgeführt
u	(<i>undo</i>) letzte Änderung zurücknehmen
U	(<i>undo</i>) akt. Zeile wiederherstellen
!:Befehl	führt den Unix-Befehl <i>Befehl</i> aus
:sh	Startet eine Shell
:r!Befehl	übernimmt die Ausgabe des Unix-Befehls <i>Befehl</i> in den Textpuffer

man vi vi Unix-Handbuchseite lesen (RTFM)