

Textmarken setzen

m(a-z)	Markiert die aktuelle Position mit einem Buchstaben von a-z. Bsp.: ma
^(a-z)	Bewegt Cursor an die Position a-z
" od. ``	Bewegt Cursor an die Position bevor das letzte / ? oder G Kommando ausgeführt wurde

UNIX Kommandos im vi ausführen

! <i>cmd</i>	Führt Shellkommando <i>cmd</i> aus. Diese speziellen Zeichen können eingefügt werden für: % Name der aktuellen Datei # Name der zuletzt editierten Datei
::!	Führt das letzte Shellkommando aus
:r <i>cmd</i>	Fügt die Ausgabe des Kommandos <i>cmd</i> ein
:f <i>datei</i>	Nennt die aktuelle Datei in <i>datei</i> um
:w <i>cmd</i>	Sendet die aktuell editierte Datei zu <i>cmd</i> als Standardeingabe und führt <i>cmd</i> aus
:cd <i>dir</i>	Wechselt aus dem aktuellen Verzeichnis in <i>dir</i> (\$HOME ist Standard)
:sh	Startet eine Subshell (CTRL - d führt zurück zum Editor)
:so <i>datei</i>	Führt die Befehle der <i>datei</i> aus (<i>datei</i> ist ein Shellsript)
! <i>cursor_cmd cmd</i>	Sendet den Text von der aktuellen Position bis zur <i>cursor_cmd Stelle</i> an das UNIX-Kommando <i>cmd</i> . Der Originaltext der Datei wird mit der Ausgabe von <i>cmd</i> überschrieben.

Beispiel:

!) sort <RETURN> Sortiert ab der aktuellen Position bis zum Ende eines Paragraphen und ersetzt Text mit sortiertem Text

Makros und Abkürzungen

:map <i>key cmd_seq key</i>	definieren, um bei Tastendruck <i>cmd_seq</i> zu starten
:map anzeigen	alle erzeugten Macros in der Statuszeile
:unmap <i>key</i>	Macrodefinition für <i>key</i> löschen
:ab <i>str string</i>	wenn <i>str</i> eingefügt wird, durch <i>string</i> ersetzen
:ab	alle Abkürzungen anzeigen
: una <i>str</i>	<i>str</i> nicht mehr abkürzen

Status-Kommandos

::=	Gibt aktuelle Zeilennummer aus
:=#	Gibt Zeilenanzahl der Datei aus
CTRL - g	Gibt Dateiname, aktuelle Zeilennummer, insgesamt Zeilenanzahl und Prozent der Dateielage an

Optionen setzen (Auswahl)

:set all	alle Optionen drucken
:set nooption	<i>option</i> abstellen
:set ai	automatische Einrückung aktivieren
:set eb	Fehlermeldungen Glockenton voranstellen
:set ic	während der Suche Groß- und Kleinschreibung ignorieren
:set list	Zeigt Tabulatoren (^!) und Zeilenenden (\$) an
:set nu	Numeriert Zeilen auf Bildschirm
:set ro	ändert Datei in "read only" um
:set scroll= <i>n</i>	Gibt Zeilen für "CTRL-d" und "z" an
:set sw=8	Gibt Schrittweite des Tabulators an
:set term	Gibt Terminaltyp an
:set wa	Schreibt auf Dateien ohne vorherige Prüfung zurück
:set window = <i>n</i>	Gibt Zeilenanzahl eines Fensters an

Kurzanleitung vi-Editor

Starten einer vi Sitzung

vi <i>datei</i>	Editiere <i>datei</i>
vi -r <i>datei</i>	Editiere die zuletzt gesicherte Version der Datei nach einem System oder Editor Crash
vi + <i>n datei</i>	Editiere <i>datei</i> und plaziere Cursor auf die Zeile <i>n</i>
vi + <i>datei</i>	Editiere <i>datei</i> und plaziere Cursor auf die letzte Zeile
vi <i>datei1 datei2</i> ...	Editiere <i>datei1</i> bis <i>datein</i> . Nach Sicherung, wechseln in nächste Datei mit :n
vi +/str <i>datei</i>	Editiere <i>datei</i> und plaziere Cursor auf die Zeile, die <i>str</i> enthält

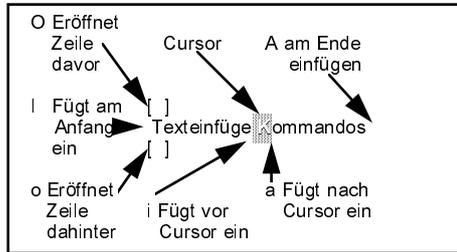
Textsicherung und vi beenden

ZZ od. :wq od. :x	sichert Text und verläßt vi
:w <i>datei</i>	Sichert <i>datei</i> , ohne vi zu verlassen. Weglassen von <i>datei</i> sichert aktuelle Datei
:n, <i>mw datei</i>	Schreibt Zeile <i>n</i> bis <i>m</i> in <i>datei</i>
:n, <i>mw >></i> <i>datei</i>	Hängt Zeilen <i>n</i> bis <i>m</i> an <i>datei</i> an
:q	Verläßt vi ohne Speicherung Fehlermeldung wenn etwas geändert wurde
:q!	Verläßt vi ohne Speicherung, überschreibt aktuelle Änderungen
Q	Verläßt vi und geht in den ex Editor. :vi wechselt zurück
:e!	Editiert, die zuletzt auf Festplatte gespeicherte Version der aktuellen Datei nochmals

Text einfügen

Um den Einfügemodus zu verlassen, muß ESC gedrückt werden

a	Fügt Text hinter dem Cursor ein
A	Fügt Text am Ende der aktuellen Zeile ein
i	Fügt vor dem Cursor ein
I	Fügt am Anfang der Zeile ein
o	Eröffnet neue Zeile, hinter der aktuellen und fügt ein
O	Eröffnet neue Zeile, vor der aktuellen und fügt ein
:r <i>datei</i>	Füge <i>datei</i> hinter der aktuellen Zeile ein
:nr <i>datei</i>	Füge <i>datei</i> nach der Zeile <i>n</i> ein



Text ändern

Diese Kommandos werden *n*-mal wiederholt, wenn ihnen *n* (eine Zahl) vorangestellt wird.

r <i>char</i>	ersetze aktuelles Zeichen durch <i>char</i>
R <i>text</i> ESC	ersetze aktuelle(s) Zeichen durch <i>text</i>
s <i>text</i> ESC	setze <i>text</i> für aktuelles Zeichen ein
S oder cc <i>text</i> ESC	setze <i>text</i> für aktuelles Zeichen ein
cw <i>text</i> ESC	Ändere aktuelles Wort in <i>text</i>
C <i>text</i> ESC	Ändere Rest der aktuellen Zeile in <i>text</i>
ccursor_ <i>cmd</i> <i>text</i> ESC	Ändere von aktueller Position bis zu <i>cursor_ <i>cmd</i></i> in <i>text</i>

Text löschen

dd	aktuelle Zeile löschen
ndd	<i>n</i> Zeile(n) löschen
D	bis Ende der Zeile löschen
x	aktuelles Zeichen löschen
nx	<i>n</i> Zeichen löschen (rechts vom Cursor)
nX	<i>n</i> Zeichen löschen (links vom Cursor)

Text suchen

/ <i>muster</i>	vorwärts nach <i>muster</i> suchen
? <i>muster</i>	rückwärts nach <i>muster</i> suchen
n	Wiederholt das letzte / oder ? Suchkommando
N	Wiederholt das letzte / oder ? Suchkommando in gegenläufiger Richtung

Text kopieren und plazieren

<i>ny</i> oder <i>nY</i>	Kopiert <i>n</i> Zeilen (in den Puffer); <i>n</i> für aktuell zukopierende Zeilen weglassen
y <i>Cursor_ cmd</i>	Kopiert von Cursor bis <i>Cursor_ cmd</i> (z.B.: yG kopiert aktuelle Zeile in die letzte Zeile der Datei)
?(<i>a-z</i>) <i>ny</i> od. ?(<i>a-z</i>) <i>ndd</i>	Kopiert oder löscht <i>n</i> Zeilen in den angegebenen Puffer (<i>a-z</i>); <i>n</i> für aktuelle Zeile weglassen
p	Schreibt Text aus Puffer nach aktueller Zeile (auch gelöschten Text)
P	Schreibt Text aus Puffer vor aktuelle Zeile (auch gelöschten Text)
?(<i>a-z</i>)p od. ?(<i>a-z</i>)P	Schreibt die Zeilen aus dem angegebenen Puffer (<i>a-z</i>); vor oder hinter der aktuellen Zeile

Rückgängig machen und wiederholen von Kommandos

u	Macht letztes Kommando rückgängig
U	Bringt die aktuelle Zeile in Originalzustand
"np	Stellt die zu <i>n</i> .letzst gelöschte Zeile wieder her (<i>9</i> sind im Puffer)
"!pu.u.	Rollt zwischen den gelöschten Zeilen im Puffer, bis man die gewünschte Löschung gefunden wurde (wiederholen u)

Zeilen verbinden

J	Hängt die nächste Zeile ans Ende der aktuellen
nJ	Hängt <i>n</i> Zeilen an die aktuelle an

Cursor plazieren und Bildschirm ausrichten

h,l,k,j	Cursor nach links, rechts, oben, unten bewegen
w,b	Cursor ein Wort nach rechts, links bewegen
0,\$	Cursor zum Anfang, Ende einer Zeile bewegen
G	Gehe in die letzte Zeile
H	Cursor in die oberste Bildschirmzeile bewegen
nH	Cursor in die <i>n</i> -te Zeile von oben bewegen
M	Cursor in die Mitte des Bildschirms setzen
L	Cursor in die letzte Bildschirmzeile bewegen
nL	Cursor in die <i>n</i> -te Zeile von unten bewegen
CTRL - u	Bildschirm 1/2 Seite hinaufschieben
CTRL - d	Bildschirm 1/2 Seite hinunterschieben
CTRL - e	Bildschirm eine Zeile hinaufschieben
CTRL - y	Bildschirm eine Zeile hinunterschieben
CTRL - b	Bildschirm eine Seite hinaufschieben
CTRL - f	Bildschirm eine Seite hinunterschieben
CTRL - l	Bildschirm neu aufbauen
z Return	Aktuelle Zeile als oberste Bildschirmzeile
nz Return	Zeile <i>n</i> als oberste Bildschirmzeile
z.	aktuelle Zeile als mittlere Bildschirmzeile
nz.	Zeile <i>n</i> als mittlere Bildschirmzeile
z-	aktuelle Zeile als unterste Bildschirmzeile
nz-	Zeile <i>n</i> als unterste Bildschirmzeile
:n	Gehe nach Zeile <i>n</i>