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

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

Beispiel:

:1;$ s/alt/neu/g <RETURN>	ersetzt (s) von Zeile 1 bis Ende ($) alle (g) Textstellen "neu" gegen "alt"
!} sort <RETURN>	Sortiert ab der aktuellen Position bis zum Ende eines Paragraphen und ersetzt Text mit sortiertem Text

Makros und Abkürzungen

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

Status-Kommandos

:.=	Gibt aktuelle Zeilennummer aus
:=	Gibt Zeilenanzahl der Datei aus
CTRL - g	Gibt Dateiname, aktuelle Zeilennummer, insgesamte Zeilenanzahl und Prozent der Dateienlage an

Optionen setzen (Auswahl)

:set all	alle Optionen drucken
:set no*option*	*option* abstellen
:set ai	automatische Einrückung aktivieren
:set eb	Fehlermeldungen Glockenton voranstellen
:set ic	während der Suche Groß- und Kleinschreibung ignorieren
:set list	Zeigt Tabulatoren (^I) und Zeilenenden ($) an
:set nu	Numeriert Zeilen auf Bildschirm
:set ro	ändert Datei in "read only" um
:set scroll=*n*	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 =*n*	Gibt Zeilenanzahl eines Fensters an

Kurzanleitung vi-Editor

Starten einer vi Sitzung

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

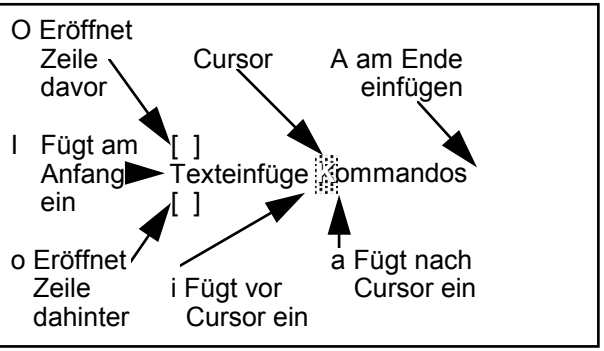
Textsicherung und vi beenden

ZZ od. :wq od. :x	sichert Text und verläßt vi
:w *datei*	Sichert *datei,* ohne vi zu verlassen. Weglassen von *datei* sichert aktuelle Datei
:*n,m*w *datei*	Schreibt Zeile *n* bis *m* in *datei*
:n,mw >> datei	Hängt Zeilen n bis m an *datei* 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 *datei*	Füge *datei* hinter der aktuelen Zeile ein
:*n*r *datei*	Füge *datei* nach der Zeile *n* ein



Text ändern

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

r*char*	ersetze aktuelles Zeichen durch *char*
R*text* ESC	ersetze aktuelle(s) Zeichen durch *text*
s*text* ESC	setze *text* für aktuelles Zeichen ein
S oder cc *text* ESC	setze *text* für aktuelles Zeichen ein
cw*text* ESC	Ändere aktuelles Wort in *text*
C*text* ESC	Ändere Rest der aktuellen Zeile in *text*
c*cursor_cmd text* ESC	Ändere von aktueller Position bis zu *cursor_cmd* in *text*

Text löschen

dd	aktuelle Zeile löschen
*n*dd	*n* Zeile(n) löschen
D	bis Ende der Zeile löschen
x	aktuelles Zeichen löschen
*n*x	*n* Zeichen löschen (rechts vom Cursor)
*n*X	*n* Zeichen löschen (links vom Cursor)

Text suchen

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

Text kopieren und plazieren

*n*yy oder *n*Y	Kopiert *n* Zeilen (in den Puffer); *n* für aktuell zukopierende Zeilen weglassen
y*Cursor_cmd*	Kopiert von Cursor bis *Cursor_cmd* (z.B.: yG kopiert aktuelle Zeile in die letzte Zeile der Datei)
*"(a-z)n*yy od. *"(a-z)n*dd	Kopiert oder löscht n Zeilen in den angegebenen Puffer (a-z); *n* 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)
"(a-z)p od. "(a-z)P	Schreibt die Zeilen aus dem angegebenen Puffer (a-z); vor oder hinter der aktuellen Zeile

Rückgängig machen und wiederholen von Kommandos

u	Macht letztes Kommando rückgängig
U	Bringt die aktuelle Zeile in Originalzustand
"*n*p	Stellt die zu *n*.letzt gelöschte Zeile wieder her (9 sind im Puffer)
"1pu.u.	Rollt zwischen den gelöschten Zeilen im Pufer, bis man die gewünschte Löschung gefunden wurde (wiederholen u)

Zeilen verbinden

J	Hängt die nächste Zeile ans Ende der aktuellen
*n*J	Hängt *n* 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
*n*H	Cursor in die *n*-te Zeile von oben bewegen
M	Cursor in die Mitte des Bildschirms setzen
L	Cusor in die letzte Bildschirmzeile bewegen
*n*L	Cursor in die *n*-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
*n*z Return	Zeile *n* als oberste Bildschirmzeile
z.	aktuelle Zeile als mittlere Bildschirmzeile
*n*z.	Zeile *n* als mittlere Bildschirmzeile
z-	aktuelle Zeile als unterste Bildschirmzeile
*n*z-	Zeile *n* als unterste Bildschirmzeile
:*n*	Gehe nach Zeile *n*