

Normal Mode

ESC

kehrt in den 'normal mode' zurück

Zahl + Kommando

führt das Kommando entsprechend oft aus

.

wiederholt die letzte Aktion

u

(undo) macht die letzte Aktion rückgängig

Strg + r

wiederherstellen

gg

erste Zeile

k

Zeile hoch

Öffnen, Speichern, Schließen

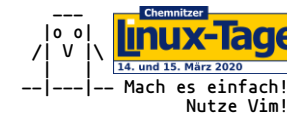
vim file.txt +zahl
öffnet Vim an einer bestimmten Zeile

vim scp://server/path
öffnet eine Datei von einem (entfernten) Server

:w file2.txt
speichern unter "file2.txt"

ZZ oder **:wq** oder **:x**
speichern und schließen

ZQ oder **:q!**
schließen, ohne Änderungen zu speichern



Hilfe!

:h öffnet die Hilfe in Vim
man vim verrät Dir alles über Vim

↑ CAPSLOCK bringt Vim total durcheinander

Strg + s sperrt die Eingabe (in einigen Konsolen)

Strg + q hebt die Sperre auf (in einigen Konsolen)

🔍 vimtutor

CC-BY-SA: m.mann@pengutronix.de



vorherige Textstellen

0

Zeilenanfang

B

Wortanfang

b

Wortanfang/Sonderzeichen

h

Zeichen

nächste Textstellen

l

Zeichen

w

Wortanfang

W

Wortanfang/Satzzeichen

e

Wortende

E

Wortende/Satzzeichen

\$

Zeilenende

Visual Mode

v

visual line- markiert Text zeichen- und zeilenweise

V

visual line'- markiert Text zeilenweise

Strg + v

visual block' - markiert Text zeichen- und spaltenweise

j

Zeile runter

G

letzte Zeile

Insert Mode

i

schreibt vor dem Cursor

a

schreibt hinter dem Cursor

I

schreibt am Anfang der Zeile

A

schreibt am Ende der Zeile

o

schreibt in einer neuen Zeile unter der Position

O

schreibt in einer neuen Zeile über der Position

Textmanipulation

cc

löscht die aktuelle Zeile und schreibt

C

löscht den Rest der Zeile und schreibt

r

ersetzt ein Zeichen

R

ersetzt beliebig viele Zeichen

x

löscht ein Zeichen

ci"

(change in) ersetzt Text zwischen " dies funktioniert auch mit (oder <

y

(yank) kopieren

yy

kopiert die aktuelle Zeile

dd

löscht die aktuelle Zeile

dw

löscht das aktuelle Wort

d\$

löscht die restliche Zeile ab der aktuellen Position

p

(paste) fügt Text über der aktuellen Zeile ein

P

(paste) fügt Text unter der aktuellen Zeile ein

~

tauscht Groß- und Kleinschreibung

:r file.txt

(read) fügt den Inhalt der Datei ein

:set paste

ermöglicht das Einfügen eines formatierten Textes aus dem Buffer

Sowohl **c** (change), **d** (delete) als auch **y** (yank) sind mit **w** (Wort), **Shift** oder **\$** (Zeilenende) kombinierbar.

im 'visual block':

I

lorem ipsum

ESC

fügt die Eingabe in allen Zeilen im Text ein

s

(substitute) ersetzt den markierten Text

<

oder

>

Text verrücken

Einrückung

==

automatische Einrückung

>>

Text einrücken Länge der 'shiftwidth'

<<

Text ausrücken Länge der 'shiftwidth'

gqap

formatiert einen Absatz auf die Länge von 'textwidth'

Suchen und Ersetzen

/foo sucht nach 'foo'

n

nächstes Suchergebnis

N

vorheriges Suchergebnis

:%s/foo/bar/g

ersetzt foo durch bar im gesamten Dokument



Regular Expressions

Befehle

: leitet Befehle ein (Tabcompletion ist möglich)

:42 geht zur entsprechenden Zeile (in diesem Fall 42)

:! ermöglicht das Ausführen von Shell-Kommandos

:! fügt das Ergebnis eines Shell-Kommandos ein

Tabs und Fenster

:tabe file2.txt
öffnet file2.txt in einem neuen Tab

vim -p file.txt file2.txt
öffnet beide Dateien in Tabs

gt wechselt zum nächsten Tab

gT wechselt zum vorherigen Tab

:split
splittet das Fenster horizontal

:vsplit
splittet das Fenster vertikal

Strg + w wechselt den Fokus

Einstellungen

:set option
schaltet eine Option ein (Tabcompletion ist möglich)

:set nooption
Option wird ausgeschaltet, indem dieser ein 'no' vorangestellt wird

vim ~/.vimrc
Einstellungen können in der vimrc gespeichert werden

Rechtschreibkorrektur

:set spell
schaltet Rechtschreibkorrektur ein

:set spelllang=de
ändert die Wörterbuchsprache

z= öffnet das Wörterbuch mit Korrekturvorschlägen