

vi überleben

Für Debian/Ubuntu empfohlen: Das Paket `vim` installieren, welches eine komfortablere Version von `vi` enthält. Dann wird u.a. immer angezeigt, wenn sich der Editor im Einfüge-Modus befindet.

Befehl	Funktion
<code>i</code>	Einfügemodus (i nsert)
<code><ESC></code>	Befehlsmodus
<code>:w</code>	speichern (w rite)
<code>:q</code>	beenden (q uit)
<code>:q!</code>	beenden ohne speichern
<code>:wq</code>	speichern und beenden
<code>u</code>	Rückgängig (u ndo)
<code>Strg+r</code>	undo Wiederherstellen (r edo)

In älteren Vim-Versionen nützlich:

<code>J</code>	hängt die nächste Zeile an das Ende der aktuellen Zeile an (j oin)
----------------	--

VIM lernen

- mit einem Browser-Spiel: <http://vim-adventures.com/>
- mit vimtutor:

```
vimtutor
```

Grundlagen

Navigationsbefehle I



Aufbau: `[Anzahl] Befehl`

Beispiel: `10w` (10 Wörter weiter springen)

<code>w</code>	ein Wort weiter (w ord)
<code>b</code>	ein Wort zurück (b ack)
<code>e</code>	zum nächsten Wortende (e nd)
<code>0</code>	zum Zeilenanfang

\$	zum Zeilenende
gg	Dateianfang (g o)
10gg	Zeile 10
G	letzte Zeile
/	Vorwärts suchen
?	Rückwärts suchen
n	Zum nächsten Treffer in gleicher Richtung springen (n ext)
N	Zum nächsten Treffer in entgegengesetzter Richtung springen

Änderungsbefehle I

Aufbau: [Anzahl] Befehl Anwendungsbereich

Anzahl ist dabei optional. Es gibt an, wie oft der Befehl ausgeführt werden soll. Mit Anwendungsbereich bezeichne ich Navigationsbefehle. Sie werden benutzt, um sich innerhalb der Datei zu bewegen. Manche Befehle erwarten keinen Anwendungsbereich.

Beispiel: cwERSATZTEXT (Text bis zum nächster Wort durch ERSATZTEXT ersetzen)

c...	ersetzen (change)
cw	Wort ersetzen (change word)
cc	Zeile ersetzen
C	bis zum Ende der Zeile ersetzen
.	letzten Befehl wiederholen

Anwendungsbeispiel: suchen und ersetzen I

/	suchen
..	Suchwort eingeben ..
cw	1 Wort ersetzen
..	Ersatzwort eingeben ..
<ESC>	Änderung beenden
n	zum nächsten Treffer springen
.	Ersetzung wiederholen

copy & paste

d...	ausschneiden (delete)
dw	Wort ausschneiden (delete word)
dd	Zeile ausschneiden
D	bis zum Ende der Zeile ausschneiden
y...	kopieren (yank)
yw	Wort kopieren
yy	Zeile kopieren
Y	Zeile kopieren
y\$	bis zum Ende der Zeile kopieren
	einfügen (paste)

d...	ausschneiden (delete)
P	an aktueller Position einfügen
p	nach aktueller Position einfügen

Visueller Modus

v	markieren (visual)
V	markieren, zeilenweise
Strg+v	markieren, blockweise

Anwendungsbeispiel: suchen und ersetzen II

/	suchen
..	Suchwort eingeben ..
vec	1 Wort ersetzen
..	Ersatzwort eingeben ..
<ESC>	Änderung beenden
n	zum nächsten Treffer springen
.	Ersetzung wiederholen

Text Markieren

vip	aktuellen Absatz markieren
vit	Text innerhalb des aktuellen xml-Tags markieren
vi(oder vib	Text innerhalb der (Klammern) markieren
vi{ oder viB	Text innerhalb der {Klammern} markieren
vi[Text innerhalb der [Klammern] markieren
vi"	Text innerhalb der "Anführungszeichen" markieren
vi'	Text innerhalb der 'Anführungszeichen' markieren
vap	aktuellen Absatz inklusive Leerzeilen außenrum markieren
vat	Text um das aktuelle xml-Tag markieren
va(oder vab	Text inklusive der (Klammern) markieren
va{ oder vaB	Text inklusive der {Klammern} markieren
va[Text inklusive der [Klammern] markieren
va"	Text inklusive der "Anführungszeichen" markieren
va'	Text inklusive der 'Anführungszeichen' markieren

Quelle: <http://vimdoc.sourceforge.net/html/doc/motion.html#object-select>

weitere Wege in den Einfügemodus

i	an der aktuellen Position
I	am Zeilen- bzw. Blockanfang
a	hinter der aktuellen Position
A	am Zeilen- bzw. Blockende
o	neue Zeile darunter einfügen
O	neue Zeile darüber einfügen

Beispiel Texte

[vim beispiel texte](#) zum Üben

Suchen und Ersetzen III

Befehl	Funktion
<code>:%s/suchwort/ersatz/g</code>	suchen und ersetzen (s ubstitute, g lobal)
<code>:g/^\$/d</code>	Alle Leerzeilen löschen

Eingabehilfen im Einfügemodus

Strg+n	Wort-Ergänzung
Strg+x f	Dateinamen-Ergänzung a la Tab-Taste

Mehrere Dateien bearbeiten

<code>vi -o file1 file2</code>	horizontaler Split (Fenster wechseln mit Ctrl+W Cursortaste)
<code>vi -O file1 file2</code>	vertikaler Split (Dateien nebeneinander)
<code>vi -p file1 file2</code>	Tab-Modus (Dateitabs wechseln mit g t (<u>g</u> oto) - T für rückwärts)

Einstellungen

Den vim.tiny smart machen

Folgende Zeilen in der `~/.vimrc` sorgen dafür, dass auch im vim.tiny im Insert-Modus die Cursortasten und Backspace vernünftig, auch über die Grenzen des gerade eingefügten Textes hinaus, funktionieren:

```
set nocompatible
set backspace=2
```

Groß-/Kleinschreibung beim Suchen ignorieren

```
:set ignorecase
```

Konfigurationsdateien ohne Kommentare anzeigen

```
:set fdm=expr
:set
fdm=getline(v:lnum)=~'^\\s*#'?1:getline(prevnonblank(v:lnum))=~'^\\s*#'?1:ge
tline(nextnonblank(v:lnum))=~'^\\s*#'?1:0
```

Syntax-Highlighting einschalten

```
:syntax on
```

Zeilennummern ein-/ausschalten

```
:set number
:set nonumber
```

Steuerzeichen anzeigen

```
:set list
```

Navigationsbefehle II

{	zum nächsten Absatz
}	zum vorherigen Absatz
*	
#	

Weitere Informationen und Befehle

Modi:

- Command/normal Mode
- Insert Mode (zum normalen Schreiben)
- Ex Mode (das was man kriegt, wenn man im command-Mode „:“ drückt)
- Visual Mode

Ex Mode


Befehl	Funktion
ZZ	speichern & verlassen
:x	speichern & verlassen
:w <Datei>	Speichert es in einer neuen Datei
:e .	Eine andere Datei auf dem aktuellen Verzeichnis öffnen
:e /etc/passwd	Die Datei passwd öffnen
:r /etc/hosts	Den Inhalt der angegebenen Datei an die Cursorposition des offenen Dokuments einlesen
:tabnew	Neuen Tab erstellen
:tabnext	In den nächsten Tab springen
:tabclose	Tab schließen
:redo	Holt den Text wieder wenn ich zuviele undos gemacht habe
:help <Befehl>	
:set all	Zeigt alle Befehle an in Kombination mit set

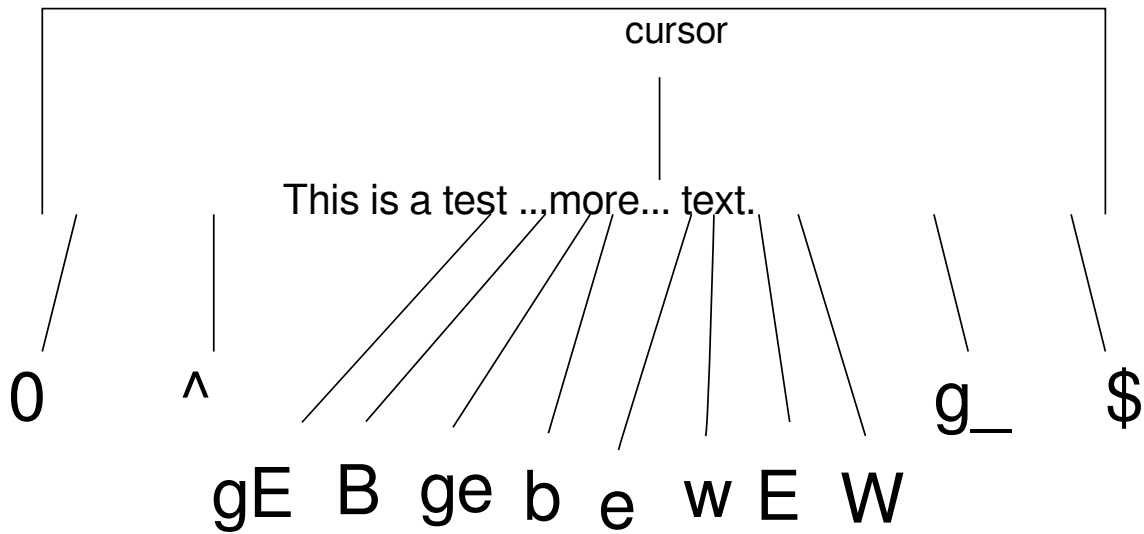
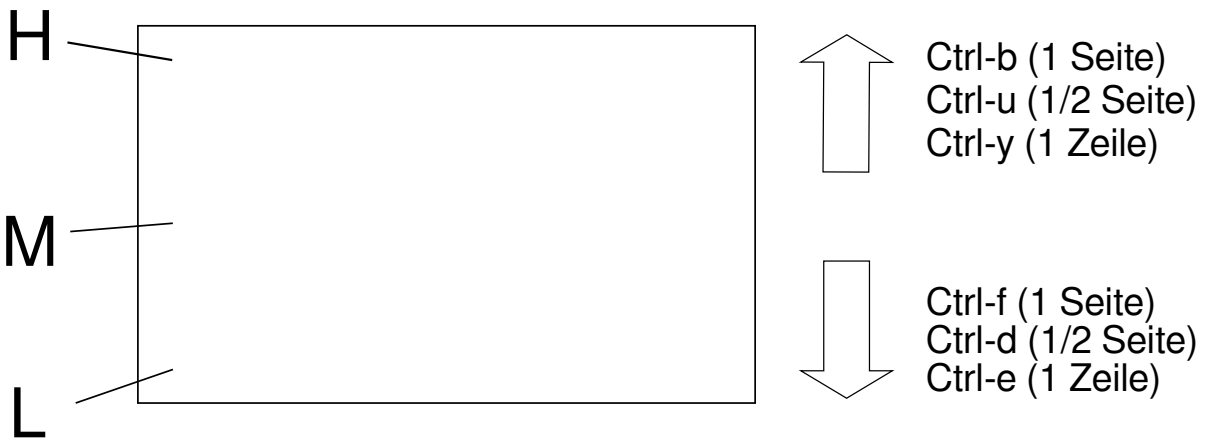
Alternative Editoren

Eigentlich keine.

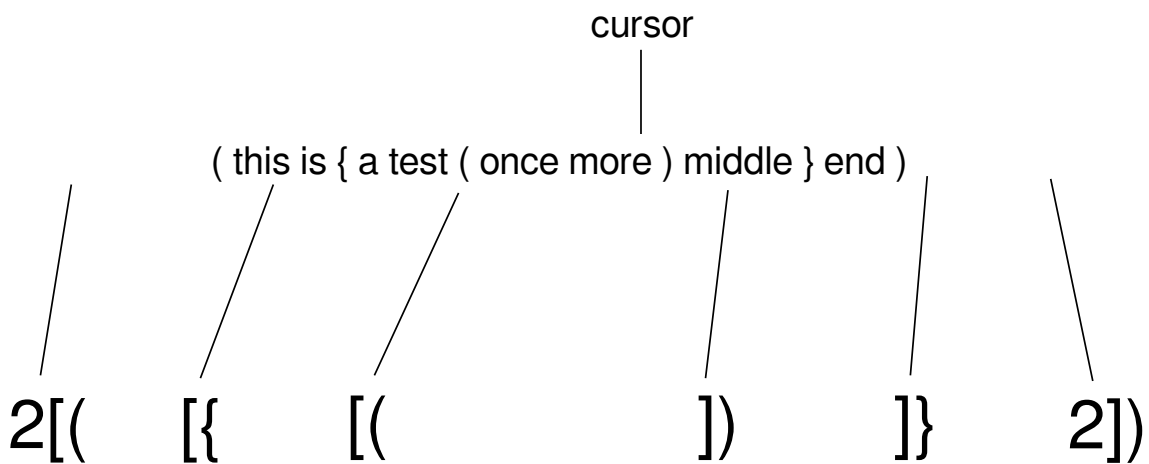
Befehl	SuSE Paket	Debian Paket	CentOS Paket	Beschreibung
mcedit	mc	mc		Teil vom Midnight Commander
nano				

Dokus und Links

- http://www.debian-administration.org/article/Hiding_comments_in_configuration_files
- [Tutorial mit Graphiken zur Navigation](#)
- Interview mit Bill Joy über die Entstehung von „vi“
<http://web.cecs.pdx.edu/~kirkenda/joy84.html>
- 



g0, g^ and g\$ work for screen-lines (if the line is longer than the screen).



These commands work over many lines! Use % to jump to the matching bracket.

From:

<https://wiki.lab.linuxhotel.de/> - **Linuxhotel Wiki**

Permanent link:

<https://wiki.lab.linuxhotel.de/doku.php/lpi2:vim>

Last update: **2022/09/28 14:25**

