

Der **vi**-Editor in Übersichten

Inhaltsverzeichnis

- [1. Start einer vi-Sitzung](#)
- [2. Text sichern und vi verlassen](#)
- [3. Eingeben von Text](#)
- [4. Rollen und Ausrichten des Bildschirminhalts](#)
- [5. Bewegen des Cursors in einem File](#)
- [6. Text löschen](#)
- [7. Ändern von Text](#)
- [8. Änderungen zurücksetzen](#)
- [9. Text kopieren oder verschieben](#)
- [10. Globales Suchen und Ersetzen von Text](#)
- [11. Suchen im Text](#)
- [12. Shellkommandos](#)
- [13. Vi-Optionen](#)

1. Start einer vi-Sitzung

```
vi file      Editiere file
vi -r file   Editiere die letzte gerettete Version
             von file nach Systemabsturz
vi +n file   Editiere file mit Cursorpositionierung
             auf Zeile n
vi + file    Editiere file mit Cursorpositionierung
             auf letzte Zeile
vi file1 ... fileN Editiere file1 bis fileN
vi +/str file Editiere file mit Cursorpositionierung
             auf Zeile, die str enthält
```

2. Text sichern und vi verlassen

```
ZZ oder :wq   aktuelles File sichern und vi verlassen
:q!           vi verlassen, ohne das aktuelle File
             zu sichern
:w           aktuelles File sichern
:w file      aktuelles File in file sichern
:w! file     Inhalt von file mit dem aktuellen
             File überschreiben
:m,nw file   Zeile m bis Zeile n des aktuellen
             Files in file sichern
```

3. Eingeben von Text

```
a           Text nach dem Cursor einfügen
A           Text am Ende der Zeile einfügen
i           Text vor dem Cursor einfügen
I           Text am Anfang der Zeile einfügen
o           unter dem Cursor Leerzeile für
             Texteingabe eröffnen
O           über dem Cursor Leerzeile für
             Texteingabe eröffnen

:r file     file lesen und nach der aktuellen
             Zeile einfügen
:nr file    file lesen und nach der n-ten
             Zeile einfügen
```

Der Wechsel vom vi-Eingabemodus zurück in den Kommandomodus erfolgt mit der Taste <ESC>

12. Rollen und Ausrichten des Bildschirminhaltes

```
<CTRL>-f    eine Seite vorwärts rollen
<CTRL>-b    eine Seite rückwärts rollen
<CTRL>-d    eine halbe Seite vorwärts rollen
<CTRL>-u    eine halbe Seite rückwärts rollen

z <RET>     aktuelle Zeile wird oberste Zeile
             des Bildschirms
z.          aktuelle Zeile wird zur mittlere Zeile
z          aktuelle Zeile wird zur untersten Zeile
```

13. Bewegen des Cursors in einem File

```
h oder <--  Cursor nach links bewegen
l oder <--> Cursor nach rechts bewegen
j           Cursor nach unten bewegen
k           Cursor nach oben bewegen
$           Cursor zum Ende der aktuellen Zeile bewegen
0 (Null)   Cursor zum Anfang der aktuellen Zeile bewegen
^           Cursor zum ersten sichtbaren Zeichen der
             aktuellen Zeile bewegen
tchar      Cursor vor erstes Zeichen char bewegen
ntchar     Cursor vor n-tes Zeichen char bewegen

w           Cursor zum Anfang des nächsten Wortes bewegen
W           Cursor zum Anfang des nächsten Wortes bewegen
             (ohne Berücksichtigung von Sonderzeichen)
b           Cursor zum Anfang des vorhergehenden Wortes
             bewegen
B           Cursor zum Anfang des vorhergehenden Wortes
             bewegen (ohne Berücksichtigung von
             Sonderzeichen)
e           Cursor zum Ende des nächsten Wortes bewegen
E           Cursor zum Ende des nächsten Wortes bewegen
             (ohne Berücksichtigung von Sonderzeichen)

H           Cursor auf oberste Zeile des Bildschirms
M           Cursor auf mittlere Zeile des Bildschirms
L           Cursor auf unterste Zeile des Bildschirms
1G          Cursor auf 1. Zeile im File bewegen
nG          Cursor auf n-te Zeile im File bewegen
```

```

G          Cursor auf letzte Zeile im File bewegen
n+        Cursor n Zeilen vorwärts bewegen
n-        Cursor n Zeilen rückwärts bewegen

(          Cursor zum Anfang des Satzes bewegen
)          Cursor zum Anfang des nächsten Satzes bewegen
{          Cursor zum Anfang des Absatzes bewegen
}          Cursor zum Anfang des nächsten Absatzes
bewegen
% (auf {}[]()) Cursor auf zugehörnde Klammer bewegen

```

20. Text löschen

Im vi-Kommandomodus:

```

x          durch Cursor markiertes Zeichen löschen
nx        n Zeichen ab Cursor löschen
dd        aktuelle Zeile löschen
nnd       die nächsten n Zeilen löschen
           (einschl. der aktuellen)
dw        aktuelles Wort löschen (vom Cursor bis
           Wortende)
dW        aktuelles Wort einschl. Sonderzeichen löschen
ndw       n Wörter ab markiertem Wort löschen
db        vorhergehendes Wort löschen
dB        vorhergehendes Wort einschl. Sonderzeichen
           löschen
ndb       n Wörter vor dem Cursor löschen
d$ oder D aktuelle Zeile vom Cursor bis Zeilenende
           löschen
d0 (Null) aktuelle Zeile vom Cursor bis Zeilenanfang
           löschen
dcursor_cmd Text vom Cursor bis zu der durch cursor_cmd
           bestimmten Stelle löschen

:m,nd     Zeilen m bis n löschen

```

Im vi-Eingabemodus:

```

<CTRL>-h oder Backspace im Eingabemodus vorhergehendes
Zeichen löschen
<CTRL>-w im Eingabemodus vorhergehendes Wort löschen
<CTRL>-x im Eingabemodus zuletzt eingegebenen Text
vollständig löschen

```

21. Ändern von Text

```

rchar     aktuelles Zeichen durch char ersetzen
Rtext     aktuelle(s) Zeichen durch text ersetzen
stext     aktuelles Zeichen durch text ersetzen
Stext oder cctext aktuelle Zeile durch text ersetzen
cwttext   aktuelles Wort durch text ersetzen
ncwtext   n Wörter ab Cursor durch text ersetzen
Ctext     aktuelle Zeile von Cursor bis Zeilenende
           durch text ersetzen
ccursor_cmd text Text von Cursor bis cursor_cmd durch
           text ersetzen
.         letztes Kommando wiederholen

```

26. Änderungen zurücksetzen

```

u          zuletzt ausgeführten Befehl rückgängig machen
U          Zustand der aktuellen Zeile vor der Änderung
           herstellen
:q!        vi-Sitzung ohne Sicherung der Änderung
           verlassen

```

27. Text kopieren oder verschieben

```

Y oder yy  aktuelle Zeile zwischenspeichern
nY oder nyy n Zeilen ab Cursor zwischenspeichern
ycursor_cmd Text von Cursor bis cursor_cmd
           zwischenspeichern

dd        aktuelle Zeile löschen
ndd       n Zeilen ab Cursor löschen
dcursor_cmd Text von Cursor bis cursor_cmd löschen

P         gelöschten oder zwischengespeicherten Text
           hinter (oder unter) dem Cursor einfügen
P         gelöschten oder zwischengespeicherten Text
           vor (oder über) dem Cursor einfügen

xp        zwei Zeichen vertauschen

J         zwei Zeilen zusammenfügen
nJ        n Zeilen zusammenfügen

```

28. Globales Suchen und Ersetzen von Text

```

:m,ns/str1/str2/ von Zeile m bis Zeile n wird str1
           durch str2 ersetzt
:m,ns/str1/str2/g von Zeile m bis Zeile n werden alle
           str1 durch str2 ersetzt
&         das letzte :s-Kommando wird wiederholt

:g/str/cmd   cmd wird für alle Zeilen ausgeführt, die
           str enthalten
:v/str/cmd   cmd wird für alle Zeilen ausgeführt, die
           str nicht enthalten
~          aktueller Buchstabe wird in Groß- bzw.
           Kleinbuchstabe umgewandelt
n~         n Buchstaben ab Cursor in Groß- bzw.
           Kleinbuchstaben umwandeln

```

29. Suchen im Text

```

fchar     vorwärts suchen von char in der aktuellen
           Zeile
Fchar     rückwärts suchen von char in der aktuellen
           Zeile
/str<RET> vorwärts suchen von str ab der aktuellen
           Zeile
?str<RET> rückwärts suchen von str ab der aktuellen
           Zeile

```

n nächstes Muster (vorwärts) suchen

12. Shellkommandos

```
!:cmd        Kommando cmd wird der Shell zur Ausführung  
             übergeben  
:r !cmd      Kommando cmd wird ausgeführt und Ergebnis  
             hinter der aktuellen Zeile eingefügt
```

13. Vi-Optionen

Durch das Belegen von vi-Optionen (Kommando :set) mit einem Wert oder das Setzen von Optionen kann die Arbeitsweise des Editors beeinflusst werden. Die Kommandos set können in ein File mit dem Namen \$HOME/.exrc eingetragen werden, welches bei jedem vi-Editoraufruf abgearbeitet wird.

```
:set all      Anzeige Belegung aller Optionen
```

Hier eine kleine Auswahl:

```
:set number    Zeilennummern anzeigen  
:set nonumber  Zeilennummern nicht anzeigen  
:set ignorecase beim Suchen soll nicht zwischen Groß- und  
              Kleinbuchstaben unterschieden werden  
:set noignorecase beim Suchen soll zwischen Groß- und  
              Kleinbuchstaben unterschieden werden  
:set showmatch bei Eingabe einer schließenden Klammer wird  
              die dazugehörige öffnende Klammer angezeigt  
:set noshowmatch keine Klammernprüfung  
:set autoindent automatische Einrückung  
:set noautoindent keine automatische Einrückung  
:set wrapmargin=n ab n Zeichen vor Zeilenende wird  
              automatisch an einer Wortgrenze getrennt  
              und eine neue Zeile begonnen
```
