Die wichtigsten Befehle in `vi`

Karl Ewald, Nov 1994, Nov 1995

Start

`vi [+`*Zeilennr*`] [`*Datei* ...`]`

WICHTIG: Modi

Eingabemodus: Texteingabe (`INPUT MODE`)
Kommandomodus: Befehlseingabe

falls noch nicht geschehen:
`:set showmode` *(nicht auf allen Systemen)*

Cursortasten und Backspace funktionieren auch im Eingabemodus. *(nicht auf allen Systemen)*

Eingabe- → Befehlsmodus: Esc
Befehls- → Eingabemodus: `i a I A o O s R`

Cursorbewegung

`h` ←   `l` →   `k` ↑   `j` ↓

- `w` Ein Wort nach rechts (`W` ein großes Wort nach rechts)
- `b` Ein Wort nach links (`B` ein großes Wort nach links)
- `0` (Null) An den Zeilenanfang
- `$` An das Zeilenende
- `%` Klammernsuche
  - auf Klammer: zum Gegenstück
  - vor (: vorwärts zur deren Gegenstück
  - vor): zurück zur deren Gegenstück
  - zwischen letzter Klammer und Textende: keine Aktion
- `Ctrl-B` Eine Bildschirmseite zurück
- `Ctrl-F` Eine Bildschirmseite vorwärts
- *Zeilennr*`G` Gehe zu Zeile (0=Dateiende)

Marken

Marken werden mit Kleinbuchstaben (*klbu*) bezeichnet.

- `m`*klbu* Marke setzen
- `'`*klbu* zu Markenzeile springen
- `` ` ``*klbu* genau zur Marke springen

Texteingabe

Einfügen ...

- `i` (*insert*): vor (auf) der Cursorposition
- `a` (*append*): nach der Cursorposition
- `I` am Zeilenanfang
- `A` am Zeilenende
- `o` (*open*): in eine neue Zeile nach der aktuellen
- `O` in eine neue Zeile vor der aktuellen

Ersetzen ...

- `r` ein Zeichen durch ein anderes (dann weiter im Befehlsmodus)
- `s` ein Zeichen durch eine Zeichenfolge
- `R` zeichenweise durch die Eingabe (Überschreiben)
- `c`*Kommando* Text zwischen Cursorposition und Ziel von *Kommando* durch Eingabe (Zeilen- oder Zeichenmodus je nach *Kommando*)
- `C` den Rest der Zeile
- `~` vertauschen Groß-/Kleinschreibung

Löschen

- `x` Zeichen auf der Cursorposition
- `d`*Kommando* Text zwischen Cursorposition und Ziel von *Kommando* (Zeilen- oder Zeichenmodus je nach *Kommando*)
- `dd` aktuelle Zeile
- `D` den Rest der Zeile
- `J` den Zeilenumbruch nach der aktuellen Zeile

Der zuletzt mit `d` gelöschte Text befindet sich in einem Puffer.

- `p` Puffer nach dem Cursor (Zeichen/Zeile) einfügen
- `P` Puffer vor dem Cursor (Zeichen/Zeile) einfügen

`y` statt `d` kopiert in den Puffer, ohne zu löschen.

`"`*klbu* (vor `d`/`y`/`p` Befehl) benannten Puffer verwenden
`"`*n* (*n*=2..9), bei `p` und `P`: ältere Löschpuffer (`"1p` ist `p`)

Rückgängig

- `u` letzte Operation rückgängig machen
- `U` Änderung der aktuellen Zeile rückgängig machen

Suche

- `/`*regexp* Suche vorwärts
- `?`*regexp* Suche rückwärts
- `n` wiederhole Suche gleiche Richtung
- `N` wiederhole Suche Gegenrichtung

Dabei enthält *regexp*:

- `.` ein beliebiges Zeichen
- `*` null- oder mehrmalige Wiederholung des vorigen Zeichens
- `\`*Spezialzeichen* das *Spezialzeichen* selbst
- `[`*Zeichen*|*Bereich* ...`]` ein Zeichen aus der Menge
- `[^`*Zeichen* ...`]` alles außer der Menge
- `^` Zeilenanfang
- `$` Zeilenende
- `\(` Beginn Gruppe
- `\)` Ende Gruppe

Ersetzen

`:[`*Anfang*, *Ende*`]s/`*Alt*`/`*Neu*`[/`*Optionen*`]`

In *Neu* kann stehen:

- `\`*n* *n*te Gruppe
- `&` ganzer Trefferstring

Optionen sind:

- `g` für global (auch mehrmals pro Zeile)
- `c` (*confirm*): mit Rückfrage (`y` = Ja, sonst Nein)

Dateioperationen

- `:w [`*Name*`]` speichert Text [unter *Name*]
- `:w!` *Name* speichert Text unter Name, auch wenn die Datei überschrieben wird
- `:r` *Name* fügt Datei *Name* nach Cursorzeile in aktuellen Text ein
- `:0r` *Name* fügt Datei *Name* am Textbeginn ein
- `:q` beendet `vi`
- `:wq` speichert Text und beendet `vi` (ebenso `ZZ`)
- `:e` *Name* lädt neuen Text statt aktuellem
- `:n` geht zur nächsten Datei (wenn mehrere auf Kommandozeile angegeben wurden)
- `:e#` geht zur vorher editierten Datei
- `:rewind` geht wieder durch Dateiliste
- `:q! :e! :n!` auch wenn dadurch Änderungen verloren gehen