

Vi-Referenz

Hilfe

zeige Hilfe an `:help^M`

In Vim kann mit `^J` (Link folgen) und `^T` (zurück) durch die Hilfe navigieren. Außerdem kann man auch gleich einen Befehl als Argument an `help` übergeben. Die Hilfe verlässt man mit wieder mit `:q^M`.

Beenden

beende mit Speichern `ZZ, :x^M`
beende ohne Speichern `:q^M`
erzwinge Beenden ohne Speichern `:q!^M`

Bewegen

Vor die meisten Bewegungskommandos kann man eine Zahl n schreiben, um sie n -mal auszuführen. Viele Editierkommandos müssen von einer Bewegung gefolgt werden.

links/runter/rauf/rechts `h, j, k, l`
nächstes/vorheriges Wort `w, b`
nächstes/vorheriges d. Leerzeichen getrenntes Wort `W, B`
Ende des (d. Leerzeichen getrennten) Wortes `e, E`
nächster/vorheriger Satz `), (`
nächster/vorheriger Absatz `}, {`
nächster/vorheriger Abschnitt `]l, [l`
Zeile n `nG, :n^M`
Anfang/Ende der Datei `gg, G`
Anfang/Ende der Zeile `0, $`
erstes nicht-Leerzeichen der Zeile `^`
nächstes/vorheriges Zeichen c in der Zeile `fc, Fc`
vor nächstes/vorheriges Zeichen c in der Zeile `tc, Tc`
nächstes/vorheriges Vorkommen von r `/r^M, ?r^M`
wiederhole letzte Suche (in Gegenrichtung) `n, N`
zu korrespondierender Klammer `%`
zur Markierung c (gesetzt mit `mc`) `'c`
Anfang/Mitte/Ende des Bildschirms `H, M, L`

Einfügen und Ersetzen

Die meisten dieser Kommandos wechseln in den Eingabemodus, aus dem man mit Escape (`^I`) wieder herauskommt. Um Kontrollzeichen einzugeben, muss man ihnen ein `^v` voranstellen.

füge vor/nach dem Cursor ein `i, a`
füge am Anfang/Ende der Zeile ein `I, A`
füge in neuer Zeile unter/über der aktuellen ein `o, O`
überschreibe `R`
ändere Zeichen unter dem Cursor in c `rc`
ersetze Zeichen, die m überspringt `cm`
ersetze Zeile/bis zum Ende der Zeile `cc, C`

Löschen und Kopieren

lösche Zeichen unter/vor dem Cursor `x, X`
lösche Zeichen, die die Bewegung m überspringt `dm`
lösche Zeile/Rest der Zeile `dd, D`

kopiere Zeichen, die m überspringt, in Register `ym`
kopiere Zeile `yy, Y`
füge Register hinter/vor dem Cursor ein `p, P`
Verbinde Zeilen `J`
führe Kopier-/Einfügebefehl e mit Register c aus `"ce`

Rückgängig und Wiederholen

mache letzte Änderung der Datei/Zeile rückgängig `u, U`
wiederhole letzten Befehl `.`
stelle letztes rückgängig Machen wieder her (Vim) `^R`

Suchen und Ersetzen

ersetze pat durch str `:s/pat/str/f^M`
 pat ist dabei ein regulärer Ausdruck, f (Flags) kann u.a. `g` (mehrere Ersetzungen pro Zeile) oder `c` (bei jeder Ersetzung nachfragen) sein. Siehe auch *Bereiche*, um in mehreren Zeilen zu ersetzen.

Shell

führe Shell-Befehl $shcmd$ aus `!:shcmd^M`
filtere Bereich r durch Shell-Befehl $shcmd$ `:r!shcmd^M`
filtere Zeilen, die m überspringt, durch Shell-Befehl s `!ms^M`

Dateien

schreibe Puffer in Datei $file$ `:w file^M`
editiere (neue) Datei $file$ `:e file^M`
nächste/vorherige Datei aus Argumentliste `:n^M, :N^M`
füge Inhalt von $file$ ein `:r file^M`
füge Ausgabe des Kommandos $shcmd$ ein `:r !shcmd^M`

Makros

definiere Abkürzung `:ab lhs rhs^M`
entferne Abkürzung `:unab lhs^M, :unab rhs^M`
definiere Befehlsmodus-Makro `:map lhs rhs^M`
definiere Eingabemodus-Makro `:map! lhs rhs^M`
lösche Makro `:unm lhs^M, :unm! lhs^M`
nimm Eingaben in Register r auf (Vim) `qr...q`
führe den Inhalt des Registers r aus `@r`

Bereiche

Die meisten Befehle, die mit einem Doppelpunkt beginnen (Ex-Kommandos), können nur auf einen bestimmten Bereich angewendet werden, indem man ihnen einen Bereich voranstellt, ansonsten beziehen sie sich nur auf die aktuelle Zeile.

Zeilen n bis m `n,m`
aktuelle Zeile `.`
aktuelle Zeile + n `+n`
letzte Zeile `$`
ganze Datei `%, 1, $`
alle Zeilen, auf die das Muster pat passt `g/pat/`