```
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
Thema:   "Vim − der Supereditor"                (C) 2007 T.Birnthaler OSTC GmbH
Autor:   Thomas Birnthaler
Ort:     KNF Kongress 2007 in Nürnberg
Datum:   So, 25.11.2007, 14:30−15:30
WWW:     http://www.ostc.de/vim−supereditor−25−11−2007.pdf
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

* Unterschiede Vim <−> Normaler Editor
    Editor für Programmierer (insb. C/C++−Programmierung, Tags)
    Modus−orientiert
    Logische Struktur (Kommando = Buchstabe = englischer Begriff)
    Bedienung per (Buchstaben)Tastatur (Cursor−/Funktions−Tasten, Maus unnötig)
    10−Fingersystem sollte man können (!)
    Viele Bewegungskommandos (auch Textauswahl)
    Wiederholungfaktor für Kommandos
    Editierbefehl = Komb. Wiederholungsfaktor + Basis−Operation + Bewegung
    Exzessiver Gebrauch der im Vi freien Tasten: g z Ctrl−w v/V/Ctrl−V
    Externe Programme als Textfilter aufrufbar (z.B. "sort")
    Reguläre Ausdrücke für Suchen/Ersetzen (+/− Offset)
    26 lokale + 26 globale Zwischenspeicher
    Eigene Funktionen und Kommandos
    Abkürzungen und Makros
    Batch−fähig
    Charityware (Uganda) − http://www.vim.org
    usw...

* Start−Modi
    ex                  = Zeilenorientierter Vorläufer (Batch−geeignet)
    vi                  = Veralteter Vorläufer (oder Alias)
    vim                 = Editor
    view                = Readonly
    vimdiff             = Unterschiede zw. 2 Dateien anzeigen/bearbeiten
    gvim gview gvimdiff = GUI−Version
    evim eview          = Easy Mode (immer im Einfügemodus)
    rvim rview          = restricted (keine Shell / externe Kommandos startbar)
    rgvim rgview                     (nicht in Hintergrund schaltbar mit Ctrl−z)

* Vim−Modi
    n = normal           = Vim−Kommandos (nach ESC)
    i = insert           = Einfügen/Ersetzen (nach a A i I o O r R s S c C)
    v = visual           = Visual (nach v V Ctrl−v)
    o = operator−pending = Auf Operator wartend (z.B. nach d...  c...  y...)
    c = command−line     = Ex−Kommandos und Suchen (: / ?)

* Automatisch Dateityperkennung (statt ^M am Zeilenende)
    :set fileformats=unix,dos,mac   = Prinzipielle Erkennungsreihenfolge
    :set fileformat=dos             = Zeilenende <CR><NL>
    :set fileformat=unix            = Zeilenende <NL>
    :set fileformat=mac             = Zeilenende <CR>

* Eingebaute Hilfe
    Hypertext
    Weitreichend und umfassend
       129 Dateien (ASCII *.txt)
   117.360 Zeilen
       4,7 MByte
    /usr/share/vim/(vim)current/doc/* (./vim2html.pl tags *.txt −> HTML−Form)

    :help                     = Hilfeeinstieg
    :help TOPIC<Ctrl−d>       = Zu TOPIC pass. Hilfethemen anzeigen (display)
    :help TOPIC               = Hilfe anzeigen zu TOPIC
    :he   TOPIC               = Hilfe anzeigen zu TOPIC
    :q<CR>                    = Hilfe beenden (quit)
    :help tutor               = Hilfe zu Tutorprogramm
    :help index               = Hilfe−Index
    :help user−manual         = Hilfe User−Manual

  Präfixe für Hilfethemen   Präfix   Beispiel
    Normal mode cmd         (nix)    :help d
    Control character       Ctrl−    :help Ctrl−u
    Visual mode cmd         v_       :help v_u
    Insert mode cmd         i_       :help i_<Esc>
    ex mode cmd             :        :help :quit
    Command−Line edit       c_       :help c_<Del>
    Vim cmd arg             −        :help −r
    Option                  ’        :help ’textwidth’

    :mouse=a          = Maus überall funktionsfähig (nicht gut!)
    :mouse=h          = Maus nur in Hilfe funktionsfähig (Hyperlink anklickbar!)
    Ctrl−]            = Referenz |Tag| folgen (deu Tastatur: Ctrl−AltGr−9!)
                        (:map ^ ^V^]  =  Ctrl−V Ctrl−V Ctrl−V Ctrl−AltGr−9)
```

```
      Ctrl−t / Ctrl−o   = Vor/zurückspringen (to/old)
      Ctrl−d + TAB      = Vervollständigen

  * Datei wiederherstellen (Recover, "screen" wäre natürlich besser!)
      :set directory=.,~/tmp,/var/tmp,/tmp   = Bevorzugte Stellen (schreibbar!)
      :set updatecount=200                   = Alle 200 Zeichen
      :set updatetime=4000                   = Alle 4 Sekunden
      :set fsync                             = Datei sofort auf Platte sync.
      :set swapsync=fsync                    = Swap sofort auf Platte sync.
      vim −r                                 = Liste wiederherstellbarer Dateien
      vim −r FILE                            = FILE wiederherstellen (.FILE.swp)

  * Cursor besser sichtbar machen ("Fadenkreuz")
      :set cursorline               = Cursorzeile hervorheben
      :set cursorcolum              = Cursorspalte hervorheben
      :highlight CursorLine ...     = Hervorhebung festlegen
      :highlight CursorColumn ...   = Hervorhebung festlegen

  * Zeilen numerieren
      :set number              = Zeilennummern anzeigen
      :set numberwidth=NN      = Breite der Zeilennummern festlegen
      :highlight LineNr ...    = Hervorhebung der Zeilennummer festlegen

  * "Undo" und "Redo undo" beliebig oft
    Vor Änderungen am Text braucht man keine Angst haben, alles zurücknehmbar
       u      = Undo
       Ctrl−R = Redo undo
       U      = Überflüssig (!)

  * Digraphs (Sonderzeichen)
      :dig         = Vollständige Liste aller Digraphen
      Ctrl−k a :   = ä (key/combine)
      Ctrl−k : a   = ä
      Ctrl−k C o   = ©
      :set digraph = Kombination von 2 Zeichen mit <BS> aktivieren
      a <BS> :     = ä
      : <BS> a     = ä

  * Drucken
      :hardcopy            = Text ausdrucken (unter Windows erscheint Druckdialog)
      :hardcopy!           = Text sofort ohne Druckdialog ausdrucken (Windows)
      :set printdevice=...
      :set printexpr=...
      :set printoptions=...
      :set printfont=...

  * Wiederholen (sehr wichtig!)
      NNN CMD     = Kommando NNN−mal ausführen
      .           = Letztes Kommando wiederholen
      u Ctrl−R    = Undo / Redo undo
      p P         = Letzten gelöschten Text wieder einfügen (put/paste)
      , ;         = Letzte t/T/f/F−Suche wiederholen / umgekehrt wiederholen
      n N         = Letzte /.../−Suche wiederholen / umgekehrt wiederholen (next)
      // ??       = Letzte /.../−Suche wiederholen
      /<CR> ?<CR> = Letzte /.../−Suche wiederholen
      & :s<CR>    = Letzte Suche+Ersetzung wiederholen
      Ctrl−N      = Passendes Wort vorwärts suchen (next)
      Ctrl−P      = Passendes Wort rückwärts suchen (previous)
      !!          = Letzten Betriebssystembefehl wiederholen (auf akt. Zeile)

  * Suchen (wiederholen, inkrementell, schlau, alle Treffer markieren)
      *  #             = Wort unter Cursor suchen (vorwärts/rückwärts, exakt)
      g*  g#           = Wort unter Cursor suchen (vorwärts/rückwärts, Teil)
      //  ??           = Letzte Suche wiederholen
      /<CR>  ?<CR>     = Letzte Suche wiederholen
      n  N             = Letzte Suche gleiche Richtung / andere Richtung (next)
      Ctrl−o           = Zur vorherigen Suchstelle zurück (old)    zurueck
      :set incsearc    = Inkrementell suchen während tippen
      :set hlsearch    = Alle Treffer markieren
      :set ignorecase  = GROSS/kleinschreibung ignorieren
      :set smartcase   = GROSS/kleinschreibung intelligent berücksichtigen
                         Nur Kleinbuchstaben oder \c in Muster −> ignorecase
                         EIN Grossbuchstaben oder \C in Muster −> noignorecase
      :set wrapscan    = Über Dateiende/anfang hinwegsuchen
      :nohlsearch      = Such−Markierungen entfernen (stören manchmal)
      :invhlsearch     = Such−Markierung an/ausschalten
                         (:nnoremap <CR> :set invhls<CR>/<BS>)

  * Kommando−Historie
      :history=500
    Für 5 Eingabearten vorhanden
      :−Kommandos (ex)
```

```
      /?−Suche
      Ausdrücke
      Eingabezeilen (input()−Funktion)
      Debug−Kommandos
    Bedienung
      Cursor−^/v     = In Kommandoliste blättern (auf/ab)
      Cursor−</>     = Zeichenweise in altem Befehl bewegen (links/rechts)
      Ctrl−Cursor−<> = Wortweise in altem Befehl bewegen (oder Shift−...)
      Ctrl−b  Ctrl−e = Zu Zeilenanfang/ende bewegen (begin, end)
      ESC            = Abbrechen

  * Visual Modus
      v      = Zeichen markieren
      V      = Zeilen markieren
      Ctrl−v = Block markieren (Spaltenbereich)
    Mit Bewegung (h j k l w b e Ctrl−f Ctrl−b ...) erweiterbar (mehrfach!)

  * Operationen auf markiertem Text (Visual)
      c d y  = Ändern / Löschen / Merken (change delete yank)
      c r s  = Ändern (change replace substitute)
      C R S  = Ändern (Ctrl−V = Rechteck, sonst ganze Zeile)
      x X Y  = Löschen ("cross out") / Merken (ganze Zeile)
      D      = Löschen (Ctrl−V = Rechteck, sonst ganze Zeile)
      ~ U u  = GROSS/kleinschreibung invertieren / GROSS / klein (uppercase)
      J gJ   = Zeilen verketten (mit / ohne Space am Ende) (join)
      g?     = ROT13 kodieren
      < >    = Ausrücken / Einrücken
      !      = Filtern (UNIX−Kommando, z.B. "sort")
      =      = Formatieren mit externem Programm (equalprg)
      gq     = Formatieren gemäß Option 'textwidth'
      :      = Ex−Kommando aufrufen (:'<,'>)
      Ctrl−] = Tagsuche (Strg−AltGr−9)
      0 $    = Überall bis Zeilenanfang/ende markieren (^ nicht mögl.)
      o O    = Zur diagonal gegül. Ecke/Ecke in gleicher Zeile springen (other)

  * Visual Mode
      :  = '<,'>:EX−BEFEHL<CR>
      !  = '<,'>!OS−KOMMANDO<CR>
      gv = Letzten Visual−Block erneut markieren (go visual)

  * Textobjekte
      :he text−objects
    Cursor INNERHALB Objekt, Ausdehnung nach beiden Seiten!
    Nach Visual−Kmdo oder vor c/d/y/... verwendbar (auch bei Faltung)
    Kürzel: i=inner, a=an/all/outer (inner ist immer weniger als outer!)
      Absatz = Leerzeile begrenzt                    (:he sentence)
      Satz   = .!? + Leerzeichen/Zeilenende begrenzt (:he paragraph)
      wort   = Folge von a−zA−Z_0−9                  (:he word, :set isident=...)
      WORT   = Folge von NICHTLeerraum               (:he WORD)
    Wiederholungsfaktor davor vervielfacht Treffer/trifft auch Verschachtelung
      aw iw = Wort             mit/ohne Space       (word)
      aW iW = WORT             mit/ohne Space       (WORD)
      as is = Satz             mit/ohne Space       (sentence)
      ap ip = Absatz           mit/ohne Space       (paragraph)
      at it = Text in HTML−Tag mit/ohne <tag></tag> (tag)           Analog
      ab ib = Text in (...)    mit/ohne ()          (block/brace)   a( i( a) i)
      aB iB = Text in {...}    mit/ohne {}          (Block/Brace)   a{ i{ a} i}
      a[ i[ = Text in [...]    mit/ohne []                          a] i]
      a< i< = Text in <...>    mit/ohne <>                          a> i>
      a" i" = Text in "..."    mit/ohne ""          (String)
      a' i' = Text in '...'    mit/ohne ''          (String)
      a` i` = Text in `...`    mit/ohne ``          (String)
    Beispiele
      <div class='intro'><p>Der <a href="#top">Name</a> ist Hase</p></div>
      echo "Ein String mit 'Hochkomma' und Kommando `ls −l /etc` drin"
      ({[< ({[< ({[< ({[< text >]}) >]}) >]}) >]})
      /* /* /* C−Kommentar */ */ */ ist damit leider nicht matchbar

  * Wort−Definition frei wählbar
      :set isident=@,48−57,_,192−255                  = Bezeichner    (identifier)
      :set iskeyword=a−z,A−Z,48−57,_,.,−,>,:          = Schlüsselwort (keyword)
      :set isfname=@,48−57,/,.,−,_,+,,,#,$,%,~,=      = Dateiname     (filename)
      :set isprint=@,~−255                            = Druckbares Zeichen

  * Insert−Vervollständigung (Completion)
      :help ins−completion
      Ctrl−N        = Passendes Wort vorwärts suchen      (next)
      Ctrl−P        = Passendes Wort rückwärts suchen     (previous)
      Ctrl−X Ctrl−K = Passenden Dictionary−Eintrag suchen (keyword)
      Ctrl−X Ctrl−F = Passenden Dateinamen suchen         (filename)
      Ctrl−X Ctrl−L = Passende vollständige Zeile suchen  (line)
```

```
    * Rechtschreibung
        :he spell               = Hilfe (sehr ausführlich)
        :set spell              = Schreibfehler markieren
        :set spellcapcheck=...  = Wort nach Satzende muss GROSS anfangen
        :set spellfile=...      = Eig. Datei (un)gültiger Worte ("zg" und "zw")
        :set spelllang=de       = en, en_us, de, fr, es, ... (alles klein!)
        :set spelllang=de,en,it = Mehrere Sprachen durch Komma getrennt
        :set spellsuggest=NN    = Max. NN Korrekturvorschläge (best,double,fast)
        :highlight SpellBad     = Unbekanntes Wort
        :highlight SpellCap     = Wort nicht groß geschrieben ('spellcapcheck')
        :highlight SpellRare    = Seltenes Wort
        :highlight SpellLocal   = Falsche Schreibweise für ausgewählte Region
        z=                      = Korrektur−Vorschläge zu Wort unter Cursor
        zg                      = Wort unter Cursor zu eig. Wortliste dazu (good)
        zw                      = Wort unter Cursor aus eig. Wortliste weg (wrong)
        zug                     = Undo "zg" (good word)
        zuw                     = Undo "zw" (wrong word)
      Beispiel
        :set spell spl=de,en spc sps=20 spf=~/.vim/spell/de.latin1.add
        :set nospell
      Englische Regionen
        en      = Alle
        en_au   = Australien
        en_ca   = Canada
        en_gb   = England
        en_nz   = Neusealand
        en_us   = USA
      Deutsche Regionen
        de      = Alle
        de_de   = Alte und neue Rechtschreibung
        de_19   = Alte Rechtschreibung
        de_20   = Neue Rechtschreibung
        de_at   = Österreich
        de_ch   = Schweiz
      Weiter Wörterbücher zum Download
        ftp://ftp.vim.org/pub/vim/runtime/spell

    * Autokommandos (Autocmd)
        Ereignis + Dateinamen−Match −> Aktion / Keine Aktion
          :autocmd EVENT FILEPAT CMD
          :autocmd! EVENT FILEPAT CMD
        EVENT
          FileReadPre  FileReadPost
          FileWritePre FileWritePost
          BufReadPre   BufRead(Post)
          BufWritePre  BufWrite(Post)
          BufNewFile
          WinEnter     WinLeave
          VimEnter     VimLeave
          ...          ...
        FILEPAT
          Analog Shell−Muster
        CMD
          Ex−Kommandos (mehrere werden gesammelt)
      Beispiel
        :autocmd  BufReadPre,FileReadPre  *.[chy]  set cindent
        :autocmd  BufNewFile              *.pl     0r ~/.vim/template/pgm.pl

    * Multiwindowing (Fenster)
        Ctrl−w n       = Neues Fenster mit leerer Datei öffnen (new)
        Ctrl−w s       = Neues Fenster mit gleicher Datei öffnen (split)
        Ctrl−w c       = Fenster schließen (close)

        Ctrl−w h j k l = Fenster links/unten/oben/rechts
        Ctrl−w t       = Oberstes Fenster (top)
        Ctrl−w b       = Unterstes Fenster (bottom)
        Ctrl−w p       = Vorheriges Fenster (preceding/previous)
        Ctrl−w Ctrl−w  = Vorheriges Fenster (window)
        Ctrl−w w       = Vorheriges Fenster (window)

        :wall          = Alle Fenster schreiben (write)
        :qall          = Alle Fenster verlassen (quit)
        :qall!         = Alle Fenster auf jeden Fall verlassen (VORSICHT!)
        :wqall         = Alle Fenster schreiben und verlassen (write quit)

        :argument ...  = Dateiliste angeben
        :all           = Ein Fenster pro angegebener Datei
        :split         = Neues Fenster mit gleicher Datei öffnen (horizontal)
        :vsplit        = Neues Fenster mit gleicher Datei öffnen (vertikal)
        :new           = Neues Fenster mit leerer Datei öffnen

        Ctrl−^         = Alternative Datei anzeigen
```

```
        Ctrl−w Ctrl−^  = Alternative Datei in anderem Fenster anzeigen
        Ctrl−w Ctrl−i  = Fenster Split + Suche nach Wort unter Cursor

        Ctrl−w _       = Maximale Höhe für Fenster
        Ctrl−w +       = Zeile mehr für Fenster
        Ctrl−w −       = Zeile weniger für Fenster
        Ctrl−w =       = Alle Fenster gleich hoch
        z N <CR>       = Fenster auf Höhe N setzen (zoom)

    * Kurioses
        gm              = Sprung zur Zeilenmitte ('textwidth')
        N|              = Sprung zur Spalte N
        N%              = Sprung zur prozentualen Zeile gemäß ganzer Datei
        %               = Sprung zur korresp. Kl., /*...*/−Kmtr, #if−#else−#endif
        Ctrl−A          = Zahl hochzählen (auch negative!)     −100   025   0x0fd
        Ctrl−X          = Zahl runterzählen (auch negative!)
        Ctrl−Y          = Zeichen in Zeile darüber kopieren (im Einfügemodus)
        Ctrl−E          = Zeichen in Zeile darunter kopieren (im Einfügemodus)
        g?              = ROT13−Verschlüsselung
        ga              = ASCII−Code des akt. Zeichens (in Statuszeile)
        gf              = Dateiname unter Cursor öffnen (Ctrl−O = zurück)
        :help 42        = The meaning of life (Douglas Adams)
        Ctrl−V Return   = Carriage Return eingeben = "^M" (verbose)
        Ctrl−V 123      = Zeichen mit ASCII−Code 123 eingeben = "{" (verbose)
        :set insertmode = Easy−Mode (Vim−Novizen, Insertmode + Ctrl−O = out)
        :set revins     = Text von rechts nach links eingeben (Hebräisch, Farsi)
        g Ctrl−g        = Zeichen + Worte + Zeilen zählen + akt. Position ausgeben
        K auf Befehl    = Manpage zu Befehl anzeigen
        :g/^/m 0        = Reihenfolge aller Zeilen rumdrehen (g=global, m=move)
        1GVG~           = Alle klein− und GROSSbuchstaben vertauschen
        1GVGu           = Alle GROSS− in kleinbuchstaben umwandeln (U=umgekehrt?)

    * Options−Typen
        Boolesch   = z.B. compatible + nocompatible
        Numerisch  = z.B. shiftwidth=4
        String     = z.B. shell=/bin/sh

    * Optionen Langform + Kurzform (2−3 Bst., nicht immer vorhanden)
        shiftwidth + sw
        showmode   + sm
        number     + nu

    * Optionen anzeigen/setzen/löschen
        :set               = Alle abweichend vom Standardwert anzeigen
        :set all           = Alle (außer Terminaloptionen) anzeigen
        :set all&          = Alle Optionen auf Stdwert setzen (außer Terminal)
        :set termcap       = Terminaloptionen anzeigen
        :set OPTION?       = Optionswert anzeigen
        :set OPTION        = Boolsche Option setzen
        :let &OPTION = 1   = Boolsche Option setzen
        :set noOPTION      = Boolsche Option abschalten
        :set OPTION!       = Boolsche Option invertieren (!)
        :set invOPTION     = Boolsche Option invertieren (!)
        :set OPTION&       = Option auf Stdwert setzen (!)
        :set OPTION=...    = Zahl/String−Option setzen (Leerz. nur VOR = ok!)
        :set OPTION:...    = Zahl/String−Option setzen (Leerz. nur VOR : ok!)
        :set OPTION+=...   = Option hinten erweitern (bzw. num. addieren)
        :set OPTION−=...   = Option reduzieren       (bzw. num. subtrahieren)
        :set OPTION^=...   = Option vorne erweitern  (bzw. num. multiplizieren)

    * Hilfe zu Optionen
        :help 'OPT            = Hilfe zu Option OPT
        :help option−list     = Alphabetisch sortiert, 1−zeilig
        :help option−summary  = Alphabetisch sortiert, ausführlich
        :help options         = Vollständige Anleitung zu Optionen

    * Optionen interaktiv editieren
        :opt     = Nach Themengebieten sortiert
        Ctrl−w _ = Fenster maximieren
        <ENTER>  = Auf Überschrift −> Sprung zu Optionen
        <ENTER>  = Auf Option−Name −> Sprung zu vollständiger Erklärung
        :q       = Erklärung schließen
        <ENTER>  = Auf set Boole−Option −> hin/herschalten


    * Links−Rechts scrollen (lange Zeilen nicht umbrechen)
        :set nowrap            = Nicht umbrechen (= scrollen)
        :set sidescroll=5      = Scrollbreite
        :set sidescrolloff=5   = Fester Rand links und rechts
        zl = Cursor nach links scrollen
        zh = Cursor nach rechts scrollen
        zs = Cursor zum linken Rand scrollen (start)
```

```
      ze = Cursor zum rechten Rand scrollen (end)

  * Zeilen umbrechen (reine Darstellung, kein echter Umbruch)
      :set wrap
      :set linebreak                  = Statt bei Zeilenbreite an "schönen" Stellen
      :set breakat=\ ^I!@*−+;:,./?"   = Erlaubte Umbruchstellen
      :set showbreak=»                = Prefix vor Fortsetzungszeile
    In umbrochener Zeile (Bildschirmzeile)...
      g0 = zu Zeilenanfang
      g^ = zu 1. Zeichen
      g$ = an Zeilenende
      gj = 1 Zeile nach unten
      gk = 1 Zeile nach oben

  * Erweiterte Reguläre Ausdrücke
      \|     = Oder
      \+     = 1−oo Mal
      \=     = 0−oo Mal

      \{n,m} = n−m Mal  (greedy)      \{−n,m} = n−m Mal  (non−greedy)
      \{n}   = n Mal    (greedy)      \{−n}   = n Mal    (non−greedy)
      \{n,}  = n−oo Mal (greedy)      \{−n,}  = n−oo Mal (non−greedy)
      \{,m}  = 0−m Mal  (greedy)      \{−,m}  = 0−m Mal  (non−greedy)

      \i     = Bezeichner−Z.    (isident)        \I = Ohne Ziffern
      \k     = Schlüsselwort−Z. (iskeyword)      \K = Ohne Ziffern
      \f     = Dateinamen−Z.    (isfilename)     \F = Ohne Ziffern
      \p     = Druckbares Z.    (isprintable)    \P = Ohne Ziffern
      \s     = Leerraum         (TAB, SPACE)     \S = KEIN Leerraum

      \b     = Backspace
      \e     = Escape
      \r     = Carriage Return
      \t     = Tabulator
      \n     = Reserviert (derzeit)

      \(..\) = Gruppieren/Merken       \1 \2 ... \9 = Gemerktes Element
      ~      = Letztes Suchmuster

      \l \u  = Folgenden Buchstaben klein/gross (lowcase uppercase)
      \L \U  = Folgende Buchstaben klein/gross bis \E

  * Viminfo
      ~/.viminfo
      Zustand aller Editiersitzungen speichern
        Ex−Kommandozeilen−Historie
        Such−Historie
        Register−Inhalte
        Datei−Marken
        Letztes Such/Ersetzungsmuster (n &)

  * Dateinamen
      %      = Aktueller Dateiname (current)
      #      = Vorheriger Dateiname (previous)
      Ctrl−^ = Alternative Datei laden (Wechsel zw. % und #, deu. Tast. Ctrl−&)

  * Source−Datei einlesen (include)
      :source FILE = Vim−Befehle einlesen (Include)
      :map <C−J> :w!<CR>:source ~/.vimrc<CR>

  * Externe Linux−Befehle (Filter) aufrufen (z.B. sort, uniq, fmt, ls)
      :!CMD   = Aufrufen und Ergebnis ansehen
      !CMD    = Auf akt. Zeile anwenden
      !!CMD   = Auf akt. Zeile anwenden
      :%!CMD  = Auf alle Zeilen anwenden
      :r!CMD  = Ergebnis nach akt. Zeile einfügen
      :0r!CMD = Ergebnis am Dateianfang einfügen

  #−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
  # Vim−Programmierung
  #−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

  * Kommentar
      " ...          Auf Zeile für sich
      Hinter Kommandos nur falls nicht als Bestandteil missinterpretiert
      set sw=1       "Kommentar erlaubt
      map <F2> "ap   "Kein Kommentar erlaubt

  * Tastaturcodes (für :map, Eingabe mit Ctrl−k + Taste)
      TIPP: Rauskriegen im Ex−Modus mit Ctrl−k + Taste

      <S−XXX>         = Shift   + Taste XXX
```

```
        <C−XXX>         = Control + Taste XXX
        <A−XXX>         = Alt     + Taste XXX
        <M−XXX>         = Meta    + Taste XXX (analog Alt)
        <D−XXX>         = Command + Taste XXX (nur Apple!)

        <F1>...<F35>    = Funktionstasten

        <ESC>           = ESC (Ctrl−[, 27)
        <BS>            = Backspace (Ctrl−h, 8)
        <BACKSPACE>     =   (analog)
        <TAB>           = Tabulator (Ctrl−i, 9)
        <NL>            = Newline (Ctrl−j, 10)
        <NEWLINE>       =   (analog)
        <LF>            = Linefeed (Ctrl−l, 12)         = Formfeed
        <LINEFEED>      =   (analog)
        <CR>            = Return (Ctrl−m, 13)
        <RETURN>        =   (analog)
        <ENTER>         =   (analog)

        <NUL>           = "\0" Zero−Byte
        <SPACE>         = " " Leertaste (32)
        <BSLASH>        = "\" Backslash (92)
        <LT>            = "<" (60)
        <BAR>           = "|" Pipe (124)

        <INS>           = Insert
        <INSERT>        =   (analog)
        <DEL>           = Delete (127)
        <DELETE>        =   (analog)

        <LEFT>          = Cursor links
        <RIGHT>         = Cursor rechts
        <UP>            = Cursor auf
        <DOWN>          = Cursor ab

        <HOME>          = Home
        <END>           = End
        <PAGEUP>        = Seite auf
        <PAGEDOWN>      = Seite ab

        <CSI>           = Command Sequence Intro (Alt−ESC, 155)
        <EOL>           = End of line (<CR>, <NL> oder <CR><NL>, System + fileformat)
        <HELP>          = Hilfe−Taste
        <UNDO>          = Undo−Taste

        <kHOME>         = Zahlentastatur Home
        <kEND>          = Zahlentastatur End
        <kPageUp>       = Zahlentastatur Seite auf
        <kPageDown>     = Zahlentastatur Seite ab
        <kPlus>         = Zahlentastatur +
        <kMinus>        = Zahlentastatur −
        <kMultiply>     = Zahlentastatur *
        <kDivide>       = Zahlentastatur /
        <kEnter>        = Zahlentastatur Enter
        <kPoint>        = Zahlentastatur Dezimalpunkt
        <k0>−<k9>       = Zahlentastatur Ziffer

        <Mouse>         = Mauscode−Einleitung
        <MouseUp>       = Scrollrad nach oben drehen
        <MouseDown>     = Scrollrad nach unten drehen
        <LeftDrag>      = Linke Maustaste ziehen
        <LeftMouse>     = Linke Maustaste drücken
        <LeftRelease>   = Linke Maustaste loslassen
        <MiddleDrag>    = Mittlere Maustaste ziehen
        <MiddleMouse>   = Mittlere Maustaste drücken
        <MiddleRelease> = Mittlere Maustaste loslassen
        <RightDrag>     = Rechte Maustaste ziehen
        <RightMouse>    = Rechte Maustaste drücken
        <RightRelease>  = Rechte Maustaste loslassen

    * Variablen
        :let VAR = ZAHL       = Zahl zuweisen
        :let VAR = "STRING"   = String zuweisen
        :echo VAR             = Variable ausgeben (Statuszeile)
        :unlet VAR...         = Löschen
        :unlet! VAR...        = Löschen ohne Warnung falls nicht existent
        :let &showmode = 1    = Option setzen
        :set showmode         = (analog)
        :let &showmode = 0    = Option zurücksetzen
        :set noshowmode       = (analog)

    * Variablen Gross/Kleinschreibung + Präfixe
```

```
        UPPERCASE = In viminfo−Datei wenn viminfo+=!
        Uppercase = Von :mksession gespeichert
        lowercase = Nicht in irgendeinem Save−File gespeichert
        $  = Environment (Shell)
        @  = Register (a−z)
        &  = Option des Vim
        b: = Buffer−lokal
        w: = Window−lokal
        g: = Global
        a: = Argument einer Funktion
        v: = Vim−intern
           = Funktions−lokal (nix!)

    * Vim−Interne Variablen
        v:count        = Zähler des letzten Normal−Mode Kommandos
        v:count1       = Analog, aber Default 1 wenn kein Zähler definiert
        v:errmsg       = Letzte Fehlermeldung
        v:warningmsg   = Letzte Warnung
        v:statusmsg    = Letzte Statusmeldung
        v:shell_error  = Exit−Status des letzten Shell−Kommandos (0=ok, sonst Fehler)
        v:this_session = Dateiname der letzten gespeicherten/gelesenen Session−Datei
        v:version      = Vim−Versionsnummer (5.01 als 501)

    * Ausgabe (Statuszeile)
        :echo ...     = Elemente durch Blank getrennt + Newline am Ende
        :echon ...    = Analog ohne Newline am Ende
        :echohl GROUP = Highlighting gemäß Gruppe einstellen
        :echohl None  = Highlighting abschalten
        :highlight    = Alle Highlightgruppen anzeigen

    * Funktions−Definition
        :function NAME(ARG1, ARG2,...)     = Müssen mit GROSSbuchstaben anfangen
        :function! NAME(ARG1, ARG2,...)    = Funktion überschreiben (! = force)
            ANWEISUNG                      = a:ARG1 = Argument−Variablen
            ...                            = Variablen immer lokal zu Funktion
            echo g:var                     = (außer g:VAR = global)
            :return RESULTAT
        :endfunction

      Beispiel

        :function Min(x, y)
        :   if a:x < a:y
        :       let min = a:x
        :   else
        :       let min = a:y
        :   endif
        :   return min
        :endfunction

    * Funktions−Aufruf
        :let erg = Min(10, 20)
        :RANGE call Min(10, 20)            = Für jede Zeile aus Bereich aufrufen

    * Funktionen auflisten und löschen
        :function                          = Liste der benutzerdefinierten Fkt.
        :function NAME                     = Inhalt von F. NAME auflisten
        :delfunction NAME                  = Funktion NAME löschen

    * Kontrollstrukturen
        :if COND
           ANWEISUNG
           ...
        :elseif COND
           ANWEISUNG
           ...
        :else
           ANWEISUNG
           ...
        :endif

        :while COND
           ANWEISUNG
           ...
           :continue
           ...
           :break
           ...
        :endwhile

        :for item in mylist
            :call remove(mylist, 0)
```

```
    :endfor

    Weitere

#−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
# Programmieren mit Vim
#−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

* Suche nach "unmatched" Klammern (gemäß Verschachtelungshierarchie)
    :set showmatch                    = Korresp. Kl. kurz markieren
    :set matchpairs=(:),[:],{:},<:>   = Welche korresp. Kl.typen markieren
    :set matchtime=5                  = Wie lange korresp. Kl. markieren (1/10s)
    %     = Sprung zur korresp. Klammer, /*...*/−Kommentar, #if−#else−#endif
    [(    = Sprung zur vorherigen nicht gematchten "("
    ])    = Sprung zur nächsten   nicht gematchten "("
    [{    = Sprung zur vorherigen nicht gematchten "{"
    [}    = Sprung zur nächsten   nicht gematchten "}"
    [#    = Sprung zum vorherigen nicht gematchten "#if" oder "#else"
    ]#    = Sprung zum nächsten   nicht gematchten "#else" oder "#endif"
    [* [/ = Sprung zum vorherigen nicht gematchten C−Kommentar "/*"
    ]* ]/ = Sprung zum nächsten   nicht gematchten C−Kommentar "*/"

* Suche nach Identifier unter Cursor
    :set include
    [i            = Anzeige 1. Vorkommen ab Dateianfang
    ]i            = Anzeige 1. Vorkommen ab akt. Position
    [I            = Anzeige aller Zeilen ab Dateianfang
    ]I            = Anzeige aller Zeilen ab akt. Position
    [ Ctrl−i      = Sprung zum 1. Vorkommen ab Dateianfang
    ] Ctrl−i      = Sprung zum 1. Vorkommen ab akt. Position
    Ctrl−w i      = Neues Fenster öffen mit 1. Vorkommen
    Ctrl−w Ctrl−i = Neues Fenster öffen mit 1. Vorkommen

* Suche nach Makro unter Cursor
    :set define
    [d            = Anzeige 1. Definition ab Dateianfang
    ]d            = Anzeige 1. Definition ab akt. Position
    [D            = Anzeige aller Definitionen ab Dateianfang
    ]D            = Anzeige aller Definitionen ab akt. Position
    [ Ctrl−d      = Sprung zu 1. Definition ab Dateianfang
    ] Ctrl−d      = Sprung zu 1. Definition ab Dateiposition
    Ctrl−w d      = Neues Fenster öffen mit 1. Definition
    Ctrl−w Ctrl−d = Definition Vorkommen ab Dateiposition

* Einrücken (Smart Indenting)
    :set autoindent        = Analog Zeile darüber (primitiv)
    :set smartindent       = Gemäß C (besser)
    :set cindent           = Gemäß C (noch besser)
    :set formatoptions=... = Formatierungsverhalten definieren
    :set comments=...      = Aussehen von Kommentar definieren
    Ctrl−D                 = Automatische Einrückung zurücknehmen
    TAB                    = Automatische Einrückung erweitern

* Textfaltung (z = gefaltetes Stück Papier)
  Verkürzte Darstellung von Text (bleibt unverändert)
  Für bessere Navigation/Arbeiten in langen Dokumenten/Programmen
  Schachtelbar
  Geschlossene Faltung wie eine Zeile behandelbar (umsortieren)
    :he fold.txt
    :he usr_28
    :he fold−commands
    :he fold−methods
    :he fold−manual
    :he fold−marker
    :he fold−create−marker
    :he fold−foldlevel
    :he fold−foldtext
    :he fold<Ctrl−D>
  Faltungs−Methode (set foldmethod=..., set fdm=...)
    :set foldmethod=...
      manual = Manuell definiert (Standard)
      marker = Marken im Text def. Faltung ({{{ und }}} bzw. {{{1 und }}}1)
      indent = Einrückungstiefe def. Faltung (Python!)
      syntax = Syntaxgesteuerte Faltung (nicht für jede Sprache!)
      expr   = Ausdrücke def. Faltung (Performance!)
      diff   = Unveränderter Text wird gefaltet (vimdiff)
  Optionen
    :he 'fold<Ctrl−D>
    :he 'fold<CR>
    foldclose=all   = Autom. schließen wenn Cursor außerhalb (Std: "")
    foldcolumn=10   = Rand links für Faltungstiefe freihalten (Std: 0)
    foldenable      = Alle Faltungen ein/ausschalten (zi toggelt)
```

```
        foldexpr=...    = Methode "expr: Pro Zl. ausgew. Ausdruck -> Tiefe (Std: 0)
        foldignore=#    = Methode "indent": Anfänge zu ign. Zeilen (Std: #)
        foldlevel=5     = Faltungen höheren Levels autom. schließen (Std: 0)
        foldlevelstart  = Startwert für 'foldlevel' (Std: −1)
        foldmarker={,}  = Methode "marker": Start/Ende−Marke (Std: {{{,}}})
        foldmethod=...  = manual,indent,expr,marker,syntax,diff
        foldminlines=.. = Min. Anz. Zeilen um Faltung durchzuführen (Std: 1)
        foldnestmax=... = Max. verschachtelte Faltungen (Std: 20 = Vim−Limit)
        foldopen=...    = Welche Kmdos öffnen Faltung wenn mit Cursor darin
                          (Std: block,hor,mark,percent,quickfix,search,tag,undo)
                          (Not: all,insert,jump)
        foldtext=...    = Für Faltung anzuzeig. Text (Std: foldtext())
      Bei jedem Wechsel der Methode alte Faltungen vergessen (außer "manual")
        V% = Block markieren
        zf = Faltung erzeugen (nur für "manual", "marker")  (fold)
        zo = Faltung unter Cursor öffnen                    (open)
        zO = Verschachtelte Faltungen vollständig öffnen    (Open)
        zc = Faltung unter Cursor schließen                 (close)
        zC = Verschachtelte Faltung vollständig schließen   (Close)
        zi = ALLE Faltungen aktivieren/deaktivieren         (switch, (no)foldenable)
        zR = Verschachtelte Faltungen vollständig öffnen    (Reduce, foldlevel)
        zr = Verschachtelte Faltungen vollständig öffnen    (reduce, foldlevel)
        zM = Verschachtelte Faltungen vollständig schließen (More, foldlevel)
        zm = Verschachtelte Faltungen vollständig schließen (more, foldlevel)
        zj = Faltungen vorwärts                             (j=Cursor abwärts)
        zk = Faltungen rückwärts                            (k=Cursor abwärts)
        :mkview   = Faltungen speichern
        :loadview = Faltungen einlesen
      Beispiel
        # Beginn
        {
            {
                {
                    {
                        dies ist Text
                    }
                }
            }
        }

    * (Re)Formatierung
        :he formatting
        :R left [OFS]   = Bereich linksbündig (Rand OFS)
        :R right [WID]  = Bereich rechtsbündig (Breite WID oder Opt. 'textwidth')
        :R center [WID] = Bereich zentriert (Breite WID oder Opt. 'textwidth')

        gqBEWEGUNG = Ausgewählten Bereich formatieren
        gwBEWEGUNG = Ausgewählten Bereich formatieren (Cursor bleibt stehen)
        gqq        = Zeile formatieren
        gww        = Zeile formatieren (Cursor bleibt stehen)
        VISUALgq   = Visuell markierten Bereich formatieren
        VISUALgw   = Visuell markierten Bereich formatieren (Cursor bleibt stehen)

    * Statuszeile formatieren
        :help statusline
        :set statusline=FORMAT = Format der Statuszeile festlegen (riesig!)
        :set statusline=%t\ [%M%r%{&ff}]\ [%L/%04l,%04v,%p%%]
        :set laststatus=2      = Statuszeile ständig anzeigen (gut)
        :set laststatus=0      = Statuszeile + Kommando−Zeile kombinieren (schlecht)
```