

```
-----
Thema: "Vim - der Supereditor"                                (C) 2007 T.Birnthaler OSTC GmbH
Autor: Thomas Birnthaler
Ort: Practical-Linux in Gießen
Datum: Sa, 27.10.2007, 16:35-17:50
WWW:   http://www.ostc.de/vim-supereditor-27-20-2007.pdf
-----
```

- * Unterschiede Vim <-> Normaler Editor
 - 10-Fingersystem
 - Editor für Programmierer (insb. C/C++-Programmierung, Tags)
 - Modus-orientiert
 - Extrem logische Struktur (Kommando = Buchstabe = englischer Begriff)
 - Bedienung per (Buchstaben)Tastatur (Cursort., Funktionstasten, Maus unnötig)
 - Sehr viele Bewegungskommandos
 - Textauswahl durch Bewegungskommandos
 - Wiederholungsfaktor für Kommandos
 - Kommando = Kombination von Wiederholung + Operation + Bewegungskommandos
 - Externe Programme zur Textmanipulation aufrufbar ("sort")
 - Suche mit Regulären Ausdrücken (+/- Offset)
 - 26 + 26 Zwischenspeicher
 - Eigene Funktionen und Kommandos
 - Abkürzungen und Makros
 - Batch-fähig
 - :global/PATTERN/-Kommando
 - Charityware (Uganda) - <http://www.vim.org>
 - ...

- * Starten in verschiedenen Modi (r=restricted, view=read only, g=gui, diff)
 - ex
 - (r)vim
 - (r)view
 - vimdiff
 - (r)gvim
 - (r)gview
 - gvimdiff
 - restricted = Keine Shell oder externe Kommandos startbar
 - Editor nicht in Hintergrund schaltbar
 - ...

- * 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
 - :set fileformat=dos (<CR><NL>)
 - :set fileformat=unix (<NL>)
 - :set fileformat=mac (<CR>)

- * Eingebaute Hilfe
 - Hypertext
 - Weitreichend und umfassend
 - 50 ASCII Dateien
 - 25000 Zeilen Text
 - /usr/share/vim/current/doc

 - :help = HilfeEinstieg
 - :help TOPIC<Ctrl-d> = Passende Hilfethemen anzeigen
 - :help TOPIC = Hilfe anzeigen zu TOPIC
 - :he TOPIC = Hilfe anzeigen zu TOPIC
 - :q<CR> = Hilfe beenden
 - :help tutor = Hilfe zu Tutorprogramm
 - :help index = Hilfe-Index
 - :help user-manual = Hilfe User-Manual

 - Normal mode cmd (nichts) :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_
 - 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 (Ctrl-AltGr-9 auf deutscher Tastatur)
 - Ctrl-t / Ctrl-o = Vor/zurückspringen (to/old)

```

Ctrl-d + TAB      = Vervollständigen

* Datei wiederherstellen
vim -r            = Liste wiederherstellbarer Dateien anzeigen (recover)
vim -r FILE      = Datei FILE wiederherstellen (aus .FILE.swp)

* Statuszeile formatieren
:help statusline
:set statusline=FORMAT
:set statusline=%F%m%r%h%w\ [FMT=%{&ff}]\ [TYPE=%Y]\ [ASCII=%\%03.3b]\ [HEX=%\%02.2B]\ [POS=
%04l,%04v,%p%]\ [LEN=%L]
:set laststatus=2      = Statuszeile ständig anzeigen
:set laststatus=0     = Statuszeile mit Kommando-Zeile kombinieren

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

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

* Rechtschreibung prüfen (ftp://ftp.vim.org/pub/vim/runtime/spell = Download Wörterbücher)
:set spell          = Fehler markieren
:set spelllang=de  = en, en_us, us, American
:set spelllang=en,da,de,it = Mehrere Sprachen
:highlight SpellBad = Hervorhebung der Fehler
z=                 = Korrektur-Vorschläge zu Wort unter Cursor
:set spellsuggest=NN = Max. NN Korrekturen vorschlagen

* "Undo" und "Redo undo" beliebig oft
Vor Änderungen am Text braucht man keine Angst haben, alles zurücknehmber
:set undolevels=1000
u = Undo
Ctrl-R = Redo undo

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

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

* Wiederholen
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

* 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)
:set incsearch = Inkrementell suchen während tippen
:set hlsearch  = Alle Treffer markieren
:set ignorecase = GROSS/kleinschreibung ignorieren
:set smartcase = GROSS/kleinschreibung intelligent berücksichtigen
(\C=abschalten, \c=einschalten)

```

```
:set wrapscan      = Über Dateiende/anfang hinwegsuchen
:nohlsearch        = Such-Markierungen entfernen
:invhlsearch       = Such-Markierung an/ausschalten
```

* Historie

```
:history=500
Für die 5 Eingabearten vorhanden
Für ':'-Kommandos (ex)
Für Suche
Für Ausdrücke
Für Eingabezeilen (input()-Funktion)
Für Debug-Kommandos
Cursor-^v          = In Kommandoliste blättern
Cursor-<>          = Zeichenweise in altem Befehl bewegen
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 beliebigen Bewegungskommandos (h j k l w b e Ctrl-f Ctrl-b ...) erweitern
```

* Operationen auf markiertem Text (Visual)

```
$          = Überall bis Zeilenende markieren
o O        = Zur diagonal gegü. Ecke / Ecke in gleicher Zeile springen (other)
~ 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)
=          = Formatieren mit externem Programm (equalprg)
gg         = Formatieren gemäß Option 'textwidth'
:         = Ex-Kommando aufrufen (:'<','>')
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         = Löschen ("cross out")
X Y       = Löschen / Merken (ganze Zeile) ("cross out" yank)
D         = Löschen (Ctrl-V = Rechteck, sonst ganze Zeile)
Ctrl-]    = Tagsuche
```

* Visual Mode

```
: -> '<','>':EX-BEFEHL
! -> '<','>!OS-KOMMANDO
gv = Letzten Visual-Block erneut markieren (go visual)
```

* Objekte auswählen (z.B. Name)

Nach Visual-Kommando oder vor Operation c/d/y/... verwendbar
Wiederholungsfaktor davor angebbar

```
(i=inner, a=all/outer)
aw      = Wort + Space danach
iw      = Wort ohne Space aussenrum
aW      = WORT + Space danach
iW      = WORT ohne Space aussenrum
as      = Satz + Space danach (sentence)
is      = Satz ohne Space aussenrum
ap      = Absatz + Space danach (paragraph)
ip      = Satz ohne Space aussenrum
a( ab   = Text in (...) mit ()
i( ib   = Text in (...) ohne ()
a<      = Text in <...> mit <>
i<      = Text in <...> ohne <>
a[      = Text in [...] mit []
i[      = Text in [...] ohne []
a{ aB   = Text in {...} mit {}
i{ iB   = Text in {...} ohne {}
```

* Insert-Vervollständigung (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 Dateiname suchen (filename)
Ctrl-X Ctrl-L = Passende vollständige Zeile suchen (line)
:help ins-completion
```

* Options-Typen

```
Boolesche (z.B. compatible + nocompatible)
Numerische (z.B. shiftwidth=4)
String (z.B. shell=/bin/sh)
```

```

* Optionen Langform und 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
:set sidescroll=5
zl = Cursor nach links scrollen
zh = Cursor nach rechts scrollen
zs = Cursor zum linken Rand scrollen (start)
ze = Cursor zum rechten Rand scrollen (end)
gj = In umbrochener Zeile eine Zeile nach unten
gk = In umbrochener Zeile eine 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. (isfilename)  \P       = Ohne Ziffern
\s           = Leerraum (TAB, SPACE)  \S       = KEIN Leerraum
\b           = Backspace
\e           = Escape
\r           = Carriage Return
\t           = Tabulator
\n           = Reserved
\(...\)      = Gruppieren/Merken  \1 \2 ... \9 = Gemerktes Element
~           = Letztes Suchmuster

* (Re)Formatierung
ggBEWEGUNG   = Ausgewählten Bereich formatieren
gwBEWEGUNG   = Ausgewählten Bereich formatieren (Cursor bleibt stehen)
gq          = Zeile formatieren
gww         = Zeile formatieren (Cursor bleibt stehen)
VISUALgq    = Visuell markierten Bereich formatieren
VISUALgw    = Visuell markierten Bereich formatieren (Cursor bleibt stehen)

* Kurioses
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
:help 42    = The meaning of life (Douglas Adams)
Ctrl-V Return = Carriage Return eingeben = "^M"
Ctrl-V 123   = Zeichen mit ASCII-Code 123 eingeben = "{"
: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 und Zeilen zählen + akt. Position ausgeben
K auf Befehl = Manpage zu Befehl anzeigen

```

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

:argument ... = Dateien angeben
:all         = Ein Fenster pro angegebener Datei
:split       = Neues Fenster mit mir gleicher Datei öffnen
:vsplit     = Neues Fenster mit mir 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)

```

* Viminfo

```

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

```

* Dateinamen

```

%          = Akt. Dateiname (current)
#          = Vorheriger Dateiname (previous)
Ctrl-^    = Alternative Datei laden (Wechsel zw. % und #, deu. Tast. Strg-&)

```

* Source-Datei einlesen (include)

```

:source FILE = Vim-Befehle einlesen (Include)
:map <Ctrl-j> ...

```

* Externe Linux-Befehle (Filter) aufrufen (z.B. sort, uniq, fmt)

```

:!CMD      = Aufrufen und Ergebnis ansehen
!CMD       = Auf akt. Zeile anwenden
!!CMD      = Auf akt. Zeile anwenden
:%!sort    = Auf alle Zeilen anwenden
:r!ls      = Ergebnis nach akt. Zeile einfügen
:0r!ls     = Ergebnis am Dateianfang einfügen

```

* Autokommandos (Autocmd)

```

Ereignis + Dateinamen-Match -> Aktion / Keine Aktion
:autocmd EVENT FILEPAT COMMAND
:autocmd! EVENT FILEPAT COMMAND
EVENT
  FileReadPre  FileReadPost
  FileWritePre FileWritePost
  WinEnter     WinLeave
  ...
FILEPAT
  Analog Shell-Muster
CMD
  Ex-Kommandos (mehrere werden gesammelt)

```

```
Beispiel
:autocmd BufReadPre,FileReadPre *. [chy] set cindent
```

```
-----
# Vim-Programmierung
#-----
```

```
* Kommentar
" ...           Auf Zeile für sich
Anhängen an Befehle nur falls Kommentar nicht als Bestandteil erkennbar
set sw=1        "Kommentar erlaubt
map <F2> "ap    "Kein Kommentar erlaubt
```

```
* Tastaturcodes (für :map, Eingabe mit Ctrl-k + Taste)
TIPP: Rauskriegen im Ex/Such-Modus mit Ctrl-k + Taste
```

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

<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-Präfixe

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

* 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
:echon ...       = Analog ohne Newline
: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
:endifunction
```

Beispiel

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

* 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)
:endifor

```

```

#-----
# Programmieren mit Vim
#-----

```

* Suche nach "unmatched" Klammern (gemäß Verschachtelungshierarchie)

```

:set showmatch
:set matchpairs
%      = 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 öffnen mit 1. Vorkommen
Ctrl-w Ctrl-i = Neues Fenster öffnen 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 öffnen mit 1. Definition
Ctrl-w Ctrl-d = Definition Vorkommen ab Dateiposition

```

* Einrücken (Smart Indenting)

```

:set autoindent      = Analog Zeile darüber (privimiv)
:set smartindent    = Gemäß C (gut)
:set cindent         = Gemäß C (besser)
:set formatoptions=... = Formatierungsverhalten definieren
:set comments=...   = Aussehen von Kommentar definieren
Strg-D              = Automatische Einrückung zurücknehmen
TAB                 = Automatische Einrückung erweitern

```

* Beispiele

```

Wortsuche
Wort ersetzen
Funktionstastenbelegung
NEW-Makro
Return -> :nohlsearch
Kommentare Ein/Ausrücken per Visual-Makro
Funktionstastenbelegung (Tristate)
vimtutor
vimtutor de

```