

- h moves the cursor one character to the left.w
- k moves the cursor up one line.
- j moves the cursor down one line.
- l moves the cursor one character to the right.
- 0 moves the cursor to the beginning of the line.
- ^ move to first non-block character in line.
- \$ moves the cursor to the end of the line.
- w move forward one word.
- b move backward one word.
- G move to the end of the file.
- gg move to the beginning of the file.
- line + G or gg moves curse to line.
- '. move to the last edit.
- A insert at the end of the line
- e Puts the cursor at the end of a word
- ctrl + u fast upwards
- ctrl + d fast downwards
- ctrl + e scroll down without cursor
- ctrl + p in commandmode will go history upwards
- ctrl + n " " down
- in NERDTree use s to open file horiyontaly and i vertically
- n move the cursor to the next instance of the text from the last search. this will wrap to the beginning of the document.
- n move the cursor to the previous instance of the text from the last search.

- d starts the delete operation.
 - x deletes current char
 - dd will delete the whole line.
 - dw will delete a word.
 - d0 will delete to the beginning of a line.
 - d\$ will delete to the end of a line.
 - dgg will delete to the beginning of the file.
 - dG will delete to the end of the file.
 - u will undo the last operation.
 - Ctrl-r will redo the last undo.
-
- gt - move to the next tab
 - gT - move to the previous tab
 - #gt - move to a specific tab number (e.g. 2gt takes you to the second tab)
 - =G - indents code below
 - CTRL-W h - moves in window above, j k l h for other windows
 - CTRL-] - in helpfile on makfiles will open them
 - |space; j or k - opens new buffer in direction of key
-
- - /*text* search for *text* in the document, going forward.
 - ?*text* search for *text* in the document, going backwards.
 - :%s/*text*/replacement text*/g search through the entire document for *text* and puts *replacement text* in front*. without text/g it replace
 - :%s/text/replacement/g replace all text

- `:%s/*text*/replacement text*/gc` search through the entire document and `*confirm*` before replacing text.
- `Crtl + V` after selection `Shift + I` insert what you want for these line
- `/+text` to be searched — with `n` move forward trough words — `N` backwards
- `shift + i + text` replaces what was marked in visual mode with text
- `:nohl` to stop marking text
- `z=` on misspelt word for suggestens
- `zg` add misspelt word to dictionary
- `zw` say word i wrong
- `s/textreplace/textsubtution [g&]`(<https://www.notion.so/g-4a820f8ef73f420580ed02d2b726782b>) to replace text for one line and then all
- `noremap` (`inoremap`, `nnoremap`, `vnoremap`) + `"keytopress"` + `"what i does"` ignores ever other map
- `autocmd` "looks always what you write
- `Filetype + "filetype"` in what filetype the folowing wil happen
- `:! + command` "makes commmand in terminal bvut beeing in vim
- `gt` switches tabs
- `:setlocal spell spelllang=de` sets spell checking in german local buffer with just `:set` it is global
- `⌘F5` turns off spell checking
- `:"lineNumberStart", "lineNumberEnd"s/^"replaceItem"` / puts on beggining of the lines between `lineNUmberStart` and `Linenumberend` the `replaceItem`, works with `crl-V` to
- `q:` shows `commandhistory` that is executable
- `⌘$` represents brakes
- `K` show documentation for thing on cursor, the commant in lua doesnt work yet

- v highlight one character at a time.
 - V highlight one line at a time.
 - V_j to move selected lines to left.
 - V_l to move selected lines to right.
 - Ctrl-V highlight by columns.
 - # in normal mode highlights all the strings i am hovering in the file
 - p paste text on the current line.
 - P paste before the cursor.
 - y yank text into the copy buffer.
 - y\$ yank till the end of the line.
 - 'y\$' - Yank (copy) everything from the cursor to the end of the line.
 - 'y[^]' - Yank (copy) everything from the cursor to the start of the line.
 - "+y copys selected data into clipboard
-
- :w save the file
 - :e filename - opens filename and closes open file
 - :tabedit file - opens a new tab and will take you to edit "file"
 - :tabe "filename" opens file in same terminal
 - :tabm i move tab to i beginning with 0
 - :tabm +i/-i move tab i related to position of tab
 - :tabs - list all open tabs
 - :tabclose - close a single tab

- :w filename safe with new name
 - :q finish vim if no safe u have to ! for safe
 - :q! quit without safe
 - :split "file" splits window in half on top will be file
 - :vsplit "file" " " left will be file
 - **pluginmanager**
 - :Telescope notify opens history off notifications
 - :Dbee opens tabs for database access, the connection Angaben are in lua/database.lua
 - in Dbee, select query, BB executes the query
 -]s or [s go to prev/next spellerror
 - zg adds wrong word to custom list of words
 -]d/[d go to next/prev error
-
- ZZ safe and quit
 - ctrl + x + f filenamecompleten in vim
 - ;it textit
 - ;ib textbf
 - ;ig include graphics
 - ;sec ;sec* ;sub ;sub*
 - ;im itemize
 - ;cen center
 - ;tb table with [H]
 - ;bg begin something

tabby movment

- "leader"tn next tab
- "leader"tp previous tab
- "leader"to closes all other tabs
- C-t opens new tab
- C-s closes tab
- "leader"tmp moves tab to previous position
- "leader"tmn moves tab to next position
- :help window-moving shows comands for moving windows

Custom/Plugs

- cdacac

lsp, codehelp

- K shows manuel entry of hovered item
- jspacejrn renames variable everywhere
- gr shows rwhere variable/function is used/defined
- jLeader>e or d shows error on line
- :Telescope notify shows history of notifications
- "leader"v finding file with telescope
- "leader"s source file

- gr hover over word in java, shows you evry usage
- gd " ", go to defenition
- :LSPinfo gives info about running language server