

# Vim

## Verbs

- **d** delete (keep visual mode)
- **c** change (move to in insert mode)
- **>** ident
- **v** visally select
- **control + v** vertical visal selection
- **y** yank

## Noun

- **b** back
- **i** inner
- **2j** two lines down
- **"** quotes
- **p** paragraph
- **s** sentence
- ex.: **dip** delete inner paragraph, inner make vim remembe the object type
- **w** move forward
- **b** move backward
- **e** move end of word
- **f** find a character forward
- **F** find a character backward
- **t** find a character forward and stop before
- **T** find a character backward and stop before
- **;** next searched item
- **\$** move to the end of a line
- **0** move to the beginning of a line
- **^** fist non-blank character of line
- **{}** move between paragraph
- **()** move between sentence
- **g;** last edited place
- **gg** beginning of file
- **G** end of file
- **H** jump to top of screen
- **M** jump to middle of screen
- **L** jump to middle of screen
- **zz** posiciona o cursor no meio da tela
- **z.** posiciona o cursor no meio da tela
- **zt** posiciona o cursor no início da tela (top)
- **zb** posiciona o cursor no fim da tela (bottom)
- **control d** down half screen
- **control u** up half screen
- **control f** down all screen
- **control b** down all screen

## Registers

- **@** ou **"** in normal mode
- **control r** in insert mode

## Macros

- **q[letter]** enter recording mode
- **q** finished the macro
- **@[letter]** is the address the recorded macro
- **10@w** execute the macro 10 times

## Windows

- **ctrl-w +** mair verticalmente
- **ctrl-w -** menor verticalmente
- **ctrl-w <** mais a esquerda
- **ctrl-w >** mais a direita
- **ctrl-w ctrl-x** inverter as janelas entre direita e esquerda, cima e baixo
- **ctrl-w ctrl-r** inverter as janelas entre direita e esquerda, cima e baixo
- **ctrl-w k** posiciona as janelas horizontalmente divididas
- **ctrl-w h** posicionada as janelas ver verticalmente

## Fuzzy Finder fzf

- **;** file here
- **;**h file home
- **;**w file ~/Wiki
- **;**s file ~/Sync
- **;**b buffers
- **;**l lines
- **;**L LinesB
- **control x** resultado em buffer horizontal
- **control v** resultado em buffer vertical
- **control t** resultado em tab

## Search

- **/** search top do down
- **?** search down do top
- **:vimgrep palavra-buscada**
- **:vimgrep palavra\_buscada \*.qmd**
- **:vimgrep /palavra\_buscada/g \*.qmd** adiciona resultados para quickfix
- **:vimgrep palavra\_buscada %** busca no arquivo atual
- **:vimgrep /foo/ foo.txt bar.txt gar.txt** buscar em vários arquivos
- **:vimgrep /palavra\_buscada/j \*.Rmd** don't jump to the first match
- **:vimgrep /palavra\_buscada/ 'find . -type f'**
- **:control r** insere a última query buscada

## Others

- **:Tabularize /**, arquivos separados por vírgula
- **:Tabularize \t** arquivos separados por tab
- **:Goyo** no distraction

# Plugins

## jalvesaq/Nvim-R

- Start/Close
  - \rf Start R (default)
  - \rq Close R (no save)
  - :RStop Stop (interrupt) R
- Send
  - \aa File
  - \bb Block (cur)
  - \bd Block (cur, down)
  - \cc Chunk (cur)
  - \cd Chunk (cur, down)
  - \ch Chunk (from first to here)
  - \ff Function (cur)
  - \fd Function (cur and down)
  - \ss Selection
  - \sd Selection (and down)
  - \m{motion} Send motion region
  - \pp Paragraph
  - \pd Paragraph (and down)
  - \l Line
  - \d Line (and down)
  - \q Line (and new one)
  - \r<Left> Left part of line (cur)
  - \r<Right> Right part of line (cur)
  - \o Line (evaluate and insert the output as comment)
  - \su All lines above the current one
- Command
  - \rh Help
  - \rl List space
  - \rr Clear console
  - \rm Remove objects and clear console
  - \rp Print (cur)
  - \rn Names (cur)
  - \rt Structure (cur)
  - \qr Quarto render (cur file)
  - \qp Quarto preview (cur file)
  - \qs Quarto stop preview (all files)
- Edit
  - \gn Go (next R chunk)
  - \gN Go (previous R chunk)
- Object Browser
  - \ro Open/Close
  - \r= Expand (all lists)
  - \r- Collapse (all lists)
  - Enter Toggle (cur)

## vimcmdline: Python and Julia

- \s to start the interpreter.
- space to send the current line to the interpreter.
- \space to send the current line to the interpreter and keep the cursor.
- \q to send the quit command to the interpreter.
- space to send a selection of text to the interpreter.
- \p to send from the line to the end of paragraph.
- \b to send block of code between the two closest marks.
- \f to send the entire file to the interpreter.
- \m to send the text in the following motion to the interpreter. For example
- \miw would send the selected word.

## tpope/vim-commentary

- gc{motion} Comment or uncomment lines that {motion} moves over.
- ex:
  - gc4j comentar da 4 linhas abaixo
  - gcip comentar o parágrafo
  - :4,8Commentary comentar das linhas 4 a 8

## tpope/vim-surround

- c chance
- d delete
- y you surround

"Hello *world!"	ds"	Hello world!
(123+4*56)/2	ds)	123+456/2
<div>Yo!*</div>	dst	Yo!
"Hello *world!"	cs"	'Hello world!'
"Hello *world!"	cs"<q>	<q>Hello world!</q>
(123+4*56)/2	cs)]	[123+456]/2
(123+4*56)/2	cs)[	[ 123+456 ]/2
<div>Yo!*</div>	cst<p>	<p>Yo!</p>
Hello w*orld!	ysiw)	Hello (world)!
Hello w*orld!	yssB	{Hello world!}

Obs.: \* é a posição do cursor na sentença

## tpope/vim-unimpaired quickfix

- ]q is :cnext
- [q is :cprevious
- ]a is :next
- [b is :bprevious

## neoclide/coc.nvim

- ;y yank
- ;x explorer
- ;a Show all diagnostics
- ;e Manage extensions.
- ;c Show commands.
- ;p Resume latest coc list.

# Git

- `stsewd/fzf-checkout.vim`
  - `gB` branches
    - \* `enter` checkout
    - \* `alt-enter` tack
    - \* `control b` create
    - \* `control d` delete
    - \* `control e` merge
    - \* `control r` rebase
  - `gC` commit history
  - `gT` tags
- `tpope/vim-fugitive`
  - `:Git <commando>`
  - `:G <commando>`
  - ex.: `:G init | :G add . | :G commit | :G remove --cached`
- `:G <enter>` Stage alternativas
  - `s` Stage (add) the file or hunk under the cursor.
  - `u` Unstage (reset) the file or hunk under the cursor.
  - `-` Stage or unstage the file or hunk under the cursor.
  - `U` Unstage everything.
  - `X` Discard the change under the cursor. This uses `checkout` or `clean`.
  - `=` Toggle an inline diff of the file under the cursor.
  - `>` Insert an inline diff of the file under the cursor.
  - `<` Remove the inline diff of the file under the cursor.
  - `gI` Open `.git/info/exclude` in a split and add the file under the cursor. Use a count to open `.gitignore`.
- `:G <enter>` Commit maps
  - `cc` Create a commit.
  - `ca` Amend the last commit and edit the message.
  - `ce` Amend the last commit without editing the message.
  - `cw` Reword the last commit.
  - `cvc` Create a commit with `-v`.
  - `cva` Amend the last commit with `-v`
  - `c?` Show this help.
- `:G <enter>` Diff maps
  - `dd` Perform a `|:Gdiffsplit|` on the file under the cursor.
  - `dv` Perform a `|:Gvdiffsplit|` on the file under the cursor.
  - `ds` Perform a `|:Ghdiffsplit|` on the file under the cursor.
  - `dh`
  - `dq` Close all but one diff buffer, and `|:diffoff|!` the last one.
  - `d?` Show this help
- Comandos especiais:
  - `:Gwrite`
  - `:Gread`
  - `:GRemove`
  - `:GDelete`
  - `:GMove`
  - `:GRename`
  - `:GBrowse`
  - `:Gclog`
  - `:Gdiffsplit`
  - `:Ghdiffsplit`
  - `:Gvdiffsplit`
  - `:Gdiffsplit`
  - `:Git blame`
  - `:Git merge`
  - `:Git_mergetool`

# i3wm

## Navegation

- **super** **u** focus parent
- **super** **d** focus child
- **super** **f** fullscreen toggle

## Change container layout (stacked, tabbed, default)

- **super** **w** tabbed
- **super** **e** stacking
- **super** **r** toggle split

## SetNext container as: split[h|v], stab, stack

- **super** **-** horizontal
- **super** **\** vertical
- **super** **shift t** tabbed
- **super** **shift s** stacking

## Float

- **super** **m** floating toggle
- **super** **n** focus mode\_\_toggle

## Action

- **alt** **[h,j,l,l]** move focused window
- **shift alt** **[0-9]** move focused window
- **super control** **[l,h]** the whole workspace to the next output
- **super** **;** **r** (resize) “Normal:Resize: [h]kl[HJKL]”
- **super** **;** **m** (move) “Normal:Resize: [h]kl[HJKL]”

# Fuzzy Finder in Terminal

- **zahtura** **\*\*<TAB>**
- **nvim** **\*\*<TAB>**
- **nvim** **~/Downloads\*\*<TAB>** ou
- **nvim control t**
- **control r** history
- **cd** **\*\*<TAB>** change directory
- **alt c** change directory
- **z** **<dir-name><TAB>\*\*<TAB>** z plugin + fzf to change directory
- Options:
  - **ctrl-b** backward-kill-word
  - **ctrl-g** top
  - **ctrl-u** page-up
  - **ctrl-d** page-down
  - **ctrl-f** toggle-all
  - **ctrl-q** clear-query
  - **ctrl-s** clear-selection
  - **ctrl-o** jump

- **ctrl-c** cancel
- **ctrl-y** yank
- **ctrl-p** toggle-preview
- **ctrl-t** toggle-all
- **ctrl-z** deselect-all
- **tab:toggle** select and down
- **shift-tab** select and up

# Vimium

- Commands:
  - **gg** scroll to top of the page
  - **G** scroll to bottom of the page
  - **d** scroll down half a page
  - **u** scroll up half a page
  - **f** open a link in the current tab
  - **F** open a link in a new tab
  - **r** reload
  - **gs** view source
  - **i** enter insert mode
  - **yy** copy the current url to the clipboard
  - **yf** copy a link url to the clipboard
  - **o** Open URL, bookmark, or history entry
  - **O** Open URL, bookmark, history entry in a new tab
  - **/** enter find mode
  - **n** cycle forward to the next find match
  - **N** cycle backward to the previous find match
  - **H** go back in history
  - **L** go forward in history
- Manipulating tabs:
  - **J**, **gT** go one tab left
  - **K**, **gt** go one tab right
  - **g0** go to the first tab. Use **ng0** to go to n-th tab
  - **g\$** go to the last tab
  - **^** visit the previously-visited tab
  - **t** create tab
  - **yt** duplicate current tab
  - **x** close current tab
  - **X** restore closed tab (i.e. unwind the ‘x’ command)
  - **T** search through your open tabs
  - **W** move current tab to new window