VIM

Vimrc

* `<Leader>ev` - edit .vimrc
* `ZZ or <Leader>wq` - save .vimrc and close vertical split
* `<Leader>sv` - source .vimrc

Vim Plugins Installed

* [Pathogen instructions](https://gist.github.com/romainl/9970697) and [Tim's repo](https://
* [vimwiki](https://vimwiki.github.io/)
* [commentary](https://vimawesome.com/plugin/commentary-vim)
* [markdown syntax](https://vimawesome.com/plugin/markdown-syntax)
* [NERDTree]()
* [vim-Flake8]()
* [vim-latex]()
* [tagbar](https://vimawesome.com/plugin/tagbar)
* [csv-vim](https://vimawesome.com/plugin/csv-vim)
* [vimux](https://github.com/benmills/vimux/blob/master/README.mkd) with bindings
* [solarized](https://github.com/vim-scripts/Solarized) see note for Term users
* [vim airline](https://github.com/vim-airline/vim-airline) with [powerline fonts](), https:
* [vim airline themes](vimairline-theme://github.com/vim-airline/vim-airline-themes) there i
* [tmuxline](https://github.com/edkolev/tmuxline.vim/blob/master/README.md) vim colors for t
* [fugitive](https://vimawesome.com/plugin/fugitive-vim) - git plugin
* [vim pandoc markdown prepandoc view](https://github.com/conornewton/vim-pandoc-markdown-pr

Plugins to consider

* [vimproc](https://vimawesome.com/plugin/vimproc-vim)
* [surround](https://vimawesome.com/plugin/surround-vim)
* [tbone](https://vimawesome.com/plugin/tbone-vim)
* vimux (<--already installed) with [pyUtils](https://github.com/julienr/vimux-pyutils)
* alternatively there is [vim ipython](https://github.com/ivanov/vim-ipython/)
* [vim markdown preview](https://github.com/JamshedVesuna/vim-markdown-preview) lets you liv
* [or this one](https://github.com/iamcco/markdown-preview.vim)

1

Vim Tricks

Count every stroke http://www.vimgolf.com

bing your arrows to hjkl https://askubuntu.com/questions/684459/configure-caps-lock-as-altgr-and-arrows-like-in-vim/898462#898462

test your setup with vanilla vim `-u NONE` disables .vimrc `-N` enabled no-compatible mode `vim -u NONE -N`

source a different .vimrc file `vim -u .custom_vimrc`

get shell access `:sh`

Search forward to a char (ex: `+`), substitute it with space+space `f+ s<Space>+<Space><Esc>` Then simply `;.`

Switch case to lower

whatever you select in visual mode you can convert with u or U `vu` or upper `vU` Toggle cases wihtout selecting ~ https://vim.fandom.com/wiki/Switching_case_of_characters

`:[range]w[rite] [++opt] !{cmd}` Range is either `%` which means entire file or `'<,'>` which means visual selection Send selection by email `:%w !mutt -s "here are my notes" -- joe@example.com`

Insert Mode - Delete del back one char `<C-h>` del back one word `<C-w>` del back to statt of line `<C-u>`

Insert Notmal Mode from insert mode invoke normal mode for a one-off command `<C-o>{cmd}`

Center Screen (vertivally) with current line in the middle zz

Spell Check changes are written to `.vim/spell/<lang-file>`

Folding https://vim.fandom.com/wiki/Folding

Splits Maximize current split - Width wise `<Ctrl-w>|` Maximize current split - Height wise `<Ctrl-w>_` Swap splits `<Ctrl-w>R` Resize all windows to equal dimensions based on their splits `<Ctrl-w>=` https://vim.fandom.com/wiki/Resize_splits_more_quickly Help `:help splits`

Buffers `:bn` next `:bp` previous `:b` list `bd #` delete buffer, where # is the number of the buffer (no # means current) `:vert sb #` open a # buffer in a vertical split https://vim.fandom.com/wiki/Buffers

Special Chars display the char's (under cursor) ascii code do `ga` To enter the char from Insert mode do the hex code `<C-v>u{code}` Or `<C-v>{code}` for _____ code the other combination was.. `<C-k>{code}`

Moving `<S-{>` and `<S-}>` moves you to the next paragraph `<S-(>` and `<S-)>` moves you to the next sentence `<[>` and `<]>` moves you to the next header *does it really work (?)*

Upper and Lower Case

`gU` - upper case `gu`- lower case `gUU` - on the whole line `gUap` - around paragraph `gR` - virtual replace mode that treats tab as multiple spaces `gv` - select again the last visual selection

`o` - go to the other end of visual selection. E.g. toggle the free end

Tables

Create horizontal line to separate table header `yyp` - copy and paste first row of the table `Vr-` - visually select this second line and replace each char with -

Reduce space between table columns `<C -v>jjj x...` - del one column of chars at a time

Add vertical line to separate columns `gvr|` - select the same range again and replace with |

Append after visual block

You can select multiple lines in visual block mode, from cursor to the end `<C-v>jj$ A;` - Append at the end of line (selection?) `<Esc>`