

Vim

Devpoint Labs - Dave Jungst / Jake Sorce

Command Line Editors

Vim is not the only command line editor other popular choices are:

- nano
- emacs
- vi
- Pico
- NeoVim

Vim (Vi iMproved) comes installed on your mac however it is an older version

Letting VIM replace your IDE

- Tab completion
- Syntax Highlighting
- Method / Class Stubbing
- Jump to declaration
- File tree explorer
- And More...

If your IDE can do it VIM can do it and likely in a more customizable way

Pathogen

Pathogen is one of many vim plugin managers that allows you to add plugins to your vim environment.

It your terminal:

```
mkdir -p ~/.vim/autoload ~/.vim/bundle && \
```

```
curl -LSso ~/.vim/autoload/pathogen.vim https://tpo.pe/pathogen.vim
```

vimrc

A vimrc is a config file for your vim environment. First lets edit the vimrc using VIM!

```
vim ~/.vimrc
```

```
execute pathogen#infect()
```

```
syntax on
```

```
filetype plugin indent on
```

type esc :wq to write and quit

vimrc (cont)

#show numbers by default

set nu

#set your vim colorscheme (Need to install colorschemes)

colorscheme desert

#ide indentation

set autoindent

set smartindent

set smarttab

set shiftwidth=2

set tabstop=2

set softtabstop=2

set expandtab

set hidden

set showmatch

filetype plugin on

filetype indent on

match Todo /\s\+\$/

autocmd BufWritePre * :%s/\s\+\$/e

Plugins

Now that pathogen is installed you can clone vim plugins from github to ~/.vim/bundle and they will automatically work

Command-T	tlib_vim	vim-jsx
angular-vim-snippets	vim-addon-mw-utils	vim-less
ctrlp.vim	vim-angular	vim-rails
ember.vim	vim-cjsx	vim-rake
html5.vim	vim-coffee-script	vim-ruby
javascript-libraries-syntax.vim	vim-colors-solarized	vim-snipmate
neocomplcache.vim	vim-endwise	vim-snippets
nerdtree	vim-fugitive	vim-vinegar
syntastic	vim-haml	vim-wakatime
taglist.vim	vim-javascript	

My advice would be to install a minimal amount of plugins and learn how to use them before installing more

Useful IDE plugins

vim-rails & vim-ruby

git clone <https://github.com/tpope/vim-rails.git>

git clone <https://github.com/vim-ruby/vim-ruby.git>

With vim Rails there are a lot of cool useful shortcuts. Let's say you have a model view and corresponding controller and you are editing the controller you can type

:Emodel and you will be taken to the model

:Eview home will take you to the controllers home view

:Emigration will take you to the migration

NerdTree

git clone <https://github.com/scroolose/nerdtree.git>

vim ~/.vimrc and add

```
map <C-n> :NERDTreeToggle<CR>
```

esc, :wq

Now whenever you are in vim you can type ctrl+n to show the project tree using regular vim movement to get around.

SnipMate

git clone https://github.com/tomtom/tlib_vim.git

git clone <https://github.com/garbas/vim-snipmate.git>

git clone <https://github.com/MarcWeber/vim-addon-mw-utils.git>

git clone https://github.com/honza/vim-snippets.git

When using snipmates typing the beginning of a block then hitting tab will complete the block

This works for

def, if, else, each, map, collect, select, scan, class

There is a snippets file in bundle where you can insert snippets for other languages

Moving Up, Down, Left, Right

The best way to get good at vim is to take away the arrow keys and practice moving around without using them. This is painful at first but once muscle memory kicks in you will likely stop using the arrow keys.

```
vim ~/.vimrc
```

```
noremap <Up> <NOP>
```

```
noremap <Down> <NOP>
```

```
noremap <Left> <NOP>
```

```
noremap <Right> <NOP>
```

Now you can only move around using h,j,k,l (left, down, up, right)

There are other movement keys but it is important to master these first

Insert / Append / Open

To move around in VIM we want to be in command mode (easily accessed by typing esc)

To enter insert mode type **i** at this point you will need to use arrows to move around.

You can also enter insert mode in other ways for example **a** will move to the next character over and put you in insert mode.

o will put you in insert mode 1 line below while **O** will put you in insert mode 1 line above.

Deleting

With your cursor over a letter type **x** to delete a character and stay in command mode.

To delete a line type **dd**

To delete a word type **dw**

Moving through lines and words

w will move you forward 1 word

b will move you backward 1 word

^ will move you to the start of the line

\$ will move you to the end of the line

:5 go to line 5

:/word find word typing **n** will go to next occurrence while typing **N** will go to previous occurrence

gg top of file **G** bottom of file

Operator, Number, Motion

In command mode all vim commands work with An operator a number and a motion. For example to delete 2 words to the right

`:d2w`

To delete 10 lines `10dd`

To move right 8 words `8w`

As you get better at vim you will start to combine operators, numbers and motion

Find and replace

`:s/find/replace/` find and replace on single line first occurrence

`:%s/find/replace/` find and replace file wide first occurrence

`:%s/find/replace/g` find and replace file wide all occurrences

`:4,12/s/find/replace/g` find and replace on lines 4-12 all occurrences

Visual mode

We know command mode and insert mode but there is also visual mode:

ctrl+v to enter visual mode now we can highlight in columns

DEMO changing a block of code with visual mode

Exiting vim

Ways to exit vim

`:wq` (write quit)

`:q` (fails if file has been changed)

`:q!` (quit without saving)

`:x`

Two more very useful command `u` for undo `ctrl+r` for redo

Tips

Keep a vim cheatsheet or 2 handy:

<http://www.fprintf.net/vimCheatSheet.html>

Learn commands slowly only a few at a time.

If you seem to be doing the same thing over and over there is probably a vim way of doing it.

Pick a vim command or 2 and practice it for the entire day until it is muscle memory.

Go through vimtutor