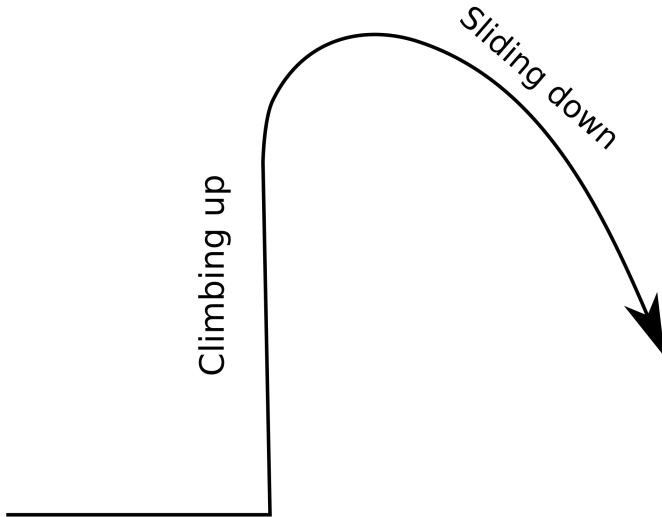


# Why You Should Give Vim a Try

Mauri Mustonen (Kazhuu)

October 3, 2019  
TampereJS

# Vim Learning Wall



# Where To Find Me

- ▶ Web page: <https://mauri.codes>
- ▶ GitHub: Kazhuu
- ▶ LinkedIn: Mauri Mustonen

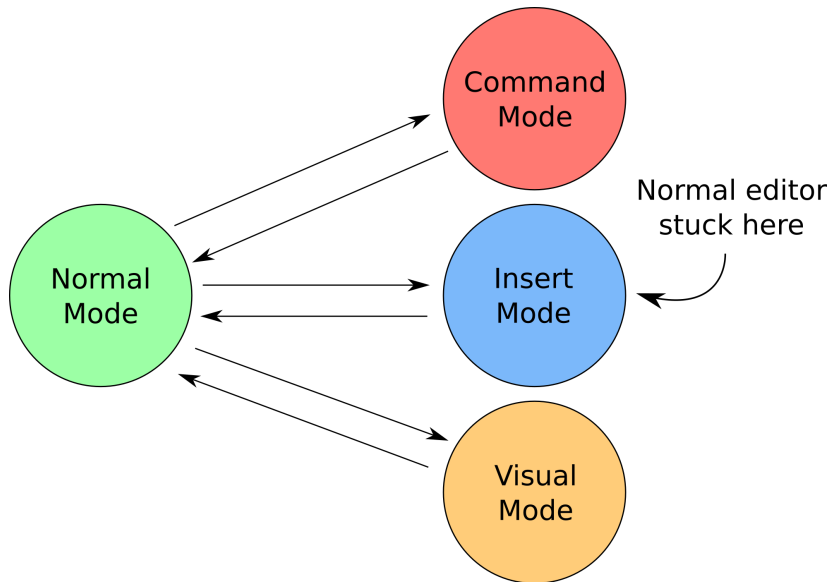


# Why To Learn Vim

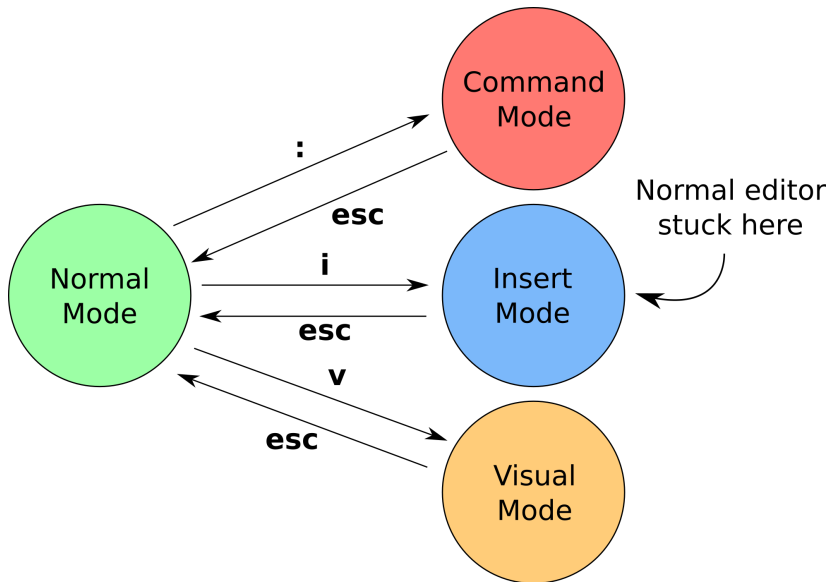


- ▶ Get rid of your mouse
- ▶ Less time on unnecessary hand movements
- ▶ Much more efficient code editing
- ▶ Highly customizable and extensible
- ▶ Learn new things for many years to come
- ▶ It will stick with you
- ▶ Become one of **those** people

# Vim Is All About Modes



# Vim Is All About Modes



# Basic Movement

- **h, j, k, l**      ▶ left, down, up, right
- **w**              ▶ move **f**orward one word
- **e**                ▶ move forward to **e**nd of the word
- **b**                ▶ move **b**ackward one word
- **gg**              ▶ move to the top of the file
- **G**                ▶ move to the bottom of the file
- **O**                ▶ move to the beginning of the line
- **\$**                ▶ move to the end of the line

And more...

# Basic Editing

- **i** ▶ **i**nsert
- **d** ▶ **d**elete
- **c** ▶ **c**hange
- **y** ▶ copy (**y**ank)
- **p** ▶ **p**aste
- **u** ▶ **u**ndo
- **ctrl+r** ▶ redo
- **.** ▶ repeat last command

And more...



# Composing Commands

<number><command><text object/motion>

- **dw** ▶ **d**delete next **w**word
- **yw** ▶ copy/**y**ank **w**word
- **2dw** ▶ **d**delete two following **w**words
- **5j** ▶ move down five lines
- **dd** ▶ delete line
- **50G** ▶ go to line 50

# Command Language

How to delete all function parameters?

```
function randomFunction(param1, param2, param3) {  
    ...  
}
```

# Command Language

How to delete all function parameters?

```
function randomFunction(param1, param2, param3) {  
    ...  
}
```

ci)

change **i**nside **p**arentheses

# Command Language

How to change string inside quotes?

```
function hello() {  
    return 'Hello World!';  
}
```

# Command Language

How to change string inside quotes?

```
function hello() {  
  return 'Hello World!';  
}
```

ci'  
change inside quotes

# Command Language

How to rewrite the whole function body?

```
function getMax(a, b) {  
  if (a < b) {  
    return b;  
  }  
  return a;  
}
```

# Command Language

How to rewrite the whole function body?

```
function getMax(a, b) {  
  if (a < b) {  
    return b;  
  }  
  return a;  
}
```

ci}  
change **i**nside **b**races

# Configuring .vimrc

```
1: home 2 3
vimrc (-) - VIM
14 set hidden          " allow switch between buffers if not saved
13 set nospell         " set no spell checking
12 set backspace=indent,eol,start
11 set textwidth=0     " set no text width
10 set scrolloff=9999  " keep cursor on the middle all the time
9 set relativenumber   " Put on relative numbers.
8 syntax on           " syntax always on
7
6 " Vim swap and backup files to home different directory instead
5 set backupdir=~/.vimtmp,.
4 set directory=~/.vimtmp,.
3
2 " Use system clipboard instead of + and * registers
1 set clipboard=unnamedplus

73
1 " Remap <leader> key. Defaults to backslash '\'
2 let mapleader=","
3
4 " Coloschema settings
5 colors zenburn
6
7 " Default indentation config
8 set tabstop=4
9 set shiftwidth=4
10 set noexpandtab
11
12 " Enable folding
13 set foldmethod=indent
14 set foldlevel=99
NORMAL .vimrc
vim utf-8[unix] 32% 73/224 1 : 1
```



## Some Useful Plugins

- ▶ NERDTree
- ▶ Command-line fuzzy finder (fzf)
- ▶ Silver Searcher (ag)
- ▶ YouCompleteMe

# Useful Tips

- ▶ Escape to Caps Lock
- ▶ Tabs as **views**
- ▶ Registers as multiple clipboards
- ▶ Record macros
- ▶ Repeat with .
- ▶ Own cheat sheet
- ▶ Get help with :help

## Lesson 1.1: MOVING THE CURSOR

**\*\* To move the cursor, press the h,j,k,l keys as indicated. \*\***

^  
k  
< h      l >  
j  
v

Hint: The h key is at the left and moves left.  
The l key is at the right and moves right.  
The j key looks like a down arrow.

1. Move the cursor around the screen until you are comfortable.
2. Hold down the down key (j) until it repeats.  
Now you know how to move to the next lesson.
3. Using the down key, move to Lesson 1.2.

NOTE: If you are ever unsure about something you typed, press <ESC> to place you in Normal mode. Then retype the command you wanted.

NOTE: The cursor keys should also work. But using hjkl you will be able to move around much faster, once you get used to it. Really!



# vi / vim graphical cheat sheet

**Esc**  
normal  
mode

␣ not used!	! external filter	"*, reg. spec <sup>1</sup>	£ not used!	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	( begin sentence	) end sentence	"soft" bol	+ next line
., goto mark	1 <sup>2</sup>	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto <sup>3</sup> format
Q ex mode	W next word	E end word	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.
q record macro	w next word	e end word	r, replace char	t, 'till	y yank	u undo	i insert mode	o open below	p paste after	[ misc	]	misc
A append at eol	S subst line	D delete to eol	F, "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	@ play macro	~ toggle case	
a append	s subst char	d delete	f, find char	g extra cmds <sup>6</sup>	h ←	j ↓	k ↑	l →	: repeat t/T/TF	! goto mk. bol	# prev ident	
bol/ goto col	Z, quit <sup>4</sup>	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen midl	< un-indent <sup>3</sup>	> indent <sup>3</sup>	? find (rev.)		
\, not used!	Z, extra cmds <sup>5</sup>	X delete char	c change	V visual mode	b prev word	n next (find)	m, set mark	, reverse t/TF	. repeat cmd	/ find		

<b>motion</b>	moves the cursor, or defines the range for an operator
<b>command</b>	direct action command, if <b>red</b> , it enters insert mode
<b>operator</b>	requires a motion afterwards, operates between cursor & destination
<b>extra</b>	special functions, requires extra input
q.	commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line,  
mk = mark, yank = copy

words: **quux(foo, bar, baz)**  
WORDS: **quux(foo, bar, baz)**

Main command line commands ('ex'):  
:w (save), :q (quit), :q! (quit w/o saving)  
:e f (open file f),  
:%s/x/y/g (replace 'x' by 'y' filewide),  
:h (help in vim), :new (new file in vim),

Other important commands:  
CTRL-R: redo (vim),  
CTRL-F/B: page up/down,  
CTRL-E/-Y: scroll line up/down,  
CTRL-V: block-visual mode (vim only)

Visual mode:  
Move around and type operator to act on selected region (vim only)

## Notes:

- use "x before a yank/paste/del command to use that register ('clipboard') (x=a,z,\*) (e.g.: "ay\$ to copy rest of line to reg 'a')
- type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5l, d4j)
- duplicate operator to act on current line (dd = delete line, >> = indent line)
- ZZ to save & quit, ZQ to quit w/o saving
- zt: scroll cursor to top, zb: bottom, zz: center
- gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to [www.viemu.com](http://www.viemu.com) - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

# Vim Elsewhere



- ▶ Chrome with Vimium
- ▶ Linux i3 window manager
- ▶ Vim on Windows Subsystem for Linux (WSL)

# Summary

- ▶ Vim is hard because it's different
- ▶ Start with basics
- ▶ Make your own .vimrc and understand it
- ▶ Read other people's .vimrc files
- ▶ Make your own cheat sheet
- ▶ Take time and learn
- ▶ You will become much more efficient editing code
- ▶ Highly customizable and extensible

# Thank you for your time!

Questions?

<https://mauri.codes>