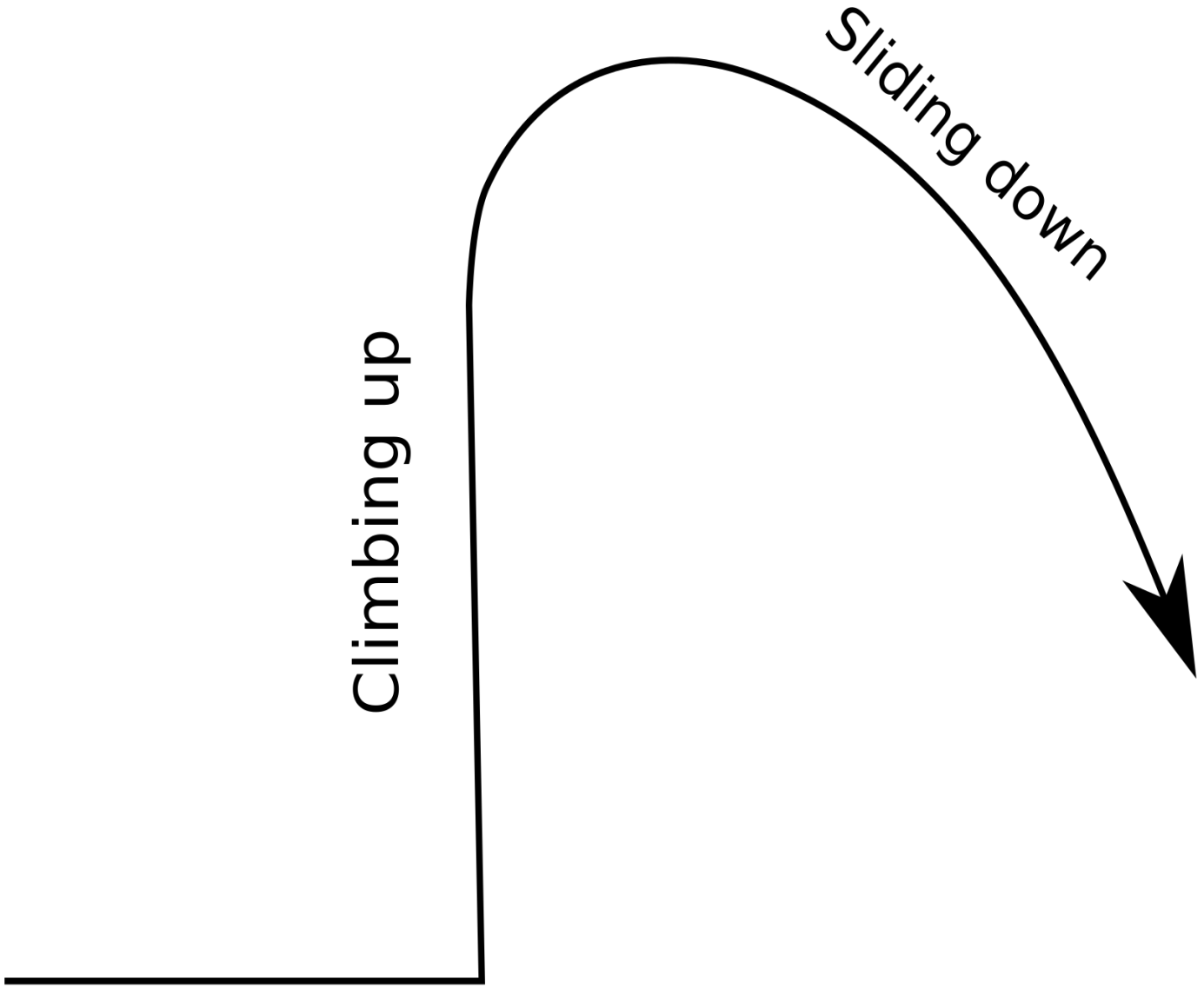
# Why You Should Give Vim a Try

Mauri Mustonen (Kazhuu)

October 3, 2019
TampereJS

# Vim Learning Wall



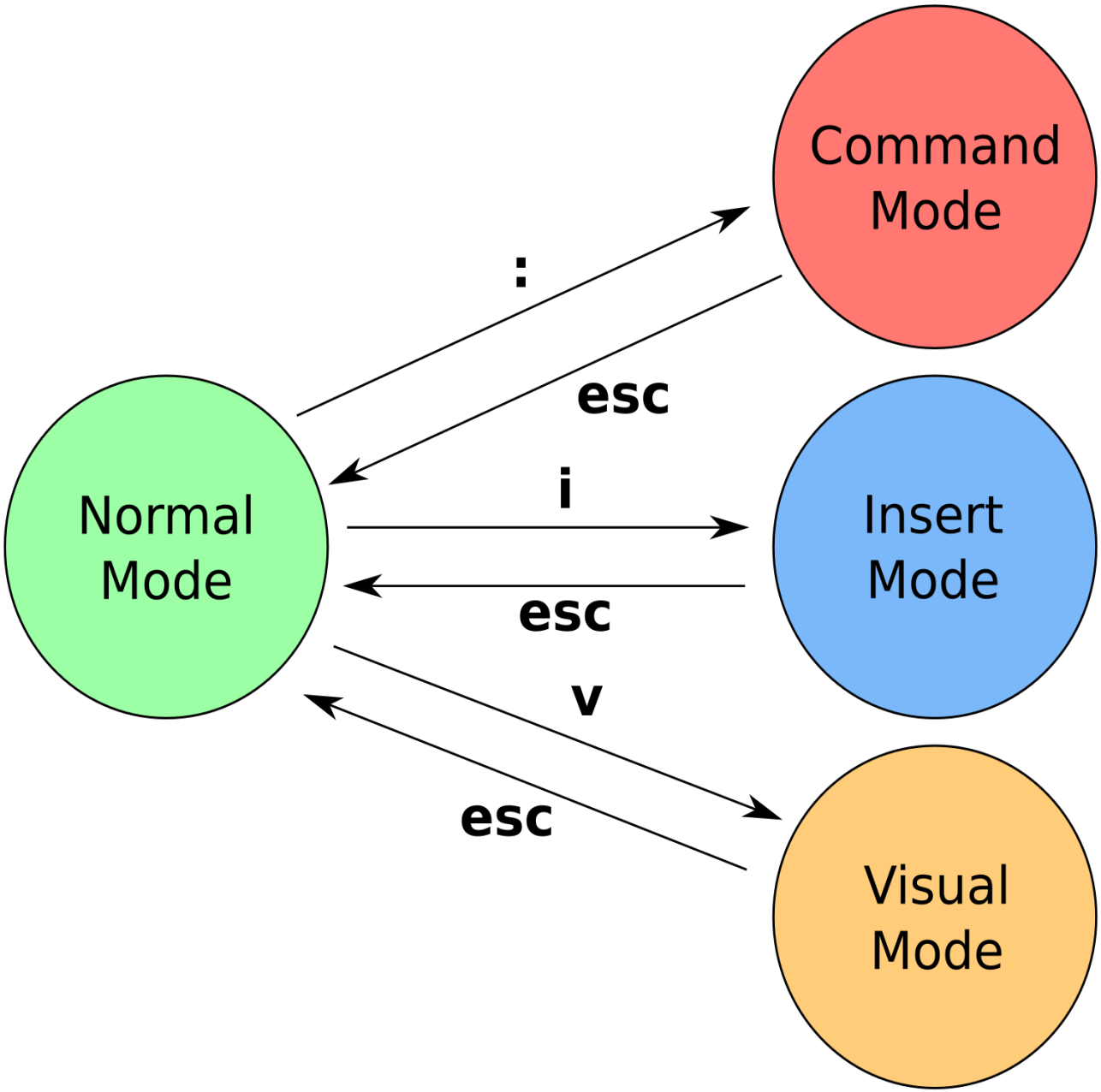
# Where To Find Me

- Web page: `https://mauri.codes`
- GitHub: Kazhuu
- LinkedIn: Mauri Mustonen

# Why I Recommend 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



# Basic Movement

| | |
|---|---|
| – **h**, **j**, **k**, **l** | ▶ left, down, up, right |
| – **w** | ▶ move for**w**ard 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 |
| – **0** | ▶ 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** — **r**edo
- **.** — repeat last command

And more...

# Composing Commands

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

- **dw**
- **yw**
- **2dw**
- **5j**
- **dd**
- **50G**

- **d**elete next **w**ord
- copy/**y**ank **w**ord
- **d**elete two following **w**ords
- move down five lines
- delete line
- 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)

**c**hange **i**nside **p**arentheses

# Command Language

## How to change returned string?

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

# Command Language

## How to change returned string?

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

ci'

**c**hange **i**nside **q**uotes

# Command Language

## How to rewrite the whole function body?

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

# Command Language

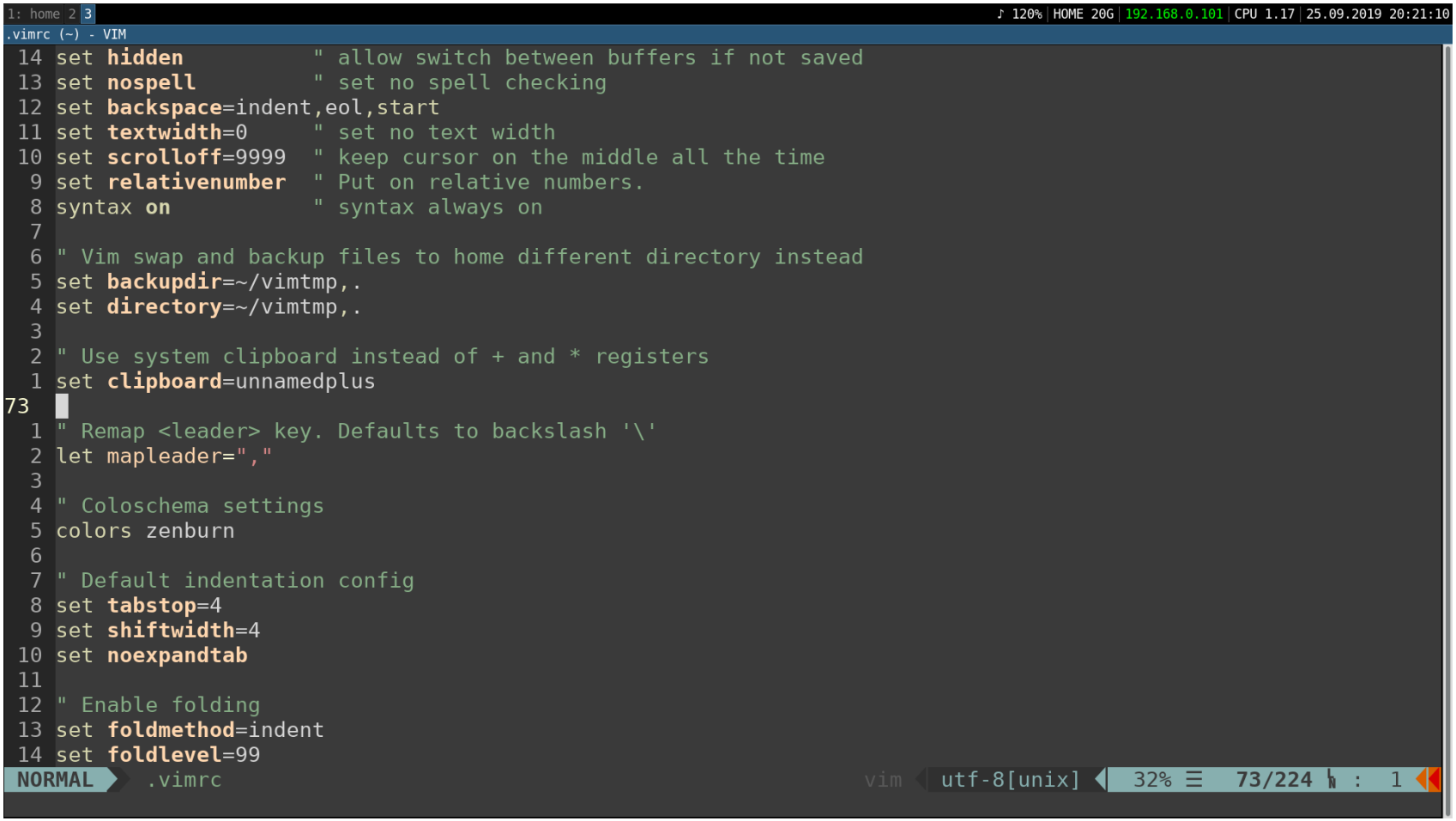
## How to rewrite the whole function body?

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

ci}

**c**hange **i**nside **b**races

# Configuring .vimrc



# Some Useful Plugins

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

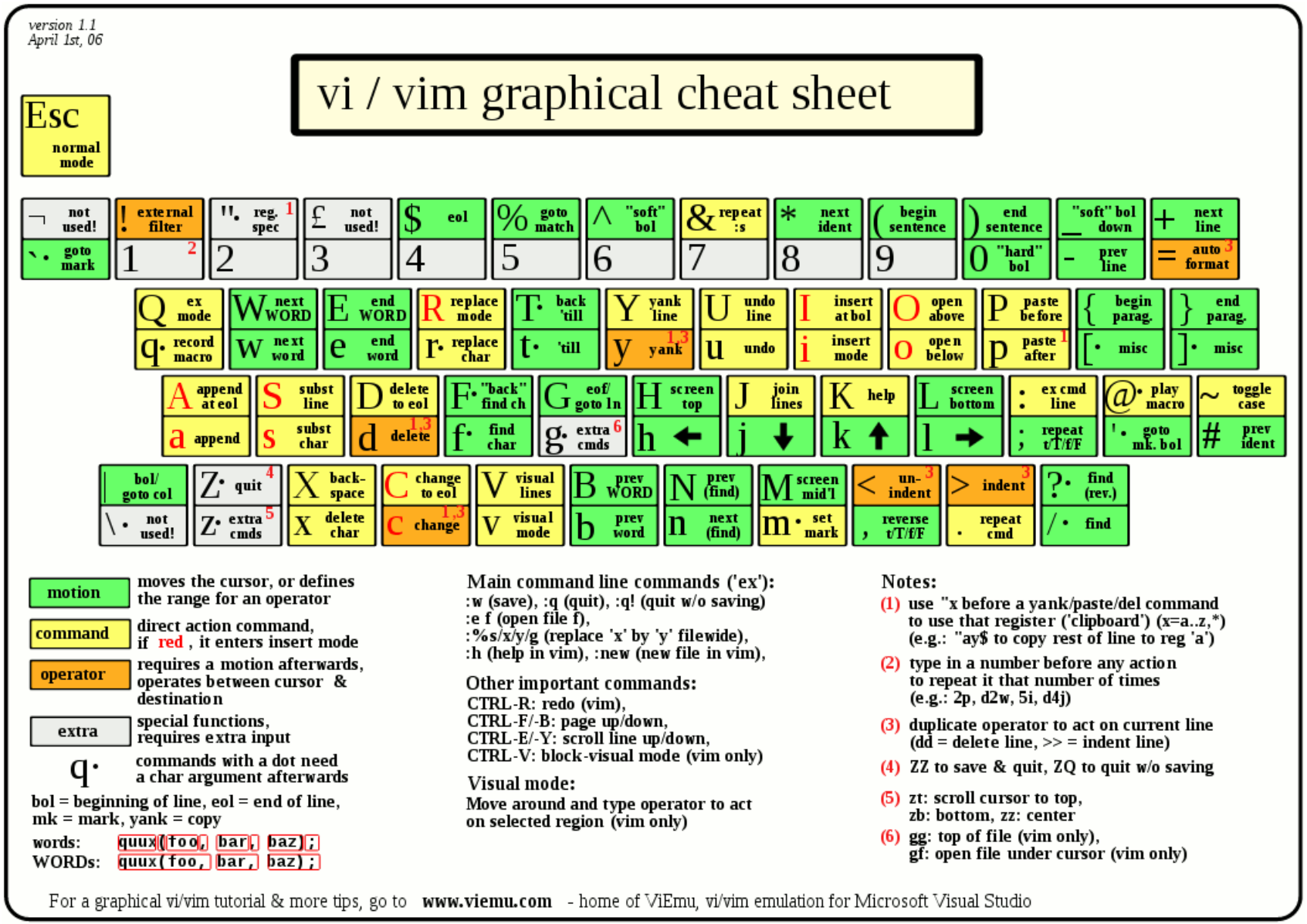
# Useful Tips

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

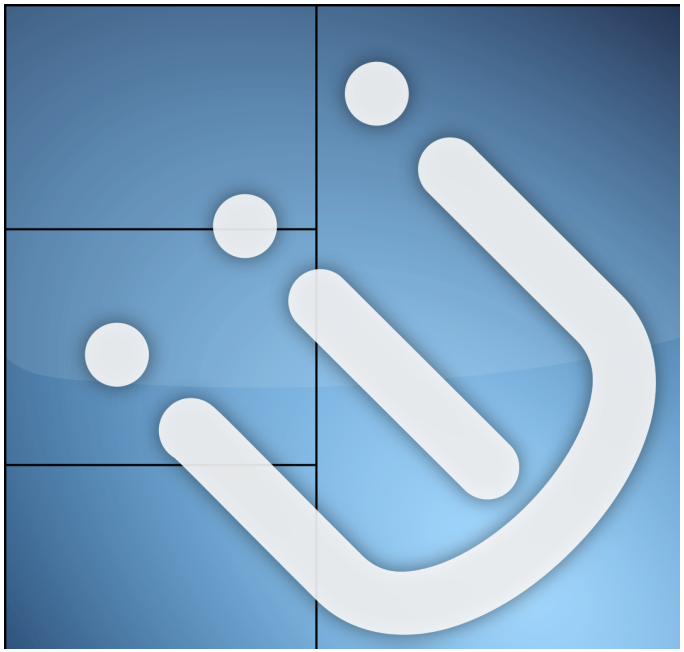
# Vimtutor

```
                    Lesson 1.1:  MOVING THE CURSOR


  ** To move the cursor, press the h,j,k,l keys as indicated. **
           ^
           k               Hint:  The h key is at the left and moves left.
      < h      l >                The l key is at the right and moves right.
           j                      The j key looks like a down arrow.
           v
  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!
```

# Vim Adventures


version 1.1
April 1st, 06
vi / vim graphical cheat sheet
Esc
normal mode
¬ not used!
` goto mark
! external filter
1 2
" reg. spec 1
2
£ not used!
3
$ eol
4
% goto match
5
^ "soft" bol
6
& repeat :s
7
* next ident
8
( begin sentence
9
) end sentence
0 "hard" bol
_ "soft" bol down
- prev line
+ next line
= auto format 3
Q ex mode
q· record macro
W next WORD
w next word
E end WORD
e end word
R replace mode
r· replace char
T· back 'till
t· 'till
Y yank line
y yank 1,3
U undo line
u undo
I insert at bol
i insert mode
O open above
o open below
P paste before
p paste after 1
{ begin parag.
[· misc
} end parag.
]· misc
A append at eol
a append
S subst line
s subst char
D delete to eol
d delete 1,3
F· "back" find ch
f· find char
G eof/ goto ln
g· extra cmds 6
H screen top
h
J join lines
j
K help
k
L screen bottom
l
: ex cmd line
; repeat t/T/f/F
@· play macro
'· goto mk. bol
~ toggle case
# prev ident
| bol/ goto col
\· not used!
Z· quit 4
Z· extra cmds 5
X back-space
x delete char
C change to eol
c change 1,3
V visual lines
v visual mode
B prev WORD
b prev word
N prev (find)
n next (find)
M screen mid'l
m· set mark
< un-indent 3
, reverse t/T/f/F
> indent 3
. repeat cmd
?· find (rev.)
/· find
motion
moves the cursor, or defines the range for an operator
command
direct action command, if red , it enters insert mode
operator
requires a motion afterwards, operates between cursor & destination
extra
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:
(1) 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')
(2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)
(3) duplicate operator to act on current line (dd = delete line, >> = indent line)
(4) ZZ to save & quit, ZQ to quit w/o saving
(5) zt: scroll cursor to top, zb: bottom, zz: center
(6) 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 - 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

# Thank you for your time!

Questions?

https://mauri.codes