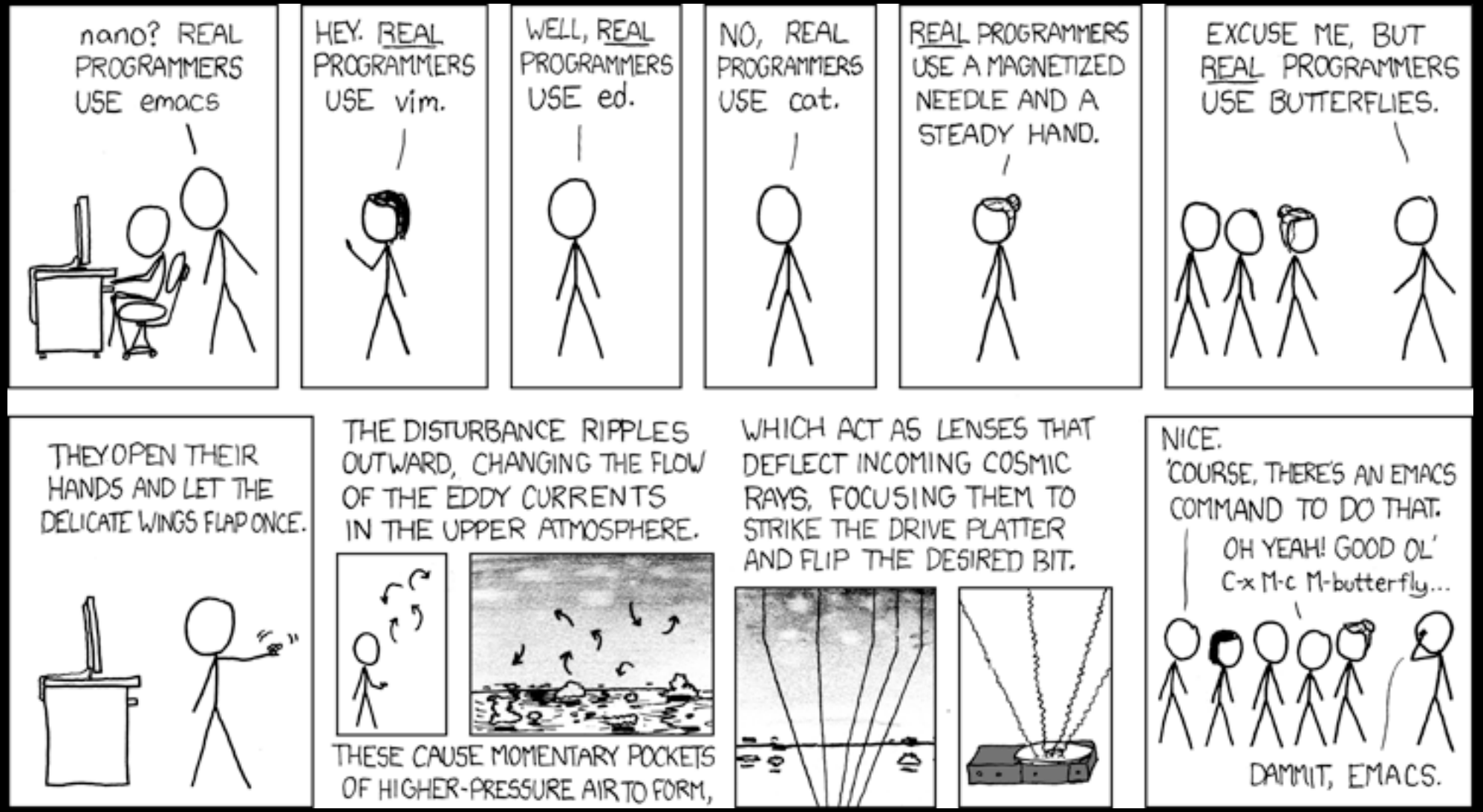
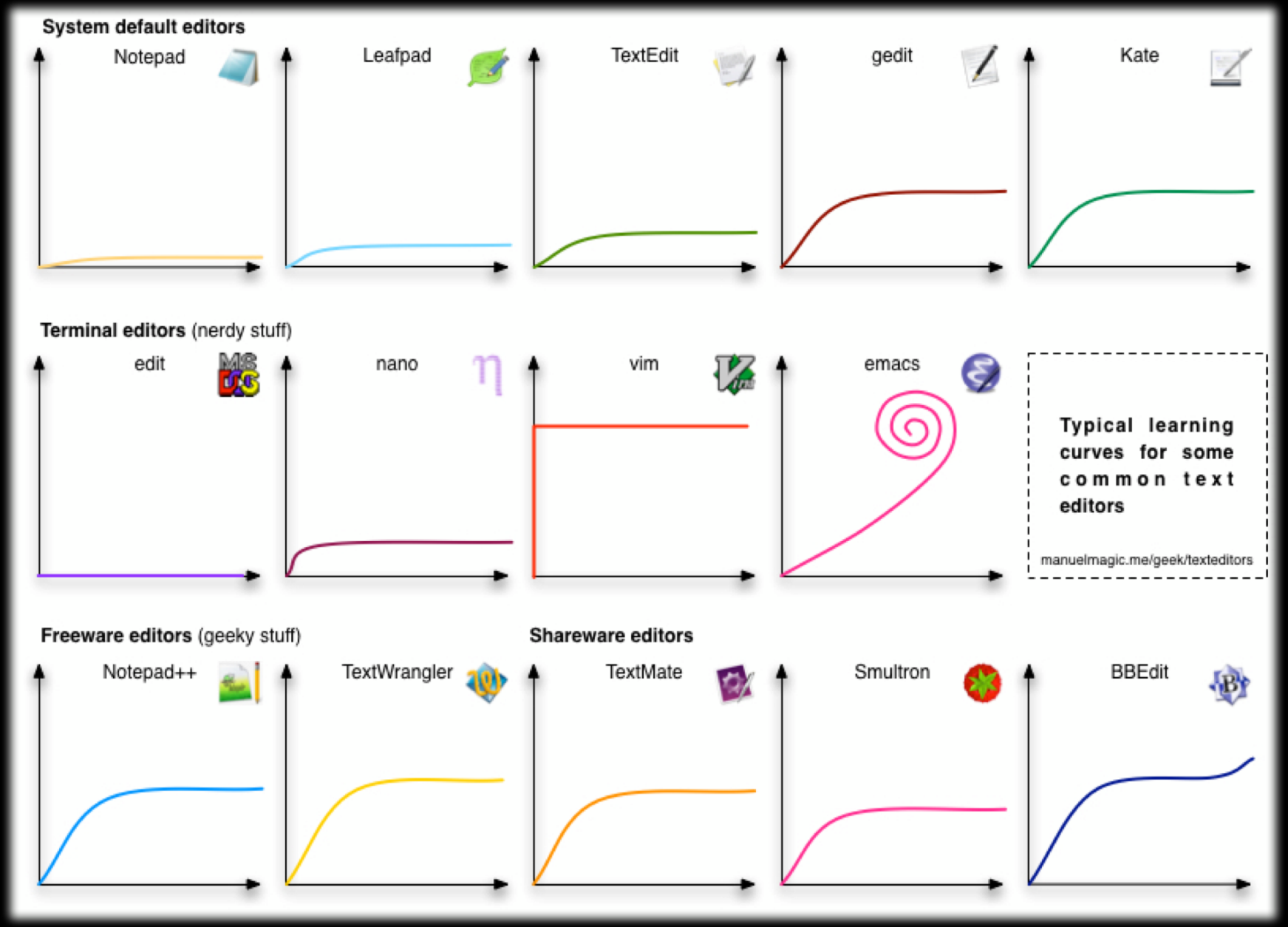
Text Editors



Learning



Experience

vi

Emacs

Do you need to bring lots of unnecessary items camping?

Maybe you will like a text editor that comes with a mail client, a web browser, and a clock

vi

Or do you appreciate beauty and simplicity?

vi seems challenging, but it's actually more natural and simple than emacs

What's the real difference?

Emacs: One mode

- Text and commands must be different combinations of keys
  - Press modifier key every time
  - Commands can be complicated
  - Move your hands a lot
  - Easy to learn, hard to mess up, but annoying and inefficient to use

vi: Two modes

- Text and commands can use the same alphanumeric keys
  - Press escape or i once
  - Commands are simple and intuitive
  - Rarely move your hands
  - Until you get used to modality, you'll be a little frustrated, but the end result is a better experience

vi

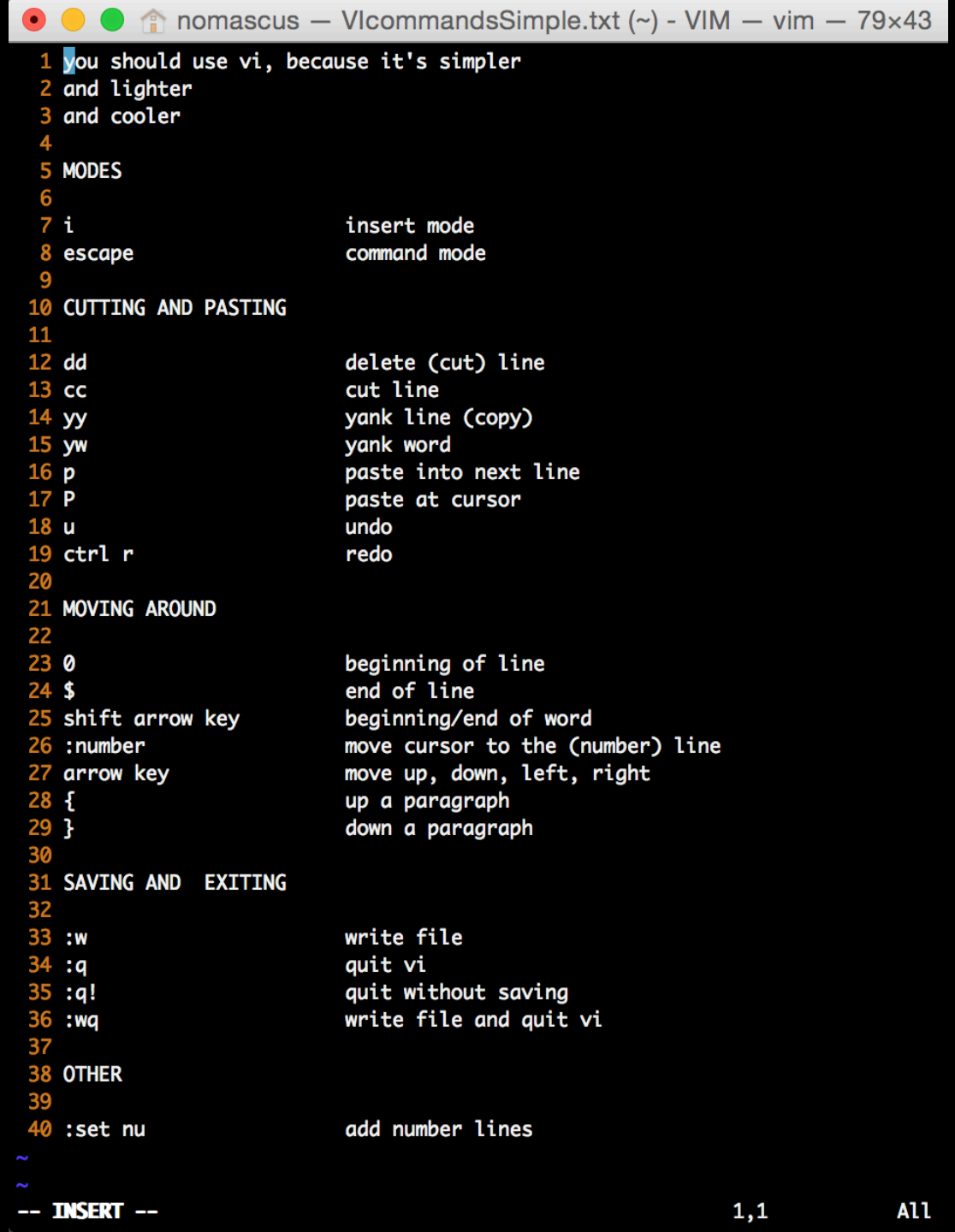
My take on EmacsVsVi (for text editing, ignoring all the additional stuff Emacs can do) has always been that Emacs is easier to *learn*, but Vi is easier to *use*. A modeless tool is easier to get a handle on, because there's never any question as to what state it's in. But it means you need a lot more knobs and buttons to get everything done, making it necessarily bigger and klunkier. A tool with modes is harder to learn, but once you get the hang of it, can be quicker to use if well designed. –

Stefan Vorkoetter (who uses Vi after a brief stint with Emacs in the late 80's).

Insert Mode

Press the i key to start INSERTING text

Move around with arrow keys



Command Mode

Press the escape key to leave insert mode

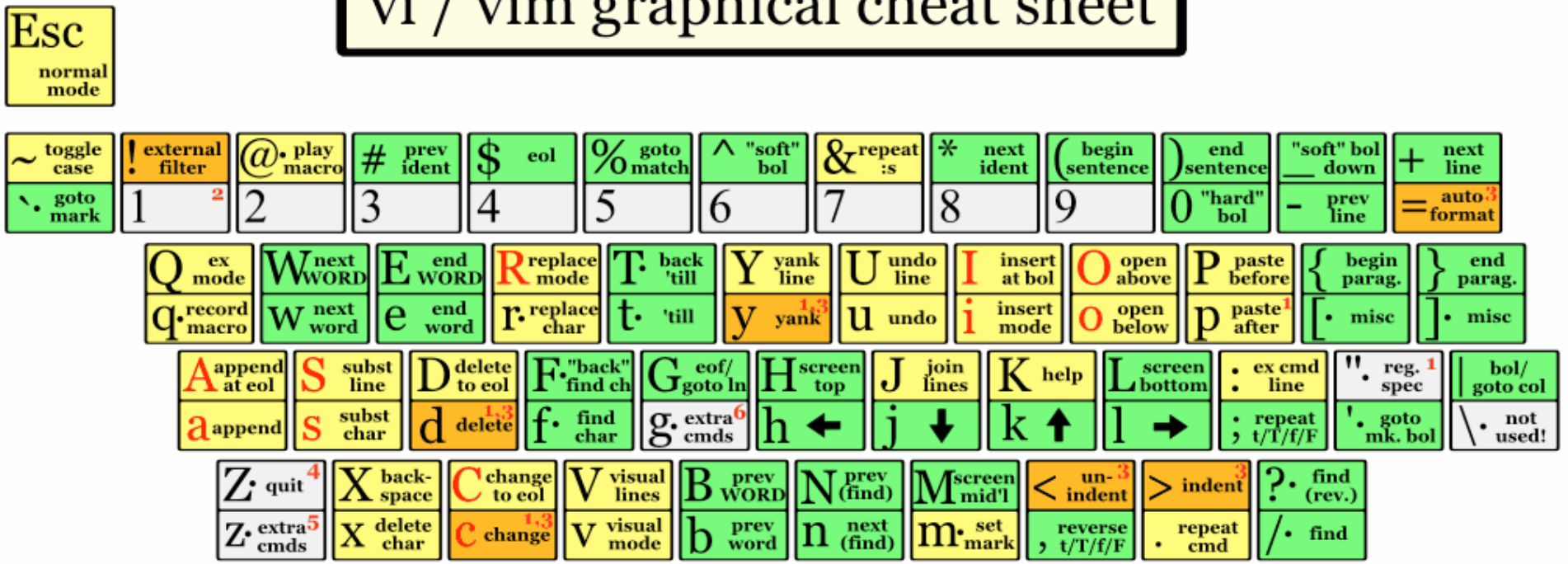
Cut & paste
Move
Search
Save & exit

Commands are simple and intuitive

You don’t have to constantly press the control key

version 1.1
April 1st, 06

vi / vim graphical cheat sheet



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

There's also a video game to help you learn vi

(you might be able to run this in emacs if you mash enough keys together…who knows…)