

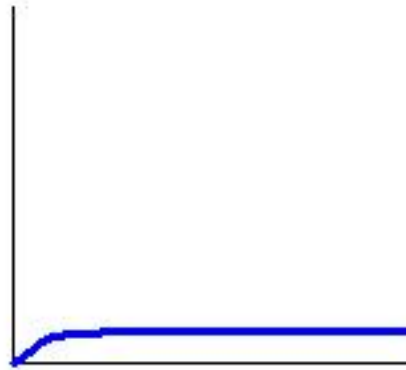
VIM
for
VICTORY

@RoyOsheroVe

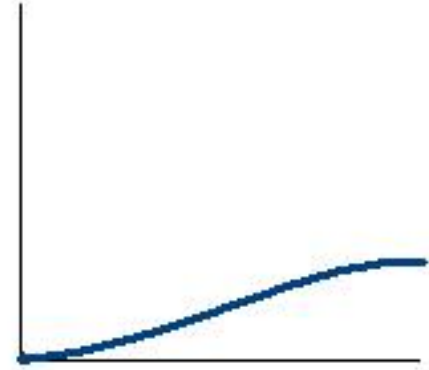
Learning

Classical learning curves for some common editors

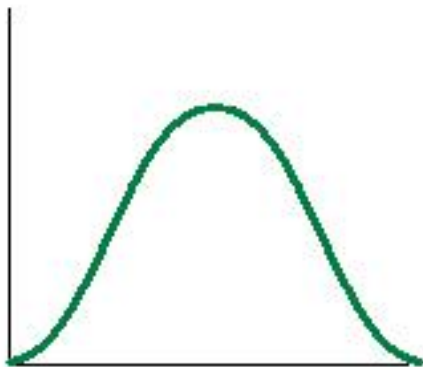
Notepad



Pico



Visual Studio



vi



emacs



11-17-09

Why VIM?

- Lightweight
- Makes some editing really easy
- Everywhere
- Remote coding via ssh
- Pair programming (ssh + tmux)
- Important skill (like regex)
- Feel Stupid, Learn Something

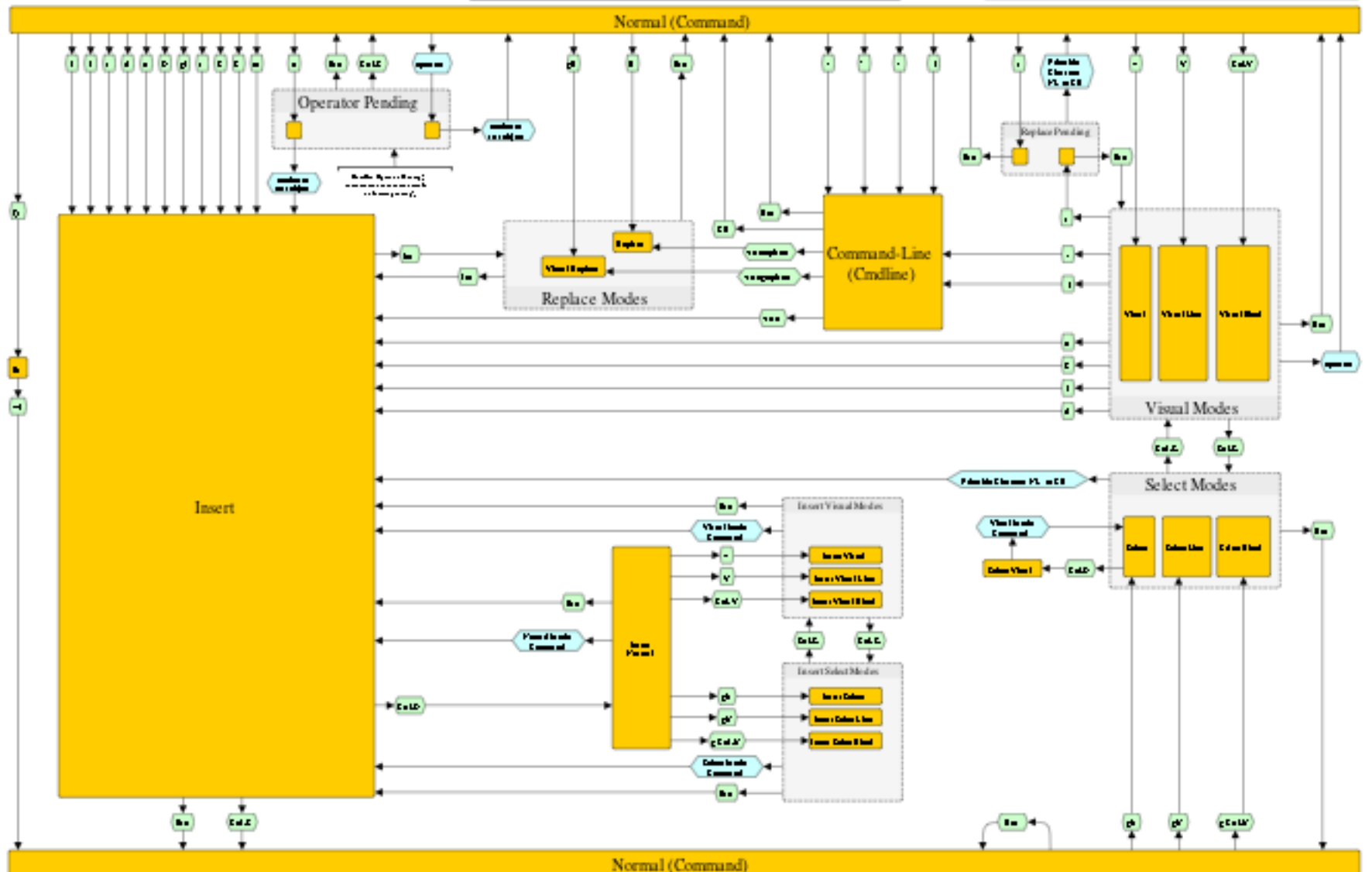
Demo

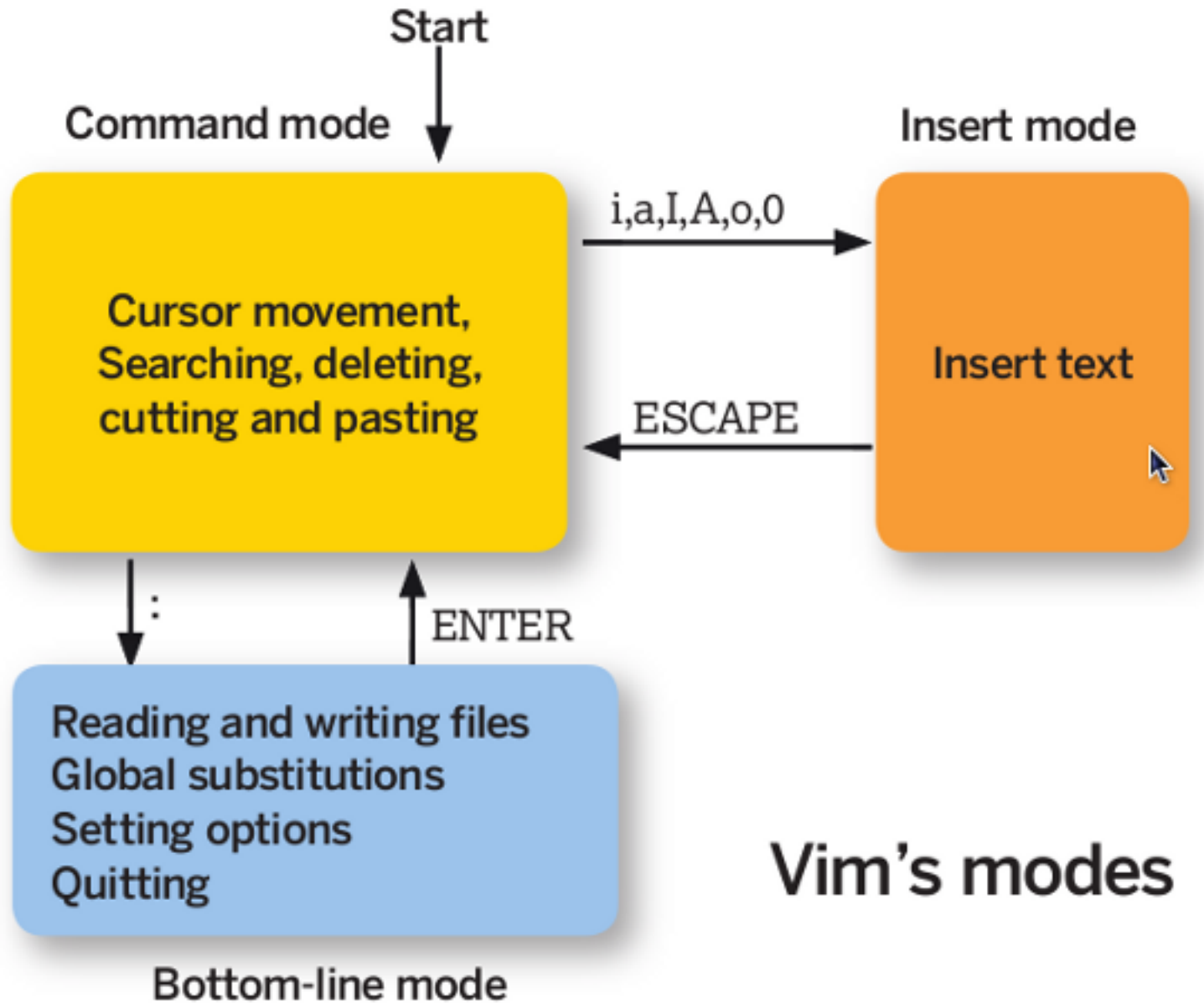
Favorite keys

- O and O
- Viw
- Cw and ciw
- Shift-C
- Shift-v to line selection
- Cc to change line
- Dd to delete line

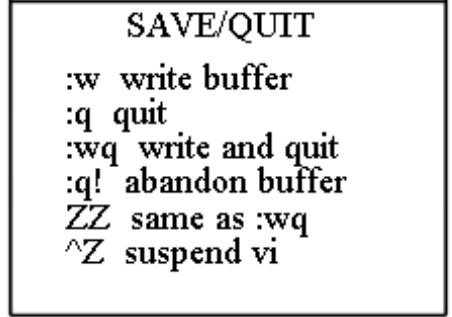
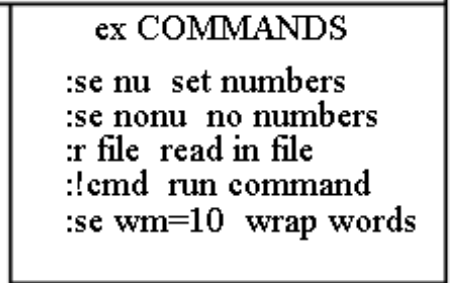
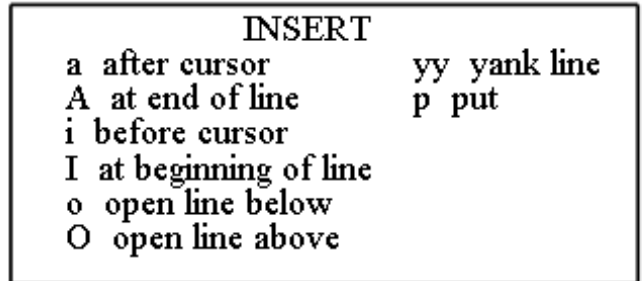
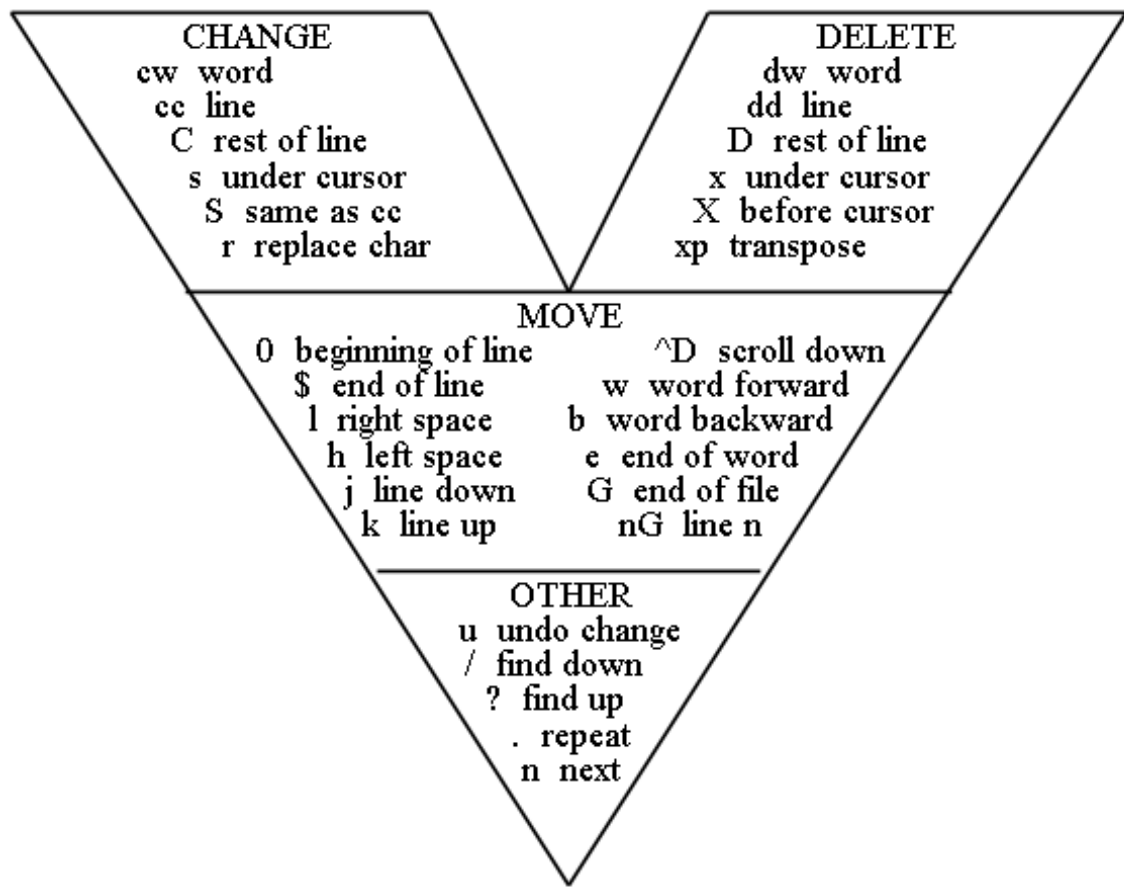
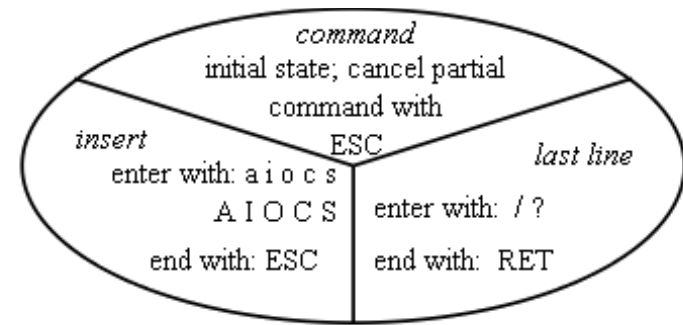
VIM Modes Transition Diagram

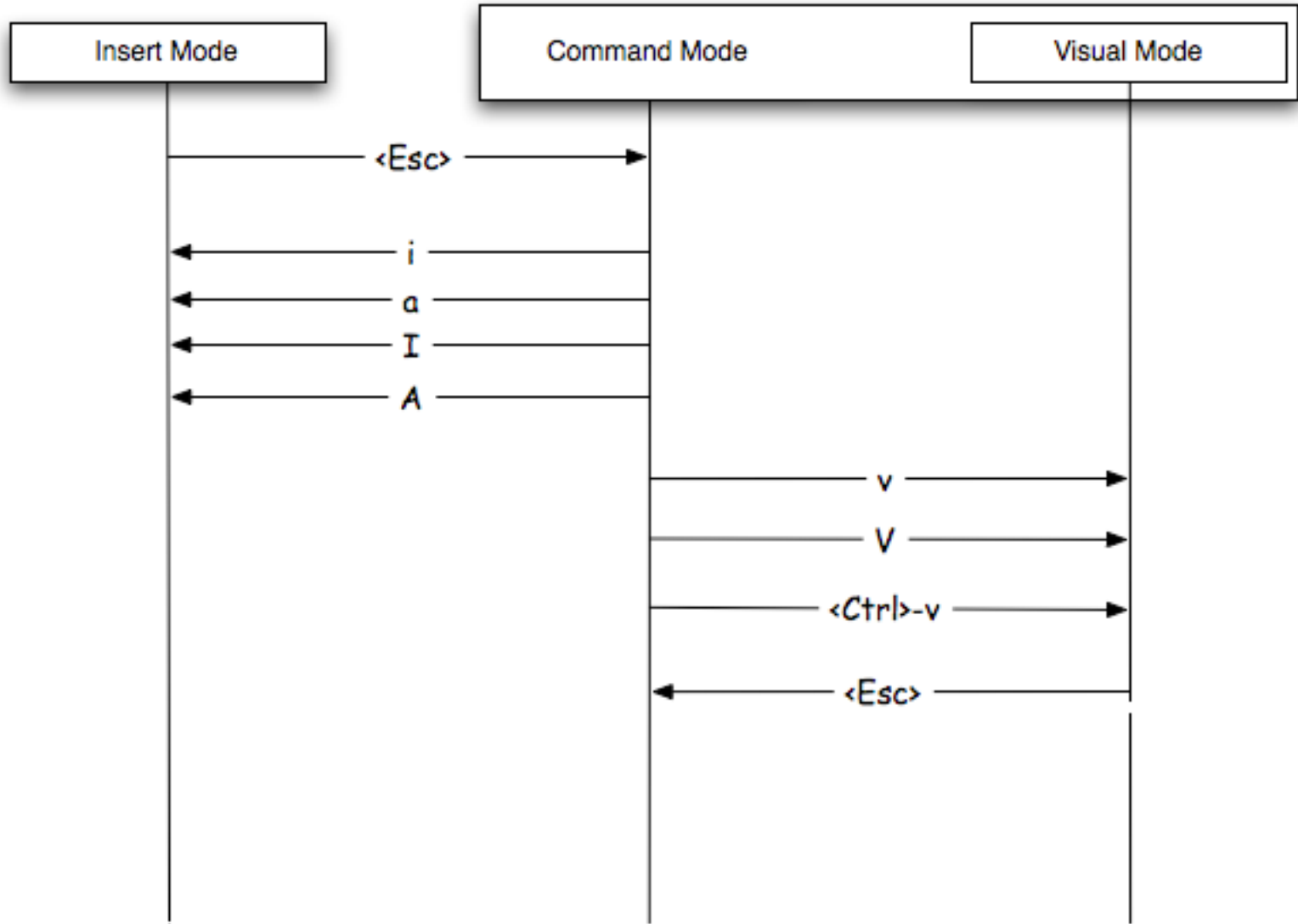
By darcyparker@gmail.com Draft (2/22/2012) Feedback welcome





Vim's modes





vi / vim graphical cheat sheet

Esc
normal mode

~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
· goto mark	1	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto ³ 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 ^{1,3}	u undo	i insert mode	o open below	p paste ¹ 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	" reg. ¹ spec	bol/ goto col	
a append	s subst char	d delete ^{1,3}	f find char	g extra ⁶ cmds	h ←	j ↓	k ↑	l →	· repeat ; t/T/f/F	' spec mk. bol	\ · not used!	
Z quit ⁴	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un- ³ indent	> indent ³	? find (rev.)			
Z extra ⁵ cmds	X delete char	c change ^{1,3}	v visual mode	b prev word	n next (find)	m set mark	, reverse t/T/f/F	· repeat cmd	/ 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, · = 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)

Configuration

- .vim file
- .vim directory
- Vim scripts and plugins
- Google “**vim Janus**”
- Github **astrails dotvim**

Favorite Plugins

- Surround plugin
- Rails
- NERdtree
- Command-t (fuzzy finding)

Other vims

- Vi-like command line in zsh (bindkey -v)
- vsVim for visual studio
- ViEmu for Word
- Vimium and vimperator for firefox and chrome