vi / vim graphical cheat sheet

Esc — normal mode

Keyboard keys and their functions:

- `~` toggle case
- `` ` `` goto mark
- `!` external filter
- `@.` play macro
- `q.` record macro
- `Q` ex mode
- `1`, `2` [2], `3`, `4`, `5`, `6`, `7`, `8`, `9`, `0` "hard" bol
- `#` prev ident
- `$` eol
- `%` goto match
- `^` "soft" bol
- `&` repeat :s
- `*` next ident
- `(` begin sentence
- `)` end sentence
- `_` "soft" bol down
- `+` next line
- `=` auto[3] format

- `A` append at eol
- `a.` append
- `S` subst line
- `s` subst char
- `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

- `Z.` quit [4]
- `Z.` extra cmds [5]
- `X` back-space
- `x` delete char
- `D` delete to eol
- `d` delete [1,3]
- `C` change to eol
- `c` change [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
- `"` reg, spec [1]
- `'` goto mk. bol

- `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]
- `>` indent [3]
- `,` reverse t/T/f/F
- `.` repeat cmd
- `?.` find (rev.)
- `/.` find
- `|` bol/ goto col
- `\.` not used!

Legend

- **motion** (green): moves the cursor, or defines the range for an operator
- **command** (yellow): direct action command, if **red**, it enters insert mode
- **operator** (orange): requires a motion afterwards, operates between cursor & destination
- **extra** (grey): 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
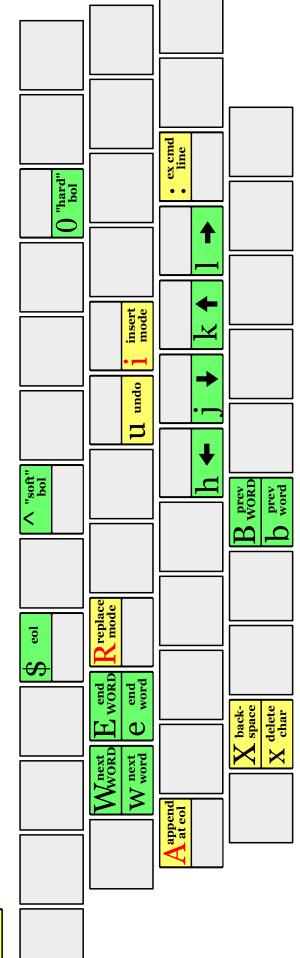
version 1.1
April 1st, 06

vi/vim lesson 1 – basic editing

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

Esc normal mode

Esc

Key	Function
$	eol
^	"soft" bol
0	"hard" bol
W	next WORD
w	next word
E	end WORD
e	end word
R	replace mode
A	append at eol
i	insert mode
u	undo
h ←	
j ↓	
k ↑	
l →	
:	ex cmd line
X	back-space
x	delete char
B	prev WORD
b	prev word

Basics:

h j k l are vi/vim cursor keys – use them as they are much closer than regular cursor keys!

Use **i** to enter insert mode, cursor turns from a block into a vertical line, and you can type in text. Use **Esc** to return to normal mode.

Use **x** to delete the current character, or **X** to delete the one to the left

Use **A** to go insert text at the end of the line (wherever you are in the line!)

(Note: insert mode is actually very similar to a regular editor, you can use cursor/navigation keys, backspace, delete...)

Extras:

u to undo the last action – traditional vi has a single level, while vim supports unlimited undo (CTRL - **R** to redo)

0 jumps directly to the beginning of the line, **$** to the end, and **^** to the first non-blank

Use **w b e** to move along 'words'. A 'word' is a sequence of all alphanumeric or punctuation signs: quux[foo, bar], baz) ;

Use **W B E** to move along WORDs. A 'WORD' is a sequence of any non-blank characters: quux(foo, bar, baz) ;

Use **R** to enter insert mode with an overstrike cursor, which types over existing characters.

: **w** and press enter to save, **:** **q** and enter to quit.

For the rest of the tutorial & a full cheat sheet, go to **www.viemu.com** – home of ViEmu, vi/vim emulation for Microsoft Visual Studio