

VI (VIM) FOR WINDOWS USERS

Learn by comparison with Windows editor

`vi` or `Vim` (Vi IMproved) is a powerful text editor with origins in UNIX and Linux. It's available for Windows too (www.vim.org). For the uninitiated user of a GUI text editor (Word, Notepad, Notepad++, etc.), it can be a challenge. There are some conceptual differences that can make it frustrating. A way to bridge this gap is to compare and map the two behaviors.

Concept

Basic Operations

Useful Settings

More Operations

Tips and Tricks

Conclusion

Concept

`vi` operates in two modes, Command mode and Insert mode. Standard GUI text editors are always in one mode, which is the Insert mode.

Command mode (also called Normal mode) – All keys pressed execute a command rather than adding text to the document. Note that there are different forms – single key, multiple keys in sequence and colon commands (where the command is prefixed with “:” character).

Insert mode – Standard GUI text editor behavior. All keys pressed adds text to the document.

Typically, the bottom left corner indicates the mode. When in Insert mode, the text “--INSERT--” is displayed. It's empty in Command mode.

Listed below are some conceptual differences. Note the comments in **red** below. These usually frustrate a beginner.

VI

Command

GUI TEXT EDITOR

Shortcut key

Commands are case sensitive

Buffer

Starts in Command mode

Press i, a, I, A to enter into Insert mode.

You have to press one of these keys before text gets added to the document.

Press ESC key to return to Command mode.

When in Insert mode, you have to press the ESC key before executing a command.

Commands do not work in Insert mode.

For copy/cut/paste, you can use the local clipboard or the system clipboard.

Mouse and scroll wheel may not work to move the cursor. Have to use one of the many keyboard commands for cursor movement.

Shortcut keys are not case sensitive

Document

Starts in Insert mode

Always in Insert mode, so you just keep typing to add text.

Always in Insert mode. No mode switching is required.

Shortcuts are active and can be used.

Shortcuts are active in Insert mode.

Copy/cut/paste always uses the system clipboard

Mouse and scroll wheel works as expected.

Basic Operations

Following are the minimal set of commands which will help you operate `vi`. They may not be the most efficient way in `vi` but will keep you out of trouble. The equivalent Windows shortcut keys in typical text editors are mentioned for comparison.

vi COMMAND	SHORTCUT KEY	DESCRIPTION
<code>:help {command}</code>		show command usage
ESC		Return to command mode
File operations		
<code>:q</code>	ALT F4	quit/close the application
<code>:q!</code>	ALT F4, 'No' to save	quit/close without saving
<code>:e</code>	CTRL o	edit/open a file
<code>:w</code>	CTRL s	write/save to file
<code>:bn</code>	CTRL TAB	cycle forward through open buffers/documents
<code>:bp</code>	CTRL SHIFT TAB	cycle reverse through open buffers/documents

:bd	CTRL F4	Close current buffer/document
:buffers	Window menu bar item	Show all buffers/documents

Cursor Movements

h	Left Arrow	
j	Down Arrow	
k	Up Arrow	
l	Right Arrow	
w	CTRL Right Arrow	Move forward by a word
b	CTRL Left Arrow	Move backward by a word
CTRL f	Page Down	
CTRL b	Page Up	
gg	CTRL Home	Beginning of document
G	CTRL End	End of document
:n	CTRL g (in some editors)	Go to line number <i>n</i>

Enter into Insert mode

i		Insert before the cursor
a		Insert after the cursor
I	Home	Insert at the beginning of the line
A	End	Insert at the end of line
R	Insert	Insert and overwrite text as you type

Deletion

x	Delete
X	Backspace

Undo/Redo

u	CTRL z	Undo
CTRL r	CTRL y	Redo

Find

<code>/ {string}</code>	CTRL f	Search <i>{string}</i>
<code>*</code>	CTRL F3 (in some editors)	Word on the cursor is set as the find string
<code>n</code>	F3 (in some editors)	Find Next
<code>N</code>	SHIFT F3 (in some editors)	Find Previous
<code>:noh</code>		clear last search highlights

Find and Replace

<code>:%s/ {Foo}/ {Bar}/gc</code>	CTRL h (in some editors)	<i>{Foo}</i> is the search string and <i>{Bar}</i> is the replace string. The <code>/gc</code> asks for confirmation before replace.
-----------------------------------	--------------------------	--

Copy Cut Paste

<code>v</code>	SHIFT Arrow keys	Enter visual mode (character) and start marking by using cursor movement keys h, j, k, l
<code>V</code>		Enter visual mode (line) and start marking full lines by using up/down cursor movement keys j, k
<code>CTRL v</code> or <code>CTRL q</code>		Enter visual mode (block) and start marking by using cursor movement keys h, j, k, l
<code>ggVG</code>	CTRL a	mark/select full buffer/document
<code>y</code>	CTRL c	yank/copy
<code>d</code>	CTRL x	delete/cut
<code>p</code>	CTRL v	Lowercase p, paste after the current cursor position
<code>P</code>		Uppercase P, paste before the current cursor position
<code>" + y</code>	CTRL c	copy to system clipboard
<code>" + d</code>	CTRL x	cut to system clipboard
<code>" + p</code>	CTRL v	Lowercase p, paste after the current cursor position from system clipboard
<code>" + P</code>		Uppercase P, paste before the current cursor position from system clipboard

File Status

<code>CTRL g</code> or <code>:f</code>		Prints the current file information at the bottom status line
--	--	---

Useful Settings

Following are some settings in `vi` which will help getting comfortable with it. These are `colon commands` which can be applied in command mode.

<code>:set autochdir</code>	Opening a file, the base directory is set to the location of the current buffer
<code>:set hlsearch</code>	Highlight the text found during search
<code>:set ignorecase</code>	Ignore Case during find
<code>:set incsearch</code>	Incremental search, find as you type
<code>:set list</code>	Display hidden character like tab and end of line
<code>:set number</code>	Show line numbers
<code>:set ruler</code>	Display the current cursor position (row and column) at the bottom
<code>:set tabstop=4</code>	Tab shifts by 4 characters
<code>:set shiftwidth=4</code>	Indentation shift with <code>>></code> and <code><<</code> commands
<code>:set expandtab</code>	Insert spaces instead of tab character

More Operations

The following are more commands in `vi` which might give you some insight into its power and flexibility. There are no real equivalents in standard text editors.

<code>.</code>	Repeat previous command
<code>\$</code>	Move to end of current line
<code>0</code>	Move to start of current line
<code>o</code>	Open a new line below and switch to insert mode
<code>O</code>	Open a new line above and switch to insert mode

<code>r</code>	Overwrite one character under the cursor
<code>D</code>	Delete characters under the cursor until end of line
<code>J</code>	Join/Combine two lines
<code>yy</code>	Copy the full line where the cursor is
<code>yw</code>	Copy a word
<code>dd</code>	Delete the full line where the cursor is
<code>dw</code>	Delete a word, remain in command mode
<code>cw</code>	Change Word, delete the word and switch to insert mode
<code>zt</code>	Scroll current cursor position to top of page
<code>zz</code>	Scroll current cursor position to center of page
<code>zb</code>	Scroll current cursor position to bottom of page

Useful in source code

<code>%</code>	Jump to matching brace
<code>[{</code>	Jump to previous brace
<code>>></code>	Indent right
<code><<</code>	Indent left
<code>></code>	Indent right in visual mode
<code><</code>	Indent left in visual mode
<code>m {name}</code>	Named bookmark, where <i>{name}</i> is a lower case character
<code>` {name}</code>	Go to <i>{name}</i> bookmark
<code>`.`</code>	Go to last edited line

Window manipulation

<code>CTRL w s</code>	Horizontal split window
<code>CTRL w v</code>	Vertical split window
<code>CTRL w w</code>	Switch to other window
<code>CTRL w c</code>	Close split window

:tabnew

New tab window

:tabnext

Switch to next tab window

:tabprevious

Switch to previous tab window

:tabclose

Close tab window