

How to use vi like a PRO?

Introducing VIM

What is VIM?

- ▶ It stands for VI-IMproved
- ▶ It is one of the most popular *vi clones*
- ▶ Bram Moolenaar created it in 1988 when he felt that the vi editor was lacking a lot of features
- ▶ It aims at extending the original vi features as well as adding new ones.
- ▶ The official website is www.vim.org
- ▶ It does not differ than vi in the basic usage and functionality. For example, it does have insert mode and command mode, all the commands that you previously learned in this course will apply to vim.

How to get it?

- ▶ Many versions of UNIX and UNIX like systems contain vim already.
- ▶ You can check by typing vim at the command line
- ▶ If it is not installed, use your package installer command to obtain it. For example yum install vim on Red Hat and Red Hat like Linux, apt-get install vim on Ubuntu and other debian systems.
- ▶ For Microsoft Windows, and if you want to compile from source, you can visit <http://www.vim.org/download.php>
- ▶ The current version of vim is 7. It is recommended that you upgrade to the latest version. You can check your current version by issuing `:version` command
- ▶ The following slides will mention some of the most important improvements of vim over vi

Integrated help system

- ▶ You can invoke help anytime by using the `:help` command
- ▶ Vim supports autocomplete feature. It is activated when pressing TAB after the first character(s) of the topic is entered
- ▶ Pressing ENTER will display the help screen associated with this topic
- ▶ You can move inside the help screen the same way you use navigation in vi. That is, CTRL-F, CTRL-B ...etc.
- ▶ You can exit the help screen at anytime using ESC-q

VIM startup arguments

- ▶ The command line options dictate the way vim will operate. They can either start with a single hyphen (-) or a double one (--).
- ▶ The following is some of the command line options that can be used with VIM:
 - ▶ `-b` : edits the file in binary mode. Example usage: inserting text in an image for cryptography purposes.
 - ▶ `-c command` : executes *command* after the VIM starts. Vi has this feature, but vim allows up to 10 commands each preceded by `-c`. Example usage: open a file and perform a case insensitive search for a word: `vim -c "set ic" -c /word`
 - ▶ `-d` : performs *diff*-like command on two or more files. However, it's much powerful than using the default diff command.
 - ▶ `-o file1 file2 filex` : opens the following files in separated windows. You can cycle through the windows using CTRL-w w. You can use CTRL-w followed by h,j,k, or l to cycle through file windows.
 - ▶ `-O` : same as `-o` but opens windows in vertical mode instead of horizontal.

Introducing easier movement

- ▶ *number*?: move the cursor to the first nonblank character of the line *number*, where % is the percentage of the total number of lines.
- ▶ :go n: go to *byte n*. This command considers all characters including whitespaces and newlines. It may be useful when editing binary files.

Visual mode

- ▶ It is used to highlight text as you navigate through the file. It provides the convenience of seeing the text that you will further process using vim commands.
- ▶ You can switch on and off the visual mode by pressing `v`.
- ▶ The following commands can be used in visual mode:
 - ▶ `naw, naW`: select the `n` number of words (whitespace ignored). The `w` (lowercase) separates words by punctuation, while `W` (uppercase) ignores punctuation.
 - ▶ `niw, niW` : select the `n` number of words (whitespace considered a word). The `w` and `W` act the same way as in `aw` and `aW`.
 - ▶ `nip` : select the `n` number of paragraphs (blank lines considered)
 - ▶ `nap` : select the `n` number of paragraphs (blank lines ignored)

Cheat Sheet

Command	Usage
<code>vim -b</code>	Start vim in binary mode
<code>vim -c command1 -c command2...</code>	Execute the command(s) following the -c. Multiple commands can be supplied.
<code>vim -d file1 file2</code>	Show the differences between file1 and file2
<code>vim -o file1 file2</code>	Open file1 and file2 in separate vertical windows
<code>vim -O file1 file2</code>	Open file1 and file2 in separate horizontal windows
<code>%number</code>	Move the cursor to the specific <i>number</i> percentage of the file. Percentage is calculated on the total number of lines and not characters.

Cheat Sheet

Command	Usage
<code>go n</code>	Moves the cursor to the <code>n</code> bytes of the file
<code>v</code>	Toggle the visual mode on and off
<code>naw</code>	In visual mode, adds the <code>n</code> words to the selection, whitespace ignored. Punctuation considered
<code>naW</code>	Same as <code>naw</code> but punctuation is ignored
<code>niw</code>	In visual mode, adds <code>n</code> words to the selection, whitespace is considered a word. Punctuation is considered
<code>niW</code>	Same as <code>niw</code> but punctuation is ignored
<code>nip</code>	In visual mode, add paragraphs to the current selection. Blank lines are considered as paragraphs
<code>nap</code>	Same as <code>niw</code> but blank lines are ignored.