

A VIM Editor Crash Course

VIM is a powerful command line interface (CLI) text editor. It was designed to be used solely with a keyboard, so a lot of commands and actions that we may be accustomed to are very different. Because VIM uses the keyboard for commands and text entry, it needs to be able to distinguish between a command being issued, or text being typed into the document.

Table 1 - Opening VIM

| Command | Action Taken |
|----------------|--|
| vim | Opens vim |
| vim <filename> | Opens <filename> in vim; creates file if it does not exist |

When VIM is first opened, it places the user in EDIT mode. This is the mode that allows users to delete, cut, copy, paste, etc. Some useful commands are shown below. **IMPORTANT: All commands are case sensitive! You will not receive any feedback that you are pressing keys until the command executes.** Patterns can be observed in how commands are utilized along with movement commands.

Table 2 - Keyboard Shortcuts for Editing

| Key Stroke | Action Taken |
|------------|--|
| G | Move cursor to last line of the file |
| gg | Move cursor to the first line of the file |
| 5G | Move cursor to line 5 |
| \$ | Move cursor to the end of the current line |
| 0 (zero) | Move cursor to the beginning of the current line |
| x | Delete single character under the cursor |
| dd | Cut/delete current line |
| dG | Cut/delete from current line to the end of the file |
| 5dd | Including the current line, cut/delete the next 5 lines |
| yy | Yank/copy current line. NOTE: the variations shown for deletion apply to yanking as well |
| p | Put/paste below current line |
| P | Put/paste on current line. NOTE: can be thought of as above, it shoves everything down |
| h j k l | Move cursor left (h), down (j), up (k), or right (l) |

These commands should demonstrate most of the tasks needed for everyday use. VIM is capable of doing much more than what is being shown in this handout, and an Internet search can show the many capabilities and resources available for learning VIM. As an example, here is a link to a VIM cheat sheet: <https://vim.rtorr.com>

When it comes to actually inputting text, VIM offers many different ways to enter INSERT mode. Only a couple are shown below. You will know if you are in EDIT or INSERT mode by looking in the bottom left corner of the terminal window. If it is empty, you are in EDIT mode. If it shows -- INSERT -- , then you are in INSERT mode. *If you ever get lost, hit the Escape key twice. This will always ensure you are in EDIT mode.*

Table 3 - Entering INSERT Mode

| Key Strokes | Action Taken |
|-------------|---|
| i | Insert cursor before current position. NOTE: Likely the only method you'll need for now |
| o | Insert a new line below cursor and move cursor there |
| O | Insert a new line at cursor position. NOTE: Can also be thought of as above |

When you are in INSERT mode, you are able to type and use the arrow keys to move around. The Backspace key works in INSERT mode, but not EDIT mode. After some typing, if you find that you need to make some edits, press Escape to enter EDIT mode, make your edits, and then return to INSERT mode.

To issue specific commands to VIM such as save, quit, search, the user needs to type: (colon) followed by the command. This can **only** be done from EDIT mode. The colon puts the user into COMMAND mode.

Table 4 - File Commands

| Key Strokes | Action Taken |
|---------------|---|
| :w <filename> | Write/save to <filename>. If <filename> is not specified, the current file is saved if it already exists. |
| :q | Quit VIM. You cannot quit if you have modified the file in any way. |
| :q! | Force quit. This is how you quit without saving. |
| :wq | Write/save and quit |