

# ViM Editor 101

Written by Alistair Ross, The Ultimate Linux Newbie Guide

<http://www.linuxnewbieguide.org>



# Getting familiar with VIM

- Moving around - hjkl
- Repetition: 3j
- Saving and Quitting via command mode.
- The modes:
  - NORMAL MODE
  - INSERT MODE
  - REPLACE MODE
  - COMMAND / EX MODE
  - VISUAL MODE

# NORMAL Mode

- Normal mode is when you:
  - Don't want to insert or replace text
  - Want to move around your document
  - Delete text
  - Search for text (using /).
  - Can enter another mode
- For example
  - Movement: hjkl, 3w,
  - Deletion: dw, 3x
- You get into normal mode by pressing ESC.

# INSERT Mode

- Insert mode is when you:
  - Want to type some text into your file.
- To get into INSERT mode you can use the following options from NORMAL mode:
  - a - append text after the cursor
  - i - insert text at the cursor
  - o- opens a new line after the present one
  - O- opens a new line in between the current line and the line immediately above it.
  - INSERT KEY - toggles between INSERT mode and REPLACE mode.

# REPLACE Mode

- Replace mode is when you:
  - Want to replace lots of text into your file.
- To get into REPLACE mode you press R (capital r)!
- To be honest, you'll probably not use Replace mode that much.

# COMMAND Mode

- Command mode is when you:
  - Want to set a variable for the environment: eg: `:set syntax=on`, `:set number` (nonnumber sets it off).
  - Want to execute a shell statement (`:!` ), or just `:shell`
  - Search and replace like SED: `%s/old/new/g` (g is global, c=confirm).
- You can enter COMMAND mode from NORMAL mode (ESC), then entering the colon symbol (:).

# VISUAL Mode

- Visual mode is when you:
  - Want to work on large areas of text.
  - Replace large blocks of text.
- It's like selecting text with the mouse.
- To enter Visual mode, press `v` from NORMAL mode.
- To work on a block of text, eg a column (VISUAL BLOCK mode), press `Ctrl-V`.

# Moving about

- Use the 'Home' row to move about much faster than the arrow keys!
- See how your hand sits naturally on the row of keys and allows you to move quickly onto command or insert mode.
  - h = left
  - j = down
  - k = up
  - l = right
- To move forward a word, use w. To move 3 words, use 3w.
- To move backward a word, use b.
- To navigate between { (braces in code), use %



# Moving about (continued)

- To find a character and move to it, use f, eg: fn to move to the first occurrence of n. To move to the 3rd occurrence of n, use 3fn.
- To go to the end of a line, use \$
- To go to the start of a line, use ^
- To search for a string of text use /
- CTRL-G - show a status bar, showing what line you are on etc.

# Deleting characters and words

- To delete a single char, use `x`. To delete 3 chars use `3x`.
- To delete a word, use `dw`. To delete 3 words, use `3dw`.
- To delete an entire line enter `dd`. To delete 3 lines, enter `3dd`.

# Replay and Undo

- Use `u` to undo.
- Press `.` to replay the last operation (eg: `dd`)
  - You can combine this with a number (`3.`) to replay the same operation 3 times.

# Cut, Paste, Copy

- **CUT:** As stated before `dd` deletes a line. `dw` deletes a word.
  - What I didn't say is it actually 'Cuts' it like `Ctrl+X` does in MS Office.
  - `x` cuts a character.
- **PASTE:**
  - To paste it at the cursor, simply use `p` (regardless of whether it's a line, a word or a character).
  - `P` pastes before before the cursor.
- **COPY:** `y` copies (yanks) a line or a word:
  - `yw` (word) `yy` (line)

# Joining

- Use J to join the line below the current line to the current line. Eg:
- BEFORE:
  - Stacy's Mom
  - Has got it going on
- AFTER:
  - Stacy's Mom Has got it going on

# COMMAND Mode Tips

- `:wq` - write the file and quit
- `:q!` - quit immediately, do not save etc.
- `:q` - quit if no changes have been made
- `:r!ls` - reads the output of `ls` and puts it in your vim buffer.
- `:r!grep "ERROR" file.log` - reads the output of `grep` into your vim buffer.
- `:10,15 !xxd` - uses the `xxd` tool to convert the contents of line 10-15 into hex

# COMMAND Mode Tips (Continued)

- `:split <filename>`
- `:new`
- `:wincmd j / k` or `:bn :bp`
- `:syntax on` (`:set syntax=on` for `.vimrc`)
- `:set background=dark`
- `:set number` - show line numbers

# INSERT Mode Tips

- CTRL-O allows you to do normal mode commands whilst in INSERT, but it's not that useful
- CTRL-W deletes the previous word.
- CTRL-u deleted the line.
- CTRL-n invokes auto-completion of words. Very handy for developing code.



# Extras

Vim config settings are made in `.vimrc` or if you use graphical client, `.gvimrc`

There are thousands of plugins for ViM. Here are just a few I like/use:

<https://github.com/shawncplus/phpcomplete.vim> <-- omnicomplete plugin

SuperTab - <https://github.com/ervandew/supertab.git>

YouCompleteMe - <https://github.com/Valloric/YouCompleteMe.git>

(<https://valloric.github.io/YouCompleteMe/#ubuntu-linux-x64>)

<https://github.com/StanAngeloff/php.vim> <-- php syntax highlighting

pathogen to install vim plugin bundles

phpcomplete.vim bundle

Turn your Vim into a modern IDE: <https://www.safaribooksonline.com/blog/2014/11/23/way-vim-ide/>