

# introductory vim

# Why vim?

`vim` stands for “VI Improved.” `vi` is the predecessor of `vim`.

Why would you possibly want to use a random, esoteric text editor like `vim`? Here are some reasons:

- `vim` takes little time to start up (unlike `emacs`) and runs in a shell, so it’s great for quick editing jobs and working remotely.
- `vim` lets you switch from edit mode to insertion mode (more on these later) which means that you can cut, copy, paste, insert, edit, search, replace, open cans, and do your dishes all without having to touch the mouse, let alone the control key.
- Many people have found that frequently switching from the mouse to the keyboard in order to edit text causes injury to their wrists and hands over a long period of time. In `vim` your hands never need to leave the keyboard.

- Sometimes it's all you have to use.

## Edit and Insert modes

In vim, you can switch between two different modes generally known as “edit” and “insert.”

**Insert mode** is just like your average text editor. You type stuff and it shows up on the screen. You are *inserting* text into your document.

**In edit mode** all of those keys you were using before to type your lovely code are now used for editing the text in your buffer. For example, if you type “j”, your cursor will move down a line. If you type “dd” you will delete the line your cursor is currently sitting on.

# Why Modal Editing?

vim's brand of bi-modal editing is favored by many because it lets the user rapidly edit their code. It takes a bit of getting used to, but once they do, many people find it significantly increases their productivity.

## Moving around

h move the cursor to the left  
j move the cursor down  
k move the cursor up  
l move the cursor to the right  
w move forward one word  
W move forward one chunk of non-whitespace  
H move to the first line in the current screen  
M move to the middle line in the current screen  
L move to the bottom line in the current screen  
<n> g move to the nth line of the file  
G move to the bottom line of the file  
g move to the top line of the file  
0 move to the beginning of the current line  
\$ move to the end of the current line  
ctrl-u move up a page  
ctrl-d move down a page

Most of the commands in edit mode are mnemonics to help you remember what they do.

## Editing your text

`dw` delete a word (like cutting)

`yw` yank a word (yanking is like copying)

`dW` delete a chunk of non-whitespace

`yW` yank a chunk of non-whitespace

`dd` delete a line

`yy` yank a line

`r<char>` replace the current character with `<char>`

`y$` yank from where you are to the end of the current line

`x` delete the current character

`p` paste in whatever you just deleted/yanked

## Saving, opening, and quitting

Jokes are told of the poor sap who found him/herself stuck in vim and unable to quit. To spare you the embarrassment, we offer you the following:

```
:w write (save) your file
:w <file>write your file to <file>
:q! quit without saving your file
ZZ save your file and quit
:e <file>open <file>
```

## Getting More Help

Vim is kind of an aquired taste. It'll take you a while to learn it but once you do you'll never want to go back.

# vilearn

Also, you may want to check out the book *Learning the vi Editor* from O'Reilly. Buy it at [bookpool.com](http://bookpool.com).

# Start using vi!

You will now use `vi` and `vim` to make the necessary edits to your dotfiles that will make you and `vim` happy.

# 1. `.vim` and `.vimrc`

First, we'll copy over the `vim` initialization files. This actually just uses shell commands.

```
cd
cp /pro/consult/pub/sample.vimrc .vimrc
cp -r /pro/consult/pub/sample.vim .vim
```

## 2. Aliases for vim

```
vi .alias
G
o
alias vim '/pro/gnome/bin/vim' <Esc>
ZZ
```

If you don't have a `.alias` file, uhm, see a consultant for help.

## 3. Colors

```
vi .Xdefaults
```

type `/Rxvt <CR>`, and see if anything is found. If something comes up, see one of us for help.

Assuming nothing comes up, type `G` to go to the end of the file, `o`, then `Escape` to insert a blank line. Then, to add our colors, type

```
:r /pro/consult/pub/vimshellcolors.Xdefaults
```

and press `ZZ` to save.

## 4. Repeat Rate

```
vi .login  
/openwin <CR>  
n  
A  
-ar1 200 -ar2 30 -dev /dev/fb defdepth 24 <Esc>  
ZZ
```

# Testing

Log off, then log on again.

Then type `rxvt` in a shell to start a new shell.

Type `which vim` and make sure it says `vim: aliased to /pro/gnome/bin/vim`.

Type `vim`, and you'll see a funny startup screen.

Type `:e FunnyClass.java` to see a really funny java program.

Type `ZZ` when you're bored.