Vim
(an overview)

Wilson Zhao

A simple outline

- What
- Why
- How

I will try to focus on the big ideas to get you started

What is VIM?

- Vim is a **screen-oriented** text editor.
- Vim is **old**
  - A clone of “Vi” which was released in the “1970s”
  - But is one of the most used text editors around today!

Vim...seems wack

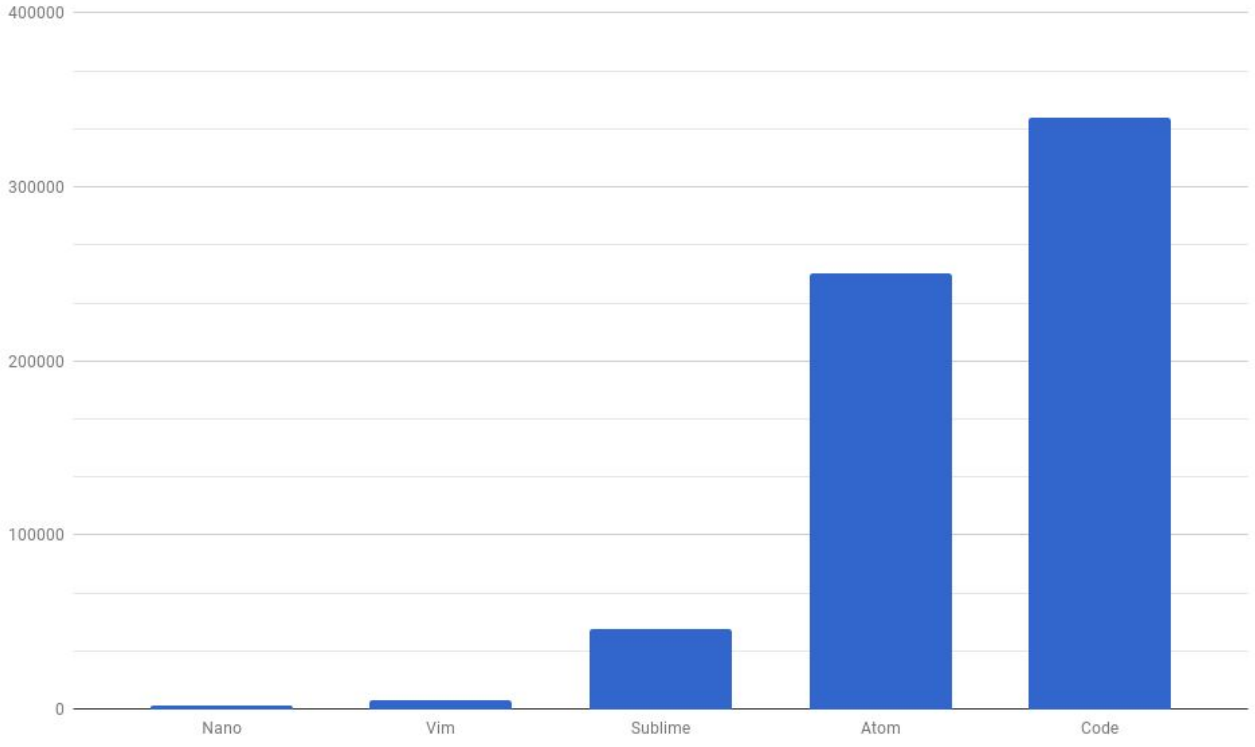
- A couple facts straight
  - No mouse
  - No ‘GUI’
  - Handicapped?

Vim...why bother?

- Heavily used for
  - System administration
  - Programming
  - Working with HTML, LaTeX, or other markup languages
  - Heavy editing of plain text files
- It's **EVERYWHERE.**
  - **And it's on andrew!**

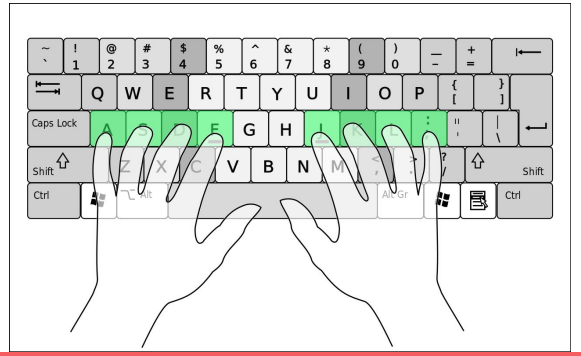
Vim...why bother?

- Lightweight



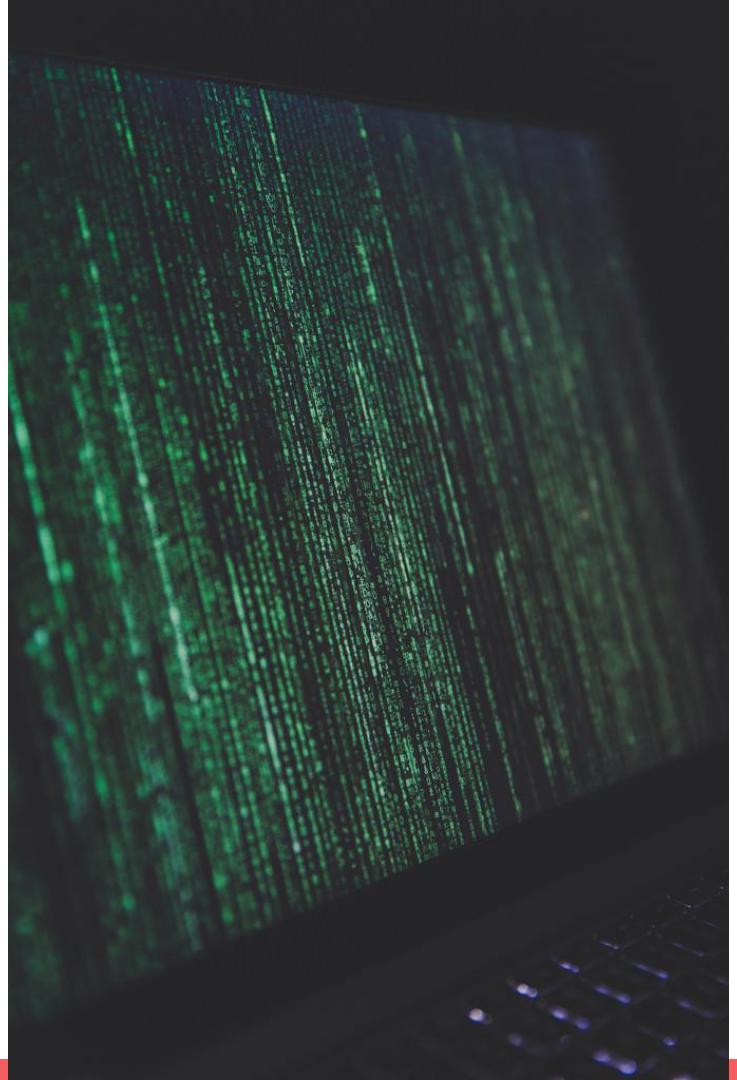
Vim...why bother?

- Ergonomic and Efficient!
- You'll be coding for the next 4 years...and possible the rest of your life...
  - Carpal Tunnel is real
- Fingers will ALWAYS stay near the home row.
  - Decreases latency



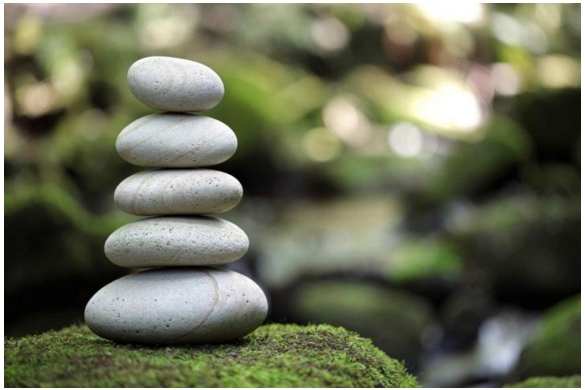
Vim...why bother?

- Hacker Cred
- Hackertyper.com but irl



How: The philosophy of Vim

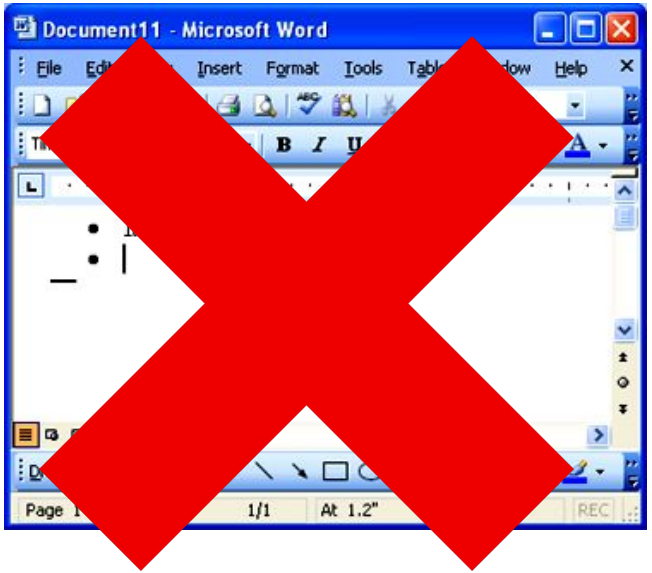
- Vim is a “**high-level**” text editor
- We want to be terse and accurate.
- Writing is filled with repetitive tasks
- We want the most efficient and ergonomic way to accomplish our goals



Vim: A higher-level text editor

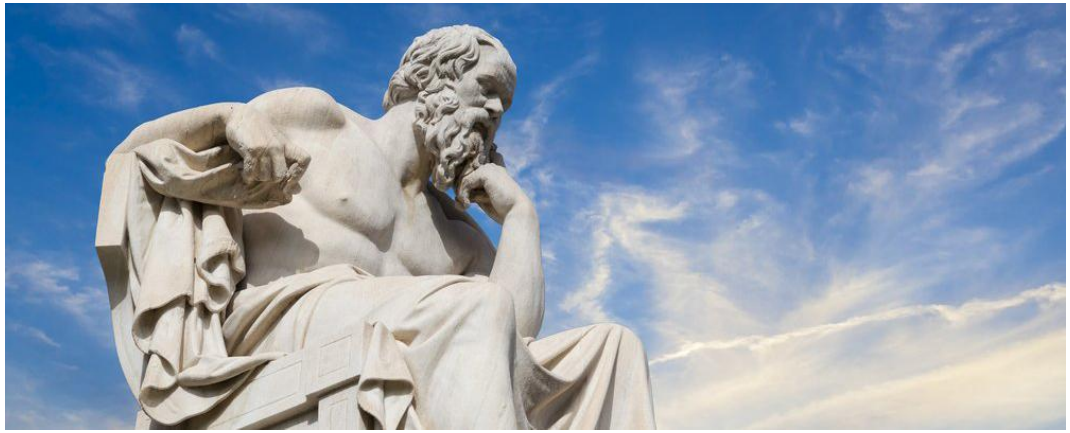
- In Microsoft Word, you manually to move the cursor a certain amount
- You tell Vim **exactly** what to do and it will do it
  - The text-editor will do the low-level things for you

Ex: dog -> cat



The philosophy of Vim

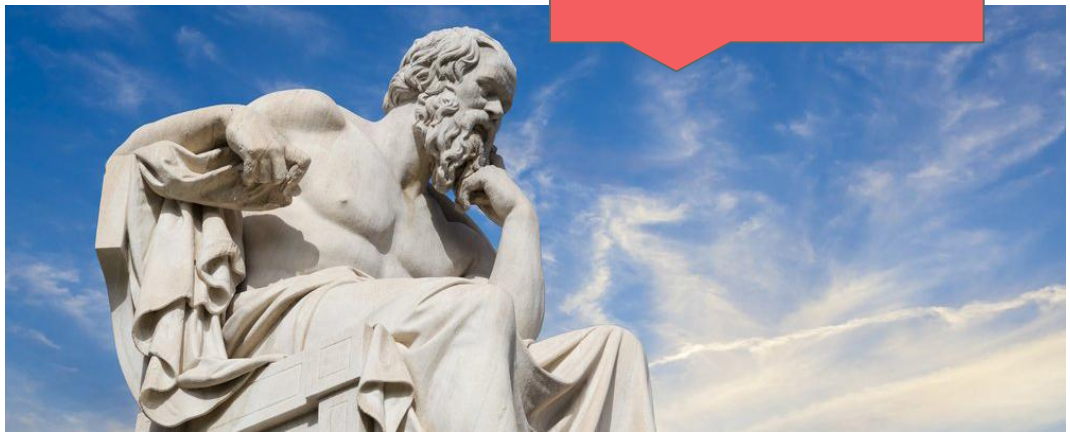
1. Write
2. Locate
3. Edit
4. ???
5. Profit



The philosophy of Vim

!!!!!!!

1. Write
2. Locate
3. Edit
4. REPEAT
5. Profit



Leads to amazing efficiency/ergonomic gains!

Demoing the “Vim philosophy”

I’m writing an apology letter to Jacobo.

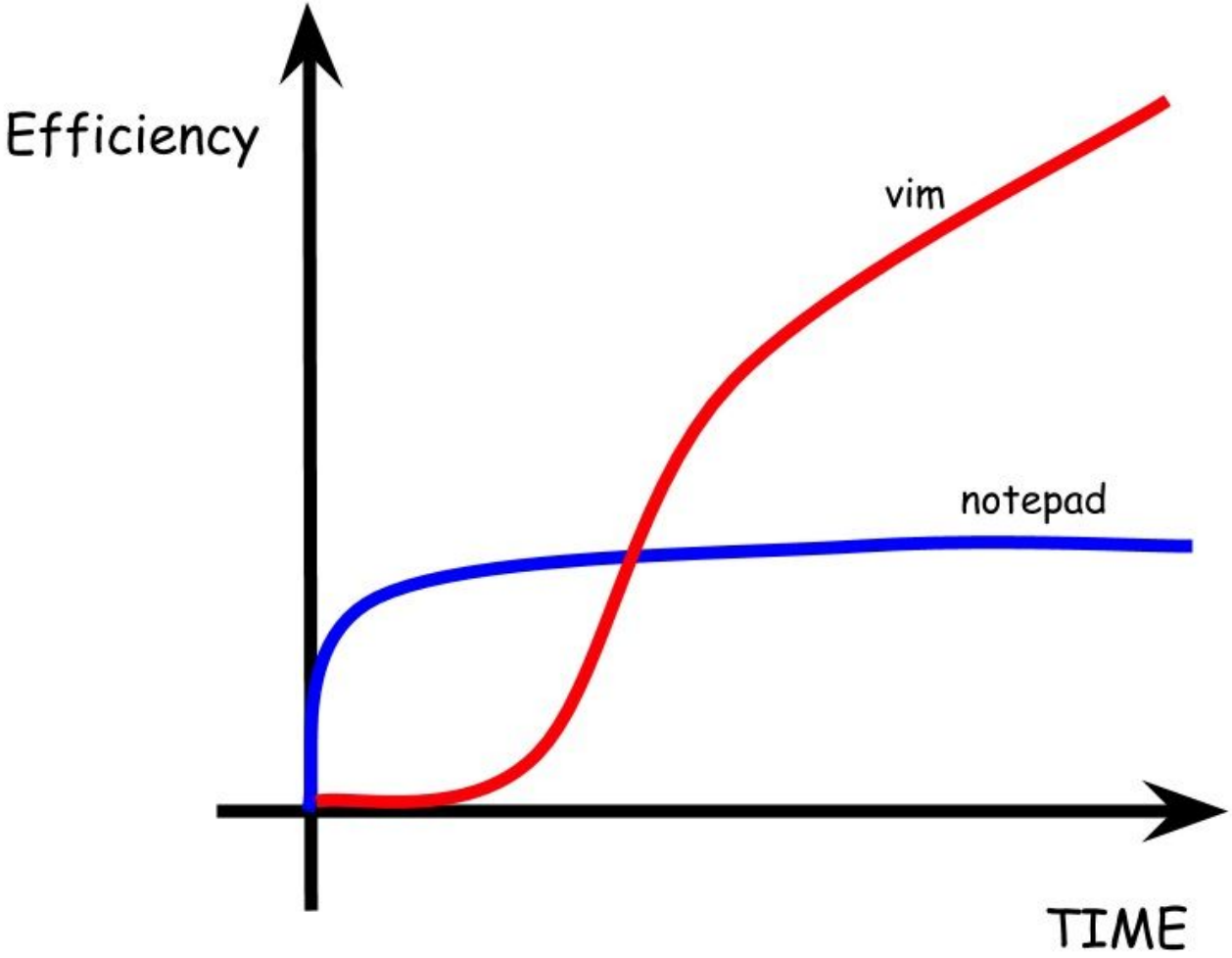
There’s a lot of errors.

I’m going to edit this...**blindfolded.**

Learning to Walk

This presentation just covers enough to get you started

- Don’t memorize! Too much content
- Think: How can I do this faster?
- Practice: Best way to learn!
- **Repeat.**


Efficiency
vim
notepad
TIME

Vim is a MODAL editor

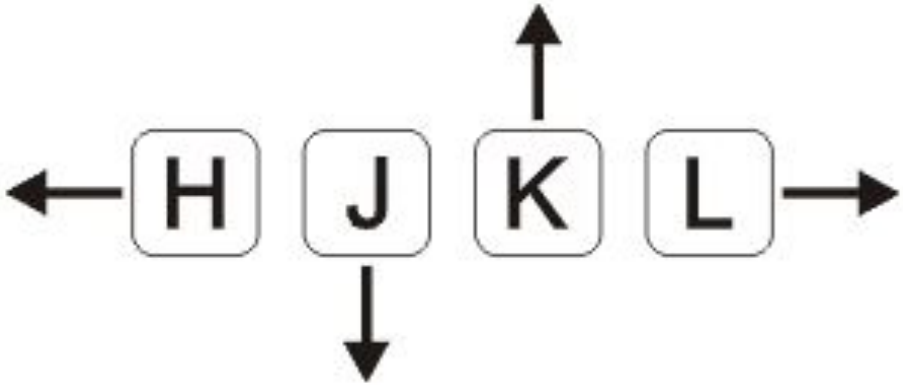
- **Insertion (i)**
  - The “Drafting mode”
  - This is for actually writing text
- **Normal (Esc)**
  - The “Editing mode,” the default mode
  - Navigation and Revising
- Visual Mode (v)
  - Hilighting blocks of text for copy-pasting/deleting
- Command/Ex mode (:)
  - “The “Filewide mode”
  - For filewide operations like searching, saving, etc.
  - Other stuff I don’t even understand yet

Normal Mode: Navigating

So how do we move around without a mouse?

Letterwise:

h j k l - arrow keys.



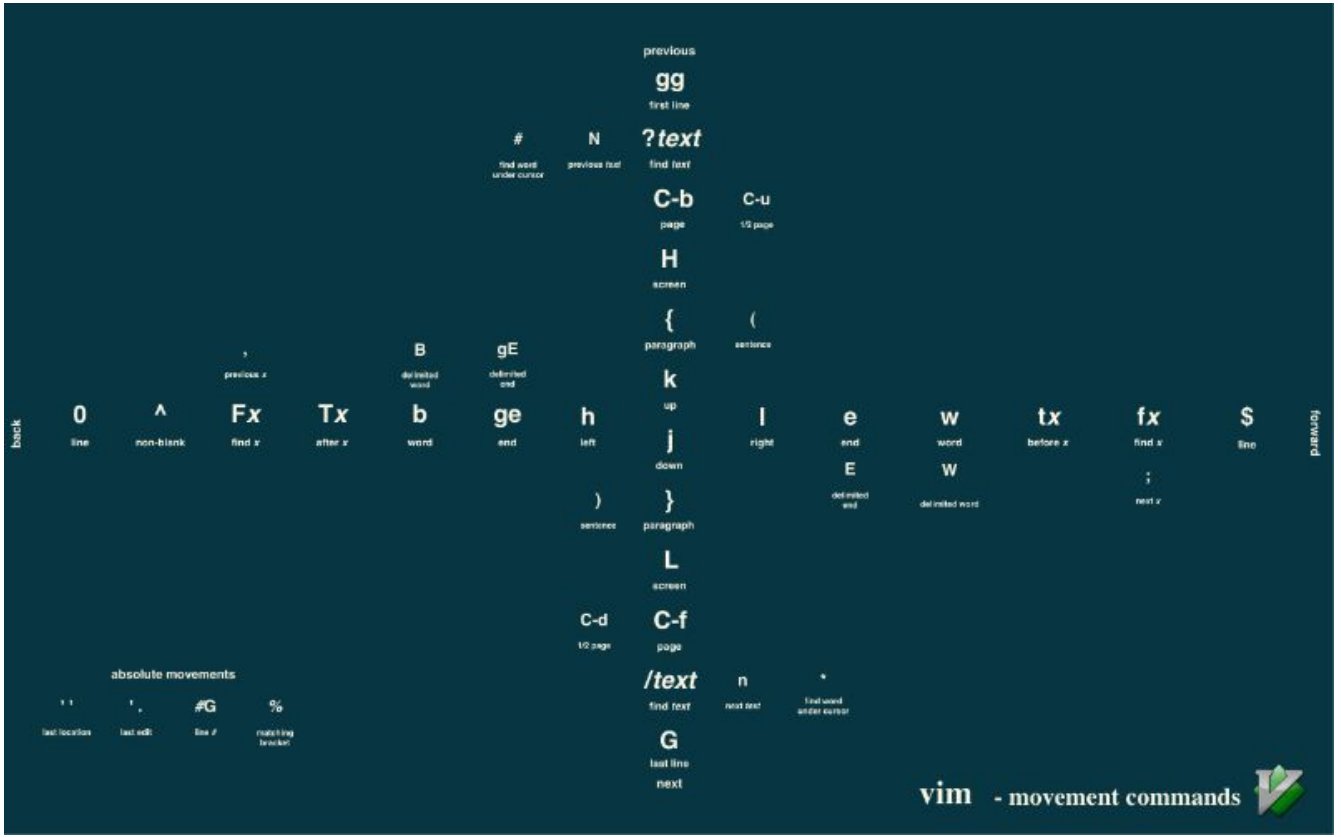
Normal Mode: Navigating

Inline

- find <next occurrence of letter>
- to <next occurrence of letter>
- 0 - Beginning of line
- $ - End of line

Filewide

- /<search text><enter>
- Go to the end of the file
  - ggo to the beginning
- next occurence of search text


previous
gg
first line
find word under cursor
N previous /text
?text find /text
C-b page
C-u 1/2 page
H screen
{ paragraph
(sentence
, previous x
B delimited word
gE delimited end
k up
back
0 line
^ non-blank
Fx find x
Tx after x
b word
ge end
h left
j down
l right
e end
w word
tx before x
fx find x
$ line
forward
E delimited end
W delimited word
; next x
) sentence
} paragraph
L screen
C-d 1/2 page
C-f page
/text find /text
n next /text
* find word under cursor
G last line
next
absolute movements
'' last location
'. last edit
#G line #
% matching bracket
vim - movement commands

Normal Mode: Editing

Vim has its own intuitive “language.” It has verbs and objects, and the commands are similar to the english counterparts. You can create “clauses” to manipulate the text

<COUNT> <VERB> <TEXT OBJECT>

2 delete word

deletes two words from the current cursor position

Doing stuff (memorize this later)

Basic Commands

- yank
- delete
- paste

Objects

- word
- Repeat the command (the entire line)
- $ (cursor to end of line)

Going to insert mode from normal mode

- What's the normal way?
- change - Deletes the following text object and enter insert mode
- Append - Jump to the end of the line and enter insert mode
- open - "Opens" a new line and enter insert mode

The most important command

1. Locating

   /<pattern><enter>

2. **Repeating**

   .

Exiting Vim

How do you generate a random string? Put a web designer in front of VIM and tell him to save and exit.

Exiting Vim

:w to save

:q to quit

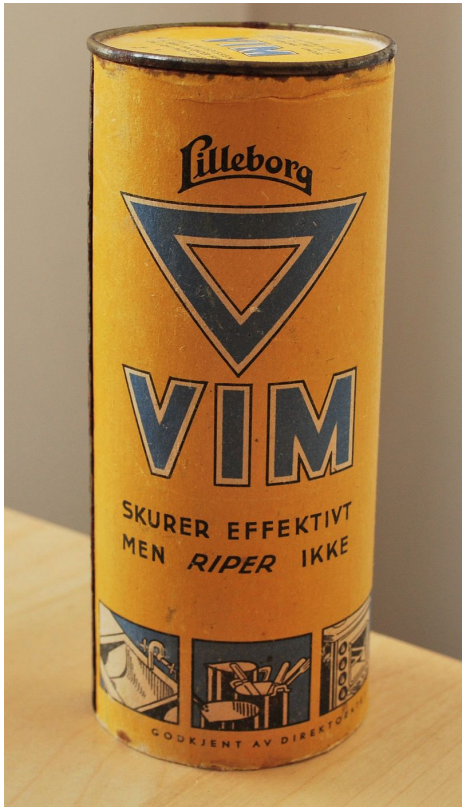
.

Demo!

Go slowly this time

Further resources/reading

- vimtutor
- Practical Vim - Drew Neil



Self-sufficiency

Practice Practice Practice!

Google is your friend!

Your friends are your friends!

Check out the Vim Wiki!

Similar to man pages, you can use

:help <topic>

A whole new world

- Registers
- Macros
- Interfacing with STDIN/STDOUT
- Customizing Vim to become an IDE
- Codecompletion, folding, markers, buffers, windows, etc….
- Using vim...in emacs and other text editors

LPT: Lab pro tips

- If you get an error message that says something like “Vundle” just press enter to continue
- If you get a “merge commit” screen on pull, then type “:wq!” and press enter to exit vim and complete the merge
  - Vim itself is used for other command tools such as Git
- Treat both words “More dust” as dust
- The last instruction on sliding should be (I always practice in grassy fields).
- “Written by yours truly” (no quotes) should both be directly ABOVE and BELOW those lines
- If you screw up on a file, `git checkout <filename>` to reset