

Vim

(an overview)

Jack Cameron

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**
 - An improved 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' - Graphic User Interface
 - Handicapped?



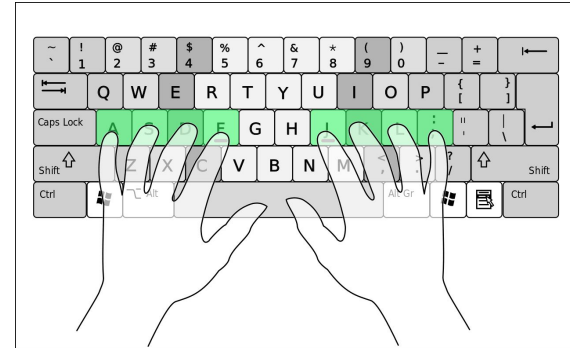
Vim...why bother?

- 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?

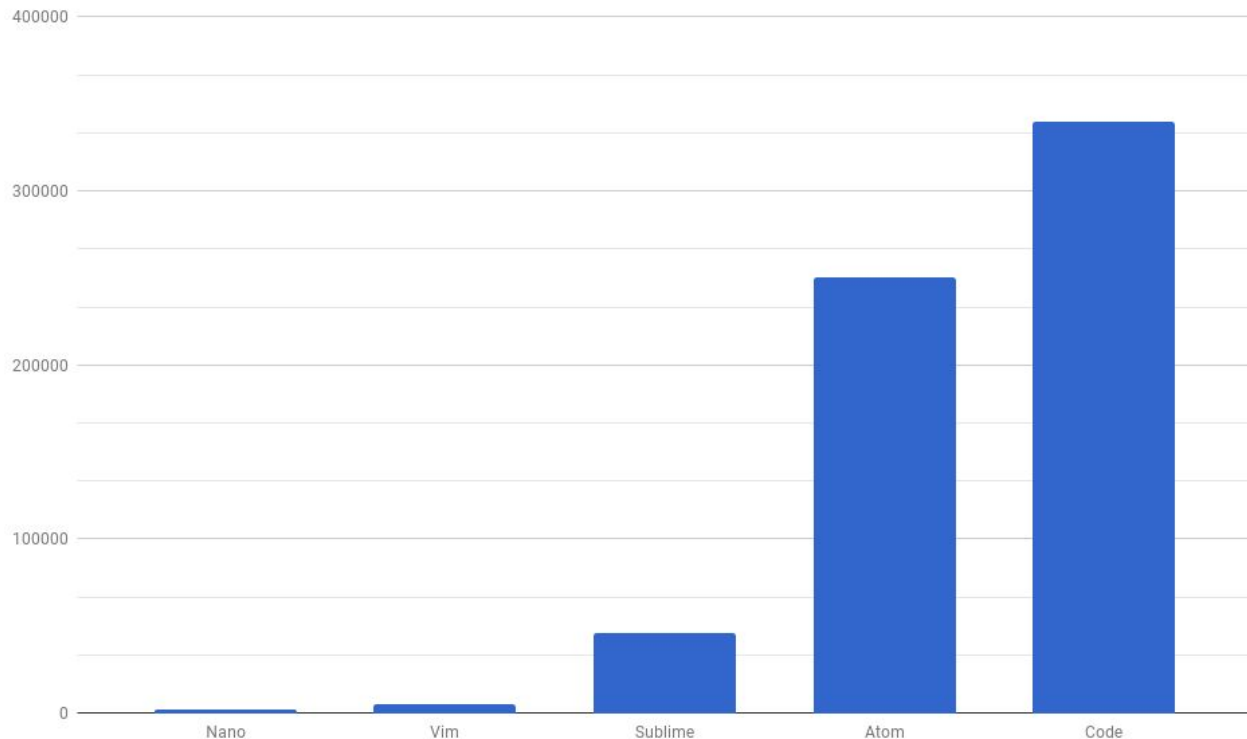
- Ergonomic and Efficient!
- You'll be coding for the next 4 years...and possibly the rest of your life...
 - Carpal Tunnel is real
- Fingers will ALWAYS stay near the home row.
 - Decreases latency

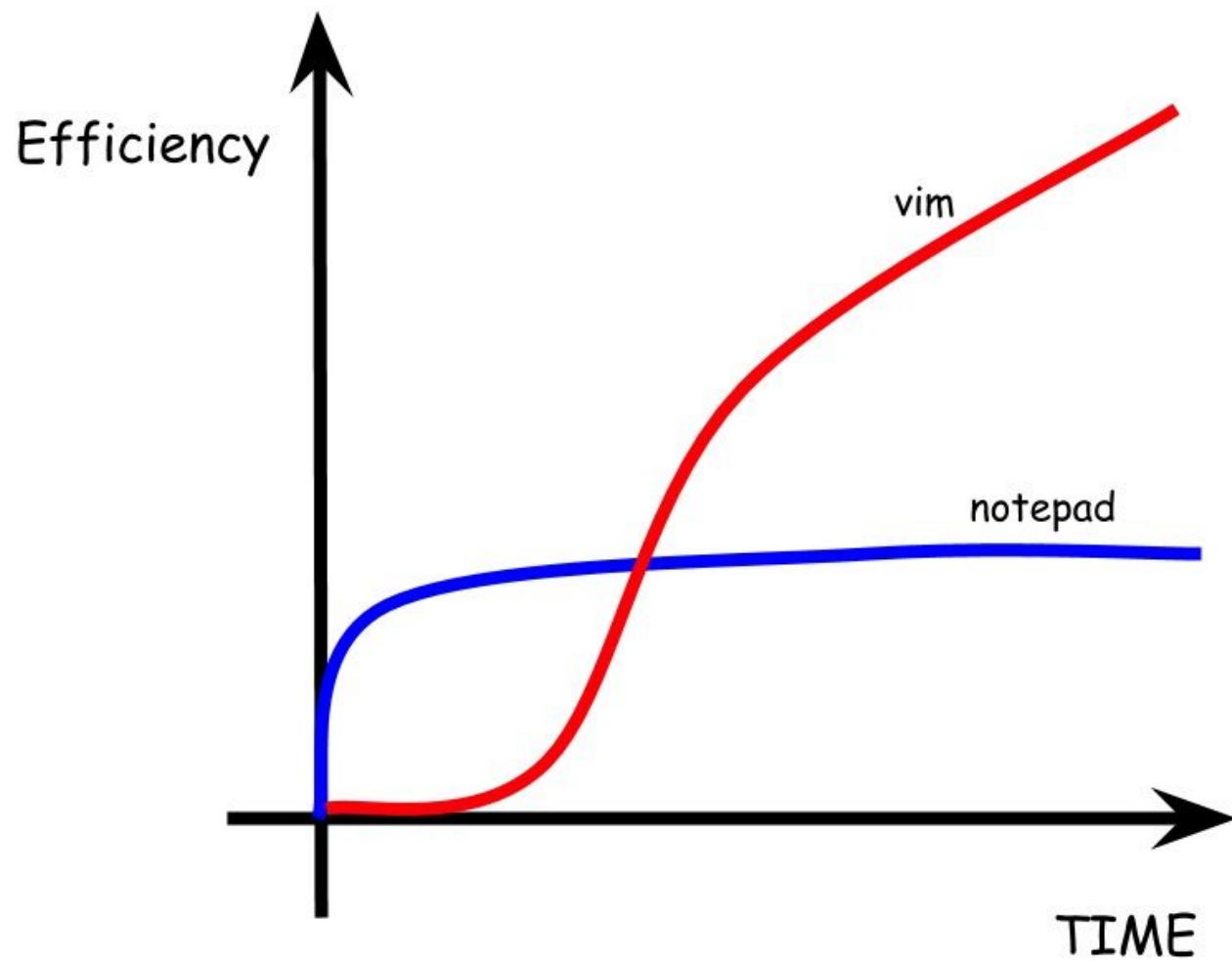


Vim...why bother?

- Lightweight

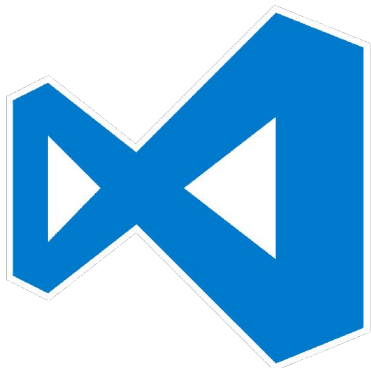
Battery Usage





Vim...why bother?

- Everything you learn expands
- Vim keybindings are everywhere



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



Learning to Walk

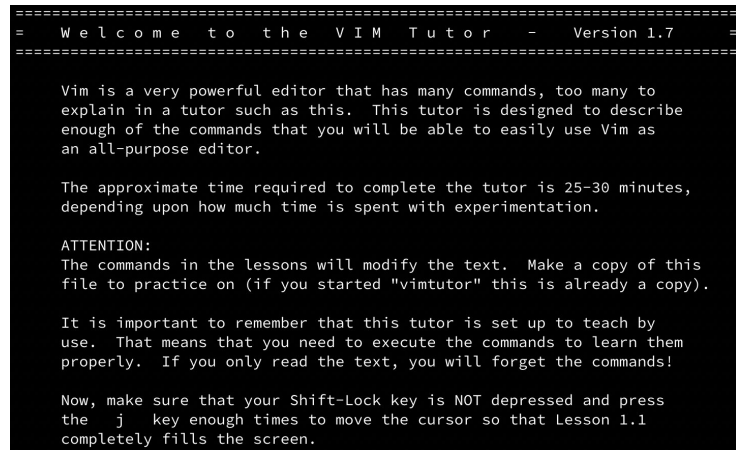
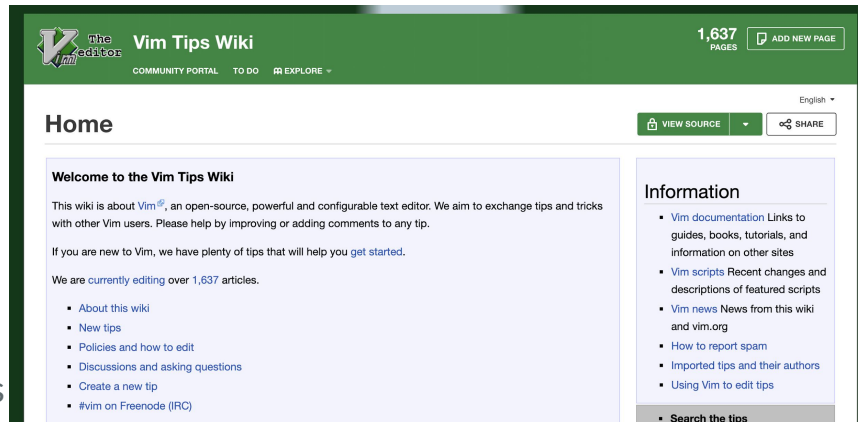
This presentation just covers enough to get you started

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



Teach Yourself!

- There are great resources out there to help you learn!
 - 50 + years of documentation to guide you
- Vim Wiki
 - Great online resource for answering questions
- Vimtutor
 - Installed on almost all machines with vim (practically everything)
 - Just type vimtutor into the terminal
 - The best way to learn!
- :help <cmmd>
 - Internal vim help pages

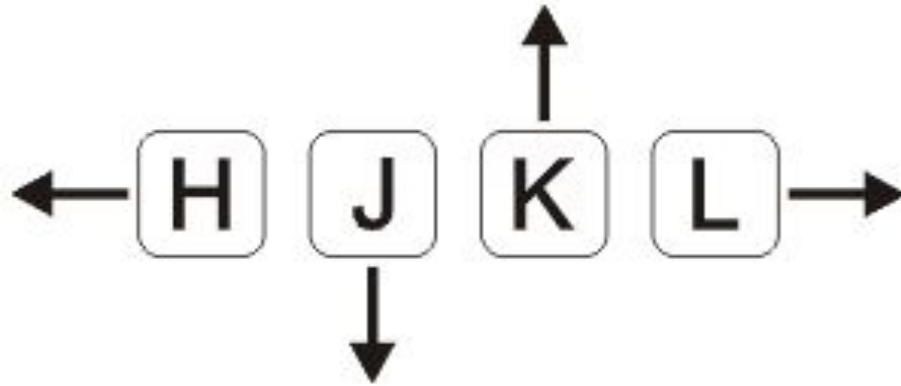


Normal Mode: Navigating

So how do we move around without a mouse?

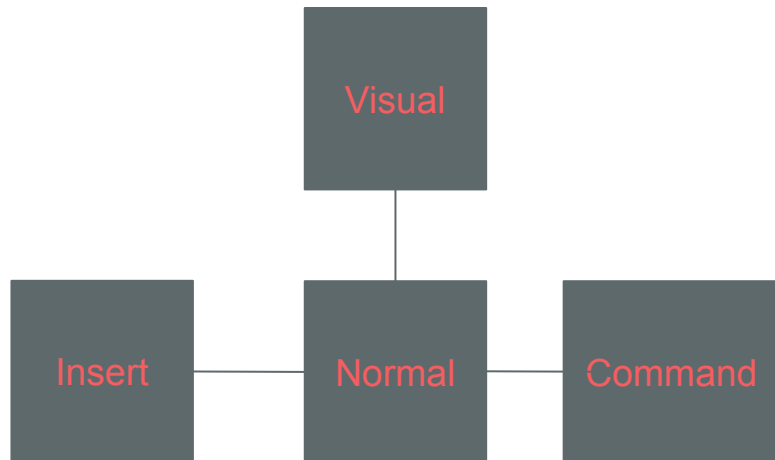
Letterwise:

h j k l - arrow keys.



Vim is a MODAL editor

- **Insertion (i)**
 - The “Drafting mode”
 - This is for actually writing text
- **Normal (Esc ctrl-[])**
 - The “Editing mode,” the default mode
 - Navigation and Revising
- **Visual Mode (v)**
 - Highlighting 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



The Parallel Shift

- Normally there is a pattern to vim commands

f <char>	Look forwards for char	F <char>	Look backwards for char
o	Open new line below	O	Open new line above
i	Insert before cursor	I	Insert at beginning of line
a	Insert after cursor	A	Insert at end of line

NOT
ALWAYS
TRUE!!!!

Normal Mode: Navigating

Inline

- **f**ind <next occurrence of letter>
 - **F**ind <previous>
- **w** <next **w**ord>
 - **b** <**b**ack a word>
- **O** - Beginning of line
- **\$** - End of line

Filewide

- **/**<search text><enter>
 - **?** for backwards
- **G**o to the end of the file
 - **g**go to the beginning
- **n**ext occurrence of search text
 - **N** - previous

back

O

line

^

non-blank

Fx

find x

Tx

after x

b

word

ge

end

h

left

k

up

j

down

l

right

e

end

E

delimited
end

w

word

W

delimited word

tx

before x

fx

find x

;

next x

\$

line

forward

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

B

delimited
word

gE

delimited
end

}

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 **d**delete **w**ord

deletes two words from the current cursor
position

Doing stuff (memorize this later)

Basic Commands

- **y**ank
- **d**delete
 - **d**delete line
- **p**aste - below
 - **P**aste - above
- **u**ndo
- **ctrl-r**edo

Objects

- **w**ord
- Repeat the command (the entire line)
- **\$** (cursor to end of line)

Going to insert mode from normal mode

- What's the normal way?
- **c**hange - Deletes the following text object and enter insert mode
- **A**ppend - Jump to the end of the line and enter insert mode
- **o**pen - "Opens" a new line and enter insert mode

The most important command

1. Locating

`/<pattern><enter>`

`:<line number>`

2. Repeating

A Very Useful Command

- From command mode (:)
- `%s:<find>:<replace>;g`
 - The `%` searches the whole file
 - The `s` is the command which is being run (search)
 - The `g` is a flag which replaces every instance

Exiting Vim



I Am Developer

@iamdeveloper



Following

I've been using Vim for about 2 years now, mostly because I can't figure out how to exit it.

↩ Reply ↻ Retweet ★ Favorite ... More

RETWEETS

4,846

FAVORITES

2,105



4:56 AM · 18 Feb 2014



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

`:wq` to save and quit

.

Once you quit you lose your undo information. To learn how to avoid that come to the extratation this weekend!

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