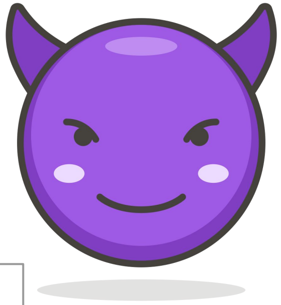
Scope’s first seminar :

Intro to Vim :)

by Jeffrey Hertzog :)

What is Vim?



- Vim is a “modal” text editor

Normal : (command mode)

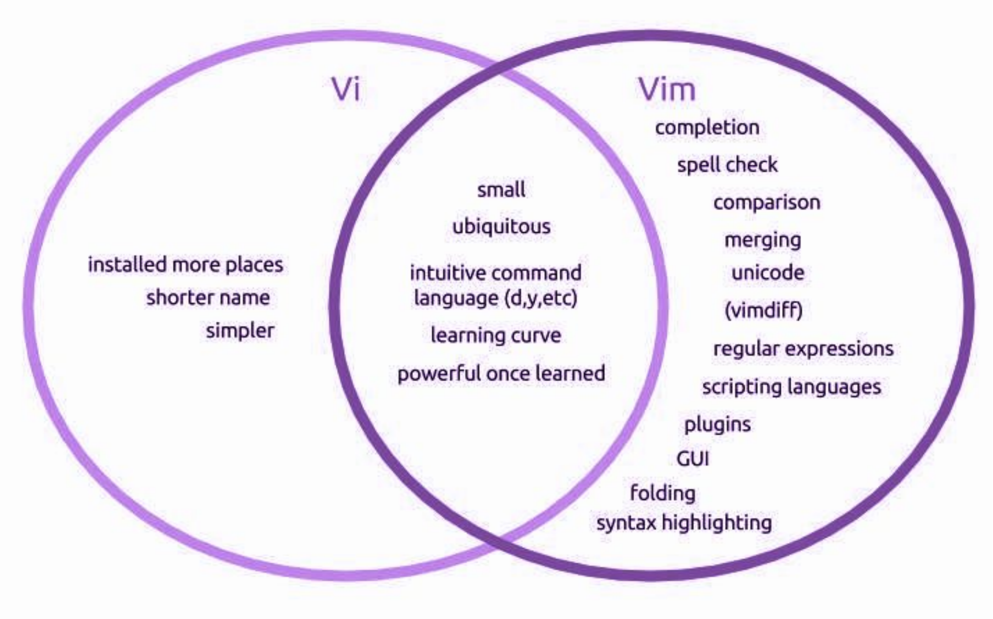
Insert : (typing mode)

Visual : (highlighting mode)

Origins

1. ed
2. em (editor for mortals)
3. ex
4. vi (extension of ex)
5. Stevie (ST editor for VI enthusiast)

6. vim (vi - improved!) ★



Vi

installed more places
shorter name
simpler

small
ubiquitous
intuitive command language (d,y,etc)
learning curve
powerful once learned

Vim

completion
spell check
comparison
merging
unicode
(vimdiff)
regular expressions
scripting languages
plugins
GUI
folding
syntax highlighting

Why should I use vim over modern text editors?

Efficient and Powerful

- Switching between modes and using commands allow the user to edit and move around text extremely fast, and without the use of the mouse.
- Very visually simple, no bloated UI elements to get in your way while editing

Ubiquity

- it's already everywhere
- can have all your configurations and plugins one file away (.vimrc)

```
[jeffrey:~/Documents/notes/leetcode/easy/strings] $ 
```

:set number

Exiting Vim

I Am Devloper
@iamdevloper

Always enjoy seeing someone trying to exit Vim for the first time.

Lady Gaga @ladygaga
AAAAAAAAAAAAAHHHHHRHRGRGRGRRRGUR BHJB EORWPSOJWPJORGWOIRGWSGODEWPGOHE PW09GJEDPOKSD!!!!!!!!!!!!!!! 0924QU8T63095JRGHWPE09UJ0PWHRGW

12:37 PM · 18 Sep 18

157 Retweets 386 Likes

:q quit without saving,
(error if you have unsaved changes)

:q! force close, disposes of unsaved changes

:wq save and quit

ZZ another way to say wq

Basic Movement

- use these keys to move around!

h	move left
j	move down
k	move up
l	move right

(resist the urge to use the arrow keys)

Find, n, and N

/{string} where string is the pattern you want to find

n move to next instance

N move to prev. instance

Writing text with insert mode

i I

i

go into insert mode AT cursor location

I

goes into insert mode at BEGINNING of the line

a A

a

go into insert mode AFTER cursor location

A

goes into insert mode at END of the line

o O

o

open a new line in insert mode BELOW current

O

open a new line in insert mode ABOVE current

Command Syntax

(verb + noun)

d : delete (action)
w : a word (object)

d + **w** = deletes a word

- what happens if we include #’s?

Basic Verbs

d → deletes a noun

x → deletes a character

c → “change” (deletes + insert mode)

y → “yank” copies a noun

p → pastes whatever was d/y last

Basic Nouns

w → forward one word

e → to the end of the next word

$ → until the end of the line

0 → to the beginning of the line

f{char} → to the next instance of char

Important! ⭐

u → undo last command

ctrl r → redo last command

. → repeat last command

- These are also considered verbs!

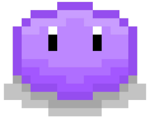
Try it!

d4j	deletes four lines
4p	pastes last deletion four times
c$	deletes to eol, and insert mode
3u	undoes the last three changes
ea	insert mode at the end of current word

More Movement

- gotta go fast :)

gg → jumps to first line
G → jumps to last line
:n → :5, jumps to line 5
g; → jumps back to place of last edit
{ } → jumps to prev/next paragraph



More Nouns

iw → modify the entire word
it → forward three words
i” → inside quotations
ip → inside paragraph
as → a sentence (up to .)

(remember: these can all be used with numbers to do many operations at once)

Visual Mode

v →

- enters “visual mode” when you move around you will select text to modify

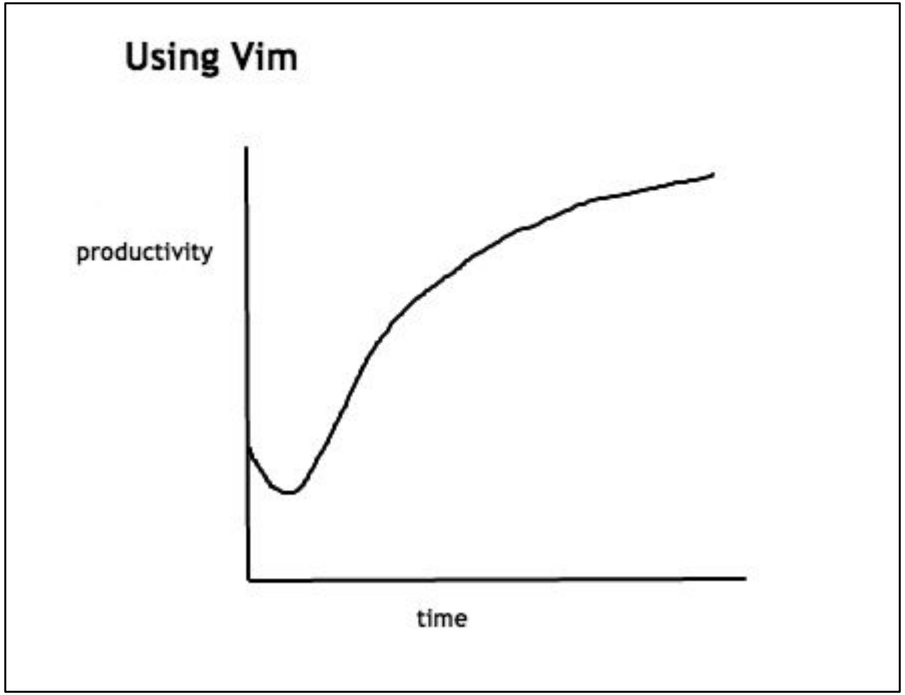
shift v→

- enters “visual line”
- same as visual just selects by entire line

- press v or shift v again to exit visual mode

Learning

- If you want to learn more and practice vim after this seminar, consider using vimtutor!



What’s my favorite thing about vim?

it’s hard

there’s so much to learn and so many unique ways to quickly solve problems
that would take exponentially longer if done in a “modern” editor

The end :0

Any questions?

http://www.viemu.com/vi-vim-cheat-sheet.gif

hertzog@usc.edu
https://github.com/j-hertzog