Intro to Vim

Hackers @ Berkeley

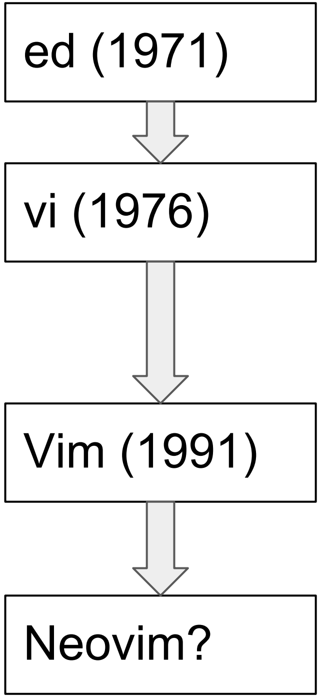
Pandu Rendradjaja
pandu@berkeley.edu

http://tinyurl.com/hellovim

What is Vim?

- Obviously, a text editor. What makes it different?
    - A second-to-none keyboard-based *language* for manipulating and navigating text.
        - *Very fast.*
        - Mouse optional.
    - Extremely extensible and configurable; has a large community with plugins, etc. for every use imaginable. (See also: Emacs.)
    - A practical note: Vi(m) is everywhere. A bit of familiarity comes in handy surprisingly often.

Vi(m) is OLD!



ed (1971)

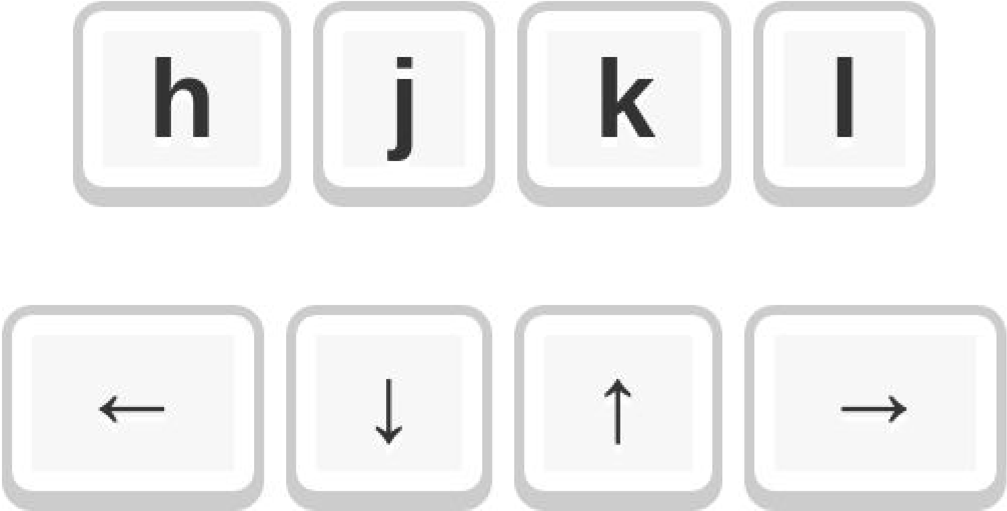
vi (1976)

Vim (1991)

Neovim?

Image shamelessly lifted from: http://hackaday.com/2014/02/08/raspi-powered-adm-3a-dumb-terminal/

http://tinyurl.com/hellovim



h j k l

← ↓ ↑ →

Instructions for exercise 1

- Use the arrow keys or hjkl to move your cursor to below the “Welcome to Intro to Vim!” line.
- Then press i, type “Hello world!”, and then press the Escape key.

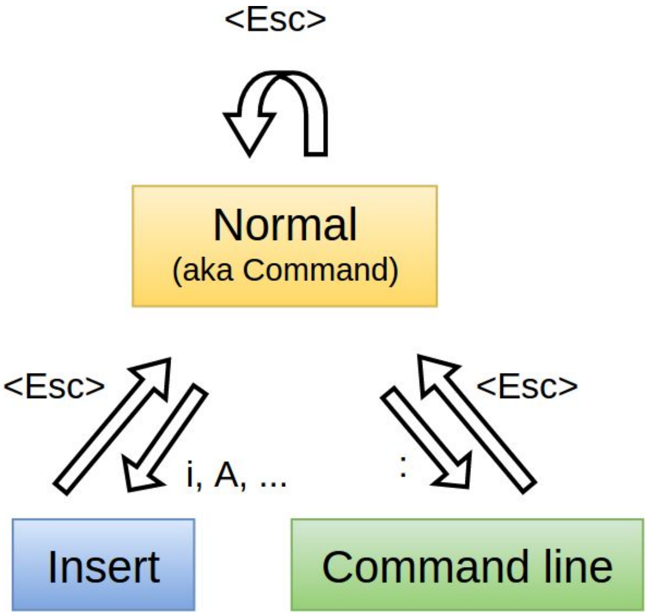
i

Hello, world!

Esc

What was that all about?

- Vim is a **modal** editor.
  - Three main modes.
  - Not as confusing as it seems!
  - Save yourself some headaches: “default” to Normal.



:

set nocompatible

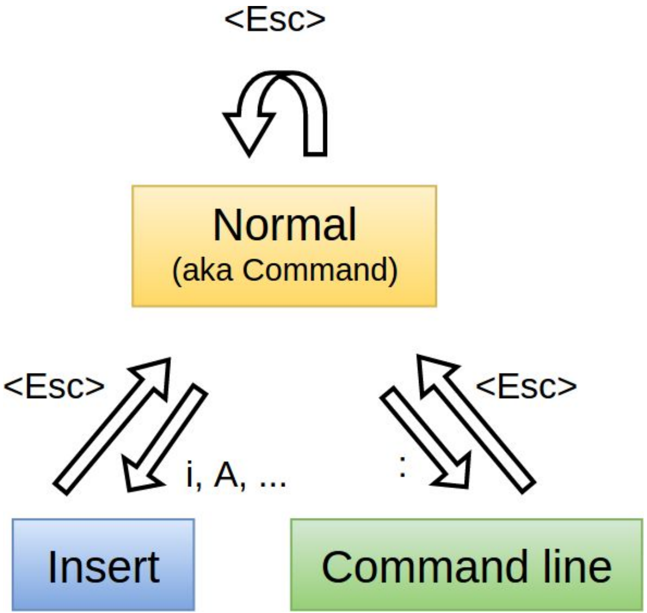
Enter

Colon commands

```
:set nocompatible
```

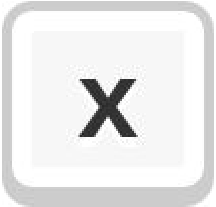
- Change settings
- Save, open and close files
- Compile
- Search and replace
- Access the help
- And so on...

- Turns off backwards-compatibility with vi: saner defaults
- Affects many other settings, e.g.:
  - Showing current mode
  - Multiple undo!
  - And others...



<Esc>
Normal
(aka Command)
<Esc>
i, A, ...
:
<Esc>
Insert
Command line

A


x

Commands for exercises 2-5

`i`	Enter Insert mode before the cursor
`<Esc>`	Return to Normal mode (<Esc> is Vim notation for the Escape key.)
`A`	Enter Insert mode at the end of the current line
`u`	Undo
`x`	Delete the character at the cursor
`dw`	Delete a word
`d%`	Delete a pair of parens/brackets/etc. (and what’s in between)

Instructions for exercises 2-5

- #1: Use the arrow keys or hjkl to move your cursor to below the “Welcome to Intro to Vim!” line. Then press i, type “Hello world!”, and press Escape.
- #2: Add the missing punctuation! Notice that you’ll need to use A instead of i in order to get a question mark at the end of the line.
- #3: Delete the extra characters with x.
- #4: Delete the extra word with dw.
- #5: Delete the parenthesized text by moving your cursor to one of the parentheses and pressing d%.

d w

d %

w

to next word

dw

delete to next word

dw

delete to next word

operator motion

d%

delete to matching paren

operator motion

This is REALLY cool.

Easy-to-remember commands

Other editors

Home

End

Ctrl-Left

Ctrl-Right

Ctrl-Shift-K

Ctrl-T

Ctrl-Del

Ctrl-Backspace

Ctrl-K K

Ctrl-K Backspace

Ctrl-Shift-D

Easy-to-remember commands

Other editors

Home
End
Ctrl-Left
Ctrl-Right
Ctrl-Shift-K
Ctrl-T
Ctrl-Del
Ctrl-Backspace
Ctrl-K K
Ctrl-K Backspace
Ctrl-Shift-D

Vim

	d	y	<	>
w				
b				
0				
$				
/				
{				
}				
%				

Easy-to-remember commands

Other editors

Home
End
Ctrl-Left
Ctrl-Right
Ctrl-Shift-K
Ctrl-T
Ctrl-Del
Ctrl-Backspace
Ctrl-K K
Ctrl-K Backspace
Ctrl-Shift-D

Vim

	d	y	<	>
w	**dw**	**yw**	**<w**	**>w**
b	**db**	**yb**	**<b**	**>b**
0	**d0**	**y0**	**<0**	**>0**
$	**d$**	**y$**	**<$**	**>$**
/	**d/**	**y/**	**</**	**>/**
{	**d{**	**y{**	**<{**	**>{**
}	**d}**	**y}**	**<}**	**>}**
%	**d%**	**y%**	**<%**	**>%**

Easy-to-remember commands

Other editors

Home
End
Ctrl-Left
Ctrl-Right
Ctrl-Shift-K
Ctrl-T
Ctrl-Del
Ctrl-Backspace
Ctrl-K K
Ctrl-K Backspace
Ctrl-Shift-D

Vim

	d	y	<	>
w	**dw**	**yw**	**<w**	**>w**
b	**db**	**yb**	**<b**	**>b**
0	**d0**	**y0**	**<0**	**>0**
$	**d$**	**y$**	**<$**	**>$**
/	**d/**	**y/**	**</**	**>/**
{	**d{**	**y{**	**<{**	**>{**
}	**d}**	**y}**	**<}**	**>}**
%	**d%**	**y%**	**<%**	**>%**

Operators for exercises 6-8

`d`	Delete
`c`	Change (= delete + insert)
`y`	Yank

Yank is called “copy” in modern editors. Use p to “put,” which is Vim’s term for “paste.”

This is not to be confused with “yank” in Emacs, which is like “put”/”paste.”

By the way, “delete” actually corresponds better to modern “cut,” because you can also use p to paste deleted text.

Motions for exercises 6-8

`w e b`	Move by words
`W E B`	Move by WORDs (sequences of non-whitespace characters)
`( )`	Move by sentences
`{ }`	Move by paragraphs
`0 $`	To beginning/end of line
`%`	To matching paren, etc.

Instructions for exercises 6-8

- #6: Experiment with the operators and motions from the previous two slides!
- #7: On each of the six lines, place your cursor on the “o” in “one.” Before performing the indicated motion, guess where your cursor will end up. Then do it, and see if you were right!
- #8: Change the word “because” to “therefore,” fill in the missing code, and copy and paste the long word.

d i '

Text objects

d

operator

w

motion

d

operator

w

motion

i”

text object

d

operator

w

motion (“where to”)

i”

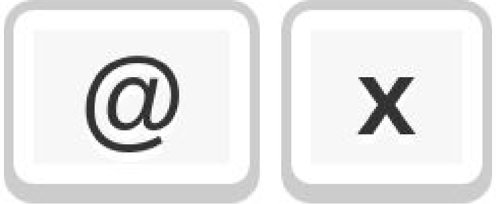
text object (“what”)

Macros

q x

(do any Vim commands here)

q


@
x

The rest of the workshop

- Practical things
- Demo of some other features
- Where to go from here

Save and quit

vimrc

A few more useful things

-
- text
  - normal size
  - TODO ...
-

-
- text
  - normal size
  - TODO ...
-