

# introduction to vi

Paul Cobbaut

# a tutorial

- for newbies
- every slide is incomplete!
- there is much MUCH more to learn
- google

# Getting the software

- `aptitude install vim`
- `yum install vim`

# vimtutor

- aptitude install vimtutor
- yum install vimtutor
  - vimtutor replaces the current speaker ;-)
  - vimtutor often included in vim package

# Getting the software

- aptitude install vim vimtutor
- yum install vim vimtutor
- ssh or putty to another laptop

# Ready ?

- Sample file
  - <http://linux-training.be/files/studentfiles/dates.txt>
  - or use any other text file...
  - or an empty file...

# Starting vi

- vi
  - work in an empty unnamed file
- vi file42.txt
  - open the file named file42.txt

# vi modes

- command mode
- insert mode (aka text mode)
- Get back to command mode with 'Esc'



# part 1: the minimum

- a : start typing/editing
- u : undo the last mistake
- :q! : quit without saving my mistakes
- :w : write (aka save) my changes
- :q : quit the vi editor

# part 1: a

- a is a command to put vi in 'insert mode'
- you can use arrow keys
  - always before typing a
  - not always after typing a (vi vs vim)

# part 1: u

- is a command (did you forget 'Esc' ?)
- is short for undo
- undoes the last change to the file

# part1: :q!

- don't panic when you mess up a file
- just use 'Esc :q!' to quit without saving changes
- 'Esc' to get in command mode
- : to get at the bottom of your screen
- q to quit (only works when nothing changed)
- q! to quit without saving changes

# part1: :w

- save your work when done editing
- :w (saves the file, uses existing name)
- :wq (saves and quits)
  - shortcut for :w followed by :q

# end of part 1: the minimum

- a : start typing/editing
- u : undo the last mistake
- :q! : quit without saving my mistakes
- :w : write (aka save) my changes
- :q : quit the vi editor
- :wq : write and quit

# part 2

- more commands
- commands working together
- macro's
- configuration

# start typing

- a A
- i I
- o O
- all six command put vi in 'insert mode'



# replace and delete

- r
- x X
- p

# our first combo

- xp
- switches two characters

# copy paste a line

- dd
- p P
- yy

# copy paste multiple lines

- 3dd
- 8472dd
- p P
- 42yy

# some line combos

- ddp (switch two lines)
- yyp (duplicate a line)

# start and end of line

- 0 (moves cursor to start of line)
- \$ (moves cursor to end of line)

# more combos

- d0
- d\$ (delete from cursor to end of line)

# joining lines

- J
- 2J (same as J)
- 4J (joins four lines into one)



# words

- w (go forward one word)
- b (go back one word)
- 42w (go forward 42 words)
- 33b

# word combos

- dw
- 5dw (cut the next five words)
- 5db
- yw
- 4yw
- 3ywP

# save and quit

- :w
- :w filename
- :wq
- ZZ (you are using qwerty right ;-)
- :q!
- :w! (chmod write chmod)

# searching

- /string
- ?string
- n (jump to next occurrence)

# advanced search

- `/^string`
- `/string$`
- `/K[iou]m`
- `^\<he\>` (beginning and end of word)

# replace

- `:1,$ s/foo/bar`
- `:1,$ s/foo/bar/g`
- `:33,42 s/adama/zaphod`

# read input

- `:r filename`
- `:r !cmd`

# options

- :set number
- :se nu
- :se nonu
- :syntax on
- :set all



# .vimrc

- \$ cat ~/.vimrc  
set number  
set tabstop=8  
map <F6> :set number!<bar>set number?<CR>

# a simple macro

- qa\$q
  - q starts the macro
  - a names the macro
  - \$ is the content
  - q ends the recording

# Q & A

- Thanks for listening
- Share with others
- Be honest in life
- Be fair
- Buy my book!