

Preparation

`git clone https://github.com/aswen/vim-workshop.git`
or

`github.com/aswen/vim-workshop/archive/master.zip`
(bit.ly/16s4tMQ)

```
vim-workshop/  
|-files/ (exercises)  
|-presentation/ (this presentation and handouts.pdf)  
|-vim-config/  
|-startvim <= Script to startup Vim
```

Not yet installed Vim? => www.vim.org => download
No Internet, ask me for usb stick.



Alexander Swen | vim@swen.nu

Based on workshop created together with
Joël Stemmer for Nedap

Why Vim?

Example header

=====

This is header 2

Why Vim?

Example header

=====

This is header 2

=====

Why Vim?



Why Vim?

Example header

=====

This is header 2

=====

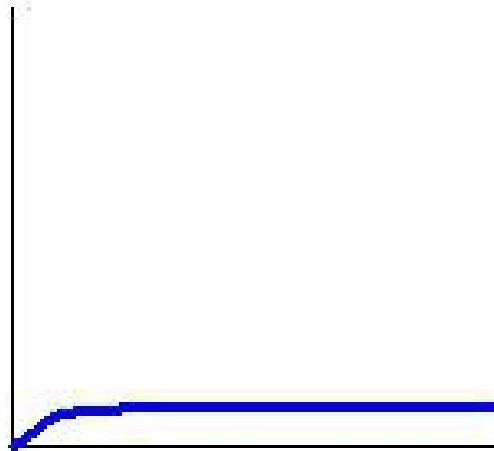
This is a somewhat longer header...



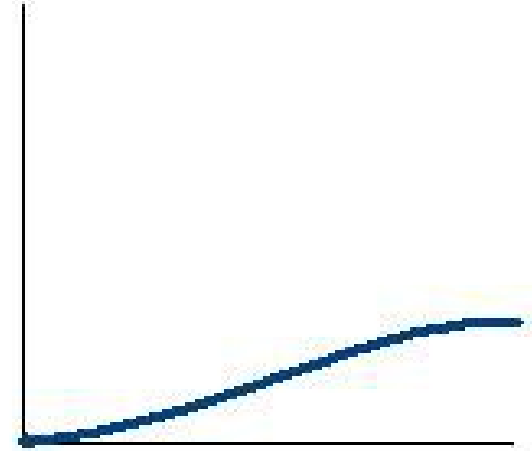
Why Vim?

Classical learning curves for some common editors

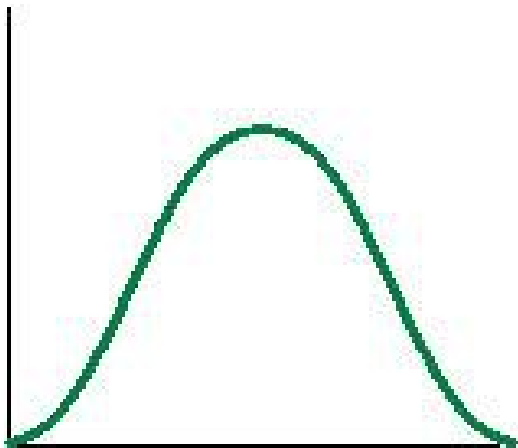
Notepad



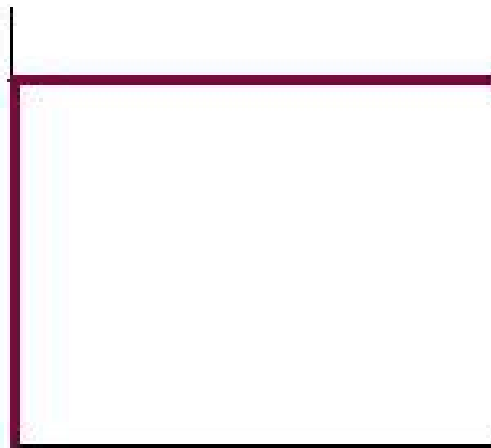
Pico



Visual Studio



vi



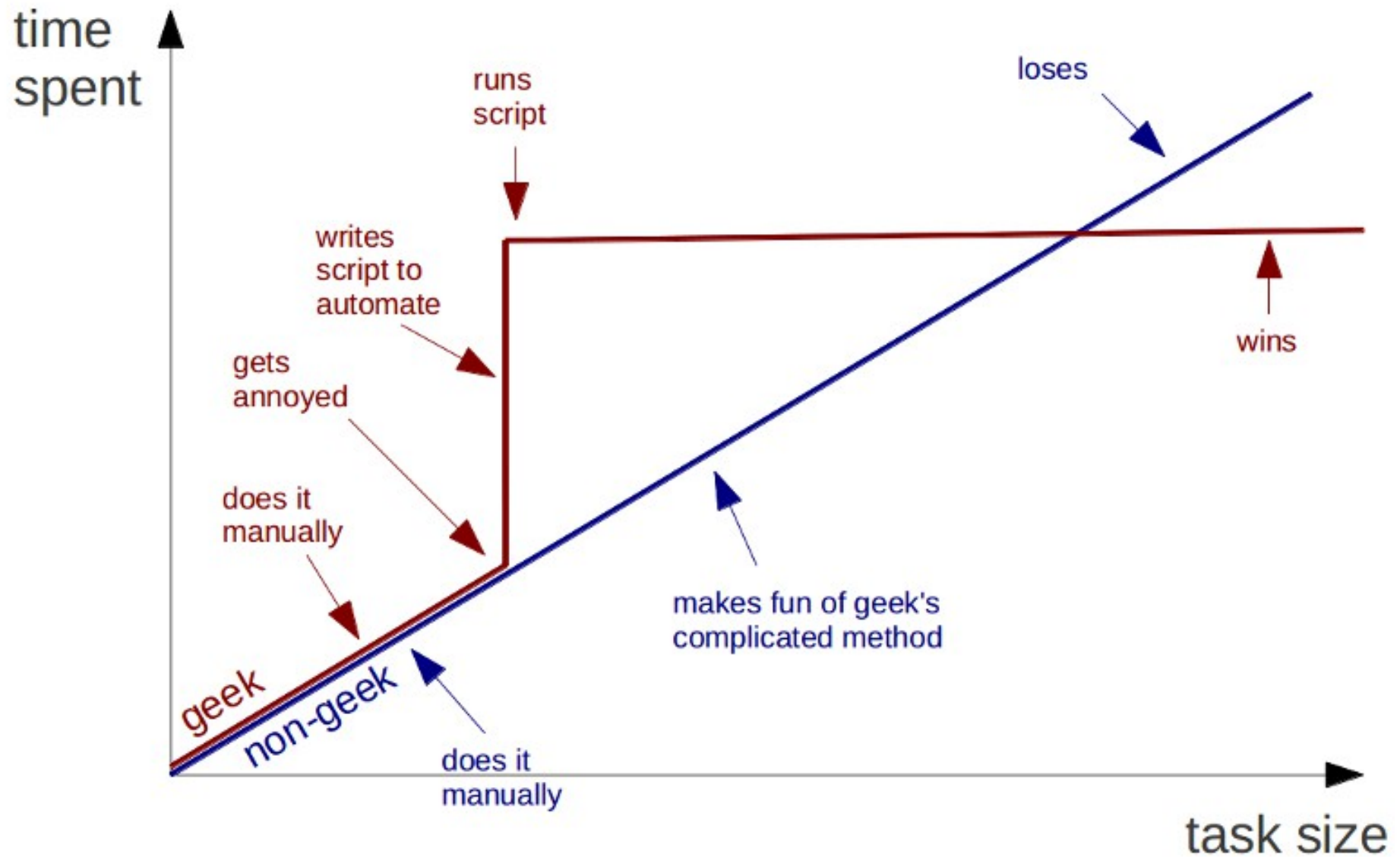
emacs



11-17-04

Why Vim?

Geeks and repetitive tasks



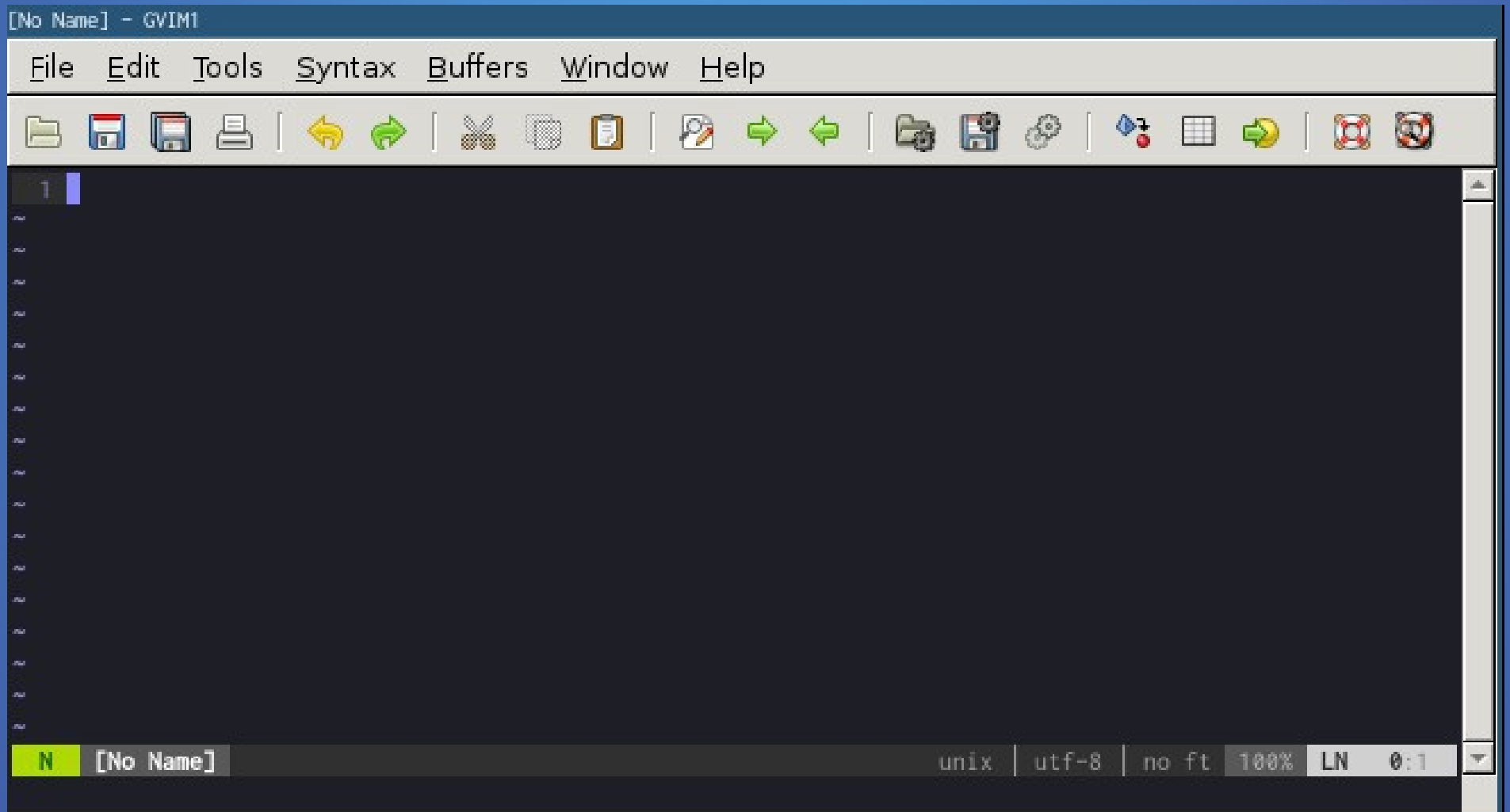
Brief history of Vim



Brief history of Vim



Vim Gui



Don't Panic

<code><esc></code>	Back to normal mode
<code>:q!</code>	Quit discarding changes
<code>:wq</code>	Write changes & quit (or ZZ, or :x)
<code>:e [file]</code>	Edit [file] (:e! <enter> reloads file)
<code>:h</code>	Help!

Modes

Vim Modes

- Normal mode (<esc>)
 - Move cursor/view
 - Text manipulation
 - Ex mode commands (:<cmd>)
 - Search (/ {string})
- Insert mode (i)
 - Input text
- Visual mode (v)
 - Select text

Vim Modes

NORMAL mode

<esc>

INSERT mode

i	i nsert
I	I nsert @begin of 1st word.
a	a ppend
A	A ppend @eol (End Of Line)
o	o pen line
O	O pen line above

VISUAL mode

v, V or <ctrl>+v

Undo/ Redo

u

undo

<ctrl>+r

redo

Repeat actions



Repeat actions n times

#<action> executes the action # times

Example:

4l moves cursor 4 places to the right

4i{some tekst }<esc> creates:

some tekst some tekst some tekst some tekst

Motions

Motions

h

j

k

l



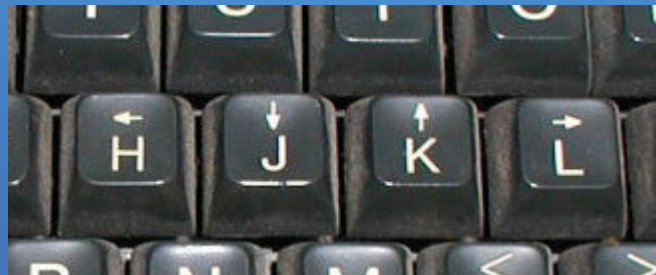
Motions

h

j

k

l



Exercise

Startup vim (./startvim)

- i f f b r <esc>
- a a <esc>
- a e <esc>
- 3 h
- i e <esc> . (an actual dot)
- l a <space><esc>
- A k <esc>
- I c o <esc>
- /a <enter>
- xp

reference

i insert

a append

h j k l left, dn, up, right

A Append *eol*

. Repeat cmd

I Insert *bol*

/a Search “a”

xp Delete/paste

Coffee break

More motions

<home>	0	begin of line (<i>bol</i>)
	^	Begin of 1st word
<end>	\$	end of line (<i>eol</i>)
<ctrl>+<pgup>	gg	begin of file (<i>bof</i>)
<ctrl>+<pgdn>	G	end of file (<i>eof</i>)

More motions

<pgup>

<ctrl>+f

Page **f**orward

<pgdown>

<ctrl>+b

Page **b**ackward

{ }

Paragraph up/down

zz

Center file on cursor

More motions

w next **w**ord

e **e**nd of word

b **b**eginning of word

:22 goto line 22

Operators

Operators

- c **c**hange (has to be followed by motion)
- C **C**hange line from cursor to *eo/*
- cc **c**hange entire line
- d **d**elete (has to be followed by a motion)
- D **D**elete from cursor to *eo/*
- dd **d**elete entire line

Operator + Motion

Operator + Motion

c w **c**hange to end of **w**ord

d b **d**eleate to **b**egin of word (*bow*)

c ^ **c**hange to *bol*

d G **d**eleate to *eof*

Exercise

reference

	:e [file]	Open [file]
Startup vim (./startvim)	{ }	Prev. / next paragraph
– open files/1-lorem.txt	c4w	Change 4 words
– go to 3rd paragraph		
– change 3rd word to “bla”	c	change
– change the following 4 words to “blabla”	c^	Change to <i>bol</i>
– delete the 6th (empty) line		
– go to the end of the 7th word of 4 paragraphs down and change from there to the beginning of that line to “this is fun”.	dd 4}	Delete line Move down 4 paragraphs
	d G	Delete from cursor to <i>eof</i> .

Even more motions

f {char} **f**ind character

t {char} **t**o/till character

Even more motions

f {char} **f**ind character

t {char} **t**o/till character

;
repeat

,
reverse

Operator + motion

c f {char}	c hange f ind character (<u>incl</u> that char)
d t {char}	d elete t o/till character
v f {char}	v isual select f ind character (<u>incl</u> that char)
c 2 t {char}	c hange t o/till 2nd character

Exercise

Open files/4-boilerplate.html

reference

(Only use the motions and operators we learned so far)

– line 16: select navigation/extensions and swap them

c f {char} **change find**
character (incl
that char)

d t {char} **delete to**
character

– line 18: change deck.hashes.css to deck.hash.css

v f {char} **Select find**
character (incl
that char)

– line 82: same as 18

Searching

Searching

`/ {string}` Search string (may be regex)

`n` next occurrence

`<shift>+n` previous occurrence (backwards)
`(N)`

Operator + search

c / {string} Change to string

d / {string} Delete to string

v / {string} Select till 1st char
of string!

Exercise

Open files/4-boilerplate.html

(Only use the motions and operators we learned so far)

	reference
/ {char}	Search {char}
n	Next occurrence

– line 16: select navigation/extensions and swap them

– line 18: change deck.hashes.css to deck.hash.css

– line 82: same as 1

Cut/Copy/Paste
& visual mode

Cut/Yank(copy)/Paste

y w	y ank w ord
y y	y ank current line
d w	d elete w ord
p	p aste after
shift+p (P)	P aste before

Visual mode

v	v isual mode
shift+v (V)	V isual line mode
ctrl+v	v isual block mode

Visual mode + motions

`v w` **v**isual select **w**ord

`shift+v 3j` **V**isual select 4 lines
(V)

`ctrl+v G` **v**isual select a block to eof

Visual mode + operators

- c **c**hange selection
- d **d**elete selection
- y **y**ank (copy) selection
- r **r**eplace each character in
selection
- p **p**aste over selection
- : Ex mode command
(example: s/search/replace/)

Exercise

Open files/6-digits.txt

- Remove lines 1, 3 and last
- Make a csv file

	reference
v	v isual mode
<shift>+v	V isual line mode
<ctrl>+v	v isual block mode
c	c hange selection
d	d elete selection
y	y ank (copy) selection
r	r eplace selection
p	p aste over selection
:	Ex command mode (e.g. :s/search/replace/)

Search/replace

<code>:s/before/after/</code>	Replaces the first occurrence of the word “before” with “after” on this line.
<code>:s/before/after/g</code>	Replaces each occurrence of the word “before” with after on this line or selection (global)
<code>:%s?/usr/lib?/usr/lib64?g</code>	Entire file, ? is allowed as well as a delimiter
<code>:24,27:s<ctrl<shift<g</code>	On lines 24-27, < is also a valid delimiter
<code>&</code>	Repeat last search/replace

Exercise

Open files/6-digits.txt

- Remove lines 1, 3 and last
- Make a csv file (using search/replace)

reference

:s/search/replace/

:%s?search?replace?g

Exercise

Open files/6-digits.txt

- Remove lines 1, 3 and last
- Make a csv file (using search/replace)

reference

:s/search/replace/

:%s?search?replace?g

:%s?\D\+(\d\+)\D\+(\d\+)\D\+?\1,\2?

Text objects

Text objects

i **i**n/inner

a **a**ll/around

More text objects

“ ’ ` ([{ }]) Delimiters (ci”)

w **w**ord

s **s**entence

p **p**aragraph

t tag (bla)

Operator + text objects

c i “ **change inner** string

d i t **delete inner t**ag

y a p **yank all p**aragraph

v a s **visual select all s**entence

c a w **change all w**ord

Fun for if we have time left...

Fun for if we have time left:

vimrc

buffers

windows

plugins

Vimrc

<code>:so %</code>	Reload vimrc while editing
<code>:so \$MYVIMRC</code>	General reload
<code>repo:/files/(g)vimrc-alex</code>	My vimrc... bit messy. (g => gvimrc => if/else thing works only half afaik)

Tip: store your vimrc in a git repo
like so:

<https://github.com/aswen/dotvim>

buffers

- \ b open buffer explorer
- <enter> switch to selected buffer
- d delete buffer (close file)

buffers

- \ b open buffer explorer
- <enter> switch to selected buffer
- d delete buffer (close file)

windows

<ctrl> + w s

window **s**plit

<ctrl> + w v

window **v**ertical split

<ctrl> + w q

window **q**uit

<ctrl> + w h j k l

window **f**ocus

plugins

Pathogen
(github.com/tpope)
/bundle

Tool to make it easy to
manage plugins
Git clone > bundle

```
git submodule add ...  
git submodule init  
git submodule update
```

Easier to use git submodules.
Easier updates.

Vim in other software

More/less/man

Motions and search

Vimium/vrome

Browser plugin voor GC

pentadactyl

Browser plugin voor FF

Bash (set -o vi)

Vim cmd line!

Plugins voor eclipse en netbeans (etc)

vifm

Vim file manager

That's all

Alexander Swen

questions and feedback: vim@swen.nu

irc:freenode:#vim: aswen

You can hire me to do this for your colleagues!

www.vim.org

Drew Neil: vimcasts.org

(he wrote the book “Practical Vim”)

vimcheatsheet.com

vimgolf.com

vim-adventures.com

Tim Pope: github.com/tpope

This workshop has been created with Joël Stemmer
for our colleagues at Nedap