

Voorbereiding

```
git clone github.com/nedap/vim-workshop.git  
of  
github.com/nedap/vim-workshop/archive/master.zip  
(bit.ly/115jxMA)
```

```
vim-workshop/  
|-files/ (oefenbestanden)  
|-presentation/ (deze presentatie en handouts.pdf)  
|-vim-config/  
|-startvim <= hiermee starten we vim vandaag
```

Nog geen vim geïnstalleerd => www.vim.org => download
Geen net? Vraag de usb stick.



nedap

technology that matters















STEPPING STONE







Welcome to You!

September 6,

dap
logy that matters





Example header

=====

This is header 2

Example header

=====

This is header 2

=====



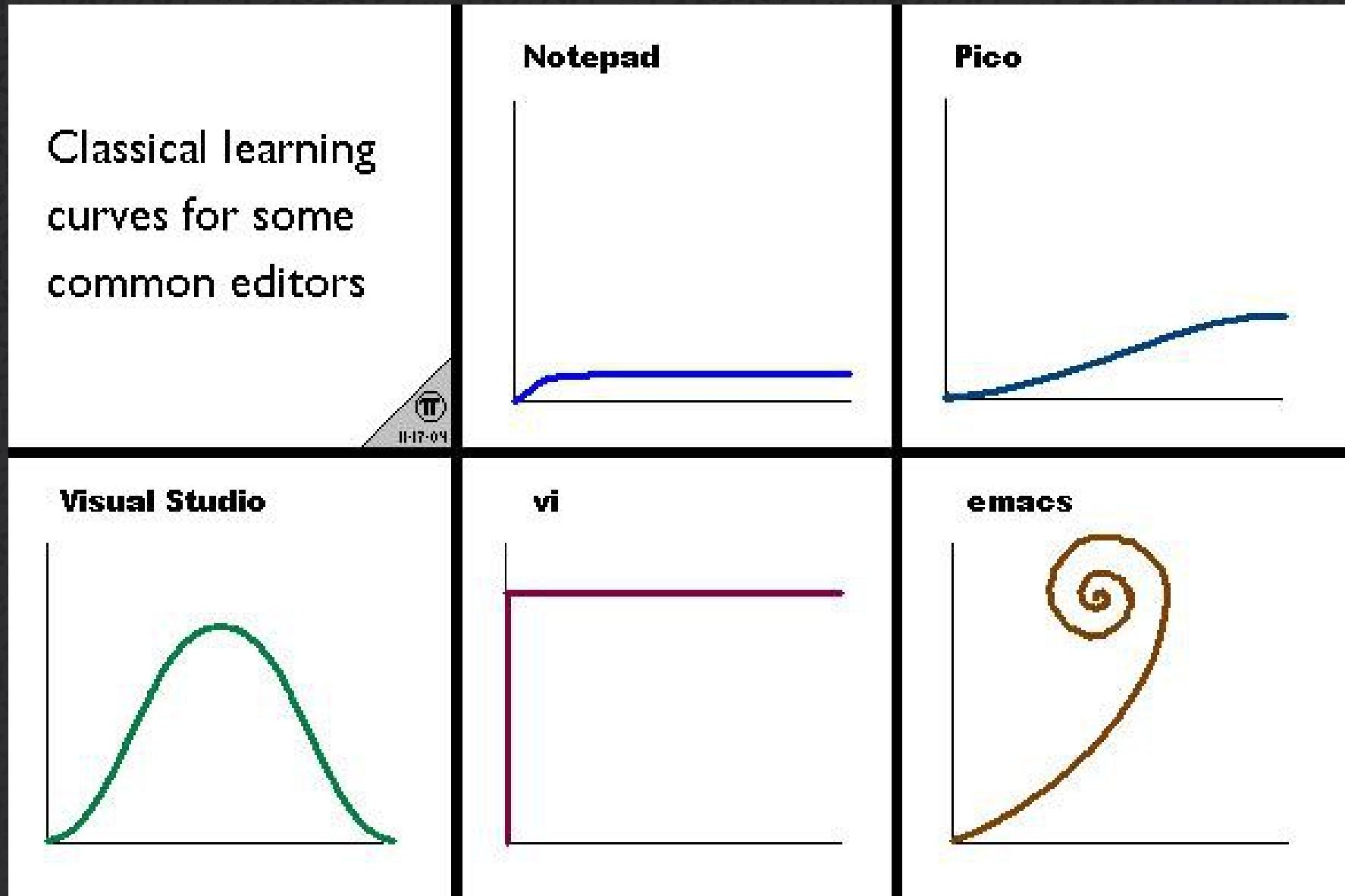
Example header

This is header 2

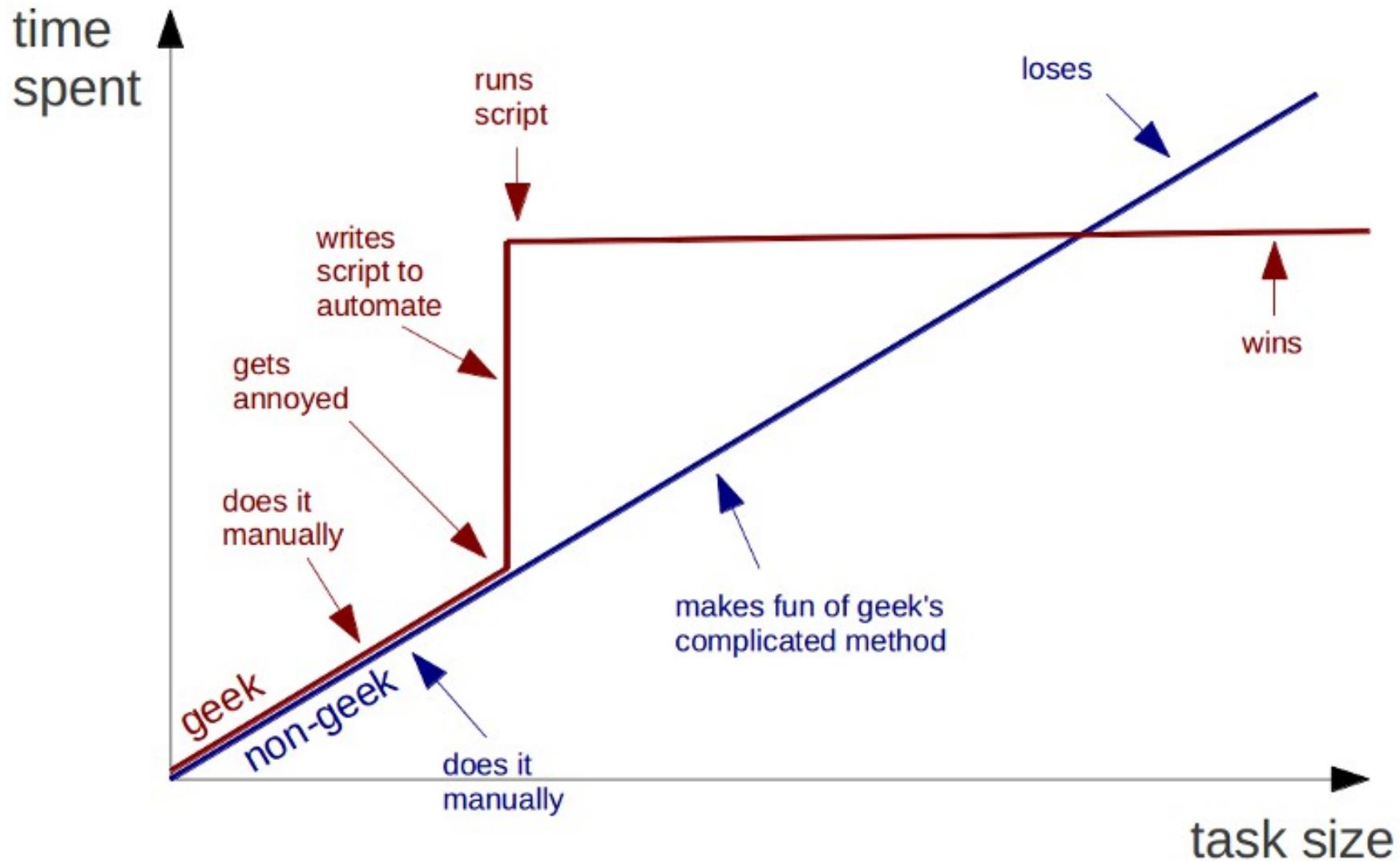
This is a somewhat longer header

imagecredit: fanpop.com





Geeks and repetitive tasks



Vim History



Image credit: <http://www.catonmat.net/blog/why-vim-uses-hjkl-as-arrow-keys/>

Don't panic

<esc>

back to normal mode

:q!

quit discarding changes

:wq

write changes & quit (or ZZ, or :x)

:e [file]

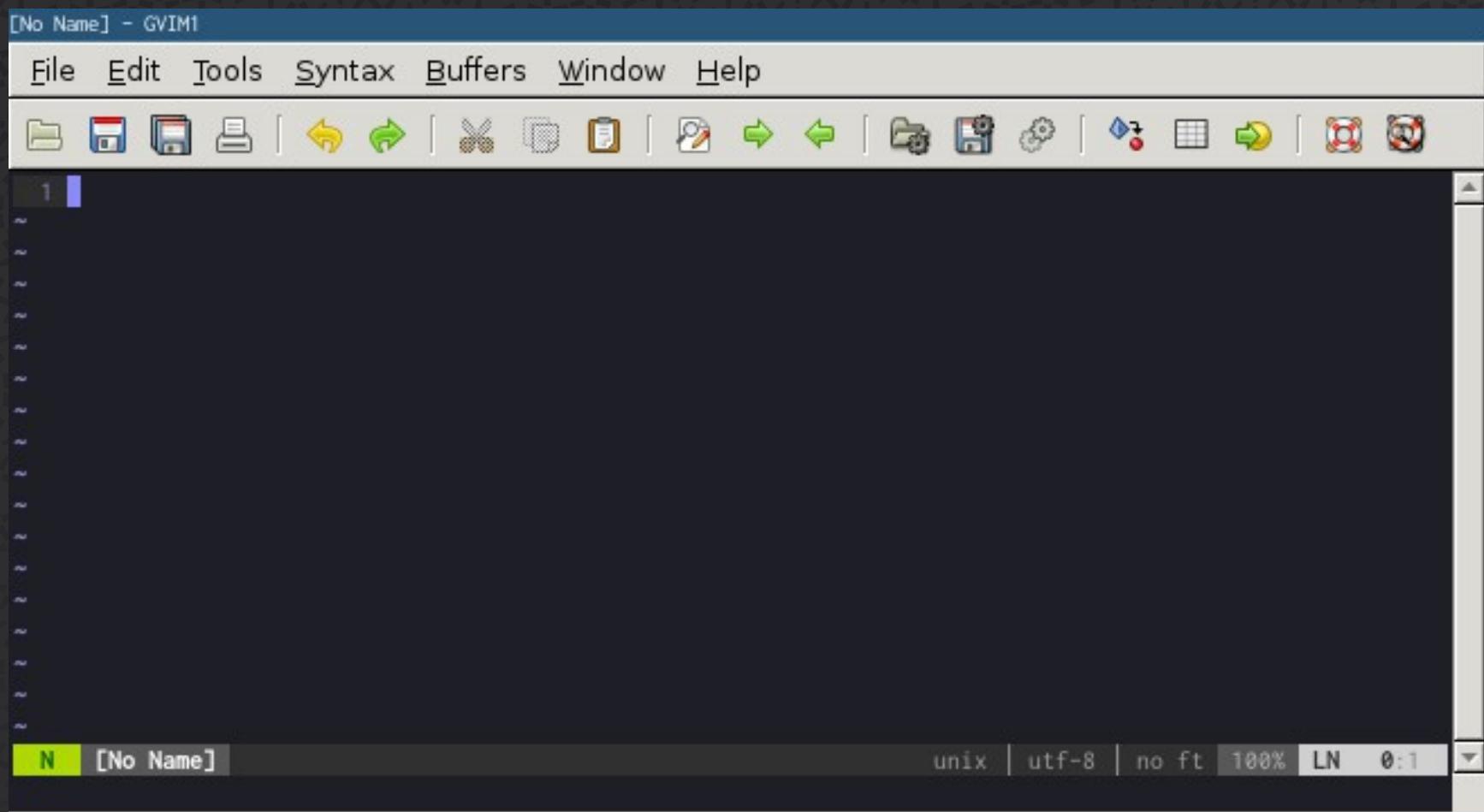
edit [file] (:e! <enter> reload file)

:h

vim help



Vim display



Vim Modes

Vim modes

- * Normal mode (esc)
 - move cursor/view
 - Text manipulation
- * Insert mode (i)
 - Input of text
- * Visual mode (v)
 - Text selection

Vim modes

- * NORMAL mode <esc>
- * INSERT mode i insert
 I insert bol (begin of line)
 a append
 A append eol (end of line)
 o open line
 O open line above
- * VISUAL mode v, V or <ctrl>+v

Undo / Redo

u

Undo

<ctrl>+r Redo

Repeat actions



repeat action n times

4<action> executes the action 4 times
(example: 4l moves 4 characters to the right)

Motions

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*)
- A k <esc>
- I c o <esc>
- /a <enter>
- xp

		reference
	i	insert
	a	append
	h	To left
	A	Append eol
	.	Repeat cmd
	I	Insert bol
	/a	Search “a”
	xp	Delete/paste

coffeebreak



| Steppingstone

Motions

<home> ^ begin of line

<end> \$ end of line

<ctrl>+<pgup> gg begin of file

<ctrl>+<pgdn> G end of file

Motions

<pgup> <ctrl>+f Page forward

<pgdown> <ctrl>+b Page backward

{ } Paragraph up/down

zz Center on cursor

Motions

w next word

e end of word

b beginning of word

:22 goto line 22

Operators

c change (has to be followed by motion)

C Change line from cursor to end of line

cc change entire line

d delete (has to be followed by motion)

D Delete from cursor to end of line

dd delete entire line

Operator + Motion

operator + motion

c w change word

d b delete to begin of word

c ^ change to bol

d G delete to eof

Exercise

Startup vim (./startvim)

- open files/1-lorem.txt
- go to 3rd paragraph
- move down 1 line
- change 3rd word to “bla”
- change the following 4 words to “blabla”
- delete the 6th line
- go to the end of the 7th word of 4 paragraphs down and change from there to the beginning of that line.

reference	:e [file] Open [file]	{ / }	Prev. / next paragraph
4cw		c	Change 4 words
c^			change
dd			Delete line
7}			Move down 7 paragraphs
d G			Delete from cursor to eof.

More motions

f {char} Find character

t {char} to/till character

operator + motion

c f {char} change till character (incl
that char)

d t {char} delete to character

v f {char} Select till character (incl
that char)

exercise

Open files/4-boilerplate.html

Line 16: select
navigation/extensions and swap
them

- line 18: deck.hashes.css should
be deck.hash.css
- line 82: same as 18

c f
{char}

d t
{char}

v f
{char}

reference

change till
character (incl
that char)

delete to
character

Select till
character (incl
that char)

Searching



| Steppingstone

Searching

`/string` Search string

`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

Line 16: select
navigation/extensions and swap
them

/ {char} Search {char}
n Next occurrence

- line 18: deck.hashes.css should be deck.hash.css
- line 82: same as 18
- line 71: “Permalink” => uppercase

reference

Cut/Copy/Paste & visual mode

Cut/Yank(copy)/Paste

y w **yank word**

y y **yank line**

d w **delete word**

p **paste after**

shift+p **paste before**
(P)

Visual mode

v

Visual mode

shift+v
(V)

visual line mode

ctrl+v

visual block mode

Visual mode + motions

v w

visual select word

3 shift+v
(v)

Visual select 3 lines

ctrl+v G

visual select a block
to eof

Visual mode + operators

- c change selection
- d delete selection
- y yank (copy) selection
- r replace selection
- p paste over selection
- :
- Modifier command
(example: s/search/replace/)

Excercise

Open files/6-digits.txt

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

v	Visual mode
<shift>+v	Visual line mode
<ctrl>+v	Visual block mode
c	change selection
d	delete selection
y	yank (copy) selection
r	replace selection
p	paste over selection
:	Modifier command (e.g.: s/search/replace/)

Search/replace

:s/before/after/

Replaces the first occurrence of the word “before” with “after” on this line.

:s/before/after/g

Replaces each occurrence of the word “before” with after on this line or selection (global)

:%s?/usr/lib?/usr/lib64?g

Entire file, ? is allowed as well as a delimiter

:24,27:s<ctrl<shift<g

On lines 24-27, < is also a valid delimiter

Excercise

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

operator + text object

Text objects

i in/inner
a all/around

Text objects

“ ” ‘ ’ [{ }]) Delimiters (ci”)

w

word

s

sentence

p

paragraph

t

tag (bla)

Operator + Text objects

c i “ change **i**nner string

d i t delete **i**nner **t**ag

y a p yank **a**ll **p**aragraph

v a s visual select **a**ll **s**entence

c a w change **a**ll **w**ord

.vimrc

```
:so %
```

Reload vimrc while editing

```
:so $MYVIMRC
```

General reload

```
repo:/files/(g)vimrc-alex
```

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 & Windows

buffers: bufexplorer

\ b open buffer explorer

enter switch to selected buffer

d delete buffer (close file)

Windows

ctrl + w s

window split

ctrl + w v

window vertical split

ctrl + w q

window quit

ctrl + w h j k l

window focus

plugins

Pathogen
(github.com/tiago)
/bundle

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

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

Easier to use git submodules.
Easier updates.

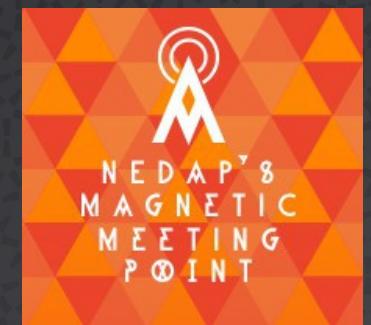
www.vim.org
(vim website, vim scripts)

vimcasts.org

vimgolf.com

vim-adventures.com

Nedap events:
bubbleconf.com
nedapatlowlands.com



Nedap hires: werkenbijnedap.com