

# 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](http://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

---

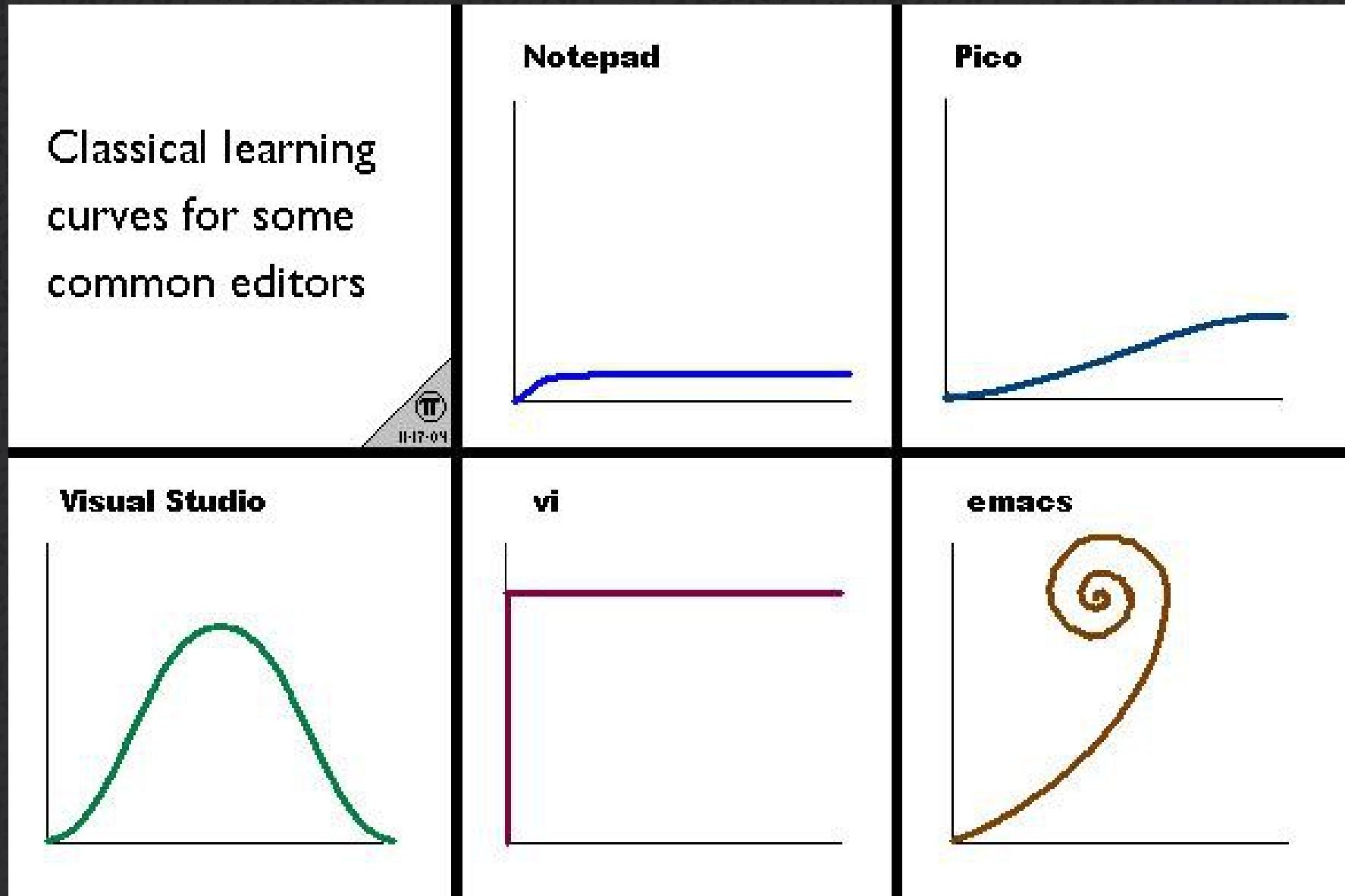
This is header 2

---

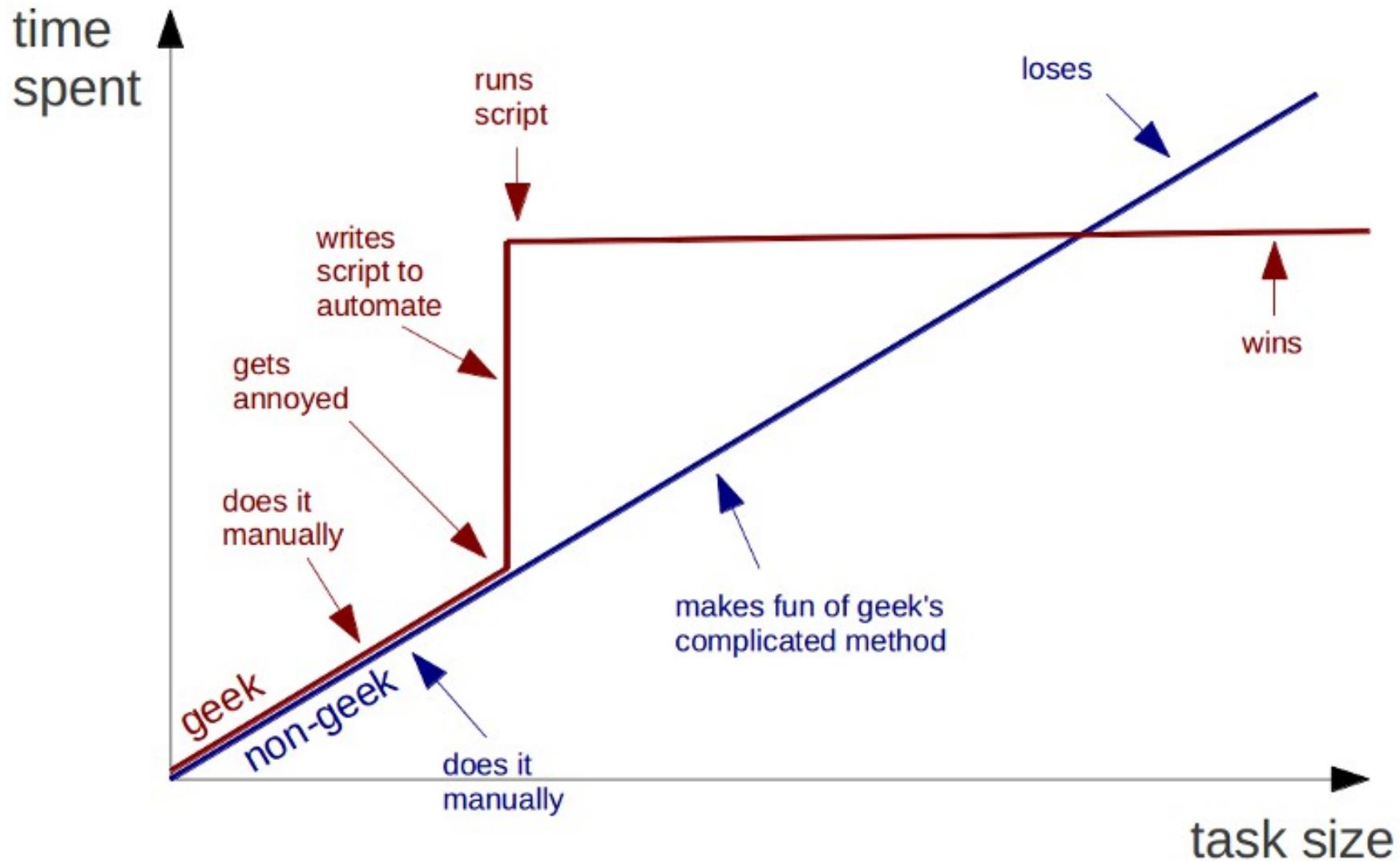
This is a somewhat longer header

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]

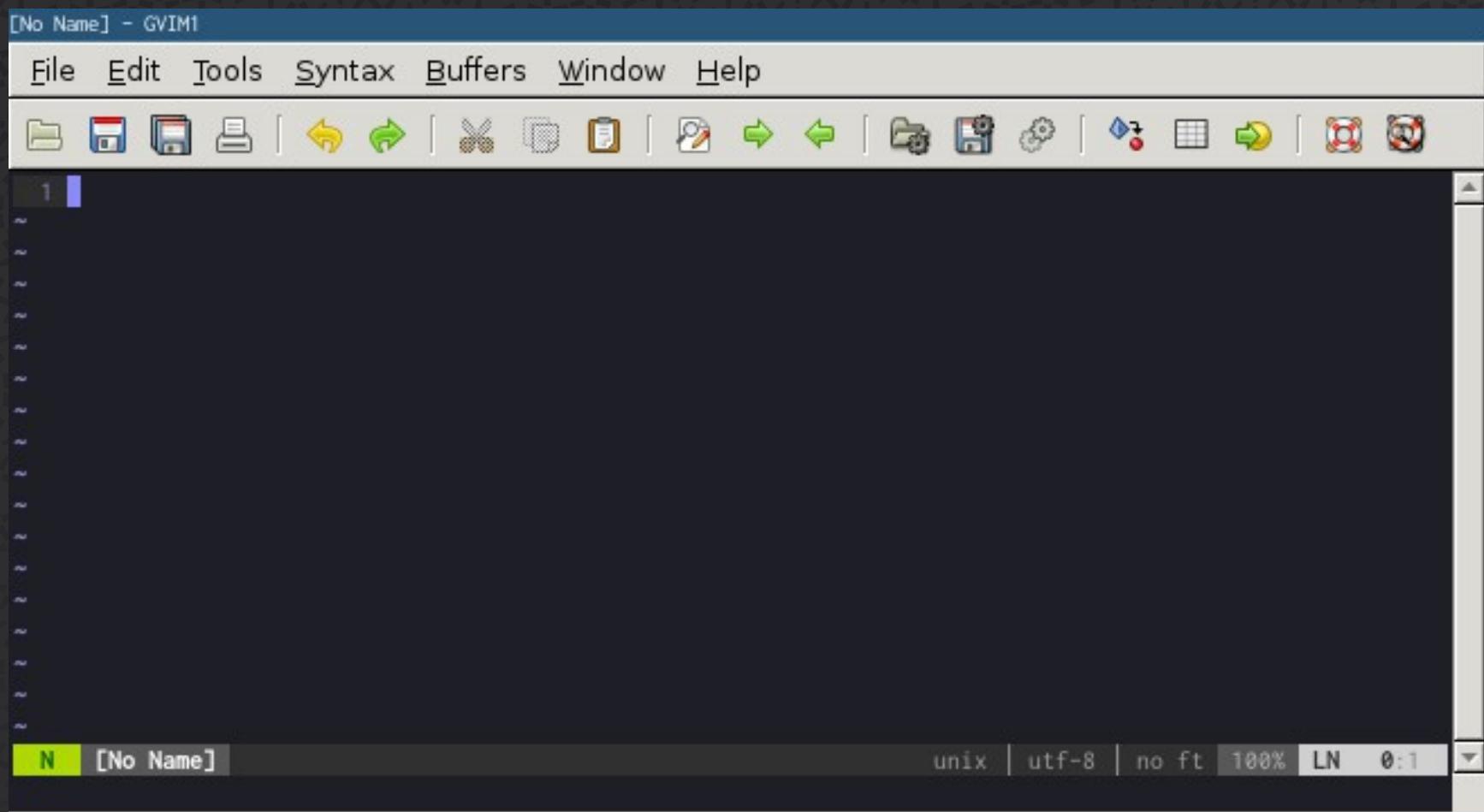
edit [file] (:e! <enter> reload file)

:h

vim help



# Vim display



# Vim Modes

# Vim modes

- \* Normal mode (esc)
  - move cursor/view
  - Text manipulation
  - Ex mode commands ([:<cmd>])
  - search (//{string})
- \* 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 (*bol*)  
                  a    append  
                  A    Append end of line (*eol*)  
                  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*)
- i `<space><esc>`
- A k `<esc>`
- I c o `<esc>`
- /a `<enter>`
- xp

		reference
	i	insert
	a	append
	hjk1	left,dn,up,right
	A	Append <i>eol</i>
.	.	Repeat cmd
	I	Insert <i>bol</i>
	/a	Search “a”
	xp	Delete/paste

# coffee break



| Steppingstone

# Motions

<home> ^ begin of line (*bol*)

<end> \$ end of line (*eol*)

<ctrl>+<pgup> gg begin of file (*bof*)

<ctrl>+<pgdn> G end of file (*eof*)

# 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 *eol*

cc change entire line

d delete (has to be followed by motion)

D Delete from cursor to *eol*

dd delete entire line

# Operator + Motion

# operator + motion

c w change word

d b delete to begin of word (*bow*)

c ^ change to *bol*

d G delete to *eof*

# Exercise

reference

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.
- |           |                                    |
|-----------|------------------------------------|
| :e [file] | Open [file]                        |
| { }       | Prev. / next paragraph             |
| c4w       | Change 4 words                     |
| c         | change                             |
| c^        | Change to <i>bol</i>               |
| dd        | Delete line                        |
| 4}        | Move down 4 paragraphs             |
| d G       | Delete from cursor to <i>eof</i> . |

# More motions

f {char} find character

t {char} to/till character

# operator + motion

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

d t {char} delete to character

v f {char} visual select find character (incl that char)

c 2 t {char} change to 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
- line 18: change deck.hashes.css to deck.hash.css
- line 82: same as 18

c f {char} **change** find character (incl that char)  
d t {char} **delete** to character  
v f {char} Select **find** character (incl that char)

# Searching



| Steppingstone

# 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 again  
*(now use the search method to navigate faster)*

– line 16: select navigation/extensions and swap them

– line 18: deck.hashes.css should be deck.hash.css

– line 82: same as 18

– line 71: “Permalink” (in title) => uppercase

reference

/ {char}      Search {char}

n                  Next occurance

# 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

shift+v 3j   Visual select 3 lines  
(V)

ctrl+v G   visual select a block  
to eof

# Visual mode + operators

- c      change selection
- d      delete selection
- y      yank (copy) selection
- r      replace each character in  
selection
- p      paste over selection
- :
- Ex mode command  
(example: s/search/replace/)

# Excercise

Open files/6-digits.txt

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

v

<shift>+v

<ctrl>+v

c

d

y

r

p

:

reference

visual mode

Visual line mode

visual block mode

change selection

delete selection

yank (copy) selection

replace selection

paste over selection

Ex command mode  
(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

# 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

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

# operator + text object

# Text objects

i      in/inner  
a      all/around

# Text objects

“ ” ‘ ’ [ { } ] ) Delimiters (ci”)

w

word

s

sentence

p

paragraph

t

tag (<b>bla</b>)

# 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.

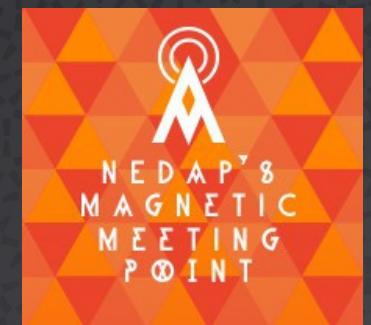
[www.vim.org](http://www.vim.org)  
(vim website, vim scripts)

[vimcasts.org](http://vimcasts.org)

[vimgolf.com](http://vimgolf.com)

[vim-adventures.com](http://vim-adventures.com)

Nedap events:  
[bubbleconf.com](http://bubbleconf.com)  
[nedapatlowlands.com](http://nedapatlowlands.com)



Nedap hires: [werkenbijnedap.com](http://werkenbijnedap.com)