*nedap

technology that matters

STEPPING STONE

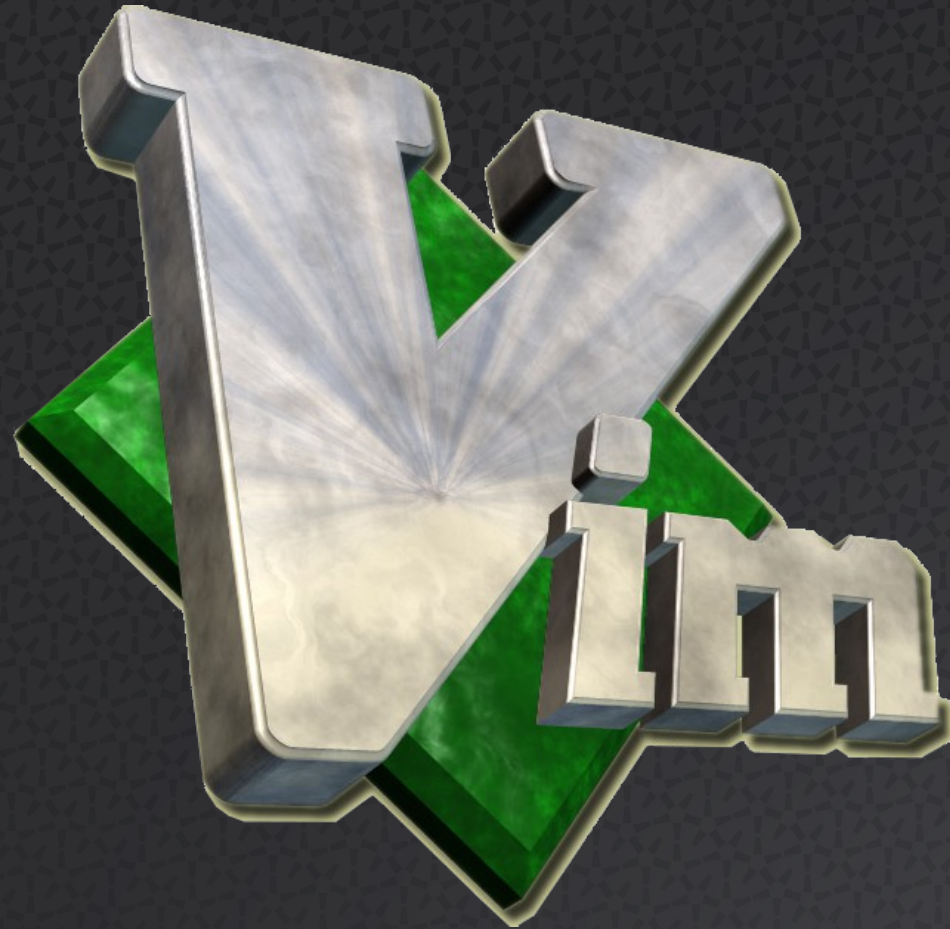# Example header

This is header 2

Example header

This is header 2

Example header

This is header 2

This is a somewhat longer header

Today

* Brief history of Vim

* The git repo

* Modes

* Motions

* Operators

* Searching

* Cut/copy/paste and visual mode

* Operators and tekstobjects

* Buffers and windows

* Search/replace

* Vim config file: ~/.vimrc

* Plugins (if time left)

Vim History

nedap | Steppingstone

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

Vim-workshop/
|-files/ (oefenbestanden)
|-presentation/
|-vim-config/
|-startvim
```

Don't panic

Esc   back to normal mode

:q!   quit discarding changes

:wq   write changes & quit

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

:h   vim help

```
 N   [No Name]          unix   utf-8   no ft   100%        0:1
:q!
```

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

* Stay only in INSERT mode when actually entering text

* Move Cursor in NORMAL mode

* Text manipulations in NORMAL mode

* Undo/redo of actions in NORMAL mode

* Repeat last action in NORMAL mode

* Text selection either in NORMAL or in VISUAL mode

Vim modes

* NORMAL mode Esc
* INSERT mode   i   insert
                  I   insert begin
                  a   append
                  A   append end
                  o   open line
                  O   open line above

* VISUAL mode   all move commands work

Undo / Redo

u   Undo

ctrl+r   Redo

Repeat actions

Motions

Motions

h   j   k   l

←   ↓   ↑   →

Excercise

Startup vim (./startvim)

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

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
	ctrl+d	begin of file
	ctrl+u	end of file
	zz	Center on cursor

Motions

w   next **w**ord

e   **e**nd of word

b   **b**eginning of word

Operators

c   **c**hange

d   **d**elete

y   **y**ank (copy)

R {char}   **r**eplace

Operator + Motion

operator + motion

c w   **c**hange **w**ord

d b   **d**elete to **b**egin of word

c ^   **c**hange to begin of line

d G   **d**elete to end of file

operator shortcuts

c c   change line

d d   delete line

y y   yank (copy) line

More motions

`f {char}`   Find character

`t {char}`   to/till character

operator + motion

c f {char}	**c**hange till character (incl that char)	
d t {char}	**d**elete to character	
v f {char}	Select till character (incl that char)	
g U t {char}	Uppercase to character	

Searching

Searching

`/{string}`	Search string
`n`	next occurrence
`shift+n` `(N)`	previous occurrence (backwards)

operator + search

c / {char}   Won't work!

d / {char}   **d**elete to character

v / {char}   Select till character

Cut/Copy/Paste
& visual mode

Yank (copy) / Paste

y w	**y**ank **w**ord
y y	yank line
p	**P**aste after
shift+p (P)	Paste before

Visual mode

v	Visual mode
shift+v (V)	visual line mode
ctrl+v	visual block mode

Visual mode + operators

c   **c**hange selection

d   **d**elete selection

y   **y**ank (copy) selection

p   **p**aste over selection

:   Modifier command
(e.g.: s/search/replace/)

operator + text object

Text objects

i   in/inner

a   all/around

Text objects

" ' ` ([{ }])	delimiters
w	word
s	sentence
p	paragraph
t	Tag (bla)

Operator + Text objects

c i "   change inner string

d i t   delete inner tag

y a p   yank all paragraph

v a s   visual select all sentence

c a w   change all word

Buffers & Windows

Search/replace

`:s/before/after/` Replaces the first occurance of the word "before" with "after" on this line.

`:s/before/after/g` Replaces each occurance 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 lins 24-27, < is also a valid delimiter

.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)

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/tpope)

/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

Fun:
vimgolf.com
vim-adventures.com