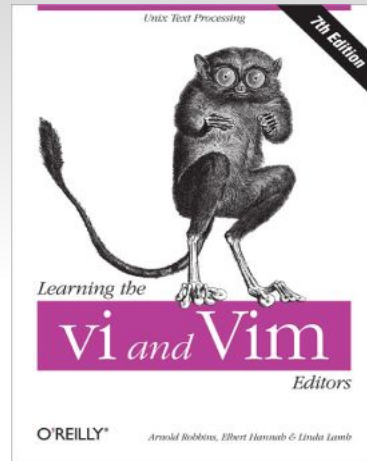


# De editor vi



In dit hoofdstuk worden de basistechnieken voor het werken met de text editor `vi` besproken

---

## Doelstelling

Aan het eind van dit hoofdstuk is de cursist bekend met:

- waarom is kennis van `vi` nodig?
- starten van `vi`
- de verschillende modi in `vi`
- elementaire handelingen in `vi`
- het bewaren (*saven*) van werk
- afsluiten van `vi`
- de profile bestanden `.exrc` en `.vimrc`

---

## Waarom vi?

- vi is altijd aanwezig op een Unix systeem (en soms de enige editor die u tot uw beschikking heeft).
- weinig overhead
  - weinig belasting voor het systeem, ook met meerdere gebruikers
  - start snel op
  - ook met groter bestanden een snelle navigatie
- zeer aanpasbaar
- visual full screen editor!

---

## Starten van vi

- `$ vi [bestandsnaam ...]`

Indien geen bestandsnaam wordt opgegeven of bestand bestaat niet:



Anders wordt de inhoud van het bestand getoond met ook de cursor links bovenaan.

---

## Afsluiten van vi

Opslaan en vi verlaten:

- Indien gestart zonder bestandsnaam:
    - `:wq naam_van_nieuw_bestand`
  - Indien gestart met een bestandsnaam:
    - `:wq` (of `:x` of `ZZ`)
- Let op: `:qw` = fout! geen geldig commando

Niet opslaan en vi verlaten (de ultieme ontsnapping..):

- `:q!`

Gebruik geen `<Ctrl>-C` en zeker **nooit** `<Ctrl>-Z` !

---

## Verschillende modi

Het verschil met vele andere editors is dat vi 3 verschillende modi kent:

### 1. command mode

- de default: bij opstarten is vi in command mode
- in deze mode worden heel veel zaken gedaan (cursorverplaatsing, deleten van tekst, copy & paste, etc..) maar is niet om tekst in te typen

### 2. input mode

- in deze mode kun je tekst intypen

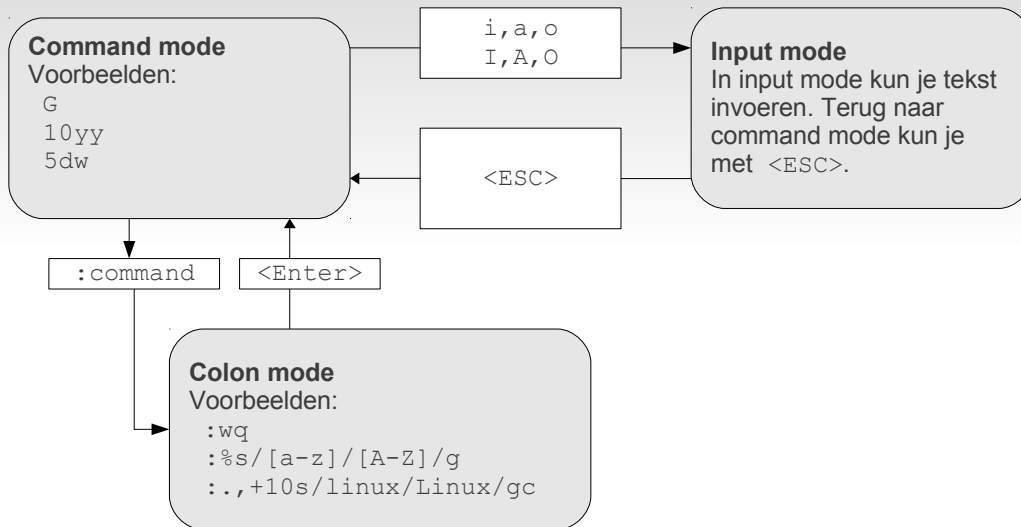
### 3. colon mode:

- ook wel regel-edit mode: hier kunt u 1-regelige functies uitvoeren die betrekking hebben op het bestand (waaronder bv. `:wq`)

---

# Switchen tussen modi

Dit kan het best in het volgende plaatje worden weergegeven:



---

# De command mode

In de command mode kun je o.a. de volgende zaken regelen:

- cursorverplaatsing
- verwijderen (en plakken) van tekst
- kopiëren (en plakken) van tekst
- ongedaan maken van aanpassingen

---

## Cursorverplaatsing (1/2)

(command mode)



- de toetsen: h,j,k en l
- ^ begin van de huidige regel
- \$ eind van de huidige regel
- *nw* n woorden vooruit (*n* = getal)  
(woord: reeks tekens gescheiden door witruimte of leestekens)
- *nW* n WOORDEN vooruit (grotere stappen dan *nw*)  
(WOORD: reeks tekens gescheiden door witruimte)
- *nb* n woorden achteruit
- *nB* n WOORDEN achteruit

---

## Cursorverplaatsing (2/2)

(command mode)

Grotere stappen:

- H naar het begin van het scherm
- L naar het eind van het scherm
- ^B een heel scherm terug (^ = <Ctrl>)
- ^F een heel scherm vooruit
- *nG* naar regel *n*
- G naar laatste regel van het bestand (Go = Begin onder de laatste regel gelijk in input mode)
- */pattern* Ga naar *pattern* (vooruit vanaf de cursor)
- *?pattern* Ga naar *pattern* (achteruit vanaf de cursor)
  - n (next) naar volgende instantie van *pattern*

---

## Verwijderen (en plakken) van tekst

(command mode)

- *nx* delete *n* characters onder en na de cursor ("delete")
- *nX* delete *n* characters onder en voor de cursor ("backspace")
- *ndd* delete *n* lijnen
- *ndw* delete *n* woorden vanaf de cursor (*dnw* kan ook)
- *ndW* delete *n* WOORDEN vanaf de cursor (*dnW* kan ook)
- *dG* delete vanaf de cursor tot einde bestand
- *D* delete tot einde lijn (*nD* = delete *n* lijnen)
- *nJ* Join lines (delete EOL's)

Delete and put (cut and paste):

- *p* plaats (put) verwijderde tekst achter de cursor
- *P* plaats (Put) verwijderde tekst voor de cursor

## Kopieren (en plakken) van tekst

(command mode)

- *nyy* kopieer *n* lijnen
- *nY* kopieer *n* lijnen
- *nyw* kopieer *n* woorden vanaf de cursor (*ynw* kan ook)
- *nyW* kopieer *n* WOORDEN vanaf de cursor (*ynW* kan ook)

Yank and put (copy and paste):

- *p* plaats (put) gekopieeerde tekst achter de cursor
- *P* plaats (Put) gekopieeerde tekst voor de cursor

---

## veranderen van tekst

(command mode)

Al voorbeelden gezien bij delete en copy

Verder:

`rx` (replace) vervangen teken onder de cursor door karakter `x`.

`R` vervangen teken onder de cursor door een of meerder tekens, afsluiten met `<ESC>`

`cw` Vervang woord met nieuwe karakters, afsluiten met `<ESC>`

`cW` Vervang WOORD met nieuwe karakters, afsluiten met `<ESC>`

`c$` Vervang tot einde regel, afsluiten met `<ESC>`

`cH` Vervang begin van bestand tot en met de lijn met de cursor, afsluiten met `<ESC>`.

---

## ongedaan maken van aanpassingen

(command mode)

- `nu` undo laatste `n` wijzigingen
- `U` herstel de laatst gewijzigde lijn
- `n^R` redo de laatste `n` undo's (`^` = `<Ctrl>`)

---

## Input mode

De input mode is niet zo ingewikkeld: gewoon typen.

Herstellen van bv. een type fout kan zolang u nog geen `<Enter>` heeft gegeven door er met de backspace naartoe te gaan.

Maar voor alle overige handelingen is de command mode of de colon mode.

---

## Colon mode (1/3)

Door vanuit de command mode de dubbele punt (:) te typen enabled u de colon mode.

In colon-mode heeft u de beschikking over de regel-editor ed (voorloper van vi) met alle bijbehorende ed-commando's.

U kunt niet alleen uw bestand opslaan en vi verlaten maar ook:

- een ander bestand inlezen,
- bewerkingen uitvoeren op alle of een selectie van regels,
- regels wegschrijven naar een ander file, etc...

Met ed-commando's is veel mogelijk!

---



## Colon mode (2/3)

Bewaar en verlaat acties:

<code>:w</code>	bewaar de inhoud van uw bestand
<code>:w file</code>	bewaar de inhoud in <i>file</i>
<code>:q</code>	verlaat (quit) vi
<code>:q!</code>	verlaat vi zonder uw tekst op te slaan
<code>:wq</code>	combieer opslaan en quit (kan ook met <code>:x</code> of <code>ZZ</code> )

---

## Colon mode (3/3)

Werken met selecties:

<code>:%s/ios/Linux/gic</code>	vervang op alle regels (%), alle (g) patterns "ios" door "Linux", ignore case (i), confirm (c) is nodig.
<code>:. ,+10s/^/□□/</code>	vervang vanaf de huidige lijn (.) t/m 10 regels verder het begin (^) van elk van deze regels met 2 spaties (□ = spatie).
<code>:3,10w file</code>	schrijf regels 3 t/m 10 naar <i>file</i>
<code>!:cmd</code>	voer <i>cmd</i> uit, daarna <Enter> om terug te keren naar vi

---

## The power of vi

<code>:%s/[A-Z]/\L&amp;/g</code>	vervang alle (g) hoofdletters ([A-Z]) door kleine letters (\L&)
<code>:wn (:wprev)</code>	ga naar het volgende (vorige) file (indien vi gestart is met meerdere files).
<code>:!cmd</code>	voer cmd uit, daarna <Enter> om terug te keren naar vi
<code>:g//m0</code>	verwissel alle regels (laatste eerst, etc...)
<code>:%s!sort</code>	sorteer alle regels
<code>.</code>	herhaal de laatste actie

---

## vi aanpassen

<code>:set all</code>	geeft een overzicht van alle beschikbare opties om vi aan te passen
-----------------------	---

Bijvoorbeeld:

<code>:set number</code>	zet regelnummering aan
<code>:set showmode</code>	laat de mode zien waar vi zich bevind

Deze settings zijn permanent te maken in `~/ .exrc` of `~/ .vimrc`

- De dubbele punt (:) is dan nog nodig.
- `.vimrc` gaat voor `.exrc`

Als een `.vimrc` bestaat dan zal vim in "nocompatible" mode opstarten: vim is dan minder compatable met vi.

---

## vi of vim?

- vim = Vi IMproved: een *superset* of vi  
Alles wat in vi kan kan ook in vim.
- vim is *ported* naar meer OS-en
- vim heeft "multi undo's en redo's"
- vim kan het scherm splitsen bij werken met meerdere files
- vim kan editen in *compressed* files (gzip, zip, tar, etc..)
- vim heeft support voor populaire programmeertalen, bv.:
  - *syntax highlighting*
  - *code folding*
  - *parentheses matching*

---

## Referenties

<http://www.vim.org/>

<http://thomer.com/vi/vi.html> ("The Vi Lovers Home Page")

[http://vim.wikia.com/wiki/Vim\\_Tips\\_Wiki](http://vim.wikia.com/wiki/Vim_Tips_Wiki)

---

# Oefeningen

Tijd voor oefening!

---