

Programmeren voor kinderen

***Geschreven door Peter Armstrong
Vertaald door Yves Hanoulle***



Programmeren voor kinderen

Peter Armstrong en Yves Hanouille
(@YvesHanouille)

Dit boek is te koop bij
<http://leanpub.com/programmerenvoorkinderen>

Deze versie is gepubliceerd op 2017-12-30



Leanpub

Dit is een [Leanpub](#) boek. Leanpub stelt auteurs en uitgevers in staat om volgens het Lean uitgeefproces te werken. [Lean Publishing](#) is het uitgeven van een boek dat nog onderhanden is met lichtgewicht gereedschap en vele iteraties om feedback te krijgen van de lezers. Op deze manier kun je aanpassingen maken tot je het juiste boek hebt, en als je zover bent helpt het om te zorgen dat je een positie krijgt in de markt.

© 2013 - 2017 Peter Armstrong en Yves Hanouille (@YvesHanouille)

Tweet This Book!

Please help Peter Armstrong en Yves Hanouille (@YvesHanouille) by spreading the word about this book on [Twitter](#)!

The suggested hashtag for this book is [#ProgrammerenVoorKinderen](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

[#ProgrammerenVoorKinderen](#)

Also By These Authors

Books by [Peter Armstrong](#)

[Lean Publishing](#)

[Programming for Kids](#)

[La Programmation Pour Les Enfants](#)

[Programación para Niños](#)

[The Markua Specification](#)

[Writing a Leanpub Book in Multiple Word Documents](#)

[Writing a Leanpub Book in One Word Document](#)

[An Unofficial Mapping from Markua to HTML5](#)

[The Pubcoin Whitepaper](#)

[The Ten Meal Week](#)

[Forum Test](#)

Books by [Yves Hanouille \(@YvesHanouille\)](#)

[Who is agile? Volume 1](#)

[The Leadership Game](#)

[Who is agile in South Africa?](#)

[Who is agile in Australia & New Zealand?](#)

[Who is agile in Ukraine?](#)

[Who is agile in India](#)

[Das Leadership-Spiel](#)

[The Scaling ball game](#)

[Pair Programming](#)

[Who is agile in Singapore?](#)

Ego workshop

Gereedschapskist voor de Agile Coach – Visualisatievoorbeelden

Peter: Letterlijk voor Evan.

Inhoudsopgave

Inleiding	1
Hoofdstuk 1: Hallo, opdrachtregel!	2
Oefeningen	11
Hoofdstuk 2: Hallo Wereld!	12
Oefeningen	16
Voor de ouders	17
Over de auteur:	19
Over de vertaling	20
Over de vertaler	21

Inleiding

In dit boek zal je zelf computerprogramma's leren schrijven!

Je hebt een Mac computer nodig om mee te kunnen doen. De programma's zijn kort, zodat je ze allemaal zelf kan typen. Dat kan zelfs als je zelf niet zo goed kan typen.

Dit boek bevat verschillende hoofdstukken. Elke hoofdstuk gaat over een idee.

Als je ouders meer willen weten over dit boek dan kunnen ze het samen met jou lezen. Of ze kunnen alleen de hoofdstuk [Voor de ouders](#) op het einde van het boek lezen.

Als je op de computer van je ouders gaat werken zullen ze waarschijnlijk het boek samen met jou willen doornemen. Dat lijkt me leuk. Zo leren ze ook nog wat!

Als je op jouw Mac wil volgen ga je waarschijnlijk het PDF bestand lezen dat je ouders hebben gedownload. Als je het PDF document dubbelklikt zal het openen in het programma Preview. Je zal dan heen en weer schakelen tussen dit boek en de ingave in Terminal.

Als je op Preview venster geklikt hebt om van pagina te veranderen, zal je terug op het Terminal venster moeten klikken om de focus op dat venster te zetten **voor** je een commando kan typen.

Als je het vervelend vindt om te wisselen tussen Preview en Terminal kan je, als je een printer hebt, dit boek afprinten. Als je een iPad hebt kunnen je ouders de EPUB file downloaden en het aan iBooks toevoegen om het dan zo op de iPad te kunnen lezen.

Laten we [beginnen](#)!

Hoofdstuk 1: Hallo, opdrachtregel!

Dit hoofdstuk gaat over de opdrachtregel, in het Engels is dat de “command line”!

Dit boek is voor kinderen zoals jij die, thuis of op school, een Mac computer gebruiken.

Normaal gebruik je een Mac door met de muis te klikken nu ga je je Mac ook op een andere manier leren gebruiken. Je zal met het toetsenbord leren werken via de opdrachtregel.

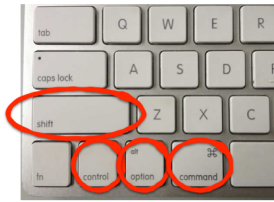
Vroeger, zo’n 30 jaar geleden, hadden de computers nog geen muis en was er alleen de opdrachtregel om de computer te bedienen.

Hoewel de opdrachtregel zeer eenvoudig is, is hij zeer krachtig. Als je weet hoe je de opdrachtregel moet gebruiken zal je je een tovenaarsvoelen die kan typen in vreemde taal en zo zijn computer ongelooflijke dingen kan laten doen.

Misschien maakt dit je een beetje bang maar dat is voor niets nodig. Ik heb dit boek zodanig opgebouwd dat als je niets kapot kan maken wanneer je iets verkeerd doet.

We gaan eerst de basis leren.

Je weet al wat een toetsenbord is. Op dat toetsenbord staan toetsen die je waarschijnlijk nog nooit gebruikt hebt zoals de Command, Option, Control en Shift toetsen.

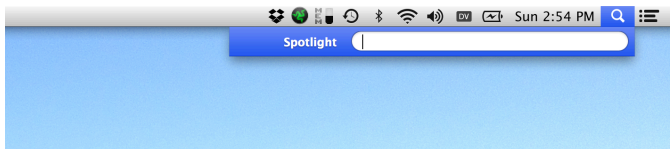


The Command, Option, Control and Shift Keys

We gaan het programma met de naam Terminal starten. Terminal is het programma dat je de opdrachtregel laat gebruiken op je Mac.

Om Terminal te starten, druk je de Command toets en tegelijk op de spatiebalk. Een kortere manier om dit te zeggen is: typ “**Command + Spatiebalk**”. Dat is hoe ik dat vanaf nu zal zeggen.

Als je **Command + Spatiebalk** typt, open je het programma Spotlight in de rechterbovenhoek van je scherm.

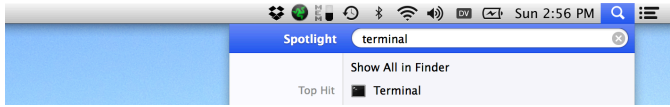


Spotlight

Spotlight laat je toe om programma's te starten door hun naam in te typen in Spotlight.

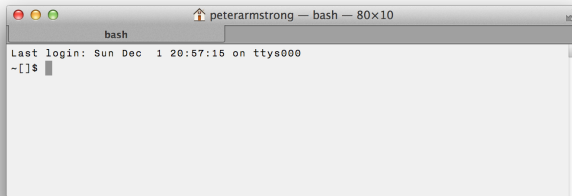
We gaan Terminal laden, dus typ **terminal** in Spotlight en druk op de ENTER toets.

Het kan zijn dat het woord terminal al verschijnt in een DropDown lijst nadat je **term** hebt getypt. Als dat zo is, dan mag je daar gewoon op klikken in plaats van **terminal** voluit te typen.



Running Terminal

Nu zie je Terminal die er ongeveer zo uit ziet.



Terminal

Terminal geeft de opdrachtregel voor je Mac.

(Negeer de ~[]\$ aan het begin van de lijn maar. Dat is gewoon hoe de “prompt” er op mijn computer uit ziet. Het is mogelijk dat jouw prompt er anders uit ziet. Bij jou staat er waarschijnlijk > of ~.)

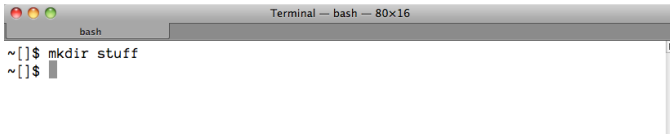
Voor we beginnen laten we de opdrachtregel “Hello” tegen ons zeggen. Dat is Engels voor “Hallo”, maar dat wist je waarschijnlijk al. Om dat te doen maken we gebruik van het programma **say**.

Typ **say hello** in Terminal en druk op ENTER.

Hopelijk heeft je Mac “Hello” tegen jou gezegd! Indien niet, kijk eens of het geluid aan staat en probeer opnieuw.

Trouwens, nadat je een opdracht ingeeft in Terminal moet je op ENTER drukken om het programma uit te voeren.

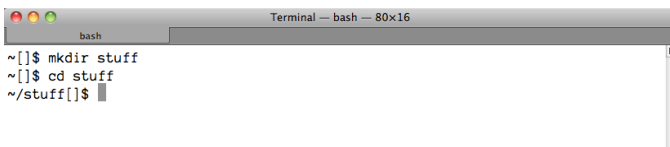
Laten we nu een folder maken, waarin we al de dingen die we gaan maken voor dit boek gaan bewaren. Het Engelse woord voor “dingen” is “stuff”. Typ **mkdir stuff** in Terminal en druk op ENTER.



```
Terminal — bash — 80x16
bash
~[]$ mkdir stuff
~[]$
```

De opdracht **mkdir** spreken we uit als “make dir” en dat is de afkorting van het Engelse “make directory”, dat wil zeggen “maak folder”. In het Engels gebruikt men zowel het woord directory als het woord folder. Je hebt dus zojuist een nieuwe folder gemaakt met als naam **stuff**.

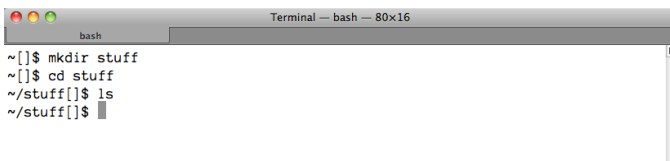
In een volgende stap gaan we naar die folder. Typ **cd stuff** in Terminal en druk op ENTER.



```
Terminal — bash — 80x16
bash
~[]$ mkdir stuff
~[]$ cd stuff
~/stuff[]$
```

De **cd** opdracht staat voor “change directory” of “wijzig folder”. Je wijzigt dus je locatie naar de folder “stuff” die je daarnet maakte.

Laten we eens kijken in de folder. In Terminal, typ **ls** (een klein letter L en een kleine letter S) en druk dan op ENTER.



```
Terminal — bash — 80x16
bash
~[]$ mkdir stuff
~[]$ cd stuff
~/stuff[]$ ls
~/stuff[]$
```

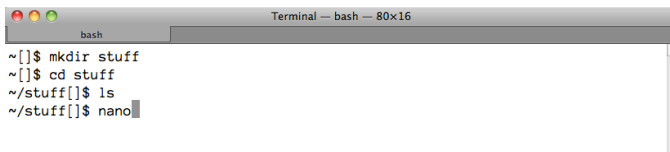
Als je **ls** typt, dan krijg je niets te zien!

De **ls** opdracht staat voor het Engelse “list” of “lijst”. Die opdracht wil zeggen: geef mij een lijst van alle bestanden en folders die in deze folder staan. Er staat niets in de folder. Weet je ook waarom? Juist, je hebt de folder daarnet aangemaakt en daarom is die nog leeg. Als je dus **ls** typt, krijg je niets te zien.

Laten we dat veranderen door een bestand aan te maken in die folder.

We gaan een tekstverwerker gebruiken om een bestand te maken en te bewerken. Een tekstverwerker laat je woorden typen in een bestand dat je nadien kan bewaren, net zoals je een spel kan bewaren om nadien verder te spelen. We gaan een kleine tekstverwerker gebruiken met de naam **nano**. Nano betekent heel klein, en dat is een goed gekozen naam, aangezien **nano** een hele kleine en eenvoudige tekstverwerker is.

In Terminal, typ je **nano** en dan druk je op de ENTER toets.

A screenshot of a macOS Terminal window titled "Terminal — bash — 80x16". The prompt is "bash". The user has entered the following commands: `mkdir stuff`, `cd stuff`, `ls`, and `nano`. The cursor is at the end of the `nano` command.

```
bash
~[]$ mkdir stuff
~[]$ cd stuff
~/stuff[]$ ls
~/stuff[]$ nano
```

Dit start Nano. Het scherm zal er nu zo uitzien.

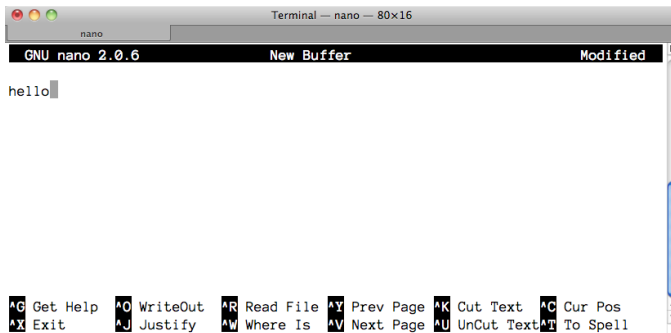
A screenshot of the Nano text editor interface running in a Terminal window titled "Terminal — nano — 80x16". The top status bar shows "GNU nano 2.0.6" and "New Buffer". The editor is currently empty. At the bottom, there is a help menu with various keyboard shortcuts and their functions.

```
GNU nano 2.0.6      New Buffer

[Empty editor content]

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^_ To Spell
```

In Nano, typ je **hello**. Nu mag je **niet** op de ENTER-toets drukken.



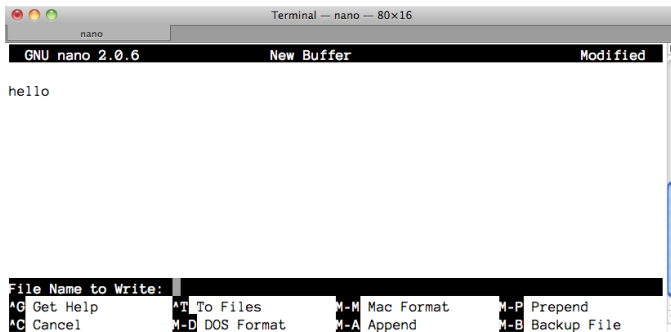
```
Terminal - nano - 80x16
nano
GNU nano 2.0.6      New Buffer      Modified

hello

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^_ To Spell
```

Druk nu de **Control**-toets in en typ de **o**-toets om het bestand te bewaren.

Nano zal je vragen hoe je het bestand wil noemen dat je wil bewaren. Hij doet dat door te vragen “File Name to Write”, of “Wat is de naam om je bestand weg te schrijven”?



```
Terminal - nano - 80x16
nano
GNU nano 2.0.6      New Buffer      Modified

hello

File Name to Write:
^G Get Help  ^O To Files  ^M Mac Format  ^P Prepend
^C Cancel    ^D DOS Format ^A Append     ^B Backup File
```

Typ **hello.txt** en druk op de **ENTER** toets.

```

Terminal -- nano -- 80x16
nano
GNU nano 2.0.6      New Buffer      Modified

hello

File Name to Write: hello.txt
^G Get Help      ^T To Files      ^M Mac Format      ^P Prepend
^C Cancel        ^D DOS Format    ^A Append        ^B Backup File

```

Nano zal het bestand bewaren en zeggen dat het één lijntje lang was, “Wrote 1 line”.

```

Terminal -- nano -- 80x16
nano
GNU nano 2.0.6      File: hello.txt

hello

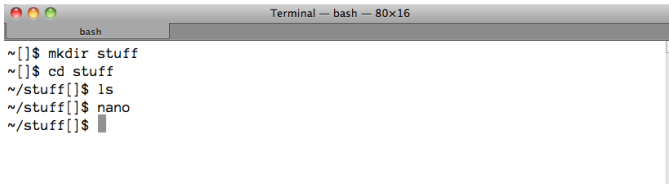
[Wrote 1 line]
^G Get Help      ^O WriteOut      ^R Read File      ^Y Prev Page      ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is       ^N Next Page      ^U UnCut Text    ^S To Spell

```

Als je toch op ENTER gedrukt had nadat je “hello” getypt had, dan zal Nano zeggen “Wrote 2 lines”. Dat is ok. Dat wil gewoon zeggen dat Nano 2 lijnen heeft weggeschreven.

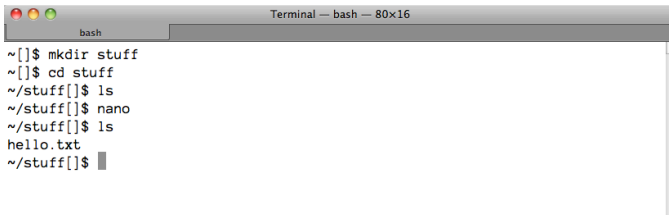
Nu dat we het bestand weggeschreven hebben, kunnen we Nano verlaten door de Control-toets in te drukken en **x** te typen. Een kortere manier om dat te zeggen is “typ **Control + x**”. Ik zal het vanaf zo schrijven.

Nu ben je terug op de opdrachtregel van Terminal.



```
Terminal — bash — 80x16
bash
~[]$ mkdir stuff
~[]$ cd stuff
~/stuff[]$ ls
~/stuff[]$ nano
~/stuff[]$
```

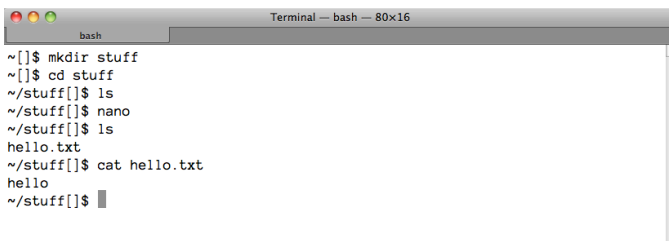
Laten we terug naar onze **stuff**-folder kijken. Herinner je je nog wat we ingegeven hadden om te kijken wat er in de folder stond? Juist: we schreven **ls** en we zagen dat de folder leeg was. Laten we dat nu eens opnieuw proberen. Typ opnieuw **ls** in Terminal. Je ziet nu dat er het bestand “hello.txt” staat dat je zonet zelf gemaakt hebt.



```
Terminal — bash — 80x16
bash
~[]$ mkdir stuff
~[]$ cd stuff
~/stuff[]$ ls
~/stuff[]$ nano
~/stuff[]$ ls
hello.txt
~/stuff[]$
```

Laten we nu in dit bestand kijken. Typ **cat hello.txt** in Terminal en druk op ENTER.

Hoewel “cat” het Engelse woord is voor “kat”, gaat **cat** hier niet over het dier maar over “catalog”. Het is een opdracht of commando dat je de inhoud toont van een bestand. Aangezien we het woord “hello” typten in het bestand is het dat woord wat we hier zien.



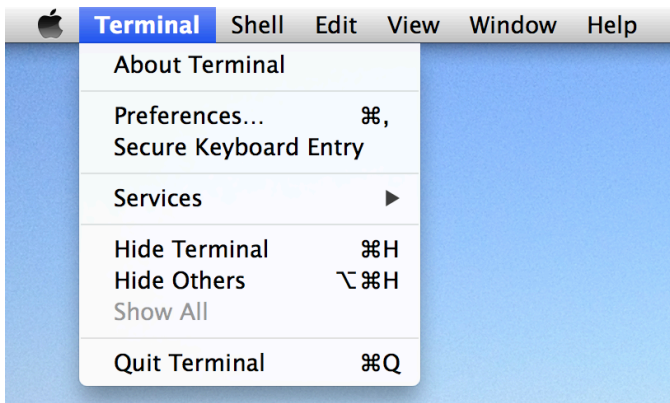
```
Terminal — bash — 80x16
bash
~[]$ mkdir stuff
~[]$ cd stuff
~/stuff[]$ ls
~/stuff[]$ nano
~/stuff[]$ ls
hello.txt
~/stuff[]$ cat hello.txt
hello
~/stuff[]$
```

Aangezien in het bestand hello.txt “hello” stond, zag je dus “hello” als je **cat hello.txt** typte.

Je kent nu sommige van de basisopdrachten van de opdrachtregel! Je leerde een aantal rare opdrachten, zoals `mkdir`, `cd`, `ls` en `cat`. Wist je dat deze opdrachten of commando's eigenlijk ook programma's zijn die door andere mensen geschreven zijn? Je maakte ook je eerste bestand aan.

In het volgende hoofdstuk ga je je eerste programma schrijven. En dat programma zal je ook kunnen aanroepen vanop de opdrachtregel!

Als je even een pauze wil, kan je terminal op elk moment stoppen door **Command + Q** te typen of "Quit Terminal" te kiezen van het bestand menu in Terminal.



Trouwens, als je denkt dat er iets fout gaat op de opdrachtregel, en je weet niet meer wat je kan doen, kan je altijd Terminal afsluiten en daarna gewoon opnieuw starten.

Maak zeker de oefeningen op de volgende pagina voor je verder gaat!

Oefeningen

1.1. Start Terminal en `cd` naar de `stuff`-folder. Hoe kan je controleren dat je in de juiste folder staat?

1.2. Gebruik Nano om een bestand `hooray.txt` aan te maken. Dit bestand moet de tekst “hooray” bevatten. Sluit Nano af wanneer je klaar bent.

1.3. Gebruik het `cat` commando om de inhoud te zien van je `hooray.txt` bestand.

1.4. Verlaat Terminal.

De [antwoorden op de oefeningen](#) kan je achteraan bekijken of je kan gewoon doorgaan met [volgende hoofdstuk](#).

Hoofdstuk 2: Hallo Wereld!

Laten we ons eerste computerprogramma schrijven!

Als je nog steeds volgt dan heb je Terminal al open en ben je in de **stuff**-folder. Laat je Terminal gewoon open en volg verder mee.

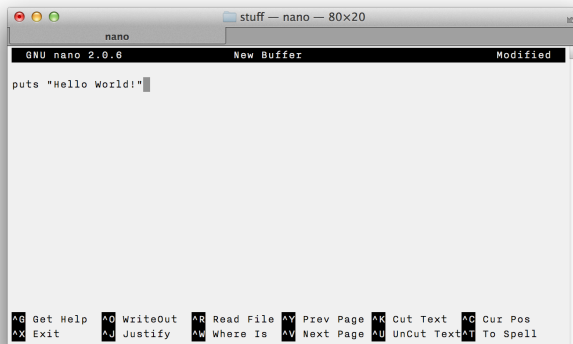
Als Terminal niet meer draait start je Terminal en ga je in de **stuff**-folder door het volgende te doen:

1. Typ **Command + Spatiebalk** om Spotlight te starten
2. Typ **terminal**
3. Druk op de **ENTER**-toets
4. Typ **cd stuff** in Terminal.

Je bent nu in het programma Terminal in de **stuff**-folder.

Om Nano te starten, typ je gewoon **nano** in het Terminal venster en drukt op **ENTER**.

We gaan een zeer eenvoudig programma schrijven. Typ **puts "Hello World!"** in Nano. Het is niet nodig om op **ENTER** te drukken. Ik zal later meer uitleg geven over **puts**.



Het aanhalingsteken ” vind je onderaan op de toets “3”. Er is geen verschil tussen het aanhalingsteken vooraan en achteraan.

Dat is het!

Typ **Control** + **o** om het bestand te bewaren.

Nano zal je vragen welke naam je het bestand wil geven. Hij doet dat in het Engels middels de tekst “File Name to Write:” Typ **hello.rb** en druk op de **Enter**-toets. Als je de naam ingeeft zal je die in de “File Name to Write”-locatie aan de onderkant van Terminal zien.

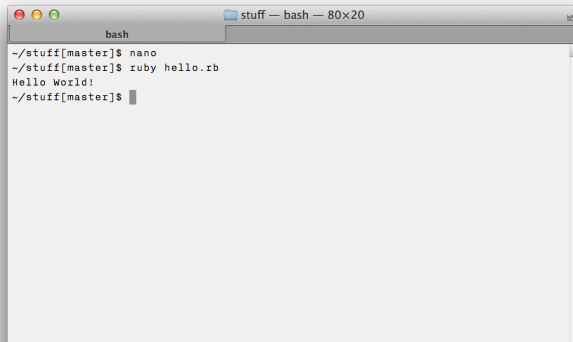


Nano zal het bestand bewaren en je vertellen dat het één lijntje lang was door te zeggen “Wrote 1 line”. Als je toch op de ENTER toets had gedrukt zal het zeggen “Wrote 2 lines”. Dat is ook goed.

Nu je het bestand opgeslagen hebt, geef je **Control + x** in om Nano te stoppen.

Laten we jouw programma uitvoeren!

In Terminal typ je **ruby hello.rb** en druk op **Enter**. Let op dat er een spatie is tussen **ruby** en **hello.rb**.

A screenshot of a macOS terminal window titled 'stuff — bash — 80x20'. The terminal shows a user at the prompt '~/.stuff[master]\$' running the command 'nano'. The next prompt is '~/.stuff[master]\$ ruby hello.rb', followed by the output 'Hello World!'. The prompt returns to '~/.stuff[master]\$' with a cursor. The terminal has a light gray background and a dark gray title bar with standard macOS window controls (red, yellow, green buttons) on the left.

```
stuff — bash — 80x20
bash
~/.stuff[master]$ nano
~/.stuff[master]$ ruby hello.rb
Hello World!
~/.stuff[master]$
```

Je eerste computerprogramma is af!

Proficiat, je hebt nu je eerste computerprogram geschreven! Je ziet dat **puts** iets op het scherm weergeeft.

Maar wacht eens even, wat was dat vreemd woord “ruby”?

Ruby is een programmeertaal. Je spreekt Nederlands, maar er zijn nog heel veel andere talen die mensen spreken. Ook bij het programmeren zijn er veel verschillende talen die je kan gebruiken om een computer te vertellen wat hij moet doen. Ruby is een van de meest gebruiksvriendelijke en je het werd standaard geïnstalleerd op een Mac.

In het volgende hoofdstuk zullen we met Ruby leren werken op zijn eigen opdrachtregel!

Oefeningen

2.1. Schrijf een programma in Ruby dat `hi.rb` noemt en dat `Hi!` op het scherm toont en voer het uit.

Je kan [de antwoorden op de oefeningen](#) lezen of je kan gewoon doorgaan met het [volgende hoofdstuk](#).

Voor de ouders

Dit boek is bedoeld om samen met je kind door te nemen. Je kan het ook door je kind alleen laten te lezen.

Ik heb het boek geschreven om mijn zoon de basis van het programmeren te leren. (Dat is ook de reden waarom Yves het boek in het nederlands vertaalde.) Mijn zoon is nu (2013) 9 jaar, maar tegen de tijd dat het boek klaar zal zijn zal hij 10 zijn. Hij speelt al jaren video spelletjes en hij wil leren programmeren omdat hij op een dag zijn eigen video spelletjes wil schrijven. Dit boek wil daarvoor de eerste stap zijn. (Je leert niet om het soort videospelletjes te schrijven dat een negenjarige kan verzinnen; dat vraagt heel wat meer kennis!)

Ik wil met dit boek het beste boek maken voor een kind dat wil beginnen leren programmeren. Programmeren is een goede vaardigheid om te hebben onafhankelijk van wat je kind later - als volwassene - zal doen. We denken zelfs dat het belangrijker is dan de wiskunde die je in het middelbaar leert. Maar nog belangrijker; als je leert programmeren leer je ook de nauwkeurig en gestructureerd denken. Dat is bruikbaar in elk beroep. Dit boek wil je kinderen helpen bij de eerste stappen voor het beginnen programmeren van computers.

Met de uitzondering van deze bijlage, is dit boek geschreven zoals een boek voor kinderen van de lagere school. Mijn doel is dat dit boek begrijpbaar is voor kinderen tussen 9 en 12 jaar oud. In Noord-Amerika - net als in België - zitten die kinderen in het derde tot zesde leerjaar. Mijn zoon werkt zich door het boek terwijl ik het schrijf. Ik denk niet dat het een goed boek is voor kinderen van 7 tot 8 jaar oud. Ik begon dit boek te schrijven toen mijn zoon 7 was, maar toen was hij er nog niet klaar voor. Dus heb ik het twee jaar in de frigo gelegd. Als jouw kind 7 tot 8 jaar oud is, dan denk ik iets als Scratch een betere keuze is.

Er wordt geen voorkennis van programmeren verwacht. De voorbeelden zijn zo kort mogelijk, aangezien ik veronderstel dat de lezer nog niet vlot kan typen. Mijn zoon kan niet blind typen. Als mijn voorbeelden te lang zijn laat hij me dat weten!

Na het lezen van dit boek kunnen je kinderen volgen wat hen interesseert. Dit boek veronderstelt dat ze een Mac gebruiken. Ik denk dat een Mac de beste computer is om kinderen te leren programmeren. Aangezien dit boek bedoeld is voor kinderen die beginnen te programmeren kan ik dit niet schrijven voor Mac, Windows of Linux. Ik moest voor een besturingsstelsel kiezen, zodat jouw kind letterlijk kan volgen.

Naast het aanleren van programmeren leert het boek ook de basis in om te werken met de opdrachtregel in het programma Terminal op de Mac. Ik denk dat het de beste manier om te volgen en alles over te typen. Echte programmeurs gebruiken de opdrachtregels dagelijks. Als je wil leren programmeren gebruik je Terminal en bestanden. Ja, je kan ook leren op websites als Codacademy - en hoewel dit zeer gebruiksvriendelijke en leerrijk is - is wat je leert fundamenteel verschillend van wat echte programmeurs doen. En hoewel het gemakkelijk is geeft het minder voldoening.

Als je jouw kind **jouw** Mac laat gebruiken voor de oefeningen, raad ik je **sterk** aan, dat je naast hem of haar zit en mee volgt! Ik leer bijvoorbeeld niet hoe ze bestanden moeten verwijderen want het is zeer snel gebeurd!

De voorbeelden zijn in Ruby. Ruby is een vrij eenvoudige programmeertaal. Dit is echter geen boek om Ruby te leren. De voorbeelden konden even goed in JavaScript, CoffeeScript of Python geschreven zijn.

Over de auteur:

Ik ben de oprichter van Ruboss, een software consultancy bedrijf in Vancouver, BC, Canada. We zijn de bedenkers van Leanpub, een website die iedereen kan gebruiken om zelf boeken uit te geven en te publiceren als e-boek, zoals dit boek. Ik ben ook een programmeur, een auteur en een vader. Ik schreef eerder, twee boeken voor programmeurs (*Flexible Rails* en *Hello! Flex 4*), dus ik weet hoe je dingen moet uitleggen aan programmeurs. Mijn zoon is een zeer intelligente jongen van 9. Ik heb heel wat ervaring met dingen uit te leggen aan een slim kind. Dit boek schrijven is waarschijnlijk een van mijn grootste uitdagingen, aangezien ik het beste inleidende boek over **programmeren voor kinderen** wil schrijven.

Over de vertaling

Ik, Yves, heb dit boek vertaald omdat ik mijn eigen kinderen wou leren programmeren. Ik ben geen professionele vertaler. Ik ben er zeker van dat ik er foutjes in liet staan maar ik wou mijn kinderen niet tegenhouden omdat ik zelf niet klaar was met de vertaling van het boek. Want naast leren programmeren wil ik mijn kinderen vooral leren dat ze hun eigen droom moeten volgen. Gelukkig zijn er mensen die me hielpen om dit boek na te lezen. Dank aan:

- Thomas Bouve
- Guy De Bruyn
- Johan De Croppe
- Iris De Caluwé
- Els Ryssen

Als u als lezer vijf fouten doorgeeft bezorg ik u een coupon waarmee u het boek gratis kan aanbieden aan een vriend.

Over de vertaler

Ik heb drie kinderen: Joppe (2002), Bent(2004), Geike (2007). Ze zijn alle drie heel geïnteresseerd in computers en technologie. Ze leerden de eerste stappen van programmeren van het bordspel: [Robot Turtles](#)¹. Het was voor mij een logische volgende stap om het boek te vertalen. Ik ben ook een van de twee lead coaches van [CoderDojo Gent](#)², waar we kinderen die kunnen lezen en schrijven leren om te programmeren.

Mijn vrouw werkt als opvoedster en werkt met mensen met autisme.

Ikzelf werk als creative collaboration agent. Dat wil zeggen dat ik software teams help om beter samen te werken.

[Yves Hanouille](#)³

¹<http://www.robotturtles.com>

²<http://gent.coderdojobelgium.be>

³<https://leanpub.com/u/YvesHanouille>