

# Inhoud

	<b>Voorwoord</b>	<b>xv</b>
	<b>Dankwoord</b>	<b>xvii</b>
	<b>Over de auteur</b>	<b>xviii</b>
	<b>Deel I: Opensourcesoftware</b>	<b>1</b>
<b>1</b>	<b>Introductie van opensourcesoftware</b>	<b>3</b>
	<b>Definitie van broncode</b>	<b>4</b>
	Gesloten bron	5
	Open bron	6
	'Free' software	6
	<b>Kiezen opensourcelicentie</b>	<b>8</b>
	Opties	9
	Sleutelwoorden	9
	Voorbeelden	9
	Bruikbare links	10
	<b>Samenvatting</b>	<b>11</b>
	<b>Deel II: Linux: de basis</b>	<b>13</b>
<b>2</b>	<b>Introductie tot Linux</b>	<b>15</b>
	<b>Wat Linux is</b>	<b>16</b>
	<b>Werken met een Linux-systeem</b>	<b>16</b>
	De juiste distributie kiezen	17
	Inloggen	18
	<b>De GUI gebruiken</b>	<b>21</b>
	<b>Werken met de opdrachtregel</b>	<b>23</b>
	Opbouw opdrachtregel	24
	Hulp krijgen	24
	<b>Samenvatting</b>	<b>30</b>

<b>3</b>	<b>Bestandssysteem</b>	<b>31</b>
	<b>Het bestandssysteem begrijpen</b>	<b>32</b>
	Meestgebruikte directory's	32
	Regels naamgeving	33
	<b>Navigeren in het bestandssysteem</b>	<b>34</b>
	<b>Bestandsbeheer</b>	<b>37</b>
	Directory's beheren	39
	Bestanden beheren	41
	<b>Samenvatting</b>	<b>50</b>
<b>4</b>	<b>Essentiële opdrachten</b>	<b>51</b>
	<b>Opdrachtregelgereedschap</b>	<b>52</b>
	Bestanden bekijken	53
	Bestanden vinden	57
	Bestanden vergelijken	59
	Mogelijkheden van de shell	60
	Rechten	62
	<b>Gereedschap voor de ontwikkelaar</b>	<b>65</b>
	Bestanden comprimeren	65
	De opdracht <code>grep</code>	68
	<b>Samenvatting</b>	<b>73</b>
<b>5</b>	<b>Tekstverwerkers</b>	<b>75</b>
	<b>De tekstverwerker <code>vi</code></b>	<b>76</b>
	Waarom <code>vi</code> leren?	76
	Wat is <code>vim</code> ?	77
	Essentiële opdrachten in <code>vi</code>	78
	<b>Andere tekstverwerkers</b>	<b>88</b>
	Emacs	88
	Gedit en <code>kwrite</code>	90
	Nano en <code>joe</code>	90
	Lime en <code>bluefish</code>	91
	<b>Samenvatting</b>	<b>91</b>
<b>6</b>	<b>Systeembeheer</b>	<b>93</b>
	<b>Belangrijke taken</b>	<b>94</b>
	Krijgen van roottoegang	94
	Harddiskruimte	97
	<b>Software beheren</b>	<b>99</b>
	Overzicht en vinden van software	100
	Software installeren	101

	<b>Gebruikersaccounts</b>	<b>102</b>
	Toevoegen gebruikersaccounts	103
	Aanpassen gebruikersaccounts	104
	Verwijderen gebruikersaccounts	104
	Gebruikersgroepen begrijpen	104
	Gebruikersgroepen beheren	105
	<b>Samenvatting</b>	<b>106</b>
	<b>Deel III: Programmeertalen in Linux</b>	<b>107</b>
<b>7</b>	<b>Programmeertalen in Linux</b>	<b>109</b>
	<b>Soorten programmeertalen</b>	<b>110</b>
	<b>Scripttalen</b>	<b>110</b>
	BASH-scripts	110
	Perl-scripts	112
	Python-scripts	113
	Overige scripttalen	114
	<b>Gecompileerde talen</b>	<b>120</b>
	C	120
	C++	121
	Java	121
	<b>IDE's</b>	<b>121</b>
	<b>Samenvatting</b>	<b>122</b>
<b>8</b>	<b>BASH-scripts</b>	<b>123</b>
	<b>De basis van BASH-scripts</b>	<b>124</b>
	<b>Voorwaardelijke regels</b>	<b>126</b>
	Integers vergelijken	128
	Testen van bestanden	128
	<b>Afhankelijk programmaverloop</b>	<b>129</b>
	while-lus	129
	for-lus	130
	case-blok	130
	<b>Gebruikersinteractie</b>	<b>131</b>
	<b>Meer informatie</b>	<b>132</b>
	<b>Samenvatting</b>	<b>132</b>

<b>9</b>	<b>Perl-scripts</b>	<b>133</b>
	<b>De basis van Perl-scripts</b>	<b>134</b>
	Perl-code uitvoeren	134
	Meer Perl-documentatie	135
	Variabelen en waarden	137
	<b>Afhankelijk programmaverloop</b>	<b>140</b>
	<b>Voorwaarden</b>	<b>142</b>
	<b>Andere mogelijkheden</b>	<b>143</b>
	<b>Samenvatting</b>	<b>145</b>
<b>10</b>	<b>Python-scripts</b>	<b>147</b>
	<b>De basis van Python-scripts</b>	<b>148</b>
	Python-code uitvoeren	149
	Meer informatie	150
	Variabelen en waarden	150
	<b>Programmaverloop</b>	<b>153</b>
	<b>Voorwaarden</b>	<b>154</b>
	<b>Andere mogelijkheden</b>	<b>155</b>
	<b>Samenvatting</b>	<b>156</b>
<b>11</b>	<b>C, C++ en Java</b>	<b>157</b>
	<b>Systeembibliotheken begrijpen</b>	<b>158</b>
	Gedeelde bibliotheken beheren	158
	Gedeelde bibliotheken bekijken	160
	<b>Pakketten maken</b>	<b>161</b>
	RPM-pakket maken	161
	Debian-pakket maken	164
	<b>Java installeren</b>	<b>164</b>
	<b>Samenvatting</b>	<b>165</b>
	<b>Deel IV: Git gebruiken</b>	<b>167</b>
<b>12</b>	<b>De essentie van Git</b>	<b>169</b>
	<b>Dutch of Engels?</b>	<b>170</b>
	<b>Idee achter versiebeheer</b>	<b>170</b>
	Eerste generatie	170
	Tweede generatie	171
	Derde generatie	172

	<b>Git installeren</b>	<b>175</b>
	<b>Idee achter Git</b>	<b>177</b>
	Stappen in Git	177
	Keuze Git-hosting	178
	Git configureren	179
	<b>Samenvatting</b>	<b>182</b>
<b>13</b>	<b>Bestanden beheren met Git</b>	<b>183</b>
	<b>Basale configuratie</b>	<b>184</b>
	<b>De opdracht git status</b>	<b>185</b>
	Problemen met meerdere locaties oplossen	188
	Git een bestand laten negeren	189
	<b>Bestanden verwijderen</b>	<b>191</b>
	<b>Vertakkingen beheren</b>	<b>192</b>
	<b>Vertakkingen overdragen</b>	<b>194</b>
	<b>Samenvatting</b>	<b>195</b>
<b>14</b>	<b>Verschillen tussen bestanden beheren</b>	<b>197</b>
	<b>Verschillen opsporen</b>	<b>198</b>
	Omgaan met witruimte	200
	Vertakkingen vergelijken	201
	<b>Bestanden samenvoegen</b>	<b>203</b>
	<b>Samenvatting</b>	<b>208</b>
<b>15</b>	<b>Geavanceerde mogelijkheden van Git</b>	<b>209</b>
	<b>Opslagruimten beheren</b>	<b>210</b>
	Inhoud ophalen van de opslagruimte op afstand	211
	Verbinden met SSH	213
	<b>Patching</b>	<b>214</b>
	<b>Samenvatting</b>	<b>215</b>
	<b>Index</b>	<b>217</b>

# Voorwoord

In gedachten zag ik het boek als het begin van een reis. Uw startpunt van deze reis kan verschillen van dat van anderen, maar het doel van dit boek is het geven van kennis die u nodig hebt voor het starten van softwareontwikkeling onder het Linux-besturingssysteem.

Sommige lezers zullen al ervaring hebben met softwareontwikkeling op het Windows-platform. Voor hen zal het boek een gids zijn bij de verschillen tussen softwareontwikkeling onder Linux en het vertrouwde platform.

Misschien werkt u al onder Linux, maar wilt u beginnen met het schrijven van code. Ook nu geeft dit boek u een perfect startpunt voor die reis.

Dit boek bevat vier delen.

- Deel I, Opensourcesoftware, bevat één hoofdstuk – hoofdstuk 1, Introductie van opensourcesoftware. U leert wat opensourcesoftware is, inclusief de voordelen boven closedsourcesoftware en de basis van softwarelicenties.
- Deel II, Linux: de basis, is een introductie van het besturingssysteem Linux. Het doel van dit deel is u de kennis aanreiken die softwareontwikkelaars moeten hebben van het werken onder Linux. Dit omvat onderwerpen voor zowel eindgebruikers als beheerders. De hoofdstukken in dit deel zijn:
  - Hoofdstuk 2, Introductie tot Linux. In dit hoofdstuk leert u de basis van Linux, waaronder hoe u toegang krijgt tot een Linux-systeem, hoe u de Linux-GUI gebruikt en het werken met de opdrachtregel.
  - Hoofdstuk 3, Bestandssysteem. Dit hoofdstuk richt zich op hoe bestanden georganiseerd zijn onder Linux. U krijgt een introductie van de ideeën achter het bestandssysteem, hoe u door de bestanden navigeert en hoe u het bestandssysteem beheert.
  - Hoofdstuk 4, Essentiële opdrachten. In dit hoofdstuk leert u meerdere Linux-opdrachten die elke ontwikkelaar moet kennen.
  - Hoofdstuk 5, Tekstverwerkers. Als ontwikkelaar moet u weten hoe bestanden bewerkt worden. Dit hoofdstuk richt zich op vi, een veelvoorkomende tekstverwerker onder Linux en Unix. Tevens worden een aantal andere tekstverwerkers voorgesteld.
  - Hoofdstuk 6, Systeembeheer. Het is voor een ontwikkelaar handig om te weten hoe enkele systeembeheerstaken worden uitgevoerd. In dit hoofdstuk leert u het toevoegen van software en het beheren van gebruikers.

- Deel 3, Programmeertalen in Linux, geeft een overzicht van verschillende programmeertalen die onder Linux beschikbaar zijn. U hebt hierbij veel keuzemogelijkheden. Het doel is niet u alles te leren over elke taal, maar een introductie te geven, zodat u kunt beslissen welke taal u het best bevalt.
  - Hoofdstuk 7, Programmeertalen in Linux. Dit hoofdstuk geeft een overzicht van programmeertalen met als aandachtspunt het begrijpen van de verschillen tussen scripttalen en gestructureerde (of gecompileerde) talen.
  - Hoofdstuk 8, BASH-scripts. In dit hoofdstuk wordt de scripttaal BASH geïntroduceerd. U leert hoe u code schrijft die interactie met gebruikers afhandelt en andere kenmerken van BASH.
  - Hoofdstuk 9, Perl-scripts. Het aandachtspunt van dit hoofdstuk is hoe u programma's in de scripttaal Perl maakt. Onderwerpen zijn onder andere afhankelijk programmaverloop en gebruiken van variabelen.
  - Hoofdstuk 10, Python-scripts. U leert in dit hoofdstuk de basis van Python-scripts, waaronder de grote verzameling Python-variabelen, hergebruik van code en afhankelijk programmaverloop.
  - Hoofdstuk 11, C, C++ en Java. In dit hoofdstuk leert u de essentie van programmeren in C, C++ en Java op een op Linux-gebaseerd systeem.
- Tot slot behandelt deel 4, Git gebruiken, een zeer populair softwareproduct voor versiebeheer met de naam Git. Het gebruik van versiebeheersoftware, zeker wanneer gewerkt wordt met een team van ontwikkelaars, kan u veel tijd, geld en moeite besparen.
  - Hoofdstuk 12, De essentie van Git. In dit hoofdstuk leert u het idee achter Git. Onderwerpen die behandeld worden zijn de ideeën achter versiebeheer, installatie en mogelijkheden van Git.
  - Hoofdstuk 13, Bestanden beheren met Git. In dit hoofdstuk leert u over concepten van Git, zoals opstelruimte, overdracht en vertakkingen.
  - Hoofdstuk 14, Verschillen tussen bestanden beheren. In dit hoofdstuk draait het om hoe u omgaat met verschillende versies van bestanden. U leert hoe u overzicht krijgt in verschillen en hoe u verschillende bestanden samenvoegt.
  - Hoofdstuk 15, Geavanceerde mogelijkheden van Git. U leert in dit hoofdstuk het beheren van Git-opslagruimten en het patchen van bestanden.

Veel succes met uw reis!

William 'Bo' Rothwell  
22 december 2016

# Opensource- software

**E**en van de belangrijkste vragen bij het ontwikkelen van software is onder welke licentie het resultaat wordt gepubliceerd. De zoektocht naar het antwoord kan moeizaam zijn.

U moet besluiten welke bescherming voor uw code nodig is en wat anderen met uw software mogen doen. Dit onderdeel heeft slechts één hoofdstuk en richt zich op de beslissing welke licentie het beste past.

U leert het volgende:

*Het verschil tussen closed- en opensourcesoftware.*

*Uitleg opensourcebescherming.*

*Verschillen tussen de belangrijkste opensourcelicenties.*



# Introductie van opensourcesoftware

**H**et kiezen van de licentie waaronder uw software zal worden verspreid, is niet eenvoudig en dit hoofdstuk geeft ook geen pasklaar antwoord. Wel geeft het u meer inzicht in de verschillen tussen open en gesloten broncode en het begrip 'free'. Ook krijgt u een overzicht van opensourcelicenties, de mogelijkheden en belangrijke terminologie.

U leert in dit hoofdstuk:

*Het verschil tussen closedsourcesoftware en opensourcesoftware.*

*Dat free niet hoeft te betekenen dat software gratis is.*

*Welke typen opensourcelicenties er bestaan.*

*Wat termen als copyleft en permissiveness betekenen.*

## Definitie van broncode

U hebt een geweldig programma gemaakt en nu is het tijd om het beschikbaar te stellen aan het publiek. U moet een belangrijke beslissing nemen: de licentie waaronder de software wordt verspreid.

Deze beslissing heeft een aantal belangrijke gevolgen, waaronder:

- hoe de software gebruikt kan worden;
- is de code publiekelijk beschikbaar of niet;
- kunnen andere programmeurs uw code hergebruiken;
- kunnen anderen het programma verkopen.



### Disclaimer

Licentienuances kunnen complex zijn en een enorme invloed hebben op hoe uw software gebruikt kan en mag worden. De uitleg in dit boek is bedoeld als globale basis om de verschillen tussen licenties te begrijpen, maar is zeker geen juridisch advies. Overweeg altijd om juridisch advies in te winnen voordat u een licentie kiest.

---

Zeer waarschijnlijk is de eerste vraag die u moet beantwoorden: wordt het *closed-sourcesoftware* of *opensourcesoftware*?" Voor het beantwoorden van deze vraag moet u weten wat broncode (*source code*) is.

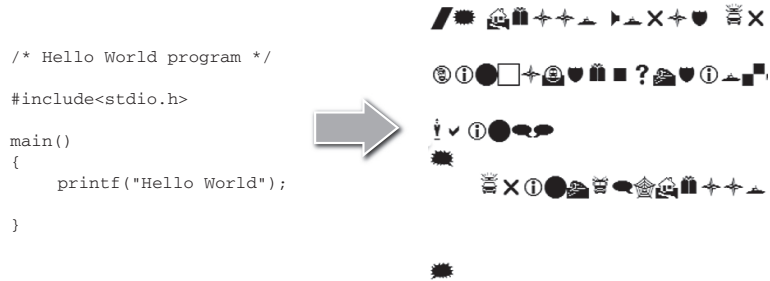
Software bestaat uit regels met instructies geschreven in een programmeertaal. Daarvan bestaan zeer veel varianten, waaronder C, C++, Java, Perl, Python en meer. Alle regels met instructies vormen tezamen de broncode. Zie afbeelding 1.1 voor een voorbeeld van broncode geschreven in C.

```
/* Hello World program */  
  
#include<stdio.h>  
  
main()  
{  
    printf("Hello World");  
}
```

**Afbeelding 1.1** *Broncode geschreven in C.*

Over het algemeen kan de broncode niet direct gebruikt worden om een programma te starten. De meeste talen worden gecompileerd, wat neerkomt op het vertalen van de broncode naar instructies die het besturingssysteem begrijpt. Het resultaat is onleesbaar voor ons, maar gesneden koek voor het

besturingssysteem. Zie afbeelding 1.2 voor een voorbeeld van broncode naar gecompileerde code.



**Afbeelding 1.2** Voorbeeld van source code naar gecompileerde code.

Wanneer u kiest voor closedsourcesoftware komt dit neer op het verspreiden van de gecompileerde code. Opensourcesoftware geeft toegang tot de source code.

## Gesloten bron

Een term die ook wel gebruikt wordt is proprietary (lees: eigendom) software omdat het doel van closedsourcesoftware het geheim houden van de broncode inhoudt. Het idee hierachter is dat anderen de broncode kunnen kopiëren of illegaal gebruiken. Concurrenten kunnen zo een negatieve invloed hebben op de financiële situatie van het bedrijf dat de software heeft ontwikkeld. Het mag duidelijk zijn dat kopiëren van andermans software veel goedkoper is dan zelf ontwikkelen.

Vaak wordt de term gesloten bron (*closed source*) gebruikt als vervanging van commerciële software, maar dat is onjuist. Commerciële software moet voor gebruik aangeschaft worden. Zowel closed- als opensourcesoftware kan commercieel zijn. De gebruikte licentie maakt duidelijk of de software commercieel of 'free'<sup>1</sup> is.

Voorbeelden van closedsourcesoftware zijn:

- Microsoft Windows
- Adobe Photoshop
- Apple Mac OS

1 Er is een reden voor de aanhalingstekens. Zoals u later zult zien moet deze kreet in de context van software juridisch gedefinieerd worden.

### Open bron

Software waarvan de gecompileerde<sup>2</sup> code en de broncode beschikbaar is, wordt meestal open bron (*open source*) genoemd. De softwarelicentie van de copyrighthouder geeft vervolgens aan onder welke voorwaarden de code bekeken, aangepast of verspreid mag worden. Er is een verscheidenheid aan opensourcelicenties beschikbaar om deze rechten vast te leggen.

Hoewel sommige opensourcesoftware gratis is in het economische verkeer, is dat geen vereiste! Open bron verwijst naar de mogelijkheid om de broncode in te zien, niet hoe de software gebruikt mag worden en of daar kosten aan verbonden zijn.

Bekende voorbeelden van opensourcesoftware zijn:

- Linux<sup>3</sup>
- Apache HTTP-server
- Firefox
- Git
- Openoffice.org

### 'Free' software

Het begrip 'free'<sup>4</sup> betekent bij het gebruik van software niet voor iedereen hetzelfde. Sommige mensen zien het als gratis oftewel geen kosten voor het gebruik. Met andere woorden, het kost niets om de software te bemachtigen en te gebruiken.

Maar wat bedoelen we met gebruiken van software? Mag de software op elke manier ingezet worden die de gebruiker wil of zijn er restricties? Mag de software overall ter wereld vrij gebruikt worden of zijn er geografische beperkingen? Bent u vrij om de software aan te passen en deze vervolgens te verspreiden of is dat verboden? U ziet dat 'free' in relatie tot software niet eenduidig is.

- 2 Soms wordt alleen de broncode verspreid en moet de gebruiker deze zelf compileren. Verder zijn er programmeertalen die niet gecompileerd worden en ook dan is alleen de broncode beschikbaar.
- 3 Hoewel bijna iedereen met Linux het complete besturingssysteem bedoelt, is het technisch gezien alleen de kernel oftewel het hart. De overige software om van de Linux-kernel een echt besturingssysteem te maken is vaak ook opensourcesoftware, maar dit is geen vereiste.
- 4 Toevoeging vertaler: In het Engels kan het begrip 'free' zowel vrij als gratis betekenen. De bedoelde betekenis blijkt meestal uit de context. In het Nederlands bestaat weliswaar het begrip vrije toegang, maar dat wordt nauwelijks meer gebruikt. In onze taal is het woord 'vrij' veel eenduidiger dan in het Engels. Daar moet het woord 'free' software dus meer verduidelijkt worden.

Een manier om het begrip ‘free’ te omschrijven is de definitie van Richard Stallman die is gepubliceerd door de Free Software Foundation (FSF):

“Het woord ‘free’ in onze naam verwijst niet naar de prijs; het verwijst naar vrijheid. Als eerste de vrijheid om het programma te kopiëren en te distribueren naar uw burens, zodat ook zij het kunnen gebruiken. Ten tweede de vrijheid om een programma aan te passen zodat u beslist hoe het werkt in plaats van andersom; hiervoor moet de broncode beschikbaar zijn.”

In deze definitie is ‘free’ software gelijk aan opensourcesoftware. Niet iedereen is het hiermee eens en daarom kunt u bijvoorbeeld closedsourcefreeware downloaden.

Een andere manier om het woord ‘free’ voor software te definiëren zijn de vier vrijheden zoals gepubliceerd door de FSF:

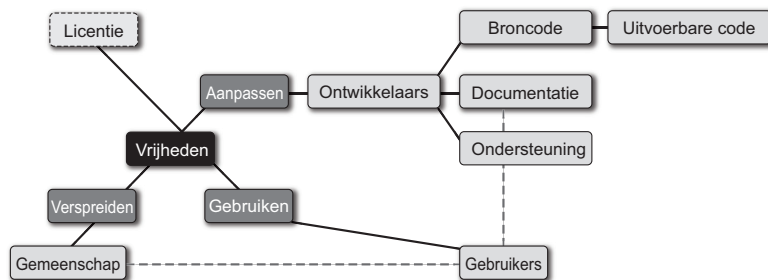
- **Vrijheid 0** De vrijheid om het programma te gebruiken zoals u wilt en voor elk doel.
- **Vrijheid 1** De vrijheid om te bestuderen hoe het programma werkt en aan te passen zodat het aan uw eisen voldoet. Beschikbaarheid van de source code is dan een vereiste.
- **Vrijheid 2** De vrijheid om kopieën te verspreiden om zo uw burens te helpen.
- **Vrijheid 3** De vrijheid om een aangepaste versie te verspreiden zodat iedereen van deze veranderingen gebruik kan maken. Beschikbaarheid van de broncode is dan een vereiste

Deze vier vrijheden vormen de basis van het begrip FOSS (Free and Open Source Software).<sup>5</sup> FOSS doet een goede poging om het begrip ‘free’ software te definiëren. Let vooral op het feit dat niet alle vier de vrijheden om ‘free’ software te definiëren gelijk staan aan open source. Dat betekent tevens dat opensourcesoftware een licentie kan hebben met slechts een deel van de vier vrijheden.

Begrijpen wat het begrip opensourcesoftware betekent en de rol die vrijheid speelt, zal de nodige tijd kosten. Zie afbeelding 1.3 voor de samenhang tussen de begrippen die opensourcesoftware beschrijven.

Afbeelding 1.3 belicht de complexe combinatie van begrippen die samen opensourcesoftware beschrijven. U ziet dat ontwikkelaars broncode schrijven, documentatie maken en ondersteuning geven. Gebruikers van de software hebben echter vaak ook een grote rol in dit proces. Sterker nog, sommige opensource-

5 Vaak in dezelfde betekenis gebruikte kreet is FLOSS (Free/Libre Open Source Software).



**Afbeelding 1.3** Weergave van de opensourcegemeenschap.

softwareprojecten krijgen nauwelijks documentatie of ondersteuning van de ontwikkelaars en zijn voor deze essentiële zaken compleet afhankelijk van de gemeenschap.

U ziet dat de vrijheden voor aanpassen, gebruiken en verspreiden ook in afbeelding 1.3. zijn opgenomen. De licentie legt deze vast.

## Kiezen opensourcelicentie

Uiteindelijk moet u bepalen of de software als gesloten of open bron gelicentieerd wordt. Deze beslissing worden deels beïnvloed door de nuances van verschillende licenties. Dat wordt in de rest van dit hoofdstuk belicht. Overweeg echter ook een paar van de algemene voordelen die het ontwikkelen van opensourcesoftware heeft:

- **Open source lijkt meer vertrouwen te geven** De reden is dat anderen precies kunnen nagaan wat de software doet door het bekijken van de broncode.
- **Open source kan resulteren in betere code en minder ontwikkeltijd** Omdat andere ontwikkelaars uw code bekijken kunnen zij commentaar geven, bugs oplossen en verbeteringen voordragen. Dat alles vaak zonder extra kosten voor u.
- **Open source die 'free' is, kan meer gebruikers opleveren** Meer gebruikers zijn bereid om software uit te proberen als voor het testen niet betaald hoeft te worden.
- **U kunt nog steeds geld verdienen aan 'free' opensourcesoftware** Andere bronnen van inkomsten zijn training, onderhoudscontracten en extra diensten.

## Opties

Er bestaan tientallen standaard opensourcelicenties en verder een groot aantal op maat gemaakte. Over het algemeen vallen ze in de volgende vier categorieën:

- **Standaard** Dit zijn licenties die vaak worden hergebruikt voor andere software. Meestal zijn ze specifiek voor een land en vele zijn gebaseerd op de wetgeving van de Verenigde Staten of Europa.
- **Internationaal** Dit zijn licenties die vaak worden hergebruikt voor andere software. In tegenstelling tot de standaardlicenties zijn deze gemaakt om over de hele wereld te gebruiken.
- **Speciale toepassing** Licenties die zijn gemaakt voor speciale omstandigheden.
- **Niet herbruikbaar** Licenties die expliciet het gebruik voor andere software verbieden. Ofwel alleen te gebruiken bij de software waarvoor ze is bedacht.

## Sleutelwoorden

Wat betreft opensourcelicenties zijn er een aantal sleutelwoorden die u moet kennen. Bijvoorbeeld *copyleft* om intellectueel eigendom (Engels: Intellectual Property of IP) als opensourcesoftware te kunnen kopiëren en verspreiden. De twee vormen van copyleft zijn:

- **Sterk** Elk afgeleid werk moet de originele copyleft bevatten.
- **Zwak** Afgeleid werk hoeft de originele restricties niet over te nemen.

Een ander sleutelwoord in opensourcelicenties is *permissiveness* (verdraagzaamheid). Dit heeft voornamelijk betrekking op afgeleid werk en of dat onder een andere licentie verspreid mag worden. De twee mogelijkheden zijn:

- **Strikt** Beperkingen aan andere licenties (geen gesloten bron of meer toestaan).
- **Tolerant** Andere licenties zijn toegestaan

## Voorbeelden

De volgende lijst beschrijft een aantal van de populairste opensourcelicenties:

- Apache License 2.0:
  - zeer verdraagzaam
  - geen copyleft
  - bruikbaar voor elk doel
  - verspreiden en aanpassen
  - staat afgeleid werk toe

- MIT License:
  - ook bekend als X11-licentie
  - vergelijkbaar met de Apache License 2.0
  - zeer tolerant
  - geen copyleft
  - bruikbaar voor elk doel
  - originele copyright moet in stand blijven
  - gebruiker moet instemmen met geen garantie
- GNU General Public License (GPL):
  - sterke copyleft
  - niet verdraagzaam
  - afgeleid werk moet eveneens GPL gebruiken
  - twee versies: v2 en v3
- BSD License:
  - zeer verdraagzaam
  - geen copyleft
  - drie typen:
    - twee voorwaarden – zelfde als MIT
    - drie voorwaarden – afgeleid werk wordt niet ondersteund door de originele ontwikkelaar
    - vier voorwaarden – in advertenties moet de originele ontwikkelaar erkend worden

### Bruikbare links

Hopelijk heeft u nu een beter begrip wat opensourcesoftware en bijbehorende licenties inhouden. Het mag duidelijk zijn dat dit geen eenvoudig onderwerp is en dat meer tijd aan het zoeken naar de juiste licentie voor uw software en uw bedrijf goed besteed is. Naast juridisch advies kunnen de volgende links mogelijk behulpzaam zijn:

- **<http://choosealicense.com>** Deze website gebruikt een serie vragen om u te helpen bij het kiezen van de beste licentie in uw situatie. Het geeft een goede start, maar u moet nog steeds juridisch advies inwinnen voordat u een definitieve beslissing neemt.
- **<http://fsf.org>** De website van de Free Software Foundation (FSF) geeft veel nuttige informatie over opensourcesoftware en licenties.
- **<http://opensource.org>** Een andere bron om meer te leren over open source software en licenties.



Open Source Humor

Open source: free as in 'free speech' not as in 'free beer' – Anoniem.

---



## Samenvatting

In dit hoofdstuk hebt u de verschillen tussen closed- en opensourcesoftware geleerd. Het concept 'free' software is eveneens aan bod gekomen. Tot slot heeft u de basis van opensourcelicenties gezien. Op dit moment moet u in staat zijn om het keuzeproces voor de licentie waaronder uw software verspreid gaat worden te starten. Bedenk echter dat u veel tijd, energie en denkwerk in deze stap moet stoppen, want de gekozen licentie heeft een enorme invloed op hoe uw software door de gemeenschap gebruikt gaat worden.