

Leren programmeren met Python

Leren programmeren met Python

Robert R. Saunders

Boom beroepsonderwijs · Meppel · 2020

Boom beroepsonderwijs
info@boomberoepsonderwijs.nl
www.boomberoepsonderwijs.nl

Auteur: Robert R. Saunders
Redactie en opmaak: Henk Pel
Titel: Leren programmeren met Python
ISBN 978 90 372 5746 5
Eerste druk / eerste oplage
© Boom beroepsonderwijs 2020

Behoudens de in of krachtens de Auteurswet gestelde uitzonderingen mag niets uit deze uitgave worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voor zover het maken van reprografische verveelvoudigingen uit deze uitgave is toegestaan op grond van artikel 16h Auteurswet dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (www.reprorecht.nl). Voor het overnemen van gedeelte(n) uit deze uitgave in compilatiewerken op grond van artikel 16 Auteurswet kan men zich wenden tot de Stichting PRO (www.stichting-pro.nl).

De uitgever heeft ernaar gestreefd de auteursrechten te regelen volgens de wettelijke bepalingen. Degenen die desondanks menen zekere rechten te kunnen doen gelden, kunnen zich alsnog tot de uitgever wenden.

Door het gebruik van deze uitgave verklaart u kennis te hebben genomen van en akkoord te gaan met de specifieke productvoorwaarden en algemene voorwaarden van Boom beroepsonderwijs, te vinden op www.boomberoepsonderwijs.nl.

Inhoud

Deel een **Introductie**

Voorwoord 1

Waarom Python? 1

1 Python installeren 5

Microsoft Windows 5

Mac™ OS 8

Ubuntu Linux 11

2 Interactieve Python-interpretersessie 13

Read-Evaluate-Process-Loop 13

Handige commando's in de REPL 13

exit() 13

help() 14

3 Rekenen met Python 15

De REPL als rekenmachine 15

Een som maken 15

Integers en floating points 16

Nog meer sommen – Python v2.x versus v3.x 17

Prioriteiten 17

Opgaven 3 19

4 Onthouden met Python 21

Variabelen 21

Regels voor variabelen 21

Conventies voor variabelen 23

Opgaven 4 23

5 Verschillende soorten data 24

Datatypes 24

Integers 24

Floating points 24

Strings 25

Booleans 26

	Converteren	27
	<i>Van floating point naar integer</i>	27
	<i>Van integer naar floating point</i>	28
	<i>Van een datatype naar string</i>	28
	Opgaven 5	29
6	Strings	30
	Strings als documentatie	30
	Strings formatteren	31
	<i>Speciale karakters</i>	33
	<i>Uitlijnen binnen een tekstblok</i>	34
	Opgaven 6	35
7	List	36
	Verschillende soorten data	36
	Elementen toevoegen	38
	<i>Append</i>	38
	<i>Insert</i>	39
	<i>Extend</i>	40
	Elementen verwijderen	42
	<i>Remove</i>	42
	<i>Pop</i>	43
	Snijden	44
	<i>Impliciet snijden</i>	45
	Tuple	48
	Opgaven 7	49
8	Een programma maken	50
	Lege ruimte	50
	Invoer van de gebruiker	51
	<i>Invoer van getallen</i>	52
	Beslissingen: als dit, dan dat	53
	<i>Een voorbeeld met beslissingen</i>	54
	Opgaven 8a	55
	Herhalingen	56
	<i>Herhalingen met for</i>	57
	<i>For met een index</i>	58
	<i>For onderbreken met break</i>	59
	<i>For met continue door laten gaan</i>	61
	<i>Herhalingen met while</i>	63
	<i>While onderbreken met break</i>	64
	<i>While met continue door laten gaan</i>	66
	Opgaven 8b	68

- 9 Bestanden schrijven en lezen 70**
 - Folders en bestanden 70
 - Alleen folders weergeven 71
 - Alleen bestanden weergeven 72
 - Folderpaden 72
 - Folderstructuur 73
 - Schrijven 74
 - Lezen 76
 - Bestandsinformatie 77
 - Opdracht 9 – Bestand schrijven 78
- 10 De Zen van Python 79**

Deel twee Gaming

- 1 Games 83**
- 2 Integrated Development Environment 84**
 - PyCharm 84
 - Installeren 84*
- 3 Een game 103**
 - Game play/mechanics 103
 - Hoger/Lager spelregels 103
 - Stappenplan 104
 - Een nieuw project met PyCharm 104
 - Virtual environment 105*
 - Het stappenplan in code 108
 - Stap 1 108*
 - Stap 2 109*
 - range() 111*
 - Stap 3 112*
 - Stap 4 113*
 - Stap 5 114*
 - Stap 6 119*
 - Stap 7 121*
 - De game nog een keer bekeken 121
 - Opgaven 3 121
- 4 Game engine 123**
 - Installeren van PyGame 124
 - PyGame 128
 - Lineaire animatie 136
 - Opgaven 4 138

- 5 **Vliegtuig 139**
 - Stappenplan 139
 - Afbeeldingen 140
 - Roteren van het vliegtuig 142
 - Functies 145
 - Parameters* 145
 - Naamgeving* 146
 - Functiedefinitie* 146
 - Documentatiestrings 146
 - Herstructureren (refactoring) 147
 - Opgaven 5 150

- 6 **Vliegen 151**
 - Toelichting: goniometrie 151
 - Berekenen nieuwe positie van het vliegtuig 152
 - Besturing 155
 - Realistisch 157
 - Toelichting: spelen met realisme* 161
 - Herstructurering (refactor) 162
 - Vliegroute 165
 - Waypoint toevoegen* 165
 - Waypoint weergeven* 168
 - Waypoint bereikt* 172
 - Opgaven 6 180

- 7 **Waypoint aanvliegen 182**
 - Stappenplan 182
 - Modules 183
 - import* 183
 - Alias* 184
 - Pad naar module* 184
 - Refactor naar modules 185
 - Initialisatie* 185
 - Grafisch* 190
 - main()* 194
 - Afloop van de main* 195
 - Waypoint-generator 196

- 8 **Een state-machine 198**
 - Toelichting: wat is een state? 198
 - Grafische weergave van een state* 199
 - State-machine in code* 200

Game-states	205
<i>Toelichting Dictionaries</i>	206
<i>Weergave</i>	209
Interactie game states	223
<i>Game state Start Menu</i>	223
<i>Game state Vliegen</i>	224
<i>Vliegen – Pauze</i>	239
Head Up Display	242
<i>Beperken speelveld</i>	243
<i>Score, waypoints, level en tijd</i>	248
Game-over	259
<i>Module game_over.py</i>	259
<i>Refactor waypoint_main_game.py</i>	260
<i>Refactor grafisch.py</i>	260
Levels	263
Geluid toevoegen	266
<i>PyGame-muziek</i>	267
<i>PyGame-geluiden</i>	269
Visuele elementen	271
<i>Refactor initialisatie.py</i>	272
<i>Refactor grafisch.py</i>	273
Einde game	276

Deel drie Raspberry Pi

1	Hardware	279
	Raspberry Pi installeren	279
	Installeren Visual Studio Code	279
	Sense HAT	286
2	Software	287
	Installeren SenseHat-bibliotheek	287
	<i>Simulatie</i>	288
	Initialiseren van de Sense HAT	288
3	Sensoren	290
	Omgevingssensoren	290
	<i>Nog een keer stringformatting</i>	291
	Bewegingsensoren	292
	<i>Kleuren met sensoren</i>	293
	<i>Configuratie bewegingsensoren</i>	296
	Joystick	297

- 4 **LED-matrix 299**
 - Punten 299
 - Stappenplan 299*
- 5 **Thermometer met geschiedenis 307**
 - Stappenplan 307*
- 6 **Communicatie 314**
 - Netwerkprogrammering 314
 - Berichtgeoriënteerde communicatie 314
 - Installeren PyZMQ 314
 - Raspberry Pi 315*
 - Computer PyCharm 315*
 - Vraag/antwoordcommunicatie 317
 - De berichten 317*
 - Server 318*
 - Client 320*

Deel vier Objectgeoriënteerd programmeren

- 1 **Objectgeoriënteerd 325**
 - Concepten van objectgeoriënteerd programmeren 325
 - Encapsulation 325*
 - Inheritance 326*
 - Polymorphism 326*
 - Voordelen 327*
 - Nadelen 327*
 - Objecten, klassen en instanties 327
 - Objecten 327*
 - Klassen 328*
 - Instanties 328*
 - Connecties 328*
- 2 **Netwerktekenprogramma 330**
 - Kandidaatobjecten 330
 - Ontleden van de kandidaatklassen 330*
 - Visuele weergave 331
 - Toelichting – Een klasse in Python 332
 - Methoden 333*
 - Ontwerpbeslissingen 336
 - Hoofdprogramma en de TekenApplicatie 337
 - Programma main.py 337*
 - Programma teken_applicatie.py 338*

- Schermpopbouw 340
 - Programma canvas.py* 340
 - Programma netwerkcomponent.py* 342
 - Herstructureer canvas.py* 346
- Selecteren van een netwerkcomponent 348
 - Herstructureer tekenapplicatie.py* 349
 - Herstructureer canvas.py* 350
 - Herstructureer netwerkcomponent.py* 351
- Slepen netwerkcomponent 353
 - Herstructureer teken_applicatie.py* 354
 - Herstructureer canvas.py* 355
- Laten vallen van een netwerkcomponent 357
 - Herstructureer canvas.py* 358
- Verwijderen van een netwerkcomponent 360
 - Herstructureer teken_applicatie.py* 360
 - Herstructureer canvas.py* 361
 - Herstructureer netwerkcomponent.py* 364
- Verbindingen maken 366
 - Statusinformatie* 366
 - Herstructureer canvas.py* 371
 - Herstructureer teken_applicatie.py* 374
 - Herstructureer canvas.py* 375
 - Herstructureer canvas.py deel 2* 379
 - Herstructureer netwerkcomponent.py* 384
- Hulp bij verbinden 386
 - Herstructureer canvas.py* 387
 - Herstructureer netwerkcomponent.py* 390
- Herstructureer methode verwijderen van Canvas-klasse 393
 - Toelichting klasse- en instantievariabele* 395
 - Herstructureer canvas.py* 397
- Overerven server, internet, router, switch en computer 399
 - Toelichting – Overerven* 399
 - Aanmaken Server-klasse* 403
 - Herstructureer canvas.py* 404
 - Herstructureer netwerkcomponent.py* 407
- Opslaan van de tekening 410
 - Herstructureer teken_applicatie.py* 410
 - Herstructureer canvas.py* 412
 - Herstructureer netwerkcomponent.py* 414
- Inlezen van de opgeslagen tekening 416
 - Herstructureer teken_applicatie* 416
 - Herstructureer canvas.py* 417

Appendix

Begrippen en termen 424

Bronvermelding 428

Gebruikte software 428

Gebruikte bibliotheken 428

Informatieve websites 428

Gebruikte afbeeldingen 428

Voorwoord

In dit boek leren we aan de hand van praktische voorbeelden de programmeertaal Python. Er wordt geen voorkennis op het gebied van programmeren vereist. Wel wordt er praktische inzet verwacht. Het leren van een programmeertaal is theoretisch vrijwel niet te doen. Je moet de code werkelijk typen en leren van de fouten die je daarbij maakt. Van kopiëren en plakken leer je niets.

Bij elk onderdeel worden er opdrachten gegeven, zodat de stof kan worden geoefend. Met de opdrachten wordt getracht om realistische voorbeelden uit de praktijk weer te geven.

Ik wil iedereen bedanken voor de steun, wijze raad en feedback tijdens het maken van dit boek. In het bijzonder mijn vrouw Charmène en onze zoon Er Le voor hun steun en geduld, terwijl ik druk bezig was met de research en het schrijven van dit boek.

Waarom Python?

Python is een krachtige, objectgeoriënteerde taal die makkelijk te leren is. De taal groeit hard in populariteit, zowel bij beginnende ontwikkelaars als bij professionals. Python draait op meerdere platforms, waaronder Microsoft Windows, Linux en Mac OS. Een standaardinstallatie van Python is erg compleet, waardoor je al heel veel functionaliteit tot je beschikking hebt. Je hebt direct de beschikking over zeer complete bibliotheken voor de ontwikkeling van de meest uiteenlopende applicaties. Vandaar dat ze zeggen: Python, batteries included.

De taal is ontworpen door de Nederlander Guido van Rossum en uitgebracht in 1991. De belangrijkste eis was een hoge leesbaarheid van de code. Dit wordt vooral bereikt door de zogenaamde 'whitespace'* en een eenvoudige krachtige syntax.

Grote bedrijven als Google, Instagram, DropBox en Mozilla gebruiken Python voor hun backend- en/of frontend-applicaties. In de wereld van de wetenschap wordt Python veel gebruikt voor onderzoeksprojecten, big-data-analyse enzovoort. In de servicemanagementwereld wordt het door DevOps vooral als scripting-taal gebruikt voor het automatiseren en beheren van enorme serverparken, bijvoorbeeld in de cloud.

De lage instapdrempel van Python zorgt ervoor dat je eenvoudig krachtige scripts kunt maken. Wanneer de applicatie complexer wordt en meer functionaliteit vereist, kun je eenvoudig opschalen naar modulair programmeren. Nog een stap

* Woorden met een * worden verklaard in de Appendix: Begrippen en termen.

verder is objectgeoriënteerd programmeren. Het is mogelijk om deze ontwikkelmethodes door elkaar te gebruiken. Je kunt daardoor van zeer eenvoudige scripts tot en met complete objectgeoriënteerde enterprise-applicaties* ontwikkelen.

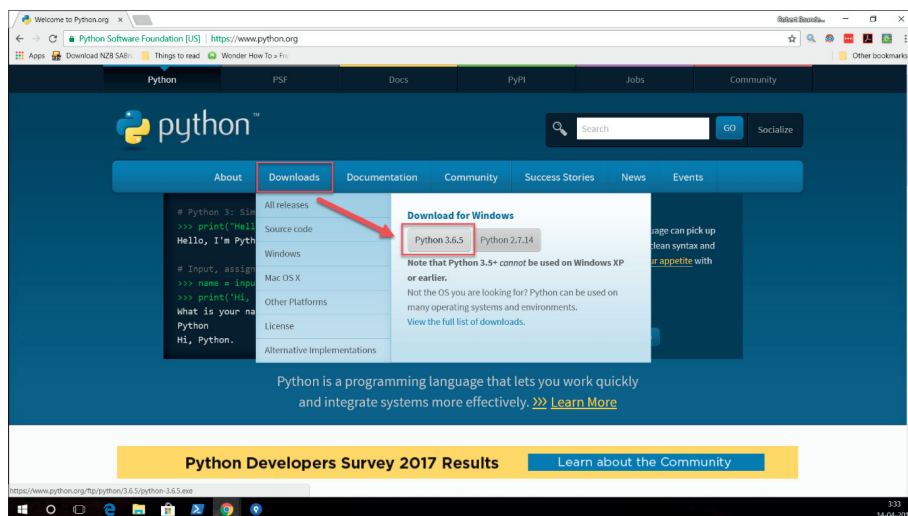
Deel een Introductie

1 Python installeren

Microsoft Windows

Het installeren van Python onder Windows gaat in een aantal stappen. Eerst wordt de laatste Python-versie van de Python-website gedownload. Ga daarvoor naar www.python.org.

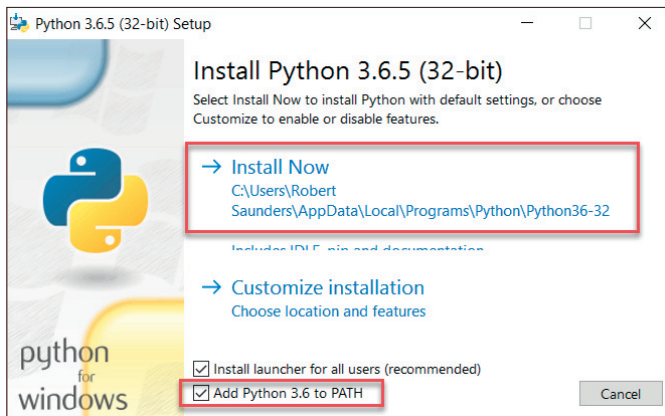
Via het menu *Downloads* kun je de download-button voor de 3.x.x Python-versie kiezen. Dit was ten tijde van het schrijven van het boek versie 3.6.5.



Figuur 1 – Download-knop op de Python-website

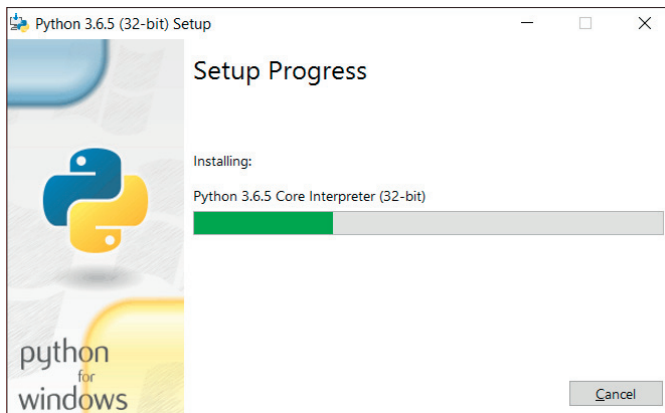
Ga naar de locatie waar het Python-installatiebestand, Python-3.6.5.exe, is opgeslagen en start het bestand met administrator-rechten.

Plaats een vink bij *Add Python 3.6 to PATH* en klik vervolgens op *Install Now*.



Figuur 2 – Het installeren van Python

Python wordt geïnstalleerd. De voortgangsindicator laat zien hoe ver het installatieproces is.



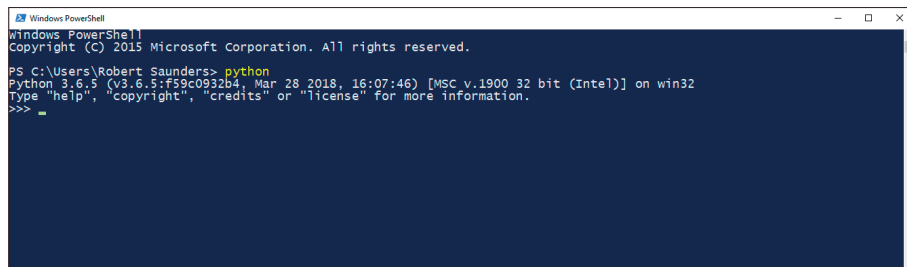
Figuur 3 – Voortgang Python-installatie

De installatie is succesvol doorlopen. Klik op *Close* om de setup af te sluiten.



Figuur 4 – Installatie is succesvol voltooid

Start PowerShell en typ op de command prompt *python.exe*. PowerShell start de Python-interpreter.

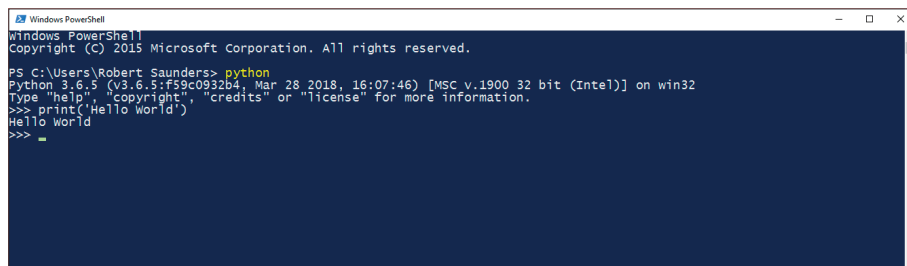


Figuur 5 – PowerShell met de Python-interpreter

De Python-interpreter is te herkennen aan de command prompt `>>>`. Vanaf hier kunnen we Python-commando's geven en Python-programma's en Python-applicaties starten.

Om aan te tonen dat het werkt typen we het volgende klassieke eerste program-ma-voorbeeld:

```
print("Hello World")
```



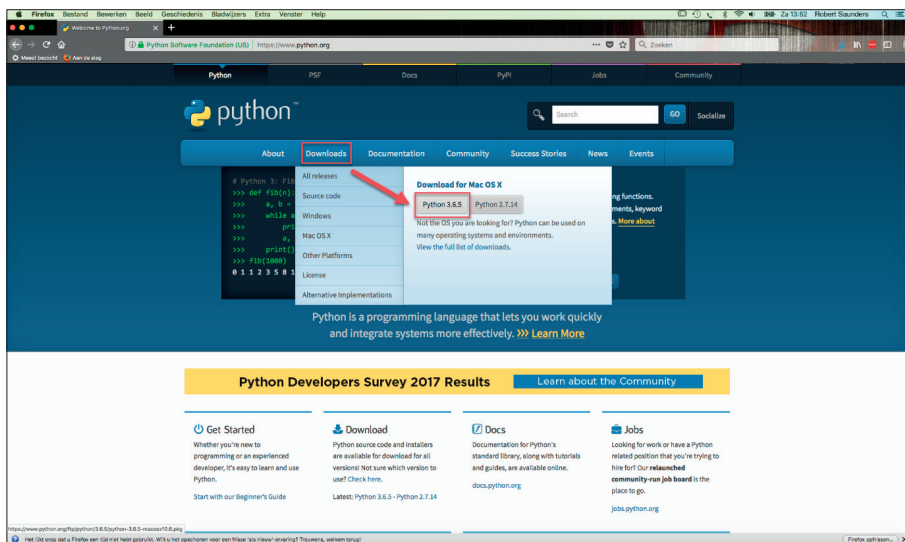
Figuur 6 – Eerste Python-programma

Mac™ OS

Standaard heeft de Mac Python al geïnstalleerd. Dit is echter versie 2.x. In dit boek gebruiken we versie 3.x. Versie 3.x kan zonder problemen naast de standaardinstallatie worden geïnstalleerd.

Het installeren van Python versie 3.x onder Mac OS gaat in een aantal stappen. Eerst wordt de laatste Python-versie van de Python-website gedownload. Ga daarvoor naar www.python.org.

Via het menu *Downloads* kun je de download-button voor de 3.x.x Python-versie kiezen. Dit was ten tijde van het schrijven van het boek versie 3.6.5.



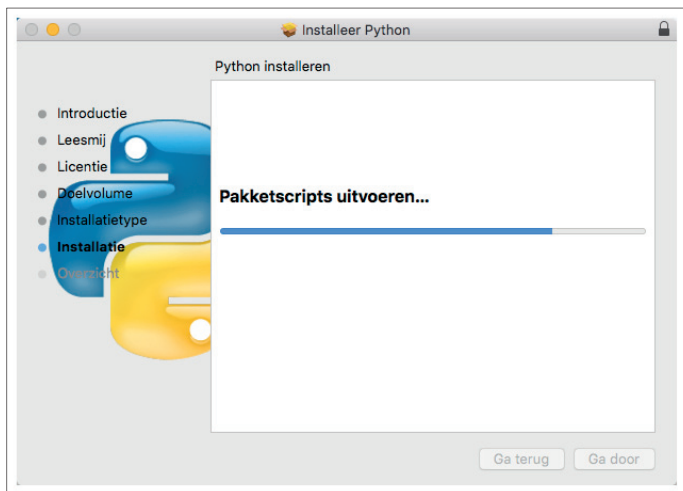
Figuur 7 – Download-knop op de Python-website

Ga naar de locatie waar het Python DMG-installatiebestand is opgeslagen en start de installatie.



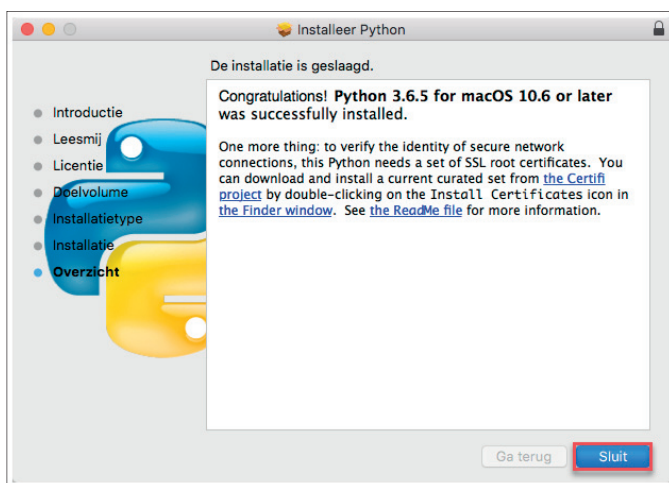
Figuur 8 – Het installeren van Python

Python wordt geïnstalleerd. De voortgangsindicator laat zien hoe ver het installatieproces is.



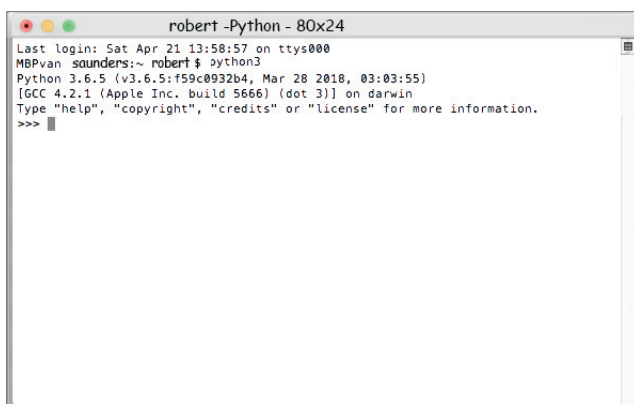
Figuur 9 – Voortgang Python-installatie

De installatie is succesvol verlopen. Klik op *Close* om de setup af te sluiten.



Figuur 10 – Installatie is succesvol voltooid

Start Terminal en typ op de command prompt `python3`. De terminal start de Python-interpreter.



Figuur 11– Shell met de Python-interpreter

De Python-interpreter is te herkennen aan `>>>` als command prompt. Hier kunnen we Python-commando's geven en Python-programma's en Python-applicaties starten.

Om aan te tonen dat het werkt typen we het volgende klassieke eerste programma-voorbeeld:

```
print("Hello World")
```