

Uitgever: Twigt D4
Drukwerk: bravenewbooks.nl

© 2019 Arie Twigt

Tweede druk (paperback) april 2019
Reviewer 1: Rosemarijn Twigt
Reviewer 2: Arno Broekhof
Omslagontwerp: Arie Twigt
Omslagillustratie: Arie Twigt
Foto Auteur: Rosemarijn Twigt

ISBN 9789402185652 (paperback)

<https://www.arietwigt.nl>

Code – Tijd om te gaan programmeren in Python

INHOUDSOPGAVE

VOORWOORD	5
INLEIDING	8
VOOR WIE DIT BOEK IS BEDOELD	9
WAT LEER JE IN DIT BOEK.....	12
HOE DIT BOEK WERKT.....	14
PYTHON TEN OPZICHT VAN ANDERE PROGRAMMEERTALEN.....	14
EEN KORTE TOUR DOOR HET BOEK	16
0 COMMAND LINE FUNDAMENTALS.....	18
1. INSTALLATIE	21
PYTHON INSTALLEREN	22
PIP PYTHON PACKAGE MANAGER.....	25
VIRTUALENV INSTALLEREN	27
HET PROJECT OPZETTEN	28
GIT INSTALLEREN.....	31
SUBLIME TEXT INSTALLEREN	34
JUPYTER NOTEBOOK INSTALLEREN EN GEBRUIKEN.....	35
SAMENVATTING	39
2 PYTHON-SCRIPTS ONTWIKKELEN EN GEBRUIKEN.....	41
EEN PYTHON SCRIPT.....	42
TEXT EDITORS EN IDE'S.....	44
EEN PYTHON-SCRIPT UITVOEREN.....	54
SAMENVATTING	57
OEFENINGEN.....	58
3 KENNISMAKING MET DE PYTHON-SYNTAX.....	59
VARIABLEN AANMAKEN IN PYTHON	60
GEBRUIK MAKEN VAN COMMENTS.....	61
DATATYPEN EN STRUCTUREN	61
BASIC DATABEWERKINGEN IN PYTHON.....	62
CALCULATIES IN PYTHON	68
UITGEBREIDE CALCULATIES EN STATISTISCHE FUNCTIES IN PYTHON	69
WERKEN MET DATUMS	70
IF-ELSE STATEMENTS.....	78
FUNCTIES GEBRUIKEN.....	80
LOOPS IN PYTHON.....	81
SAMENVATTING	83
OEFENINGEN.....	84
4 PYTHON-MODULES GEBRUIKEN.....	85
MODULES INSTALLEREN MET PIP	86
MODULES GEBRUIKEN	87
ONDERDELEN VAN EEN MODULE GEBRUIKEN	92
EEN FUNCTIE UIT EEN EIGEN SCRIPT IMPORTEREN	92
EEN WORD-BESTAND GENEREREN MET DE DOCX MODULE	96
SAMENVATTING	104
OEFENINGEN.....	104
5 WERK BESPAREN DOOR FUNCTIES EN LOOPS TE GEBRUIKEN IN PYTHON.....	105
EEN FUNCTIE SCHRIJVEN.....	106
WAAR HET ALLEMAAL BIJ ELKAAR KOMT.....	116
SAMENVATTING	123
OEFENINGEN.....	124
6 WERKEN MET TABELLEN EN DATA FRAMES IN PYTHON	125

DE PANDAS MODULE IN PYTHON	126
EEN DATA FRAME MAKEN MET DE PANDAS MODULE	126
DATA FRAME BEWERKINGEN	128
DATAFRAMES RANGSCHIKKEN	134
CALCULATED COLUMNS MAKEN IN DATA FRAMES	135
DATATYPEN IN DATAFRAMES	136
SAMENVATTING	141
OEFENINGEN.....	142
7 VERSIE-CONTROLE MET GIT	144
BEGINNEN MET GIT	145
WIJZIGINGEN DOORVOEREN MET GIT COMMIT	151
REMOTE REPOSITORIES.....	154
REPOSITORIES CLONEN.....	158
BRANCHING.....	160
MERGEN IN GIT.....	164
OVERIGE ONDERWERPEN IN GIT	168
SAMENVATTING	169
OEFENINGEN.....	169
8 WERKEN MET API'S EN WEBSERVICES IN PYTHON.....	170
TWITTER API	171
DE WEATHER API GEBRUIKEN OM INFORMATIE OVER HET WEER OP TE HALEN	177
CRYPTOCURRENCY DATA OPHALEN IN PYTHON	180
SAMENVATTING	186
OEFENINGEN.....	186
9 OBJECT GEORIËNTEERD PROGRAMMEREN IN PYTHON.....	187
EEN OBJECT AANMAKEN IN PYTHON	188
DEFAULT ATTRIBUTES.....	191
METHODS	192
METHODS TOEVOEGEN MET PARAMETERS.....	195
UITGEBREIDE METHODS.....	197
CLASS INHERITANCE	200
SAMENVATTING	206
OEFENINGEN.....	207
10 EEN WEB INTERFACE MAKEN	208
EEN WEB INTERFACE.....	209
WEERSVOORSPELLINGEN OPHALEN	219
VARIABLEN DOORGEVEN IN FLASK TEMPLATES MET JINJA TEMPLATING.....	232
PARAMETERS GEVEN AAN EEN FUNCTIE.....	236
DE FRONTEND EEN MOOI DESIGN GEVEN MET BOOTSTRAP	246
STYLING TOEPASSEN OP EEN PANDAS DATA FRAME IN FLASK	252
EMOJI'S TOEVOEGEN AAN DE TABEL MET WEERSVOORSPELLINGEN.....	256
11 EEN CHATBOT MAKEN MET PYTHON	266
DE BENODIGDE MODULES EN SOFTWARE INSTALLEREN	267
HET PYTHON SCRIPT SCHRIJVEN OM DE BOT TE CONFIGUREREN	272
FUNCTIES MET INPUT AAN DE BOT GEVEN	276
WERK MET EEN SCRIPT IN PLAATS VAN MET EEN NOTEBOOK	278
PERSOONLIJKE INSTELLINGEN VOOR DE BOT.....	278
SAMENVATTING	282
ANTWOORDEN OP OEFENINGEN	283

Voorwoord

Allereerst wil ik jullie bedanken dat jullie de tijd willen nemen om aan de hand van dit boek kennis te maken met programmeren in Python. Voor de mensen die al ervaring hebben, bedankt voor jullie tijd en nederigheid om dit boek erbij te pakken. Voor beide type lezers ben ik ervan overtuigd dat ik jullie in dit boek kan inspireren om met Python code prachtige eindproducten te maken. We leven we in een prachtige tijd waarbij we dichterbij zijn dan ooit om onze dromen, qua software, werkelijkheid te maken. Met de huidige mogelijkheden zoals de beschikbare open source software en zeer behulpzame communities is het veel prettiger om software te schrijven of om nieuwe skills te leren.

Nu ben ik niet iemand die graag met oneliners strooit, maar zinnen als *fasten your seatbelts* spoken door mijn hoofd bij het schrijven van deze paragraaf. Met dit boek staat je een mooi avontuur te wachten tijdens het leren van een nieuwe skill, programmeren. Ongetwijfeld zullen er tijdens het doornemen van dit boek nieuwe ideeën in je hoofd schieten om aan te werken. In dit boek heb ik slechts een aantal voorbeelden gegeven van geweldige eindproducten die je kunt bouwen met Python. Gaan jouw ideeën nog verder dan deze voorbeelden? Dan heb je met Python de goede programmeertaal gekozen. In mijn carrière als Consultant, Freelance Software Engineer, Machine Learning Engineer en Data Scientist ben ik geen enkele programmeertaal zo vaak tegengekomen als Python.

Ik wil mijn twee reviewers, Rosemarijn Twigt en Arno Broekhof, heel erg bedanken voor de tijd die zij hebben gestoken in het reviewen van dit boek. Rosemarijn Twigt is, al voor dat ik met haar getrouwd ben, mijn trouwe reviewer voor zo goed als alles dat ik publiceer. Ook bij mijn vorige boek *Data Analyseren en Programmeren met R*. Van mijzelf weet ik dat ik geen ster ben in

de Nederlandse taal en al helemaal niet in de Nederlandse spelling. Ik wil haar daarom heel erg bedanken voor het opsporen van de spel- en stijlfouten in dit boek waardoor het prettiger leesbaar is. En geloof me, hier heeft ze een zware kluit aan gehad.

Arno heb ik leren kennen tijdens mijn tweede gesprek voor de sollicitatie bij Dataworkz. Dataworkz is een geweldige startup waar ik na mijn tijd bij Sogeti/Capgemini en nu nog naast mijn rol als Freelancer werkzaam ben. Tijdens ons gesprek met z'n drieën (Arno Broekhof, Joep Kokkeler en ik) ging het heel erg over de inhoud maar op een zeer gepassioneerde manier. Alsof we het over voetbal hadden. Al snel kwam ik erachter dat Arno wel heel veel weet.... Hij gaf mij bijvoorbeeld de oplossing voor een probleem die ik had met Elasticsearch waar ik al 2 maanden mee aan het stoeien was. Dat kan natuurlijk ook iets zeggen over mijn skills met Elasticsearch, maar de voor deze geef ik de credits aan Arno. Maar dat was nog niet alles, tijdens de afgelopen maanden dat ik hier werkzaam ben bij Dataworkz ben ik erachter gekomen dat Arno gewoon veel weet over ontzettend veel onderwerpen die ook maar een klein beetje met Data Engineering te maken hebben. Als het nu gaat over Deep Learning voor Image Recognition, Machine Learning automation of webapplicaties schrijven in Python Flask, van ieder onderwerp krijg je het idee dat het zijn stokpaardje is. Het was voor mij dan ook niet moeilijk om antwoord te geven op zijn voorstel om mijn boek te reviewen.

In dit boek zal je merken dat ik het type ontwikkelaar ben die zich vooral richt op het eerst werkend krijgen van code en het later strakker en efficiënter inrichten. Echter moet je in een boek over Python wel rekening houden dat je de lezers inspireert om code op de juiste manier te schrijven. Dat is waar de opmerkingen van de review van Arno mij vooral bij geholpen hebben. Bij deze

wil ik nogmaals Rosemarijn en Arno bedanken voor het *prettiger leesbaar maker* van mijn boek.

Contact

Verder wens ik je veel plezier en zie ik er naar uit om met mijn lezers in contact te komen. Ook werk ik op dit moment aan extra hulpmiddelen om de ervaring van dit boek nog meer te verbeteren zoals video's en bestanden met de voorbeeld code. Houdt daarom mijn LinkedIn posts in de gaten.

LinkedIn - voor algemene updates

<https://www.linkedin.com/in/arietwigt/>

Website – blog

<https://arietwigt.nl>

Instagram – voor wat minder formele updates

<https://www.instagram.com/arietwigt4/>

Github – code achter de voorbeelden in dit boek

https://github.com/ArieTwigt/boek_code_python

Inleiding

We beginnen met het hoofdstuk waarin je de minste praktijk zal tegenkomen, de inleiding. Ondanks het feit dat dit een praktisch boek is, leek het mij niet verkeerd om eerst uit te leggen wat je van dit boek kan verwachten. Ik leg kort uit waarom dit misschien wel een van de beste keuzes is die je de afgelopen tijd hebt gemaakt. Programmeren zorgt ervoor dat er een wereld voor je opengaat, helemaal met de toenemende digitalisering. Ook zorgt het ervoor dat je voldoening krijgt door dingen te creëren. Het mooie aan de digitale wereld is dat de bouwstenen hiervoor eindeloos zijn. Nu klinkt het overdreven om te vertellen dat er een hele nieuwe wereld op je staat te wachten, maar het komt aardig in de buurt.

Voor wie dit boek is bedoeld

Dit boek is bedoeld voor iedereen die geïnteresseerd is in programmeren en hier een keer echt mee wil beginnen. Ook voor ontwikkelaars die al ervaring hebben in andere talen en kennis willen maken met Python, is dit een uitstekend boek.

Iedereen kan leren programmeren, maar dat wil niet zeggen dat het makkelijk is. Je kan het zo moeilijk maken als je zelf maar wil. Het ligt eraan wat je wilt bouwen of zelfs waar je dromen liggen, alles is mogelijk in de digitale wereld.

Voorwaarden om te beginnen

Wat zijn de voorwaarden om te beginnen? Simpel, een computer of laptop en een internetverbinding. Voor zowel de laptop als de internetverbinding geldt dat het voldoende is als deze op een basaal niveau zijn. Je hoeft dus geen extreem snelle internetverbinding te hebben of een laptop met zware hardware. De nodige software die geïnstalleerd moet worden, wordt later uitgelegd in dit boek. Alle software is trouwens gratis. Het eerste hoofdstuk gaat hier uitgebreid op in.

Ook maakt het niet uit op welk operating system je de voorbeelden uitwerkt. De voorbeelden in dit boek zijn gemaakt in Mac OSX, maar Python code werkt op ieder operating system hetzelfde. In de voorbeelden leg ik ook zo goed mogelijk voor Mac OSX **en** Windows te instructies uit.

De installatie is belangrijk omdat je er later ook aan moet geloven dat je in een nieuwe omgeving, bij een nieuwe baan of gewoon een nieuwe laptop, er ook

aan moet geloven dat je alles opnieuw moet installeren. Helaas bestaat er geen kant en klare versie die met 1 druk op de knop geheel volgens jouw wensen geconfigureerd wordt. Gelukkig is het installeren van software niet moeilijk.

Geduld

Al is dit boek geschreven om te werken in de vorm die lijkt op een bootcamp, geduld blijft een van de belangrijkste eigenschappen. Het bekende gezegde *Rome is niet in één dag gebouwd* is ook bij programmeren van toepassing. In je hoofd zie je misschien al helemaal voor je wat je gebouwd wilt hebben. Toch begint dit met kleine stapjes en misschien wel suffe onderdelen, maar die wel belangrijk zijn. Dat was nog een positieve situatie waar geduld bij nodig is. Een voorbeeld van een meer negatieve situatie is als je bepaalde zaken niet voor elkaar krijgt en dat het, tot jouw frustratie, veel te lang duurt voordat je tot een oplossing komt.

Neem van mij aan dat je hiermee niet de enige zal zijn en bedenk je dat exact op datzelfde moment wereldwijd duizenden andere ontwikkelaars die frustratie met jou delen. Neem ook van mij aan dat je er altijd wel uit komt, misschien niet volgens de benadering die je zelf voor ogen had, maar een manier die door een andere ontwikkelaar wordt voorgesteld. Er zijn tegenwoordig genoeg online platforms om met andere ontwikkelaars in contact te komen zoals **StackOverflow** of **Github**. Beide platforms worden in dit boek uitgebreid behandeld.

Creativiteit

Besef je dat je met programmeren iets creëert uit niets meer dan regels code die jij intypt. De mogelijkheden zijn eindeloos. Vooral in de laatste decennia zijn er

enorm veel producten en services gecreëerd die alleen uit software bestaan. Platforms zoals Facebook, Google, Dropbox en Uber zijn allemaal begonnen met een regel code en zijn door goed teamwork en doorzettingsvermogen uitgegroeid tot de misschien wel belangrijkste services ter wereld.

Het is daarom belangrijk dat als je zelf je eigen idee wilt uitwerken, je het eindresultaat al voor je ziet. Teken het uit en verdeel het in kleine hapbare stukjes die je kan uitwerken. Zelf teken ik heel veel situaties die ik wil aanpakken uit. Binnen een paar jaar heb ik al een flinke stapel aan notitieblokken vol.



Afbeelding 1. Uittekenen en verdelen in kleine stukken

Accepteer dat het een tijdje gaat duren voordat het af is en houd er rekening mee dat je onderdelen tegenkomt waar je vast zal komen te zitten. Als dit het geval is, neem dan even afstand door even te gaan wandelen, sporten, douchen of zelfs even op de Playstation te gaan. Deze laatste optie heeft mij altijd goed

geholpen. Een Gran Turismo race op de Playstation zorgde er bijvoorbeeld voor dat ik mijn hoofd even leeg kon maken en de situatie van een afstand, helikopter view, kon bekijken.

Wat leer je in dit boek

We zullen iedere term uitgebreid uitleggen, maar tegelijkertijd wel het tempo er voldoende in houden. Ook gaan we niet extreem veel tijd besteden aan uitgebreide theorieën of verschillende meningen die ontwikkelaars hebben om bepaalde onderdelen te bouwen. Het is een kennismaking met programmeren en je zult merken dat, naarmate je meer ontwikkeld bent in het programmeren, je code op efficiëntere wijze gaat schrijven. Dit wordt ook wel *refactoring* genoemd.

Het laatste wat ik in die boek wil doen, is jou overspoelen met informatie. Het is zonde als je hierdoor zal afdwalen voordat het echte programmeerwerk begint. Echter adviseer ik je, naar mate je een meer ervaren ontwikkelaar wil worden, boeken of artikelen lezen die jou leren om op de beste manier code te schrijven. Mijn mening over het leren van een nieuwe programmeertaal is in ieder geval om het echte werk zo snel mogelijk “aan te raken” en met kleine beetjes aan de slag te gaan.

Voor dit boek heb ik onderwerpen geselecteerd waarvan ik denk dat je deze het eerste zal tegenkomen als je Python in de praktijk gaat gebruiken. Hierbij heb ik gekeken naar mijn eigen ervaring uit de praktijk, bijvoorbeeld door bij een nieuwe klant te kijken naar welke onderwerpen er geconfigureerd moeten worden voordat we aan de slag kunnen gaan met Python. Met deze mix van onderwerpen zal je aan het einde van dit boek vol zelfvertrouwen kunnen

zeggen dat je kan programmeren (*developen*) in Python en de basics om te beginnen onder de knie hebt.

Eraan werken om programmeren in Python onder de knie te krijgen, is één van de betere beslissingen die je de afgelopen tijd hebt genomen. Ongetwijfeld krijg je later te maken met onderdelen die niet werken door fouten. Sterker nog, er zouden moment kunnen komen dat je denkt dat programmeren niets voor jou is en dat je er beter mee kunt stoppen... Luister naar dat stemmetje en antwoord vriendelijk dat je gewoon doorgaat. Op dat moment weet je dat je jouw grenzen aan het verleggen bent en dat je iets nieuws leert. Zelfs de meest doorgewinterde programmeurs hebben weleens mindere momenten of een dag dat ze niet zover zijn als ze zouden willen. Dit hoort er allemaal bij en is normaal.

In de meeste gevallen is een fout, of foutje, die over het hoofd wordt gezien de oorzaak van lang vastzitten. In andere gevallen is de manier waarop je iets wil aanpakken misschien niet juiste en moet je voor een andere methode kiezen. Bezie dat als een leermoment en een teken van jouw flexibiliteit om het roer om te gooien. Probeer in ieder geval altijd het eindresultaat voor ogen te zien. Ik heb zelf ook meerdere keren dit soort momenten gehad en ben er altijd doorheen gekomen waardoor ik terug kan kijken op een mooi leermoment. Denk eraan dat als ze mensen naar de maan kunnen sturen, jij ook wel dat stukje software waar je tegen aanloopt kan schrijven.

Het echte programmeren leer je door dingen te doen die je nog niet kan. Je zal daarom heel veel Google gebruiken, net zoals de meeste ontwikkelaars. Het zal je verbazen als je ziet hoeveel ontwikkelaars op Google of StackOverflow bijna standaard op hun tweede scherm open hebben staan. Denk dus niet dat je een amateur bent als je veel Google gebruikt, je bent dan juist op weg om een echte

ontwikkelaar te worden. Bij sommige onderdelen geef ik aan dat je het kunt opzoeken in Google. Dit is niet uit luiheid, maar om je te leren dat je via Google het snelst bij de meest recente en relevante informatie kan komen waar je op dat moment naar op zoek bent. Je zal er uiteraard een betere ontwikkelaar door worden.

Hoe dit boek werkt

Zoals aangegeven, richt dit boek zich vooral op het verkrijgen van praktijkervaring. Daarom bevat dit boek relatief veel stukken met voorbeeld code. Maak je geen zorgen, al het nodige wordt uitgelegd.

Ik raad je aan een paar uren in je agenda te blokken als je aan de slag gaat met dit boek. De drempel kan soms hoog zijn om alles op te zetten om er klaar voor te zijn om te programmeren, al stelt dit niet veel voor. Eenmaal aan de slag, ben je niet meer te stoppen en komt het goed uit dat je hier een paar uur voor hebt gereserveerd. Pak bijvoorbeeld een (lang) weekend. Hoe kan je deze nou beter besteden dan werken aan waardevolle skills waar je nog jaren plezier van gaat hebben?

Python ten opzichte van andere programmeertalen

Python is een van de meest gebruikte programmeertalen over verschillende vakgebieden heen. Deze verschillende vakgebieden zijn bijvoorbeeld *Web Development*, *Data Science*, *Big Data*, *Marketing*, *Web Scraping*, *Artificial*

Intelligence en *Testing*. Je kan Python uitstekend gebruiken om producten te bouwen zoals een online Social Media platform of bijvoorbeeld een chatbot.

Voor dit boek hadden we voor een andere programmeertaal kunnen kiezen, maar vanwege de flexibiliteit, efficiëntie en mogelijkheden kiezen we nu voor Python. Met efficiëntie bedoel ik dat Python vrij recht toe recht aan is voor wat je wil bouwen. De wat “zwaardere” programmeertalen zoals *Java* en *C#* (.NET), de zogenaamde *gecompileerde* programmeertalen, vereisen meer code om iets voor elkaar te krijgen. Je krijgt hier over het algemeen meer performance voor terug omdat de *compiler* van deze programmeertalen de code makkelijker kan interpreteren. Python is een zogenaamde *interpreted programming language*. Het hoeft niet gecompileerd te worden en gaat direct in de Python interpreter. Er zijn bergen artikelen geschreven wat hier de voor- en nadelen van zijn. Het is daarbij ook niet zo dat Python **altijd** “langzamer” is dan compiled programmeertalen.

Echter bestaan er wel *pyc*-bestanden. Dit zijn gecompileerde Python scripts en verhogen de performance. Deze *pyc*-bestanden kom je vooral tegen tijdens het bouwen van webapplicaties, bijvoorbeeld in het Flask-framework. Deze files worden automatisch gegenereerd en hoef je niets voor te doen. Webapplicaties bouwen met het Flask framework wordt ook in dit boek behandeld.

Zoals aangegeven kan Python voor veel verschillende onderwerpen worden gebruikt. Daarbij heeft Python een zeer grote community die zeer behulpzaam is. Een grote community heeft als voordeel dat je vragen snel beantwoord worden op platforms zoals **StackOverflow.com** of nog makkelijker zelfs, dat iemand anders jouw vraag al heeft gesteld en deze al beantwoord is. Je kunt dan bijna letterlijk de code in het voorgestelde antwoord kopiëren en plakken.

Een ander voordeel van een grote community is dat er veel extensies, *modules*, zijn gebouwd voor Python. Hiermee heb je een indrukwekkend arsenaal aan functies die je kan gebruiken om te bouwen wat jij voor ogen hebt. Je kunt er bijna vanuit gaan dat voor hetgeen dat jij wilt programmeren er een Python module is geschreven die functies bevat die jij hier goed bij kan gebruiken. Het onderwerp *modules* gaan we uitgebreid behandelen in dit boek en gaan we ook goed gebruiken.

Een korte tour door het boek

De bedoeling van dit boek is om de programmeertaal Python snel in de vingers te krijgen. Daarom zal dit boek vooral bestaan uit voorbeelden en minder uit theorie. Natuurlijk kun je dit boek ook lezen zonder dat je de voorbeelden uitwerkt. Daarnaast kan je dit boek ook gebruiken als naslagwerk.

Aan het einde van een hoofdstuk vind je een aantal oefeningen. Hiermee kan je jezelf toetsen of je de onderwerpen uit het hoofdstuk nog goed in de vingers hebt. De oefeningen zijn gerangschikt van eenvoudig naar uitgebreid.

Die rangschikking is ook toegepast op de hoofdstukindeling van dit boek. Het niveau gaat van de basics naar steeds meer uitgebreide onderwerpen. Je zal zien dat je in de uitgebreide onderwerpen veel handelingen uit de voorgaande hoofdstukken gaat gebruiken. Aan het einde van dit boek maak je bijvoorbeeld een eigen webapplicatie. Hierbij heb je bijna alle skills nodig die je in de eerdere hoofdstukken hebt geleerd.

Misschien wel de belangrijkste eigenschap van dit boek is dat de nadruk meer ligt op programmeren, code kloppen, dan de theorie doornemen. Daarom heb

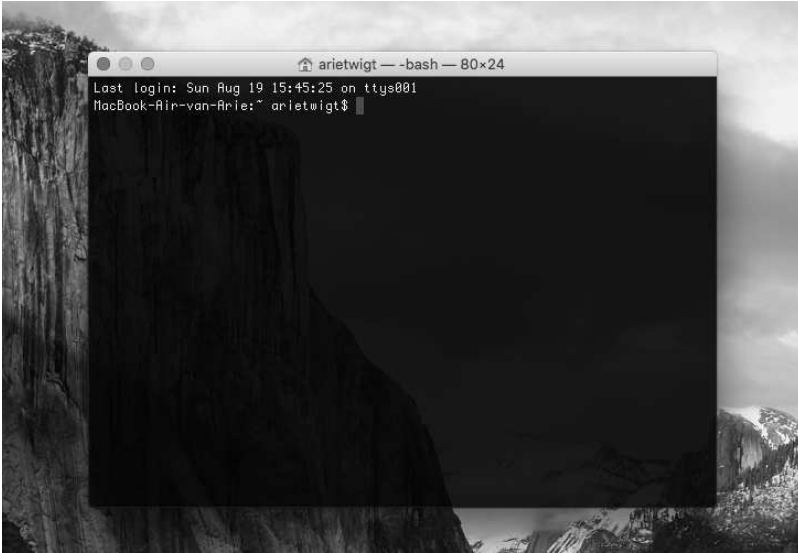
ik geprobeerd om de stukken tekst tussen de voorbeelden zo kort mogelijk te houden. Het is de bedoeling dat je lekker aan de slag gaat en liefde krijgt voor het vak. Verder hoop ik dat je een leuke en uitdagende ervaring zult hebben tijdens het lezen van dit boek en het uitwerken van de voorbeelden en oefeningen. Veel plezier en succes!

0 Command Line fundamentals

Als je hebt besloten om te beginnen met programmeren, is het belangrijk dat je kennis maakt met de *Command Line*. De Command Line heeft in verschillende operating systems een andere naam:

- Windows: **Command Prompt**
- Mac OSX en Linux: **Terminal**

Met de Command Line kan je bijvoorbeeld navigeren door je computer, bestanden of mappen aanmaken en scripts uitvoeren. In dit boek gebruiken we slechts de basics van Command Line.



Afbeelding 0.1 De Command Line in Mac OS

Op de verschillende operating systems (Windows, Linux, Mac) werkt de Command Line over het algemeen hetzelfde. Echter zijn sommige commands verschillend. Deze verschillen worden in dit boek duidelijk aangegeven zodra hier sprake van is. Voor nu is het belangrijk dat we even stilstaan bij de volgende commands. Je hoeft ze nog niet direct te demonstreren. Bij de instructies in dit boek wordt aangegeven wanneer het tijd is om deze commands te gebruiken.

- **ls** (Mac OSX/Linux) / **dir** (Windows): Met dit command kan je de bestanden in een map weergeven. Je kan het vergelijken met de *Verkenner* van Windows of de *Finder* op een Mac waarmee je kan zien welke bestanden en mappen zich in de huidige map bevinden;

- **cd**: Het cd command is een afkorting voor “Current Directory”. Hiermee selecteer je de map waar je naar wilt navigeren. Dit kan je vergelijken met het openen van een map in Verkenner of Finder.
- **cd ../**: Heeft dezelfde functie als het **cd** command, maar gaat een map terug. Dit is vergelijkbaar met de **vorige** (←) knop in Verkenner of Finder.

1. Installatie

Het begint allemaal met de installatie van de benodigde software. Dit is nodig om de voorbeelden in dit boek te volgen. Daarbij leer je ook hoe de software is geconfigureerd op jouw computer. Mocht je een keer op je werk of met je eigen project gebruik moeten maken van Python, is het handig dat je weet wat er bij komt kijken om de benodigde software te installeren. Geloof mij als ik zeg dat je genoeg momenten tegenkomt waarbij je Python (opnieuw) moet installeren, bijvoorbeeld als je een nieuwe computer of laptop hebt gekocht.

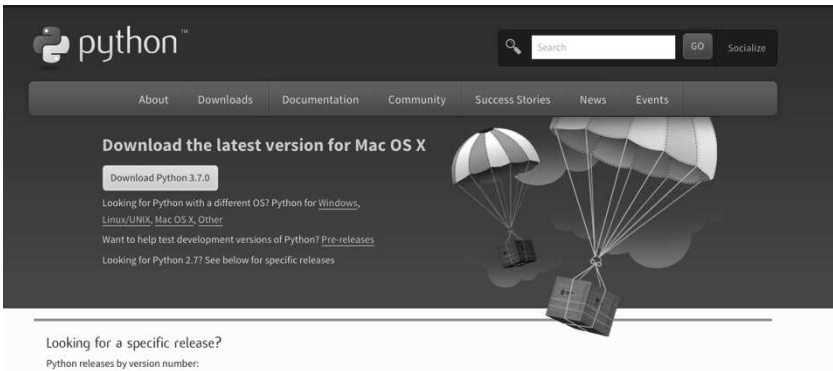
Naast Python, installeren we software die we kunnen gebruiken om Python scripts te bouwen en demonstreren. Ook installeren we handige hulpmiddelen zoals software voor versie-controle. De software die in dit hoofdstuk behandeld wordt is voor een groot deel een persoonlijke keuze. Uit de praktijk ben ik deze collectie aan software om Python scripts te creëren vaak tegengekomen en omdat beginnende ontwikkelaars deze manier als prettig ervaren.

Python installeren

Het installatiebestand

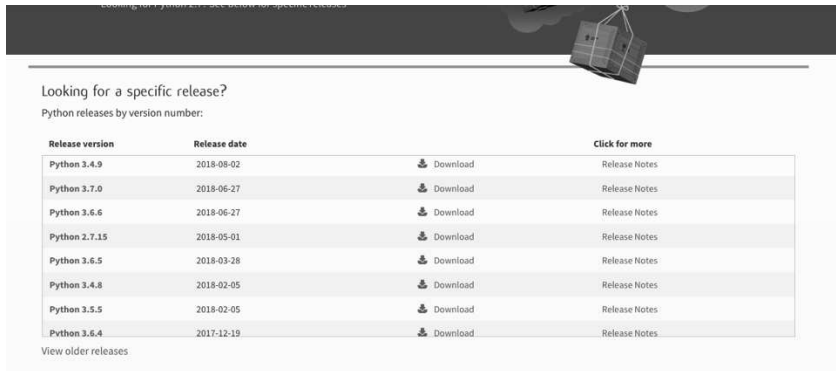
Laten we eerst Python installeren. Hiervoor gaan we naar de URL: <https://www.python.org/downloads/> . Op deze pagina kan je de twee verschillende Python versies downloaden: Python 2.7 en Python 3.7. Stoor je voor nu even niet aan het feit dat er twee Python versies zijn... Voor deze cursus gebruiken wij Python 3. Hiermee bedoelen we alle Python versies die met **3** beginnen zoals Python 3.5, Python 3.6 en Python 3.7. De voorbeelden in dit boek zijn over al deze verschillende versies probleemloos te gebruiken.

Klik op de grote gele knop **Download Python 3.7** om het installatiebestand te downloaden.



Afbeelding 1.1 Python downloaden

Wil je er echt zeker van zijn dat je dezelfde Python versie wil gebruiken als de voorbeelden in dit boek? Download dan alsnog Python 3.6 door te zoeken onder het kopje *Looking for an older release?* In dat kadertje zie je Python versies als Python 3.6.5 of Python 3.6.6. Met deze versies bedoelen wij gewoon **Python 3.6**.



Afbeelding 1.2 Een andere Python versie downloaden

- Het downloaden van het installatiebestand duurt ongeveer een minuut.
- Als het bestand is gedownload, open het installatiebestand.
- Let op! Bij de vraag, of checkbox, "*add Python to Pythonpath*", kies je voor **Yes**. Dit is belangrijk om Python op een makkelijke manier te openen. Kies vervolgens zo veel mogelijk voor de standaard instellingen die worden voorgesteld.

Testen of Python werkt

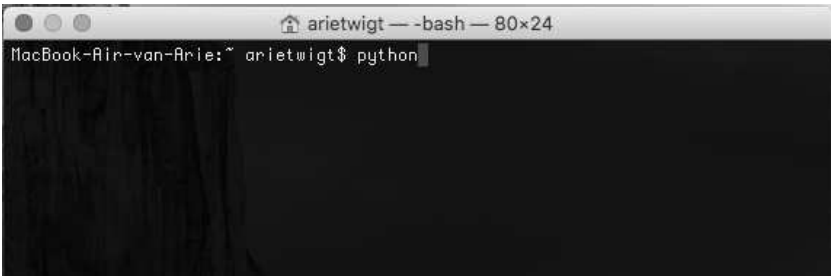
We gaan nu testen of Python werkt op onze computer. Dit kunnen we zien door te checken of de *Python Shell* wordt geopend als we het woord **python** invoeren in Command Line:

1. Open de Command Line, dus de **Terminal** op een Mac of het **Command Prompt** in Windows



2. Typ **python** in:

3.



Afbeelding 1.3 python openen vanaf the command line

3. Als het *Python* programma verschijnt, zoals te zien is in **Afbeelding 1.4**, is Python succesvol geïnstalleerd


```
Python 3.6.1 (v3.6.1:69c0db5050, Mar 21 2017, 01:21:04)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Afbeelding 1.4 Python gebruiken in de Command Line

Omgevingsvariabelen

Om programma's te kunnen openen vanuit de Command Line, moeten deze zijn toegevoegd aan de omgevingsvariabelen. Met de optie *Add Python to Pythonpath* bij de installatie van Python, geven we aan dat we Python willen toevoegen aan de omgevingsvariabelen. Als we niet voor deze optie kiezen, moet Python handmatig worden toegevoegd aan de omgevingsvariabelen. Dit komt in sommige gevallen weleens voor bij Windows. In dat geval kan je de instructies op deze pagina volgen:

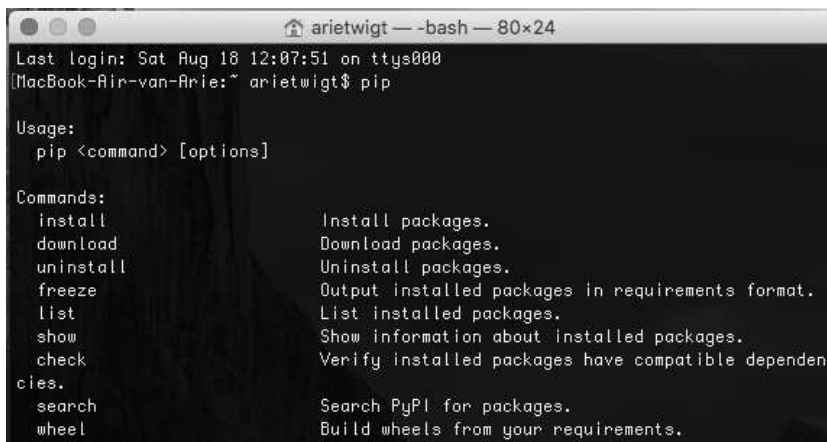
<https://stackoverflow.com/questions/3701646/how-to-add-to-the-pythonpath-in-windows>

Zoals in het begin is besproken, is Google de beste vriend van een ontwikkelaar. Later in dit boek wordt de website StackOverflow uitgebreid besproken.

Pip Python package manager

Pip is de package manager van Python waarmee je extra modules, dus extra functies, van Python kan installeren. Omdat we in dit boek extra modules gaan installeren, is het belangrijk dat we checken of **pip** al is geïnstalleerd.

1. Open de Command Line en typ het woord **pip** in:



```
arietwigt --bash-- 80x24
Last login: Sat Aug 18 12:07:51 on ttys000
[MacBook-Air-van-Arie:~ arietwigt$ pip

Usage:
  pip <command> [options]

Commands:
  install           Install packages.
  download          Download packages.
  uninstall         Uninstall packages.
  freeze           Output installed packages in requirements format.
  list             List installed packages.
  show            Show information about installed packages.
  check           Verify installed packages have compatible dependen
cies.
  search          Search PyPI for packages.
  wheel          Build wheels from your requirements.
```

Afbeelding 1.5 Controleren of 'pip' is geïnstalleerd

2. Als er opties worden weergegeven, zoals in **Afbeelding 1.5**, betekent het dat **pip** succesvol is geïnstalleerd.

Het komt bijna niet voor dat Python wordt geïnstalleerd zonder dat **pip** ook tegelijkertijd wordt geïnstalleerd. Als dit bij jou niet het geval is, heb je de optie om **pip** handmatig te installeren. Schrik niet, het klinkt spannender dan dat het daadwerkelijk is. Ga naar de URL:

<https://pip.pypa.io/en/stable/installing/#do-i-need-to-install-pip> en volg de eenvoudige instructies:

1. Download het **get-pip.py** bestand;
2. Navigeer in de Command Line naar de map waar het **get-pip.py** bestand zich bevindt;

3. Voer het **get-pip.py** bestand uit met het command: **python get-pip.py**.
4. Pip wordt vervolgens geïnstalleerd. Doe na de installatie voor de zekerheid alsnog de controle of **pip** is geïnstalleerd.

Virtualenv installeren

Een *Virtual Environment* zorgt ervoor dat ieder project op jouw computer zijn eigen Python installatie heeft. Op deze manier kan je bijvoorbeeld verschillende Python versies gebruiken per project. Belangrijker nog, is dat je de Python modules die je gebruikt over de verschillende projecten heen netjes worden gescheiden. In het verdere verloop van dit boek zullen we de voordelen van het werken met een **Virtual Environment** verder uitleggen. Het hoort bij de “best practices”, beste manier, van het programmeren met Python. We richten ons nu op de installatie.

Virtualenv wordt de eerste Python module die we installeren. Open de Command Line en voer het volgende command in:

- **pip install virtualenv**

Bij het installeren van een Python module, krijg je o.a. te zien dat de bestanden worden gedownload (**Downloading...**) en bezig is de module te installeren (**Installing...**). Uiteindelijk krijg je het bericht dat de module succesvol is geïnstalleerd (**Successfully installed ...**).

Het project opzetten

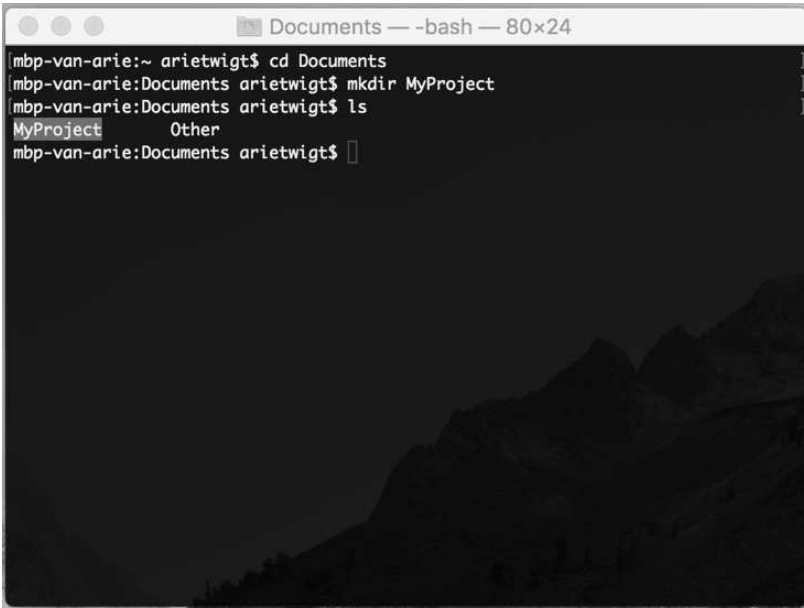
De projectmap aanmaken

Met de term **Project** bedoelen we een map waar de scripts worden opgeslagen. Dit is zeer gebruikelijk voor ontwikkelaars. Een website is bijvoorbeeld een map met scripts, afbeeldingen en andere type bestanden om de website te laten draaien. Dit is voor andere type softwareprojecten dan websites niet anders.

Kies voor jezelf een locatie die je handig vindt om een project in op te slaan. Een voorbeeld is de **Documents** (Mac OSX) of **Mijn Documenten** (Windows) map. Het is af te raden om het project op het bureaublad de plaatsen of een nog vreemdere plek zoals de **Program Files** map.

Laten we vanaf de Command Line een map aanmaken die gaat dienen als onze projectmap. Navigeer naar de locatie/map waar je jouw project wil opzetten. Bijvoorbeeld:

- **cd Documents**
- **mkdir MyProject**

A terminal window titled "Documents — -bash — 80x24" with a dark background and light text. The window shows a sequence of commands and their outputs: the user changes to the "Documents" directory, creates a directory named "MyProject", and lists the contents of the directory, which shows "MyProject" and "Other".

```
mbp-van-arie:~ arietwigt$ cd Documents ]
mbp-van-arie:Documents arietwigt$ mkdir MyProject ]
mbp-van-arie:Documents arietwigt$ ls ]
MyProject      Other
mbp-van-arie:Documents arietwigt$
```

Afbeelding 1.6 Een projectmap aanmaken in de Command Line

Klaar! Je hebt een map aangemaakt met de naam van jouw project. In dit boek gaan we dit project vullen met onze scripts.

Mocht je er om een bepaalde reden niet uit komen met de bovenstaande commands in de Command Line, dan kun je de map ook handmatig aanmaken met Finder of Verkenner. De map aanmaken via de Command Line, zoals aangegeven in het bovenstaande voorbeeld, is bedoeld om alvast warm te draaien met de Command Line commands.