

INLEIDING



Welkom bij *Programmeren met Minecraft*! In dit boek leer je programma's schrijven in de programmeertaal Python. Met deze programma's kun je allerlei dingen regelen die in jouw Minecraft wereld gebeuren. Je leert

wat het betekent om te programmeren en daarna gebruik je wat je geleerd hebt om met programmeercode gebouwen te maken, mini-spellen te schrijven en saaie Minecraft items om te toveren tot leuk nieuw speelgoed. Aan het eind van het boek heb je zoveel vaardigheden opgedaan dat je de wildste ideeën kunt uitvoeren.

Programmeren is creatief en fantasievol, net als Minecraft. Je leert in dit boek programmeertechnieken waarmee je allerlei dingen kunt bouwen, zoals spellen, apps en nuttige hulpmiddelen, maar je leert ook programma's maken die niets met Minecraft te maken hebben. Dit is de eerste stap op weg naar een toekomst als geweldige programmeur en Minecraft master!

WAAROM LEER JE PROGRAMMEREN?

Als je leert programmeren, leer je tegelijkertijd ook problemen oplossen. Dat is een belangrijke reden om hiermee aan de slag te gaan. Je leert ook grote problemen opdelen in kleinere stukjes die je gemakkelijker oplost.

Daarnaast kom je veel problemen tegen waarbij je op een creatieve manier moet denken en verschillende ideeën moet uitproberen.

Een ander voordeel is dat je logisch leert denken. Je begrijpt daardoor beter hoe jouw programma in elkaar zit en je kunt de opbouw en ‘flow’ van je programma’s beter snappen. De vaardigheid om problemen op te lossen, creativiteit en logisch denken zijn allemaal waardevolle eigenschappen die je ook op andere terreinen kunt gebruiken.

Bovendien is een loopbaan als programmeur ook nog eens heel leuk. Je mag elke dag creatieve oplossingen bedenken voor verschillende problemen. Als je geen programmeur wilt worden, is het toch een leuke en spannende hobby. Ik ben zelf met programmeren begonnen als hobby en later heb ik er mijn werk van gemaakt.

Het allerbelangrijkste is dat je ongelooflijk veel plezier kunt hebben bij het programmeren! Er is niks fijner dan een programma coole dingen zien doen die jij zelf hebt geprogrammeerd!

WAAROM PYTHON?

Waarom zou je eigenlijk leren programmeren met Python? Python is een geweldige taal om als beginnend programmeur mee te starten. Het is een gemakkelijke taal om te leren lezen en schrijven en krachtig genoeg om echte computerprogramma’s mee te bouwen. Python is één van de populairste programmeertalen ter wereld!

WAAROM MINECRAFT?

Minecraft is erg populair omdat het leuk en creatief is. In jouw Minecraft wereld ben je helemaal vrij om te maken wat je maar wilt. Je kunt je fantasie de vrije loop laten. Als je Minecraft integreert of combineert met je eigen Python programma’s heb je nog meer controle over Minecraft en kun je nóg creatiever bezig zijn. Je kunt dingen doen die niet mogelijk zijn als je alleen maar Minecraft gebruikt, bijvoorbeeld binnen een paar seconden een gigantisch gebouw maken.

Soms kan het lastig zijn om met programmeren te beginnen, omdat je heel veel code moet leren waarmee je niet echt spannende dingen doet, maar door het combineren van Python en Minecraft verschijnen de resultaten van jouw indrukwekkende programma’s direct in jouw Minecraft wereld.

WAT STAAT ER IN DIT BOEK?

Elk hoofdstuk gaat over één enkel begrip in Python. Naarmate je verder in het boek komt, wordt jouw kennis van Python groter. In elk hoofdstuk wordt uitgelegd hoe Python werkt, zie je voorbeelden van Python in actie en vind je Minecraft missies. In deze missies schrijf je programma’s die samenwerken met Minecraft. Ik geef je wat basiscode om mee te starten en dan moet jij zelf de ontbrekende delen aanvullen en de programma’s afmaken. Hierdoor ontwikkel je probleemoplossende vaardigheden die van groot belang zijn voor iedere programmeur.

Hieronder zie je wat je in elk hoofdstuk ontdekt.

- **Hoofdstuk 1: Zet alles klaar voor je avontuur** helpt je met het klaarzetten van Python en Minecraft. Daarna begin je met programmeren!
- **Hoofdstuk 2: Teleporter met variabelen** leert je om je speler direct te teleporteren door variabelen te manipuleren. Je leert wat variabelen zijn en hoe zij gegevens kunnen onthouden in een programma. Jouw bijzondere teleporteervaardigheden gebruik je zelfs om een magische rondleiding te maken door jouw eigen wereld.
- **Hoofdstuk 3: Snel bouwen en ver reizen door te rekenen** laat zien hoe je rekenen gebruikt om jezelf super machten toe te kennen en je creaties super snel te bouwen. Wil je binnen één seconde een Minecraft huis bouwen? Rekenkundige operatoren helpen je hierbij. Wil je super hoog de lucht in springen? Hierbij krijg je ook hulp van rekenkundige operatoren!
- In **Hoofdstuk 4: Chatten met strings** leer je alles over strings en bouw je hiermee een interactieve chat. In een programmeertaal wordt met een *string* eigenlijk gewoon tekst bedoeld. Je leert Minecraft Python programma's schrijven die boodschappen overbrengen aan jou en jouw spelers.
- **Hoofdstuk 5: Gebruik het Booleaanse data type om true en false te onderscheiden** leert je Booleaanse operatoren en logica gebruiken om je programma's vragen te laten beantwoorden. Dat betekent dat je programma's kunt bouwen die jou kunnen vertellen of iets waar (true) of onwaar (false) is. Jouw Minecraft Python programma's kunnen allerlei soorten vragen beantwoorden: Ben ik onder water? Zit ik in een boom? Ben ik vlakbij mijn huis?
- **Hoofdstuk 6: Mini-spellen maken met if statements** neemt je mee naar een volgend niveau van Booleaanse logica. Door if statements te gebruiken leer je programma's bouwen die beslissingen nemen op grond van de gegevens die zij krijgen. Heb je ooit bedacht dat het leuk zou zijn om een geheime doorgang te openen in Minecraft door een bepaald blok op een bepaalde plek neer te zetten? Nou, dit kun je dus doen door if statements te gebruiken!
- **Hoofdstuk 7: Dance party's en bloemenparades met while lussen** laat je een heel slimme manier zien om jouw programma's dingen te laten herhalen met behulp van lussen. Hiermee voer je code automatisch uit en er kunnen fantastische dingen gebeuren. Stel je eens voor dat een speler gevolgd wordt door een spoor van bloemen of bouw een betoverde dansvloer die verschillende kleuren aanneemt. Dit zijn de programma's die ik het liefst aan anderen laat zien om een beetje op te scheppen.
- In **Hoofdstuk 8: Super krachten met functies** leer je in één klap complete bossen en steden bouwen met behulp van functies. Je leert ook hoe je jouw leven als programmeur gemakkelijker maakt door delen van je programma's opnieuw te gebruiken.
- In **Hoofdstuk 9: Dingen raken met lijsten en woordenboeken** leer je mini-spellen maken met behulp van lijsten. Lijsten zijn hele krachtige

programmaonderdelen, omdat je belangrijke informatie kunt opslaan op één plek. Je gebruikt lijsten om jouw programma alle blokken te laten onthouden die je met je zwaard hebt geraakt. Met een paar extra regels code tover je alles om tot een mini-spel. Vrij cool, zou ik zeggen!

- **Hoofdstuk 10: Minecraft magie met for lussen** laat zien hoe je dingen bouwt met for lussen, bijvoorbeeld piramides. Je kunt for lussen zelfs gebruiken om pixelvormen te tekenen of jouw Minecraft gebouwen te dupliceren. Bouw een prachtig beeld en dupliceer dit, dan heb je een heel leger van beelden!
- In **Hoofdstuk 11: Gebouwen opslaan en laden met bestanden en modules** gebruik je jouw programma's om bestanden te maken en bewaren waarin je de objecten opslaat die je hebt gebouwd. Vervolgens laad je die in verschillende spelwerelden in. Eigenlijk zet je op deze manier je gebouwen om in bestanden en deze kun je dan overal naar toe verplaatsen. Wil je dat geweldige landhuis bewaren dat je laatst hebt gebouwd? Geen punt! Door bestanden te gebruiken sla je het huis op en laad je het weer op elke plek die je wilt.
- **Hoofdstuk 12: Objectgeoriënteerd programmeren geeft je klasse** laat je kennismaken met een paar onderwerpen voor gevorderden: klassen, objecten en overerving. Tegen de tijd dat je dit hoofdstuk hebt doorgevoerd ben je een ware Python master. In de missies begin je met de code waarmee je een gebouw bouwt. Daarna gebruik je klassen, objecten en overerving om op een simpele manier duplicaten en variaties te bouwen. Met maar een paar extra regels code kun je bijvoorbeeld hele dorpen en hotels bouwen.
- Het stukje **Blok ID Cheat Sheet** is een handig naslagwerk om Minecraft blok ID's op te zoeken die je in jouw programma's kunt gebruiken.

ONLINE BRONNEN

Alle code en bronnen die in dit boek worden gebruikt vind je op de website bij dit boek www.visualsteps.nl/minecraft. Als je vast komt te zitten en de oplossing wilt bekijken, download je de code voor de Minecraft missies. Deze code kun je ook aanpassen om daarmee je eigen fantastische programma's te bouwen! Hier kun je ook de setup bestanden downloaden. In Hoofdstuk 1 begeleid ik je bij het klaarzetten van alle bestanden.

GA NU MAAR OP AVONTUUR

Ik hoop dat je net zoveel zin hebt als ik om eindelijk aan de slag te gaan. Ik vond het ontzettend leuk om dit boek te schrijven en alle Minecraft missies te maken die je gaan helpen bij het leren programmeren. Laten we maar snel beginnen!

2

TELEPORTEREN MET VARIABELEN



Ben je zover dat je de kracht van Python kunt gebruiken om jouw Minecraft wereld te beheersen? In dit hoofdstuk krijg je een korte rondleiding langs de basisbegrippen van Python. Daarna test je je nieuwe vaardigheden door je eigen rondreis te maken waarin je door jouw Minecraft wereld teleporteert!

De begrippen in dit hoofdstuk horen niet alleen maar bij Minecraft Python. Je kunt ze dus ook in andere Python programma's gebruiken die je gaat maken.

WAT IS EEN PROGRAMMA?

Een *programma* is een set instructies die ervoor zorgt dat jouw computer een bepaalde taak of bepaalde taken uitvoert. Bijvoorbeeld een stopwatch app op een mobiele telefoon. Het stopwatch programma heeft instructies voor wat er moet gebeuren als jij op start en stop drukt. Het programma heeft ook instructies om de tijd op het beeldscherm te tonen terwijl de tijd loopt. Er is iemand geweest, een jongen of een meisje, die de stopwatch heeft geprogrammeerd om zo te werken.

Elke dag worden over de hele wereld miljoenen programma's gebruikt. De berichtenapp op een mobieltje is een programma, verkeerslichten worden door programma's bestuurd en zelfs computerspellen zoals Minecraft zijn programma's.

In dit boek leer je de basisbeginselen van het programmeren en je leert ook hoe je programma's schrijft die jouw ideeën tot leven brengen in Minecraft.

GEGEVENS OPSLAAN MET VARIABLEN

We beginnen met te leren hoe je gegevens (ook wel data genoemd) opslaat in variabelen. Met *variabelen* kun je gegevens opslaan om die later in het programma weer te gebruiken. Met *data* wordt alle informatie bedoeld die je zou willen bewaren, bijvoorbeeld getallen, namen, allerlei tekst, lijsten met items, enzovoorts. Hier is bijvoorbeeld een variabele pikhouwelen die de waarde bewaart van het getal 12:

```
>>> pikhouwelen = 12
```

Variabelen kunnen getallen, woorden en zelfs hele zinnen bewaren, zoals “Maak dat je wegkomt, engerd!” Je kunt variabelen ook veranderen waardoor je een paar hele mooie dingen kunt doen in Minecraft. Je gebruikt straks al variabelen om te profiteren van de kracht van het teleporteren!

Om in Python een variabele te maken heb je een variabele naam nodig, een is-gelijk teken (=) en een waarde. Stel dat je aan het begin staat van een groot avontuur; dan wil je wel genoeg eten meenemen. Je kunt dit voedsel in de vorm van een variabele opvoeren. In onderstaande Python shell is brood de naam van de variabele en is 145 de waarde:

```
>>> brood = 145
```

De naam van de variabele staat altijd links van het is-gelijk teken en de waarde die je wilt opslaan staat altijd rechts, zoals je in Figuur 2-1 ziet. Deze Python code regel *declareert* de variabele *brood* en *wijst* hieraan de waarde 145 toe.

Nadat je een variabele hebt gedeclareerd en hieraan een waarde hebt toegewezen, voer je de naam van de variabele in de Python shell in om te kijken wat deze variabele inhoudt:

```
>>> brood
145
```

brood = 145
[] []
variabele naam waarde

Figuur 2-1: De delen van een variabele declaratie. Je hebt wel erg veel honger als je 145 broden nodig hebt.

Je kunt bijna elke naam gebruiken voor een variabele. Maar het is het beste om een naam te gebruiken die beschrijft wat het doel is van de variabele. Zo begrijp je beter wat er in jouw programma gebeurt. Het is niet echt een regel, maar het is beter om de variabele naam te beginnen met een

kleine letter, geen hoofdletter. Deze stijl wordt door Python programmeurs gevolgd en het is een goede gewoonte om deze stijl aan te houden. Zo kunnen anderen gemakkelijk jouw code lezen als je die ooit wilt delen.

LET OP!

Hoewel de waarde van een variabele wordt opgeslagen, wordt die niet permanent bewaard. De waarde van een variabele wordt opgeslagen in het tijdelijke geheugen van de computer. Dat betekent dat de waarde van de variabele weg is zodra de computer wordt uitgeschakeld, of het programma stopt. Probeer maar eens om IDLE te sluiten en dan weer te openen. Wat gebeurt er als je de waarde van brood wilt zien?

DE STRUCTUUR VAN PROGRAMMEERTALEN

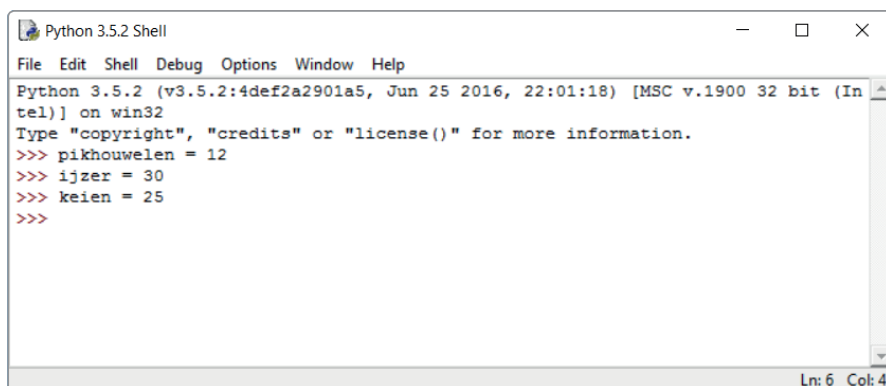
De set regels die beschrijft hoe de grammatica en interpunctie (spaties, punten en komma's) van een programmeertaal eruit moeten zien, heet de *syntax*. Dit lijkt op de grammatica en interpunctie in een gewone menselijke taal. Wanneer je de syntax van Python begrijpt, kun je betere programma's schrijven die een computer goed kan volgen. Maar als je de verkeerde syntax gebruikt, begrijpt de computer niet wat je hem wilt laten doen.

Stel je eens voor dat één enkele instructie in jouw code hetzelfde is als een zin. Een zin eindigt in het Nederlands altijd met een punt. In plaats van een punt, gebruikt Python een nieuwe regel om het einde van een instructie aan te geven. De volgende instructie begint dan op de nieuwe regel. Elke instructie die op een nieuwe regel staat wordt een *statement* genoemd.

Stel dat je wilt bijhouden hoeveel hakbijlen, blokken ijzererts en keien je hebt. In de Python shell schrijf je dit als volgt:

```
>>> pikhouwelen = 12
>>> ijzer = 30
>>> keien = 25
```

In Figuur 2-2 zie je hoe dit eruit ziet in de Python shell.



Figuur 2-2: Code invoeren in de Python shell

Je ziet dat elk statement op een aparte regel staat. Doordat er elke keer een nieuwe regel gemaakt is, begrijpt Python dat je drie verschillende items wilt bijhouden. Als je elk statement niet op een nieuwe regel zet, raakt Python in de war en krijg je een foutmelding die *syntax error* heet:

```
>>> pikhouwelen = 12 ijzer = 30 keien = 25
SyntaxError: invalid syntax
```

Een *syntax error* is de manier waarop Python je vertelt dat hij het niet begrijpt. Python kan deze instructies niet opvolgen, omdat hij niet weet waar het ene statement eindigt en het andere begint.

Als je een regel begint met een blanco spatie, weet Python ook niet meer wat hij moet doen:

```
>>>  ijzer = 30
SyntaxError: unexpected indent
```

Als je goed kijkt, zie je dat er spaties staan vóór de code, aan het begin van de regel. Als je zo'n *unexpected indent syntax error* (onverwachte inspringing) krijgt, zoals deze, weet je dat er spaties staan aan het begin van een code regel.

Python is erg kieskeurig als het gaat om de manier waarop je de code schrijft. Als je tijdens het uitwerken van de voorbeelden in dit boek een *syntax error* krijgt, moet je heel nauwkeurig alles nakijken. Meestal vind je wel ergens een foutje.

SYNTAXREGELS VOOR VARIABELEN

Om Python duidelijk te maken dat het om een variabele gaat, moet je je ook aan een paar *syntax* regels houden. Dan gaat het vooral om de naam van de variabele:

- Gebruik geen symbolen in je variabele namen, behalve underscores (`_`); anders krijg je een *syntax error*.
- Laat de naam van een variabele niet met een cijfer beginnen, zoals in `9brood`. Verderop in de naam mag je wel een cijfer gebruiken, bijvoorbeeld `brood9`.
- Je hoeft aan beide kanten van het is-gelijk teken (`=`) géén spaties te gebruiken. Jouw programma draait ook goed zonder die spaties. Maar ze maken het wel gemakkelijker om de code te lezen, dus gebruik ze toch maar wel.

Variabelen zijn erg nuttig. Straks leer je hoe je de waarde van variabelen verandert. Daarna ben je klaar voor het teleporteren van je speler!

DE WAARDEN VAN VARIABELEN VERANDEREN

Je kunt op elk moment de waarde van een variabele veranderen, op dezelfde manier als je een variabele declareert. Stel dat je vijf Minecraft

katten tegenkomt en je wilt deze waarde als een variabele opslaan. Eerst declareer je een variabele `katten` en wijs je de waarde 5 toe aan deze variabele. In een Python shell ziet dat er zo uit:

```
>>> katten = 5
>>> katten
5
```

Later kom je nog vijf andere katten tegen en besluit je dat je deze waarde wilt ophogen. Wat gebeurt er als je de waarde van `katten` verandert in 10?

```
>>> katten = 10
>>> katten
10
```

Als je Python nu vraagt naar de nieuwe waarde van `katten`, is dat geen 5 meer! Als je vanaf nu de variabele `katten` in een programma gebruikt, heeft die de nieuwe waarde 10.

In een variabele kun je verschillende soorten data opslaan. *Data types* vertellen aan de computer hoe ze met een bepaalde soort data moeten omgaan. Ik begin met het bespreken van één van de meest gebruikte data types. Dit zijn integers. Verderop in het hoofdstuk heb ik het ook over het floats data type.

INTEGERS

Integers zijn hele getallen die zowel positief als negatief kunnen zijn. Waarden als 10, 32, -6, 194689 en -5 zijn integers, maar 3.14 en 6.025 zijn dat niet.

Waarschijnlijk gebruik je elke dag integers zonder erbij na te denken, zelfs in Minecraft! Zo kun je bijvoorbeeld zomaar 12 koeien op een heuvel zien staan terwijl je op weg bent om 5 diamanten op te graven, met 2 verse appels in jouw inventory. Al deze getallen zijn integers.

Stel dat je in jouw Minecraft wereld vijf varkens hebt en je wilt een programma schrijven dat op de een of andere manier dit aantal varkens gebruikt. In Python zou je dan een integer variabele declareren om het aantal varkens weer te geven:

```
>>> varkens = 5
```

Je kunt ook negatieve waarden opslaan in een variabele. Als je bijvoorbeeld wilt zeggen dat de temperatuur negatief is, stel min vijf graden, dan declareer je de volgende variabele:

```
>>> temperatuur = -5
```

Om Python variabelen en integers samen met Minecraft te gebruiken, moet je de eerste missie uitvoeren.