PROGRAMMEREN VOOR KINDEREN

LEER STAP VOOR STAP PROGRAMMEREN
EN JE EIGEN COMPUTERGAMES MAKEN

LANNOO

DK Londen

Redactie Sam Priddy, Sam Atkinson, Lizzie Davey,Daniel Mills, Ben Morgan, Maud Whatley
Design Fiona Macdonald, Simon Murrell, Laura Brim, Sophia MTT
Hoofdredactie Paula Regan, Owen Peyton Jones
Productie Ben Marcus, Mary Slater
Uitgever Sarah Larter
Art director Phil Ormerod
Associate publishing director Liz Wheeler
Publishing director Jonathan Metcalf

DK India
Redactie Devika Dwarkadas, Suefa Lee, Neha Pande, Sanjay Chauhan,
Shreya Anand Virmani, Vanya Mittal
Design Sachin Gupta, Suhita Dharamjit, Harish Aggarwal
Hoofdredactie Rohan Sinha, Sudakshina Basu
Productie Balwant Singh

Oorspronkelijke titel *Computer Coding for Kids*
Oorspronkelijke uitgever Dorling Kindersley Limited, Londen
Copyright © 2014 Dorling Kindersley Limited, Londen
Copyright © Nederlandse vertaling Uitgeverij Lannoo, Tielt
Vertaling Ronald Schrijber, www.groenlingua.nl
Redactie Nederlandse editie Jaap Verschoor / Kantoor Verschoor Boekmakers, Heemstede
Zetwerk Peter Verwey Grafische Produkties, Heemstede

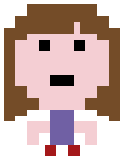
ISBN 978 94 014 1965 9
D/2014/45/359
NUR 211 / 241

www.lannoo.com

Gedrukt en gebonden in China bij South China Printing Company

Alle rechten voorbehouden. Niets uit deze uitgave mag worden verveelvoudigd,
opgeslagen in een geautomatiseerd gegevensbestand en/of openbaar gemaakt in enige vorm
of op enige wijze, hetzij elektronisch, mechanisch of op enige andere manier
zonder voorafgaande schriftelijke toestemming van de uitgever.

Registreer u op onze website en we sturen u regelmatig een nieuwsbrief
met informatie over nieuwe boeken en met interessante, exclusieve aanbiedingen.

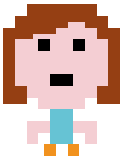


CAROL VORDERMAN is een van de populairste Britse tv-presentatrices en een bekend wiskundige. Ze heeft technische wetenschappen gestudeerd in Cambridge. Carol pleit ervoor dat elk kind de kans moet krijgen de waardevolle wereld van het coderen te leren kennen. Ze heeft talloze tv-shows gepresenteerd over wetenschap en techniek op de BBC, ITV en Channel 4. Ze was 26 jaar lang medepresentatrice van *Countdown* op Channel 4 en werd daardoor bijna de bestverkopende vrouwelijke non-fictieauteur vanaf 2000. En ook uit haar adviserende rol aan de Britse premier David Cameron over de toekomstige potentie van wiskundeonderwijs bleek overduidelijk haar passie en toewijding om op een spannende en voor iedereen te begrijpen manier wiskunde, wetenschap en techniek over te brengen op het grote publiek.

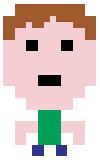
DR. JON WOODCOCK heeft natuurwetenschappen gestudeerd aan de University of Oxford en is doctor *Computational Astrophysics* aan de University of London. Hij begon met coderen toen hij acht was heeft alle soorten computers geprogrammeerd, van enkelchips microcomputers tot wereldwijde supercomputers. Tot zijn vele projecten behoren gigantische ruimtesimulatoren, onderzoek voor hightechbedrijven en intelligente robots die gemaakt zijn van afvalmaterialen. Jon heeft een passie voor wetenschappelijke en technologische educatie en geeft op scholen lezingen over ruimtecomputertechnologie en computerprogrammeren. Hij heeft aan talloze boeken over wetenschap en technologie meegewerkt.

SEAN McMANUS leerde programmeren toen hij negen was. Zijn eerste programmeertaal was Logo, nu is hij een ervaren schrijver en journalist op technologisch gebied. Tot zijn boeken behoren *Scratch Programming in Easy Steps*, *Web Design in Easy Steps* en *Raspberry Pi For Dummies*. Op zijn website www.sean.co.uk vind je Scratch-games en handleidingen.

CRAIG STEELE is gespecialiseerd in computerwetenschappelijk onderwijs. Hij is projectmanager bij CoderDojo Scotland, dat gratis toegankelijke ruimtes beheert waar jongeren kunnen coderen. Hij heeft gewerkt bij de Scottish Qualification Authority, Glasgow Science Centre en de University of Glasgow. Craigs eerste computer was een ZX Spectrum.



CLAIRE QUIGLEY studeerde computerwetenschappen in Glasgow, waarin ze bachelor en doctor werd. Ze heeft gewerkt op de computerafdeling van de Cambridge University en aan een project over het ontwikkelen van computervaardigheden bij basisschoolkinderen. Ze is mentor bij CoderDojo Scotland, een codeerclub voor jongeren.



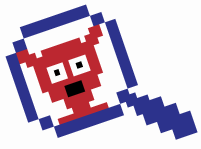
DANIEL McCAFFERTY heeft computerwetenschappen gestudeerd aan de University of Strathclyde. Daarna heeft hij software ontwikkeld voor de grootste investeringsbanken ter wereld. In zijn vrije tijd begeleidt hij kinderen bij CoderDojo in Schotland, een codeerclub voor jongeren.

Inhoud

4 COMPUTERS VANBINNEN

5 PROGRAMMEREN IN HET ECHT

Lees meer op:
www.dk.com/computercoding
(Engelstalig)



Voorwoord

Nog maar een paar jaar geleden leek programmeren iets geheimzinnigs dat alleen maar door specialisten kon worden gedaan. Veel mensen konden zich niet voorstellen dat programmeren leuk kon zijn. Maar ineens werd alles anders. Internet, e-mail, sociale netwerken, smartphones en apps overvielen ons als een wervelwind en veranderden ons leven compleet.

Computers zijn niet meer weg te denken uit ons leven. We bellen nauwelijks meer iemand op, maar sturen een sms of gebruiken de sociale media. Shoppen of vermaak, nieuws of games, alles doen we tegenwoordig via de computer. Maar we kunnen nog veel meer dan alleen gebruik maken van de computertechniek, we kunnen die ook zelf maken. Als we leren coderen, kunnen we zelf onze digitale meesterwerken maken.

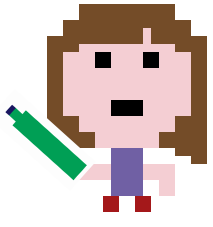
Alles wat computers doen wordt bepaald door regels met codes die iemand op een toetsenbord heeft ingevoerd. Het lijkt dan wel een vreemde taal, maar het is een taal die iedereen snel kan leren. Volgens velen is coderen een van de belangrijkste dingen die je je in de 21ste eeuw kunt aanleren.

Leren coderen is ontzettend leuk, want je ziet meteen resultaat, ook al heb je nog veel te leren. Het is zelfs zo leuk om games en programma's te maken dat je niet eens merkt dat je er moeite voor moet doen als je er eenmaal mee bezig bent. En het is creatief – misschien wel de eerste wetenschap die kunst, logica, verhalen en business combineert.

Bovendien is coderen geweldig voor je ontwikkeling. Je leert logisch te denken en je kunt sneller problemen oplossen – heel belangrijk in veel verschillende gebieden in je leven, van wetenschap en techniek tot medicijnen en rechten, maar ook in creatieve beroepen. Het aantal banen waarvoor kennis van coderen nodig is, zal in de toekomst enorm toenemen; er is nu al een groot tekort aan goede codeerders. Leer coderen en de digitale wereld ligt aan je voeten!

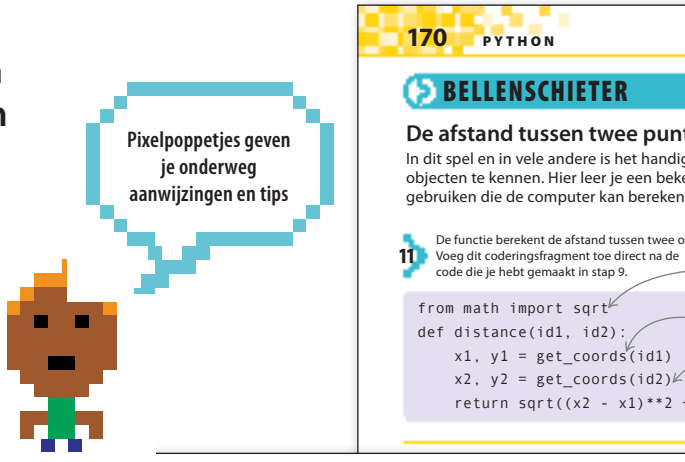
Carol Vorderman

CAROL VORDERMAN



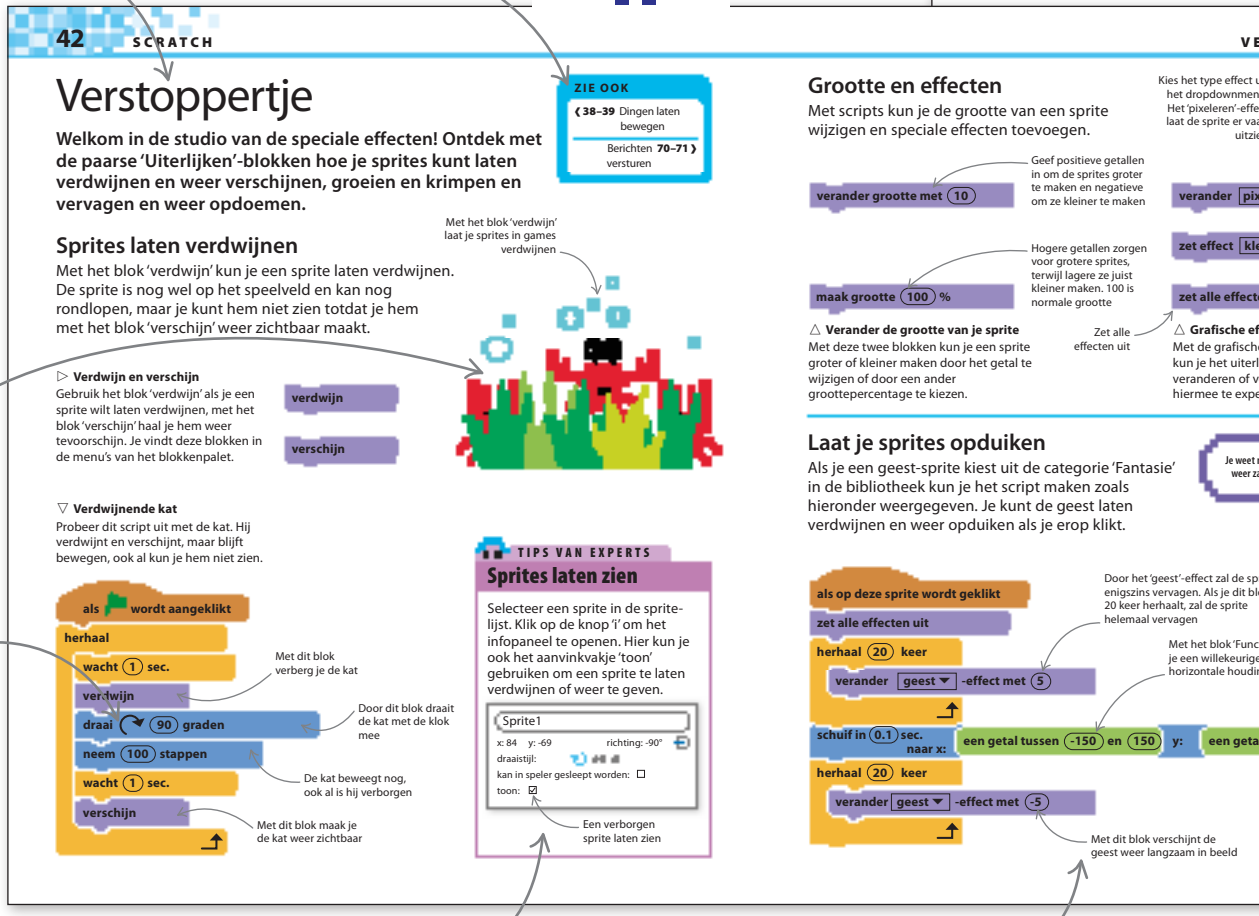
Zo werkt dit boek

In dit boek leer je alles wat nodig is om te kunnen coderen. Met allerlei leuke projecten breng je dit in de praktijk. We hebben alles in kleine stukken opgedeeld, zodat je het gemakkelijk kunt volgen en begrijpen.



Elk onderwerp wordt in detail beschreven met voorbeelden en oefeningen

In 'Zie ook'- kaders vind je andere onderwerpen over het thema



Kleurrijke illustraties wijzen je op verschillende programmeer-ideeën

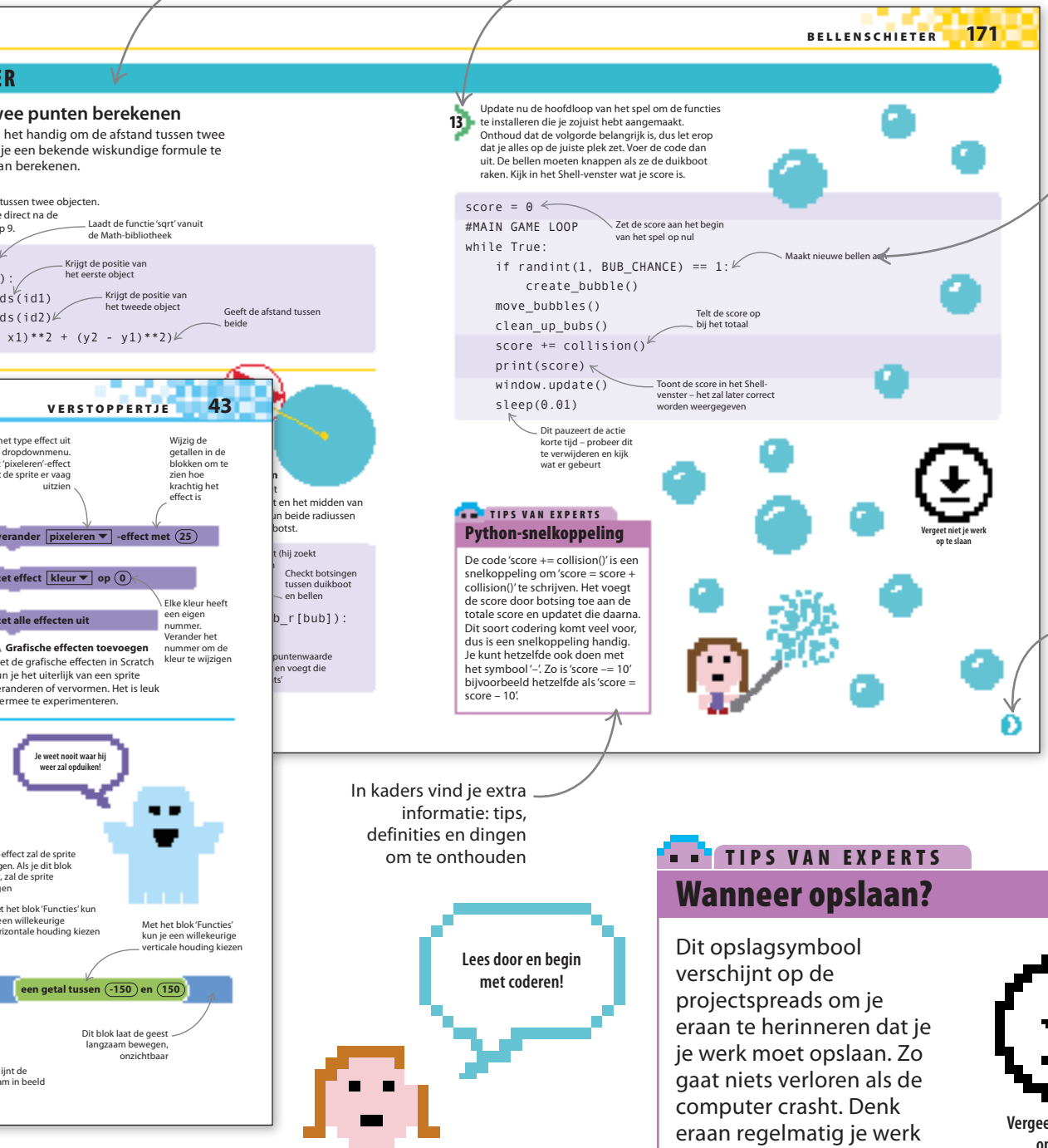
Programmeer-scripts en codes worden stap voor stap uitgelegd

Instructies om te klikken, slepen of selecteren

Labels helpen je elke stap te begrijpen

Via zeven projecten bouw je je codeervaardigheden op. Projectpagina's vallen op door de blauwe balk

Via eenvoudige instructies word je stap voor stap door elk project geleid



Elke coderingsregel heeft een duidelijk label, dus je kunt je niet vergissen

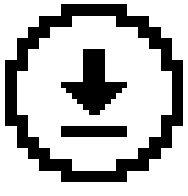
Dit icoontje geeft aan dat het project op de volgende bladzijde verder gaat

In kaders vind je extra informatie: tips, definities en dingen om te onthouden

TIPS VAN EXPERTS

Wanneer opslaan?

Dit opslagsymbool verschijnt op de projectspreads om je eraan te herinneren dat je je werk moet opslaan. Zo gaat niets verloren als de computer crasht. Denk eraan regelmatig je werk op te slaan.



Vergeet niet je werk op te slaan

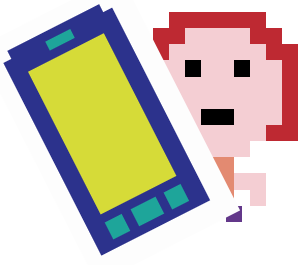
Wat is een computerprogramma?

ZIE OOK
Denk als een computer **16–17** ❯
Word een codeerder **18–19** ❯

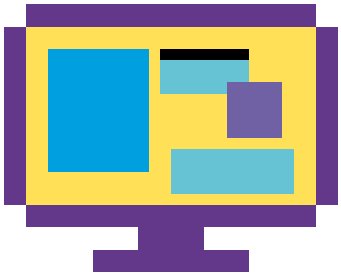
Een computerprogramma is een reeks instructies die een computer volgt om een taak uit te voeren. 'Coderen' of 'programmeren' wil zeggen dat je de stap-voor-stapinstructies schrijft die de computer vertellen wat hij moet doen.

Computerprogramma's zijn overal

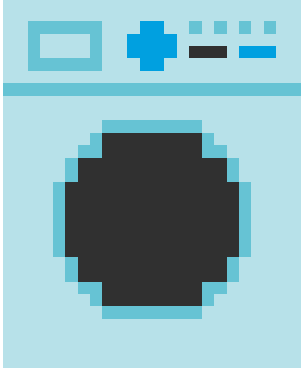
We worden omgeven door computerprogramma's. Veel apparaten die we elke dag gebruiken worden erdoor aangestuurd. Al deze machines volgen stap-voor-stap-instructies die een computerprogrammeur heeft geschreven.



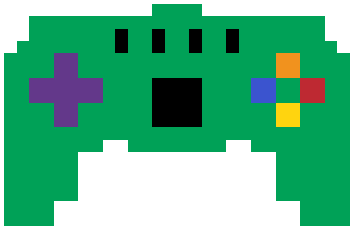
◁ **Mobiele telefoons**
Dankzij programma's kun je bellen en berichten versturen. Als je in je contactenlijst zoekt, vindt een programma het juiste nummer voor je.



△ **Computersoftware**
Alles wat een computer doet, van browsen op het internet tot tekstverwerken of muziek afspelen, werkt door coderingen die een computerprogrammeur heeft ingevoerd.

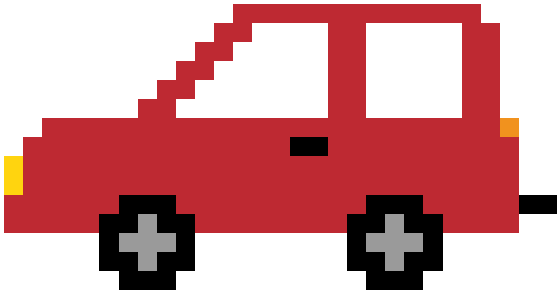


△ **Wasmachines**
Wasmachines zijn geprogrammeerd om verschillende wasprogramma's uit te voeren. Via computercoderingen worden de temperatuur van het water en de duur van de wasbeurt geregeld.



◁ **Games**
Consoles zijn gewoon ook een soort computers en alle games die je erop speelt, zijn programma's. Alle afbeeldingen, geluiden en besturingen zijn in computercodes geschreven.

▷ **Auto's**
In veel auto's controleren computerprogramma's de snelheid, temperatuur en het benzinepeil in je tank. Ze kunnen voor de veiligheid van de bestuurder zelfs checken of de remmen nog goed werken.

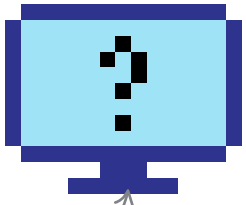


Zo werkt een computerprogramma

Computers lijken dan wel heel slim, maar het zijn in feite gewoon dozen die heel snel en heel precies instructies opvolgen. Omdat mensen intelligent zijn, gebruiken ze computers om bepaalde taken te verrichten aan de hand van geschreven programma's, lijsten of instructies.

1 Computers kunnen niet denken

Een computer zal niets uit zichzelf doen. Het is aan de programmeur om de computer te instrueren.



Zonder instructies is een computer stuurloos

2 Schrijf een programma

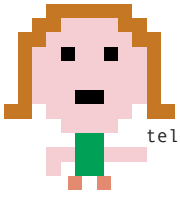
Je kunt een computer zeggen wat hij moet doen door hem een gedetailleerde reeks instructies te geven, een programma. Elke instructie moet zo beperkt zijn dat de computer hem kan begrijpen. Kloppen de instructies niet, dan zal de computer niet doen wat jij wilt.

Dit computerprogramma telt af om van start te gaan

```
tellen binnen bereik (10, 0, -1):
    print ('aftellen', tellen)
```

3 Programmeertalen

Computers kunnen alleen instructies opvolgen die in een taal geschreven zijn die ze begrijpen. De programmeur bepaalt welke taal voor een bepaalde taak het geschiktst is.



```
tellen binnen bereik (10, 0, -1):
print ('aftellen', tellen)
```

Alle programma's worden uiteindelijk geconverteerd in een 'binaire code', een standaardcomputertaal die uit nullen en enen bestaat

LINGO

Hardware en software

'Hardware' wil zeggen de materiële delen van de computer die je kunt zien of aanraken (alle bedrading, de stroomkring, het circuit, het toetsenbord, het scherm enz.). De 'software' zijn de programma's die in de computer draaien en regelen hoe de computer werkt. Software en hardware zorgen er samen voor dat de computer nuttige dingen kan doen.

Denk als een computer

Een programmeur moet leren denken als een computer. Alle taken moeten in kleine stukjes worden opgedeeld, zodat ze makkelijk te volgen zijn en onmogelijk verkeerd te begrijpen.

ZIE OOK
❮ 14–15 Wat is een computerprogramma?
Word 18–19 ❯ een codeerder

Denk als een robot

Stel je voor dat de ober in het café een robot is. De robot heeft een simpel computerbrein en je moet hem vertellen hoe hij de bestelling vanuit de keuken naar de gasten aan tafel kan brengen. Als eerste moet je nu het proces in eenvoudige taken verdelen die de computer kan begrijpen.

LINGO

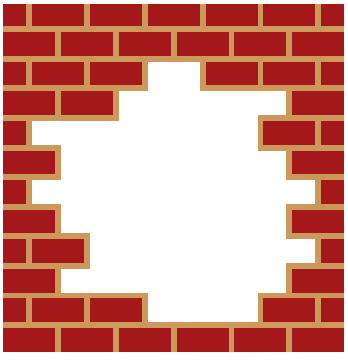
Algoritme

Een algoritme is een reeks eenvoudige instructies om een taak uit te voeren. Een programma is een algoritme dat vertaald is in een taal die computers kunnen begrijpen.

1 Ober-robot programma 1

Met dit programma grijpt de robot het eten van het bord, ramt dwars door de keukenmuur naar de eetzaal en zet het voedsel op de grond. Het algoritme was niet gedetailleerd genoeg.

1. Pak het eten op
2. Loop van de keuken naar de eettafel
3. Zet het eten neer

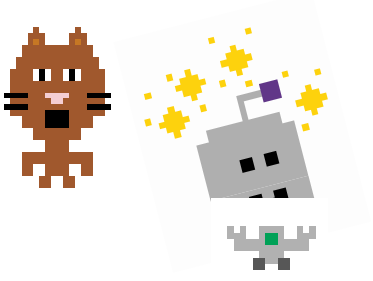


◁ **Een ramp!**
De instructies waren niet duidelijk: we zijn vergeten de robot te vertellen dat hij door de deur moet lopen. Voor mensen mag dat duidelijk zijn, maar computers kunnen niet zelf denken.

2 Ober-robot programma 2

Deze keer hebben we de ober-robot verteld dat hij de deur moet gebruiken. Dat doet hij nu ook, maar dan struikelt hij over de kat en valt het bord op de grond.

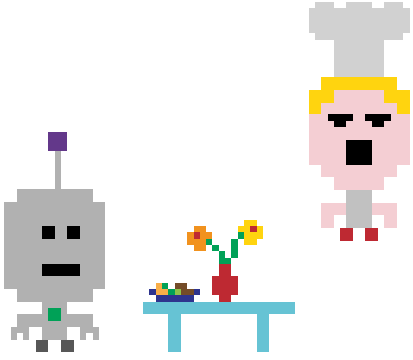
1. Pak een bord met eten op
2. Loop van de keuken naar de eettafel via:
   - Loop naar de deur tussen keuken en eetzaal
   - Loop van de deur naar de tafel
3. Zet het bord vóór de gast op tafel



△ **Nog niet perfect**
De robot weet niet hoe hij met obstakels als een kat moet omgaan. Dus moet het programma nog preciezere instructies geven, zodat hij veilig kan lopen.

3 **Ober-robot programma 3**
Met deze versie van het programma brengt de robot het eten succesvol bij de gast en omzeilt alle obstakels. Maar nadat hij het bord heeft neergezet, blijft hij staan terwijl andere borden zich ophopen in de keuken.

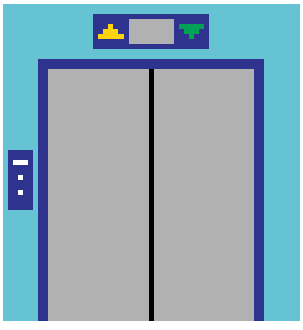
```
1. Pak een bord met eten op en houd dit de hele tijd omhoog
2. Loop van de keuken naar de eettafel via:
    Loop naar de deur tussen keuken en eetzaal
        Kijk of er obstakels zijn en loop eromheen
    Loop van de deur naar de tafel
        Kijk of er obstakels zijn en loop eromheen
3. Zet het bord vóór de gast op tafel
```



△ **Eindelijk goed zo?**
Eindelijk lukt het de robot het eten succesvol af te leveren. Maar we zijn vergeten hem te instrueren terug te gaan naar de keuken om het volgende bord te pakken.

Voorbeeld uit de echte wereld

De ober-robot mag dan fantasie zijn, maar dit soort algoritmes doen hun werk overal om ons heen. Een computergestuurde lift bijvoorbeeld heeft met precies dezelfde problemen te maken. Moet hij omhoog of omlaag? Naar welke verdieping moet hij hierna?



```
1. Wacht tot de deuren dicht zijn
2. Wacht tot er op een knop is gedrukt
    Als de gekozen verdieping hoger ligt dan de huidige:
        Beweeg de lift omhoog
    Als de gekozen verdieping lager ligt dan de huidige:
        Beweeg de lift omlaag
3. Wacht tot verdieping gelijk is aan gekozen bestemming
4. Open deuren
```

◁ **Liftprogramma**
Om de lift correct en veilig te laten werken moet elke stap precies en duidelijk zijn en elke andere mogelijkheid uitsluiten. De programmeurs moeten zeker weten dat ze een geschikt algoritme maken.