

D.B.Baarda / A.van Londen

# Leren omgaan met de ZX SPECTRUM

BASIC voor jonge computergebruikers



Kluwer Technische Boeken

Leren omgaan  
met de  
**ZX Spectrum**



# Leren omgaan met de **ZX Spectrum**

*BASIC voor jonge  
computergebruikers*

Drs. D. B. Baarda  
en  
Drs. A. van Londen



**Kluwer Technische Boeken B.V. - Deventer - Antwerpen**



Omslag en illustraties: W. Niessink

ISBN 90 201 1766 1

D/1985/0108/166

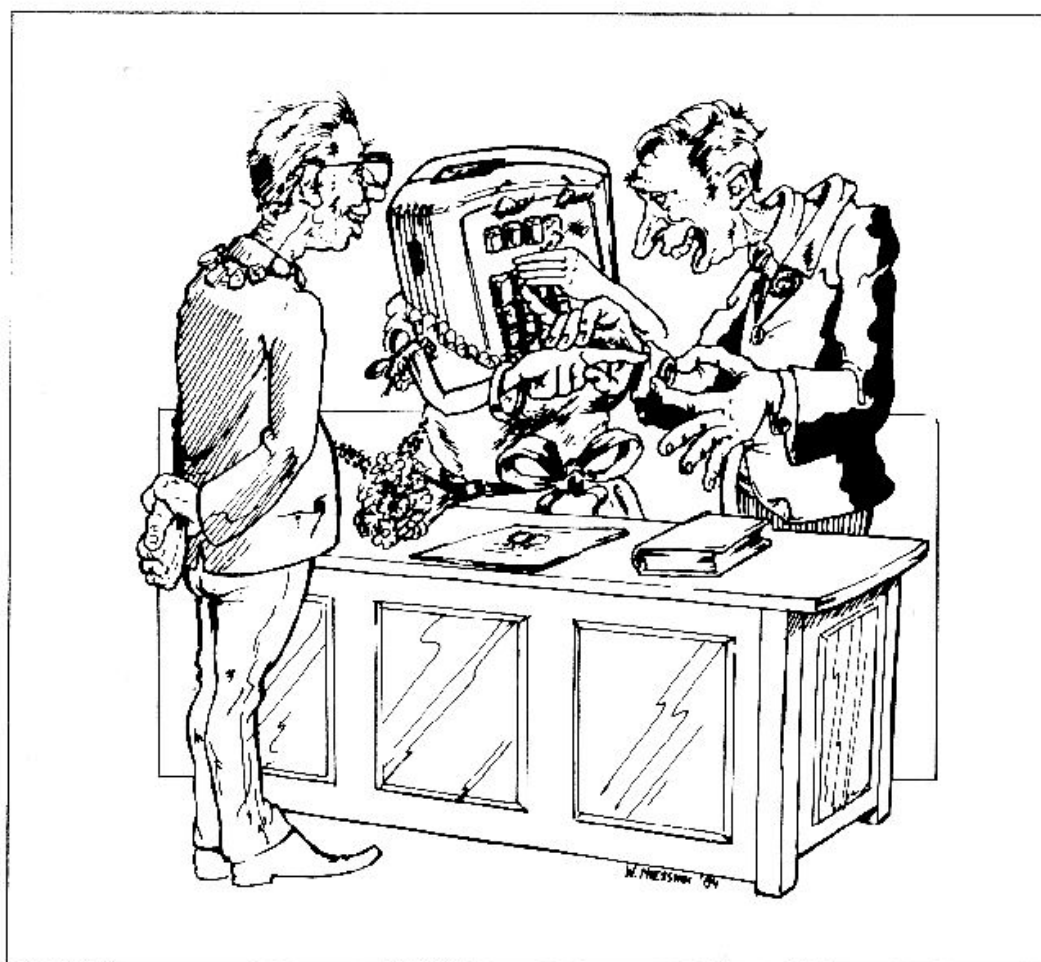
© 1985 Kluwer Technische Boeken B.V. – Deventer

1e druk 1985

Niets uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke andere wijze ook, zonder voorafgaande schriftelijke toestemming van de uitgever.

No part of this book may be reproduced in any form, by print, photoprint, microfilm or any other means without written permission from the publisher.

Ondanks alle aan de samenstelling van de tekst bestede zorg, kan noch de redactie noch de uitgever aansprakelijkheid aanvaarden voor eventuele schade, die zou kunnen voortvloeien uit enige fout, die in deze uitgave zou kunnen voorkomen.



*Met dank aan:*

Arleen  
Andre  
Christian  
Jonathan  
Mariska  
Oscar  
Raf  
Ruth

De programma's zijn geschikt voor de meeste BASIC-versies en  
getest op de ZX Spectrum



# INHOUD

## Waarover gaat dit boek?

### Deel 1

#### WAT KAN EEN COMPUTER?

Toetsenbord. . . . .	13
Hoofdletters, kleine letters en andere tekens . . . . .	19
Rekenen op de computer. . . . .	22
Kleuren met de computer . . . . .	28
Geluid op de ZX Spectrum . . . . .	30

### Deel 2

#### PROGRAMMEREN IN BASIC

Klaar om te programmeren? . . .	35
NEW. . . . .	37
CLS . . . . .	38
LIST. . . . .	38
RUN. . . . .	39
BREAK/SPACE . . . . .	40
CONT . . . . .	40
Fouten verbeteren in een programma . . . . .	41

#### LEREN PROGRAMMEREN

Inleiding . . . . .	45
PRINT. . . . .	47
INPUT . . . . .	53
LET . . . . .	58
GO TO. . . . .	61
GO SUB EN RETURN. . . . .	65
REM. . . . .	71
IF ... THEN . . . . .	74

FOR ... NEXT. . . . .	78
READ en DATA . . . . .	84
RND. . . . .	90

### Deel 3

#### PROGRAMMA'S

Rekenen met de computer . . . . .	97
Toetsenbord. . . . .	102
Regenboog . . . . .	103
Toonladder . . . . .	103
Old Mac Donald . . . . .	104
Happy Birthday. . . . .	104
Grootste Gemene Deler . . . . .	104
Uitrollende dobbelsteen . . . . .	105
Twee dobbelstenen . . . . .	105
Getallen omkeren. . . . .	105
Namen omkeren . . . . .	106
Huiswerkhulp. . . . .	107
Priemgetallen . . . . .	108
Rekenkundige tafels . . . . .	108
Telefoonboek . . . . .	109
Overhoren tafels . . . . .	109
Nonsens generator . . . . .	110

### Bijlage 1

#### DE CASSETTERECORDER

Bewaren. . . . .	113
Laden . . . . .	114

### Bijlage 2

#### ANTWOORDEN OP DE VRAGEN 115



# WAAROVER GAAT DIT BOEK?

Het boek dat je nu begint te lezen gaat over het omgaan met computers en het is speciaal voor kinderen geschreven. (Dit betekent niet, dat een volwassene er niks uit kan leren!) We leren je om zelf de ZX Spectrum te gebruiken. Als het goed gaat heb je er geen volwassene bij nodig.

In dit boek kun je BASIC leren. Dit is een computertaal. Daarover gaat het tweede deel van dit boek. Als je al kunt omgaan met het toetsenbord, kun je beginnen met het hoofdstuk 'Klaar om te programmeren?'. Anders kun je beter vooraan beginnen en in ieder geval 'Toetsenbord' en 'Hoofd, kleine letters en andere tekens' lezen voordat je aan het tweede deel begint.

Het eerste deel van het boek behandelt hoe je met het toetsenbord (de toetsen) moet omgaan en hoe je kleuren en geluid kunt maken met de computer. Zo gebruik je de computer als een soort schrijf- of tekenpapier, als kleurdoos of muziekinstrument. We leren je welke toetsen je hiervoor moet indrukken. Het verschil met papier is alleen, dat je woordjes of sommetjes weg zijn als je de computer uitzet en dat is soms jammer.

In het tweede deel kun je leren, hoe je de computer ook voor andere zaken kunt gebruiken. Daarbij kun je denken aan het onthouden en bewaren van dingen (bijvoorbeeld het opstel dat je wilt gaan schrijven of de tekening die je gaat maken, of het muziekstukje).

En hij kan ook snel karweitjes voor je uitvoeren waar je zelf veel meer tijd voor nodig hebt, zoals een moeilijke som uitrekenen, duizend strafregels schrijven, je huiswerkhulp zijn en spelletjes ontwerpen. Maar dit kan de computer niet zonder jouw hulp. Jij moet hem vertellen, wat je wilt dat hij doet. Je schrijft de computer daarvoor een soort 'brief' via het toetsenbord. Zo'n 'brief' noem je een 'programma'. Je kunt niet beginnen met

'Beste computer, luister eens...', want dat snapt hij niet. Je moet het programma in zijn eigen taal schrijven, een computertaal. De computertaal die wij je leren heet 'BASIC'. BASIC is een schrijftaal. Er is niemand die BASIC spreekt. Er worden alleen computerprogramma's in geschreven. Dit leer je in het tweede deel:

#### **PROGRAMMEREN IN BASIC.**

In het derde deel vind je voorbeelden van programma's die al gemaakt zijn. Je kunt ze over typen. Als je daarna RUN typt en vervolgens **ENTER** indrukt, doet de computer wat er in het programma staat. Sommige van deze programma's kun je in deel 1 al gebruiken.

DEEL 1

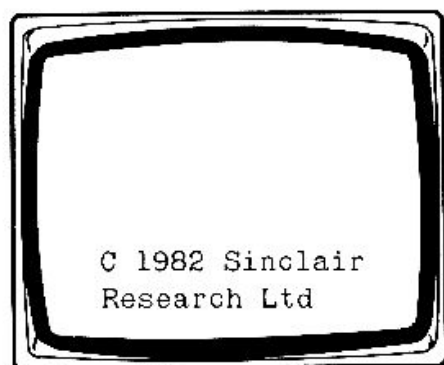
# **WAT KAN EEN COMPUTER?**





# TOETSENBORD

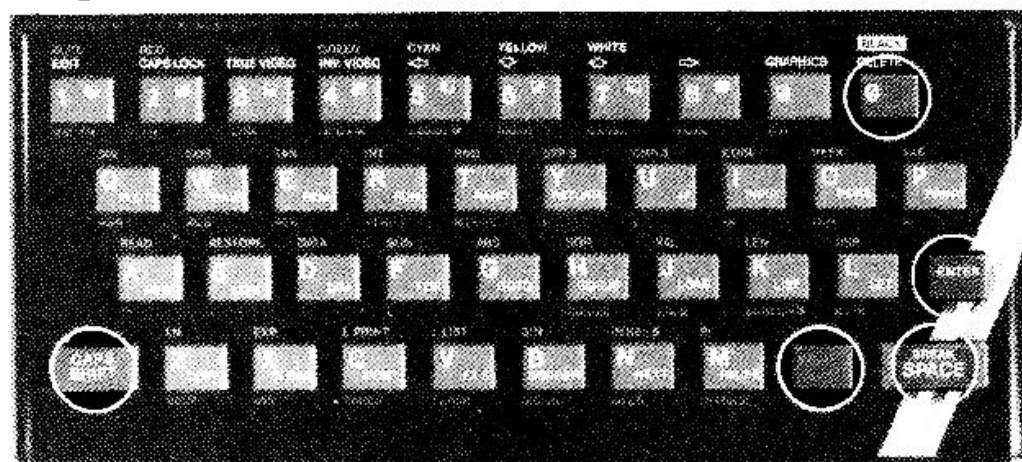
Het toetsenbord van de ZX Spectrum is heel kleurrijk. Je ziet toetsen met witte, rode en groene tekens. Witte letters, witte woordjes, rode woordjes, groene woordjes; een regenboog lijkt het wel. We gaan snel kijken hoe je al die tekens op het TV-scherm kunt krijgen. Als je de computer aanzet zie je als het goed is:



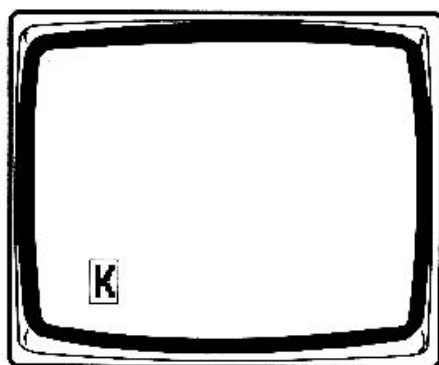
Als dit bij jou niet zo is, kijk dan nog eens of alles goed aangesloten is.

Je ziet niet zoals bij een gewoon TV-toestel een testbeeld. Wat je ziet is een TV-scherm waar jij letters, cijfers en andere tekens op kunt krijgen via het toetsenbord. Het lijkt dus gewoon op een blanco velletje papier.

We gaan er nu iets op zetten.



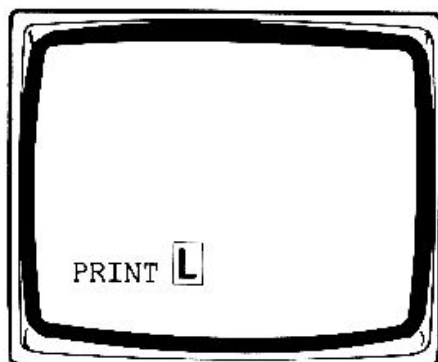
**ENTER** Zie je de toets met **ENTER** er op? Druk daar maar op.



- cursor

Nu staat er links onderin een blokje te knippen met een K erin. Daarmee geeft de computer aan, dat jij aan de beurt bent om iets op het scherm te zetten. Als de computer zelf iets gaat schrijven, verdwijnt dat blokje (de cursor heet die in het Engels). In dit deel van het boek zullen we vooral kijken wat jij op het scherm kunt krijgen. Hoe je er voor zorgt dat de computer iets gaat doen leer je in het tweede deel. Maar het blokje staat er, dus jij kunt iets typen op de plek waar het blokje staat. Druk maar eens op **P**.

Heb je dat gedaan? Dan staat er als het goed is op het TV-scherm.



Nu staat er een L in het knipperende blokje.

Druk nog eens op **P**. Je ziet dat er nu een p naast staat. Het blokje schuift iedere keer een plekje naar opzij als je iets intypt.

Het blokje is een soort stuur.

Druk eens op de **A**.

Druk eens op de **B**.

Druk eens op de **1**.

Houd de **C** eens een paar tellen ingedrukt.

Je krijgt nu een hele rij ccccccc. Als je een toets een paar tellen ingedrukt houdt, gaat hij snel door.

Zo, je ziet dat je zowel één letter of cijfer als een hele reeks op het TV-scherm kunt krijgen.

Nu moet je de computer even uit en weer aan zetten.

Druk maar weer op **ENTER**.

Het blokje (de cursor) bevat de letter K. Dat is een afkorting van Engels voor Keyword. Dat betekent 'een belangrijk woord'. Opdrachten zullen we die noemen. Als meneer Sinclair een Nederlander was geweest had hij er vast een O in gezet in plaats van een K.

*Als er een K in de cursor staat kun je de witte woordjes die op de toetsen staan op het TV-scherm krijgen. Opdrachten heten die.*

Druk maar eens op **G** en floep GO TO verschijnt.

Voordat je verder kunt gaan met de andere woordjes moet je even weten hoe je weer een K in de cursor krijgt. Je kunt altijd de computer uitzetten maar dat is zeer onhandig. Nee laten we het met

**CAPS SHIFT** en **DELETE** doen.

**DELETE** Als je op **CAPS SHIFT** en **DELETE** tegelijk drukt zie je dat de cursor teruggaat. GO TO is verdwenen en er staat weer een K in de cursor.

Probeer het volgende eens:

Druk op

**P** en **CAPS SHIFT** en **DELETE**

**I** en **CAPS SHIFT** en **DELETE**

**L** en **CAPS SHIFT** en **DELETE**

**K** en **CAPS SHIFT** en **DELETE**

Druk eens op **P**, daarna nergens op drukken. Eerst even doorlezen.

Je ziet nu PRINT **L** . De letter L staat in de cursor. Dat is een afkorting voor 'letters'.

*Met de L in de cursor kun je allerlei tekens (letters, cijfers, punten, komma's, enz) op het TV-scherm krijgen.*

## ROOD

Maar eerst moeten we je nog leren hoe je de rode tekens op het scherm kunt krijgen. Een belangrijk rood teken is het aanhalingsteken " . Dit vind je op de P-toets in het rood.

Hoe krijg je die nu?

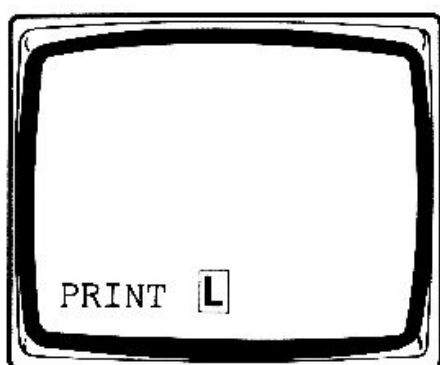
Met de rode **SYMBOL SHIFT** toets.

---

Met	<b>SYMBOL SHIFT</b>	krijg je de	rode letters en tekens op de toetsen	in het blokje staat L
-----	---------------------	-------------	--	-----------------------------

---

Druk nu op **SYMBOL SHIFT** en **P** , nu verschijnt " .



aanhalingsteken

Kijk nu eens of je de volgende tekens op het scherm kunt krijgen:

" . . . , , , , ! ! ! ! ! \$ \$ \$ \$ \$ \* \* \* \* \* " als laatste dus weer een " .

Druk nu weer op **ENTER** .

De regel verschijnt nu op het TV-scherm.

Als je nu op **ENTER** drukt staat er in de cursor weer een K.

Is het niet gelukt begin dan nog eens bij DELETE.  
Probeer nu het volgende eens met PRINT.

```
PRINT 10 ENTER  
PRINT 123 ENTER  
PRINT "huis" ENTER
```

Je ziet dat de computer wat er achter PRINT staat op het scherm schrijft. Als het een woord is moet dit tussen " en " staan.

## SPACE

Tussen woorden zie je vaak witte stukjes staan; spaties heten die. Net als op een typemachine kun je die ook op het TV-scherf krijgen. Gewoon door op **BREAK SPACE** te drukken. Die toets vind je in de gekleurde streep rechts.

```
PRINT "huis ?" ENTER  
PRINT " " ENTER  
PRINT "$$$$$$" ENTER  
PRINT "+++++" ENTER  
PRINT "*****" ENTER
```

## CLS

druk op **✓** er verschijnt CLS druk op **ENTER** en whoeps ... alles is weg.

Je hebt nu al een paar keer onder aan het TV-scherf OK en een paar cijfertjes zien staan. Het woordje OK betekent dat het goed was wat je deed en de computer zegt er mee dat je een nieuwe regel kunt intypen.

Je hebt al heel veel geleerd. We zullen het nog eens op een rijtje zetten wat je al geleerd hebt.

## K

Is de cursor. Een knipperend blokje dat een soort stuur is. Als er een K in de cursor staat kun je woordjes als PRINT, GO TO, NEW, enz. op het TV-scherf krijgen. Dit zijn dan de opdrachten die je geeft.

## L

Als er een L in de cursor staat, kun je alle witte tekens op het TV-scherf krijgen. De a, b, c, 1, 2, 3, enz.

## SYMBOL

Rode tekens.

## SHIFT

De rode tekens als " \$ , . ? ( ) krijg je door **SYMBOL SHIFT** ingedrukt te houden en op de toets met het teken te drukken.

- DELETE** Door op **SHIFT** en **DELETE** tegelijk te drukken gaat de cursor naar links en stuft hij gelijk dat tekentje uit. Houd je die toetsen een paar tellen vast, dan gaat hij snel uitvegen.
- CLS** Door vanuit de K-cursor op **V** te drukken komt er CLS op het TV-scherm. Wanneer je nu op **ENTER** drukt wordt het TV-scherm weer schoon.
- ENTER** Dat is de toets waarop je moet drukken als je klaar bent met een regel en je wilt dat de computer die leest en uitvoert.

## opgaven

- 1** Maak eerst het scherm schoon. En typ dan het volgende huisje.

PRINT "        * "	druk op	<b>ENTER</b>
PRINT "    *        * "	druk op	<b>ENTER</b>
PRINT "+                +"	druk op	<b>ENTER</b>
PRINT "+                +"	druk op	<b>ENTER</b>
PRINT "+                +"	druk op	<b>ENTER</b>
PRINT "-----"	druk op	<b>ENTER</b>

- 2** Maak het scherm weer schoon. En teken dan het beest dat je hieronder ziet staan. Het is een soort computerhond.

PRINT "                    \$ "	druk op	<b>ENTER</b>
PRINT "                    \$ S "	druk op	<b>ENTER</b>
PRINT "\$                    \$ \$ \$ "	druk op	<b>ENTER</b>
PRINT "    \$ \$ \$ \$ \$ "	druk op	<b>ENTER</b>
PRINT "    \$ \$ \$ \$ \$ "	druk op	<b>ENTER</b>
PRINT "    \$ \$ \$ \$ \$ "	druk op	<b>ENTER</b>
PRINT "    \$                \$ "	druk op	<b>ENTER</b>
PRINT "    \$                \$ "	druk op	<b>ENTER</b>

- 3** Maak het scherm schoon. En teken een boom, boot, auto of iets wat je zelf bedenkt.

# HOOFDLETTERS, KLEINE LETTERS EN ANDERE TEKENS

De computer heeft tot nu toe de opdrachten in hoofdletters geschreven. De rest stond in kleine letters. Ook daar kun je hoofdletters van maken. Dat gaat eigenlijk net als op een typemachine. Laten wij maar eens:

*dit is mijn eerste tekst*

op het TV-scherf zien te krijgen.

Als er een K in de cursor staat druk je op de **P** en PRINT verschijnt. De " krijg je door **SYMBOL** en **P** in te drukken.

## CAPS SHIFT

Als je nu op **Q** zou drukken komt er een kleine letter en dat willen we niet. Door eerst op **CAPS SHIFT** te drukken en dan op **Q** krijg je de hoofdletter. Je moet de **CAPS SHIFT** wel vast blijven houden als je op de letters drukt. Zie je de CAPS/SHIFT-toets? Ja, helemaal links onderaan. Als je die toets ingedrukt houdt krijg je hoofdletters. Als je klaar bent, moet je eerst weer op " (aanhalingsteken op de letter P) drukken en dan op **ENTER**. Als het goed gaat staat er nu:

DIT IS MIJN EERSTE TEKST




Je hoeft niet de SHIFT-toets ingedrukt te houden als je eerst op **CAPS SHIFT** en tegelijk op **CAPS LOCK** drukt. Kun je die toets vinden (boven de 2)? Als je daar op drukt komt er een C in de cursor. C is een afkorting van Capital en dat is Engels voor hoofdletters. Elke letter die je nu typt wordt een hoofdletter. Door nogmaals op **CAPS LOCK** te drukken verdwijnt de C weer om plaats te maken voor de L in de cursor. Dan kun je weer kleine letters krijgen. Als je naar de toetsen kijkt zie je dat er niet alleen letters en woordjes op en rond de toetsen staan, maar ook allerlei tekens. We hebben er al een paar geprobeerd,



maar laten we eens kijken hoe je ze op het TV-scherm kunt krijgen.

Met	<b>SYMBOL SHIFT</b>	krijg je de	rode letters en tekens op de toetsen	in het blokje staat L
Met	<b>CAPS SHIFT</b> + <b>SYMBOL SHIFT</b>	krijg je de	groene woordjes boven de toetsen	in het blokje staat E
Met	<b>CAPS SHIFT</b> + <b>SYMBOL SHIFT</b>	krijg je met <b>SYMBOL SHIFT</b> ingedrukt	rode woordjes onder de toetsen	in het blokje staat E

Je zult de rode woordjes onder de toetsen in het boek bijna niet tegen komen. Ze worden in het begin weinig gebruikt.

**GRAPHICS** Er zijn nog een aantal tekens die wij bijna niet zullen gebruiken. Dat zijn de witte figuurtjes die je op de cijfertoetsen ziet staan. Bij de 1 staat: 

Met	<b>CAPS SHIFT</b> + <b>GRAPHICS</b>	krijg je de	witte tekens op de cijfers	in het blokje staat G
-----	---	-------------	----------------------------------	-----------------------------

De **GRAPHICS** zit bij cijfertoets 9. Als je de G in de cursor hebt en je drukt op de cijfertoetsen dan krijg je de figuurtjes, probeer het maar.

Je kunt weer letters en cijfers gaan typen als je nogmaals op **GRAPHICS** (toets 9) drukt. De L verschijnt dan weer in de cursor.

Het valt waarschijnlijk niet mee, snel de verschillende letters en tekens te vinden. Wij hebben een programma gemaakt dat je daarbij kan helpen. Het staat achterin en het heet 'TOETSENBORD'. Misschien kan iemand het even voor je intypen. Wij hebben het met opzet kort gehouden.

Het programma geeft telkens een letter of teken en jij moet die dan zo snel mogelijk intypen. Aan het eind krijg je nog te zien hoeveel je er goed had. Wij hebben alleen de letters en tekens die het meest voorkomen gebruikt.

**SCROLL?** Als het TV-scherf vol is vraagt de computer SCROLL? Als je nu een toets indrukt, het geeft niet welke, schuift hij de tekst op het scherm een stukje op, zodat er een plek komt voor nieuwe tekst. Alleen als je op de n van nee drukt stopt hij.

## opgaven

---

**1** Typ eens:

Ik leer al aardig typen. Het gaat goed!

Je moet zowel de hoofd- als de kleine letters en ook het uitroepteken (!) typen.

**2** Typ eens:

2 " 5 \$ ( ) ? + \*

**3** Oefen met het programma 'TOETSENBORD'.

**4** Maak eens een gedicht op de computer.

# REKENEN OP DE COMPUTER

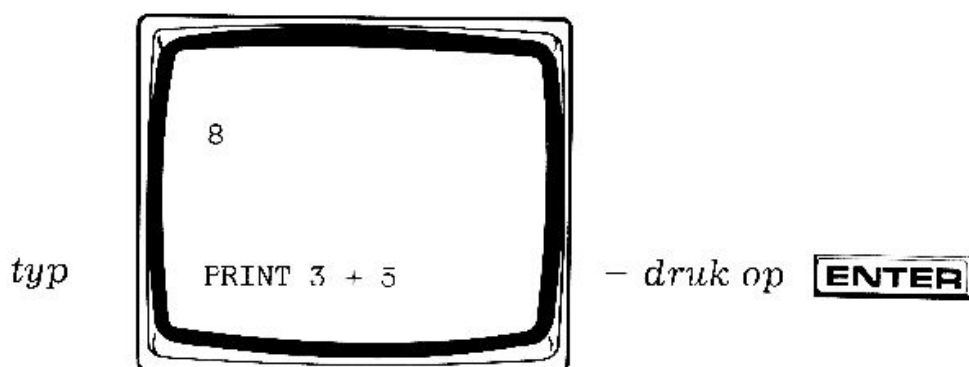
Computers werden vroeger alleen maar als rekenmachines gebruikt: dat kunnen ze veel sneller en nauwkeuriger dan mensen! Daarvoor zijn ze ook uitgevonden. Je zult zien hoe snel de computer je rekenhuiswerk kan maken.

Je hoeft alleen maar je som in te typen en hij weet het antwoord al. Als je voor je som PRINT typt, dan schrijft hij het antwoord op het TV-scherm.

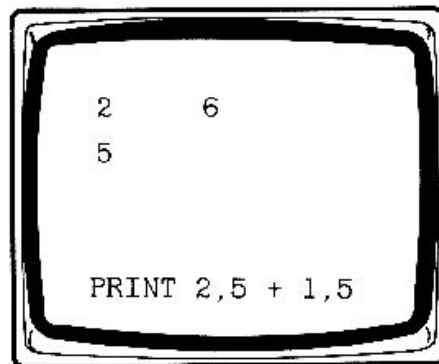
De computer gebruikt voor vermenigvuldigen en delen niet dezelfde tekens als die jij op school leert. Hieronder komt een lijstje:

	<i>BASIC wat de computer begrijpt.</i>	<i>Op school leer je.</i>
Optellen	+	+
Aftrekken	-	-
Vermenigvuldigen	*	×
Delen	/	:

Als je de computer als rekenmachine wilt gebruiken, dan typ je PRINT en daarachter de getallen en het teken voor optellen, aftrekken of iets anders. De uitkomst verschijnt boven aan het scherm. Laten we een paar voorbeelden maken.



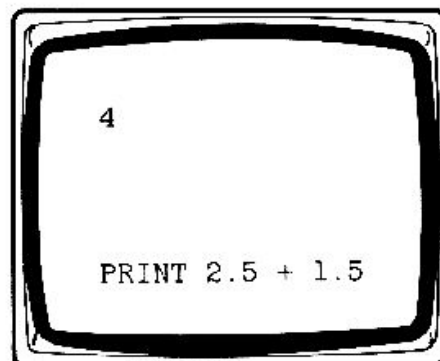
*typ*



– druk op **ENTER**

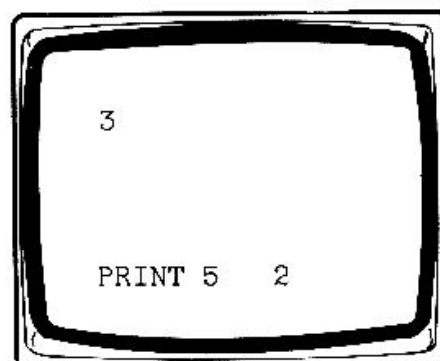
Hé, dat gaat fout. Je dacht dat er 4 zou komen en er staat nu 2 6 en 5. Hoe zou dat komen? De computer gebruikt de , (komma) voor iets anders dan waar wij hem nu voor nodig hebben. De komma betekent dat de volgende uitkomst een stukje verder op het scherm komt. Eerst de 2, dan een stukje verder 5 + 1 en weer een stukje verder 5. We hebben nu een . (punt) nodig. Dus onthouden: EEN . (PUNT) IN PLAATS VAN EEN , (KOMMA) midden in een getal.

*typ*



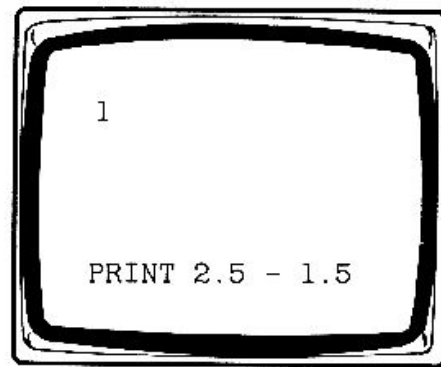
– Let op een punt  
in plaats van  
een komma

*typ*



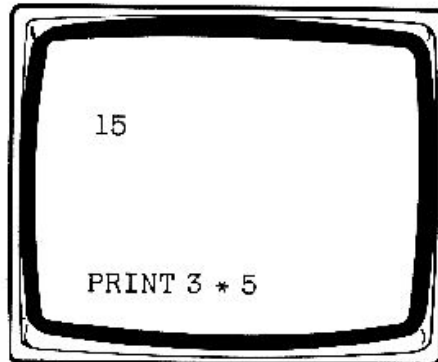
– Aftrekken

*typ*



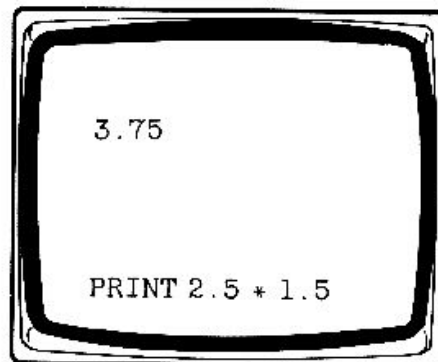
– Let op een punt  
in plaats van  
een komma

*typ*



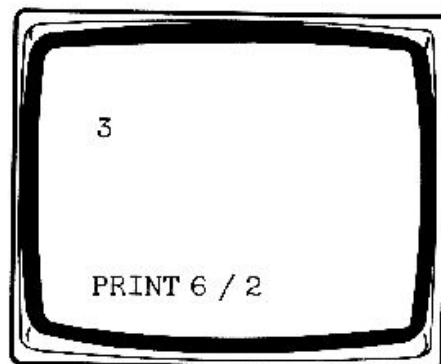
– Vermenigvuldigen.  
Let op een \*  
i.p.v. een ×

*typ*

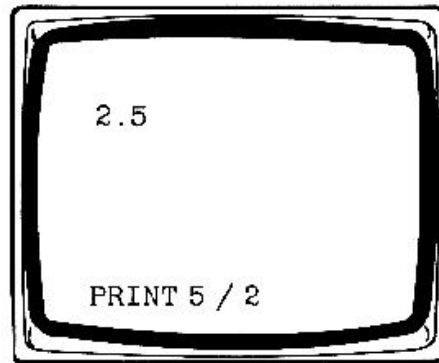


– Delen. Let op  
een / i.p.v. :.

*typ*



*typ*



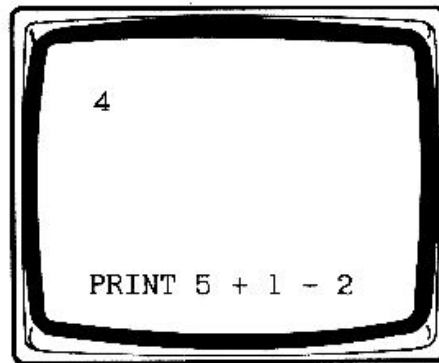
Als je deze voorbeelden gemaakt hebt, dan weet je hoe je de computer als rekenmachine kunt gebruiken.

Laten wij eens een som gaan maken:

Je hebt 5 gulden. Je krijgt er een gulden bij. Je koopt een boekje van 2 gulden. Hoeveel heb je over?

De computer kan hier snel bij helpen:

*typ*



Dus 4 gulden over. Heb je in de gaten, dat je  $5 + 1 - 2$  gewoon achter elkaar kunt zetten?

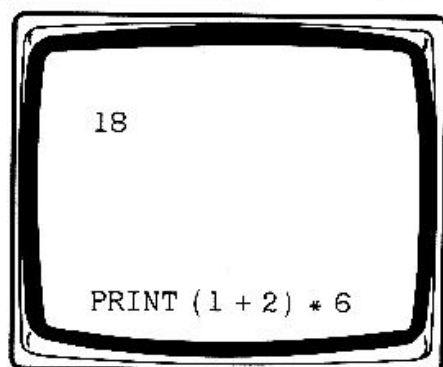
Als je langere sommen maakt, moet je net als op school, haakjes gebruiken.

*Voorbeeld:* Jan heeft 6 gulden en Kees en Ans hebben ieder 6 gulden. Samen hebben ze:

$$(1 + 2) * 6 = 18 \text{ gulden} \quad (\text{goed})$$

$$1 + 2 * 6 = 13 \text{ gulden} \quad (\text{fout})$$

*typ*



De mogelijkheden voor rekenen, boekhouden, wiskunde, etc. van de computer zijn onbegrensd. Maar dat is te veel om uit te leggen.

**De volgende fouten komen vaak voor als je gaat rekenen:**

---

- 1** Een komma in plaats van een punt getypt.

PRINT 2,5 + 1,5                   (*fout*)

Als je dit typt krijg je drie getallen (2 6 5).  
Je hebt in plaats van een punt een komma gezet.

PRINT 2.5 + 1.5                   (*goed*)

- 2** Aanhalingstekens gebruikt.

PRINT "5 + 3"                   (*fout*)  
5 + 3

Wat tussen aanhalingstekens (") wordt gezet komt precies zo op het TV-scherm. En dat was niet de bedoeling. Je wou 5 + 3 laten uitrekenen. Dus:

PRINT 5 + 3  
8                                   (*goed*)

## opgaven

---

Maak de volgende sommen. De oplossingen vind je achterin.

- 1 Hoeveel is  $7 + 6$   
 $18 + 33$   
 $9 - 3$   
 $19 - 17$   
 $2 * 4$   
 $12 * 15$   
 $4 / 2$   
 $32 / 8$
  - 2 Tel 7 en 6 bij elkaar op en trek er 3 vanaf.
  - 3 Je hebt f 2,50 zakgeld. Je geeft drie kwartjes aan je vriend. Hoeveel heb je over?
  - 4 Drie zusjes willen samen een computerspel kopen van f 50,00. Ze krijgen ieder f 12,50 van een tante. Daarvan moet f 3,00 naar de bank van hun ouders. Hoeveel moeten ze nog sparen?
-



# KLEUREN MET DE COMPUTER

Zoals je misschien zelf al ontdekt hebt kun je verschillende kleuren krijgen, als je de computer op een kleurentelevisie hebt aangesloten.

Die kleuren zitten bij de cijfertoetsen.

- 0 = zwart
- 1 = blauw
- 2 = rood
- 3 = paars
- 4 = groen
- 5 = lichtblauw
- 6 = geel
- 7 = wit

Je kunt die kleuren voor verschillende dingen gebruiken.

**INK** INK (bij de X-toets) is de inkt, daar maak je de kleur van de letters mee.

**PAPER** PAPER (bij de C-toets) maakt de kleur van het papier.

**BORDER** BORDER (op de B-toets) maakt de kleur van de rand van het scherm.


Als je zou typen INK 4, krijg je dus groene inkt en met PAPER 6 komt het op geel papier.

Wij moeten nog even vertellen hoe je INK en PAPER kunt typen. Het zijn rode woordjes onder de toetsen.

---

		<i>krijg je</i>	<i>rode</i>	
	+	<i>met</i>	<i>woordjes</i>	<i>in het</i>
<i>Met</i>			<i>onder de</i>	<i>blokje</i>
		<i>ingedrukt</i>	<i>toetsen</i>	<i>staat E</i>

---

Als er een E in de cursor staat kun je de rode woordjes onder de toetsen dus krijgen door  ingedrukt te

houden terwijl je op een andere toets drukt. Druk je op de C-toets dan krijg je PAPER, druk je op de X-toets dan krijg je INK. Voor BORDER hoef je geen E in het blokje te hebben, die krijg je meteen als je op de B-toets drukt.

Zo kun je de letters, het papier en de 'tafel' waar het papier op ligt een verschillende kleur geven.

Schrijf zo eens je eigen naam.

Een voorbeeld:

<i>typ</i>	INK 6	(letters worden geel)	<b>ENTER</b>
<i>typ</i>	PAPER 1	(papier wordt blauw)	<b>ENTER</b>
<i>typ</i>	BORDER 2	(rand wordt rood)	<b>ENTER</b>

als je nog een keer op **ENTER** drukt zie je blauw papier op rode tafel.

*typ* PRINT "moniek" (of eigen naam tussen de aanha-  
lingstekens) **ENTER**

Wij hebben een programma gemaakt dat je al de kleuren van de ZX Spectrum laat zien. Het staat achterin en heet REGENBOOG.

# GELUID OP DE ZX SPECTRUM

Je kunt met je computer ook geluiden maken. Je doet dat met BEEP, die zit onder de Z-toets. Je kunt hem krijgen

met	<div style="border: 1px solid black; padding: 2px; display: inline-block;">CAPS SHIFT</div> + <div style="border: 1px solid black; padding: 2px; display: inline-block;">SYMBOL SHIFT</div>	krijg je met ingedrukt	rode woordjes onder de toetsen	in het blokje staat E
-----	---	------------------------------	---	-----------------------------

Als er een E in de cursor staat kun je BEEP krijgen door 

SYMBOL  
SHIFT

 ingedrukt te houden terwijl je op de Z-toets drukt.

Achter BEEP moet je twee getallen typen, bijvoorbeeld BEEP 5,2. Druk nu eens op 

ENTER

 Hoor je het? Bij BEEP zegt het eerste getal hoeveel seconden het geluid moet duren. Hier duurt het geluid 5 seconden. Wil je dat het 10 seconden duurt dan type je BEEP 10,2. Met het tweede getal bepaal je het soort geluid. Des te hoger het getal des te hoger het geluid. Je kunt tot ongeveer 60 gaan.

## BEEP van -60 tot 60

Typ je BEEP 5,60 dan krijg je een gepiep dat 5 seconden duurt. Typ je BEEP 5,-30 dan krijg je gebrom dat 5 seconden duurt. Het laagste getal is -60. Je kunt alle getallen tussen -60 en 60 nemen.

De volgende regels moet je precies zo typen.

```
10 FOR x = -60 TO 60    (druk op 

ENTER

 )
20 BEEP .25, x          (druk op 

ENTER

 )
30 NEXT x               (druk op 

ENTER

 )

RUN                     (druk op 

ENTER

 )
```

Hoor je de tonen?

Als je goed luistert hoor je dat je met die geluiden ook allerlei dingen na kunt doen. Het laatste stuk lijkt een

beetje op het stijgen van een ruimteschip. Als je het geluid sneller maakt door in regel 20 van .25 te maken .10, lijkt het helemaal net echt.

Je kunt met de ZX Spectrum ook muziek maken. Met de volgende regels krijg je de toonladder do, re, mi, fa, enz.

```
10 FOR x = 1 TO 8
20 READ a
30 BEEP 1,a
40 NEXT x
50 DATA 0,2,4,5,7,9,11,12
```

RUN (druk op **ENTER**)

Als je naar regel 50 kijkt zie je de getallen voor de toonladder staan.

0 = do	2 = re	4 = mi	5 = fa
7 = sol	9 = la	11 = si	12 = do

Nu kun je zelf muziek maken. Achterin staan een paar voorbeelden van liedjes.



DEEL 2

# **PROGRAMMEREN IN BASIC**



# KLAAR OM TE PROGRAMMEREN?

Je weet nu hoe je het toetsenbord kunt gebruiken. Wat je op het toetsenbord intypt, kun je op het TV-scherm krijgen. Ook weet je hoe je letters en getallen op het TV-scherm kunt veranderen of uitvegen. Dat zijn de belangrijkste dingen die je moet weten om de fantastische wereld van het programmeren binnen te stappen. Als je kunt programmeren, kun je de computer opdrachten gaan geven die hij voor jou moet gaan uitvoeren. Je kunt hem snel laten rekenen, je kunt hem dingen laten onthouden zodat hij je bijvoorbeeld kan overhoren, je kunt spelletjes ontwerpen en nog veel, veel meer.

## **Wat is programmeren?**

Programmeren is het schrijven van een programma, een soort brief, waarin precies staat wat de computer moet gaan doen. Zo'n programma bestaat uit regels, die allemaal moeten beginnen met een nummer.

## **REGEL- NUMMER**

Zo'n nummer noemen we een **REGELNUMMER**. De regelnummers geven aan, in welke volgorde een programma uitgevoerd moet worden. Je zou daarvoor nummer 1, 2, 3 enz. kunnen nemen. Als je de computer dan de opdracht geeft om een programma te gaan uitvoeren begint hij bij regel 1, dan regel 2, dan regel 3, enz.

Nu is programmeren een heel creatief en precies werkje. Vaak wil je tijdens het werken aan je programma tussen twee regels nog een regel zetten.

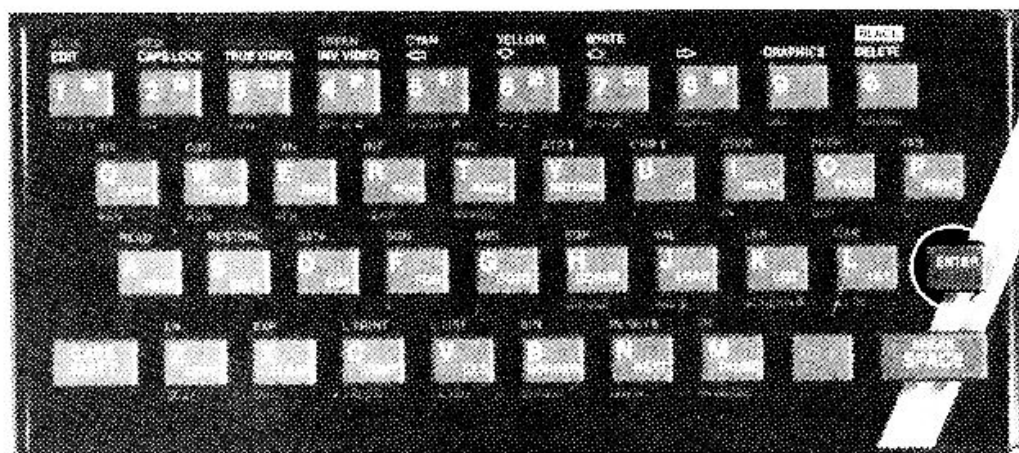
Maar wanneer je als eerste regel 1 hebt gekozen en als tweede regelnummer 2, kun je er niets meer tussen zetten.

Je mag alleen hele nummers gebruiken. Daarom beginnen programma's bijna allemaal met regelnummer 10 en daarna 20, 30, 40, enz. Later kun je dan



gemakkelijk tussen de eerste regel 10 en de tweede regel 20 nog een opdracht zetten op bijvoorbeeld regel 12, 15 of 18. Net zoals je een regel moet beginnen -met een heel nummer- moet je ook iedere regel eindigen.

**ENTER** Je beëindigt een regel altijd, door op de ENTER-toets te drukken. Dat is de toets die rechts ongeveer in het midden zit.



De ENTER-toets is heel belangrijk. Alles wat je op het TV-scherm typt krijgt pas betekenis voor de computer als je op de ENTER-toets drukt. Dan pas gaat de computer controleren of wat op die regel staat ook zo in zijn geheugen mag. Na het indrukken van de ENTER-toets gaat de computer heel snel kijken of er een regelnummer voorstaat, of er begrijpelijke opdrachten tussen staan en nog veel meer. Als alles goed gaat, accepteert de computer die regel en stopt hem in zijn geheugen. Op het scherm is die controle niet te zien. Je ziet alleen maar dat het blokje (de cursor) weer links onderaan staat. Je kunt dan een nieuwe regel intypen. De computer kan op één regel vrij veel tekens onthouden.

Meestal stoppen wij er maar 10 of 20 tekens in. Een aantal van die regels vormen dan samen een programma.

Je moet dus onthouden dat een programma in BASIC er altijd zo uit moet zien:

---

*regelnummer*

*bijv. 10.....      iedere regel eindigen*

*20 .....      met een druk op **ENTER***

*30 .....*

---

## **OP- DRACH- TEN**

Je weet nu hoe een programma er uit ziet. In een programma kun je opdrachten zetten, deze heten: programma-opdrachten. Verder bestaan er opdrachten die de computer kent om een programma te beginnen, te laten bekijken en uit te voeren, deze noemen we directe opdrachten. Hier volgen de opdrachten die wij je zullen leren. Er zijn er nog meer maar met deze opdrachten kun je heel ver komen.

---

<i>directe opdrachten</i>	<i>programma-opdrachten.</i>
NEW	PRINT
CLS	INPUT
LIST	LET
RUN	GO TO
BREAK/SPACE	GO SUB en RETURN
CONT	REM
	IF ... THEN
	FOR ... NEXT
	READ en DATA
	RND

---

De programma-opdrachten komen in de volgende hoofdstukken aan bod. Dan leren we je hoe je ze kunt gebruiken in een programma. De directe opdrachten zullen we nu uitleggen. Dan weet je hoe je ervoor kunt zorgen dat de computer je programma's uitvoert.

## **NEW**

NEW is heel eenvoudig het Engelse woordje voor nieuw. Als je een nieuw programma wilt intypen, dan begin je met:

*typ    NEW      ( **ENTER** indrukken)*

De computer maakt dan eerst zijn geheugen schoon, zodat jij je programma kunt intypen. Het lijkt een beetje op eerst je tafel of bureau opruimen en een nieuw stuk papier en een potlood pakken. Of een cassettebandje aan het begin zetten zodat je een nieuwe l.p. kunt opnemen. Na NEW is het geheugen schoon en kun je van voren af aan beginnen.

Typ nu het volgende kleine programmaatje eens in.

<i>typ</i>	NEW	(druk op <b>ENTER</b> )
<i>typ</i>	10 PRINT "dag";	(druk op <b>ENTER</b> )
<i>typ</i>	20 PRINT "arleen"	(druk op <b>ENTER</b> )
<i>typ</i>	30 GO TO 10	(druk op <b>ENTER</b> )

## CLS

Zo maak nu het TV-scherm maar schoon. (Je weet dat je dan de CLS-toets en **ENTER** in moet drukken). En whoeps, alles wat er stond is weg.

Zou je programma nu helemaal weg zijn zodat je opnieuw moet beginnen? Wel nee, het programma is niet weg, je hebt het alleen maar even verstoppt. Het nieuwe blaadje met je programma erop zit als het ware in de la van je bureau. Je kunt het blaadje er weer uit halen en bekijken. Je hebt daar de directe opdracht LIST voor nodig.

## LIST

Als je LIST en dan ENTER indrukt verschijnt het blaadje weer op het TV-scherm en kun je het programma weer bekijken. LIST staat op de K-toets. Probeer het maar eens:

*typ* LIST (druk op **ENTER**)

Staat er nu:

```
10 PRINT "dag";  
20 PRINT "arleen"  
30 GO TO 10
```

Ja, dan is het goed.

Als je heel lange programma's hebt, die niet op het TV-scherm passen, dan komen alleen de eerste regels op het TV-scherm. Geen nood, want je kunt de volgende regels ook bekijken.

Onderaan het scherm zie je namelijk:

scroll?

Dat betekent zoiets als doorgaan?

Je mag dan iedere toets indrukken alleen de N niet. Het best kun je de J-toets van ja indrukken. De computer laat je dan de volgende regels zien.

Eenvoudig hè?

Je weet wat je moet doen als je een nieuw programma wilt gaan maken (NEW typen).

Je kunt dat programma, nadat je het hebt ingetypt, bekijken of het goed in het geheugen staat (LIST typen). Nu moet je de computer nog de directe opdracht geven om het programma uit te voeren.

## RUN

Dat doen we met de directe opdracht RUN.

RUN staat op de R-toets.

*typ*      RUN      (*druk op* **ENTER**)

Zie je nu



op het TV-scherm staan. Als het scherm vol is vraagt de computer weer scroll?

Ja?, gefeliciteerd. Je hebt nu je eerste programma gemaakt en door de computer laten uitvoeren.

RUN is het Engels voor hardlopen. En hardlopen kan de computer. In vliegende vaart zoekt de computer de eerste regel (10) op, voert die uit, gaat snel naar de volgende regel (20) net zo lang tot de computer aan het eind is. Ook programma's van 100 regels of meer rent de computer moeiteloos langs.

**BREAK  
SPACE**

Maar hoe kun je hem nu laten stoppen? Door op **BREAK SPACE** te drukken. Midden in het programma houdt de computer op met uitvoeren.

**CONT**

Je ziet dat er nu onderaan staat:

D BREAK – CONT repeats, 10:1

Als je wilt dat hij weer verder gaat, druk je op de CONT-toets (staat op de C) en op **ENTER**.

Zo dit waren de directe opdrachten waarmee je programma's kunt beginnen, kunt bekijken en uitvoeren. Onthoud maar:

NEW	Het geheugen wordt dan schoon gemaakt en je kunt een nieuw programma beginnen.
CLS	Maakt scherm schoon, maar je programma blijft bewaard.
LIST	De computer laat zien wat in zijn geheugen staat.
RUN	Is het startsein voor de computer om het programma te gaan uitvoeren.
BREAK SPACE	Door op <b>BREAK SPACE</b> te drukken stopt de computer midden in een programma.
CONT	Door op deze toets te drukken gaat hij weer verder.

## FOUTEN VERBETEREN IN EEN PROGRAMMA

Type het volgende programma, maar dan met je eigen naam, eens in.

```
10 PRINT "daf"  
20 PRINT "arleen"  
30 PRINT "tod ziens"
```

Als je het hele programma hebt ingetypt ontdek je fouten. Daf moet natuurlijk dag zijn en tod moet tot zijn. Je kunt dit eenvoudig verbeteren. Hoe gaat dat? Je kunt de regel opnieuw intypen. Je typt dan

```
10 PRINT "dag" ( ENTER )
```

De computer veegt nu de oude regel weg en zet er de nieuwe voor in de plaats en zo is de fout verbeterd. Probeer het maar eens. Dat is makkelijk hè.

Er is nog een manier. Zie je het pijltje > op het scherm staan? Waarschijnlijk staat dat bij regel 10. Het staat bij de regel die er het laatst bij is gezet.

Je kunt dit pijltje ook sturen.

Als je goed kijkt, zie je boven de toetsen 5, 6, 7 en 8 pijltjes staan.

Als je nu op **SHIFT** en < (toets 6) drukt gaat het pijltje (>) een regel naar beneden.

Als je nu op **SHIFT** en > (toets 7) drukt gaat het pijltje weer een regel naar boven.

Zet het pijltje eens bij regel 30.

Als je nu op **SHIFT** en EDIT (toets 1) drukt komt regel 30 onder in het scherm te staan. Dat betekent dat je er wat aan kunt veranderen.

Laten we 'tod' veranderen in 'tot'.

Met **SHIFT** en → (toets 8) kun je een plaatsje naar rechts gaan. Zet nu de cursor (het blokje) even voorbij de d van tod.

Met **CAPS SHIFT** en **DELETE** (toets 0) kun je de d uitvegen en er een t voor in de plaats zetten. Als je nu weer op **ENTER** drukt zie dat regel 30 veranderd is. Er staat nu netjes:

```
30 PRINT "tot ziens"
```

Mooi he! Vooral bij grote programma's is dit een handige manier om de regels die al boven aan het scherm staan te veranderen.

We zetten nog even op een rijtje hoe je de cursor kunt verplaatsen.

met **CAPS SHIFT** en  $\downarrow$  (toets 6) gaat pijltje een regel naar beneden

met **CAPS SHIFT** en  $\uparrow$  (toets 7) gaat pijltje een regel naar boven

met **CAPS SHIFT** en **EDIT**(toets 1) komt de regel waar het pijltje (>) staat onderaan het scherm

met **CAPS SHIFT** en  $\rightarrow$  (toets 8) gaat cursor naar rechts

met **CAPS SHIFT** en  $\leftarrow$  (toets 5) gaat cursor naar links

Een letter uitvegen gaat met **CAPS SHIFT** en **DELETE** (toets 0).

Oefen er maar mee. Het is de moeite waard om dit te leren.

# **Leren Programmeren**





# INLEIDING

Dit tweede deel van het boek gaat over de taal BASIC. In BASIC worden programma's geschreven die de computer kan begrijpen. BASIC is een taal zoals Nederlands, Engels en Frans talen zijn. Een computer is alleen geen mens die talen als Nederlands en Engels zomaar kan leren en begrijpen. De computer moet je in zijn eigen taal vertellen wat hij moet doen, anders snapt hij het niet.

Dat betekent, dat jij BASIC moet leren, want jij bent een mens en mensen kunnen wel nieuwe talen leren. Jij kunt ook – misschien doe je dat later nog – andere computertalen leren als PASCAL, ALGOL, LOGO, etc. BASIC is de meest gebruikte taal die bijna iedere computer snapt. De kunst is dus, om als je de computer iets voor je wilt laten doen, bijvoorbeeld reken eens een som voor me uit, dat zo te doen dat hij je begrijpt en doet wat je zegt. Bedenk dan eerst in gewoon Nederlands wat je van hem wilt en vertaal dat in BASIC. In dit deel leer je de computertaal BASIC. Je moet dan de hoofdstukken op volgorde lezen. Als je het woordje *typ* ziet, moet je die regels op je eigen computer typen en doen wat er staat.

BASIC heeft maar een paar werkwoorden. Opdrachten noemen we die. Meestal staat er in elke zin (regel) één zo'n opdracht. Bijvoorbeeld het woord PRINT is de vertaling van het Nederlands: schrijf op het TV-scherm. En GO TO is de vertaling voor het Nederlands: ga naar de regel ...

In ieder van de volgende hoofdstukken leer je een belangrijke opdracht. Als je deze hebt geleerd kun je de computer al veel voor je laten doen, zoals je helpen bij het huiswerk, telefoonnummers van je vrienden onthouden, uitrekenen hoe je met je geld kunt omgaan, je eigen spelletjes maken, enz.

Elk hoofdstuk ziet er hetzelfde uit. Eerst lees je welke opdracht je gaat leren, en wat je de computer met die opdracht kunt laten doen. Bijvoorbeeld:

*Met PRINT kun je tekst op  
het TV-scherm krijgen*

Dan volgt een voorbeeld hoe de opdracht gebruikt wordt en de vertaling daarvan in het Nederlands. Bijvoorbeeld:

*In BASIC:*

`10 PRINT "dag"`

*In Nederlands:*

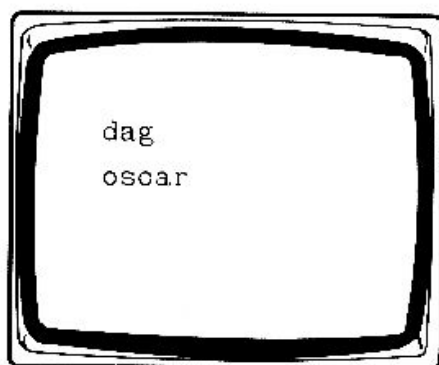
*zet dag op het TV-scherm*

Daarna krijg je programmaatjes met die opdracht die je moet typen. Bijvoorbeeld:

`typ 10 PRINT "dag"`

`typ 20 PRINT "oscar"`

Het zal je waarschijnlijk niet altijd direct lukken, de opdrachten zo te geven dat de computer je snapt. Je moet zijn taal dan nog iets beter leren spreken. Daarom staan achter elk hoofdstuk de vergissingen die je vaak maakt als je deze taal leert. Ook staat erbij hoe je de fouten kunt verbeteren. Soms zie je een TV-scherm staan.



Wat je op dit scherm ziet, kun je op het TV-scherm van jouw computer verwachten.

Behalve de opdrachten tussendoor, staan er aan het eind van elk hoofdstuk opdrachten die je moet kunnen maken als je de stof goed hebt gelezen. De antwoorden vind je achter in het boek. *Veel succes.*

# DE PRINT- OPDRACHT

## PRINT

---

*Met PRINT kun je tekst op het  
TV-scherm schrijven*

---

*In BASIC:*

*In Nederlands:*

10 PRINT "dag"

*zet dag op het TV-scherm*

20 PRINT 10

*zet 10 op het TV-scherm*

---

We gaan met het PRINT-commando de computer leren om netjes dag te zeggen. het programma ziet er zo uit: (vergeet niet NEW te geven)

*typ* 10 PRINT "dag";           *- let op de punt/komma*  
*typ* 20 PRINT "oscar"       *- of jouw naam*  
*typ* 30 GO TO 10

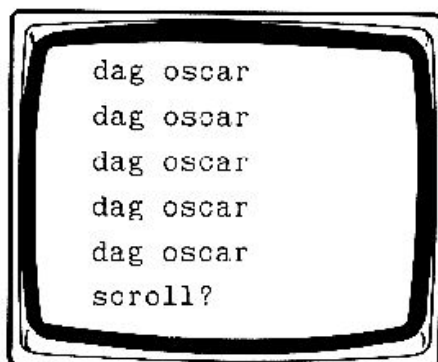
Zo, wat heb je nu gedaan?

Je hebt 3 regels in het geheugen van de computer gezet. Laten we eens kijken wat ze betekenen.

op regel 10 staat    'zet 'dag' op het TV-scherm'.  
op regel 20 staat    'zet 'oscar' op het TV-scherm'  
op regel 30 staat    'ga weer naar regel 10 en begin  
                         opnieuw'

Als de drie regels er precies staan zoals het moet, dan volgt nu een spannend moment. Je geeft de computer de opdracht om jouw programma uit te voeren.

*typ* RUN           *(niet vergeten op **ENTER** te drukken.)*



Als je dit op het scherm ziet, dan doet de computer precies wat jij wilde.

Soms komt er een foutmelding. Bijvoorbeeld:

```
2 variable not found, 30:1
```

Dit betekent dat je foutmelding 2 hebt gekregen. Variable not found is Engels voor Variabele (meestal een letter) niet gevonden. Het volgende getal (30) geeft aan dat de fout op regel 30 zit.

Typ regel 30 in dit geval opnieuw in.

Zo nu heb je de computer geleerd om jou gedag te zeggen.

Netjes he?

Laten we nog een voorbeeld maken.

### Programma NAAM

```
typ NEW  
typ 10 PRINT "*****"  
typ 20 PRINT "dit programma"  
typ 30 PRINT  
typ 40 PRINT "    is van"  
typ 50 PRINT  
typ 60 PRINT "oscar baarda"  
typ 70 PRINT "*****"  
  
typ RUN
```



Op regel 30 en 50 staat alleen PRINT. Als je alleen PRINT typt en niets daarachter, dan krijg je er een regel tussen.

Daarmee kun je wat op het scherm komt vaak mooier maken.

Je zult het al wel gezien hebben. Telkens als je tekst op het scherm wilt laten schrijven, moet je die tekst tussen aanhalingstekens "en" zetten. Dus:

```
PRINT "*****"
```

Zie je de aanhalingstekens?

*Alleen PRINT geeft een regel waar niets staat*

Kijk eens naar de volgende voorbeelden

```
typ 10 PRINT "dag"      ← niets er achter
```

```
typ 20 PRINT "oscar"
```

```
typ 30 PRINT
```

```
typ 40 PRINT "dag ";    ← punt/komma er achter
```

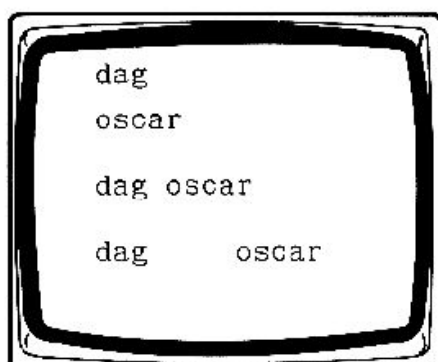
```
typ 50 print "oscar"
```

```
typ 60 PRINT
```

```
typ 70 PRINT "dag",     ← komma er achter
```

```
typ 80 PRINT "oscar"
```

```
typ RUN
```



Je kunt dezelfde teksten op verschillende manieren op het TV-scherm laten komen.

- 
- |                       |                                 |
|-----------------------|---------------------------------|
| - onder elkaar        | <i>doet hij uit zichzelf</i>    |
| - achter elkaar       | <i>dan ; (punt/komma typen)</i> |
| - met ruimte ertussen | <i>dan , (komma typen)</i>      |
- 

*Met PRINT kun je tekst op het TV-scherm schrijven*

## herstel

---

- 1** Ben je de " (aanhalingstekens) vergeten?

Dit komt in het begin wel eens voor. Na **ENTER** komt er dan een vraagteken op de plaats waar de " moeten zijn. Zet ze er alsnog tussen.

- 2** Ben je de spaties vergeten?

Meestal is dat niet zo erg. Het programma draait toch wel. Het programma ziet er echter niet zo mooi uit. Het is verstandiger om de spaties toch aan te brengen. Zoals je weet, krijg je een spatie door de SPACE-toets in te drukken.

- 3** Ben je de ; (punt/komma) vergeten?

Dan schrijft de computer de volgende regel op een nieuwe lijn en dat wou je niet. De punt/komma zorgt er voor dat de teksten achter elkaar worden gezet. Regel opnieuw typen.

---

Geef zonodig een LIST om te kijken of het programma nu goed in het geheugen zit. Geef dan de computer de opdracht om jouw programma uit te voeren.

*typ* RUN (niet vergeten een **ENTER** te geven)

---

## opgaven

---

- 1** Zet op regel 20 de dag, bijvoorbeeld zondag, neer en kijk of je het met RUN op het TV-scherm krijgt.

*typ* 10 PRINT "vandaag is het"  
*typ* 20 .....

- 2** Verander het volgende programma zodanig dat de teksten achter elkaar komen op het TV-scherm.

*typ* 10 PRINT "de zon"  
*typ* 20 PRINT "schijnt"



*Nu staat er*



*Het moet  
worden*



**3** *Nu moet  
het worden*



**4** *Nu moet  
het worden*



*– een regel  
ertussen*

# DE INPUT- OPDRACHT

## INPUT

---

*Met INPUT vraag je de gebruiker  
om een getal of een naam*

---

*In BASIC:*

10 INPUT x  
20 INPUT n\$

*In Nederlands:*

*geef een getal  
geef een naam*

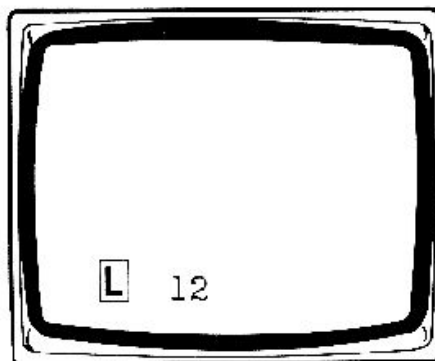
---

Met de opdracht INPUT vraag je de gebruiker vanuit het programma om een getal of een naam. Een getal kan 1, 5 of veel groter zijn. Een naam kan a, h, piet of wat dan ook zijn. Elke combinatie van een aantal tekens is een naam. Je kunt zo complete zinnen typen.

Twee voorbeelden:

10 INPUT x  
20 PRINT x

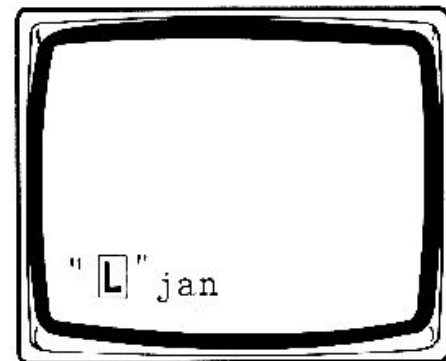
*typ* RUN



*typ een getal*

10 INPUT n\$  
20 PRINT n\$

*typ* RUN



*typ een naam*

Opletten: Als je na INPUT geen \$ geeft, bijvoorbeeld X, L of K, dan moet je een getal intypen en niets anders. Als je namen wilt gaan gebruiken in je programma, moet je na de letter achter INPUT een \$ zetten. De \$ vind je op de 4-toets.

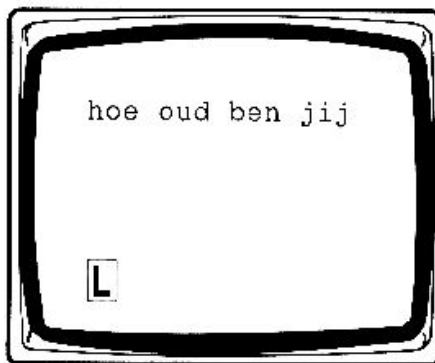
### Een \$ noemen we een stringteken

Een stringteken geeft aan, dat je met andere tekens dan alleen getallen wilt werken. Als je er getallen instopt (bijvoorbeeld een telefoonnummer als 888945), kun je niet met deze getallen rekenen. De computer denkt dat het getal een naam is.

Je kunt heel goed zien of de computer een naam of een getal vraagt. Bij een naam zie je "L". Wanneer hij om een cijfer vraagt zie je alleen een L. Zou je nu toch letters als antwoord geven, dan antwoordt de computer dat je een fout hebt gemaakt.

#### Programma **LEEFTIJD**

```
typ 10 PRINT "hoe oud ben jij"  
typ 20 INPUT l  
typ 30 PRINT "welk jaar is het"  
typ 40 PRINT "bijvoorbeeld 1985"  
typ 50 INPUT j  
typ 60 PRINT "dan ben jij in het"  
typ 70 PRINT "jaar 2000"; l + (2000 - j)  
  
typ RUN
```



– typ een getal,  
bijvoorbeeld 12

Je ziet dat er hier gevraagd wordt om getallen via INPUT l en INPUT j. Een naam is niet mogelijk, want er staat achter l en j geen \$ (stringteken).

## Programma NAAM

```
typ 10 PRINT "wat is je voornaam"
typ 20 INPUT v$
typ 30 PRINT "dank je wel "; v$
typ 40 PRINT "wat is je achternaam"
typ 50 INPUT a$
typ 60 PRINT "dus jij bent "; v$; " "; a$
typ 70 PRINT "tot ziens"

typ RUN
```

Als je zelf programma's gaat maken, is het wel eens nodig dat je de computer laat vragen of je door wilt gaan. Dat kan met het volgende gedeelte dat past in het programma 'NAAM'.

```
typ 70 PRINT "wil je stoppen. ja of nee"
typ 80 INPUT j$
typ 90 IF j$ = "nee" THEN GO TO 10

typ RUN
```

Als je een naam vraagt mag je maar één letter en \$ gebruiken. Dus a\$, b\$ of x\$ en niet aa\$, ba\$ of xx\$. Wanneer je getallen vraagt mag het wel een langere naam hebben zoals getal of aantal.

<p><i>Met INPUT vraag je de gebruiker om een getal of een naam</i></p>
--

## herstel

---

- 1** Een punt in plaats van een komma.

**L** 12,5 (fout)

De computer geeft, als je probeert 12,5 te typen, een vraagteken. Dat komt omdat hij denkt dat het om twee getallen gaat: 12 en 5. Hij heeft er maar om één gevraagd, en raakt in de war. Als je toch 12,5 wilt typen moet je een punt zetten in plaats van een komma, dus 12.5.

**L** 12.5 (goed)

- 2** Aanhalingstekens vergeten als je \$ bij INPUT hebt gebruikt. Als je in je programma bijvoorbeeld XS gebruikt, zoals in:

20 IF x\$ = jan THEN GO TO 100 (fout)

dan moet je steeds aan de aanhalingstekens denken. Ook bij getallen.

Dus

20 IF x\$ = 6 THEN GO TO 100 (fout)

20 IF x\$ = "6" THEN GO TO 100 (goed)

- 3** Je hebt teveel letters voor het stringteken \$ gezet. Je mag er maar één voor zetten.

20 INPUT naam\$ (fout)

30 INPUT n\$ (goed)

---

Geef zonodig een LIST om te kijken of het programma nu goed in het geheugen zit. Geef dan de computer de opdracht om jouw programma uit te voeren.

typ RUN (Niet vergeten een **ENTER** te geven)

## opgaven

---

**1** Is dit programma goed?

```
10 PRINT "wat is je naam?"  
20 INPUT 1  
30 PRINT "je naam is"; 1
```

**2** Is dit programma goed?

```
10 PRINT "wat is je leeftijd?"  
20 INPUT 1$  
30 LET n = 1$ * 12  
40 PRINT "je bent "; n; " maanden jong"
```

**3** Is dit programma goed?

```
10 PRINT "wat is je naam?"  
20 INPUT naam$  
30 PRINT naam$
```

**4** Verander programma 1, 2 en 3 zodanig dat ze goed zijn.

---

# DE LET- OPDRACHT

## LET

---

*Met LET kun je een getal of naam  
in een hokje zetten*

---

*In BASIC:*

10 LET goed = 0

20 LET n\$ = "jitske"

*In Nederlands:*

*zet een 0 in vakje goed*

*zet jitske in vakje n\$*

---

LET betekent eigenlijk LAAT. Als je hierboven kijkt zie je dat je in regel 10 vraagt, LAAT vakje goed een 0 krijgen.

Zoals je in regel 20 ziet kun je zelfs namen of hele zinnen in een vakje zetten. Je moet dan niet vergeten het stringteken \$ te gebruiken.

Je kunt LET ook gebruiken om nieuwe dingen in een programma te berekenen. Laten we het eens gebruiken om de rente van je spaargeld uit te rekenen.

### Programma **RENTE**

```
10 PRINT "hoeveel geld heb je?"
```

```
20 INPUT geld
```

```
30 PRINT "hoeveel procent rente krijg je?"
```

```
40 INPUT percentage
```

```
50 LET rente = geld * percentage / 100
```

```
60 PRINT "na een jaar krijg je "; rente;  
" gulden"
```

Je ziet dat in regel 50 LET gebruikt wordt om de rente uit te rekenen.

LET kun je ook goed gebruiken in spelletjes om de stand bij te houden. Iedere keer als iemand het goed doet laat je de computer naar een regel gaan waar bijvoorbeeld staat:

```
LET goed = goed + 1
```

Achterin zie je programma's staan die dit gebruiken. Je kunt LET ook gebruiken om woorden of zinnnetjes in een hokje te zetten. Vooral bij grote programma's is dat handig.

### Programma **GROET**

```
10 LET a$ = "typ je naam"
20 LET d$ = "goede morgen "
30 PRINT a$
40 INPUT n$
50 PRINT d$ ; n$
```

*Met LET kun je een getal of naam  
in een hokje zetten*

## herstel

---

Met LET kun je heel weinig verkeerd doen.

- 1** Het komt wel eens voor dat je bij LET dingen laat optellen of vermenigvuldigen die nog niet in het programma staan.

```
10 INPUT a
30 LET c = a + b
40 PRINT c
```

*(fout)*

De computer zal dan zeggen 'variable not found'. Dit betekent dat hij de b die je bij a wilt optellen niet kent. Je bent vergeten die te vragen.

```
10 INPUT a
20 INPUT b
30 LET c = a + b
40 PRINT c
```

*(goed)*



## 2 Het stringteken vergeten?

```
10 LET a$ = "typ je naam"  
20 PRINT a$  
30 INPUT n  
40 PRINT n$
```

*(fout)*

In regel 10 vraag je om een naam. Maar in regel 30 vraag je om een getal. Zet er maar een stringteken achter.

```
30 INPUT n$
```

*(goed)*

---

## opgaven

---

- 1 Schrijf een programma waarin je twee getallen a en b vraagt en tel die met behulp van de LET-opdracht bij elkaar op. Laat de computer het antwoord op het TV-scherm schrijven.

- 2 Wat is er fout in het volgende programma.

```
10 LET a$ = "typ je naam"  
20 LET d$ = "goede morgen "  
30 PRINT a$  
40 INPUT n  
50 PRINT d$ ; n$
```

---

# DE GO TO- OPDRACHT

## GO TO

---

*Met GO TO kun je een sprong  
maken in het programma*

---

*In BASIC:*

10 GO TO 50  
50 PRINT "einde"

*In Nederlands:*

*ga naar regel 50  
zet einde op het scherm*

---

GO TO bestaat uit de Engelse woordjes GO en TO en betekent 'ga naar'. Na GO TO volgt meestal een regelnummer, bijvoorbeeld GO TO 80 of GO TO 10. Met deze opdracht kun je een sprong maken in het programma. Er zijn veel mogelijkheden:

spring naar een bepaalde regel  
spring naar laatste regel  
spring naar eerste regel.

Naar boven springen of naar beneden springen is allemaal mogelijk.

Voorbeeld van met GO TO springen.

```
typ 10 LET r = 3
typ 20 PRINT "raad een getal?"
typ 30 INPUT x
typ 40 IF x = r THEN GO TO 70
typ 50 PRINT "probeer het nog eens"
typ 60 GO TO 20
typ 70 PRINT "ja, het is "; x
```

Als in regel 40 x gelijk is aan r wordt er naar regel 70 gesprongen. En in regel 60 springt de computer terug naar regel 20. Je ziet dat je zowel naar boven als naar beneden kunt springen met GO TO.

Typ geen NEW. Als je het volgende programma achter het vorige zet heb je een spelletje.

Voorbeeld van met GO TO naar het einde (hier regel 140) springen.

```
typ 80 PRINT "wil je stoppen?"  
typ 90 PRINT "typ ja of nee"  
typ 100 INPUT j$  
typ 110 IF j$ = "ja" THEN GO TO 140  
typ 120 LET r = r + 2  
typ 130 GO TO 20  
typ 140 PRINT "einde"  
  
typ RUN
```

Dus je ziet dat je met GO TO naar het begin of het einde kunt springen. Maar ook naar een regel die ergens in het midden staat is mogelijk. Met GO TO kun je overal naar toe springen.

*Met GO TO kun je een sprong  
maken in het programma*

## herstel

---

### 1 Naar de volgende regel springen

```
60 GO TO 70 (fout)
70 PRINT .....
```

Dit is niet zozeer fout maar overbodig. Na regel 60 voert de computer zelf al de volgende regel (70) uit. Deze sprong hoeft je niet te maken.

### 2 Naar een letter springen

```
60 GO TO a
```

Dit kan alleen als je vooraf een getal in vakje a hebt gezet. Meestal komt er achter GO TO een regelnummer (bijvoorbeeld 100) en niet een letter (bijvoorbeeld a).

```
50 LET a = 100
60 GO TO a
```

```
100 PRINT "einde" (goed)
```

### 3 Naar zichzelf springen.

```
60 GO TO 60 (fout)
```

Deze 'ga naar' springt naar zichzelf. Het gevolg is dat de computer eindeloos doorgaat. Van 60 naar 60 naar 60 naar 60 enz. Je zult hem dan met CAPS/SHIFT en BREAK/SPACE stil moeten zetten.

---

Geef zonodig een LIST om te kijken of het programma nu goed in het geheugen zit. Geef dan de computer de opdracht om jouw programma uit te voeren.

typ RUN (niet vergeten een **ENTER** te geven).

## opgaven

---

- 1** Welke regelnummers moet je hier intypen?

```
typ 10 LET y = 6
typ 20 PRINT "geef een getal tussen 1 en 10"
typ 30 INPUT x
typ 40 IF x = y THEN GO TO ...
typ 50 PRINT "nee, dat is mijn getal niet"
typ 60 GO TO ...
typ 70 PRINT "ja, dat is mijn nummer " ; x
```

- 2** Geef geen NEW na opgave 1, we gaan door met dat programma.  
Welke regelnummers kun je intypen om dit programma af te maken?

```
typ 80 PRINT "wil je stoppen?"
typ 90 PRINT "typ ja of nee"
typ 100 INPUT j$
typ 110 IF j$ = "ja" THEN GO TO ...
typ 120 PRINT "dus je wilt niet stoppen"
typ 130 GO TO ...
typ 140 PRINT "einde"
```

- 3** Hoe kun je het volgende programma mooier maken door van GO TO gebruik te maken?

```
typ 10 LET n = 1
typ 20 PRINT n
typ 30 LET n = n + 1
typ 40 PRINT n
typ 50 LET n = n + 1
typ 60 PRINT n
typ 70 LET n = n + 1
typ 80 PRINT n
```

---

# DE GO SUB- EN RETURN-OPDRACHTEN

## GO SUB EN RETURN

*Met GO SUB maak je een sprong en met  
RETURN ga je weer terug*

*In BASIC:*

```
100 GO SUB 100  
20  
30  
100 PRINT "hulpje"  
120 RETURN
```

*In Nederlands:*

*ga naar regel 100  
begin hulpprogramma  
ga terug en begin bij  
regel 20*

GO SUB betekent 'doe eerst het hulpprogramma vanaf regel ..'.

Met GO SUB maak je een sprong naar een hulpprogramma. Ieder hulpprogramma moet eindigen met een RETURN. Dat betekent 'ga terug'. GO SUB en RETURN horen bij elkaar. Als je GO SUB gebruikt, moet je ook RETURN gebruiken.

### HOOFDPROGRAMMA

```
100 GO SUB 200  
110  
120  
300 PRINT "einde"
```

### HULPJE

```
200  
210  
220 RETURN
```

Vanuit het hoofdprogramma spring je naar een gedeelte van het programma (hulpprogramma) en daarna weer terug naar de volgende regel in het hoofdprogramma.

## Programma **REKEN**

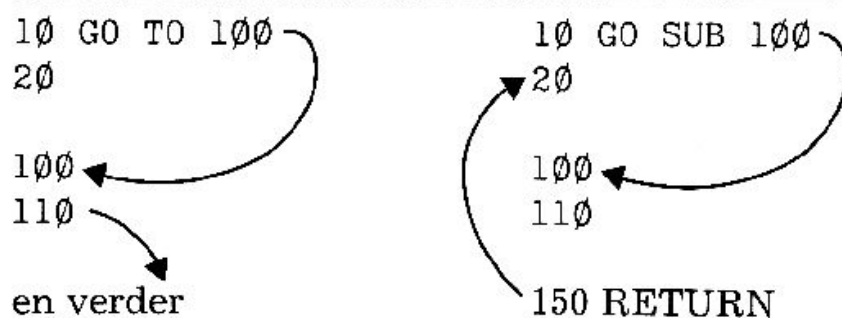
```
typ 10 PRINT "rekenprogramma"
typ 20 PRINT "geef een getal"
typ 30 INPUT g1
typ 40 PRINT "geef nog een getal"
typ 50 INPUT g2
typ 60 GO SUB 100
typ 70 GO SUB 200
typ 80 GO TO 300

typ 100 PRINT g1;" + ";g2;" = ";g1 + g2
typ 110 RETURN
typ 200 PRINT g1;" - ";g2;" = ";g1 - g2
typ 210 RETURN
typ 300 PRINT "einde"

typ RUN
```

Nu zie je gelijk wat het voordeel is van **GO SUB** en **RETURN**. Je kunt er zo vaak als nodig is naar toe springen. Als je vaak hetzelfde moet doen in een programma, dan is het handig om een hulpprogramma te maken waar je iedere keer naar toe springt. **GO SUB** lijkt een beetje op **GO TO** ('ga naar'). Maar met **GO SUB** kom je weer terug en met **GO TO** niet. Met **GO TO** blijf je waar je bent.

### Verschil **GO TO** en **GO SUB**



GO SUB en RETURN zijn handig als je vaak hetzelfde moet doen. Net zoals in sommige liedjes waar je achter ieder couplet een refrein moet zingen.

### HOOFDPROGRAMMA

Alles in de wind,  
alles in de wind,  
't is maar een schipperskind.

GO SUB

O, wat een spijt,  
o, wat een spijt,  
nu is m'n zusje kwijt.

GO SUB

O, wat een nood,  
o, wat een nood,  
Nu is mijn zusje dood.

GO SUB

Op die brug,  
op die brug,  
vond ik m'n zusje terug.

GO SUB

Einde.

### HULPPROGRAMMA

refrein:

Kom hier, Rosa,  
je bent m'n zusje,  
je bent m'n zusje,  
Kom hier, Rosa,  
je bent m'n zusje, ja, ja.

RETURN



Hier moet je op vier verschillende plaatsen het refrein 'kom hier, Rosa' doen. Dat is vier keer hetzelfde. En als je vaak hetzelfde op verschillende plaatsen moet doen kun je er een hulpprogramma van maken met GO SUB en RETURN.

Je ziet dat er vier keer GO SUB staat en maar één keer RETURN.

### Programma **GETALLEN RADEN**

```
typ 20 LET x=7
typ 30 GO SUB 100
typ 40 LET x=4
typ 50 GO SUB 100
typ 60 LET x=9
typ 70 GO SUB 100
typ 80 PRINT "en nu zijn we klaar"
typ 90 GO TO 200

typ 100 PRINT "welk getal heb ik?"
typ 110 INPUT getal
typ 120 IF getal = x THEN GO TO 150
typ 130 PRINT "probeer het nog eens"
typ 140 GO TO 110
typ 150 PRINT "goed zo, de volgende."
typ 160 RETURN

typ 200 PRINT "einde"

typ RUN
```

*Met GO SUB maak je een sprong en met  
RETURN ga je weer terug*

## herstel

---

### 1 Een RETURN vergeten.

```
10 GO SUB 50
50 PRINT "dit is regel 50"      (fout)
100 PRINT "einde"

10 GO SUB 50
20 GO TO 100

50 PRINT "dit is regel 50"      (goed)
60 RETURN
100 PRINT "einde"
```

### 2 GO TO in plaats van GO SUB gebruikt.

```
10 GO TO 100
100 PRINT "dit is regel 100"    (fout)
120 RETURN
```

RETURN is alleen mogelijk in combinatie met een GO SUB-opdracht. Dus:

```
10 GO SUB 100
100 PRINT "dit is regel 100"    (goed)
120 RETURN
```

### 3 Achter de laatste GO SUB moet altijd een regel staan. Anders springt een RETURN terug naar niets en dat kan de computer niet.

---

Geef zonodig een LIST om te kijken of het programma nu goed in het geheugen zit. Geef dan de computer de opdracht om jouw programma uit te voeren.

typ RUN (niet vergeten een **ENTER** te geven)

## opgaven

---

- 1** De bedoeling is dat alle getallen x geprint worden.  
Waarom doet hij dat niet?

```
10 LET x = 7
20 GO SUB 200
30 LET x = 9
40 GO SUB 200
50 LET x = 12
60 GO SUB 200
70 GO TO 300
200 PRINT x
300 PRINT "einde"
```

- 2** De bedoeling is dat alle getallen x geprint worden.  
Waarom doet hij dat niet?

```
10 LET x = 7
20 GO TO 200
30 LET x = 9
40 GO TO 200
50 LET x = 12
60 GO TO 200
70 GO TO 300
200 PRINT x
210 RETURN
300 PRINT "einde"
```

- 3** Wanneer kun je beter een GO SUB en RETURN  
gebruiken dan een GO TO?
-

# DE REM- OPDRACHT

## REM

---

*Via REM kun je mededelingen  
in het programma zetten (voor jezelf)*

---

*In BASIC:*

*In Nederlands:*

*10 REM programma 1*

*mededeling programma*

---

REM is een afkorting van het Engelse woordje REMark en betekent opmerking. Wat achter REM staat wordt door het programma niet uitgevoerd. Het is meer een handigheidje. Je weet hier bijvoorbeeld dat het om programma 1 gaat.

REM doet in het programma niets. De REM-opdracht wordt overgeslagen. Je kunt er dus alles wat je wilt achter zetten, bijvoorbeeld mededelingen of een deel van het programma mee uitleggen. Vooral als je een programma aan een ander wilt geven kun je achter REM mededelingen zetten. Kijk maar eens naar het volgende programma:

```
typ 10 REM dit programma is van aad en ben
typ 20 REM vraag om een voornaam
typ 30 PRINT "wat is je voornaam?"
typ 40 INPUT v$
typ 50 REM vraag om een achternaam
typ 60 PRINT "wat is je achternaam?"
typ 70 INPUT a$
typ 80 REM zet voor- en achternaam op het
      tv-scherm
typ 90 PRINT "dag "; v$; " "; a$
typ 100 PRINT "leuk met je kennis gemaakt te
      hebben"
typ 110 PRINT "einde"
typ RUN
```

Je ziet dat de opmerking achter de REM-opdracht door de computer niet uitgevoerd wordt. Waarom is het dan nodig dat er REM-opdrachten in staan?

Met REM-opdrachten kun je het programma veel duidelijker maken. Zowel voor jezelf als voor anderen. Als je het programma later nog eens wilt bekijken dan weet je door de REM-opdrachten waar alles voor dient. Ook als je het programma aan een ander wilt geven kun je via REM-opdrachten, mededelingen in het programma zetten. Vooral voor lange programma's zijn REM-opdrachten handig. De programma's zijn dan veel duidelijker.

Kijk maar eens naar het volgende programma. Het is precies hetzelfde als het programma dat we net gemaakt hebben.

Alleen de REM-opdrachten zijn er uitgelaten.

```
typ  30 PRINT "wat is je voornaam?"
typ  40 INPUT v$
typ  60 PRINT "wat is je achternaam?"
typ  70 INPUT a$
typ  90 PRINT "dag "; v$;" ";a$
typ 100 PRINT "leuk met je kennis gemaakt te
        hebben"
typ 110 PRINT "einde"
```

*Via REM kun je mededelingen  
in het programma zetten (voor jezelf)*

## **herstel**

---

- 1** De REM-opdracht gaat niet zo snel fout. Alles achter de REM-opdracht wordt als een mededeling gezien en precies zo in de LISTing van een programma opgenomen.
- 

## **opgaven**

---

- 1** Maak programma LEEFTIJD op blz. 54 mooier door met REM-opdrachten er mededelingen tussen te zetten. Kies mededelingen die anderen kunnen gebruiken.
  - 2** Doe dit ook voor programma NAAM op blz. 55.
-

**IF ...  
THEN**

Met IF ... THEN moet de computer een keuze maken

10 IF  $x = 6$  THEN GO TO 100

```
100 PRINT "einde"
```

als het 6 is ga dan  
naar regel 100

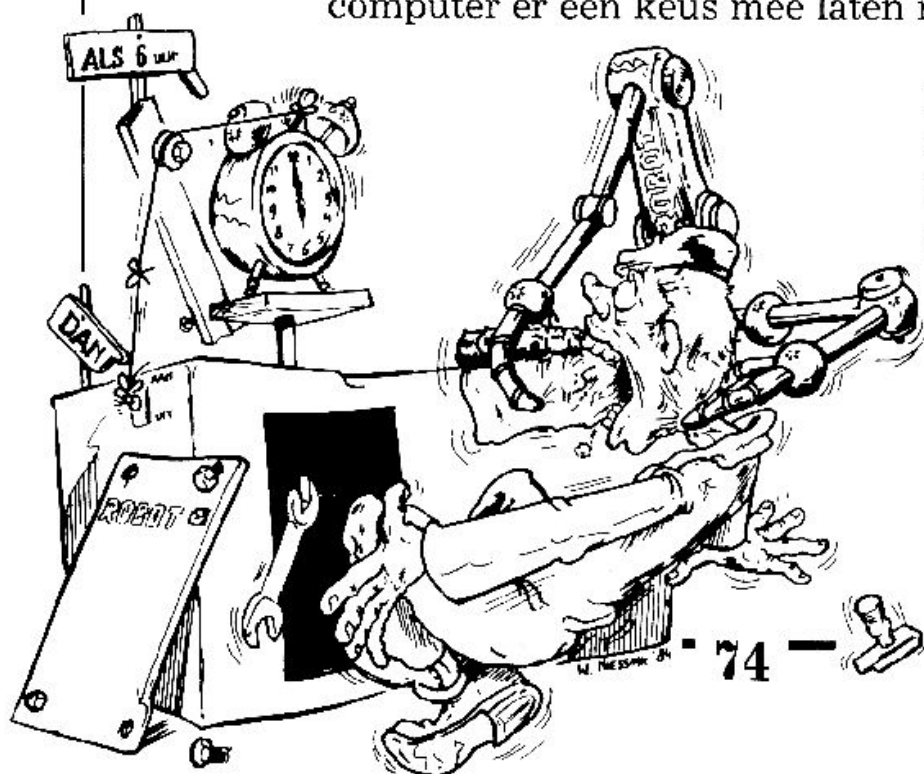
einde

IF en THEN zijn twee Engelse woordjes. IF betekent ALS en THEN betekent DAN. Zou de man die BASIC gemaakt heeft een Nederlander geweest zijn, dan had hij er vast ALS ... DAN van gemaakt. Maar die man sprak Engels en dus is het IF ... THEN geworden.

De IF ... THEN-opdracht is heel handig. Je kunt de computer er een keus mee laten maken.

Laten we een voorbeeld maken:

Men heeft tegen je  
gezegd: ALS het zes uur  
is DAN moet je gaan  
eten. Anders kun  
je doorgaan met  
waar je mee  
 bezig bent.



### Programma TIJD

```
typ 10 PRINT "hoe laat is het?"  
typ 20 PRINT "in uren"  
typ 30 INPUT t  
typ 40 IF t = 6 THEN PRINT "snel naar huis"  
typ 80 PRINT "einde"  
typ RUN
```



–Typ de tijd. Bijv. 5

Alleen maar als  $t = 6$  uur dan wordt de opdracht uitgevoerd.

Als  $t$  niet 6 (uur) is dan wordt de opdracht niet uitgevoerd.

Die regel wordt dan overgeslagen en gaat de computer de volgende regel uitvoeren.

Tussen IF ... THEN kunnen de volgende vergelijkingen gemaakt worden:

```
typ 40 IF t = 6 THEN PRINT "het is 6 uur"  
typ RUN
```

```
typ 50 IF t > 6 THEN PRINT "het is al 6 uur geweest"  
typ RUN
```

```
typ 60 IF t < THEN PRINT "het is voor 6 uur"  
typ RUN
```

```
typ 70 IF t <> 6 THEN PRINT "het is geen 6 uur"  
typ RUN
```



---

<i>Dus onthouden</i>	=	(bij de L-toets)	gelijk aan	6 = 6
	>	(bij de T-toets)	groter dan	8 > 6
	<	(bij de R-toets)	kleiner dan	5 < 6
	<>	(bij de W-toets)	ongelijk aan	3 <> 6

---

Er zijn nog meer vergelijkingen, maar die hoeft je nog niet te onthouden.

### Programma **GELUKSGETAL**

```

typ 10 PRINT "raad mijn geluksgetal"
typ 20 PRINT "typ een getal tussen 1 en 10"
typ 30 INPUT x
typ 40 IF x = 7 THEN GO TO 70
typ 50 PRINT "probeer het nog eens"
typ 60 GO TO 20
typ 70 PRINT "ja, 7 is mijn geluksgetal"
typ 80 PRINT "einde"

typ RUN

```

Als je een foutmelding krijgt dan weet je dat de computer je niet begrijpt. Eerst de vergissing herstellen en dan nog een keer RUN typen. Lukt het niet, kijk dan even in Herstel.

### Programma **MENEER of MEVROUW**

```

typ 10 PRINT "ben je een man of een vrouw"
typ 20 PRINT "typ een m of een v"
typ 30 INPUT m$
typ 40 IF m$ = "m" THEN LET m$ = "meneer"
typ 50 IF m$ = "v" THEN LET m$ = "mevrouw"
typ 60 PRINT "wat is je voornaam?"
typ 70 INPUT v$
typ 80 PRINT "dag "; m$; " "; v$
typ 90 PRINT "einde"

typ RUN

```

Je ziet dat er bij INPUT weer een stringteken (\$) achter de letters staat. Je hebt het nodig omdat je een naam vraagt.

*Met IF ... THEN moet de computer een keuze maken*

## herstel

### 1 Aanhalingstekens vergeten?

40 IF m\$ =  $\swarrow \nearrow$  m THEN ... (fout)

Aanhalingstekens vergeten.

Het stringtekentje \$ vraagt om een naam of letters en die moeten tussen aanhalingstekens. Herstel die regel.

typ 40 IF m\$ = "m" THEN PRINT m\$ - "meneer" (goed)

Geef zonodig een LIST om te kijken of het programma nu goed in het geheugen zit. Geef dan de computer de opdracht om jouw programma uit te voeren.

typ RUN (niet vergeten een **ENTER** te geven)

## opgaven

- 1 Verander het programma TIJD op blz. 75 zodanig dat je om 5 uur naar huis moet.
- 2 Wat betekent het = teken?
- 3 Wat betekent het > teken?
- 4 Wat betekent het < teken?
- 5 Wat betekent het <> teken?
- 6 Verander het programma GELUKSGETAL op blz. 76. Laat nu 3 het geluksgetal van de computer zijn.

# DE FOR ... NEXT-OPDRACHT

## FOR ... NEXT

*Met de FOR ... NEXT-opdracht kun je de opdracht die er tussen staat herhalen*

*In BASIC:*

10 FOR x = 1 to 4

20 PRINT "ben"

30 NEXT x

*In Nederlands:*

*doe van 1 tot 4 keer het volgende*

*zet ben op TV-scherm  
ga weer terug*

Deze opdracht  
wordt 4 keer herhaald.

FOR vind je op de F-toets en TO ook, maar dan in rood. TO krijg je door gelijktijdig de **SYMBOL SHIFT** en de F-toets in te drukken.

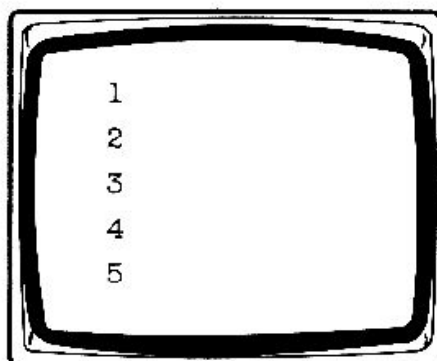
Voorbeeld:

typ 10 FOR j = 1 TO 5

typ 20 PRINT j

typ 30 NEXT j

typ RUN



Je ziet dat de FOR ... NEXT-lus een soort herhaling is van de opdracht die er tussen staat. FOR (voor) geeft het begin aan en NEXT (volgende) stuurt de computer terug naar FOR net zolang tot dat hij klaar is. 'FOR j = 1 TO 5' betekent, dat er 5 keer een opdracht moet worden uitgevoerd.

*De FOR ... NEXT-opdracht wordt een lus genoemd*

Je kunt een lus ook goed als een pauze in je programma gebruiken.

```
typ 10 PRINT "begin pauze"
typ 20 FOR p = 1 TO 1000
typ 30 NEXT p
typ 40 PRINT "einde pauze"

typ RUN
```

Wat er tussen regel 20 en regel 30 staat wordt 1000 keer uitgevoerd. Maar tussen regel 20 en 30 staat geen opdracht.

De computer telt eigenlijk alleen 1000 keer en dat kost tijd. Het is net als met verstoppertje spelen. De computer moet tot 1000 tellen voordat hij mag gaan zoeken. Als de computer aan het tellen is heb jij even pauze. Je kunt de pauze korter of langer maken door de computer korter of langer te laten tellen.

Let er op dat er maar één letter achter FOR mag komen.

Dus FOR pauze = 1 TO 100 is fout.

---

FOR p = 1 TO 100 : NEXT p	is 1 sec.
FOR p = 1 TO 1000 : NEXT p	is 5 sec.
FOR p = 1 TO 10000 : NEXT p	is 45 sec.

---

De getallen die hier genoemd worden, kunnen per computer verschillen. Vaak is het een kwestie van proberen hoe lang je de computer moet laten tellen.

Voorbeeld:

```
typ 100 FOR p = 1 TO 10
typ 110 NEXT p
typ 120 PRINT "dag";
typ 130 FOR p = 1 TO 100
typ 140 NEXT p
typ 150 PRINT "hoe";
typ 160 FOR p = 1 TO 1000
typ 170 NEXT p
typ 180 PRINT "gaat het";
typ 190 FOR p = 1 TO 10000
typ 200 NEXT p
typ 210 PRINT "met jou?"
typ 220 PRINT "einde"

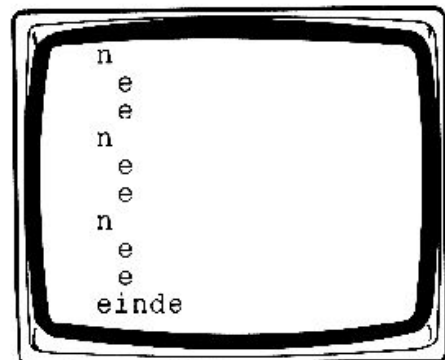
typ RUN
```

Bij deze voorbeelden staat er keurig een lus (bijv. regel 100 en 110) en daarna een andere lus (bijv. regel 130 en 140). Maar, je kunt de ene lus ook in de andere zetten. Wat je moet onthouden is, dat de computer zoekt welke FOR en NEXT het dichtst bij elkaar staan. Deze maakt hij ook voor het eerst af.

Kijk maar eens naar het volgende voorbeeld:

```
typ 300 FOR a = 1 TO 3
typ 310 PRINT "n"
typ 320 FOR b = 1 TO 2
typ 330 PRINT "e"
typ 340 NEXT b
typ 350 NEXT a
typ 360 PRINT "einde"

typ RUN
```



Goed opletten dat je ze niet door elkaar haalt, want dan loopt je programma niet goed.

#### Programma **STRAFWERK**

```
typ 10 LET aantal = 100  
typ 20 FOR s = 1 TO aantal  
typ 30 PRINT "ik moet stilzitten"  
typ 40 NEXT s  
typ 50 PRINT "einde"
```

Het voordeel van deze lus is, dat je het getal in vakje aantal makkelijk kunt veranderen. Dus als je de volgende keer 200 strafregels moet schrijven dan verander je alleen: aantal = 200.

*Met de FOR ... NEXT-opdracht kun je de opdracht die er tussen staat herhalen*

## herstel

---

### 1 Heeft elke NEXT een FOR?

```
10 FOR j = 1 to 5
20 PRINT j                                (fout)

10 FOR j = 1 TO 5
20 PRINT j                                (goed)
30 NEXT j
```

### 2 Staan de goede lussen bij elkaar?

```
10 FOR a = 1 TO 3
20 FOR b = 1 TO 3
30 NEXT a                                (fout)
40 NEXT b
```

Lus a loopt door lus b heen. Kijk maar:



De lussen mogen niet door elkaar lopen.

```
10 FOR a = 1 TO 3
20 FOR b = 1 TO 3
30 NEXT b                                (goed)
40 NEXT a
```

Geef zonodig een LIST om te kijken of het programma nu goed in het geheugen zit. Geef dan de computer de opdracht om jouw programma uit te voeren.

typ RUN (niet vergeten een **ENTER** te geven)

---

## opgaven

---

- 1** Schrijf met behulp van een FOR ... NEXT-lus een programma dat de getallen 1 tot en met 10 op het TV-scherm zet.
- 2** Zet 10 keer jouw naam op het TV-scherm met behulp van FOR ... NEXT.
- 3** Verander dit programma door van een FOR ... NEXT-lus gebruik te maken.

```
typ 10 LET j = 0
typ 20 LET j = j + 1
typ 30 PRINT j
typ 40 IF j = 10 THEN GO TO 60
typ 50 GO TO 20
typ 60 PRINT "einde"
```

- 4** Met FOR ... NEXT kun je ook goed een programma schrijven dat je bijvoorbeeld laat zien wat de tafel van twee is, maar het kan ook voor de tafel van 5 of van 25 of van ...  
Probeer het maar eens, achterin hebben wij er een mooi voorbeeld van.
-



# DE READ- EN DATA-OPDRACHTEN

## READ ... DATA

---

*READ ... DATA zijn opdrachten waarmee  
je zegt Lees (READ) de Gegevens (DATA) in*

---

*In BASIC:*

10 READ a

50 DATA 7

100 READ a\$

150 DATA "piet"

*In Nederlands:*

*lees a (een getal)*

*getal is 7*

*lees a\$ (een naam)*

*naam is piet*

---

READ en DATA zijn groene woordjes boven de A- en D-toetsen. Je krijgt ze door tegelijk **CAPS SHIFT** en **SYMBOL SHIFT** in te drukken. Er verschijnt een E in de cursor. Als je nu op A drukt verschijnt READ. Als je vanuit de E-cursor op de D-toets drukt verschijnt DATA.

READ is het Engelse woord voor lees. DATA is het Engelse woord voor gegevens. READ zoekt de gegevens op die in een DATA-opdracht staan. Net zoals bij INPUT, is een \$ voor namen (of getallen waar je niet mee hoeft te rekenen). De DATA moet in het geval van een naam wel tussen " " staan. Het verschil tussen INPUT en READ is, dat INPUT gegevens vraagt aan de gebruiker. READ haalt de gegevens ergens anders uit het programma. Als er in een programma een READ-opdracht staat, gaat het programma zoeken naar een regel waar DATA staat. DATA is een ander woord voor gegevens. Met READ en DATA kun je veel leuke dingen doen.

*typ* 10 READ a, b

*typ* 20 PRINT a, b

*typ* 30 DATA 5, 3

*typ* 40 PRINT "einde"

*typ* RUN



De READ-opdracht in regel 10 laat de computer zoeken naar een regel waar DATA staat. Het programma springt even naar regel 30. In de vakjes a en b worden dan de getallen 5 en 3 gestopt. Dan gaat het programma weer terug om de volgende regel (20) uit te voeren. Deze regel 20 zet de gegevens op het TV-scherm. Daarna is het programma klaar.

Als je net zoals hier om twee dingen vraagt (a en b), dan moeten er ook twee dingen bij DATA staan. Staat er bijvoorbeeld maar één of staan er drie, dan gaat het fout.

```
typ 10 DATA 5, 3
typ 30 READ a, b
typ 100 PRINT a, b
typ 300 PRINT "einde"

typ RUN
```

Je ziet dat het er niet toe doet of eerst de DATA komt en dan de READ-opdracht of andersom. Meestal wordt wel eerst de READ-opdracht gekozen en daarna de DATA-opdracht. De DATA-opdracht staat in grote programma's vaak aan het einde.

Je kunt ook namen en telefoonnummers in de DATA-opdracht zetten. Daarmee kun je dan een telefoonlijstje maken van je vrienden.

```
typ 10 READ n$, t$
typ 20 PRINT n$, t$
typ 30 DATA "piet", "030-876321"
typ 40 PRINT "einde"

typ RUN
```



Je ziet, dat je hier het stringteken (\$) moet gebruiken voor telefoonnummers.

Je begrijpt dat we met READ en DATA hele leuke programma's kunnen maken. Wat dacht je hiervan?

### Programma **TELEFOONNUMMERS**

```
typ 10 READ n$, t$
typ 20 IF n$ = "stop" THEN GO TO 500
typ 30 PRINT n$, t$
typ 40 GO TO 10

typ 100 DATA "piet", "030-123456"
typ 110 DATA "els", "010-654321"
typ 120 DATA "hans", "070-112233"
typ 130 DATA "bert", "20-776655"

typ 490 DATA "stop", ""
typ 500 PRINT "einde"
typ RUN
```

Hoe werkt dit programma nu?

10 READ n\$, t\$

*Lees de gegevens.  
Zoek regel waarin  
DATA staat.*

20 IF N\$ = "stop" THEN GO TO  
500

*Als in vakje n\$  
'stop' komt, ga dan  
naar regel 500.*

30 PRINT n\$, t\$

*Zet de gegevens op  
TV.*

40 GO TO 10

*Ga weer naar regel 10 om via READ de volgende gegevens in te lezen.*

100 DATA "piet", "030-123456"  
110 DATA "els", "010-654321"  
120 DATA "hans", "070-112233"  
130 DATA "bert", "020-776655"

*Hier staan de gegevens. Door de READ-opdracht op regel 10 worden er telkens twee gelezen.*

*Tussen regel 130 en 490 kun je nog veel meer namen en telefoon nummers zetten.*

490 DATA "stop", ""

*Eind van de gegevens.*

500 PRINT "einde"

*Eind van programma.*

*READ ... DATA zijn opdrachten waarmee je zegt Lees (READ) de Gegevens (Data) in*

## herstel

---

### 1 Een komma te veel.

10 DATA 5, 10

20 READ a, b, (fout)

De komma achter b is te veel.

### 2 Vergeten de DATA-opdracht te geven:

10 PRINT 7, 10

(fout)

20 READ a, b

10 DATA 7, 10

20 READ a, b

(goed)

30 PRINT a, b

### 3 Teveel gegevens achter de DATA-opdracht.

10 DATA 5, 10, 9

20 READ a, b (fout)

Achter READ moeten net zoveel letters staan als getallen achter DATA. Als er 3 getallen staan, moeten er ook 3 letters zijn.

10 DATA 5, 10, 9

20 READ a, b, c (goed)

### 4 Komma in plaats van een punt.

10 READ a

(fout)

20 DATA 25,50

Als je een komma achter de 25 zet, dan wordt 50 het volgende getal. De computer denkt dat het twee getallen zijn, 25 en 50. Met READ a vertel je de computer dat hij één getal moet lezen. Dus dat gaat fout.

10 READ a

20 DATA 25,50 (goed)

**5** Naam en getal verwisseld.

```
10 READ a, n$ (fout)
20 DATA "piet", 17
```

Met READ vraag je eerst om een getal voor vakje a. Maar in DATA staat eerst een naam en dat gaat niet. Namen kunnen alleen in vakjes waar een \$ (stringtekentje) achter staat. Vergeet de aanhalingstekens niet!

```
10 READ a, n$
20 DATA 17, "piet" (goed)
```

---

## **opgaven**

---

- 1** Je vrienden Jan, Klaas en Wim zijn 10, 12 en 15 jaar oud. Maak een programma met READ en DATA. Laat het programma de naam en leeftijd van je vrienden op het TV-scherm zetten.
  - 2** Je hebt een vriendin Mariska van 14 jaar erbij gekregen. Pas het programma aan.
  - 3** Maak een telefoonlijst van je beste vrienden. Gebruik de READ- en DATA-opdracht.
  - 4** Doe nu hetzelfde voor vrienden, familie en kennissen. (Achterin staat, hoe je dit programma op cassette kunt bewaren.)
  - 5** Wat moet er eerst in een programma gebruikt worden, READ of DATA?
-

# DE RND- OPDRACHT

## RND

---

*RND is een opdracht waarmee je een verrassingsgetal van de computer krijgt*

---

*In BASIC:*

10 LET x= RND

*In Nederlands:*

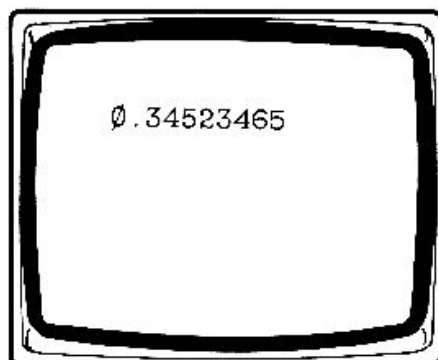
*zet in vakje x een verrassingsgetal*

---

RND krijg je door **SHIFT** en **SYMBOL** tegelijk in te drukken. In de cursor verschijnt een E. Als je nu de T-toets indrukt krijg je RND.

Voor al voor veel spelletjes is het handig, als je een soort dobbelsteen hebt. Iets dat getallen geeft die je van te voren niet wist, die een verrassing zijn. Ook bij een dobbelsteen weet je van te voren niet, welk getal er boven zal liggen. RND is een soort van computerdobbelsteen, dit geeft telkens een verrassingsgetal. Dit getal ligt tussen de 0 en 1. Typ het volgende programma maar in, dat zal het je laten zien.

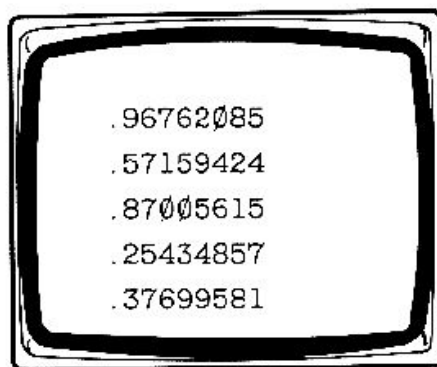
```
type 20 LET x = RND
typ 30 PRINT x
typ RUN
```



De computer heeft hier 0.34523465 als verrassingsgetal genomen. Bij jou was het waarschijnlijk een ander getal. En als je nog eens RUN intypt, komt er weer een ander getal.

Volgende voorbeeld:

```
typ 10 FOR i = 1 TO 5
typ 20 LET x = RND
typ 30 PRINT x
typ 40 NEXT i
typ RUN
```



5 verrassingsgetallen.

Met deze getallen kunnen we nog niet zoveel. Veel mooier is het, als we getallen tussen 1 en 10 kunnen krijgen. Dat gaat op de volgende manier.

```
typ 10 LET x = RND
typ 20 PRINT x
typ 30 LET y = 10 * x
typ 40 PRINT y
typ 50 LET z = INT (y)
typ 60 PRINT z
typ RUN
```

Wat hebben we nu gedaan:



Een RND-getal is bijvoorbeeld 0.34523465

Dat nemen we 10 keer  $\frac{10 \text{ keer}}{3.4523465}$

Afkappen met INT is 3

INT maakt er een heel getal van. Bijvoorbeeld:

$INT(3.757865) = 3$

$INT(2.489265) = 2$

Zo, nu kunnen we de computer een getal tussen 1 en 10 laten kiezen. Het kan nog iets mooier en dat staat hieronder:

---

LET x = INT(RND \* 10 + 1) geeft getal van 1 t/m 10.

LET x = INT(RND \* 100 + 1) geeft getal van 1 t/m 100.

LET x = INT(RND \* 1000 + 1) geeft getal van 1 t/m 1000.

---

#### Programma **RAAD EEN GETAL**

```
typ 10 PRINT "raad een getal van 1 t/m 10"
typ 20 LET x = INT (RND * 10 + 1)
typ 30 INPUT g
typ 40 IF g = x THEN GO TO 70
typ 50 PRINT "nee, nog niet geraden"
typ 60 GO TO 30
typ 70 PRINT "ja, je hebt mijn getal geraden"
typ 80 PRINT "het getal is "; x
typ 90 PRINT "einde"

typ RUN
```

Gaat alles goed? Dan heb je een spelletje gemaakt.

*RND is een opdracht waarmee je een verrassingsgetal van de computer krijgt*

## herstel

---

Een haakje te weinig.

Bijv.  $10 \text{ LET } x = \text{INT} (\text{RND} * 10 + 1$

↖ haakje vergeten

Er moeten namelijk aan de linkerkant evenveel haakjes staan als haakjes aan de rechterkant en dat klopt hier niet.

---

## opgaven

---

- 1** Maak het programma GETALLEN RADEN op blz. 90 leuker door op de regels 20, 40 en 60 een RND-getal te nemen van 1 t/m 10.
  - 2** Hoe kun je een RND (random)-getal tussen 1 en 5 krijgen?
  - 3** Hoe kun je een RND-getal tussen 1 en 250 krijgen?
  - 4** Verander het programma RAAD EEN GETAL van 1 t/m 10 zo, dat hij dobbelsteengetallen geeft, dus van 1 t/m 6. (Zie voor een mooi voorbeeld achterin op bladzijde 105.)
-



DEEL 3

# **PROGRAMMA'S**



# REKENEN MET DE COMPUTER

Heb je vaak (huis)werk waarbij je moet rekenen? Als het antwoord hier ja op is, dan is het volgende iets voor jou. Je kunt de computer heel goed gebruiken bij het maken van je rekenwerk. De computer is een geduldige leraar die je goed kan helpen met leren rekenen of rekensommen maken. Optellen, aftrekken, delen of vermenigvuldigen, het maakt de computer niets uit. Grote of kleine getallen, veel sommen of weinig sommen; de computer doet het voor je. Het programma dat je daarvoor nodig hebt kun je zelf maken. Laten we eens een rekenprogramma stap voor stap opzetten. Je moet dan iedere keer als er staat typ, bijvoorbeeld:

```
typ 10 PRINT "rekenen"
```

die regel precies zo intypen. Tussen die regels zullen we stap voor stap kijken waar die regel voor nodig is en wat je verder moet doen. Aan het eind van die regels heb je een compleet rekenprogramma wat je net zo vaak kunt gebruiken als je maar wilt. Aan het eind zullen we het programma nog eens in zijn geheel geven.

Dus als je niet alle stappen wilt volgen kun je gelijk het hele programma intypen. Je kunt altijd daarna nog kijken welke stappen je gemaakt hebt en waar ze voor nodig zijn.

Zo, laten we beginnen.

## **Programma REKENEN**

We willen een programma maken waarmee we kunnen optellen, aftrekken, delen en vermenigvuldigen.

Allereerst typ je dan de regels waarmee je de computer vraagt of je wilt optellen, aftrekken, delen of vermenigvuldigen.

```

typ 10 PRINT "*** rekenen ***"
typ 20 PRINT "typ 1 voor optellen"
typ 30 PRINT "typ 2 voor aftrekken"
typ 40 PRINT "typ 3 voor delen"
typ 50 PRINT "typ 4 voor vermenigvuldigen"

```

Na deze regels moeten we er voor zorgen dat de keuze in een vakje wordt opgeborgen. Laten we dat vakje de naam keuze geven.

```
typ 60 INPUT keuze
```

Regel 70 slaan wij even over die gebruiken wij straks. Zo nu kunnen we in dat vakje het getal 1, 2, 3 of 4 zetten. De getallen waarmee we gaan rekenen moeten ook in vakjes komen. In het eerste vakje een getal en in het tweede vakje een getal. Die getallen kunnen we daarna bij elkaar optellen, van elkaar aftrekken, op elkaar delen of vermenigvuldigen. Laten we de vakjes eerste en tweede noemen.

```

typ 80 PRINT "geef nu twee getallen op"
typ 90 PRINT "eerste getal ="
typ 100 INPUT eerste
typ 120 PRINT "tweede getal ="
typ 130 INPUT tweede

```

Zo nu hebben we 3 vakjes gemaakt waarin de keuze zit (optellen, aftrekken, delen of vermenigvuldigen), het eerste getal en een vakje waarin het tweede getal zit.

<i>Bijv.</i>	<i>keuze</i>	<i>eerste</i>	<i>tweede</i>
	1	8	4

Nu moeten we de computer laten kijken of er in het vakje keuze een 1, 2, 3 of 4 zit

want is het een 1 dan moet hij optellen  
           is het een 2 dan moet hij aftrekken  
           is het een 3 dan moet hij delen  
           is het een 4 dan moet hij vermenigvuldigen.

Zoals je weet kun je daar de IF ... THEN-opdracht voor gebruiken. IF ... THEN betekent ALS ... DAN.

Dus ALS 1 DAN +  
ALS 2 DAN -  
ALS 3 DAN / (delen)  
ALS 4 DAN \* (vermenigvuldigen)

Het getal 1, 2, 3 of 4 staat in het vakje keuze. En de getallen die we bijvoorbeeld willen optellen staan in de vakjes 'eerste' en 'tweede'.

Als we een sommetje op papier maken ziet dat er bijvoorbeeld zo uit:

$$12 = 8 + 4$$

Getal 8 zit in het eerste vakje en getal 4 zit in het tweede vakje. Waar moet het getal 12, de som, dan in?

Daar moeten we ook een vakje voor maken en dat kunnen we som noemen.

We hebben nu de volgende vakjes:

<i>Bijv.</i>	<i>keuze</i>	<i>som</i>	<i>eerste</i>	<i>tweede</i>
	1	12	8	4

We moeten nu de keuze door de computer laten maken. Als de computer het getal van de keuze (bijv. 1) gevonden heeft moet hij de optelling gaan maken. Laten we het eens uitschrijven:

als keuze = 1 dan som = eerste + tweede  
als keuze = 2 dan som = eerste - tweede  
als keuze = 3 dan som = eerste / tweede (delen)  
als keuze = 4 dan som = eerste \* tweede (verm.)

Nu nog iets aanpassen zodat de computer het begrijpt:

```
typ 140 IF keuze = 1 THEN LET som = eerste + tweede  
typ 150 IF keuze = 2 THEN LET som = eerste - tweede  
typ 160 IF keuze = 3 THEN LET som = eerste / tweede  
typ 170 IF keuze = 4 THEN LET som = eerste * tweede
```



Zo we zijn nu bijna klaar. De keuze is gemaakt. Het eerste en tweede getal hebben we en de som is gemaakt. We moeten het antwoord nog op het TV-scherm krijgen. Dat gaat als volgt:

```
typ 300 PRINT "antwoord = " ; som
```

We zouden het programma kunnen stoppen. Als je een som moet maken of je wilt wat oefenen met rekenen dan kun je dit programma gebruiken. Iedere keer typ je RUN en het programma helpt je.

Het programma is alleen nog onhandig. Je hebt meestal meer dan een som. Soms wel 10 of nog meer. Dan is iedere keer RUN typen vervelend. We kunnen het programma heel gemakkelijk handiger maken door de computer iedere keer weer de volgende som te laten vragen. Dit gaat als volgt:

```
typ 310 GO TO 80
```

GO TO 80 betekent ga naar regel 80. En daar staat 'geef nu de twee getallen op'. Je kunt dan gewoon met de volgende som beginnen.

Door de computer nu iedere keer opnieuw te laten beginnen heb je bereikt dat je een aantal sommen achter elkaar kunt maken. Je kunt nu maar doorgaan en doorgaan.

Je moet zelfs doorgaan omdat het programma niet weet hoe te stoppen. Als je bijvoorbeeld 10 sommen hebt gemaakt en je wilt stoppen, hoe moet dat dan? Je kunt altijd de computer uitzetten of een stoptoets gebruiken, maar dat is niet mooi. Nee, laten we de mogelijkheid om te stoppen netjes in het programma opnemen.

```
typ 70 PRINT "je kunt stoppen door 999"
```

Dit moeten we iedere keer controleren. Bijvoorbeeld:

als 999 dan stoppen

Het best kunnen we dat doen na de vraag om het eerste getal (regel 90). Komt in het vakje eerste 999 dan wil je stoppen. In computertaal wordt dit:

```
typ 110 IF eerste = 999 THEN GO TO 400
```

En op regel 400 moeten we dan het programma stoppen. Want GO TO 400 betekent 'ga naar regel 400'.

```
typ 400 PRINT "einde"
```

Zo, dit is het hele programma. Als je overal waar typ voor staat hebt ingetypt dan ben je nu klaar met het programma. Hier volgt het hele programma achter elkaar zonder commentaar erbij.

### Programma **REKENEN**

---

```
typ 10 PRINT "*** rekenen ***"
typ 20 PRINT "typ 1 voor optellen"
typ 30 PRINT "typ 2 voor aftrekken"
typ 40 PRINT "typ 3 voor delen"
typ 50 PRINT "typ 4 voor vermenigvuldigen"
typ 60 INPUT keuze
typ 70 PRINT "je kunt stoppen door 999"
typ 80 PRINT "geef nu twee getallen op"
typ 90 PRINT "eerste getal = "
typ 100 INPUT eerste
typ 110 IF eerste = 999 THEN GO TO 400
typ 120 PRINT "tweede getal = "
typ 130 INPUT tweede
typ 140 IF keuze = 1 THEN LET som = eerste + tweede
typ 150 IF keuze = 2 THEN LET som = eerste - tweede
typ 160 IF keuze = 3 THEN LET som = eerste / tweede
typ 170 IF keuze = 4 THEN LET som = eerste * tweede
typ 300 PRINT "antwoord = " ; som
typ 310 GO TO 80
typ 400 PRINT "einde"
```

---

Het programma is klaar en je kunt er mee rekenen. Je kunt het programma ook nog uitbreiden. Heel leuk is het om eerst zelf een antwoord te bedenken en dan de computer te laten controleren of jouw antwoord goed is. Je gebruikt de computer dan om jou te overhoren. En je weet dat de computer heel veel geduld heeft en nooit kwaad wordt.

## Programma TOETSENBORD

```
100 REM een klein programma om
110 REM vertrouwd te raken met
120 REM het toetsenbord
130 REM
140 REM Baarda en van Londen
150 REM
160 INPUT "na hoeveel fouten wil je stoppen ";
    fouten
170 IF fouten = 0 THEN LET fouten = 1
180 PRINT:PRINT "typ het laatste teken na"
190 PRINT:PRINT "je kunt stoppen met SPACE"
200 PRINT
210 LET false = 0 : LET ok = 0
215 LET b$=INKEY$ : LET a$=""
220 LET x = 32 + INT (RND * 90 +1)
230 IF x = 91 THEN GO TO 220
240 IF x = 92 THEN GO TO 220
250 IF x = 93 THEN GO TO 220
255 LET x$=CHR$ (x)
260 PRINT "-> " ; x$ ; " ";
265 LET b$=a$
270 LET a$ = INKEY$
280 IF a$ = " THEN GO TO 270
285 IF a$ = b$ THEN GO TO 270
290 PRINT a$;
300 IF a$ = x$ THEN PRINT "is goed" : LET
    ok=ok+1
310 IF a$ = CHR$ (32) THEN GO TO 360
320 IF a$ < > x$ THEN PRINT " is fout. Het was
    ";x$: LET false=false+1
330 PRINT
340 IF false>=fouten THEN GO TO 360
350 GOTO 220
360 PRINT:PRINT "je score is "; INT
    ( (ok/(ok+false) ) * 100); " procent"
```

### Programma **REGENBOOG**

```
10 REM regenboog sinclair
20 FOR x = 0 TO 7
30 BORDER x
40 FOR y = 0 TO 7
50 PAPER y
60 CLS
70 FOR z = 0 TO 7
80 INK z
90 PRINT "kleuren van de sinclair"
100 FOR p = 1 TO 20 : NEXT p
110 NEXT z
120 NEXT y
130 NEXT x
140 PAPER 7
150 INK 1
```

### Programma **TOONLADDER**

```
10 FOR x = 1 TO 8
20 READ a
30 BEEP 1,a
40 NEXT x
50 DATA 0,2,4,5,7,9,11,12
```

Op regel 50 zie je de getallen voor de toonladder.

0 = do	2 = re	4 = mi	5 = fa
7 = sol	9 = la	11 = si	12 = do

Hiermee kun je zelf muziek maken.

### Programma **OLD MAC DONALD**

```
10 REM old mac donald
20 REM Baarda en de Goede
30 READ a
40 IF a=20 THEN GO TO 110
50 BEEP .4,a
60 GO TO 30
70 DATA 7,7,7,7,4,4,2,2,11
80 DATA 11,9,9,7,7,7,2,7,7
90 DATA 7,7,4,4,2,2,11,11,9
100 DATA 9,7,7,7,20
110 PRINT "EINDE"
```

### Programma **HAPPY BIRTHDAY**

Als je de regel 70, 80, 90 en 100 verandert zoals hieronder staat krijg je Happy Birthday.

```
70 DATA 0,0,2,0,5,4,0
80 DATA 0,2,0,7,5,0,0
90 DATA 12,9,5,5,4,2,11
100 DATA 11,9,5,7,5,20
```

### Programma **GROOTSTE GEMENE DELER**

```
10 REM grootste gemene deler
20 REM Baarda en van Londen
30 PRINT "Grootste Gemene Deler"
40 INPUT "eerste getal "; a
50 INPUT "tweede getal" ; b
60 REM kijken of a kleiner is dan b
70 REM zo ja, dan a en b omkeren
80 IF a<b THEN LET h=a : LET a=b : LET b=h
90 REM GGD berekening
100 LET g = INT (a/b) : LET r = a - (g*b)
110 IF r < > 0 THEN LET a=b : LET b=r : GO TO 100
120 PRINT:PRINT "de GGD is "; b
```

### Programma **UITROLLENDE DOBBELSTEEN**

```
10 REM uitrollende dobbelsteen
20 REM Baarda en van Londen
30 LET x = 2
40 FOR j = 1 TO x : NEXT j
50 LET x = x * 1.3 + 50
60 LET r = INT (RND * 6 + 1)
70 PRINT r; " ";
80 IF x < 1000 THEN GO TO 40
90 PRINT "einde"
```

### Programma **TWEE DOBBELSTENEN**

```
10 REM twee dobbelstenen
20 REM Baarda en van Londen
30 PRINT "hoeveel keer wil je gooien ";
40 INPUT n
50 PRINT
60 PRINT "eerste tweede totaal"
70 REM lus voor aantal worpen
80 FOR i = 1 to n
90 REM de dobbelsteen getallen
100 LET r1 = INT (RND * 6 + 1)
110 LET r2 = INT (RND * 6 + 1)
120 LET som = r1 + r2
130 PRINT " " ; r1; " " ; r2; " " ; som
140 NEXT i
150 PRINT "einde"
```

### Programma **GETALLEN OMKEREN**

```
10 REM getallen omkeren
20 REM Baarda en van Londen
30 PRINT "getallen omkeren"
40 PRINT
50 PRINT "typ 2 getallen door een komma  
gescheiden"
60 PRINT "bijvoorbeeld 21,14"
70 INPUT a,b
```

```

80 PRINT "jouw getallen zijn "
90 PRINT a;" en ";b
100 PRINT
110 LET r=a : PRINT "reserve krijgt eerste
    getal = ";r
120 LET a=b : PRINT "eerste krijgt nu de
    tweede = ";a
130 LET b=r : PRINT "tweede krijgt nu de
    reserve = ";b
140 PRINT
150 PRINT "de computer maakt ervan ";a;" en ";b
160 PRINT "einde"

```

### Programma **NAMEN OMKEREN**

```

10 REM namen omkeren
20 REM Baarda en van Londen
30 PRINT "namen omkeren"
40 PRINT
50 PRINT "typ 2 namen door een komma
    gescheiden"
60 PRINT "bijvoorbeeld arleen, oscar"
70 INPUT a$,b$
80 PRINT "jouw namen zijn "
90 PRINT a$;" en ";b$
100 PRINT
110 LET r$=a$ : PRINT "reserve krijgt eerste
    naam = ";r$
120 LET a$=b$ : PRINT "eerste krijgt nu de
    tweede = ";a$
130 LET b$=r$ : PRINT "tweede krijgt nu de re-
    serve = ";b$
140 PRINT
150 PRINT "de computer maakt ervan ";a$;" en ";
    b$
160 PRINT "einde"

```

## Programma HUISWERKHULP

```
10 REM Baarda en van Londen
20 PRINT "huiswerkhulp. Je moet telkens"
30 PRINT "twee dingen opgeven. Bijv."
40 PRINT "Land en Hoofdstad of"
50 PRINT "Engels en Nederlands"
60 REM maak plaats voor 25 vragen
70 DIM a$(25) : DIM x$(25,15) : DIM w$(15)
80 DIM b$(25) : DIM y$(25,15)
90 LET n=0 : LET t=0 : LET g=0
100 PRINT
110 INPUT "1 = "; a$
120 INPUT "2 = "; b$
130 PRINT
140 PRINT "je moet eerst de stof opgeven"
150 PRINT "bijv. nederland en amsterdam"
160 PRINT "als je klaar bent typ je stop"
170 PRINT
180 REM het huiswerk
190 FOR i= 1 TO 25
200 PRINT a$;i: INPUT x$(i)
210 IF x$(i) (1 TO 4) = "stop" THEN GO TO 250
220 PRINT b$;i: INPUT y$(i)
230 LET n=n+1
240 NEXT i
250 LET x=0
260 PRINT:PRINT "ik geef je nu ";a$
270 LET rep=x : LET x=INT (RND * n +1)
280 IF x=rep THEN GO TO 270
290 LET t=t+1
300 PRINT:PRINT x$(x)
310 INPUT "--> ";w$
320 IF w$ (1 TO 4) = "stop" THEN GO TO 360
330 IF w$=y$(x) THEN PRINT "ja goed !!" : LET
    g=g+1 : GO TO 270
340 PRINT "nee jammer, het antwoord is ": PRINT
    y$(x)
350 GO TO 270
```



```

360 PRINT
370 PRINT "je had er ";g;" goed"
380 PRINT "van de ";t-1;" pogingen"
390 PRINT "tot de volgende keer"
400 PRINT "einde"

```

#### Programma **PRIEMGETALLEN**

```

10 REM priemgetallen
20 REM Baarda en van Londen
30 REM
40 PRINT "tot welk getal wil je"
50 PRINT "de priemgetallen weten"
60 INPUT max
70 PRINT 2,
80 FOR i=3 TO max
90 LET q=SQR (i)
100 LET n=1
110 LET n=n+1
120 IF INT(i/n) <> i/n THEN GO TO 140
130 GO TO 160
140 IF n<q THEN GO TO 110
150 PRINT i,
160 NEXT i
170 PRINT "einde"

```

#### Programma **REKENKUNDIGE TAFELS**

```

10 REM rekenkundige tafels
20 REM Baarda en van Londen
30 PRINT "welke tafel wil je"
40 INPUT "de tafel van " ;x
50 PRINT
60 FOR n=1 TO 10
70 PRINT n;" maal ";x;" = ";n*x
80 NEXT n
90 PRINT "einde"

```

## Programma **TELEFOONBOEK**

```
10 REM telefoonboek
20 REM Baarda en van Londen
100 PRINT:PRINT "wat is de naam"
110 INPUT n$
120 READ a$, t$
130 IF a$="stop" THEN PRINT "die ken ik niet" :
    GO TO 500
140 IF n$ < > a$ THEN GO TO 120
150 PRINT "telefoonnummer= ";t$
200 DATA "kees", "010-12345"
210 DATA "marie", "01234-3344"
220 DATA "opa", "070-99999"
300 REM
310 REM *****
320 REM vanaf regel 230 t/m 480
330 REM ruimte voor meer namen
340 REM en telefoonnummers
350 REM *****
360 REM
490 DATA "stop", ""
500 PRINT "einde"
```

## Programma **OVERHOREN TAFELS**

```
10 REM overhoren tafels
20 REM Baarda en van Londen
30 PRINT "met welke tafel wil je oefenen"
40 INPUT "de tafel van" ;x
50 PRINT
60 FOR n=1 TO 10
70 PRINT n; " maal "; x; " = ";
80 INPUT a
90 IF a=n*x THEN PRINT "ja goed!!!"
100 IF a < > n*x THEN PRINT "jammer het is ";
    n*x
110 NEXT n
120 PRINT "einde"
```

## Programma NONSENSGENERATOR

```
10 REM nonsens generator
20 REM Baarda en van Londen
30 PRINT "De computer is gek"
40 PRINT "geworden. Hij praat"
50 PRINT "nonsens"
60 PRINT:PRINT
70 LET max=INT (RND * 5 + 1)
80 IF max <2 THEN GO TO 70
90 LET r$=""
100 FOR j=1 TO max
120 LET x=INT (RND * 34 + 56)
130 IF (x<65) OR (x>87) OR (x=81) THEN GO TO
    120
140 LET w$=CHR$ (x)
150 LET r$=r$+w$
160 NEXT j
170 PRINT r$;" ";
180 GO TO 70
```

# **BIJLAGEN**



## BIJLAGE 1

# DE CASSETTERECORDER

### BEWAREN

Het is vervelend dat telkens als je de computer afzet je programma weg is. Je moet dan steeds opnieuw programma's intikken. Dat is ook niet nodig. Je kan ze namelijk bewaren op een cassettebandje. Als je de volgende keer het programma wilt gebruiken, stop je de cassette in de recorder die bij de computer zit. Je geeft dan de computer de opdracht te lezen wat er op de cassette staat. En in plaats van typen zit het programma in een paar seconden weer in het geheugen van de computer. Als je zelf een programma hebt gemaakt, kun je die ook op een cassette bewaren.

Laten wij eens een voorbeeldje nemen. Typ het volgende programma maar eens in:

```
10 PRINT "is het mooi weer"  
20 PRINT "typ j of n"  
30 INPUT w$  
40 IF w$ = "j" THEN PRINT "fijn, hopelijk  
    blijft het zo"  
50 IF w$ = "n" THEN PRINT "jammer hopelijk is  
    het morgen beter"
```

### SAVE

Wij willen dit programma bewaren onder de naam HET WEER.

Je doet dit met SAVE en die zit op de S-toets. Als je SAVE "HET WEER" typt, zegt de computer "Start tape, then press any key". Dat wil zeggen dat je de REC- en de PLAY-toets op de cassetterecorder tegelijk in moet drukken, je moet wel stevig drukken. Controleer nog even als de cassette draait of beide toetsen ingedrukt zijn, je kunt dat zo zien. Bij sommige cassetterecorders hoeft je alleen de REC-toets in te drukken. Dat moet je even uitproberen.

Als de cassette recorder draait, moet je op een toets van de computer, bijvoorbeeld **ENTER** drukken.

Als de computer klaar is met het programma op de cassette zetten, dan schrijft hij OK onder in het beeld. Probeer het maar eens.

### **LADEN**

Zoals je weet, kun je je eigen programma's op een cassettebandje bewaren (zie het deel over BEWAREN). Maar je kunt ook de programma's van anderen in jouw computer laden via een cassettebandje. Het programma van een vriend bijvoorbeeld. En programma's die je in winkels koopt. Spelletjes als PACMAN en DONKEY zijn niets anders dan hele lange programma's, geschreven in een computertaal, die gebruik maken van tekenen, schrijven, rekenen en geluid maken met de computer. Stel je krijgt een cassette met PACMAN erop en je wilt dat spelletje gaan spelen. Hoe gaat dat dan?

### **LOAD**

Je doet dat met **LOAD** en die zit op de J-toets. Je stopt eerst de cassette in de recorder. Je moet wel de cassette terugspoelen naar waar PACMAN begint, je doet dat met de **REW**-toets die op de cassette recorder zit. Als je daar op drukt spoelt hij terug. Je moet nu de computer de opdracht geven dat hij PACMAN moet laden. Je doet dat met **LOAD "PACMAN"**. Als je de **PLAY**-toets van de cassette recorder indrukt gaat hij op zoek naar PACMAN. Als hij PACMAN gevonden heeft zegt hij

Program: PACMAN

en hij gaat het programma laden. Je ziet dat aan de bewegende streepjes aan de rand van het beeld.

Dat laden duurt eventjes en als hij klaar is zegt hij OK. Je moet dan op de **STOP**-toets van de cassette recorder drukken.

Je kunt beginnen door **RUN** in te typen en op **ENTER** te drukken. Probeer het maar eens met de cassette die je hebt. Je moet dan alleen geen PACMAN typen maar de naam van het programma dat jij op de cassette hebt. Weet je de naam niet, typ dan alleen **LOAD ""**

## BIJLAGE 2

# ANTWOORDEN OP DE VRAGEN

### REKENEN OP DE COMPUTER

- 1** PRINT 7 + 6 druk op **ENTER**  
13
- PRINT 18 + 33 druk op **ENTER**  
51
- enzovoort
- 2** PRINT 7 + 6 - 3 druk op **ENTER**  
10
- 3** Let erop dat je een punt in plaats van een komma typt.  
PRINT 2.50 - 0.75 druk op **ENTER**  
1.75
- 4** PRINT 50 - 3 \* (12.50 - 3) druk op **ENTER**  
21.50

### DE PRINT-OPDRACHT

- 1** 10 PRINT "vandaag is het"  
20 PRINT "zondag"
- 2** 10 PRINT "de zon "; spatie  
20 PRINT "schijnt" punt/komma
- 3** 10 PRINT "de zon", komma  
20 PRINT "schijnt"
- 4** 10 PRINT "de zon"  
15 PRINT  
20 PRINT "schijnt"



## DE INPUT-OPDRACHT

- 1** Dit programma is niet goed. In regel 10 wordt er een naam gevraagd. In regel 20 wordt via INPUT 1 een getal gevraagd. Als je een naam vraagt moet je achter 1 het stringteken \$ gebruiken. Als er staat INPUT 1\$ dan verwacht de computer een naam.
- 2** Dit programma is niet goed.  
Via regel 20 vraag je om een naam. Het \$ (stringteken) betekent dat de computer alles als een naam zal behandelen. In regel 30 laat je dan de computer rekenen met een naam en dat kan hij niet.
- 3** Dit programma is niet goed.  
Als je een naam vraagt mag je maar één letter en \$ gebruiken.
- 4** Opgave 1 moet worden.

```
10 PRINT "wat is je naam?"
20 INPUT 1$
30 PRINT "je naam is"; 1$
```

Opgave 2 moet worden.

```
10 PRINT "wat is je leeftijd?"
20 INPUT 1
30 n = 1 * 12
40 PRINT "je bent " ; n ; " maanden jong"
```

Opgave 3 moet worden.

```
10 PRINT "wat is je naam?"
20 INPUT n$
30 PRINT n$
```

## DE LET-OPDRACHT

**1** 10 PRINT "eerste getal = "  
20 INPUT eerste  
30 PRINT "tweede getal = "  
40 INPUT tweede  
50 LET som=eerste + tweede  
60 PRINT "som = ";  
70 PRINT som

**2** Op regel 40 moet staan

40 INPUT n\$

## DE GO TO-OPDRACHT

**1** 40 IF x = y THEN GO TO 70  
60 GO TO 30

Als je in regel 60 GO TO 20 getypt hebt is dat ook goed.

**2** 110 IF j\$ = "ja" THEN GO TO 140  
130 GO TO 20

**3** De computer moet iedere keer  $n = n + 1$  uitvoeren.  
Je zou daar het volgende programma voor kunnen maken:

10 n = 1  
20 PRINT n  
30 n = n+1  
40 GO TO 20

Om een programma te maken dat precies hetzelfde doet als in de opgave staat kun je ook het volgende nemen.

10 n = 1  
20 PRINT n  
25 IF n = 4 THEN GO TO 50  
30 n = n + 1  
40 GO TO 20  
50 PRINT "einde"

## DE GO SUB- EN RETURN-OPDRACHTEN

- 1** Er staat geen RETURN achter het hulpprogramma. Als je  
210 RETURN  
er bij zet is het programma goed.
- 2** De GO TO-opdrachten op regel 20, 40 en 60 moeten GO SUB-opdrachten zijn. Een RETURN zonder GO SUB werkt niet.
- 3** Als je vaak hetzelfde moet doen in een programma is het handiger om van dat stukje een hulpprogramma te maken. Je hoeft het dan maar één keer te schrijven. Je kunt er dan iedere keer met GO SUB naar toe springen en met RETURN weer terugkomen.

## DE REM-OPDRACHT

Het gebruik van REM-opdrachten is heel persoonlijk. De een zal er veel gebruik van maken en de ander maar weinig. Je zou op de volgende manier je programma's kunnen beginnen:

```
10 REM dit programma is van .....  
20 REM datum .....
```

## DE IF ... THEN-OPDRACHT

- 1** Verander regel 40 in  
40 IF t = 5 THEN PRINT "snel naar huis"  
Mooier is:  
5 LET y = 5  
40 IF t = y THEN PRINT "snel naar huis"  
Als je nu nog eens de tijd wilt veranderen, hoef je alleen maar regel 5 te veranderen en die is een stuk korter dan regel 40.
- 2** = gelijk aan

**3** > groter dan

**4** < kleiner dan

**5** <> ongelijk aan

**6** Verander regel 40 in

```
40 IF x = 3 THEN GO TO 70
```

Mooier is:

```
5 LET y = 3
```

```
40 IF x = y THEN GO TO 70
```

### DE FOR ... NEXT-OPDRACHT

**1** 20 FOR j = 1 TO 10  
30 PRINT j  
40 NEXT j

**2** 20 FOR j = 1 TO 10  
30 PRINT "arleen"  
40 NEXT j

**3** 20 FOR j = 1 TO 10  
30 PRINT j  
40 NEXT j

**4** Zie op bladzijde 108 het programma REKENKUNDIGE TAFELS.

Misschien heb je zelf zoiets gemaakt:

```
10 PRINT "welke tafel"  
20 INPUT tafel  
30 FOR j = 1 TO 10  
40 PRINT j * tafel  
50 NEXT j
```

## DE READ- EN DATA-OPDRACHTEN

- 1**    10 READ n\$ , 1  
      20 IF n\$ = "stop" THEN GO TO 500  
      30 PRINT n\$,1  
      40 GO TO 10  
      100 DATA "jan",10  
      110 DATA "klaas",12  
      120 DATA "wim",15  
      490 DATA "stop",0  
      500 PRINT "einde"
- 2**    Zet op regel 130  
      130 DATA "mariska", 14
- 3**    Zie het programma TELEFOONBOEK op bladzijde 109.
- 4**    In het programma TELEFOONBOEK kun je van regel 140 tot en met regel 480 zoveel namen en telefoonnummers zetten als je wilt.
- 5**    Dat maakt niet uit.  
      Meestal zet je eerst de READ-opdracht en daarna de DATA-opdracht.

## DE RND-OPDRACHT

- 1**    Programma GETALLEN RADEN  
      20 x = INT (RND \* 10 + 1)  
      40 x = INT (RND \* 10 + 1)  
      60 x = INT (RND \* 10 + 1)
- 2**    x = INT (RND \* 5 + 1)
- 3**    x = INT (RND \* 250 + 1)
- 4**    Je moet regel 10 en 20 veranderen.  
      10 PRINT "raad een getal van 1 t/m 6"  
      20 x = INT (RND \* 6 + 1)





Een nieuwe serie computerboeken, ditmaal speciaal voor kinderen geschreven. (Dit betekent niet dat de volwassene er niets uit kan leren!) De auteurs, Drs. B. Baarda en Drs. A. van Londen, zijn beide werkzaam aan de subfaculteit der pedagogische en andragogische wetenschappen van de Rijksuniversiteit Utrecht.

Het boek leert een jonge computergebruiker, zo veel mogelijk zonder hulp van een volwassene, zijn computer te gebruiken.

Het eerste deel van het boek behandelt op welke wijze een computer kan tekenen, schrijven, kleuren en geluid maken.

In het tweede deel wordt bekeken voor welke toepassingen de computer nog meer is te gebruiken. Bijvoorbeeld het onthouden en bewaren van een opstel, een tekening of een muziekstukje. Ook het uitvoeren van karweitjes zoals moeilijke sommen uitrekenen, strafregels schrijven en spelletjes ontwerpen, komen aan de orde. Kortom in dit deel wordt op speelse en nuttige wijze kennis gemaakt met de programmeertaal BASIC.

Het derde deel ten slotte geeft een aantal voorbeeldprogramma's om zelf in te toetsen en te spelen. Jonge computergebruikers zullen niet alleen door de speelse benadering van de tekst, maar ook door de vormgeving zeker worden aangesproken!

ISBN 90 201 1766 1

D.B. Baarda/A. van Londen

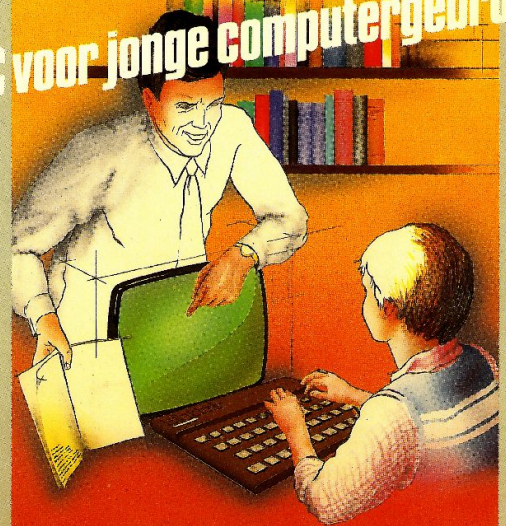
Leren omgaan met de ZX SPECTRUM



D.B. Baarda/A. van Londen

# Leren omgaan met de ZX SPECTRUM

BASIC voor jonge computergebruikers



Kluwer Technische Boeken