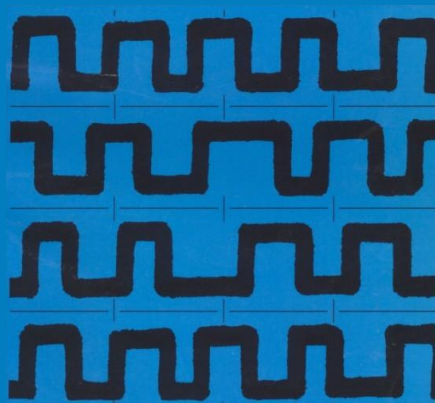


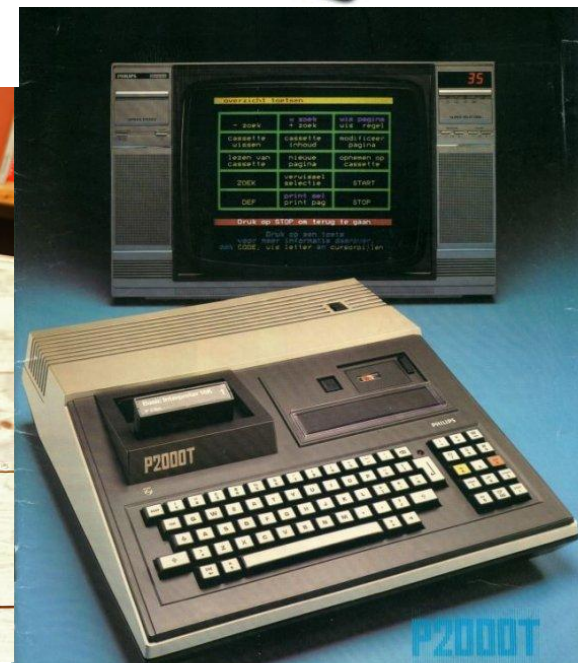
# basiccode

the 8-bit API  
that crossed the Berlin Wall



Rob Hagemans :: FOSDEM 2020 :: Brussels

# The 1980s: a Cambrian explosion of 8-bit “micro” computers



# BASIC, BASIC everywhere

... nor any one the same

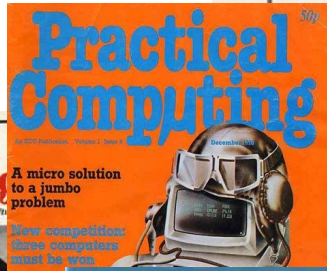
```
10 FOR I=0 TO 100
20 CLS
30 PRINT USING"FOSDEM ####";2020+I
40 NEXT I
```

```
10 FOR I=0 TO 100:PRINT CHR$(144)
20 PRINT"FOSDEM ";2020+i
30 NEXT:GOTO 10
```

```
10 TEXT
20 FOR I=0 TO 100:HOME
30 PRINT "FOSDEM";2020+I
40 NEXT I:GOTO 10
```



# Magazines & type-in listings



```

00 599 REM POPPING SOUND
00 600 FOR S=15 TO 0 STEP -1
      :SOUND 0,15,0,S:SOUND
      1,16,0,S:NEXT S
16 610 SOUND 0,0,0,0:SOUND 1
      ,0,0,0:RETURN
1A 999 REM CHECK FOR COLLISI
      ON
NK 1000 PEK=PEEK(53261):IF P
      EK=0 THEN RETURN
LQ 1002 GOTO 1010
NL 1005 FOR I=25 TO 10 STEP
      -5:SOUND 0,1,4,8:SOU
      ND 1,I+2,2,8:NEXT I:
      SOUND 0,0,0,0:SOUND
      1,0,0,0:RETURN
LM 1009 REM DID BALLOON HIT
      HEAD?
010 IF BY-BB<3 THEN PM$(
      H+BB,P0+BB)=B$:BB=BB
      -11:BAI=BAI-1:BOH=BO
      H+1:BAI(BOH)=C(I-1,B
      R/3):POP:GOTO 60
019 REM DID ARM HIT BALL
      OON?
020 IF STRIG(0)=0 THEN I
      F ABS(BY-178)<3 THEN
      1040
      ((A>PP)-(A<=PP))
      POKE 53249,A:POKE
      1:RETURN
      IF HIT THEN P=PP
      P>PP)-(P<=PP))*3
      K=BY TO BL STEP
      GOSUB 500:SOUND 0
      10,8:PM$(P1+K,F+K
      ):POKE 53249,P
      (((P>PP)-(P<=PP)
  
```



Player/missile graphics animate the player and balloons in the Atari version of "Balloon Crazy."

```

IF SC>BONUS THEN LF=
LF+1:BONUS=BONUS+100
0*LVL:COLOR 72:PLOT
LF,23
J 2040 IF BOH<1 THEN 2060
# 2050 FOR K=1 TO BOH:FOR J
=158-K*11 TO 168-K*1
1:PM$(P0+J,P0+J+12)=
B$:NEXT J:NEXT K:NEX
T I
B 2060 PM$(PT,PM)=N$:RETURN
ES 2499 REM CHOOSE BALLOON T
O FALL
JN 2500 BR=3+INT(RND(0)*6)*3
:FOR I=5 TO 2 STEP -
1:GOSUB 500:LOCATE B
R,1,BT:IF BT=32 THEN
  
```



## COMMODORE 64 AND VIC 20:

Change lines 10, 40 and 150 to:

PRINT CHR\$(147)

Change line 190 to:

190 R = INT (RND (1) \* 2) + 1

## IBM PC AND PCJR: Change lines

10, 40, and 150 to: CLS

Add or change these lines:

5 RANDOMIZE TIMER

190 R = INT (RND \* 2) + 1

## TRS-80 COLOR COMPUTER:

Change lines 10, 40 and 150 to: CLS

Change line 190 to:

190 R = RND(2)

## TI 99/4A: Change lines 10, 40 and

150 to: CALL CLEAR

Add or replace these lines:

5 RANDOMIZE

45 I = 0

Kijk bij gebruik van deze poster eerst in het lijstje hiernaast. Daar staan de weinige opdrachten die op (bijna) alle machines gelijk zijn. Door de kleine afmetingen van de hokjes is het niet mogelijk van alle opdrachten hun volledige kracht te laten zien. De meest voorkomende gevallen van een opdracht worden vermeld, maar een bredere betekenis blijft altijd mogelijk.  
Bij wel op: een zelfde syntax wil nog niet zeggen een zelfde betekenis.  
Bijvoorbeeld RND(exp) op de Enterprise levert een willekeurig geheel getal op, maar op andere machines veelal een willekeurig getal tussen 0 en 1.  
Gehuld- en grafische opdrachten zijn niet opgenomen: die zijn te ingewikkeld en te verschillend per computer.

### OVEREENKOMSTIGE INSTRUCTIES

**ABS(exp)** geeft de absolute waarde van de meegegeven expressie  
**COS(exp)** geeft de cosinus van de meegegeven expressie  
**END** einde van het programma.  
Opm.: niet beschikbaar op de Sinclair QL. Op de Enterprise moet deze opdracht soms parameters hebben.  
**FOR var = exp TO exp** begin van een lus die loopt van de eerste expressie tot de tweede expressie, waarbij de stapgrootte (STEP), indien opgegeven, gelijk is aan de derde expressie, anders gelijk aan 1.

**LEN (string)** geeft de lengte van de meegegeven string.  
Opm.: spatie is verplicht op de Memotech.  
**LET var=exp** toekenning van de expressie aan de opgegeven variabele.  
Opm.: op veel machines mag LET weggelaten worden wat echter deze opdracht staat is commentaar en de computer zelf dat niet als opdrachten  
**SIN(exp)** geeft de sinus van de meegegeven expressie.  
**SQR(exp)** bepaalt de wortel van de expressie, die niet negatief mag zijn.  
**STOP** stopt de uitvoering van het programma, het programma kan weer verder draaien met behoud van de variabelen.

## STANDAARD MICROSOFT

### BASICODE

### SCHNEIDER 464, 664 & 6128

### APPLE II

### ATARI XL-SERIE

### ELECTRON/BBC/MASTER

### COMMODORE 128/+4/C-16

### VIC 20/COMMODORE 64

### ENTERPRISE

### IBM PC — BASIC A/GW-BASIC

### MEMOTECH

### MSX1/MSX2

### DRAGON/TRS-80 level 2

### SHARP MZ-700 (cassette Basic)

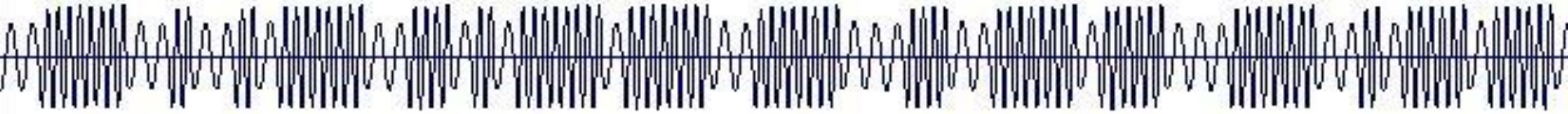
### ATARI ST-SERIE

### SINCLAIR QL

### P2000T

ASC	ATN	AUTO	CALL	CHAIN	CHR\$	CLEAR	CLOSE	CONT	DATA	DEF	DELETE	DIM	EDIT
Geeft ASCII waarde van de eerste teken van de string	Arctangens van de expressie		roep een assemblaantal subroutine aan	roep een nieuw programma op en geeft alle variabelen door	Geeft een string van het teken dat de ASCII-code van 'expressie' heeft	wist alle (of de geselecteerde) variabelen	sluit de disk bestanden of andere tekenen af alle bestanden	doorgaan met uitvoeren van het programma	lijst van data op in een READ-opdracht te gebruiken	definieert een rekenkundige string functie	haalt de geselecteerde programma regels weg	reserveert geheugenruimte voor arrays en specificiert maximale sub-subscript	maakt werkelementen op en op gelijk
ASC(string)	ATN(exp)	AUTO(regelnr, waarde)	CALL var[,var,...]	CHAIN "bestandsnaam"	CHR\$(exp)	CLEAR [exp,exp,...]	CLOSE	CONT	DATA const[,const,...]	DEF FNvar [param,param,...] = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	EDIT
ASC(string)	ATN(exp)	(m)		(m)	CHR\$(exp)		GO SUB 900		DATA const[,const,...]		(m)	DIM variab[,var[sub],...]	(m)
ASC(string)	ATN(exp)	AUTO [regelnr, toename]	CALL adr [,param,param,...]	CHAIN "bestandsnaam" [regelnr,exp]	CHR\$(exp)	CLEAR[all]	CLOSE IN "bestandsnaam" OUT Opm.: cassette moet open zijn	CONT	DATA const[,const,...]	DEF FNvar [param,param,...] = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	EDIT
ASC(string)	ATN(exp)		CALL adr [var, var,...]	CHAIN "bestandsnaam"	CHR\$(exp)	CLEAR	CLOSE "bestandsnaam"	CONT	DATA const[,const,...]	DEF FNvar [param,param,...] = exp	DEL [regelnr,regelnr]	DIM variab[,var[sub],...]	tekst met cursor
ASC(string)	ATN(exp)		USR[adr]var, var,...	RUN "bestandsnaam"	CHR\$(exp)	CLR	CLOSE "bestandsnaam" var, var,... "bestandsnaam"	CONT	DATA const[,const,...]	DEF FN [var,param,param,...] = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	tekst met cursor
ASC(string)	ATN(exp)	AUTO [regelnr, toename]	CALL adr [var, var,...]	CHAIN "bestandsnaam"	CHR\$(exp)	CLEAR	CLOSE # bestandsnaam Opm.: CLOSE #0 alle bestanden	CONT	DATA const[,const,...]	DEF FN naam [var] = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	tekst met cursor
ASC(string)	ATN(exp)	AUTO [toename]	CALL adr [,param,param,...]	geen CHAIN	CHR\$(exp)	CLR	CLOSE "bestandsnaam" [bestandsnaam]	CONT	DATA const[,const,...]	DEF FNvar = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	tekst met cursor
ASC(string)	ATN(exp)		SYSDR [adr,exp]		CHR\$(exp)	CLR	CLOSE "bestandsnaam" [bestandsnaam]	CONT	DATA const[,const,...]	DEF FNvar = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	tekst met cursor
ORD(exp)	ATN(exp)	AUTO [AT regelnr, toename]	CALL USR (naam,param,...)	CHAIN "bestandsnaam" [param,param,...]	CHR\$(exp)		CLOSE "kanaal"	CONTINUE	DATA const[,const,...]	DEF FNparam [param,param,...] = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	tekst met cursor
ASC(string)	ATN(exp)	AUTO [regelnr, toename]	CALL adr [,var,...]	CHAIN bestandsnaam	CHR\$(exp)	CLEAR	CLOSE [bestandsnaam]	CONT	DATA const[,const,...]	DEF FNvar [param,param,...] = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	tekst met cursor
ASC(string)	ATN(exp)	AUTO [regelnr, toename]	USR[adr]		CHR\$(exp)	CLEAR	DISC CLOSE # poortnr	CONT	DATA const[,const,...]	DEF FNvar [param,param,...] = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	tekst met cursor
ASC(string)	ATN(exp)	AUTO [regelnr, toename]	USR[adr]	RUN "top" [bestnr] of LOAD [toemr]	CHR\$(exp)	CLEAR	CLOSE #bestandsnaam	CONT	DATA const[,const,...]	DEF FNvar [param,param,...] = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	tekst met cursor
ASC(string)	ATN(exp)		EXEC adr	RUN "top" [bestnr] of LOAD [top] [bestnr]	CHR\$(exp)	CLEAR [exp]		CONT	DATA const[,const,...]	DEF FNvar [param,param,...] = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	tekst met cursor
ASC(string)	ATN(exp)	AUTO	USR [adr]	[alleen beschikbaar in Disk-Basic]	CHR\$(exp)	CLR		CONT	DATA const[,const,...]	DEF FNvar [param,param,...] = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	tekst met cursor
ASC(string)	ATN(exp)	AUTO [regelnr, toename]	CALL naam [var, var,...]	CHAIN [MERGE] bestandsnaam	CHR\$(exp)	CLEAR	CLOSE [bestandsnaam] [bestandsnaam]	CONT	DATA const[,const,...]	DEF FNvar [param,param,...] = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	tekst met cursor
CODE(string)	ATN(exp)	AUTO [regelnr, toename]	CALL adr [param]	MRUN eenh "bestandsnaam"	CHR\$(exp)	CLEAR	CLOSE # poortnr	CONTINUE	DATA const[,const,...]	DEF FNvar [param,param,...] = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	tekst met cursor
ASC(string)	ATN(exp)	AUTO [regelnr, toename]	USR exp	RUN "bestandsnaam"	CHR\$(exp)	CLEAR [exp,exp]		CONT	DATA const[,const,...]	DEF FNvar [param,param,...] = exp	DELETE [regelnr,regelnr]	DIM variab[,var[sub],...]	tekst met cursor

# 1981: transmitting programs by radio



# 1983: an API based on GOSUBs

```
1000 A=100: GOTO 20: REM *** BASICODE DEMO ***
1010 FOR I = 0 TO 100
1020 GOSUB 100
1030 PRINT "FOSDEM ";2020+I
1040 NEXT I
1050 GOTO 1010
1060 END
```

1000: first line of the BASICODE-2 program. It must be in the following form: 1000 A=(value): GOTO 20: REM program name. (value) is the maximum number of characters that can be used by all strings together. Line 20 is used to reserve memory space for the strings in those computers which need it.

1010-32767: the main program. There are no restrictions on this section, except that line numbers above 32767 are forbidden.

## 3. Operation of the BASICODE-2 standard routines

These are written specifically for each computer brand and form part of the translation program. They look different in the various BASIC dialects but in fact they do exactly the same in each case.

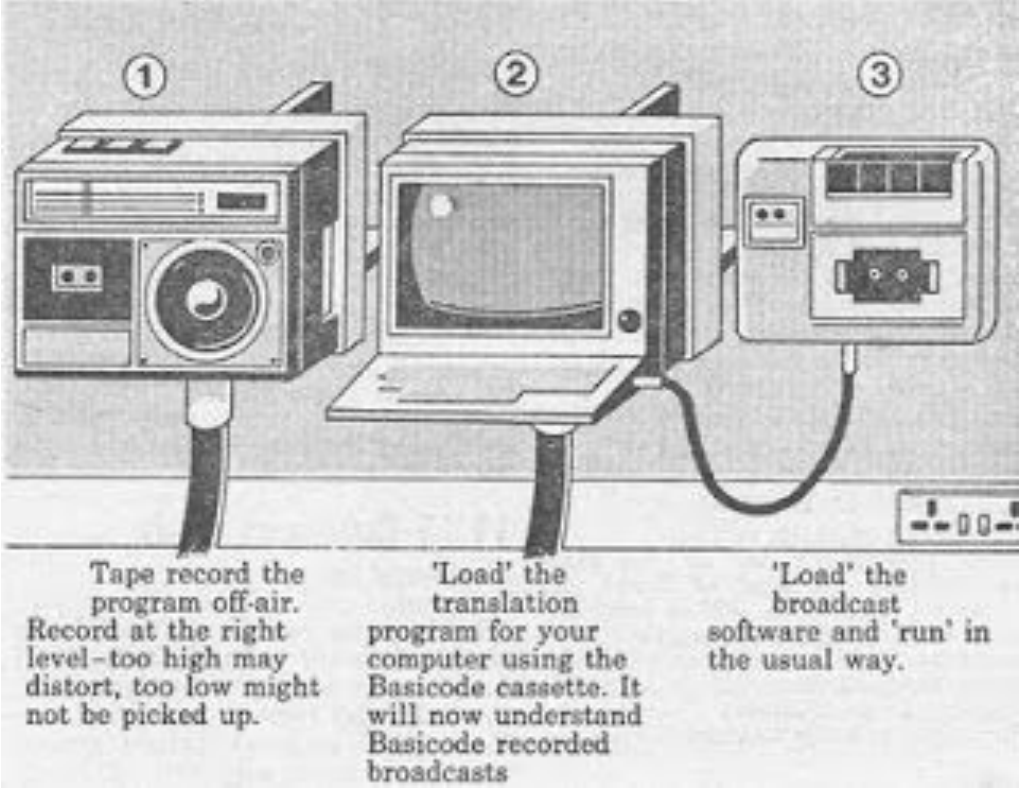
### GOSUB 100:

This subroutine is used to clear the screen and set the cursor at position 0,0 (i.e. the top left-hand corner of the screen.)

### GOSUB 110:

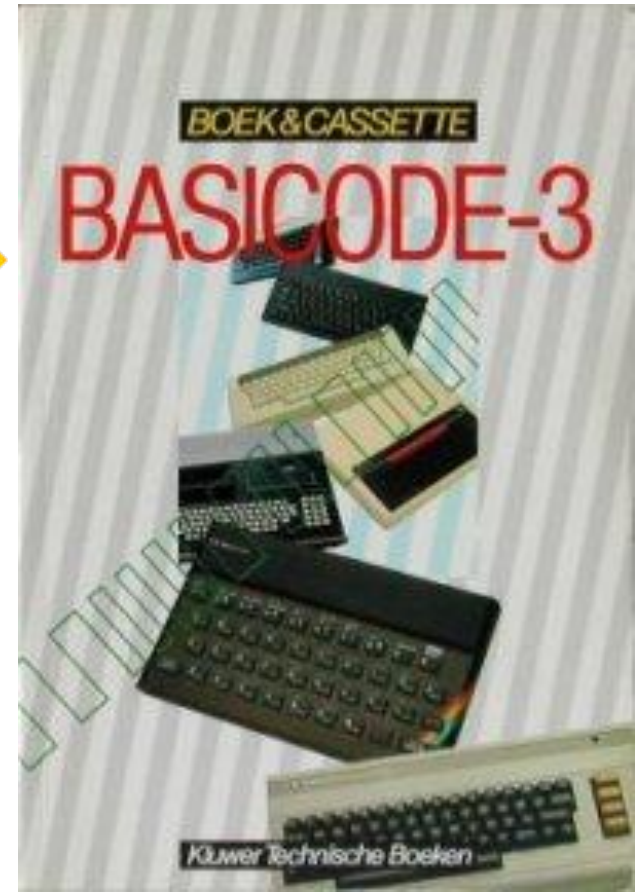
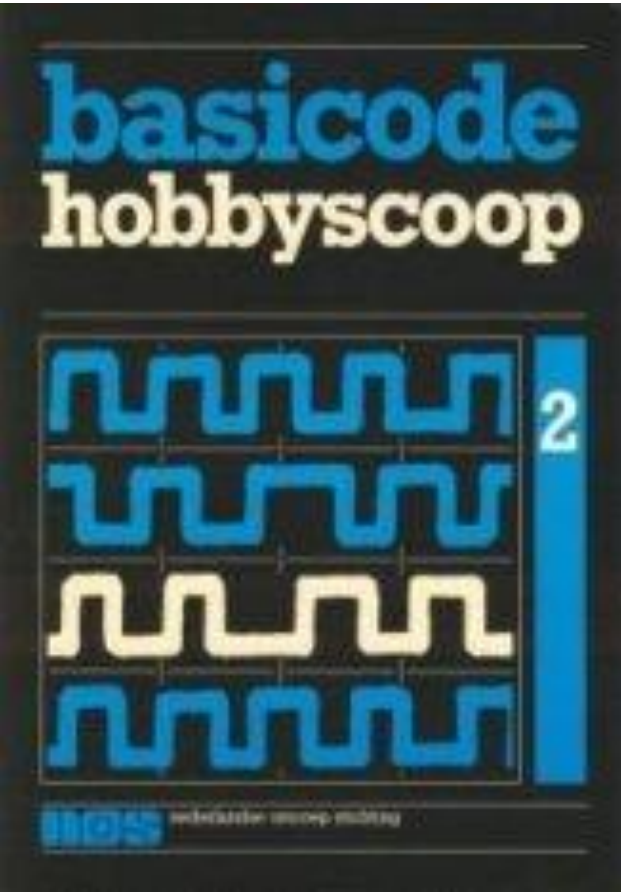
This places the cursor at a specific point on the screen. The exact point can



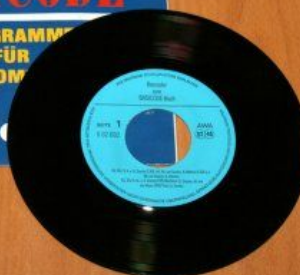




# 1985: licence wars and a fork



# Early 1989: across the Wall



**robotron<sup>®</sup>**



# BASICODE in JavaScript

```
<html>
```

```
<head>
```

```
<script src="basicode.js"></script>
```

```
</head>
```

```
<body>
```

```
<script type="text/basicode">
```

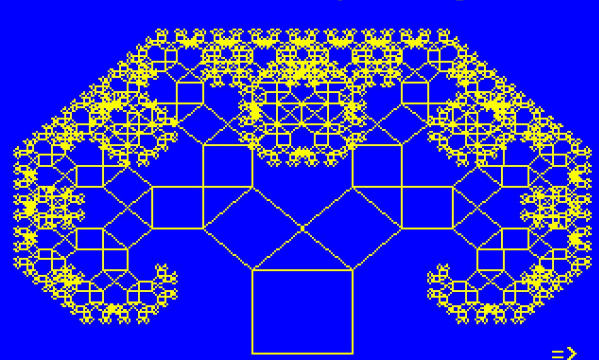
```
1010 PRINT "Hello, world!"
```

```
</script>
```

```
</body>
```

```
</html>
```

De BOOM van PYTHAGORAS, 10 diep



=>

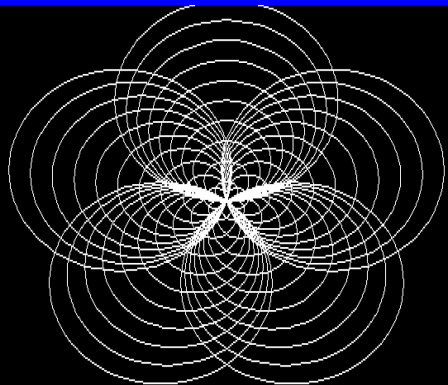
РУССКИЙ АЛФАБИТ

АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ

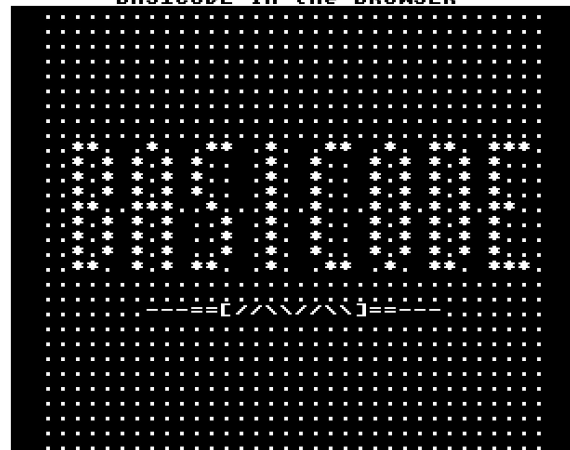
VERTAAL IN HET NEDERLANDS:

ЗДРАВСТВУЙТЕ

?



BASICODE in the BROWSER



(c) 2016, 2017 Rob Hagemans

Welcome!

Welcome to the growing group of  
Basicode users. On this side of the  
cassette are a number of example  
programs

can try to  
then run  
Taken number  
a



DAGEN

maand 2  
dag 1  
uur 16

druk toets =>



# Thank you!

## History & sources

Janny Looyenga, Frank Veraart, Jan Bredenbeek, André van der Leeden, Thomas Rademacher, Victor Reijs  
and many others

## References

- Hans G. Janssen (ed.), *BASICODE Hobbyscoop 2*, Nederlandse Omroep Stichting, Hilversum, 1983.  
Trevor Taylor (ed.), *The Chip Shop BASICODE-2*, Broadcasting Support Services, London, 1984.  
Jacques Haubrich (ed.), *Het BASICODE-3 Boek*, Kluwer Technische Boeken, Deventer, 1986.  
Michael Wiegand, Heike & Manfred Fillinger, *BASICODE. Mit Programmkassette*, Ravensburg, 1986.  
'basicode-2', *Elektuur* **10** 1983.  
Prof. Dr. Horst Völz, 'Datenaustausch mit BASICODE', *Radio Fernsehen Elektronik* **1** 1990.  
Frank Veraart, 'Basicode: Co-Producing a Microcomputer Esperanto', *History of Technology* **28** 2008

links & materials at

**<https://github.com/robhagemans/basicode>**