

**LESEPROBE**

Holger  
Weßling

# FOR BASIC TO ONE-LINERS



# Inhalt

|                                                  |           |                                                    |           |
|--------------------------------------------------|-----------|----------------------------------------------------|-----------|
| <b>1..... Vorwort.....</b>                       | <b>8</b>  | 4.6..... Dateiformate.....                         | 25        |
| 1.1..... Holger Weßling (Harvey).....            | 8         | 4.6.1..... PRG.....                                | 25        |
| 1.2..... Martin Roscher.....                     | 8         | 4.6.2..... P00.....                                | 25        |
| 1.3..... Robin Harbron.....                      | 8         | 4.6.3..... D64.....                                | 26        |
| 1.4..... Martin Gutenbrunner.....                | 8         | 4.7..... Systemgrundlage.....                      | 26        |
|                                                  |           | 4.8..... 50KB Basic Speicher.....                  | 26        |
| <b>2..... Einleitung.....</b>                    | <b>9</b>  | <b>5..... BASIC Abkürzungen &amp; Token.....</b>   | <b>26</b> |
| 2.1..... BASIC-Persönlichkeiten.....             | 9         | 5.1..... BASIC.....                                | 26        |
| 2.2..... Das erste Basic-Programm.....           | 10        | 5.2..... Varianten von Commodore BASIC.....        | 27        |
| 2.3..... One-Liner-Geschichte.....               | 10        | 5.2.1..... Simons' BASIC.....                      | 27        |
| 2.4..... Was ist ein Einzeiler?.....             | 11        | 5.2.2..... TSB (Tuned Simons' BASIC).....          | 27        |
| 2.4.1..... Echte / natürliche Einzeiler.....     | 11        | 5.2.3..... Vision BASIC.....                       | 27        |
| 2.5..... Erweiterte Einzeiler.....               | 12        | 5.2.4..... XC=BASIC.....                           | 27        |
| 2.5.1..... Der 80-Zeichenmodus.....              | 12        | 5.3..... Andere Systeme mit BASIC.....             | 28        |
| des Commodore 128.....                           | 12        | 5.4..... Optimierer, Compiler und Interpreter..... | 28        |
| 2.6..... Assembler in BASIC.....                 | 12        | 5.4.1..... Optimierer.....                         | 28        |
| 2.6.1..... In Speicher schreiben mit PRINT.....  | 13        | 5.4.2..... Compiler.....                           | 29        |
| 2.6.2..... Grafik-Code als Maschinensprache..... | 14        | 5.4.3..... Interpreter.....                        | 29        |
| in BASIC.....                                    | 14        | 5.5..... Nützliche Cross-Development-Tools.....    | 29        |
| 2.6.3..... Cross-Assembler.....                  | 15        | für BASIC.....                                     | 29        |
|                                                  |           | 5.6..... Was in BASIC möglich ist.....             | 30        |
| <b>3..... BASIC Basics.....</b>                  | <b>16</b> | 5.7..... BASIC PLUS/4 Emulator.....                | 30        |
| 3.1..... Mathematische Prioritäten.....          | 16        | <b>6..... Zeitmessung.....</b>                     | <b>31</b> |
| 3.2..... Rundung.....                            | 17        | 6.1..... Wie generiert der C64 seine Zeit?.....    | 31        |
| 3.3..... Variablen.....                          | 17        | 6.2..... Wie kann man die Zeit auslesen.....       | 31        |
| 3.4..... Systemvariablen.....                    | 17        | und nutzen?.....                                   | 31        |
| 3.5..... Konstanten.....                         | 18        | 6.3..... Was sind Jiffys?.....                     | 31        |
| 3.5.1..... DATA.....                             | 19        | <b>7..... Einsteigertipps.....</b>                 | <b>32</b> |
| 3.5.2..... Mathematische Funktionen.....         | 19        | 7.1..... Übung 1.....                              | 32        |
| 3.6..... Operatoren.....                         | 19        | 7.1.1..... Das Ausprobieren.....                   | 33        |
| 3.7..... (PEEK(A)ANDB)ORC.....                   | 19        | 7.1.2..... Das Optimieren.....                     | 33        |
| 3.8..... Das GOTO-Problem.....                   | 20        | 7.1.3..... Mehr Features?.....                     | 34        |
|                                                  |           | 7.1.4..... Mehr Features!.....                     | 36        |
| <b>4..... BASIC Speicher.....</b>                | <b>21</b> | 7.2..... Übung 2.....                              | 37        |
| 4.1..... Vergleichskriterien.....                | 21        | 7.3..... Übung 3.....                              | 38        |
| 4.2..... Speicheraufbau BASIC.....               | 21        | 7.4..... Wie man Code verdichtet:.....             | 39        |
| 4.2.1..... Einfaches Beispiel.....               | 23        | Alle Fehlermeldungen.....                          | 39        |
| 4.3..... Beispiel CBM prg Studio.....            | 24        | 7.5..... Teste deine Alternativen.....             | 43        |
| 4.4..... Speicherermittlung mit FRE().....       | 24        |                                                    |           |
| 4.5..... Extra-Speicherbedarf für Variablen..... | 24        |                                                    |           |

## 8. Zeichen sparen..... 44

|      |                           |    |
|------|---------------------------|----|
| 8.1  | LET A                     | 44 |
| 8.2  | Leerzeichen               | 45 |
| 8.3  | AA=0:NAME\$=""            | 45 |
| 8.4  | A=0:A\$=""                | 46 |
| 8.5  | GOTO 0                    | 46 |
| 8.6  | A=2.0                     | 46 |
| 8.7  | 0.05 = .05                | 46 |
| 8.8  | A% VS. INT(A)             | 47 |
| 8.9  | INT(XAND1)                | 47 |
| 8.10 | INT(X.X*Y)                | 48 |
| 8.11 | DATA 1                    | 48 |
| 8.12 | DATA 2                    | 48 |
| 8.13 | DATA 3                    | 49 |
| 8.14 | DATA X,Y,Z                | 49 |
| 8.15 | Klammern                  | 50 |
| 8.16 | (NOTA)ANDB                | 50 |
| 8.17 | BOOLE vs. IF              | 50 |
| 8.18 | A=(X)AND(Y)               | 50 |
| 8.19 | PRINT;CHR\$(X); MID\$(.)  | 51 |
| 8.20 | PRINT"TEXT"               | 51 |
| 8.21 | A\$=""                    | 51 |
| 8.22 | Strings                   | 52 |
| 8.23 | IF A\$<>"                 | 52 |
| 8.24 | FORA=1TO100               | 52 |
| 8.25 | FOR TO NEXT STEP-         | 52 |
| 8.26 | NEXT A:NEXT B             | 53 |
| 8.27 | Festwerte                 | 53 |
| 8.28 | Potenzen                  | 54 |
| 8.29 | 10er Exponenten           | 55 |
| 8.30 | Potenz statt SQR          | 55 |
| 8.31 | $\pi$ als Multiplikator   | 55 |
| 8.32 | RND mal anders: 0 bis 255 | 56 |
| 8.33 | RND mal anders: 0 bis 20  | 57 |
| 8.34 | RND mal anders: 0 bis -1  | 58 |
| 8.35 | Mehrfache RND             | 58 |
| 8.36 | INT(RND(1))               | 59 |
| 8.37 | WAIT0,1                   | 59 |
| 8.38 | WAIT0,0                   | 59 |
| 8.39 | GET & WAIT                | 60 |
| 8.40 | IF THEN / GOTO            | 60 |
| 8.41 | IF<X>THEN                 | 60 |
| 8.42 | IF(X)THEN 1               | 61 |
| 8.43 | IF(X)THEN 2               | 61 |
| 8.44 | Mehrfaches IF THEN        | 61 |
| 8.45 | IF X<>0 THEN              | 62 |

|      |                             |    |
|------|-----------------------------|----|
| 8.46 | IF X>0ANDX<100 THEN         | 62 |
| 8.47 | IF X=1THENX=5:IF X=5THENX=1 | 62 |
| 8.48 | A<=X                        | 63 |
| 8.49 | DIM                         | 63 |
| 8.50 | Bildschirmfarben            | 63 |
| 8.51 | Cursorfarbe setzen          | 63 |

## 9. Nicht-Optimierungen..... 64

|     |                   |    |
|-----|-------------------|----|
| 9.1 | A<B*2 vs. (A<B)*2 | 64 |
| 9.2 | A=B=1             | 64 |

## 10. Videokurse-Optimierung..... 64

|      |            |    |
|------|------------|----|
| 10.1 | Videokurse | 64 |
|------|------------|----|

## 11. Tipps und Tricks ..... 65

|         |                                              |    |
|---------|----------------------------------------------|----|
| 11.1    | Bildschirm löschen 1                         | 65 |
| 11.2    | Bildschirm löschen 2                         | 65 |
| 11.3    | Bildschirmfarben 1                           | 65 |
| 11.4    | Bildschirmfarben 2                           | 66 |
| 11.5    | Cursor oben links positionieren              | 67 |
| 11.6    | Cursor irgendwo positionieren                | 67 |
| 11.7    | Cursor positionieren<br>ohne SYS             | 68 |
| 11.8    | Cursor positionieren<br>ohne Zeilenumbruch   | 68 |
| 11.9    | A=0:A\$="" vs. CLR                           | 69 |
| 11.10   | GOTO x                                       | 69 |
| 11.11   | GOTO 10,20,30                                | 69 |
| 11.12   | Gerade oder ungerade Zahl                    | 70 |
| 11.13   | IF THEN ELSE                                 | 70 |
| 11.14   | IF ohne IF                                   | 71 |
| 11.15   | REPEAT UNTIL DO WHILE                        | 71 |
| 11.16   | Input mit allen Sonderzeichen                | 72 |
| 11.17   | IF A\$="X"THEN                               | 72 |
| 11.18   | MID\$ statt RIGHT\$                          | 73 |
| 11.19   | RND mal anders: 0 oder 1                     | 73 |
| 11.19.1 | RND mal anders: 0 bis 255 per Timer          | 73 |
| 11.19.2 | RND mal anders: 0 bis 255 per SID            | 74 |
| 11.19.3 | RND mal anders: 0 oder 1 per AND             | 74 |
| 11.19.4 | RND mal anders: 0 oder 1 kurz<br>und schnell | 74 |

|         |                                                    |    |
|---------|----------------------------------------------------|----|
| 11.19.5 | RND mal anders: 0 oder 1<br>(kürzer und schneller) | 75 |
| 11.19.6 | RND 1 oder 2                                       | 75 |
| 11.19.7 | RND von X%                                         | 75 |
| 11.19.8 | RND mehrfach                                       | 76 |
| 11.19.9 | RND STEP X                                         | 76 |
| 11.20   | PRINT TAB                                          | 77 |
| 11.21   | Horizontale Rahmen beschneiden                     | 77 |
| 11.22   | WAIT vs. TI vs. FOR NEXT                           | 78 |
| 11.23   | WAIT ohne STOP                                     | 78 |
| 11.24   | WAIT mit Fortsetzung                               | 79 |
| 11.25   | Modulo                                             | 79 |
| 11.26   | Gerade vs. ungerade Zahlen                         | 79 |
| 11.27   | SGN & INT                                          | 80 |
| 11.28   | INT(-X.X)                                          | 80 |
| 11.29   | Auf zwei Stellen runden                            | 80 |
| 11.30   | TOGGLE 1/0, -1/0, -1/1                             | 80 |
| 11.31   | FOR NEXT = +1                                      | 81 |
| 11.31.1 | FOR NEXT STEP-                                     | 82 |
| 11.31.2 | FOR NEXT Unendlich 0 > X                           | 82 |
| 11.31.3 | FOR NEXT Unendlich -X > -1/0                       | 83 |
| 11.31.4 | FOR NEXT Unendlich -X > 0 > X                      | 84 |
| 11.31.5 | FOR NEXT Unendlich -X > 0 > X < 0 < -X             | 85 |
| 11.31.6 | FOR NEXT Unendlich X > Y > X > Y > X               | 85 |
| 11.32   | Infinite Schleifen ohne GOTO                       | 86 |
| 11.33   | XOR/EOR/EXOR Exklusiv-Oder                         | 86 |
| 11.34   | LIST im Programm                                   | 86 |
| 11.35   | LOAD "*" ,8                                        | 86 |
| 11.36   | Save & Replace                                     | 87 |
| 11.37   | ASCII > Bildschirmcode                             | 87 |
| 11.38   | Bildschirmcode > ASCII                             | 87 |
| 11.39   | Zeile löschen                                      | 88 |
| 11.40   | Löschen von Bildschirmzeilen                       | 88 |
| 11.41   | Kopieren von Bildschirmzeilen                      | 88 |
| 11.42   | Bildschirm-Textfarbe ändern                        | 88 |
| 11.43   | Kleinschrift per Steuerzeichen                     | 89 |
| 11.44   | PRINT ohne Umbruch                                 | 89 |
| 11.45   | Joystickabfrage 1                                  | 90 |
| 11.46   | Joystickabfrage 2                                  | 90 |
| 11.47   | Passwortschutz                                     | 91 |
| 11.48   | Langsame Bildschirmausgabe                         | 91 |
| 11.49   | 16 Funktionstasten abfragen                        | 91 |
| 11.50   | Weiter mit Feuerknopf                              | 91 |
| 11.51   | BASIC-Programm UNDO nach<br>NEW oder RESET         | 91 |
| 11.52   | Mehr DATAs in einer Zeile                          | 92 |
| 11.53   | RESTORE → RUN                                      | 93 |

|       |                                    |    |
|-------|------------------------------------|----|
| 11.54 | Modifizierer und ausführender Code | 93 |
| 11.55 | Rasterinterrupt                    | 94 |
| 11.56 | VICE doppelt fix                   | 95 |
| 11.57 | Selten genutzte Symbole            | 95 |

## 12 ..... Speed ..... 96

|         |                                                       |     |
|---------|-------------------------------------------------------|-----|
| 12.1    | Kurze Zeilennummern                                   | 98  |
| 12.2    | Schnelleres BASIC: Bildschirm-<br>ausgabe ausschalten | 98  |
| 12.2.1  | Schnelleres BASIC: Interrupt 1                        | 98  |
| 12.2.2  | Schnelleres BASIC: Interrupt 2                        | 98  |
| 12.2.3  | Schnelleres BASIC: Timer A                            | 99  |
| 12.2.4  | Schnelleres BASIC: ROM > RAM                          | 99  |
| 12.3    | DIMA%(1000)                                           | 100 |
| 12.4    | Variablen wiederverwenden                             | 100 |
| 12.5    | A=0 vs. A=                                            | 101 |
| 12.6    | NEXT A                                                | 101 |
| 12.7    | NEXT A,B                                              | 101 |
| 12.8    | A vs. AA vs. AAA                                      | 101 |
| 12.9    | 10er Exponenten                                       | 102 |
| 12.10   | A=1:B=1                                               | 102 |
| 12.11   | A+1                                                   | 102 |
| 12.12   | π vs. 3                                               | 103 |
| 12.13   | Besser multiplizieren 1                               | 103 |
| 12.14   | Besser multiplizieren 2                               | 103 |
| 12.15   | Multiplizieren statt Potenzieren                      | 104 |
| 12.16   | Addieren statt Multiplizieren                         | 104 |
| 12.17   | Arithmetische Operationen: Schleifen                  | 104 |
| 12.18   | Gemeinsame Basis                                      | 105 |
| 12.19   | Variablen und Festwerte                               | 105 |
| 12.19.1 | Variablen und Festwerte 1                             | 106 |
| 12.19.2 | Variablen und Festwerte 2                             | 106 |
| 12.19.3 | Variablen und Festwerte 3                             | 107 |
| 12.20   | Vaiablen und Array-Variablen                          | 107 |
| 12.21   | Print und Variablen                                   | 107 |
| 12.22   | Steuer-/Zeichen vs. CHR\$                             | 108 |
| 12.23   | MID\$ vs. PRINT                                       | 108 |
| 12.24   | RND schneller machen 1                                | 108 |
| 12.25   | RND schneller machen 2                                | 109 |
| 12.26   | Boolesche Operationen                                 | 109 |
| 12.27   | A=1/10,B=.1                                           | 109 |
| 12.28   | FOR NEXT vs. IF THEN GOTO                             | 110 |
| 12.29   | Mehrfach AND                                          | 110 |
| 12.30   | IF AND THEN                                           | 110 |
| 12.31   | IF THEN <> IF GOTO                                    | 111 |

|       |                           |     |
|-------|---------------------------|-----|
| 12.32 | IF THEN <> ON GOTO        | 111 |
| 12.33 | Zahlen vordefinieren      | 111 |
| 12.34 | A\$="A" vs. LEN(A\$)      | 112 |
| 12.35 | Exponenten verwenden      | 112 |
| 12.36 | USR                       | 112 |
| 12.37 | TIME/TI/TI\$ zurücksetzen | 113 |

### 13. Was es zu vermeiden gilt 113

|      |                        |     |
|------|------------------------|-----|
| 13.1 | ON(A=X)GOTO            | 113 |
| 13.2 | A=ASC(A\$) / A=ASC("") | 114 |
| 13.3 | A=B=0                  | 114 |
| 13.4 | INT = Falsch           | 114 |

### 14. One-Liner 114

|          |                                      |     |
|----------|--------------------------------------|-----|
| 14.1     | One-Liner-Bedingungen                | 114 |
| 14.2     | FROG                                 | 115 |
| 14.3     | Mandelbrot                           | 116 |
| 14.4     | Basic-Game-Compo                     | 117 |
| 14.5     | Der erste Commodore 64 Einzeiler?    | 118 |
| 14.6     | Der erste One-Liner-Wettbewerb       | 121 |
| 14.7     | One-Liner-Games                      | 121 |
| 14.7.1   | BASIC-Einzeiler Star Collider (Game) | 121 |
| 14.7.2   | Number Game                          | 122 |
| 14.8     | Blue Cross BASIC Contest             | 122 |
| 14.9     | Commodore FREE                       |     |
|          | One Liners competition               | 125 |
| 14.10    | First International GOTO-less        |     |
|          | Basic Coding Compo                   | 125 |
| 14.11    | One-Liner zum Abtippen,              |     |
|          | Anschauen und Lernen                 | 125 |
| 14.11.1  | Alle Befehle                         | 126 |
| 14.11.2  | Aus der Zeitschrift The Transactor 4 | 126 |
| 14.11.3  | 10PRINT with custom characters       | 127 |
| 14.11.4  | Weird colourful patterns             | 127 |
| 14.11.5  | Colorful Pattern                     | 128 |
| 14.11.6  | Eagle                                | 128 |
| 14.11.7  | Bird can fly                         | 128 |
| 14.11.8  | Laser C64                            | 128 |
| 14.11.9  | Laser                                | 129 |
| 14.11.10 | SID & MAT64                          | 129 |
| 14.11.11 | Screensaver 1                        | 130 |
| 14.11.12 | Screensaver 2                        | 130 |
| 14.11.13 | Colored scrolling stars              | 130 |

|           |                                  |     |
|-----------|----------------------------------|-----|
| 14.11.14  | Sinful                           | 131 |
| 14.11.15  | Random Walk                      | 131 |
| 14.12     | BASIC-Demos                      | 132 |
| 14.12.1   | 512/256/128/64/32/16/8           |     |
|           | Bytes Demos                      | 132 |
| 14.12.1.1 | LoveByte                         | 132 |
| 14.12.1.2 | 256Byte Games                    | 132 |
| 14.13     | TRS-80 One-Liners                | 132 |
| 14.14     | Programmierwettbewerbe Einzeiler | 133 |

### 15. Kuriositätsensammlung 135

|      |                                 |     |
|------|---------------------------------|-----|
| 15.1 | Rechnen mit PEEK                | 135 |
| 15.2 | TI als Zähler, unendliches NEXT | 135 |
| 15.3 | ON TI GOTO                      | 136 |
| 15.4 | ON ERROR-GOTO                   | 136 |
| 15.5 | Randomized Text                 | 136 |
| 15.6 | TRUE=FALSE                      | 136 |
| 15.7 | Zeile 350800                    | 137 |
| 15.8 | Merkwürdiges Verhalten          | 137 |
| 15.9 | Spaß mit TI\$                   | 137 |

### 16. PEEKs, POKEs, SYSs, WAITs & CHR\$s 140

|      |        |     |
|------|--------|-----|
| 16.1 | PEEKs  | 140 |
| 16.2 | POKEs  | 143 |
| 16.3 | SYSs   | 146 |
| 16.4 | WAITs  | 148 |
| 16.5 | CHR\$s | 149 |

### 17. Weblinks 150

|      |             |     |
|------|-------------|-----|
| 17.1 | Magazine    | 150 |
| 17.2 | Listings    | 150 |
| 17.3 | BASIC-Tools | 150 |

### 18. Danksagungen & Quellennachweise 151

### 19. Bildschirmcodes des C64 152

### 20. PETSCII-Code-Tabelle des C64 154

### 21. BASIC-Dialekte Befehlstabelle 156

## Zeichen sparen

Weil auch Einsteiger dieses Buch lesen werden, gilt auch hier, dass alles zusammengetragen wurde, was Zeichen einspart – von den allereinfachsten und offensichtlichen Möglichkeiten bis zu komplexen Wegen, sodass Anfänger wie auch Fortgeschrittene einen Nutzen davon haben werden.

### INFO: Tipps-Aufteilung

Manche Tipps passen in dieses Kapitel, aber auch zu den *Tipps & Tricks* oder zum Abschnitt *Speed*. Wir haben die Tipps so themengerecht wie möglich eingeordnet und hoffen, dass Überschneidungen oder die Platzierung mancher Inhalte in einem anderen Kapitel für dich kein Störfaktor sind.

### 8.1

#### LET A

**LET** wurde unter anderem als *Befehl* eingeführt, um ein Listing übersichtlich zu machen und auch um die Kompatibilität zu anderen BASIC-Dialekten zu gewährleisten. Es kann im Commodore BASIC ersatzlos gestrichen werden.

 **10 FORI=0TO100:LETA=A+1:NEXT**

So lässt sich die Variablenzuweisung etwas besser lesen als mit:

 **10 FORI=0TO100:A=A+1:NEXT**

Die Zeitersparnis mag gering erscheinen, doch in diesem Fall zählt jedes bisschen, da sich die Vorteile über die Zeit aufsummieren.

 **10 FORB=1TO1000:LETA=1:NEXT**  
 = 23 Zeichen (19 gekürzt) [Benchmark = 173]

 **10 FORB=1TO1000:A=1:NEXT**  
 = 18 Zeichen (15 gekürzt) [Benchmark = 170]

## 11.20

## PRINT TAB

Wenn du mit **PRINT** eine Ausgabe 10 Zeichen nach rechts versetzt darstellen willst, dann verwende das Komma anstatt **TAB(10)**.

```
10 PRINT"A";TAB(10)"B"
```

funktioniert verkürzt auch so:

```
10 PRINT"A","B"
```

Indem du pro 10 Zeichen ein zusätzliches Komma einfügst, kannst du auch mehrere Zeilen auf einmal überspringen.

```
10 PRINT"A",,,,,,,,,,"B"
READY.
RUN
A                                     B
```

Wenn du etwa nur neun Zeichen brauchst, verwende Cursor-Links im Quote-Modus vor dem nächsten Zeichen.

```
10 PRINT"A","B"
READY.
RUN
A                                     B

READY.
10 PRINT"A";TAB(9);"B"
RUN
A                                     B
```

## 11.21

## Horizontale Rahmen beschneiden

Mit **POKE53265,19** werden am oberen und unteren Rahmen eine halbe Zeile überblendet, womit Scrolleffekte verschönert werden, da das Ruckeln in den nicht sichtbaren Bereich verschwindet. Zurücksetzen lässt sich das mit **POKE53265,27**.

```
POKE53265,19  OBERSTE ZEILE
READY.

UNTERSTE ZEILE
```

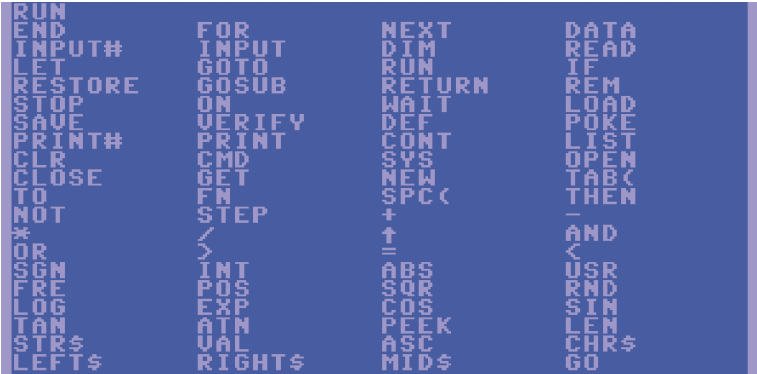
14.11.1

Alle Befehle

Mit diesem Einzeiler werden sämtliche Befehle des Commodore 64 BASIC V2 aufgelistet:



```
1 POKE769,177:FORI=1TO76:POKE73,255:POKE781,I:
  SYS42794:PRINT,:NEXT:POKE769,227
```



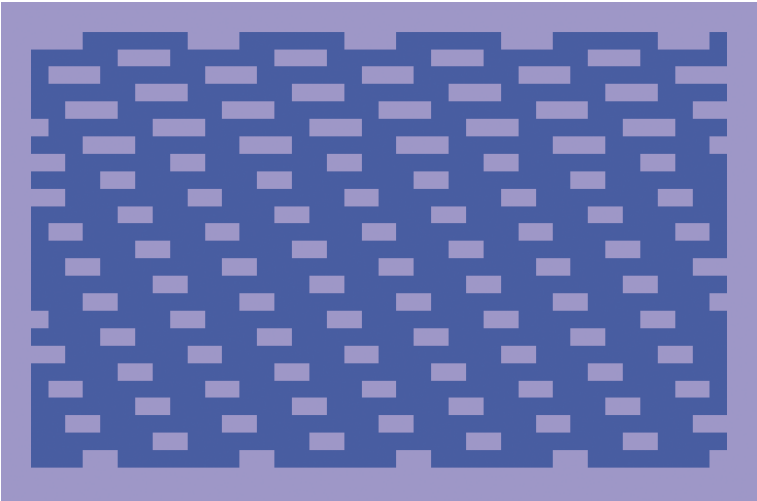
14.11.2

Aus der Zeitschrift  
The Transactor 4

Ursprünglich für den Commodore PET entwickelt, wurde der Code angepasst, damit er auf dem Commodore 64 läuft:



```
1 C=32:FORN=1TO41:C=192-C:FORA=0TON:FORB=1024+
  AT02047STEPN:POKEB,C:NEXTB,A,N
```



# PEEKs, POKEs, SYSs, WAITs & CHR\$s

## 16.1 PEEKs

| PEEK                  | Funktion                                                |
|-----------------------|---------------------------------------------------------|
| PEEK(17)              | Wert letzter Eingabeart. 0= INPUT, 64=GET, 128=READ     |
| PEEK(18)              | SGN Resultat TAN- oder SIN-Funktion                     |
| PEEK(19)              | Aktueller Ausgabekanal (CMD Logische Dateinummer)       |
| PEEK(21)+PEEK(22)*256 | Zeilennummer für GOTO, LIST,ON & GOSUB                  |
| PEEK(23)+PEEK(24)*256 | Adresse zum letzten STRING                              |
| PEEK(25)+PEEK(X)*256  | Stack für temporäre STRING (X = 26 bis 33)              |
| PEEK(34) - (37)       | Temporäre Datenstacks                                   |
| PEEK(36)/(37)         | Zeiger für indirekte Speicherung                        |
| PEEK(38)/(41)         | Fließkommaergebnis der Multiplikation (Mantisse)        |
| PEEK(43)+PEEK(44)*256 | BASIC-Programm Anfang                                   |
| PEEK(45)+PEEK(46)*256 | BASIC-Programm Ende                                     |
| PEEK(47)+PEEK(48)*256 | BASIC-Arrayspeicher Anfang                              |
| PEEK(49)+PEEK(50)*256 | BASIC-Arrayspeicher Ende                                |
| PEEK(51)+PEEK(52)*256 | BASIC-Stringspeicher Anfang                             |
| PEEK(53)+PEEK(54)*256 | BASIC-Temporäre Pointer für Strings/Vorheriger String   |
| PEEK(55)+PEEK(56)*256 | Pointer zur höchsten Adresse die von BASIC genutzt wird |
| PEEK(57)+PEEK(58)*256 | Aktuelle BASIC-Zeilenummer                              |
| PEEK(59)+PEEK(60)*256 | Vorherige BASIC-Zeilenummer (wird von CONT verwendet)   |
| PEEK(61)+PEEK(62)*256 | Aktuelle BASIC-Adresse zum nächsten Befehl              |
| PEEK(63)+PEEK(64)*256 | Aktuelle DATA Zeilennummer                              |
| PEEK(65)+PEEK(66)*256 | Zeiger zum aktuellen DATA Inhalt                        |
| PEEK(67)+PEEK(68)*256 | Vektor INPUT-Routine                                    |
| PEEK(69)+PEEK(70)*256 | Aktueller BASIC-Variablenname                           |
| PEEK(71)+PEEK(71)*256 | Zeiger zur aktuellen Variable                           |
| PEEK(73)+PEEK(74)*256 | Variablenzeiger für FOR/NEXT                            |
| PEEK(75)+PEEK(96)*256 | Zwischenspeicher für BASIC-Zeiger                       |
| PEEK(97)              | FAC-Exponent                                            |
| PEEK(98) - (101)      | FAC-Mantisse                                            |
| PEEK(102)             | FAC-Vorzeichen                                          |
| PEEK(103)             | FLAG: Polynomauswertung                                 |
| PEEK(104)             | FAC-Überlauf                                            |
| PEEK(105)             | ARG-Exponent                                            |
| PEEK(106) - (109)     | ARG-Mantisse                                            |
| PEEK(110)             | ARG-Vorzeichen                                          |
| PEEK(111)             | Ergebnis Vorzeichenvergleich FAC/ARG                    |
| PEEK(112)             | FAC-Niederwertigste Stelle (Rundung)                    |
| PEEK(113) / (114)     | Zeiger Kassettenpuffer                                  |

# BASIC-Dialekte Befehlstabelle

| Atari      | Amstrad    | Apple      | Commodore  | MSX        |
|------------|------------|------------|------------|------------|
| ABS        | ABS        | ABS        | ABS        | ABS        |
| AND        | AND        | AND        | AND        | AND        |
| ASC        | ASC        | ASC        | ASC        | ASC        |
| ATN        | ATN        | ATN        | ATN        | ATN        |
| BIN\$      | BIN\$      | BIN\$      | -          | BIN\$      |
| BORDER     | BORDER     | -          | -          | BORDER     |
| CHR\$      | CHR\$      | CHR\$      | CHR\$      | CHR\$      |
| CLOSE      | CLOSE      | CLOSE      | CLOSE      | CLOSE      |
| CLS        | CLS        | HOME       | CLR        | SCREEN     |
| COLOR      | INK        | COLOR      | -          | COLOR      |
| CONT       | CONT       | CONT       | CONT       | CONT       |
| COS        | COS        | COS        | COS        | COS        |
| DATA       | DATA       | DATA       | DATA       | DATA       |
| DATE       | DATE\$     | -          | -          | -          |
| DEF        | DEF        | DEF        | DEF        | DEF        |
| DIM        | DIM        | DIM        | DIM        | DIM        |
| DRAW       | DRAW       | -          | -          | DRAW       |
| END        | END        | END        | END        | END        |
| EXP        | EXP        | EXP        | EXP        | EXP        |
| FN         | FN         | FN         | FN         | FN         |
| FOR...NEXT | FOR...NEXT | FOR...NEXT | FOR...NEXT | FOR...NEXT |
| FRE        | FRE        | FRE        | FRE        | FRE        |
| -          | -          | -          | GET#       | -          |
| GOSUB      | GOSUB      | GOSUB      | GOSUB      | GOSUB      |
| GOTO       | GOTO       | GOTO       | GOTO       | GOTO       |
| HEX\$      | HEX\$      | HEX\$      | -          | HEX\$      |
| IF...THEN  | IF...THEN  | IF...THEN  | IF...THEN  | IF...THEN  |
| INKEY\$    | INKEY\$    | -          | -          | -          |
| INPUT      | INPUT      | INPUT      | INPUT      | INPUT      |
| -          | -          | -          | INPUT#     | -          |
| INT        | INT        | INT        | INT        | INT        |
| JOY        | JOY        | -          | -          | -          |
| LEFT\$     | LEFT\$     | LEFT\$     | LEFT\$     | LEFT\$     |
| LEN        | LEN        | LEN        | LEN        | LEN        |
| LET        | LET        | LET        | LET        | LET        |
| LIST       | LIST       | LIST       | LIST       | LIST       |
| LOAD       | LOAD       | LOAD       | LOAD       | LOAD       |
| LOG        | LOG        | LOG        | LOG        | LOG        |
| MID\$      | MID\$      | MID\$      | MID\$      | MID\$      |
| NEW        | NEW        | NEW        | NEW        | NEW        |
| NOT        | NOT        | NOT        | NOT        | NOT        |