

Inhaltsverzeichnis zur Anleitung zum TOOL3  
\*\*\*\*\*

Allgemeines zum TOOL3 . . . . .	Seite 3-2
CLR . . . . .	Seite 3-3
DEC . . . . .	Seite 3-5
ERL . . . . .	Seite 3-7
ERN . . . . .	Seite 3-7
FRC . . . . .	Seite 3-9
FRE . . . . .	Seite 3-11
HEX\$ . . . . .	Seite 3-13
INSTR . . . . .	Seite 3-15
PGM . . . . .	Seite 3-17
PSIZE . . . . .	Seite 3-18
PURGE . . . . .	Seite 3-19
REDIM . . . . .	Seite 3-21
RESUME . . . . .	Seite 3-24
STRING\$ . . . . .	Seite 3-26
SWAP . . . . .	Seite 3-28
VKEEP . . . . .	Seite 3-30
VLIST . . . . .	Seite 3-32
Liste der Funktionen und Instruktionen . . . . .	Seite 3-34

Allgemeines zum TOOL3  
\*\*\*\*\*

TOOL3 stellt vorallem weitere Funktionen zur Verarbeitung von Variablen zur Verfügung.

Bei der Auswahl der Funktionen wurde auf folgendes Wert gelegt:

1. Die Funktion lässt sich nicht durch ein BASIC-Programm realisieren (z.B. REDIM, PURGE)
2. Die Funktion lässt sich zwar durch ein BASIC-Programm realisieren, braucht aber viel Aufwand (z.B. INSTR, CLR) oder, bei häufiger Anwendung, viel Rechenzeit (z.B. FRC, STRING\$).

Variablen-Bezeichnungen in dieser Anleitung:

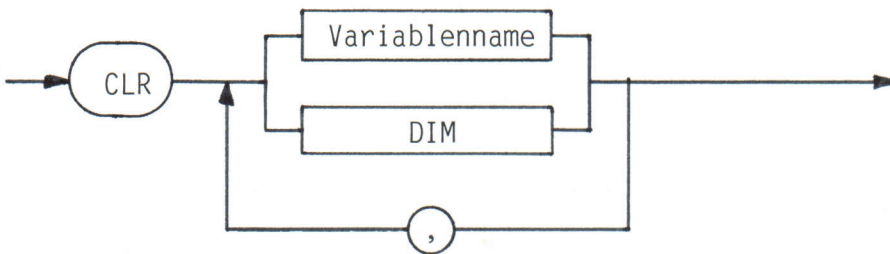
- Festvariablen: Variablen die nicht im Programmspeicher angelegt sind, sondern die durch das System immer zur Verfügung stehen.  
Festvariablen heissen A ... Z oder A\$ ... Z\$ und haben keinen Index.
- 2-Zeichen-Variablen: Variablen, deren Namen aus 2 Zeichen bestehen und die nicht durch DIM angelegt werden, sondern durch eine Zuweisung.  
2-Zeichen-Variablen haben keinen Index.
- DIM-Felder: DIM-Felder werden durch die Anweisung DIM angelegt, haben einen Namen aus einem oder zwei Zeichen und haben einen Index.

Anmerkung: Das \$-Zeichen gehört nicht zum Namen einer Variablen oder eines DIM-Feldes.

## CLR

```
*****  
*  
* Setzt den Inhalt der angegebenen Variablen, bzw.  
* DIM-Felder null.  
*  
*****
```

## Syntax



- Parameter Beschreibung:
- Variablenname: Festvariable A ... Z, A\$ ... Z\$ oder 2-Zeichen-Variablen die null gesetzt werden soll, bzw. DIM-Feld, das null gesetzt werden soll.
  - DIM: Wird DIM angegeben, so werden alle 2-Zeichen-Variablen und DIM-Felder null gesetzt.

Hinweise: - Mit @(\*) werden A ... Z, mit @\$(\*) A\$ ... Z\$ null gesetzt.

- Es können nur ganze DIM-Felder null gesetzt werden, aber keine einzelnen Elemente von Feldern.
- Es können nur bestehende 2-Zeichen-Variablen, bzw. DIM-Felder null gesetzt werden !
- TIME kann mit CLR nicht null gesetzt werden.
- CLR kann manuell oder programmkontrolliert ausgeführt werden.

## CLR (Beispiele)

### Beispiel 1

Die Anweisung

```
CLR DIM,@(*),@$(*)
```

setzt alle Festspeicher, 2-Zeichen-Variablen und DIM-Felder null.

### Beispiel 2

Die Programmanweisung

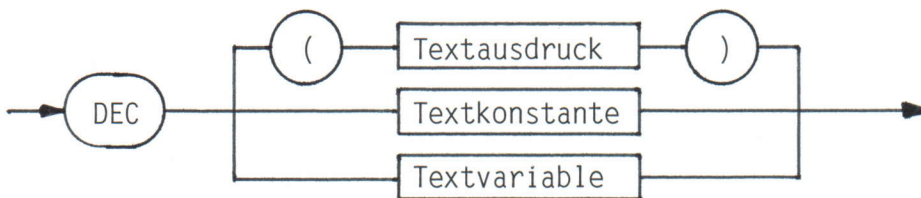
```
100: CLR A,B,XX$,T(*),IN$(*),AA,X,Z$
```

setzt die Festspeicher A,B,X und Z\$, die 2-Zeichen-Variablen AA und XX\$, sowie die DIM-Felder IN\$(\*) und T(\*) null.

## DEC

```
*****  
* Die Funktion DEC berechnet den Dezimalwert einer als *  
* Zeichenfolge angegebenen hexadezimalen Zahl. *  
*****
```

### Syntax



Ergebnis: numerischer Wert

- Funktionsbeschreibung:
- Die Funktion berechnet den Dezimalwert einer hexadezimalen Zahl, ähnlich wie die Funktion VAL (siehe Handbuch zum PC-1500, Seite 92).
  - Die Zeichenfolge wird aber als Hexadezimalzahl aufgefasst wie nach &. Im Gegensatz zu & kann die Zeichenfolge aber auch in einer Textvariablen gespeichert sein oder einen Textausdruck darstellen.
  - DEC ist die Umkehrfunktion zu HEX\$ (siehe Seite 3-13).
  - Die Konvertierung beginnt mit dem ersten Zeichen des Textausdruckes und wird abgebrochen, wenn ein Zeichen auftritt, das nicht in der Liste (0...9,A...F) enthalten ist.

- Funktionsbereich:
- Die Funktion verarbeitet nur vorzeichenlose Textausdrücke von 0...FFFF (hexadezimal).
  - Vornullen sind zugelassen, Zwischenräume (SPACE) nicht.

- Hinweis:
- DEC kann manuell oder programmkontrolliert ausgeführt werden.

## DEC (Beispiele)

Manuelles Rechnen (Protokoll-Ausdruck):

```
DEC "3E8"           1000
DEC "00000F1"      241
DEC ("5A"+"C3")    23235
```

Programm-Beispiel:

```
100:"Z"REM DEC
110:A$="5AC3"
120:A=DEC (MID$ ("
      * 5DC *", 3, 3))
130:LPRINT A
140:LPRINT DEC A$
150:END
```

Resultat des obigen Programmablaufs:

```
1500
23235
```