

Datentypen

Suffix	Datentyp	Länge	Wertebereich, Bemerkungen
%	INTEGER	2 Bytes	(Ganzzahl) -32 768...32 767
&	LONG	4 Bytes	(lange Ganzzahl) -2 147 483 648...2 147 483 647
!	SINGLE (einfach lange Gleitpkt.zahl)	4 Bytes	-3.402823 E38...-1.40129 E-45 für negative Werte 1.40129 E-45 ... 3.402823 E38 für positive Werte
#	DOUBLE (doppelt lange Gleitpunktzahl)	8 Bytes	-1.79769313486231 D308... -4.94065 D-324 für negative Werte 4.94065 D-324 ... 31.79769313486231 D308 für positive Werte
\$	STRING		max 32k Textzeichen

Standard-Farbcodes

0 = Schwarz	3 = Türkis	6 = Braun	9 = Hellblau	12 = Hellrot	15 = Weiß
1 = Blau	4 = Rot	7 = Hellgrau	10 = Hellgrün	13 = Rosa	
2 = Grün	5 = Violett	8 = Grau	11 = Helltürkis	14 = Gelb	

In SCREEN 0 sind als Hintergrundfarben nur die Codes 0...7 verwendbar.

Die wichtigsten Bildschirmmodi

SCREEN	Beschreibung, Auflösung, Farben, [Speicherbedarf], Bildschirmseiten
0	Textmodus, für alle Grafikkarten, 80*25 43 50 und 40*25 43 50 Spalten * Zeilen mit WIDTH einstellbar, 16 Farben, 8 Bildschirmseiten (0-7)
1	CGA/EGA/VGA-Karte, 320*200 Grafik, 30*25 Text, 4 aus 16 Farben, [2 Bits pro Pixel in 1 Ebene für GET/PUT], 1 Bildschirmseite (0)
2	CGA/MCGA/EGA/VGA-Karte, 640*200 Grafik, 80*25 Text, 2 aus 16 Farben, [1 Bit pro Pixel in 1 Ebene für GET/PUT], 1 Bildschirmseite (0)
7	EGA/VGA-Karte, 320*200 Grafik, 40*25 Text, 16 Farben, 8 Bildschirmseiten (0-7). Ruckelfreie Animationen auch auf langsamen Rechnern möglich, [4 Bits pro Pixel in 4 Ebenen für GET/PUT]
9	EGA/VGA-Karte, 640*350 Grafik, 80*15 Text, bis 16 Farben, [4 Bits pro Pixel (bei 16 Farben) in 4 Ebenen für GET/PUT], 2 Bildschirmseiten (0-1)
11	VGA-Karte, 640*480 Grafik, 80*25 30 50 60 Text (Voreinstellung: 80*30), 2 aus 256 Farben, gut geeignet für s/w-Grafiken, [1 Bit pro Pixel in 1 Ebenen für GET/PUT], 1 Bildschirmseite
12	VGA-Karte, 640 x 480 Grafik, 80*30 50 60 Text (Voreinstellung: 80*30), 6 aus 256 Farben, [4 Bits pro Pixel in 4 Ebenen für GET/PUT], 1 Bildschirmseite
13	VGA-oder MCGA-Karte, 320 x 200 Grafik, 40*25 Text, 256 Farben, [8 Bits pro Pixel in 1 Ebene für GET/PUT], 1 Bildschirmseite

Mathe-Funktionen

Befehl	Beschreibung
+ - * /	Grundrechenarten
x \ y	Ganzzahl-Division, liefert d. ganzzahligen Anteil von x/y; z.B. 13\5 = 2
x MOD y	Modulo-Operation (liefert d. Rest der Division x/y; z.B. 13 MOD 5 = 3)
x^y	Exponentialfunktion x ^y
SQR (x)	Quadratwurzel aus x
y = x^(1/n)	n-te Wurzel aus x
SGN(x)	Vorzeichen von x (Signum-Funktion)
ABS (x)	Absolutbetrag von x
SIN (x)	Sinus von x (x im Bogenmaß; 360° entspr. 2*Pi; z.B. sin 30° = SIN(30 * 3.1416/180) ::::...
COS (x)	Cosinus von x (x im Bogenmaß)
TAN (x)	Tangens von x (x im Bogenmaß)
ATN (x)	Arcustangens von x (ergibt den Winkel im Bogenmaß)
EXP (x)	x-te Potenz zur Basis e (e ^x)

LOG (x)	Natürlicher Logarithmus ln(x) (Logarithmus zur Basis e)
INT (x!)	Gleitpunkt-Integer-Wandlung mit Abrundung nach unten
FIX (x!)	Nachkommastellen abtrennen ohne Rundung
CINT (x!)	SINGLE Gleitpunktzahl zu Ganzzahl "wissenschaftlich" zur nächsten geraden Zahl runden (4,5 -> 4 5,5 -> 6)
CLNG(x#)	DOUBLE Gleitpunktzahl zu Ganzzahl "wissenschaftlich" zur nächsten geraden Zahl runden (4,5 -> 4 5,5 -> 6)
CSNG	Numerischen Wert in SINGLE-Gleitpunktzahl umwandeln
CDBL	Numerischen Wert in DOUBLE-Gleitpunktzahl umwandeln
HEX\$ (x)	Zahl x in Hexadezimalzahl-String umwandeln

Zeichenkettenfunktionen

Befehl	Beschreibung
LEFT\$ (t\$, n)	Liefert n Zeichen links aus dem String t\$ zurück
RIGHT\$ (t\$, n)	Liefert n Zeichen rechts aus dem String t\$ zurück
MID\$ (t\$, a, n)	Liefert n Zeichen ab dem a-ten zeichen aus einem String zurück
INSTR (a, t\$, s\$)	Durchsucht den String t\$ ab dem a-ten Zeichen nach dem String t\$ und liefert die Zeichenposition der Fundstelle zurück bzw. 0 wenn s\$ nicht gefunden
LCASE\$ (t\$)	Groß- in Kleinbuchstaben umwandeln (nicht für Umlaute)
UCASE\$ (t\$)	Klein- in Großbuchstaben umwandeln (nicht für Umlaute)
MID\$ (t\$, a, n) =e\$	n Zeichen im String t\$ ab Zeichen-Position a durch den String e\$ ersetzen
SPACE\$ (n)	Liefert einen String mit n Leerzeichen
STRING\$ (n, t\$)	Liefert einen String , der n mal das 1. Zeichen von t\$ enthält
LEN (t\$)	Liefert die Länge des Strings t\$ in Anzahl Zeichen
CHR\$ (code)	Liefert das Textzeichen mit dem angegebenen ASCII-Code
ASC (t\$)	Liefert den ASCII-Code des ersten Zeichens von t\$
t\$ = t1\$ + t2\$	Strings zusammenfügen (z.B. "Es" + "sel" = "Esel")
STR\$ (z)	Numerischen Wert z in String umwandeln
VAL (t\$)	String t\$ in numerischen Wert umwandeln

Die wichtigsten Tastencodes (ASCII-Tabelle in "Hilfe | Inhalt | ASCII-Zeichen-Codes")

Taste	Code	Taste	Code
Eingabe	CHR\$(13)	Cursor hoch	CHR\$(0) + "H"
Leertaste	CHR\$(32)	Cursor tief	CHR\$(0) + "P"
Backspace	CHR\$(8)	Cursor links	CHR\$(0) + "K"
Esc	CHR\$(27)	Cursor rechts	CHR\$(0) + "M"
Einf	CHR\$(0) + "R"	Bild hoch	CHR\$(0) + "I"
Entf	CHR\$(0) + "S"	Bild tief	CHR\$(0) + "Q"
Pos1	CHR\$(0) + "G"	F1...F10	CHR\$(0) + CHR\$(59).....(68)
Ende	CHR\$(0) + "O"	F11...F12	CHR\$(0) + CHR\$(133)...(134)

Befehle zur Bildschirmausgabe

PRINT [numerische oder Textausdrücke] [; | ,] => [; = ohne Zeilenvorschub | , = mit Tabulator]

Befehle zur Tastatureingabe

INPUT Text\$ {;|,} Variable => Aufforderungs-Text anzeigen und Variableneingabe anfordern (mit | ohne Fragezeichenanzeige)

t\$ = INPUT\$ (n) => Wartet bis n Textzeichen von der Tastatur in t\$ eingelesen

Wartezeit einfügen

Zeit = TIMER : DO : LOOP WHILE TIMER < Zeit + Wartezeit *Wartezeit in sec*