

C64 BASIC V2.0: Funktion: RIGHT\$

geschrieben von Andreas Potthoff | 31. Mai 2025

RIGHT\$

- Token: 201 / \$C9 | Abkürzung: rI | ROM Exec: 46892 / \$B72C
- Typ: Funktion, string | Funktion: Stringbearbeitung
- Syntax: RIGHT\$(<string>,<numeric>)

Aktion, Hinweise, Tipps, Tricks:

Die Funktion RIGHT\$ gibt einen Teilstring vom rechten Ende des <string>-Arguments zurück. Die Länge des Teilstrings wird durch das <numeric>-Argument definiert, das eine beliebige Ganzzahl im Bereich von 0 bis 255 sein kann. Ist der Wert des numerischen Ausdrucks Null, wird ein Nullstring ("") zurückgegeben. Ist der im <numeric>-Argument angegebene Wert größer als die Länge des <string>-Arguments , wird der gesamte String zurückgegeben.

Beispiele:

```
10 MSG$="COMMODORE COMPUTERS"  
20 PRINT RIGHT$(MSG$,9)  
RUN
```

```
COMPUTERS
```

Befehle	CLOSE, CLR, CMD, CONT, DATA, DEF, DIM, END, FOR, GET, GET#, GOSUB, GOTO, IF, INPUT, INPUT#, LET, LIST, LOAD, NEW, NEXT, ON, OPEN, POKE, PRINT, PRINT#, READ, REM, RESTORE, RETURN, RUN, SAVE, STOP, SYS, VERIFY, WAIT
Befehle Spezial (Bywords)	FN, GO, NOT, SPC(, TAB(, THEN, TO, STEP
Arithmetische und logische Operatoren	+, −, *, /, ^, >, =, <, AND, OR
Funktionen	ABS, ASC, ATN, CHR\$, COS, EXP, FRE, INT, LEFT\$, LEN, LOG, MID\$, PEEK, POS, RIGHT\$, RND, SGN, SIN, SQR, STR\$, TAN, USR, VAL
Konstanten und Systemvariablen	Pi, ST, TI, TI\$

C64 BASIC V2.0: Funktion: COS

geschrieben von Andreas Potthoff | 31. Mai 2025

COS

- Token: 190 / \$BE | Abkürzung: – | ROM Exec: 57956 / \$E264
- Typ: Funktion, numerisch | Funktion: Arithmetik
- Syntax: COS (<number>)

Aktion, Hinweise, Tipps, Tricks:

Diese mathematische Funktion berechnet den Kosinus der Zahl, wobei die Zahl ein Winkel im Bogenmaß ist.

Beispiele:

```
10 PRINT COS(0)
20 X = COS(Y* {pi} /180) : REM Grad in Bogenmaß umrechnen
```

Befehle	CLOSE, CLR, CMD, CONT, DATA, DEF, DIM, END, FOR, GET, GET#, GOSUB, GOTO, IF, INPUT, INPUT#, LET, LIST, LOAD, NEW, NEXT, ON, OPEN, POKE, PRINT, PRINT#, READ, REM, RESTORE, RETURN, RUN, SAVE, STOP, SYS, VERIFY, WAIT
Befehle Spezial (Bywords)	FN, GO, NOT, SPC(, TAB(, THEN, TO, STEP
Arithmetische und logische Operatoren	+, −, *, /, ^, >, =, <, AND, OR
Funktionen	ABS, ASC, ATN, CHR\$, COS, EXP, FRE, INT, LEFT\$, LEN, LOG, MID\$, PEEK, POS, RIGHT\$, RND, SGN, SIN, SQR, STR\$, TAN, USR, VAL
Konstanten und Systemvariablen	Pi, ST, TI, TI\$

C64 BASIC V2.0: Funktion: FRE

geschrieben von Andreas Potthoff | 31. Mai 2025

FRE

- Token: 184 / \$B8 | Abkürzung: fR | ROM Exec: 45949 / \$B37D
- Typ: Funktion, numerisch | Funktion: Arithmetik
- Syntax: FRE (<variable>)

Aktion, Hinweise, Tipps, Tricks:

Diese Funktion zeigt an, wie viel RAM für Ihr Programm und dessen Variablen verfügbar ist. Versucht ein Programm, mehr Speicherplatz zu verwenden, als verfügbar ist, wird der Fehler „OUT OF MEMORY“ angezeigt. Die <variable> Zahl in Klammern kann einen beliebigen Wert haben und wird bei der Berechnung nicht berücksichtigt. Wenn das Ergebnis von FRE negativ ist, addieren Sie 65536 zum Ergebnis.

Beispiele:

```
PRINT FRE(0)
10 X = (FRE(K) - 1000) / 7
950 IF FRE(0) < 100 THEN PRINT "NOT ENOUGH ROOM"
```

```
PRINT FRE(0) - (FRE(0) < 0) * 65536
```

Die Funktion zeigt immer den aktuell verfügbaren RAM an

Befehle	CLOSE, CLR, CMD, CONT, DATA, DEF, DIM, END, FOR, GET, GET#, GOSUB, GOTO, IF, INPUT, INPUT#, LET, LIST, LOAD, NEW, NEXT, ON, OPEN, POKE, PRINT, PRINT#, READ, REM, RESTORE, RETURN, RUN, SAVE, STOP, SYS, VERIFY, WAIT
----------------	---

Befehle Spezial (Bywords)	FN, GO, NOT, SPC(, TAB(, THEN, TO, STEP
Arithmetische und logische Operatoren	+, −, *, /, ^, >, =, <, AND, OR
Funktionen	ABS, ASC, ATN, CHR\$, COS, EXP, FRE, INT, LEFT\$, LEN, LOG, MID\$, PEEK, POS, RIGHT\$, RND, SGN, SIN, SQR, STR\$, TAN, USR, VAL
Konstanten und Systemvariablen	Pi, ST, TI, TI\$

C64 BASIC V2.0: Systemvariable: ST

geschrieben von Andreas Potthoff | 31. Mai 2025

ST

- Token: − / − | Abkürzung: − | ROM Exec: 65463 / \$FFB7
- Typ: Systemvariable | Funktion: Status
- Syntax: ST

Aktion, Hinweise, Tipps, Tricks:

Gibt den Abschlussstatus für die letzte Ein-/Ausgabeoperation zurück, die an einer geöffneten Datei ausgeführt wurde. Der Status kann von jedem Peripheriegerät gelesen werden. Das Schlüsselwort STATUS (oder einfach ST) ist ein systemdefinierter Variablenname, in den der KERNAL den Status von Ein-/Ausgabeoperationen einfügt.

Eine Tabelle mit den Statuscodewerten für Band-, Drucker-, Disketten- und RS-232-Dateioperationen:

ST Bit Position	ST Numeric Value	Cassette Read	Serial Bus R/W	Tape Verify + Load
0	1		time out write	
1	2		time out read	
2	4	short block		short block
3	8	long block		long block
4	16	unrecoverable read error		any mismatch
5	32	checksum error		checksum error
6	64	end of file	E0I	
7	-128	end of tape	device not present	end of tape

Beispiele:

```

10 OPEN 1,4:OPEN 2,8,4,"MASTER FILE,SEQ,W"
20 GOSUB 100:REM CHECK STATUS
30 INPUT#2,A$,B,C
40 IF STATUS AND 64 THEN 80:REM HANDLE END-OF-FILE
50 GOSUB 100:REM CHECK STATUS
60 PRINT#1,A$,B;C
70 GOTO 20
80 CLOSE1:CLOSE2
90 GOSUB 100:END
100 IF ST > 0 THEN 9000:REM HANDLE FILE I/O ERROR
110 RETURN

```

Befehle	CLOSE, CLR, CMD, CONT, DATA, DEF, DIM, END, FOR, GET, GET#, GOSUB, GOTO, IF, INPUT, INPUT#, LET, LIST, LOAD, NEW, NEXT, ON, OPEN, POKE, PRINT, PRINT#, READ, REM, RESTORE, RETURN, RUN, SAVE, STOP, SYS, VERIFY, WAIT
Befehle Spezial (Bywords)	FN, GO, NOT, SPC(, TAB(, THEN, TO, STEP
Arithmetische und logische Operatoren	+, −, *, /, ^, >, =, <, AND, OR
Funktionen	ABS, ASC, ATN, CHR\$, COS, EXP, FRE, INT, LEFT\$, LEN, LOG, MID\$, PEEK, POS, RIGHT\$, RND, SGN, SIN, SQR, STR\$, TAN, USR, VAL
Konstanten und Systemvariablen	Pi, ST, TI, TI\$

C64 BASIC V2.0: Funktion: RND

geschrieben von Andreas Potthoff | 31. Mai 2025

RND

- Token: 187 / \$BB | Abkürzung: rN | ROM Exec: 57495 / \$E097
- Typ: Funktion, numerisch | Funktion: Arithmetik
- Syntax: RND (<numeric>)

Aktion, Hinweise, Tipps, Tricks:

RND erzeugt eine Gleitkommazahl mit einem Wert zwischen 0,0 und 1,0. Der Computer generiert eine Folge von Zufallszahlen, indem er Berechnungen mit einer Startzahl durchführt, die im Computerjargon als Seed bezeichnet wird. Die RND-Funktion wird beim Systemstart gestartet. Das <numeric>-Argument ist ein Dummy-Argument, mit Ausnahme seines Vorzeichens (positiv, null oder negativ). Ist das <numeric>-Argument positiv, wird dieselbe pseudozufällige Zahlenfolge zurückgegeben, beginnend mit einem bestimmten Seed-Wert. Unterschiedliche Seeds ergeben unterschiedliche Zahlenfolgen, aber jede Folge ist wiederholbar, wenn man von derselben Seed-Zahl ausgeht. Eine bekannte Folge von Zufallszahlen ist beim Testen von Programmen hilfreich. Wählt man als <numeric>-Argument Null, generiert RND eine Zahl direkt aus einer freilaufenden Hardware-Uhr (der System-Jiffy-Clock). Negative Argumente führen dazu, dass die RND-Funktion bei jedem Funktionsaufruf neu gestartet wird.

Dieser Speicherort enthält den von der RND-Funktion zurückgegebenen 5-Byte-Gleitkommawert. Er ist zunächst auf einen aus dem ROM kopierten Startwert gesetzt (die fünf Bytes lauten 128, 79, 199, 82, 88 – \$80, \$4F, \$C7, \$52, \$58). \$008B-\$008F RND Function Seed Value

Beim Aufruf der Funktion RND(X) hat der numerische Wert von X keinen Einfluss auf die zurückgegebene Zahl, wohl aber sein Vorzeichen. Ist X gleich 0, generiert RND einen Startwert aus Hardware-Timern auf Chipebene. Ist X positiv, gibt RND(X) die nächste Zahl in einer arithmetischen Folge zurück. Diese Folge wiederholt sich so lange, ohne sich zu wiederholen, und ergibt eine so gleichmäßige Zahlenverteilung, dass sie als zufällig angesehen werden kann. Ist X negativ, wird der Startwert in eine Zahl geändert, die einer verschlüsselten Gleitkommadarstellung der Zahl X selbst entspricht.

Bei einem bestimmten Startwert wird immer dieselbe pseudozufällige Zahlenfolge zurückgegeben. Dies kann für Debugging-Zwecke nützlich sein, jedoch nicht, wenn Sie

wirklich zufällige Zahlen benötigen.

Die traditionelle Commodore-Methode zur Auswahl eines Zufallsstartwerts verwendet den Ausdruck `RND(-TI)`, hauptsächlich weil `RND(0)` auf frühen PETs nicht korrekt funktionierte. Obwohl die `RND(0)`-Form auch auf dem 64er nicht richtig funktioniert (siehe Position 57495 (\$E097)), kann der Ausdruck `RND(-RND(0))` einen zufälligeren Startwert erzeugen.

Beispiele:

```
220 PRINT INT(RND(0)*50)           (Gibt Zufallszahlen  
zwischen 0 und 49 zurück)  
100 X=INT(RND(1)*6)+INT(RND(1)*6)+2 (Simuliert zwei  
Würfel)  
100 X=INT(RND(1)*1000)+1          (Zufallszahlen  
zwischen 1 und 1000)  
100 X=INT(RND(1)*150)+100         (Zufallszahlen  
zwischen 100 und 249)  
100 X=RND(1)*(U-L)+L              (Zufallszahlen  
zwischen der oberen (U) und unteren (L) Grenze)
```

Befehle	CLOSE, CLR, CMD, CONT, DATA, DEF, DIM, END, FOR, GET, GET#, GOSUB, GOTO, IF, INPUT, INPUT#, LET, LIST, LOAD, NEW, NEXT, ON, OPEN, POKE, PRINT, PRINT#, READ, REM, RESTORE, RETURN, RUN, SAVE, STOP, SYS, VERIFY, WAIT
Befehle Spezial (Bywords)	FN, GO, NOT, SPC(, TAB(, THEN, TO, STEP
Arithmetische und logische Operatoren	+, -, *, /, ^, >, =, <, AND, OR

Funktionen	ABS, ASC, ATN, CHR\$, COS, EXP, FRE, INT, LEFT\$, LEN, LOG, MID\$, PEEK, POS, RIGHT\$, RND, SGN, SIN, SQR, STR\$, TAN, USR, VAL
Konstanten und Systemvariablen	Pi, ST, TI, TI\$

C64 BASIC V2.0: Funktion: SIN

geschrieben von Andreas Potthoff | 31. Mai 2025

SIN

- Token: 191 / \$BF | Abkürzung: sI | ROM Exec: 57963 / \$E26B
- Typ: Funktion, numerisch | Funktion: Arithmetik
- Syntax: SIN (<numeric>)

Aktion, Hinweise, Tipps, Tricks:

SIN gibt den Sinus des <numeric>-Arguments im Bogenmaß an. Der Wert von COS(X) entspricht SIN($x + 3,14159265/2$).

Beispiele:

```
235 AA=SIN(1.5):PRINT AA
    .997494987
```

Befehle	CLOSE, CLR, CMD, CONT, DATA, DEF, DIM, END, FOR, GET, GET#, GOSUB, GOTO, IF, INPUT, INPUT#, LET, LIST, LOAD, NEW, NEXT, ON, OPEN, POKE, PRINT, PRINT#, READ, REM, RESTORE, RETURN, RUN, SAVE, STOP, SYS, VERIFY, WAIT
Befehle Spezial (Bywords)	FN, GO, NOT, SPC(, TAB(, THEN, TO, STEP
Arithmetische und logische Operatoren	+, -, *, /, ^, >, =, <, AND, OR
Funktionen	ABS, ASC, ATN, CHR\$, COS, EXP, FRE, INT, LEFT\$, LEN, LOG, MID\$, PEEK, POS, RIGHT\$, RND, SGN, SIN, SQR, STR\$, TAN, USR, VAL
Konstanten und Systemvariablen	Pi, ST, TI, TI\$