

Appendix F - Math and Derivative Helpers

Numeric and string helpers BASIC provides at the prompt. Each one is documented in full in Chapter 2; this appendix is the single-page reference.

F.1 Numeric functions

Function	Domain	Result
ABS(x)	any number	Absolute value $ x $.
SGN(x)	any number	-1 if $x < 0$, 0 if $x = 0$, 1 if $x > 0$.
INT(x)	any number	Integer part of x , truncated toward zero.
SQR(x)	$x \geq 0$	Square root.
EXP(x)	any number	e^x .
LOG(x)	$x > 0$	Natural logarithm.
SIN(x)	radians	Sine.
COS(x)	radians	Cosine.
TAN(x)	radians	Tangent.
ATN(x)	any number	Arctangent, result in $(-\pi/2, \pi/2)$.
RND(x)	sign-sensitive	$x > 0$: next pseudo-random in $[0, 1)$. $x = 0$: repeat last value. $x < 0$: reseed with INT(x) and emit first draw.
FRE(x)	argument ignored	Bytes of free BASIC program / variable storage.
POS(x)	argument ignored	Current cursor column on the terminal.
USR(x)	address	Call an IE64 user machine-code routine; see Chapter 2.
MEMALLOC(size[,align])	byte count, optional alignment	Allocate a public low32 buffer and return its exact address.
PEEK(a)	address	Byte-width read; alias for PEEK8(a).
PEEK8(a)	address	Byte-width read.
PEEK16(a)	address	16-bit aligned read.
PEEK32(a)	address	32-bit aligned read.
PEEK64(a)	address	64-bit aligned read; returns an exact integer qword.

F.2 String functions

Function	Result
LEN(s\$)	Number of bytes in s\$.

Function	Result
VAL (s\$)	Numeric value parsed from the start of s\$.
ASC (s\$)	ASCII code of the first byte of s\$.
CHR\$(n)	One-byte string with ASCII code n.
STR\$(n)	Decimal text form of n, with a leading space for non-negative numbers.
LEFT\$(s\$, n)	First n bytes of s\$.
RIGHT\$(s\$, n)	Last n bytes of s\$.
MID\$(s\$, p [, n])	n bytes of s\$ starting at byte position p (1-based). With one argument, returns everything from p to the end.
TAB(n)	Move cursor to column n; only valid inside PRINT.
HEX\$(n)	Uppercase hexadecimal text form of n.
BIN\$(n)	Binary text form of n.

F.3 Derived identities

These are not separate functions: they are equivalences a program can rely on when composing the helpers above.

Wanted	Compute as
natural log	$\text{LOG}(x)$
base-10 log	$\text{LOG}(x) / \text{LOG}(10)$
base-2 log	$\text{LOG}(x) / \text{LOG}(2)$
arcsine of x	$\text{ATN}(x / \text{SQR}(1 - x*x))$
arccosine of x	$1.5707963 - \text{ATN}(x / \text{SQR}(1 - x*x))$
truncated remainder	$x - \text{INT}(x / m) * m$
round to nearest	$\text{INT}(x + 0.5)$ for $x \geq 0$; $\text{INT}(x - 0.5)$ for $x < 0$
fractional part	$x - \text{INT}(x)$
min(a,b)	$(a + b - \text{ABS}(a - b)) / 2$
max(a,b)	$(a + b + \text{ABS}(a - b)) / 2$
pi	$4 * \text{ATN}(1)$
e	$\text{EXP}(1)$

F.4 Range and precision

Numeric functions return BASIC double-precision values, except for helpers such as PEEK64 and MEMALLOC that deliberately return exact integer payloads for hardware work. The trigonometric helpers accept any radian argument; very large arguments lose precision in the usual way during range reduction. Programs that need accuracy across many octaves of input should fold the argument into $[-\pi, \pi]$ themselves.

RND is a 32-bit linear-congruential generator. Its sequence is deterministic and reproducible after a RND(-seed) reseed; the first draw immediately after the reseed depends on seed and the generator's fixed multiplier.