

MEMNEW

The MEMNEW function allocates new memory and returns a MEM memory block referring to it.

Syntax

```
memoryBlock = _MEMNEW(byteSize)
```

Parameters

- The *byteSize* parameter is the desired byte size of the memory block based on the variable type it will hold.

Description

- The *memoryBlock* value created holds the elements .OFFSET, .SIZE, .TYPE and .ELEMENTSIZE.
- MEMNEW does not clear the data previously in the memory block it allocates, for speed purposes.
- To clear previous data from a new memory block, use MEMFILL with a byte value of 0.
- When a new memory block is created the memory .TYPE value will be 0.
- **If the read only memory block .SIZE is 0, the memory block was not created.**
- **All values created by memory functions must be freed using MEMFREE with a valid MEM variable.**

Examples

Example: Shows how SINGLE numerical values can be passed, but non-fixed STRING lengths cannot get the value.

```
DIM m AS MEM  
DIM f AS STRING * 5
```

Contents

[Syntax](#)

[Parameters](#)

[Description](#)

[Examples](#)

[See also](#)

```
m = MEMNEW(5) 'create new memory block of 5 bytes
a = 12345.6
MEMPUT m, m.OFFSET, a 'put single value
MEMGET m, m.OFFSET, b 'get single value
PRINT "b = "; b
c$ = "Doggy"
MEMPUT m, m.OFFSET, c$ 'put 5 byte string value
MEMGET m, m.OFFSET, d$ 'get unfixed length string value
MEMGET m, m.OFFSET, f 'get 5 byte string value
e$ = MEMGET(m, m.OFFSET, STRING * 5) 'get 5 byte string value
PRINT "d$ = "; d$; LEN(d$) 'prints empty string
PRINT "e$ = "; e$; LEN(e$)
PRINT "f = "; f; LEN(f)
```

```
b = 12345.6
d$ = 0
e$ = Doggy 5
f = Doggy 5
```

See also

- MEM, MEMPUT
- MEMGET, MEMGET (function)
- MEMFILL, MEMFREE

Navigation:

Main Page with Articles and Tutorials
Keyword Reference - Alphabetical
Keyword Reference - By usage

