GxScript 3.0 System Runtime Library

1 Math functions

sin(x), cos(x), tan(x), asin(x), acos(x), atan(x), log(x), exp(x), round(x), floor(x), ceil(x), abs(x)

2. Console

read()
returns a string representing what the user is entered in the console.

write(obj)
prints the content of obj to console.

3 String processing

StrLen(str), StrLeft(str, size), StrRight(str, size), SubStr(str, pos, count), StrLTrim(str), StrRTrim(str),
StrTrim(str), StrDelete(str, pos, count)

4 Array manipulation (as member methods/properties)

Size : Property. Set/Get length of the array
Insert(array|variable, pos=end); Insert an element or an array of elements at specified position.
Delete(pos,count): Remove a range of elements from array
Subset(pos,count) : Get a subset copy of the array
Clone(): Create a new copy of this array

5 Reflection

GetClassCtor(str, n) : Get constructor with n parameters of type specified in str.

Common members of all classes for reflection purposes:

_prototype member:

ClassName Property. Gets the name of this class
Members member(indexable through [] operator)

Count: property. Gets the number of members in this class

Add: Method

Exists: Method

Remove: Method

Ancestors: Member(indexable through [] operator)

Count: Property. Gets the number of ancestors of this class

Add: Method. Add an ancestor to this class

Exists: Method. Returns whether an object is an ancestors of this class

Remove: Method. Removes an ancestor

Members[i] : the i-th member in the class

Name : name of the i-th member

Visibility : accessibility of i-th member, where (0=public, 1= protected, 2= private)

Ancestors[i] : returns info about i-th ancestor of this class

ClassName : class name of i-th ancestor

Object : return the base object of this ancestor

6 Dynamic code compilation

__compile(source, scope, [ErrList])

Compiles the GxScript code specified in source.

scope: the scope of the compiled function, where

0: Compiles to global scope. All existing code can access the compiled function, and the compiled function can access all global variables.

1: Compiles to the scope of local function. Only current function can call the compiled function.

2: Compiles to a stand alone scope. Only current function can call the compiled function, and the compiled function cannot acesss any variable defined in existing code.

ErrList is an list for storing compiler errors. This parameter is optional.

If compilation succeeds, the function returns a function object representing the compiled function. The function will return 0 otherwise.