Math-c Documentation

Math-c functions

clc()

clear the text of command window.

return

y -> 0.

clear()

clear(a..N)

erase the environment variables in the current scope

a..N -> string, variable number of parameters

returns

y -> 0. if the parameter is empty erase the environment variables, if a..N are defined, erase of the variable name selected.

If parameter "a" is equal to "reset functions", erase all the custom functions defined by the user and set default build in functions

do

dowhile

do-while loop

example to loop 10 times:

```
i = 0;

do

 //code here

 i+=1

dowhile(i<10)     ///also can be written as dowhile i<10
```

error(x)

display the error string x and exits the script

x -> string

returns

y -> 0.

exit()

exits the script

returns

y -> 0.

for

for loop

example to loop 10 times:

```
for i=1:10     ///also can be written as for(i=1:10)
//code here
end
```

function

return

keyword to define a function, must return a value or matrix.

example:

```
function myfunctionName(num1,num2)
return num1 + num2;
end
```

then to call this function in code:

```
value = myfunctionName(3,2)
```

import

import the functions defined in other file.

Note: the file is taken from the selected directory path is set in Math-c > Preferences, if is not exist, is taken from iCloud.

only imports the functions and doesn't validate them until are executed.

Does not run any code from the imported file, only add the functions.

Other limitation from the imported files is that can not be debugged.

Example:

```
#import signalfuncs.mc
```

y = input(x)

stop program and ask for input to user displaying the string x.

x -> String

returns

y -> the value entered by user.

y = isArray(x)

return if is an array.

x -> value

returns

y -> 0 if false and 1 if true

y = isinf(x)

return if value x is infinite.

x -> value

returns

y -> 0 if false and 1 if true

y = isInteger(x)

return if value x is integer.

x -> value

returns

y -> 0 if false and 1 if true

y = isnan(x)

return if value x is nan (undefined number).

x -> value

returns

y -> 0 if false and 1 if true

y = isString(x)

return if value x is string.

x -> value

returns

y -> 0 if false and 1 if true

print(x)

displays string x to command window.

x -> String

returns

y -> 0.

y = str(format,x)

format the number to format .

format -> string

x -> scalar value

returns

y -> formated string

more information can be found in the printf function of C programming

f ~ for float

g ~ short notation

x ~ hexadecimal notation

u ~ unsigned integer

o ~ octal notation

%l#x ~ hexadecimal prefix

%.(number) number of decimals to display

example:

```
>>> x=7+3i
 >>> str("%lf",x)                //only real float
ans = 7.000000
>>> str("%.2lf im-> %lg",x)      //2 decimals float and short representation
 ans = 7.00 im-> 3
 >>> str("%ld im-> %l#x",x)       //integer and hexadecimal format
 ans = 7 im-> 0x3
>>> str("%+ld im-> %lo",x)       //integer with sign and octal
ans = +7 im-> 0x3
```

while

while loop

example to loop 10 times:

```
i=0
while i<10
//code here
 i+=1
 end
```