

MATHSTUDIO

Quick Reference

VARIABLE DECLARATION AND UTILIZATION

There's no distinction for strings, integers and real variables. To declare them, simply name them and assign them a value:

```
[variableName] = value
```

For mathematical expression:

```
[variableName] = calc value
```

Variables usage example:

```
[a] = 2  
[b] = 5  
[sum] = calc {a}+{b}
```

Sum is 5; note that if you do not put the `calc` keyword, sum will be a string of value "2+5".

For user input, just use:

```
[var] = input
```

SELECTION AND ITERATION

Selection syntax:

```
if {condition} then  
    //code  
else  
    //code  
endif
```

Iteration syntax:

```
loop {condition} then  
    //code  
endloop
```

Goto syntax:

```
:mark  
//code  
goto mark
```

COMMANDS

NAME	DESCRIPTION	EXAMPLE
OUTPUT	Prints a text	output HelloWorld
INPUT	Insert a value	[var] = input
CALC	Calculates a mathematical expression	[var] = calc {a}^2+{b}
FDEF	Defines a function	fdef f(x) = x^2
ARGDEF	Defines an argument	argdef y=4
CONSTDEF	Defines a constant	constdef pi=3.14
PLOT	Plots a function graph	plot "x^2+2"
EXE	Executes an external script	exe "script" "line1" "line2" "argument"
CALL	Starts a process	call "process" "argument"
TCOLOR	Changes text color	tcolor red
BCOLOR	Changes background color	bclolor white
TITLE	Changes interpreter title	title Custom Title
GOTO	Go to a mark in the code	goto mark
HELP	Shows help for a command	help calc
DELFILE	Deletes a file	delfile "path"
PUTFILE	Adds lines to a file	putfile "path" "line1" "line2" ... "lineN"
CLEAR	Clear console output	clear
PAUSE	Pause code execution	pause
WAIT	Wait for milliseconds	wait 1000
STREQ	String comparison	if streq "string1" "string2" then
CHECKFILE	Checks if a file exists	if checkfile "path" then

For a list of all available functions in `calc` command, open console with CTRL+SHIFT+C and type "help calc".

PLUGINS

To install plugins, simply put them in the **Plugins** directory in the main folder of MathStudio. No DLLs from the MathStudio namespace are required to be put in this folder.

Plugins must implement the **MathStudio.Core.Plugin** interface:

```
string Name { get; } //Plugin name
string Description { get; } //Plugin description
string Version { get; } //Plugin version
string Author { get; } //Plugin author
bool ExecutableInEditor { get; } //Is executable in the editor?
bool ExecutableInConsole { get; } //Is executable in the console?
string[] CommandsList { get; } //Commands list (Console)
string Help(string cmd); //Commands help (Console)
string Execute(string cmd); //Execute (Console)
void ExecuteInEditor(object sender, EventArgs e); //Execute (Editor UI)
```

Plugins can use all the members of the MathStudio namespace and must be written in a .NET compatible language. More information about APIs at math.gomanga.it/docs.

For samples, look at the **Samples** folder in the main directory of MathStudio. There are also some implementation of the API.