# Create Variable

>{variable name} as (num|dec|word|binary) (reachable|reachable\_all|closed) (=assignment=

>foo as num closed = 10+8

>bar as binary reachable = [{foo} is 18 or false]

# Use variables

Instead of using fixed values, use {varname}

Ex.

>{foo} + {bar}

# Unload variables

>uload:varname

# Math

>3+7 (only for console)

>out:3+7 (usage in script)

# Dec/Num Comparison

[num {is|not|smaller|bigger|smallerIs|biggerIs} num1 ….]

Ex. Console:

>[3+7 is 10]

>[3 not 3]

Ex. Script

>out: [3+7 is 10]

>out: [3 not 3]

# Conditions

>case[logic|dec/num comparison]

>endcase

ex.

>case[3+7 is 10]

>out:‘I am a condition‘

>endcase

# Loops

>runala[logic|dec/num comparison]

>endrunala

ex.

>runala[3+7 is 10]

>out:‘I am a loop‘

>endrunala

# Dump variables

>dumpVars:{num|dec|word|binary|all}

# Logic

[condition {or|and|is|not} condition1 ….]

Ex. Console:

>[true or false]

>[(3+7 is 10) and false]

Ex. script:

>out:[true or false]

>out:[(3+7 is 10) and false]

# Get datatype of a variable

Ex. Console

>type:foo

Ex. Script

>bar = type:foo

# Print

>out:‘word‘

Ex.: > out:‘Hello world‘

>out:3+7

>out:[true or false]

>out:variable

Ex.: >out:foo

# Assign a non word to a word

foo as word reachable = out:3+7

foo as word reachable = out:[true or false]

# Call a method

a->GetData:void (when there is not input parameter, void comes after the method name)

a->GetData:url

# Load another scriptfile

load ‘Helloworld.hades‘ (loads and executes the file as a script)

load ‘Helloworld.hades‘ as a (loads the script and puts it into a)

a->print:void

# Exit the program

exit:{exitcode}