Page Up and Page Down scrolls the console. Any keypress will move it back down.
Page Up and Page Down WITH SHIFT changes the size of the Comm Window

Property Spaces can be used in the following commands

SAY
PAUSE

The SAY command and the Property Space – printing to the console.

Property Space can be used with the SAY command.

It can be used anywhere in the text parameter and is included between { and } characters.

E.g.
SAY hello, this will be green {green}

The property space goes between the parenthesis { and }. This is optional.
You can put a number between 8 and 24 in the property space to change display size.
You can put a valid font name in the property space to change display font.
You can put a color in the property space to change the display color.

E.g.
SAY {24, red, verdana} hello, this will be large and red in verdana.

Keywords in the property space can be in any order, upper or lower case, etc. The property space is designed to be flexible. It can exist anywhere in the SAY text parameter. But it ***must*** open with { and close with }. Separate each with a space or comma.

E.g.
SAY hello, this will be large {24, red, verdana} and red in verdana.

The table below shows, on the right hand side, the keyword which can be used in the Property Space.

Colors	Purple, pink, orange, white, black, orange, lorange, blue, dblue, lblue, green, dgreen, lgreen, brown, lbrown, dbrown, marone, grey, dgrey, lgrey, red, lred, dred, gold, yellow, lyellow, dyellow
Font Size	8, 9, 10, 11, … up to 48
Font Name	Arial, arial black, comic sans ms, courier new, Georgia, impact, lucida console, Tahoma, times new roman, trebuchet ms,
Attributes	bold, nobold, italic, noitalic, underline, nounderline, strikethrough,
Extras	flash, flashfast, flashslow, aligncenter,

The characters { and } can be printed independently (so they wont be recognized as a property space by using (** for { and **) for }

FILES AND DIRECTORIES SHOULD NOT HAVE SPACES, THEY MAY BE CONVERTED TO - DASHES

Parameters which include a directory name or file name can be specified as being from the local directory or from the root directory.

e.g a file \system\file.ds

From ANY directory, can be accessed with: \system\file.ds

However it can only be accessed from its parent directory (\system) with simply just: file.s

This goes for directories as well.

---

for a new line, type: say -

You can also run a file as a script simply by typing its name in the console. if you have a script called mscript, type ***myscript*** in the console to run the file as a script.

VARIABLE NAMES are like
$varname
They should be **no more than 36** characters in length. Variables are NOT case sensitive. $VaRiAbLE is the same as $variable.

Global Variables Which Always Exist are: $time, $date, $username

Variables do not need to be declared.
Examples of assigning values to variables:

$myvar = hello
or
$myvar = 12* 500
or
$multiplier = 10
$myvar = 12 * multiplier

In the last case, $myvar will be equal to 120

-
This can be done both in scripts or even typed in the console. (i.e., you can create and set variables by typing them in the console).

FOR LOOPS

Example

```
FOR $var = 1 to 5
  say var is $var
NEXT
```

you can also make loops backwards, or change the loop interval, by appending the optional STEP keyword:
the loop below will count down: 10, 8, 6, 4, 2

```
FOR $var = 10 to 0 step -2
  say var is $var
```

NEXT

IF STATEMENTS

Example

```
IF $var = 5 then
        say it is 5
ELSE IF $var = 6 then
        say it is 6
ELSE
        say it is not 5 or 6! must be something else..
END IF
```

Available Operators
= is equal to
> is greater than
< is less than
^ contains
~ doesn't contain

If statements can be nested within each other, etc, i.e.

```
IF $var = 5 then
        IF $var2 = 6 then
                say hello
        END IF
ENDIF
```

Goto Works

```
@START

GOTO START
```

NOTE ABOUT ASSIGNING VALUES

If you have problems with your data acting as numbers as opposing to a string, you may choose to encase your value in quotes, e.g.

$p = $date

Above, if the date is 5/5/2010, the value of $p will be seen as the mathematical result of the equation 5 / 5 / 10.
To ensure that strings are not calculated mathematically, simply use instead the following, noting the quotes are the value.

$p = "$date"

Cancel a running script

You can cancel a current script that is running by holding down the CTRL key and pressing C to cancel.

Switch between the four primary consoles with the F keys (F1, F2, F3, F4).

GETTING USER INPUT FROM SCRIPTS

Below shows the structure of how to get a user to enter data , which can then be processed by the script.

$var = input(message)
and
$var