ArnScript Instructions

Comments: Comments will not be read by the program. To write a comment, simply write “**” at the beginning of the line.

String: To create a string literal, the following format must be used: lit (quote). Some important notes:

- There must be a space between lit and the parentheses
- The content inside the parentheses can only be one word
- There is no need for quotation marks inside the parentheses

There is a variable (more on variables below) called “space” that is declared when the program begins (i.e. you do not need to create this variable). This variable can be added with other string literals to form more complex strings that include spaces and are more than one word.

Operations: Math operations supported by this program include addition, subtraction, multiplication, division, and modulo of integers, as well as addition of strings. Integers can be added to strings in a similar fashion to the addition in Java. Below are the symbols for each operation.

Addition: a + b
Subtraction: a - b
Multiplication: a * b
Division: a / b
Modulo: a % b

You can nest operations into one line.

IMPORTANT: Expressions will be evaluated from left to right, rather than using PEMDAS.
EVEN MORE IMPORTANT: Keep a space between the operands and operators in each expression.

Declaring Variables: There are two types of variables: strings and integers. The two keywords to create variables are decInt and decStr. The format for declaring variables is shown below:

```
decStr b -> lit (hey)
```

In this example, the variable b has been initialized with the value “hey.” Note that the value can only be one word, unless you add literals together. The “->” symbol is an assignment operator and establishes an equality relationship between the two sides of the “equation.”

Changing Variables: If you want to change the value of a variable, use the keywords changeInt and changeStr. You cannot change the type (int or String) of a variable, only its value. The format is similar to declaring variables, as shown below:

changeInt a -> 15

The only difference is the keyword. However, the benefit of this is that you can reuse the variable and change its value without creating a duplicate.

Reading Input: To read input from a user, use the following methods: for strings, use the command readStr, while for integers, use readInt. If you want to provide instructions to the user to input their data, simply use display before reading the input (see section below). The two methods mentioned will return a new value of their type (string or integer).

Note that these methods can be used in an expression with operators and variables. They do not have to be used in their own separate line or stored as a variable.

Output: To show the output of a variable or expression on the console, simply use the keyword “display” followed by what you want to display. Note that if you place an expression after display, the expression will be evaluated before being displayed. Make sure to put a space before display and the printed expression.

Format: display 7 + 3

Conditions: In a check statement, the keyword check is followed by a statement involving one of the following: >, <, >=, <=. If the statement is true, the code following this check will be run. Otherwise, it will be skipped. After writing the code for the check statement, put the line “end check” to tell the compiler that the code afterwards is not part of the check statement. Make sure to put spaces between each variable/expression/operator.

Loops: A repeat loop, as suggested by its name, repeats one line of code a specified number of times. The format of the repeat loop is as follows:

“repeat variable start end -> code” (NOTE THE SPACES)

After the repeat keyword, place a temporary variable. This variable is much like the “i” in a Java for loop. It can be used only in that line of code and will be forgotten afterwards.

The start and end values are integers. The temporary variable will start from the value “start” and end at “end”-1. The code after the arrow is the part that will be repeated (end-start) times. This code does not need any special instructions; it can be just like code you would put on a new line.

Errors: If the code does not compile, there will be an error. The error will specify the line in the ArnScript program where the error occurred. Note that blank lines are not included in the calculation of line numbers.

NOTE THAT BETWEEN ANY KEYWORD OR OPERATOR, THERE MUST BE A SPACE.

List of Keywords:

- decStr
- decInt
- changeInt
- changStr
- repeat
- check
- lit
- space