# Code Test

Summary

The goal of this test is to create a basic system that can process our fictitious scripting language (FSL). FSL is written in JSON.

This system receives FSL as input. The FSL defines variables and functions. Functionality is limited to create, delete, update, add, subtract, multiply, divide, print, as well as function calls. Variables are integers, floating point numbers, and strings.

Requirements

The finished project must support receiving multiple FSL scripts. Functions and variables must persist between FSL scripts. Resolve conflicts by overwriting existing variables or functions.

The system will create a representation of the script processed. The init function is immediately called after each script is processed.

The input is a JSON object of named variables and named functions. Variables are defined as a key value pair. References to variables are preceded by a hash mark (#).

A function is an array of command objects. An attribute called “cmd” is required and will define which operation to perform. All parameters passed to a function are referenced by a $.

Function calls are defined in the “cmd” attribute by preceding the function name with a hash mark (#).

|  |
| --- |
| Note  If you decide to use a language without built in JSON parsing please use a JSON library. Many libraries are available at: <http://www.json.org/> |

Sample Script

sample-script.txt  
init is the first function called by the system. It has a single command "#setup"  
setup has a series of commands:

1. var1 is set to 7
2. int1 is set to 5
3. int1’s value is printed to the screen
4. sum is called with the id int1, value1 equal to the value of int1, and value2 equal to the value of int2
5. int1’s value is printed to the screen
6. int3 is created with a value of 5
7. int1 is deleted
8. printAll is called

sum has a single command

1. The id passed in as $id is set to the value of $value1 added to $value2

printAll has 5 commands

1. var1’s value is printed to the screen
2. int1’s value is printed to the screen
3. int2’s value is printed to the screen
4. int3’s value is printed to the screen
5. float1’s value is printed to the screen
6. float2’s value is printed to the screen

Sample Output

7  
5  
10  
7  
undefined  
5  
5  
3.5  
1.5