# **Requirements Document**

Gage Peterson

Jun 27 2015

## Definitions

#### **Standard out**

Usually the main output of a console. Some place specified for most output of the program to go.

#### **The Primitives**

The basic built in types of the language including: Floats, Strings, Lists and Booleans

## Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **functional-requirement** | **demonstration scenarios** | **success measure** |
| 0 | The programing language **shall** be able to print the primitives to standard out | 1. print 1.0 2. print 'hi' 3. print [1.0 2.0 3.0] | 1. prints 1.0 to standard out 2. prints hi to standard out 3. prints [1.0 2.0 3.0] to standard out |
| 1 | The programing language **shall** have constructs that allow for a block of code to be executed or not executed based on a condition | 1. if true then print 'hello' 2. if false then print 'hello' | 1. prints 'hello' to standard out 2. does nothing |
| 2 | The programing language **shall** be able to store primitives in a variable | 1. box a = 1.0 2. print a 3. box a = [1.0 2.0 3.0] 4. print a 5. box a = "hello" 6. print a | 1. 1.0 is printed to standard out 2. [1.0 2.0 3.0] is printed to standard out 3. 'hello' is printed to standard out |
| 3 | The programing language **shall** be able to repeat blocks of code | 1. repeat 4 print 'hello ' | 1. standard out contains: 'hello hello hello hello' |
| 4 | The programing language **shall** be able to define and run functions with return values | 1. fun add x y = x + y 2. print (add 2.0 5.0) | 1. 7.0 is printed to standard out |
| 5 | The programing language **shall** be able to check if two floats are equal | 1. 2.5 == 2.5 2. 2.5 == 2.0 3. 2.0 == 3.0 | 1. returns true 2. returns false 3. returns false |
| 6 | The programing language **shall** be able to check if one float is greater than another | 1. 2.0 < 0.0 2. 2.0 < 3.0 3. 2.0 > 3.0 4. 2.0 > 2.0 | 1. returns false 2. returns true 3. returns false 4. returns false |
| 7 | The programing language **shall** be able to tell if one float is less than or equal to another | 1. 2.0 <= 3.0 2. 2.0 <= 1.0 3. 2.0 <= 2.0 | 1. returns true 2. returns false 3. returns true |
| 8 | The programing language **shall** be able to tell if one float is greater than or equal to another | 1. 2.0 >= 3.0 2. 2.0 >= 1.0 3. 2.0 >= 2.0 | 1. returns false 2. returns true 3. returns true |
| 9 | The programing language **shall** be able to tell if two lists are equal | 1. [] == [] 2. [1.0 2.0 3.0] == [1.0 2.0 3.0] 3. [1.0 10.0 3.0] == [1.0 2.0 3.0] 4. [] == [1.0 2.0 3.0] | 1. return true 2. return true 3. return false 4. return false |
| 10 | The programing language **shall** be able to tell if two strings are equal | 1. '' == '' 2. 'a' == 'a' 3. 'yellow' == 'hello' 4. 'yellow' == 'yellow' | 1. return true 2. return true 3. return false 4. return true |
| 11 | The programing language **shall** be able to negate a boolean statement | 1. not true 2. not false | 1. returns false 2. returns true |
| 12 | The programing language **shall** be able to connect two boolean statements with an 'and' operator | 1. false and false 2. false and true 3. true and false 4. true and true | 1. returns false 2. returns false 3. returns false 4. returns true |
| 13 | The programing language **shall** be able to connect two boolean expressions with a 'or' operator | 1. false or false 2. false or true 3. true or false 4. true or true | 1. returns false 2. returns true 3. returns true 4. returns true |
| 14 | The programing language **shall** be able to parse and execute the basic statement outlined in demonstration scenarios in a file in 1 second | 1. print 'hello' 2. print 2+2 | 1. the program prints 'hello' and exits within 1 second 2. the program prints '4' and exits within 1 second |
| 15 | The programing language **may** be able to step through the code line by line and see how variables change | 1. debugger run on a code file containing: 2. box a = 2 3. box a = a + 2 | 1. The debugger outputs: 2. box a = 2 3. box a = 4 |
| 16 | The programing language **shall** be able to add two floats together | 1. 2.0 + 3.0 2. 2.0 + -3.0 | 1. 5.0 2. -1.0 |
| 17 | The programing language **shall** be able to divide one float by another | 1. 6.0 / 3.0 2. 6.0 / -3.0 | 1. 2.0 2. -2.0 |
| 18 | The programing language **shall** be able to multiply one number by another | 1. 6.0 \* 2.0 2. 6.0 \* -2.0 | 1. 12.0 2. -12.0 |
| 19 | The programing language **shall** be able to subtract one float from another | 1. 6.0 - 2.0 2. 6.0 - -2.0 | 1. 4.0 2. 8.0 |