# LEDscript

Led script is a programming language designed to simplify the controlling of RGB led lights. Specially the manipulation of the colour values. The language will allow users to declare and set each light of a led array to a specific value. For this assignment, the language will focus on the manipulation of number. The led arrays will be printed to the console showing the values that each light has been set too. To output to hardware the print operator will needed to be edited to output the values instead of printing to console. Key features include being able to declare and edit multiple light arrays and specify which one should be displayed. Additional the ability to preview the colours that will be output will help speed up development.

The language is implemented as a java interpreter and follows the Turing machine computational model. LEDscript is an imperative based language. Programs should be written in an ordered fashion with each statement changing the programmes state. In addition to being imperative the language has one key object oriented feature. The ability to declare multiple arrays of led lights and work on each separately closely resembles features of an object-oriented language.

Syntax for LEDscript is based upon the syntax from C. slight changes have been made to simplify the users experience and allow programmes to be made by less experiences users.

|  |  |
| --- | --- |
| Function | Syntax |
| Assign | r=59;  or  int r = 59;  or  Colour(r=10;g=20;b=30); |
| Loop | for(int q=1;q<5;q = q + 5;) |
| If | if(r==50) |

## Semantics

Variables are mutable. Once declared a variable can be both redeclared with a new value or edited using several different operators. This is important as it will allow the values being given to the led lights to be changed easily.

There are 3 main types in LEDscript. The most important is integers these are declared in the same was as an integer in java. ”INT I = 10;”. Additional 3 integers can be declared at the same time using the prefix Colour. When declared integers will be checked to see if they are between 1 and 255. If a number is out of the boundary than it will be set to the nearest boundary value. This check will happen every time a integer value is edited or assigned other types can be declared without a prefix for example: name = ”kain”.

All literal and assigned variables will be evaluated as expressions. Identifiers will be deference to get the value they are storing and then evaluated in the same way. Additionally, code cannot be reused. As the language aims to be as simple as possible programmes will be written in a single file that completes tasks in the order they appear. This will keep source code easy to follow for users.

LEDscript implements value semantics. Identifiers are only used for their values. There are no references or pointers. This is again done to keep the language as simple to use as possible.

Any identifier will work in the statement it is declared and any statements that are inside that. Identifiers will however not work outside of the scope that they are declared.

To make the writing of applications as simple as possible there are many operators. The majority are semantic sugar designed to simplify the syntax and make the code more readable. The chart below outlines the operators, gives an example of the syntax and how describes the purpose of the operator.

|  |  |  |
| --- | --- | --- |
| Operator | Syntax | Description |
| Brighten | brighten(r,5;g,5;b,5);  or  brighten (r,5); | Adds the specified amount to the value given. Equivalent to += |
| Darken | darken(r,5;g,5;b,5);  or  darken (r,5); | subtracts the specified amount from the value given. Equivalent to -= |
| Divide | divide(r,5;g,5;b,5);  or  divide (r,5); | Divides the value by the number specified |
| Multiply | multiply(r,5;g,5;b,5);  or  multiply (r,5); | Multiplies the value by the number specified |
| Flip | flip(r;g;b); | Subtracts the value given from 255. This flips the value |
| Clear | Clear I; | Sets the value given to 0 |
| Fill | Fill I ; | Sets the value given to 255 |
| Delay | delay 1; | Waits the specified number of seconds. This is useful when changing what the led lights are showing. |
| Update | update (r;g;b;1;"ledarray"); | Finds the array with the name specified and sets the light specified to the input value, |
| Save | save "ledarray"; | Outputs the array specified to a text file |
| LEDS | leds 50;"ledarray" | Creates a LED array of the size and name specified |
| Info | info "ledarray"; | Prints the name of each colour to in the specified array to the console |
| Write | WRITE "ledarray" ; | Print the RGB values of the led array to the console |