

A guide to Keya

Krishna Sudhakar

1 Memory Layout

The operations of Keya is done on a grid layout. Standard Keya memory layout is the size of 20 by 20 bytes. This can be changed in the source code of the interpreter. The initial configuration is that the memory is initialized to 0, and the pointer points to the first cell of the first row, of the memory grid.

2 Commands

There are different kinds of commands in Keya. They are:

- Movement of the Pointer
- Input/Output
- Data Manipulation
- Loops

2.1 Movement of pointer

8		Move up by one row
2		Move down by one row
4		Move left by one cell
6		Move right by one cell
r		Reset to initial configuration

2.2 Input/Output

.		Output the data at the Pointer
,		Input data at Pointer

2.3 Data Manipulation

+		Increment byte at the Pointer
-		Decrement byte at the Pointer
>		Copy data to next cell on the right of the Pointer
<		Copy data to next cell on the left of the Pointer
^		Copy data to next cell above the Pointer
_		Copy data to next cell below the Pointer

2.4 Loops

[If the byte at the Pointer is zero, then instead of moving the instruction pointer forward to next command, jump it forward to the command after the next corresponding] command.
]		if the byte at the pointer is nonzero, then instead of moving the instruction pointer forward to the next command, jump it back to the command after the previous corresponding [command.

2.5 Comments

! | All the text right of ! is considered as comments, till the first occurrence of a new line character.

3 Directory Structure

3.1 src

keya.c

Source code of the interpreter in C.

keya.py

Source code of the interpreter in Python. [Current development suspended]

3.2 bin

Executable interpreter for Linux. Usage:

```
keya-linux <filename>.keya
```

3.3 Example

Hello World!.keya

This is an example program which outputs “Hello World!”

factorial.keya

The program prints the factorial of a number less than or equal to 5. Since the size of a cell is one byte, it can hold the value of at most 5!