LOLCODE

20 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

LOLCODE is an esoteric programming language inspired by lolspeak, the language expressed in examples of the lolcat Internet meme.[1] The language was created in 2007 by Adam Lindsay, a researcher at the Computing Department of Lancaster University.[2][3]

The language is not clearly defined in terms of operator priorities and correct syntax, but several functioning interpreters and compilers exist. One interpretation of the language has been proven Turing-complete.[4]

LOLCODE	
	
Paradigm	esoteric
Designed by	Adam Lindsay
First appeared	2007
Filename extensions	.lol, .lols
Website	lolcode.org
Influenced by	
Lolcats	

Language structure and examples [edit]

LOLCODE's keywords are drawn from the heavily compressed (shortened) patois of the lolcat Internet meme. Here follow a "Hello, World!" program and a simple program to output a file to a monitor.[5] Similar code was printed in the *Houston Chronicle*.[1]

- `:)` represents a newline (`\n`)
- `:>` represents a tab (`\t`)
- `:o` represents a bell character (`\a`)
- `:"` represents a literal double quote (`"`)
- `::` represents a single literal colon (`:`)
- `:(<hex>)` converts a single hexadecimal Unicode code point to local environment encoding (for example, UTF-8)
- `:{<variable>}` interpolates the value of the enclosed variable, cast as a string
- `:[<character name>]` converts normative name of a single Unicode character to local environment encoding

Example 1 [edit]

```
1 HAI 1.2
2 CAN HAS STDIO?
```

```
VISIBLE "HAI WORLD!"
KTHXBYE
```

Code	Comment
`HAI [VERSION]`	In all LOLCODE programs, `HAI` ("Hi!") introduces the program and specifies the version (although this isn't actually used yet).
`CAN HAS [LIBRARY]?`	In many programming languages, one of the first statements will be a library inclusion for common functions such as input and output. Typically this is included by a call such as `#include <stdio.h>` (stdio standing for standard input/output library). This command is a tongue-in-cheek corruption of that, asking if a library is obtainable, obtaining it if possible, and raising an exception if not.[6] It is there primarily for verisimilitude—in fact, it is ignored in current implementations of LOLCODE.
`VISIBLE [MESSAGE]`	Prints a message to the screen.
`KTHXBYE`	Just as `HAI` introduces the program, `KTHXBYE` (which is "K," "THX," and "Bye" all strung together, meaning "OK, thanks, bye") terminates it.
`BTW [MESSAGE]`	To write a single line comment in LOLCODE, you use the `BTW` keyword. Comments are ignored by the compiler and are written for better understanding of the program.
`OBTW [MESSAGE] TLDR`	Similar to the `BTW` keyword, the `OBTW` keyword marks a multiline comment, a comment that spans multiple lines. In LOLCODE, the `OBTW` keyword signifies the start of a multiline comment while the `TLDR` keyword ends it.

Example 2 [edit]

```
HAI 1.2
CAN HAS STDIO?
PLZ OPEN FILE "LOLCATS.TXT"?
    AWSUM THX
        VISIBLE FILE
    O NOES
        INVISIBLE "ERROR!"
KTHXBYE
```

[5]

In this example, commands to open a file (`PLZ OPEN FILE "NAME"?`—"Please open this file?"), and error handling (`AWSUM THX`—"Awesome, thanks!", and `O NOES`—"Oh no!") are introduced.

Example 3 [edit]

Other commands include `I HAS A variable` for declaring variables, `variable R value` ("variable [is/are/being] value") for assigning them, sending error messages to the front end via `INVISIBLE` instead of `VISIBLE`, and `BTW` ("by the way") to denote a comment, making the parser ignore the rest of the line.

Loops are created with `IM IN YR label` (inspired by the "Im in ur *noun*, *verb*ing yr *related noun*" LOLcat meme), and ended with `IM OUTTA YR label`. Loops can be broken with the keyword `ENUF` ("enough"), or in older versions, `GTFO`.[7] Loops can also be ended with the conditional `IZ` command, as demonstrated in the next example.

```
HAI 1.0
CAN HAS STDIO?
I HAS A VAR
IM IN YR LOOP
    UP VAR!!1
    VISIBLE VAR
    IZ VAR BIGGER THAN 10? KTHX
IM OUTTA YR LOOP
KTHXBYE
```

[5]

This simple program displays the numbers 1–11 and terminates (as of specification 1.0). The same program as of specification 1.2 is (assuming VAR starts at 0):

```

HAI 1.2
CAN HAS STDIO?
IM IN YR LOOP UPPIN YR VAR TIL BOTH SAEM VAR AN 10
    VISIBLE SUM OF VAR AN 1
IM OUTTA YR LOOP
KTHXBYE
```

Example 4 [edit]

```
HAI 1.0
CAN HAS STDIO?
VISIBLE "U SEE THIS"

BTW VISIBLE "U SEE NOTHING"
```

```
6
7  OBTW
8  VISIBLE "U SEE NOTHIN"
9  VISIBLE "U STIL SEE NOTHIN"
10 TLDR
11
12 VISIBLE "U SEE THIS"
13 KTHXBYE
```

The above example will return the following:

```
U SEE THIS
U SEE THIS
```

This is because line 3 outputs `U SEE THIS` but line 5 is ignored due to the fact that it is commented out by the `BTW` keyword. Lines 8 and 9 aren't run because they are in a multiline comment that starts in line 7, and ends on line 10. Line 12 outputs `U SEE THIS` and line 13 terminates the program.

Implementations [edit]

The most recent and up-to-date interpreter for the LOLCODE language is lci, written in C by Justin Meza. It interprets LOLCODE efficiently on a variety of platforms.[8]

The first LOLCODE implementation was a PHP parser written by Jeff Jones.[9][10] The parser's website was also the first website using LOLCODE as an actual web scripting language. Being open source with a BSD style licence, it has been forked and used by multiple websites to implement LOLCODE scripting. The winning Pecha Kucha presentation at PHP Works 2008 was about this parser.[11][12]

There is a .NET compiler for LOLCODE written by Nick Johnson,[13] and featured in Microsoft developer training seminars, TechEd 2007 Conference (Australia).[14][15][16]

PL/LOLCODE, a project headed by Josh Tolley, makes LOLCODE available as a server-side programming language inside PostgreSQL.[17]

Microsoft Dynamic Language Runtime has an implementation of LOLCODE for testing purposes.[18]

lolcode-java (A Java grammar / interpreter for the LOLCODE programming language) is a project also available[19] but it appears to not yet be compliant with the version 1.3 specification.

A LOLCODE to JavaScript translator is also available.[20]

There is also a LOLCODE compiler included with the Parrot virtual machine as one of the languages demonstrating the use of Parrot's compiler tools.[21]

A compiler, virtual machine and debugger, created by Piper, for a LoLCode like language, LoLCode 1337, written in C.[22]

A version for parallel and distributed computing can be found.[23]

Related projects [edit]

LOLCODE has also inspired LOLPython, written by Andrew Dalke. LOLPython uses LOL-inspired syntax similar to that of LOLCODE, but with a Python-like style. It operates by translating the LOLPython source into Python code.[24]

ArnoldC is an offshoot of LOLCODE that replaces lolspeak with quotes from different Arnold Schwarzenegger movies.[25]

References [edit]

1. ^ a b Dwight Silverman (2007-06-06). "I'm in ur newspaper writin mah colum" . Chron.com. Retrieved 2007-06-06.
2. ^ "Computing Department at Lancaster University – News" . Lancaster University, Computing Department. Archived from the original on 2009-05-28. Retrieved 2009-06-09.
3. ^ Hammock, Anne (2008-05-01). "The new fame: Internet celebrity" . *CNN*. Archived from the original on 2009-04-30.
4. ^ Arachnid. "Proof that LOLCode is turing complete:BrainF*** interpreter in LOLCode" . forum.lolcode.com. Archived from the original on 2007-07-11. Retrieved 2008-10-05.
5. ^ a b c Adam Lindsay (2007-05-25). "LOLCODE main page" . lolcode.com. Archived from the original on 2007-10-02. Retrieved 2007-10-02.
6. ^ "keywords:can-has · LOLCODE" . Archived from the original on 2008-02-06. Retrieved 2008-02-10.
7. ^ "SORN.net" . Archived from the original on February 16, 2009. Retrieved 2007-11-07.
8. ^ "LOLCODE + lci" .
9. ^ "First Parser Comment" . Lindsay.at. Archived from the original on 2009-05-19. Retrieved 2009-06-09.
10. ^ "Tetraboy's LOLCODE parser" . Tetraboy.com. Archived from the original on 2009-02-27. Retrieved 2009-06-09.
11. ^ " "The Best Web Language: LOLCODE" Slides" . Slideshare.net. 7 March 2009. Retrieved 2009-06-09.
12. ^ "PHP Works" . mtacon.com. Retrieved 4 November 2014.
13. ^ LOLCODE .NET compiler at Google Code
14. ^ TechEd Day 2: Microsoft announces LOLCode support , Long Zheng
15. ^ LOLcode in next Visual Studio? For young and funny cats , NetworkWorld.com
16. ^ "Video of LOLCODE presentation at TechEd 2007" . Blip.tv. 2007-08-17. Archived from the original on 2009-02-05. Retrieved 2009-06-09.
17. ^ PL/LOLCODE Archived 2010-11-28 at the Wayback Machine, pgFoundry
18. ^ Deep DLR , John Lam and Martin Maly
19. ^ "lolcode grammar interpreter written in Java" .
20. ^ "LolCode" . Fullvolume.co.uk. Retrieved 2009-06-09.
21. ^ "Lightning-Parrot" . lolcode.com. Archived from the original on 2009-03-22. Retrieved 2009-06-20.

22. ^ "LoLCode 1337" .
23. ^ "Parallel and Distributed Computing with LOLCODE" .
24. ^ "LOLPython" . Dalkescientific.com. 2007-06-01. Retrieved 2009-06-09.
25. ^ "ArnoldC" .

External links [edit]

- Official website

v · t · e **Esoteric programming languages** [hide]

Beatnik · Befunge · Brainfuck · FRACTRAN · INTERCAL · Iota and Jot · JSFuck · Leet · **LOLCODE** · Malbolge · One-instruction set computer · Piet · Shakespeare Programming Language · Unlambda · Whitespace

Category

Categories: Esoteric programming languages | Lancaster University | Programming languages created in 2007 | Science and technology in Lancashire