

TABLE OF CONTENTS

📖 FRONTMATTER

Dedication

Acknowledgements

I. WELCOME

1. Introduction

Design Note: What's in a Name?

2. A Map of the Territory

3. The Lox Language

Design Note: Expressions and Statements

II. A TREE-WALK INTERPRETER

4. Scanning

Design Note: Implicit Semicolons

5. Representing Code

6. Parsing Expressions

Design Note: Logic Versus History

7. Evaluating Expressions

Design Note: Static and Dynamic Typing

8. Statements and State

Design Note: Implicit Variable Declaration

9. Control Flow

Design Note: Spoonfuls of Syntactic Sugar

10. Functions

11. Resolving and Binding

12. Classes

Design Note: Prototypes and Power

13. Inheritance

III. A BYTECODE VIRTUAL MACHINE

14. Chunks of Bytecode

Design Note: Test Your Language

15. A Virtual Machine

Design Note: Register-Based Bytecode

16. Scanning on Demand

17. Compiling Expressions

Design Note: It's Just Parsing

18. Types of Values

19. Strings

Design Note: String Encoding

20. Hash Tables

21. Global Variables

22. Local Variables

23. Jumping Back and Forth

Design Note: Considering Goto Harmful

24. Calls and Functions

25. Closures

Design Note: Closing Over the Loop Variable

26. Garbage Collection

Design Note: Generational Collectors

27. Classes and Instances

28. Methods and Initializers

Design Note: Novelty Budget

29. Superclasses

30. Optimization

📖 BACKMATTER

A1. Appendix I: Lox Grammar

A2. Appendix II: Generated Syntax Tree Classes

Handcrafted by Robert Nystrom — © 2015–2020