

# CONTENTS

<b>Acknowledgments</b>	<b>3</b>
<b>Introduction</b>	<b>4</b>
The Monkey Programming Language & Interpreter . . . . .	5
Why Go? . . . . .	7
How to Use this Book . . . . .	8
<b>1 Lexing</b>	<b>9</b>
1.1 - Lexical Analysis . . . . .	9
1.2 - Defining Our Tokens . . . . .	10
1.3 - The Lexer . . . . .	12
1.4 - Extending our Token Set and Lexer . . . . .	20
1.5 - Start of a REPL . . . . .	25
<b>2 Parsing</b>	<b>28</b>
2.1 - Parsers . . . . .	28
2.2 - Why not a parser generator? . . . . .	30
2.3 - Writing a Parser for the Monkey Programming Language . . . . .	31
2.4 - Parser's first steps: parsing let statements . . . . .	32
2.5 - Parsing Return Statements . . . . .	44
2.6 - Parsing Expressions . . . . .	46
Expressions in Monkey . . . . .	47
Top Down Operator Precedence (or: Pratt Parsing) . . . . .	48
Terminology . . . . .	48
Preparing the AST . . . . .	49
Implementing the Pratt Parser . . . . .	52
Identifiers . . . . .	53
Integer Literals . . . . .	56
Prefix Operators . . . . .	59
Infix Operators . . . . .	63
2.7 - How Pratt Parsing Works . . . . .	69
2.8 - Extending the Parser . . . . .	77
Boolean Literals . . . . .	79
Grouped Expressions . . . . .	82
If Expressions . . . . .	84
Function Literals . . . . .	89
Call Expressions . . . . .	93
Removing TODOs . . . . .	97
2.9 - Read-Parse-Print-Loop . . . . .	99
<b>3 Evaluation</b>	<b>102</b>

3.1 - Giving Meaning to Symbols . . . . .	102
3.2 - Strategies of Evaluation . . . . .	103
3.3 - A Tree-Walking Interpreter . . . . .	104
3.4 - Representing Objects . . . . .	105
Foundation of our Object System . . . . .	107
Integers . . . . .	107
Booleans . . . . .	108
Null . . . . .	108
3.5 - Evaluating Expressions . . . . .	109
Integer Literals . . . . .	110
Completing the REPL . . . . .	112
Boolean Literals . . . . .	113
Null . . . . .	115
Prefix Expressions . . . . .	115
Infix Expressions . . . . .	119
3.6 - Conditionals . . . . .	124
3.7 - Return Statements . . . . .	126
3.8 - Abort! Abort! There's been a mistake!, or: Error Handling . . . . .	130
3.9 - Bindings & The Environment . . . . .	135
3.10 - Functions & Function Calls . . . . .	139
3.11 - Who's taking the trash out? . . . . .	149
<b>4 Extending the Interpreter</b> . . . . .	<b>151</b>
4.1 - Data Types & Functions . . . . .	151
4.2 - Strings . . . . .	151
Supporting Strings in our Lexer . . . . .	152
Parsing Strings . . . . .	154
Evaluating Strings . . . . .	155
String Concatenation . . . . .	157
4.3 - Built-in Functions . . . . .	159
len . . . . .	159
4.4 - Array . . . . .	163
Supporting Arrays in our Lexer . . . . .	164
Parsing Array Literals . . . . .	165
Parsing Index Operator Expressions . . . . .	168
Evaluating Array Literals . . . . .	171
Evaluating Index Operator Expressions . . . . .	173
Adding Built-in Functions for Arrays . . . . .	176
Test-Driving Arrays . . . . .	179
4.5 - Hashes . . . . .	180
Lexing Hash Literals . . . . .	181
Parsing Hash Literals . . . . .	183
Hashing Objects . . . . .	187
Evaluating Hash Literals . . . . .	191
Evaluating Index Expressions With Hashes . . . . .	194
4.6 - The Grand Finale . . . . .	196
<b>Resources</b> . . . . .	<b>199</b>
<b>Feedback</b> . . . . .	<b>201</b>
<b>Changelog</b> . . . . .	<b>202</b>