

Contents

About the Book xiii

I The Basics 1

1 Getting Started 3

- 1.1 Chunks 4
- 1.2 Some Lexical Conventions 5
- 1.3 Global Variables 6
- 1.4 Types and Values 7
- 1.5 The Stand-Alone Interpreter 9
- Exercises 10

2 Interlude: The Eight-Queen Puzzle 13

Exercises 15

3 Numbers 17

- 3.1 Numerals 17
- 3.2 Arithmetic Operators 19
- 3.3 Relational Operators 20
- 3.4 The Mathematical Library 21
- 3.5 Representation Limits 23
- 3.6 Conversions 24
- 3.7 Precedence 25
- 3.8 Lua Before Integers 26
- Exercises 27

Strings 29

- 4.1 Literal strings 30
- 4.2 Long strings 31
- 4.3 Coercions 32
- 4.4 The String Library 33
- 4.5 Unicode 36
- Exercises 38

Tables 39

- 5.1 Table Indices 40
- 5.2 Table Constructors 41
- 5.3 Arrays, Lists, and Sequences 43
- 5.4 Table Traversal 45
- 5.5 Safe Navigation 46
- 5.6 The Table Library 46
- Exercises 48

Functions 49

- 6.1 Multiple Results 50
- 6.2 Variadic Functions 53
- 6.3 The function `table.unpack` 55
- 6.4 Proper Tail Calls 56
- Exercises 57

The External World 59

- 7.1 The Simple I/O Model 59
- 7.2 The Complete I/O Model 62
- 7.3 Other Operations on Files 64
- 7.4 Other System Calls 64
- Exercises 66

Filling some Gaps 67

- 8.1 Local Variables and Blocks 67
- 8.2 Control Structures 69
- 8.3 **break**, **return**, and **goto** 72
- Exercises 74

Real Programming 77**Closures 79**

- 9.1 Functions as First-Class Values 79
- 9.2 Non-Global Functions 81
- 9.3 Lexical Scoping 83
- 9.4 A Taste of Functional Programming 85

Exercises 87**10 Pattern Matching 89**

- 10.1 The Pattern-Matching Functions 89
- 10.2 Patterns 91
- 10.3 Captures 95
- 10.4 Replacements 97
- 10.5 Tricks of the Trade 100
- Exercises 103

11 Interlude: Most Frequent Words 105

- Exercises 106

12 Date and Time 109

- 12.1 The Function `os.time` 109
- 12.2 The Function `os.date` 110
- 12.3 Date-Time Manipulation 111
- Exercises 114

13 Bits and Bytes 115

- 13.1 Bitwise Operators 115
- 13.2 Unsigned Integers 116
- 13.3 Packing and Unpacking Binary Data 118
- 13.4 Binary files 121
- Exercises 122

14 Data Structures 125

- 14.1 Arrays 125
- 14.2 Matrices and Multi-Dimensional Arrays 126
- 14.3 Linked Lists 128
- 14.4 Queues and Double-Ended Queues 129
- 14.5 Reverse Tables 129
- 14.6 Sets and Bags 131
- 14.7 String Buffers 132
- 14.8 Graphs 133
- Exercises 135

15 Data Files and Serialization 137

- 15.1 Data Files 138
- 15.2 Serialization 139
- Exercises 145

16 Compilation, Execution, and Errors 147

- 16.1 Compilation 147
- 16.2 Precompiled Code 151

- 6.3 Errors 152
- 6.4 Error Handling and Exceptions 154
- 6.5 Error Messages and Tracebacks 154
- Exercises 156

Modules and Packages 159

- 17.1 The Function require 160
- 17.2 The Basic Approach for Writing Modules in Lua 164
- 17.3 Submodules and Packages 165
- Exercises 168

Lua-isms 169

Iterators and the Generic for 171

- 18.1 Iterators and Closures 171
- 18.2 The Semantics of the Generic for 173
- 18.3 Stateless Iterators 174
- 18.4 Traversing Tables in Order 176
- 18.5 True Iterators 177
- Exercises 178

Interlude: Markov Chain Algorithm 181

- Exercises 182

Metatables and Metamethods 185

- 20.1 Arithmetic Metamethods 186
- 20.2 Relational Metamethods 189
- 20.3 Library-Defined Metamethods 190
- 20.4 Table-Access Metamethods 190
- Exercises 196

Object-Oriented Programming 197

- 21.1 Classes 198
- 21.2 Inheritance 200
- 21.3 Multiple Inheritance 202
- 21.4 Privacy 204
- 21.5 The Single-Method Approach 206
- 21.6 Dual Representation 207
- Exercises 208

The Environment 211

- 22.1 Global Variables with Dynamic Names 211
- 22.2 Global-Variable Declarations 213
- 22.3 Non-Global Environments 214

- 22.4 Using _ENV 217
- 22.5 Environments and Modules 219
- 22.6 _ENV and load 220
- Exercises 221

23 Garbage 223

- 23.1 Weak Tables 223
- 23.2 Memorize Functions 225
- 23.3 Object Attributes 227
- 23.4 Revisiting Tables with Default Values 227
- 23.5 Ephemeron Tables 228
- 23.6 Finalizers 229
- 23.7 The Garbage Collector 232
- 23.8 Controlling the Pace of Collection 233
- Exercises 234

24 Coroutines 237

- 24.1 Coroutine Basics 237
- 24.2 Who Is the Boss? 240
- 24.3 Coroutines as Iterators 241
- 24.4 Event-Driven Programming 245
- Exercises 249

25 Reflection 251

- 25.1 Introspective Facilities 251
- 25.2 Hooks 257
- 25.3 Profiles 258
- 25.4 Sandboxing 260
- Exercises 264

26 Interlude: Multithreading with Coroutines 265

- Exercises 269

IV The C API 271

27 An Overview of the C API 273

- 27.1 A First Example 274
- 27.2 The Stack 276
- 27.3 Error Handling with the C API 282
- 27.4 Memory Allocation 285
- Exercises 286

28 Extending Your Application 289

- 28.1 The Basics 289

	28.2 Table Manipulation	291
	28.3 Calling Lua Functions	295
	28.4 A Generic Call Function	297
	Exercises	298
29	Calling C from Lua	301
	29.1 C Functions	301
	29.2 Continuations	304
	29.3 C Modules	306
	Exercises	308
30	Techniques for Writing C Functions	309
	30.1 Array Manipulation	309
	30.2 String Manipulation	311
	30.3 Storing State in C Functions	313
	Exercises	320
31	User-Defined Types in C	321
	31.1 Userdata	322
	31.2 Metatables	324
	31.3 Object-Oriented Access	327
	31.4 Array Access	328
	31.5 Light Userdata	329
	Exercises	330
32	Managing Resources	333
	32.1 A Directory Iterator	333
	32.2 An XML Parser	337
	Exercises	344
33	Threads and States	347
	33.1 Multiple Threads	347
	33.2 Lua States	351
	Exercises	359
	Index	361