

# CONTENTS

<b>Acknowledgments</b>	<b>3</b>
<b>Introduction</b>	<b>4</b>
Evolving Monkey . . . . .	5
The Past and Present . . . . .	5
The Future . . . . .	7
Use This Book . . . . .	8
<b>1 Compilers &amp; Virtual Machines</b>	<b>10</b>
Compilers . . . . .	11
Virtual and Real Machines . . . . .	14
Real Machines . . . . .	14
What Is a Virtual Machine? . . . . .	19
Why Build One? . . . . .	21
Bytecode . . . . .	23
What We're Going to Do, or: the Duality of VM and Compiler . . . . .	24
<b>2 Hello Bytecode!</b>	<b>26</b>
First Instructions . . . . .	27
Starting With Bytes . . . . .	28
The Smallest Compiler . . . . .	32
Bytecode, Disassemble! . . . . .	37
Back to the Task at Hand . . . . .	41
Powering On the Machine . . . . .	43
Adding on the Stack . . . . .	49
Hooking up the REPL . . . . .	55
<b>3 Compiling Expressions</b>	<b>57</b>
Cleaning Up the Stack . . . . .	57
Infix Expressions . . . . .	61
Booleans . . . . .	65
Comparison Operators . . . . .	69
Prefix Expressions . . . . .	76
<b>4 Conditionals</b>	<b>82</b>
Jumps . . . . .	85
Compiling Conditionals . . . . .	87
Executing Jumps . . . . .	100
Welcome Back, Null! . . . . .	103
<b>5 Keeping Track of Names</b>	<b>110</b>

The Plan . . . . .	111
Compiling Bindings . . . . .	112
Introducing: the Symbol Table . . . . .	114
Using Symbols in the Compiler . . . . .	117
Adding Globals to the VM . . . . .	120
<b>6 String, Array and Hash</b>	<b>125</b>
String . . . . .	125
Array . . . . .	130
Hash . . . . .	134
Adding the index operator . . . . .	140
<b>7 Functions</b>	<b>146</b>
Dipping Our Toes: a Simple Function . . . . .	146
Representing Functions . . . . .	147
Opcodes to Execute Functions . . . . .	148
Compiling Function Literals . . . . .	151
Compiling Function Calls . . . . .	164
Functions in the VM . . . . .	166
A Little Bonus . . . . .	176
Local Bindings . . . . .	176
Opcodes for Local Bindings . . . . .	177
Compiling Locals . . . . .	180
Implementing Local Bindings in the VM . . . . .	191
Arguments . . . . .	200
Compiling Calls With Arguments . . . . .	201
Resolving References to Arguments . . . . .	205
Arguments in the VM . . . . .	208
<b>8 Built-in Functions</b>	<b>216</b>
Making the Change Easy . . . . .	217
Making the Change: the Plan . . . . .	222
A New Scope for Built-in Functions . . . . .	223
Executing built-in functions . . . . .	227
<b>9 Closures</b>	<b>233</b>
The Problem . . . . .	234
The Plan . . . . .	235
Everything's a closure . . . . .	236
Compiling and resolving free variables . . . . .	246
Creating real closures at run time . . . . .	257
<b>10 Taking Time</b>	<b>262</b>
<b>Resources</b>	<b>267</b>
<b>Feedback</b>	<b>270</b>
<b>Changelog</b>	<b>271</b>