

CONTENTS IN DETAIL

FOREWORD	xv
-----------------	-----------

ACKNOWLEDGMENTS	xvii
------------------------	-------------

INTRODUCTION	xix
---------------------	------------

Who Should Read This Book	xix
Why Users Are Interested in WebAssembly	xx
Why the World Needs WebAssembly	xxi
What's in This Book	xxi

1	
AN INTRODUCTION TO WEBASSEMBLY	1

What Is WebAssembly?	2
Reasons to Use WebAssembly	3
Better Performance	3
Integrating Legacy Libraries	4
Portability and Security	4
JavaScript Skeptics	4
WebAssembly's Relationship with JavaScript	5
Why Learn WAT?	6
WAT Coding Styles	7
The Embedding Environment	11
The Browser	11
WASI	11
Visual Studio Code	12
Node.js	12
Our First Node.js WebAssembly App	14
Calling the WebAssembly Module from Node.js	15
The .then Syntax	16
The Time Is Now	17

2	
WEBASSEMBLY TEXT BASICS	19

Writing the Simplest Module	20
Hello World in WebAssembly	20
Creating Our WAT Module	21
Creating the JavaScript File	23
WAT Variables	25
Global Variables and Type Conversion	25
Local Variables	29
Unpacking S-Expressions	30
Indexed Variables	32
Converting Between Types	32
if/else Conditional Logic	34

Loops and Blocks	37
The block Statement.	37
The loop Expression	38
Using block and loop Together	39
Branching with br_table	42
Summary	43

3 FUNCTIONS AND TABLES 45

When to Call Functions from WAT	46
Writing an is_prime Function	46
Passing Parameters	46
Creating Internal Functions	47
Adding the is_prime Function	49
The JavaScript	52
Declaring an Imported Function	54
JavaScript Numbers.	55
Passing Data Types	55
Objects in WAT	55
Performance Implications of External Function Calls	55
Function Tables	59
Creating a Function Table in WAT	59
Summary	67

4 LOW-LEVEL BIT MANIPULATION 69

Binary, Decimal, and Hexadecimal	70
Integers and Floating-Point Arithmetic	71
Integers	71
Floating-Point Numbers	73
High- and Low-Order Bits	77
Bit Operations	78
Shifting and Rotating Bits	79
Masking Bits with AND and OR	80
XOR Bit Flip	83
Big-Endian vs. Little-Endian	84
Summary	85

5 STRINGS IN WEBASSEMBLY 87

ASCII and Unicode	88
Strings in Linear Memory	88
Passing the String Length to JavaScript.	88
Null-Terminated Strings	90
Length-Prefixed Strings	93
Copying Strings	95
Creating Number Strings	101
Setting a Hexadecimal String	105
Setting a Binary String	110
Summary	113

6		
LINEAR MEMORY		115
Linear Memory in WebAssembly		116
Pages		117
Pointers		119
JavaScript Memory Object		121
Creating the WebAssembly Memory Object		121
Logging to the Console with Colors		122
Creating the JavaScript in store_data.js		123
Collision Detection		125
Base Address, Stride, and Offset		126
Loading Data Structures from JavaScript		127
Displaying the Results		129
Collision Detection Function		130
Summary		138
7		
WEB APPLICATIONS		139
The DOM		140
Setting Up a Simple Node Server		140
Our First WebAssembly Web Application		142
Defining the HTML Header		143
The JavaScript		143
The HTML Body		145
Our Completed Web App		146
Hex and Binary Strings		147
The HTML		147
The WAT		150
Compile and Run		154
Summary		156
8		
WORKING WITH THE CANVAS		157
Rendering to the Canvas		158
Defining the Canvas in HTML		158
Defining JavaScript Constants in HTML		159
Creating Random Objects		161
Bitmap Image Data		162
The requestAnimationFrame Function		163
The WAT Module		164
Imported Values		164
Clearing the Canvas		165
Absolute Value Function		166
Setting a Pixel Color		167
Drawing the Object		169
Setting and Getting Object Attributes		171
The \$main Function		173
Compiling and Running the App		183
Summary		184

9		
OPTIMIZING PERFORMANCE		185
Using a Profiler		186
Chrome Profiler.		186
Firefox Profiler		192
wasm-opt		196
Installing Binaryen.		196
Running wasm-opt.		196
Looking at Optimized WAT Code		198
Strategies for Improving Performance		199
Inlining Functions		199
Multiply and Divide vs. Shift		202
DCE		204
Comparing the Collision Detection App with JavaScript		205
Hand Optimizing WAT		208
Logging Performance		209
More Sophisticated Testing with benchmark.js		213
Comparing WebAssembly and JavaScript with <code>-print-bytecode</code>		219
Summary		222
10		
DEBUGGING WEBASSEMBLY		223
Debugging from the Console		224
Logging Messages to the Console		229
Using Alerts.		232
Stack Trace		233
The Firefox Debugger		238
The Chrome Debugger		243
Summary		245
11		
ASSEMBLYSCRIPT		247
AssemblyScript CLI		248
Hello World AssemblyScript		249
JavaScript for Our Hello World App		251
Hello World with the AssemblyScript Loader.		253
AssemblyScript String Concatenation.		254
Object Oriented Programming in AssemblyScript		256
Using Private Attributes		258
JavaScript Embedding Environment		259
AssemblyScript Loader.		260
Extending Classes in AssemblyScript		263
Performance of Loader vs. Direct WebAssembly Calls		264
Summary		267
FINAL THOUGHTS		269
INDEX		271