

Learn You Some Erlang for Great Good!

A Beginner's Guide



Fred Hébert

Foreword by Joe Armstrong



CONTENTS IN DETAIL

ABOUT THE AUTHOR **xvii**

FOREWORD by Joe Armstrong **xix**

PREFACE **xxi**

To the Foreigner xxi
To the Erlang Regular xxii
To the Person Who Has Read This Online xxii

ACKNOWLEDGMENTS **xxiii**

INTRODUCTION **1**

So What's Erlang? 2
Don't Drink Too Much Kool-Aid 4
What You Need to Dive In 5
Where to Get Help 6

1
STARTING OUT **7**

Using the Erlang Shell 7
 Entering Shell Commands 8
 Exiting the Shell 8
Some Erlang Basics 10
 Numbers 10
 Invariable Variables 11
 Atoms 12
 Boolean Algebra and Comparison Operators 14
 Tuples 16
 Lists 18
 List Comprehensions 21
Working with Binary Data 23
 Bit Syntax 23
 Bitwise Binary Operations 26
 Binary Strings 27
 Binary Comprehensions 28

2
MODULES **31**

What Are Modules? 31
Creating Modules 32

Compiling Code	36
Compiler Options	37
Defining Macros	38
More About Modules	40
Metadata	40
Circular Dependencies.	41

3 SYNTAX IN FUNCTIONS 43

Pattern Matching	43
Fancier Patterns	45
Variables in a Bind	46
Guards, Guards!	48
What the If?!	49
In case ... of	52
Which Should We Use?	54

4 TYPES (OR LACK THEREOF) 55

Dynamite-Strong Typing	55
Type Conversions.	57
To Guard a Data Type	58
For Type Junkies.	60

5 HELLO RECURSION! 61

How Recursion Works	62
Length of a List	63
Length of a Tail Recursion.	64
More Recursive Functions	66
A Duplicate Function	66
A Reverse Function	66
A Sublist Function	68
A Zip Function	69
Quick, Sort!	70
More Than Lists	72
Thinking Recursively	75

6 HIGHER-ORDER FUNCTIONS 77

Let's Get Functional	77
Anonymous Functions	79
More Anonymous Function Power	80
Function Scope and Closures	81
Maps, Filters, Folds, and More	83
Filters.	83
Fold Everything	84
More Abstractions.	86

7		
ERRORS AND EXCEPTIONS		87
A Compilation of Errors		88
Compile-Time Errors		88
No, YOUR Logic Is Wrong!		89
Runtime Errors		90
Raising Exceptions		93
Error Exceptions		93
Exit Exceptions		94
Throw Exceptions		95
Dealing with Exceptions		96
Handling Different Types of Exceptions		96
After the Catch		99
Trying Multiple Expressions		99
Wait, There's More!		100
Try a try in a Tree.		103
8		
FUNCTIONALLY SOLVING PROBLEMS		105
Reverse Polish Notation Calculator		106
How RPN Calculators Work		106
Creating an RPN Calculator		107
Testing the Code		110
Heathrow to London		111
Solving the Problem Recursively		112
Writing the Code		114
Running the Program Without the Erlang Shell		118
9		
A SHORT VISIT TO COMMON DATA STRUCTURES		121
Records		122
Defining Records		122
Reading Values from Records		123
Updating Records		125
Sharing Records		126
Key/Value Stores		127
Stores for Small Amounts of Data		127
Larger Dictionaries: Dicts and GB Trees		128
A Set of Sets		130
Directed Graphs		131
Queues		132
End of the Short Visit		133
10		
THE HITCHHIKER'S GUIDE TO CONCURRENCY		135
Don't Panic		136
Concurrency Concepts		137
Scalability		137

Fault Tolerance	138
Concurrency Implementation	140
Not Entirely Unlike Linear Scaling	140
So Long and Thanks for All the Fish!	142
Spawning Processes	142
Sending Messages	144
Receiving Messages	145

11 MORE ON MULTIPROCESSING 149

State Your State	150
We Love Messages, But We Keep Them Secret	152
Time Out	153
Selective Receives	156
The Pitfalls of Selective Receives	156
More Mailbox Pitfalls	159

12 ERRORS AND PROCESSES 161

Links	162
It's a Trap!	164
Old Exceptions, New Concepts	165
Monitors	168
Naming Processes	170

13 DESIGNING A CONCURRENT APPLICATION 175

Understanding the Problem	176
Defining the Protocol	178
Lay Them Foundations	180
An Event Module	181
Events and Loops	181
Adding An Interface	184
The Event Server	186
Handling Messages	188
Hot Code Loving	191
I Said, Hide Your Messages	192
A Test Drive	194
Adding Supervision	195
Namespaces (or Lack Thereof)	196

14 AN INTRODUCTION TO OTP 199

The Common Process, Abstracted	200
The Basic Server	201
Introducing the Kitty Server	201
Generalizing Calls	203

Generalizing the Server Loop	204
Starter Functions	206
Generalizing Kitty Server	207
Specific vs. Generic	209
Callback to the Future	210
The init Function	210
The handle_call Function	211
The handle_cast Function	212
The handle_info Function	212
The terminate Function	212
The code_change Function	213
.BEAM Me Up, Scotty!	213

15

RAGE AGAINST THE FINITE-STATE MACHINES **219**

What Is a Finite-State Machine?	220
Generic Finite-State Machines	223
The init Function	223
The StateName Function	224
The handle_event Function	225
The handle_sync_event Function	225
The code_change and terminate Functions	225
A Trading System Specification	225
Show Me Your Moves	226
Defining the State Diagrams and Transitions	227
Game Trading Between Two Players	233
The Public Interface	233
FSM-to-FSM Functions	235
The gen_fsm Callbacks	236
That Was Really Something	245
Fit for the Real World?	246

16

EVENT HANDLERS **247**

Handle This! *pumps shotgun*	248
Generic Event Handlers	249
The init and terminate Functions	250
The handle_event Function	250
The handle_call Function	251
The handle_info Function	251
The code_change Function	251
It's Curling Time!	252
The Scoreboard	252
Game Events	253
Alert the Press!	257

17		
WHO SUPERVISES THE SUPERVISORS?		263
Supervisor Concepts		264
Using Supervisors		266
Restart Strategies		266
Restart Limits		268
Child Specifications		268
Band Practice		271
Musicians		271
Band Supervisor		274
Dynamic Supervision		277
Using Standard Supervisors Dynamically		277
Using a simple_one_for_one Supervisor		279
18		
BUILDING AN APPLICATION		281
A Pool of Processes		282
The Onion Layer Theory		283
A Pool's Tree		284
Implementing the Supervisors		286
Working on the Workers		290
Writing a Worker		296
Run Pool Run		298
Cleaning the Pool		301
19		
BUILDING APPLICATIONS THE OTP WAY		303
My Other Car Is a Pool		304
The Application Resource File		305
Converting the Pool		307
The Application Behavior		309
From Chaos to Application		310
Library Applications		314
20		
THE COUNT OF APPLICATIONS		315
From OTP Application to Real Application		316
The Application File		317
The Application Callback Module and Supervisor		318
The Dispatcher		319
The Counter		329
Run App Run		331
Included Applications		333
Complex Terminations		333

21		
RELEASE IS THE WORD		335
Fixing the Leaky Pipes		336
Terminating the VM		336
Updating the Application Files		336
Compiling the Applications		337
Releases with systools		338
Creating a Boot File		339
Packaging the Release		340
Releases with Reltool		341
Reltool Options		345
Reltool Recipes		348
Released from Releases		352
22		
LEVELING UP IN THE PROCESS QUEST		353
The Hiccups of Appups and Relups		354
The Ninth Circle of Erl		356
Process Quest		357
The regis-1.0.0 Application		358
The processquest-1.0.0 Application		359
The sockserv-1.0.0 Application		360
The Release		360
Making Process Quest Better		363
Updating code_change Functions		363
Adding Appup Files		365
Upgrading the Release		369
Relup Review		372
23		
BUCKETS OF SOCKETS		375
IO Lists		375
UDP and TCP: Bro-tocols		377
UDP Sockets		379
TCP Sockets		381
More Control with Inet		384
Sockserv, Revisited		387
Where to Go from Here?		396
24		
EUNITED NATIONS COUNCIL		397
EUnit—What’s an EUnit?		398
Test Generators		402
Fixtures		404
More Test Control		406
Test Documentation		407
Testing Regis		408
He Who Knits EUnits		417

25

BEARS, ETS, BEETS: IN-MEMORY NOSQL FOR FREE! 419

Why ETS.	420
The Concepts of ETS.	421
ETS Phone Home	423
Creating and Deleting Tables	423
Inserting and Looking Up Data	425
Meeting Your Match.	427
You Have Been Selected.	428
DETS	433
A Little Less Conversation, a Little More Action, Please	434
The Interface.	434
Implementation Details.	435

26

DISTRIBUNOMICON 441

This Is My Boomstick.	442
Fallacies of Distributed Computing	445
The Network Is Reliable	445
There Is No Latency.	446
Bandwidth Is Infinite	446
The Network Is Secure.	447
Topology Doesn't Change	448
There Is Only One Administrator.	449
Transport Cost Is Zero	449
The Network Is Homogeneous	450
Fallacies in a Nutshell	451
Dead or Dead-Alive	451
My Other Cap Is a Theorem	453
Consistency	453
Availability.	453
Partition Tolerance	454
Zombie Survivors and CAP.	454
Setting Up an Erlang Cluster	458
Through the Desert on a Node with No Name	458
Connecting Nodes	459
More Tools.	460
Cookies	462
Remote Shells	464
Hidden Nodes.	465
The Walls Are Made of Fire, and the Goggles Do Nothing	466
The Calls from Beyond	467
The net_kernel Module.	467
The global Module	467
The rpc Module	469
Burying the Distribunomicon	471

27		
DISTRIBUTED OTP APPLICATIONS		473
Adding More to OTP		474
Taking and Failing Over		475
The Magic 8 Ball		476
Building the Application		477
Making the Application Distributed		480
28		
COMMON TEST FOR UNCOMMON TESTS		485
What Is Common Test?		485
Common Test Structure		486
Creating a Simple Test Suite		488
Running the Tests		489
Testing with State		491
Test Groups		493
Defining Test Groups		494
Test Group Properties		495
The Meeting Room		496
Test Suites Redux		500
Test Specifications		501
Specification File Contents		501
Creating a Spec File		503
Running Tests with a Spec File		503
Large-Scale Testing		504
Creating a Distributed Spec File		506
Running Distributed Tests		507
Integrating EUnit Within Common Test		508
Is There More?		509
29		
MNESIA AND THE ART OF REMEMBERING		511
What's Mnesia?		512
What Should the Store Store?		513
The Data to Store		513
Table Structure		514
From Record to Table		515
Of Mnesia Schemas and Tables		516
Creating Tables		519
Installing the Database		519
Starting the Application		522
Access and Context		523
Reads, Writes, and More		524
Implementing the First Requests		526
A Test for Adding Services		526
Tests for Lookups		529
Accounts and New Needs		532

Meet the Boss	534
Deleting Stuff, Demonstrated	536
Query List Comprehensions	539
Remember Mnesia	541

30
TYPE SPECIFICATIONS AND DIALYZER **543**

PLTs Are the Best Sandwiches	543
Success Typing	545
Type Inference and Discrepancies	547
Typing About Types of Types	550
Singleton Types	550
Union and Built-in Types	551
Defining Types	554
Types for Records	555
Typing Functions	556
Typing Practice	560
Exporting Types	564
Typed Behaviors	566
Polymorphic Types	567
We Bought a Zoo	568
Some Cautions	570
You're My Type	572
That's All, Folks	572

AFTERWORD **573**

Other Erlang Applications	574
Community Libraries	575
Your Ideas Are Intriguing to Me and I Wish to Subscribe to Your Newsletter	576
Is That It?	576

APPENDIX
ON ERLANG'S SYNTAX **577**

The Template	577
The English Sentence	579
And, Or, Done	580
In Conclusion	580

INDEX **581**