

The
Pragmatic
Programmers

Programming Erlang

Software for a
Concurrent World



Joe Armstrong

Extracted from:

Programming Erlang

Software for a Concurrent World

This PDF file contains pages extracted from Programming Erlang, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragmaticprogrammer.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

Copyright © 2007The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Contents

1	Begin	14
1.1	Road Map	15
1.2	Begin Again	18
1.3	Acknowledgments	19
2	Getting Started	21
2.1	Overview	21
2.2	Installing Erlang	24
2.3	The Code in This Book	26
2.4	Starting the Shell	27
2.5	Simple Integer Arithmetic	29
2.6	Variables	30
2.7	Floating-Point Numbers	36
2.8	Atoms	36
2.9	Tuples	38
2.10	Lists	41
2.11	Strings	43
2.12	Pattern Matching Again	44
3	Sequential Programming	46
3.1	Modules	46
3.2	Back to Shopping	51
3.3	Functions with the Same Name and Different Arity	55
3.4	Funs	55
3.5	Simple List Processing	61
3.6	List Comprehensions	64
3.7	Arithmetic Expressions	68
3.8	Guards	68
3.9	Records	73
3.10	case and if Expressions	75
3.11	Building Lists in Natural Order	77
3.12	Accumulators	78

4	Exceptions	80
4.1	Exceptions	80
4.2	Raising an Exception	81
4.3	try...catch	82
4.4	catch	85
4.5	Improving Error Messages	86
4.6	Programming Style with try...catch	86
4.7	Catching Every Possible Exception	87
4.8	Old- and New-Style Exception Handling	88
4.9	Stack Traces	88
5	Advanced Sequential Programming	90
5.1	BIFs	90
5.2	Binaries	91
5.3	The Bit Syntax	93
5.4	Miscellaneous Short Topics	102
6	Compiling and Running Your Program	123
6.1	Starting and Stopping the Erlang Shell	123
6.2	Modifying the Development Environment	124
6.3	Different Ways to Run Your Program	126
6.4	Automating Compilation with Makefiles	132
6.5	Command Editing in the Erlang Shell	135
6.6	Getting Out of Trouble	135
6.7	When Things Go Wrong	136
6.8	Getting Help	139
6.9	Tweaking the Environment	139
6.10	The Crash Dump	140
7	Concurrency	142
8	Concurrent Programming	146
8.1	The Concurrency Primitives	147
8.2	A Simple Example	148
8.3	Client-Server—An Introduction	149
8.4	How Long Does It Take to Create a Process?	153
8.5	Receive with a Timeout	155
8.6	Selective Receive	158
8.7	Registered Processes	159
8.8	How Do We Write a Concurrent Program?	161
8.9	A Word About Tail Recursion	162
8.10	Spawning with MFAs	163

8.11	Problems	163
9	Errors in Concurrent Programs	164
9.1	Linking Processes	164
9.2	An <code>on_exit</code> Handler	165
9.3	Remote Handling of Errors	167
9.4	The Details of Error Handling	167
9.5	Error Handling Primitives	175
9.6	Sets of Linked Processes	176
9.7	Monitors	177
9.8	A Keep-Alive Process	178
10	Distributed Programming	180
10.1	The Name Server	182
10.2	The Distribution Primitives	187
10.3	Libraries for Distributed Programming	190
10.4	The Cookie Protection System	191
10.5	Socket-Based Distribution	192
11	IRC Lite	196
11.1	Message Sequence Diagrams	198
11.2	The User Interface	199
11.3	Client-Side Software	200
11.4	Server-Side Software	204
11.5	Running the Application	207
11.6	The Chat Program Source Code	209
11.7	Exercises	216
12	Interfacing Techniques	217
12.1	Ports	218
12.2	Interfacing an External C Program	219
12.3	<code>open_port</code>	225
12.4	Linked-in Drivers	226
12.5	Notes	230
13	Programming with Files	231
13.1	Organization of the Libraries	231
13.2	The Different Ways of Reading a File	232
13.3	The Different Ways of Writing to a File	239
13.4	Directory Operations	244
13.5	Finding Information About a File	245
13.6	Copying and Deleting Files	246

13.7	Bits and Pieces	246
13.8	A Find Utility	247
14	Programming with Sockets	250
14.1	Using TCP	251
14.2	Control Issues	260
14.3	Where Did That Connection Come From?	263
14.4	Error Handling with Sockets	263
14.5	UDP	265
14.6	Broadcasting to Multiple Machines	268
14.7	A SHOUTcast Server	269
14.8	Digging Deeper	277
15	ETS and DETS: Large Data Storage Mechanisms	278
15.1	Basic Operations on Tables	279
15.2	Types of Table	280
15.3	ETS Table Efficiency Considerations	281
15.4	Creating an ETS Table	282
15.5	Example Programs with ETS	284
15.6	DETS	289
15.7	What Haven't We Talked About?	292
15.8	Code Listings	293
16	OTP Introduction	296
16.1	The Road to the Generic Server	297
16.2	Getting Started with <code>gen_server</code>	306
16.3	The <code>gen_server</code> Callback Structure	310
16.4	Code and Templates	313
16.5	Digging Deeper	316
17	Mnesia: The Erlang Database	318
17.1	Database Queries	318
17.2	Adding and Removing Data in the Database	322
17.3	Mnesia Transactions	324
17.4	Storing Complex Data in Tables	328
17.5	Table Types and Location	330
17.6	Creating the Initial Database	333
17.7	The Table Viewer	334
17.8	Digging Deeper	335
17.9	Listings	336

18 Making a System with OTP	340
18.1 Generic Event Handling	341
18.2 The Error Logger	344
18.3 Alarm Management	351
18.4 The Application Servers	353
18.5 The Supervision Tree	356
18.6 Starting the System	359
18.7 The Application	363
18.8 File System Organization	365
18.9 The Application Monitor	366
18.10 Digging Deeper	366
18.11 How Did We Make That Prime?	367
19 Multicore Prelude	371
20 Programming Multicore CPUs	373
20.1 How to Make Programs Run Efficiently on a Multicore CPU	374
20.2 Parallelizing Sequential Code	378
20.3 Small Messages, Big Computations	381
20.4 mapreduce and Indexing Our Disk	385
20.5 Growing Into the Future	395
A Documenting Our Program	396
A.1 Erlang Type Notation	397
A.2 Tools That Use Types	400
B Erlang on Microsoft Windows	402
B.1 Erlang	402
B.2 Fetch and Install MinGW	402
B.3 Fetch and Install MSYS	403
B.4 Install the MSYS Developer Toolkit (Optional)	403
B.5 Emacs	403
C Resources	405
C.1 Online Documentation	405
C.2 Books and Theses	406
C.3 Link Collections	406
C.4 Blogs	406
C.5 Forums, Online Communities, and Social Sites	407
C.6 Conferences	407
C.7 Projects	407
C.8 Bibliography	408

D	A Socket Application	409
D.1	An Example	409
D.2	How lib_chan Works	412
D.3	The lib_chan Code	415
E	Miscellaneous	425
E.1	Analysis and Profiling Tools	425
E.2	Debugging	428
E.3	Tracing	437
E.4	Dynamic Code Loading	441
F	Module and Function Reference	445
F.1	Module: application	445
F.2	Module: base64	446
F.3	Module: beam_lib	447
F.4	Module: c	447
F.5	Module: calendar	449
F.6	Module: code	450
F.7	Module: dets	451
F.8	Module: dict	454
F.9	Module: digraph	455
F.10	Module: digraph_utils	456
F.11	Module: disk_log	457
F.12	Module: epp	458
F.13	Module: erl_eval	458
F.14	Module: erl_parse	459
F.15	Module: erl_pp	459
F.16	Module: erl_scan	460
F.17	Module: erl_tar	460
F.18	Module: erlang	461
F.19	Module: error_handler	470
F.20	Module: error_logger	470
F.21	Module: ets	471
F.22	Module: file	474
F.23	Module: file_sorter	476
F.24	Module: filelib	477
F.25	Module: filename	477
F.26	Module: gb_sets	478
F.27	Module: gb_trees	480
F.28	Module: gen_event	481
F.29	Module: gen_fsm	482

F.30	Module: gen_sctp	483
F.31	Module: gen_server	483
F.32	Module: gen_tcp	484
F.33	Module: gen_udp	485
F.34	Module: global	485
F.35	Module: inet	486
F.36	Module: init	487
F.37	Module: io	487
F.38	Module: io_lib	488
F.39	Module: lib	489
F.40	Module: lists	489
F.41	Module: math	493
F.42	Module: ms_transform	493
F.43	Module: net_adm	493
F.44	Module: net_kernel	494
F.45	Module: os	494
F.46	Module: proc_lib	495
F.47	Module: qlc	495
F.48	Module: queue	496
F.49	Module: random	497
F.50	Module: regexp	497
F.51	Module: rpc	498
F.52	Module: seq_trace	499
F.53	Module: sets	500
F.54	Module: shell	501
F.55	Module: slave	501
F.56	Module: sofs	502
F.57	Module: string	505
F.58	Module: supervisor	507
F.59	Module: sys	507
F.60	Module: timer	508
F.61	Module: win32reg	509
F.62	Module: zip	510
F.63	Module: zlib	510
Index		513

The Pragmatic Bookshelf

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style, and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help you stay on top of your game.

Visit Us Online

Programming Erlang

<http://pragmaticprogrammer.com/titles/jaerlang>

Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates

<http://pragmaticprogrammer.com/updates>

Be notified when updates and new books become available.

Join the Community

<http://pragmaticprogrammer.com/community>

Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

New and Noteworthy

<http://pragmaticprogrammer.com/news>

Check out the latest pragmatic developments in the news.

Buy the Book

If you liked this PDF, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: pragmaticprogrammer.com/titles/jaerlang.

Contact Us

Phone Orders:	1-800-699-PROG (+1 919 847 3884)
Online Orders:	www.pragmaticprogrammer.com/catalog
Customer Service:	orders@pragmaticprogrammer.com
Non-English Versions:	translations@pragmaticprogrammer.com
Pragmatic Teaching:	academic@pragmaticprogrammer.com
Author Proposals:	proposals@pragmaticprogrammer.com