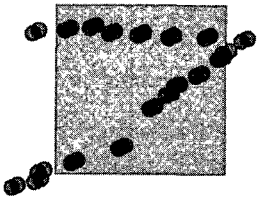
Beautiful Code



Edited by Andy Oram and Greg Wilson

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

CONTENTS

	FOREWORD *by Greg Wilson*	xv
	PREFACE	xvii
1	A REGULAR EXPRESSION MATCHER *by Brian Kernighan*	1
	The Practice of Programming	2
	Implementation	3
	Discussion	4
	Alternatives	5
	Building on It	6
	Conclusion	8
2	SUBVERSION'S DELTA EDITOR: INTERFACE AS ONTOLOGY *by Karl Fogel*	11
	Version Control and Tree Transformation	12
	Expressing Tree Differences	16
	The Delta Editor Interface	17
	But Is It Art?	23
	Abstraction As a Spectator Sport	25
	Conclusions	27
3	THE MOST BEAUTIFUL CODE I NEVER WROTE *by Jon Bentley*	29
	The Most Beautiful Code I Ever Wrote	30
	More and More with Less and Less	31
	Perspective	36
	What Is Writing?	38
	Conclusion	39
	Acknowledgments	40
4	FINDING THINGS *by Tim Bray*	41
	On Time	41
	Problem: Weblog Data	42
	Problem: Who Fetched What, When?	50
	Search in the Large	55
	Conclusion	57

5 CORRECT, BEAUTIFUL, FAST (IN THAT ORDER): LESSONS FROM DESIGNING XML VERIFIERS 59
by Elliotte Rusty Harold

The Role of XML Validation 59
The Problem 60
Version 1: The Naïve Implementation 62
Version 2: Imitating the BNF Grammar O(N) 63
Version 3: First Optimization O(log N) 64
Version 4: Second Optimization: Don't Check Twice 66
Version 5: Third Optimization O(1) 68
Version 6: Fourth Optimization: Caching 72
The Moral of the Story 74

6 FRAMEWORK FOR INTEGRATED TEST: BEAUTY THROUGH FRAGILITY 75
by Michael Feathers

An Acceptance Testing Framework in Three Classes 76
The Challenge of Framework Design 78
An Open Framework 79
How Simple Can an HTML Parser Be? 80
Conclusion 83

7 BEAUTIFUL TESTS 85
by Alberto Savoia

That Pesky Binary Search 87
Introducing JUnit 89
Nailing Binary Search 90
Conclusion 102

8 ON-THE-FLY CODE GENERATION FOR IMAGE PROCESSING 105
by Charles Petzold

9 TOP DOWN OPERATOR PRECEDENCE 129
by Douglas Crockford

JavaScript 130
Symbol Table 131
Tokens 132
Precedence 133
Expressions 134
Infix Operators 134
Prefix Operators 136
Assignment Operators 137
Constants 138
Scope 138
Statements 140

Functions 143
Array and Object Literals 144
Things to Do and Think About 145

10 THE QUEST FOR AN ACCELERATED POPULATION COUNT 147
by Henry S. Warren, Jr.

Basic Methods 148
Divide and Conquer 149
Other Methods 151
Sum and Difference of Population Counts of Two Words 152
Comparing the Population Counts of Two Words 153
Counting the 1-Bits in an Array 154
Applications 158

11 SECURE COMMUNICATION: THE TECHNOLOGY OF FREEDOM 161
by Ashish Gulhati

The Heart of the Start 162
Untangling the Complexity of Secure Messaging 163
Usability Is the Key 165
The Foundation 167
The Test Suite 172
The Functioning Prototype 172
Clean Up, Plug In, Rock On... 173
Hacking in the Himalayas 177
The Invisible Hand Moves 182
Speed Does Matter 184
Communications Privacy for Individual Rights 185
Hacking the Civilization 185

12 GROWING BEAUTIFUL CODE IN BIOPERL 187
by Lincoln Stein

BioPerl and the Bio::Graphics Module 188
The Bio::Graphics Design Process 192
Extending Bio::Graphics 210
Conclusions and Lessons Learned 214

13 THE DESIGN OF THE GENE SORTER 217
by Jim Kent

The User Interface of the Gene Sorter 218
Maintaining a Dialog with the User over the Web 219
A Little Polymorphism Can Go a Long Way 221
Filtering Down to Just the Relevant Genes 224
Theory of Beautiful Code in the Large 225
Conclusion 228

14 HOW ELEGANT CODE EVOLVES WITH HARDWARE: THE CASE OF GAUSSIAN ELIMINATION 229
by Jack Dongarra and Piotr Luszczek

The Effects of Computer Architectures on Matrix Algorithms 230
A Decompositional Approach 232
A Simple Version 233
LINPACK's DGEFA Subroutine 235
LAPACK DGETRF 237
Recursive LU 240
ScaLAPACK PDGETRF 243
Multithreading for Multi-Core Systems 247
A Word About the Error Analysis and Operation Count 250
Future Directions for Research 251
Further Reading 252

15 THE LONG-TERM BENEFITS OF BEAUTIFUL DESIGN 253
by Adam Kolawa

My Idea of Beautiful Code 253
Introducing the CERN Library 254
Outer Beauty 255
Inner Beauty 261
Conclusion 266

16 THE LINUX KERNEL DRIVER MODEL: THE BENEFITS OF WORKING TOGETHER 267
by Greg Kroah-Hartman

Humble Beginnings 268
Reduced to Even Smaller Bits 273
Scaling Up to Thousands of Devices 276
Small Objects Loosely Joined 277

17 ANOTHER LEVEL OF INDIRECTION 279
by Diomidis Spinellis

From Code to Pointers 280
From Function Arguments to Argument Pointers 282
From Filesystems to Filesystem Layers 285
From Code to a Domain-Specific Language 287
Multiplexing and Demultiplexing 289
Layers Forever? 290

18 PYTHON'S DICTIONARY IMPLEMENTATION: BEING ALL THINGS TO ALL PEOPLE 293
by Andrew Kuchling

Inside the Dictionary 295
Special Accommodations 296
Collisions 298
Resizing 299
Iterations and Dynamic Changes 300
Conclusion 301
Acknowledgments 301

19 MULTIDIMENSIONAL ITERATORS IN NUMPY 303
by Travis E. Oliphant

Key Challenges in N-Dimensional Array Operations 304
Memory Models for an N-Dimensional Array 305
NumPy Iterator Origins 307
Iterator Design 307
Iterator Interface 313
Iterator Use 314
Conclusion 318

20 A HIGHLY RELIABLE ENTERPRISE SYSTEM FOR NASA'S MARS ROVER MISSION 319
by Ronald Mak

The Mission and the Collaborative Information Portal 320
Mission Needs 321
System Architecture 322
Case Study: The Streamer Service 325
Reliability 328
Robustness 336
Conclusion 338

21 ERP5: DESIGNING FOR MAXIMUM ADAPTABILITY 339
by Rogerio Atem de Carvalho and Rafael Monnerat

General Goals of ERP 340
ERP5 340
The Underlying Zope Platform 342
ERP5 Project Concepts 346
Coding the ERP5 Project 347
Conclusion 351

22 A SPOONFUL OF SEWAGE 353
by Bryan Cantrill

23 DISTRIBUTED PROGRAMMING WITH MAPREDUCE 371
by Jeffrey Dean and Sanjay Ghemawat

A Motivating Example 371
The MapReduce Programming Model 374
Other MapReduce Examples 375
A Distributed MapReduce Implementation 377
Extensions to the Model 380
Conclusion 381
Further Reading 381
Acknowledgments 382
Appendix: Word Count Solution 382

24 BEAUTIFUL CONCURRENCY 385
by Simon Peyton Jones

A Simple Example: Bank Accounts 386
Software Transactional Memory 388
The Santa Claus Problem 396
Reflections on Haskell 404
Conclusion 404
Acknowledgments 406

25 SYNTACTIC ABSTRACTION: THE SYNTAX-CASE EXPANDER 407
by R. Kent Dybvig

Brief Introduction to syntax-case 411
Expansion Algorithm 413
Example 425
Conclusion 428

26 LABOR-SAVING ARCHITECTURE: AN OBJECT-ORIENTED FRAMEWORK FOR NETWORKED SOFTWARE 429
by William R. Otte and Douglas C. Schmidt

Sample Application: Logging Service 431
Object-Oriented Design of the Logging Server Framework 433
Implementing Sequential Logging Servers 439
Implementing Concurrent Logging Servers 444
Conclusion 450

27 INTEGRATING BUSINESS PARTNERS THE RESTFUL WAY 451
by Andrew Patzer

Project Background 452
Exposing Services to External Clients 452
Routing the Service Using the Factory Pattern 456
Exchanging Data Using E-Business Protocols 457
Conclusion 462

28 BEAUTIFUL DEBUGGING 463
by Andreas Zeller

Debugging a Debugger 464
A Systematic Process 466
A Search Problem 467
Finding the Failure Cause Automatically 468
Delta Debugging 470
Minimizing Input 472
Hunting the Defect 473
A Prototype Problem 475
Conclusion 476
Acknowledgments 476
Further Reading 476

29 TREATING CODE AS AN ESSAY 477
by Yukihiro Matsumoto

30 WHEN A BUTTON IS ALL THAT CONNECTS YOU TO THE WORLD 483
by Arun Mehta

Basic Design Model 484
Input Interface 487
Efficiency of the User Interface 500
Download 500
Future Directions 500

31 EMACSPEAK: THE COMPLETE AUDIO DESKTOP 503
by T. V. Raman

Producing Spoken Output 504
Speech-Enabling Emacs 505
Painless Access to Online Information 516
Summary 522
Acknowledgments 525

32 CODE IN MOTION 527
by Laura Wingerd and Christopher Seiwald

On Being "Bookish" 528
Alike Looking Alike 529
The Perils of Indentation 530
Navigating Code 531
The Tools We Use 532
DiffMerge's Checkered Past 534
Conclusion 536
Acknowledgments 536
Further Reading 536

33 WRITING PROGRAMS FOR "THE BOOK" 539
by Brian Hayes

The Nonroyal Road 540
Warning to Parenthophobes 540
Three in a Row 541
The Slippery Slope 544
The Triangle Inequality 545
Meandering On 547
"Duh!"—I Mean "Aha!" 548
Conclusion 550
Further Reading 550

AFTERWORD 553
by Andy Oram

CONTRIBUTORS 555

INDEX 565