

# Table of Contents

<b>Introduction</b>	<b>1</b>
<hr/>	
<b>A Potted History</b>	<b>1</b>
<b>Why Perl?</b>	<b>2</b>
It's Free	3
What Is Perl Used For?	3
<b>Windows, UNIX, and Other Operating Systems</b>	<b>4</b>
The Prompt	4
<b>What Do I Need To Use This Book?</b>	<b>5</b>
How Do I Get Perl?	6
How To Get Help	10
Perl Resources	13
<b>Conventions</b>	<b>15</b>
Downloading the Source Code	16
Exercises	16
Errata	17
<b>Customer Support</b>	<b>17</b>
<b>Chapter 1: First Steps In Perl</b>	<b>19</b>
<hr/>	
<b>Programming Languages</b>	<b>19</b>
Interpreted vs. Compiled Source Code	20
Libraries, Modules and Packages	21
<b>Why is Perl Such A Great Language?</b>	<b>22</b>
It's Really Easy	22
Flexibility Is Our Watchword	22
Perl on the Web	22
The Open Source Effort	23
Developers Releases and Topaz	23

<b>Our First Perl Program</b>	<b>24</b>
Program Structure	28
Documenting Your Programs	28
Keywords	29
Statements and Statement Blocks	29
<b>ASCII and Unicode</b>	<b>31</b>
Escape Sequences	32
White Space	32
<b>Number Systems</b>	<b>33</b>
<b>The Perl Debugger</b>	<b>34</b>
<b>Summary</b>	<b>34</b>
<b>Exercises</b>	<b>35</b>
<b>Chapter 2: Working with Simple Values</b>	<b>37</b>
<hr/>	
<b>Types of Data</b>	<b>37</b>
Numbers	38
Binary, Hexadecimal and Octal Numbers	39
Strings	41
Single- vs Double-quoted strings	41
Alternative Delimiters	44
Here-Documents	44
Converting between Numbers and Strings	45
<b>Operators</b>	<b>46</b>
Numeric Operators	46
Arithmetic Operators	46
Bitwise Operators	49
Truth and Falsehood	51
Boolean Operators	53
String Operators	55
String Comparison	57
Operators To Be Seen Later	59
Operator Precedence	59
<b>Variables</b>	<b>60</b>
Modifying A Variable	61
Operating and Assigning at Once	62
Autoincrement and Autodecrement	63
Multiple Assignments	64

---

Scoping	65
How It Works	66
Variable Names	67
<b>Variable Interpolation</b>	<b>68</b>
Currency Converter	69
Introducing <STDIN>	70
<b>Summary</b>	<b>71</b>
<b>Exercises</b>	<b>72</b>
<b>Chapter 3: Lists and Hashes</b>	<b>75</b>
<hr/>	
<b>Lists</b>	<b>75</b>
Simple Lists	76
Less Simple Lists	77
Accessing List Values	80
List Slices	82
Ranges	84
Combining Ranges and Slices	86
<b>Arrays</b>	<b>86</b>
Assigning Arrays	87
Scalar vs List Context	89
Adding to an Array	90
Accessing an Array	91
Accessing Single Elements	91
Accessing Multiple Elements	94
Running Through Arrays	96
Array Functions	99
<b>Hashes</b>	<b>104</b>
Creating a Hash	104
Working with Hash Values	106
Adding, Changing and Taking Values Away from a Hash	107
<b>Accessing Multiple Values</b>	<b>108</b>
<b>Summary</b>	<b>110</b>
<b>Exercises</b>	<b>111</b>
<b>Chapter 4: Loops and Decisions</b>	<b>113</b>
<hr/>	
<b>Deciding If...</b>	<b>114</b>
Logical Operators Revisited	119
Comparing Numbers	120

## Table of Contents

---

Comparing Strings	121
Other Tests	123
Logical Conjunctions	123
Running Unless...	124
Statement Modifiers	124
Using Logic	125
Multiple Choice	125
if elsif else	126
More Elegant Solutions	128
<b>1, 2, Skip A Few, 99, 100</b>	<b>128</b>
for Loops	129
Choosing an Iterator	130
What We Can Loop Over	131
Aliases and Values	131
Statement Modifiers	132
<b>Looping While...</b>	<b>134</b>
while (<STDIN>)	135
Infinite Loops	136
Running at Least Once	138
Statement Modifying	138
Looping Until	139
<b>Controlling Loop Flow</b>	<b>139</b>
Breaking Out	140
Going onto the Next	141
Goto	144
<b>Summary</b>	<b>144</b>
<b>Exercises</b>	<b>145</b>
<b>Chapter 5: Regular Expressions</b>	<b>147</b>
<hr/>	
<b>What Are They?</b>	<b>148</b>
Patterns	148
Interpolation	151
Escaping Special Characters	152
Anchors	153
Shortcuts and Options	155
Posix and Unicode Classes	158
Alternatives	158
Repetition	159
Summary Table	161
Backreferences	162
How the Engine Works	163

---

<b>Working with RegExps</b>	<b>166</b>
Substitution	166
Changing Delimiters	167
Modifiers	168
Split	169
Join	170
Transliteration	171
Common Blunders	171
<b>More Advanced Topics</b>	<b>172</b>
Inline Comments	172
Inline Modifiers	172
Grouping without Backreferences	173
Lookaheads and Lookbehinds	174
Backreferences (again)	176
<b>Summary</b>	<b>176</b>
<b>Exercises</b>	<b>177</b>
<b>Chapter 6: Files and Data</b>	<b>179</b>
<b>Filehandles</b>	<b>179</b>
Reading Lines	181
Creating Filters	183
Reading More Than One Line	185
What's My Line (Separator)?	186
Reading Paragraphs at a Time	188
Reading Entire Files	189
<b>Writing To Files</b>	<b>189</b>
Opening a File for Writing	189
Writing on a Filehandle	190
Accessing Filehandles	195
Writing Binary Data	196
Selecting a Filehandle	197
Buffering	199
<b>Permissions</b>	<b>200</b>
<b>Opening Pipes</b>	<b>201</b>
Piping In	202
Piping Out	205
<b>File Tests</b>	<b>207</b>

<b>Directories</b>	<b>212</b>
Globbing	212
Reading Directories	213
<b>Summary</b>	<b>214</b>
<b>Exercises</b>	<b>215</b>
<b>Chapter 7: References</b>	<b>217</b>
<hr/>	
<b>What Is a Reference?</b>	<b>217</b>
Anonymity	218
<b>The Lifecycle of a Reference</b>	<b>218</b>
Reference Creation	218
Anonymous References	220
Using References	222
Array Elements	224
Reference Modification	225
Hash References	226
Notation Shorthands	227
Reference Counting and Destruction	230
Counting Anonymous References	231
<b>Using References for Complex Data Structures</b>	<b>231</b>
Matrices	231
Autovivification	232
Trees	236
Linked Lists	239
<b>Summary</b>	<b>240</b>
<b>Exercises</b>	<b>241</b>
<b>Chapter 8: Subroutines</b>	<b>243</b>
<hr/>	
The 'Difference' Between Functions and Subroutines	244
Usually	244
In Perl	244
<b>Understanding Subroutines</b>	<b>245</b>
Defining a Subroutine	245
Order of Declaration	247

---

<b>Subroutines for Calculation</b>	<b>249</b>
Parameters and Arguments	249
Return Values	250
The return Statement	252
Caching	252
Context	253
Subroutine Prototypes	254
<b>Understanding Scope</b>	<b>255</b>
Global Variables	255
Lexical Variables	258
Runtime Scope	258
When to Use my() And When to Use local	260
<b>Passing More Complex Parameters</b>	<b>260</b>
@_ Provides Aliases!	260
Lists Always Collapse	261
Passing References to a Subroutine	262
Passing Arrays and Hashes to a Subroutine	263
Passing Filehandles to a Subroutine	264
Default Parameter Values	265
Named Parameters	266
<b>References to Subroutines</b>	<b>266</b>
Declaring References to Subroutines	266
Calling a Subroutine Reference	266
Callbacks	267
Arrays and Hashes of References to Subroutines	268
<b>Recursion</b>	<b>268</b>
<b>Style Point: Writing Big Programs</b>	<b>275</b>
<b>Summary</b>	<b>276</b>
<b>Exercises</b>	<b>277</b>
<b>Chapter 9: Running and Debugging Perl</b>	<b>279</b>
<b>Error Messages</b>	<b>280</b>
Syntax Error Checklist	281
Missing Semicolons	281
Missing Open/Close Brackets	281
Runaway String	282
Missing Comma	283
Brackets around Conditions	283
Barewords	283

<b>Diagnostic Modules</b>	<b>283</b>
warnings	284
strict	286
diagnostics	289
<b>Perl Command Line Switches</b>	<b>290</b>
-e	291
-n and -p	292
-c	294
-i	295
-M	295
-s	296
-I and @INC	298
-a and -F	298
-l and -O	299
-T	299
<b>Debugging Techniques</b>	<b>300</b>
Before the Debugger...	300
Debugging Prints	300
Pare It Down	300
Context	301
Scope	301
Precedence	301
<b>Defensive Programming</b>	<b>302</b>
Strategy	302
Check Your Return Values	303
Be Prepared for the Impossible	303
Never Trust the User	303
Definedness and Existence	303
Have Truthful, Helpful Comments	304
Keep the Code Clean	304
<b>Summary</b>	<b>305</b>
<b>Exercises</b>	<b>305</b>
<b>Chapter 10: Modules</b>	<b>309</b>
<b>Types of Module</b>	<b>309</b>

---

<b>Why Do I Need Them?</b>	<b>310</b>
Including Other Files	310
do	310
require	311
use	312
Changing @INC	312
<b>Package Hierarchies</b>	<b>312</b>
<b>Exporters</b>	<b>313</b>
<b>The Perl Standard Modules</b>	<b>314</b>
File::Find	314
Getopt::Std	316
Getopt::Long	317
File::Spec	318
Benchmark	318
Win32	320
<b>CPAN</b>	<b>322</b>
Installing Modules with PPM	323
Installing a Module Manually	324
The CPAN Module	326
Bundles	329
Bundle::LWP	330
Bundle::libnet	331
Submitting Your Own Module to CPAN	331
<b>Summary</b>	<b>332</b>
<b>Chapter 11: Object-Oriented Perl</b>	<b>335</b>
<b>Working with Objects</b>	<b>335</b>
Turning Tasks into OO Programs	336
Are Your Subroutines Tasks?	336
Do You Need Persistence?	336
Do You Need Sessions?	336
Do You Need OO?	336
Do You Want The User To Be Unaware Of The Object?	337
Are You Still Unsure?	337
Improving Your Vocabulary	337
Objects	337
Attributes	338
Methods	338
Classes	339
Polymorphism	339

## Table of Contents

---

Encapsulation	340
Inheritance	340
Constructors	341
Destructors	341
<b>Rolling Your Own</b>	<b>345</b>
Bless You, My Reference	345
Storing Attributes	347
The Constructor	348
Considering Inheritance	349
Providing Attributes	349
Creating Methods	351
Distinguishing Class and Object Methods	353
Get-Set Methods	354
Class Attributes	355
Privatizing Your Methods	358
Utility Methods	360
Death of an Object	361
Our Finished Class	362
<b>Inheritance</b>	<b>364</b>
What is it?	364
Adding New Methods	365
Overriding Methods	366
<b>Ties</b>	<b>369</b>
<b>Summary</b>	<b>374</b>
<b>Exercises</b>	<b>375</b>
<b>Chapter 12: Introduction to CGI</b>	<b>377</b>
<b>How Do I Get It to Work?</b>	<b>377</b>
Setting Up CGI on UNIX	377
Apache	378
Starting and Stopping Apache	378
DocumentRoot and cgi-bin	379
Setting up Perl CGI on Windows	379
Internet Information Server	380
Personal Web Server	380
Using Windows Web Servers	381
<b>Writing CGI Scripts</b>	<b>381</b>
Basic CGI	381
Plain Text	382
HTML Text	382

---

The CGI Environment	384
HTTP Commands	388
The GET Method	388
The POST Method	389
<b>Writing Interactive CGI Scripts</b>	<b>389</b>
A Form-Based Example	390
Passing Parameters with CGI.pm	390
Checking the HTTP Method	391
Determining the Execution Environment	392
Generating HTML Programmatically	392
The Environment Dumper Rewritten	397
Generating the HTTP Header	398
Generating the Document Header	400
Producing Human-Readable HTML	403
Generating HTML Forms	404
Generating Self-Referential URLs	405
Using the Same Script to Generate and Process Forms	407
Saving and Loading CGI State	409
Redirecting from a CGI Script	411
Regenerating Pages with Server Push	412
Cookies and Session Tracking	415
<b>Debugging CGI Scripts</b>	<b>420</b>
Using CGI.pm to Debug Scripts from the Command Line	421
<b>CGI Security</b>	<b>422</b>
An Example of an Insecure CGI Script	422
Executing External Programs	423
Reading and Writing to External Programs	425
Taint Checking	426
An Example of a More Secure CGI Script	428
CGI Wrappers	429
A Simple Security Checklist	429
<b>Summary</b>	<b>431</b>
<b>Chapter 13: Perl and Databases</b>	<b>433</b>
<b>Perl and DBM</b>	<b>434</b>
Which DBM Implementation to Use	434
Accessing DBM Databases	435
Opening a DBM Database	435
Checking the State of a DBM Database	436

## Table of Contents

---

Creating DBM Databases	437
Emptying the Contents of a DBM Database	437
Closing a DBM Database	437
Adding and Modifying DBM Entries	437
Reading DBM Entries	438
Deleting from a DBM Database	438
Writing Portable DBM Programs with the AnyDBM Module	443
Copying from One DBM Format to Another	444
Complex Data Storage	445
Multi-Level DBM (MLDBM)	446
<b>Beyond Flat Files and DBM</b>	<b>449</b>
Introducing Relational Databases	450
<b>Introducing DBI</b>	<b>450</b>
So What Do We Need?	451
Installing DBI	452
What's Available	454
Our DB of Choice - MySQL	456
Installing on Windows	457
Installing on Linux	457
Setting up the Root Account	459
Testing Our MySQL Server	459
Installing DBD::MySQL	459
What's Available Again?	460
<b>First Steps - The Database Cycle</b>	<b>460</b>
Connecting To A Database	460
Connecting To A Remote Database	461
Connecting With The Environment	462
The Fourth Parameter – Connection Flags	463
Disconnecting From a Database	464
Interacting With The Database	465
Creating a Table	467
Populating a Table With Information	470
A Note on Quoting	472
Keeping the Table up to Date	474
Pulling Values from the Database	475
Where Do the Records Go?	478
Fetching a Single Value	480
Binding Columns	480
Fetching All Results	482
Extracting Column Information From Statements	482
Removing Information From The Table	484
<b>Summary</b>	<b>485</b>

---

<b>Chapter 14: The World of Perl</b>	<b>487</b>
<b>IPC and Networking</b>	<b>487</b>
Running Programs	488
system	489
Processes and IPC	490
Signals	490
Trapping Signals	492
Fork, Wait and Exec	493
Networking	494
IP Addresses	495
Sockets and Ports	495
Domain Name Service	496
Networking Clients	496
Writing Clients	497
IO::Socket	497
Blocking and IO::Select	498
Servers with IO::Socket	499
<b>Graphical Interfaces</b>	<b>502</b>
Widgets	502
Perl/Tk	503
Perl/GTK+ and Perl/GNOME	503
Glade	504
Perl/Qt	505
Perl Win32 Module	506
<b>Perl Math</b>	<b>506</b>
BigInt and BigFloat	506
Perl Data Language (PDL)	510
Simple Trigonometry	510
Adding Complex Number Support	512
Security and Cryptography	513
crypt – Password Security	513
Public Key Cryptography	515
<b>Working With Data</b>	<b>518</b>
LDAP	518
Different Types of Data - One Way to Present It	519
<b>Working on the Web</b>	<b>520</b>
Log Files	520
PerlScript	520
<b>Communicating with C</b>	<b>520</b>
Using C from Perl	520
Embedding Perl	521
<b>The End of the Beginning</b>	<b>521</b>

<b>Appendix A: Regular Expressions</b>	<b>523</b>
<b>Appendix B: Special Variables</b>	<b>531</b>
<b>Appendix C: Function Reference</b>	<b>539</b>
<b>Appendix D: The Perl Standard Modules</b>	<b>567</b>
<b>Appendix E: Command Line Reference</b>	<b>579</b>
<b>Appendix F: The ASCII Character Set</b>	<b>585</b>
<b>Appendix G: Licenses</b>	<b>593</b>
<b>Appendix H: Solutions to Exercises</b>	<b>603</b>
<b>Appendix J: Support, Errata and P2P.Wrox.Com</b>	<b>623</b>
<b>Index</b>	<b>631</b>

---



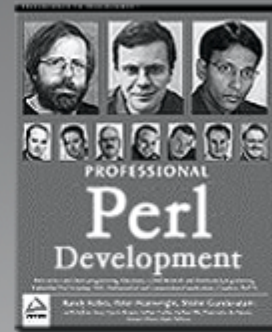
Source code available at : [www.wrox.com](http://www.wrox.com)

Peer discussion at : [lamplists.com](http://lamplists.com)

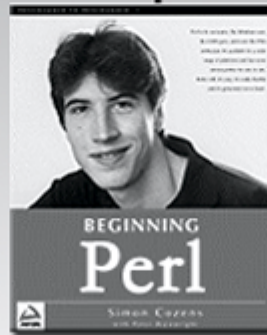
Also from Wrox



<http://www.wrox.com/books/1861004494.htm>



<http://www.wrox.com/books/1861004389.htm>



<http://www.wrox.com/books/1861003145.htm>

[lamplists.com](http://lamplists.com)  
The Open Source Programmer's Resource Centre

This work is licensed under the Creative Commons **Attribution-NoDerivs-NonCommercial** License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd-nc/1.0> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

The key terms of this license are:

**Attribution:** The licensor permits others to copy, distribute, display, and perform the work. In return, licensees must give the original author credit.

**No Derivative Works:** The licensor permits others to copy, distribute, display and perform only unaltered copies of the work -- not derivative works based on it.

**Noncommercial:** The licensor permits others to copy, distribute, display, and perform the work. In return, licensees may not use the work for commercial purposes -- unless they get the licensor's permission.