

## Symbols

---

' character 293  
(<<). *See* append operator  
[]. *See* square brackets  
\*\_eval method family 18  
\*\_dbm libraries 189  
/ character 293  
/. *See* slash 61  
\. *See* backslash 61  
% 60  
%s format specifier 201

## A

---

abstract class 11  
access control 317  
Action Mailer 134  
  on top of TMail 127  
Active Directory 234  
Active Record 7  
ActiveDirectory 238–240  
  using ruby-activedirectory 240  
  using ruby-net-ldap 238  
  using with OpenID 251  
ActiveLdap 238  
ActiveMQ 141–145  
  consuming a message 143  
  queuing a message 142  
ActiveRecord 7, 29, 84, 203,  
  206, 217, 238, 261, 288  
  database support 267  
  methods 267  
  migration 243

  objects 271  
  storing encrypted  
    passwords 234  
  storing messages from  
    WMQ 157  
  storing work orders for asyn-  
    chronous processing 147  
  used in a SOAP service 121  
  without Rails 121  
ActiveResource 114–117  
acts\_as\_ferret 267–270  
acts\_as\_list 84  
acts\_as\_solr 269–270  
add\_class 216  
add\_element 214  
  function 213  
add\_to\_cart method 44, 46  
after block 38  
agile practices 37  
AIM 127, 134  
AJAX 82, 275  
Aksyonoff, Andrew 270  
Amazon S3 55  
Amazon Web Services 56  
anonymous class definition 11  
Ant 17, 21  
AOL Instant Messenger. *See* AIM  
Apache 97, 259  
  load balancing 321–323  
  for reverse proxy 321  
Apache Lucene. *See* Lucene  
API 44, 184–185  
  (g)dbm 189–197  
  low-level database 21  
  MySolr 263  
  web search 273–275  
  wrapper 228–229  
APOP 132  
append operator 186–187  
Apple iCal. *See* iCal  
AppleScript 59, 63  
application  
  scripting 59  
ApplicationController 215  
Appscript 63  
  library 62  
arguments  
  simulating named 12  
assembly language 5  
assert methods 30  
assert\_equal method 30  
assertion 29–30  
Astrotrain 134  
Asynchronous messaging  
  ActiveMessaging 148  
  AP4R 148  
  consuming a message with  
    reliable-msg 147  
  queuing a message with  
    reliable-msg 145  
  queuing a message with  
    RubyWMQ 151, 155  
  RubyWMQ and WMQ  
    149, 155–159  
  StompServer 144  
  testing your application 153  
  using ActiveMQ 141–145  
  using reliable-msg 145–149  
  using Stomp 141–145

ATOM 245  
 Atom 208–209, 228, 243  
   generating with Rails 111  
 attr method 219  
 attr\_accessor 17–18  
 attribute  
   owner definition 18  
   reader 193  
 attributes method 211  
 Authenticated POP. *See* APOP  
 Authentication 233  
   ActiveDirectory 238–240  
   LDAP 237–238  
   username/password 234–237  
 authentication  
   credentials 58  
   with token 244  
 automated testing 25  
 automating with OLE and  
   OSA 59–64

**B**

---

backslashes 61  
 Base64 246  
 baseline setup 42  
 base-row renderer 303  
 Bash 21  
 BDD 35–36, 39  
 BDUF 37  
 before block 38  
 behavior expectation setup 46  
 behavior-driven  
   development 242  
   *See also* BDD  
 benchmarking  
   Rails applications 86–88  
   Rails views 87  
 Berkeley database. *See* DBM  
 Big Design Up Front. *See* BDUF  
 blog data aggregation 216  
 boost  
   query results 266  
   technique 268  
   values 266  
 <br> tag 217  
 broken HTML 216  
 Brooks, Fred 5  
 Builder 21, 95  
   creating an XML message 141  
   using with Rails 113  
 built-in converter 282

business logic test 186  
 byline class 219

**C**

---

C programming language  
   89, 214, 263  
 C/C++ 5  
 C# 17  
 C0 coverage 50  
 calculate\_total method 26  
 calculator  
   configuration 209  
   object instantiation 209  
 <calculator> node 221  
 calendar  
   application 62  
   using Cocoa API 64  
 Capfile 167  
 Capistrano 167–170  
   Capfile 167  
   deploying to different  
     environments 169  
   finding recipes 169  
   rollback 168  
   using Deprec 169  
   using sudo and run 169  
 CGI 246  
   escape 92  
   work around limitations 316  
 Chronic  
   gem 270  
   time-handling library 82  
 class  
   definition 13, 18  
   definition change 17  
   definition checking 20  
 Class object 19  
 class\_eval 18  
 clobber task 68, 70  
 clobber\_rdoc 66  
 close method 193  
 COBOL 5  
 Cocoa API 64  
 code  
   behavior 26  
   duplication reduction 10  
   refactoring 27  
 color 66  
 command-line language,  
   bash 55  
 command-line tool  
   cap 167  
   creating with Ruby gems 164  
   gem 161, 165

God 178  
 https 322  
 IRB 310  
 jirb 312  
 jruby 312  
 lsof 320  
 make 164  
 newgem 162  
 rake 313  
 rcov 50  
 RI 310  
 spec 313  
 warble 315  
 wsdl2ruby 119  
 xsd2ruby 122  
 comma-separated values. *See* CSV  
 comparison report 290  
 compiler  
   type-checking 13  
   verifying behavior 27  
 complex query 259  
 component reference  
   incapsulation 203  
 COMPONENTS constant 193  
 Config  
   class 213  
   object 214, 221  
 @config instance variable 225  
 configuration file  
   format 209  
   parsing 209, 211  
 configuration object 211  
 configuration value, storage 209  
 constant value 203  
 constructor 11, 13  
 contact  
   retrieval 186  
   variable 195  
 Contact class 185, 189, 192, 203  
   close method 193  
   COMPONENTS constant 193  
   implementation 187  
   new instance 185  
   open method 193  
 Contact object 186–187,  
   197, 200  
   caching 194  
   creation 206  
   part of a ContactList 191  
 ContactList  
   class 185–186, 188, 192–193,  
     196  
   new instance 188  
   object 188, 192, 194, 197

@contacts array 187  
 contacts table 199  
 :contains selector 219  
 conventions  
   ActiveResource and REST 115  
   resource mapping and Rails 104  
   response templates in Rails 110  
 convert program 56, 58  
 converter building 282  
 CRM 10, 155  
 cron  
   removal 58  
   scheduling tasks 58  
   starting monitoring service on boot 178  
 CSS 17, 21, 80  
   stylesheet 215  
 CSS3 214  
   selector 216, 219  
   syntax 219  
 CSV 184, 209, 290, 296  
   headers 281  
   processing 279–283  
   on the web 91  
 currency converter 279  
   file output 282  
 custom  
   mock 48  
   PDF 296  
 customer-relationship management. *See* CRM  
 customized reports 295

**D**

---

daemons  
   controlling with Capistrano 167  
   controlling with Vlad 171  
   monitoring with God 173, 176  
 daily report generation 283  
 Darcs 265  
 data  
   format selection 208  
   serialization 209, 230  
 :data option 287  
 data serialization 196  
   format 184  
 data source, swapping 288  
 database 7, 183  
   connection 205

  driver. *See* DBD  
   relational 183  
 database interface library. *See* DBI  
 Davis, Ryan 50–51  
 DBD 197, 204  
   setting 205  
 DBI 184, 204, 206  
 DBM 184, 189  
 debugging 27, 29  
 DELETE command 10  
 delta indexing 272  
 deployment  
   using Capistrano 167–170  
   using Ruby gems 164–166  
   using Vlad 171–173  
   web applications 316  
 Deprec 169  
 describe command 35  
 developer cycle optimization 5  
 Dir.glob 61  
   method 60, 62  
 directory removal 196  
 Distributed Ruby. *See* DRb  
 <div> tag 217  
 docs directory 70, 72  
 document  
   indexing 261  
   processing 279  
   scoring 258  
   search 255  
   sharable 296  
 DOM 213  
 domain model 7  
 domain-specific language 276  
   *See also* DSL  
 DOS command line 61  
 drag and drop 21  
 DRb 145  
 DRY 71, 82  
   efficiency language 14  
 DSL 21  
   intentional programming 23  
   success 23  
   *See also* domain-specific language  
 DTD 20  
 duck typing 11–12  
 dynamic languages  
   testing 27  
 dynamic method  
   definition 20  
 dynamic typing 13

**E**


---

each\_with\_index 60  
 e-commerce site 275  
 EDSL 22  
 EJB 5  
 Elements array 220  
 email 283–289  
   processing email 132–134  
   property 187  
   sending notifications with God 176  
   using APOP 132  
   using POP3 130, 132–134  
   using SMTP 127–129, 132  
   using TMail 132  
 embedded DSL. *See* EDSL  
 empty name attribute 52  
 empty? method 187, 196  
 encryption 317  
 environment variable,  
   HTTP\_PROXY 93  
 error handling 58  
 eRuby 60  
 establish\_connection 8  
 :even selector 216  
 Event Machine 317  
 Excel. *See* Microsoft Excel  
 exception raising 26, 29  
 excerpt 272  
 execute\_purchase method 44  
 \$expectations hash 48  
 extconf.rb 164  
 Extensible Messaging and Presence Protocol. *See* XMPP  
 extraction process 23  
 extras  
   component 193  
   hash 200, 206  
   table 206

**F**

---

fake object 44  
 false response 13  
 FastCGI, disadvantage 316  
 FasterCSV 91, 279–283, 301  
   object 282  
 Fedora Directory Server 237  
 feed reader 221  
 FeedTools 225–229  
 Fernandez, Mauricio 50, 263  
 Ferret 254–259, 263, 267–270  
   compared to FTSearch 265  
   index 257

FieldInfos  
  class 265  
  object 258

File  
  :ALT\_SEPARATOR 61  
  :SEPARATOR 61  
  class 189

file  
  IO 185  
  matching rule 72  
  method 65  
  using glob patterns 98

FileList 70, 72

filename extension  
  substitution 72

FileUtils 194, 196  
  library 69

fill\_textfield 276

find\_by\_contents method 269

find\_by\_name 23

finish! method 266

:first extension 219

FIXME  
  find 59  
  punctuating comment 60

fixtures 41–44

flat files 6

FlexMock 48

float conversion 211

Float object 282

for loop 11

foreign key 199, 203

formatting library 279

FORTRAN 36

Fowler, Chad 231

from\_xml 214  
  method 213

frontend web server 318–319

FTSearch 254, 263–266

fuzz testing 51

## G

gdbm 189–197, 206  
  API 191  
  database 204  
  key/value pair 191

gem 198  
  heckle 51  
  install ferre 255  
  install rspec 36  
  rcov 50  
  Thin 318  
  *See also* Ruby gem

Gems 64  
  ansi-color 170  
  builder 113  
  capistrano 167  
  Chronic 82  
  god 173  
  Highline 171  
  Mongrel 97  
  mysql 308–309  
  net-ssh 170  
  newgem 162  
  RedCloth 82  
  ruby-fcgi 316  
  rubyforge 164  
  rubyzip 98  
  server 310  
  Ultraviolet 78  
  vlad 171  
  warble 317  
  Warbler 315  
  wirble 310  
  xml-simple 93  
  *See also* Ruby Gems

GET command 10

get\_results\_as\_string method 26

global variable 47

glue language 200

GNU 189

GNU Database Manager. *See*  
  gdbm

God  
  lifecycle events 175  
  running as a daemon 178  
  sending notifications 176–178

God.rb 173–178

Google 254, 273, 275–277  
  MapReduce algorithm 15  
  pulling data from Google  
  Finance 91  
  scraping 275

graph  
  building 9  
  themes 9

Gray, James Edward II 279

Greasemonkey 55

Grouping  
  object 294

Growl 64

## H

-h. *See* -help

Hash 227

hash 12, 63, 191, 202

Haskell 15

Heckle 51  
  -heckle option 51  
  -help 224  
  helper module 289

Hibernate 5

Highline 171

home component 193

Hpricot 209, 214, 218, 220–224  
  for JRuby 313  
  object 219

href attribute 219

htaccess 21

HTML 21, 50, 72, 208,  
  214–219, 303  
  data postprocessing 215  
  files converted from  
  Textile 70  
  generating 301  
  parsing 214

HTML/XHTML 50

HTTP 91, 262  
  201 (Created) 94  
  404 (Not Found) 98  
  405 (Method Not  
  Allowed) 100  
  406 (Not Acceptable) 111  
  415 (Unsupported Media  
  Type) 111  
  authentication 246–247  
  Basic Authentication using  
  ActiveResource 114  
  content negotiation 111  
  Content-Disposition 101  
  Content-Length 101  
  GET 91  
  HEAD, OPTIONS 96  
  headers 100, 246  
  Keep-Alive 96  
  Location header 108  
  making requests with  
  headers 94  
  POST 93  
  returning 404 (Not  
  Found) 108  
  returning 412 (Unprocess-  
  able Entity) 108  
  serving requests 97  
  serving requests with  
  Mongrel 97–101  
  status codes 94, 100  
  streaming large files 101  
  using DELETE to remove a  
  resource 115  
  using HTTPS. *See* HTTPS  
  using Net::HTTP 93–96

HTTP (*continued*)  
 using open-uri 91–93  
 using POST to create a resource 115  
 using proxy server 93  
 using PUT to update a resource 115  
 WebDAV 96  
 HTTP\_PROXY 93  
 HTTPS 94, 247  
 Apache reverse proxy 322  
 Nginx reverse proxy 324  
 Hunt, Andy 231  
 Hyett, PJ 272

**I**


---

I/O network library 317  
 iCal automation 61  
 IDE wizard 17  
 identifier 12, 89, 116, 120, 152  
 IM 126, 134  
 integrating with Rails 136  
 Jabber and XMPP 136–139  
 using AIM 134–136  
 using Net::TOC 134  
 using xmpp4r 137  
 image  
 link posting 55  
 transformation 56, 58  
 ImageMagick 58  
 incremental iterations 37  
 :index 258  
 index  
 deletion 262  
 query 261, 264  
 storing 258  
 indexer 254  
 building 264  
 indexing  
 integration 273  
 library 255  
 text only 255  
 initialization method 218  
 initialize method 193–194  
 inject method 16  
 insertion cycle 201  
 instance variable 17, 42  
 instant messaging. *See* IM  
 InstantRails 307  
 Integer 214  
 integer  
 conversion 211  
 test 27  
 integration test 28

intentional programming 23  
 Interactive Ruby Interpreter. *See* IRB  
 IRB  
 interface 11  
 Internet Movie Database 275  
 IRB 281, 307  
 enhancing 310  
 is\_indexed 271  
 items array 224

**J**


---

J2EE 318  
 size of API 5  
 Jabber 127, 134  
*See also* XMPP 137  
 Java 5, 17, 29, 36, 255,  
 314–315, 317  
 CLASSPATH 315  
 deploying Rails  
 applications 315  
 ScriptEngine 315  
 scripting with JRuby 315  
 Java Message Service. *See* JMS  
 Java System Directory 237  
 Java Virtual Machine. *See* JVM  
 JavaBean 20  
 JavaScript 21, 70, 214–215  
 JMS 141  
 join command 219  
 JPEG 56  
 jQuery 214, 220  
 JRuby 312, 317  
 calling Java code 314  
 classpath 315  
 deploying web  
 applications 315  
 ENV\_JAVA 315  
 import 314  
 installation 312–313  
 scripting 315  
 side by side with Ruby 313  
 to\_java 314  
 using Java WS-\* stacks 125  
 js directory 70–71  
 JSON 259  
 JVM 312, 317

**K**


---

key/value pair 191, 206  
 Keyboard  
 class 212–213  
 object 213–214

keyboard specification  
 object 212  
 keyboard.to\_xml 214  
 Keyboard#to\_xml function 214

**L**


---

landscape report 299  
 language features 6  
 large dataset report 301  
 :last selector 219  
 LDAP 90, 234, 237–238  
 authentication 237  
 using ActiveLdap 238  
 using ruby-ldap 238  
 using ruby-net-ldap 237  
 library  
 formatting 279  
 packaging and  
 distribution 161  
 processing 279  
 libxml 95  
 LICENSE 68  
 Lightweight Directory Access  
 Protocol. *See* LDAP  
 Linux 266  
 installing 308  
 installing JRuby 312  
 RubyGems 309  
 setting up Mongrel as a  
 service 321  
 setting up the Thin  
 service 320  
 LISP 15  
 list object 194  
 literal node creation 220  
 load  
 balancing 317–318  
 method 188, 231  
 lock mechanism 15  
 logging 317  
 LOGIN command 10–11  
 login method 11  
 logs, tailing with Capsitrano 170  
 lookup table generation 280  
 Lucene 255, 259, 264

**M**


---

Mac OS X 59, 209  
 automating application 63  
 Fink 308  
 installing 308  
 installing JRuby 312

Mac OS X (*continued*)  
 MacPorts 308  
 One-Click Installer 64, 308  
 RubyGems  
   C extensions 308  
 Mailer configuration 287  
 mainframe applications 5  
 Make 21, 64, 309  
 make method call 63  
 Marohni, Mislav 272  
 measurement  
   pixels 299  
   ruler 299  
 Merb 64, 215  
 metadata 255  
 Metaprogramming  
   BlankSlate and  
     BasicObject 114  
   method\_missing 114  
 metaprogramming  
   16–24, 82–84  
 method 194  
   argument 13  
   calling 29  
 method\_missing 6, 20, 23  
   usage in Builder 113  
 method-defining methods 17  
 Microsoft 238  
 Microsoft Excel 209  
 Microsoft Office 59  
 Microsoft Outlook 209  
 Microsoft SQL Server 267  
 Microsoft Windows 209  
   *See also* Windows  
 missing\_data() method 293  
 mixins 10  
 Mocha 48  
   expectations 48  
 mock  
   custom 46, 48  
   objects 46  
 mocking  
   interface 48  
   library 48  
 Module object 19  
 module\_function 295  
 Mongrel 97–101, 317  
   configuration 321  
   HttpHandler 97  
   for JRuby 313  
   request parameters 97  
   running a cluster 321  
   sending response 97  
   starting a new HttpServer 99  
   using with SOAP4R 122

monit 173  
 monitoring  
   alerts and notifications 176  
   flapping 175  
   process limits and  
     intervals 175  
   using God.rb 173  
 Mowers, Matt 127  
 multicore architectures 54  
 multifield document search 256  
 multiple  
   platform planning 44  
   return values 246  
 mutex 15  
 MySolr 262  
   class 260  
 MySQL 29, 184, 197–204, 267,  
   270, 286  
   API 200, 203  
   database 283  
   database data transfer  
     197, 206  
   :Error 204  
   gem driver 198  
   installing on Linux 309  
   installing on Mac OS X 308  
   managing via email 130  
   managing via IM 137  
 Mythical Man-Month 5

## N

name  
   field population 227  
   lookup 187  
   property 187  
 ndbm 189  
 needed? method 68  
 nested containers 193  
 .Net 318  
 Net::HTTP 93–96  
   HTTP Basic  
     Authentication 96  
   making requests with  
     headers 94  
   working with status codes  
     94, 100  
 Net::TOC  
   responding to IMs 135  
   sending an IM 134  
 New Database Manager. *See*  
   ndbm  
 Nginx 97, 318  
   load balancing 323  
   setup 323

nil 29, 31, 195  
 Nitro 215  
 North, Dan 35  
 Novell eDirectory 237  
 :nth-child 216  
 null test 27  
 numeric\_keypad 211

## O

-o option 50  
 object  
   definition 27  
   extension 11  
 object-oriented 212  
 object-relational mapper. *See*  
   ORM  
 ODBC 286  
 Og 203, 206  
 OLE Automation 59  
 Olson, Rick 241  
 One-Click Installer 64  
   for Mac OS X 308  
   Ruby installer 307  
 open  
   classes 246  
   method 193, 195, 274  
 Open Scripting Architecture. *See*  
   OSA  
 open source  
   Jabber 136  
   messaging servers 140–149  
 OpenID 234, 247–251  
   Rails plugin 249  
 OpenLDAP 237, 240  
 OpenSSL, cryptographically  
   secure random  
   numbers 236  
 open-uri 91–93, 221,  
   224, 262, 274  
   using proxy server 93  
 options object 300  
 optparse 224  
 Oracle 267  
 order\_on method 23  
 ordinal indices 281  
 ORM 203  
 OS X. *See* Mac OS X  
 OSA 59, 63  
   API 64  
 Outlook automation 59  
   *See also* Microsoft Outlook  
 output  
   directory specification 50  
   generation code 287

**P**


---

<p> element 219  
 paginated search results 272  
 paper orientation 299  
 partial updates 21  
 Passwords  
   hashing 234  
   salting 234  
 :path parameter 259  
 path separator 61  
 pathmap 72  
 pattern matching 293  
 PDF 6  
   custom 302  
   custom documents 296  
   drawing 299  
   generating 296–301  
   report 279, 285  
 PDF::SimpleTable 304  
 PDF::Writer 279, 299–301  
   API 305  
 PenBalance 318  
 percent operator 60  
 Perl 55  
 persistence 184  
 PHP 318  
 PID files, cleaning with God 174  
 plain text  
   data storage 184  
   file 183  
 PNG 56  
   file 9  
 Polaroid effect 55  
 POP3 130, 132–134  
 populate\_contact method  
   194–195, 197  
 Post object 268  
 Postgres 267  
 PostgreSQL 197, 270  
 Pound 318  
 predicates 38  
 prepare task 70  
 prepare\_report 299  
 -prereqs option 69  
 Pretty Printing 8  
 primary key 199  
 printf 60  
 proc 225  
 Proc object 282  
 process\_text\_file 295  
   method 292–293  
 ProcessChecker class 42  
 processing library 279  
 productivity 5

Profiling, Rails applications  
   88–89  
 progress formatter 66  
 proxy objects 192  
 proxy server 317  
   reverse 317  
 ps command 43  
 pull-down menus 21  
 purchase method 47  
 Python 29, 55, 318

**Q**


---

query  
   script 264  
   test 8  
 Query configuration 287  
 querying 254

**R**


---

Rails 4, 6, 23, 42, 77,  
   261, 267, 318  
   accepting XML  
   documents 112  
   ActionWebService 122  
   ActiveResource 114–117  
   adding request formats with  
   params\_parser 113  
   API 23  
   ApplicationController 102  
   associating user with  
   session 108  
   atom\_feed 111  
   authentication 245  
   benchmarking 86–88  
   code generation 240  
   content types 109–113  
   core team 249  
   create action 103  
   creating and editing  
   forms 104  
   customizing routes 105  
   deploying to Java web  
   server 315  
   deployment 317  
   form\_for 112  
   helpers 78–82  
   hiding passwords in log 237  
   index and show actions 102  
   libraries 78  
   link\_to 106, 113  
   listing routes 105  
   map.resources 102

named routes 113  
 performance 86  
 processing and sending  
   email 132  
 processing forms 112  
 profiling 88–89  
 queuing messages to  
   WMQ 152  
 queuing work using  
   reliable-msg 147  
 redirect\_to 103  
 rescue errors 108  
 respond\_to 103, 110  
 REST conventions 104  
 RESTful resources 101–109  
 restful\_authentication  
   240–243  
 routing and resources 102  
 sending and receiving  
   IMs 136  
 server script 42  
 sessions 241  
 templates and content  
   types 110  
 to\_param 109  
 update and destroy  
   actions 103  
 url\_for 106  
 using ActiveRecord on its  
   own 121  
 using Builder templates 113  
 using JRuby 313  
 using to\_xml to create a  
   message 151  
 validation 82  
 Windows installation 307  
 XMPP and  
   ActionMessenger 137  
 Rails plugins 78–80, 85  
   creating 84–86  
   and tasks 86  
 Rake 23, 36, 64, 73, 271  
   :RDocTask 66  
   Rakefile 162, 171  
   using Vlad for  
   deployment 171  
 rake  
   command 67  
   rdoc 66  
   spec failures 66  
 Rake API 65  
 Rakefile 65, 69, 71, 162  
   DRY 71  
   referencing a task 67  
   Vlad the deployer 171

- rakefile 68
- rank\_offsets array 266
- raw hits 266
- rcov command 50
- RDoc 65–66
- rdoc 66, 68
- README 68
- RedCloth 72, 82
  - gem 71
  - library 70
- regular expressions 21
- relational database 21, 183, 197, 204
- reliable-msg 145–149
  - consuming a message 147
  - delivery options 148
  - queuing a message 145
- remote service 45
  - interaction 44
- remote\_session method 11
- render\_grouping 301
- renderable\_data method 287
- report
  - customized 295
  - email 285
  - generation 36
- Report class 287
- report definition 287
  - ActiveRecord-backed 288
- reporting 278
- Representational State Transfer. *See* REST
- require 37
  - method 30
- rerdoc 66
- respond\_to? method 13
- REST 101–117
  - content types 109–113
  - creating a resource 103
  - handling PUT and DELETE 103
  - identifying resources 102
  - multiple representations 111
  - resources 101–109
  - uniform interface 107
  - using DELETE to remove a resource 115
  - using POST to create a resource 115
  - using PUT to update a resource 115
- working with
  - ActiveResource 114
- restful\_authentication
  - 237, 240–243
  - test helpers 242
- @results variable 271
- return command 15
- reverse proxy 317–318
  - architecture 318
  - server 317
- REXML 95, 210–214, 220
  - parsing a message 156
  - processing XML messages 143
- RI 310
- ri 263
- rm\_rf method 69, 196
- RMagick 7, 57
- rope code generator 283–289
- rounded\_text\_box method 300
- row-based processing, drawbacks 303
- RSpec 22, 34, 36, 51, 65, 242
  - mocking library 48
- RSS 209, 221–229, 243, 245
  - feed 39, 216, 218
  - generating 225
  - object 224
  - parsing 221
- RSS feed 39
  - creation 20
- Ruby 55
  - gdbm extension 191
  - history 4
  - idiom 10, 211
  - Linux installation 308
  - Mac OS X installation 308
  - one-click installer. *See* one-click installer for Mac OS X
  - one-click installer. *See* one-click Ruby installer
  - popularity 6
  - S option 313
  - scripting 55–58
  - side by side with JRuby 313
  - simplicity 14
  - standard library 185
  - standard library processor 281
  - testing library 22, 29
  - using JRuby 312
  - Windows installation 307
  - wrapping C/C++ libraries 164
- Ruby Gems
  - ActiveLdap 238
  - ActiveSalesForce 158
  - installing gems with C extensions 150
  - MailFactory 128
  - mongrel-soap4r 122
  - net-toc 134
  - reliable-msg 145
  - ruburple 136
  - ruby-activedirectory 240
  - ruby-ldap 238
  - ruby-net-ldap 237–238
  - RubyWMQ 149
  - SOAP4R 122
  - stomp 141
  - TMail 134
  - xmpp4r 137
- Ruby gems as Rails plugins 79, 85
- Ruby on Rails 6, 12, 42, 215, 254
  - authentication 240
  - core team 241
  - email library 129
  - sourcing plugins 240
  - YAML usage 230
  - See also* Rails
- Ruby One-Click Installer. *See* One-Click Installer
- Ruby standard library
  - OpenSSL 236
- Ruby/Cocoa bridge 64
- RubyDBI 287
- RubyGems 7–8, 255
  - automating release tasks 163
  - binaries 164
  - C extensions 164
  - gem server 165, 310
  - JRuby 312
  - loaded by default 309
  - packaging 161
  - repository 164–166
  - specification 162
  - updating to latest version 161
  - using newgem 162
  - using RUBYOPT 309
  - wrapping C/C++ libraries 164
- Ruby-OLE bridge 60
- RubyOSA 63
- ruby-prof 89
- RubyWMQ 155
  - queuing a message 151, 155
  - running admin commands 154
  - testing your application 153
  - using synchpoint 156
  - YAML configuration 151, 155

- RubyZip 98
    - ZipOutputStream 98
  - RuleML 17
  - run 272
  - Ruport 279, 301
    - API 301
    - customized formatting 297
    - installation 283
    - PDF formatter 299
  - ruport-util package 283
- S**
- 
- SalesForce
    - integrating with WMQ 158
    - using ActiveSalesForce 158
  - save method 188
  - scanf 60
  - SCGI, disadvantage 316
  - scheduled reports 283
  - scheduling software 285
  - scraping 275
  - script run schedule 55
  - scripting
    - languages 55
    - with JRuby 315
  - scRUBYt! 275–276
  - Scruffy 7, 9
  - Scruffy::Graph instance 9
  - sdbm 189
  - search 254
    - automation 275
    - method 44, 46
    - result order 272
  - search\_each method 257
  - Security
    - authentication 233
    - cryptographically secure
      - random numbers 236
    - hiding passwords in Rails
      - log 237
  - security
    - SHA1 235, 244
    - SHA2 235
  - send method 211
  - separate indexer 256
  - serialization 185
    - method 220
  - Serialization, sending binary
    - messages with reliable-  
msg 146
  - server farms 54
  - Service Oriented Architecture.
    - See* SOA
  - Servlet API 317
  - set\_author method 228
  - setup method 31, 186
  - sharable documents 296
  - shared-nothing architecture 318
  - should method 38
  - should\_not method 38
  - SimpleIdentifierAnalyzer 265
  - SimpleRSS 40
  - simulating named arguments 12
  - single sign-on 247
  - size 187, 196
    - method 13
  - slash and backslash 60
  - slashes 61
  - SMTP 127–129, 132
    - ActionMailer and TMail 127
    - faking a server with
      - MailTrap 127
    - sending a message with
      - Net::SMTP 128
    - session authentication 286
    - using MailFactory 128
    - using Net::SMTP 132
  - SOA 90
  - SOAP 6, 275
    - generating dynamic stubs with
      - SOAP4R 123
    - generating stubs with
      - SOAP4R 124
    - Java stacks and JRuby 125
    - Rails ActionWebService 122
  - SOAP4R 117–125
    - client stubs 124
    - dynamic stubs using
      - WSDLDriverFactory 123
    - mapping XML to Ruby 120
    - naming conventions 120
    - servants 120
    - using WSDL 119
  - sock parameter 11
  - Solr 254, 259–263, 267, 269–270
  - source code
    - generation 20
    - generator 17
  - SOX compliance 4
  - spec 66
    - command 40
    - command-line tool 36
  - specification 35, 66
    - object instantiation 213
    - report 35
  - Sphinx 267, 270–273
    - indexing conditions 272
  - Spring 5
  - sprintf 60
  - SQL 7, 17, 21
    - avoiding raw 288
    - generation 203
    - hacks 267
    - query 284
  - sqlite 197
  - square brackets 187
  - SSH 170
    - using for deployment 167
  - Standard Library
    - CGI 92
    - Dir.glob 98
    - Net::HTTP 94
    - Net::POP3 130, 132
    - Net::SMTP 127–128, 132
    - open-uri 91
    - SOAP4R 122
    - Tempfile 98
    - URI 92, 94
  - standard library
    - mkmf 164
    - SHA1 235
  - standardized export format 289
  - <startup> node 221
  - state change check 26
  - static typing 13
  - static-content caching 317
  - Stomp 141
    - acknowledging messages 144
    - authentication 144
    - consuming a message 143
    - queuing a message 142
    - running StompServer 144
    - using Stomp with
      - ActiveMQ 141–145
  - storage object 12
  - :store 258
  - Streaming Text Orientated Mes-  
saging Protocol. *See* Stomp
  - String 214, 221, 227
  - string
    - deserialization 185
    - object representation 184
    - testing 36
    - value test 27
  - string-based manipulation 221
  - Struct class 20
  - structured format 208
  - stub 44
    - automatic creation 48
  - stubbed class 44
  - stubbing 46
  - Substitute DBM. *See* sdbm
  - Subversion installation 269

subversion, using for  
 deployment 167, 172  
 suffix-array search engine 263  
 SupplierInterface stubbed 45  
 SVN. *See* subversion  
 SWIG 164  
 synchronization issues 290  
 Szinek, Peter 275

## T

---

-T option 50  
 Table object 290, 293  
 tables array 205  
 tabular data 279  
 tag manipulation 19  
 task method 65  
 TDD 28–35, 50, 241  
 teardown 11  
 method 31  
 temporary files, working with 98  
 test  
 adding 29  
 business logic 186  
 case 29  
 coverage 49–51, 242  
 fail 29  
 pass 29  
 refactoring 29  
 runner 29  
 test case  
 setup 31  
 teardown 31  
 test code length 26  
 test suite 190  
 example 185  
 Test::Unit 31  
 Test::Unit::TestCase 29  
 test/unit 186  
 test-driven development. *See*  
 TDD  
 test-first development 28  
 testing  
 assert 30  
 automated 26  
 benefits 26  
 coverage 49–50  
 dynamic languages 27  
 empty name attribute 52  
 environment 41  
 fuzz 51  
 integration test 28  
 library 22, 29  
 metrics 49  
 principles 26

quality 51  
 specification 29  
 team 28  
 types 27  
 unit tests 27  
 workflow 28  
 ZenTest 50  
 text  
 editability 184  
 format 50  
 Textile 71–72  
 converting files to HTML 70  
 TextMate 78  
 text-only indexing 255  
 Thin 317  
 configuration 319  
 live statistics 319  
 running as a service 320  
 setting up 318  
 Thomas, Dave 231  
 thread 15  
 throttling 317  
 TMail 127  
 formatting email 132  
 parsing email 132  
 to\_atom method 228  
 to\_f 211, 282  
 to\_html method 224  
 to\_i 211  
 to\_rss method 228  
 to\_s 214  
 to\_yaml method 231  
 TODO  
 find 59  
 punctuating comment 60  
 -trace flag 67  
 transaction 203  
 e-commerce 28  
 OpenID 248  
 transitive dependency  
 tracking 68  
 Trollop library 225  
 true response 13, 46  
 truncating prevention 304  
 Twitter 55, 57  
 type  
 attribute 211  
 casting 27  
 checking 13  
 @type instance variable 214  
 typing  
 duck 12  
 dynamic 13  
 static 13

## U

---

Ubuntu, installing 309  
 Ultrasphinx 270–272  
 plugin 270  
 Ultraviolet 78  
 UML 21  
 unary operator 193  
 unit test 27  
 unit-testing library 25  
 unmatched\_data() 294  
 URI 92  
 parse 94  
 URL 91  
 escaping 92  
 format suffix 112  
 HTTP Basic  
 Authentication 96  
 rewriting 317  
 validating 82

## V

---

validation in Rails 82  
 values method 202  
 variable declaration 13  
 Varnish 318  
 VBA 59  
 virtual hosting 317  
 Visual Basic for Applications. *See*  
 VBA  
 Vlad the Deployer 64, 171–173  
 Voice over IP. *See* VOIP  
 VOIP 139

## W

---

Warble 315  
 Weaver, Evan 270  
 web application framework 12  
 web framework 215  
 search 273, 275  
 web server  
 building a simple service 97  
 FCGI and SCGI 316  
 JRuby and Warbler 315  
 JRuby and WARs 317  
 Mongrel 97–101, 321  
 Nginx 323  
 processing forms with  
 Rails 112  
 processing XML requests with  
 Rails 112  
 RESTful resources 101

- web server (*continued*)
  - SOAP processing with
    - SOAP4R 120
  - streaming large files 101
  - Thin 318–320
  - using Atom with Rails 111
  - using Capistrano 167
  - working with
    - ActiveResource 114
- web services
  - SOAP and Ruby 117–125
  - using CSV 91
  - using HTTP GET 91
  - using HTTP POST 93
  - using XML 93
- WEBrick 97, 317
- WebSphere MQ 6
- WebSphere MQ. *See* WMQ
- Weirich, Jim 48
- why the lucky stiff 214
- wildcard 259
- will\_paginate 272
- win32ole 60
- Windows
  - installing 307
  - installing JRuby 312
- Wirble 310
- WMQ 90, 149–159
  - queuing a message with
    - RubyWMQ 151, 155
  - running admin commands
    - with RubyMQ 154
  - testing your application 153
  - using Ruby to integrate with
    - SalesForce 158
  - using RubyWMQ 155
  - using synchpoint 156
- work component 193
- wrapper 57
- write-only code 5

- WSDL 21
  - code generation with
    - SOAP4R 119
  - dynamic stubs with
    - SOAP4R 123
  - generating stubs with
    - SOAP4R 124

---

**X**

---

- XHTML 209, 217
- XML 44, 93, 184, 209–214, 259, 273
  - application test 28
  - creating message with
    - Builder 141
  - element nodes 210
  - generating Ruby code from
    - XML Schema 122
  - generation 220
  - mapping to Ruby using
    - SOAP4R 120
  - new document creation 213
  - parsing 210
  - parsing a message with
    - REXML 156
  - processing messages with
    - REXML 143
  - reading 211
  - report 32
  - Schema 17, 20, 22
  - serialization 231
  - serializing 212
  - templates 6
  - transforming an XML
    - document to Hash 156
  - using to\_xml to create a
    - message 151
  - writing 212
  - XmlSimple 34

- XML Path Language. *See* XPath
- XML::Builder 20
- XmlSimple 34, 93, 95
- XMPP 136
  - ActionMessenger 137
  - contact authorization 137
  - responding to IMs 137
  - using xmpp4r 137
- XPath 156, 211, 276
  - query 276
- XSLT 21
- xUnit architecture 31

---

**Y**

---

- Yahoo!
  - IM 136
    - searching with 273–274
  - Web Search API 274
- YAML 42, 183–191, 196, 208, 230–231
  - data serialization format 184
  - deserialization 188
  - example file 189, 198
  - fixtures 242
  - loading 189
  - serialization 188, 231
    - using to configure
      - RubyWMQ 151, 155
  - YAML.load method 185
  - yield statement 11

---

**Z**

---

- ZenTest 50
- zip files, using RubyZip to
  - compress files 98
- zip project package 67