

Index

A

- abs method, 26, 169–170
- Absolute value, 26, 169–170
- Abstract Factory pattern, 16, 239–241
- Abstract syntax trees (ASTs) for parsers
 - building, 264–267
 - complex, 277
 - file-finding interpreters, 267–272
 - parser-less interpreters, 274–276
 - simple, 272–274
 - XML and YAML, 276–277
- AbstractAdapter class, 173
- Account class, 134
- Account initialize method, 47
- AccountProtectionProxy class
 - creating, 178–179
 - for delegation, 187–188
- AccountProxy class, 186–187
- ActiveRecord class
 - Adapter pattern, 173
 - and DSL, 295
 - factory method, 244–245
 - magic methods, 260
 - migration, 155–156
 - observers, 108
 - relationships, 310–311
- ActiveSupport class
 - Decorator pattern, 205
 - Singleton pattern, 224–225
- adapter_for method, 319–320
- Adapter pattern and adapters, 16, 163
 - alternatives, 168–170
 - in Convention Over Configuration
 - example, 316
 - examples, 173–174
 - file, 319
 - FTP, 326
 - HTTP, 318–319
 - loading, 320–323
 - vs. modification, 172
 - selecting, 319–320
 - single instance modifications, 170–172
 - SMTP, 318
 - summary, 174
 - text rendering, 167–168
 - working with, 173
- add_child method, 120
- add_observer method, 99, 105
- add_students method, 48
- add_sub_population method, 305
- add_sub_task method, 118
- AddDryIngredientsTask class, 115–116
- AddEmployee class, 158
- Addition in Ruby, 24–26
- AddLiquidsTask class, 116
- addr_writer method, 308
- Alexander, Christopher, 4
- Algae class, 233, 236
- alias keyword, 202

- alias_method_chain method, 205
 - All class, 267–268
 - all? method, 133
 - Alternatives
 - Adapter pattern, 168–170
 - Decorator pattern, 201–203
 - Singleton pattern, 215–218
 - Amperands (&) in comparisons, 29
 - And class, 271–272
 - and operator, 29
 - Animal class, 311
 - any? method, 133
 - Appending array elements, 39
 - Arguments
 - options, 47–49
 - parentheses for, 22
 - Arithmetic operations, 24–26
 - Array class and arrays
 - appending elements to, 39
 - as composites, 119
 - creating, 38–39
 - iterators, 128–133, 135
 - methods, 182
 - sorting, 39–40, 92
 - as strings, 35
 - subtraction operator, 270
 - ArrayIterator class, 128–133, 135
 - assert method, 72
 - assert_equal method, 72
 - assert_not_nil method, 72
 - Assignments
 - shortcuts, 25
 - variables, 24, 44
 - ASTs (Abstract syntax trees) for interpreters
 - building, 264–267
 - complex, 277
 - file-finding interpreters, 267–272
 - parser-less interpreters, 274–276
 - simple, 272–274
 - XML and YAML, 276–277
 - At signs (@)
 - for class variables, 208
 - for instance variables, 42
 - attr_accessor method
 - BankAccount, 45, 308–309
 - Employee, 96–97
 - attr_reader method, 45, 308–309
 - attr_writer method, 45, 308
 - authorize method, 325
 - authorized? method, 324–325
 - authorizer_for method, 324
- ## B
- Backslashes (\) for extended statements, 22
 - Backup class, 287, 291–292
 - backup method, 285–292
 - backup_newies task, 295
 - backup_oldies task, 295
 - balance method, 44–45
 - BankAccount
 - example class, 41–47
 - Proxy pattern, 174–184
 - Bars (|)
 - in comparisons, 29
 - Proc objects, 86
 - basename method, 268
 - BasicCPU class, 250
 - Beck, Kent, 4
 - begin/rescue statement, 52
 - Berra, Yogi, 14
 - Big files, finding, 269–270
 - Bigger class, 269–270
 - Bignum class, 25–26
 - Blocks
 - as commands, 147–148
 - as observers, 104–105
 - Strategy pattern, 84–88
 - Blogs, Ruby, 337
 - body field in message gateways, 317
 - Boolean operators, 28–30
 - break statement, 33
 - BritishTextObject class, 167–169
 - BritishTextObjectAdapter class, 168
 - bto class, 170–172
 - build utility, 224
 - builder class, 251–253
 - Builder pattern, 16, 249
 - computers, 250–253
 - examples, 259–260
 - magic methods, 258–259
 - polymorphic, 253–256
 - reusable, 257–258
 - preventing mistakes, 256–257

- summary, 260–261
 - working with, 259
 - Button class, 145
- C**
- C programming language, 263–264
 - C# programming language, 7
 - data types, 27
 - declarations, 69
 - strings, 36–37
 - unit tests, 71
 - call method, 84, 86
 - camelCase variable names, 23
 - Car class, 6, 9–10
 - delegation, 12–13
 - meta-programming, 310
 - Carnivore module, 300
 - Case-sensitivity of variables, 23
 - ChangeAddress class, 159
 - changed method, 104
 - ChangeResistantArrayIterator class, 135–136
 - Chatty module, 50
 - check_access method, 179
 - check_sum method, 199
 - checksumming_write_line method, 195
 - CheckSummingWriter class, 199
 - Checksums, 194–196, 199
 - Children in composites, 121
 - class method, 26–27
 - class_eval method, 305, 310–311
 - class keyword, 42
 - class level methods, 210
 - ClassBasedLogger class, 216–217, 219–220
 - Classes
 - creating, 41–43
 - inheritance, 8
 - loading, 320–323
 - methods, 209–211
 - as singletons, 216–217
 - variables, 208–209
 - ClassVariableTester class, 208–209
 - clone method, 220
 - close method, 201
 - Closure, 84
 - Code blocks
 - and commands, 147–148
 - and observers, 104–105
 - and strategies, 84–88
 - Collections. *See* Iterator pattern
 - Colons (:) for symbols, 37
 - Command class, 148
 - Command pattern, 16, 143
 - ActiveRecord, 155–156
 - for buttons, 145–146
 - code blocks, 147–148
 - Madeleine implementation, 156–160
 - queues, 154
 - for recording, 148–151
 - subclasses, 144–145
 - summary, 160–161
 - undo operations, 151–153
 - working with, 154–155
 - Comments, 21
 - Compilers, 71
 - Complex parsers, 277
 - Complex searches, 270–272
 - Component class, 200
 - Composite class, 114–115
 - composite_of method, 305–307, 309
 - Composite pattern, 15, 111
 - arrays, 119
 - composites vs. leaf objects, 120
 - creating composites, 114–118
 - examples, 123–124
 - operators, 118–119
 - pointers, 120–122
 - wholes and parts, 112–114
 - working with, 122–123
 - CompositeBase class
 - analyzing, 304–306
 - meta-programming, 306–308
 - CompositeCommand class, 150, 153
 - CompositeTask class, 117–119, 121–123
 - Composition and inheritance, 7–12
 - Computer class, 250–251, 259
 - computer method, 256–257
 - ComputerBuilder class, 252–254
 - Computers, Builder pattern example, 250–253
 - Concatenation of strings, 34
 - ConcreteComponent class, 200

Configuration. *See* Convention Over Configuration pattern
 Configuration-dependent code, 314
 const_get method
 adapters, 320
 IOFactory, 243
 Constants, 24, 216
 Contexts
 ASTs, 266
 data sharing with strategy, 80–82
 Convention Over Configuration pattern, 17, 313–314
 adapter selection, 319–320
 class loading, 320–323
 examples, 328
 message gateways, 317–319, 326–327
 security, 323–325
 summary, 328–329
 working with, 327
 Conventions in GUI development, 316
 CopyFile class, 149, 153
 Copying files, 149
 CPU class, 250
 CreateBookTable class, 156
 CreateFile class, 148–149, 152
 Creators in Factory Method pattern, 232–233
 Curly braces ({})
 for code blocks, 85
 for expressions, 35–36
 for hashes, 40
 Custom objects. *See* Meta-programming pattern
 Cygwin environment, 333

D

Data types, 27
 DatabaseConnectionManager class, 222–223
 DataPersistence class, 221
 DataSource class, 288
 <<db_type>>_connection method, 246
 Declaring variables, 24, 68–71
 Decorator class, 200
 Decorator pattern, 16, 193
 components, 200
 delegation, 200–201
 dynamic alternatives, 201–203
 examples, 205
 modules, 202–203
 overview, 193–199
 summary, 206
 working with, 204
 wrapping methods, 202
 def_delegators method, 201
 def keyword, 42
 Delegation
 Car class, 12–13
 Decorator pattern, 200–201
 for proxies, 187–188
 Strategy pattern, 78–80
 delete_observer method, 99
 delete_sub_population method, 305
 DeleteCommand class, 152
 DeleteEmployee class, 158–159
 DeleteFile class, 149, 152–153
 Deleting
 files, 149
 observers, 99
 deposit method
 account, 183
 BankAccount, 43
 missing_method for, 186, 190
 describe method, 148
 describe_hero method, 49
 Description class, 311
 DesktopBuilder class, 254–255
 DesktopComputer class, 254
 <destination_host>Authorizer class, 323, 325
 DirectCall class, 190
 Directors in builder classes, 253
 Distributed Ruby (drb) package, 190–191
 Diurnal module, 301
 Division, 24–26
 do statement, 84–85
 Documentation
 Convention Over Configuration pattern, 327
 rdoc utility, 90–91
 doesNotUnderstand method, 184
 Dollar signs (\$) for global variables, 215
 Domain-Specific Language (DSL) pattern, 17, 283
 examples, 294–295
 file backups, 284–285
 overview, 283–284

- PackRat DSL, 284–285
 - building, 287–289
 - data files, 285–286
 - evaluating, 289–290
 - improving, 290–293
 - summary, 295
 - working with, 293–294
- Double quotes (") for strings, 22, 34–35
- down method in ActiveRecord migrations, 156
- downcase method, 34
- drb (Distributed Ruby) package, 190–191
- Drive class, 251
- drive method, 6
- DSL. *See* Domain-Specific Language (DSL)
 - pattern
- Duck class, 228–229, 233, 236
- Duck typing
 - safety and flexibility, 69–71
 - vs. static typing, 68–69
 - Strategy pattern, 82–84
- DuckAlgaePond class, 236
- DuckPond class, 231
- DuckWaterLilyPond class, 234–236
- Dynamic alternatives in Decorator pattern, 201–203
- Dynamic typing, 24
 - with arrays, 39
 - safety and flexibility, 69–71
 - vs. static typing, 68–69
 - Strategy pattern, 82–84

E

- each method
 - Array, 131
 - Hash, 138
 - IO, 138
 - String, 136
- each_byte method
 - IO, 139
 - String, 136
- each_entry method, 139
- each_filename method, 139
- each_index method, 136
- each_key method, 137
- each_line method, 139
- each_object method, 139–140
- each statement, 33
- each_value method, 137
- Eager instantiation, 214–215
- else keyword, 30–31
- elsif keyword, 30–31
- Email messages
 - creating, 259–260
 - gateways, 317–319
- Employee class, 50–51
 - Madeleine implementation, 157
 - Observer pattern, 95–105
- EmployeeManager class, 157–158
- EmployeeObserver class, 108
- empty? method, 70–71
- EmptyTest class, 71
- Encapsulation
 - commands, 151
 - increasing, 11
 - modules for, 49
 - preserving, 172
 - Proc objects, 147
- encrypt method, 164
- Encrypter class, 164–166, 173
- end statements
 - code blocks, 84–85
 - loops, 32
- Engine class, 10–11, 310
- EnhancedIO class, 195
- EnhancedWriter class, 194–195
- Enumerable module, 133–134
- Enumeration interface, 128
- eof? method, 165, 173
- Equal signs (=)
 - for assignments, 44
 - with regular expressions, 41
- Etc module, 179
- eval statement, 286, 289
- evaluate method, 265
- Exceptions, 52–53
- Exclamation points (!)
 - for not operator, 29
 - with regular expressions, 41
- execute method
 - AddEmployee, 158
 - Command, 150–151
 - CompositeCommand, 153

- execute method (*continued*)
 - CreateFile, 148–149
 - Interpreter, 265
 - Oracle, 173
 - execute_command method, 160
 - Expression class, 267, 275
 - expression method, 274
 - Expressions
 - regular, 40–41, 274, 278, 337
 - in strings, 35–36
 - extend method, 202–203
 - Extended statements, 21–22
 - Extending Factory Method pattern, 237–239
 - Extensibility. *See* Convention Over Configuration pattern
 - External DSLs, 284
 - External iterators, 127–133
- F**
- Factory Method pattern, 16, 227–228
 - classes, 236–237, 241–242
 - examples, 244–246
 - extending, 237–239
 - names, 242–243
 - parameterized methods, 233–236
 - pond simulation, 228–230
 - summary, 246–247
 - template method pattern and, 231–233
 - working with, 244
 - false, 28–30
 - FalseClass class, 29
 - __FILE__ constant, 322
 - File backup DSLs, 284–285
 - File class, 322
 - File-finding interpreters, 267–272
 - FileAdapter class, 319
 - FileDeleteCommand class, 154–155
 - FileName class, 268
 - Files
 - copying, 149
 - creating, 148
 - data, 285–286
 - deleting, 149
 - finding, 268–270
 - sending messages to, 319
 - source, 54–55
 - Find class, 268
 - find method, 268
 - Finding files, 268–270
 - Fixnum class, 25–26
 - vandalizing, 169–170
 - Flexibility in duck typing, 69–71
 - Floating-point numbers, 25–26
 - fnmatch method, 268
 - for_each_element method, 130–131
 - for loops, 32–33
 - Formatter class, 78, 81–84
 - Forward slashes (/) for regular expressions, 40–41
 - Forwardable module, 200–201, 310
 - FOX widgets, 124–125
 - Frame classes, 124
 - Frog class, 230, 233, 236
 - FrogAlgaePond class, 234, 236
 - FrogPond class, 231–232
 - FrogWaterLilyPond class, 236
 - from field in message gateway, 317
 - FtpAdapter class, 320, 326
 - FXButton class, 124
 - FXHorizontalFrame class, 124
 - FXLabel class, 124
 - FXMainWindow class, 125
 - FXRuby, 124
 - FXVerticalFrame class, 124
 - FXWindow class, 124
- G**
- Gamma, Erich, 4
 - Gang of Four (GoF), 4
 - Gateways, message
 - extending, 326–327
 - overview, 317–319
 - GenericServer class in WEBrick, 74
 - Geographical Information System (GIS), 263
 - get_child method, 120
 - get_time_required method
 - MakeBatterTask, 116
 - Task, 115
 - getc method, 165, 173
 - getter and setter methods, 45
 - GetWeatherByZipCode method, 180
 - GIS (Geographical Information System), 263
 - Global access, 207–208, 215–216, 220

GoF (Gang of Four), 4
 Graphical user interfaces (GUIs), 315
 libraries, 123–124
 needs anticipation, 315–316
 templates, 316
 greater than operator (>) for expressions, 28

H

Habitat class, 238–240
 Hand waving, 227
 Hash class, 137–138
 Hashes, 40
 Hello World program, 20–22
 HelloModule module, 49–50
 HelloServer class, 74–75
 Helm, Richard, 4
 Herbivore module, 300
 Hook methods, 66–68
 HTML, 277
 HTMLFormatter class, 78, 81–83
 HTMLReader class, 243
 HTMLReport class, 63–65
 HTMLWriter class, 243
 HTTP Post requests, 317
 HttpAdapter class, 318
 HttpServlet class, 314

I

if statement, 30–31
 Immutable objects
 singleton methods disallowed, 171
 strings, 36–37
 include? method, 133
 include statement, 49–50
 increment method, 209
 Indexing
 arrays, 38
 hashes, 40
 strings, 35
 Inflections class, 224
 Inheritance
 composition over, 7–12
 overview, 46–47
 Initialization syntax for hashes, 40
 Installing Ruby
 Mac OS X, 334
 Microsoft Windows, 333
 UNIX-style systems, 333–334
 instance method, singleton, 212, 214
 Instance methods, 42
 instance_of method, 27
 Instance variables, 42–45
 Instances
 modifying, 170–172
 reflection features, 306
 Instantiation in Singleton pattern, 214–215
 Integers, 25
 interactive Ruby (irb) shell, 20–21
 InterestBearingAccount class, 46–47
 Interfaces, programming to, 5–7
 Internal Domain-Specific Language (DSL)
 pattern. *See* Domain-Specific Language (DSL) pattern
 Internal iterators, 130–133
 Interpolation, string, 36
 interpret method, 265
 Interpreter, ruby, 21
 Interpreter pattern, 16, 263–264
 AST creation, 272–277
 examples, 278–279
 languages, 264
 parsers
 building, 264–267
 complex, 277
 file-finding interpreters, 267–272
 parser-less interpreters, 274–276
 simple, 272–274
 XML and YAML, 276–277
 process, 264–267
 summary, 279–280
 working with, 277–278
 interval method, 285–292
 IO class, 138–139
 IOFactory class, 243
 irb (interactive Ruby) shell, 20–21
 isEmpty method, 70
 item method, 129
 Iterator interface, 128
 Iterator pattern, 15, 127
 Enumerable module, 133–134
 examples, 136–140

Iterator pattern (*continued*)
 external iterators, 127–133
 internal iterators, 130–133
 summary, 140–141
 working with, 134–136

J

Java language
 interfaces, 7
 servlets, 314
 Johnson, Ralph, 4
 join method, 54
 Jungle class, 303
 JungleOrganismFactory class, 240

K

keys, encryption, 164

L

lambda method, 84
 Languages
 DSL. *See* Domain-Specific Language (DSL)
 pattern
 interpreters. *See* Interpreter pattern
 LaptopBuilder class, 255, 257–258
 LaptopComputer class, 254–255
 Lazy instantiation, 214–215
 Leaf classes, 114–115, 120
 leaf method, 299
 Leaf nodes in ASTs, 265
 Length and length method
 aggregate classes, 129
 arrays, 38
 strings, 34
 Less than signs (<)
 arrays, 39
 Composite pattern, 118–119
 for expressions, 28
 Libraries
 GUI, 123–124
 standard, 55
 Linux, installing Ruby on, 333–334
 Literals, string, 34–35

load method, 286
 load_adapters method, 321–322
 Loading classes, 320–323
 Logging class, 211–215
 Loops, 32–34
 Lowercase letters for variables, 23

M

Mac OS X, installing Ruby on, 334
 Madeleine, 156–160
 Magic methods, 258–260
 MailFactory builder, 259–260
 MakeBatter class, 113
 MakeBatterTask class, 113, 116–118
 MakeCakeTask class, 117–118
 Manager class, 219, 295
 Marshal package, 156
 math_service proxy, 191–192
 MathService class, 191
 max method, 133
 member_of method, 305–306, 309
 member_of_composite? method, 306
 merge method, 132
 Message class, 317
 MessageGateway class, 322
 Messages
 creating, 259–260
 gateways, 317–319, 326–327
 passing, 183–184
 sending, 185
 Meta-programming pattern, 17, 297–298
 custom-tailored objects
 method by method, 298–300
 module by module, 300–301
 new methods, 301–306
 reflection features, 306
 examples, 308–311
 summary, 311–312
 working with, 306–307
 method_missing method
 vs. forwardable, 201
 with magic methods, 258
 Object class, 183–184
 performance, 190
 working with, 184–187

method_name method, 210
 MethodMissingCall class, 190
 Methods
 class-level, 209–211
 custom-tailored objects, 298–300
 defining, 42
 hook, 66–68
 instance, 208
 magic, 258–260
 new, 301–306
 wrapping, 202
 Microsoft Windows, installing Ruby on, 333
 min method, 133
 Mixins, 51–52
 MixTask class, 115–116
 Modifying single instances, 170–172
 ModuleBasedLogger module, 218, 220
 Modules, 49–52
 custom-tailored objects, 300–301
 Decorator pattern, 202–203
 Singleton pattern, 214
 as singletons, 218
 Monitor class, 54
 Motherboard class, 250
 Multiplication, 24–26
 Mutable strings, 37
 MySQL adapter, 245

N

\n character, 22
 Names, variable 23–24
 Need for wisdom, 332
 Needs anticipation in GUI development, 315
 new_animal method
 DuckPond, 231
 DuckWaterLilyPond, 235
 Pond, 232
 new_plant method
 DuckWaterLilyPond, 235
 meta-programming, 298–299
 NewDocumentButton class, 145
 Newline characters, 22
 next statement, 33
 nil value, 27–28, 30
 NilClass class, 28
 Nocturnal module, 300
 NoMethodError class, 183
 Nonterminals in ASTs, 265
 Not class, 270–271
 not operator
 comparisons, 29
 parsers, 276
 notify_observers method
 Employee, 98, 104
 Subject, 102
 NumberedWriter class, 203
 numbering_write_line method, 195
 NumberingWriter module, 197–198, 203
 Numbers for strings, 35

O

Object class, 27
 method_missing method, 183–184
 Module module, 309
 Object-oriented programming inheritance,
 7–12
 Objects
 custom. *See* Meta-programming pattern
 everything is, 26–27
 Factory Method pattern, 236–237, 241–242
 ObjectSpace module, 139–140
 Observable class, 101
 Observable module, 103
 Observer pattern, 15, 95
 code blocks, 104–105
 examples, 108–109
 implementing, 100–104
 for information, 95–100
 summary, 109–110
 variations, 105–106
 working with, 106–108
 old_write method, 202
 old_write_line method, 202
 on_button_push method, 144
 One-Click Ruby Installer, 333
 Operators
 arithmetic, 25–26
 Boolean, 28–30
 Composite pattern, 118–119
 Or class, 271
 or operator, 29
 Oracle adapter, 245

OrganismFactory class, 242
 OS X, 334
 output_end method, 66
 output_line method, 65
 output_report method
 HTMLReport and PlainTextReport,
 65, 82
 Report, 80–81
 output_start method
 PlainTextReport, 66
 Report, 67

P

PackRat class, 292–293
 PackRat DSL, 284–285
 building, 287–289
 data files, 285–286
 evaluating, 289–290
 improving, 290–293
 Parameterized factory methods, 233–236
 parent_classification method, 307
 parent_<composite_name> method, 308
 Parentheses () for arguments, 22
 Parents in composites, 121
 parse method, 274
 Parser class, 273–274
 Parser-less interpreters, 274–276
 Parsers, 265
 complex, 277
 simple, 272–274
 XML and YAML, 276–277
 Parts, assembling. *See* Composite pattern
 Pass the buck technique, 13
 Pathname class, 138–139
 Payroll class, 96, 99–100
 PDFReader class, 243
 PDFWriter class, 243
 PlainTextFormatter class, 78, 82–83
 PlainTextReport class, 64–66
 Plus signs (+) for concatenation, 34
 Pointers, 120–122
 Polymorphic builders, 253–256
 Pond class, 229–232, 234–237
 Pond simulation, 228–230
 extending, 237–239
 objects, 236–237
 parameterized, 233–236
 templates, 231–233
 PondOrganismFactory class, 239
 Portfolio class, 54, 134
 pos method, 201
 Post requests, 317
 Pound signs (#)
 for comments, 21
 for string interpolation, 35
 PreferenceManager class, 221–223
 PrefReader class, 223
 Prevayler project, 156
 private_class_method call, 213
 Proc class
 code blocks, 84–89
 closures, 84
 lambda, 84
 process_message method, 322
 Products
 builder classes, 253
 Factory Method pattern,
 232–233
 Programming to interfaces, 5–7
 Protection proxies, 178–179
 <protocol>Adapter class, 320, 322
 Protocols
 adapters, 319–320
 message gateways, 317–318
 Proxy pattern, 16, 175–176
 examples, 190–192
 message passing, 183–184
 message sending, 185
 method_missing method,
 184–185
 protection proxies, 178–179
 purpose, 176–178
 remote proxies, 179–180
 repetitive code, 182–183
 simplifying, 185–189
 summary, 192
 virtual proxies, 180–182
 working with, 189–190
 public_class_method method, 219
 public_methods method, 306
 pull strategy, 106
 push strategy, 106
 puts method, 21–22

Q

- Queries with file-finding interpreters, 267–272
- Queuing up commands, 154

R

- Racc parser, 277
- Race conditions, 54
- raise statement, 52–53
- rake utility, 224, 294–295
- .rb suffix, 55
- rdoc utility, 90–91
- readable_attribute method, 309
- Recording commands, 148–151
- Redoing commands, 151
- Regular expressions, 40–41
 - information on, 337
 - interpreters for, 278
 - for parsers, 274
- Remote procedure call (RPC) systems, 180
- Remote proxies, 179–180
- remove_child method, 120
- Renderer class, 167
- Rendering text, 167–169
- Report class
 - declarations, 68–69
 - hook methods, 66–68
 - Strategy pattern, 79–81, 83, 88–89
 - Template Method pattern, 59–64, 73–75
- require statement, 54–55, 169, 321–322
- Resources, 335–337
- respond_to? method, 306
- ResultSet, 128
- return statement, 44
- Reusable builders, 257–258
- reverse method, 39
- reverse! method, 39
- reverse_each method, 136
- Reversing array elements, 39, 136
- rewind method, 201
- REXML package, 108
- ri command, 91
- RIGenerator class, 91
- round method, 26
- RPC (remote procedure call) systems, 180

Ruby language overview

- arguments, 47–49
- arithmetic operations, 24–26
- arrays, 38–40
- benefits, 17, 19–20
- Boolean operations, 28–30
- classes, 41–43
- current object, 46
- decision statements, 30–32
- exceptions, 52–53
- hashes, 40
- Hello World program, 20–22
- inheritance, 46–47
- instance variables, 43–45
- interactive, 20
- interpreter, 21
- loops, 32–34
- modules, 49–52
- nil value, 27–28
- objects, 26–27
- regular expressions, 40–41
- source Files, 54–55
- strings, 34–37
- summary, 55–56
- symbols, 37
- threads, 53–54
- variables, 23–24
- RubyGems packaging system, 55, 333
- run method
 - Backup, 287
 - HelloServer, 75
- run_it method, 87
- run_it_with_parameters method, 87–88
- Runt library, 55, 278
- RuntimeException class, 53
- RussolsenDotComAuthorizer class, 324

S

- Safety
 - in duck typing, 69–71
 - Singleton pattern, 219–220
- SaveButton class, 144
- SaveCommand class, 146–147
- SAX2Parser class, 108–109
- scaffolding, 325
- scan method, 274

- Searches by file-finding interpreters, 267–272
- Security in Convention Over Configuration
 - example, 323–325
- select_all method, 173–174
- self keyword, 46, 210
- SelfCentered class, 46
- Semicolons (;) in statements, 21
- send method, 185
- send_message method, 318
- Sending messages, 185
- Separating changes from stable code, 5, 61–64
- Servlets, 314
- set_balance method, 44
- setup method, 72
- Sharing data between context and strategy, 80–82
- Shortcuts, assignment, 25
- Simple parsers, 272–274
- SimpleLogger class, 211–215, 224
- SimpleWriter class, 197–199
- Single instance modifications, 170–172
- Single quotes (') for strings, 22, 34–35
- singleton class, 171
- singleton_methods method, 172
- Singleton pattern, 16, 207
 - alternatives, 215–218
 - classes, 216–217
 - counts, 221
 - examples, 224–225
 - global access, 207–208
 - lazy and eager, 214–215
 - logging class, 211–213
 - methods, 209–211
 - modules, 214, 218
 - safety features, 219–220
 - scattered uses, 221–223
 - summary, 225
 - testing, 223–224
 - variables, 208–209, 215–216, 220
- size method, 38
- Skeletal methods. *See* Template Method pattern
- SlickButton class, 144–145, 147
- Smalltalk programming language, 184, 335–336
- SmtAdapter class, 318
- SOAP, 180
- Software adapters, 164–167
- sort method
 - arrays, 39, 92
 - Enumerable, 133
- sort! method, 39
- Sorting arrays, 39–40, 92
- Source files, 54–55
- Special characters for delimiting strings, 36
- Species class, 303, 307, 311
- SQL language, 277
- sql method, 173
- Square braces ([]) for arrays, 38
- start_engine method
 - Car, 12–13
 - Engine, 310
- Statements, 21–22
- Static typing, 68–71
- stem method, 299
- stop_engine method
 - Car, 12–13
 - Engine, 310
- Strategy pattern, 15, 77
 - delegation, 78–80
 - duck typing, 82–84
 - examples, 90–92
 - procs and blocks, 84–88
 - quick-and-dirty strategies, 88–90
 - sharing data between context and strategy, 80–82
 - summary, 92–93
 - working with, 90
- String class and strings, 22
 - with ArrayIterator, 130
 - iterators in, 136
 - methods, 182
 - object representation as, 27
 - regular expression pattern matching, 40–41
 - scan method, 274
 - working with, 34–37
- String interpolation, 36
- StringIO class, 174
- StringIOAdapter class, 165–167, 173–174
- StringTokenizer interface, 128
- sub_populations method, 305
- Subclasses, 8, 46–47, 144–145
- Subject class, 100–101
- subject method, 181–182

Subject module, 102–104
 Subjects in proxies, 176
 Subtasks, 113
 Subtraction, 24–26, 270
 succ method, 26
 super method, 103
 Superclasses, 8, 46–47
 Symbols
 for hash keys, 40
 as immutable identifiers, 37

T

\t character, 22
 Tab characters, 22
 Task class, 113, 115, 121–123
 Tate, Bruce, 332
 TaxMan class, 99–100
 teardown method in tests, 72
 Template Method pattern, 15, 59–61
 drawbacks, 77
 examples, 74–75
 hook methods, 66–68
 overview, 65
 for separation of code, 61–64
 unit tests, 71–72
 working with, 73–74
 Temporal expressions, 278
 Terminals in ASTs, 265
 TestMethodMissing class, 184–185
 Tests
 Singleton pattern, 223–224
 unit, 71–72
 Text rendering, 167–169
 TextObject class, 167–168
 Thread class, 53–54
 Thread safe code, 54
 Threads, 53–54
 Tiger class, 238, 302–303
 Tildes (-) with regular expressions, 41
 timestamping_write_line method, 195
 TimeStampingWriter class, 199
 TimeStampingWriter module, 203
 to field in message gateways, 317
 to method, 285–292
 to_s method, 27
 total_num_of_tasks method, 122

Tree class, 238, 302–303
 true, 28
 TrueClass class, 28–29
 truncate method, 26
 TurboCPU class, 250
 Types, 68–71

U

Underscores (_) for variables, 23
 Undo operations, 151–153
 unexecute method
 Command, 151
 CopyFile, 153
 CreateFile, 152
 Unit tests, 71–72
 UNIX-style systems, installing Ruby on,
 333–334
 unless statement, 31–32
 until loops, 32
 up method, ActiveRecord migrations, 156
 upcase method, 34
 update method, 98, 106
 Uppercase letters for constants, 24
 URI class, 317
 <user name>_authorized? method, 324–325

V

Variables, 23–24
 classes, 208–209
 instance, 42–45
 Vehicle class, 9–11
 Vertical bars (|)
 in comparisons, 29
 Proc objects, 86
 Virtual proxies, 180–182
 VirtualAccountProxy class, 181–182
 VirtualProxy class, 188–189
 Vlissides, John, 4

W

WaterLily class, 233, 236
 Web sites, information from, 337
 web.xml file, 314

- WEBrick library
 - singletons in, 207
 - Template pattern, 74
- while loops, 32
- Wisdom gap, 332
- withdraw method, 43, 190
- Wrapping methods, 202
- Writable class, 270
- Writable files, finding, 269–270
- write method
 - wrapping, 202
 - WriterDecorator, 201
- write_line method, 195, 205
- write_line_with_numbering method, 205
- write_line_with_timestamp method, 205
- write_line_without_timestamp method, 205
- Writer class, 203
- WriterDecorator class, 197–198, 200–201

X

- XML parsing
 - Factory Method pattern, 244
 - Limitations in Interpreter pattern, 276–277
 - package, 108
- XMLRPC implementation, 244
- XOR encryption, 164

Y

- YAGNI (You Ain't Gonna Need It) principle,
 - 13–14, 244
- YAML (YAML Ain't Markup Language) parsers,
 - 276–277
- yield keyword
 - Backup, 292
 - internal iterators, 130