

index

Symbols

- `` (backticks system command operator) 435
 - calling system program with 436
- ^ (caret) 223
- ^(bitwise EXCLUSIVE OR) operator 93
- : (single colon) Symbol literal constructor 187
- :: (double colon) constant resolution operator 84–85
- :: separator 134
- ! (bang)
 - at end of method names 190
 - operator 150, 190
- !@ method 190
- != method 206
- ?c 361
- ?d 361
- ?e 361
- ?f 361
- ?z 361
- . (dot)
 - in regular expressions 339
 - wildcard character 323
- ... (range literal constructor) 187
- .svn 363
- " (string literal constructor) 187
- "" (String literal constructor) 187
- [] variety of meanings 251
 - Array literal constructor 187
 - hash constructor method 259
- { } (curly braces)
 - Hash literal constructor 187
 - to specify pattern repetition count 333
- operator, customizing 190
- @ (at sign) 65
- @ method 190
- @hash 275
- * (asterisk)
 - and to_a 197
 - expands array into bare list 197
 - prefix operator 197
 - star operator 416
 - un-arraying effect of 197
 - various names for 197
 - zero-or-more quantifier 331
- *args parameter 106
- / (slash character) 72
- // (regular expression literal constructor) 187, 321
- \ (backslash) escape character 215, 323
- \A beginning-of-string anchor 335
- \b word-boundary anchor 335
- \n 217
 - as target of chomp method 227
- \Z end-of-string (except final newline) anchor 335
- \z end-of-string anchor 335
- &
 - block argument prefix 275
 - flag 410
 - special method signature character 408
 - to_proc operator 410
- &&= (and equals) operator 93
- # (hash mark) 84
- #{...} interpolation 215, 338
- % (literal string constructor) 69
- %{} 216
- %= (modulo) operator 93
- %c (date format specifier) 215, 244
- %d (date format specifier) 243
- %m 243
- %Q (literal string constructor) 216
- %q (literal string constructor) 215
- %W (literal array constructor) 253
- %w (literal array constructor) 252
- %x (system command operator) 244, 437
- %Y (date format specifier) 243
- +
 - day-wise conversion 245
 - method, defining and using as infix operator 187
 - method, for strings 220
 - operator, customizing 190
- +@ method 190
- ++= (raise to a power) operator 93
- + = operator 93
- < symbol 76
- << array-append operator 136

- << method 254
 - for strings 220
- << month-wise conversion 245
- << operator 217
- << singleton class notation 378
 - difference between obj.method and 378–379
 - most commonly used for class methods 379
- <<EOM 217
 - as method argument 218
- <=> 433
- <=> (spaceship) operator 207
 - piggybacking on another object 207
- <=> method 224, 301
- = (equal sign), in methods 68, 71
- = (setter) method 219, 253, 260
- = method 219, 253, 394
- == method 206, 224, 257
- === case equality (threeequal) method 156, 345
- === method 187
 - implementing 157
 - in String class 157
- => operator 178, 258
 - in a hash 259
- ~ pattern-matching operator 322
- > (literal lambda constructor) 406
- >> month-wise conversion 245
- | (pipe), as code block parameter delimiters 167
- |= (bitwise OR) operator 93
- ||= (or-equals) operator 93
- \$(library load path) global variable 459
- \$: global variable 126
- \$? global variable 436
- \$/ global variable 299
- \$~ global variable 459
- \$\$ global variable 127
- \$0 (startup file) vs. Perl's \$0 326
- \$0 global variable 126, 460
- \$endrange, sets 274
- \$FILENAME global variable 460
- \$LOAD_PATH global variable 459
- \$SAFE global variable 422
- \$stderr 351
- \$stdin 350
- \$stdout 351

A

- access
 - private 140
 - protected 140
 - public 140
- accessor method 139
- ActiveSupport library 391
- additive change to core functionality 391–393
 - pass-through hybrid change 395
- alias keyword 387, 394
- alias_method method 394
- ancestors method 383
- and-equals operator 93
- anonymous function, basic 406–416
- archdir 16
- ARGF.class object 457
- argument list 164
- ArgumentError 175, 178, 180
- ArgumentList class 408
- arguments 46–50, 408
 - and reference 56
 - assignment to variable 52
 - default value 48
 - defined 9, 36
 - list, rules for 51
 - optional 47
 - order 48
 - required 47
 - syntax 51
- arithmetic expression 238
- arithmetic method 189
- arithmetic operation 238–239
- arithmetic operators
 - 238
 - / 238
 - % 238
 - + 238
 - actually methods 239
 - calling as a method 239
 - syntactic sugar 239
- arity 416
- Array class
 - implementing
 - map_with_index in 391
 - literal constructor 187
 - new method 250
- Array method 251
- Array.new method 250
- Array#size method 168
- arrays 93, 249–258, 285
 - << (append) method 254
 - adding object 254
 - as universal result format 285
 - can contain any object 249
 - combinations 255
 - compact method 258
 - compact! method 258
 - compared to hash 248–249
 - concat method 255
 - conversion methods 197
 - count method 258
 - creating 250–253
 - delete_if method 363
 - determining if nil is an element 284
 - each method 408
 - empty? method 258
 - find method 284
 - first method 258
 - flatten method 257
 - flatten! method 257
 - flattening 257
 - getter method 253
 - grep method 452
 - include? method 233, 258
 - inner, nesting 257
 - in-place mapping 297
 - inserting elements 253–255
 - join method 257, 457
 - last method 258
 - length method 258
 - literal constructor 251
 - map 463
 - map method 411
 - map! method 297
 - of strings, generating with %w and %W 252
 - pack method 317
 - pop method 255
 - printing 47
 - push method 254
 - querying 258
 - rejecting items 285
 - removing duplicate elements 257
 - removing elements 253–255
 - removing object 255
 - replace method 256
 - replacing contents 256
 - retrieving elements 253–255
 - reverse method 257
 - reverse! method 257
 - reversing 257
 - setting/getting multiple elements 253
 - shift method 255
 - size method 258

arrays (*continued*)
 slice method 254
 slice! method 254
 sort method 452, 457
 to_s method 194–195
 transformations 257
 uniq method 257
 uniq! method 257
 unshift method 254
 values_at method 254
 zero-origin indexing 253
 ascend method for
 pathnames 369
 ASCII code of a string 223
 assert_equal method 466
 assertion 465
 assignment
 in conditional test 154
 syntax 153
 assignment (variable) 5, 72
 associative array 248
 asterisk notation 47
 atime attribute 362
 attr method 75
 attr_* method
 family 72–75
 symbols as arguments 234
 attr_accessor method 74
 attr_reader method 74
 attr_writer method 74
 attribute reader method 72
 attribute writer method 72
 attributes 72–75
 automating creation of 73–75

B

backtrace method 178
 bang (!) methods 190–193, 225
 conventions and best
 practices 192
 examining for specific
 objects 453
 bang notation, when to use 192
 bare list 197
 bareword style 9
 basename method 368–369
 BasicObject class 78, 399–403
 instantiating 43
 BasicObject.new 43
 begin/end blocks with while
 keyword 161–162
 behavior, encapsulating in a
 module 92
 Bignum class 238

bind method 418
 Binding class 421
 binding method 421
 binding method to object 418
 bitwise AND 93
 bit-wise operator 189
 BlankSlate class 401
 library 78
 block and proc
 interoperability 408
 block parameter 165, 167, 171,
 174
 block variable, flavors 174
 block_given? method 402
 block, testing for presence
 of 402
 boolean
 comparison test, on a
 range 268
 every object has true or false
 state 201
 expressing in a method 41
 objects 201–205
 objects (true and false) 203
 states 201–205
 values, only nil and false have
 false value 205
 vs. boolean objects 201
 break keyword 160
 broadcast method 430
 Builder library 400
 Builder::XmlMarkup
 method 401

C

-c 11, 21
 C extensions directory 16
 C standard library 348
 call method 406
 callable object defined 406
 callbacks 442
 in practice 464–470
 caller method 460
 capitalize method 225
 capture_block method 409–410
 capture, getting 328
 Casals, Pablo 357
 case
 equality 345
 method 156
 operator 189
 transformation 225
 case statement 155–159
 behavior, programming 157
 testing pattern matches 345
 truth-test 158
 using return value as method
 argument 159
 case-changing methods 225
 center method 226
 chaining 307
 change per object 396–399
 character
 alphanumeric, matching
 325
 encoding 229–231
 escaping 215, 323
 literal, in patterns 323
 non-whitespace,
 matching 325
 putting back into file-input
 stream 354
 chdir method 365
 check syntax 21
 chomp method 226
 removing newlines 227
 chop method 226
 chr method 224
 Class class 79
 class/object paradox 79
 singleton classes as instances
 of 377
 writing class methods on 377
 class definition, default
 object 117
 class keyword 79, 423
 << (singleton class) form
 of 378
 for singleton classes 377
 vs. class_eval method 423
 class methods 27, 384–385
 adding with extend 397
 defining with class 379
 definition 82
 use of term 82
 vs. instance method 84
 writing 82
 class module inclusion, vs. sin-
 gleton module
 inclusion 382–384
 class objects
 calling methods 80
 creating 79
 maintaining per-class state
 with 138
 class variable 7, 134–140
 and class hierarchy 137
 pros and cons 137
 class_eval method 420, 423–425

- classes 379–380
 - as message receiver 79–84
 - as object 79–84
 - class_eval method 423, 468
 - core, modifying 386–399
 - define_method method 424
 - definition 9
 - breaking up 64
 - definition block, local
 - scope 128
 - design and naming 110–113
 - false argument to method
 - querying methods 210
 - hierarchy, and class
 - variables 137
 - inherited method 448, 468
 - limitations of 448
 - instance methods
 - getting 455–457
 - viewing 210
 - introduction 61–65
 - mixing module into 94–96
 - nesting 112
 - numerical 238
 - opening with class keyword vs.
 - class_eval 423
 - private_instance_methods
 - method 211
 - protected_instance_methods
 - method 211
 - public_instance_methods
 - method 211
 - reopening 63–64
 - subclassing, hooking into 448
 - vs. module 112
- clear method 227
- clone method 57
- cloning objects 57
- close method, for files 352
- closures 172
 - defined 414
 - Proc object as 413
- code
 - conditional execution
 - 149–159
 - repetition, reducing 73
- code blocks 80, 159
 - { } (curly brace) delimiters 163
 - and iterators 163–174
 - and Proc objects 407
 - and yielding from
 - method 165
 - as body of Proc 407
 - as closure 172
 - as part of method-call
 - syntax 408
- as syntactic constructs
 - 164–165
- delimiter precedence 166
- do/end vs. {} delimiters 160, 165
 - ignored by methods 166
 - in a loop 160
 - not an object 408
 - not method arguments 408
 - not part of argument list 164
- parameter bindings 167
- parameter semantics vs.
 - methods 167, 171
- parameters vs. new local
 - variables 172
- parentheses in parameter
 - list 249
- Proc objects 409–411
 - returning values to a
 - method 170
 - temporary local variables
 - in 173–174
 - using to scope file
 - operations 357
 - variable scoping rules 172
- CodeBlock class 408
- collection, merging into set 273
- colon, in symbol
 - constructor 231
- command, trying without
 - executing 367
- command-line switches 21–24
 - combining command-line
 - switches 23
- comment, magic 229
- comments 6
- commercial method 241
- compact method 258
- Comparable module 206–208, 224
 - and sorting 301
 - defines comparison methods
 - in terms of 208
 - defining for use in connection with 207
- comparison methods 189, 206
- Complex method 251
- concat method 255
- conditional
 - code execution 149–159
 - execution 6
 - expression in assignment
 - syntax 153
 - logic 41
 - modifier 152
 - test, assignment in 154
- conditionals, alternate
 - notations 155
- const_missing method 449
- constant 7
 - basic use 85
 - definition 61
 - forcing absolute path 133
 - introduction 85–87
 - introspection 459
 - lookup 133
 - modifying 87
 - predefined 85
 - reassigning 86
 - resolving, and scope 132–134
- constructors
 - literal 186
- constructors, literal 186
- conversion
 - methods 193–201
 - vs. typecasting 199
- conversions
 - arrays 197
 - numerical 198
- core class, modifying 386–399
- core functionality
 - additive changes 391–393
 - additive/pass-through hybrid
 - changes 395
 - modifying safely 390
 - pass-through changes
 - 393–396
 - per-object changes with
 - extend 396–399
 - risks of modifying 386–391
 - superset-driven overrides 396
- core method, not rewriting to
 - perform unexpectedly 390
- core module, modifying
 - 386–399
- core object, changing behavior
 - of 399
- count method 222
- cp method 367
- crypt method 228
- curly braces 160
 - in string interpolation 221
 - vs. do/end 165
- current object 116

D

- dangerous methods
 - indicated with !-terminated
 - names 190
 - receiver-changing effects 191

- data, normalizing 71
 - Date class 240
 - constructors 241
 - date operations 240–245
 - package 240
 - Date.civil 241
 - Date.new 241
 - Date.today 240
 - dates
 - arithmetic 244
 - conversion methods 244
 - format specifiers 243
 - formatting methods 243
 - iterating over range 245
 - object, instantiation 240
 - parse constructor 241
 - query methods 242
 - datetime
 - civil constructor 242
 - conversion methods 244
 - formatting methods 243
 - iterating over range 245
 - new constructor 242
 - now constructor 242
 - object, instantiation 242
 - parse constructor 242
 - query methods 242
 - DateTime class 240
 - constructors 242
 - datetime objects
 - sec method 243
 - second method 243
 - Daylight Saving Time 243
 - day-of-week method 243
 - debugging with pass-through
 - overrides 393
 - def keyword 35
 - local scope 128
 - def/end keywords 176
 - default object 6, 116–125
 - as message receiver 74
 - top-level 117
 - define_method method 424
 - delegating 443
 - Delegator class 443
 - delete method 227
 - for a set 271
 - for directories 366
 - delete_if method 363
 - desc command 28
 - dictionary 248
 - Dir class 362–366
 - chdir method 365
 - entries method 363
 - glob method 364
 - and use of flags 364
 - instantiating 365
 - mkdir method 365
 - object, creating 362
 - pwd method 366
 - rmdir method 365
 - unlink (rmdir, delete) method 366
 - dir, as Pathname object 369
 - Dir.glob method 364
 - Dir.pwd method 366
 - Dir#path method 363
 - directories
 - creating 365
 - deleting 365
 - entries, reading 363–365
 - globbing 363–365
 - manipulating 362–366
 - navigating to 365
 - removing 367
 - directory (Dir) objects
 - creating 365
 - entries method 363
 - path method 363
 - display method 196
 - do/end vs. curly braces 165
 - dot 164
 - dot operator 36
 - definition 8
 - double colon 84
 - double equal-sign comparison
 - operator 45
 - double quotes
 - and escaping characters 215
 - and interpolation 215
 - double-quoted string,
 - generating 216
 - DryRun module 367
 - dst? method 243
 - dup method 56
 - duplicating objects 56
- E**
-
- e 21
 - each method 168
 - and enumerability 279–281
 - as basis for complex iterations 169
 - for IO objects 349
 - implementing 168
 - relatives of 292
 - return value 168
 - simple use with array 168
 - underlying logic 168
 - each_object method 457
 - each_with_index method 249
 - largely supplanted by with_index 391
 - each, looping through
 - collection 29
 - else keyword 150
 - Ruby's avoidance of ambiguity 151–152
 - elsif keyword 150
 - empty? method 222, 264
 - encode method 230
 - __ENCODING__ 229
 - encoding
 - of source files 229
 - of strings 230
 - end keyword 149
 - end of message (EOM) 217
 - end_with? 222
 - English.rb file, global variable
 - descriptions 127
 - ensure clause 180
 - entries method for
 - directories 363–365
 - enum_for method 306
 - enumerability, adding to
 - objects 311
 - Enumerable module 456
 - methods 281
 - mixing in to classes 281
 - overrides, listing 456
 - enumerables
 - method 291
 - === (case equality operator) 286
 - boolean queries 281–283
 - cycle method 294
 - drop method 290
 - drop_while method 290
 - each method 279, 281
 - each_cons method 293
 - each_index method 292
 - each_slice method 293
 - each_with_index
 - method 249, 292, 315, 391
 - element-wise operations 288–292
 - find method 280, 284
 - find_all method 285
 - finding all matches 285
 - finding minimum and maximum values 291
 - first method 288
 - grep method 286, 452
 - group_by method 287
 - indexing 315

- enumerables (*continued*)
 - inject method 294
 - in-place mapping 297
 - IO objects 349
 - map method 296
 - map_with_index method 315
 - mapping 296–297
 - max method 291
 - max_by method 291
 - min method 291
 - min_by method 291
 - minmax method 291
 - minmax_by method 291
 - none? method 283
 - one? method 283
 - one-object search 284
 - organizing selection
 - results 287
 - partition method 287
 - reduce method 294
 - reducing 294
 - reject method 285–286
 - reject! method 286
 - returning all elements multiple times 294
 - returning arrays of elements 293
 - returning first item 288
 - returning integers with ordinal positions of items 292
 - searching and selecting 284–288
 - select 457
 - selecting based on case equality 286
 - sort method 452
 - sort_by method 302
 - sorting 299–302
 - sort-order logic 301
 - take method 290
 - take_while method 290
 - taking and dropping elements 290
 - using grep with 286
 - using regexps in find operations 341
 - with_index method 315, 392
 - enumeration 267
 - Enumerator::Yielder class 303
 - yield instance method 303
 - Enumerator.new 307
 - enumerators 166, 302–307
 - and fibers 427
 - creating implicitly by blockless iterator calls 307
 - creating with a code block 303–305
 - each method 308
 - each_slice method 314
 - exclusive-or operations on strings 316
 - fine-grained iteration 311
 - inject method 314
 - logic of each method 303, 308
 - method chaining 313–318
 - protecting objects with 309
 - select method 309
 - semantics 307–313
 - un-overriding Enumerable methods 308
 - using to add enumerability to objects 311
 - vs. iterator 311
 - with_index method 315, 391
 - endl? method 206
 - equal sign in methods 68, 71
 - equal? method 206
 - equality-test methods 206
 - Errno namespace 359
 - constant 360
 - Errno::EACCES 359
 - Errno::ENOENT 359
 - Errno::error 175
 - Errno::EISDIR 359
 - error handling 174–182
 - error, file I/O related 359
 - escaping 323
 - characters 215
 - eval method 420
 - dangers 421–422
 - executing strings as code with 420–422
 - family of 420–425
 - event, examples 442
 - exceptions 174–182
 - capturing in rescue clause 178–180
 - class, custom 181–182
 - classes,
 - ZeroDivisionError 175
 - common 175
 - EOFError 431
 - Errno namespace 359
 - file I/O related 359
 - namespace 182
 - NoMethodError 442
 - object, assigning to variable 178
 - raising and rescuing 174–175
 - raising explicitly 177
 - re-raising 179
 - exclusive-or operation 316
 - execute literal script 21
 - execution
 - conditional 6
 - tracing 460–464
 - exit message 426
 - expression examples 37
 - extend event, intercepting 446
 - extend method
 - adding class methods with 397
 - adding to object's functionality with 396
 - making per-object changes with 396–399
 - extensions 17–20
 - defined 18
 - loading 17–20
 - external file, loading 17–20
- ## F
-
- false 6
 - object 41, 201, 203–204
 - state 201
 - state vs. value 204
 - FalseClass 203
 - feature 18
 - Feuermann, Emanuel 357
 - Fiber class 427
 - Fiber.new method 427
 - Fiber.yield method 427
 - fibers
 - basis of enumerators 427
 - resume method 427
 - vs. threads 427
 - yield method 427
 - FIFO. *See* first in, first out
 - File class 348, 352
 - basename method 369
 - examining class methods of 458
 - getting information from 360
 - glob-controlling constants 364
 - join method 363
 - methods for reading files 355
 - open method 357
 - with code block 357
 - query methods 360
 - read method 355–356
 - readlines method 355
 - File object
 - creating 352
 - enumerability 358

- hashes (*continued*)
 - fetch method 261
 - has_key? method 264
 - has_value? method 264
 - include? method 264
 - invert method 263
 - inverting 263
 - iteration 282
 - key
 - as basis for set
 - members 275
 - nonexistent 261
 - key insertion order 259
 - key, symbol as 235
 - key? method 264
 - key/value pairs 258
 - literal constructor 259
 - member? method 264
 - merge method 263
 - ordered 248
 - plethora of methods for testing presence of key 264
 - querying 264
 - retrieving multiple values
 - from 261
 - select method 308–309
 - separating keys and values 259
 - size method 264
 - special syntax for symbol
 - keys 266
 - store method 260
 - transformations 263
 - uniqueness of keys 248
 - unwrapping 266
 - update method 262
 - value? method 264
 - values_at method 261
 - Heifetz, Jascha 357
 - help 23
 - here-doc 217–218
 - in literal object
 - constructor 217
 - single-quoted 217
 - hex method 228
 - hexadecimal integer 239
 - hook 442
- I**

- i modifier 337
- I/O
 - architecture 348–351
 - exceptions and errors 359
 - low-level operations 356
- id number, of object 43
- identifier
 - scoping rules 126
 - types 6–8
- if 6
- if keyword 149–152
- if/else ambiguity in C 152
- immediate value 54
- immutability of symbols 232
- in keyword 163
- include 91
- include? method 264
- included method 445
- inclusion 267
- index method 223
- infix operators 158
- inheritance 75–79
 - chain 75
 - intercepting with
 - inherited 448
 - multiple 77
 - relationship, and singleton
 - classes 385
 - single 77
 - superclass/subclass
 - relationship 112
 - tree 88
- inherited callback,
 - limitations 448–449
- inherited method, for
 - classes 448
- inheriting a class vs. mixing in a
 - module 92
- initialization, automatic 67
- initialize method 66, 136
- input separator, for IO
 - objects 349
- inspect method 195
- inspection 208–211
- installation, Ruby, anatomy
 - of 14–17
- instance method 27, 376
 - getting 455–457
 - introduction 62
 - vs. class method 84
- instance variable 7, 65–68
 - listing 460
 - resolving through self 124
- instance_eval method 132, 420, 422
- instance_exec method 132
- instance_method method 418
- instance_methods method
 - 210, 456
- instance_variables method 460
- instance, introduction 61–65
- instruct! command 400
- Integer class 238
- Integer method 251
 - stricter than to_i for
 - strings 198
- integers
 - hexadecimal 239
 - octal 239
 - times method 328, 434, 465
 - to_i method 410
- Interactive Ruby 5
- interactive Ruby
 - interpretation 24–26
- intermediate object 313
- intern method 229
- interpolation 214
 - string 40
- interpreter 3
 - command-line switches 21–24
 - warning mode 21
- introspection 45
 - of variables and constants 459
- IO class 348
 - eof? instance method 348
 - getc instance method 348
 - in File ancestry 459
 - low-level methods 356
 - SEEK constants 355
 - seek instance method 348
- IO global variables 350
- IO objects 348
 - as enumerable 349
 - iteration 349
 - querying 360–362
- IOError 175
- irb 5, 24–26
- is_a? method 87, 276
- iteration, ingredients of 163
- iterator 29
 - and code blocks 163–174
 - definition 163
 - vs. enumerator 311
- iterators 171

J

- jd method 241
- join method 257, 425

K

- Kernel module 99
 - built-in private instance
 - methods 146
 - private methods 361

Kernel#extend method, making
 per-object changes
 with 396–399
 Kernel#test 361
 key? method 264
 key/value pair 258
 adding to a hash 260
 keyboard input 12, 351
 keys method 432
 keyword 7
 definition 35, 57
 kill message 426

L

-l 22
 lambda 406, 416
 as fussy Proc objects 416
 as Proc object 416
 literal constructor 406
 vs. Proc object 417
 last in, first out (LIFO) 92
 leap year 243
 leap? method 243
 length method 222, 264
 library 18
 library load path 459
 library subdirectory 15
 LIFO. *See* last in, first out
 line mode 22
 link_to method 265
 Linux 360
 literal array constructor 251
 literal constructors 186
 of a hash 259
 overloading of 187
 ljust method 226
 load 95
 load method 17
 load path 18
 local scope 128–130, 172
 and self 131
 local variable 7, 52–58, 129, 174
 and threads 431
 assignment in
 conditional 153
 listing 459
 parse-time recognition of 153
 local_variables method 459
 logging with pass-through
 overrides 393
 loop 159–163
 and yield 163
 conditional 160–162
 controlling 160
 unconditional 159
 yielding to block 166
 lstrip method 226
 lvalue 55, 70

M

m modifier 337
 magic comment 229
 main 119
 definition 119
 map method 166, 169–170
 block evaluation 297
 implementing 170
 return value 170, 296
 vs. each method 170
 map, building on each 171
 match, constraining with
 quantifiers 330–332
 MatchData class 154, 322, 325,
 327–330
 begin method 330
 captures method 328, 463
 end method 330
 getting captures 328
 post_match method 329
 pre_match method 329
 querying instances 154
 MatchData objects, []
 method 328
 mathematical operations 5
 member? method 264
 merge method, for a set 273
 message 8
 receiver 8
 defined 36
 self as receiver of 122
 sending to object 35, 45–46
 syntax 35
 unrecognized, intercepting
 with method-missing
 442–445
 metaclass 385
 method 253, 364
 class vs. instance 84
 method arguments, braceless
 hash pairs 265
 method call syntax, and code
 blocks 408
 Method class, method
 method 418
 method definition, default
 object 117
 method method 418
 Method object 418
 method_added method 450
 method_missing method 99,
 105–110, 400, 442–445
 combining with super
 106–110
 overriding 106
 vs. respond_to? 455
 method-access rules 140–144
 method-lookup path, singleton
 classes on 380–384
 methods 187, 219, 253, 364
 !-terminated not always
 destructive 191–193
 access levels 140
 access rules 140–144
 accessor 139
 added 391
 adding directly to objects 88
 anatomy of calls to 164
 argument, symbol as 234
 arguments 36, 46–50
 arithmetic 189
 as object 417–420
 rationale for using 418
 attribute reader 72
 attribute writer 72
 bang 190–193
 conventions and best
 practices 192
 binding to object 418
 bound 418
 call 8
 similarity to local
 variable 57
 syntax 164
 calling scenarios 80
 chaining 313
 class vs. instance 84
 comparison 189, 206
 conversion 193–201
 dangerous 190
 definition 8
 anatomy of 35
 jumping to with super
 keyword 104
 local scope 128
 definitions, multiple 101–103
 delegating 443
 don't have to yield to a
 block 166
 equality-test 206
 exercising caution with
 destructive 192
 filtering lists of 210
 formal parameter 36

- methods (*continued*)
 - hashes as arguments 265–266
 - inspection, in practice 464–470
 - instance, introduction 62
 - lookup 98–105
 - basics 98
 - failure 105–110
 - rules 101
 - missing 105–110
 - name 164
 - = (equal sign) in 68
 - names 8
 - non-destructive equivalents
 - return copies 191
 - non-private, listing 452–454
 - notation 84
 - overriding 62
 - pass-through change 393
 - private 140–143
 - listing 454–455
 - singleton 142
 - private setter 142
 - protected 143
 - listing 454–455
 - querying 455
 - receiver-changing (dangerous) 191
 - receivers 74
 - return value 37
 - role-playing 199
 - search path 100
 - searching, and singleton
 - classes 380–384
 - setter 68–72
 - singleton 62, 101
 - listing 457
 - that starts with an uppercase letter 251–252
 - top-level 144
 - built-in 146
 - defining 145
 - unbinding from object 418
 - with operator-style syntactic sugar notation 189
 - yielding 164
 - methods method 209, 452–454, 458
 - includes singleton methods 209
 - MiniTest 465
 - MiniTest::Test::Unit class
 - assert_equal method 466–467
 - autorun feature 467
 - mix modifiers 340
 - mix-in 91, 111
 - mkdir method 365
 - mode of a file 14
 - modifier position for if and unless conditions 152, 162
 - Module class, `alias_method` method 394
 - module definition, default object 117
 - `module_eval` method 420, 423–425
 - `Module#extended` method 446
 - `Module#include` method 446
 - modules
 - ancestors method 398
 - callbacks used in singleton classes 447
 - `const_missing` method 449
 - core, modifying 386–399
 - creation and use 91–97
 - `define_method` method 424
 - definition 77
 - definition block, local scope 128
 - design and naming 110–113
 - encapsulating behaviors 92–94
 - extended method 446
 - included method 445–446
 - including in singleton class 381
 - including more than once 103
 - inclusion 98
 - instance methods, getting 455–457
 - `instance_methods` method 210
 - `method_added` 450
 - `method_added` method 468
 - mixing in 91, 94–96, 111
 - `module_eval` method 423
 - nesting 112
 - over-modularization 110
 - trapping include operations 445
 - vs. class 112
 - `ModuleTester` 91
 - multiple inheritance 77
 - `mv` method 367
- N**
-
- name, reserved 173
 - `NameError` 175
 - nesting
 - class definition inside module definition 112
 - classes and modules 130
 - newline character, as target of `chomp` method 227
 - next keyword 160
 - next method 228, 245
 - `next_unit` methods 245
 - `nil` 6, 201–205
 - boolean value 42
 - has boolean value of false 203
 - object 41, 205
 - string and integer representations of 205
 - removing from arrays 257
 - `nil?` method 275
 - `NilClass` 205
 - `-noecho` 25
 - `NoMethodError` 175
 - exception 442
 - normalizing data 71
 - not keyword 150
 - notation, overloading 187
 - `NoWrite` module 367
 - number 237–239
 - random, generating 289
 - `Numeric` class 238
 - numerical conversions 198
- O**
-
- `Object` class 77
 - `object_id` method 43–44, 132
 - `Object`, private instance methods 145
 - `Object.new` 43
 - objects 8
 - adding to stack 94
 - ancestry 77
 - basic 43
 - behavior 38
 - change 396
 - defining 35
 - encapsulating 92
 - extending 443
 - callable, defined 406
 - cloning 57
 - comparison 206–208
 - dereferencing 54
 - duplication 56
 - equality 44
 - extending with a module 447
 - freezing 56

- objects (*continued*)
 - functionality, adding to with extend 396
 - generic 43
 - creating 34–36
 - id number 43
 - identifying uniquely 43–44
 - individuation 89
 - initializing with state 66–68
 - innate behaviors 42–46
 - intermediate 313
 - manipulating 368
 - methods method 452
 - methods, listing 209
 - nature vs. nurture 87
 - non-private methods,
 - listing 452–454
 - numerical 237–239
 - orientation 33
 - private_methods method 454
 - protecting with
 - enumerators 309
 - querying 44, 210
 - relationship to variable 53
 - removing from stack 94
 - responding to messages 35
 - sending message to 45–46
 - special-value 6
 - state 65
 - to which methods can't be added 384
 - total method-calling capabilities 377
 - ObjectSpace module 457
 - oct method 228
 - octal integer 239
 - open family of methods 437–440
 - open method 371
 - communicating with external program 438
 - Open3.popen3 method 438
 - open-uri library 371
 - operator
 - bit-wise 189
 - case equality 189
 - defining, by defining a method 188
 - method-wrapping 93
 - unary, customizing 190
 - or operator, combining flags with 364
 - ord method 223
 - or-equals operator 93, 139
 - overloading, of notation 187
 - over-modularization 110
 - overriding methods 62
- P**
-
- p method 47, 196
 - returns its argument 196
 - package, installing with
 - gem 29–31
 - parameters
 - block 165, 167, 171, 174
 - formal 36
 - order 48
 - syntax 49
 - parentheses
 - in conditional statements 150
 - in submatch capture 325
 - stylistic considerations 151
 - parser 153
 - pass-through change to core
 - functionality 393–396
 - Pathname class 368
 - pathnames
 - ascend method 369
 - basename 368
 - basename method 368
 - directory, examining 369
 - dirname method 369
 - extension, getting 369
 - extname method 369
 - to_s method 368
 - pattern
 - literal characters in 323
 - match, searching for 219
 - specified by regular expression 321
 - patterns, expressed in plain language 321
 - Peel, Mrs. Emma 325
 - percent-sign technique 69
 - pop method 255
 - pos attribute of File object 354
 - prev_unit methods 245
 - print
 - help 23
 - version 23
 - print command 6, 11
 - print method
 - for files 356
 - for IO objects 349
 - printing array 47
 - printing to screen 6
 - private method 140–143
 - vs. singleton method 142
 - private methods 454
 - private setter method 142
 - private_instance_methods
 - method 146, 456
 - private_methods method 211, 454–455
 - Proc class 406–416
 - call method 407
 - history of terminology 407
 - instances vs. lambdas 406
 - new method 406
 - to_proc method 410
 - proc method 406–407
 - Proc object 406–407
 - and code blocks 407
 - and scope 414
 - as closure 413
 - converting to code block 409–411
 - creating 407
 - parameters and arguments 415
 - vs. lambda 417
 - Proc.new method 407
 - Process::Status object 436
 - procs
 - and scope 414
 - indifference to arity 415
 - parameter and argument syntax 415
 - program
 - execution, tracing 460–464
 - external, communicating with 438
 - running 11–12
 - shortening 13
 - protected methods 143, 454
 - protected_instance_methods
 - method 456
 - protected_methods
 - method 211, 454–455
 - public_instance_methods
 - method 456
 - public_method_defined?
 - method 109
 - public_methods method 211
 - public_send method 46
 - push method 254
 - puts method
 - and STDOUT 350
 - calls to_s 194
 - command 6, 11
 - for files 356
 - nil return value 42

Q

quantifiers 330–332
 question mark, using to evaluate true/false 41
 quotes
 double 215
 mechanisms 215
 single 214–215

R

-r flag 15
 rake 28
 namespace 29
 task, defining 29
 Rakefile 28
 rand method 289
 random number 289
 Range class
 instantiating 267
 literal constructor 187
 ranges 266–270
 .. vs. ... 268
 as boolean inclusion tools 268
 backward 269–270
 backward, as index
 argument 269–270
 cover? method 268
 creating 267
 enumeration 267, 283
 exclusive 267
 include? method 268–269
 inclusion 267
 testing 268
 inclusion logic 268–270
 inclusive 267
 member? method 268
 starting and ending
 points 268
 Rational method 251
 rbconfig library 15–16
 RDoc 26
 read attribute 73
 read method 352
 read/write attribute 73
 readbyte method 354
 readchar method 354
 reader/writer attribute,
 creating 74
 reading from a file 13
 readline method 353
 readlines method 353
 real world, defining 33–34
 receiver object 164

receivers
 default 74
 explicit 74
 of a message 8
 recursion 131
 references 54
 and method argument 56
 in variable assignment 55
 reflection 45, 208–211
 Regexp class 321
 escape method 339
 literal constructor 187
 modifying 386
 new method 340
 passing a string to new
 method 340
 Regexp#match method
 modifying 386
 regular expressions 320
 ^ (caret) beginning-of-line
 anchor 335
 ^ (caret) character class nega-
 tion symbol 324
 ? (question mark) non-greedy
 quantifier symbol 332
 ? (question mark) zero-or-one
 quantifier 330–331
 . (dot) wildcard character 323
 {} (curly braces) count
 specifier 333
 * (asterisk) zero-or-more
 quantifier 331
 \A beginning-of-string
 anchor 335
 \b word boundary anchor 335
 \d any digit 325
 \D any non-digit 325
 \S any non-whitespace
 character 325
 \s any whitespace
 character 325
 \w any alphanumeric or
 underscore 325
 \W any non-w character 325
 \Z end-of-string (except final
 newline) anchor 335
 \z end-of-string anchor 335
 + (plus sign) one-or-more
 quantifier 332
 =~ (equal sign/tilde)
 method 322
 \$ (dollar sign) end-of-line
 anchor 335
 \$-style capture variables 431
 adding comments to 337
 anchor 335
 anchors 330, 335
 as constraints 335
 as representations of
 patterns 321
 assertions 335
 atoms 334
 backtracking 333
 building a pattern in 323–325
 captures in string
 scanning 342
 capturing submatches with
 parentheses 325
 case equality 345
 character classes 324
 consuming characters 335
 converting to strings 338–341
 default string
 representation 195, 340
 difference between z and Z
 anchors 336
 engine 332
 escape sequences 324
 global variables for captured
 submatches 326
 greedy and non-greedy
 quantifiers 332–334
 instantiating from string 340
 limitations of parentheses in
 matching 334
 literal characters in 321
 literal constructor 321
 lookahead assertions 336
 lookbehind assertions 336
 match method for regular
 expressions 322
 match success and failure 327
 matching patterns 321
 modifiers 330, 337–338
 negated character classes 324
 negative lookahead
 assertions 336
 negative lookbehind
 assertions 336
 positive lookahead
 assertions 336
 positive lookbehind
 assertions 336
 pre-defined character
 classes 324
 quantifiers 330–332
 range of characters in a char-
 acter class 324
 reputation for crypticness 320
 special characters 323

- regular expressions (*continued*)
 - string methods that use 341, 346
 - submatches 326
 - using captures in string (g)sub operations 344
 - whitespace in 338
 - writing 321–322
 - zero-width property of assertions 336
 - reject method 453
 - repetition, reducing 73
 - replace method 227, 256
 - require 95
 - require method 17, 19
 - require named file or extension 22
 - rescue clause 174
 - capturing exception in 178–180
 - rescue keyword 175–177
 - inside methods and blocks 176
 - reserved name 173
 - respond_to? method 44, 88
 - vs. method_missing 455
 - return keyword 37
 - in lambda or Proc object 417
 - vs. yield 166
 - return value 37
 - reverse method 257
 - rewind method for files 353
 - RFC 2616 format 244
 - RFC 2822 format 244
 - ri 26
 - rindex method 223
 - rjust method 226
 - rm method 367
 - rm_rf method 367
 - rmdir method 365
 - rname 22
 - Rock/Paper/Scissors 432
 - role-playing methods 199
 - to_ary 200
 - to_str 199
 - rstrip method 226
 - Ruby
 - dynamics 419
 - fundamental levels 4
 - identifier literacy 7
 - installation 14–17
 - interactive interpretation 24
 - literacy 3–5
 - program, running 11–12
 - source code tree 15
 - Ruby 1.8
 - block parameter semantics 167
 - Ruby Documentation (RDoc) 26
 - Ruby Index (ri) 26
 - Ruby on Rails, link_to helper method 265
 - RubyGems 17
 - runtime id, permissions 361
 - RuntimeError 175, 177
- S**
-
- scalar objects 213
 - scope
 - and resolving constants 132–134
 - determining 126–140
 - global 126–128
 - local 128–130, 172
 - and self 131
 - scoping rules 126
 - script, execute literal 21
 - search negation 223
 - searching for pattern match 219
 - sec method 243
 - second method 243
 - seek method for files 354
 - self 6, 9, 116–125
 - and local scope 131
 - as default message receiver 74
 - as default receiver of messages 122
 - determining which object is 117–119
 - in class 380
 - and module definitions 119
 - in class, module, and method definitions 119–121
 - in instance method definition 120
 - in singleton method and class method definitions 120
 - resolving instance variables through 124
 - top-level 118
 - using instead of hard-coded class names 121
 - value of in a class definition block 392
 - self keyword 116
 - self.talk 122
 - semicolon 149
 - in block parameter lists 173
 - __send__ method 46
 - send method 45–46, 469
 - alternatives to 46
 - symbol as argument 234
 - Set class 270–274
 - instantiating 270
 - set, manipulating 271–273
 - set.rb source code 274–277
 - Set#initialize method, source code for 274
 - sets 270
 - ^ (XOR) method 272
 - & (intersection) method 272
 - + (intersection) method 272
 - << (append) method 271
 - | (intersection) method 272
 - add method 272, 276
 - add? method 272, 276
 - adding/removing one object 271
 - changing elements in place 285
 - creating 270
 - difference method 272
 - enumeration 283
 - include? method 276
 - initializing with a code block 271
 - in-place mapping 297
 - intersection method 272
 - map! method 285, 297
 - proper_subset method 274
 - proper_superset method 274
 - subset method 274
 - subsets and supersets 274
 - superset method 274
 - union method 272
 - uniqueness of elements 270
 - setter method 68–72
 - of an array 253
 - private 142
 - return values 72
 - shell globbing 364
 - shift method 255
 - SimpleDelegator class 443
 - simple-prompt 24
 - single inheritance 77
 - single quotes
 - and escaping characters 215
 - and interpolation 214
 - single-quoted string, generating 215
 - singleton class 101, 376–385
 - behavior with extended and included 447

- singleton class (*continued*)
 - class keyword notation 378
 - dual determination of object through 377
 - getting as an object 392
 - in inheritance
 - relationships 385
 - including modules 380–381
 - modifying 377–380
 - on method-lookup path 380–384
- singleton method 62, 101, 376
 - calling on subclasses 384
 - defining 377
 - listing 457
 - location 377
 - of a class object, calling 81
 - vs. private method 142
- singleton module, inclusion vs. class module
 - inclusion 382–384
- singleton_class method, implementing 392
- singleton_method_added method 450
 - triggering its own execution 450–451
- singleton_methods method 211, 457
- site_ruby directory 16
- sitedir 16
- size method 222, 264
- sleep method 425
- slice method 219, 254
- slice! method 219, 254
- sort method 452
- source code tree 15
- source file, encoding, setting 229
- spaceship method 224
- spaceship operator 207, 291
- special-value object 6
- split method 72, 165
- sprintf method 69
- stack
 - adding object to 94
 - removing object from 94
- stack method 93
- stack trace
 - examining with caller 460
 - parsing 462–464
- stacklikeness 92
- standard library 17
 - file tools 366–372
- start_with? 222
- state
 - initializing object with 66–68
 - maintaining 66
 - maintaining per class 138
 - of an object 65
- STDERR 348
 - output 350
- STDIN 348
 - and gets method 350
- STDOUT 348
 - and puts method 350
 - used by display method 196
- strftime method 243, 396
- String class 214
 - literal constructor 187
- string conversion 194
- string interpolation 40
- string literal 214
- String method 251
- string notation 214–218
- String#bytes 312
- String#gsub!, modifying 388
 - reasons not to do so 390
- StringIO class 370–372
- strings
 - [] method 218
 - * (multiplication) method 198
 - + method 220
 - << (append) method 220
 - <=> (spaceship) method 224
 - == method 224
 - =~ method 322
 - appending to another 220
 - as quasi-enumerable 298
 - boolean queries 222
 - bytes method 298
 - capitalize method 225
 - capitalizing 225
 - captures in scan method 342
 - case transformation 225
 - center method 226
 - centering 226
 - changing contents of 344
 - chomp method 226
 - chop method 226
 - chr method 224
 - clear method 193, 227
 - combinations 220
 - comparing to symbols 236
 - comparison and ordering 224
 - concatenating 40
 - content queries 222
 - content transformation 226–228
 - conversions 228
 - converting symbols to 231
 - converting to regular expressions 338–341
 - count method 222
 - counting letters 222
 - counting range of letters 223
 - creating new from other strings 220
 - crypt method 228
 - delete method 227
 - deleting characters 227
 - determining ASCII code 223
 - double-quoted, generating 216
 - downcase method 225
 - each_byte method 298
 - each_char method 298
 - each_line method 299
 - empty? method 222
 - emptying 227
 - encoding 229–231
 - encrypting 228
 - end_with? method 222
 - eq!? method 224
 - equal? method 225
 - equality comparison 224
 - escaping special regular expression characters 339
 - exclusive-or operation with enumerators 316
 - finding substrings in 218
 - global substitution 344
 - gsub method 344
 - gsub! method 344
 - hex method 228
 - include? method 222
 - incrementing 228
 - index method 223
 - indexing from end 218
 - individual, encoding 230
 - instantiating regular expression from 340
 - intern method 229
 - interpolation 214, 338
 - iterating
 - through bytes 298
 - through characters 298
 - through lines 299
 - justifying 226
 - length method 222
 - ljust method 226
 - lstrip method 226
 - manipulation, basic 218–221

- strings (*continued*)
 - methods that use regular expressions 341–346
 - next method 228
 - obfuscation 316
 - object identities 225
 - oct method 228
 - ord method 223
 - querying 222
 - for absence of content 222
 - size and length 222
 - removing
 - characters from 219
 - end characters 226
 - end newlines 227
 - replace method 227
 - replacing content 227
 - representing as base number 228
 - rindex method 223
 - rjust method 226
 - rstrip method 226
 - scan method 341
 - called with code block 342
 - search negation 223
 - searching for pattern match 219
 - setting part to a new value 219
 - single-quoted, generating 215
 - size method 222
 - size, transforming 226
 - slice method 219
 - slice! method 219
 - split method 343
 - start_with? method 222
 - strip method 226
 - stripping whitespace 226
 - sub method 344
 - sub! method 344
 - substring searching 222
 - succ method 228
 - swapcase method 225
 - testing for identical content 224
 - testing for start or end 222
 - to_f for non-digits 198
 - to_f method 229
 - to_i for non-digits 198
 - to_i method 228, 239
 - to_s method 194, 229, 410
 - to_sym method 229
 - transcoding 230
 - transformation 225–228
 - transforming formats 225
 - upcase method 225
 - using captures in (g)sub operations 344
 - working with 214–231
 - StringScanner class 342–343
 - strip method 226
 - strptime method 241
 - subclass 75
 - subset, proper 274
 - substring
 - getting and setting 218–220
 - search 219
 - succ method 228, 245
 - super keyword 104–105
 - super method, combining with method_missing 106–110
 - superclass 75
 - superset, proper 274
 - superset-driven override of core functionality 396
 - switches, combining 23
 - Symbol class 214
 - literal constructor 187
 - Symbol.all_symbols method 233
 - Symbol#to_proc method 411–413
 - symbolic identifier 233
 - symbols 231–237
 - adding to symbol table 234
 - and identifiers 233
 - as hash key 235
 - as method argument 234
 - casecmp method 237
 - characteristics 232
 - comparing to strings 236
 - conversion of strings to 229
 - converting strings to 231
 - converting to string 231
 - converting to strings 231
 - creating 231
 - definition 73
 - immutability 232
 - in practice 234–236
 - literal constructor 231
 - string-like methods 236
 - succ method 237
 - table 233
 - to_proc method 411
 - uniqueness 232
 - upcase method 237
 - syntactic sugar 70, 156, 187, 253, 259, 265
 - categories of for operator/methods 189
 - for methods with operator-style calling notation 189
 - for unary operators 190
 - infix operators for method calls 187
 - syntax
 - elements 5
 - error, checking for 11, 174
 - sysread method 356
 - sysseek method 356
 - system command, issuing 435–440
 - system method 422, 435
 - system program
 - executing with backticks 436
 - executing with system method 435
 - syswrite method 356
- T**
-
- tap method 389
 - TCPServer class 428
 - temperature conversion 163
 - Tempfile class 370
 - temporary variable 173
 - terminate message 426
 - test character 361
 - test method 361
 - Test::Unit::TestCase class 465
 - then keyword 149
 - Thomas, Dave 26
 - Thread class 425
 - Thread.new method 425
 - threads
 - and variables 431–435
 - example 428
 - green 425
 - join method 425
 - joining 426
 - keys 432–435
 - killing 426
 - native 425
 - parallel execution with 425–435
 - state, examining 426
 - stopping and starting 426
 - techniques for manipulating 426
 - time
 - adding/subtracting seconds 244
 - arithmetic 244
 - conversion methods 244
 - format specifiers 243

- time (*continued*)
 - format, locale-specific 244
 - formatting methods 243
 - query methods 242
 - Time class 64, 240
 - at constructor 241
 - constructors 241
 - local constructor 241
 - mktime constructor 241
 - new constructor 241
 - now constructor 241
 - parse constructor 241
 - time objects, rfc822 method 395
 - time operations 240–245
 - time.rb 64
 - times method 166–168, 465
 - implementing 166
 - times object, instantiation 241
 - Tk graphics library 16
 - to_* methods 193–201
 - role-playing 199
 - to_a method 197, 252
 - to_ary method 200, 252
 - to_date method 244
 - to_datetime method 244
 - to_enum method 306, 309
 - to_f method 198, 229
 - stricter than to_i for strings 198
 - to_i method 13, 198
 - converting numbers in any base to decimal 239
 - string conversion 228
 - to_i operation 71
 - to_proc method 410
 - generalizing 411
 - to_s 240
 - to_s method 229
 - available for every object 194
 - defining on new object 194
 - for generic new object 194
 - for pathnames 368
 - of the Time class 395
 - to_str method 199
 - in string addition 199
 - to_sym method 229, 231
 - to_time method 244
 - tool, out-of-the-box 20–31
 - top level, defined 118
 - top-level method 144
 - built-in 146
 - defining 145
 - top-level methods
 - defined in Kernel 454
 - user-defined 454
 - transcoding 230
 - true 6
 - state vs. value 204
 - true object 41, 201, 203–204
 - true state 201
 - TrueClass 203
 - truth-value 41
 - typecasting vs. conversion 199
 - TypeError 175
- U**
-
- u escape sequence 231
 - unary operators,
 - customizing 190
 - unbind method 418
 - unbinding method from
 - object 418
 - underscore, matching 325
 - uniq method 257
 - uniq! method 257
 - uniqueness of symbols 232
 - unless keyword 151
 - with else 151
 - unlink method for
 - directories 366
 - unshift method 254
 - until keyword 162
 - as modifier 162
 - at end of begin/end block 162
 - US-ASCII encoding 229
 - use_a_binding method 421
- V**
-
- v 23
 - value, immediate 54
 - value? method 264
 - values_at method 254
 - values, comparing 6
 - variable
 - as object label 55
 - assignment 52–58
 - references in 55
 - block, flavors 174
 - class 7, 134–140
 - and class hierarchy 137
 - pros and cons 137
 - global 7
 - and global scope 126–128
 - and threads 431
 - built-in 126
 - list in English.rb 127
 - listing 459
 - pros and cons 127
 - instance 7, 65–68
 - listing 460
 - introspection 459
 - local 7, 52–58, 129, 174
 - and threads 431
 - assignment in
 - conditional 153
 - listing 459
 - similarity to method call 57
 - relationship to object 53
 - scope 52
 - temporary 173
 - vendor_ruby directory 16
 - vendordir 16
 - verbose mode 23
 - version 23
 - version, print 23
- W**
-
- w 21
 - w flag 11
 - warning mode 21
 - warnings 154
 - Weirich, Jim 28, 78, 400
 - when expression 156
 - when keyword 155
 - allows more than one argument 156
 - underlying logic 156
 - while keyword 160
 - as modifier 162
 - at beginning vs. end of block 161
 - at the end of a loop 161
 - whitespace
 - in patterns 331
 - stripping from strings 226
 - wildcard
 - constrained 324
 - partial 324
 - with_index method 391
 - write method
 - for files 356
 - for IO objects 349
 - writing to a file 14
- X**
-
- x escape sequence 230
 - x modifier 337

XML, Builder library as writing
 tool 400
xor operation 316

Y

YAML 394
yield keyword 164, 303
 vs. return 164, 166
 with argument(s) 167
yield method of yielders 303
yielder 303

Z

ZeroDivisionError 175–176, 179