

Index

SYMBOLS AND NUMBERS

- `%Q`, Ruby literal constructors, 75
- `%Q`, Ruby literal constructors, 75
- && operator**, Ruby, 72
- *** (splat operator)
 - defining methods that accept multiple arguments, 87
 - for expanding range into arrays, 79
- :: operator, for accessing constants**, 127
- _ (underscore prefix)**, indicating partials with, 172
- { } (curly brackets)**, in block syntax, 95
- || (pipes) operator**, Ruby, 72
- < (less than) operator**, Ruby, 72
- <%=h tag**, evaluating expressions, 358
- <= (less than or equal to) operator**, Ruby, 72
- == (equality) operator**, Ruby, 81, 92
- === (case equality) operator**, Ruby, 92
- > (greater than) operator**, Ruby, 74
- >= (greater than or equal to) operator**, Ruby, 74
- \ (backslash)**, convention regarding use of, 137
- .** (dot operator), for accessing methods, 127
- ! (exclamation mark)**, Ruby method syntax, 66
- / (forward slash)**, convention regarding use of, 137
- ! (not operator)**, Ruby, 72
- ? (question mark)**, Ruby method syntax, 66
- ?: (ternary operator)**, Ruby programming, 90
- 37signals**, 8–9

A

- ab tool, for stress testing**, 404
- Acceptance Testing**, 380
- accessor methods, Ruby programming**, 112–116
- action caching**, 406
- ActionController**, 303–352
 - accessing request/response environment, 326–327
 - benchmark method, 401–402
 - console for working with routes, 310–311
 - customizing RESTful routes, 315–317
 - customizing routes, 306–308
 - defining routes, 305
 - enabling/disabling sessions, 337–338
 - expiration of sessions and, 338–339
 - filters, 339–342
 - Flash objects and, 330–332
 - functional testing, 348–351
 - generating controllers, 320–321
 - HTTP authentication and, 342–344
 - improving authentication system, 344–346
 - managing state, 329–330
 - named routes, 311–312
 - namespaced resources, 318–319
 - overview of, 303–304
 - processing actions, 321–322
 - `redirect_to` method, 325–326
 - `render` method, 322–324
 - `request` object, 327–329
 - RESTful route helpers, 319
 - RESTful routes, 312–315
 - route priority, 309–310
 - `send_data` and `send_file` methods, 324–325
 - sessions and, 332–333

ActionController (continued)

ActionController (continued)

- storing session, 333–336
- summary, 352
- testing controllers, 346
- testing routes, 346–348
- working with controllers, 319

ActionMailer

- e-mail and, 381
- as standard package in Rails, 44

ActionPack

- caching and, 406
- MonoRail as clone of, 52
- as standard package in Rails, 44

actions

- adding custom REST action to blog
 - example, 191–192
- `config/routes.rb` and, 151
- defined, 303
- mapping routes to, 153–155
- processing, 321–322
- `redirect_to` method, 325–326
- `render` method, 322–324
- request/response environment and, 327–329
- `send_data` method and `send_file` method, 324–325

ActionView

- adding Ajax to blog example, 377–380
- adding RSS and Atom feeds, 360
- Builder templates, 358–359
- built-in template engines, 357
- creating helpers, 376–377
- default templates, 354
- ERb templates, 357–358
- `format.rss` and `format.atom`, 360–361
- `FormHelper` and `FormTagHelper`, 370–371
- `FormOptionsHelper`, 372
- helpers, 367–368, 375–376
- `index.rss.builder` and `index.atom.builder`, 361–366
- `JavaScriptHelper`, `PrototypeHelper`, and `ScriptaculousHelper`, 373–375

- linking to RSS and Atom feeds, 366–367
- overview of, 353
- predefined helpers, 368–369
- `render` method for bypassing templates, 354–355
- `render` method for rendering specific template, 355
- RJS templates, 359
- software alternatives, 381
- summary, 382
- view layer communicating with other layers, 356–357

ActionWebService (AWS), 391

Active Record pattern

- ASP.NET, 52
- overview of, 233

ActiveRecord

- adding inflection rules, 247–248
- advanced uses, 284
- ArticleTest test case, 299–300
- ASP.NET Active Record pattern and, 52
- association callbacks, 292
- association extensions, 280–281
- associations, 270
- auto-generated methods, 273–274
- `belongs_to` method, 274–277
- caching and, 407
- calculations, 263–265
- callbacks, 289–292
- CRUD operations, 250
- customizing migrations, 240–244
- customizing SQL queries, 262–263
- customizing table names for models, 248–250
- data types and, 141–142
- databases supported, 234–235
- defining models, 38–40
- delete operations, 269–270
- dynamic finders, 131–132, 261–262
- finding and limiting or offsetting fetched records, 256
- finding and ordering or grouping results, 254
- finding by first, last, and all, 253–254
- finding by id(s), 253

- finding with conditions, 254–255
- finding with `include` and `i` options, 257–260
- finding with `readonly` and `lock` options, 260–261
- finding with `SELECT` and `FROM` option, 256
- generating migrations, 239–240
- generating models, 237–239, 250–252
- `has_and_belongs_to_many` method, 279–280
- `has_many` method, 279–280
- `has_one` method, 278
- Inflector class and, 245–246
- many-to-many relationships, 271–273
- mapping tables to models, 244–245
- observers, 292–294
- one-to-many relationships, 271
- one-to-one relationships, 270
- ORM (Object Relational Mapping), 6, 237
- overview of, 233
- partial updates and dirty objects, 265–267
- polymorphic associations, 286–289
- read operation, 253
- sample script using, 235–237
- single table inheritance, 284–286
- as standard package in Rails, 44
- summary, 301
- testing models, 294–301
- update operations, 265
- `update_attribute/update_attributes`, 265–267
- updating multiple records, 267–269
- validation, 174, 281–283
- validation helpers, 283–284
- `ActiveRecord::Base`, **244**
- `ActiveRecordHelper`, **369**
- `ActiveRecordStore`, **333**
- ActiveResource**
 - avoiding duplicate models, 385
 - consuming/publishing REST services from .NET, 390
 - create operations, 387
 - creating ActiveResource models, 384–385
 - CRUD operations, 385
 - delete operations, 388
 - JSON vs. XML, 387
 - nested resources, 389–390
 - overview of, 383–384
 - read operations, 386
 - REST methods (`get`, `post`, `put`, and `delete`) and, 388–389
 - SOAP, XML-RPC, and ActionWebService, 391
 - as standard package in Rails, 44
 - summary, 391
 - update operations, 387–388
- ActiveSupport**
 - `BufferedLogger` class, 224
 - `catrr_reader` method, 199
 - as standard package in Rails, 44
 - utility classes in, 120
- `acts_as_textiled` **plugin, 188**
- adapters, database support and, 234–235**
- `add_index` **method, 202, 240**
- Adobe Flex, 381**
- ADO.NET, 234**
- `after_filter` **method, 339–342**
- aggregator methods, in Enumerable module, 101**
- agile development**
 - Rails and, 5–6
 - Rails culture and, 44
 - testing and, 346
 - XP (Extreme Programming) compared with, 45
- Agile Manifesto, 5–6**
- Ajax**
 - adding to blog example, 377–380
 - frameworks, 381, 418
 - issues with ASP.NET, 50
 - libraries, 418
 - RJS templates and, 359
 - `script.aculo.us`, 375
- alias methods, 86–87**
- `all`, **finder options, 253–254**
- `and` **operator, Ruby, 72**

Apache

Apache

- deploying Rails on GNU/Linux, 413
- installing/configuring, 412
- overview of, 410–411

APP (Atom Publishing Protocol), 391

app directory, 136

app\controllers directory, 157

ApplicationController, 339

application-level performance considerations, 407–409

applications

- blog example. See *blog example*
- characteristics of applications that Rails is suited for, 9
- creating new Rails application, 146–151

app\models

- defining associations and, 204
- models stored in, 156

app\views

- adapting for comments, 207–211
- embedded comments and, 216
- layouts folder, 163–164
- overview of, 163

Aptana RadRails IDE, 25–27

Ares. See *ActiveResource*

ArgumentError method, raising errors in Ruby, 109–110

arguments

- case statement passing, 91
- methods and, 87

around_filter method, 339–342

arrays

- adding conditions to finder, 254–255
- each method for iterating, 96
- hashes as associative arrays, 81–82
- map method, 99
- partition method, 100
- Ruby data types, 79–81
- treating strings as, 75

Article model

- analyzing, 156
- defining associations, 204

articles

- embedding comments in, 215–221
- listing, 150, 157–158
- New Article, 149

articles folder

- app\views, 163
- edit.html.erb template, 170–171
- index.html.erb template, 166–168
- new.html.erb template, 168–170
- show.html.erb template, 171–172

articles.html.erb layout, 164–166, 176

ArticleTest test case, 299–300

as option, customizing RESTful routes, 315

ASCII characters, converting numbers to, 97

ASP.NET

- Code Behind, 319
- developer preferences and, 49
- learning curve and, 49
- main issue with, 50–51
- master pages, 163–164
- MVC framework, 10, 52–53
- overview of, 47–48
- performance considerations, 50
- politics of, 48
- routing compared with Rails routing, 304
- sessions, 332
- similarities/dissimilarities with Rails, 48
- when to use, 51

assert methods, testing routes and, 347

associations

- auto-generated methods and, 273–274
- belongs_to method, 274–277
- callbacks, 292
- defining, 204–206
- eager loading, 281
- extensions, 280–281
- has_and_belongs_to_many method, 279–280
- has_many method, 279–280
- has_one method, 278
- include option, 257
- many-to-many relationships, 271–273
- one-to-many relationships, 271

- one-to-one relationships, 270
 - overview of, 270
 - polymorphic, 286–289
 - Atom (Atom Syndication Format) feeds, 359**
 - adding, 360
 - `format.atom`, 360–361
 - `index.atom.builder`, 361–366
 - linking to, 366–367
 - Atom Publishing Protocol (APP), 391**
 - Atom Syndication Format feeds. See Atom (Atom Syndication Format) feeds**
 - `AtomFeedHelper`, **ActionView**, 369
 - attachments, working securely with e-mail, 396**
 - `attr_accessible`, **protecting records, 399**
 - `attr_accessor`, **Account class, 115–116**
 - `attr_protected`, **protecting records, 399**
 - `attr_reader`, **Account class, 115–116**
 - attributes**
 - assigning/viewing, 251
 - Ruby programming, 112–116
 - `update_attribute/update_attributes` methods, 265–267
 - authentication**
 - filters for HTTP basic authentication, 342–344
 - HTTP authentication, 385
 - improving, 344–346
 - auto-generated methods, associations and, 273–274**
 - auto-implemented properties, C#, 114**
 - `autotest-rails`, 348
 - `average` **method, calculations, 264**
 - AWS (ActionWebService), 391**
- B**
- backslash (\), convention regarding use of, 137**
 - BDD (Behavior-Driven Development)**
 - overview of, 294
 - RSpec and Shoulda and, 348
 - `before_filter` **method, 209, 339–342**
 - Behavior-Driven Development (BDD)**
 - overview of, 294
 - RSpec and Shoulda and, 348
 - `belongs_to`
 - auto-generated methods and, 273–274
 - one-to-many relationships, 271
 - one-to-one relationships, 270
 - `belongs_to`
 - associations, 274
 - example of model employing, 274–276
 - options accepted by, 277
 - `belongs_to_many` , 204
 - `benchmark` **method, 401–402**
 - `benchmark` **script, 403**
 - `between` **operator, Ruby, 72**
 - bigdecimals, Ruby numeric classes, 68**
 - bignums, Ruby numeric classes, 68, 69–70**
 - bindings, blocks and procs associated with, 104**
 - bitwise AND, Ruby operators, 72**
 - bitwise OR, Ruby operators, 72**
 - black box testing, 380**
 - `block_given?` **method, 103**
 - blocks**
 - converting standalone block into `Proc` objects, 104
 - overview of, 95
 - syntax for, 95
 - blog example**
 - associations, defining, 204–206
 - comments added to, 201–203
 - comments embedded in articles, 215–221
 - controller adapted for, 207–211
 - controllers, analyzing, 157
 - `create` action, 160–161
 - creating Rails application for, 146–151
 - creating tables for, 141–144
 - custom environments, 223
 - custom REST action added to, 191–192
 - data types in, 141–144
 - database credentials stored in `config/database`, 137–139

blog example (continued)

blog example (continued)

- database for, 139
- debugging, 227–230
- destroy action, 162–163
- development environment and, 221
- directory structure, 230–231
- edit action, 159
- edit.html.erb template, 170–171
- filtering data, 226–227
- helper methods, 192–193
- index action for listing articles, 157–158
- index.html.erb template, 166–168
- layouts, 163–166
- logging, 223–226
- migrations, 141, 144–146
- models, analyzing, 156
- named route helpers, 155–156
- named_scope used in, 188–191
- nested resources, 206
- new action, 159
- new.html.erb template, 168–170
- pagination of, 197–200
- partials added to, 172–174, 193–195
- production environment and, 222–223
- rails command for creating, 136–137
- RESTful applications and, 151–152
- routes mapped to actions, 153–155
- scaffold generator and, 140–141
- schema-migrations table, 144
- show action, 158–159
- show.html.erb template, 171–172
- style added to, 176–182
- summary, 193–195, 231
- test environment and, 222
- textile support, 184–188
- time zones, 182–184
- update action, 161–162
- validation added to, 174–176
- view layer adapted for, 211–214
- views, analyzing, 163

BlueCloth plugin, 185

book resources, recommendations, 420

Booleans

- Ruby data types, 72–74
- validation of, 283

Bray, Tim, 22

Brooks, Fred, 34

browsers

- cross-browser compatibility and, 3
- Web developers and, 374

Buffered Logger **class**,

ActiveSupport, 224

Builder templates

- Atom feeds, 360, 363, 365–366
- RSS feeds, 360–362, 364–365
- XML and, 357–359

built-in template engines, 357

C

C#

- auto-implemented properties, 114
- exception handling, 107
- Ruby strings compared with C# strings, 76

cache

- MemCacheStore, 335–336
- performance and, 406–407

CacheHelper, **ActionView, 369**

calculations, methods for, 263–265

callbacks, 289–292

- associations and, 292
- executing code during, 291–292
- observers and, 292–294
- order of, 290–291
- overview of, 289
- types of, 290

CamelCase, 246

Capistrano, 414

CaptureHelper, **ActionView, 369**

case equality operator (===), 92

case sensitivity, in Ruby, 58

case statements, 91–93

Castle Project's MonoRail. *See MonoRail*

cattr_reader **method, ActiveSupport, 199**

cd **command, stepping into directory with, 136**

- CDATA sections, 372**
- characters**
 - ASCII, 97
 - Ruby treatment of, 58
- chomp method, Ruby, 76**
- chr method, 97**
- CI (Continuous Integration), 414**
- class keyword, 111**
- class methods**
 - defined, 85
 - finding, 133
 - overview of, 123–124
- class variables, 123**
- classes**
 - defining/instantiating, 111
 - finding, 132–133
 - initialize method, 111–112
 - in modules, 129
 - modules compared with, 126
- closures, blocks and procs as, 104**
- CMS (Content Management System), 34**
- code**
 - controllers defining, 319
 - partials for reusing, 172
- Code Behind, ASP.NET, 319**
- collection option, RESTful routes and, 315–316**
- collections, looping through, 93**
- Comment model**
 - defining associations, 204
 - validation of, 204–205
- comments**
 - adapting Comments controller, 207–211
 - adapting view layer for, 211–214
 - adding, 201–203
 - defining associations and, 204
 - embedding in articles, 215–221
 - nested resources and, 206
- Comments controller, 207–211**
- comments_controller.rb file, 209–211**
- commit monitoring, 414**
- Common Type System (CTS), 63**
- compatibility, cross-browser, 3**
- complex, Ruby numeric classes, 68**
- conditionals**
 - if/elsif/else/unless, 88–90
 - overview of, 88
- conditions, finding with, 254–255**
- conditions option, RESTful routes, 316**
- config directory, 136**
- config/database.yml, 137–139**
- config/environments**
 - creating custom environments, 223
 - development environment, 221
 - production environment and, 222–223
 - test environment, 222
- config/routes.rb, 151, 304**
- configuring Instant Rails, 17–19**
- connect method**
 - customizing routes and, 307–308
 - named routes and, 311–312
- console**
 - reload! command for restarting, 247
 - working with routes, 310–311
- constants**
 - accessing with ::, 127
 - Ruby conventions, 67
- constructors, Ruby literal, 75**
- Content Management System (CMS), 34**
- Continuous Integration (CI), 414**
- controller option, RESTful routes, 316**
- controllers. See also MVC (Model-View-Controller) pattern**
 - ActionController. See ActionController
 - adapting Comments controller, 207–211
 - analyzing blog application controller, 157
 - code defined by, 319
 - defining within modules, 311
 - destroying, 321
 - generating, 320–321
 - managing in MVC approach, 42–43
 - mapping controller actions to URLs, 153
 - partials in shared views, 356
 - testing, 346
 - view layer and, 356–357
 - working with, 319

Convention over Configuration

Convention over Configuration

- in ActiveRecord, 244
- canons of agile development, 5–6
- Rails culture and, 44–46

cookie security, 396

CookieStore, **storing sessions, 334–335**

Core library, ri (Ruby interactive), 59

Corti, Paolo, 412

count **method, 263**

create, rake db:create, **139**

create, read, update, and delete. See CRUD (create, read, update, and delete)

create **action**

- blog application example, 160–161
- comments and, 208

create **method**

- creating new resources, 387
- instantiating objects, 252

create operation. See also CRUD (create, read, update, and delete)

ActiveRecord, 250–252

ActiveResource, 387

create_table **method, 141–144, 241**

cross-browser compatibility, 3

cross-platform basis, of Rails, 4

cross-site scripting attacks. See XSS (cross-site scripting) attacks

CRUD (create, read, update, and delete)

- mapping routes to actions, 153–155
- resources defined by CRUD actions, 152
- scaffold generators and, 140

CRUD (create, read, update, and delete), in ActiveRecord

- calculations, 263–265
- create operation for generating models, 250–252
- customizing SQL queries, 262–263
- delete operations, 269–270
- dynamic finders, 261–262
- finding and limiting or offsetting fetched records, 256
- finding and ordering or grouping results, 254
- finding by first, last, and all, 253–254

finding by id(s), 253

finding with conditions, 254–255

finding with include and join, 257–260

finding with readonly and lock options, 260–261

finding with SELECT and FROM, 256

overview of, 250

partial updates and dirty objects, 265–267

read operation, 253

simplifying approach to, 6

update operations, 265

update_attribute/update_attributes, 265–267

updating multiple records, 267–269

CRUD (create, read, update, and delete), in ActiveRecord, 385

create operation, 387

delete operations, 388

read operations, 386

update operations, 387–388

CTS (Common Type System), 63

curly brackets ({}), in block syntax, 95

customization

environments, 223

helpers, 376

migrations, 240–244

RESTful actions, 191–192

RESTful routes, 315–316

routes, 306–308

SQL queries, 262–263

table names for models, 248–250

Cygwin

managing logs with, 402

One-Click Ruby Installer and, 23

D

data

filtering, 226–227

Rails Web applications storing, 6–7

retrieving with find, 252

send_data method, 324–325

storing session data, 336

data sources, list of supported, 234

- data types, Rails, 141–144**
- data types, Ruby**
 - arrays, 79–81
 - Booleans, 72–74
 - fixnums and bignums, 69–70
 - floats, 70–72
 - hashes, 81–82
 - identifiers and variables, 64–67
 - numeric classes, 67–68
 - objects, 63–64
 - vs. other programming languages, 69
 - ranges, 78–79
 - regular expressions, 77–78
 - strings, 74–77
 - symbols, 77
- databases**
 - adding indexes to, 240
 - creating, 139
 - credentials stored in `config/database.yml`, 137–139
 - installing SQLite3 and `sllite3-ruby`, 16–17
 - Rails Web applications storing data in, 6–7
 - supported, 234–235
 - testing with SQL instead of Active Record's `schema dumper`, 203
- DateHelper, ActionView, 369**
- db:create, rake , 139**
- db:drop, rake , 139**
- db:migrate, rake, 144–145**
- db:rollback, rake, 145**
- db:schema, rake, 202–203**
- debug helper, 357**
- debug level, logger, 225**
- debugger method, 227–230**
- debugging, 227–230**
 - from Console, 230
 - debugger method, 227–230
 - ruby-debug gem, 227
- DebugHelper, ActionView, 369**
- decrement operators, Ruby vs. other programming languages, 70**
- def statement, for defining methods, 86**
- default routes, 305**
- default templates, ActionView, 354**
- DELETE, HTTP methods, 152**
- delete method**
 - deleting rows, 269
 - REST methods in `ActiveResource`, 388–389
- delete operation. See also CRUD (create, read, update, and delete)**
- delete operations**
 - `ActiveRecord`, 269–270
 - `ActiveResource`, 388
- delete_all method, 270**
- deploying Rails**
 - enterprise-related topics, 414–415
 - freezing application and, 415
 - on GNU/Linux, 413
 - history of, 410–411
 - misconceptions regarding, 35
 - source version control tools, 414
 - on Windows OSs, 411
- Deploying Rails Applications: A Step-by-Step Guide, 412**
- destroy action**
 - blog application example, 162–163
 - comments and, 207–208
- destroy method**
 - callback methods and, 290
 - deleting rows, 269
 - destroying existing controllers, 321
 - destroying resources, 388
- destroy script, deleting generated files with, 238**
- developers**
 - adds-on for Web developer, 374
 - ASP.NET vs. Rails, 49
- development, agile. See agile development**
- development environment, 221**
- directory structure, 136, 230–231**
- dirty objects, ActiveRecord, 265–267**
- Distributed Ruby (DRb), 335**
- division operators, Ruby compared with other programming languages, 69**
- divmod method, Ruby, 69**
- do keyword, looping and, 94**

doc folder

doc folder, in directory structure, 230

`doc:guides, rake, 317`

Document Object Model (DOM), 373

documentation, Rails API, 324

DOM (Document Object Model), 373

Domain Specific Languages (DSLs), 2, 131

Don't Make Me Think (Krug), 41

Don't Repeat Yourself (DRY)

canons of agile development, 5–6

Rails culture and, 46–47

dot notation, chaining multiple methods, 86

dot operator (.), for accessing methods, 127

`down` **class method, migrations, 145–146**

`downto` **iterator, numeric class, 95**

DRb (Distributed Ruby), 335

DrbStore, 335

drivers, database support and, 234

`drop, rake db:drop, 139`

DRY (Don't Repeat Yourself)

canons of agile development, 5–6

Rails culture and, 46–47

DSLs (Domain Specific Languages), 2, 131

Duck Typing, 56, 122

dynamic finders

`ActiveRecord, 131–132`

overview of, 261–262

SQL injection attacks and, 397–398

E

each method, 96–98

iterating arrays, 96

iterating files, 98

iterating over ranges, 79, 96

iterating strings, 97

eager loading, associations, 281

`edit` **action**

blog application example, 159

comments and, 207

`edit.html.erb` **template, 170–171, 212**

eigenclasses

finding, 132–133

singleton methods and, 124–126

`else` **conditional, 89–90**

`elsif` **conditional, 89–90**

Emac text editor, 24

e-mail, ActionMailer and, 381

embedding comments, in articles, 215–221

encryption, SSL, 385

`end` **statement, terminating method**

definitions, 86

enterprise-related topics

overview of, 414–415

when to use ASP.NET instead of Rails, 51

`Enumerable` **module, 101**

environments

defining custom, 223

development, 221

production, 222–223

test, 222

equality operator (==), 81, 92

ERb templates

`ActionView, 357–358`

overview of, 41–42

`error` **level, logger, 225**

errors

raising, 109–110

“Template is missing”, 354

Erubis template engine, 42

escaping HTML tags, 394

`except` **helper, textual helpers, 367**

exception handling, 107–109

`execute` **method, SQL statements, 240**

expressions

`<%=h` tag when evaluating, 358

statements and, 90

`extend` **method, 130–131**

extensibility, of Ruby, 56

extensions, association, 280–281

Extreme Programming. See XP

(Extreme Programming)

F

FastCGI, 410

`fatal` **level, logger, 225**

feeds, RSS and Atom, 359

adding, 360

- format.rss and format.atom, 360–361
 - index.rss.builder and index.atom.builder, 361–366
 - linking to, 366–367
 - order of, 363
 - file_field helper, 371
 - files**
 - each method for iterating, 98
 - send_file method, 324–325
 - uploading, 371
 - FileStore, storing session data, 336
 - filter methods, 101**
 - filter_parameter_logging method, 227
 - filters**
 - HTTP authentication, 342–344
 - logging and, 226–227
 - overview of, 339–342
 - find method
 - ActiveResource, 386
 - obtaining list of records with, 237
 - records not found and, 262
 - retrieving data with, 252
 - find_by_sql method, 263
 - finders**
 - customizing SQL queries, 262–263
 - dynamic finders, 261–262
 - finding and limiting or offsetting fetched records, 256
 - finding and ordering or grouping results, 254
 - finding by first, last, and all, 253–254
 - finding by id(s), 253, 400
 - finding with conditions, 254–255
 - finding with include and join, 257–260
 - finding with readonly and lock options, 260–261
 - finding with SELECT and FROM, 256
 - SQL injection attacks and, 397–398
 - Firebug adds-on, for Web developers, 374**
 - Firefox, Mozilla, 374**
 - FireWatir, Rails tools, 380**
 - first, finder options, 253–254
 - FiveRuns, 405**
 - fixnums, Ruby data types, 67, 69–70
 - Flash objects, 330–332**
 - Flex, Adobe, 381**
 - floats, Ruby data types, 68, 70–72**
 - for keyword, 93
 - foreign key columns, adding indexes to, 201–202**
 - foreign key constraints**
 - adding to migrations, 242–244
 - applying at application level vs. database level, 240
 - for/in loops, 93–94
 - form_for arguments, 169, 212
 - format.atom, 360–361
 - format.rss, 360–361
 - formats, Rails supported, 158**
 - FormHelper, ActionView, 370–371
 - FormOptionsHelper, ActionView, 372
 - FormTagHelper, ActionView, 370–371
 - forward slash (/), convention regarding use of, 137**
 - Fowler, Martin, 233**
 - fragment caching, 406–407**
 - frameworks**
 - Ajax, 381, 418
 - ASP.NET MVC framework, 10
 - BDD, 348
 - Convention over Configuration and, 46
 - full-stack, 4
 - JavaScript frameworks, 373
 - from, finder options, 256
 - full-stack framework, Rails as, 4**
 - functional testing, 348–351**
- ## G
- generators**
 - migration generator, 144, 239–240
 - model generator, 40, 237–239
 - scaffold generator. See scaffold generator
 - GET, HTTP methods, 152, 329
 - get, REST methods in ActiveResource, 388–389
 - getter/setter methods, Ruby, 114–115**
 - GitHub, 346, 414

GNU/Linux

GNU/Linux

- deploying Rails on, 413
- installing Rails on, 20–21
- Rails on, 11–12

grep, **managing logs with**, 402

group, **finder options**, 254

guid, **as primary key for tables**, 249–250

guides, generating Rails guides, 317

H

h helper. **See** *html_escape*

habtm, **see** *has_and_belongs_to_many*

hackers, 393

Haml template engine, 42, 381

Hansson, David Heinemeier, 8

has_and_belongs_to_many
auto-generated methods and, 273
overview of, 279–280

has_many
auto-generated methods and, 273
customizing RESTful routes, 316–317
one-to-many relationships, 271
overview of, 279–280

has_one
auto-generated methods and, 273
customizing RESTful routes, 316–317
defining associations, 204
one-to-one relationships, 270
overview of, 278

hashes

- adding conditions to finder, 254–255
- Ruby data types, 81–82

Hello world program, in Ruby, 56–59

help resources, 419–420

helper **method**, 376

helper_method, 376

helpers

- ActionView, 367–368
- creating, 376–377
- defining, 192–193
- ERb and, 357
- FormHelper and FormTagHelper, 370–371

FormOptionsHelper, 372
JavaScriptHelper,
 PrototypeHelper, and
 ScriptaculousHelper,
 373–375

MigrationHelpers, 243

other options, 375–376

pluralize helper method, 215

predefined helpers, 368–369

RESTful route helpers, 319

sanitize helper method, 185

validation, 283–284

Hibbs, Curt, 17

hosting services, resources for, 419

HTML

- formats supported by Rails, 158
- representation formats, 152
- resources for, 417

HTML tags

- escaping, 394
- sanitizing, 394–396

html_escape, 394

HTTP, representation formats, 152

HTTP authentication

- filters for, 342–344
- Web services and, 385

HTTP headers, request/response environment an, 326–327

HTTP methods

- ActiveResource using, 384
- defining RESTful routes, 313–314
- mapping to controller actions, 153
- requests, 152

httpperf, **for stress testing**, 404

I

identifiers, 64–67

- for constants, 67
- mapping tables to models and, 244
- for methods, 66–67
- overview of, 64
- snake_case convention, 64
- for variables, 64–66

IDEs (Integrated Development Environments)

Aptana RadRails, 25–27
 NetBeans, 25
 overview of, 21–22
 Ruby in Steel IDE, 27–28
 RubyMine IDE, 27

ids

finding by, 253
 as primary key for tables, 249–250
 vulnerabilities and, 399

IE (Internet Explorer), 374

if **conditional**, 88–90

IIS (Internet Information Services), 412**in**

for/in loops, 93–94
 in keyword, 93

include **method, vs. extend method**,
130–131

include **option, finders**, 257–260

**increment operators, Ruby vs. other
 programming languages**, 70

index action

listing articles, 157–158
 listing comments, 207

index.atom.builder, 361–366

indexes

add_index method, 202, 240
 adding to foreign key columns, 201–202
 polymorphic tables and, 287

index.html.erb **template**, 166–168,
213–214

index.rss.builder, 361–366

inflection

adding new inflection rules, 247–248
 Inflector class, 245–246

info **level, logger**, 225

inheritance, single table, 117–120,
284–286

initialize **method**, 111–112

initializers, 234

installing Mongrel, 16

installing Rails

on GNU/Linux, 20–21
 on Mac OS X, 20

One-Click Ruby Installer, 12–13
 on Windows OSs, 12, 14–15

installing RubyGems, 13–14

installing SQLite3 and sqllite3-ruby, 16–17

installing Subversion (SVN), 17

instance methods

defined, 86
 finding, 132–133

instance variables, 114

Instant Rails, configuring, 17–19

Integer **class**, 95

Integrated Development Environments.

*See IDEs (Integrated Development
 Environments)*

integration testing, 351

Interactive Ruby (IRB)

overview of, 59
 using, 60–61

internationalization

Rails and, 245
 restful-authentication-118, 346

Internet Explorer, 374

Internet Information Services (IIS), 412

IP addresses, cookie security and, 396

IRB (**Interactive Ruby**)

overview of, 59
 using, 60–61

IronRuby

deploying, 412
 Ruby versions, 134
 software alternatives for using with Rails, 381

iterators

common, 99–101
 defining, 102–104
 each method, 96
 numeric, 95–96
 overview of, 95

J

JavaScript

libraries, 377
 rendering in, 354
 resources for, 417–418
 RJS templates, 357, 359

JavaScript Object Notation

JavaScript Object Notation. See JSON (JavaScript Object Notation)

`javascript_include_tag`, 376

`JavaScriptHelper`, `ActionView`, 373–375

join models, many-to-many relationships and, 272–273

`join` option, finders, 257–260

join tables, many-to-many relationships and, 271–272

`jQuery`, 381

`JRuby`, 134

JSON (JavaScript Object Notation)

ActiveResource using, 387

for communication among machines, 383

rendering in, 354–355

representation formats, 152

K

Kernel methods, 86

`Kernel#lambda` method, 104–106

`Kernel#raise` method, 109–110

L

`last`, finder options, 253–254

`layout` method, adapting `Comments` controller, 207

layouts

`articles.html.erb` layout, 164–166

compared with ASP.NET master pages, 163–164

partials in, 356

`layouts` folder, `app\views`, 163–164

learning curve, ASP.NET vs. Rails, 49

legacy databases, when to use ASP.NET instead of Rails, 51

lib folder, in directory structure, 230

libraries

Ajax, 418

Core and Standard, 59

JavaScript, 377, 381

Ruby, 186

Ruby's SOAP library, 391

licensing Rails, 3, 48

Lighttpd (Lighty), 410

`limit` option, finders, 256

links

to comments, 215

resources for HTML, 417

resources for JavaScript, 417–418

resources for Ruby on Rails, 418–419

to RSS and Atom feeds, 366–367

Lisp language, 56

literal constructors, Ruby, 75

local variables, controllers and, 330

localization, `country_select`, 372

`lock` option, finders, 260–261

log analyzers, 402

`log\development`, 223

`logger`

security levels and, 226

using, 225

logs/logging, 223–226

filtering data and, 226–227

overview of, 223–224

performance logs, 401–402

production logs, 225–226

redirecting logging to console, 226

security levels and, 225

SQL queries and, 254

tools for managing, 402

looping. See also iterators

`for/in` loops, 93–94

overview of, 93

`while/until` loops, 94

M

Mac OS X

installing Rails, 20

Rails and, 11

MacRuby, Ruby versions, 134

MagLev, Ruby versions, 134

many-to-many relationships, 271–273

join models and, 272–273

join tables and, 271–272

`map` method, arrays, 99

`map.connect`, for defining routes, 305

- mapping tables to models, 244–245**
- `map.resource`, **defining RESTful routes, 314–315**
- `map.resources`, **defining RESTful routes, 313–314**
- Markdown, 185**
- master pages, ASP.NET, 163–164**
- Matsumoto, Yukihiro, 7**
- maximum method, calculations, 264**
- `member` **option, RESTful routes, 315–316**
- MemCacheStore, 335–336**
- `MemoryStore`, **336**
- metaprogramming, 131–132**
- method overload, Ruby not supporting, 87**
- `method_missing` **method**
 - action processing and, 321–322
 - as dynamic finder, 261–262
 - overview of, 132
- methods**
 - arguments and, 87
 - class methods, 123–124
 - defining Ruby, 85–88
 - `include` `method` vs. `extend` `method`, 130–131
 - name resolution, 132–133
 - Ruby, 66
 - singleton methods, 121–122
 - visibility, 116–117
- Microsoft SourceSafe, 414**
- Microsoft Web Application Stress Tool, 404**
- `migrate`, `rake db:migrate`, **144–145**
- migration generator**
 - creating migration files, 144
 - generating migrations with, 239–240
- MigrationHelpers, 243**
- migrations**
 - adding foreign keys to, 242–244
 - customizing, 240–244
 - generating, 239–240
 - old style, 145
 - overview of, 144–145
 - specifying primary keys and, 250
 - up and down class methods, 145–146
- minimum method, calculations, 264**
- misconceptions regarding Rails**
 - deployment and, 35
 - overview of, 33
 - programming skill requirements, 33
 - scalability and, 35–36
 - silver bullet myth, 34–35
- MIT licensing, 3, 48**
- mixins, modules as, 129–130**
- `mod_rails` (**Phusion Passenger**)
 - deploying Rails on GNU/Linux, 413
 - overview of, 393
 - Rails deployment and, 35, 411
- `model` **generator**
 - creating models with, 237–239
 - Rails, 40
- models. See also MVC (Model-View-Controller) pattern**
 - as ActiveRecord objects, 233
 - analyzing blog application model, 156
 - create operation for generating, 250–252
 - creating ActiveRecord models, 384–385
 - defining in MVC approach, 38–40
 - generating with ActiveRecord, 237–239
 - mapping tables to, 244–245
 - ORM (Object Relational Mapping)
 - and, 237
 - setting custom table names for models, 248–250
 - testing, 294–301
 - view layer interaction with, 356
- Model-View-Controller pattern. See MVC (Model-View-Controller) pattern**
- `module`, **127**
- `Module` **class, 115**
- modules**
 - classes in, 129
 - compared with classes, 126
 - defining controllers within, 311
 - finding, 132–133
 - as mixins, 129–130
 - as namespaces, 127–129
- `modulo` **method, Ruby, 69**

Mongrel

Mongrel

- components need when installing Rails, 12
- configuring, 412
- installing, 16
- `ruby script/server` command for starting, 147
- as Web server, 410–412

Monit, 413

monitoring performance, 404–405. See also performance

monkey patching, Ruby programming, 120–121

MonoRail

- MVC framework for .NET, 52
- overview of, 10

Mozilla Firefox, 374

MVC (Model-View-Controller) pattern

- ActiveRecord implementing model component of, 233
- adapting `Comments` controller, 207–211
- adapting view layer for comments, 211–214
- analyzing blog application controller, 157
- analyzing blog application model, 156
- analyzing blog application view, 163
- applying to software development, 4–5
- ASP.NET MVC, 52–53
- defining models, 38–40
- designing views, 41–42
- interaction of model, view, and controller, 36–38
- managing controllers, 42–43
- MonoRail framework, 52
- overview of, 36

MySQL, 138

N

name resolution, 132–133

`name_prefix` option, for RESTful routes, 317

named routes, 311–312

namespaced resources, 318–319

namespaces, modules as, 127–129

naming conventions

- adding new inflection rules, 247–248
- Convention over Configuration and, 46
- `Inflector` class, 245–246
- mapping tables to models and, 244–245
- setting custom table names for models, 248–250

NAnt, Rake compared with, 139

nested resources, 389–390

- comments and, 206
- `comments_controller.rb` file adjusted for, 209–211

.NET, consuming/publishing REST services from, 388–389

NetBeans IDE, 25

new action, blog application example, 159

new option, RESTful route, 315–316

New Relic RPM service, 405

`new.html.erb` template, 168–170, 212

Nginx Web servers, 325, 410–411

`nmake`, Rake compared with, 139

not operator, Ruby, 72

`NumberHelper`, 375

numeric classes, Ruby, 67–68

numeric iterators, Ruby, 95–96

O

object creation, callback methods, 290

Object Relational Mapper. See ORM (Object Relational Mapper)

object-oriented languages, 55

objects

- defining/instantiating classes, 111–112
- finding, 132–133
- referencing current, 112
- Ruby data types, 63–64

observers, ActiveRecord, 292–294

offset option, finders, 256

One-Click Ruby Installer, 12–13, 23

one-to-many relationships, 271

one-to-one relationships, 270

online documentation, Rails API, 324

open source, 3–4

operators

- for accessing constants and methods, 127
- case equality, 92
- decrement/increment, 70
- division, 69
- equality operator, 81, 92
- list of Ruby, 72, 74
- splat, 79, 87
- ternary, 90

optimistic locking, 261

or **operator, Ruby, 72**

order **attribute, finder options, 254**

ORM (Object Relational Mapper). See

also ActiveRecord

- ActiveRecord and, 237
- defining models in MVC architecture, 38–40
- Rails use of, 6

orthogonality, DRY principle and, 47

P

p method, vs. puts command, 80

packages, standard in Rails, 44

page caching, 406

pagination

- adding pagination menu to view, 200
- adding to blog example, 197–200
- overview of, 197
- `per_page` method and, 199–200
- `will_paginate` plugin, 198–199

pagination menu, 200

Paperclip plugin, for attachment security, 396

parameterized queries, ActiveRecord, 263

partial updates, ActiveRecord, 265–267

partials

- ActionView and, 356
- adding to blog application example, 172–174, 193–195
- reusability and, 47

partition method, arrays, 100

patches, 402

path_prefix option, for RESTful routes, 317

Patterns of Enterprise Application

Architecture (Fowler), 6, 233

`per_page` **method, of pagination, 199–200**

performance

- application-level considerations, 407–409
- ASP.NET vs. Rails, 50
- caching and, 406–407
- logs, 401–402
- measuring, 400
- monitoring, 404–405
- scalability and, 405
- tools for, 402–404
- when to use ASP.NET instead of Rails, 51

performance folder, scripts in, 403

Performance Testing Rails Applications guide, 404

Perl language, 56

pessimistic locking, 261

Phusion Passenger (`mod_rails`)

- deploying Rails on GNU/Linux, 413
- overview of, 393
- Rails deployment and, 35, 411

plain text

- rendering in, 354
- representation formats, 152

platforms, Rails cross-platform basis, 3

pluralize helper, textual helpers, 215, 217, 367

pluralize method, of Inflector class, 245

POLS (principle of least surprise), 56

polymorphic associations, ActiveRecord, 286–289

POST, HTTP methods, 152, 329

post, REST methods in ActiveResource, 388–389

predefined helpers, ActionView, 368–369

prefixes, specifying for table names, 249

primary keys, tables, 249–250

principle of least surprise (POLS), 56

principles

- Convention over Configuration, 44–46
- DRY (Don't Repeat Yourself), 46–47
- overview of, 44

private methods

private methods

- method visibility and, 116–117
- security and, 399

`Proc.new` **method**, 104–106

production

- application-level performance
 - considerations, 407–409
- caching and, 406–407
- cookie security, 396
- costly mistakes, 399
- deploying on GNU/Linux, 413
- deploying on Windows OSs, 411
- deployment history, 410–411
- enterprise-related topics, 414–415
- escaping HTML tags, 394
- logs, 401–402
- measuring performance, 400
- monitoring performance, 404–405
- overview of, 393
- record protection, 398–399
- Ruby on Rails Security Guide, 399
- sanitize HTML tags, 394–396
- source version control tools, 414
- SQL injection attacks, 397–398
- summary, 416
- tools for measuring performance, 402–404
- upgrading to Rail 2.3, 415–416
- XSS (cross-site scripting) attacks and, 394

production environment, 222–223

Production Log Analyzer, 402

production logs, 225–226

productivity, Convention over Configuration and, 46

`profiler` **script**, 403

profiling

- with RailsBench, 402
- with `ruby-prof`, 402–403

programming languages

- Rails compared with, 7
- Ruby compared with, 69

programming skills, Rails misconceptions, 33

protected methods, visibility and, 116–117

Prototype

- Ajax and, 381
- JavaScript frameworks, 373
- PrototypeHelper, 373–375
- PStore, for storing sessions**, 336

public methods

- security and, 399
- visibility and, 116–117

publishing REST services, from .NET, 388–389

`PUT`, **HTTP method**, 152

`put`, **REST methods in ActiveResource**, 388–389

`puts` **command**

- vs. `p` method, 80
- printing in Ruby, 58

Q

Quicksort algorithm, 99–100

R

RadRails IDE, Aptana, 25–27

`rails` **command**

- creating blog application, 136–137
- creating new Rails applications, 135

Rails introduction. See *Ruby on Rails, introduction*

RailsBench, measuring performance and profiling with, 402

`rails:update`, `rake`, 416

raising errors, 109–110

Rake

- creating databases, 139
- `rake db:create`, 139
- `rake db:drop`, 139
- `rake db:migrate`, 144–145
- `rake db:rollback`, 145
- `rake db:schema`, 202–203
- `rake doc:guides`, 317
- `rake rails:update`, 416
- `rake routes`, 155
- `rake stats`, 346

ranges

- each method for iterating, 96
- Ruby data types, 78–79

rational, Ruby numeric classes, 68**read operations, ActiveRecord. See also**

CRUD (create, read, update, and delete)

- calculations, 263–265
- customizing SQL queries, 262–263
- dynamic finders, 261–262
- finding and limiting or offsetting fetched records, 256
- finding and ordering or grouping results, 254
- finding by first, last, and all, 253–254
- finding by id(s), 253
- finding with conditions, 254–255
- finding with `include` and `join`, 257–260
- finding with `readonly` and `lock` options, 260–261
- finding with `SELECT` and `FROM`, 256
- overview of, 253

read operations, ActiveResource, 386.

See also CRUD (create, read, update, and delete)

readonly option, finders, 260–261**Really Simple Syndication feeds. See RSS (Really Simple Syndication) feeds****RecordIdentificationHelper, 376****records. See also ActiveRecord**

- adding new records to table, 250–252
- inability to find, 262
- pessimistic and optimistic locking, 261
- protecting, 398–399

RecordTagHelper, 376**RedCloth library, Ruby, 186****redirect_to method, 325–326****Reenskaug, Trygve, 36****references types, Ruby types contrasted with, 63****Regexp class, Ruby, 77–78****regular expressions, Ruby data types, 56, 77–78****relationships. See associations****reload! command, for restarting console, 247****remainder method, Ruby, 69****remove_index method, 202****render method, 194**

- ActionController, 322–324
- bypassing templates, 354–355
- rendering specific template, 355

reports, Ruby gem Ruport, 324**REpresentational State Transfer. See REST (REpresentational State Transfer)****request handling. See ActionController****Request Log Analyzer, 402****request object, request/response environment, 327–329****request script, 403****request/response cycle, 303****request/response environment, 326–327****requests, ActionController, 305****rescue clause, exception handling and, 107–109****resources. See also ActiveResource**

- creating new, 387
- CRUD actions and, 152
- destroying, 388
- namespaced resources, 318–319
- nested, 206, 389–390
- saving, 387–388

resources method, RESTful routes, 313–314**response object, request/response environment, 327****REST (REpresentational State Transfer)**

- ActiveResource and, 388–389
- consuming/publishing REST services from .NET, 390
- creating RESTful controllers, 320
- customizing RESTful actions, 191–192
- customizing RESTful routes, 315–317
- mapping REST resources, 384
- overview of, 152
- RESTful applications, 151–152
- RESTful route helpers, 319
- RESTful routes, 312–315

REST Starter Kit

REST Starter Kit, WCF (Windows Communication Foundation), 390

reusability, partials and, 47

ri (Ruby interactive)

overview of, 59

using, 61–63

RIAs (Rich Internet Applications), 2

Riding Rails blog, 400

RJS templates, 41, 359

rollback, rake db:rollback, **145**

root **method, 311**

RoR. See Ruby on Rails, introduction

route globbing, 307

routes

console for working with, 310–311

customizing, 306–308

customizing RESTful routes, 315–317

defining, 305

mapping to actions, 153–155

name route helpers, 155–156

named routes, 311–312

priority, 309–310

RESTful route helpers, 319

RESTful routes, 312–315

testing, 346–348

routes , rake, **155**

routing, ASP.NET vs. Rails, 304

rows, deleting, 269

RPM service, New Relic, 405

RSpec, BDD framework, 348

RSS (Really Simple Syndication) feeds, 359

adding, 360

format.rss, 360–361

index.rss.builder, 361–366

linking to, 366–367

Rubinius, Ruby versions, 134

Ruby

arrays, 79–81

Booleans, 72–74

components need when installing Rails, 12

constants, 67

fixnums and bignums, 69–70

floats, 70–72

hashes, 81–82

Hello world program, 56–59

identifiers, 64

IRB (Interactive Ruby), 60–61

methods, 66–67

numeric classes, 67–68

objects, 63–64

vs. other programming languages, 69

overview of, 7

ranges, 78–79

regular expressions, 77–78

ri (Ruby interactive), 61–63

SOAP library, 391

strings, 74–77

summary, 83

symbols, 77

template engines, 42

useful tools, 59

variables, 64–66

what it is, 55–56

Ruby, programming with

attributes and accessor methods, 112–116

blocks and iterators, 95

case statement, 91–93

class methods, 123–124

common iterators, 99–101

conditionals, 88

defining iterators, 102–104

defining methods, 85–88

defining/instantiating classes, 111

each method, 96–98

eigenclasses, 124–126

exception handling, 107–109

for/in loops, 93–94

if/elsif/else/unless, 88–90

include method vs. extend method,
130–131

initialize method, 111–112

looping, 93

metaprogramming, 131–132

method name resolution, 132–133

method visibility, 116–117

modules as mixins, 129–130

- modules as namespaces, 127–129
 - modules compared with classes, 126
 - monkey patching, 120–121
 - numeric iterators, 95–96
 - overview of, 85
 - `Proc.new` and `Kernel#lambda`
 - methods, 104–106
 - raising errors, 109–110
 - single inheritance, 117–120
 - singleton methods and eigenclasses, 121–122
 - summary, 134
 - ternary operator, 90
 - versions of, 133–134
 - `while/until` loops, 94
 - `ruby`, **Ruby interpreter, 57**
 - Ruby 1.9, 133**
 - Ruby in Steel IDE, 27–28**
 - Ruby interactive (`ri`)**
 - overview of, 59
 - using, 61–63
 - Ruby on Rails, introduction**
 - advantages of, 7–8
 - agile development and, 5–6
 - Aptana RadRails IDE, 25–27
 - configuring Instant Rails, 17–19
 - cross-platform basis of, 4
 - database installation, 16–17
 - database-driven, 6–7
 - example project, 29–30
 - as full-stack framework, 4
 - history of, 8–9
 - IDEs and, 21–22
 - installing on GNU/Linux, 20–21
 - installing on Mac OS X, 20
 - installing on Windows OSs, 12, 14–15
 - MVC pattern and, 4–5
 - NetBeans IDE, 25
 - One-Click Ruby Installer, 12–13
 - open source basis of, 3–4
 - popularity of, 10–11
 - resources for, 418–419
 - Ruby in Steel, 27–28
 - Ruby programming language, 7
 - RubyGems and, 13–14
 - RubyStack installer, 21
 - summary, 31
 - TextMate editor, 22–24
 - upgrading to Ruby on Rails 2.3, 415–416
 - version control for, 17
 - Vi and Emacs text editors, 24
 - Web 2.0 and, 10
 - Web server for, 16
 - what it is, 3
 - Windows vs. Unix/Linux platforms, 11–12
 - Ruby on Rails Security Guide, 399**
 - `ruby script/server` **command, for starting Web server, 147**
 - `ruby-debug` **gem, 227**
 - RubyGems**
 - components need when installing Rails, 12
 - improvements in, 15
 - installing/updating, 13–14
 - RubyMine IDE, 27**
 - `ruby-prof`
 - patch required for using, 404
 - profiling with, 402–403
 - RubyStack installer, 21**
 - runtime environments**
 - defining custom, 223
 - development environment, 221
 - production environment, 222–223
 - test environment, 222
 - `RuntimeError` **method, 109–110**
 - Ruport, Ruby gem, 324**
- S**
- SaaS (Software as a Service)**
 - Rails use by SaaS vendors, 49
 - writing CSS and, 381
 - `sanitize`
 - HTML tags, 394–396
 - using for potentially dangerous tags, 358
 - sanitize helper method, 185**

SanitizeHelper

`SanitizeHelper`, **375**

`save` **method**, **387–388**

`scaffold` **generator**

creating blog engine with, 140

creating `migrate` directory with, 140–141

creating RESTful controllers, 320

generating migration files, 144

Rails and, 40

view files generated by, 163

scalability

misconceptions regarding, 35–36

performance and, 405

`schema`, `rake db:schema`, **202–203**

`schema-migrations` **table**, **144–145**

SciTE editor, **57**

scope, using `named_scope` in blog example, **188–191**

Scout, **405**

`script.aculo.us`, **375**

scripting languages, Ruby as, **56**

scripts

in performance folder, 403

sample script using `ActiveRecord`, 235–237

`ScriptttaculousHelper`, **373–375**

security

cookie security, 396

escaping HTML tags, 394

record protection, 398–399

Ruby on Rails Security Guide, 399

sanitize HTML tags, 394–396

SQL injection attacks, 397–398

XSS (cross-site scripting) attacks, 394

`select` **method**, **Quicksort algorithm** and, **99–100**

`select` **option**, **finders**, **256**

`SELECT` **queries**, **197**

Selenium, **Rails tools**, **380**

`self`, **referencing current object**, **112**

`send_data` **method**, **324–325**

`send_file` **method**, **324–325**

serializing records, **ActiveRecord**, **289**

servers. See *Web servers*

Service-Oriented Architecture (SOA), **391**

Session Fixation Attacks, **396**

session IDs (SIDs), **396**

`session` **method**, for enabling/disabling **sessions**, **337–338**

sessions

`ActiveRecordStore`, 333

`CookieStore`, 334–335

`DrbStore`, 335

enabling/disabling, 337–338

expiration of, 338–339

`MemCacheStore`, 335–336

other storage options, 333–336

overview of, 332–333

`PStore`, 336

`shared` **methods (VB.NET)**, **123**

shared views, **partials** in, **356**

Shaw, Zed, **410**

Shoulda, **BDD framework**, **348**

`show` **action**

blog application example, 158–159

comments and, 207

`show.html.erb` **template**

comments and, 213–214

embedded comments and, 216

overview of, 171–172

SIDs (session IDs), **396**

sigils, **defining scope in Ruby with**, **65–66**

silver bullet myth, **Rails misconceptions**, **34–35**

Silverlight, **381**

Silverline plug-in, **381**

`simple_format`, for adding line breaks, **184–185**

single inheritance, **117–120**, **284–286**

Single Responsibility Principle, **293**

singleton methods

defining, 121–122

eigenclasses and, 124–126

finding, 132–133

overview of, 86

- singular option, customizing RESTful routes, 317
 - Smalltalk, 56
 - snake_case convention, 64, 246
 - SOA (Service-Oriented Architecture), 391
 - SOAP, 391
 - software
 - alternatives for using with Rails, 381
 - Convention over Configuration in design of, 46
 - Software as a Service (SaaS)
 - Rails use by SaaS vendors, 49
 - writing CSS and, 381
 - source version control. *See* SVN (Subversion)
 - splat operator (*)
 - defining methods that accept multiple arguments, 87
 - for expanding range into arrays, 79
 - SQL injection attacks, 397–398
 - SQL queries. *See also* finders
 - customizing, 262–263
 - include and join options, 258–260
 - logs, 254
 - SQL Server, 142–143
 - SQL Server Management Studio, 139
 - SQL statements, execute method for using, 240
 - SQLite3
 - components need when installing Rails, 12
 - default config/database.yml, 137
 - installing, 16–17
 - support for foreign keys, 244
 - sqlite3 command, 139
 - sqlite3-ruby
 - components need when installing Rails, 12
 - installing, 16–17
 - SSL encryption, 385
 - Standard library, 59
 - standard packages, 44
 - state management, ActionController, 329–330
 - statements, expressions and, 90
 - static methods (C#), 123
 - stats, rake, 346
 - step iterator, numeric class, 96
 - stored procedures, ActiveRecord and, 263
 - stress testing, 404
 - String Interpolation, 74
 - string literals, Ruby syntax for, 58
 - strings
 - adding conditions to finder, 254
 - each method for iterating, 97
 - Ruby data types, 74–77
 - symbols and, 77
 - strongly typed language, Ruby as, 56
 - style, adding to blog application example, 176–182
 - Subversion. *See* SVN (Subversion)
 - suffixes, specifying for table names, 249
 - sum method, calculations, 264
 - summarizer methods, Enumerable module, 101
 - SVN (Subversion)
 - components need when installing Rails, 12
 - installing, 17
 - overview of, 414
 - switch statements, 91–93
 - symbols, Ruby data types, 77
- ## T
- tables
 - adding new records to, 250–252
 - create_table method, 241
 - defining, 141
 - mapping to models, 244–245
 - polymorphic associations, 286–289
 - prefixes and suffixes for custom tables, 249
 - primary keys, 249–250
 - schema-migrations table, 144
 - setting custom table names for models, 248–250
 - single inheritance, 284–286
 - TagHelper, 376

TDD

TDD (test driven development)

- popularity in Rail community, 44
- Rails community embracing, 294

templates, Rails, 41

templates, view

- Builder templates, 358–359
- built-in template engines, 357
- default templates, 354
- ERb templates, 357–358
- render method for bypassing templates, 354–355
- render method for rendering specific template, 355
- RJS templates, 359

ternary operator (?:), Ruby programming, 90

test driven development (TDD)

- popularity in Rail community, 44
- Rails community embracing, 294

test environment, 222

test folder, in directory structure, 230

Test-First Development, 294

testing

- ArticleTest test case, 299–300
- benefits of, 294
- controllers, 346
- functional testing, 348–351
- models, 294–301
- performance and stress tests, 404
- routes, 346–348
- unit tests, 294–295, 300
- validation, 297–298

text editors

- TextMate, 22–24
- Vi and Emacs, 24

TextHelper, 375

Textile, adding Textile support to blog application example, 184–188

textilize method, 186–187

TextMate editor, 22–24

time zones, setting for blog application example, 182–184

times method, as numeric iterator, 95

timestamps, migrations and, 144

tmp folder, in directory structure, 230

tools, performance, 402–404

tools, Ruby

- IRB (Interactive Ruby), 60–61
- overview of, 59
- ri (Ruby interactive), 61–63

transactions, unit testing and, 300

TuneUp and Manage services, FiveRuns, 405

U

Unicode, when to use ASP.NET instead of Rails, 51

Uniform Resource Identifiers (URIs), 152

unit testing

- defined, 294
- transactions and, 300

Unix. See GNU/Linux

unless conditional, 90

unpublished action, 191

until, while/until loops, 94

up class method, migrations, 145–146

update, rake rails:update, 416

update action

- blog application example, 161–162
- comments and, 207–208

update method, 267–269

update operations, ActiveRecord. See also CRUD (create, read, update, and delete)

- overview of, 265

- update_attribute/update_attributes methods, 265–267
- updating multiple records, 267–269

update operations, ActiveResource, 387–388. See also CRUD (create, read, update, and delete)

update_all method, 267–269

update_attribute method, ActiveRecord, 265–267

update_attributes method, ActiveRecord, 265–267

updates, callback methods and, 290
upgrading to Rail 2.3, 415–416
uploading files, 371
 upto **iterator, numeric class, 95**
URLs (Uniform Resource Identifiers), 152
 UrlHelper, **375**

URLs

default format in Rails, 151–152
 mapping to actions, 153
 Rails and, 37
 security of, 399

user interface. See ActionView

UTC

migration timestamps, 144
 setting default time zone, 182–184

utility classes, ActiveSupport, 120

V

valid_password **method, 344**
 validate **method, 281–282**
 validate_on_create **method, 281–282**
 validate_on_update **method, 281–282**
 validates_format_of, **205**

validation

ActiveRecord, 281–283
 adding to blog application example,
 174–176
 callback methods and, 290
 of Comment model, 204–205
 helpers, 283–284
 testing, 297–298

value types, Ruby types contrasted with, 63

variables, 64–66

class variables, 123
 declaring local, 330
 instance variables, 114
 not switching types, 64–65
 options for creating, 252
 sigils for defining scope of, 65–66
 snake_case convention, 64

VB (Visual Basic), 107

VB.NET, 76

vendor folder, in directory structure, 230

verbs. See HTTP methods

verify method, filtering with, 342

version control. See SVN (Subversion)

versions, Ruby, 133–134

Vi text editor, 24

views. See also MVC

(Model-View-Controller) pattern

adapting, 211–214
 analyzing blog application view, 163
 designing in MVC approach, 41–42
 interaction with models, 356
 pagination menu added to, 200
 templates. *See templates, view*
 view layer communicating with other layers,
 356–357

ViewState, issues with ASP.NET, 50

Virtual Private Server (VPS), 35, 411

visibility, of methods, 116–117

Visual Studio, 27–28

Visual Studio Team System, 414

VPS (Virtual Private Server), 35, 411

vulnerabilities

ids and, 399
 XSS (cross-site scripting) attacks, 393, 394

W

warn **level, logger, 225**

Watir (Web Application Testing), 380

WCF (Windows Communication

Foundation), 390

Web 2.0, 2, 10

Web Application Testing (Watir), 380

**Web applications, shift from desktop
 applications to, 2**

**Web Developer toolbar, adds-on for Web
 developers, 374**

Web developers, 374

Web development, challenges of, 2–3

Web forms, issues with ASP.NET, 50

Web servers

Web servers

Apache, 410–413
mapping data types between ActiveRecord
and SQL Server, 142–143
Mongrel. *See Mongrel*
Nginx, 325, 410–411
Rails deployment and, 410–411
`ruby script/server` command for
starting, 147
VPS (Virtual Private Server) for Rails
deployment, 35
WEBrick server, 3, 147

Web Service Definition Language (WSDL), 391

Web services. *See*

also ActiveRecord

benefits of working with through
ActiveResource, 383–384
consuming/publishing from .NET, 390
HTTP authentication and, 385
overview of, 383

WEBrick server

overview of, 3
starting, 147

while/until loops, 94

will_paginate plugin

overview of, 198
`per_page` method, 199–200
using, 198–199

Windows Communication Foundation (WCF), 390

Windows OSs

deploying Rails on, 411
installing Rails on, 12

Rails and, 11–12
working with Ruby files on, 57

WSDL (Web Service Definition Language), 391

X

XHTML, 357

XML

ActiveResource using, 387
Builder templates and, 357–359
communication among machines, 383
formats supported by Rails, 158
rendering in, 354–355
RSS and Atom feeds, 360

XML Builder, 41

XML-RPC, 391

XP (Extreme Programming)

agile development compared with, 45
Rails and, 5
Rails culture and, 44
testing and, 294

XSS (cross-site scripting) attacks

escaping HTML tags, 394
overview of, 394
sanitize HTML tags, 394–396
`textilize` method and, 187
vulnerabilities, 393

Y

YAGNI (You Ain't Gonna Need It), 5–6

YAML, 289

yield statement, 102

You Ain't Gonna Need It (YAGNI), 5–6