

Index

NUMERICS

- 0.3 specification, Atom, 39, 211
- 1.0 specification, RSS, 39
- 2.0 specification, RSS, 39
- 4Suite package, 20–21, 433

A

- “A Plan for Spam” (Graham), 451
 - access logs, Apache, 304–312
 - ACCESS_LOG constant, 305
 - ACCESS_RE constant, 310
 - Active Channels, Internet Explorer, 4
 - addConnection() method, 117–118
 - addTrack() method, 192
 - AddType configuration, Apache, 25
 - AdiumX IM client, 105
 - Advanced Packaging Tool, Debian Linux, 18
 - agg01_pollsubs.py program, 52–59
 - agg01_subscribe.py program, 50–51
 - agg02_pollsubs.py program, 68–77
 - agg03_emailsubs.py program, 80–86
 - agg04_emailsubs.py program, 86–91
 - agg05_im_subs.py program, 105–111
 - agg06_aggbot.py program, 114–124
 - AggBot class, 116–124
 - agglib.py module, 77–79, 112–114, 145–146
 - aggregator. *See* feed aggregator
 - AIM (AOL Instant Messenger). *See also* IM (Instant Messenger)
 - definition of, 93–94
 - handling connections from, 97–101
 - protocols for, 94
 - AIMConnection class, 97–101
 - Amazon Web Services (AWS)
 - definition of, 394
 - inserting related items into feeds, 506–512
 - product searches, building feeds from, 403–407
 - queries, building, 395–398
 - registering as AWS developer, 395, 506
 - REST-style interface for, 395
 - searches, building feeds from, 398–403
 - wish list items, building feeds from, 407–412
 - wish lists, searching for, 396–398
 - XSLT, building feeds using, 413
 - AmazonAdFeed class, 508–511
 - AMAZON_KEY constant, 508–509
 - amazon.lib.py module, 398–403
 - AmazonScraper class, 398–402
 - amazon_search() method, 510–511
 - AmazonSearchScraper class, 405–406
 - AmazonWishlistScraper class, 409–411
 - America Online Instant Messenger. *See* AIM (AOL Instant Messenger)
 - AmphetaDesk feed aggregator, 8
 - AmphetaRate service, 481
 - AOL Instant Messenger. *See* AIM
 - Apache Web server
 - compression, enabling, 231–232
 - conditional GET, enabling, 235–236
 - configuring MIME types for feeds, 25
 - monitoring access logs, 304–312
 - monitoring error logs, 301–304
 - API_BLOGID constant, 525
 - API_PASSWD constant, 525
 - API_URL constant, 525
 - API_USER constant, 525
 - append_entry() method, 297–298
 - AppleScript
 - accessing from Python, 160
 - importing MP3s into iTunes, 189–194
 - application ID, Yahoo! Web services, 384–385, 387
 - ARCHIVE_FN constant, 517
 - ASSOCIATE_TAG constant, 508–509
 - “ATOM 0.3” links, 23
 - Atom feeds. *See also* feeds
 - conversion to RSS feeds
 - commercial programs for, 444
 - definition of, 418
 - with feedparser, 437–443
 - with XSLT, building normalizer for, 420–432
 - with XSLT, data model for, 418–419
 - with XSLT, processor for, 433
 - with XSLT, testing normalizer for, 434–436
 - definition of, 3, 13–18
 - generating from HTML content, 209–217
 - 0.3 specification, 39, 211
- “Atom” links, 23
 - ATOM_DATE_FMT template, 248
 - ATOM_ENTRY_TMPL template
 - amazon.lib.py module, 399–400
 - ch07_feedmaker.py program, 557
 - ch18_mod_event_feed_filter.py program, 560
 - mailfeedlib.py module, 363–364
 - scraperlib.py module, 248
 - atomfeed feed generator, 223
 - ATOM_FEED_TMPL template, 211, 248, 560
 - AtomTemplateDocWrapper class, 215

audio feeds, creating, 158–167
 <author> tag, 17
 autodiscovery of feeds, 26–32
 AvantGo program, 167
 AWS. *See* Amazon Web Services
 AWS_ID constant, 396, 404
 AWS_INDEX constant, 404
 AWS_KEYWORDS constant, 404
 AWS_URL constant, 396, 399

B

Babu, Vattekkat Sathesh (*spycyroll* feed aggregator), 61
 “Bake, Don’t Fry” (Swartz), 227
 baked feeds, 226–229
 bandwidth, hosting feeds and
 caching feeds, 229–230
 compressing feeds, 230–233
 delivering partial feeds, 240
 how often to regenerate feeds, 226–227
 issues involving, 225–226
 metadata with update schedule hints, 237–239
 redundant downloads, reducing, 233–237
 uploading feeds with FTP, 227–229, 240
 BASE_HREF constant, 210, 250, 260–261
 Bayes class, 452, 453
 BayesFilter class, 465–467
 Bayesian classifier
 filtering feeds using, 452–459, 467–469
 suggesting feed entries using, 463–467
 training with feedback mechanism, 459–463
 Bayesian statistics, 450–451
 Beautiful Soup module, 288
 BitTorrent, 180–189
 BitTorrentDownloader class, 182–188
 blagg feed aggregator (Dornfest), 61
 blending feeds
 adding related links to feeds, 488–497
 commercial programs for, 513
 inserting related items into feeds, 506–512
 merging feeds, 483–487
 mixing daily links into feeds, 497–505
 Blogdex diffusion index, merging feeds from, 485–486
 BLOG_FN constant, 517
 blog-like aggregators, 7–8
 blogs. *See* Web logs
 Bloxom blogging package, 223
 bloxom blogging package (Dornfest), 61
 Bond, Julian (Gnews2RSS, Google News scraper), 412
 bookmarks
 in database, 292–294
 del.icio.us bookmarks manager
 API for, 497–498
 definition of, 497

 inserting bookmarks into feeds, 498–505
 merging feeds from, 485–486
 bookmark_tailgrep() function, 292–294, 302–303
 Bradbury, Nick (FeedDemon feed aggregator), 64–65
 buddy list. *See* IM (Instant Messenger)
 build() method, 532–533
 build_guid_for_message() method, 366
 buildNotesForEntries() function, 152–153
 buildNotesForFeeds() function, 150–152
 buildPluckerDocument() function, 138–139

C

Cache-Control header, 236–237
 cached_feed_reader.py module, 583–584
 CACHE_DIR constant, 577
 caching
 dynamically generated feeds, 229–230
 shared feeds, 575–584
 calendar events
 finding, 541–543, 570
 harvesting from feeds, 564–569
 parsing from HTML into feeds, 548–557
 rendering as HTML pages, 543–548
 Carlo’s Bootleg RSS Fedpalooza Web site, 287
 CDF (Channel Definition Format), 4
 cgi_buffer package (Nottingham), 232–233, 236
 ch05_ipod_notes_aggregator.py program, 146–153
 ch05_plucker_aggregator.py program, 137–140
 ch05_speech_aggregator.py program, 161–166
 ch06_enclosure_finder.py program, 171
 ch06_podcast_tuner.py program, 176–180, 188–189, 192–194
 ch07_feedmaker.py program, 209–215, 217–219
 ch08_feed_fetch.py program, 229–230
 ch08_feedfilter.py program, 232–233
 ch08_ftpupload.py program, 228–229
 ch09_fcc_scraper.py program, 270–273
 ch09_loc_scraper.py program, 259–263
 ch09_whitehouse_scraper.py program, 284–286
 ch10_apache_error_feed.py program, 301–304
 ch10_apache_referrer_feed.py program, 305–312
 ch10_bookmark_tailgrep.py program, 292–294
 ch10_logins_feed.py program, 312–317
 ch10_pytailgrep.py program, 290–291
 ch11_cvs_history_scraper.py program, 333–339
 ch11_svn_log_scraper.py program, 343–348
 ch12_imaplib_example.py program, 356–357
 ch12_mailing_list_feed.py program, 369–372
 ch12_poplib_email_example.py program, 358–359
 ch12_poplib_example.py program, 354–355
 ch13_amazon_find_wishlist.py program, 396–398
 ch13_amazon_search_scraper.py program, 403–407

- ch13_amazon_wishlist_scraper.py program, 408–411
 - ch13_google_search_scraper.py program, 378–381
 - ch13_yahoo_news_scraper.py program, 390–393
 - ch13_yahoo_search_scraper.py program, 386–389
 - ch14_feed_normalizer.py program, 437–441
 - ch14_xslt_feed_normalizer.py program, 433
 - ch14_xslt_feed_normalizer.xsl transformation, 420–432
 - ch15_bayes_agg.py program, 452–459
 - ch15_bayes_filter.py program, 463–467
 - ch15_bayes_mark_entry.cgi program, 459–463
 - ch15_feed_filter.py program, 445–449
 - ch15_popular_links.py program, 470–478
 - ch16_feed_amazon_ads.py program, 507–511
 - ch16_feed_delicious_recaps.py program, 498–504
 - ch16_feed_merger.py program, 483–486
 - ch16_feed_related.py program, 491–495
 - ch16_technorati_search.py program, 490–491
 - ch17_feed_blog.py program, 515–523
 - ch17_feed_reposter.py program, 525–528
 - ch17_feed_to_javascript.py program, 530–533
 - ch18_feed_to_ical.py program, 564–567
 - ch18_ical_to_hcal.py program, 543–547
 - ch18_mod_event_feed_filter.py program, 559–563
 - Channel Definition Format (CDF), 4
 - <channel> tag, 15, 237–238
 - chatbots
 - commercial chatbots, 126
 - managing feed subscriptions using, 114–125
 - check-out operation, CVS, 322
 - _choose() method, 186
 - CIA open source notification system, 351
 - clean_entries() method, 298–299
 - clickable feed buttons, 24–26
 - closeDBs() function, 70
 - cmd_list() method, 122
 - cmd_poll() method, 123–124
 - cmd_signoff() method, 121
 - cmd_subscribe() method, 123
 - cmd_unsubscribe() method, 122
 - __cmp__() method
 - CVSHistoryEvent class, 332
 - EntryWrapper class, 55–56
 - FeedEntryDict class, 246
 - HTMLMetaDoc class, 204
 - TemplateDocWrapper class, 213
 - CMS (content management system), 13
 - Colloquy IRC client, 340–341
 - COMMAND constant, 313
 - Command Prompt, Windows, 19
 - commit operation, CVS, 322
 - compression, for feed transfers, 230–233
 - Concurrent Version System. *See* CVS
 - conditional GET, HTTP, 234–236
 - connect() method, 98–99, 101–102, 118–119
 - content management system (CMS), 13
 - CONTENT_TMPL template, 471–472
 - Content-Type header, 233
 - conversion of feeds
 - commercial programs for, 443–444
 - definition of, 417, 418
 - with feedparser, 437–443
 - with XSLT
 - building normalizer, 420–432
 - commercial programs for, 444
 - conversion enabled by, 420
 - data model for, 418–419
 - testing normalizer, 434–436
 - XPath expressions for, 419
 - XSLT processor for, 433
 - cPickle module, 297
 - CrispAds service, 513
 - cron scheduler, 60
 - cURL tool, 489
 - CVS commit triggers, 351
 - CVS (Concurrent Version System)
 - commercial programs for, 335, 351
 - definition of, 321–322
 - history events
 - feed scraper for, 333–339
 - parsing, 327–333
 - querying, 324, 325
 - installing, 324
 - log entries
 - parsing, 327–333
 - querying, 325–327
 - scrapping, 333–338
 - repositories, finding, 322–324
 - _cvs() method, 329
 - CVSClient class, 328–330
 - CVSHistoryEvent class, 331–332
 - cvslib.py module, 327–333
 - CVSLogEntry class, 331
 - CVSScraper class, 335–337
 - Cygwin, Windows, 19
- ## D
- data model for normalization of feeds, 418–419
 - date and time formats
 - converting between, in normalization, 424, 426, 428–432
 - handling when parsing feeds, 47–48
 - RFC 822 format, 218
 - W3CDTF format, 213

- date ranges
 - Google searches using, 382–383
 - Yahoo! searches using, 389
 - DATE_HDR_TMPL template, 54, 57, 457, 520–521
 - <day> tag, 238
 - Daypop search engine, 481, 485–486
 - Debian Linux, 18. *See also* Linux
 - decode_entities() method, 46, 207–208
 - __del__() method, 116–117
 - DEL_API_URL constant, 501
 - DEL_ENTRY_TMPL template, 501
 - del.icio.us bookmarks manager
 - API for, 497–498
 - definition of, 497
 - inserting bookmarks into feeds, 498–505
 - merging feeds from, 485–486
 - DeliciousFeed class, 500–504
 - DEL_LINK_TMPL template, 501
 - DEL_PASSWD constant, 501
 - DEL_TAG_TMPL template, 501
 - DEL_USER constant, 501
 - <description> tag, 15, 17
 - Desktop component, Plucker, 132
 - difflib module, Python, 312, 315
 - digg news feed, merging feeds from, 485–486
 - directories of feeds, 32–37
 - _display() method, 186–188
 - Distiller component, Plucker
 - building feed aggregator using, 135–140
 - definition of, 132
 - installing, 132
 - using, 132–135
 - Divmod (Reverend), 451–452
 - Documentation component, Plucker, 132
 - dodgeit service, 373
 - Doppler podcast receiver, 196–197
 - Dornfest, Rael
 - blagg feed aggregator, 61
 - blosxom blogging package, 61
 - downloaders.py module, 172–175, 181–188
 - DOWNLOAD_PATH constant, 176
 - downloadURL() method, 173, 183–184
 - downloadURLs() function, 178–179
 - Drupal portal, 6
- ## E
- email
 - accessing programmatically, 353–357
 - commercial programs
 - for converting email to feeds, 373
 - for routing feeds to email, 91–92
 - comparing RSS feeds to, 15–16
 - containing individual new or modified feeds, 86–92
 - containing report of new or modified feeds, 80–86
 - filtering to produce feeds from, 369–372
 - format of, 357–359
 - producing feeds from
 - building mail protocol wrappers for, 360–363
 - filtering email for, 369–372
 - generating feed entries for, 363–369
 - email package, Python, 358
 - <email> tag, 17
 - emailAggregatorPage() function, 82
 - emailEntries() function, 87–89
 - email.MIMEMultipart module, Python, 81, 82
 - email.MIMEText module, Python, 81, 82
 - <enclosure> tag, 169–170
 - enclosures, RSS
 - commercial programs for, 194–197
 - downloading content for, 171–175
 - downloading content with feed aggregator, 176–180
 - finding multimedia content using, 169–171
 - end_entry() method, 43
 - end_feed() method, 258
 - end_feed_entry() method, 258
 - end_item() method, 43
 - entries_from_messages() method, 365–366
 - ENTRIES_XPATH constant, 282–283, 284
 - ENTRY_DB_FN constant, 69, 516, 525
 - ENTRY_LINK_TMPL template, 148
 - ENTRY_RE constant, 268, 271
 - ENTRY_TMPL template
 - agg01_pollsubs.py program, 54, 57
 - agg05_im_subs.py program, 110
 - ch05_ipod_notes_aggregator.py program, 148
 - ch15_bayes_agg.py program, 457–458
 - ch17_feed_blog.py program, 521
 - ch17_feed_to_javascript.py program, 531–532
 - EntryWrapper class, 55–57, 73
 - ENTRY_XPATHS constant, 282–283, 284
 - error logs, Apache, 301–304
 - _error() method, 186
 - ERROR_LOG constant, 301
 - ETag header, 234, 236
 - <ev> tag, 538–539
 - EVDB (Events and Venues Database), 570
 - Event Log, monitoring, 290, 318
 - event_filter() function, 309
 - EventMeister package, 318
 - Events and Venues Database (EVDB), 570
 - EXCLUDE_EXACT constant, 306, 309
 - EXCLUDE_PARTIAL constant, 306, 309
 - _executeAppleScript() method, 192
 - Expires header, 236–237

- extending feeds
 - adding metadata to feeds, 538–539, 541, 557–564
 - with calendar events, 541–557
 - commercial programs for, 569–570
 - definition of, 537
 - harvesting calendar events from feeds, 564–569
 - structuring feeds with microformats, 539–541, 543–548
 - Extensible Markup Language (XML)
 - definition of, 13–14
 - tools for, 20–21
 - `extract_summary_from_message()` method, 367–368
 - Extras component, Plucker, 132
- ## F
- FCCScraper class, 270–272
 - Federal Communications Commission, example using, 264–274
 - feed aggregator
 - building
 - using iPod Notes, 144–153
 - using Plucker Distiller, 135–140
 - using Python, 52–59
 - commercial programs for, 61–65, 167
 - creating group Web log from, 515–524
 - definition of, 4–13
 - downloading multimedia content using, 176–180
 - emailing individual new or modified feeds, 86–92
 - emailing report of new or modified feeds, 80–86
 - finding feeds for
 - clickable feed buttons, 24–26
 - feed autodiscovery, 26–32
 - feed directories, 32–37
 - links on Web sites and blogs for, 23–24
 - Web Services, 32–37
 - future of, 10–13
 - generating audio files from, 160–167
 - mixed reverse-chronological aggregators, 7–8
 - module for reusable parts of, 77–79
 - parsing feeds for
 - building feed parser, 38–46
 - extending feed parser, 47–48
 - testing feed parser, 46–47
 - Universal Feed Parser (Pilgrim), 48–49
 - personalized portals, 5–7
 - retrieving only new or modified feeds, 67–77
 - scheduling, 60–61
 - subscribing to feeds for, 49–51
 - three-pane aggregators, 8–13
 - Web-based aggregators, 10
 - feed cache, for shared feeds, 575–584
 - feed handler
 - building, 37–48
 - Universal Feed Parser (Pilgrim), 48–49
 - Feed on Feeds feed aggregator (Minutillo), 61–62
 - feed playlist format, 25
 - feed: protocol scheme, 26
 - feed reader, 4–5. *See also* feed aggregator
 - `<feed>` tag, 17, 42
 - Feed Validator, 220–222
 - `FeedAutodiscoveryParser` class, 28–29
 - FeedBurner service, 443, 513
 - `FeedCache` class, 577–582
 - `feedcache.py` module, 575–584
 - `FeedCacheRecord` class, 576–577
 - `FEED_DB_FN` constant, 69, 516, 525
 - FeedDemon feed aggregator (Bradbury), 64–65
 - `FEED_DIR` constant, 301, 305, 313, 470
 - `FeedEntryDict` class, 245–247
 - `feed_fetch.py` module, 38
 - `FeedFilter` class, 447–449
 - `FEED_HDR_TMPL` template
 - `agg01_pollsubs.py` program, 54, 57
 - `agg05_im_subs.py` program, 110
 - `ch15_bayes_agg.py` program, 457
 - `ch17_feed_blog.py` program, 520–521
 - `FEED_IN_URL` constant, 559
 - `FEED_LINK_TMPL` template, 148, 149
 - `FeedMerger` class, 485–486
 - `FEED_META` constant
 - `AmazonWishlistScraper` class, 409
 - `ch07_feedmaker.py` program, 209–210, 212
 - `FCCScraper` class, 271
 - `LOCScraper` class, 260–261
 - `Scraper` class, 249–250, 253
 - `YahooSearchScraper` class, 387–388
 - `FEED_NAME_FN` constant
 - `ch10_apache_error_feed.py` program, 301
 - `ch10_apache_referrer_feed.py` program, 305
 - `ch10_logins_feed.py` program, 313
 - `ch15_feed_filter.py` program, 446
 - `ch15_popular_links.py` program, 470
 - `FeedNormalizer` class, 438–441
 - `FEED_OUT_FN` constant, 559
 - Feedpalooza Web site, 287
 - `feed_reader.py` program, 46–47
 - `FeedRelator` class, 492–495
 - feeds
 - aggregator for
 - building, 52–59, 135–140, 144–153
 - commercial programs for, 61–65, 167
 - creating group Web log from, 515–524
 - definition of, 4–13

feeds

- aggregator for (*continued*)
 - downloading multimedia content using, 176–180
 - emailing individual new or modified feeds, 86–92
 - emailing report of new or modified feeds, 80–86
 - finding feeds for, 24–37
 - future of, 10–13
 - generating audio files from, 160–167
 - mixed reverse-chronological aggregators, 7–8
 - module for reusable parts of, 77–79
 - parsing feeds for, 38–49
 - personalized portals, 5–7
 - retrieving only new or modified feeds, 67–77
 - scheduling, 60–61
 - subscribing to feeds for, 49–51
 - three-pane aggregators, 8–13
 - Web-based aggregators, 10
- audio, creating, 158–167
- baking, 226–229
- blending
 - adding related links to feeds, 488–497
 - commercial programs for, 513
 - inserting related items into feeds, 506–512
 - merging feeds, 483–487
 - mixing daily links into feeds, 497–505
- caching, 229–230, 575–584
- compressing, 230–233
- conversion of
 - commercial programs for, 443–444
 - definition of, 418
 - with feedparser, 437–443
 - with XSLT, building normalizer for, 420–432
 - with XSLT, data model for, 418–419
 - with XSLT, processor for, 433
 - with XSLT, testing normalizer for, 434–436
- definition of, 3–4, 13–18
- extending
 - adding metadata to feeds, 538–539, 541, 557–564
 - with calendar events, 541–557
 - commercial programs for, 569–570
 - definition of, 537
 - harvesting calendar events from feeds, 564–569
 - structuring feeds with microformats, 539–541, 543–548
- extracting calendar events from, 564–569
- filtering
 - commercial programs for, 481
 - definition of, 13
 - by keywords and metadata, 445–450
 - using Bayesian classifier, 452–469
- finding
 - clickable feed buttons, 24–26
 - feed autodiscovery, 26–32
 - feed directories, 32–37
 - links on Web sites and blogs for, 23–24
 - MIME types for, 25
 - URI scheme for, 26
 - Web Services for, 32–37
- fried, 226
- hosting
 - caching feeds, 229–230
 - compression used for, 230–233
 - delivering partial feeds, 240
 - how often to regenerate feeds, 226–227
 - issues involving, 225–226
 - metadata with update schedule hints, 237–239
 - redundant downloads, reducing, 233–237
 - scheduling feed generation, 227
 - testing, 240
 - uploading with FTP, 227–229, 240
- inserting items into
 - Amazon items, 506–512
 - commercial programs for, 513
 - daily links, 497–505
 - related links, 488–497
- links
 - daily, mixing into feeds, 497–505
 - related, adding to feeds, 488–497
 - sifting from feeds, 469–480
- merging, 483–487
- metadata with update schedule hints, 237–239
- monitoring login activity using, 312–317
- monitoring logs using, 290–294, 301–312
- monitoring projects in CVS repositories
 - history events, 324, 325, 333–339
 - log entries, 325–338
- monitoring projects in Subversion repositories, 340–350
- multimedia content in
 - commercial programs for, 194–197
 - downloading from URLs, 171–175
 - downloading using BitTorrent, 180–189
 - downloading with feed aggregator, 176–180
 - finding, using RSS enclosures, 169–171
 - importing MP3s into iTunes, 189–194
- normalization of
 - commercial programs for, 443–444
 - definition of, 417, 483
 - with feedparser, 437–443
 - with XSLT, building normalizer for, 420–432
 - with XSLT, data model for, 418–419
 - with XSLT, processor for, 433
 - with XSLT, testing normalizer for, 434–436
- parsing
 - building feed parser, 38–46
 - conversion of feeds using, 437–443
 - date and time formats, handling, 47–48
 - definition of, 483

- extending feed parser, 47–48
 - for feed aggregator, 38–49
 - history events in CVS, 327–333
 - HTML files into feeds, 548–557
 - HTMLParser class, Python, 27–30, 253–263
 - log entries in CVS, 327–333
 - testing feed parser, 46–47
 - Universal Feed Parser (Pilgrim), 48–49, 171, 419, 437–443
 - producing
 - with Amazon Web Services, 394–412
 - from calendar events in HTML, 548–557
 - commercial programs for, 223–224
 - from email, filtering feeds for, 369–373
 - from email, generating feeds for, 363–369
 - from email, mail protocol wrappers for, 360–363
 - from filtered email, 369–372
 - with Google Web services, 375–383
 - from HTML files, 209–215, 217–219
 - incrementally generating from logs, 294–300
 - with Yahoo! Web services, 383–394
 - publishing, 13
 - reading from PalmOS devices
 - commercial programs for, 167
 - feed aggregator for, 135–140
 - loading documents to, 141
 - Plucker package for, 130–135
 - receiving in Usenet newsgroups, 92
 - remixing, 417
 - republishing
 - commercial programs for, 535–536
 - as group web log, 515–524
 - in JavaScript includes, 529–535
 - as web log with MetaWeblog, 524–529
 - retrieving, 37–38
 - retrieving from IM chatbot, 114–125
 - retrieving only new or modified feeds
 - aggregator memory for, 67–77
 - emailing reports of, 80–91
 - retrieving using iPod Notes
 - building feed aggregator for, 144–153
 - using feed aggregator for, 153–157
 - scraping from CVS entries, 333–339
 - scraping from Subversion entries, 343–350
 - scraping Web sites for
 - building feed scraper, 244–253
 - commercial programs for, 287–288
 - definition of, 13, 243–244
 - with HTML Tidy, 274–278
 - with HTMLParser class, 253–263
 - problems with, 244
 - with regular expressions, 264–274
 - tools for, 244
 - with XPath, 274–275, 278–286
 - sending as Instant Messages, 105–111
 - subscriptions
 - managing with IM chatbot, 114–125
 - subscribing to, 49–51
 - update schedule hints in, 237–239
 - validating, 220–222
 - FEEDS_FN constant, 52, 471, 516, 525
 - Feedsplitter feed conversion, 444
 - FEED_TAGLINE constant, 470
 - FEED_TITLE constant, 470
 - FEED_TYPES constant, 29, 30, 33
 - FEED_URI constant, 446
 - FEED_URL constant, 492, 499–500, 531, 564
 - fetch_items() method
 - AmazonSearchScraper class, 406
 - AmazonWishlistScraper class, 410–411
 - fetch_messages() method
 - IMAP4Client class, 362–363
 - POP3Client class, 360–361
 - filter_aggregator_entries() method, 466–467
 - filtering email to produce feeds, 369–372
 - filtering feeds
 - commercial programs for, 481
 - definition of, 13
 - by keywords and metadata, 445–450
 - using Bayesian classifier
 - classifier for, 463–467
 - feed aggregator for, 452–459
 - feedback mechanism for, 459–463
 - performing, 467–469
 - filter_messages() method, 364–365, 370–372
 - FILTER_RE constant, 446
 - _fin() method, 185–186
 - findEntry() function, 456
 - findHTML() function, 202–203, 212
 - finish() method, 552
 - FooListScraper class, 370
 - 4Suite package, 20–21, 433
 - fried feeds, 226
 - FTP, baking feeds using, 227–229, 240
 - ftplib module, Python, 227–228
 - fuzzy logic, used in scraping, 244
- ## G
- GET, conditional, 234–236
 - _getCachedURIs() method, 581
 - getEnclosuresFromEntries() function, 171, 177–178
 - get_entry_paths() method, 299
 - getFeedEntries() function, 53
 - GetFeedFields() method, Syndic8, 34
 - GetFeedInfo() method, Syndic8, 34
 - getFeedRecord() method, 578
 - getFeeds() function, 30

- getFeedsDetail() function, 30–31, 34
 - getFeedStates() method, Syndic8, 34
 - __getitem__() method
 - CVSHistoryEvent class, 332
 - CVSLogEntry class, 331
 - EntryWrapper class, 56
 - FeedEntryDict class, 246–247
 - HTMLMetaDoc class, 203–204
 - ICalTmpWrapper class, 546
 - ScoredEntryWrapper class, 454
 - TemplateDocWrapper class, 214–215
 - TrampTmpWrapper class, 402–403
 - getNewFeedEntries() function
 - agg02_pollsubs.py program, 70–74
 - agglib.py module, 145–146, 177, 453
 - getOwner() method, 117–118
 - getSubscriptionURI() method, 119
 - _getTorrentMeta() method, 184–185
 - Gmail service, 373
 - Gnews2RSS, Google News scraper (Bond), 412
 - go() method, 118–119
 - Google News scrapers, 412
 - Google Web services
 - definition of, 375–376
 - license key for, 376–377, 379
 - searches, building feeds from, 378–383
 - SOAP interface for, accessing, 378
 - GOOGLE_LICENSE_KEY constant, 379
 - GOOGLE_SEARCH_QUERY constant, 379
 - GoogleSearchScraper class, 379–381
 - Graham, Paul (“A Plan for Spam”), 451
 - Gregorio, Joe (httpcache module), 37–38
 - grep command, log monitoring and, 290–294
 - Growl global notification system, 350
 - guessEntry() function, 455–456
 - <guid> tag, 15, 75
 - gzip module, Python, 316
- ## H
- Hammersley, Ben (tracking installed Perl modules), 317
 - handheld devices
 - iPod devices
 - commercial programs for, 167
 - creating and managing Notes on, 142–145
 - feed aggregator for, building, 144–153
 - feed aggregator for, using, 153–157
 - Note Reader for, 141–142
 - PalmOS devices
 - commercial programs for, 167
 - definition of, 129
 - feed aggregator for, 135–140
 - loading documents to, 141
 - Plucker package for, 130–135
 - handle_comment() method, 261
 - handle_data() method, 207–208
 - handle_endtag() method, 206–207, 262–263, 553–555
 - handle_entityref() method, 207–208
 - handle_starttag() method, 206, 261–262, 552
 - hash() method, 73, 74–76
 - hash_entry() method, 299
 - hCalendar microformat, 539–541, 548–557. *See also*
 - calendar events
 - HCalendarParser class, 549–556
 - hcalendar.py module, 548–556
 - <head> tag, feeds as metadata in, 26–27
 - HEVENT_TMPL template, 544, 546–547
 - history() method, 330
 - history operation, CVS
 - definition of, 324, 325
 - feed scraper for, 333–339
 - parsing results of, 327–333
 - HISTORY_FN constant, 516
 - hosting feeds
 - caching feeds, 229–230
 - compression used for, 230–233
 - delivering partial feeds, 240
 - how often to regenerate feeds, 226–227
 - issues involving, 225–226
 - metadata with update schedule hints, 237–239
 - redundant downloads, reducing, 233–237
 - scheduling feed generation, 227
 - testing, 240
 - uploading with FTP, 227–229, 240
 - HotLinks link aggregator, merging feeds from, 485–486
 - <hour> tag, 237–238
 - .htaccess file, Apache, 25
 - HTML files
 - examining to produce feed scraper for, 255–256, 264–267, 274–275
 - extracting metadata from, 201–209
 - parsing into feeds, 548–557
 - producing feeds from, 209–215, 217–219
 - rendering calendar events as, 543–548
 - HTML Tidy, scraping Web sites using, 274–278
 - HTML_DIR constant, 544
 - HTML_FN constant, 52, 134
 - HTMLMetaDoc class, 203–204
 - htmlmetalist.py module, 201–209
 - HTMLMetaParser class, 204–207
 - _HTMLMetaparserFinished class, 204–205
 - HTMLParser class, Python, 27–30, 253–263
 - HTMLScraper class, 257–259
 - HTMLStripper class, 163, 165–166
 - HTTP 1.1
 - conditional GET facilities, 234–236
 - content coding allowing transformation, 230
 - HTTP status codes, 72

- HTTPCache class, 395–396, 398, 490–491
 - httpcache module (Gregorio), 37–38, 229–230
 - HTTPDownloader class, 172–175
- I**
- iCalendar standard, 539, 541. *See also* calendar events
 - ICalTmplWrapper class, 546–547
 - ICAL_URL constant, 544
 - ICS_FN constant, 564
 - ID for Yahoo! Web services, 384–385
 - <id> tag, 17, 75
 - If-Modified-Since header, 234, 236
 - If-None-Match header, 234, 236
 - Iframe-style includes, republishing feeds using, 536
 - IM (Instant Messenger)
 - AIM (AOL Instant Messenger)
 - definition of, 93–94
 - handling connections from, 97–101
 - protocols for, 94
 - commercial programs for, 126
 - Jabber instant messaging
 - commercial chatbots for, 126
 - definition of, 94–95
 - handling connections from, 101–104
 - managing feed subscriptions using, 114–125
 - sending new feeds as Instant Messages, 105–111
 - test function for, 95–97
 - IMAP4 mailbox, accessing programmatically, 355–357
 - IMAP4Client class, 362–363
 - imaplib module, Python, 356
 - IM_CHUNK constant, 106
 - IM_CLASS constant, 106
 - imconn.py module, 95–105
 - IM_PASSWD constant, 106
 - impedence mismatch, 225
 - importAudioFiles() function, 193–194
 - importSoundFile() function, 163–165
 - IM_TO constant, 106
 - IM_USER constant, 106
 - inbox, accessing programmatically, 353–357
 - includes, JavaScript, 529–535, 536
 - INCLUDE_TMPL template, 531–532
 - “Industrial Strength Web Publishing” (Kallen), 227
 - __init__() method
 - AggBot class, 116–117
 - AmazonAdFeed class, 508–509
 - AmazonSearchScraper class, 405
 - AmazonWishlistScraper class, 409–410
 - BayesFilter class, 465–466
 - CVSClient class, 329
 - CVSHistoryEvent class, 332
 - CVSLogEntry class, 331
 - CVSScraper class, 335–336
 - DeliciousFeed class, 501
 - FeedCache class, 577–578
 - FeedEntryDict class, 246
 - FeedFilter class, 447
 - FeedNormalizer class, 438
 - FeedRelator class, 493
 - GoogleSearchScraper class, 380
 - HCalendarParser class, 549–550
 - HTMLMetaDoc class, 203–204
 - ICalTmplWrapper class, 546
 - IMAP4Client class, 362
 - JavaScriptFeed class, 531–532
 - LogBufferFeed class, 296–297
 - MailScraper class, 363–364
 - ModEventFeed class, 562
 - POP3Client class, 360–361
 - ScoredEntryWrapper class, 454
 - SVNScraper class, 345
 - TemplateDocWrapper class, 213
 - TrampTmplWrapper class, 402
 - YahooNewsScraper class, 392
 - YahooSearchScraper class, 387–388
 - INSERT_ITEM_TMPL template, 493, 508–509
 - INSERT_TMPL template, 493, 508–509
 - Instant Messenger (IM). *See* IM
 - Internet Engineering Task Force, 94
 - Internet Mail Access Protocol version 4 mailbox,
 - accessing programmatically, 355–357
 - Internet Message Format (RFC 2822), 357–359
 - iPod Agent program, 167
 - iPod devices
 - commercial programs for, 167
 - creating and managing Notes on, 142–145
 - feed aggregator for
 - building, 144–153
 - using, 153–157
 - Note Reader for, 141–142
 - iPodder podcast receiver, 194–195, 322–324
 - iPodderX podcast receiver, 196
 - IPOD_FEEDS_DIR constant, 146–147
 - IPOD_FEEDS_IDX constant, 146–147
 - IPOD_FEEDS_IDX_TITLE constant, 146–147
 - IPOD_NOTES_PATH constant, 146–147
 - <issued> tag, 17
 - isValidAudioFile() function, 193–194
 - <item> tag, 15
 - items() method, 551
 - ITEM_TRACK constant, 399
 - iTunes
 - accessing from Python, 160
 - importing MP3s into, 189–194
 - iTunesMac class, 190–192

J

- Jabber instant messaging. *See also* IM (Instant Messenger)
 - commercial chatbots for, 126
 - definition of, 94–95
 - handling connections from, 101–104
- JabberConnection class, 101–104
- JabRSS chatbot, 126
- JavaScript includes, republishing feeds using, 529–535, 536
- JavaScriptFeed class, 531–533
- JS_FN constant, 531
- js_format() method, 533
- Julian date ranges, Google searches using, 382–383
- julian.py module, 382–383

K

- Kallen, Ian (“Industrial Strength Web Publishing”), 227
- keywords, filtering feeds by, 445–450
- Kibo, 310–312
- KNewsTicker (K Desktop Environment), 5

L

- last command, 313
- Last-Modified header, 234, 235
- learning machines, 4
- license key, Google Web services, 376–377, 379
- <link> tag, 15, 17, 26–27
- LINKER_TMPL, 471–472
- links
 - daily, mixing into feeds, 497–505
 - indicating feeds, 23–24
 - related, adding to feeds, 488–497
 - sifting from feeds
 - building feed generator for, 469–478
 - using feed generator for, 478–480
- LinkSkimmer class, 478
- LINK_TMPL template, 306–307, 335, 471–472
- Linux
 - installing CVS on, 324
 - installing 4Suite on, 21
 - installing HTML Tidy, 276
 - installing Python on, 19
 - installing Subversion on, 340
 - installing XPath on, 279
 - loading Plucker documents to PalmOS, 141
 - login activity, monitoring, 312–317
 - UNIX-based tools for, 18–19
- _loadRecord() method, 581–582
- loadSubs() function, 112
- localhost, SMTP server set to, 81
- LOCScraper class, 260–263
- log operation, Subversion, 342–343
- LogBufferFeed class, 296–299, 302

- LOG_CONF constant, 471
- login activity, monitoring, 312–317
- LogMeister package, 318
- LOG_PERIOD constant, 345
- logs
 - CVS
 - parsing, 327–333
 - querying, 325–327
 - scraping, 333–338
 - monitoring
 - Apache access logs, 304–312
 - Apache error logs, 301–304
 - commercial programs for, 317–318
 - incremental feeds for, 294–300
 - login activity, 312–317
 - UNIX system logs, 290–294
 - Windows Event Log, 290
 - Subversion
 - querying, 341–343
 - scraping, 343–350

M

- M3U format, 25
- Mac OS X
 - creating audio feeds with Text-to-Speech, 158–167
 - importing MP3s into iTunes, 189–194
 - installing CVS on, 324
 - installing 4Suite on, 21
 - installing HTML Tidy, 276
 - installing Python on, 20
 - installing Subversion on, 340
 - installing XPath on, 279
 - loading Plucker documents to PalmOS, 141
 - NetNewsWire feed aggregator, 63–64
 - Radio UserLand feed aggregator, 62–63
 - UNIX-based tools for, 19
- machine learning, 450
- MaggieRSS feed parser, 61
- mailbox, accessing programmatically, 353–357
- MailBucket service, 373
- MAIL_CLIENT constant, 369
- mailfeedlib.py module, 359–369
- Mailscraper class, 363–368
- makeEntryID() function, 456–457
- Manalang, Rich (XSLT feed conversion), 444
- MAX_ENTRIES constant
 - ch07_feedmaker.py program, 209–210, 212
 - ch17_feed_blog.py program, 516–517
 - scraperlib.py module, 250, 253
- MAX_ENTRY_AGE constant, 471
- MAX_ITEMS constant, 508–509
- md5 module, Python, 68, 75–76
- md5_hash() function, 150

- merging feeds, 483–487
 - `_messageCB()` method, 103–104
 - `<meta>` tag, 208
 - metadata
 - adding to feeds, 538–539, 541, 557–564
 - extracting from HTML files, 201–209
 - filtering feeds by, 445–450
 - update schedule hints in, 237–239
 - MetaWeblog API, reposting feeds using, 524–529
 - microcontent, 4
 - microformats
 - specifications for, 570
 - structuring feeds with, 539–541, 543–548
 - MIME (Multipurpose Internet Mail Extensions), 25, 357–359
 - MIMEMultipart module, Python, 81, 82
 - MIMEText module, Python, 81, 82
 - `minifeedfinder.py` module, 27–32
 - MiniFeedParser class, 40
 - `minifeedparser.py` module, 39–48
 - MIN_LINKS constant, 471
 - Minutillo, Steve (Feed on Feeds feed aggregator), 61–62
 - mixed reverse-chronological aggregators, 7–8
 - `mod_deflate` module, Apache, 232
 - `mod_event` module, 538–539
 - ModEventFeed class, 560–563
 - `mod_expires` module, Apache, 235–236
 - `mod_gzip` module, Apache, 231
 - `<modified>` tag, 17
 - `monitorfeedlib.py` module, 294–300
 - monitoring
 - Apache access logs, 304–312
 - Apache error logs, 301–304
 - commercial programs for, 317–318
 - incremental feeds for, 294–300
 - login activity, 312–317
 - projects in CVS repositories
 - history events, 324, 325, 333–339
 - log entries, 325–338
 - projects in Subversion repositories, 340–350
 - UNIX system logs, 290–294
 - Windows Event Log, 290, 318
 - Movable Type content management package, 226, 524
 - MP3 files, importing into iTunes, 189–194
 - MP3 links, 25
 - MP3PLAYER constant, 192
 - `mp3players.py` module, 189–192
 - multimedia content
 - commercial programs for, 194–197
 - downloading from URLs, 171–175
 - downloading using BitTorrent, 180–189
 - downloading with feed aggregator, 176–180
 - finding, using RSS enclosures, 169–171
 - importing MP3s into iTunes, 189–194
 - Multipurpose Internet Mail Extensions (MIME), 25, 357–359
 - My Netscape portal, 6
- ## N
- `<name>` tag, 17
 - named groups, 267
 - `ndiff()` function, 315
 - NetNewsWire feed aggregator (Ranchero Software), 63–64
 - NEW_FEED_LIMIT constant, 176, 177–178
 - news searches, Yahoo!, 390–394
 - newsmaster, 4
 - Newspipe program, 91–92
 - `nntp/rss` aggregator, 92
 - normalization of feeds
 - commercial programs for, 443–444
 - definition of, 417, 483
 - with feedparser, 437–443
 - with XSLT
 - building normalizer, 420–432
 - conversion enabled by, 420
 - data model for, 418–419
 - testing normalizer, 434–436
 - XPath expressions for, 419
 - XSLT processor for, 433
 - `normalize_entries()` method, 440–441, 485–486
 - `normalize_feed_meta()` method, 439–440
 - “Normalizing Syndicated Feed Content” (Pilgrim), 419
 - Note Reader for iPod
 - creating and managing Notes, 142–145
 - definition of, 141–142
 - feed aggregator for
 - building, 144–153
 - using, 153–157
 - NOTE_TEMPL template, 148
 - Nottingham, Mark (`cgi_buffer` package), 232–233, 236
 - NUM_DAYS constant, 501
- ## O
- `on_IM_IN()` method, 100–101
 - Open Source projects, tracking
 - CVS (Concurrent Version System)
 - commercial programs for, 335, 351
 - definition of, 321–322
 - history events, feed scraper for, 333–339
 - history events, parsing, 327–333
 - history events, querying, 324, 325
 - installing, 324
 - log entries, feed scraper for, 333–338
 - log entries, parsing, 327–333
 - log entries, querying, 325–327
 - repositories, finding, 322–324

Open Source projects, tracking (*continued*)

- Subversion repositories
 - commercial programs for, 335, 351
 - definition of, 340
 - finding, 340–341
 - installing, 340
 - logs, querying, 341–343
 - logs, scraping, 343–350

Open Source Web community, 6

`openDBs()` function, 70

`osascript` command line tool, 160, 165

OSCAR (Open System for Communication in Real Time), 94

`os.popen` function, Python, 160

`os.walk()` function, Python, 203

P

`PAGE_TMPL` template, 54, 520–521

PalmOS devices

- commercial programs for, 167
- definition of, 129
- feed aggregator for, 135–140
- loading documents to, 141
- Plucker package for, 130–135

`parse()` function, 39–40, 550

`parse()` method, 578, 582–583

`parse_file()` method, 205, 259

`parse_results()` method, 389

`parse_uri()` method, 550

parsing

- conversion of feeds using, 437–443
- date and time formats, handling, 47–48
- definition of, 483
- for feed aggregator
 - building feed parser, 38–46
 - extending feed parser, 47–48
 - testing feed parser, 46–47
 - Universal Feed Parser (Pilgrim), 48–49
- history events in CVS, 327–333
- HTML files into feeds, 548–557
- `HTMLParser` class, Python, 27–30, 253–263
- log entries in CVS, 327–333
- Universal Feed Parser (Pilgrim)
 - definition of, 48
 - downloading, 48–49
 - enclosures and, 171
 - normalizing and converting feeds with, 419, 437–443
 - using, 49

pattern recognition, used in scraping, 244

`PERC_STEP` constant, 182

performance. *See* bandwidth, hosting feeds and

Perl modules, installed, tracking (Hammersley), 317

personal aggregators, 7–8

personalized portals, 5–7

PHP iCalendar, 541

PHP includes, republishing feeds using, 536

`Pickle` module, 297

Pilgrim, Mark

“Normalizing Syndicated Feed Content”, 419

Ultra-Liberal Feed Finder module, 36–37

Universal Feed Parser, 48–49, 419

Planet feed aggregator, 535

playlist files, 25

Plone portal, 6

PLS format, 25

Plucker package for PalmOS

components in

downloading, 131–132

list of, 131–132

definition of, 130–131

Desktop component, 132

Distiller component

building feed aggregator using, 135–140

installing, 132

using, 132–135

Documentation component, 132

Extras component, 132

`PLUCKER_BPP` constant, 134–135

`PLUCKER_DEPTH` constant, 134–135

`PLUCKER_DIR` constant, 134

`PLUCKER_FN` constant, 134–135

`PLUCKER_TITLE` constant, 134

podcast feeds, listings of, 179

podcasting, 169. *See also* multimedia content

PointCast, Inc., 5

`pollFeed()` method, 119–120

polling, 225

POP3 mailbox, accessing programmatically, 353–355

POP3 standard, 354

`POP3Client` class, 360–361

`popen2` module, Python, 312

`popen4()` method, 314

`poplib` module, Python, 354

portals, personalized, 5–7

Post Office Protocol version 3 mailbox, accessing programmatically, 353–355

`produce_entries()` method

`AmazonAdFeed` class, 509–510

`AmazonScraper` class, 400–402

`BayesFilter` class, 466

`CVSScraper` class, 336

`DeliciousFeed` class, 501–504

`FCCScraper` class, 272

`FeedFilter` class, 448–449

`FeedMerger` class, 485–486

`FeedNormalizer` class, 438–439

`FeedRelator` class, 493–495

- GoogleSearchScraper class, 380–381
 - HTMLScraper class, 257
 - LogBufferFeed class, 296–297
 - MailScraper class, 364–365
 - ModEventFeed class, 562
 - RegexScraper class, 269
 - SVNScraper class, 346–347
 - XPathScraper class, 283–284
 - YahooNewsScraper class, 392–393
 - YahooSearchScraper class, 388
 - PROG_OUT constant, 182
 - <pubDate> tag, 15
 - pull technology, 225
 - push technology, 225
 - PyBlosxom blogging package, 223
 - pygoogle package, 378
 - PyPlucker.Spider module, 137
 - PyRSS2Gen feed generator, 223
 - Python
 - feed aggregator using, 52–59
 - feed autodiscovery using, 27–32
 - fetching feeds using, 37
 - installing on Linux, 19
 - installing on Mac OS X, 20
 - installing on Windows, 20
 - spycyroll feed aggregator (Babu), 61
 - Web Services access using, 33–37
 - Py-TOC module, Python (Turner), 94
- Q**
- QueryFeeds() method, Syndic8, 34
 - QuickNews feed aggregator, 167
- R**
- Radio UserLand feed aggregator, 7–8, 62–63
 - Ranchero Software, NetNewsWire feed aggregator, 63–64
 - <rdf> tag, 42
 - Really Simple Syndication. *See* RSS feeds
 - reBlog package, 536
 - receiveIM() method, 120–121
 - _recordFN() method, 581
 - referrer spam, 306
 - referrers, monitoring, 304–312
 - REFER_SEEN constant, 305
 - refreshFeed() method, 579–581
 - refreshFeeds() method, 578–579
 - REFRESH_PERIOD constant, 577
 - RegexScraper class, 267–269, 271
 - regular expressions
 - for Apache access log, 310
 - definition of, 266
 - feed scraper using, 264–274
 - for feeds with category of “python”, 446
 - resources for learning about, 266, 267, 269
 - Reinacker, Greg (Windows Event Log, monitoring), 318
 - remixing feeds, 417. *See also* conversion of feeds; normalization of feeds
 - repositories, CVS, finding, 322–324
 - reset() method, 41, 204–205, 552
 - reset_doc() method, 205
 - resources. *See* Web site resources
 - REST style of Web services
 - for Amazon, 395
 - for Yahoo!, 384
 - Reverend (Divmod), 451–452
 - revision control systems. *See* CVS (Concurrent Version System)
 - RFC 822 format, 218, 424, 426, 428–432
 - RFC 2822 format, 357–359
 - RFC3229 format, 240
 - rlog() method, 329
 - rlog operation, CVS
 - definition of, 325–327
 - feed scraper for, 333–338
 - parsing results of, 327–333
 - “RSS 2.0” link, 23
 - RSS Digest service, 536
 - RSS enclosures
 - commercial programs for, 194–197
 - downloading content for, 171–175
 - downloading content with feed aggregator, 176–180
 - finding multimedia content using, 169–171
 - RSS feeds. *See also* feeds
 - conversion to Atom feeds
 - commercial programs for, 444
 - definition of, 418
 - with feedparser, 437–443
 - with XSLT, building normalizer for, 420–432
 - with XSLT, data model for, 418–419
 - with XSLT, processor for, 433
 - with XSLT, testing normalizer for, 434–436
 - definition of, 3, 13–18
 - generating from HTML content, 217–220
 - mod_event extension to, 538–539
 - 1.0 specification, 39
 - 2.0 specification, 39
 - update schedule hints in, 237–239
 - “RSS” links, 3, 23
 - <rss> tag, 15, 42
 - rss2email program (Swartz), 91
 - rss2jabber chatbot, 126
 - RSSCalendar calendar events, 570
 - RSS_DATE_FMT template, 248–249
 - RSS_ENTRY_TMPL template, 249, 558, 561
 - RSS_FEED_TMPL template, 248–249, 561
 - RSS-IM Gateway IM chatbot, 126
 - RSSTemplateDocWrapper class, 218–219
 - runOnce() method, 99–100, 103–104, 118–119

S

- `_saveRecord()` method, 582
- `saveSubs()` function, 112
- say command, Mac OS X, 158–160
- scheduled tasks, Windows XP, 60–61
- `ScoredEntryWrapper` class, 453–454
- `scoreEntries()` function, 454
- `scoreEntry()` function, 454–455
- `scrape()` method, 251–253
- `scrape_atom()` method, 250–251
- Scraper class, 249–253
- `_ScraperFinishedException()` method, 247
- `scraperlib.py` module, 245–253, 257–259, 268–269, 282–284
- `scrape_rss()` method, 250–251
- SCRAPE_URL constant, 250, 260–261
- scraping CVS entries, 333–339
- scraping Subversion entries, 343–350
- scraping Web sites
 - building feed scraper, 244–253
 - commercial programs for, 287–288
 - definition of, 13, 243–244
 - with HTML Tidy, 274–278
 - with `HTMLParser` class, 253–263
 - problems with, 244
 - with regular expressions, 264–274
 - tools for, 244
 - with XPath, 274–275, 278–286
- ScrappyGoo, Google News scraper (Yang), 412
- `<script>` tag, 530
- searches
 - Amazon, building feeds with, 398–407
 - Google, building feeds from, 375–383
 - Technorati, adding to feeds, 489–495
 - Yahoo!, building feeds with, 383–394
- SEARCH_URL_TMPL template, 493
- Secure FTP (SFTP), 240
- security, for feeds generated from server monitoring, 289
- seeders, BitTorrent, 180
- SEED_TIME constant, 182
- SEED_TORRENT constant, 182
- select module, Python, 96, 99
- `sendEntriesViaIM()` function, 108–109
- `sendIM()` method, 100, 103–104
- `sendIMwithTemplate()` function, 109
- server. *See* Web server
- `__setitem__()` method, 246
- SFTP (Secure FTP), 240
- `SGMLParser`, parsing feeds using, 38–46
- shared feed cache, implementing, 575–584
- SHELL_TMPL template, 521–523
- shelve module, Python, 67–68
- sifting feeds
 - building feed generator for, 469–478
 - using feed generator for, 478–480
- SITE_NAME constant, 301, 305, 313
- SITE_ROOT constant, 305
- `<skipdays>` tag, 238
- `<skipHours>` tag, 237–238
- SMTP server, set to localhost, 81
- smtplib module, Python, 81
- SOAP modules, Python, 378
- SOAP Web services, 376
- socket module, Python, 99
- Source Code component, Plucker, 132
- SourceForge CVS repositories, 322–324
- spam, referrer, 306
- `speakTextIntoSoundFile()` function, 163
- speech synthesis, creating audio feeds using, 158–167
- spycyroll feed aggregator (Babu), 61
- `start()` method, 98–99
- `start_entry()` method, 43
- `start_feed()` method, 258
- `start_feed_entry()` method, 258
- `start_item()` method, 43
- START_TIMEOUT constant, 182
- state machine, 41
- STATE_FN constant
 - `AmazonScraper` class, 399
 - `CVSScraper` class, 335
 - `LOCScraper` class, 260–261
 - `Scraper` class, 250
 - `SVNScraper` class, 345
- status codes, HTTP, 72
- `stop()` method, 118–119
- Straup Cope, Aaron (XSLT feed conversion), 444
- string formatting, 54
- SUBJECT constant, 87
- subscriptions to feeds
 - managing, 114–125
 - subscribing, 49–51
- SUBSCRIPTION_TMPL template, 520–521
- `SubsException` class, 113
- `SubsNotSubscribed` class, 113
- Subversion repositories
 - commercial programs for, 335, 351
 - definition of, 340
 - finding, 340–341
 - installing, 340
 - logs
 - querying, 341–343
 - scraping, 343–350
- `summarizeEntry()` function, 455–456
- `<summary>` tag, 17

SUMMARY_TMPL template, 306–307, 313, 400
 svn command, 341–343
 SVN_BIN constant, 345
 svn_log() method, 347–348
 SVNScraper class, 345–348
 swarming, used by BitTorrent, 180–181
 Swartz, Aaron
 “Bake, Don’t Fry”, 227
 rss2email program, 91
 xmltramp module, 395–396, 490–491
 Syndic8 feed directory, 32
 Syndic8 Web Services, 34, 287
 syndic8feedfinder.py module, 33–36
 “Syndicate this Site” links, 3, 23
 syndication feeds. *See* feeds
 system logs, monitoring, 290–294

T

TAG_DATE constant, 399
 TAG_DOMAIN constant, 210, 363–364, 399
 TAG_PREFIX constant, 335
 tail command, log monitoring and, 290–294
 Talk to OSCAR (TOC), 94
 Technorati API
 account required for, 488–489
 definition of, 488
 searches, adding to feeds, 489–495
 searching with, 489
 TECHNORATI_KEY constant, 493
 technorati_search() method, 495
 TemplateDocWrapper class, 213–215
 templates
 for Amazon feeds, 399–400
 for Amazon items in feeds, 508–509
 for Atom feeds, 211, 213–215, 248
 for Bayes-enabled feeds, 457–458
 for CVS history feeds, 335
 for del.icio.us recaps in feeds, 501
 for email feeds, 363–364
 for feed aggregator output, 54, 57
 for feed metadata, 557, 558, 560, 561
 for instant messages, 109, 110
 for iPod-based feed aggregator, 148, 149
 for JavaScript includes, 531–532
 for links sifted from feeds, 471–472
 for login activity feeds, 313
 for microformat content, 544, 546–547
 for referrer feeds, 306–307
 for related links in feeds, 493
 for RSS feeds, 218–219, 248–249
 for scraped feeds, 245–249
 for Web logs, 520–523
 Terminal application, Mac OS X, 19

Text-to-Speech, creating audio feeds using, 158–167
 three-pane aggregators, 8–13
 TIDY_CMD constant, 277
 tidylib.py module, 276–278
 TIDY_OPTS constant, 277
 tidy_string() function, 278
 tidy_url() function, 278
 time format strings, Python, 69
 time module, Python, 68
 <title> tag, 15, 17, 208
 TITLE_TIME_FMT template, 471
 TITLE_TMPL template, 313, 335, 400, 471
 TOC (Talk to OSCAR), 94
 TocTalk class, 97
 tracker, BitTorrent, 180
 trainEntry() function, 455–456
 TrampTplWrapper class, 402–403
 translate_prop_name() method, 46
 <ttl> tag, 237–238
 Turner, Jamie (Py-TOC module, Python), 94

U

Ultra-Liberal Feed Finder module (Pilgrim), 36–37
 UNICODE_ENC constant, 56, 531–532
 United States Library of Congress, example using, 253–263
 Universal Feed Parser (Pilgrim)
 definition of, 48
 downloading, 48–49
 enclosures and, 171
 normalizing and converting feeds with, 419, 437–443
 using, 49
 UNIX. *See also* Linux; logs
 command shell tools for, 18–19
 system logs, monitoring, 290–294
 unknown_endtag() method, 44
 unknown_starttag() method, 44
 Unpolluted tool, 240
 unsubscribeFeed() function, 113
 update operation, CVS, 322
 <updateBase> tag, 239
 <updateFrequency> tag, 239
 <updatePeriod> tag, 239
 URI scheme for feeds, 26
 urlencode() function, 397–398
 urljoin() module, Python, 27, 31
 urllib module, 395–396
 urllib2.urlopen() module, Python, 27, 31
 urlopen() module, Python, 27, 31
 urlparse.urljoin() module, Python, 27, 31
 Usenet newsgroups, receiving feeds in, 92
 UserLand Software, Radio UserLand feed aggregator, 62–63

V

validating feeds, 220–222
 VALID_AUDIO_EXT constant, 192
 Version Control with Subversion, 340
 ViewCVS package, 335
 Viewer component, Plucker, 131

W

W3CDTF format, 213, 424, 426, 428–432
 walk() function, Python, 203
 Walsh, Norman (XSLT stylesheet for Subversion), 351
 Web logs (blogs)
 commercial programs for, 535–536
 creating from feed aggregator, 515–524
 links to feeds on, 23–24
 reposting feeds using MetaWeblog API, 524–529
 Web server
 compression, enabling, 231–233
 conditional GET, enabling, 235–236
 configuring MIME types for, 25
 monitoring
 Apache access logs, 304–312
 Apache error logs, 301–304
 incremental feeds for, 294–300
 issues involving, 289
 logs, 290–294
 security of, 289
 Web services
 Amazon, building feeds with
 from product searches, 403–407
 queries for, 395–398
 searches for, 398–403
 from wish lists, 407–412
 commercial programs for, 412–413
 for finding feeds, 32–37
 Google, building feeds with, 375–383
 SOAP Web services, 376
 Yahoo!, building feeds with, 383–394
 Web site resources. *See also* Web sites
 “A Plan for Spam” (Graham), 451
 AdiumX IM client, 105
 Amazon Web Services (AWS), 506
 AmphetaDesk feed aggregator, 8
 AmphetaRate service, 481
 Atom 0.3 specification, 211
 atomfeed feed generator, 223
 AvantGo program, 167
 baked-versus-fried feeds, 227
 Bayesian statistics, 451
 Beautiful Soup module, 288
 BitTorrent, 181
 Blogdex diffusion index, 485
 Bloxom blogging package, 223
 calendar events, 542

CIA open source notification system, 351
 Colloquy IRC client, 340–341
 compression, 230
 conditional GET, HTTP, 234
 cPickle module, 297
 CrispAds service, 513
 cURL tool, 489
 CVS commit triggers, 351
 CVS (Concurrent Version System), 322, 324
 Cygwin, Windows, 19
 Daypop search engine, 481, 485
 del.icio.us bookmarks manager, 485, 497
 difflib module, Python, 312
 digg news feed, 485
 dodgeit service, 373
 Doppler podcast receiver, 196
 email package, Python, 358
 ETag header, 234
 EVDB (Events and Venues Database), 570
 EventMeister package, 318
 Federal Communications Commission, 264
 Feed on Feeds feed aggregator (Minutillo), 61
 Feed Validator, 220, 222
 FeedBurner service, 443, 513
 FeedDemon feed aggregator (Bradbury), 64
 Feedpalooza Web site, 287
 Feedsplitter feed conversion, 444
 4Suite package, 20–21
 Gmail service, 373
 Gnews2RSS, Google News scraper (Bond), 412
 Google Web services, 376
 Growl global notification system, 350
 <guid> tag, 75
 gzip module, Python, 316
 hCalendar microformat, 539–541
 history operation, CVS, 325
 Hot Links link aggregator, 485
 HTML Tidy, 276
 HTMLParser class, Python, 29
 HTTP caching, 237
 HTTP status codes, 72
 httpcache module (Gregorio), 38
 iCalendar standard, 539, 541
 <id> tag, 75
 If-Modified-Since header, 234
 If-None-Match header, 234
 imaplib module, Python, 356
 Internet Message Format (RFC 2822), 357
 iPod Agent program, 167
 iPodder podcast receiver, 194–195
 iPodderX podcast receiver, 196
 Jabber instant messaging, 94, 95
 JabRSS chatbot, 126
 Kibo, 312

- Last-Modified header, 234
- LogMeister package, 318
- MailBucket service, 373
- md5 module, Python, 75
- MetaWeblog API, 524
- microformat specifications, 570
- MIME (Multipurpose Internet Mail Extensions), 357
- MIMEMultipart module, Python, 82
- MIMEText module, Python, 82
- mod_gzip module, Apache, 231
- Movable Type content management package, 226, 524
- NetNewsWire feed aggregator (Ranchero Software), 63
- Newspipe program, 91
- nntp/rss aggregator, 92
- “Normalizing Syndicated Feed Content” (Pilgrim), 419
- Note Reader for iPod, 142
- Open Source Web community packages, 6
- osascript command line tool, 160
- os.popen function, Python, 160
- Perl modules, installed, tracking, 317
- PHP iCalendar, 541–542
- Pickle module, 297
- Planet feed aggregator, 535
- Plucker package for PalmOS, 130, 131
- POP3 standard, 354
- popen2 module, Python, 312
- poplib module, Python, 354
- PyBlosxom blogging package, 223
- pygoogle package, 378
- PyRSS2Gen feed generator, 223
- Python, 20
- Py-TOC module, Python, 94
- QuickNews feed aggregator, 167
- Radio UserLand feed aggregator, 62–63
- reBlog package, 536
- referrer spam, 306
- regular expressions, 266, 267, 269
- REST style of Web services, 384
- RFC 822 format, 218
- rlog operation, CVS, 326
- RSS Digest service, 536
- RSS enclosures, 170
- rss2email program (Swartz), 91
- rss2jabber chatbot, 126
- RSSCalendar calendar events, 570
- RSS-IM Gateway IM chatbot, 126
- say command, Mac OS X, 158
- ScrappyGoo, Google News scraper (Yang), 412
- select module, Python, 99
- SGMLParser, 39, 41
- shelve module, Python, 67–68
- SOAP Web services, 376
- socket module, Python, 99
- SourceForge CVS repositories, 322
- speech on Mac OS, 158
- spycyroll feed aggregator (Babu), 61
- string formatting, 54
- Subversion repositories, 340
- Syndic8 feed directory, 32
- Syndic8 Web Services, 34, 287
- Technorati API, 488
- time format strings, Python, 69
- TOC protocol for AIM, 94
- Ultra-Liberal Feed Finder module (Pilgrim), 36
- United States Library of Congress, 253
- Universal Feed Parser (Pilgrim), 48, 419
- urllib2.urlopen() module, Python, 31
- urlparse.urljoin() module, Python, 31
- Version Control with Subversion, 340
- ViewCVS package, 335
- W3CDTF format, 213
- webSearch class, 389
- WebSVN application, 351
- The White House, 274
- WordPress publishing system, 224, 524
- xmlrpc.lib module, Python, 33, 527
- xmltramp module (Swartz), 395–396, 490
- XMPP Presence events, 102–103
- Xmpppy modules, Python, 95
- XPath, 278, 279
- XSLT, building feeds from Amazon using, 413
- XSLT feed conversion (Manalang), 444
- XSLT feed conversion (Straup Cope), 444
- XSLT stylesheet for Subversion (Walsh), 351
- Yahoo! News RSS feeds, 412
- Yahoo! Web services, 384
- Web sites. *See also* Web site resources
 - links to feeds on, 23–24
 - scraping
 - building feed scraper, 244–253
 - commercial programs for, 287–288
 - definition of, 13, 243–244
 - with HTML Tidy, 274–278
 - with HTMLParser class, 253–263
 - problems with, 244
 - with regular expressions, 264–274
 - tools for, 244
 - with XPath, 274–275, 278–286
- Web-based aggregators, 10
- <WebMaster> tag, 15
- webSearch class, Yahoo!, 389
- WebSVN application, 351
- The White House, example using, 274–287
- whitehouseScraper class, 285–286

Windows

- Event Log, monitoring, 290, 318
- FeedDemon feed aggregator (Bradbury), 64–65
- installing CVS on, 324
- installing 4Suite on, 20
- installing HTML Tidy, 276
- installing Python on, 20
- installing Subversion on, 340
- installing XPath on, 279
- loading Plucker documents to PalmOS, 141
- Radio UserLand feed aggregator, 62–63
- UNIX-based tools for, 19
- Windows XP, scheduled tasks on, 60–61
- wish list items, Amazon, 407–412
- WordPress publishing system, 224, 524
- `writeAggregatorPage()` function
 - `agg01_pollsubs.py` program, 53–54, 57
 - `agglib.py` module, 78–79, 82, 138
- `writeNote()` function, 150

X

- XML (Extensible Markup Language)
 - definition of, 13–14
 - tools for, 20–21
- “XML” links, 23
- `xmlrpc.lib` module, Python, 33–34, 527
- `xmltramp` module (Swartz), 395–396, 403, 490–491
- XMPP Presence events, 102–103
- Xmpppy modules, Python, 95
- XPath expressions for normalization of feeds, 419
- XPath, scraping Web sites using, 274–275, 278–286

- `xPathScraper` class, 282–284
- XSL Transformations. *See* XSLT
- XSLT processor, 433
- XSLT stylesheet for Subversion (Walsh), 351
- XSLT (XSL Transformations)
 - building feeds from Amazon using, 413
 - normalizing and converting feeds with
 - building normalizer, 420–432
 - commercial programs for, 444
 - conversion enabled by, 420
 - data model for, 418–419
 - testing normalizer, 434–436
 - XPath expressions for, 419
 - XSLT processor for, 433
 - tools for, 20–21

Y

- Yahoo! News RSS feeds, 412–413
- Yahoo! Web services
 - application ID for, 384–385, 387
 - definition of, 383–384
 - news searches, building feeds from, 390–394
 - REST-style interface for, 384
 - SDK for, 384–386
 - searches, building feeds from, 386–394
- `YahooNewsScraper` class, 391–393
- `YahooSearchScraper` class, 387–388
- Yang, Tim (ScrappyGoo, Google News scraper), 412
- `YWS_APP_ID` constant, 387
- `YWS_SEARCH_QUERY` constant, 387