

Index

SYMBOLS AND NUMERICS

- .. (double period symbol), 478
- / (forward slash), 477
- . (period symbol), 477
- 0 (zero) error code, 60

A

- absolute path, 7, 477
- Abstract Data Type (ADT), 208–210
- Access Control Lists (ACL), 491
- accessor methods (Objective-C), 237–238, 253–257
- ACL (Access Control Lists), 491
- actions (Interface Builder), 281
- Activity Monitor, 665
- ADC (Apple Developer Connection) website, 21, 48–53, 659
- AddFunc function, 180
- AddMediaSample function, 416
- AddMovieToStorage function, 409
- ADT (Abstract Data Type), 208–210
- aliases, 393–400
- alternate menu items, 103
- American National Standards Institute (ANSI), 154
- API (application program interface)
 - defined, 8
 - GraphicsImporter, 403
 - Human Interface Toolbox, 374
 - Macintosh File Manager, 392
 - Macintosh Toolbox, 373
- API reference
 - Cocoa, 279
 - content overview, 42–46
- AppKit framework, 16, 279
- Apple
 - MacApp application framework, 373
 - mailing lists, 660
 - website, 436
- Apple Developer Connection (ADC) website, 21, 48–53, 659
- Apple Events
 - defined, 530–531
 - Open Scripting Architecture (OSA), 531
- Apple Human Interface Guidelines, 15, 18–19, 41–42

AppleScript commands

- do shell script, 559
- exists, 565
- load script, 558
- osascript, 559
- run script, 558
- store script, 558

AppleScript scripting language

- Apple Events, 530–531
- classes, 558
- conditional branching, 540–541
- cross-platform functionality, 432
- data types, 531–537
- documentation, 436
- element specifiers, 553–555
- example script, 432–436
- exception handling, 544–545
- features, 432
- handlers, 537–540, 557–558
- inheritance, 556–557
- interapplication communication, 530
- loops, 541–543
- messaging, 530–531
- object-oriented programming (OOP), 531
- operators, 549–552
- polymorphism, 557
- properties, 555
- recordable applications, 530
- Script Editor, 30, 34–35, 528–530
- script objects, 555–558
- scriptable applications, 530
- Stock Quotes example, 546–549
- tell statement, 543–544
- variables, 531–535

AppleScript Studio

- AppleScript–Cocoa bridge, 568–570
- Interface Builder, 572–575
- Piles of Pictures
 - script, 578–586
 - user interface (UI), 575–578
- Xcode, 570–572

AppleScripts

- creating, 528–530
- example, 432–436
- loading, 558
- Piles of Pictures, 578–586

AppleScripts (continued)

- running, 530, 558–559
- saving, 530
- Slideshow Maker, user interface (UI), 559–567
- storing, 558

application frameworks

- API reference, 42–46
- CamelBones, 447, 451
- Carbon
 - ApplicationServices framework, 17
 - C interfaces, 17
 - Carbon framework, 17
 - connecting interfaces, 377–378
 - controls, 387–392
 - Core Services framework, 17
 - CoreFoundation framework, 374, 377
 - development history, 16–17
 - Email Formatter, 378–382
 - events, 382–387
 - File Manager, 392–400
 - handles, 375–376
 - memory functions, 375
 - memory management, 375–376
 - menus, 98–103
 - resizable windows, 120–123
 - user interface (UI), 374
- Cocoa
 - API reference, 279
 - AppKit framework, 16, 279
 - AppleScript–Cocoa bridge, 568–570
 - application preferences, 359–367
 - archiving objects, 356–359
 - connecting interfaces, 280–304
 - CoreFoundation framework, 374
 - development history, 16
 - documents, 306–307, 336–339
 - drawers, 343–345
 - Email Formatter, 282–304
 - Foundation framework, 16, 279
 - instances, 280–281
 - Java bindings, 16
 - keyboard events, 368–370
 - menus, 103–106, 322–327
 - Model/View/Controller design pattern, 305–306
 - NeXTSTEP technology, 229
 - nib files, 310–311
 - notification system, 340–343
 - NSApplication class, 311
 - NSAutoreleasePool class, 251–253, 296
 - NSBox class, 284
 - NSButton class, 311
 - NSCoder class, 358–359
 - NSControl class, 295–296, 320
 - NSData class, 314
 - NSDocument class, 306, 308, 311–315
 - NSDrawer class, 345

- NSImage class, 314
- NSImageView class, 304, 310, 317–318, 320–322
- NSLayoutManager class, 305–306
- NSMutableArray class, 330–333
- NSNotificationCenter class, 342–343
- NSObject class, 249, 288
- NSObjectController class, 297–298
- NSOpenPanel class, 328–330, 334–336
- NSResponder class, 310
- NSScrollView class, 307, 310
- NSString class, 295
- NSTableView class, 346–349
- NSTask class, 604–607
- NSTextField class, 289–290, 294–295
- NSTextStorage class, 305
- NSTextView class, 284
- NSUserDefaultsController class, 298
- NSWindowController class, 306, 308, 315–317
- Objective-C, 16
- QuickTime, 400
- redo support in documents, 354–355
- replacing MyDocument files, 318–319
- resizable windows, 113–120
- table data sources, 349–353
- table views, 346–349
- undo support in documents, 354–355

common features, 15

- conceptual documentation, 40–41
- defined, 4
- HI Toolbox, 374
- Java, 17–18
- MacApp, 373
- Metrowerks PowerPlant, 373
- PyObjC, 437–438
- Python, 437
- RubyCocoa, 466

application preferences

- built-in preferences system, 149–150
- Cocoa, 359–367
- editing with Property List Editor, 150
- file formats, 149–150
- restoring, 144
- storing, 148–149
- viewing, 144–148

application program interface (API)

- defined, 8
- GraphicsImporter, 403
- Human Interface Toolbox, 374
- Macintosh File Manager, 392
- Macintosh Toolbox, 373

applications

- bundles
 - building, 136–144
 - bundle bit, 128
 - defined, 127
 - executable code, 131

- features, 128
 - global resources, 131
 - Info.plist file, 130–131
 - localized resources, 131–132
 - Mail application, 132–136
 - opaque directories, 128
 - Slide Master application, 142
 - Slide Master bundle, 137–141, 143–144
 - versioned bundles, 128–129
 - Crumbs
 - archiving script, 612–619
 - controls, 608–611
 - CrumbsController class, 608–611
 - CrumbsController.h file, 609
 - CrumbsController.m file, 609–611
 - NSTimer class, 611
 - user interface (UI), 607–611
 - defined, 7
 - packages, 127
 - recordable applications, 530
 - scriptable applications, 530
 - Searcher
 - Cocoa interface, 595–601
 - search engine, 601–603
 - Searcher class, 600
 - Slide Master
 - application delegate, 325–326
 - archiving objects, 356–359
 - drawers, 343–345
 - keyboard events, 368–370
 - menu bar, 322–325
 - mutable arrays, 330–333
 - MyDocument files, 318–319
 - notifications, 340–343
 - NSDocument class, 307, 311–315
 - NSImageView class, 309–310, 317–318, 320–322
 - NSScrollView class, 309–310
 - NSWindowController class, 307, 315–317
 - open panels, 328–330, 334–336
 - preferences panel, 360–367
 - QuickTime movies, 401–420
 - redo feature, 354–355
 - scrollable image view, 307–310
 - SlideShowDocument class, 336–339
 - table data sources, 349–353
 - table views, 346–349
 - undo feature, 354–355
 - Terminal
 - Bash shell, 470–471
 - configuring, 471
 - features, 6–7, 11, 669
 - UI (user interface), 15
 - wrappers, 127
 - X11, 470–471
 - apropos **command, 621**
 - Aqua user interface, 15**
 - archivers, 358**
 - archiving objects, 356–359**
 - arguments**
 - C functions, 180, 186
 - defined, 10
 - arithmetic operators**
 - AppleScript scripting language, 549–550
 - C programming language, 159–161
 - arrays**
 - C programming language, 163–166
 - Perl, 451
 - assembler, 154**
 - AT&T, 6**
 - autocompletion feature (Bash shell), 495**
 - Automator, 663**
 - autorelease method, 251–253**
 - autorelease pools, 251–253, 296, 377**
 - autoreleasing, 24–25**
 - Autosizing control, 628–629**
 - awk command, 502–503**
- ## B
- background commands, 480**
 - background processes, 505**
 - backtrace, 23**
 - bash command, 510**
 - Bash Reference Manual website, 432**
 - Bash shell**
 - AppleScripts, 559
 - autocompletion feature, 495
 - conditional branching, 515–518
 - configuring, 473–476
 - defined, 35, 38
 - error codes, 480
 - executables, 480
 - file permissions, 491–494
 - globbing, 494–495
 - input/output (I/O), 481–483
 - interactive use, 470
 - login shell, 473
 - loops, 518–520
 - pipes, 481–483
 - processes, 480
 - script example, 427–432
 - scripts
 - commands, 509–510
 - file compression script, 521–524
 - functions, 520–521
 - quoting, 513–515
 - running, 470, 510
 - shebang, 510
 - variables, 510–513
 - Terminal application, 470–471
 - text editors, 471–472
 - X11 application, 471

Bell Labs, 153

BigTop, 665

`/bin` directory, 478

bindings

Cocoa Bindings

- conceptual documentation, 304
- connecting controls to `NSObjectController`, 298–300
- controller keys, 301
- editability, 301
- features, 280, 296
- Key Value Coding, 247, 304
- Key Value Observing, 304
- model key paths, 300–301, 303–304
- `NSArrayController` class, 298
- `NSObjectController` class, 297–298
- `NSUserDefaultsController` class, 298
- value transformers, 361

Java, 16

Python, 437

Bluetooth Explorer, 667

Bourne Again Shell (Bash). See Bash shell

branching

- AppleScript scripting language, 540–541
- Bash shell, 515–518
- C programming language, 169–173

bridges

- AppleScript–Cocoa bridge, 568–570
- Toll-Free Bridging, 374, 390, 392

BSD installer package, 21–22

bugs

- fence-post bugs, 177
- off-by-one bug, 177

build results (Xcode), 78

build styles (Xcode), 78

build tools

- Automator, 663
- Interface Builder, 22, 663
- Xcode, 22, 663

`BuildImageVideoSamples` function, 415–416

`BuildImageVideoTrack` function, 411

building

- application bundles, 136–144
- installer packages, 30–33
- Xcode projects, 72–77

built-in preferences system, 149

bundles

- application bundles
 - building, 136–144
 - defined, 127
 - localizations, 127
 - Slide Master application, 142
 - Slide Master bundle, 137–141, 143–144
- bundle bit, 128
- executable code, 131
- formats, 128

global resources, 131

`Info.plist` file, 130–131

localized resources, 131–132

Mail application, 132–136

opaque directories, 128

plug-ins, 127–128

symbolic links, 129

versioned bundles, 128–129

business resources, 53

C

C programming language

Abstract Data Type (ADT), 208–210

arrays, 163–166

assembler functionality, 154

C99 update, 154

characters, 183–184

comments, 156, 168

conditional branching, 169–173

data structures, 194–196

data types, 157–159, 194–196

development history, 153–154

functions

`AddFunc`, 180

arguments, 180, 186

calling, 180

declaring, 180–181

`fgetc`, 190

`fgets`, 190

`fprintf`, 189–190

`fscanf`, 189–190

`getchar`, 190

`gets`, 190

main, 154–155

parameter list, 155

parameters, 155, 180

pass-by-value, 182

`popen`, 594

`pow`, 626

`printf`, 155, 180, 187, 626

recursion, 181

return types, 154

return values, 180

running subprocesses, 589

`scanf`, 187–188, 626

signatures, 181

`sprintf`, 188–189

`sscanf`, 188–189

`strcat`, 185

`strcmp`, 185

`strcpy`, 184

`strlen`, 185

`strncat`, 185

`strncmp`, 185

`strncpy`, 184

- strstr, 185
 - system, 590–594
 - Grepper program, 191–194
 - header files, 154, 205
 - if statements, 161
 - if/else statements, 169–171
 - if/elseif/else statements, 171
 - infinite loops, 174
 - input/output, 186–190
 - keywords
 - enum, 194
 - NULL, 185
 - typedef, 194
 - void, 180
 - linked lists, 197–198
 - loops, 173–177
 - memory management, 196–200
 - MyAddressBook program
 - AddressBook ADT, 208–218
 - AddressBook.c file, 210–213
 - AddressBook.h file, 210
 - Controller.c file, 218–224
 - Controller.h file, 218
 - Person.c file, 207–208
 - Person.h file, 207–208
 - new line characters, 157
 - Objective-C programming language, 227
 - operators, 159–162
 - organizing C programs, 205–206
 - pointers, 164–169
 - precedence of operators, 161–162
 - preprocessor, 200–205
 - preprocessor directives, 154, 206
 - programs
 - compiling, 155–156
 - running, 156
 - source files, 205
 - statistical analysis of array data, 177–179
 - strings, 183–186
 - structured programming, 208
 - switch/case statements, 171–172
 - terminating null character, 183
 - ternary operator, 172–173
 - variable declarations, 158–159
 - variables, 157–159
- C Programming Language (Brian Kernighan and Dennis Ritchie), 154**
- Calculator program, 65–69**
- call stack, 23**
- calling functions in C, 180**
- CamelBones framework, 447, 451**
- Carbon**
- ApplicationServices framework, 17
 - C interfaces, 17
 - Carbon framework, 17
 - connecting interfaces, 377–378
 - controls, 387–392
 - Core Services framework, 17
 - CoreFoundation framework, 374, 377
 - development history, 16–17
 - Email Formatter, user interface (UI), 378–382
 - events, 382–387
 - File Manager, 392–400
 - functions
 - DisposeHandle, 375–376
 - DisposePtr, 375
 - memory functions, 375
 - NewHandle, 375–376
 - NewPtr, 375
 - NewPtrClear, 376
 - handles, 375–376
 - memory functions, 375
 - memory management, 375–376
 - menus, 98–103
 - resizable windows, 120–123
 - user interface (UI), 374
- Carbon Simulator, 95–96**
- case-sensitivity, 11**
- cat command, 501**
- cc command-line tool, 35**
- cd command, 486–488, 496**
- CFBundleDocumentTypes Info.plist **key, 130**
- CFBundleExecutable Info.plist **key, 130**
- CFBundleGetInfoString Info.plist **key, 130**
- CFBundleIconFile Info.plist **key, 130**
- CFBundleIdentifier Info.plist **key, 130**
- CFBundleName Info.plist **key, 130**
- CFBundlePackageType Info.plist **key, 130**
- CFBundleSignature Info.plist **key, 130**
- CFCreate... **function, 377**
- CFRelease **function, 377**
- CFRetain **function, 377**
- changing**
- file permissions, 493–494
 - shells, 39
- characters in C, 183–184**
- chat rooms, 661**
- checking status of processes, 505–506**
- chmod **command, 491, 493, 496**
- chown **command, 493**
- chsh **command, 39**
- CHUD Tools (Common Hardware Understanding Development Tools), 27**
- class methods (Objective-C), 232–233**
- classes**
- AppleScript scripting language, 558
 - Cocoa
 - NSApplication, 311
 - NSAutoreleasePool, 251–253, 296

classes (continued)

classes (continued)

- NSBox, 284
- NSButton, 311
- NSCoder, 358–359
- NSControl, 295–296, 320
- NSData, 314
- NSDocument, 306, 308, 311–315
- NSDocument class, 306
- NSDrawer, 345
- NSImage, 314
- NSImageView, 304, 309–310, 317–318, 320–322
- NSLayoutManager, 305–306
- NSMutableArray, 330–333
- NSNotificationCenter, 342–343
- NSObject, 249, 288
- NSObjectController, 297–298
- NSOpenPanel, 328–330, 334–336
- NSResponder, 310
- NSScrollView, 307, 310
- NSString, 295
- NSTableView, 346–349
- NSTask, 604–607
- NSTextField, 289–290, 294–295
- NSTextStorage, 305
- NSTextView, 284
- NSUserDefaultsController, 298
- NSWindowController, 306, 308, 315–317
- defined, 228
- Objective-C programming language, 229–231
- CLI (command-line interface), 9, 469–470**
- Cocoa**
 - API reference, 279
 - AppKit framework, 16, 279
 - AppleScript–Cocoa bridge, 568–570
 - application preferences, 359–367
 - archiving objects, 356–359
 - classes
 - NSApplication, 311
 - NSAutoreleasePool, 251–253, 296
 - NSBox, 284
 - NSButton, 311
 - NSCoder, 358–359
 - NSControl, 295–296, 320
 - NSData, 314
 - NSDocument, 306, 308, 311–315
 - NSDrawer, 345
 - NSImage, 314
 - NSImageView, 304, 309–310, 317–318, 320–322
 - NSLayoutManager, 305–306
 - NSMutableArray, 330–333
 - NSNotificationCenter, 342–343
 - NSObject, 249, 288
 - NSObjectController, 297–298
 - NSOpenPanel, 328–330, 334–336
 - NSResponder, 310
 - NSScrollView, 307, 310
 - NSString, 295
 - NSTableView, 346–349
 - NSTask, 604–607
 - NSTextField, 289–290, 294–295
 - NSTextStorage, 305
 - NSTextView, 284
 - NSUserDefaultsController, 298
 - NSWindowController, 306, 308, 315–317
 - connecting interfaces
 - with Cocoa Bindings, 296–304
 - with Interface Builder, 280–296
 - controls, 320–322
 - CoreFoundation framework, 374
 - development history, 16
 - documents
 - creating, 306–307, 336–339
 - redo support, 354–355
 - replacing MyDocument files, 318–319
 - undo support, 354–355
 - drawers, 343–345
 - Email Formatter
 - Cocoa Bindings, 297–298
 - connecting controls, 290–293, 298–304
 - EmailAddress class, 302–304
 - EmailController class, 287–290, 292–296
 - key view loop, 284–287
 - NSObjectController instance, 297–298
 - user interface (UI), 282–284
 - Foundation framework, 16, 279
 - instances, 280–281
 - Java bindings, 16
 - keyboard events, 368–370
 - menus
 - connecting, 322–325
 - creating, 103–106
 - menu commands, 326–327
 - Model/View/Controller design pattern, 305–306
 - NeXTSTEP technology, 229
 - nib files, 310–311
 - notification system, 340–343
 - NSTask class, 607
 - Objective-C, 16
 - QuickTime, 400
 - resizable windows, 113–120
 - table data sources, 349–353
 - table views, 346–349
- Cocoa Bindings**
 - conceptual documentation, 304
 - connecting controls to NSObjectController, 298–300
 - controller keys, 301
 - editability, 301
 - features, 280, 296
 - Key Value Coding, 247, 304
 - Key Value Observing, 304
 - model key paths, 300–301, 303–304

- NSArrayController class, 298
- NSObjectController class, 297–298
- NSUserDefaultsController class, 298
- value transformers, 361
- Cocoa Dev Central website, 659**
- CocoaDev chat room, 661**
- CocoaDev website, 659**
- Code Editor (Xcode), 65–70**
- code examples, 46–47**
- codecs (QuickTime), 408**
- command-line interface (CLI), 9, 469–470**
- command-line tools**
 - cc, 35
 - DeRez, 35, 37–38, 669
 - diff, 35–36, 669
 - gcc, 35–36, 669
 - gdb, 35–37, 669
 - GetFileInfo, 35, 37, 670
 - ld, 670
 - leaks, 670
 - make, 670
 - man, 670
 - Rez, 35, 37–38, 670
 - sample, 35, 38, 671
 - SetFile, 35, 37, 623–625, 671
 - top, 35, 39, 671
- commands (AppleScript)**
 - do shell script, 559
 - exists, 565
 - load script, 558
 - osascript, 559
 - run script, 558
 - store script, 558
- commands (Nano text editor), 472**
- commands (Unix)**
 - apropos, 621
 - arguments, 10
 - awk, 502–503
 - background commands, 480
 - bash, 510
 - cat, 501
 - cd, 486–488, 496
 - chmod, 491, 493, 496
 - chown, 493
 - chsh, 39
 - cp, 490, 496
 - crontab, 508
 - curl, 508
 - dig, 508
 - diskutil, 508
 - ditto, 508
 - echo, 11, 486, 500
 - eval, 481
 - exec, 481
 - exit, 480–481
 - find, 136, 480, 499–500
 - flags, 10
 - foreground commands, 480
 - ftp, 509
 - grep, 501–502
 - gunzip, 509
 - gzip, 509
 - hdiutil, 508
 - installer, 508
 - kill, 506–507
 - killall, 507
 - locate, 499
 - ls
 - directories, 9–10, 478–479, 487–488
 - file permissions, 492
 - file system, 487–488, 496
 - man page, 479
 - man, 10–11, 479, 621
 - mdfind, 498
 - mkdir, 491, 496
 - mount_afp, 508
 - mv, 488–490, 496
 - netstat, 509
 - nice, 509
 - open, 508
 - pipes, 481–483
 - pmset, 508
 - ps, 505–507
 - pwd, 486
 - renice, 509
 - rm, 491
 - running, 480
 - sed, 503–505
 - sftp, 509
 - sleep, 506–507
 - source, 481
 - ssh, 509
 - standard error, 483
 - sudo, 493–494, 496–497
 - system_profiler, 484–486, 508
 - SystemStarter, 508
 - tar, 509
 - touch, 501
 - unzip, 509
 - wc, 503
 - which, 479
 - zip, 509
- comments**
 - C programming language, 156, 168
 - Python, 440–441
- Common Hardware Understanding Development Tools (CHUD Tools), 27**
- compiled languages, 425–426**
- compilers**
 - defined, 426
 - interpreters, 426

compilers (continued)

- Java, 451
- Xcode, 77

compiling programs in C, 155–156

Comprehensive Perl Archive Network (CPAN) website, 446, 451

`CompressImage` function, 416

conceptual documentation, 40–41

conditional branching

- AppleScript scripting language, 540–541
- Bash shell, 515–518
- C programming language, 169–173

configuring

- Bash shell, 473–476
- Terminal application, 471

connecting

- Carbon interfaces, 377–378
- Cocoa interfaces
 - with Cocoa Bindings, 296–304
 - with Interface Builder, 280–296
- Cocoa menus, 322–325

content rects (windows), 317

control statements. See conditional branching; loops

controllers, 305

controls

- Autosizing control, 628–629
- Carbon, 387–392
- Cocoa, 320–322
- Interface Builder, 281
- windows, 113

Core Image Fun House, 664

core OS, 4

CoreFoundation framework

- autorelease pools, 377
- data types, 377, 389–390
- features, 374
- functions
 - `CFCreate...` function, 377
 - `CFRelease`, 377
 - `CFRetain`, 377
- memory conventions, 377
- Toll-Free Bridging, 374, 390, 392

`cp` command, 490, 496

CPAN (Comprehensive Perl Archive Network) website, 446, 451

`CreateMovieStorage` function, 408–409

creating

- AppleScripts, 528–530
- documents, 306–307, 336–339
- drawers, 344–345
- menus
 - Carbon menus, 98–103
 - Cocoa menus, 103–106
- Objective-C objects, 247–249
- QuickTime movies, 404–409
- user interface (UI), 93–97

- windows, 106–112
- Xcode projects, 58–60

creator code (files), 37

`crontab` command, 508

Crumbs application

- archiving script, 612–619
- controls, 608–611
- `CrumbsController` class, 608–611
- `CrumbsController.h` file, 609
- `CrumbsController.m` file, 609–611
- `NSTimer` class, 611
- user interface (UI), 607–611

`curl` command, 508

current directory, 9

custom controls, 320–322

D

daemons, 505

Darwin

- command-line interface, 9–11
- help system, 9–10
- kernel, 7–8
- source code, 6
- system library, 7–9
- Terminal application, 6–7

DarwinPorts project website, 466

Dashboard

- `DashBall` widget, 453–465
- Dashboard Programming Guide, 465
- documentation, 465
- widget bundles, 452
- widgets, 452–465

data forks, 37

data hiding, 237–238

data types

- AppleScript scripting language, 531–537
- C programming language, 157–159, 194–196
- CoreFoundation framework, 377, 389–390
- `FSRef` data type, 393
- `FSSpec` data type, 392–393

`dealloc` method, 240, 249–250, 256–257

debuggers

- `gdb`, 36–37
- Xcode's debugger, 78–84

declaring functions in C, 180–181

declaring variables

- C programming language, 158–159
- JavaScript, 457
- Perl, 448–449

`def` keyword (Python), 441

delegate objects, 325–326

delegation (Objective-C), 245–246

deleting

- files/directories, 491
- Objective-C objects, 249–250

DeRez **command-line tool**, 35, 37–38, 669

designated initializers, 248–249

developer support

API reference, 42–46

Apple Developer Connection (ADC) website, 21, 48–54

chat rooms, 661

code examples, 46–47

documentation, 39–42

MacTech Magazine, 661

mailing lists, 53, 660

man pages, 47–48

Reference Library, 39–42

developer tools

Activity Monitor, 665

Automator, 663

BigTop, 665

Bluetooth Explorer, 667

BSD installer package, 21–22

CHUD Tools, 27

Core Image Fun House, 664

DeRez, 35, 37–38, 669

FileMerge, 30–31, 667

icns Browser, 667

Icon Composer, 30–31, 667

Interface Builder, 22, 663

IORegistryExplorer, 667–668

Jar Bundler, 665

JavaBrowser, 665

MallocDebug, 23–24, 665

MONster, 665–666

ObjectAlloc, 23–25, 666

OpenGL Driver Monitor, 664

OpenGL Profiler, 664

OpenGL Shader Builder, 664

PackageMaker, 30–33, 668

PacketLogger, 668

Pixie, 30, 33, 664

profilers, 23

Property List Editor, 30, 33–34, 455, 668

Quartz Composer, 664

Quartz Debug, 23, 25–26, 666

Repeat After Me, 668

Sampler, 23, 26–27, 621–623, 666

Saturn, 666

Script Editor, 30, 34–35

Shark, 23, 27–28, 666

Spin Control, 23, 28, 667

SRLanguageModeler, 668

Terminal, 669

Thread Viewer, 23, 28–30, 667

USB Prober, 669

Xcode, 22, 663

device drivers, 8

dialog windows, 106–112

diff command-line tool, 35–36, 669

dig command, 508

directories

/bin directory, 478

current directory, 9

home directory, 9–10

Language Project directories, 131

root directory, 7

/sbin directory, 479

/usr/bin directory, 479

discounts on hardware, 53

diskutil command, 508

DisposeHandle function, 375–376

DisposePtr function, 375

ditto command, 508

do shell script command, 559

document windows, 106

documentation

Apple Developer Connection (ADC) website, 49–52

AppleScript, 436

Dashboard, 465

Python, 445

Reference Library, 39–42

Xcode, 57, 84–89

documents

creating, 306–307, 336–339

redo support, 354–355

replacing MyDocument files, 318–319

undo support, 354–355

double period symbol (..), 478

downloading

Slide Master application, 4

stock quotes, 546–549

drawers

creating, 344–345

features, 343

states, 345

DrawSlideInGWorld function, 420

dynamic menu items, 103

dynamically typed programming languages, 243

E

echo command, 11, 486, 500

editing property lists, 455

editors

AppleScript Script Editor, 528–530

Bash shell, 471–472

Code Editor (Xcode), 65–70

emacs text editor, 471

Nano text editor, 471–472

Property List Editor, 30, 33–34, 455, 668

Script Editor, 30, 34–35

vi text editor, 471

element specifiers in AppleScript, 553–555

emacs text editor, 471

Email Formatter (Carbon version), 378–382

Email Formatter (Cocoa version)

Cocoa Bindings, 297–298
connecting controls, 290–291, 293, 298–304
EmailAddress class, 302–304
EmailController class, 287–290, 292–296
key view loop, 284–287
NSObjectController instance, 297–298
user interface (UI), 282–284

empty menus, 102

encapsulation, 228, 237

EnterMovies **function, 419**

enum **keyword, 194**

enumerated types, 194–195

error codes

Bash shell, 480
zero (0) error code, 60

errors

QuickTime, 420
Xcode, 78

eval **command, 481**

events

Apple Events
 defined, 530–531
 Open Scripting Architecture (OSA), 531
Carbon events, 382–387
keyboard events, 368–370

exception handling

AppleScript scripting language, 544–545
Objective-C programming language, 235

exec **command, 481**

executable code (bundles), 131

executables (defined), 7, 480

exists **command, 565**

exit **command, 480–481**

exportMovie: **method, 419–420**

F

fence-post bugs, 177

fgetc **function, 190**

fgets **function, 190**

file extensions, 7, 60

File Manager

aliases, 393–400
FSRef data type, 392–393
FSSpec data type, 392–393

file path, 6–7

file permissions

Access Control Lists (ACL), 491
changing, 493–494
viewing, 491–492

File::Find **module (Perl), 448**

FileMerge, 30–31, 667

files

creator code, 37
Portable Document Format (PDF), 14

Rez files, 37–38

type code, 37
uncompressing, 4

find **command, 136, 480, 499–500**

flags, 10

for **loops**

C programming language, 175–177
Python scripting language, 443–444

foreground commands, 480

formats of bundles, 128

forward slash (/), 477

Foundation framework

features, 16, 279
Toll-Free Bridging, 390, 392

fprintf **function, 189–190**

frame rects (windows), 317

frame stack, 23

frames, 23

frameworks

API reference, 42–46
CamelBones, 447, 451
Carbon
 ApplicationServices framework, 17
 C interfaces, 17
 Carbon framework, 17
 connecting interfaces, 377–378
 controls, 387–389, 391–392
 Core Services framework, 17
 CoreFoundation framework, 374, 377
 development history, 16–17
 Email Formatter, 378–382
 events, 382–387
 File Manager, 392–400
 handles, 375–376
 memory functions, 375
 memory management, 375–376
 menus, 98–103
 resizable windows, 120–123
 user interface (UI), 374

Cocoa

API reference, 279
AppKit framework, 16, 279
AppleScript–Cocoa bridge, 568–570
application preferences, 359–367
archiving objects, 356–359
connecting interfaces, 280–304
CoreFoundation framework, 374
development history, 16
documents, 306–307, 336–339
drawers, 343–345
Email Formatter, 282–303
Foundation framework, 16, 279
instances, 280–281
Java bindings, 16
keyboard events, 368–370
menus, 103–106, 322–327

- Model/View/Controller design pattern, 305–306
- NeXTSTEP technology, 229
- nib files, 310–311
- notification system, 340–343
- NSApplication class, 311
- NSAutoreleasePool class, 251–253, 296
- NSBox class, 284
- NSButton class, 311
- NSCoder class, 358–359
- NSControl class, 295–296, 320
- NSData class, 314
- NSDocument class, 306, 308, 311–315
- NSDrawer class, 345
- NSImage class, 314
- NSImageView class, 304, 310, 317–318, 320–322
- NSLayoutManager class, 305–306
- NSMutableArray class, 330–333
- NSNotificationCenter class, 342–343
- NSObject class, 249, 288
- NSObjectController class, 297–298
- NSOpenPanel class, 328–330, 334–336
- NSResponder class, 310
- NSScrollView class, 307, 310
- NSString class, 295
- NSTableView class, 346–349
- NSTask class, 604–607
- NSTextField class, 289–290, 294–295
- NSTextStorage class, 305
- NSTextView class, 284
- NSUserDefaultsController class, 298
- NSWindowController class, 306, 308, 315–317
- Objective-C, 16
- QuickTime, 400
- redo support in documents, 354–355
- replacing MyDocument files, 318–319
- resizable windows, 113–120
- table data sources, 349–353
- table views, 346–349
- undo support in documents, 354–355
- common features, 15
- conceptual documentation, 40–41
- defined, 4
- HIToolbox, 374
- Java, 17–18
- MacApp, 373
- PyObjC, 437–438
- Python, 437
- RubyCocoa, 466
- Free Software Foundation**
 - gcc command-line tool, 36
 - gdb command-line tool, 36–37
- fscanf **function (C programming language), 189–190**
- FSRef **data type, 393**
- FSSpec **data type, 392–393**
- ftp **command, 509**
- full path, 7**
- functions**
 - Bash shell scripts, 520–521
 - C functions
 - AddFunc, 180
 - arguments, 180, 186
 - calling, 180
 - declaring, 180–181
 - fgetc, 190
 - fgets, 190
 - fprintf, 189–190
 - fscanf, 189–190
 - getchar, 190
 - gets, 190
 - main, 154–155
 - parameter list, 155
 - parameters, 155, 180
 - pass-by-value, 182
 - popen, 594
 - pow, 626
 - printf, 155, 180, 187, 626
 - recursion, 181
 - return types, 154
 - return values, 180
 - running subprocesses, 589
 - scanf, 187–188, 626
 - signatures, 181
 - sprintf, 188–189
 - sscanf, 188–189
 - strcat, 185
 - strcmp, 185
 - strcpy, 184
 - strlen, 185
 - strncat, 185
 - strncmp, 185
 - strncpy, 184
 - strstr, 185
 - system, 590–594
 - Carbon
 - DisposeHandle, 375–376
 - DisposePtr, 375
 - memory functions, 375
 - NewHandle, 375–376
 - NewPtr, 375
 - NewPtrClear, 376
 - CoreFoundation framework
 - CFCreate... function, 377
 - CFRelease, 377
 - CFRetain, 377
 - defined, 23
 - Perl, 449–451
 - QuickTime
 - AddMediaSample, 416
 - AddMovieToStorage, 409
 - BuildImageVideoSamples, 415–416
 - BuildImageVideoTrack, 411
 - CompressImage, 416

functions (continued)

CreateMovieStorage, 408–409
DrawSlideInGWorld, 420
EnterMovies, 419
GetMaxCompressionSize, 415
InsertMediaIntoTrack, 411–412
NewTrackMedia, 411
SaveImageMovie, 408, 419–420

G

gcc **command-line tool**, 35–36, 669
gdb **command-line tool**, 35–37, 669
getchar **function**, 190
GetFileInfo **command-line tool**, 35, 37, 670
GetMaxCompressionSize **function**, 415
gets **function**, 190
global resources (bundles), 131
global variables, 192–193
globbing, 494–495
glue languages, 426–427
graphical user interface (GUI). See **user interface (UI)**
graphics
Core Image Fun House, 664
OpenGL Driver Monitor, 664
OpenGL library, 12, 14
OpenGL Profiler, 664
OpenGL Shader Builder, 664
Pixie, 664
Portable Document Format (PDF), 14
Quartz 2D library, 12, 14
Quartz Composer, 664
Quartz Compositor, 12–13
QuickDraw graphics environment, 400
QuickTime, 12, 15
GraphicsImporter API, 403
grep **command**, 501–502
Grepper program
C version, 191–194
Objective-C version, 233–237
groups (Xcode), 64–65
GUI (graphical user interface). See **user interface (UI)**
gunzip **command**, 509
GWorld port (QuickTime), 403–404
gzip **command**, 509

H

handlers (AppleScript), 537–540, 557–558
handles, 375–376
hardware
device drivers, 8
discounts, 53
hdiutil **command**, 508
header files (C programming language), 154, 205
help system, 9–10

HFS file system, 37
HFS+ file system, 37
HIToolbox application framework, 374
home directory, 9–10
Human Interface Toolbox, 374

I

icns Browser, 667
Icon Composer, 30–31, 667
icons, 102
IDE (Integrated Development Environment). See **Xcode**
if statements
AppleScript scripting language, 540–541
C programming language, 161
Python scripting language, 445
if/else statements
AppleScript scripting language, 540–541
C programming language, 169–171
if/elseif/else statements in C, 171
implementation block (Objective-C classes), 229–231
indenting code, 70–72
infinite loops, 174
Info.plist file, 130–131, 631
informal protocols, 244–246
inheritance
AppleScript scripting language, 556–557
defined, 228, 238–241
init method, 240
initializers, 247–249
input/output (I/O)
Bash shell, 481–483
C programming language, 186–190
InsertMediaIntoTrack **function**, 411–412
Inspector window (Interface Builder), 97
installer **command**, 508
installer packages
BSD, 22
building, 30–33
instance methods (Objective-C), 232–233
instances
Cocoa, 280–281
Interface Builder, 97
NSObjectController, 297–298
NSTextView, 284
Integrated Development Environment (IDE). See **Xcode**
interapplication communication, 530
interface block (Objective-C classes), 229–231
Interface Builder
actions, 281
AppleScript Studio, 572–575
Carbon Simulator, 95–96
Control-drag tips, 293
controls, 281
features, 22, 663
Inspector window, 97

- interfaces
 - creating, 93–97
 - instances, 97
 - testing, 95–97
- menus
 - Carbon, 98–103
 - Cocoa, 103–106
- nib files, 93
- outlets, 281, 292–293
- Palettes window, 97, 281
- Release Notes window, 94
- table data sources, 349
- table views, 346–349
- windows
 - controls, 113
 - creating, 106–112
 - dialog windows, 106–112
 - document windows, 106
 - labels, 113
 - resizable windows, 113–123
 - text labels, 113
- International Standards Organization (ISO), 154**
- interpreted languages, 426**
- interpreters, 426**
- intro man pages, 625**
- I/O (input/output)**
 - Bash shell, 481–483
 - C programming language, 186–190
- IORegistryExplorer, 667–668**
- ISO (International Standards Organization), 154**

J

- Jar Bundler, 665**
- Java, 17–18**
- Java bindings, 16**
- Java virtual machine (JVM), 451–452**
- JavaBrowser, 665**
- JavaScript**
 - Dashboard, 452–465
 - features, 451
 - Java virtual machine (JVM), 451–452
 - var keyword, 457
 - variable declarations, 457
- JVM (Java virtual machine), 451–452**

K

- kernel, 7–8**
- Kernighan, Brian (C Programming Language), 154**
- Key Value Coding (Cocoa Bindings), 247, 304**
- Key Value Observing (Cocoa Bindings), 304**
- key view loop, 284–287**
- keyboard events, 368–370**
- keyboard navigation of user interfaces (UIs), 287**
- keyed archivers, 358–359**

- keyed archiving, 359**
- keys (Info.plist), 130–131**
- keywords**
 - def, 441
 - enum, 194
 - my, 449
 - NULL, 185
 - typedef, 194
 - use, 448
 - var, 457
 - void, 180
- kill command, 506–507**
- killall command, 507**
- killing processes, 506–507**

L

- labels (windows), 113**
- Language Project directories, 131**
- layers, 3–4**
- lazy initialization, 330**
- ld command-line tool, 670**
- leaks command-line tool, 670**
- linked files (Xcode), 78**
- linked lists, 197–198**
- load script command, 558**
- loading AppleScripts, 558**
- local variables, 193**
- localizations, 127**
- localized resources (bundles), 131–132**
- locate command, 499**
- logical operators in C, 161**
- login shell, 473**
- loops**
 - AppleScript scripting language, 541–543
 - Bash shell scripts, 518–520
 - C programming language, 173–177
 - Python scripting language, 443–444
- lproj directories, 131**
- ls command**
 - directories, 9–10, 478–479, 487–488
 - file permissions, 492
 - file system, 487–488, 496
 - man page, 479

M

- Mac OS X**
 - core OS, 4
 - Darwin, 6–7
 - layers, 3–4
 - release date, 3
 - versions, 3
- Mac OS X Technology Overview, 41–42**
- MacApp application framework, 373**
- Macintosh File Manager API, 392**

Macintosh Toolbox, 373

MacPython website, 445

MacScripter.net website, 660

MacTech Magazine website, 661

Mail application, 132–136

mailing lists, 53, 660

main function, 154–155

make command-line tool, 670

MallocDebug, 23–24, 665

man command, 10–11, 479, 621

man command-line tool, 670

man pages, 47–48, 625

mdfind command, 498

memory leaks

leaks command-line tool, 670

MallocDebug, 23–24, 665

memory management

autoreleasing, 24

C, 196–200

Carbon, 375–376

Objective-C, 250–253

protected memory, 8

reference counting, 24–25

menus

alternate menu items, 103

Carbon menus, 98–103

Cocoa menus

connecting, 322–325

creating, 103–106

menu commands, 326–327

dynamic menu items, 103

empty menus, 102

icons, 102

pre-built menus, 102

separators, 102

submenus, 102

messaging

AppleScript scripting language, 530–531

Objective-C programming language, 241–242

metadata, 498

methods

Objective-C

accessor methods, 237–238, 253–256

autorelease, 251–253

class methods, 232–233

dealloc, 240, 249–250, 256–257

declarations, 229, 231–233

init, 240

initializers, 247–249

instance methods, 232

invoking, 232–233

names, 231

overloading, 240

overriding, 240

parameters, 231

release, 240, 257

return values, 232

selectors, 242

object-oriented programming (OOP), 228

Python, 441–443

QuickTime

exportMovie:, 419–420

selectMovieURL:, 419

Metrowerks PowerPlant application framework, 373

mkdir command, 491, 496

model (defined), 305

Model-View-Controller (MVC), 221, 305–306

modules (Perl)

availability of, 446

File::Find, 448

strict, 448

MONster, 665–666

mount_afp command, 508

Mozilla's Gecko DOM Reference website, 465

multiple inheritance, 238–239

mutable arrays, 330–333

mv command, 488–490, 496

MVC (Model-View-Controller), 221, 305–306

MyAddressBook program (C version)

AddressBook ADT, 208–218

AddressBook.c file, 210–213

AddressBook.h file, 210

Controller.c file, 218–224

Controller.h file, 218

Person.c file, 207–208

Person.h file, 207–208

MyAddressBook program (Objective-C version)

AddressBook class, 260–266

Command subclasses, 275–277

Commands class, 266, 268–275

Controller class, 266–268, 271, 274–275

IOUtility.h file, 257–258

IOUtility.m file, 257–259

Person class, 259–260

Person.h file, 257

Person.m file, 257–258

MyDocument files, 318

N

naming conventions in Objective-C, 246–247

naming projects (Xcode), 59

Nano text editor, 471–472

Netscape JavaScript Guide website, 465

netstat command, 509

new line characters, 157

NewHandle function, 375–376

NewPtr function, 375

NewPtrClear function, 376

NewTrackMedia function, 411

NeXTSTEP technology, 229

nib files, 93, 310–311

nice **command**, 509

notifications

receiving, 340–343
sending, 340–343

NSApplication **class**, 311

NSAutoreleasePool **class**, 251–253, 296

NSBox **class**, 284

NSButton **class**, 311

NSCoder **class**, 358–359

NSControl **class**, 295–296, 320

NSData **class**, 314

NSDocument **class**, 306, 308, 311–315

NSDrawer **class**, 345

NSImage **class**, 314

NSImageView **class**, 304, 309–310, 317–318, 320–322

NSLayoutManager **class**, 305–306

NSMainNibFile Info.plist **key**, 130

NSMutableArray **class**, 330–333

NSNotificationCenter **class**, 342–343

NSObject **class**, 249, 288

NSObjectController **class**, 297–298

NSOpenPanel **class**, 328–330, 334–336

NSPrincipalClass Info.plist **key**, 130

NSResponder **class**, 310

NSScrollView **class**, 307, 310

NSString **class**, 295

NSTableView **class**, 346–349

NSTask **class**, 604–607

NSTextField **class**, 289–290, 294–295

NSTextStorage **class**, 305

NSTextView **class**, 284

NSUserDefaultsController **class**, 298

NSWindowController **class**, 306, 308, 315–317

NULL **keyword**, 185

O

object code, 77

object files, 77

ObjectAlloc performance tool, 23–25, 666

Objective-C programming language

accessor methods, 253–257
archiving objects, 356–359
C programming language, 227
classes, 229–231
Cocoa application framework, 16
data hiding, 237–238
delegation, 245–246
designated initializers, 248–249
exception handling, 235
Grepper program, 233–237
informal protocols, 244–246
inheritance, 238–241
initializers, 247–249
memory management, 250–253
messaging, 241–242

methods

accessor methods, 237–238, 253–256
autorelease, 251–253
class methods, 232–233
dealloc, 240, 249–250, 256–257
declarations, 229, 231–232
init, 240
initializers, 247–249
instance methods, 232
invoking, 232–233
names, 231
overloading, 240
overriding, 240
parameters, 231
release, 240, 257
return values, 232
selectors, 242

multiple inheritance, 238–239

MyAddressBook program

AddressBook class, 260–266
Command subclasses, 275–277
Commands class, 266, 268–275
Controller class, 266–268, 271, 274–275
IOUtility.h file, 257–258
IOUtility.m file, 257–259
Person class, 259–260
Person.h file, 257
Person.m file, 257–258

naming conventions, 246–247

object-oriented programming (OOP), 227–228

objects

creating, 247–249
deleting, 249–250
protocols, 242–246
selectors, 242
static typing extensions, 243
subclassing, 238–241
type checking, 243
variables, 229

object-oriented programming (OOP)

AppleScript scripting language, 531
classes, 228
encapsulation, 228, 237
inheritance, 228, 238–239
methods, 228
multiple inheritance, 238–239
Objective-C programming language, 227–228
objects, 228
polymorphism, 228
variables, 228

objects

AppleScript scripting language, 555–558
archivers, 358
archiving, 356–359
delegate objects, 325–326
keyed archivers, 358–359

objects (continued)

- Objective-C
 - creating, 247–249
 - deleting, 249–250
- off-by-one bug, 177**
- Omni Group Developer website, 660**
- Omni mailing lists, 660**
- online help system, 9–10**
- Oblick guide to Bourne Shell Programming website, 432**
- OOP (object-oriented programming)**
 - AppleScript scripting language, 531
 - classes, 228
 - encapsulation, 228, 237
 - inheritance, 228, 238–239
 - methods, 228
 - multiple inheritance, 238–239
 - Objective-C programming language, 227–228
 - objects, 228
 - polymorphism, 228
 - variables, 228
- opaque directories, 128**
- open command, 508**
- open panels, 328–330, 334–336**
- Open Scripting Architecture (OSA), 531**
- OpenGL Driver Monitor, 664**
- OpenGL graphic library, 12, 14**
- OpenGL Profiler, 664**
- OpenGL Shader Builder, 664**
- operators**
 - AppleScript scripting language, 549–552
 - C programming language, 159–162
- organizing**
 - C programs, 205–206
 - Xcode projects, 61–64
- OSA (Open Scripting Architecture), 531**
- osascript command, 559**
- outlets (Interface Builder), 281, 292–293**
- overloading methods, 240**
- overriding methods, 240**

P

- PackageMaker, 30–33, 668**
- packages (applications), 127**
- PacketLogger, 668**
- Palettes window (Interface Builder), 97, 281**
- parameter list (C programming language), 155**
- pass-by-value (C functions), 182**
- paths, 6–7, 476–478**
- PDF (Portable Document Format), 14**
- performance statistics, 39**
- performance tools**
 - Activity Monitor, 665
 - BigTop, 665
 - MallocDebug, 23–24, 665
 - MONster, 665–666
 - ObjectAlloc, 23–25, 666
 - Quartz Debug, 23, 25–26, 666
 - Sampler, 23, 26–27, 666
 - Saturn, 666
 - Shark, 23, 27–28, 666
 - Spin Control, 23, 28, 667
 - Thread Viewer, 23, 28–30, 667
- period symbol (.), 477**
- Perl**
 - arrays, 451
 - CamelBones framework, 447, 451
 - example script, 447–451
 - features, 446–447
 - functions, 449–451
 - modules
 - availability of, 446
 - File::Find, 448
 - strict, 448
 - my keyword, 449
 - perl command, 448
 - regular expressions, 449
 - shebang, 448
 - use keyword, 448
 - variable declaration, 448–449
- Perl Directory website, 451**
- permissions**
 - Access Control Lists (ACL), 491
 - changing, 493–494
 - viewing, 491–492
- persistence, 215**
- PHP scripting language, 466**
- PHP website, 466**
- Piles of Pictures**
 - script, 578–586
 - user interface (UI), 575–578
- pipes, 481–483**
- Pixie, 30, 33, 664**
- plug-ins, 127–128**
- pmset command, 508**
- pointers**
 - C programming language, 164–169
 - Carbon, 375–376, 385
- polymorphism**
 - AppleScript scripting language, 557
 - defined, 228
 - Python, 445
- popen function, 594**
- Portable Document Format (PDF), 14**
- ports (QuickTime), 402–404**
- pow function, 626**
- PowerPlant application framework, 373**
- pre-built menus, 102**
- precedence of operators, 161–162**
- pre-emptive multi-tasking, 8**

preferences

- application preferences
 - built-in preferences system, 149–150
 - Cocoa, 359–367
 - editing with Property List Editor, 150
 - file formats, 149–150
 - restoring, 144
 - storing, 148–149
 - viewing, 144–148
- built-in preferences system, 149
- property lists, 149–150
- storing preferences files, 148–150

preprocessor directives, 154, 205–206**preprocessors, 200–205****printf function, 155, 180, 187, 626****processes**

- background processes, 505
- Bash shell, 480
- checking status, 505–506
- daemons, 505
- defined, 7
- killing, 506–507
- status values, 505

profilers, 23**programs**

- C programs
 - compiling, 155–156
 - running, 156
- Calculator, 65–69
- defined, 7
- device drivers, 8
- functions, 23
- Grepper
 - C version, 191–194
 - Objective-C version, 233–237
- stack, 23

projects (Xcode)

- building, 72–77
- creating, 58–60
- naming, 59
- organizing, 61–64
- saving, 59
- source files, 60–64
- targets, 78

Property List Editor, 30, 33–34, 455, 668**property lists**

- application preferences, 149–150
- editing, 455
- Info.plist file, 130–131

protected memory, 8**protocols (Objective-C), 242–246****ps command, 505–507****pwd command, 486****PyObjC website, 446****Python**

- bindings, 437
- comments, 440–441

cross-platform functionality, 437

def keyword, 441

documentation, 445

example script, 438–445

features, 436–438, 445

if statements, 445

for loops, 443–444

methods, 441–443

performance, 437

polymorphism, 445

PyObjC framework, 437–438

Python framework, 437

shebang, 440

tutorials, 445–446

Python website, 436, 445**pythonmac website, 445****Q****Quartz 2D graphic library, 12, 14****Quartz Composer, 664****Quartz Compositor, 12–13****Quartz Debug, 23, 25–26, 666****QuickDraw graphics environment, 400****QuickTime**

Cocoa, 400

codecs, 408

creating movies, 404–409

errors, 420

features, 12, 15, 400

functions

AddMediaSample, 416

AddMovieToStorage, 409

BuildImageVideoSamples, 415–416

BuildImageVideoTrack, 411

CompressImage, 416

CreateMovieStorage, 408–409

DrawSlideInGWorld, 420

EnterMovies, 419

GetMaxCompressionSize, 415

InsertMediaIntoTrack, 411–412

NewTrackMedia, 411

SaveImageMovie, 408, 419–420

GraphicsImporter API, 403

GWorld port, 403–404

methods

exportMovie:, 419–420

selectMovieURL:, 419

off-screen drawing, 403

ports, 402–404

UIKit library, 400

QuickDraw graphics environment, 400

video tracks, 409–416

quitting shells, 39**quoting in Bash shell scripts, 513–515**

R

- receiving notifications, 340–343**
- recordable applications, 530**
- recursion (C programming language), 181**
- redo support in Cocoa, 354–355**
- redrawing user interfaces, 339**
- reference counting, 24–25**
- Reference Library, 39–42**
- regular expressions**
 - grep command, 502
 - Perl, 449
- relative path, 7, 477**
- release date of Mac OS X, 3**
- release method, 240, 257**
- Release Notes window (Interface Builder), 94**
- renice command, 509**
- Repeat After Me, 668**
- repeat loops in AppleScript, 541–543**
- replacing MyDocument files, 318–319**
- resizable windows**
 - Carbon, 120–123
 - Cocoa, 113–120
- resource forks, 37, 490**
- responder chain, 324–325**
- restoring application preferences, 144**
- return values in C functions, 180**
- Rez command-line tool, 35, 37–38, 670**
- Rez files, 37–38**
- Ritchie, Dennis (C Programming Language), 154**
- rm command, 491**
- root directory, 7**
- Ruby scripting language, 466**
- Ruby website, 466**
- RubyCocoa framework, 466**
- Run Log window (Xcode), 60**
- run script commands, 558**
- running**
 - AppleScripts, 530, 558–559
 - Bash shell scripts, 510
 - C programs, 156
 - commands, 480
 - executables, 480
 - scripts asynchronously, 594
 - shells, 39
 - Slide Master application, 4
 - subprocesses with C functions, 589

S

- sample code, 51–52**
- sample command-line tool, 35, 38, 671**
- Sampler, 23, 26–27, 621–623, 666**
- Saturn, 666**
- SaveImageMovie function, 408, 419–420**

- saving**
 - AppleScripts, 530
 - stock quotes, 546–549
 - Xcode projects, 59
- /sbin directory, 479**
- scanf function, 187–188, 626**
- scope (variables), 193**
- Script Editor (AppleScript), 30, 34–35, 528–529**
- scriptable applications, 530**
- scripting languages**
 - AppleScript
 - Apple Events, 530–531
 - classes, 558
 - conditional branching, 540–541
 - cross-platform functionality, 432
 - data types, 531–537
 - documentation, 436
 - element specifiers, 553–555
 - example script, 432–436
 - exception handling, 544–545
 - features, 432
 - handlers, 537–540, 557–558
 - inheritance, 556–557
 - interapplication communication, 530
 - loops, 541–543
 - messaging, 530–531
 - object-oriented programming (OOP), 531
 - operators, 549–552
 - polymorphism, 557
 - properties, 555
 - recordable applications, 530
 - Script Editor, 30, 34–35, 528–530
 - script objects, 555–558
 - scriptable applications, 530
 - Stock Quotes example, 546–549
 - tell statement, 543–544
 - variables, 531–535
 - Bash, 427–432
 - defined, 426–427
 - JavaScript
 - Dashboard, 452–465
 - features, 451
 - Java virtual machine (JVM), 451–452
 - var keyword, 457
 - variable declarations, 457
 - Perl
 - arrays, 451
 - CamelBones framework, 447, 451
 - example script, 447–451
 - features, 446–447
 - functions, 449–451
 - modules, 446, 448
 - my keyword, 449
 - perl command, 448
 - regular expressions, 449
 - shebang, 448

- use keyword, 448
- variable declaration, 448–449
- PHP, 466
- Python
 - bindings, 437
 - comments, 440–441
 - cross-platform functionality, 437
 - def keyword, 441
 - documentation, 445
 - example script, 438–445
 - features, 436–438, 445
 - if statements, 445
 - for loops, 443–444
 - methods, 441–443
 - performance, 437
 - polymorphism, 445
 - PyObjC framework, 437–438
 - Python framework, 437
 - shebang, 440
 - tutorials, 445–446
- Ruby, 466
- support for, 425
- Tcl, 466
- scripts**
 - AppleScripts
 - creating, 528–530
 - example, 432–436
 - loading, 558
 - Piles of Pictures, 578–586
 - running, 530, 558–559
 - saving, 530
 - Slideshow Maker, 559–567
 - storing, 558
 - Bash shell
 - commands, 509–510
 - conditional branching, 515–518
 - file compression script, 521–524
 - functions, 520–521
 - loops, 518–520
 - quoting, 513–515
 - running, 470, 510
 - shebang, 510
 - variables, 511–513
 - defined, 425–426
 - running asynchronously, 594
- Searcher application**
 - Cocoa interface, 595–601
 - search engine, 601–603
 - Searcher class, 600
- sed **command, 503–505**
- selectMovieURL: **method, 419**
- selectors, 242**
- sending notifications, 340–343**
- separators (menus), 102**
- SetFile **command-line tool, 35, 37, 623–625, 671**
- sftp **command, 509**
- sh (**shell**), **35, 38**
- Shader Builder, 664**
- Shark, 23, 27–28, 666**
- shebang**
 - Bash shell scripts, 510
 - Perl, 448
 - Python, 440
- shells**
 - Bash shell
 - AppleScripts, 559
 - autocompletion feature, 495
 - conditional branching, 515–518
 - configuring, 473–476
 - defined, 35, 38
 - error codes, 480
 - executables, 480
 - file compression script, 521–524
 - file permissions, 491–494
 - functions, 520–521
 - globbing, 494–495
 - input/output (I/O), 481–483
 - interactive use, 470
 - login shell, 473
 - loops, 518–520
 - pipes, 481–483
 - processes, 480
 - quoting, 513–515
 - script example, 427–432
 - scripts, 470, 509–513
 - Terminal application, 470–471
 - text editors, 471–472
 - X11 application, 471
 - case-sensitivity, 11
 - changing, 39
 - quitting, 39
 - running, 39
 - sh (shell), 35, 38
 - tcsh, 35, 38–39
 - Terminal application, 11
- signatures (C programming language), 181**
- SimpleText, 46–47**
- Sketch, 46–47**
- sleep **command, 506–507**
- Slide Master application**
 - application delegate, 325–326
 - archiving objects, 356–359
 - bundle, 137–144
 - downloading, 4
 - drawers, 343–345
 - how it works, 5–6
 - keyboard events, 368–370
 - menu bar, 322–325
 - mutable arrays, 330–333
 - MyDocument files, 318–319
 - notifications, 340–343
 - NSDocument class, 307, 311–315

Slide Master application (continued)

- NSImageView class, 309–310, 317–318, 320–322
- NSScrollView class, 309–310
- NSWindowController class, 307, 315–317
- open panels, 328–330, 334–336
- preferences panel, 360–367
- QuickTime movies, 401–420
- redo feature, 354–355
- running, 4
- scrollable image view, 307–310
- SlideShowDocument class, 336–339
- table data sources, 349–353
- table views, 346–349
- undo feature, 354–355

smart groups (Xcode), 65

source code

- Darwin, 6
- SimpleText, 46–47
- Sketch, 46–47
- TextEdit, 46–47
- Xcode, 57

source command, 481

source files

- C programs, 205
- file extensions, 60
- Xcode projects, 60–64

speech recognition, 668

Spin Control, 23, 28, 667

spinning wait cursor, 22

Spotlight search technology, 497–499

sprintf function, 188–189

SRLanguageModeler, 668

sscanf function, 188–189

ssh command, 509

stack, 23

standard error (Unix commands), 483

states of drawers, 345

statically typed programming languages, 243

statistics on system performance, 39

status values of processes, 505

Stepwise website, 660

stock quotes

- downloading, 546–549
- saving, 546–549

store script command, 558

storing

- aliases, 394–400
- AppleScripts, 558
- application preferences, 148–149
- preferences files, 148–150

strcat function, 185

strcmp function, 185

strcpy function, 184

strict module (Perl), 448

strings

- C programming language, 183–186
- Carbon, 386–388

strlen function, 185

strncat function, 185

strncmp function, 185

strncpy function, 184

strstr function, 185

structured programming, 208

subclassing (Objective-C), 238–241

submenus, 102

subprocesses

- NSTask class (Cocoa), 605–607
- running with C functions, 589
- versus threads, 590

sudo command, 493–494, 496–497

switch/case statements, 171–172

symbolic links

- bundles, 129
- defined, 129

system function, 590–594

system library, 7–9

system performance statistics, 39

system_profiler command, 484–486, 508

SystemStarter command, 508

T

- table data sources, 349–353
- table views, 346–349
- tar command, 509
- targets (Xcode projects), 78
- Tcl Developer Xchange website, 466
- Tcl scripting language, 466
- tcsh, 35, 38–39
- tell statement (AppleScript), 543–544
- temporary files, 594
- Terminal application
 - Bash shell, 470–471
 - configuring, 471
 - features, 6–7, 11, 669
- terminating null character, 183
- ternary operator, 172–173
- testing user interface (UI), 95–97
- text editors, 471–472
- text labels (windows), 113
- TextEdit, 46–47
- Thompson, Ken, creator of the C programming language, 153
- Thread Viewer, 23, 28–30, 667
- threads versus subprocesses, 590
- Toll-Free Bridging, 374, 390, 392
- Toolbox API, 373
- tools
 - command-line tools
 - cc, 35

DeRez, 35, 37–38, 669
 diff, 35–36, 669
 gcc, 35–36, 669
 gdb, 35–37, 669
 GetFileInfo, 35, 37, 670
 ld, 670
 leaks, 670
 make, 670
 man, 670
 Rez, 35, 37–38, 670
 sample, 35, 38, 671
 SetFile, 35, 37, 623–625, 671
 top, 35, 39, 671
 developer tools
 Activity Monitor, 665
 Automator, 663
 BigTop, 665
 Bluetooth Explorer, 667
 BSD installer package, 21–22
 CHUD Tools, 27
 Core Image Fun House, 664
 DeRez, 35, 37–38, 669
 FileMerge, 30–31, 667
 icns Browser, 667
 Icon Composer, 30–31, 667
 Interface Builder, 22, 663
 IORegistryExplorer, 667–668
 Jar Bundler, 665
 JavaBrowser, 665
 MallocDebug, 23–24, 665
 MONster, 665–666
 ObjectAlloc, 23–25, 666
 OpenGL Driver Monitor, 664
 OpenGL Profiler, 664
 OpenGL Shader Builder, 664
 PackageManager, 30–33, 668
 PacketLogger, 668
 Pixie, 30, 33, 664
 profilers, 23
 Property List Editor, 30, 33–34, 455, 668
 Quartz Composer, 664
 Quartz Debug, 23, 25–26, 666
 Repeat After Me, 668
 Sampler, 23, 26–27, 621–623, 666
 Saturn, 666
 Script Editor, 30, 34–35
 Shark, 23, 27–28, 666
 Spin Control, 23, 28, 667
 SRLanguageModeler, 668
 Terminal, 669
 Thread Viewer, 23, 28–30, 667
 USB Prober, 669
 Xcode, 22, 663
 top **command-line tool, 35, 39, 671**
 touch **command, 501**

type checking (Objective-C), 243

type code (files), 37

typedef **keyword, 194**

U

UI (user interface)

Apple Human Interface Guidelines, 15, 18–19, 41–42
 applications, 15
 Aqua user interface, 15
 Carbon, 374
 Cocoa, 280–281
 creating, 93–97
 Crumbs application, 607–611
 Email Formatter, 282–284, 378–382
 keyboard navigation, 287
 nib files, 93
 Piles of Pictures, 575–578
 redrawing, 339
 Slideshow Maker, 559–567
 testing, 95–97

uncompressing files, 4

undo support (Cocoa), 354–355

Unix

/bin directory, 478
 features, 6
 paths, 476–478
 /sbin directory, 479
 /usr/bin directory, 479

Unix commands

apropos, 621
 arguments, 10
 awk, 502–503
 background commands, 480
 bash, 510
 cat, 501
 cd, 486–488, 496
 chmod, 491, 493, 496
 chown, 493
 chsh, 39
 cp, 490, 496
 crontab, 508
 curl, 508
 dig, 508
 diskutil, 508
 ditto, 508
 echo, 11, 486, 500
 eval, 481
 exec, 481
 exit, 480–481
 find, 136, 480, 499–500
 flags, 10
 foreground commands, 480
 ftp, 509
 grep, 501–502

Unix commands (continued)

gunzip, 509
gzip, 509
hdiutil, 508
installer, 508
kill, 506–507
killall, 507
locate, 499
ls
 directories, 9–10, 478–479, 487–488
 file permissions, 492
 file system, 487–488, 496
 man page, 479
man, 10–11, 479, 621
mdfind, 498
mkdir, 491, 496
mount_afp, 508
mv, 488–490, 496
netstat, 509
nice, 509
open, 508
pipes, 481–483
pmsset, 508
ps, 505–507
pwd, 486
renice, 509
rm, 491
running, 480
sed, 503–505
sftp, 509
sleep, 506–507
source, 481
ssh, 509
standard error, 483
sudo, 493–494, 496–497
system_profiler, 484–486, 508
SystemStarter, 508
tar, 509
touch, 501
unzip, 509
wc, 503
which, 479
zip, 509

Unix Manual, 10

Unix shells

Bash shell
 AppleScripts, 559
 autocompletion feature, 495
 conditional branching, 515–518
 configuring, 473–476
 defined, 35, 38
 error codes, 480
 executables, 480
 file compression script, 521–524
 file permissions, 491–494

 functions, 520–521
 globbing, 494–495
 input/output (I/O), 481–483
 interactive use, 470
 login shell, 473
 loops, 518–520
 pipes, 481–483
 processes, 480
 quoting, 513–515
 script example, 427–432
 scripts, 470, 509–513
 Terminal application, 470–471
 text editors, 471–472
 X11 application, 471
case-sensitivity, 11
changing, 39
quitting, 39
running, 39
sh (shell), 35, 38
tcsh, 35, 38–39
 Terminal application, 11
unzip **command**, 509
USB Prober, 669
use **keyword (Perl)**, 448
user interface (UI)
 Apple Human Interface Guidelines, 15, 18–19, 41–42
 applications, 15
 Aqua user interface, 15
 Carbon, 374
 Cocoa, 280–281
 creating, 93–97
 Crumbs application, 607–611
 Email Formatter, 282–284, 378–382
 keyboard navigation, 287
 nib files, 93
 Piles of Pictures, 575–578
 redrawing, 339
 Slideshow Maker, 559–567
 testing, 95–97
/usr/bin **directory**, 479

V

value transformers (Cocoa Bindings), 361

var **keyword**, 457

variable declarations

 C programming language, 158–159
 JavaScript, 457
 Perl, 448–449

variables

 AppleScript scripting language, 531–535
 Bash shell scripts, 510–513
 C programming language, 157–159
 global variables, 192–193
 local variables, 193

Objective-C programming language, 229
 object-oriented programming (OOP), 228
 scope, 193

versioned bundles, 128–129

versions of Mac OS X, 3

vi text editor, 471

video tracks (QuickTime), 409–416

viewing

application preferences, 144–148
 file permissions, 491–492

views (defined), 305

void keyword, 180

W

wc command, 503

websites

Apple, 436
 Apple Developer Connection (ADC), 21, 48–53, 659
 Bash Reference Manual, 432
 Cocoa Dev Central, 659
 CocoaDev, 659
 Comprehensive Perl Archive Network (CPAN), 446, 451
 DarwinPorts project, 466
 MacPython, 445
 MacScripter.net, 660
 MacTech Magazine website, 661
 Mozilla's Gecko DOM Reference, 465
 Netscape JavaScript Guide, 465
 Omni Group Developer, 660
 Ooblick guide to Bourne Shell Programming, 432
 Perl Directory, 451
 PHP, 466
 PyObjC, 446
 Python, 436, 445
 pythonmac, 445
 Ruby, 466
 Stepwise, 660
 Tcl Developer Xchange, 466
 Wrox, 4

which command, 479

while loops in C, 174–175

widget bundles (Dashboard), 452

widgets (Dashboard)

bundles, 452
 DashBall, 453–465
 defined, 452

windows

content rects, 317
 controls, 113

creating, 106–112
 dialog windows, 106–112
 document windows, 106
 frame rects, 317
 labels, 113
 resizable windows
 Carbon, 120–123
 Cocoa, 113–120
 text labels, 113

wrappers (applications), 127

Wrox website, 4

X

X11 application, 470–471

Xcode

AppleScript Studio, 570–572
 application bundles, 136–144
 build results, 78
 build styles, 78
 Calculator program, 65–69
 Code Editor, 65–70
 compiler, 77
 debugger, 78–84
 documentation, 57, 84–89
 errors, 78
 features, 22, 663
 groups, 64–65
 indenting code, 70–72
 linked files, 78
 object code, 77
 object files, 77
 projects
 building, 72–77
 creating, 58–60
 naming, 59
 organizing, 61–64
 saving, 59
 source files, 60–64
 targets, 78
 replacing MyDocument files, 318–319
 Run Log window, 60
 smart groups, 65
 source code, 57

Xcode 2.0 User Guide, 41–42

Z

zero (0) error code, 60

zip command, 509

