

The  
Pragmatic  
Programmers

Android  
1.5 to 2.1

# Hello, Android

Introducing Google's Mobile  
Development Platform

Third Edition

Edited by Susannah Davidson Pfälzer

*Ed Burnette*

ANDROID





























































































Download from Library of Wow! eBook  
<[www.wowebook.com](http://www.wowebook.com)>



























































































































































































































































































































































































































































































































































## Appendix B

# Bibliography

---

- [Bur05] Ed Burnette. *Eclipse IDE Pocket Guide*. O'Reilly & Associates, Inc, Sebastopol, CA, 2005.
- [Gen06] Jonathan Gennick. *SQL Pocket Guide*. O'Reilly Media, Inc., Sebastopol, CA, second edition, 2006.
- [Goe06] Brian Goetz. *Java Concurrency in Practice*. Addison-Wesley, Reading, MA, 2006.
- [Owe06] Mike Owens. *The Definitive Guide to SQLite*. Apress, Berkeley, CA, 2006.

# Index

## A

- About box, [57–62](#)
- Accelerometer readings, [170](#)
- ACCESS\_COARSE\_LOCATION permission, [41](#), [162](#)
- ACCESS\_FINE\_LOCATION permission, [41](#), [162](#)
- ACTION\_DOWN event, [227](#)
- ACTION\_MOVE event, [227](#)
- ACTION\_POINTER\_DOWN event, [227](#)
- ACTION\_POINTER\_UP event, [227](#)
- ACTION\_UP event, [228](#)
- ACTION\_VIEW action, [135](#)
- Activity, [35](#), [37](#), [39](#)
  - declaring, [60](#)
  - defining, new, [58](#)
- Activity class, [35](#), [126](#)
- Activity Manager, [33](#)
- Adapter class, [153](#)
- addEvent() method, [186](#), [194](#)
- addJavaScriptInterface() method, [139–147](#)
- addPreferencesFromResource() method, [66](#)
- ADT (Android Development Toolkit), [20](#)
- AlarmManager class, [241](#)
- alert() method, [143](#), [145](#)
- AlertDialog class, [57](#)
- Alpha values, [74](#)
- ALTER TABLE statements, [183](#)
- Ambient lighting, [209](#)
- Android
  - activity in, [36](#), [37](#)
  - architecture of, [30–35](#)
  - audio formats supported by, [111](#)
  - benefits of, [130](#)
  - content providers, [193](#)
  - installation, [20](#), [21](#)
  - Java library support, [280](#)
  - language, vs. Java, [278](#)
  - libraries, [31–32](#)
  - objects, [39–40](#)
  - OpenGL and, [199](#)
  - project, basic, [24](#)
  - resources, [40](#)
  - screen rotations, [116](#)
  - sensors, support for, [169](#)
  - third-party libraries for, [281](#)
  - threads in, [279](#)
  - user ID, [127](#)
  - video formats supported by, [112](#)
  - windows in, [35–38](#)
- Android 1.5 (Cupcake), [13](#), [26](#), [32](#), [110](#), [112](#), [169](#), [175](#), [202](#), [233](#)
- Android 1.6 (Donut), [13](#), [128](#)
- Android 2.0 (Eclair), [13](#)
- Android 2.0.1 (Eclair), [13](#)
- Android 2.1 (Eclair MR1), [13](#), [243](#)
- Android 2.2 (FroYo), [13](#), [25](#), [26](#), [69](#), [128](#), [206](#), [221](#), [257](#), [268](#), [269](#)
- Android APIs, [259–265](#)
- Android AVD Error, [177](#)
- android:background attribute, [77](#)
- android:configChanges property, [116](#)
- Android Developer Forums, [266](#)
- Android Development Toolkit (ADT), [20](#)
- Android Eclipse plug-in, [103](#)
- android.graphics package, [73](#)
- android:imeOptions option, [132](#)
- android:inputType option, [132](#)
- android:installLocation attribute, [269](#)
- android:height parameter, [50](#)
- android:layout\_width parameter, [50](#)
- Android Location API, [161–168](#)
  - sensors and, [168–172](#)
- Android Market, publishing to, [35](#), [271–276](#)
  - updating published applications, [275](#)

android.media package, 105  
 Android runtime, 32  
 Android SDK  
     Setup programs, 19  
     Starter Package, 18  
 Android versions  
     building for multiple, 257–259  
     list of, 256  
 Android Virtual Device (AVD), 23, 177  
     creating, 27  
     undefined, 25  
 AndroidManifest.xml file  
     defining widgets, 234  
 AndroidManifest.xml file, 41, 60, 61, 66,  
     71, 80, 114, 138, 149, 162, 174,  
     195, 225, 234, 244, 269  
     multiversion support, 258  
 Animation, OpenGL, 212–213  
 APIs, 259–265  
 .apk files, 139, 272  
 Application Framework, 33  
 Application stack, 35  
 Applications, 34  
 Applications and Widgets layer, 34  
 APPWIDGET\_UPDATE message, 239  
 AppWidgetProvider class, 238–239  
 Architecture, 30–35  
     Application Framework, 33  
     Applications and Widgets layer, 34  
     Linux kernel, 30  
     native libraries, 31  
 ARGB for colors, 73  
 arrays.xml file, 67, 150  
 assets directory, 142, 146  
 Audio, 105–110  
 Audio formats, supported, 111  
 AUTOINCREMENT keyword, 180  
 AVD (Android Virtual Device), 23, 177  
     creating, 27  
     undefined, 25

## B

Background gradient, 78  
 Background images, stretchable, 236  
 Background music, 115–119  
 Background services, 117  
 BaseColumns interface, 182  
 BIND\_WALLPAPER permission, 244  
 Bitmap drawables, 76  
 Bitmaps, 54  
 Blending mode, 217

Bornstein, Dan, 33  
 BrowserIntent class, 131–135  
 BrowserView class, 131, 137  
 Bundle class, 124  
 Button class, 132

## C

calculateUsedTiles() method, 80, 99  
 Callable class, 147  
 callAndroid() method, 143, 145  
 callJS() method, 143, 146  
 Camera class, 177  
 Canvas class, 75  
 Carmack, John, 200  
 Chrome, 145  
 Circular paths, 75  
 .class files, 33  
 ClassNotFoundException class, 174  
 Color class, 73  
 Colors, 73  
 colors.xml file, 52, 74, 83  
 concurrent package package, 147  
 Connection error, 22  
 Constants interface, 182, 184  
 Content providers, 34, 40, 193  
 ContentProvider class, 180, 192–196  
 ContentValues class, 186  
 Context class, 126  
 Context menus, 64  
 create() method, 109, 118  
 CREATE TABLE statements, 181, 183  
 createSnapshot() method, 139  
 Criteria class, 165  
 Cube model, 206–210, 213, 216  
 Cupcake (Android 1.5), 13, 26, 32, 110,  
     112, 169, 175, 202, 233  
 Cursor class, 187

## D

d() method (Log class), 69  
 Dalvik virtual machine (VM), 32  
 Data binding, 192f, 189–192  
 Data definition language (DDL)  
     statements, 180  
 Data storage, 120–128  
     current screen position, 124–126  
     in /data directory, 113, 127  
     internal files, accessing, 126  
     options, adding, 120–122  
     pausing game, 122–124  
     secure digital (SD), 127–128

SQLite, 178–196  
 about, 178–179, 180f  
 basic application, 185f, 181–189  
 ContentProvider, 192–196  
 data binding, 192f, 189–192  
 overview of, 179–181

DDL statements, 180

Debugging, 69–71

Declarative design, 44

*The Definitive Guide to SQLite* (Owens), 196

Delayed requests, 153

Delegation, 260

delegation, 261

deleteFile() method, 126

Density-independent pixels, 54

Depth testing, 217

Developer Forums, 266

Device-specific interfaces, 267

.dex files, 33

Difficulty selection, 68

Diffuse lighting, 209

Digital signatures, 127, 272

Dips (density-independent pixels), 54

Directional lighting, 209

Directory name qualifiers, 268

Donut (Android 1.6), 13, 128

dp (density-independent pixels), 54

Drag gesture, 220, 229, 230

Drawable class, 76, 77

Drawables, 76, 77

Drawing functions, 89

dumpProviders() method, 165

## E

---

e() method (Log class), 69

Eclair (Android 2.0), 13

Eclair (Android 2.0.1), 13

Eclair MR1 (Android 2.1), 13, 243

Eclipse, 18, 20, 29, 40, 45, 71, 168  
 management of R class, 47  
 version of, 25

*Eclipse IDE Pocket Guide* (Burnette), 29

EditText class, 132

Emulator, 23, 257  
 GPS and, 168  
 landscape mode, 55  
 MapView class and, 177  
 secure digital cards, 127  
 sensors and, 170  
 sound formats and, 111

speed and, 28, 218

Emulator skins, 257

enableCompass() method, 176

enableMyLocation() method, 176

Errors  
 activities, declaring, 60  
 connection, 22  
 MediaPlayer and, 110  
 release() method, 119

Events, touch, 225–228

Events class, 185, 194

EventsData class, 182, 196

EventsProvider class, 195

executor objects, 249

ExecutorService class, 147, 155, 160

Exit button, 71

extraData area, 79

## F

---

Field of view, 198

FileInputStream class, 126

fileList() method, 126

FileOutputStream class, 126

Fill rate, limits on, 217

final keyword, 247

Finalization, 279

findViewById() method, 59, 134, 175

findViews() method, 96, 152

finish() method, 71

Fixed-point interface, 209

Flash memory card, 127–128, 268

float type, 231

Floating-point interface, 209

FontMetrics class, 87

Formats, audio, 111

Formats, video, 112

FPS (frames per second), 217

Frame rates, 217

FrameLayout class, 49

FROM parameter, SELECT statement, 187

fromPuzzleString() method, 102

FroYo (Android 2.2), 13, 25, 26, 69, 128, 206, 221, 257, 268, 269

Future class, 147, 156

## G

---

Game class, defining, 79

Game logic, 99

Games, sensors for, 169

Garns, Howard, 44, 79

Gennick, Jonathan, 196

Gesture events, [225–228](#)  
 Gestures, multi-touch, [220](#)  
 GestureWorks site, [232](#)  
 getBestProviders() method, [165](#)  
 getColor() method, [74](#)  
 getColumnIndexOrThrow() method, [188](#)  
 getContentResolver() method, [194](#)  
 getEvents() method, [186, 187, 194](#)  
 getExternalFilesDir() method, [128](#)  
 getHeight() method, [81](#)  
 getHints() method, [122](#)  
 getIntExtra() method, [124](#)  
 getLang() method, [157](#)  
 getLastNonConfigurationInstance() method, [116](#)  
 getLong() method, [188](#)  
 getMenuInflater() method, [65](#)  
 getPointerCount() method, [228, 260](#)  
 getPointerId() method, [228](#)  
 getPuzzle() method, [101, 123, 124](#)  
 getReadableDatabase() method, [187](#)  
 getRect() method, [89](#)  
 getResources() method, [74](#)  
 getSettings() method, [139](#)  
 getString() method, [188](#)  
 getSystemService() method, [164, 169](#)  
 getTitle() method, [102](#)  
 getTitleString() method, [86, 102](#)  
 getUsedTiles() method, [99](#)  
 getWidth() method, [81](#)  
 getWritableDatabase() method, [186, 194](#)  
 GL\_BLEND option, [204](#)  
 GL\_CULL\_FACE option, [204](#)  
 GL\_DEPTH\_TEST option, [204](#)  
 GL\_DITHER option, [204](#)  
 GL\_LIGHT option, [204](#)  
 GL\_LIGHTING option, [204](#)  
 GL\_LINE\_SMOOTH option, [204](#)  
 GL\_MULTISAMPLE option, [204](#)  
 GL\_POINT\_SMOOTH option, [204](#)  
 GL\_TEXTURE\_2D option, [204](#)  
 glColor4f() method, [209](#)  
 glColor4x() method, [209](#)  
 glDisable() method, [204](#)  
 glEnable() method, [204](#)  
 Global Positioning System (GPS), [161–168](#)  
 GLRenderer class, [203, 208](#)  
 GLSurfaceView class, [202, 205](#)  
 gluPerspective() method, [204](#)  
 GLView class, [201](#)

Goetz, Brian, [160](#)  
 Google Maps, [172–177, 273](#)  
 Google Translation API, [147](#)  
 Google Web Toolkit (GWT), [278](#)  
 Gradient background, [78](#)  
 Gradient drawables, [77](#)  
 Graphics, [73–104, 198–217](#)  
   animation, [212–213](#)  
   canvases, [75](#)  
   color, [73](#)  
   cube model, [206–210](#)  
   drawable graphics, [76, 77](#)  
   gradient background, [78](#)  
   Hello, Android and, [200–202](#)  
   input, [87–93](#)  
   library, [87](#)  
   lighting, [209–212](#)  
   NinePatch, [77, 236–238](#)  
   OpenGL and, [199–200](#)  
   painting, [74](#)  
   path, circular, [76](#)  
   paths, [75](#)  
   smoothness of, measuring, [217](#)  
   Sudoku example, [78–87](#)  
   texture, [212–216](#)  
   threads, [202–206](#)  
   three-dimensional, [198–199](#)  
   transparency, [216](#)  
 GraphicsView class, [75](#)  
 Grid lines, Sudoku, [85](#)  
 GROUP\_BY clause, [187](#)  
 guiSetText() method, [157](#)

## H

H.263 format, [112n](#)  
 Handler class, [144, 154](#)  
 HAVING clause, [187](#)  
 Hello, Android  
   audio and, [106](#)  
   browser view and, [136](#)  
   browsing by intent, [131](#)  
   Emulator, running in, [26](#)  
   first project, [23, 24](#)  
   JavaScript and, [141](#)  
   location test, [162](#)  
   MapView class, [172](#)  
   Multi-touch and, [222](#)  
   OpenGL and, [200–202](#)  
   SQLite and, [181](#)  
   translation program, [148](#)  
 Hints, [91, 122](#)

Hipp, Richard, 178  
 Home application, 35, 36  
 Home screen, customizing, 233–254  
   live wallpaper, 242–254  
   widgets, 233–242  
 HTML views, 63  
 HttpURLConnection package, 147, 160

## I

---

i() method (Log class), 69  
 IBinder interface, 192  
 Image transformation (multi-touch), 228–229  
 ImageView class  
   matrix transformation, 228  
   setImageMatrix() bug, 265–267  
 in. (inches) as resolution units, 54  
 index.html file, 142, 146  
 initMapView() method, 175  
 initMyLocation() method, 175  
 initThreading() method, 154  
 Inner class, 60  
 Input, 87–93  
 INSERT statements (SQL), 186  
 insertOrThrow() method, 186, 194  
 Install directory, 19  
 Installation  
   Android, 19–21  
   Eclipse, 18  
   Eclipse plug-in, 20  
   Java 5.0+, 17  
   Web Standard Tools, 21  
 Intent class, 60, 135  
 intent-filter tag, 244  
 Intents, 39, 60  
 Inter-Process Communication (IPC), 192  
 Internal files, accessing, 126  
 Internet  
   browsing by intent, 131–135  
   JavaScript and, 140–147  
   web services and, 147–160  
   web view, 135–139  
 INTERNET permission, 41, 138, 149  
 invalidate() method, 89  
 isValid() method, 98

## J

---

.jar files, 33  
 jarsigner utility, 272  
 Java, 17

constants (as interface), 184  
 Dalvik and, 33  
 intrinsic language types, 279  
 JavaScript and, 140  
 language subset and, 278  
 libraries, support for, 280  
 naming packages, 271  
 reflection, support for, 279  
 time slicing, multiple threads, 279  
 web services, 147–160  
 java.io package, 126  
 java.net.HttpURLConnection package, 147  
 java.util.concurrent package package, 147  
*Java Concurrency in Practice* (Goetz), 160  
 JavaScript, 140–147

## K

---

keyboardHidden value,  
   android:configChanges, 116  
 Keypad class, 93  
 Keypads, 93, 96  
 keytool utility, 272

## L

---

Landscape mode, 55  
 Layer containers, 77  
 Layout, widget, 236  
 Lea, Doug, 147  
 Level containers, 77  
 Libraries, 31–32  
 Lighting, OpenGL, 209–212  
 LinearLayout class, 49, 137, 150  
 Linux kernel, 30  
 ListActivity class, 189  
 ListView class, 189  
 live wallpaper, 242–254  
 loadData() method, 139  
 loadDataWithBaseURL() method, 139  
 loadTexture() method, 215  
 loadUrl() method, 138–140, 146  
 Local data storage, 120–128  
   current screen position, 124–126  
   internal files, accessing, 126  
   options, adding, 120–122  
   pausing game, 122–124  
   secure digital (SD), 127–128  
 LocalBrowser class, 131, 143, 146  
 Locale, 161–168  
   sensors, 168–172  
 Location manager, 34

Location updates, [166](#)  
 LocationListener class, [164](#), [165](#)  
 LocationManager class, [164](#), [165](#)  
 LocationManager class, [163](#)  
 LocationTest class, [164](#)  
 Log class, [69](#)  
 Log messages, [69](#)  
 LogCat view, [70](#)

## M

main.xml file, [46](#), [48](#), [50](#), [52](#), [56](#), [59](#),  
[113](#), [132](#), [136](#), [142](#), [149](#), [163](#),  
[172](#), [184](#), [188](#), [191](#), [202](#), [224](#),  
[228](#), [236](#), [245](#)  
 managedQuery() method, [195](#)  
 MapActivity class, [175](#)  
 MapController class, [175](#)  
 MapView class, [172–177](#)  
   embedding, [172](#)  
   Hello, Android, [172](#)  
   location of, [176](#)  
 Market, publishing to, [35](#), [271–276](#)  
   updating published applications,  
   [275](#)  
 Matrix transformation, [228](#)  
 Media codes, [32](#)  
 MediaPlayer, [110](#)  
 MediaPlayer class, [105](#), [107](#), [118](#)  
 MediaRecorder class, [119](#), [177](#)  
 Menu button, [64](#)  
 MenuInflater class, [65](#)  
 Menus, adding, [64–65](#)  
 meta-data tag, [234](#), [244](#)  
 Midpoint between touches  
   (multi-touch), [231](#)  
 mm. (millimeters) as resolution units,  
   [54](#)  
 Modification statements, [181](#)  
 MotionEvent class, [259](#)  
 moveToNext() method, [188](#)  
 Movies, [112–115](#)  
 MP3 audio format, [111](#)  
 Multi-touch features, [220–232](#)  
   about touch events, [225–228](#)  
   image transformation, [228–229](#)  
   implementing drag gesture, [229](#), [230](#)  
   implementing pinch zoom, [230–232](#)  
 Multimedia, [105–119](#)  
   audio, [105–110](#)  
   background music, Sudoku,  
   [115–119](#)

  video, [112–115](#)  
 Multithreading, [279](#)  
 Music class, [118](#)  
 MyLocationOverlay class, [176](#)  
 MyMap class, [174](#), [177](#)

## N

Naming Java packagers, [271](#)  
 Native Development Toolkit (NDK), [32](#)  
 Native libraries, [31–32](#)  
 NDK (Native Development Toolkit), [32](#)  
 New game button, [67–69](#)  
 Nikoli, [79](#)  
 NinePatch, [77](#), [236–238](#)  
 Notification manager, [34](#)

## O

Object finalization, [279](#)  
 Objects, [39–40](#)  
 OGG audio format, [111](#)  
 onClick() method, [60](#), [68](#)  
 onCommand() method, [254](#)  
 onConfigurationChanged() method, [116](#)  
 onCreate() method, [36](#), [46](#), [77](#), [80](#), [124](#),  
[128](#), [164](#)  
 onCreate() method (Service), [245](#), [249](#)  
 onCreateEngine() method, [246](#)  
 onCreateOptionsMenu() method, [65](#), [120](#)  
 onDestroy() method, [38](#), [116](#)  
 onDestroy() method (Service), [245](#), [250](#)  
 onDraw() method, [75](#), [81](#), [87](#), [122](#), [202](#)  
   calling drawing functions, [89](#)  
   re-creating screen from scratch, [83](#)  
   speeding up, [103](#)  
 onDrawFrame() method, [205](#), [212](#)  
   frame rate and, [217](#)  
 onJsAlert() method, [145](#)  
 onKeyDown() method, [88](#), [89](#), [107](#)  
 onLocationChanged() method, [166](#)  
 onOffsetsChanged() method, [253](#)  
 onOptionItemSelected() method, [65](#)  
 onPause() method, [37](#), [117](#), [165](#), [169](#),  
[202](#)  
 onProviderDisabled() method, [166](#)  
 onProviderEnabled() method, [166](#)  
 onRestart() method, [38](#)  
 onRestoreInstanceState() method, [38](#), [125](#),  
[126](#)  
 onResume() method, [37](#), [115](#), [117](#), [165](#),  
[169](#), [202](#)



onRetainNonConfigurationInstance()  
     **method, 116**  
 onSaveInstanceState() **method, 37, 38,**  
     **125, 126**  
 onSensorChanged() **method, 169**  
 onSizeChanged() **method, 81, 82**  
 onStart() **method, 37**  
 onStartCommand() **method (Service), 245**  
 onStatusChanged() **method, 166**  
 onStop() **method, 37**  
 onSurfaceCreated() **method, 203, 204,**  
     **210, 212**  
 onTouchEvent() **method, 90, 254**  
 onTrackballEvent() **method, 89**  
 onUpdate() **method, 241**  
 onUpgrade() **method, 183**  
 onVisibilityChanged() **method, 252**  
 openBrowser() **method, 134, 137**  
 openFileInput() **method, 126**  
 openFileOutput() **method, 126**  
**OpenGL, 198–217**  
     **animation, 212–213**  
     **cube model, 206–210**  
     **fixed- vs. floating-point interface,**  
         **209**  
     **Hello, Android, 200–202**  
     **lighting, 209–212**  
     **for live wallpapers, 248**  
     **options, disabling/enabling, 204**  
     **overview of, 199–200**  
     **texture, 212–216**  
     **threads in, 202–206**  
     **three-dimensional graphics and,**  
         **198–199**  
     **transparency, 216**  
     **versions, 206**  
**OpenGL.java file, 201**  
**OpenGL options**  
     **GL\_BLEND, 204**  
     **GL\_CULL\_FACE, 204**  
     **GL\_DEPTH\_TEST, 204**  
     **GL\_DITHER, 204**  
     **GL\_LIGHT, 204**  
     **GL\_LIGHTING, 204**  
     **GL\_LINE\_SMOOTH, 204**  
     **GL\_MULTISAMPLE, 204**  
     **GL\_POINT\_SMOOTH, 204**  
     **GL\_TEXTURE\_2D, 204**  
**OpenIntents, 171**  
 openNewGameDialog() **method, 68**

**ORDER\_BY parameter, SELECT statement,**  
     **187**  
 orientation value, **android:configChanges,**  
     **116**  
**Owens, Mike, 196**

## P

**Package names, 46, 271**  
**Paint class, 74**  
**Path class, 75**  
**PathEffect classes, 76**  
**Pausing multimedia play, 117**  
**Performance, 28, 63, 103, 188, 189,**  
     **198, 201, 204, 209, 213, 217,**  
     **218, 231, 260, 279**  
**Permissions, 41, 139, 162**  
     **ACCESS\_COARSE\_LOCATION**  
         **permission, 41, 162**  
     **ACCESS\_FINE\_LOCATION permission,**  
         **41, 162**  
     **BIND\_WALLPAPER permission, 244**  
     **INTERNET permission, 41, 138, 149**  
     **READ\_CONTACTS permission, 41**  
     **RECEIVE\_SMS permission, 41**  
     **WRITE\_CONTACTS permission, 41**  
     **WRITE\_EXTERNAL\_STORAGE permission,**  
         **128**  
**Phone**  
     **3D hardware and, 201**  
     **installation, 28**  
     **keypads, 93, 96**  
     **prepackaged systems, 34**  
     **uses for, 130**  
**Pinch zoom, 221, 230–232**  
**Pixels, 54**  
**Planet Android, 72**  
**Platform, 17**  
 play() **method, 115, 118, 122**  
**Playback, 112**  
 post() **method, 145**  
**Prefs class, 65**  
**Preparing to publish, 271**  
**PRIMARY KEY designation, 180**  
**Private intents, 60**  
**Procedural design, 44**  
**Processes vs. activities, 36**  
**Proxy, 22**  
 pt. (points) as resolution units, **54**  
**Public intents, 60**  
**Publishing to Android Market, 35,**  
     **271–276**

updating published applications,  
275

PuzzleView class, 80, 81

px. (pixels) as resolution units, 54

## Q

Queries, running, 187

query() method, 187

Query statements, 181

queueUpdate() method, 154, 155, 157

QuickTime Pro, 112

## R

R class, 40, 48

managed by Eclipse, 47

R.java file, 47

raw directory, 107

rawQuery() method, 187

READ\_CONTACTS permission, 41

RECEIVE\_SMS permission, 41

receiver tag, 234

Reflection, 260, 279

Refresh, 90

Refresh rate, smoothness and, 217

registerListener() method, 169

RelativeLayout class, 49

release() method, 109, 119

RemoteViews class, 241

removeUpdates() method, 166

Renderer interface, 203

rendering, 202–206

requestLocationUpdates() method, 165

res directory, 47, 107

Resampling artifacts, 111

Resolution, 87

Resolution units

in. (inches) as resolution units, 54

mm. (millimeters) as resolution  
units, 54

pt. (points) as resolution units, 54

px. (pixels) as resolution units, 54

Resolution-independent

measurements, 54

Resource compiler, 40

Resource manager, 34

Resources, 40

Resuming paused multimedia, 117

returnResult() method, 98

RGB color values, 74

Rows, adding, 186, 194

Runnable class, 145, 147

runOnFirstFix() method, 176

Runtime, 32

RuntimeException class, 186

## S

Sampling rates, 111

Scale, with pinch zoom gesture, 231

Scale containers, 77

Scale-independent pixels, 54

ScaleGestureDetector class, 221

Screen layouts, 55, 57

Screen position, 124–126

Screen resolution, 54

Screen rotations, 116

SD cards, 127–128, 268

SDK install directory, 19

SDK Setup program, 19

SDK Starter Package, 18

Secure Digital (SD) cards, 127–128,  
268

Security

JavaScript and, 140

permissions, 41, 128

select() method, 89

SELECT statements, 181, 187

Selecting tiles, 87

sensor types

TRICORDER type, 169n

TYPE\_ACCELEROMETER type, 169

TYPE\_LIGHT type, 169

TYPE\_MAGNETIC\_FIELD type, 169

TYPE\_ORIENTATION type, 169, 170

TYPE\_PRESSURE type, 169

TYPE\_PROXIMITY type, 169

TYPE\_TEMPERATURE type, 169, 170

SensorManager class, 169

Sensors, 168–172

Emulator and, 170

readings, interpreting, 169

simulator, 171

SensorTest class, 169

Service class, 117, 245

service tag, 244

Services, 39, 245

setAdapters() method, 152

setBackgroundResource() method, 77

setBuiltInZoomControls() method, 175

setColor() method, 74

setContentView() method, 46, 134

setDownloadListener() method, 139

setId() method, 125

- setImageMatrix() method, 265–267
- setItems() method, 69
- setListeners() method, 154
- setOnClickListener() method, 59, 146
- setOnCompletionListener() method, 109
- setSatellite() method, 175
- setText() method, 145
- settings.xml file, 65
- Settings, adding, 66, 67
- setTranslated() method, 157
- Setup program (Android SDK), 19
- setVideoPath() method, 113
- setVolumeControlStream() method, 107
- setWebChromeClient() method, 139, 145
- setWebViewClient() method, 139
- Shaking the screen (animation), 93
- Shape drawables, 77
- showEvents() method, 186, 188, 189
- Signing applications, 272
- Silicon Graphics, 199
- SimpleCursorAdapter class, 189, 197
- 64-bit Java Development Kit, 19
- Size, 81, 87
- Skins, emulator, 257
- Smoothness, measuring, 217
- Soft keyboards, 132
- Soft lighting, 209
- Sound effects, 105–110
- SoundPool class, 110n
- sp (scale-independent pixels), 54
- Spacing between fingers (multi-touch), 231
- Specular lighting, 210
- Speed, 28, 63, 103, 188, 189, 198, 201, 204, 209, 213, 217, 218, 231, 260, 279
- Spinner class, 150
- Spinners, 150
- Sps, 54
- SQL Pocket Guide* (Gennick), 196
- SQLException class, 186
- SQLite, 32, 178–196
  - about, 178–179, 180f
  - application basics, 185f, 181–189
  - blessing (license), 179
  - ContentProvider class, 192–196
  - data binding, 192f, 189–192
  - DDL statements, 180
  - modification statements, 181
  - overview of, 179–181
  - queries, running, 187
  - query statements, 181
  - rows, adding, 186, 194
- SQLiteOpenHelper class, 182
- start() method, 109, 112, 119
- startActivity method, 135
- Starter Package (Android SDK), 18
- startGame() method, 78, 124
- startManagingCursor() method, 187
- State containers, 77
- State-saving code, 38
- stop() method, 112, 117, 118
- stopLoading() method, 139
- Storing local data, 120–128
  - current screen position, 124–126
  - internal files, accessing, 126
  - options, adding, 120–122
  - pausing game, 122–124
  - secure digital (SD), 127–128
- Stretchable backgrounds, 236
- String resource, 58
- strings.xml file, 58, 64, 67, 80, 132, 138, 146, 157, 191, 239, 245
- strings.xml file, 50
- styles.xml file, 61
- Subclassing, 260
- Sudoku history, 44, 79, 118, 121
- Sudoku program, 43–72
  - About box, 57–62
  - debugging, 69–71
  - defined, 43
  - exit button, 71
  - game board, 83
  - game class for, 79
  - game logic, 99
  - graphics for, 78–87
  - hints, 91, 122
  - input and graphics, 87–93
  - interface design, 44–45
  - menus, 64–65
  - multimedia, 105–119
    - audio, 105–110
    - background music for, 115–119
    - video, 112–115
  - new game button, 67–69
  - opening screen, 45–54
  - pausing, 122
  - PuzzleView class, 81
  - screen layouts, 55–57
  - settings, 66, 67
  - starting game (code), 78
  - themes, 61–63

Surface class, [112](#), [203](#), [204](#)  
 Surface Manager, [31](#)  
 Synchronization, [279](#)

## T

---

TableLayout class, [49](#), [57](#), [149](#)  
 TableRow class, [150](#)  
 Tap gesture, [220](#)  
 Testing, state-saving code, [38](#)  
 Testing programs, [257](#)  
 Texture, OpenGL, [212–216](#)  
 TextView class, [63](#), [141](#), [145](#), [163](#)  
 Themes, [61–63](#)  
 32-bit Java Development Kit, [19](#)  
 Thread class, [147](#)  
 Threads  
   multiple, [279](#)  
   OpenGL, [202–206](#)  
 Tile selection, [87](#)  
 Time slicing, [279](#)  
 Time-based animation, [213](#)  
 Toast class, [145](#)  
 tools directory, SDK, [19](#)  
 toPuzzleString() method, [102](#)  
 Torvalds, Linus, [30](#)  
 Touch.java file, [224](#)  
 traceview profiler, [218](#)  
 Trackball, [89](#)  
 Translate class, [151](#)  
 Translate program, [131](#)  
 TranslateTask class, [156](#)  
 Translator, [147](#), [148](#)  
 Transparency, [216](#)  
 Triangle strips, [208](#)  
 TRICORDER sensor type, [169n](#)  
 2D and 3D graphics, [32](#)  
 TYPE\_ACCELEROMETER sensor type, [169](#)  
 TYPE\_LIGHT sensor type, [169](#)  
 TYPE\_MAGNETIC\_FIELD sensor type, [169](#)  
 TYPE\_ORIENTATION sensor type, [169](#), [170](#)  
 TYPE\_PRESSURE sensor type, [169](#)  
 TYPE\_PROXIMITY sensor type, [169](#)  
 TYPE\_TEMPERATURE sensor type, [169](#), [170](#)

## U

---

Uniform Resource Identifier (URI), [193](#)  
 unregisterListener() method, [169](#)  
 updatePeriodMillis parameter, [241](#)  
 Updates, widget, [239–241](#)  
 Updating published applications, [275](#)  
 Uri class, [135](#)

UriMatcher class, [196](#)  
 User ID, [127](#)  
 User interface design, [43–72](#)  
   About box, [57–62](#)  
   debugging, [69–71](#)  
   device-specific UI, [267](#)  
   exit button, [71](#)  
   menus, [64–65](#)  
   multi-touch features, [220–232](#)  
   new game button, [67–69](#)  
   opening screen, [45–54](#)  
   screen layouts, [55–57](#)  
   settings, [66](#), [67](#)  
   Sudoku overview, [43](#), [45](#)  
   themes, [61–63](#)  
   types of, [44–45](#)  
 uses-library tag, [174](#)  
 uses-permission tag, [139](#), [148](#)

## V

---

v() method (Log class), [69](#)  
 Vector graphics, [54](#), [75](#)  
 VerifyError exceptions, [260](#)  
 versionCode parameter, [272](#), [275](#)  
 versionName parameter, [272](#), [275](#)  
 Versions, Android, [256](#)  
   building for multiple, [257–259](#)  
 Video, [112–115](#)  
 Video class, [128](#)  
 Video formats, supported, [112](#)  
 Video.java file, [113](#)  
 VideoView class, [112](#)  
 View frustrum, [198](#)  
 View size, [81](#)

## W

---

w() method (Log class), [69](#)  
 wallpaper, live, [242–254](#)  
 WallpaperService class, [245](#)  
 WAV audio format, [111](#)  
 Web services, [147–160](#)  
 Web Standard Tools (WST), [21](#)  
 WebClient class, [145](#)  
 WebKit library, [32](#)  
 WebView, [135–139](#)  
 WebView class, [63](#), [136](#), [138](#)  
 WHERE clause, [187](#)  
 Widget class, [238](#)  
 widget.xml file, [234](#)  
 Widgets, [34](#), [233–242](#)

- extending `AppWidgetProvider`,  
238–239
- getting updates, 239–241
- running, 239
- stretchable backgrounds, 236–238

Windows, in Android, 35–38

`WRITE_CONTACTS` permission, 41

`WRITE_EXTERNAL_STORAGE` permission,  
128

`wtf()` method (`Log` class), 69

## X

---

`XmlHttpRequest` class, 172

`xmlns:android` parameter, 49

## Z

---

.zip files, 139

Zoom gesture (pinch zoom), 221,  
230–232

# The Pragmatic Bookshelf

---

Available in paperback and DRM-free eBooks, our titles are here to help you stay on top of your game. The following are in print as of June 2010; be sure to check our website at [pragprog.com](http://pragprog.com) for newer titles.

<b>Title</b>	<b>Year</b>	<b>ISBN</b>	<b>Pages</b>
<i>Advanced Rails Recipes: 84 New Ways to Build Stunning Rails Apps</i>	2008	9780978739225	464
<i>Agile Coaching</i>	2009	9781934356432	248
<i>Agile Retrospectives: Making Good Teams Great</i>	2006	9780977616640	200
<i>Agile Web Development with Rails, Third Edition</i>	2009	9781934356166	784
<i>Beginning Mac Programming: Develop with Objective-C and Cocoa</i>	2010	9781934356517	300
<i>Behind Closed Doors: Secrets of Great Management</i>	2005	9780976694021	192
<i>Best of Ruby Quiz</i>	2006	9780976694076	304
<i>Cocoa Programming: A Quick-Start Guide for Developers</i>	2010	9781934356302	450
<i>Core Animation for Mac OS X and the iPhone: Creating Compelling Dynamic User Interfaces</i>	2008	9781934356104	200
<i>Core Data: Apple's API for Persisting Data on Mac OS X</i>	2009	9781934356326	256
<i>Data Crunching: Solve Everyday Problems using Java, Python, and More</i>	2005	9780974514079	208
<i>Debug It! Find, Repair, and Prevent Bugs in Your Code</i>	2009	9781934356289	232
<i>Deploying Rails Applications: A Step-by-Step Guide</i>	2008	9780978739201	280
<i>Design Accessible Web Sites: 36 Keys to Creating Content for All Audiences and Platforms</i>	2007	9781934356029	336
<i>Desktop GIS: Mapping the Planet with Open Source Tools</i>	2008	9781934356067	368
<i>Developing Facebook Platform Applications with Rails</i>	2008	9781934356128	200
<i>Domain-Driven Design Using Naked Objects</i>	2009	9781934356449	375
<i>Enterprise Integration with Ruby</i>	2006	9780976694069	360
<i>Enterprise Recipes with Ruby and Rails</i>	2008	9781934356234	416
<i>Everyday Scripting with Ruby: for Teams, Testers, and You</i>	2007	9780977616619	320
<i>ExpressionEngine 2: A Quick-Start Guide</i>	2010	9781934356524	250
<i>FXRuby: Create Lean and Mean GUIs with Ruby</i>	2008	9781934356074	240
<i>From Java To Ruby: Things Every Manager Should Know</i>	2006	9780976694090	160

---

Continued on next page

<b>Title</b>	<b>Year</b>	<b>ISBN</b>	<b>Pages</b>
<i>GIS for Web Developers: Adding Where to Your Web Applications</i>	2007	9780974514093	275
<i>Google Maps API, V2: Adding Where to Your Applications</i>	2006	PDF-Only	83
<i>Grails: A Quick-Start Guide</i>	2009	9781934356463	200
<i>Groovy Recipes: Greasing the Wheels of Java</i>	2008	9780978739294	264
<i>Interface Oriented Design</i>	2006	9780976694052	240
<i>Land the Tech Job You Love</i>	2009	9781934356265	280
<i>Language Implementation Patterns: Create Your Own Domain-Specific and General Programming Languages</i>	2009	9781934356456	350
<i>Learn to Program, 2nd Edition</i>	2009	9781934356364	240
<i>Manage It! Your Guide to Modern Pragmatic Project Management</i>	2007	9780978739249	360
<i>Manage Your Project Portfolio: Increase Your Capacity and Finish More Projects</i>	2009	9781934356296	200
<i>Mastering Dojo: JavaScript and Ajax Tools for Great Web Experiences</i>	2008	9781934356111	568
<i>Metaprogramming Ruby: Program Like the Ruby Pros</i>	2010	9781934356470	240
<i>Modular Java: Creating Flexible Applications with OSGi and Spring</i>	2009	9781934356401	260
<i>No Fluff Just Stuff 2006 Anthology</i>	2006	9780977616664	240
<i>No Fluff Just Stuff 2007 Anthology</i>	2007	9780978739287	320
<i>Pomodoro Technique Illustrated: The Easy Way to Do More in Less Time</i>	2009	9781934356500	144
<i>Practical Programming: An Introduction to Computer Science Using Python</i>	2009	9781934356272	350
<i>Practices of an Agile Developer</i>	2006	9780974514086	208
<i>Pragmatic Ajax: A Web 2.0 Primer</i>	2006	9780976694083	296
<i>Pragmatic Project Automation: How to Build, Deploy, and Monitor Java Applications</i>	2004	9780974514031	176
<i>Pragmatic Thinking and Learning: Refactor Your Wetware</i>	2008	9781934356050	288
<i>Pragmatic Unit Testing in C# with NUnit</i>	2007	9780977616671	176
<i>Pragmatic Unit Testing in Java with JUnit</i>	2003	9780974514017	160
<i>Pragmatic Version Control Using Git</i>	2008	9781934356159	200
<i>Pragmatic Version Control using CVS</i>	2003	9780974514000	176
<i>Pragmatic Version Control using Subversion</i>	2006	9780977616657	248
<i>Programming Clojure</i>	2009	9781934356333	304
<i>Programming Cocoa with Ruby: Create Compelling Mac Apps Using RubyCocoa</i>	2009	9781934356197	300
<i>Programming Erlang: Software for a Concurrent World</i>	2007	9781934356005	536

Continued on next page

<b>Title</b>	<b>Year</b>	<b>ISBN</b>	<b>Pages</b>
<i>Programming Groovy: Dynamic Productivity for the Java Developer</i>	2008	9781934356098	320
<i>Programming Ruby: The Pragmatic Programmers' Guide, Second Edition</i>	2004	9780974514055	864
<i>Programming Ruby 1.9: The Pragmatic Programmers' Guide</i>	2009	9781934356081	960
<i>Programming Scala: Tackle Multi-Core Complexity on the Java Virtual Machine</i>	2009	9781934356319	250
<i>Prototype and script.aculo.us: You Never Knew JavaScript Could Do This!</i>	2007	9781934356012	448
<i>Rails Recipes</i>	2006	9780977616602	350
<i>Rails for .NET Developers</i>	2008	9781934356203	300
<i>Rails for Java Developers</i>	2007	9780977616695	336
<i>Rails for PHP Developers</i>	2008	9781934356043	432
<i>Rapid GUI Development with QtRuby</i>	2005	PDF-Only	83
<i>Release It! Design and Deploy Production-Ready Software</i>	2007	9780978739218	368
<i>SQL Antipatterns: Avoiding the Pitfalls of Database Programming</i>	2010	9781934356555	300
<i>Scripted GUI Testing with Ruby</i>	2008	9781934356180	192
<i>Ship It! A Practical Guide to Successful Software Projects</i>	2005	9780974514048	224
<i>Stripes ...and Java Web Development Is Fun Again</i>	2008	9781934356210	375
<i>TextMate: Power Editing for the Mac</i>	2007	9780978739232	208
<i>The Definitive ANTLR Reference: Building Domain-Specific Languages</i>	2007	9780978739256	384
<i>The Passionate Programmer: Creating a Remarkable Career in Software Development</i>	2009	9781934356340	200
<i>ThoughtWorks Anthology</i>	2008	9781934356142	240
<i>Ubuntu Kung Fu: Tips, Tricks, Hints, and Hacks</i>	2008	9781934356227	400
<i>Web Design for Developers: A Programmer's Guide to Design Tools and Techniques</i>	2009	9781934356135	300
<i>iPhone SDK Development</i>	2009	9781934356258	576



# Grow your Skills

## Debug It!

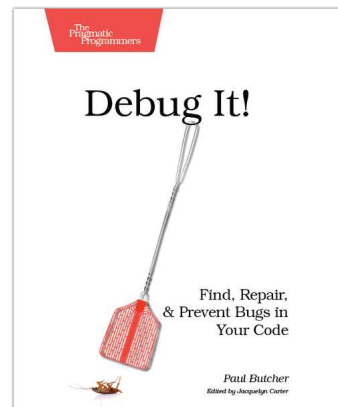
*Debug It!* will equip you with the tools, techniques, and approaches to help you tackle any bug with confidence. These secrets of professional debugging illuminate every stage of the bug life cycle, from constructing software that makes debugging easy; through bug detection, reproduction, and diagnosis; to rolling out your eventual fix. Learn better debugging whether you're writing Java or assembly language, targeting servers or embedded micro-controllers, or using agile or traditional approaches.

### **Debug It! Find, Repair, and Prevent Bugs in Your Code**

Paul Butcher

(232 pages) ISBN: 978-1-9343562-8-9. \$34.95

<http://pragprog.com/titles/pbdp>



## SQL Antipatterns

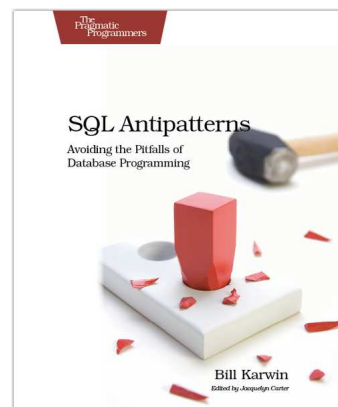
If you're programming applications that store data, then chances are you're using SQL, either directly or through a mapping layer. But most of the SQL that gets used is inefficient, hard to maintain, and sometimes just plain wrong. This book shows you all the common mistakes, and then leads you through the best fixes. What's more, it shows you what's *behind* these fixes, so you'll learn a lot about relational databases along the way.

### **SQL Antipatterns: Avoiding the Pitfalls of Database Programming**

Bill Karwin

(300 pages) ISBN: 978-19343565-5-5. \$34.95

<http://pragprog.com/titles/bksqla>



# Agile Techniques

---

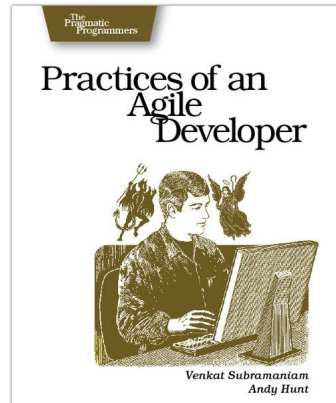
## Practices of an Agile Developer

Agility is all about using feedback to respond to change. Learn how to

- apply the principles of agility throughout the software development process
- establish and maintain an agile working environment
- deliver what users really want
- use personal agile techniques for better coding and debugging
- use effective collaborative techniques for better teamwork
- move to an agile approach

### **Practices of an Agile Developer: Working in the Real World**

Venkat Subramaniam and Andy Hunt  
(189 pages) ISBN: 0-9745140-8-X. \$29.95  
<http://pragprog.com/titles/pad>



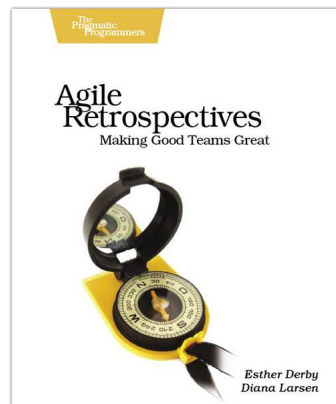
## Agile Retrospectives

Mine the experience of your software development team continually throughout the life of the project. Rather than waiting until the end of the project—as with a traditional retrospective, when it's too late to help—agile retrospectives help you adjust to change *today*.

The tools and recipes in this book will help you uncover and solve hidden (and not-so-hidden) problems with your technology, your methodology, and those difficult “people issues” on your team.

### **Agile Retrospectives: Making Good Teams Great**

Esther Derby and Diana Larsen  
(170 pages) ISBN: 0-9776166-4-9. \$29.95  
<http://pragprog.com/titles/dlret>



# iPhone/iPad

## iPhone SDK Development

Jump into application development for today's most remarkable mobile communications platform, the Pragmatic way. This Pragmatic guide takes you through the tools and APIs, the same ones Apple uses for its applications, that you can use to create your own software for the iPhone and iPod touch. Packed with useful examples, this book will give you both the big-picture concepts and the everyday "gotcha" details that developers need to make the most of the beauty and power of the iPhone OS platform.

### **iPhone SDK Development**

Bill Dudney, Chris Adamson, Marcel Molina  
(545 pages) ISBN: 978-1-9343562-5-8. \$38.95

<http://pragprog.com/titles/amiphd>



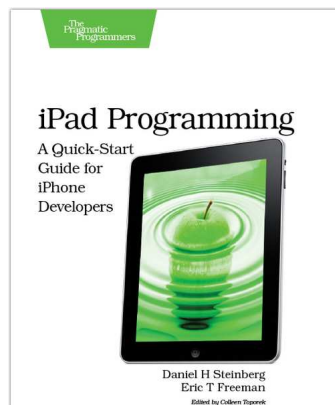
## iPad Programming

It's not an iPhone and it's not a laptop: the iPad is a groundbreaking new device. You need to create true iPad apps to take advantage of all that is possible with the iPad. If you're an experienced iPhone developer, *iPad Programming* will show you how to write these outstanding new apps while completely fitting your users' expectation for this device.

### **iPad Programming: A Quick-Start Guide for iPhone Developers**

Daniel H Steinberg and Eric T Freeman  
(250 pages) ISBN: 978-19343565-7-9. \$34.95

<http://pragprog.com/titles/sfipad>



# Refactor Your Career

## Land the Tech Job You Love

You've got the technical chops—the skills to get a great job doing what you love. Now it's time to get down to the business of planning your job search, focusing your time and attention on the job leads that matter, and interviewing to wow your boss-to-be.

You'll learn how to find the job you want that fits you and your employer. You'll uncover the hidden jobs that never make it into the classifieds or Monster. You'll start making and maintaining the connections that will drive your future career moves.

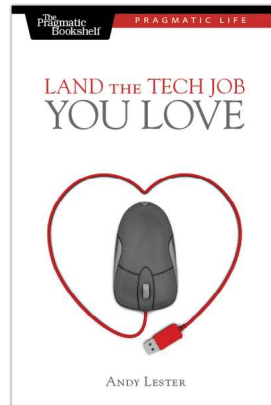
You'll land the tech job you love.

### **Land the Tech Job You Love**

Andy Lester

(280 pages) ISBN: 978-1934356-26-5. \$23.95

<http://pragprog.com/titles/algh>



## Manage It!

*Manage It!* is an award-winning, risk-based guide to making good decisions about how to plan and guide your projects. Author Johanna Rothman shows you how to beg, borrow, and steal from the best methodologies to fit your particular project. You'll find what works best for *you*.

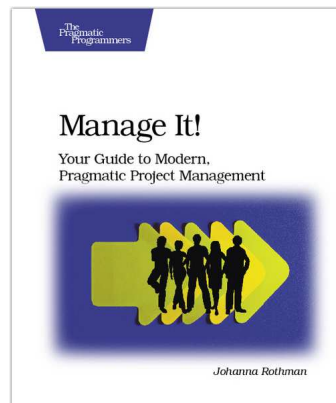
- Learn all about different project lifecycles
- See how to organize a project
- Compare sample project dashboards
- See how to staff a project
- Know when you're done—and what that means.

### **Manage It! Your Guide to Modern, Pragmatic Project Management**

Johanna Rothman

(360 pages) ISBN: 0-9787392-4-8. \$34.95

<http://pragprog.com/titles/jrpm>



# Refactor Your Career

## Pragmatic Thinking and Learning

Software development happens in your head. Not in an editor, IDE, or design tool. In this book by Pragmatic Programmer Andy Hunt, you'll learn how our brains are wired, and how to take advantage of your brain's architecture. You'll master new tricks and tips to learn more, faster, and retain more of what you learn.

- Use the Dreyfus Model of Skill Acquisition to become more expert
- Leverage the architecture of the brain to strengthen different thinking modes
- Avoid common "known bugs" in your mind
- Learn more deliberately and more effectively
- Manage knowledge more efficiently

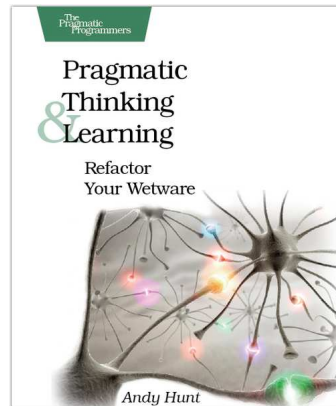
### **Pragmatic Thinking and Learning:**

#### **Refactor your Wetware**

Andy Hunt

(288 pages) ISBN: 978-1-9343560-5-0. \$34.95

<http://pragprog.com/titles/ahpt/>



## The Passionate Programmer

This book is about creating a remarkable career in software development. Remarkable careers don't come by chance. They require thought, intention, action, and a willingness to change course when you've made mistakes. Most of us have been stumbling around letting our careers take us where they may. It's time to take control.

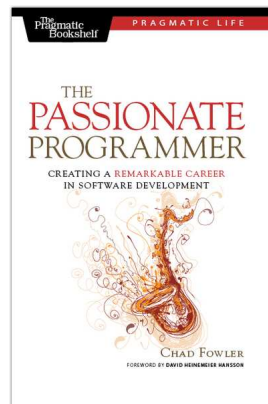
This revised and updated second edition lays out a strategy for planning and creating a radically successful life in software development (*the first edition was released as My Job Went to India: 52 Ways To Save Your Job*).

### **The Passionate Programmer: Creating a Remarkable Career in Software Development**

Chad Fowler

(232 pages) ISBN: 978-1934356-34-0. \$23.95

<http://pragprog.com/titles/cfcar2>



# The Pragmatic Bookshelf

---

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help you stay on top of your game.

## Visit Us Online

---

### Home page for Hello Android, Third Edition

<http://pragprog.com/titles/eband3>

Source code from this book, errata, and other resources. Come give us feedback, too!

### Register for Updates

<http://pragprog.com/updates>

Be notified when updates and new books become available.

### Join the Community

<http://pragprog.com/community>

Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

### New and Noteworthy

<http://pragprog.com/news>

Check out the latest pragmatic developments, new titles and other offerings.

## Buy the Book

---

If you liked this eBook, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: [pragprog.com/titles/eband3](http://pragprog.com/titles/eband3).

## Contact Us

---

Online Orders:	<a href="http://www.pragprog.com/catalog">www.pragprog.com/catalog</a>
Customer Service:	<a href="mailto:support@pragprog.com">support@pragprog.com</a>
Non-English Versions:	<a href="mailto:translations@pragprog.com">translations@pragprog.com</a>
Pragmatic Teaching:	<a href="mailto:academic@pragprog.com">academic@pragprog.com</a>
Author Proposals:	<a href="mailto:proposals@pragprog.com">proposals@pragprog.com</a>
Contact us:	1-800-699-PROG (+1 919 847 3884)