O'REILLY

Head First Android Development

A Brain-Friendly Guide

Design top-selling apps

Avoid embarrassing activities

See how Material Design can change your life

Master out-of-this-world concepts

Tap into the Android Location Service

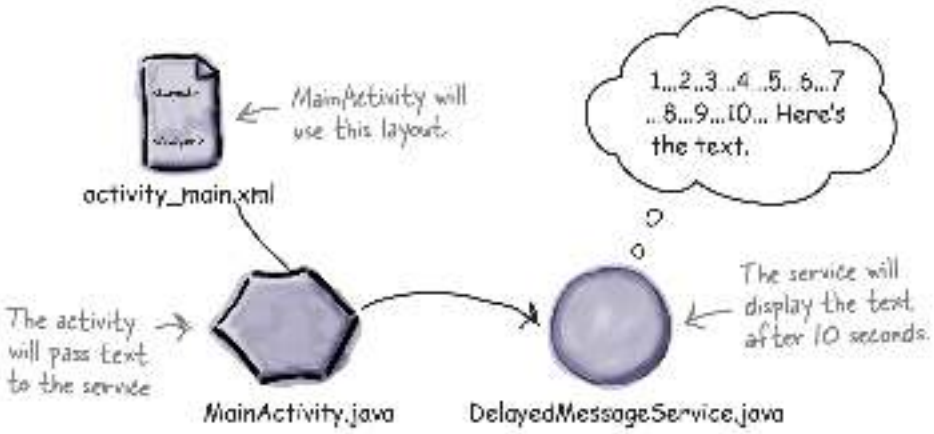
Fool around in the Android Support Libraries

Dawn Griffiths & David Griffiths

Android Development

What will you learn from this book?

If you have an idea for a killer Android app, this book will help you build your first working application in a jiffy. You'll learn hands-on how to structure your app, design interfaces, create a database, make your app work on various smartphones and tablets, and much more. It's like having an experienced Android developer sitting right next to you! All you need is some Java know-how to get started.



Why does this book look so different?

Based on the latest research in cognitive science and learning theory, *Head First Android Development* uses a visually rich format to engage your mind, rather than a text-heavy approach that puts you to sleep. Why waste your time struggling with new concepts? This multi-sensory learning experience is designed for the way your brain really works.

"A comprehensive beginner's guide to Android development, easy to read and full of excellent examples and exercises."

—Edward Yue Shung Wong (@shungeiphone)

"This is, without a doubt, the best available book for learning Android development. If you can get only one, make it this one."

—Kenneth Kousen President, Kousen IT Inc., and JavaOne Rock Star

"Become an able Android developer applying up-to-date patterns and create that next killer app. *Head First Android Development* will be your friendly, accurate, and fun-to-be-with master craftsman on that path."

—Ingo Kinzig Android Learner

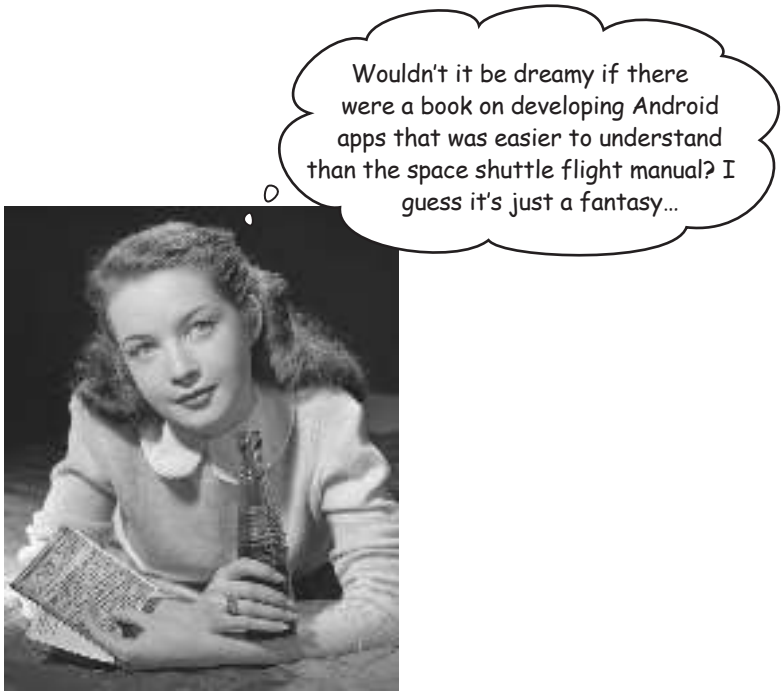
Programming / Android

US $49.99 CAN $57.99

ISBN: 978-1-449-36218-8

oreilly.com

Head First
Android Development



Dawn Griffiths
David Griffiths

O'REILLY®

Beijing ▪ Cambridge ▪ Köln ▪ Sebastopol ▪ Tokyo

Head First Android Development

by Dawn Griffiths and David Griffiths

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly Media books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://safaribooksonline.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

Series Creators: Kathy Sierra, Bert Bates

Editor: Meghan Blanchette

Cover Designer: Karen Montgomery

Production Editor: Melanie Yarbrough

Production Services: Jasmine Kwityn

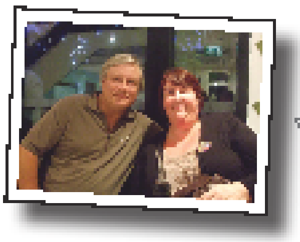
Indexer: Bob Pfahler

Page Viewers: Mum and Dad, Carl

Printing History:

June 2015: First Edition.

Mum and Dad



Rob and Lorraine

No kittens were harmed in the making of this book, but several pizzas were eaten.

ISBN: 978-1-449-36218-8

[LSI] [2015-06-19]

To our friends and family. Thank you so
much for all your love and support.

Authors of Head First Android Development

Dawn Griffiths started life as a mathematician at a top UK university, where she was awarded a first-class honors degree in mathematics. She went on to pursue a career in software development and has 20 years experience working in the IT industry.

Before writing Head First Android Development, Dawn wrote three other Head First books (*Head First Statistics*, *Head First 2D Geometry* and *Head First C*) and has also worked on a host of other books in the series.

When Dawn's not working on Head First books, you'll find her honing her Tai Chi skills, reading, running, making bobbin lace, or cooking. She particularly enjoys spending time with her wonderful husband, David.

David Griffiths began programming at age 12, when he saw a documentary on the work of Seymour Papert. At age 15, he wrote an implementation of Papert's computer language LOGO. After studying pure mathematics at university, he began writing code for computers and magazine articles for humans. He's worked as an agile coach, a developer, and a garage attendant, but not in that order. He can write code in over 10 languages and prose in just one, and when not writing, coding, or coaching, he spends much of his spare time traveling with his lovely wife—and coauthor—Dawn.

Before writing *Head First Android Development*, David wrote three other Head First books: *Head First Rails*, *Head First Programming* and *Head First C*.

You can follow us on Twitter at *https://twitter.com/HeadFirstDroid*.

Table of Contents (Summary)

Table of Contents (the real thing)

Intro

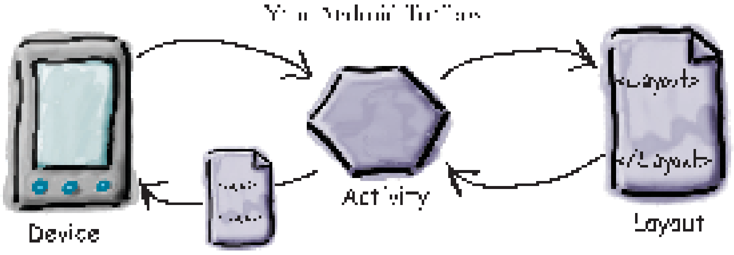
Your brain on Android. Here *you* are trying to *learn* something, while here your *brain* is, doing you a favor by making sure the learning doesn't *stick*. Your brain's thinking, "Better leave room for more important things, like which wild animals to avoid and whether naked snowboarding is a bad idea." So how *do* you trick your brain into thinking that your life depends on knowing how to develop Android apps?

getting started

1 Diving In

Android has been taking the world by storm.

Everybody wants a smart phone or tablet, and Android devices are hugely popular. In this book we'll teach you how to **develop your own apps**, and we'll start by getting you to build a basic app and run it on an Android Virtual Device. Along the way you'll meet some of the basic components of all Android apps such as *activities* and *layouts*. **All you need is a little Java know-how...**

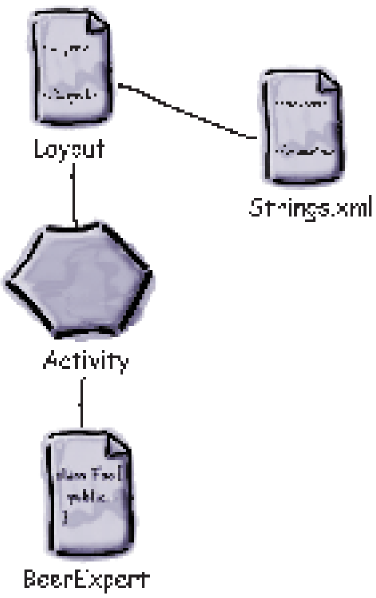


2 building interactive apps

Apps That Do Something

Most apps need to respond to the user in some way.

In this chapter you'll see how you can make your apps *a bit more interactive*. You'll see how you can get your app to **do** something in response to the user, and **how to get your activity and layout talking to each other** like best buddies. Along the way we'll take you a bit **deeper** into ***how Android actually works*** by introducing you to R, the hidden gem that glues everything together.

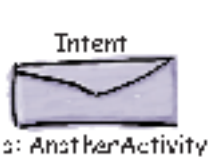


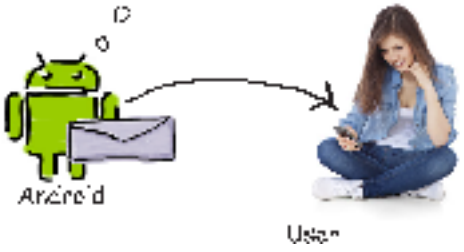
multiple activities and intents

3 State Your Intent

Most apps need more than one activity.

So far we've just looked at single activity apps, which is fine for simple apps. But when things get more complicated, just having one activity won't cut it. We're going to show you **how to build apps with multiple activities**, and how you can get your apps talking to each other using *intents*. We'll also look at how you can use intents to **go beyond the boundaries of your app and make activities in other apps on your device perform actions.** Things just got a whole lot more powerful.



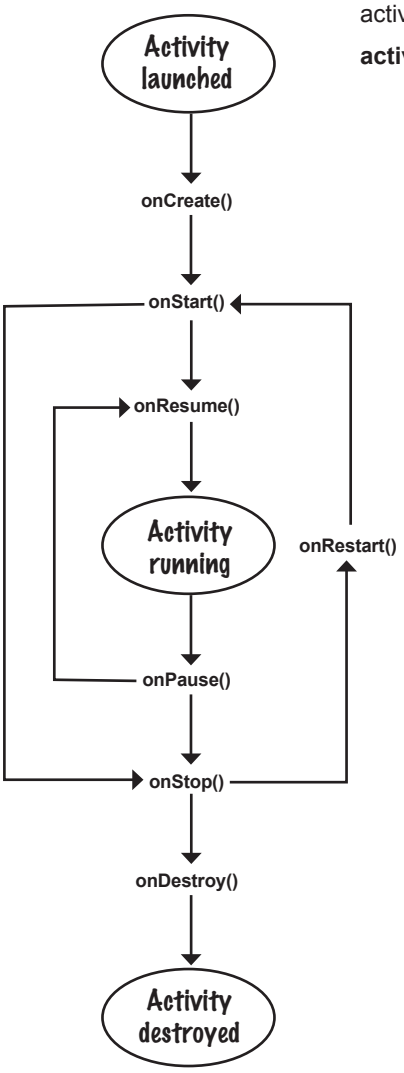


4 the activity lifecycle

Do one thing and do it well

Activities form the foundation of every Android app.

So far you've seen how to create activities, and made one activity start another using an intent. But *what's really going on beneath the hood?* In this chapter we're going to dig a little deeper into **the activity lifecycle**. What happens when an activity is ***created*** and ***destroyed***? Which methods get called when an activity is *made visible and appears in the foreground*, and which get called when the activity *loses the focus and is hidden*? And **how do you save and restore your activity's state**?

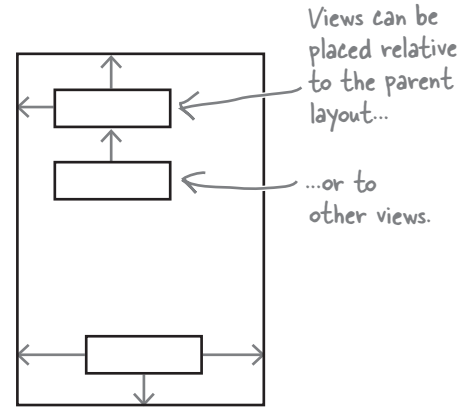


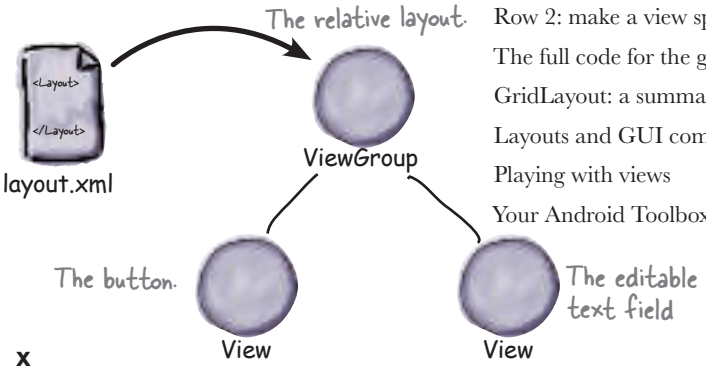
the user interface

5 Enjoy the View

Let's face it, you need to know how to create great layouts.

If you're building apps you want people to *use*, you need to make sure they **look just the way you want**. So far we've only scratched the surface when it comes to creating layouts, so it's time to *look a little deeper*. We'll introduce you to more **types of layout** you can use, and we'll also take you on a tour of the **main GUI components** and *how you use them*. By the end of the chapter you'll see that even though they all look a little different, all layouts and GUI components have ***more in common than you might think***.



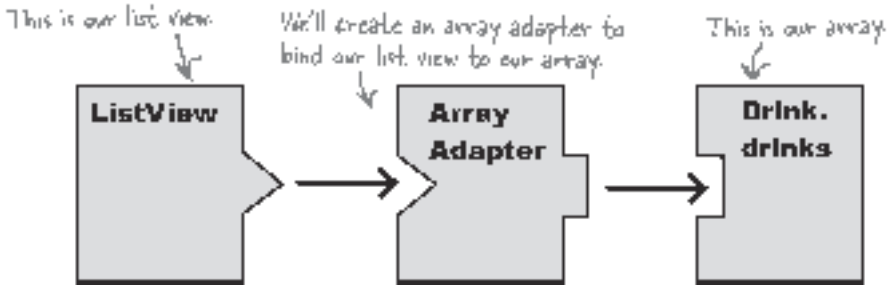


list views and adapters

6 Getting Organized

Want to know how best to structure your Android app?

You've learned about some of the basic building blocks that are used to build apps, and now it's time to get organized. In this chapter we'll show you how you can take a bunch of ideas and *structure them into an awesome app*. We'll show you how lists of data can form the core part of your app design, and how linking them together can create a *powerful and easy-to-use* app. Along the way you'll get your first glimpse of using event listeners and adapters to make your app more dynamic.

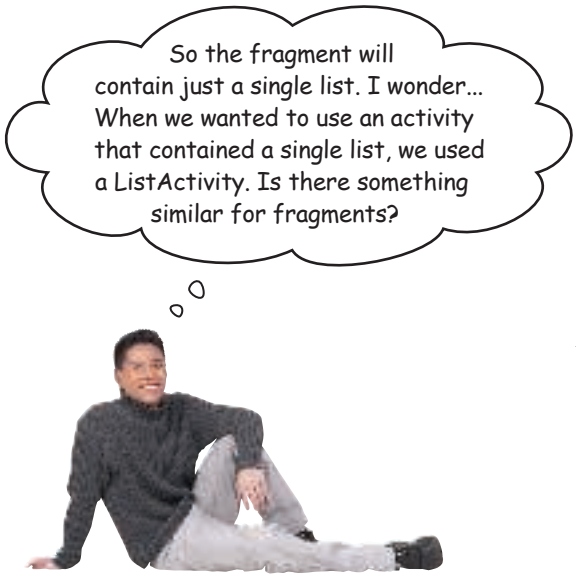


fragments

7 Make it Modular

You've seen how to create apps that work in the same way irrespective of the device they're running on.

But what if you want your app to *look and behave differently* depending on whether it's running on a *phone* or a *tablet*? In this chapter we'll show you how to make your app choose the **most appropriate layout for the device screen size**. We'll also introduce you to **fragments**, a way of creating *modular code components* that can be *reused by different activities*.

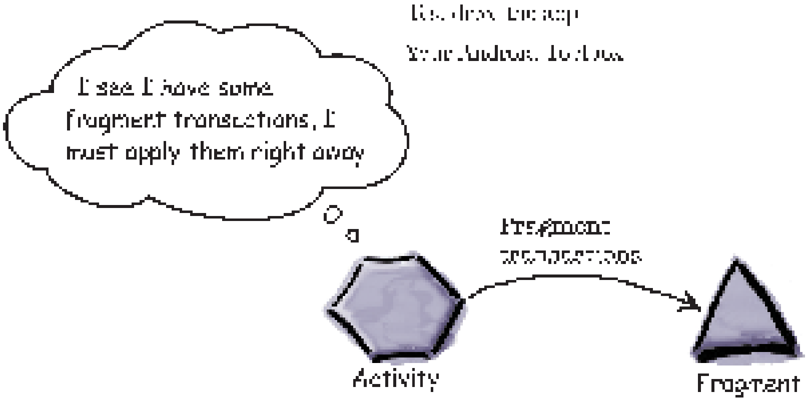


8 nested fragments

Dealing with Children

You've seen how using fragments in activities allow you to reuse code and make your apps more flexible.

In this chapter we're going to show you *how to nest one fragment inside another.* You'll see how to use the child fragment manager to tame unruly fragment transactions. Along the way you'll see why *knowing the differences between activities and fragments* is so important.

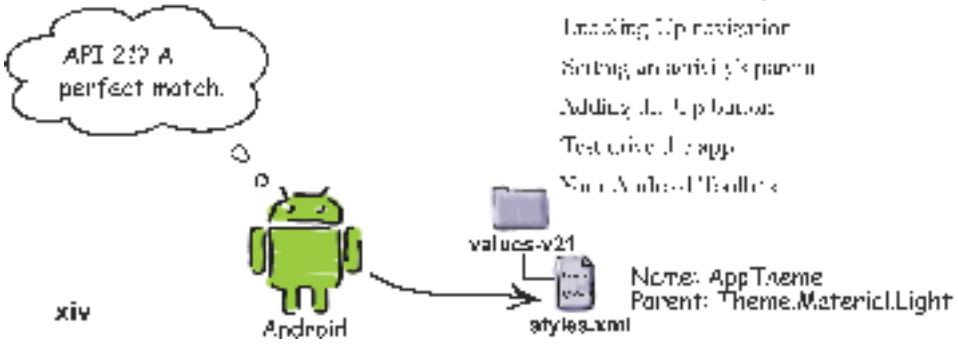


action bars

9 Taking Shortcuts

Everybody likes a shortcut.

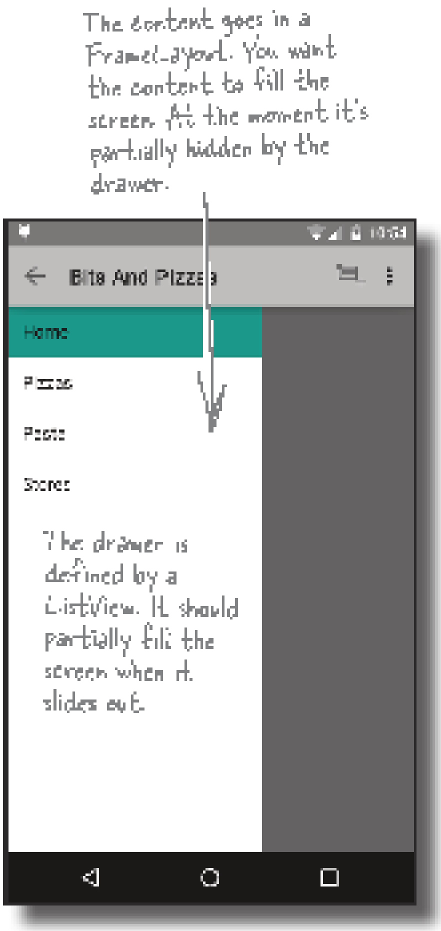
Are in this chapter you'll see how to add shortcuts to your apps using **action bars**. We'll show you how to start other activities by adding **action items** to your action bar, how to share content with other apps using the **share action provider**, and how to navigate up your app's hierarchy by implementing **the action bar's** Up button. Along the way you'll see how to give your app a consistant look and feel using **themes** and introduce you to the Android support library package.



10 navigation drawers

Going Places

Apps are so much better when they're easy to navigate. In this chapter we're going to introduce you to the **navigation drawer**, a slide-out panel that appears when you swipe the screen with your finger or click an icon on the action bar. We'll show you how to use it to display a *list of links* that take you to **all the major hubs of your app**. You'll also see how *switching fragments* makes those hubs **easy to get to and fast to display**.



11 SQLite databases

Fire up the Database

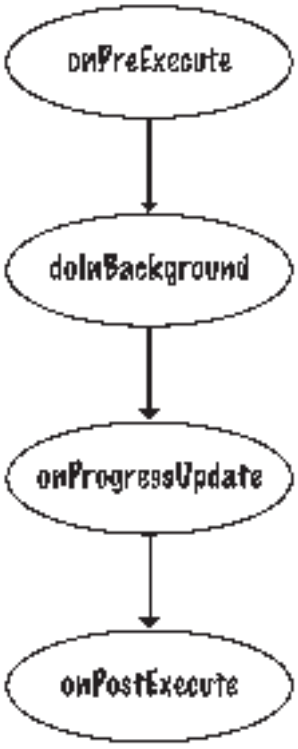
If you're recording high scores or saving tweets, your app will need to store data. And on Android you usually keep your data safe inside a **SQLite database**. In this chapter, we'll show you how to *create a database*, *add tables* to it, and *prepopulate it with data*, all with the help of the friendly **SQLite helper**. You'll then see how you can cleanly roll out *upgrades* to your database structure, and *how to downgrade* if you need to pull any changes.

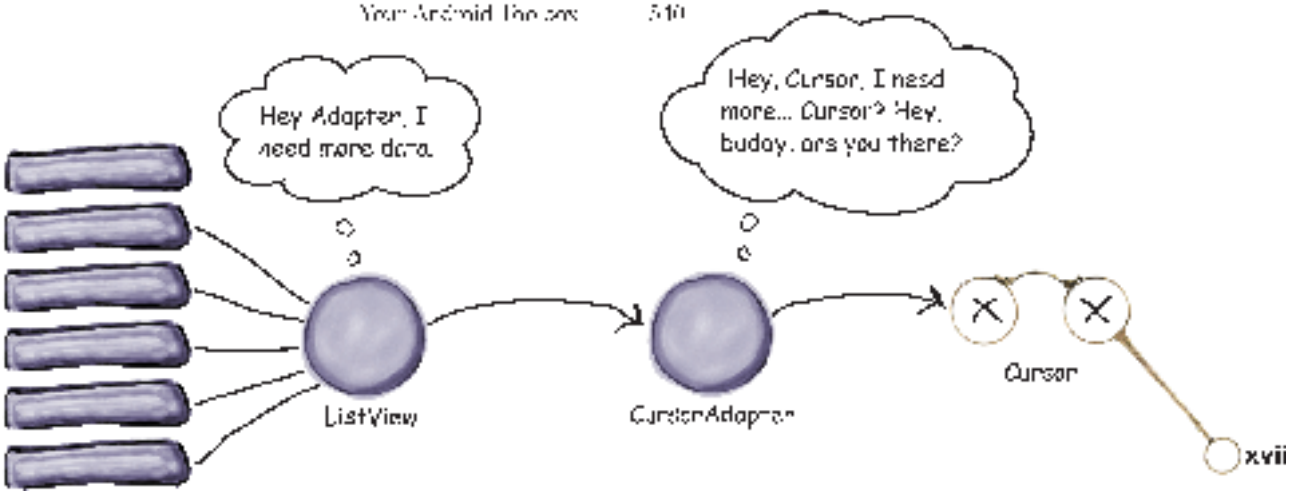
cursors and asynctasks

12 Connecting to Databases

So how do you connect your app to a SQLite database?

So far you've seen how to create a SQLite database using a SQLite helper. The next step is to get your activities to access it. In this chapter you'll find out how to use **cursors** to *get data from the database*, how to *navigate* cursors and how to *get data from them*. You'll then find out how to use **cursor adapters** to connect them to list views. Finally, you'll see how writing efficient *multi-threaded* code with **AsyncTasks** will keep your app speedy.



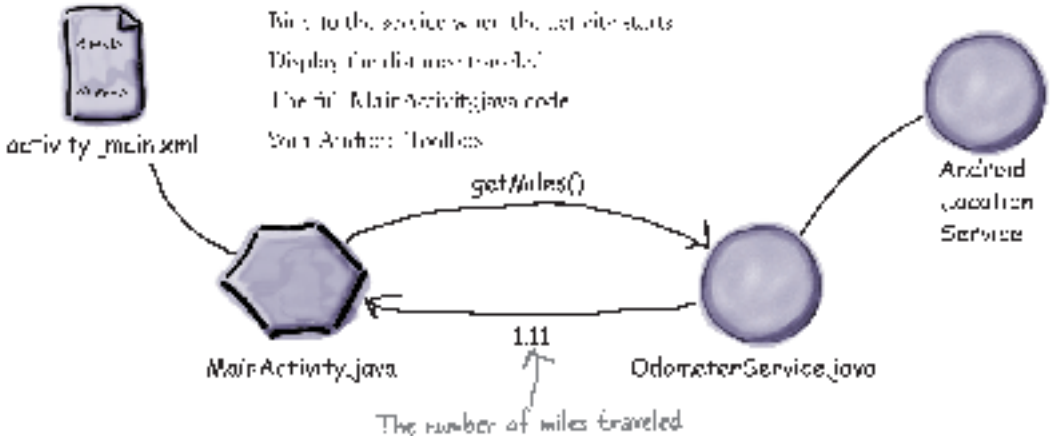


services

13 At Your Service

There are some operations you want to keep on running, irrespective of which app has the focus.

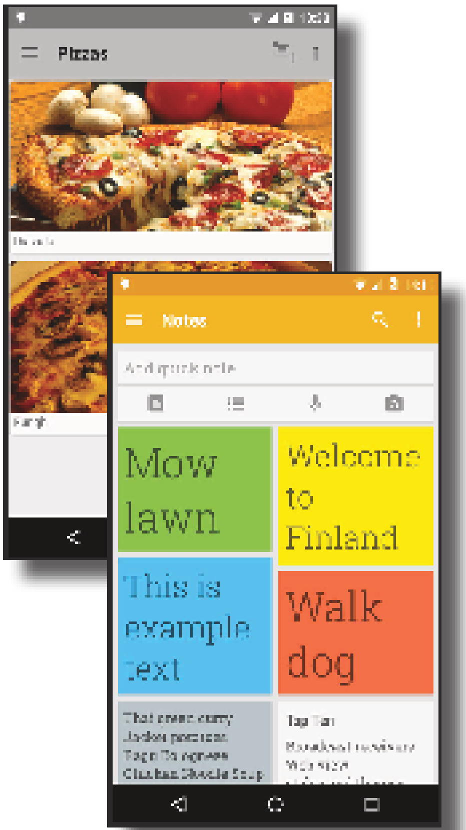
As an example: if you start playing a music file in a music app, you'd probably expect it to keep on playing when you switch to another app. In this chapter you'll see how to use services to deal with situations just like this. Along the way you'll see how use some of **Android's built-in services**. You'll see how to to keep your users informed with the *notification service*, and how the *location service* can tell you where you're located



14 material design

Living in a Material World

With API level 21, Google introduced Material Design. In this chapter we'll look at **what Material Design is**, and how to make your apps fit in with it. We'll start by introducing you to **card views** you can reuse across your app for a *consistent look and feel*. Then we'll introduce you to the **recycler view**, the list view's flexible friend. Along the way you'll see how to **create your own adapters**, and how to completely change the look of a recycler view with just *two lines of code*.



art

The Android Runtime

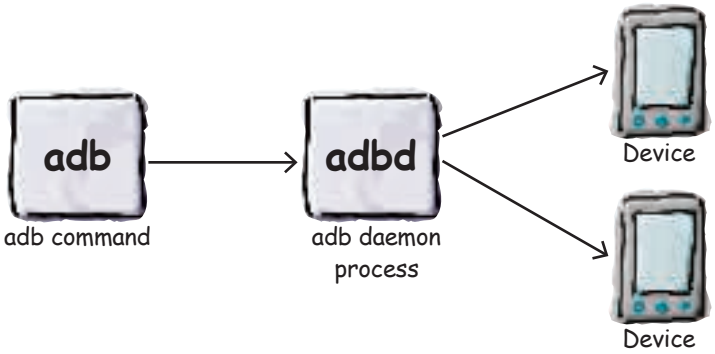
Android apps need to run on devices with low powered processors and very little memory.

Java apps can take up a lot of memory and because they run inside their own Java Virtual Machine (JVM), Java apps can take a long time to start when they're running on low-powered machines. Android deals with this by not using the JVM for its apps. Instead it uses a very different virtual machine called the Android Runtime (ART). In this appendix we'll look at how ART gets your Java apps to run well on a small, low-powered device.

adb

The Android Debug Bridge

In this book we've focused on using an IDE for all your Android needs. But there are times when using a command line tool can be plain useful, like those times when Android Studio can't see your Android device but you just *know* it's there. In this chapter we'll introduce you to the Android Debug Bridge (or adb), a command line tool you can use to communicate with the emulator or Android devices.



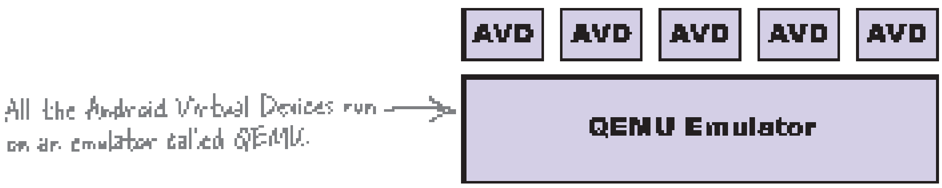
iii

the emulator

The Android Emulator

Ever felt like you were spending all your time waiting for the emulator?

There's no doubt that using the Android emulator is useful. It allows you to see how your app will run on devices other than the physical ones you have access to. But at times it can feel a little... sluggish. In this appendix, we're going to explain why the emulator can seem slow. Even better, we'll give you a few tips we've learned for **speeding it up.**



iv

leftovers

The Top Ten Things (we didn't cover)

Even after all that, there's still a little more.

There are just a few more things we think you need to know. We wouldn't feel right about ignoring them, and we really wanted to give you a book you'd be able to lift without extensive training at the local gym. Before you put down the book, **read through these tidbits.**