

A Brain-Friendly Guide

# Head First Android Development



Learn what  
matters, when  
it matters



Bend your mind  
around dozens of  
puzzles and exercises



Avoid  
embarrassing  
mistakes



Load important  
concepts directly into  
your brain



Master  
out-of-this-world  
concepts



O'REILLY®

Jonathan Simon

# Table of Contents (Summary)

	Intro	xi
1	Your first app: <i>Meet Android</i>	1
2	Give your app an action: <i>Adding behavior</i>	41
3	Pictures from space: <i>Work with feeds</i>	79
4	When things take time: <i>Long-running processes</i>	123
5	Run your app everywhere: <i>Multiple-device support</i>	
6	Tablets are not just big phones: <i>Optimizing for tablets</i>	
7	Building a list-based app: <i>Lists and adapters</i>	167
8	Navigation in Android: <i>Multi-screen apps</i>	205
9	Database persistence: <i>Store your stuff with SQLite</i>	265
10	RelativeLayout: <i>It's all relative</i>	313
11	Give your app some polish: <i>Tweaking your UI</i>	345
12	Make the most of what you can use: <i>Content proficers</i>	393
i	Leftovers: The <i>Top Ten Things (We Didn't Cover)</i>	

# Table of Contents (the real thing)

## Your First App

### So you're thinking: "What makes Android so special?"

Android is a **free and open operating system** from **Google** that runs on all kinds of devices from **phones**, to **tablets** and even **televisions**. That's a ton of different devices you can target with *just one platform!* (And the market share is gaining too!) Google provides all of the stuff you need to get started building Android apps **for free**. You can build your Android apps on Macs, Windows, or Unix and publish your apps for next to nothing (with no need for anyone's approval). Ready to get started? Great! You're going to start building your first Android app, but first there are a few things to set up...

meet android

1

## Your first app

**So you're thinking: "What makes Android so special?"**

Android is a **free and open operating system** from *Google* that runs on all kinds of devices from **phones**, to **tablets** and even **televisions**. That's a ton of different devices you can target with *just one platform!* (And the market share is gaining too!) Google provides all of the stuff you need to get started building Android apps **for free**. You can build your Android apps on **Macs**, **Windows**, or **Unix** and publish your apps for next to nothing (with no need for anyone's approval). Ready to get started? Great! You're going to start building your first Android app, but first there are a few things to setup...

adding behavior

2

## Give your app an action

**Apps are interactive!** When it comes to apps, it's what your *users can do* with your apps that make them love 'em. As you saw in Chapter 1, Android really *separates* out the **visual definition** of your apps (remember all that XML layout and String resource work you just did!) **from the behavior** that's defined in *Java code*. In this chapter, you're going to **add some behavior** to the AndroidLove haiku app. And in the process you'll learn how the XML resources and Java work seamlessly together to give you a great way to build your Android apps!

work with feeds

3

## Pictures from space!

**RSS feeds are everywhere!** From weather and stock information to news and blogs, huge amounts of content are distributed in RSS feeds and just waiting to be used in your apps. In fact, the RSS feed publishers want you to use them! In this chapter, you'll learn how to build your own app that incorporates **content** from a public RSS feed on the Web. Along the way, you'll also learn a little more about **layouts**, **permissions**, and **debugging**.

long-running processes

4

## When things take time

It would be great if everything happened instantly. Unfortunately, some things just take time. This is especially true on mobile devices, where network latency and the occasionally slow processors in phones can cause things to take a *bit* longer. You can make your apps faster with optimizations, but some things just take time. But you *can* learn how to **manage long-running processes better**. In this chapter, you'll learn how to show active and passive status to your users. You'll also learn how to perform expensive operations off the UI thread to guarantee your app is always responsive.

multiple-device support

5

## Run your app everywhere

There are a lot of different sized Android devices out **there**. You've got big screens, little screens, and everything in between. And it's your job to support them all! Sounds crazy, right? You're probably thinking right now "*How can I possibly support all of these different devices?*" But with the right strategies, you'll be able to target all of these devices *in no time* and with **confidence**. In this chapter, you'll learn how Android classifies all of these different devices into groups based on **screen size** as well as **screen density**. Using these groups, you'll be able to make your app look great on all of these different devices, and all with a **manageable** amount of work!

optimizing for tablets

6

## Tablets are not just big phones

**Android tablets are coming onto the scene.** These new larger-format Android devices give you an entirely new hardware format to present new and cool apps to your users. **But they are not just big phones!** In this chapter, you'll learn how to get your app up and running on a tablet. You'll learn about the new screen size groupings and also how to use Fragments to combine multiple Activities on a single screen. So more importantly than just running on tablets in this chapter, you'll learn about how to **make your app work better** on them.

lists and adapters

7

## Building a list-based app

**Where would we be without lists?** They display read-only information, provide a way for users to select from large data sets, or even act as navigational device by building up an app with a list-based menu structure. In this chapter, you'll learn how to build an app with a list. You learn about where lists store data (in Adapters) and how to customize how that data is rendered in your list. You'll also learn about adding additional layouts to your app (not just the layout that the Wizard creates for you) and turn that into a real view.

multi-screen apps

8

## Navigation

**Eventually you'll need to build apps with more than one screen..** So far, all of the apps you've built only have a single screen. But the great apps you're going to build may need more than that! In this chapter, you'll learn how to do just that. You'll build an app with a couple of screens, and you'll learn how to create a new Activity and layout which was previously done for you by the Wizard. You'll learn how to navigate between screens and even pass data between them. You'll also learn how to make your own Android context menu- the menu that pops up when press the Menu button!

database persistence

9

## Store your stuff with SQLite

**In memory data storage only gets you so far.** In the last chapter, you built a list adapter that only stored data in memory. But if you want the app to **remember** data between sessions, you need to **persist** the data. There are a few ways to persist data in Android including writing directly to files and using the built in SQLite database. In this chapter, you'll learn to use the more robust SQLite database solution. You learn how to create and manage your own SQLite database. You'll also learn how to integrate that SQLite database with the ListView in the TimeTracker app. And don't worry, if you're new to SQL, you'll learn enough to get started and pointers to more information.

RelativeLayout

## It's all relative

10

You've created a few screens now using **LinearLayouts (and even nested LinearLayouts)**. But that will only get you so far. Some of the screens you'll need to build in your own apps will need to do things that you just can't do with **LinearLayout**. But don't worry! Android comes with other layouts that you can use. IN this chapter, you'll learn about another super powerful layout called **RelativeLayout**. This allows you to layout Views on screen relative to each other (hence the name). It's new way to layout your Views, and as you'll see in the chapter, a way to optimize your screen layouts.

UIKit

## Giving your app some polish

11

With all the competition in the marketplace, your apps must do more than just work. They have to look great doing it! Sometimes, basic graphics and layouts will work. But other times, you'll need to crank it up a notch. In this chapter, you'll learn about a new layout manager called **RelativeLayout**. It'll let you lay out your screens in ways that you just can't do with **LinearLayout** and help you code your designs just the way you want them. You'll also learn more techniques for using images to polish up the look and feel of your app. Get your app noticed!

Content Providers

## Make the best of what you can use

12

You don't want to reinvent the wheel, do you? Of course you don't; you've got apps to build! Well, one of the awesome benefits of Android is the ease in which you can use bits of other applications with content providers. Android apps can expose functionality they want to share and you can use that in your apps. But this doesn't work only for market apps; a number of built-in apps (like the Address Book) expose stuff you can use in your apps too. In this chapter, you'll learn how to use content providers in your app. And who knows, you might like this whole content provider thing so much, you'll decide to provide some of your own content to other apps!