**Resources for Android App Development**

**Amy Funk**

**http://bit.ly/2dwLyhA**

**Objective:**

The objective of this white paper is to give developers a resource to build a working android app.  Currently Android occupies 87.6%[(1)](http://www.idc.com/prodserv/smartphone-os-market-share.jsp) of the smartphone OS market and has the one of the largest app marketplaces in the world making it a lucrative and profitable environment for developers. The following information will give you a basic understanding of Android development and show you resources to develop your own app.

**Basic Understanding:**

The Android OS is built upon the [Java](https://en.wikipedia.org/wiki/Java_%28programming_language%29) developed by Google as an open source smartphone OS.  Android works across multiple devices which include smartphones, tablets, wearables, and a number of other devices.[(2)](https://en.wikipedia.org/wiki/Android_(operating_system))

The resources below assume that the reader has an understanding of Java and has used it to program before.

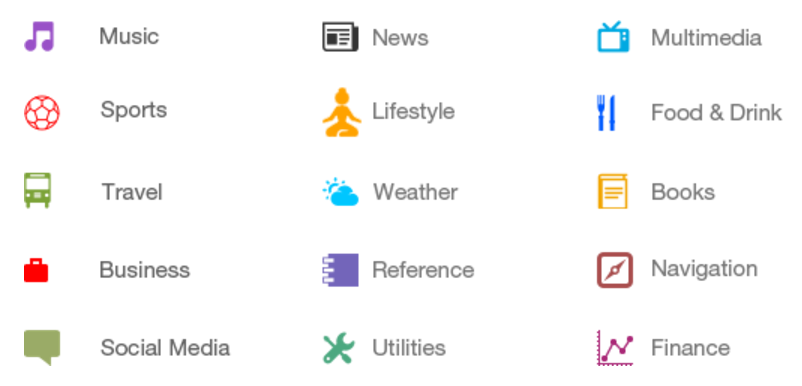
**Operating Systems you can develop Android applications on:**

* Microsoft Windows XP or later version
* Mac OS X 10.5.8 or later version
* Linux including GNU C Library 2.7 or later

**Software needed to start you Android application programming:**

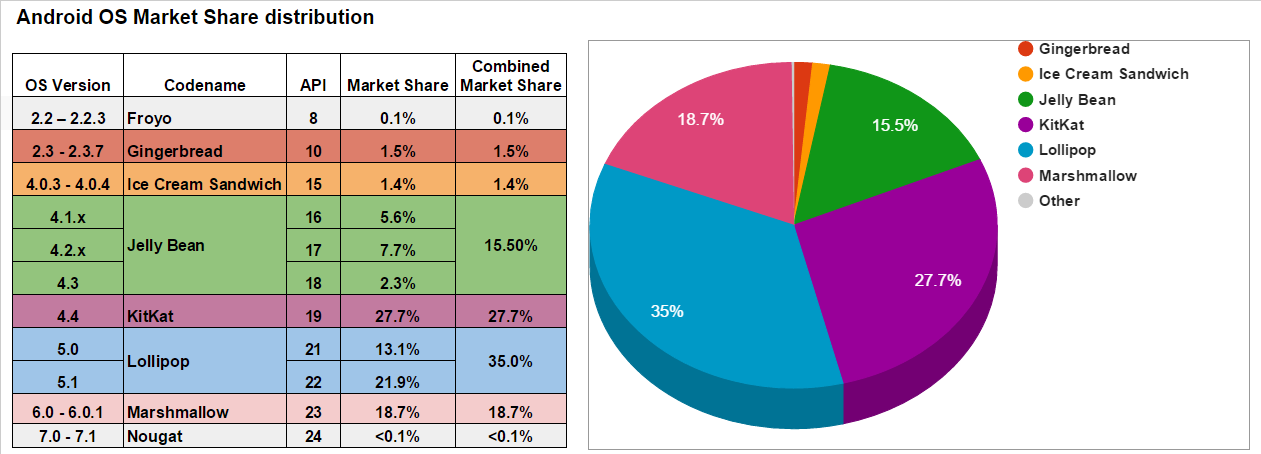
* Java SE 5 or later version
* Android SDK
* Java Runtime Environment (JRE) 6
* Android Studio [Link of download](https://developer.android.com/studio/index.html)
* Eclipse IDE for Java Developers (optional) [Link for download](http://www.eclipse.org/)
* Android Development Tools (ADT) Eclipse Plugin (optional)

**Categories of Android Apps:**

****

**Android OS Market share and why you care**

One of the fundamental decisions to make for an Android developer is which of the many editions of the Android OS their app is going to support.  Each additional OS version adds more potential customers, but also adds design complications and expands the testing matrix.



[**Data collected by Google from the devices that visited Playstore in the 7 days preceding September 5th, 2016**](https://developer.android.com/about/dashboards/index.html)

The Android dev site has a [tutorial](https://developer.android.com/training/basics/supporting-devices/platforms.html) on how to support multiple OS versions in your app. Most discussion boards on this topic point to 4.0 or 4.1 as a reasonable minimum OS version.  Do your research as these numbers change over time. Also consider if there are any particular requirements for the app you are designing that would dictate a minimum version.  As an example, [Bluetooth low energy](http://en.wikipedia.org/wiki/Bluetooth_low_energy) used by many fitness bands was only supported in Android OS v 4.3 and above.

**Android OS Tutorials**

As the Android OS has taken over the lion’s share of the mobile app market, a vast array of teaching resources has been created to help developers get up to speed.  These learning tools come in many different styles, from basic step-by-step“for dummies” types to very high level “jump start” courses that assume journeyman developer knowledge.  Fortunately there are many to choose from so you should be able to find a tutorial that fits your experience level and learning style.

Here are some popular free Android tutorials for you to consider-

* Google’s [Android developer site](https://developer.android.com) has a very extensive [Training](https://developer.android.com/training/index.html) section with guides that take you from your first [“Hello World” app](https://developer.android.com/training/basics/firstapp/index.html) to using the many sophisticated mobile features that Android has to offer.
* [Android Developers YouTube channel](https://www.youtube.com/user/androiddevelopers)  These videos are produced by the Android developers at Google, so they cover all of the latest features and how to use them.  Videos in the [Google I/O 2013 - The Android Sessions](https://www.youtube.com/playlist?list=PLWz5rJ2EKKc9WGUwq2gQ-coU3fSyexgOx) from the 2013 Google I/O conference give detailed introduction to the Android Studio IDE and other topics.   The [Android Design in Action](https://www.youtube.com/playlist?list=PLWz5rJ2EKKc8j2B95zGMb8muZvrIy-wcF) series goes more in depth with longer videos covering various aspect of Android design.
* [Lynda Training Videos](http://www.lynda.com/Android-training-tutorials/947-0.html?utm_medium=ldc-partner&utm_source=SSPRC&utm_content=524&utm_campaign=CD15737&bid=524&aid=CD15737) Lynda’s Android courses are highly rated.  They have 15 Android related courses, consisting of over 600 videos.  These range from “getting started” primers to complete examples of building specific types of apps like games or navigation.  There are exercises included with many of these courses.  Though Lynda is paid subscription service, Seattle and King County residents have free access through their library system.
* [Material Design at Sitepoint](https://www.sitepoint.com/material-design-android-design-support-library/)  Tutorial for learning [Material Design](https://developer.android.com/design/material/index.html), a new design language that gives design guidelines for Android apps and apps on other platforms, was introduced with the release of Android 5.0 Lollipop.
* [Harvard Extension School - Building Mobile Apps](https://www.youtube.com/playlist?list=PLWz5rJ2EKKc9WGUwq2gQ-coU3fSyexgOx) This program features [OpenCourseWare](https://en.wikipedia.org/wiki/OpenCourseWare)  instruction on building mobile apps.  There are 5 lectures specific to Android, but it also covers iOS and some Windows Mobile.  These videos and associated courseware can be downloaded for offline use.
* [Android Adventures - Getting Started with Android Studio 2](http://www.i-programmer.info/programming/android/5887-android-adventures-getting-started-with-android-studio.html) Written by Mike James, this tutorial video focuses on helping you get up-and-running in the latest [Android Studio](https://en.wikipedia.org/wiki/Android_Studio) IDE.
* [Udemy: Android Video Tutorials](http://click.linksynergy.com/fs-bin/click?id=9Mp0e/rCunU&subid=&offerid=323058.1&type=10&tmpid=14538&RD_PARM1=https%3A%2F%2Fwww.udemy.com%2Ftopic%2FAndroid-Video-Tutorials%2F) Udemy offers both free and paid instructional videos, but they offer some excellent Android courses at no cost.
* [Derek Banas: Android Development For Beginners](https://www.youtube.com/playlist?list=PLGLfVvz_LVvSKgnFm8-6Fz1cd6zt_KxTC) YouTube videos introducing [MIT’s App Inventor](http://appinventor.mit.edu/) which is a tool to let even non-coders build Android Apps.
* [Android Application Security Series](https://manifestsecurity.com/android-application-security/) by [Aditya Agraw](https://disqus.com/by/exploitprotocol/) Tutorial that deals exclusively with security issues, solutions and testing approaches for the Android OS.
* [HMKCode Android Tutorials](http://hmkcode.com/android-tutorial/) Another free Android tutorial, HMKCode includes an associated [Git repository](https://github.com/hmkcode/Android) of code related to their course.

**Android Libraries, Examples**

“Embrace and Extend” is an important concept in all development today.  One of the best ways to learn a new technology is to grab some sample code and start playing with it.  With Android there are volumes of example code available, and myriad libraries available to extend your functionality.

Here are a few useful places to go for Android sample code and libraries you may want-

* [Codepath/android\_guides Must Have Libraries](https://github.com/codepath/android_guides/wiki/Must-Have-Libraries) Excellent source for those Android libraries that you really should have, or at least be aware of.  These libraries are grouped into Standard and Advanced, so if you are just getting started you probably won’t want to include the Advanced set.
* [HMKCode Android Repository](https://github.com/hmkcode/Android) This Git repository relates to the [HMKCode Android tutorials](http://hmkcode.com/android-tutorial/), and has a wide variety of useful sample code that supports the training course.
* [ZXing](https://github.com/zxing/zxing) is a library of barcode and qr code functionality, very mature and supporting many platforms.  If you need to read or print barcodes of any kind these libraries will help.
* [Libphonenumber](https://github.com/googlei18n/libphonenumber) A Google team library that is a comprehensive set of tools for parsing, validating, and formatting phone numbers.
* [DBFLow](https://github.com/Raizlabs/DBFlow) Another Git repository that is focused on [ORM](https://en.wikipedia.org/wiki/Object-relational_mapping) Android database library.
* [Sanfoundry Android Examples](http://www.sanfoundry.com/java-android-programing-examples/) This page lists over 100 examples of Android coding solutions.  Each example includes a description of what it does, the actual code, an explanation of how it functions and example output if applicable.

**Android Developer Cheat Sheets**

Developing Android apps, and mobile apps in general, requires being aware of many considerations and concepts that may not be familiar to the “non-mobile” developer.  Supporting multiple screen sizes, orientations, view types, action bars, manifests etc. means keeping track of many detailed bits of information - sounds like a job for a “Cheat Sheet”.  There are many examples of Android cheat sheets available on the web, from general purpose to those aimed at a specific topic.  You will want to find those that work well for you, or that you can “extend” to meet your needs.

Here are a few examples of Android related Cheat Sheets -

* [The Android Design Cheat Sheet](https://possiblemobile.com/2014/01/android-design-cheat-sheet/), focuses on visual design specific information like device sizes, UI element specifics, padding details and more.
* [Android Testing Cheat Sheet](https://www.owasp.org/index.php/Android_Testing_Cheat_Sheet), is more a website than a true cheat sheet, but it focuses on step-by-step security testing for your Android App.
* [Android App Development Cheat Sheet](https://docs.google.com/spreadsheets/d/1Iv2DD18vl8JnDz8Mblo_mI2tBoOBKUh-9eKJuMCiEjU/edit), a Google Sheet that contains all the resources collected by one developer during his Android journey.  Easily extensible!
* [Android Cheatsheet for Graphic Designers](http://petrnohejl.github.io/Android-Cheatsheet-For-Graphic-Designers/), a website that contains VERY detailed dimensions for all the various bits of Android UI.  Covers rules for images and themes, including [nine-patch drawables](https://developer.android.com/reference/android/graphics/NinePatch.html), and covers naming conventions for graphic elements and their states.
* [iOS and Android Design Guidelines Cheat Sheet](https://www.kinvey.com/ios-and-android-design-guidelines-cheat-sheet/), High level side-by-side comparison of UI considerations for iOS vs. Android. Compares UI elements, screen sizes,  Icon sizes, and touch gestures.  Would be very handy if developing an App for both mobile operating systems.
* [Publishing Android Applications](http://www.tutorialspoint.com/android/android_publishing_application.htm), a website that shows the entire processes of getting your Android app published. Shows proper dev lifecycle for Android app.  Covers the process of signing and exporting an [Android Package (APK)](https://en.wikipedia.org/wiki/Android_application_package).
* [Java Programming Cheat Sheet](http://introcs.cs.princeton.edu/java/11cheatsheet/) provided from the Introduction to Programming in Java course at Princeton University. The appendix summarizes the most commonly-use Java language features.

**Android User Groups and Blogs**

Your best resource for help with Android developing will always be other Android developers.  As a collective, they have already tackled many of the most thorny problems in mobile development, and are typically willing to share that knowledge if approached respectfully.

Important Android Groups and Blogs -

* [Android Developer Community](https://plus.google.com/communities/105153134372062985968/) is a Google+ group with over a quarter million members and describes itself as the “official” Android development community.
* [Android Official Blog](https://blog.google/products/android/)  All things Android, not just how to build apps. Managed by members of the Android team at Google
* [Android Development Blog](http://android-developers.blogspot.com/)  Blog devoted to just Android development, also curated by Android team.
* [Androiduiux](https://androiduiux.com/) is a blog that focuses on User Interface and User experience issues and solutions.
* [Android Weekly](http://androidweekly.net/) is a free eNewsletter that “helps you stay cutting-edge” as a ‘Droid dev.
* [Android Forums](http://androidforums.com/) claims to be the “first and largest community dedicated to Android.

**Citations**

1. IDC: Smartphone OS Market Share. (n.d.). Retrieved October 06, 2016, from http://www.idc.com/prodserv/smartphone-os-market-share.jsp
2. Android (operating system). (n.d.). Retrieved October 06, 2016, from https://en.wikipedia.org/wiki/Android\_(operating\_system)

**Questions:**

If you want to reach at least 75% of the market share what is the minimum version you should support for your app?  
4.4 and up.

Isn’t it hard to develop apps for the Android OS?

Well “hard” is a relative term.  There are many Android specific details that need to be understood to build successful apps.  However there are terrific resources to help you succeed. Google provides tremendous support, including their Android Studio IDE.  Many forms of online instruction and examples are available and there are robust users groups supporting the Android environment to help get your questions answered.  So if you are interested in being a ‘Droid dev, it is probably easier than you think.

Why wouldn’t you want to develop apps for Android OS?

The Android market is fragmented by multiple types of OS, slower OS updates to phones also.  Generally more money is made through IOS apps.