1. Android’s features
   1. URIs/URLs
      1. <https://www.gazelle.com/thehorn/2014/02/10/the-android-operating-system-10-unique-features/>
      2. <https://www.elprocus.com/what-is-android-introduction-features-applications/>
      3. <https://developer.android.com/about/index.html>
   2. Android OS is an open-source piece of software that, due to being open-source, is very customizable and hackable. Some features include third-party-apps having permission to control external connections and hardware (some apps can control and/or automate everything from NFC to GPS services to the flashlight on the back of the phone), adding custom ROMS (to change appearance and grant even more third-party permissions), and widgets on the home screen.  
      The basis of Android is the Linux Kernel. Users (typically) only interact with the part of the architecture that is NOT the kernel which in turn go through the virtual machine and then to Linux.

One other major feature of Android is the Google Play application marketplace. Each *month*, Google Play delivers “1.5 billion downloads . . . and [that number is] growing.” Combined with the wide variety of existing use, Google has lowered barriers for their store (that companies like Apple have, such as relatively high costs for membership) to permit developers to publish their work more easily.

* 1. The third most important point detailed in what I read is that Android runs on top of Linux. This gives it enough simplicity while still maintaining power over its hardware.

The second most important takeaway was that of the Google Play store. Its ease-of-entry combined with being preinstalled on Android devices allows both developers and consumers to exchange services and publicity. If the Google Play store were designed such that it was much more difficult to use within Android, then Android would not have the success it sees today, due partially to the Apple app store being its direct competition.

Certainly, the biggest idea proposed in the articles was that of user customizability. No other mobile phone OS (and very few, if any, non-mobile OSs) give the user such an ability to customize the experience as Android. From top-level home-screen launchers to custom ROMs to automating when the Wi-Fi connection is enabled using an app from the Play store, customizability gives Android a major distinguishing factor.

1. XML and Java
   1. URIs/URLs
      1. <http://www.zipcodewilmington.com/blog/i-want-to-develop-android-apps-what-languages-should-i-learn>
      2. <http://www.techotopia.com/index.php/Creating_an_Android_User_Interface_in_Java_Code>
      3. <https://www.safaribooksonline.com/library/view/head-first-android/9781449362171/ch01.html>
   2. Android development it done with Java and XML. Java code controls the brains and the information of the application, while XML determines its façade and appearance. Put another way, the Java code dictates how the app should function, and the XML code should dictate how it looks while it’s functioning. Java code integrates with XML so that they work together to make a complete, visually-balanced application.

Design and styling can be done with Java (alongside the functionality), however it is much more time-efficient to use XML. Languages other than Java can be used to encode the functionality for Android apps (such as C or C++ with the Android Native Development Kit), though they tend to not be as generalized and as compatible as Java-written apps.

* 1. A very important idea to realize in Android coding is that functionality and appearance of an application can be made in the same file, but separation provides for more clear and concise code.

Even more important is the use of XML and Java in tandem. Splitting the functionality from style is itself inherently helpful, but Java and XML help even more. Java’s bytecode allows for generalized content (because of how Java works, with its own runtime environment, combined with Android’s overlay, the Android Runtime Environment (ART)), and XML is a markup language that allows for creation of personalized elements; these qualities synergize rather well.

1. Superior design
   1. URIs/URLs
      1. <https://www.smashingmagazine.com/2011/06/designing-for-android/>
      2. <https://developer.android.com/design/index.html>
      3. <https://www.computerworld.com/article/2909897/material-design-apps-android.html>
   2. Currently, many Android app designers use a set of design guidelines known as “Material Design.” Within the guidelines, vibrant, solid (non-gradient) colors are often used with simple geometric figures.

Lots of applications have a menu accessed by either touching the icon depicted by three parallel bars or swiping in from the left side of the screen; a lot of navigation can be performed from here. A similar amount of applications also have a menu with three dots, aligned vertically, in the top-right corner of the screen. This is similar to the “File” menu in many word processors/general computer applications.

Some acknowledged “best practices” include maintaining visual consistency throughout all usage aspects of the app, making major actions clear and concise, and making every animation/response system seem natural and intentional.

* 1. Material design is certainly the most important task to implement in application design, when it comes to making the app familiar. Its high usability combined with visual appeal to a wide audience make it an exceptional tool that is available to the arsenal of Android app development.

Following material design, it is most important to follow, rather closely, the current status quo for app design. The status quo can be challenged, but following it provides for a less risky chance at success. Guidelines that are currently commonly used are menus accessible from the top corners of the screen as well as fluid animations and a general fondness of contrastive colors.