What is Android?

- Originally developed by Android, Inc. in 2003 as an advanced operating system for digital cameras
- Android was acquired by Google in 2005 with intention to develop it into a mobile phone operating system
- Open source operating system for mobile devices and consumer appliances
  - Linux-based
  - Every app has its own user ID for security reasons
- Apps are developed using Java programming language
  - Java 6 fully supported
  - Java 7 partially supported (Fully supported in Lollipop)
  - Java 8 not yet supported
  - Uses Google’s proprietary Dalvik VM, ART runtime in Lollipop
  - Native programming in C using NDK
## Android Versions

<table>
<thead>
<tr>
<th>Version</th>
<th>API Level</th>
<th>Name</th>
<th>Where used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 - 2.3</td>
<td>1 - 10</td>
<td>“A”, “B”, Cupcake, Donut, Eclair, Froyo, Gingerbread</td>
<td>Phones</td>
</tr>
<tr>
<td>3.0 - 3.2</td>
<td>11 - 13</td>
<td>Honeycomb</td>
<td>Tablets</td>
</tr>
<tr>
<td>4.0 - 4.4</td>
<td>14 - 20</td>
<td>Ice Cream Sandwich, Jelly Bean, KitKat</td>
<td>Phones &amp; Tablets</td>
</tr>
<tr>
<td>5.0 - 5.1</td>
<td>21 - 22</td>
<td>Lollypop</td>
<td>Phones, Tablets, Wearables, Automotive, TV</td>
</tr>
</tbody>
</table>
What does the Android SDK Provide?

- UI Framework
- Animations, 2D & 3D Graphics
- Computation
- Media and Camera
- GPS, Compass, Accelerometers, Light sensors
- Bluetooth, WiFi, NFC, etc.
- Text Input
- Data Storage
  - Database (SQLite)
  - File Storage
  - Preference Storage
- Browser-Based Apps
  - HTML5 / Javascript
- Too many more to list here
What skills do I need to get started?

- Java
  - OOP Concepts
    - Class Inheritance - Extending existing designs
    - Polymorphism - Different objects with similar behaviors
    - Interfaces
  - Threads - Independent contexts of execution, possibly on different processor cores

- Event-Based Frameworks
  - Registering Callbacks with Interfaces
  - Quick Execution - Handle UI event quickly and return to event loop.
    - Spawn new threads when necessary.
Basic Android App Components

- **Activity**
  - Main user interface component
  - Much like a window in a desktop environment
- **Service**
  - No UI component
  - May or may not work on a separate thread
- **Intent Filter**
  - Provides a means to invoke an app component based on user action
  - Allows apps to make use of functions provided by other apps
  - Can be used to invoke an Activity or a Service
- **Fragment**
  - Provides a means of making the UI more modular
- **Content Provider**
  - Allows a standardized way for apps to access data on the device
  - Provides access to contacts, photos, databases, etc.
- **Many, many more components and subcomponents.**
Activity

- Primary means of interacting with the user
  - Somewhat similar to a window in a desktop environment
  - Activities display a UI and handle events from the UI elements
  - User interface appearance is defined using XML layouts
- Each app may have several Activities
  - Examples: List messages, Read message, Compose message
  - Activities are stacked in a process
    - Different stacking behaviors can be defined
    - Activity objects are typically re-used unless otherwise specified
- Activities are invoked using Intent Filters
  - User clicks on icon on home screen
  - User opens a file of a particular type
  - User clicks on a link to a particular type of resource
  - An app invokes the Activity through a request for a particular function (i.e., choose photo)
  - User clicks on a notification
Service

- Provide a means of centralizing functions of an app that don’t need a UI
  - Background operations or maintaining a state
  - Access to underlying database
  - Handling communication to a server
  - Performing tasks that require extensive calculation
  - Handling periodic real-time system updates (i.e., games)
- May or may not incorporate a separate thread of execution
- May continue to operate without a user interface
- Can be invoked from an Activity or through other means
  - Broadcast listeners
  - Timers
  - External network contact
Intent and Intent Filter

- **Intents**
  - Invoke an Activity from the home screen
  - Signal a timer event or other system event
  - Open a file
  - Specify what happens when a notification is clicked (PendingIntent)
  - Can be invoked by system, apps, services, notifications, etc.
  - Primary means of interprocess communication

- **Intent Filters**
  - Specify what types of Intents will trigger a certain action
  - Can specify action type, content type, file type, URL prefix, etc.
- Building blocks of a more complex UI
- Provide a means of encapsulating both UI and underlying functionality
- Can be used to as components within an Activity UI or as a Dialog
- Allow reuse of UI design in different form factors, i.e., phone and tablet
  - Email message list and message view in same Activity on tablet
  - Email message list and message view in different Activities on phone
- Can be conceptualized of as a sort of “Activity within an Activity”
Basic App Structure

- Manifest (XML) - provides information about your app to the OS
  - Activities, Services, Broadcast Listeners, etc
  - Intent Filters
  - Security Permissions
- Code
  - Java Source Code
  - Libraries (.jar)
- Resources
  - Layouts (XML)
    - UI Design and Appearance
  - Menus (XML)
  - Values (XML)
    - String constants, IDs, Color values, etc.
  - Graphics (PNG, JPG, etc)
  - Misc (XML)
  - Assets (raw data files)
Android Development Tools

- Eclipse
  - Original Development IDE for Android
  - No longer supported officially by Google
  - Recommended only if you have to or prefer to use Eclipse

- Android Studio
  - Based on IntelliJ IDE
  - Officially and actively supported by Google
  - Windows, Linux, and Mac
  - WYSIWYG user interface design with Drag/Drop UI components
  - Android-specific refactoring and auto code completion
  - Lint tools
  - Template wizards
  - Support for Android Wear
  - Built-in support for Google Cloud Messaging and App Engine
Android Studio - Layout Editing

XML Editing View
Android Studio - Layout Design

- Component Palette
- Component Hierarchy
- Layout Design View
- Component Parameters
Android Debugging

Debugging Tools:

- Android SDK Emulator
- 3rd Party Emulator (Example: Genymotion)
- On-Device Debugging with ADT (Recommended)
Useful References

- Android Developer Page
- Stack Overflow
- Vogella Tutorials - Android Development
- Android Asset Studio