

# Intro to Coding for Android

Todd DeLand  
Ronald Daugherty  
Arbormoon Software, Inc.

# Audience Background

- Any current developers?
- Prior development experience?
- Students?
- IT professionals?
- Others?

# About Todd DeLand

- Software Engineer - 11 years
  - PHP
  - .NET
  - .NET Micro
  - Demandware
  - .NET
  - Android
- GDG Ann Arbor Co-organizer
- Tech Hobbyist (this is really important!)
- Father of three
- Husband
- Skiing
- Hiking
- Gardening
- TVing

Twitter: @todddeland

# About Ronald Daugherty

- Software Developer- 3 years
  - Android
  - Java
  - Kotlin
  - Flutter
  - Dart
- Tech Hobbyist (more than Todd)
- Father of Two
- Awesome Boyfriend
- Basketball
- Movie Aficionado
- TVing
- Youtube Rabbit Holes

Twitter: @ronjdaugherty29

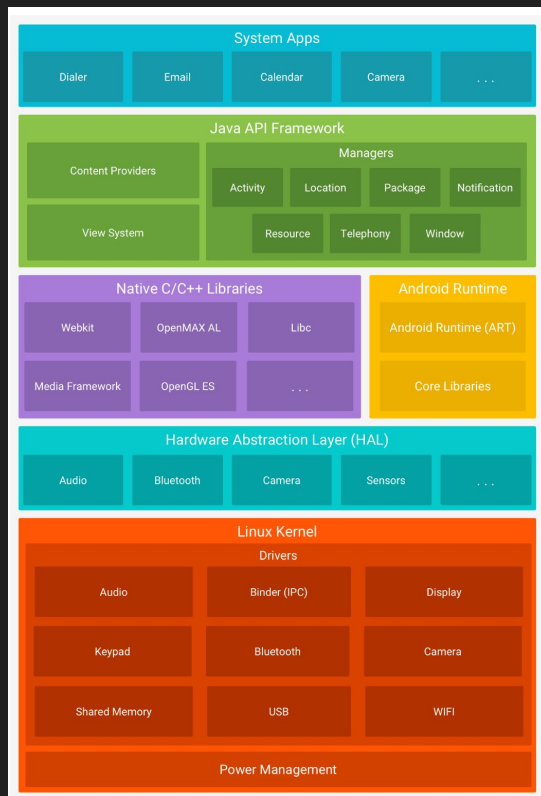
# Agenda

- Tooling
- Languages
- Fundamentals
- Resources
- Q&A

# Background

# Background

- Released 2008
- Current platform architecture



# Tooling



# Tooling

## Android Studio

<https://developer.android.com/studio/>

System Requirements at bottom of page

The image shows the Android Studio website and a screenshot of the IDE interface. The website header features the "androidstudio" logo and the text "Android Studio provides the fastest tools for building apps on every type of Android device." Below this is a prominent "DOWNLOAD ANDROID STUDIO" button, with "3.1.3 for Mac (849 MB)" and "DOWNLOAD OPTIONS" / "RELEASE NOTES" links underneath. The IDE screenshot displays the "image\_grid.xml" file in the main editor, showing XML code for a grid of images. The interface includes a project explorer on the left, a preview window on the right showing a grid of images, and an Android Profiler at the bottom with CPU, MEMORY, and NETWORK graphs.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:layout_editor_absoluteX="86dp"
tools:layout_editor_absoluteY="86dp">
    <ImageView2
        android:id="@+id/imageView2"
        android:layout_width="86dp"
        android:layout_height="86dp"
        android:contentDescription="@string/app_name"
        app:layout_constraintBottom_toTopOf="@+id/imageView6"
        app:layout_constraintEnd_toStartOf="@+id/imageView3"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/grid_1" />
    <ImageView
        android:id="@+id/imageView3"
        android.support.constraint.ConstraintLayout
        ImageView
    </Image>
</android.support.constraint.ConstraintLayout>
```

# Other Requirements

# Other Requirements

## Emulator -

- Easy to get started
- Way faster than used to be
- Variety
- Can make new profiles
- Not full featured
  - Bluetooth
  - Camera
- Pinch-zoom is tough

## Physical -

- Most realistic
- Expensive
- Hard to find
- Full featured

# Other Requirements

- Google Play Console
  - Publishing via Play Store
  - \$25 - one-time fee
  - <https://developer.android.com/distribute/console/>
- Or, just email
  - Private
  - Self, co-workers, other devices

# Languages

# Languages

## Java

- Start here
- Object-oriented
- Been around “forever”
- Helpful to know history
- Many examples

## Kotlin

- Then learn this later
- Modern. Expressive. Safe.
- Avoid null pointer exceptions
- Fairly new

# Fundamentals

# Fundamentals

- Write app in Android Studio
- Developer tests on emulator and/or physical device
- Testers test on physical devices
- Distribute via Play Console
  - Internal
  - Alpha
  - Beta
  - Production



# Fundamentals

- App packaged into an archive, \*.apk
  - This is what gets downloaded distributed to users
  - Similar to an .exe, .jar
- Each app is in its own security sandbox
  - Runs as a different user
  - Runs in different JVM (java virtual machine)
  - Runs in own process

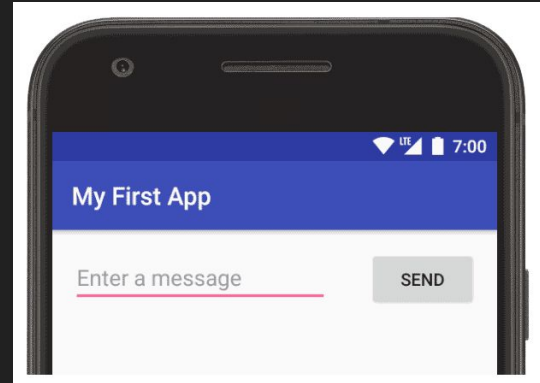
# Fundamentals

## Application components

1. Activities
2. Services
3. Broadcast receivers
4. Content providers

# Fundamentals - Activities

- Entry point for interacting with user
- A single screen with a user interface



# Fundamentals - Services

- Background work
  - Network sync
  - Music
- No UI supported

# Fundamentals - Broadcast Receiver

- App can respond to system wide events
  - Battery low
  - Picture taken
  - Screen turned off
- App doesn't have to be running

# Fundamentals - Content Providers

- Shared set of app data stored in system
  - Contacts
  - Notes
- Other apps can access data

# Resources

# Resources

- Udacity <https://www.udacity.com/google>
- Email digests
  - Medium
  - Porter.io
- Stack Overflow
- Android Weekly
- GDG Ann Arbor and/or GDG Detroit
- Android Developer website
- Android Arsenal



# Resources

- <https://android.jlelse.eu/learning-android-development-in-2018-part-1-83a514f6a205>
- <https://blog.aritraroy.in/what-my-2-years-of-android-development-have-taught-me-the-hard-way-52b495ba5c51>
- <https://dribbble.com/>
- <https://www.uplabs.com/android>
- <https://android-arsenal.com/>
- <https://github.com/pcqpcq/open-source-android-apps>
- <https://android.jlelse.eu/>
- <https://medium.com/mindorks>
- <https://blog.aritraroy.in/>

# Q&A