THE IMPORTANCE OF DESIGN

- “Enchant Me, Simplify my Life, Make me Amazing!”: Android’s creative vision
- Aesthetically pleasing as well as functioning
  - Users decide on your app in 30 seconds, looks count a lot
Apps should be

- Fun to use
- Surprise in delightful ways with animations and smooth transitions
- Let users touch objects directly (rather than through menus)
- Customizable with sensible defaults
- Shortcuts
- Don’t ask for same info twice
- Deviate with purpose

see Material Design principles here
https://developer.android.com/design/material/index.html
Toy app: Visual polish

We will start with a basic design, and customize fonts, colors, etc.
Visual mockups

- Mockup: a model of an app, used for design evaluations
- Often done by a graphics artist or designer
- Note the keylines above
Design: Which looks better?
Material Design: Colors

- Use only a few colors
- Use contrast to highlight components
- Distinguish views and interactive components
- Have 1–3 shades of the Primary Color, and an Accent color that stands out
Looking at the color resources

- You can modify the two primary colors and accent color for your app
- Use these in coloring your own views

check out:
https://www.materialpalette.com/

Toy App

```
<color name="colorPrimary">#673AB7</color>
<color name="colorPrimaryDark">#512DA8</color>
<color name="colorPrimaryLight">#D1C4E9</color>
<color name="colorAccent">#FF9100</color>
```
Quiz: Which palettes are acceptable?
● Default Android font: Roboto
● Comes in different font families:
  ○ serif(roman), sans-serif(gothic), sans-serif-thin, sans-serif-smallcaps, etc.
  ○ sans-serif is the default for Android
● Changing these settings --- see below (from the toy app)
Text: Which looks better, a single font family, or many?

- Use the same font throughout your app
- You can change styles and font families to highlight, distinguish, etc.
Both dp and sp render the same size across screen densities.

- **dp**: density independent pixels
  - Used for multiple screen densities
- **sp**: scale-independent pixels
  - Used for accessibility for text
**Styles and Themes**

**Style**

A *style* is an xml resource file, separate from the layouts, where you can set all these properties in one place. Then later, we can apply that style to any View we want.

<table>
<thead>
<tr>
<th>XML</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layouts</td>
<td>Style</td>
</tr>
</tbody>
</table>

**Theme**

And a *theme* is created in the same way, and is just a style that’s applied to an *entire* Activity or application and not just one view.
Using Styles for Uniformity: Wonky to Uniform

- Styles are declared in the style file in res/values
- Styles can be inherited

In activity layout

```
<item name="android:layout_width">match_parent</item>
<item name="android:layout_height">wrap_content</item>
<item name="android:drawablePadding">8dp</item>
<item name="android:paddingTop">4dp</item>
<item name="android:paddingBottom">4dp</item>
<item name="android:textSize">28sp</item>
<item name="android:textColor">@android:color/black</item>
```