CSc 317
Android App Basics

Benjamin Dicken

Announcements

- FYI: Readings and topics could change!
- PA due next week
- Discord?
- Office hours
- Quiz
- How to compress your project

    File -> Export

Java or Kotlin?

- Java vs Kotlin
- Kotlin is the officially preferred Android Dev language
- May use if you want to learn the new syntax
- In class will mostly stick to Java

MyFirstApp Prep

- As a part of your first reading, you’ll create a simple app
    - Do the readings!
- We will go through some of the basics today

MyFirstApp

- Install Android Studio
- Create a new app (empty activity, Java)

MyFirstApp

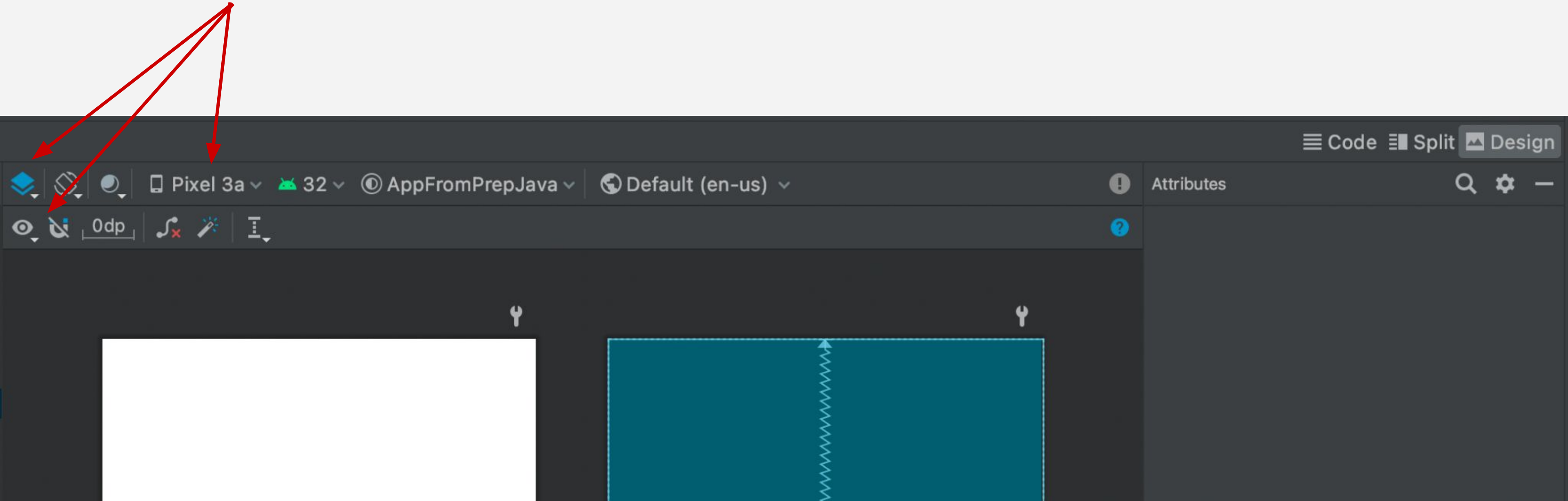
- Let’s inspect a few files:
  - `MainActivity.java`
  - `Activity_main.xml (text)`
  - `AndroidManifest.xml`
  - `build.gradle`
- Will be using java + xml heavily

Run the app in the emulator

- What are emulators?
- Can create one for simulating various devices

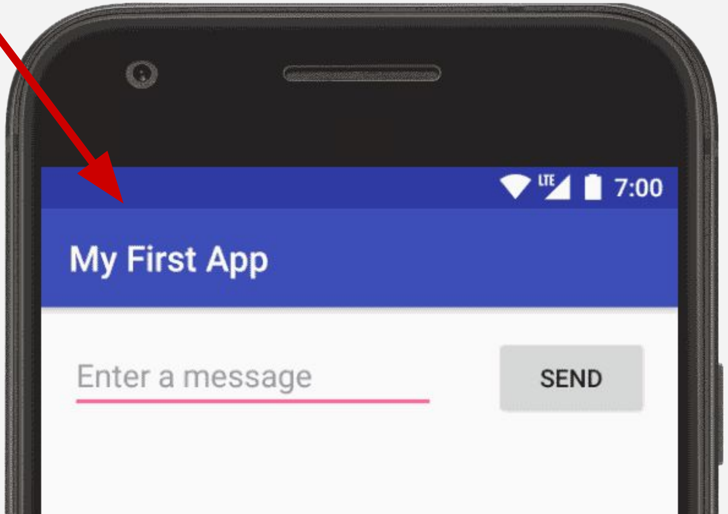
Modify the UI with the GUI

- Click on activity_main.xml
- See Code view and Design view
- Configure some settings



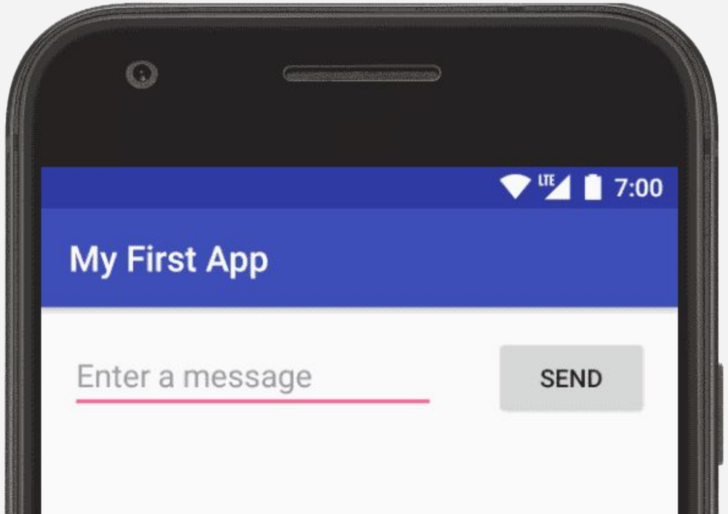
MyFirstApp

- Make the app look like this
- Add a plain text and a button
- Then, change so that it starts a new page (activity) when the button is clicked with the text from the plaintext displayed



MyFirstApp

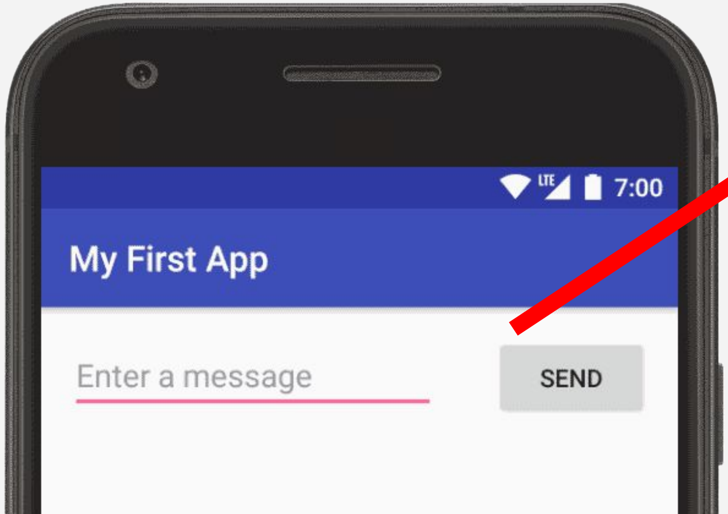
- The instructions showed you how to build the UI via GUI
- Let’s look at what happened in the XML

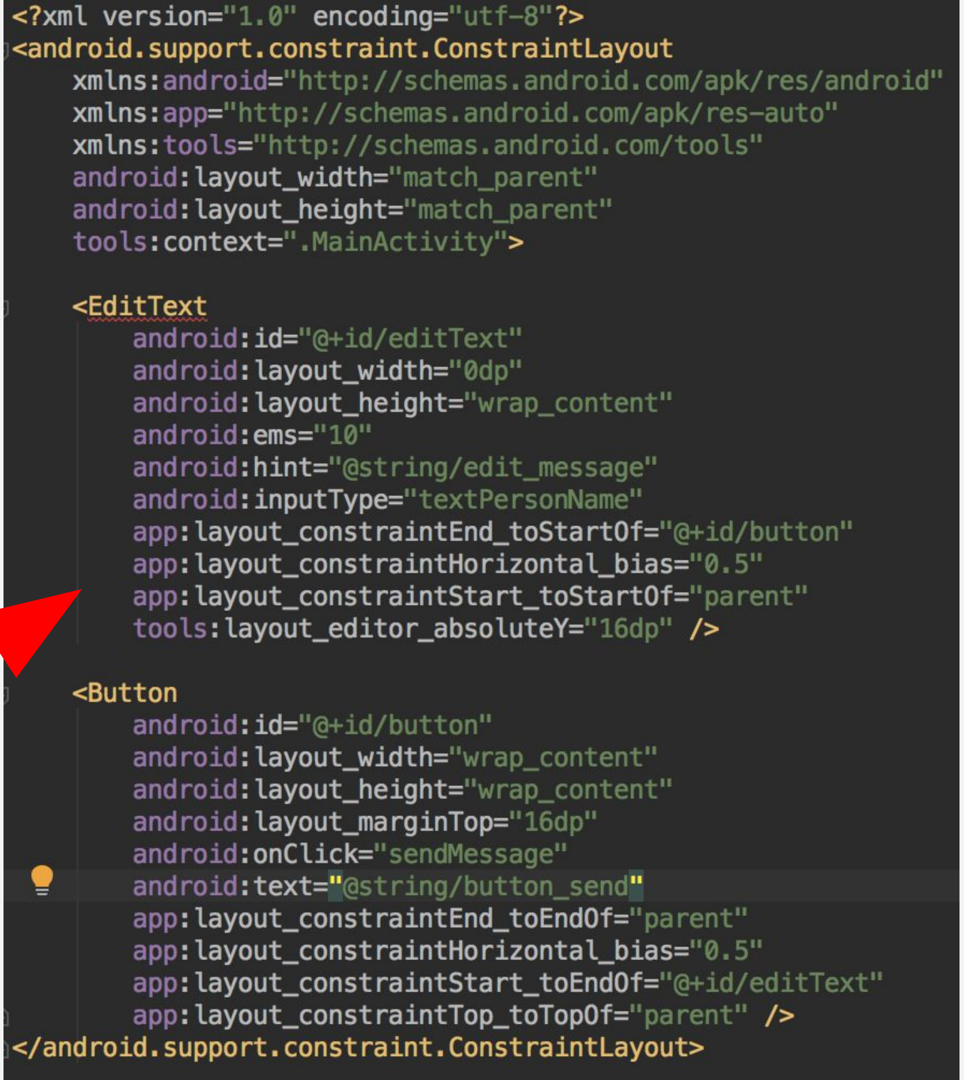


Activities

- For each activity that was created, two main files:
  - The java class file
  - The layout file

MyFirstApp





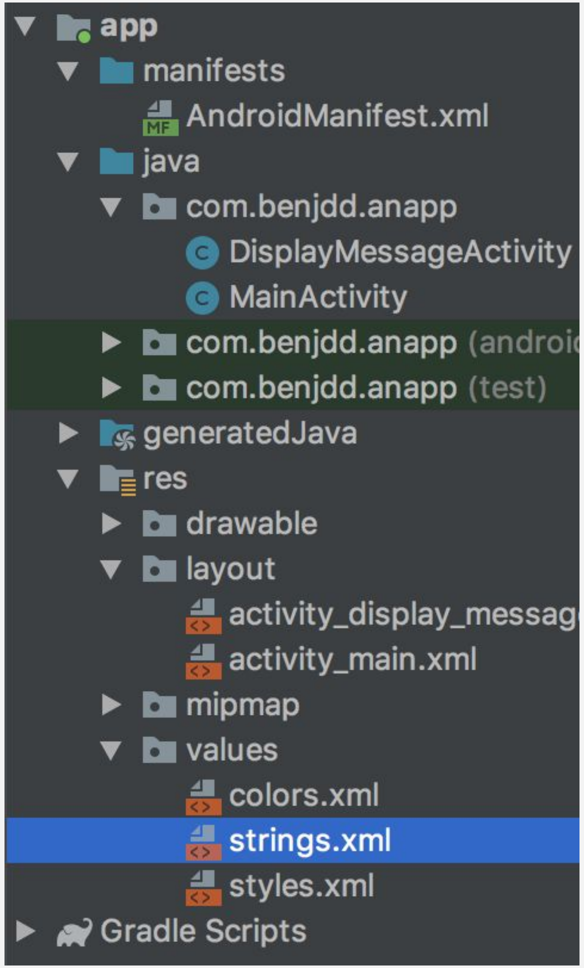
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editText"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="@string/edit_message"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toStartOf="@+id/button"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="16dp" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:onClick="sendMessage"
        android:text="@string/button_send"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toEndOf="@+id/editText"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

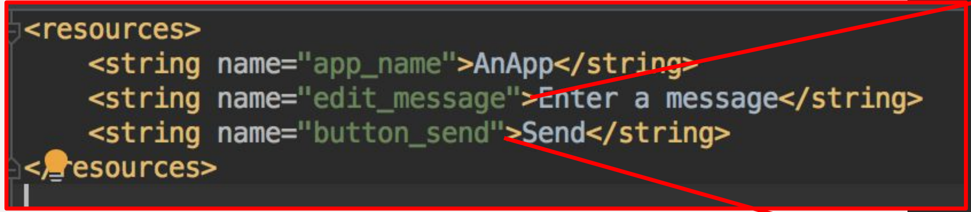
MyFirstApp

- When you created the string resources, they were actually added to the strings.xml file.
- To add resources to an android program, add them to a directory or XML file in the **res** dir

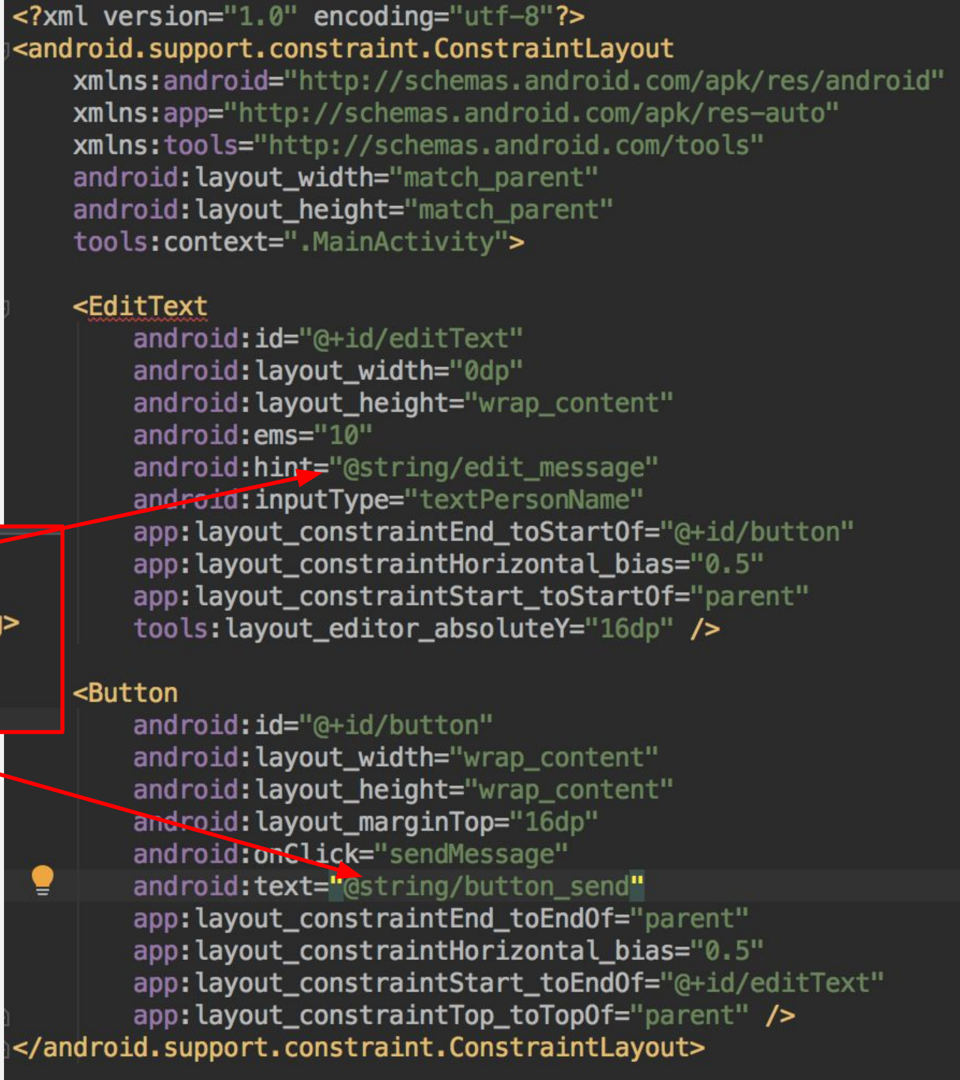


```
▼ app
  ▼ manifests
      AndroidManifest.xml
  ▼ java
    ▼ com.benjdd.anapp
        DisplayMessageActivity
        MainActivity
    ▶ com.benjdd.anapp (androi
    ▶ com.benjdd.anapp (test)
  ▶ generatedJava
  ▼ res
    ▶ drawable
    ▼ layout
        activity_display_messag
        activity_main.xml
    ▶ mipmap
    ▼ values
        colors.xml
        strings.xml
        styles.xml
▶ Gradle Scripts
```

MyFirstApp



```
<resources>
    <string name="app_name">AnApp</string>
    <string name="edit_message">Enter a message</string>
    <string name="button_send">Send</string>
</resources>
```



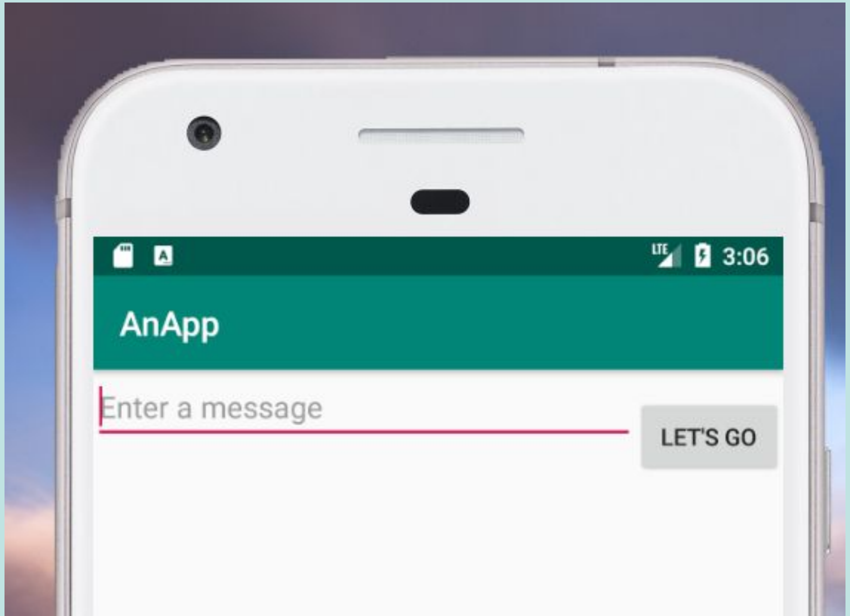
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editText"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="@string/edit_message"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toStartOf="@+id/button"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="16dp" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:onClick="sendMessage"
        android:text="@string/button_send"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toEndOf="@+id/editText"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

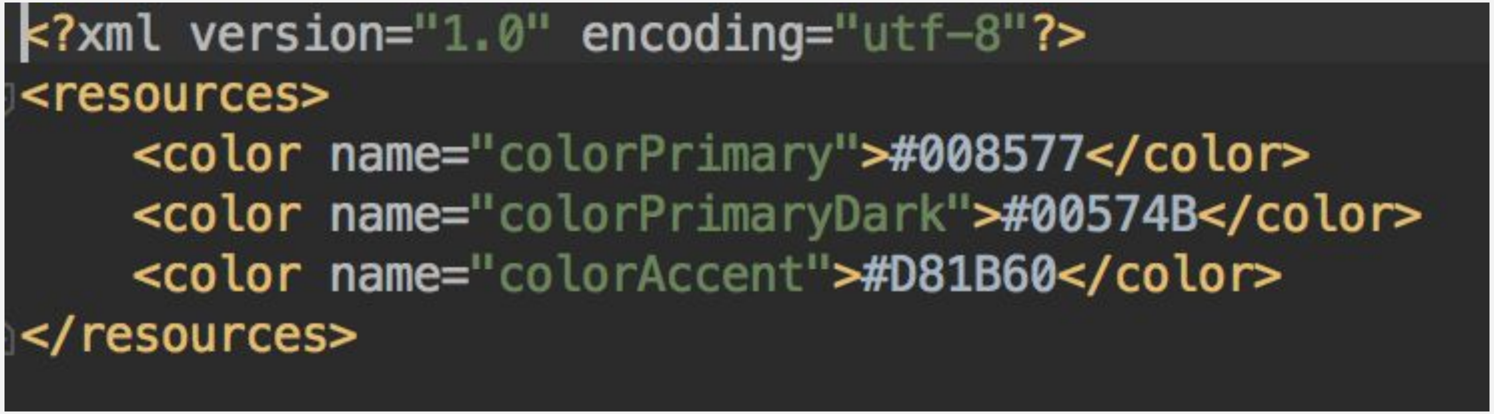
Change the button label to say “Let’s Go”

- Add a new string to strings.xml
- Change the activity_main.xml
- Run the app



Custom Color

- You can also add custom colors (colors.xml)
- You can reference the color using the name
  - Use ***meaningful*** names
- Colors based on hexadecimal value

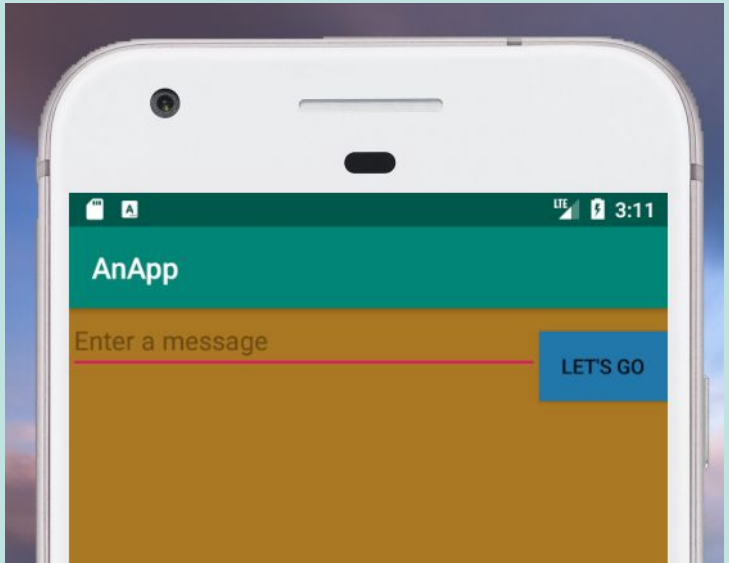


```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#008577</color>
    <color name="colorPrimaryDark">#00574B</color>
    <color name="colorAccent">#D81B60</color>
</resources>
```

ICA

Add color

- Add a new custom colors to strings.xml
  - You can use a color picker to choose a color
- Change the background color of the **`ContraintLayout`** and of the Button and the background using the **`android:background`** attribute
- Run the app

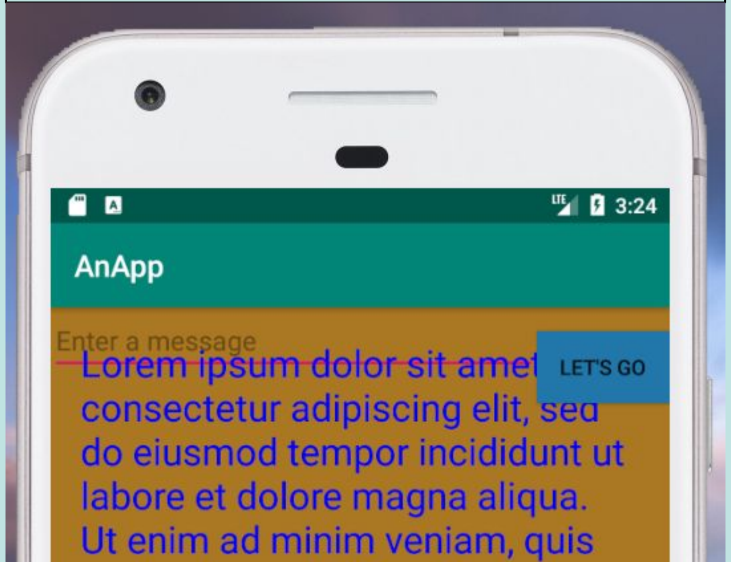


ICA

Add some text

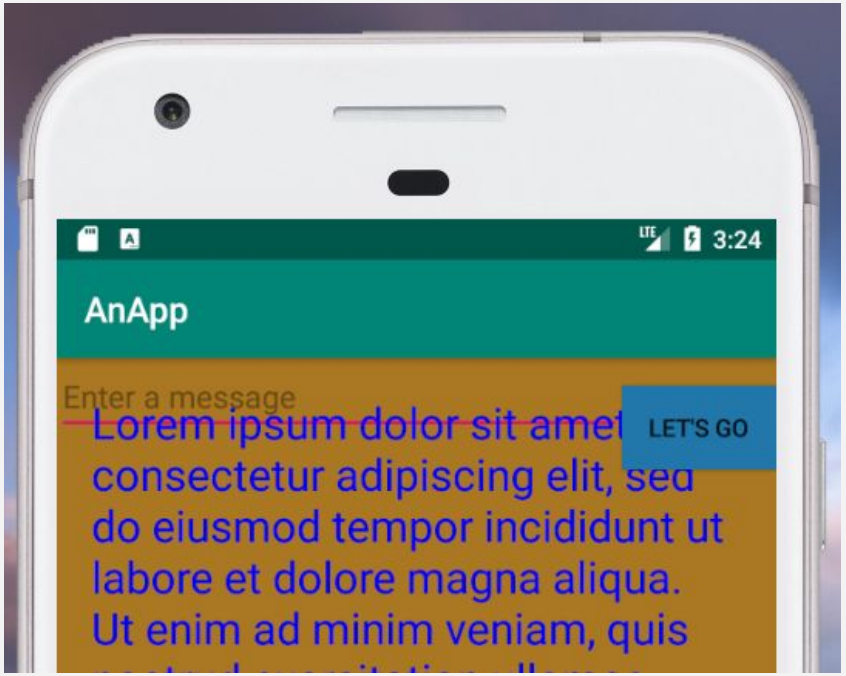
- Add a paragraph of text to the Layout using a **`TextView`** tag
- Should have:
  - Large font
  - Blue
  - Centered/padded
- Text/color should be added to strings.xml/colors.xml
- Run the app

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_gravity="center"
  android:padding=???
  android:text=???
  android:textColor=???
  android:textSize=??? />
```



Lorem ipsum dolor sit amet consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis

Why the overlap?



Lorem ipsum dolor sit amet consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis

ConstraintLayout vs LinearLayout

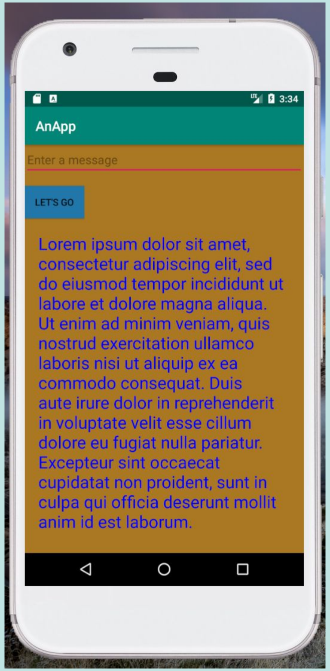
- **ConstraintLayout**: Layout out elements relative to one another
  - Notice things like:
    `app:layout_constraintEnd_toStartOf="@+id/button"`
- **LinearLayout**: Layout out sequentially, either vertically or horizontally

```
<LinearLayout
  . . .
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">
```

Change the Layout

- Use a LinearLayout
- Set to vertical layout
- Remove the **`app:layout_*`** and **`tools:layout_*`** attributes
- Ensure **`android:layout_width`** and **`android:layout_height`** are defined for the three nested elements

```
<LinearLayout
  . . .
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

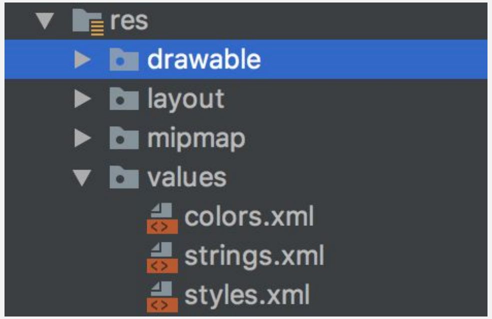
ImageView

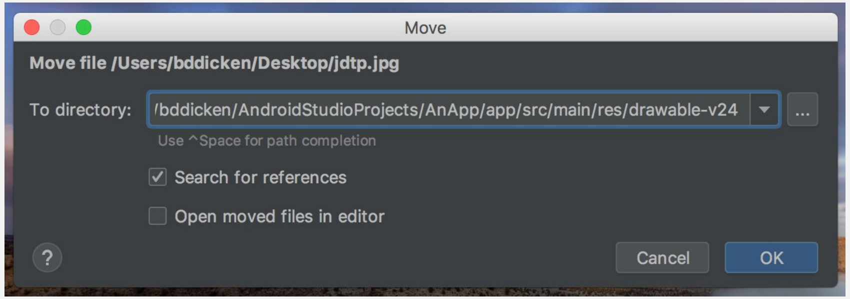
- **ImageView**: View for displaying an image
  - Specify an id
  - Specify the id of the image resource
    - **`android:src="@drawable/drawable_resource_id"`**
  - Can also specify, withd, height, etc.
  - For instance:

```
<ImageView
        android:id="@+id/suns_logo"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:src="@drawable/suns_logo" />
```

Adding an image

- Click and drag an image to the **drawable** directory
- Change directory to be named **drawable**
- Click OK
- If you didn't change anything else, the ID should be the image name, not including the extension





Move file /Users/bddicken/Desktop/jdtp.jpg

To directory: /bddicken/AndroidStudioProjects/AnApp/app/src/main/res/drawable-v24

Use ^Space for path completion

☑ Search for references

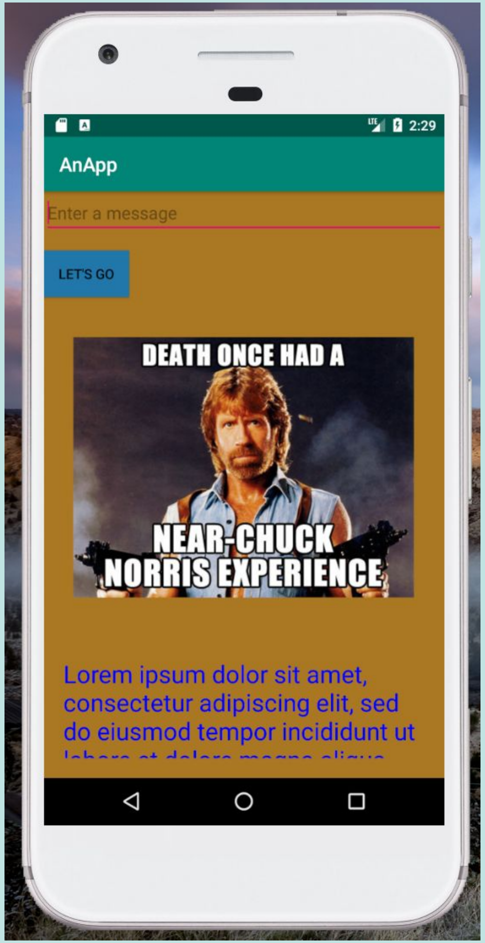
☐ Open moved files in editor

ICA

Add an image

- Download and add an image
- Use a ImageView
- Ensure **android:layout_width** and **android:layout_height** are defined

```
<ImageView
  android:id="@+id/some_id"
  android:layout_width="200dp"
  android:layout_height="200dp"
  android:src="@drawable/some_id" />
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut