
# Android Lecture #1

# What are we doing today?

- Creating our first application!
- Talk about our project structure
- Adding Activities

# Android Studio

Version 3.2.1

Start a new Android Studio project

Open an existing Android Studio project

Check out project from Version Control

Profile or debug APK

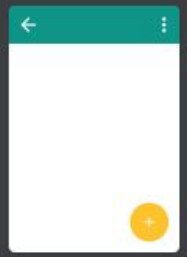
Import project (Gradle, Eclipse ADT, etc.)

Import an Android code sample

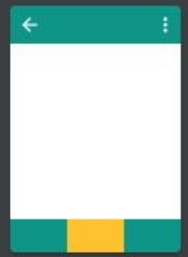
Events Configure Get Help

# Add an Activity to Mobile

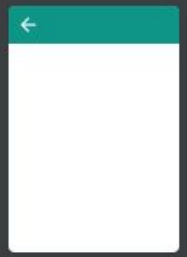
Add No Activity



Basic Activity



Bottom Navigation Activity

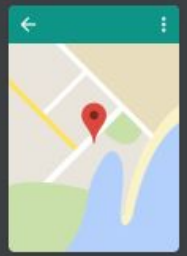


Empty Activity

Fragment + ViewModel

Fullscreen Activity


Google AdMob Ads Activity

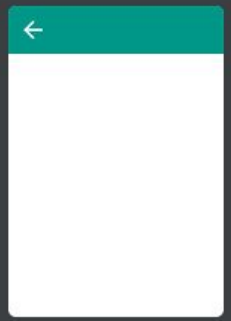


Google Maps Activity

Cancel | Previous | Next | Finish

Create New Project

# Configure your project



Empty Activity

Creates a new empty activity

Name

Click Counter

Package name

il.co.ravtech.clickcounter

Save location

C:\Projects\ClickCounter

Language

Java

Minimum API level

API 21: Android 5.0 (Lollipop)

Your app will run on approximately **85.0%** of devices.

Help me choose

This project will support instant apps

Use androidx.* artifacts

Previous Next Cancel Finish

# Target Android Devices

## Select the form factors and minimum SDK

Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

- [x] **Phone and Tablet**
  - API 21: Android 5.0 (Lollipop)
  - By targeting **API 21 and later**, your app will run on approximately **85.0%** of devices. Help me choose
  - [ ] Include Android Instant App support
- [ ] **Wear OS**
  - API 23: Android 6.0 (Marshmallow)
- [ ] **TV**
  - API 21: Android 5.0 (Lollipop)
- [ ] **Android Auto**
- [ ] **Android Things**
  - API 24: Android 7.0 (Nougat)
- [ ] **Glass**
  - Glass Development Kit Preview (API 19)

Cancel | Previous | Next | Finish

| ANDROID PLATFORM VERSION | API LEVEL | CUMULATIVE DISTRIBUTION |
|---|---|---|
| 4.0 Ice Cream Sandwich | 15 | |
| 4.1 Jelly Bean | 16 | 99.6% |
| 4.2 Jelly Bean | 17 | 98.1% |
| 4.3 Jelly Bean | 18 | 95.9% |
| 4.4 KitKat | 19 | 95.3% |
| 5.0 Lollipop | 21 | 85.0% |
| 5.1 Lollipop | 22 | 80.2% |
| 6.0 Marshmallow | 23 | 62.6% |
| 7.0 Nougat | 24 | 37.1% |
| 7.1 Nougat | 25 | 14.2% |
| 8.0 Oreo | 26 | 6.0% |
| 8.1 Oreo | 27 | 1.1% |

## Lollipop

**User Interface**

Material design support
Concurrent documents and activities in the recents screen
WebView updates
Screen capturing and sharing

**Notifications**

Lock screen notifications
Notifications metadata

**Graphics**

Support for OpenGL ES 3.1
Android Extension Pack

**Media**

Camera API for advanced camera capabilities
Audio playback
Media playback control
Media browsing

**Storage**

Directory selection

**Wireless & Connectivity**

Multiple network connections
Bluetooth Low Energy
NFC enhancements

**Battery - Project Volta**

Scheduling jobs
Developer tools for battery usage

**Android in the Workplace and in Education**

Managed provisioning
Device owner
Screen pinning

**Printing Framework**

Render PDF as bitmap

**System**

App usage statistics

**Testing & Accessibility**

Testing and accessibility improvements

**IME**

Easier switching between input languages

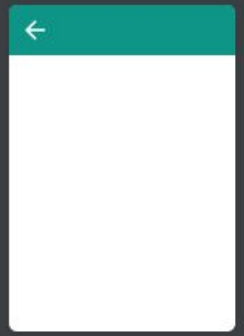
**Manifest Declarations**

Declarable required features
User permissions

https://developer.android.com/about/versions/android-5.0.html

Cancel OK

# Configure Activity



## Creates a new empty activity

Activity Name: MainActivity

☑ Generate Layout File

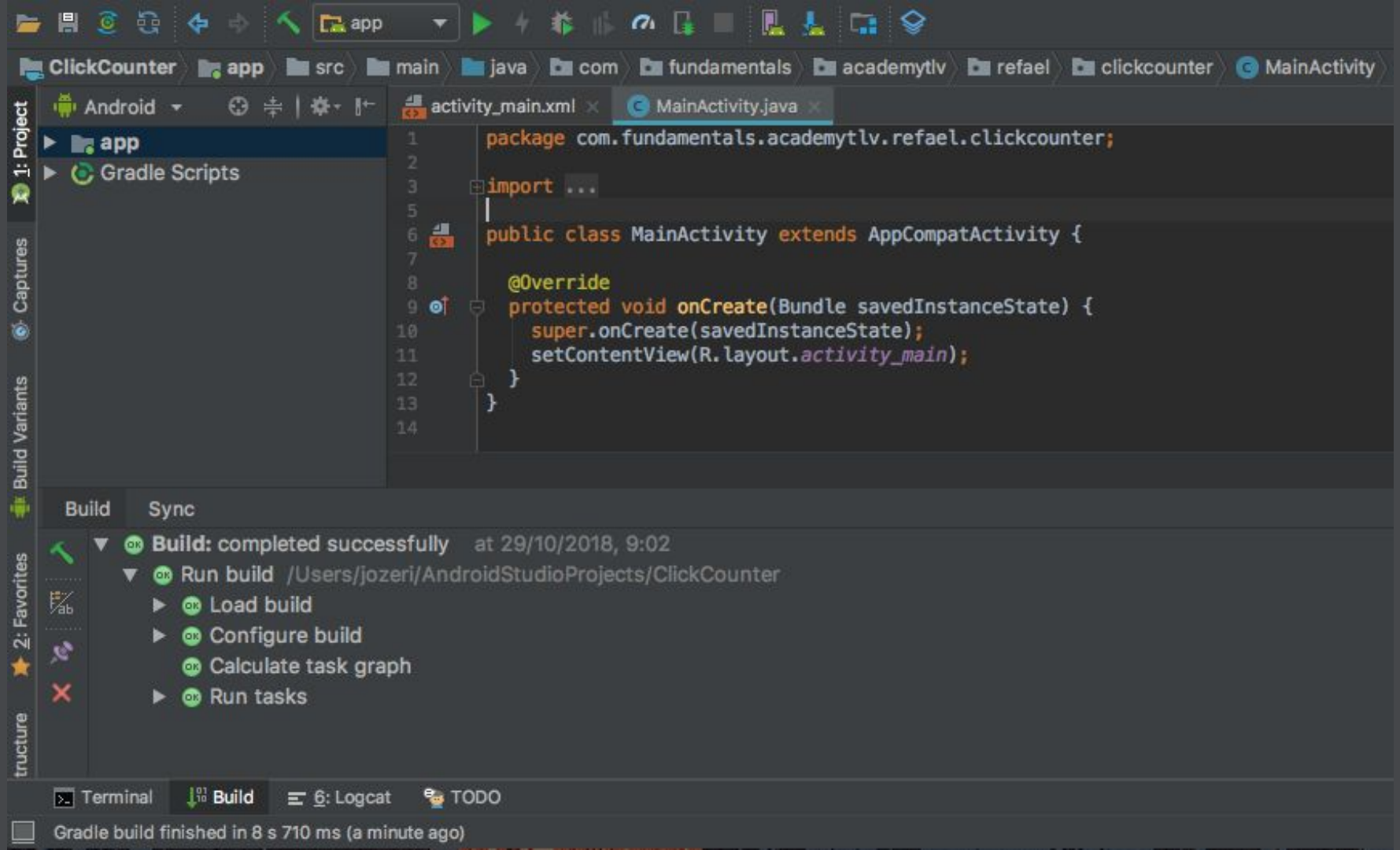
Layout Name: activity_main

☑ Backwards Compatibility (AppCompat)

The name of the activity class to create

Cancel | Previous | Next | Finish

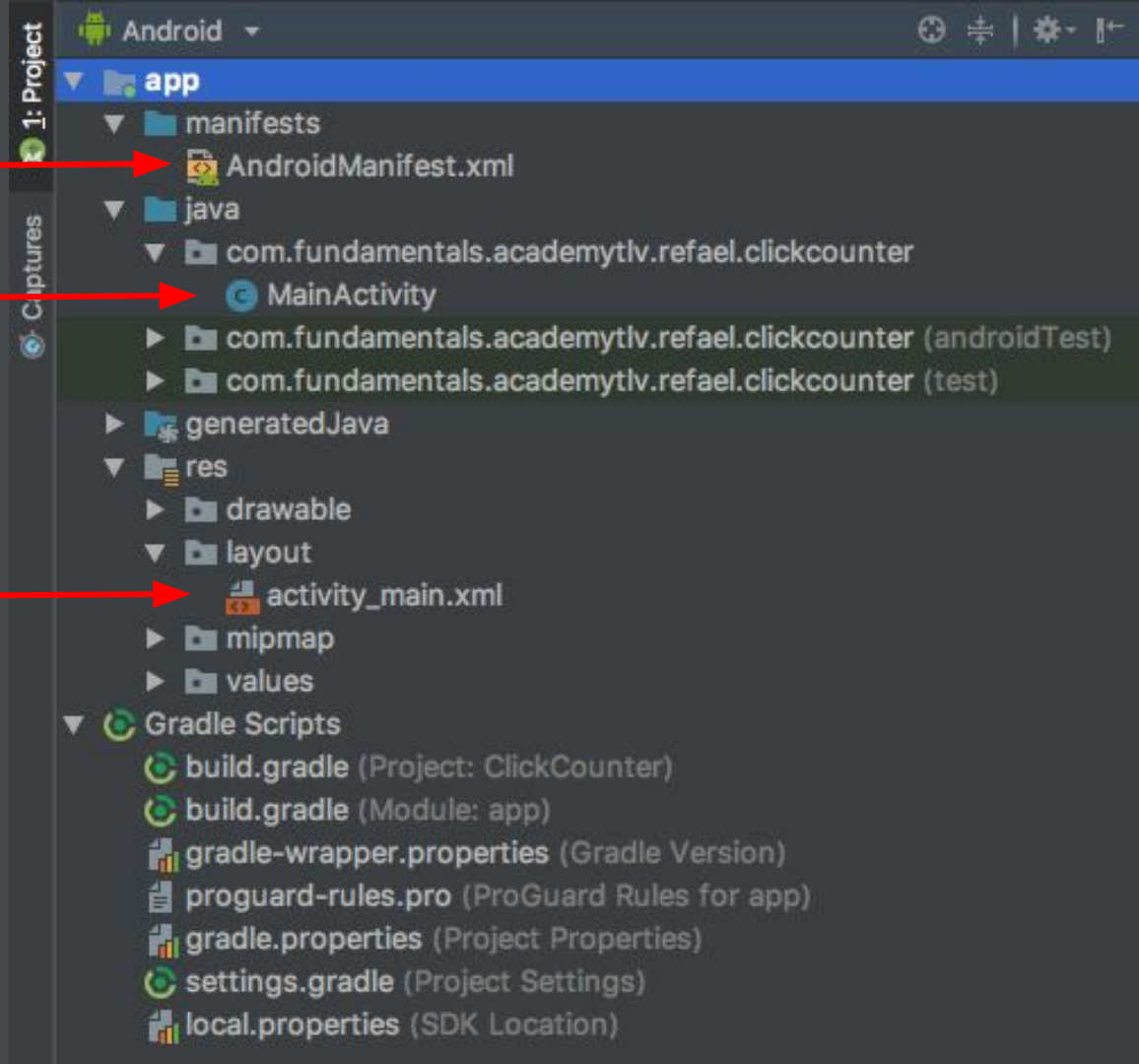

Don't be afraid. I feel it too.


app
ClickCounter
app
src
main
java
com
fundamentals
academytlv
refael
clickcounter
MainActivity
Android
app
Gradle Scripts
activity_main.xml
MainActivity.java
package com.fundamentals.academytlv.refael.clickcounter;
import ...
public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}
}
Build
Sync
Build: completed successfully at 29/10/2018, 9:02
Run build /Users/jozeri/AndroidStudioProjects/ClickCounter
Load build
Configure build
Calculate task graph
Run tasks
Terminal
Build
6: Logcat
TODO
Gradle build finished in 8 s 710 ms (a minute ago)

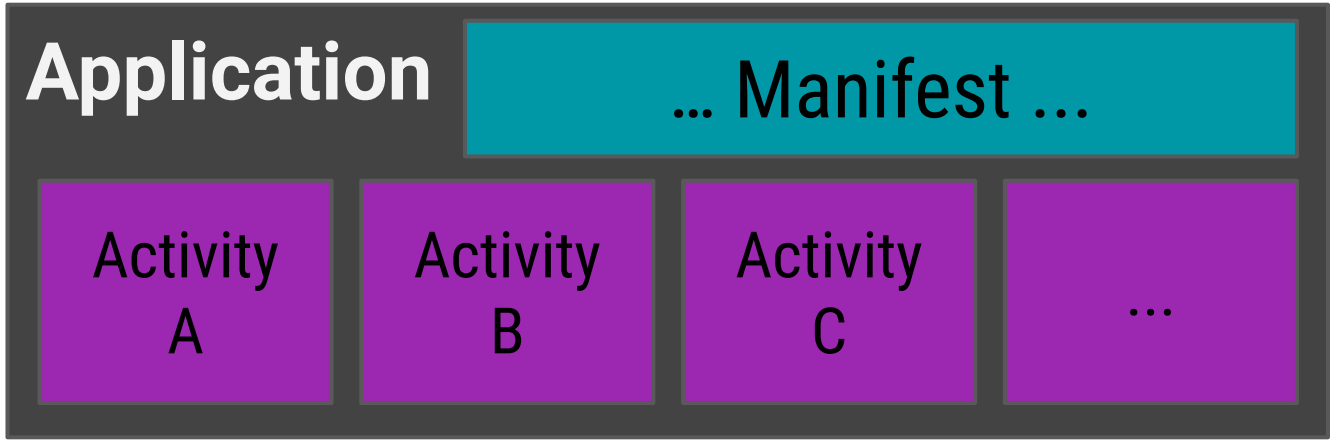
AndroidManifest

MainActivity

activity_main.xml



# Android Application Structure for now



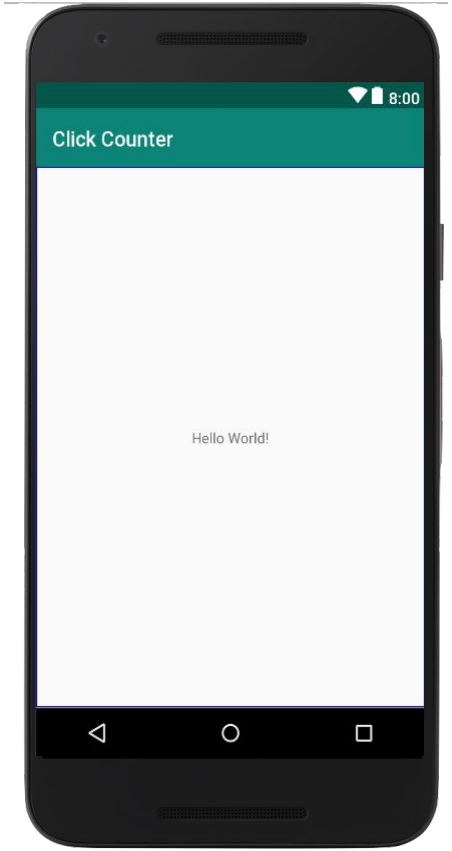
# What's an Activity ??

**Activity** - a component with a screen
with which users can interact in order to **do something.**

Examples: dial the phone, take a photo, send an email, view a map...

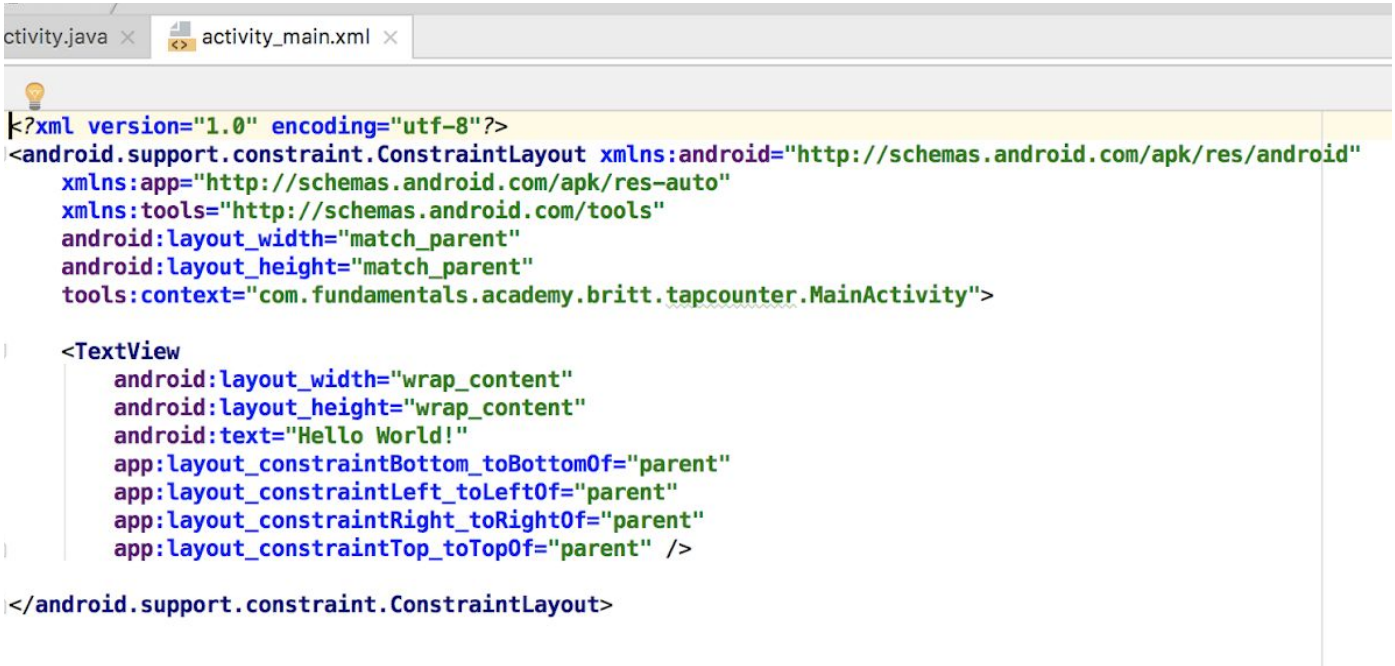
https://developer.android.com/guide/components/activities.html

# How Does An Activity Look?



# How Does An Activity Look?

But to us, it looks like this: (activity_main.xml)



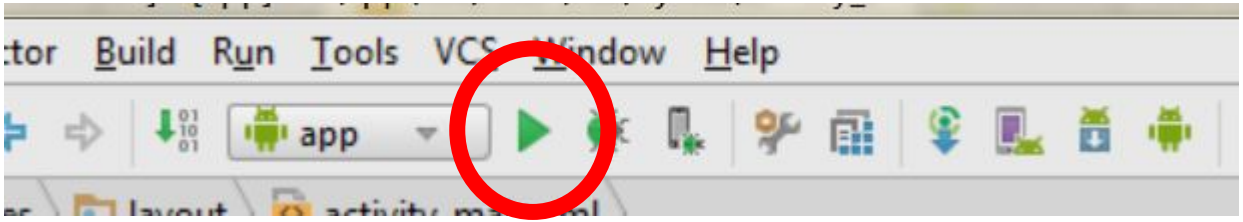
# What Does An Activity Do?

1. Helps the user to perform a **task**
   (i.e. making a call, checking the weather or taking a picture.)

2. Whatever it says in its **java** file,
   and looks as defined in its **XML** layout file.

# How to run the app?

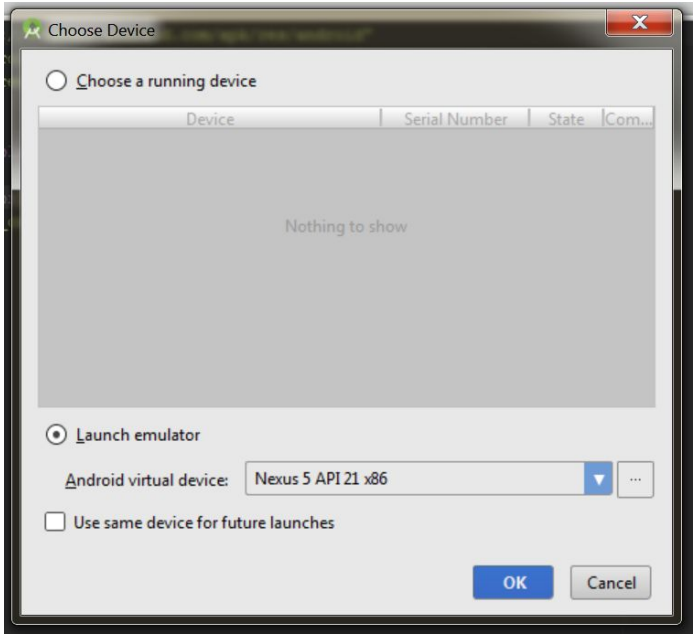
Simple answer:




https://developer.android.com/training/basics/firstapp/running-app.html

# How to run the app?

We need to choose a device.



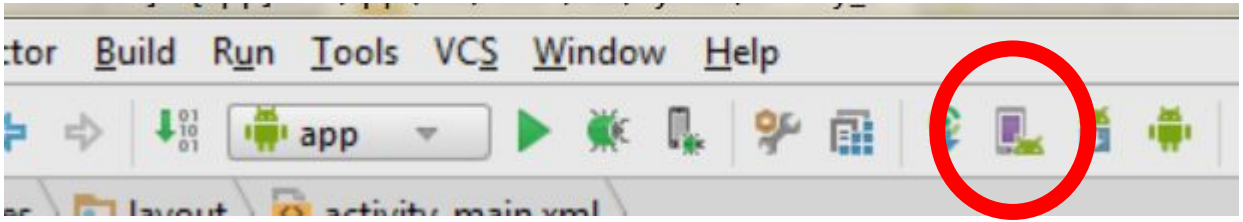
# Option 1:  A Real Device

Allow USB Debugging.

It’s in the Developer Options settings, which might be hidden
(by default from Android 4.2)


Read more: http://developer.android.com/tools/device.html

# Option 2: A Virtual Device

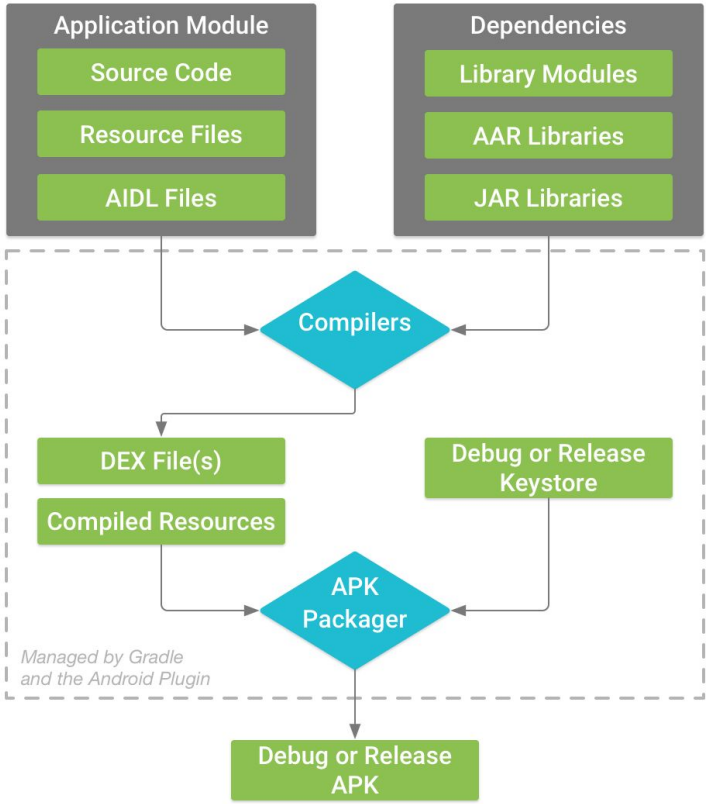



Open the AVD manager
and **create a virtual device**.

The run dialog will offer you to start it.

# What does it take to build the project?

Actually, Quite a lot.




http://developer.android.com/sdk/installing/studio-build.html

# What does it take to build the project?

But Gradle helps us with it.

http://developer.android.com/sdk/installing/studio-bu

Gradle Console

```
Executing tasks: [:app:generateDebugSources, :app:generateDebugAnd

Configuration on demand is an incubating feature.
:app:preBuild UP-TO-DATE
:app:preDebugBuild UP-TO-DATE
:app:checkDebugManifest
:app:preReleaseBuild UP-TO-DATE
:app:prepareComAndroidSupportAppcompatV72103Library UP-TO-DATE
:app:prepareComAndroidSupportSupportV42103Library UP-TO-DATE
:app:prepareDebugDependencies
:app:compileDebugAidl UP-TO-DATE
:app:compileDebugRenderscript UP-TO-DATE
:app:generateDebugBuildConfig UP-TO-DATE
:app:generateDebugAssets UP-TO-DATE
:app:mergeDebugAssets UP-TO-DATE
:app:generateDebugResValues UP-TO-DATE
:app:generateDebugResources UP-TO-DATE
:app:mergeDebugResources
:app:processDebugManifest UP-TO-DATE
:app:processDebugResources
:app:generateDebugSources
:app:preDebugAndroidTestBuild UP-TO-DATE
:app:prepareDebugAndroidTestDependencies
:app:compileDebugAndroidTestAidl UP-TO-DATE
:app:processDebugAndroidTestManifest UP-TO-DATE
:app:compileDebugAndroidTestRenderscript UP-TO-DATE
:app:generateDebugAndroidTestBuildConfig UP-TO-DATE
:app:generateDebugAndroidTestAssets UP-TO-DATE
```

# I click “Run”, You say Ho!

1. Android studio **Builds** the project,
2. **Loads** it to a device (real or virtual)
3. **Executes** it on the device,
   attach a debugger if needed.

```
adb install YourApp.apk
```

```
adb shell am start -n com.examp.name...
```

http://developer.android.com/tools/help/adb.html

# And Then What Happens?

Android Studio launches the app on a device for you.

When an app launches,

Android looks at the application’s manifest,

and decides which activity to show.

# Our Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.fundamentals.academytlv.refael.clickcounter">
       <application
               android:allowBackup="true"
               android:icon="@mipmap/ic_launcher"
               android:label="@string/app_name"
               android:theme="@style/AppTheme">

               <activity android:name=".MainActivity">
                   <intent-filter>
                       <action android:name="android.intent.action.MAIN" />
                       <category android:name="android.intent.category.LAUNCHER" />
                   </intent-filter>
        </activity>

       </application>
</manifest>
```

4 App Components:

Activities

Services

Broadcast Receivers

Content Providers

** Must be declared in our Manifest **

# Create An Activity

- Extend `Activity`.
- Set layout as the UI : `setContentView()`.
- Declare on *AndroidManifest.xml.*

http://developer.android.com/guide/components/activities.html

# Our MainActivity

```
package com.fundamentals.academytlv.refael.clickcounter;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
  }
}
```

# Our MainActivity

```
<?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">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

# Any Questions?

# A Little Bit On Layouts

# Views

Everything in that XML file is a View.

A View:

- Knows to **draw** itself
- Used for **user interaction**
- Has (at least) height and width (`match_parent` / `wrap_content/fixed`)
- May have an id (`@+id/` means to create an id if it doesn't exist)

# View Group (Layout)

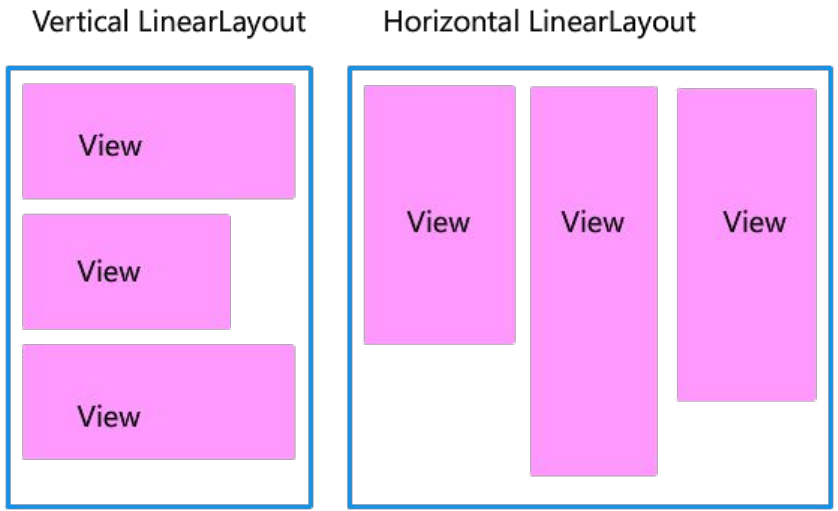
A special kind of view.

Knows to position other views on the screen.

# LinearLayout

Lays items in a row or a column.

Can also divide existing space by weight.



# RelativeLayout

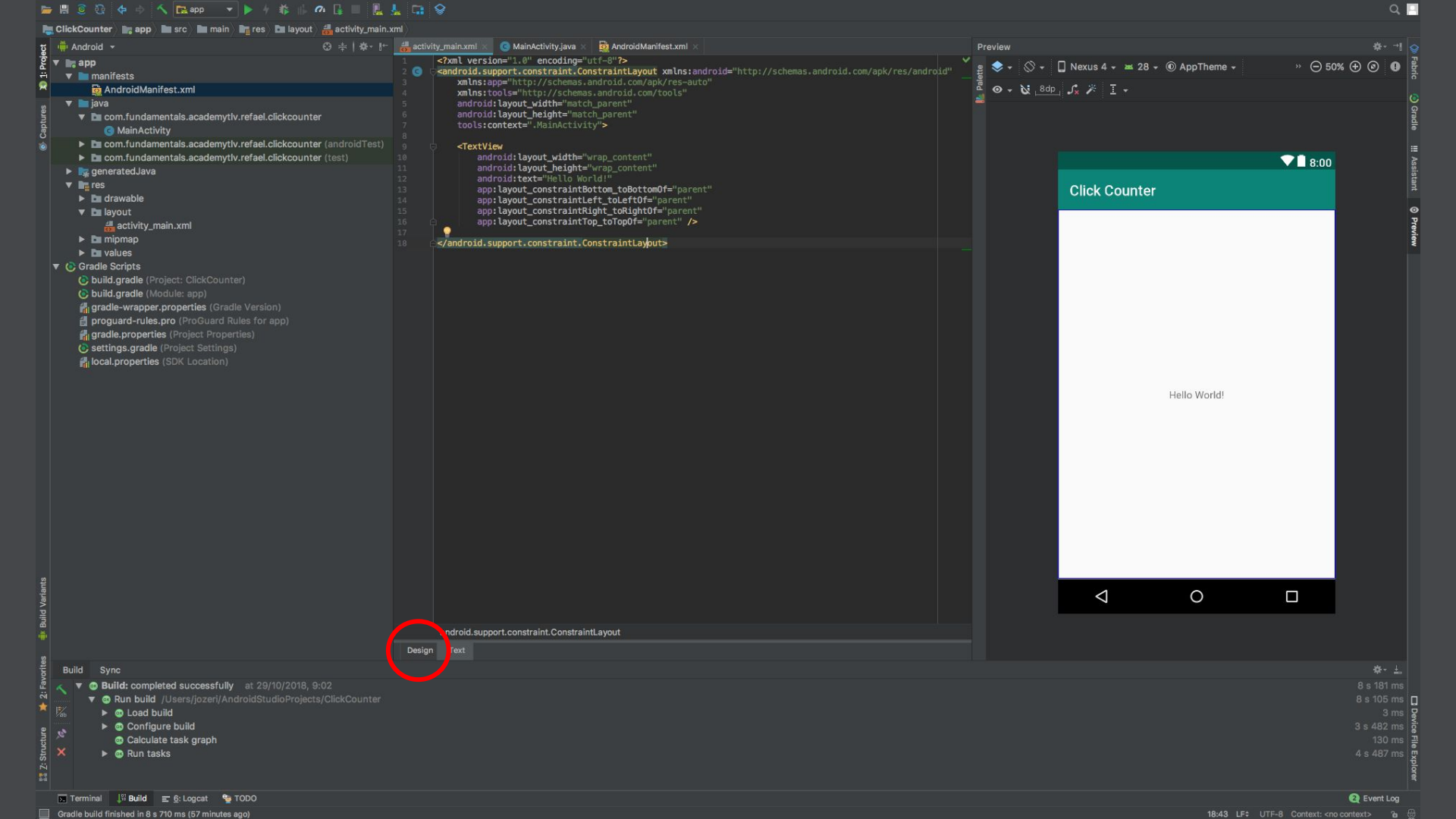
Lays items next to each other, or relative to their parent.

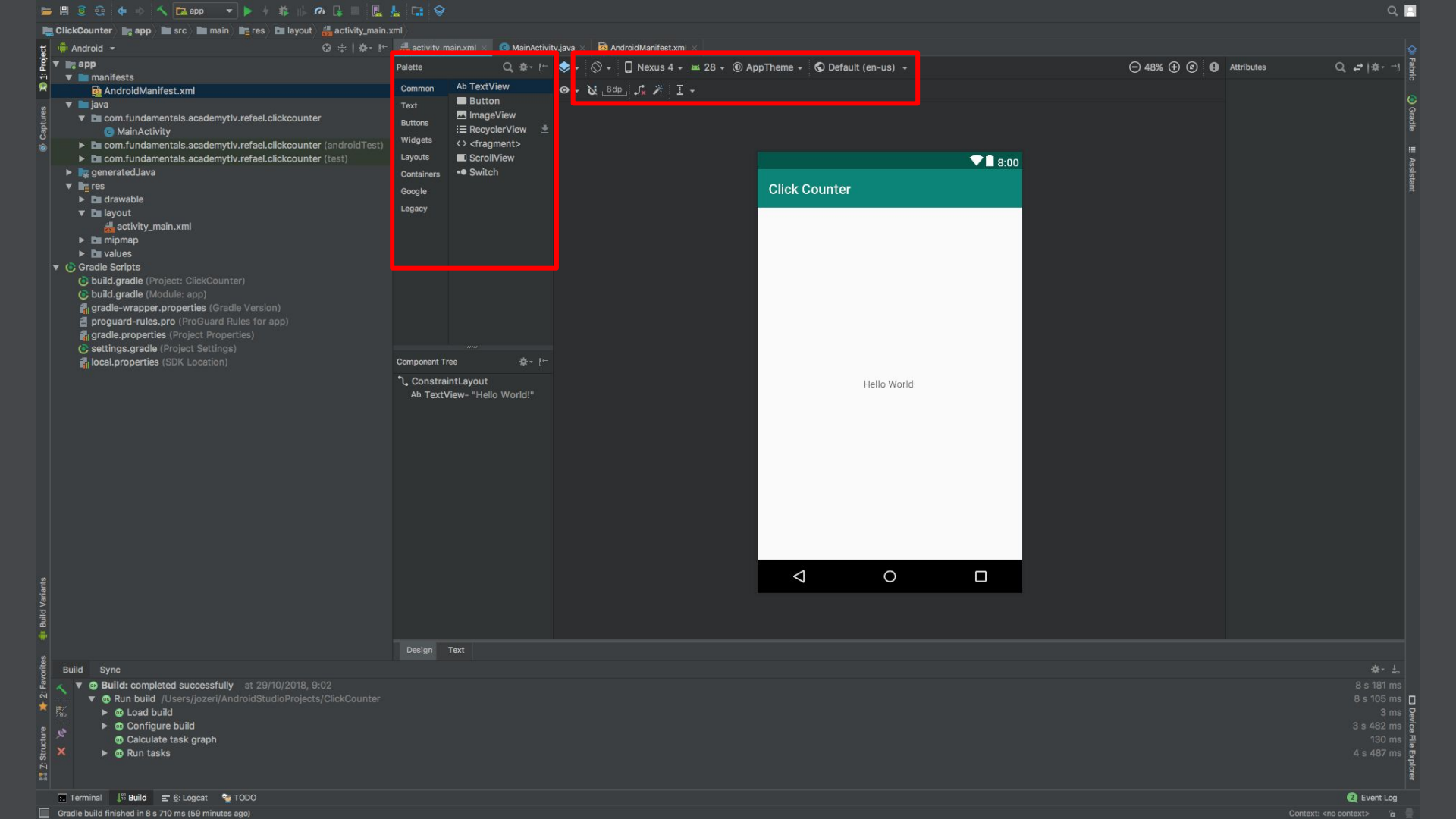
# Constraint Layout

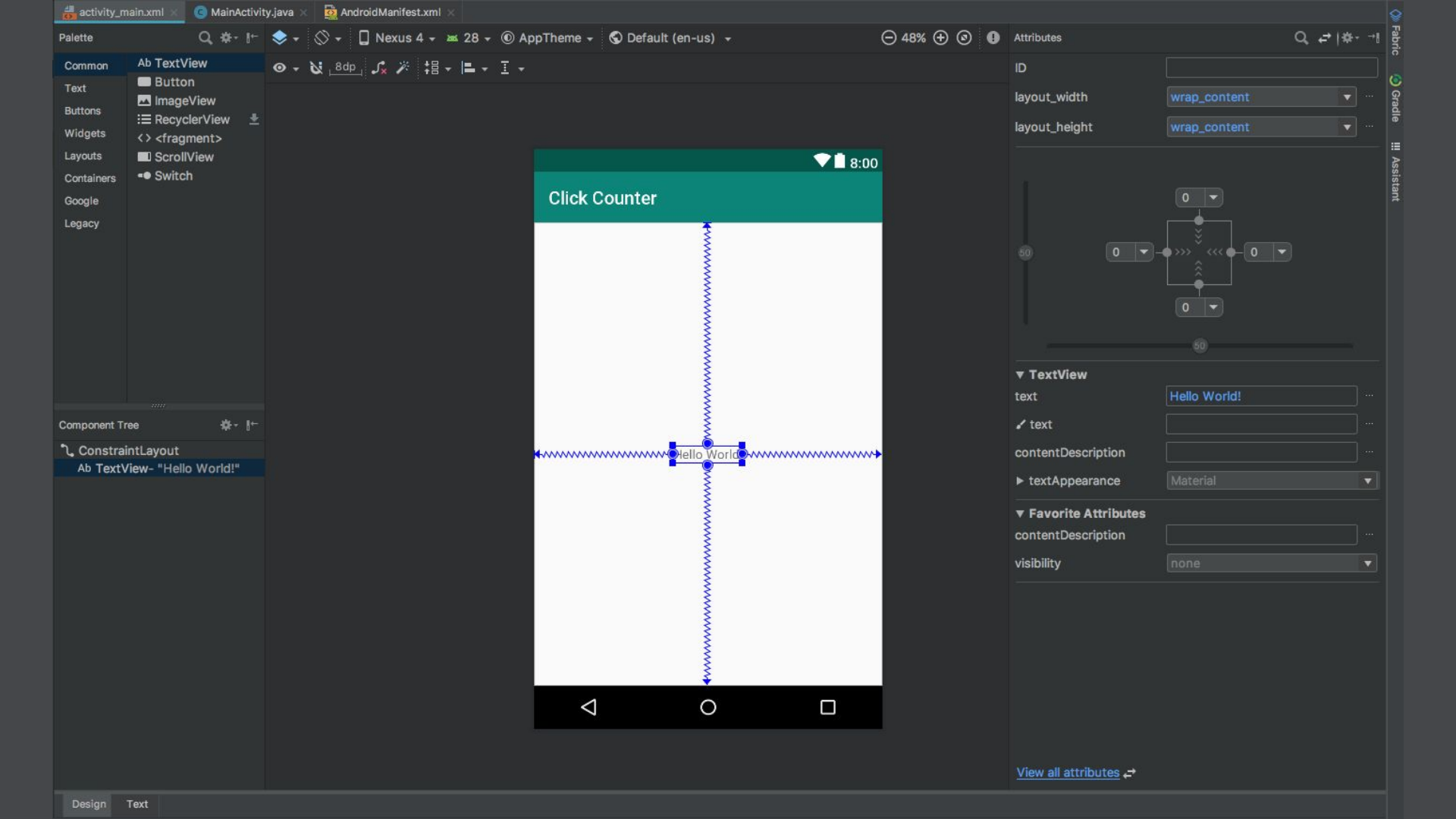
….we’ll discuss later on the course...

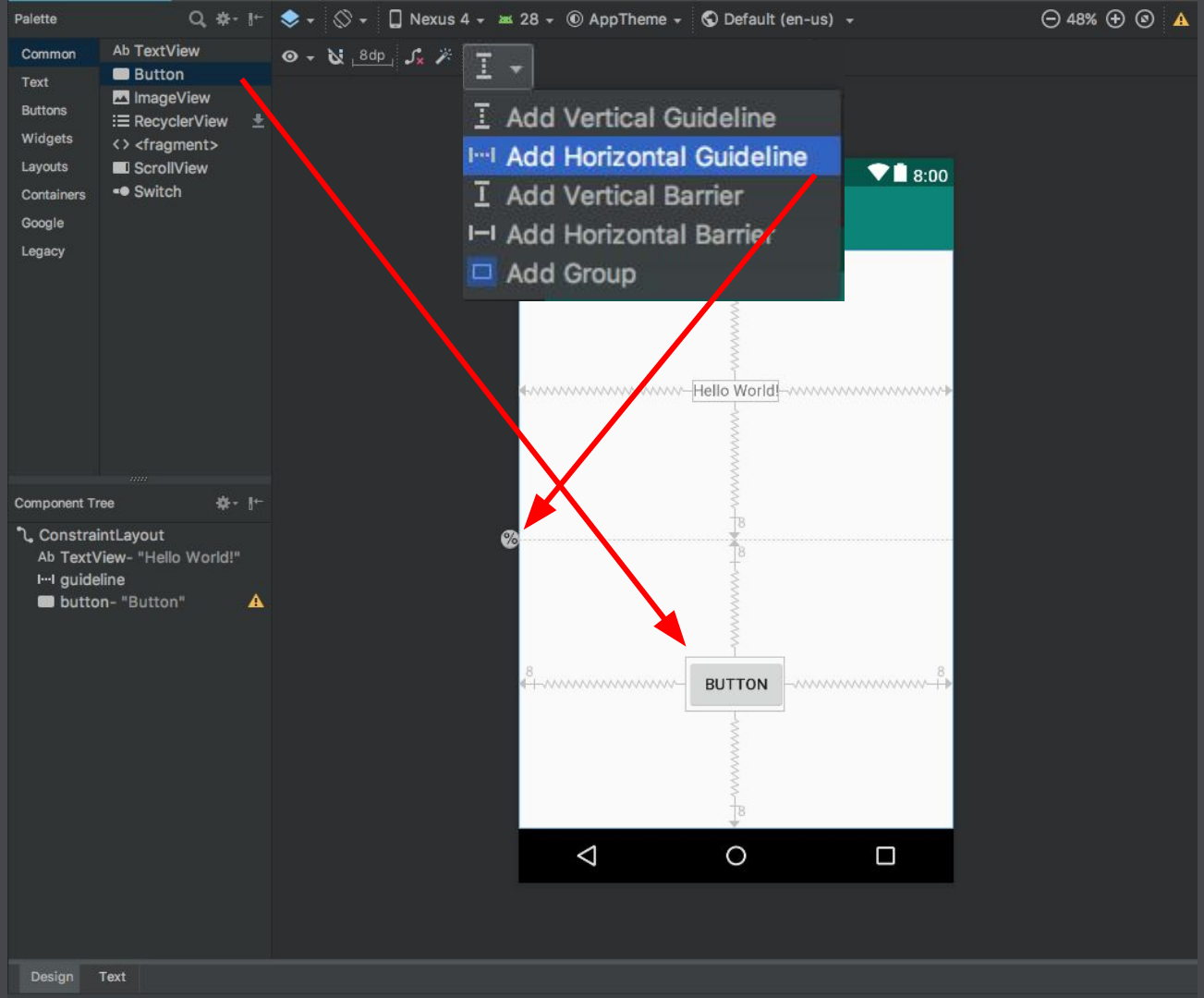
אבל תכלסססססס

How does the layout look?


ClickCounter
app
src
main
res
layout
activity_main.xml
Android
app
manifests
AndroidManifest.xml
java
com.fundamentals.academytlv.refael.clickcounter
MainActivity
com.fundamentals.academytlv.refael.clickcounter (androidTest)
com.fundamentals.academytlv.refael.clickcounter (test)
generatedJava
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
build.gradle (Project: ClickCounter)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
activity_main.xml
MainActivity.java
AndroidManifest.xml
<?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">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World!"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
android.support.constraint.ConstraintLayout
Design
Text
Preview
Nexus 4
28
AppTheme
50%
8:00
Click Counter
Hello World!
Build
Sync
Build: completed successfully at 29/10/2018, 9:02
Run build /Users/jozeri/AndroidStudioProjects/ClickCounter
Load build
Configure build
Calculate task graph
Run tasks
8 s 181 ms
8 s 105 ms
3 ms
3 s 482 ms
130 ms
4 s 487 ms
Terminal
Build
6: Logcat
TODO
Event Log
Gradle build finished in 8 s 710 ms (57 minutes ago)
18:43
LF
UTF-8
Context: <no context>


ClickCounter
app
src
main
res
layout
activity_main.xml
Android
activity_main.xml
MainActivity.java
AndroidManifest.xml
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Google
Legacy
TextView
Button
ImageView
RecyclerView
<fragment>
ScrollView
Switch
Nexus 4
28
AppTheme
Default (en-us)
48%
Attributes
8dp
app
manifests
AndroidManifest.xml
java
com.fundamentals.academytlv.refael.clickcounter
MainActivity
com.fundamentals.academytlv.refael.clickcounter (androidTest)
com.fundamentals.academytlv.refael.clickcounter (test)
generatedJava
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
build.gradle (Project: ClickCounter)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
Component Tree
ConstraintLayout
TextView- "Hello World!"
8:00
Click Counter
Hello World!
Design
Text
Build
Sync
Build: completed successfully at 29/10/2018, 9:02
Run build /Users/jozeri/AndroidStudioProjects/ClickCounter
Load build
Configure build
Calculate task graph
Run tasks
8 s 181 ms
8 s 105 ms
3 ms
3 s 482 ms
130 ms
4 s 487 ms
Terminal
Build
6: Logcat
TODO
Event Log
Gradle build finished in 8 s 710 ms (59 minutes ago)
Context: <no context>

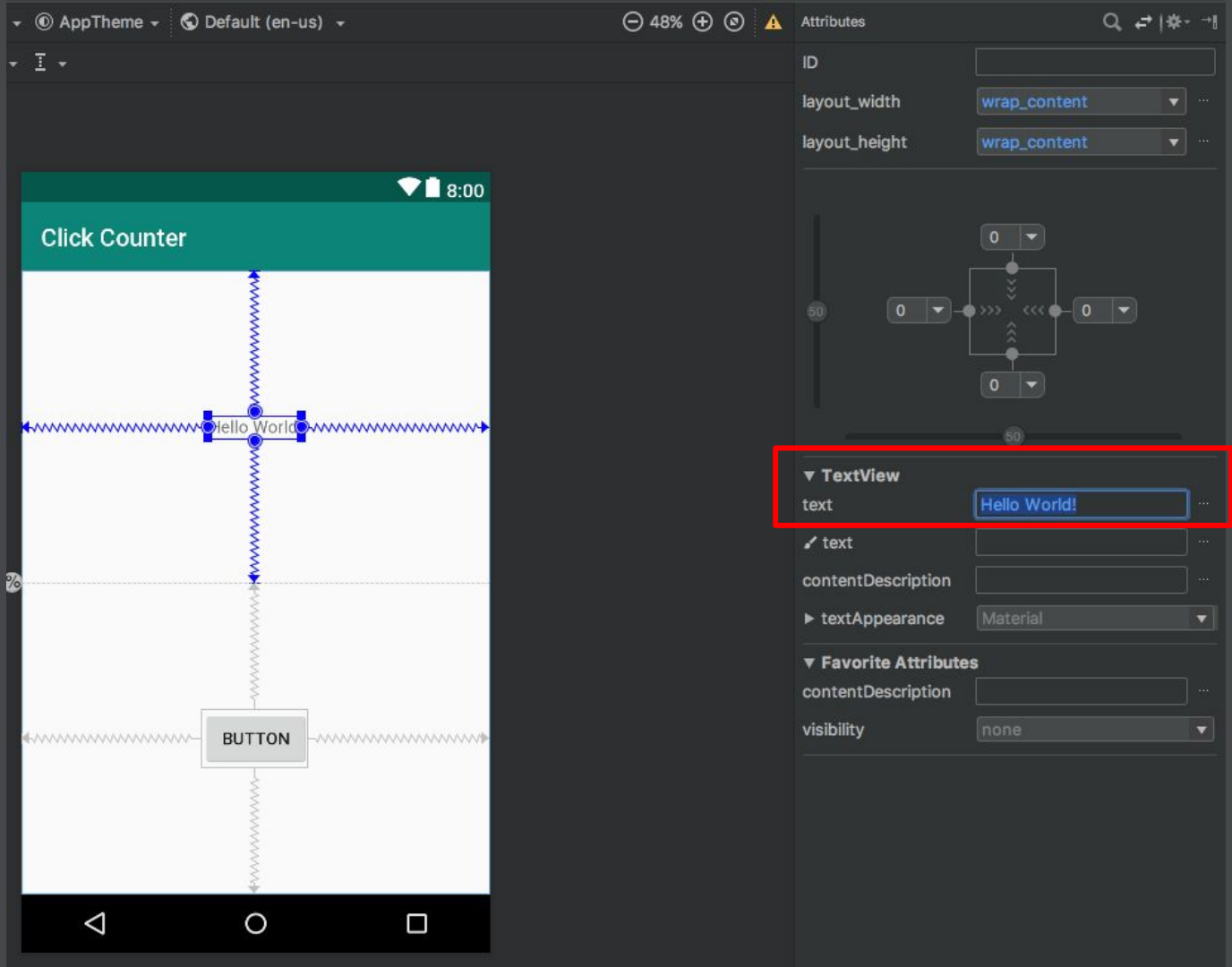

activity_main.xml
MainActivity.java
AndroidManifest.xml
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Google
Legacy
Ab TextView
Button
ImageView
RecyclerView
<fragment>
ScrollView
Switch
Component Tree
ConstraintLayout
Ab TextView- "Hello World!"
Nexus 4
28
AppTheme
Default (en-us)
48%
8dp
8:00
Click Counter
Hello World!
Attributes
ID
layout_width
wrap_content
layout_height
wrap_content
TextView
text
Hello World!
text
contentDescription
textAppearance
Material
Favorite Attributes
contentDescription
visibility
none
View all attributes
Design
Text
Fabric
Gradle
Assistant


Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Google
Legacy
Ab TextView
Button
ImageView
RecyclerView
<fragment>
ScrollView
Switch
Component Tree
ConstraintLayout
Ab TextView- "Hello World!"
guideline
button- "Button"
Nexus 4
28
AppTheme
Default (en-us)
48%
8dp
Add Vertical Guideline
Add Horizontal Guideline
Add Vertical Barrier
Add Horizontal Barrier
Add Group
8:00
Hello World!
BUTTON
Design
Text

# activity_main.xml

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintBottom_toTopOf="@+id/guideline"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<android.support.constraint.Guideline
    ... />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline" />
```

# What do we want to do?!


AppTheme
Default (en-us)
48%
Attributes
ID
layout_width
wrap_content
layout_height
wrap_content
8:00
Click Counter
Hello World
BUTTON
0
0
0
0
50
50
TextView
text
Hello World!
text
contentDescription
textAppearance
Material
Favorite Attributes
contentDescription
visibility
none

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="0"
    app:layout_constraintBottom_toTopOf="@+id/guideline"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<android.support.constraint.Guideline
    ... />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="CLICK ME!"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline" />
```

activity_main.xml

But how do we change the text size?

```
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0"
        android:textSize="36sp"
        app:layout_constraintBottom_toTopOf="@+id/guideline"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <android.support.constraint.Guideline
        ... />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="CLICK ME!"
        android:textSize="36sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
```

Question.. what's our problem here in this code?

Wait.. SP?! What is SP? Wait a few slides

We have the value 36sp saved twice!

NEVER REPEAT YOURSELF!

# The Alt+Enter Magic

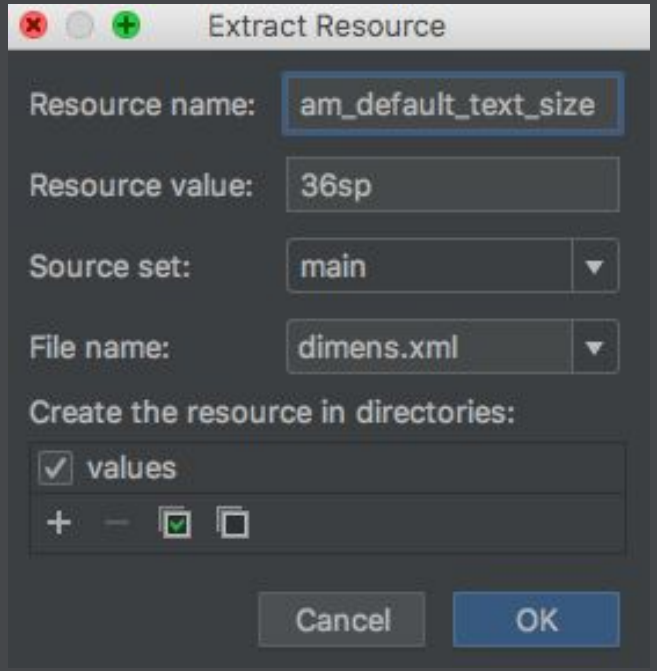
**Alt + Enter** key binding for quick fix errors.

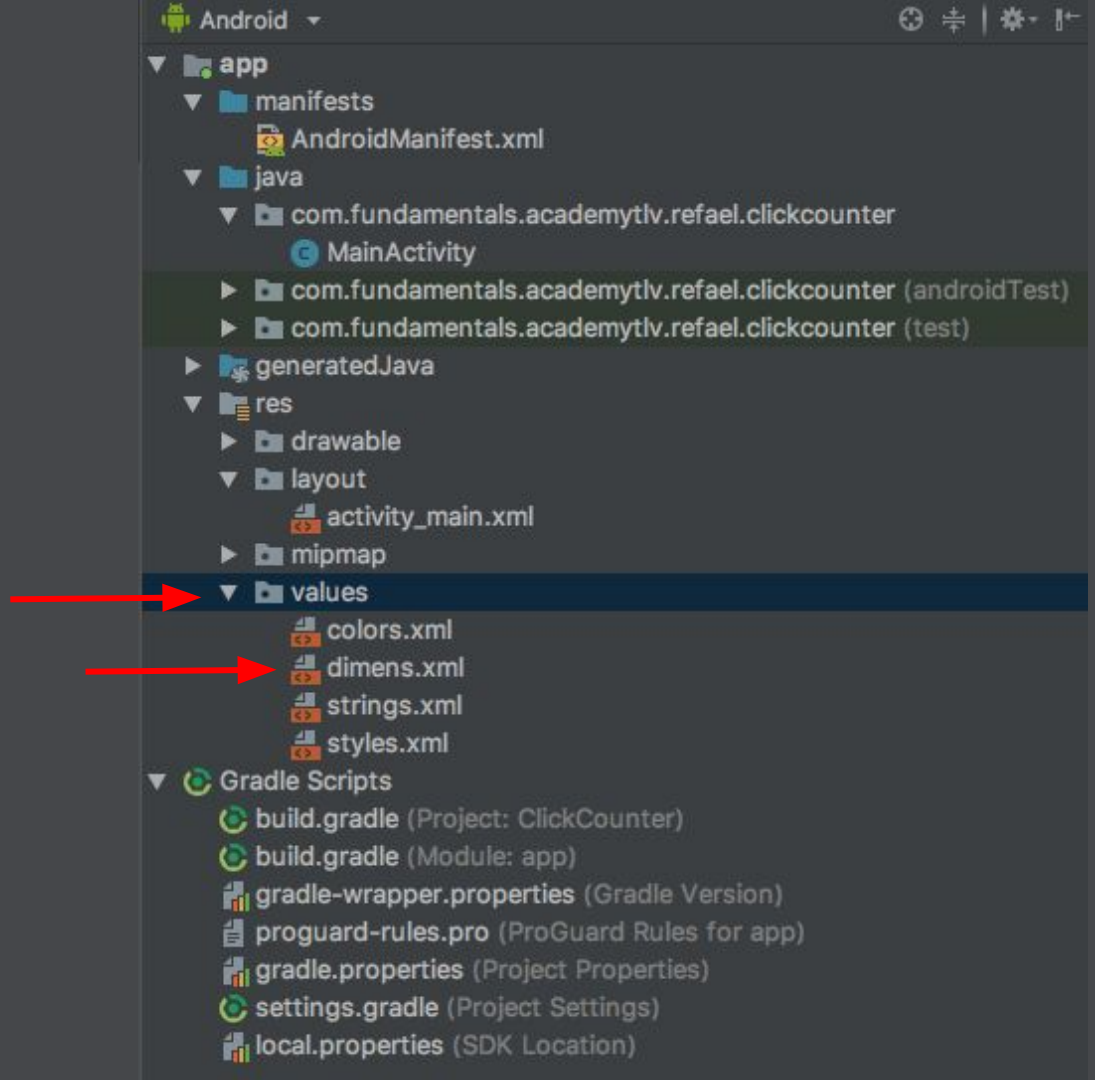
It's **context aware**,
and can do a lot of magic for you!

https://developer.android.com/sdk/installing/studio-tips.html

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="0"
    android:textSize="36sp"
    app:layout_constrain
    app:layout_constrain
    app:layout_constrain
    app:layout_constrain
```

Extract dimension resource
Override Resource in Other Configuration...
Inject language or reference




Android
app
manifests
AndroidManifest.xml
java
com.fundamentals.academytlv.refael.clickcounter
MainActivity
com.fundamentals.academytlv.refael.clickcounter (androidTest)
com.fundamentals.academytlv.refael.clickcounter (test)
generatedJava
res
drawable
layout
activity_main.xml
mipmap
values
colors.xml
dimens.xml
strings.xml
styles.xml
Gradle Scripts
build.gradle (Project: ClickCounter)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)

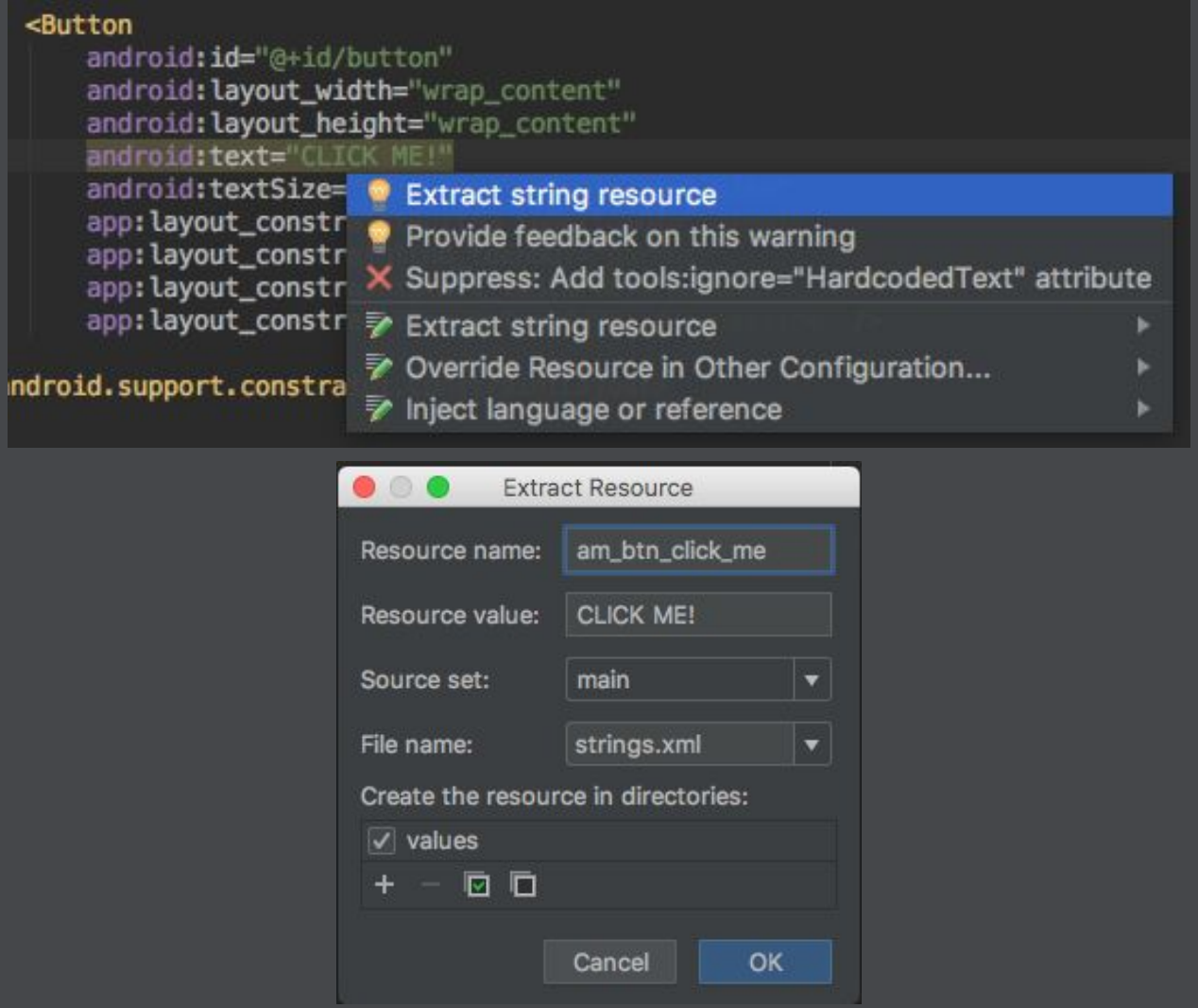
dimens.xml

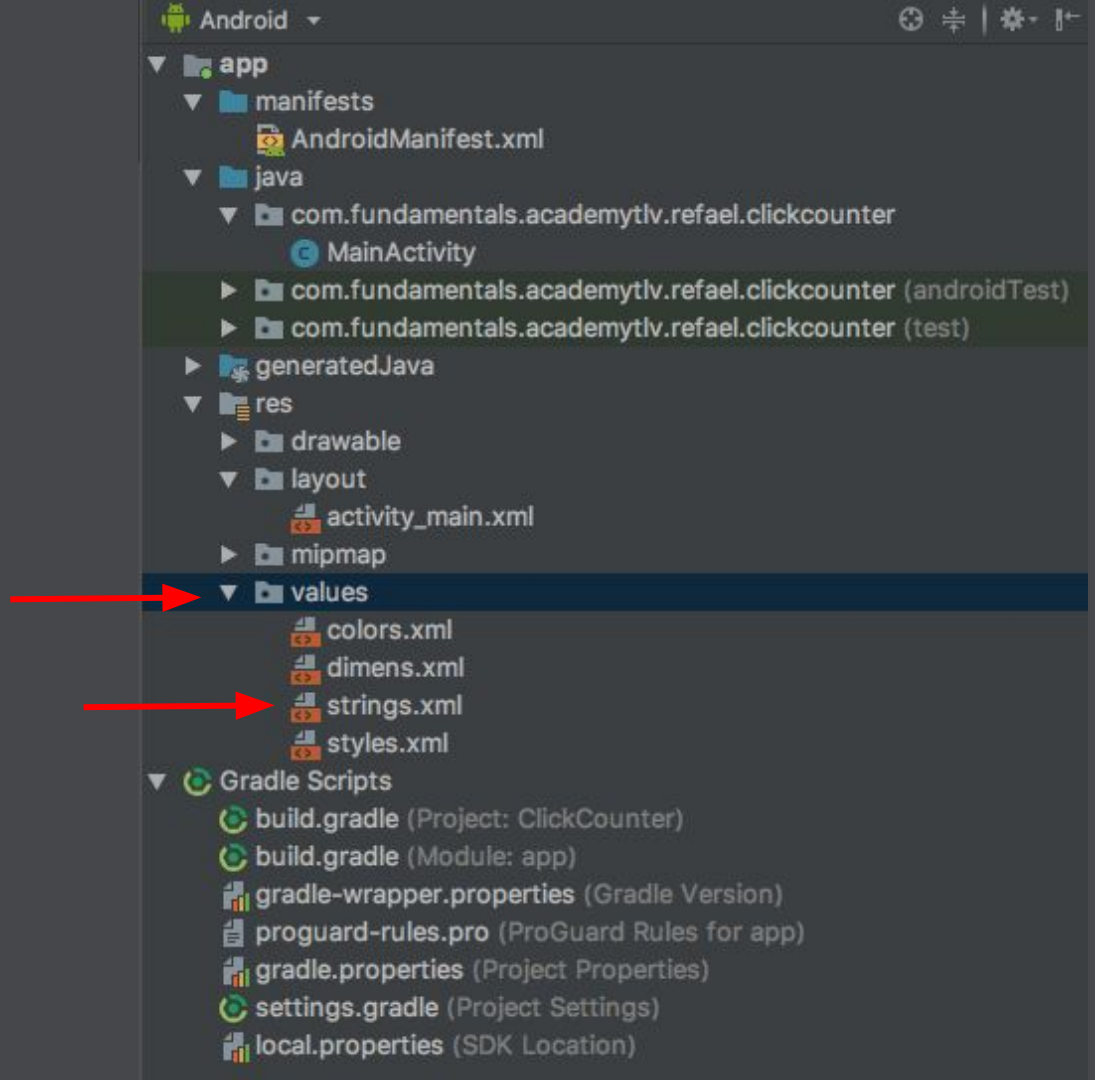
```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <dimen name="am_default_text_size">36sp</dimen>
</resources>
```

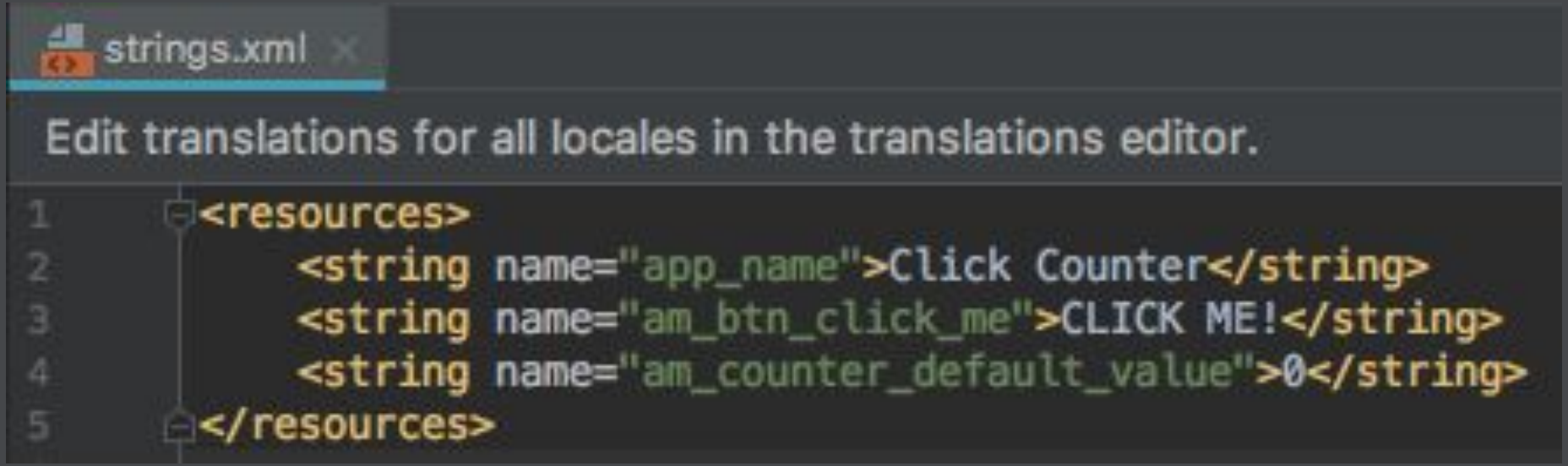
```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="0"
    android:textSize="@dimen/am_default_text_size"
    app:layout_constraintBottom_toTopOf="@+id/guideline"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<android.support.constraint.Guideline
    ... />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="CLICK ME!"
    android:textSize="@dimen/am_default_text_size"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="CLICK ME!"
    android:textSize="@dimen/am_default_text_size"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline" />
```


26 <Button
27 android:id="@+id/button"
28 android:layout_width="wrap_content"
29 android:layout_height="wrap_content"
30 android:text="CLICK ME!"
Hardcoded string "CLICK ME!", should use @string resource more... (⌘F1)
33 app:layout_constraintEnd_toEndOf="parent"
34 app:layout_constraintStart_toStartOf="parent"
35 app:layout_constraintTop_toTopOf="@+id/guideline" />


<Button
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="CLICK ME!"
android:textSize=
app:layout_constr
app:layout_constr
app:layout_constr
app:layout_constr
ndroid.support.constra
Extract string resource
Provide feedback on this warning
Suppress: Add tools:ignore="HardcodedText" attribute
Extract string resource
Override Resource in Other Configuration...
Inject language or reference
Extract Resource
Resource name:
am_btn_click_me
Resource value:
CLICK ME!
Source set:
main
File name:
strings.xml
Create the resource in directories:
values
Cancel
OK


Android
app
manifests
AndroidManifest.xml
java
com.fundamentals.academytlv.refael.clickcounter
MainActivity
com.fundamentals.academytlv.refael.clickcounter (androidTest)
com.fundamentals.academytlv.refael.clickcounter (test)
generatedJava
res
drawable
layout
activity_main.xml
mipmap
values
colors.xml
dimens.xml
strings.xml
styles.xml
Gradle Scripts
build.gradle (Project: ClickCounter)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)


strings.xml
Edit translations for all locales in the translations editor.
1 <resources>
2 <string name="app_name">Click Counter</string>
3 <string name="am_btn_click_me">CLICK ME!</string>
4 <string name="am_counter_default_value">0</string>
5 </resources>

```
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/am_counter_default_value"
        android:textSize="@dimen/am_default_text_size"
        app:layout_constraintBottom_toTopOf="@+id/guideline"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/am_btn_click_me"
        android:textSize="@dimen/am_default_text_size"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/guideline" />
```

# Click Counter

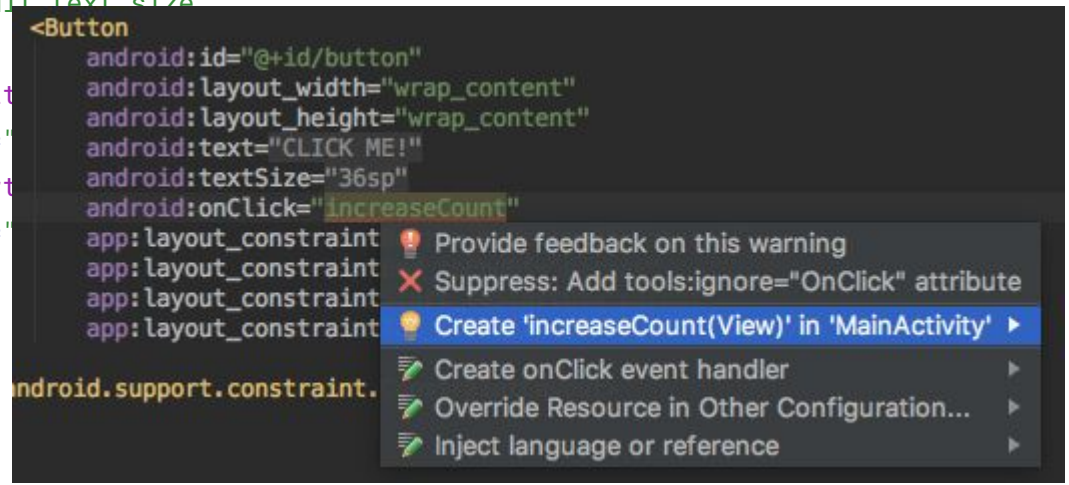
# Let’s make a click counter!

We’ll want to change the TextView’s text at runtime,

so we’ll need to grab it after the views are inflated.

# Add onClick

```
<Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/am_btn_click_me"
        android:textSize="@dimen/am_default_text_size"
        android:onClick="increaseCount"
        app:layout_constraintBottom_toBott
        app:layout_constraintEnd_toEndOf="
        app:layout_constraintStart_toStart
        app:layout_constraintTop_toTopOf="
```



# The method was created on the Activity.java

```
package com.fundamentals.academytlv.refael.clickcounter;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
  }

  public void increaseCount(View view) {


  }
}
```

# Let’s make a click counter!

We’ll need a counter variable.
Let’s make an int, and init it to 0.

```
public class MainActivity extends AppCompatActivity {

  private int mCounter = 0;

  public void increaseCount(View view) {

  }
}
```

# Fill in the method that happens on click

```
public class MainActivity extends AppCompatActivity {

  private int mCounter = 0;

  //activity code ommited...

  public void increaseCount(View view) {
    mCounter++;
  }
}
```

# Update UI

# Assign view ID

```
<TextView
        android:id="@+id/am_tv_counter"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/am_counter_default_value"
        android:textSize="@dimen/am_default_text_size"
        app:layout_constraintBottom_toTopOf="@+id/guideline"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

# Reference to the view on code

```
public class MainActivity extends AppCompatActivity {

  private int mCounter = 0;

  private TextView mCounterTextView;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mCounterTextView = findViewById(R.id.am_tv_counter);
  }

  public void increaseCount(View view) {
    mCounter++;
  }
}
```

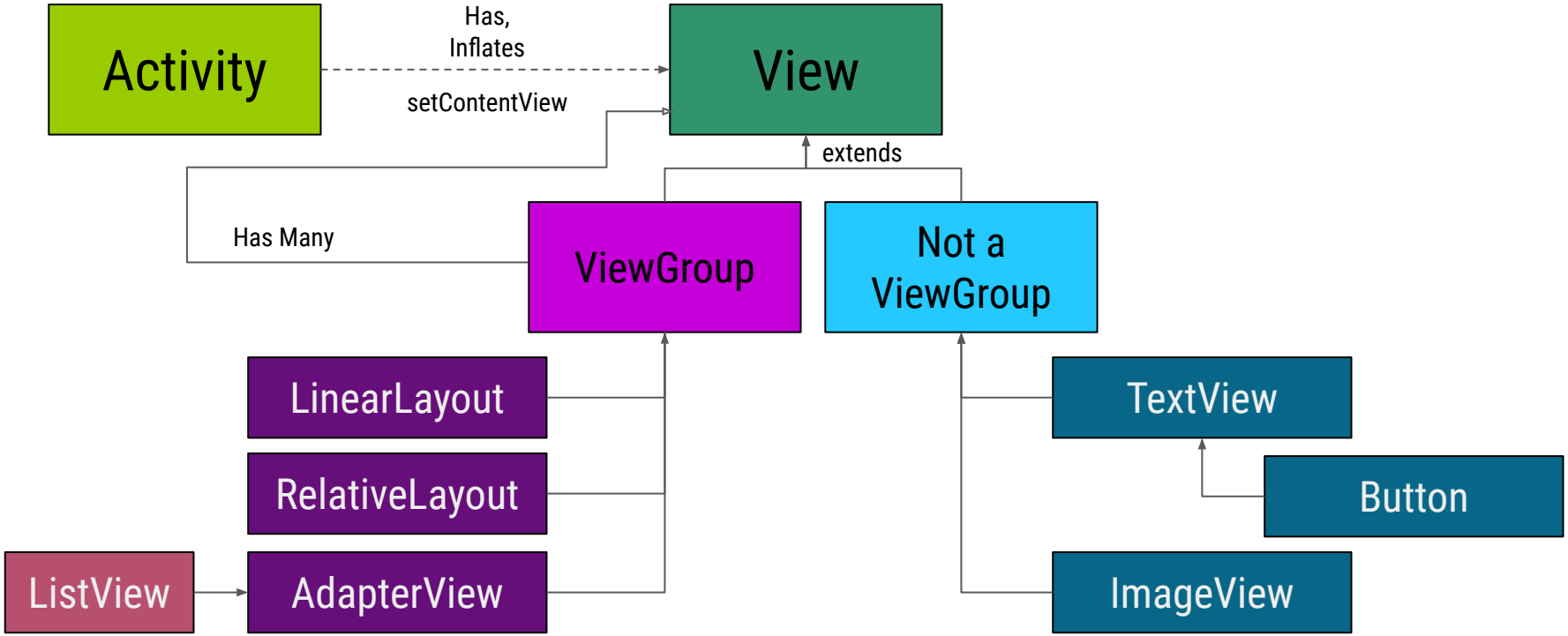
# Set View with counter

```
public class MainActivity extends AppCompatActivity {

  private int mCounter = 0;

  private TextView mCounterTextView;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mCounterTextView = findViewById(R.id.am_tv_counter);
  }

  public void increaseCount(View view) {
    mCounter++;
    mCounterTextView.setText(String.valueOf(mCounter));
  }
}
```

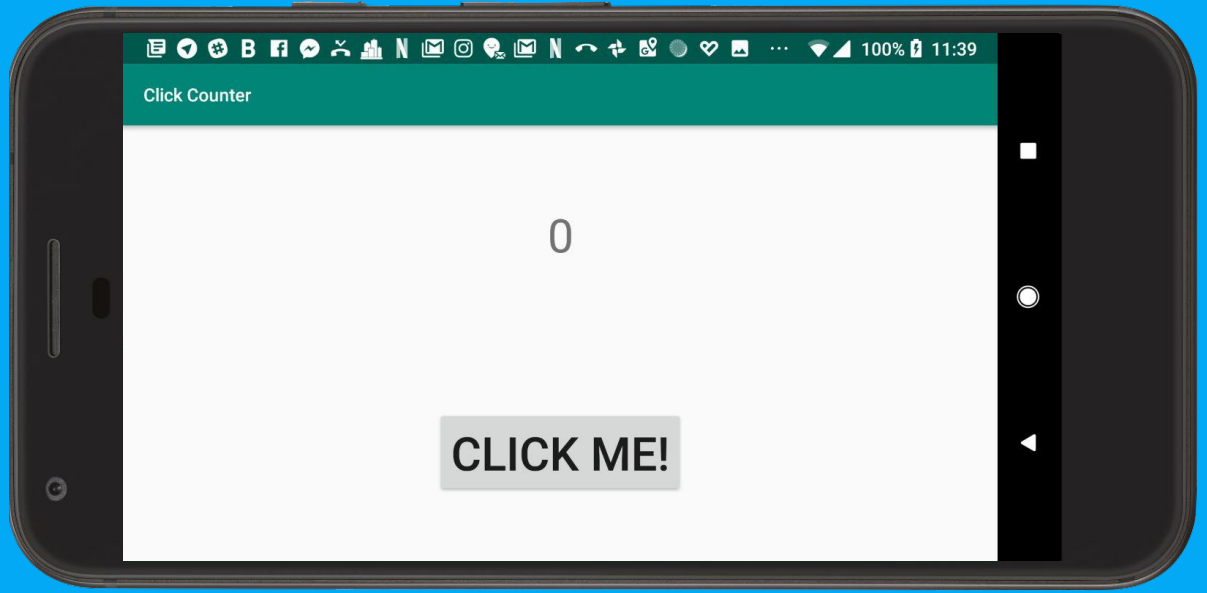
# OMG, It looks great!

# Questions?

# Quick UML Recap

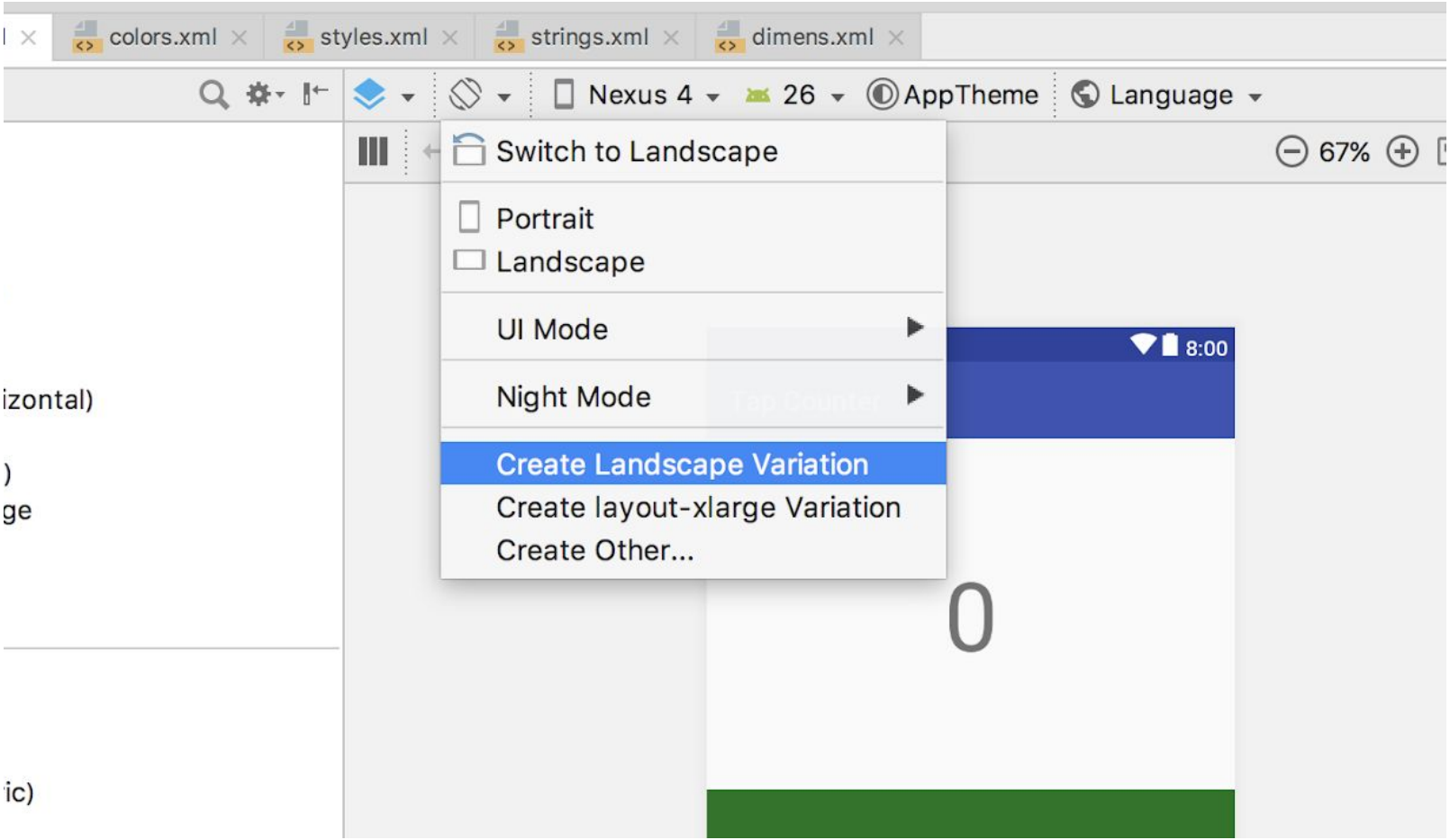


# Rotate the device



There’s a bug with the counter…
I know…
but we’ll get to it later...

# Create a different layout for landscape


colors.xml
styles.xml
strings.xml
dimens.xml
Nexus 4
26
AppTheme
Language
67%
Switch to Landscape
Portrait
Landscape
UI Mode
Night Mode
Create Landscape Variation
Create layout-xlarge Variation
Create Other...
8:00
0
izontal)
)
ge
ic)

# Activity_main.xml (#2)

```
<android.support.constraint.Guideline
        android:id="@+id/guideline2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.5" />
```

layout/activity_main.xml | land/activity_main.xml | MainActivity.java

```xml
<?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">

    <TextView
        android:id="@+id/am_tv_counter"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0"
        android:textSize="36sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/guideline2"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <android.support.constraint.Guideline
        android:id="@+id/guideline2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.5" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="increaseCount"
        android:text="@string/am_btn_click_me"
        android:textSize="36sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="@+id/guideline2"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

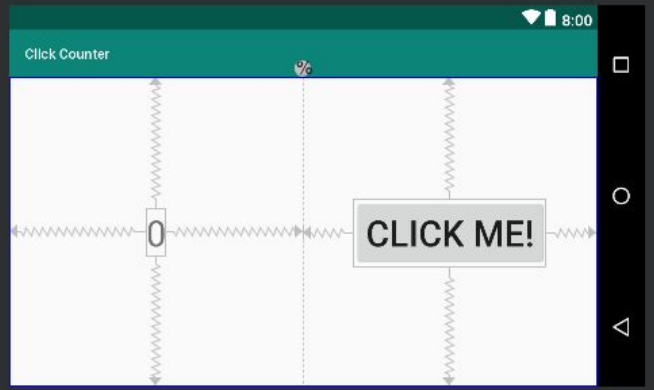
android.support.constraint.ConstraintLayout

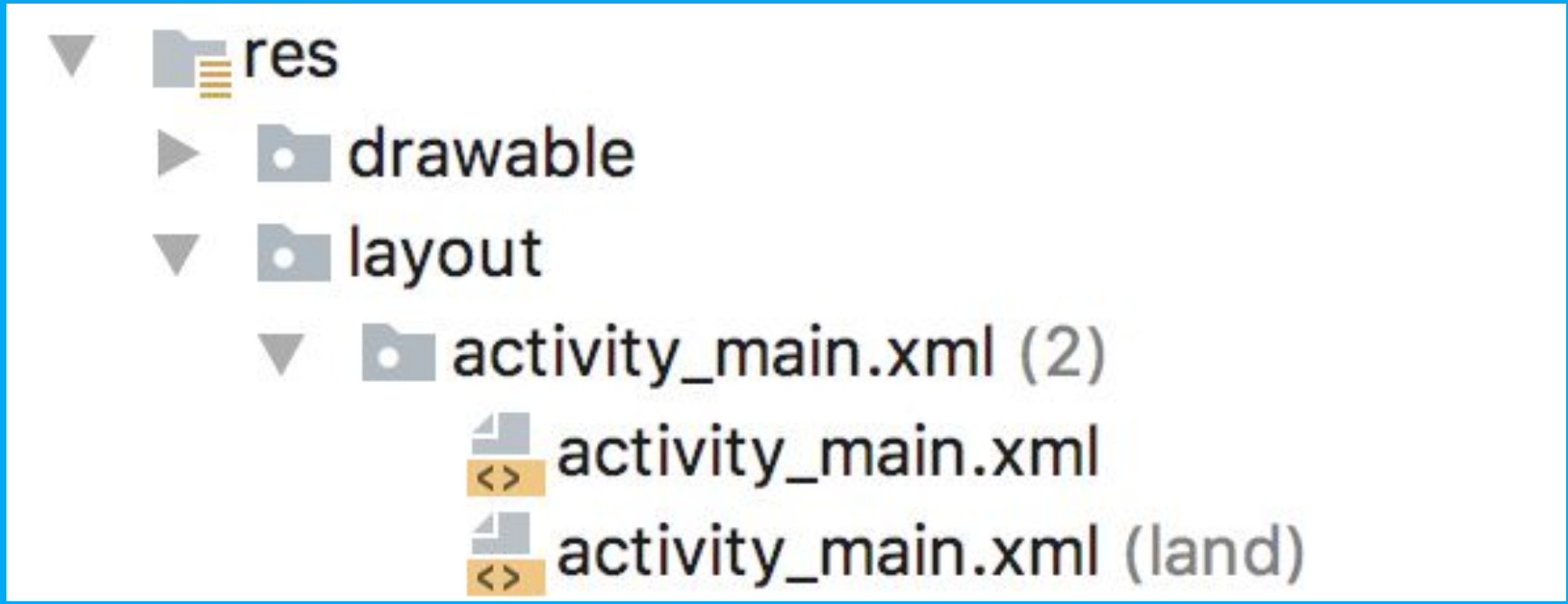
Design | Text

Preview

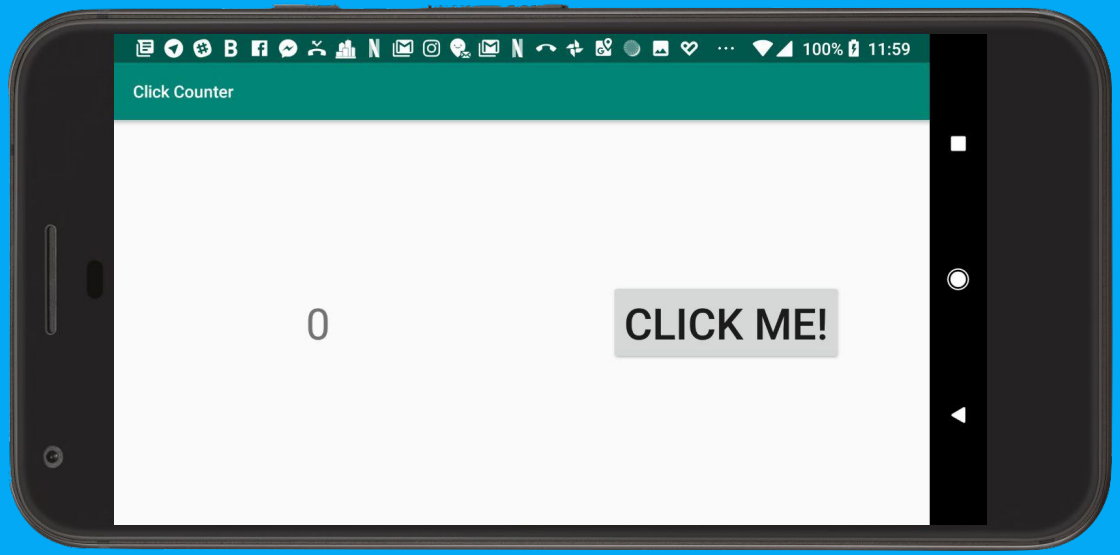
Palette

Nexus 4 | 28 | AppTheme | 36% | 8dp




res
drawable
layout
activity_main.xml (2)
activity_main.xml
activity_main.xml (land)

# Now: Rotate the device



# DEVICE FRAGMENTATION

# First android device

**HTC Dream**

320x480px (2:3)

180 ppi

3.2” (81mm)


http://en.wikipedia.org/wiki/HTC_Dream

# Current Android Device

**Samsung Galaxy Note 9**

2960x1440 (18.5:9)

516 ppi

6.4” display

# Big Android Device

**Tab S4 (Tablet)**

2560x1600px (16:10)

287ppi

10.5”

# Small Android Device

**LG G Watch**

Most

~~Same~~ techniques apply


http://en.wikipedia.org/wiki/Android_Wear

# The difference is Big.


ANDROID WEAR

PHONES

TABLETS

ANDROID TV

ANDROID AUTO

# Differences

**Pixel Density**

Screen size

Orientation

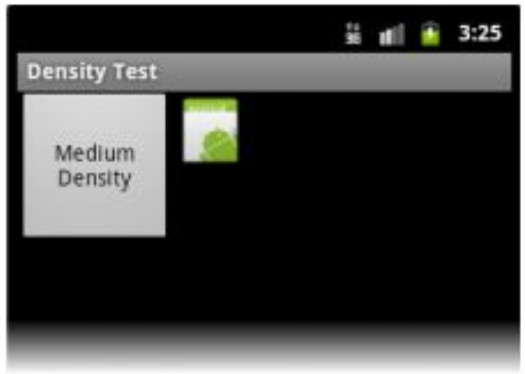
Device Class

OS Version

Language

Accessibility

# Pixel Density: Problem and solution

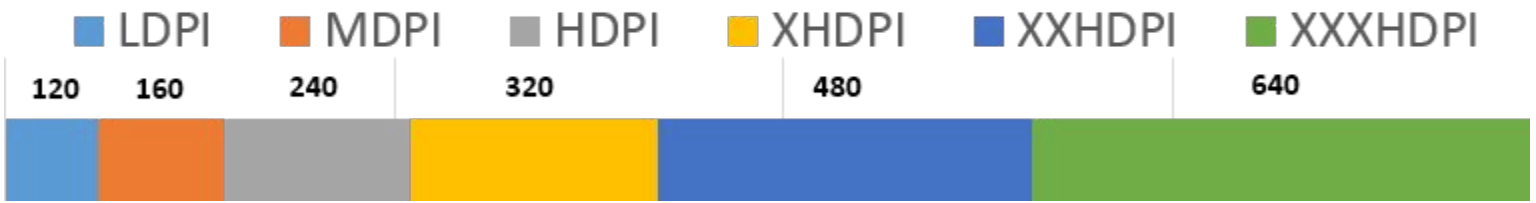







# Pixel Density: Solution #1

Divide the screen sizes to categories



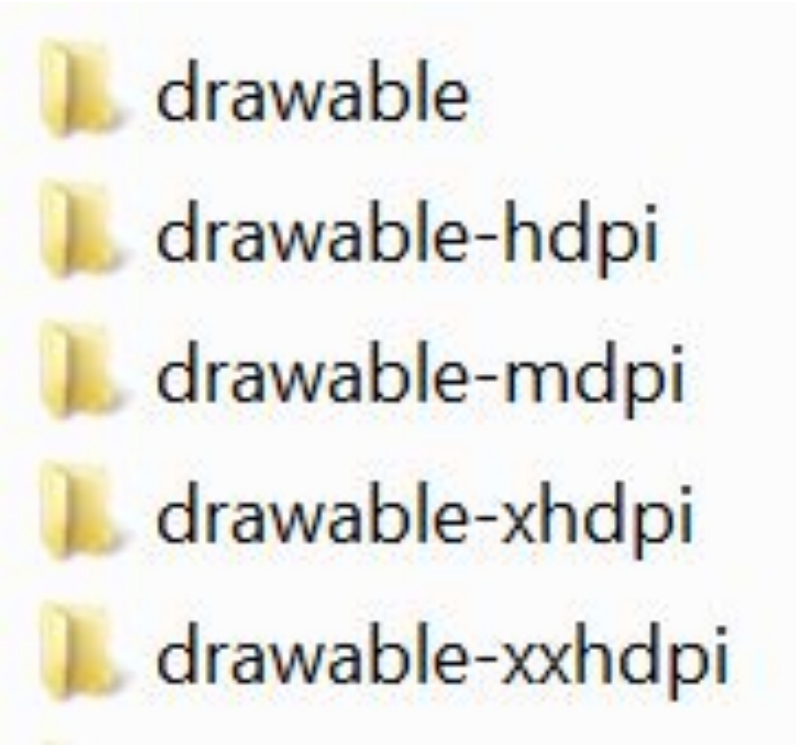

http://developer.android.com/training/basics/supporting-devices/screens.html

# Pixel Density: Solution #1

Have different versions for different screens

Remember the folders from earlier?

# Folder Structure




# Solution #2: Different Layouts

We can have multiple versions of a layout for different screens.

The technique is similar to the one used in **drawables** we saw earlier.

To support tablets, we add `-large` to the directory name. To support landscape we add `-land`.

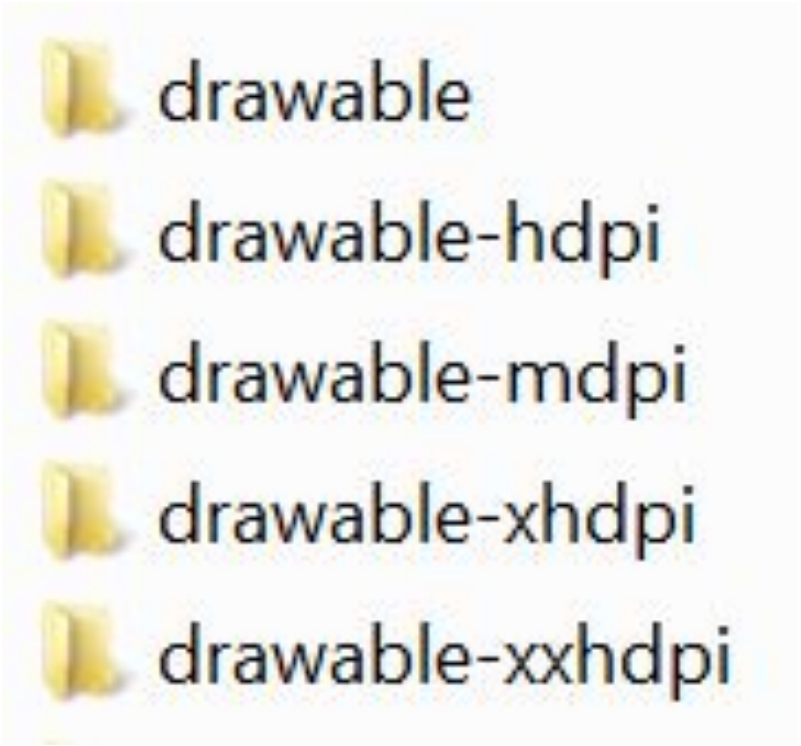
# Solution #2: Different Layouts

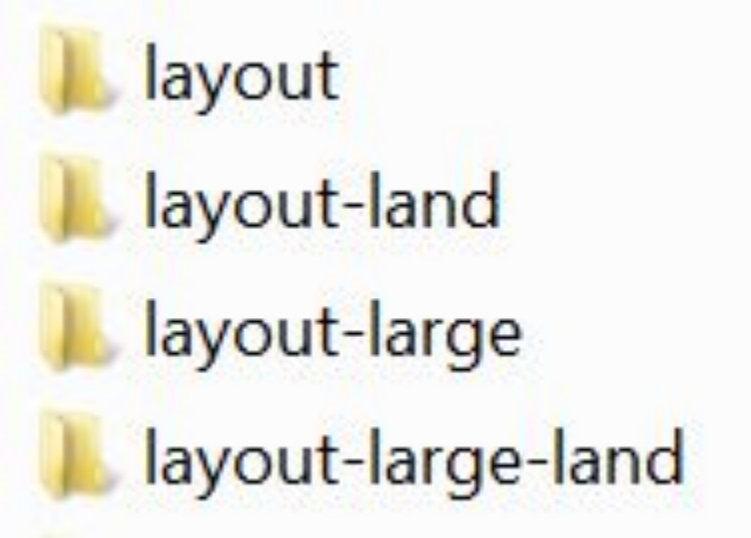
On a large device, the system uses the xml in the /layout-large/ folder.

if the required layout isn't there, the system falls-back one level, and find it in /layout/ folder.

If it won't find it at all, the program will **crash**.

# Folder Structure



# Solution #2: Different Layouts Files

Android uses this mechanism for **a lot** of things:

strings are stored in xml and enable localization easily.

colors and styles are stored in xml files and allow theming.

dimensions also, that can use different screen sizes.

Guide: http://developer.android.com/guide/topics/resources/providing-resources.html

# Resource Qualifiers

- #dpi
- w#dp
- h#dp
- land
- port
- sw#dp

# Resources Types

- Layouts
- Dimensions
- Colors
- Menus
- Styles
- Booleans/Strings/Integers

# Pixel Density: Solution #3

Instead of **Pixels**, that are affected by pixel density,

Use **dp** - Density-independent Pixels.

# Measurement of Text Sizes

Users may change text scale as they please.

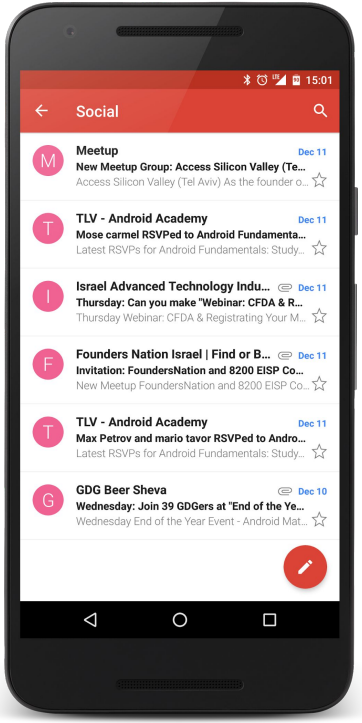
So we can’t use “**points**” or “**pixels**” to define text size
Because these are affected by pixel density, and not by the user’s choice.

Instead, we use **SP** = Scale Independent Pixels

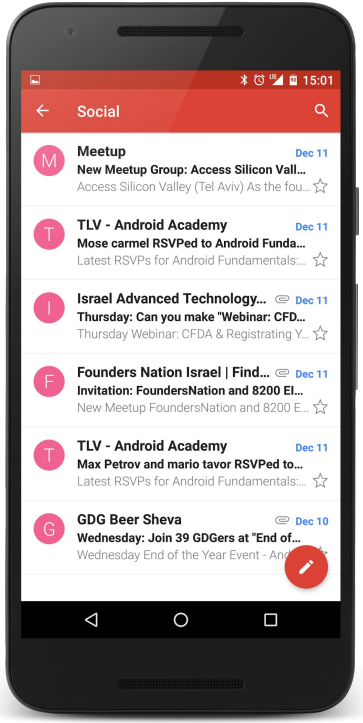
# Measurement of Text Sizes

- **16sp = 16dp @ 100% scale**
- **16sp = 20dp @ 125% scale**

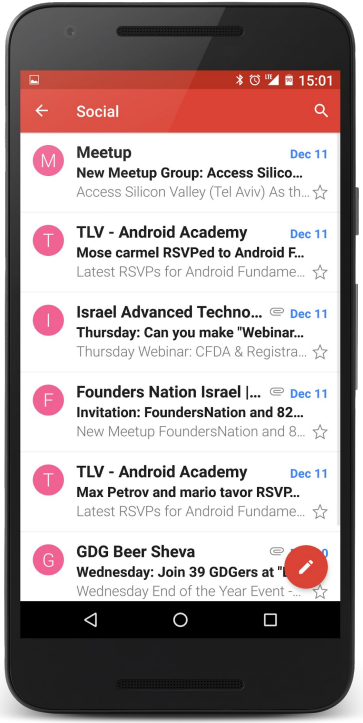
# SP - same as DP but scaleable




Normal




Large




Huge