





# What Is Android?



Android is a mobile operating system developed by Google, primarily for touchscreen smartphones and tablets.

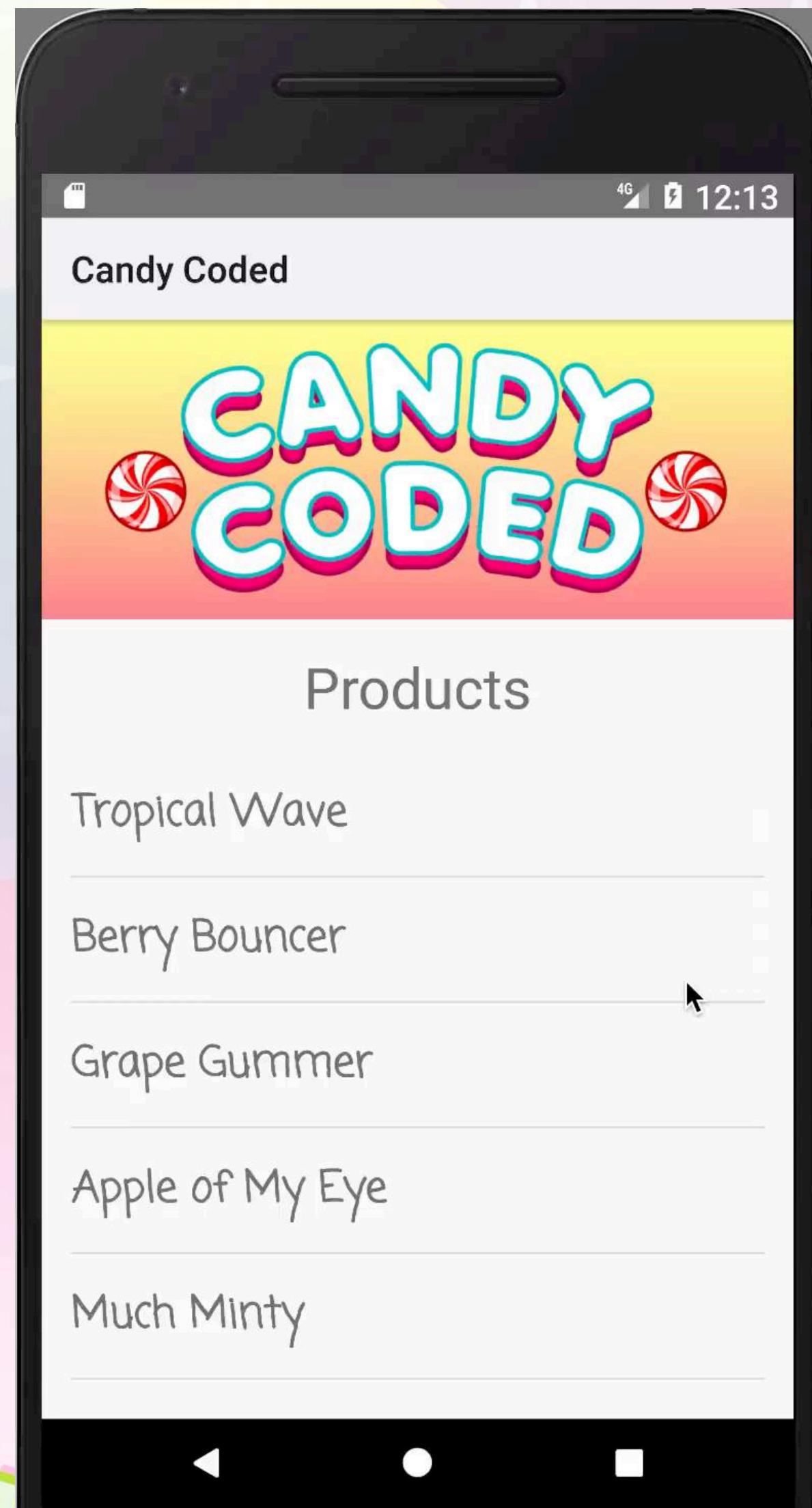
Label

<b>Alpha</b>	Sept 23, 2008	<b>Ice Cream Sandwich</b>	Oct 18, 2011
<b>Beta</b>	Feb 9, 2009	<b>Jelly Bean</b>	July 9, 2012
<b>Cupcake</b>	April 27, 2009	<b>Kit Kat</b>	Oct 31, 2013
<b>Donut</b>	Sept 15, 2009	<b>Lollipop</b>	Nov 12, 2014
<b>Eclair</b>	Oct 26, 2009	<b>Marshmallow</b>	Oct 5, 2015
<b>Froyo</b>	May 20, 2010	<b>Nougat</b>	Aug 22, 2016
<b>Gingerbread</b>	Dec 6, 2010	<b>Oreo</b>	Aug 21, 2017
<b>Honeycomb</b>	Feb 22, 2011		



# The App We're Going to Make

We're going to make an app to list the candy in our candy store: Candy Coded.



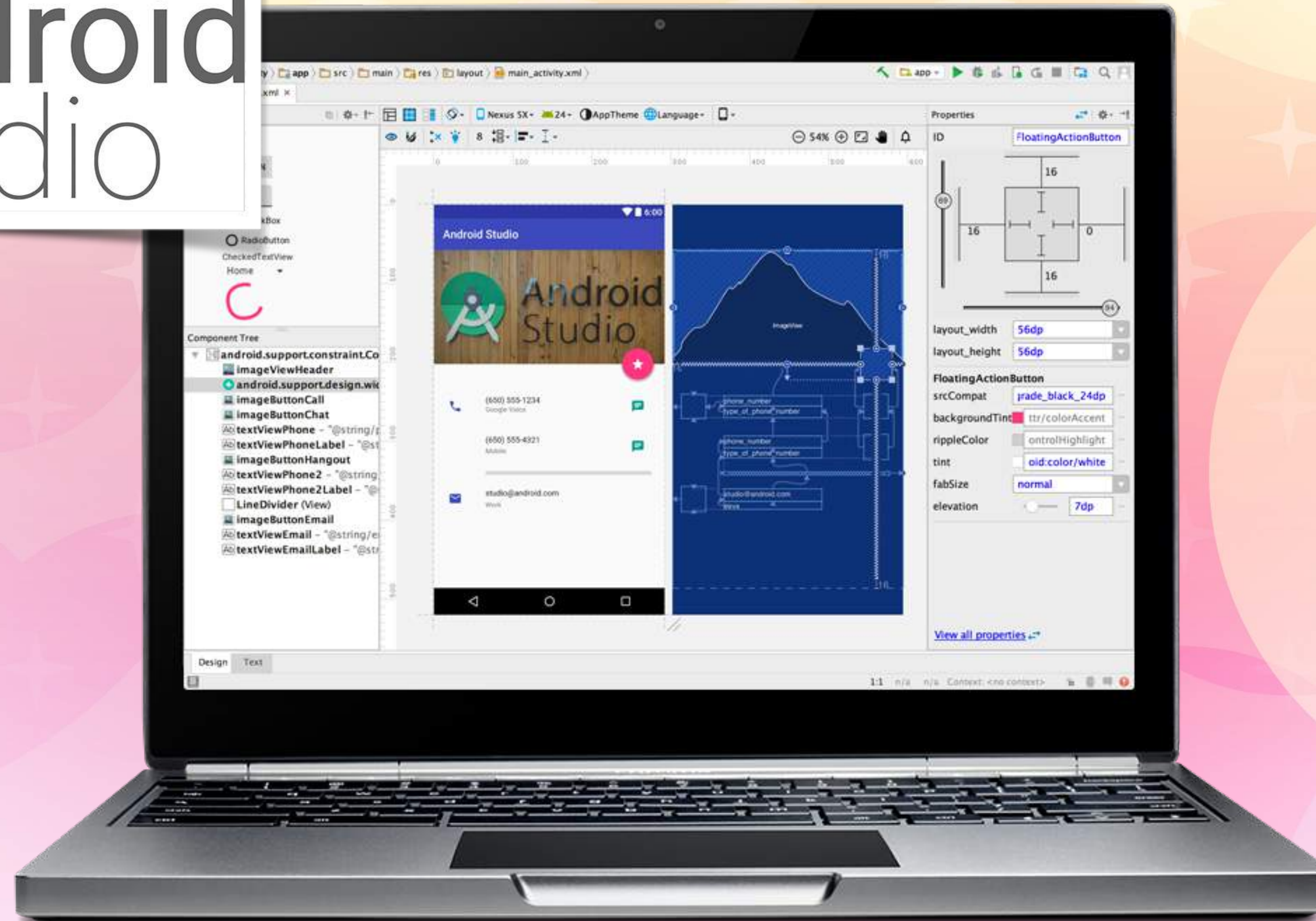
*Our app will use Activity and View classes and Layout files, which are the main building blocks of an Android app*

**TRY  
ANDROID**



# Android Studio Helps You Make Android Apps

Android Studio is the official IDE for Android, and you can download it for free here:  
[go.codeschool.com/install-android-studio](https://go.codeschool.com/install-android-studio)



TRY  
ANDROID



Level 1 – Section 1

# Cooking Up an App

Creating an Android Studio Project

TRY  
ANDROID

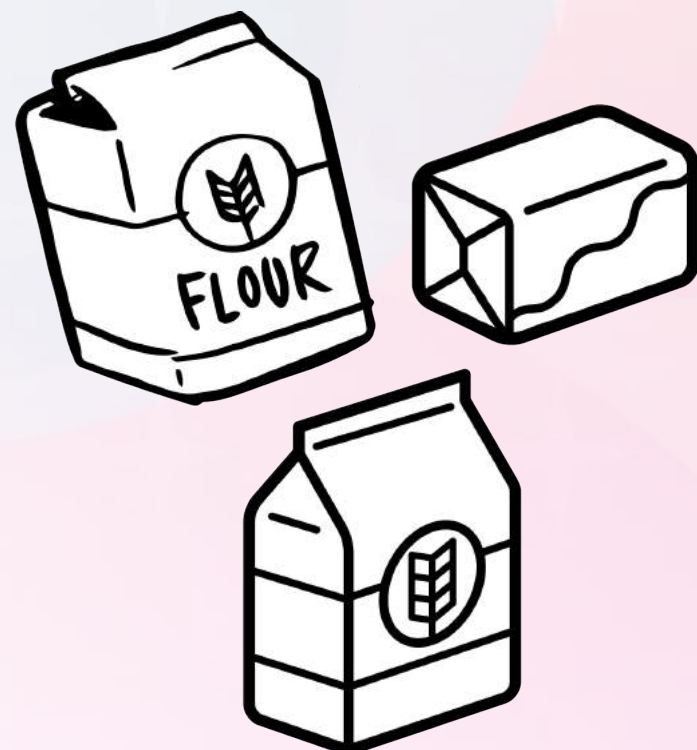


# Apps Are Like Cupcakes

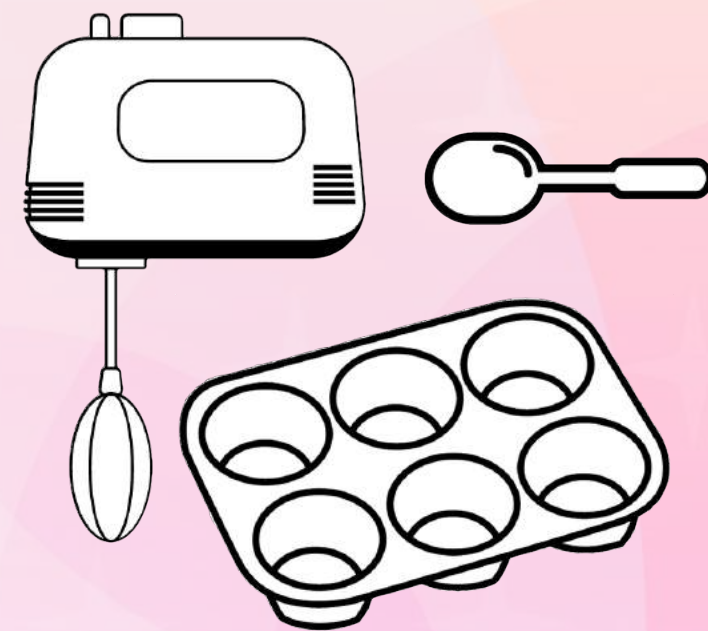


*What do we need to  
bake a cupcake?*

Ingredients



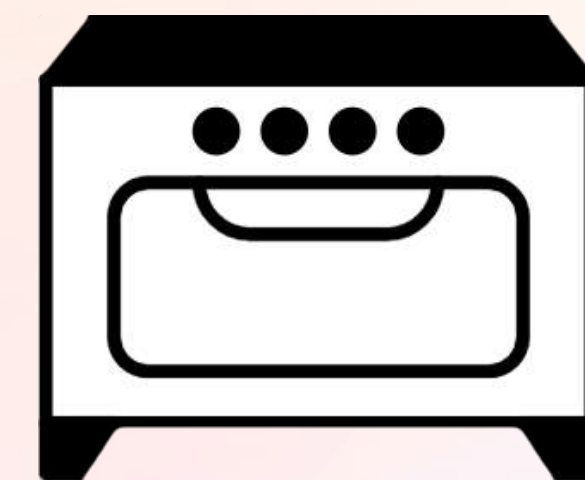
Cooking Utensils



Recipe



Oven



TRY  
ANDROID



# The Android SDK Is Like the Ingredients

The Android SDK provides standard ways to display data in your app, such as activities, layouts, text, images, and buttons.

Ingredients



Text

**Candy Coded**

Images



Buttons

**Add Favorite**

Layout

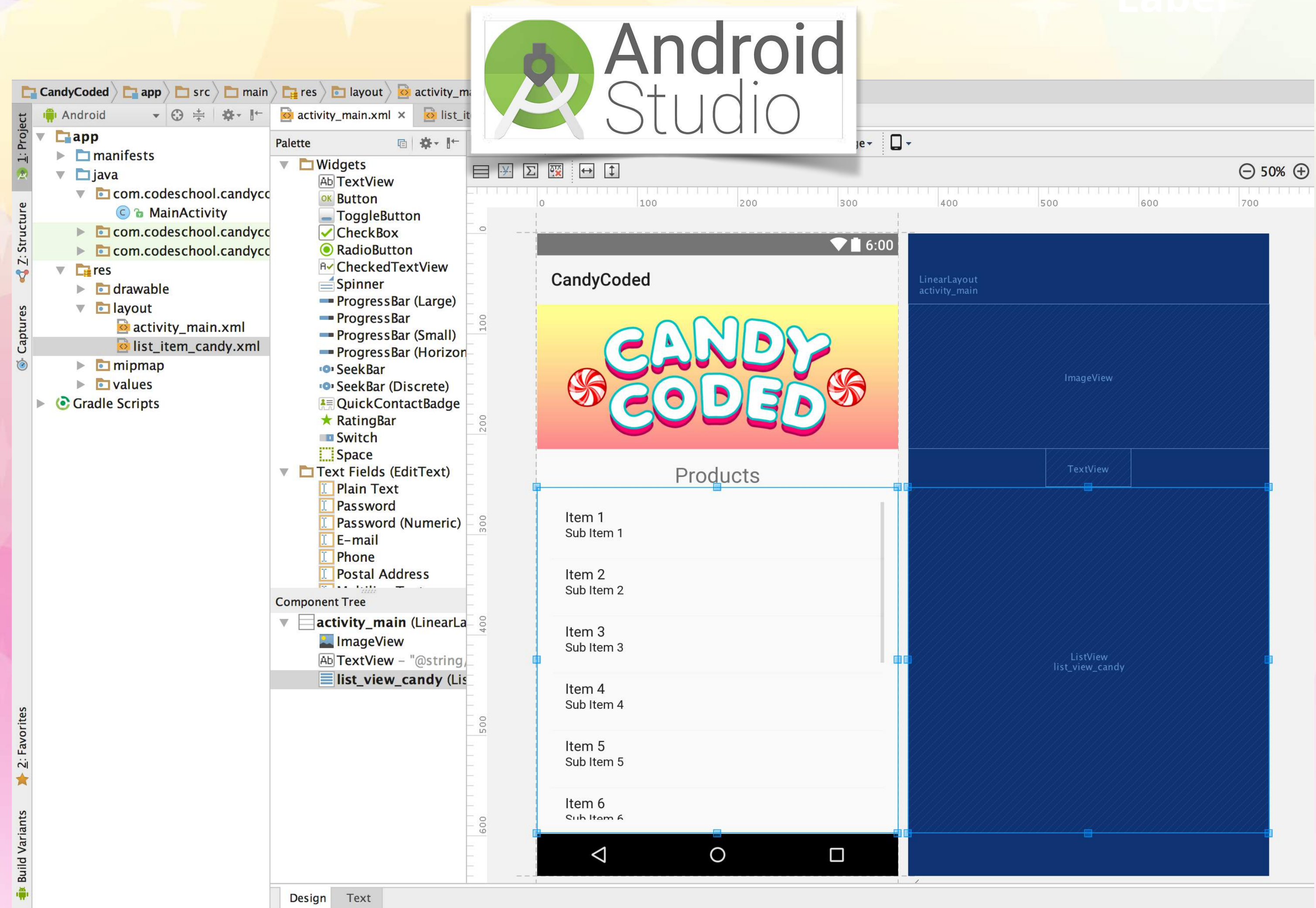
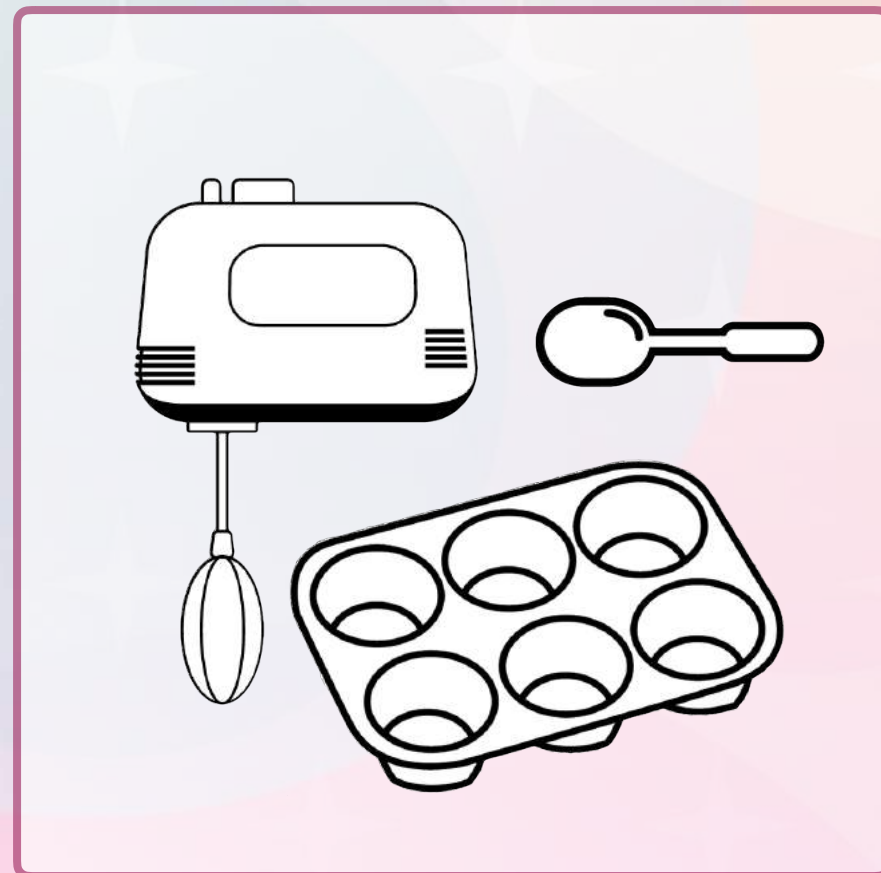
**TRY  
ANDROID**



# Android Studio Is Like the Kitchen Utensils

Android Studio is a free program provided by Google that you use to write Java code and assemble your app.

## Cooking Utensils





# Java Is Like the Instructions for Making Cupcakes

You'll write Java code in Android Studio that works with the Android SDK to display your data.

*You are the chef!*

Recipe



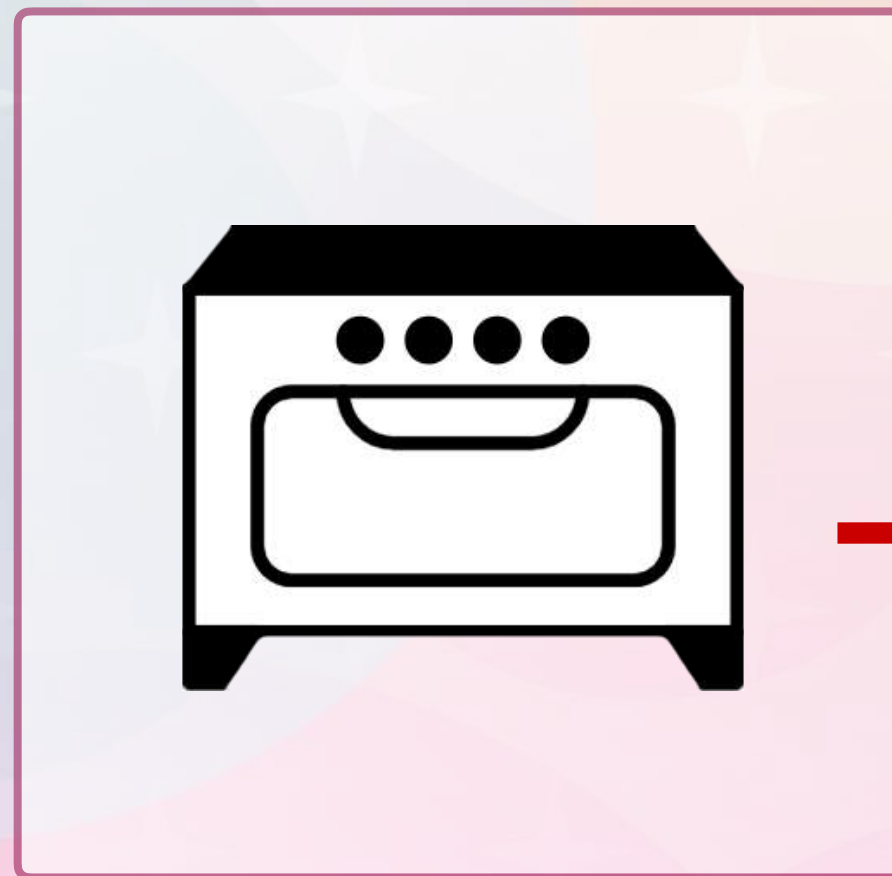
TRY  
ANDROID



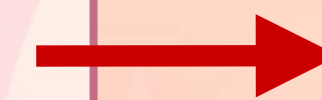
# Gradle Is Like the Oven

Finally, Gradle is the build system in Android Studio that outputs your final app (or cupcake) into a single APK file.

Oven



Gradle



*The final output!*



TRY  
ANDROID



# Screencast: Create & Set Up a New Project

---

TRY  
ANDROID



Level 1 – Section 2

# Cooking Up an App

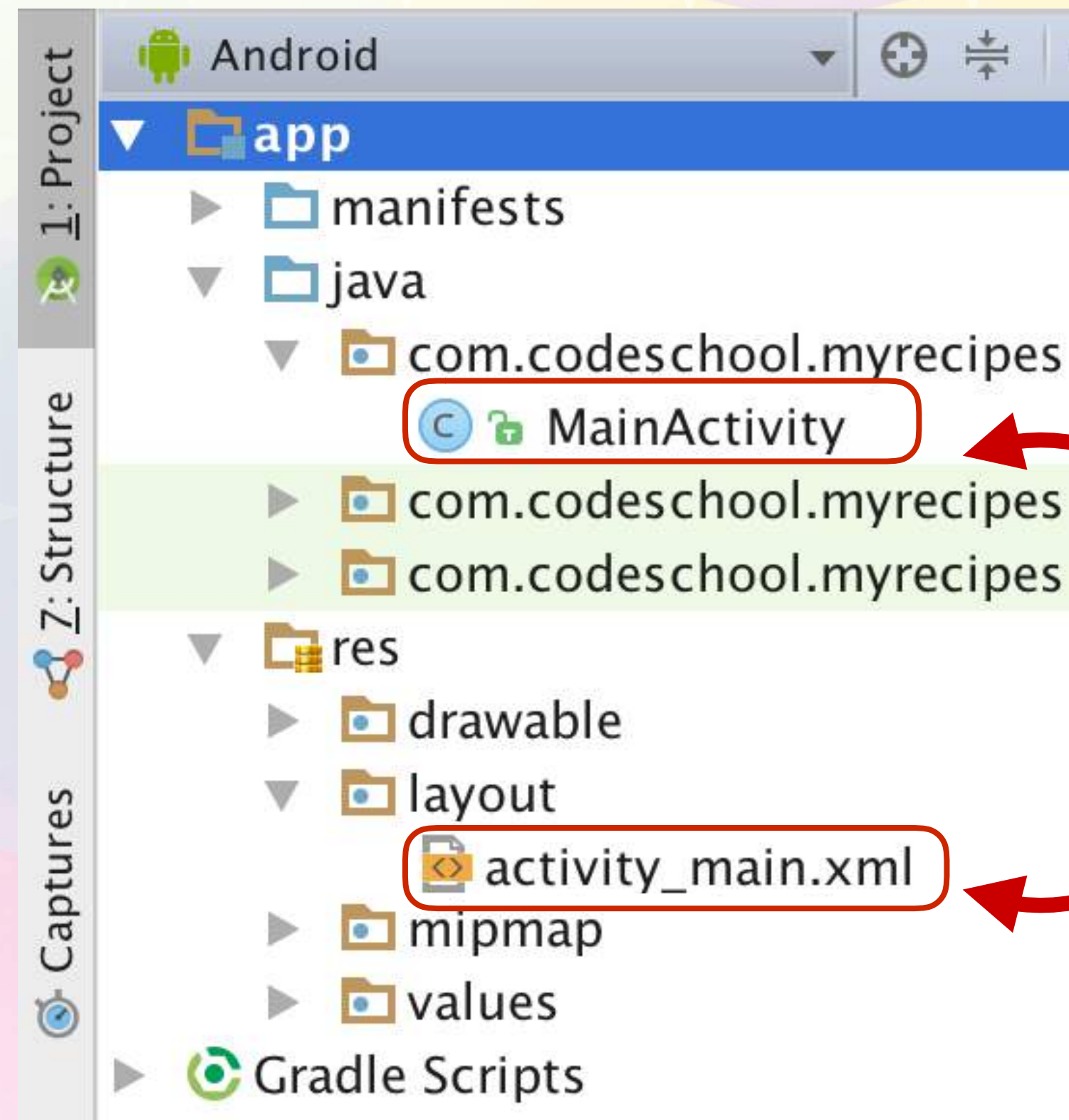
Activities, Layouts & Resources

TRY  
ANDROID



# How Is Our App Launched?

The first screen we see in our app is called the MainActivity, which is launched by the MainActivity.java file and displays the Layout defined in the activity\_main.xml file.

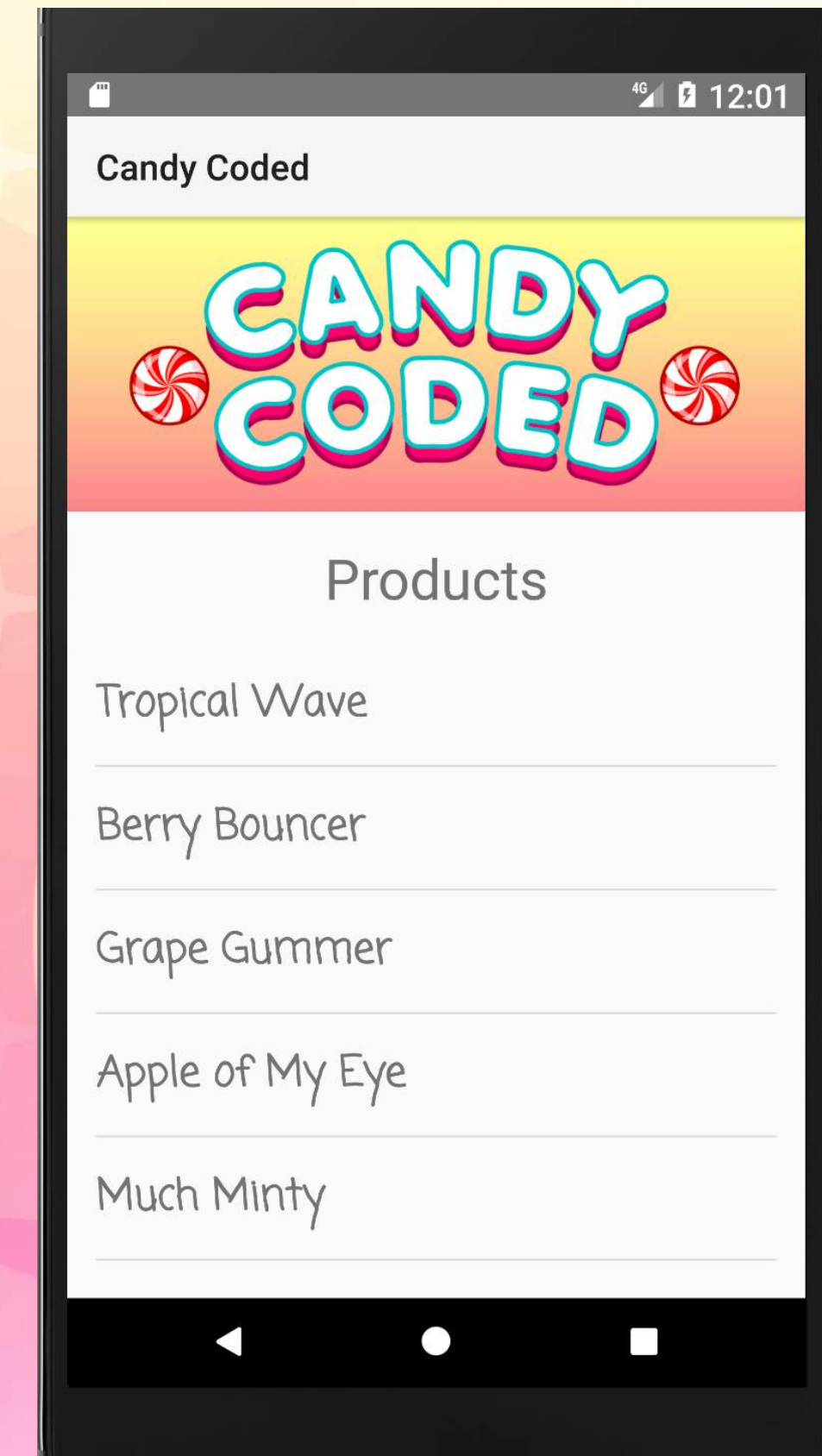


The MainActivity

+

The Layout the  
MainActivity  
displays

=



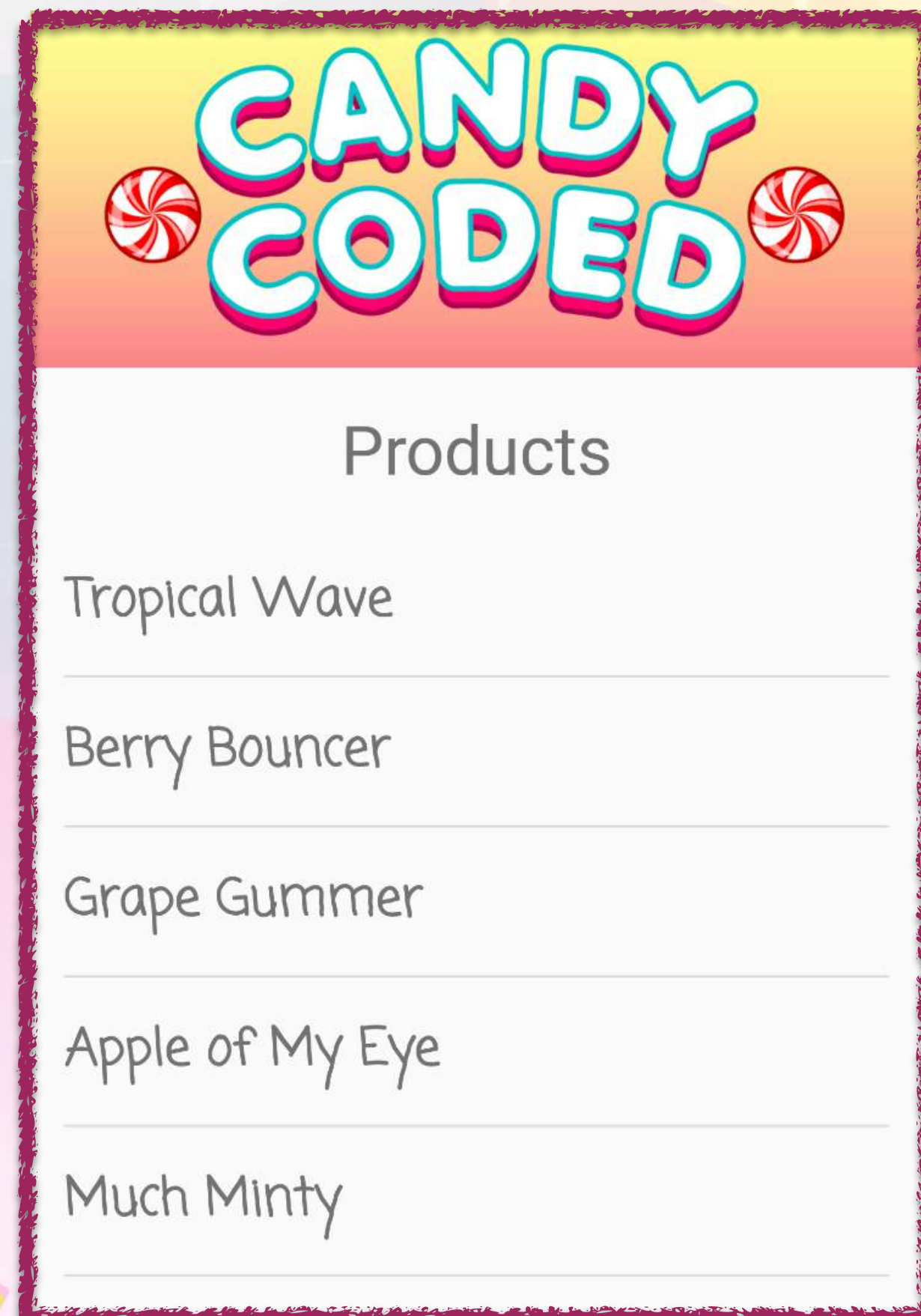
*Naming conventions: Java file names use PascalCase, and Layout file names use snake\_case where the first word is the type of layout*



# Layouts Define the User Interface

A layout describes the visual structure of everything the user will see on that screen.

Layout



*A layout can contain elements like text, images, buttons, etc. to display*

*But how do we get them to do something?*

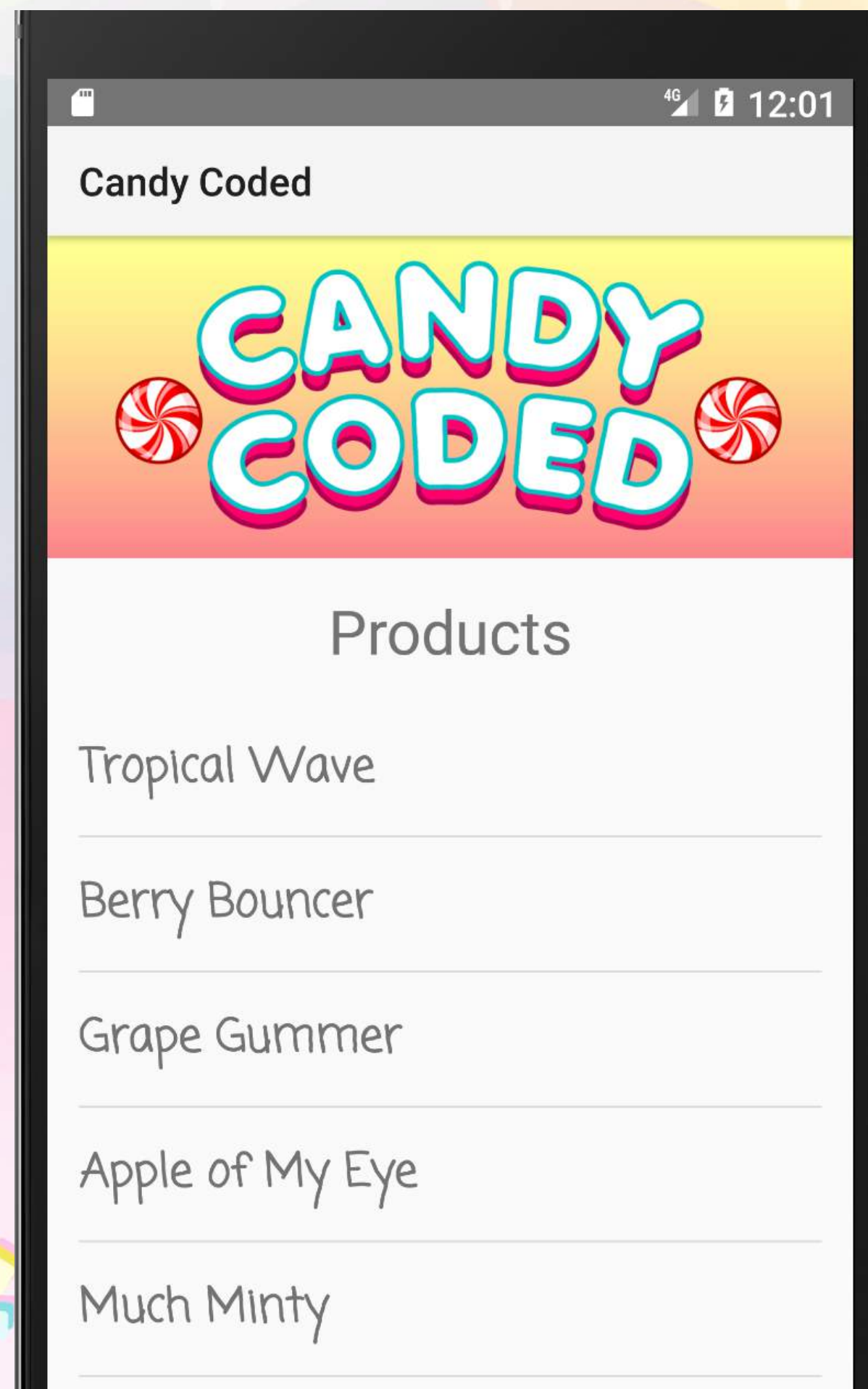
TRY  
ANDROID



# An Activity Manages a Layout

An Activity is a Java file that controls behaviors for our app. An Activity usually starts by creating a window to display the Layout.

## MainActivity



*The Activity is a Java file that holds behaviors like:*

- *Managing layouts*
- *Event handling*
- *Putting data in lists*
- *Opening other activities*

**TRY  
ANDROID**



# Screencast: Adjusting the Text in the TextView

By default, our Layout contains a TextView with the text “Hello World”. We’ll update it to say “Welcome to Candy Coded!”

TRY  
ANDROID



# What Dimensions Do We Use?

Android Studio supports the following dimensions: dp, sp, pt, px, mm, and in. We'll use only dp (for images) and sp (for text) in our app.

## dp - Density-independent Pixels

*For phones with different pixel densities and/or sizes, using dp units (instead of px units) makes the view dimensions in your layout resize properly*

## sp - Scale-independent Pixels

*sp is the same as dp but also scaled by the user's font size preferences*

```
android:textSize="24sp"
```



Moto X

300 pixels/in

Google Pixel

500 pixels/in

TRY  
ANDROID



# What Does wrap\_content Mean?

Looking at the TextView code in detail, we see its layout\_width and layout\_height are set to wrap\_content.

## activity\_main.xml

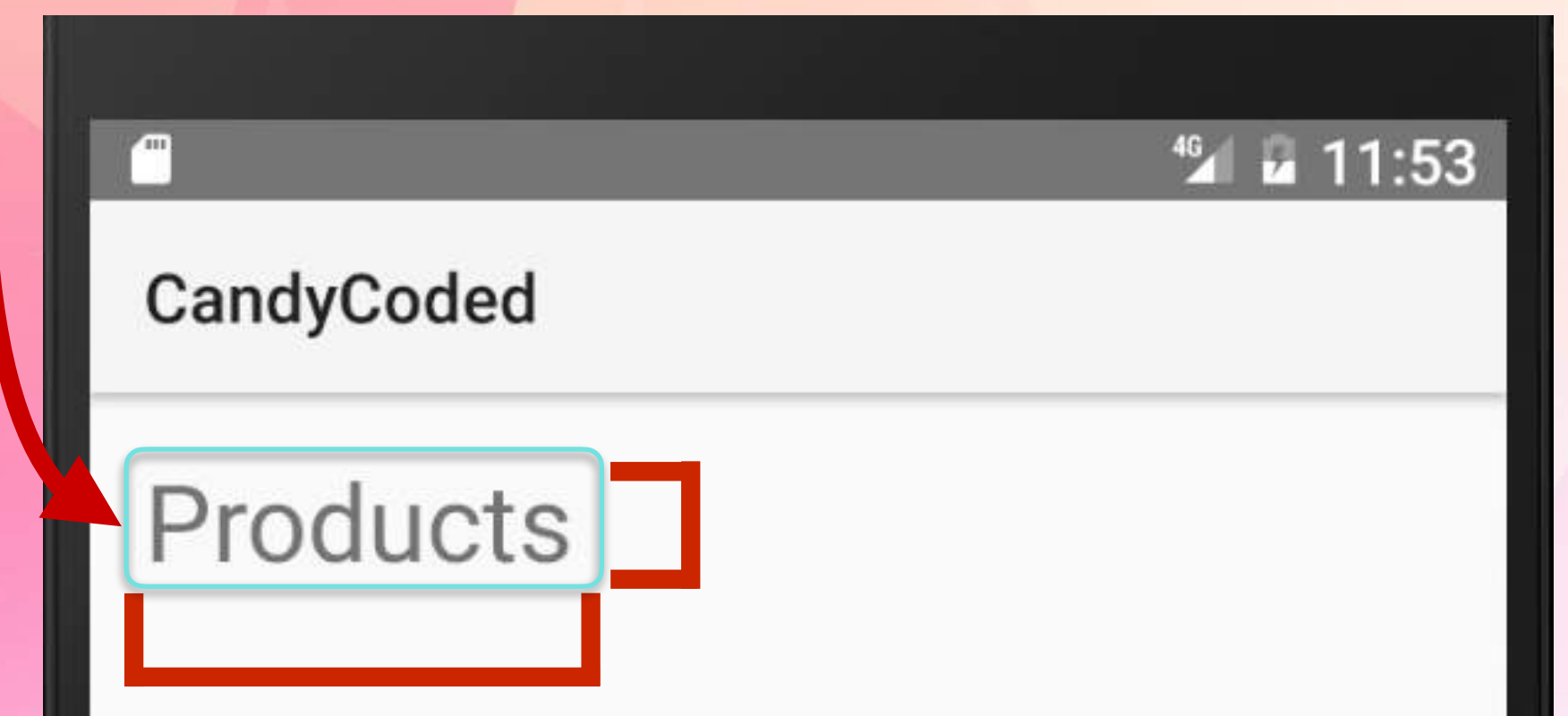
```
<?xml version="1.0" encoding="utf-8"?>
<... ConstraintLayout ...>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Products"
        android:textSize="24sp" />

</ConstraintLayout>
```

*wrap\_content means the view will only be large enough to enclose its content*

*So this TextView will only be wide enough and tall enough to hold the string "Products"*





# Best Practice: Using the String Resource File

Android Studio warns you if you use hardcoded values, you should use resources instead.

## activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<... ConstraintLayout ...>

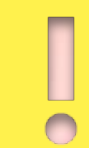
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Products"
        ... />
</ConstraintLayout>
```

*We want to replace hardcoded strings with a reference to the string resource file:*

*"@string/products\_title"*

*The variable products\_title will store the string value "Products"*

[I18N] Hardcoded string "Products", should use @string resource [more...](#)



TRY  
ANDROID



# Screencast: Using Resource Files

---

TRY  
**ANDROID**



