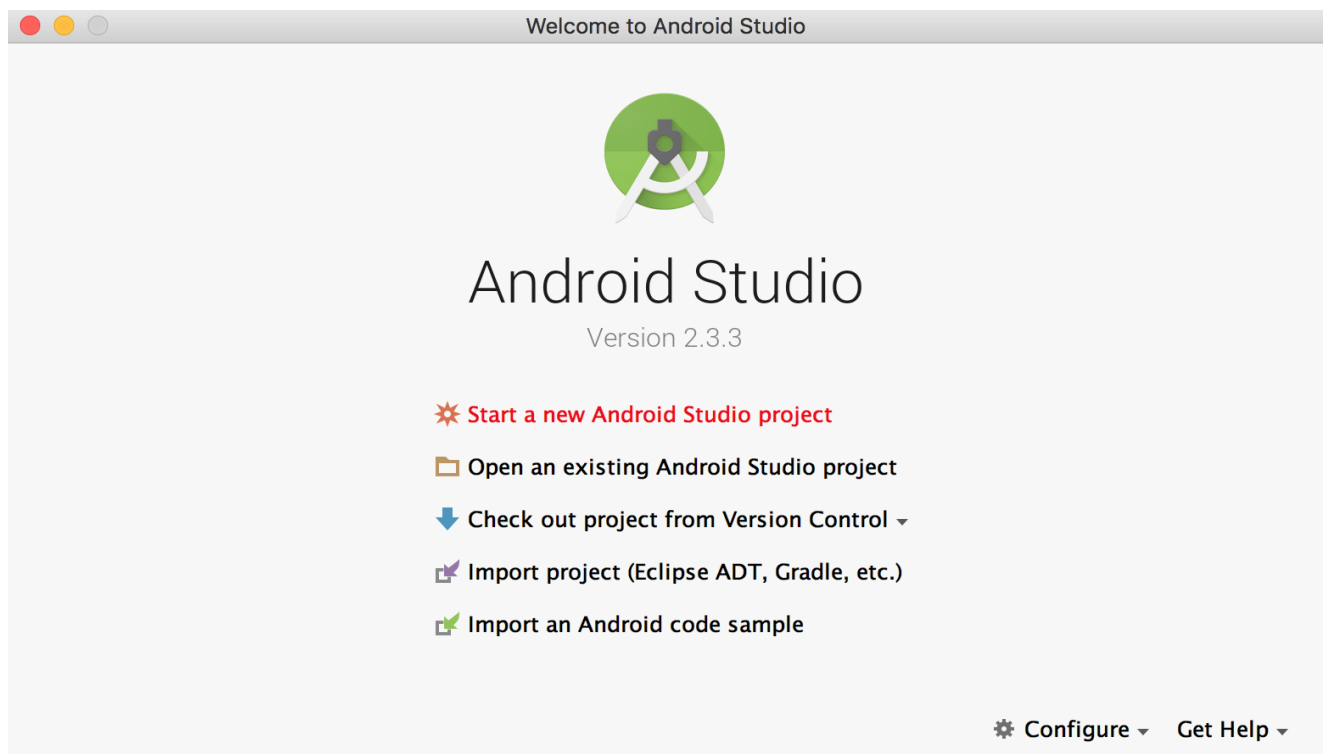


# Making Your First App : Hello World

## 1. Welcome to Android Studio Screen

First, open Android Studio. You should see this window. Go ahead and click Start a new Android Studio Project. An Android Studio Project typically means the code and files for one Android Application.

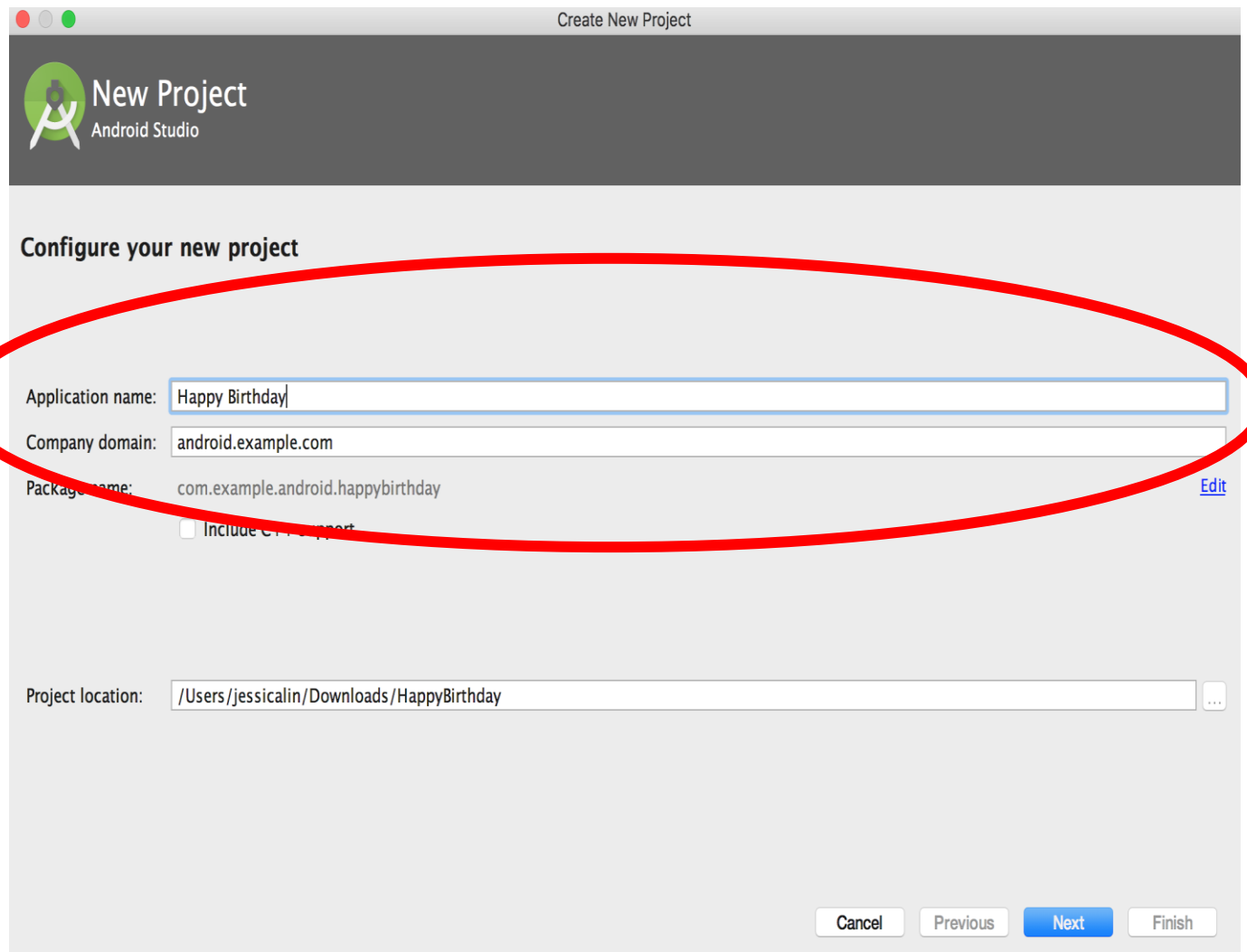


Select the "Start a new Android Studio project" option


## 2. Configure your new project

For this app, use this configurations:

- **Application Name:** Happy Birthday
- **Company domain:** android.example.com
- **Project location:** < Your choice of where to save this project on your computer >



Create New Project

 **New Project**  
Android Studio

**Configure your new project**

Application name:

Company domain:

Package name:  [Edit](#)

☐ Include Kotlin support

Project location:  ...

**These are the configurations we are using for this example project**

### **3. Select the form factors your app will run on**

We've decided to build this project for *Phone and Tablets* with Minimum SDK of API 15.

Create New Project

## Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs.

☒ Phone and Tablet

Minimum SDK

Lower API levels target more devices, but have fewer features available.  
By targeting API 15 and later, your app will run on approximately 100.0% of the devices that are active on the Google Play Store.  
[Help me choose](#)

☐ Wear

Minimum SDK

☐ TV

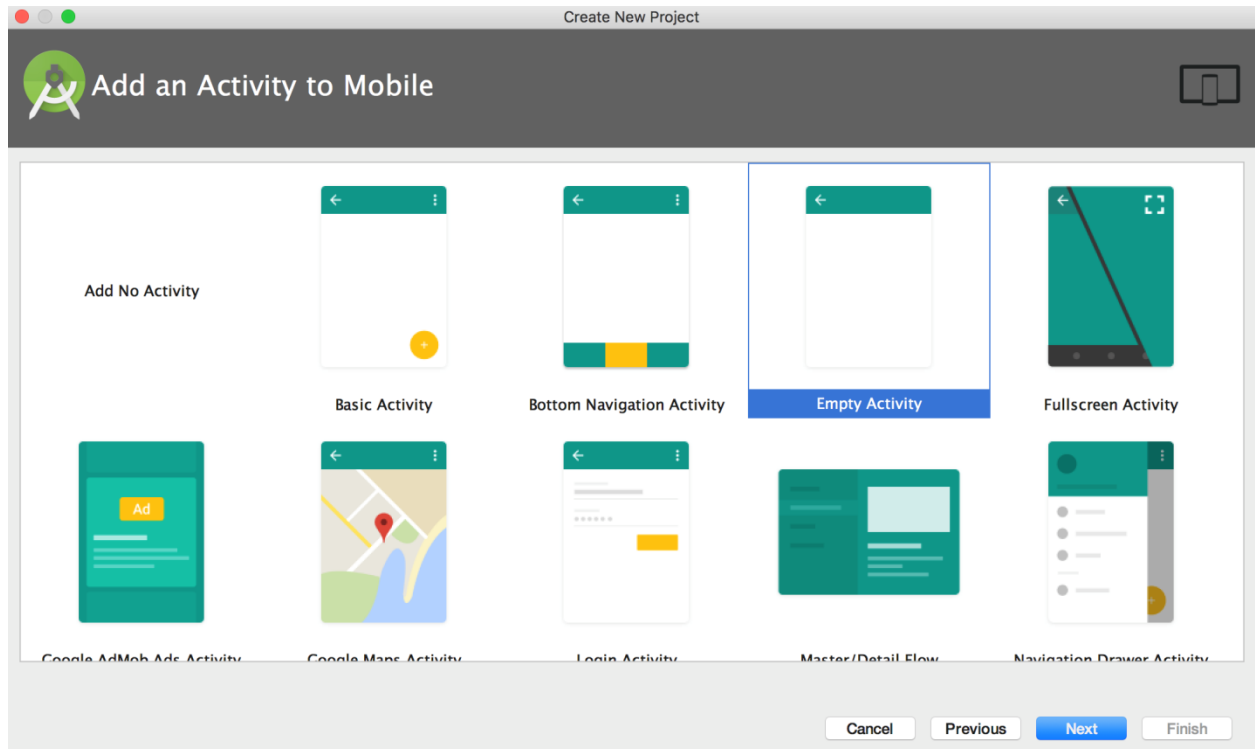
Minimum SDK

☐ Android Auto

We are building for Phone and Tablet and using API 15 as the Minimum SDK

#### 4. Add an Activity to Mobile

You will need to select the "Empty Activity" option to have starter code.

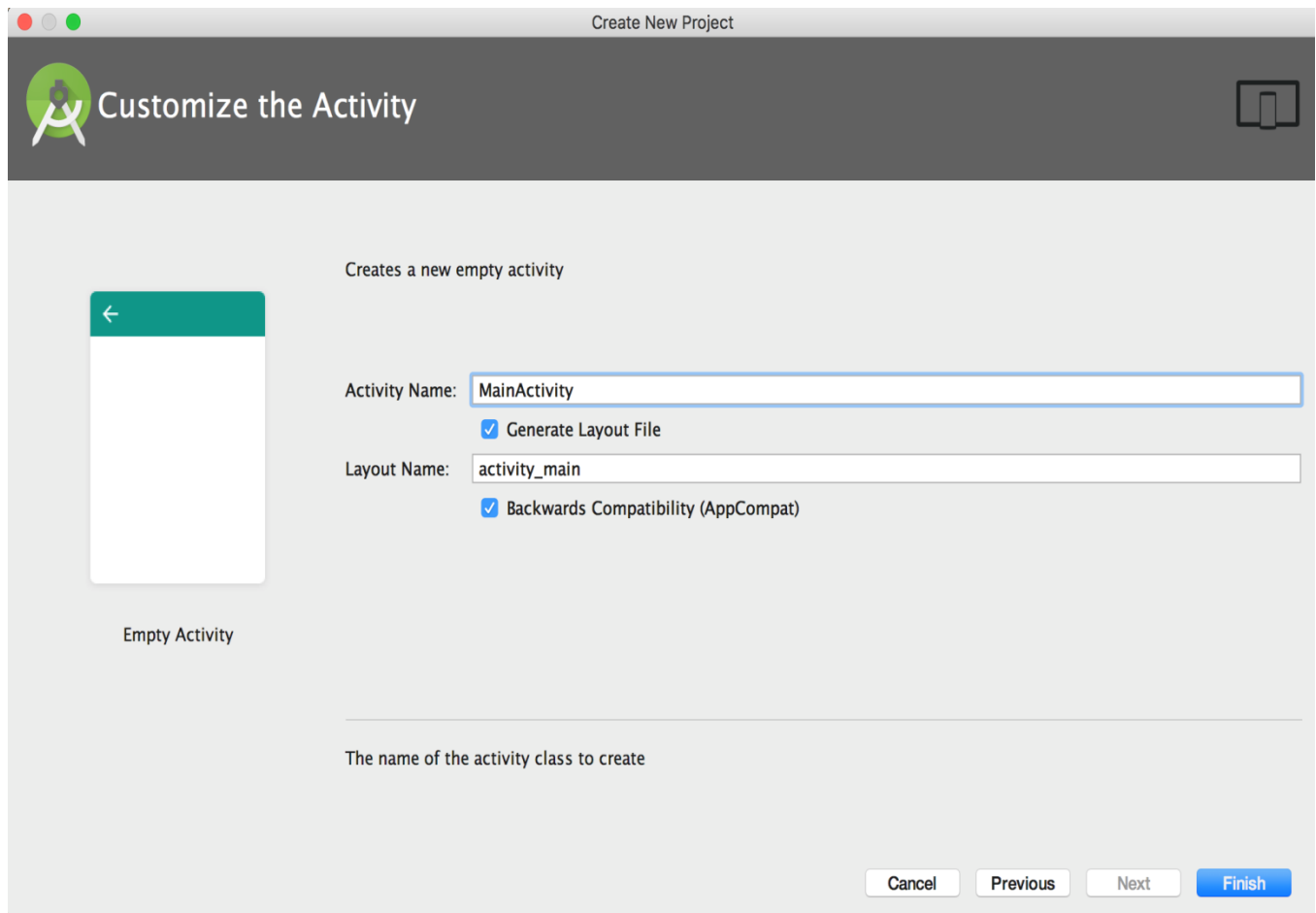


## 5. Customize the Activity and Finish

For the new activity, give it these names:

- **Activity Name:** MainActivity (Select Generate Layout File)
- **Layout Name:** activity\_main (Select Backwards Compatibility (AppCompat))

Once those options have been selected, click Finish to generate your first project!



**Name the activity *MainActivity* and the Layout Name *activity\_main***

Depending on your computer speed it might take a minute to set up your project. Go ahead and get some tea or do a stretch.

## **Keeping up with the changes**

Google is constantly improving the Android platform and adding new features. This is great for you as a developer, but it makes learning harder sometimes. Recently Google released **ConstraintLayout**; a tool that makes it super fast to create responsive UIs with many different types of components. **ConstraintLayout** is a great tool to have on the sleeve, but we will use **RelativeLayout**, **LinearLayout** .

All of this matters to you because the new project templates in Android Studio now use **ConstraintLayout** as default, which makes the code you see on your computer a bit different from what is on the screen.

## Current Layout File

In the new versions of Android Studio, after choosing the **Empty Activity** template, the layout file `app/src/main/res/layout/activity_main.xml` will look like this:

```
<?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:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.udacity.myapplication.MainActivity">

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

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

Note the use of **ConstraintLayout**, and that **TextView** has a list of limiters that position it within **ConstraintLayout**.

## Modify the Layout File

Unlike the above code, our videos and start code assume that the template looks more like the following, using as the root of the view a **RelativeLayout**:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
```

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context="com.udacity.myapplication.MainActivity">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!" />
</RelativeLayout>
```

When you create your new project, go to `app/src/main/res/layout/activity_main.xml` and copy and paste the above code. Then you're ready to go!