# 1. Getting Started

-Welcome to Androidville

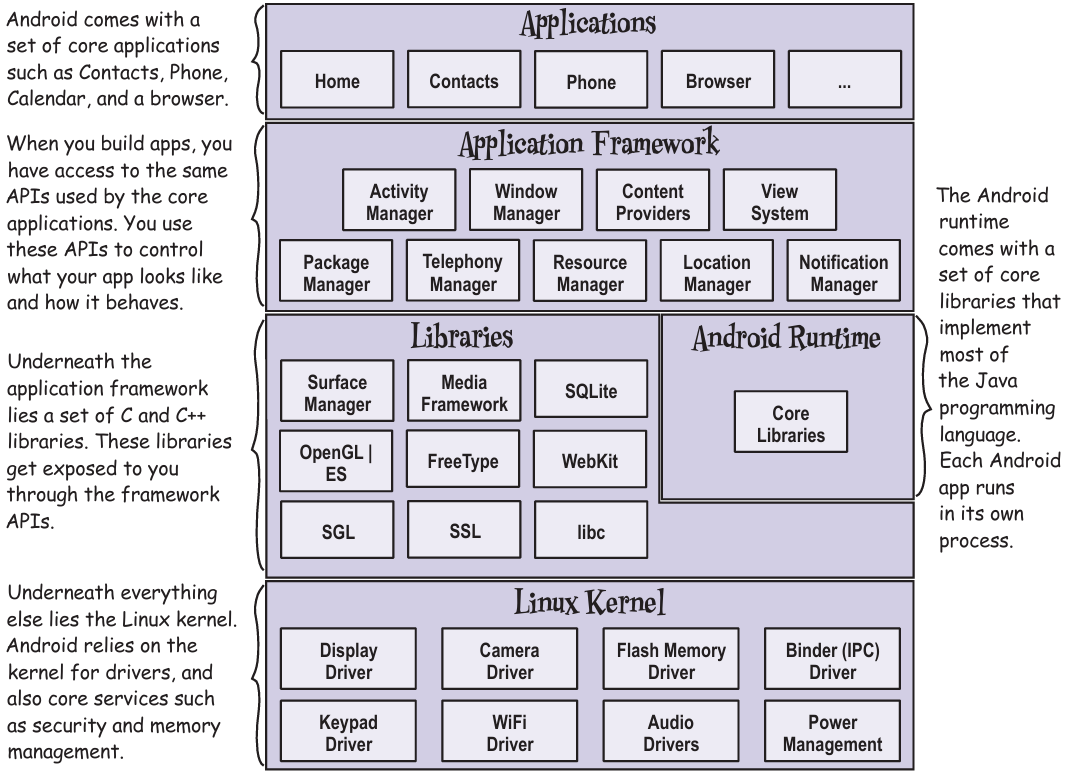
+**Android** is an **open source platform** based on **Linux** and **championed** by **Google**. It’s development **framework** using a mix of **Java** and **XML**. It enables to **deploy apps** to devices: phone, tablet…

+Android app is composed of **screens**. You **define** what each **screen looks like** by a **layout**. It’s defined in **XML** and **include GUI**: buttons, text fields, labels.

+**Activities**: **define** **what** **app does**. It’s special **Java class** that decides which layout to use and tell app how to respond to user.

+**Extra resources**: such as image files, app data

-The Android platform dissected



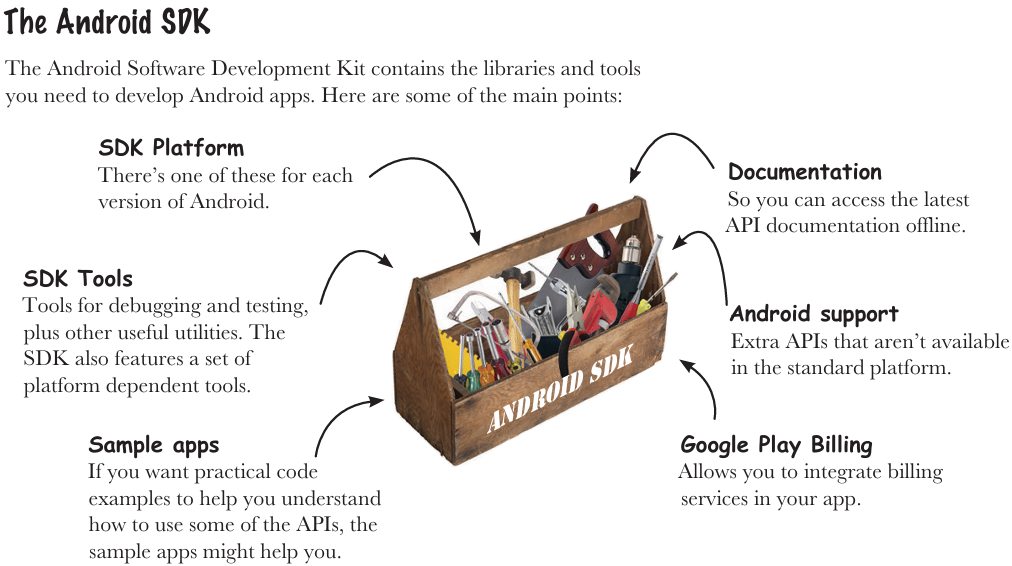
+All of **Android libraries** are **exposed** through **APIs** in **app framework**, these APIs used to **create apps**.

-Here’s what we’re going to do

+Set up a development environment->Build a basic app->Run the app in Android emulator->Change the app

-Your development environment

+Android devices don’t run .class +.jar ->Need a tool to write and compile Java code + other tools to convert compiled code into form that Android devices can run **Android SDK**



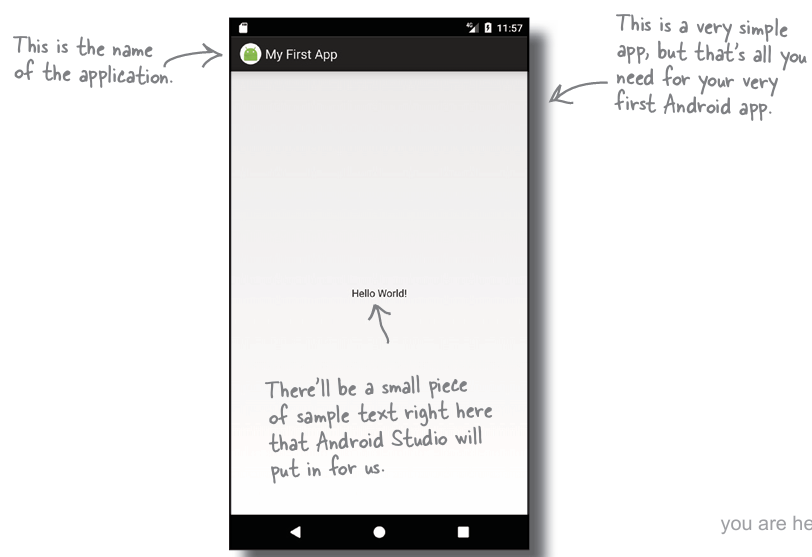
+Android Studio is a special version of IntelliJ IDEA that includes a version of Android SDK and extra GUI tools.

-Install Android Studio

<http://developer.android.com/sdk/index.html#Requirements>

https://developer.android.com/sdk/installing/index.html?pkg=studio

-Build a basic app

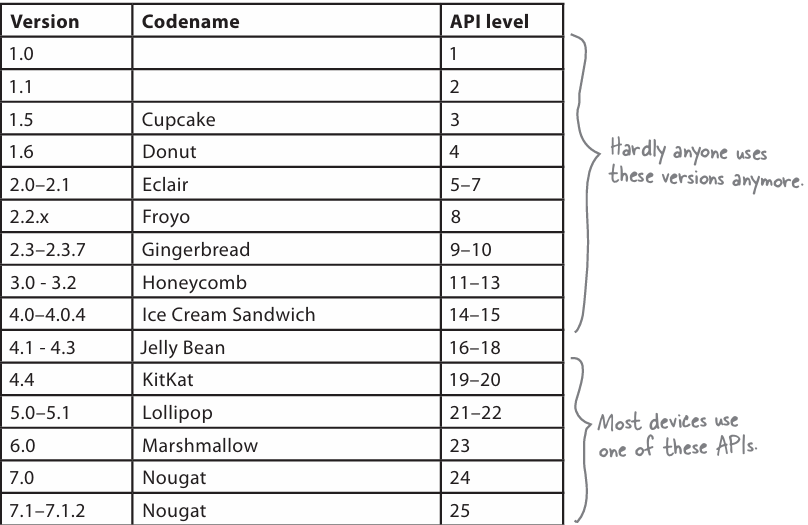


-How to build the app

+Create a new project: Start a new Android Studio project

+Configure the project: AS uses company domain + app name to form the name of package used for app ->uniquely identify the app.

+Specify the minimum SDK

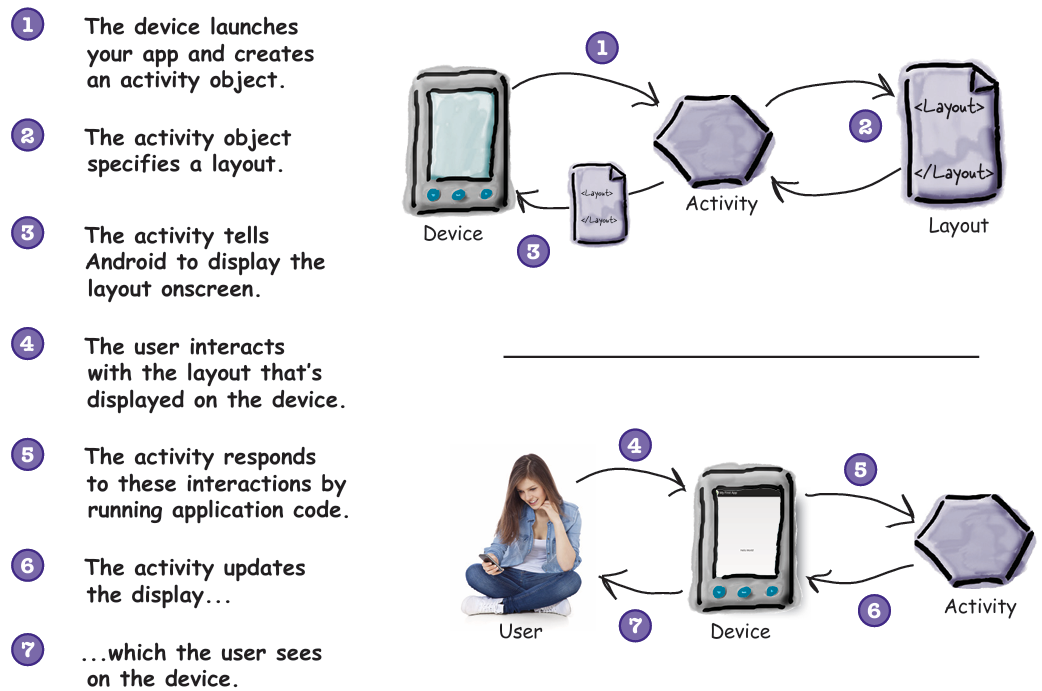


-Activities and layout from 50000 feet

+**Add** an **activity** to project. Android app is a collection of screens, each screen is composed of an activity and a layout.

+**Activity**: a single, defined thing that user can do. Activities are associated with one screen, written in Java

+**Layout**: the appearance of screen, written as XML files, tell Android how different screen elements are arranged

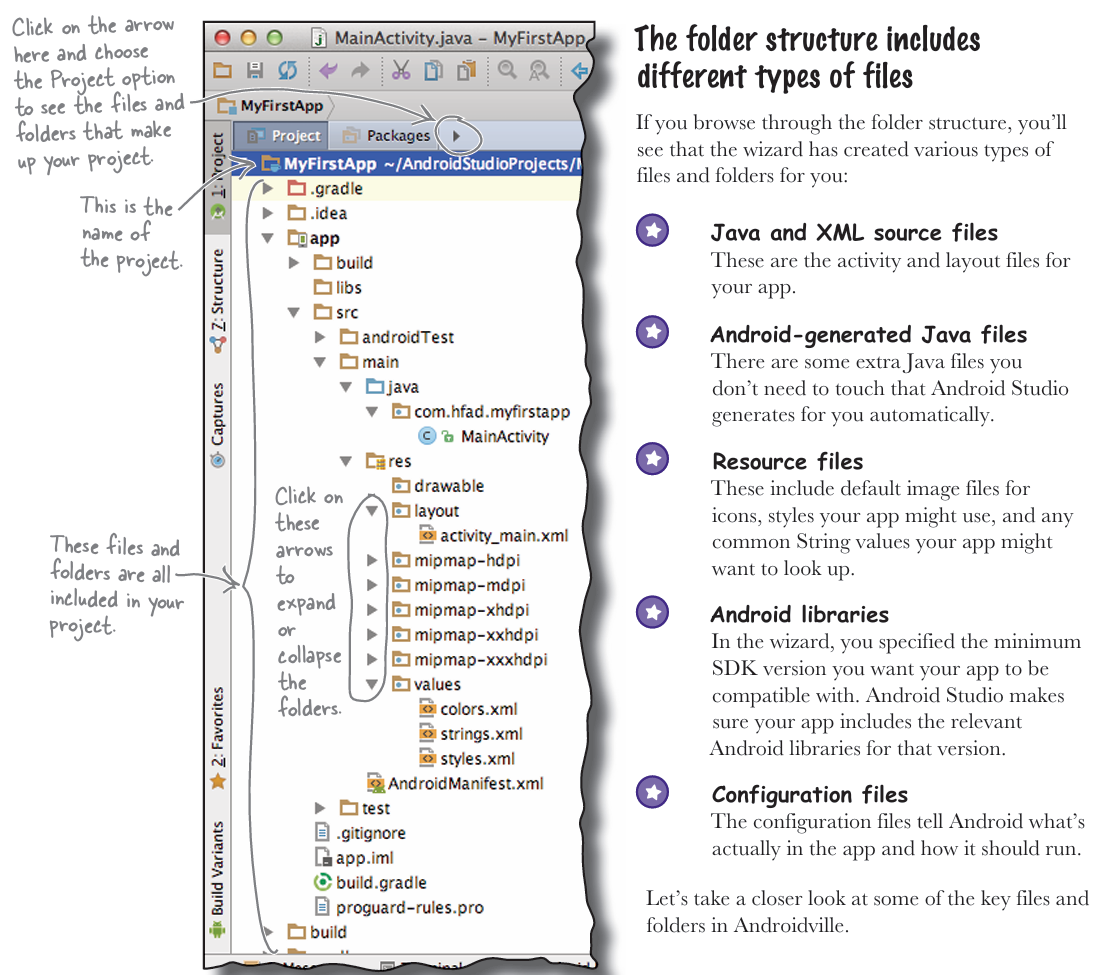


-How to build the app (continued)

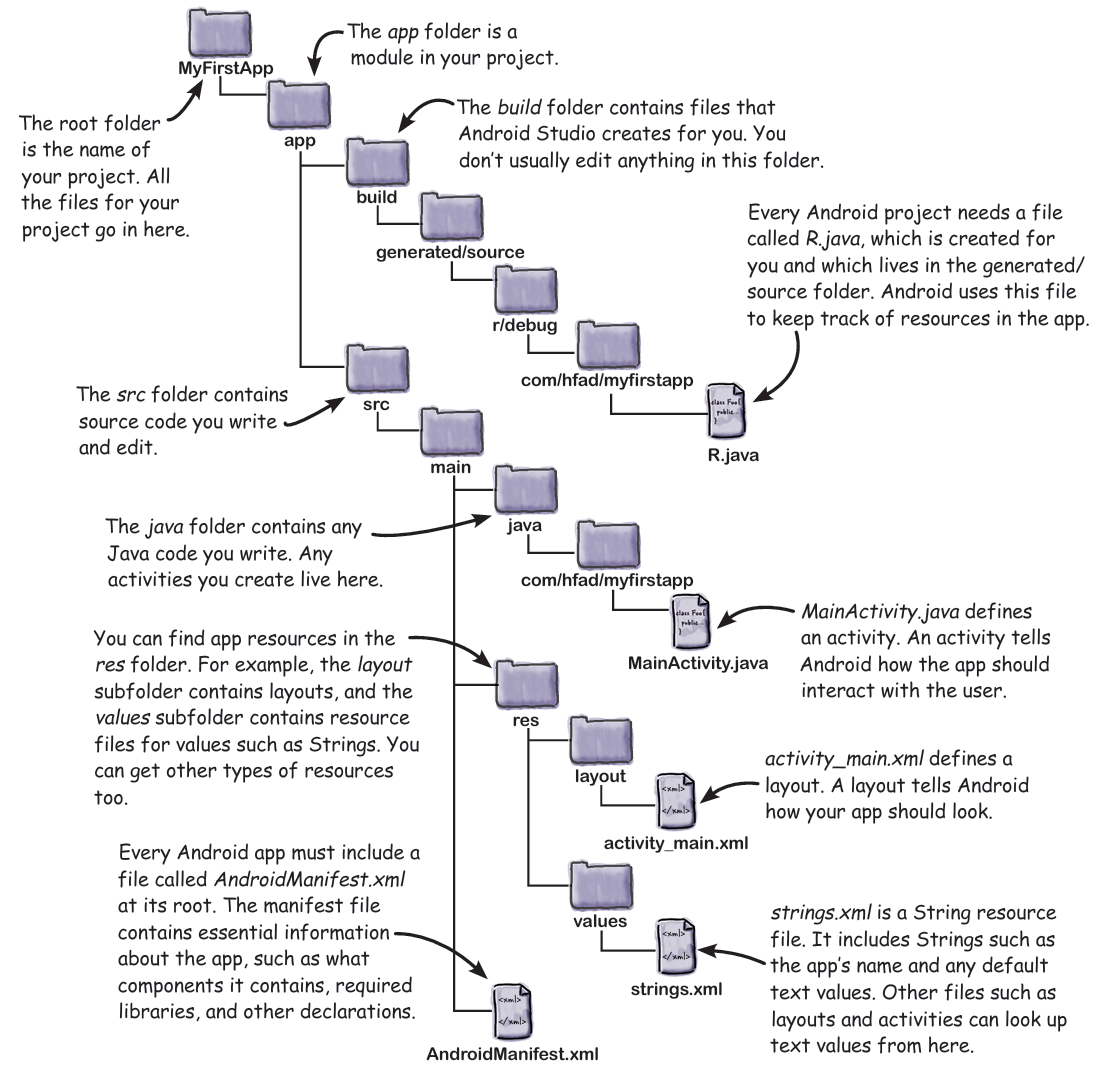
+Add an activity: choose Empty Activity, uncheck Backwards Compatibility

-You’ve just created your first Android app

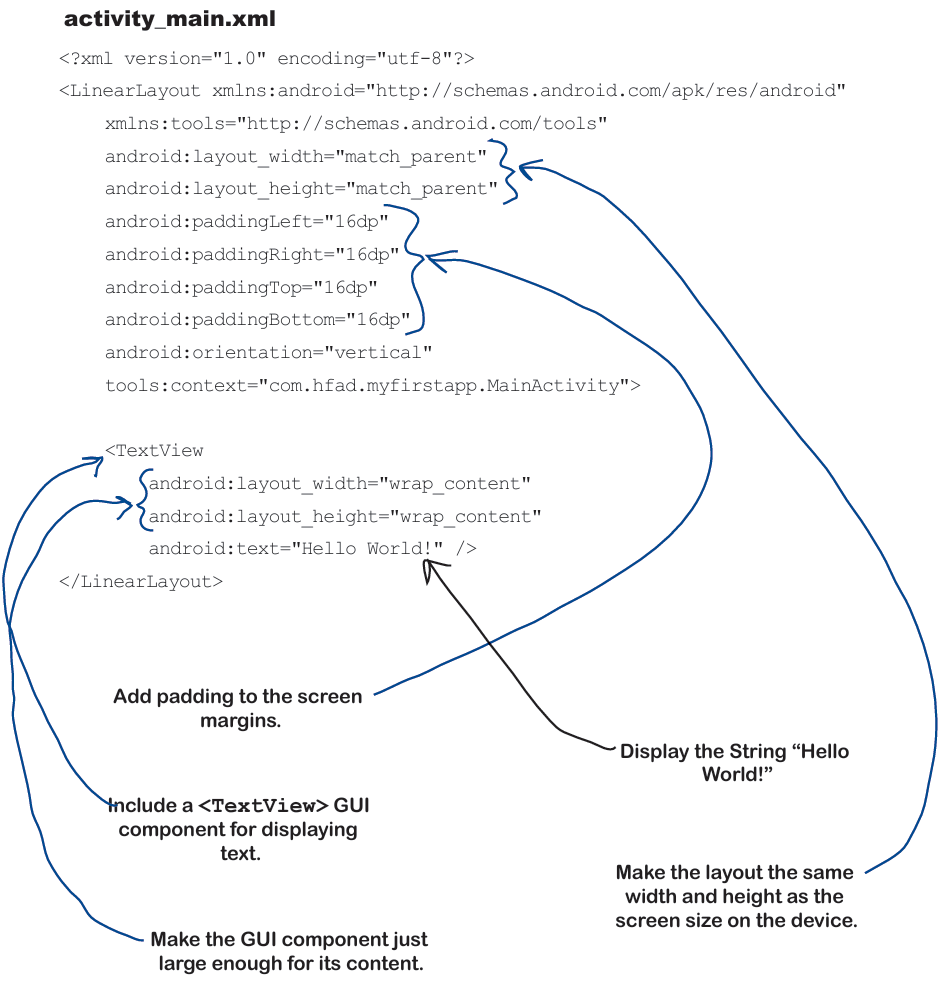
-Android Studio creates a complete folder structure for you

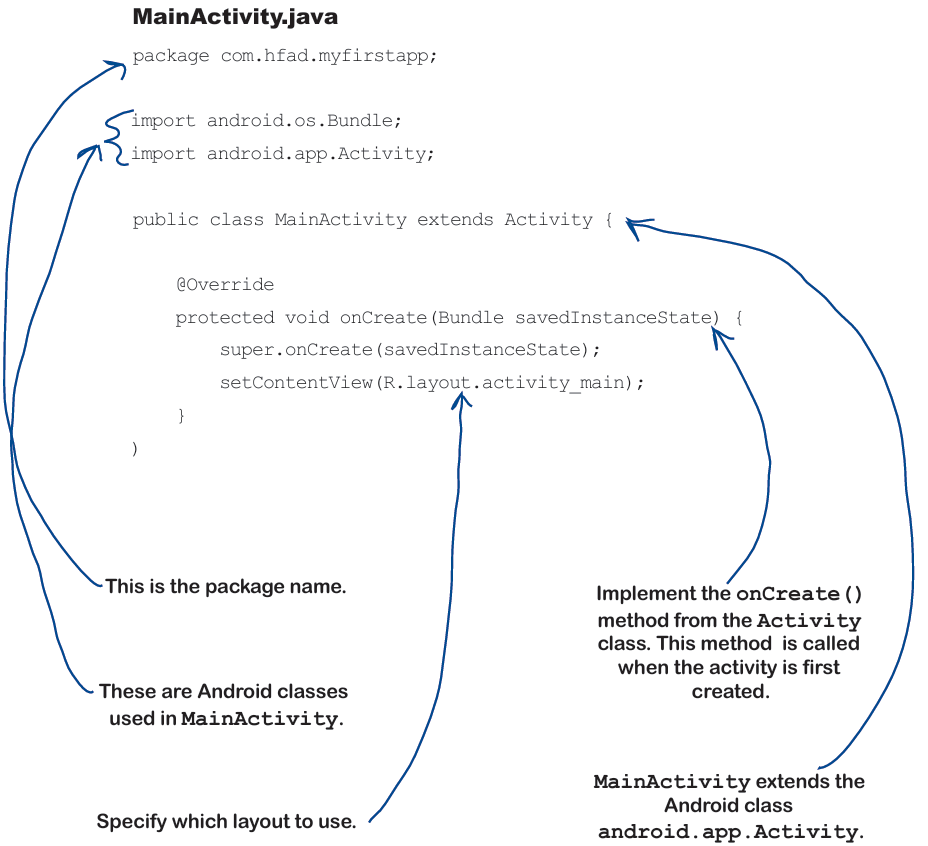


-Useful files in your project: Android Studio use Gradle build system to compile and deploy apps.



-Edit code with the Android Studio editors





-Run the app in the Android emulator

+**Android emulator** in Android SDK enables you to **set up** more **Android virtual devices**.

-Creating an Android Virtual Device

+Open Android Virtual Device Manager:Tool->AVD Manager->Create…

+Select the hardware

+Select a system image: choose API level that’s compatible with app.

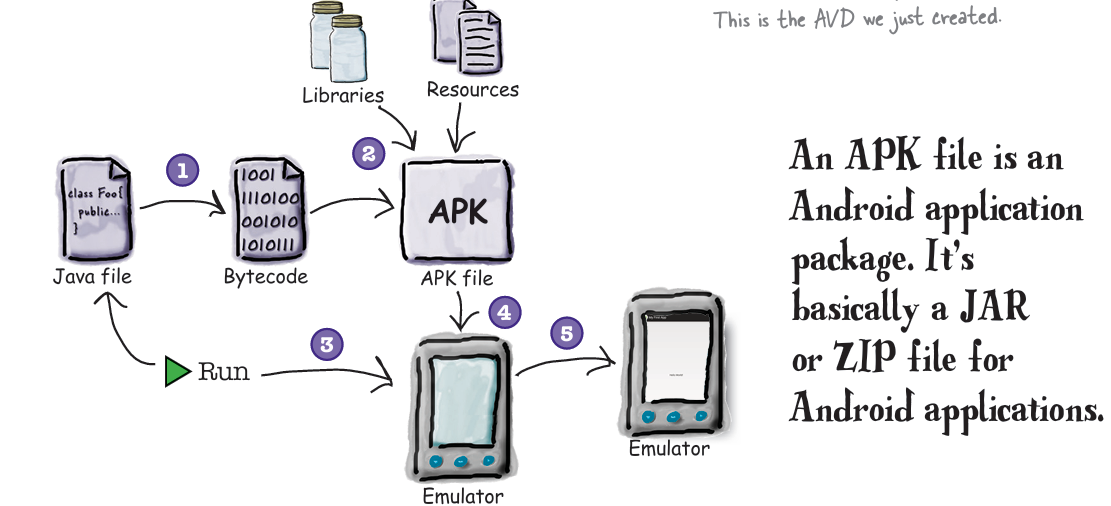
+Verify AVD configuration

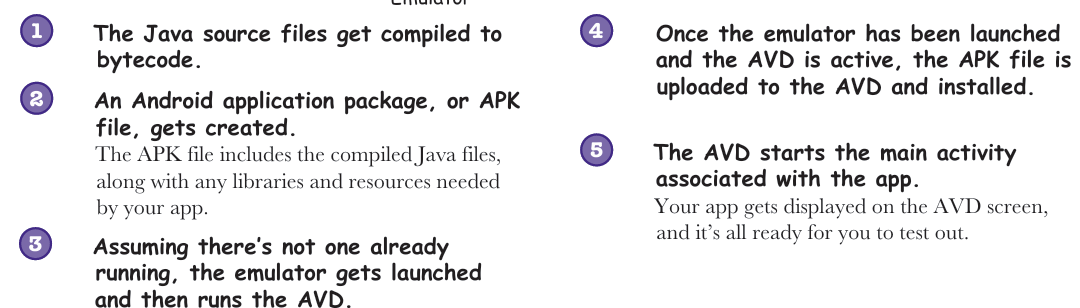
-Run the app in the emulator

+Run app from Run menu ->Choose device

If have error: <https://stackoverflow.com/questions/68387270/android-studio-error-installed-build-tools-revision-31-0-0-is-corrupted>

+Compile, package, deploy, and run:

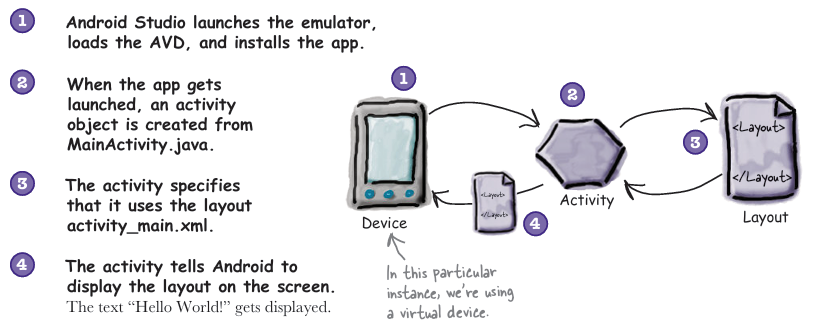




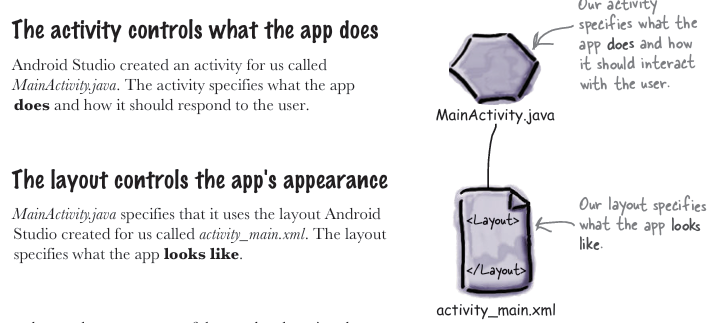
-You can watch progress in the console

+Test drive:

-What just happened?

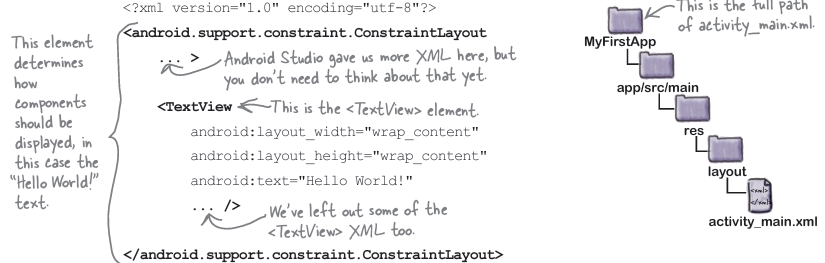


-Refining the app



-What’s in the layout: 2 ways of viewing and editing layout files: design editor + code editor

-activity\_main.xml has two elements



+…**ConstraintLayout**: type of layout element telling Android how to display components on device screen

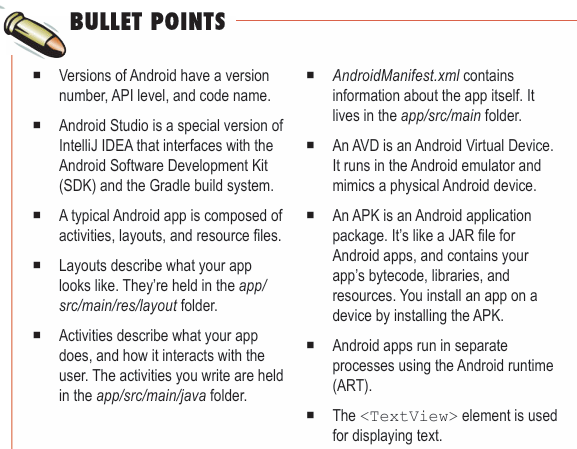
+**TextView**: display text to user



-Update the text displayed in the layout

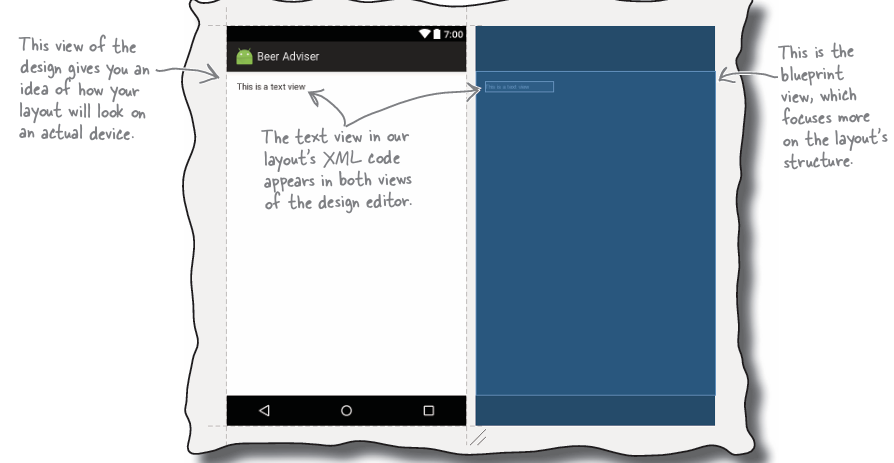
-Take the app for a test drive

-Your Android Toolbox

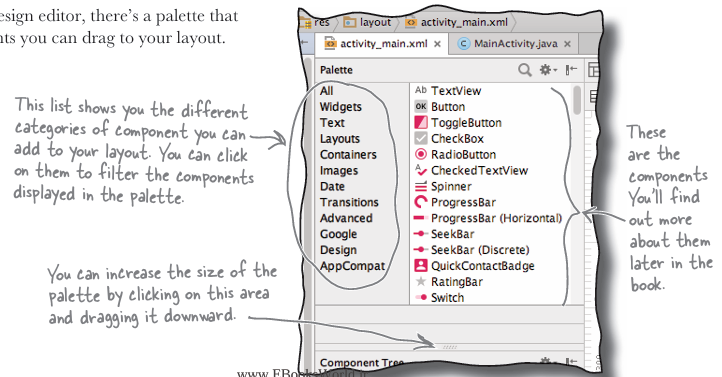


# 2. Building Interactive Apps

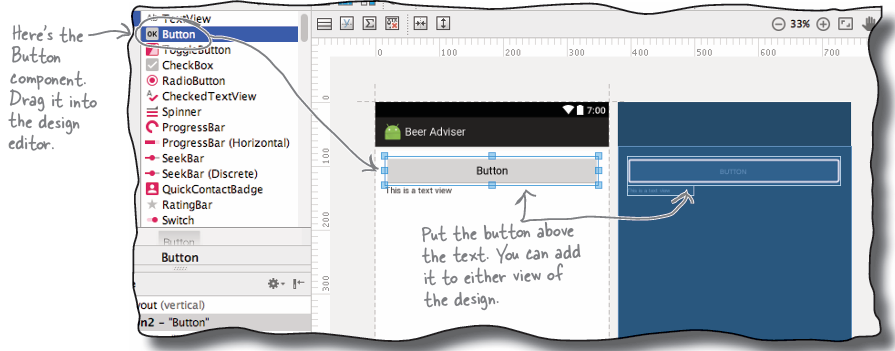
-In design editor, it features 2 different views of layout:



+To the left, there’s a **palette**:



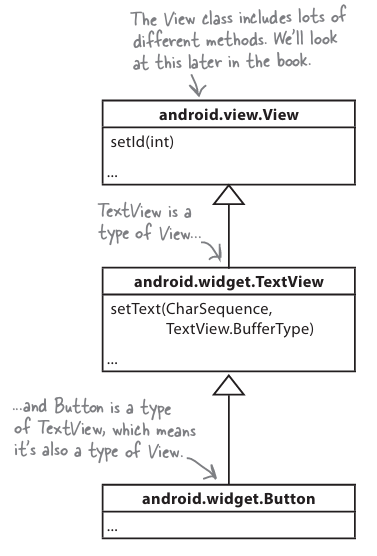
-Add a button:



+Changes in design editor are reflected in XML:



-Buttons and text views are subclasses of the same Android View class



+component an identifying name

+: wrap content means it should be just big enough for content, match\_parent means it should be as wide as layout containing it.

+: text display

-Look at layout code:

