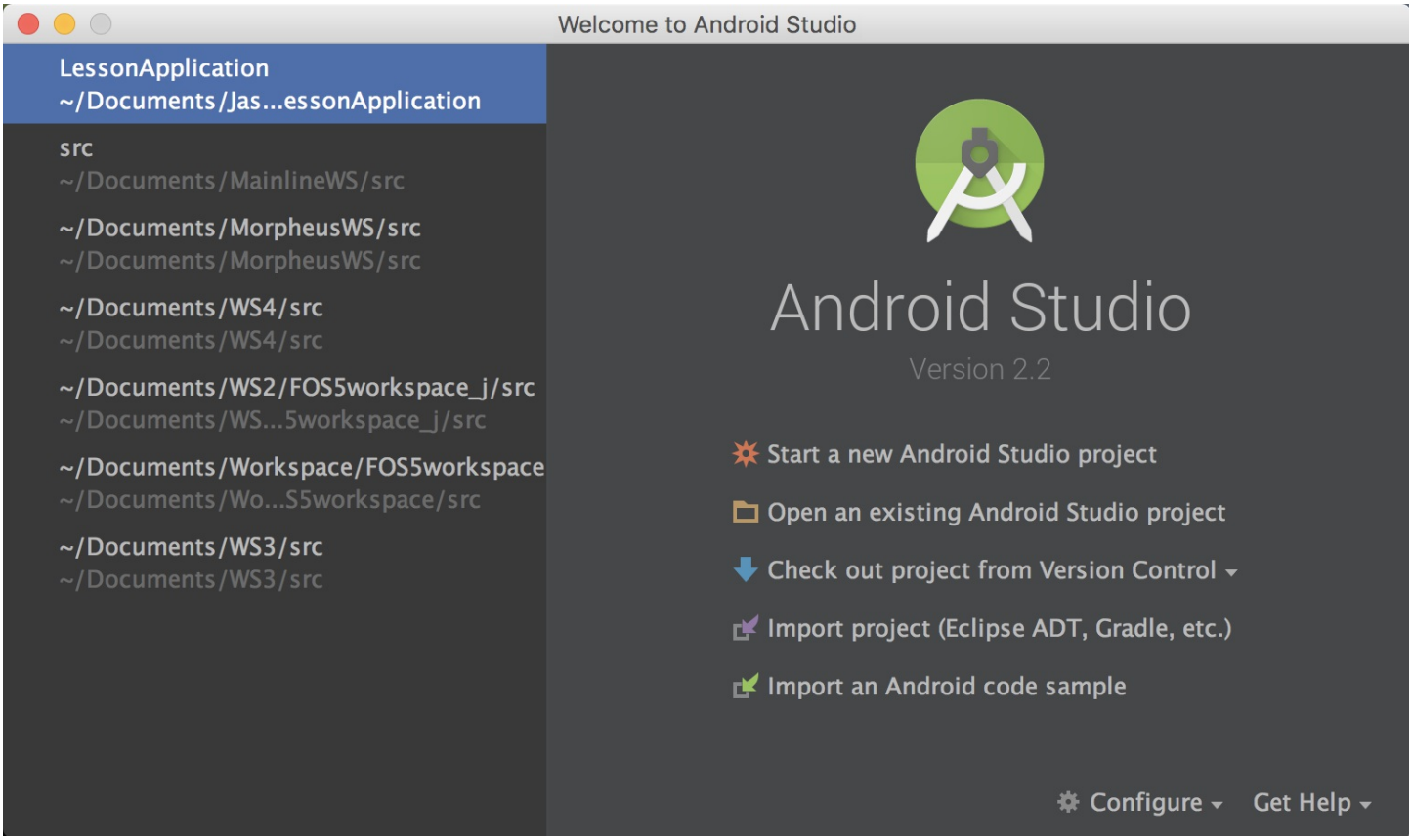
Welcome to Android!

Android is fun and it's actually quite easy to wrap your head around. In these lessons I hope to teach you the basics of Android development, such as basic core components such as the Activity life cycle, all the way to the simple things such as formatting your layouts to get the best UI possible! Let's start with setting up your workspace to start developing Android apps.

Setting Up

The first thing you're going to want to do is download Android Studio! Whatever version you have currently should work fine, but if you don't have one, just go ahead and download the latest one. I'll be using Android Studio v2.2



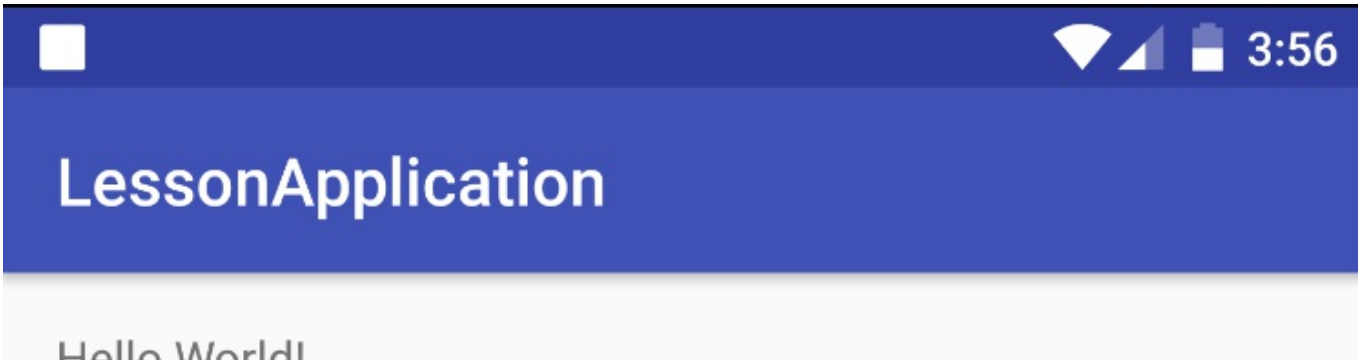
Let's go ahead and `Start a new Android Studio project`. Give your project a name and a path

> *The package name is a unique ('String') identifier to your project. It lets the Android OS know what stuff belongs to a certain project. So for example, my `Company Domain` is com.jtj.lessons (I just made it up). My `Application Name` is `LessonApplication` so my package name will be `com.jtj.lessons.LessonApplication`. It doesn't really matter what this ends up being, but once you create your app, you are stuck with this package name (identifier) forever!*

Keep following the instructions, set min SDK to 19 for now, and select Empty Activity when prompted. This is all for setup purposes and can be ignored for now (ignored as in, I won't explain much now).

> *Fine, if you care so much about it, Google's team is kind enough to give us boilerplate code for a lot of different scenarios. For example, if we want a Map UI app, we can select `Map Activity` and much of the code will be generated for us to create an Map application without much hassle.*

> *By the way, you can run your app! Click the green arrow. You should be greeted with this UI.*



Hello World!

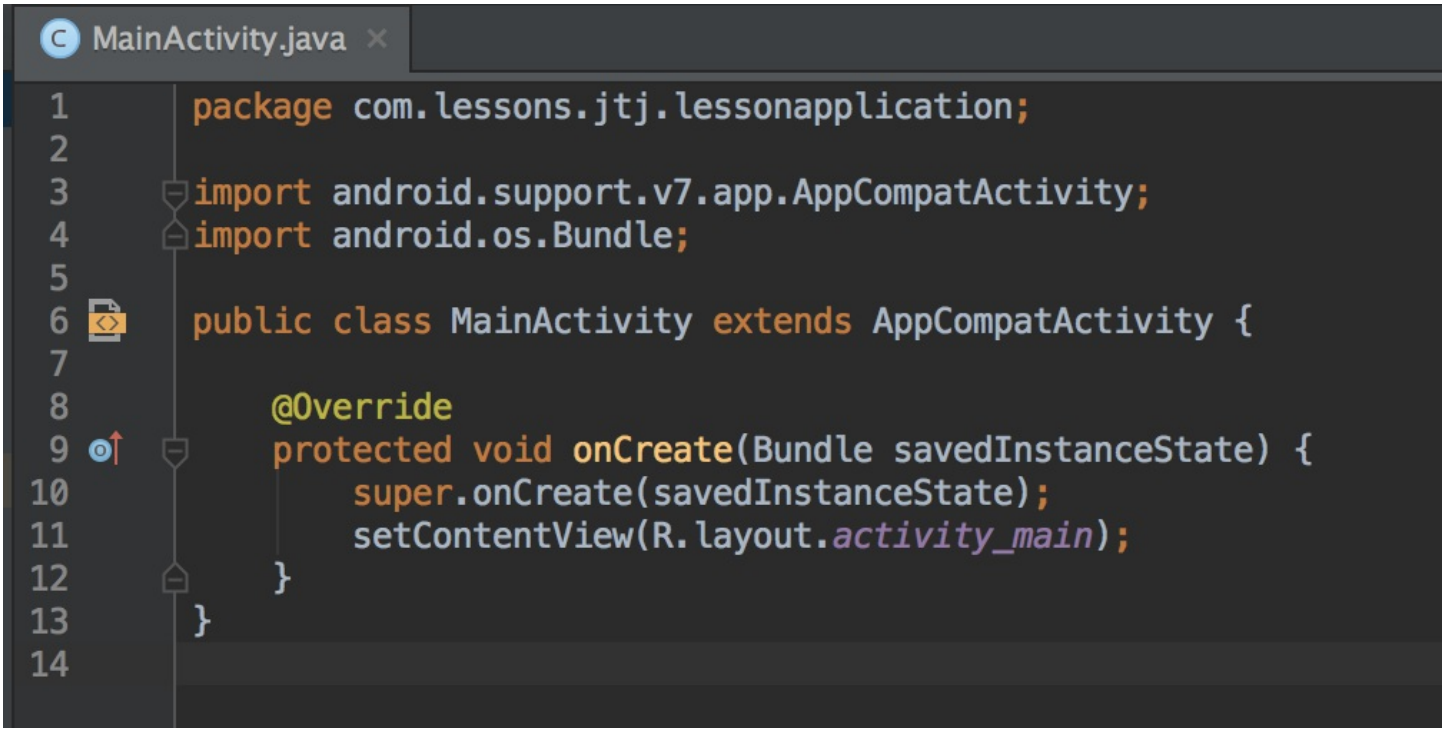
The Activity Class

We'll start with the `Activity` class.

You can think of an activity as a single UI 'page' in your application. Relating this to a website, you may have your home page (ie www.facebook.com/). When you click on your friends' profile, it will open a new URL to show your friends' profile information. Each page can be considered an `Activity.`

> *Note: this is not to be confused with the App itself. Think of an App to be like Google Chrome, and each page you view is an activity. An Application is composed of many activities that are linked together, kind of like how a web browser is composed to webpages that are linked.*

One activity can launch/display another activity (as with web hyperlinks). Each activity is also paired with a `layout` (located in <projectpath>/app/res/layout/<layout name>.xml)



In this case, our `Acitivity` is titled MainActivity, and extends an Android base class `AppCompatActivity`. Our layout is seen inside of the onCreate function, which is auto generated code. The function setContentView is called here to set our view. We'll dive into this later.

> *Do you know inheritance well? If not, reach out to me and I'll make a lesson on it. Our MainActivity class is inheriting functions and functionality from `AppCompatActivity`, which is an Android base class which handles the OS level displaying and running of Activities.*

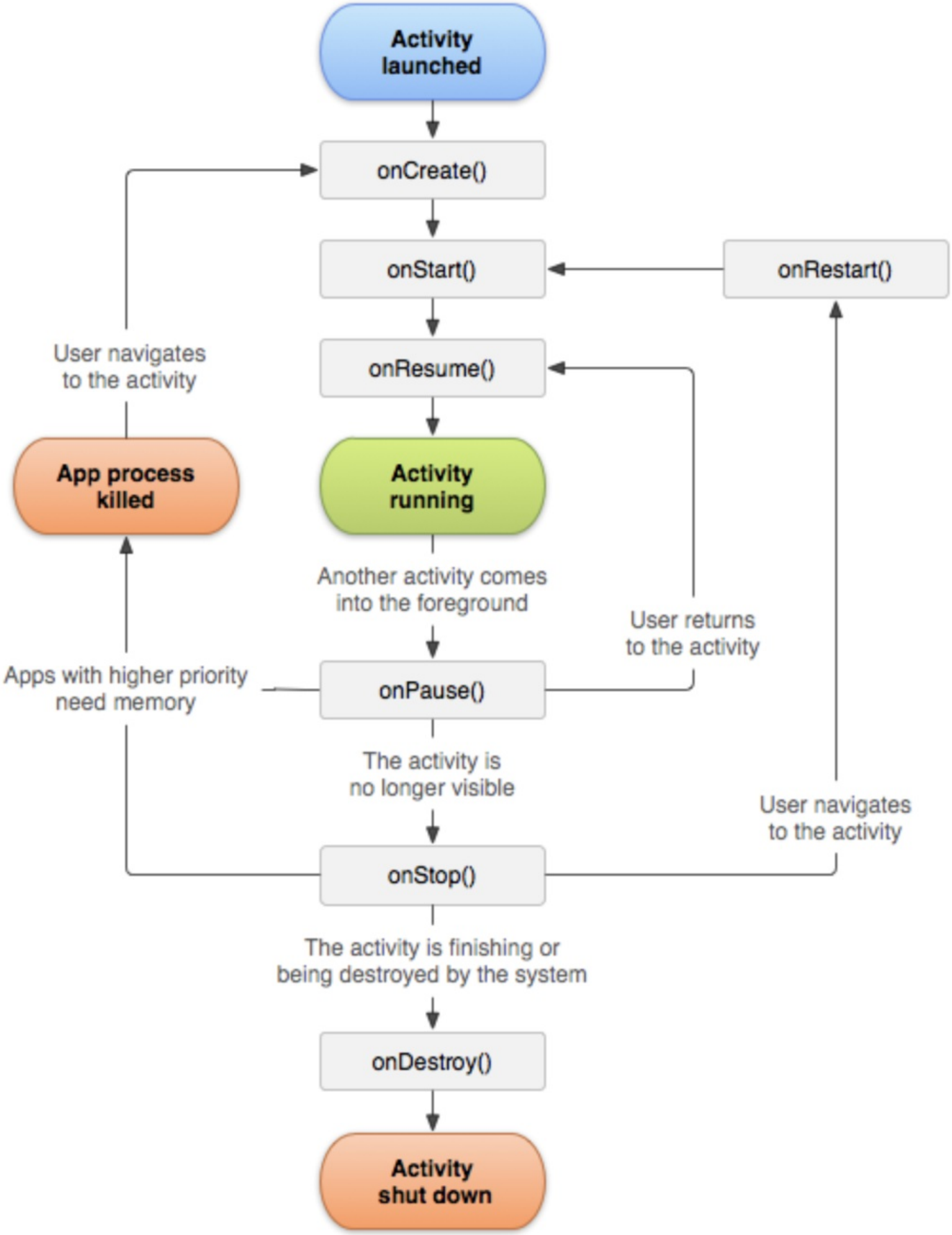
So, `onCreate` (of Activity), we are calling our superclass' onCreate function (ignore for now) and then setting the content view to that of our layout XML. That's it. Our activity has nothing more. Make sense so far? Pretty simple stuff.

In the next lesson, we'll be more hands on in actual coding. We'll be

tinkering with the layout, creating a button to click on, and opening another activity to show our users

> *Bonus info that you might want to know to be successful:*

The Activity Lifecycle



I feel this image is somewhat self explanatory. Each function here (denoted by `()`) is a function that is called automatically when different statuses are

present (ie when the app is created vs when you switch to another app, etc.). In our case, we're already familiar with `onCreate()`. `onCreate()` is the entry point into our application. Once the activity is launched and 'created', the app can start running. When we leave the application, such as when we open another activity or leave the app entirely. When the OS runs out of memory, or we kill the app ourselves, the resources need to be allocated elsewhere, so the system will begin to `destroy` the activity, losing any progress./information we haven't stored somewhere. If we go back to the activity before it is destoryed by the system, it can be `resumed`, or can be recreated if it was killed. It's a bit of a crap shoot whether or not our app/activity gets killed, it all depends on the system!