

杨兵兵

2017年8月15日

AndroidUI初步

Android包含四大组件，这四大组件有Activity、Service、Broadcast Receiver、Content Provider。其中：

Activity形象的说就是一个容器，在里面放置各种控件(按钮，文本，复选框等)，就形成了软件的界面，它是App和用户交互的窗口；

Service顾名思义就是服务，没有用户界面，但可以为你的app提供一些服务，就像你使用的微信为什么退出了还能收到消息，播放器为什么关屏了还可以播放音乐，就是因为它的作用，可以在后台（这里的后台可不是指服务端后台）执行一些指令；

Broadcast Receiver广播接收器，其实在手机里预置了好多的广播，并且允许开发者自己定义一些广播并在系统内发布以便传输一些数据。其实广播大家都不陌生可以想象成你的收音机，每一个广播电台都在实时的发布着广播，当你调频调到相同的频率你就能接收到，这里的广播也是这样的，只不过是你要先把频率调好，当发布者发布广播的时候你就可以接收到这些数据：

Content Provider内容提供者，它最主要的作用是共享数据，如果你有两个app安装在同一个手机上，你的一个app想访问你另外一个app中的数据就需要通过Content Provider来实现了，android也通过Content Provider暴露了好多的数据供终端上的app读取使用。以上简单介绍了一下activity的四大组件，我们这篇文档主要是讲一下Activity以及他所呈现的控件。

一、Activity

1、什么是Activity上面已经说了，这里先从activitys的生命周期开始。其生命周期主要包括：

onCreate()：这个方法是在activity启动时第一个调用的方法（或者activity被系统回收后重启时候也会被调用），用于一些初始化的操作。

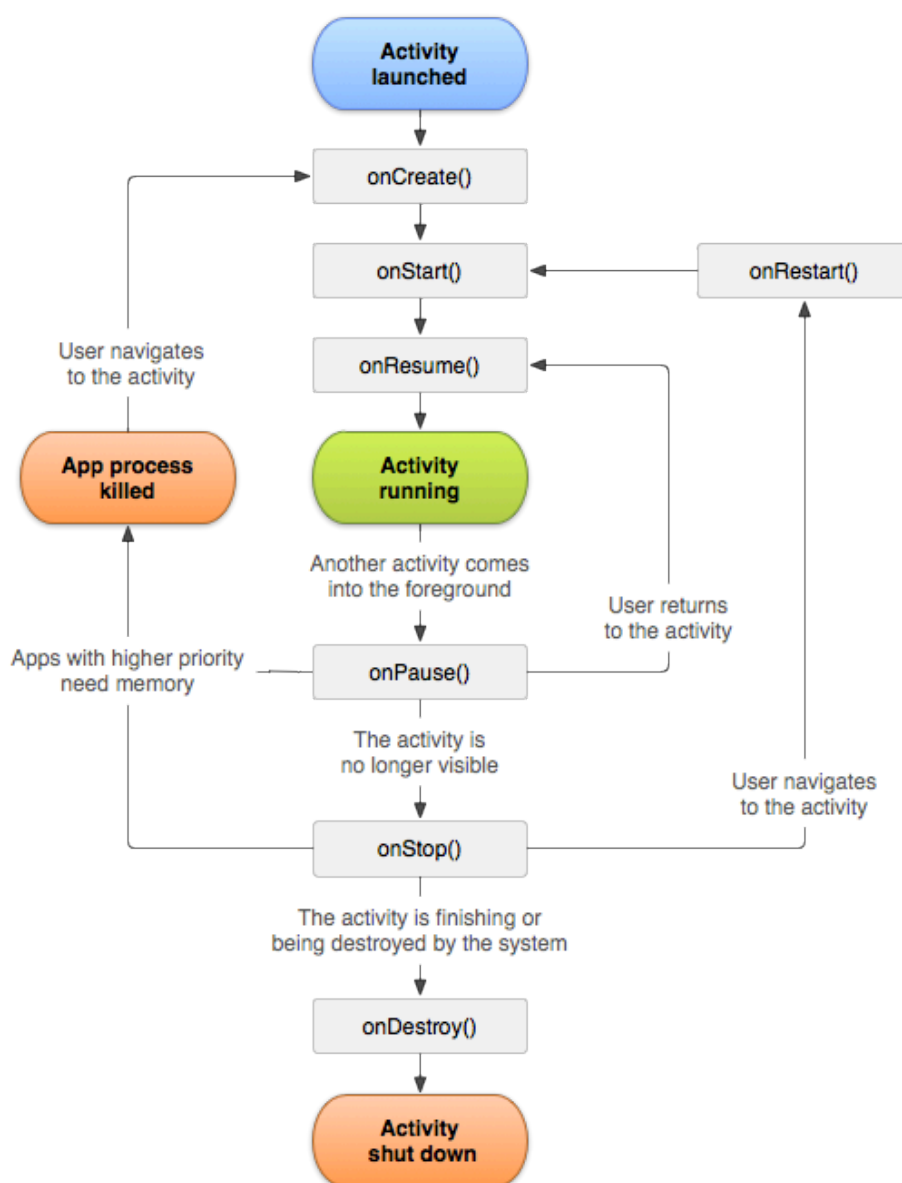
onStart()：通过activity生命周期图可以看出这个方法是在onCreate或者onRestart后调用的，（当前窗口被切走不可见了，当重新切回来的时候会调用onRestart->onStar）这个方法里做的事一般是比较少的。

onResume(): 这个方式是当activity获取焦点并可以和你交互的时候调用。这里的不宜做耗时较长的操作，因为那样当activity激活的时候时间会过长。

onPause(): 是和onResume相对的，当activity失去焦点的时候调用，这时候当系统内存低的时候activity容易被回收，在这个方法里可以保存一些重要的状态数据，但不应保存哪些耗时太长的数据，以避免第二个activity启动的时候时间过长。

onStop(): 是和onStart相对的，当activity不可见的时候被调用，这个时候activity很容易被系统回收，这里可以保存一些耗时较长或者不是很重要的数据。

onDestory(): 当activity结束的时候会调用这个方法，用于对一些资源进行释放。



activity生命周期

老师讲课的时候也说过在onCreate的时候需要调用一下setContentView这个方法，将写好的layout文件设置进去，以便activity知道需要显示哪些内容。尽管findViewById可以在任何时候调用，但是使用频繁的控件最好是声明好成员变量，在onCreate方法里通过findViewById赋一次值即可。因为findViewById也是一个比较耗时的操作，应避免频繁的findViewById操作。

activity的生命周期在开发页面的时候是比较重要的，需要牢记，当在相应的生命周期的回调里做了不该做的事情的时候你的app可能就会出现性能问题了。activity保存了展示给用户的各种状态，在哪个生命周期回调的时候需要进行什么工作，大家是可以在开发过程中根据自己的需求自由变通的，只有掌握了它们的限制条件才能做到游刃有余。讲生命周期的时候老师的demo可以练习一下，查看生命周期是怎么一个执行顺序。

2、Activity的通信方式

1) Intent通信，activity的启动是需要Intent来启动的，并且通过Intent可以传递一些数据，你的activity想要启动其他的activity也是需要使用Intent被你启动的activity可以是跟你在同一个进程里面的也可以是另外一个进程里面的activity（即app想要打开系统相机就可以通过Intent设置成相机的启动模式就可以了）。

```
Intent intent=new Intent();
intent.setClass(FirstActivity.this, SecondActivity.class);
//intent=new Intent(FirstActivity.this, SecondActivity.class);
startActivity(intent);
```

setClass就相当于启动模式，这种调用叫做显式调用，对应的还有隐式调用，隐式调用就需要AndroidManifest.xml中定义的<intent-filter/>了，首先在这里面定义好要监听的启动事件（这个事件也就是一个字符串），然后通过intent.setAction将定义好的事件写在里面就可以启动了。要启动相机就需要这种隐式调用的方式。

2) SharedPreferences: SharedPreferences 使用 xml 格式为 Android 应用提供一种永久的数据存贮方式，他是基于键值对的方式进行存储的，每一对数据都包含一个键和一个值，键就是你要记录的名称，值就是你要存储的数据，不过这种方式存储的数据形式有限。

3) 通过数据库以及文件（这个以后老师会讲，就不具体介绍了）。

3、Activity栈，栈的定义是限制仅在表的一端进行插入和删除运算的线性表。其特点是由上向下了先进的在栈底，后进的在栈顶，出栈的时候从栈顶开始出栈，是一个先进后出，后进先出的进出逻辑。

（以下四个Activity栈特点摘自老师ppt）

1) Android 是通过一种 Activity 栈的方式来管理 Activity 的，一个Activity的实例的状态决定它在栈中的位置。处于前台的Activity总是在栈的顶端，当前台的Activity因为异常或其它原因被销毁时，处于栈第二层的 Activity 将被激活，上浮到栈顶。当新的 Activity 启动入栈时，原 Activity 会被压入到栈的第二层。一个 Activity 在栈中的位置变化反映了它在不同状态间的转换。

2) 一个任务栈包含了一个activity的集合, android系统可以通过Task有序地管理每个 activity，并决定哪个Activity与用户进行交互，只有在任务栈栈顶的activity才可以跟用户进行交互。

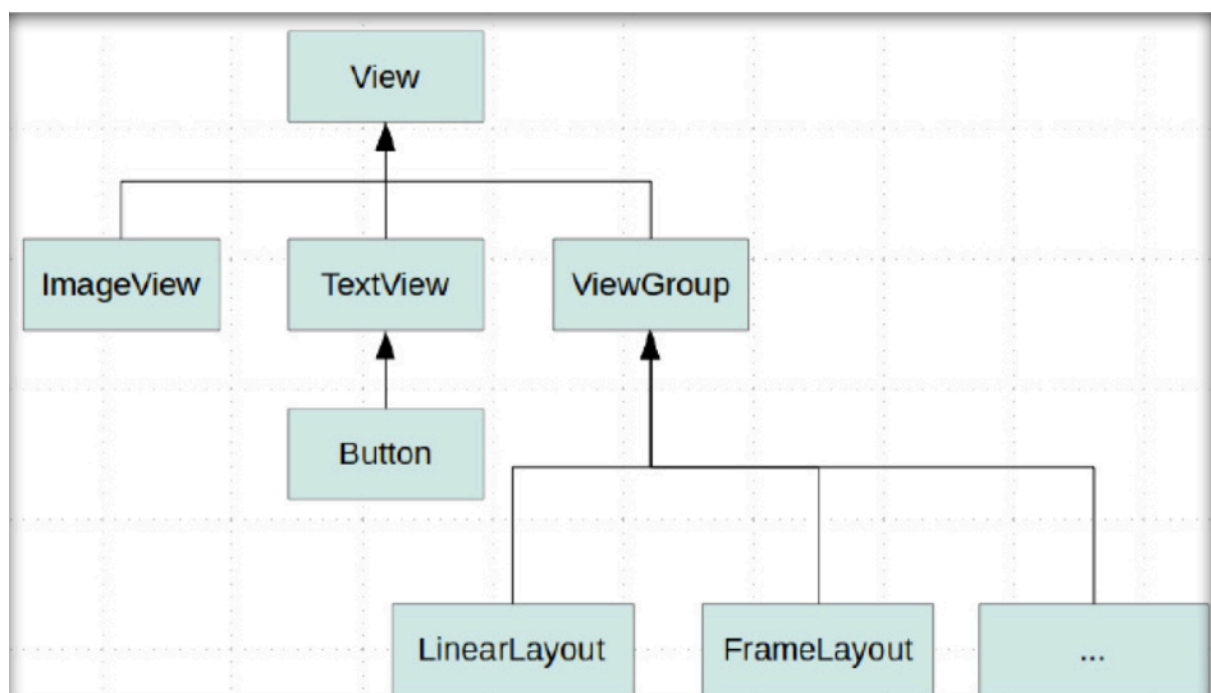
3) 除了最顶层即处在 Active 状态的 Activity 外，其它的 Activity 都有可能在系统内存不足时被回收，一个 Activity 的实例越是处在栈的底层，它被系统回收的可能性越大。系统负责管理栈中 Activity 的实例，它根据 Activity 所处的状态来改变其在栈中的位置。

4) 在退出应用程序时，必须把所有的任务栈中所有的activity清除出栈时，任务栈才会被销毁。当然任务栈也可以移动到后台, 并且保留了每一个activity的状态。可以有序的给用户列出它们的任务，同时也不会丢失Activity的状态信息。

4、Activity启动模式（包括standard，singleTop，singleTask，singleInstance）

这些启动模式会影响activity的入栈出栈的逻辑，并且singleInstance会独享一个任务栈，因为对于一些人不是很好理解，这里就不说了，有兴趣的可以百度（singleTop，singleTask是怎么入栈，singleTask是怎么出栈）。

二、View以及ViewGroup



android中用于显示的视图的主要分为两大类一个是ViewGroup一个是View。View主要是显示具体的按钮、文本、图片等视图。而ViewGroup顾名思义一个View的组，即一个容器，它继承自View，但是它用于添加并组织View的排列顺序。ViewGroup和View内部还封装了用户交互的一些操作及响应。

1、android的六大布局，这六大布局就是ViewGroup子类，也就是ViewGroup的不同实现，可以让用户以不同的方式来组织显示自己的View的。

1) **FrameLayout**(框架布局) 这个布局的特点是：放入其中的所有元素都被放置在最左上的区域，而且无法为这些元素指定一个确切的位置，下一个子元素会重叠覆盖上一个子元素。如果后一个元素比上一元素的宽高要大的话，只能看到最后一个元素，如果后一个元素比上一元素的宽高要小的话，可以看到的是一个叠加的效果。

2) **LinearLayout**（线性布局）这个布局的特点是：主要提供控件水平或者垂直排列的，每个子组件都是以垂直或水平的方式来线性排布(默认是垂直)。可以通过 `android:orientation="vertical"` 垂直线性布局,"horizontal"水平线性布局来设置。最好不要用默认，有时候AndroidStudio也会提示添加这个属性的。它包含的子元素可以通过 `android:layout_weight` 这个属性来设置自己的显示比例的。如果所有子元素设置的值一样的话就会平分这个layout的所占的空间。例如微信聊天列表用这个来实现的话就是，首先得需要两个LinearLayout，一个LinearLayout水平显示，另外一个垂直显示，水平显示的LinearLayout包含一个图片和一个垂直显示的LinearLayout，垂直显示的LinearLayout包含一个名称的文本和一个最后一条聊天记录文本。

3) **RelativeLayout**（相对布局）这个布局的特点是：以某一个组件为参照物，来定位下一个组件的位置的布局方式。顾名思义相对的布局，其中的元素可以通过它特有的属性显示在这个layout内部空间的任何位置，然后可以根据定好的元素来定义下个元素相对于上一个元素的位置。当然这个控件也可以实现FrameLayout的效果。如果用这个控件实现微信聊天列表用这个来实现的话就是，左边就是一个图片将它位置定好以后，可以设置聊天的名称在图片的右边显示，而最后一条记录的显示也是在图片的右边，并且在名称的下方的位置。

4) **AbsoluteLayout**（绝对定位布局）这个布局的特点是：采用坐标轴的方式定位组件，左上角是（0，0）点，往右X轴递增，往下Y轴递增，组件用于定位的属性为 `android:layout_x`和 `android:layout_y`来确定坐标。（但是由于Android手机的屏幕尺寸、分辨率存在较大差异，使用AbsoluteLayout无法兼顾适配问题，所以该布局已经过时，所以不推荐使用。）

5) **TableLayout**（表格布局）这个布局的特点是：类似Html里的Table。使用TableRow来布局，其中TableRow代表一行，TableRow的每一个子元素代表一列。当然

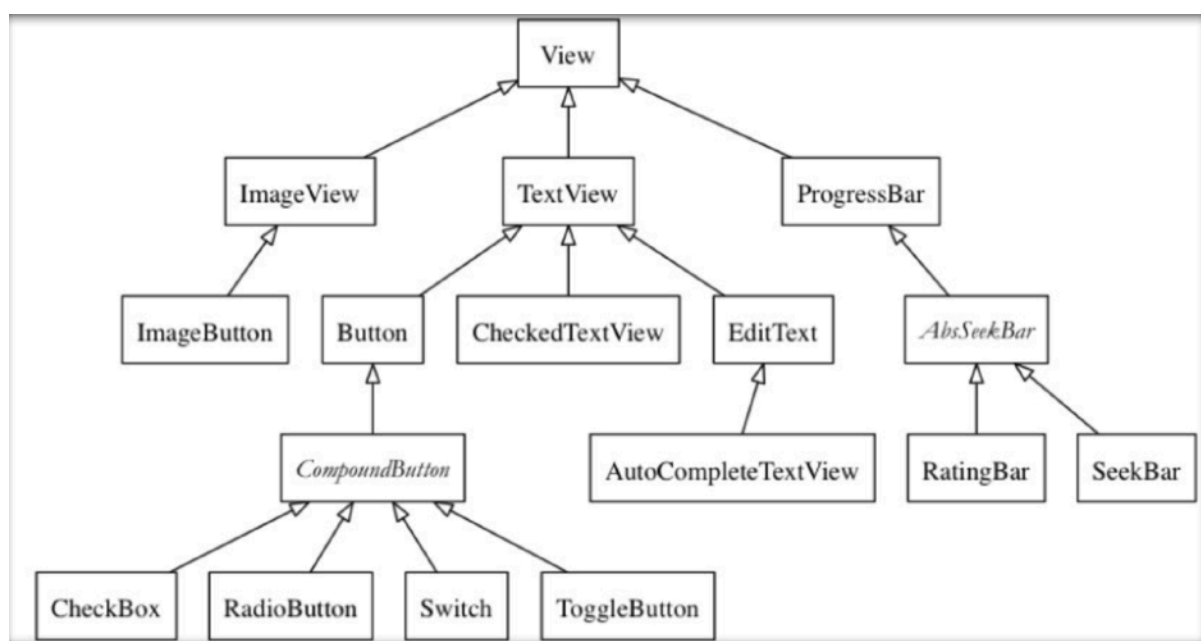
TableLayout也可以直接添加View以及ViewGroup，这个时候View或者ViewGroup就会独占一行。其实TableLayout是LinearLayout的子类，会有一些LinearLayout的影子。

6) GridLayout（网格布局）这个布局的特点是：这个控件是android 4.0才添加的，是所有布局中运用比较灵活的一种。GridLayout使用随意的网格来放置元素，可以完成row * column的布局。并且一个元素可以占多个单元格。有了这个控件计算器的布局方式可以通过它来实现，有兴趣的可以试一下。

以上这些布局可以根据自己的需要灵活运用，但要记住两点，1.尽量减少层级也就是嵌套不要太深（嵌套就是ViewGroup包含另一个ViewGroup.....），2.不要每个可以添加背景的控件都设置android:background，以避免过度绘制。过度绘制会使每一帧绘制的时间较长导致丢帧问题（如果不明白丢帧是什么就先记住那两点就好了）。

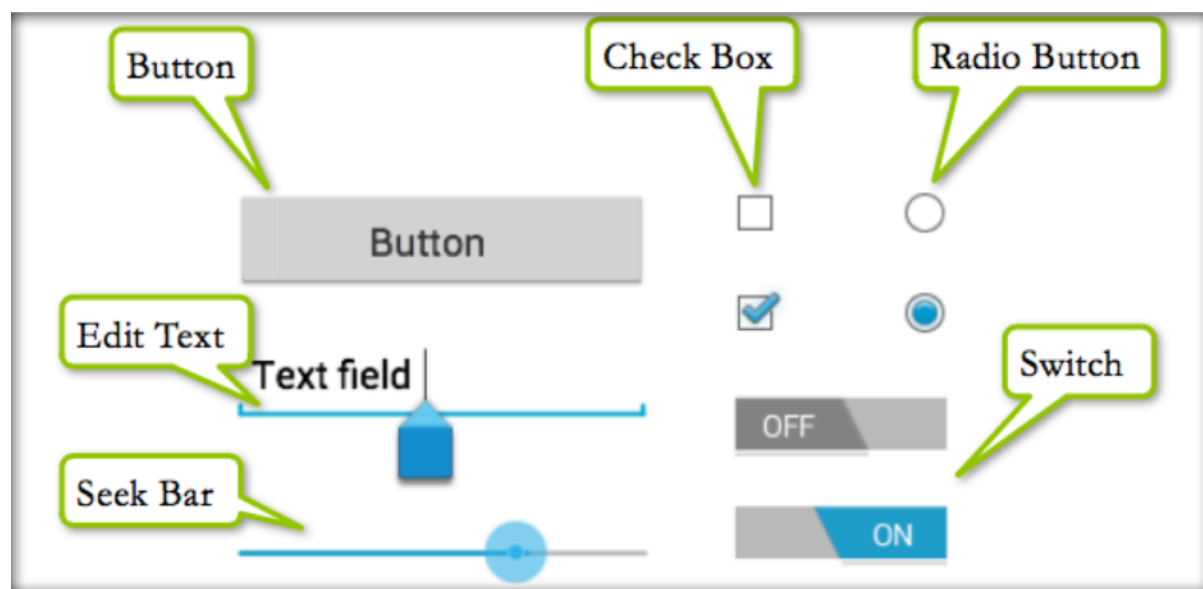
因为这个是总结性的文档，大家可以按照我写的这个百度一下各个控件的特性以及存在的属性，使用不同的layout其包含的子元素也会有一些不同的属性。这个就需要自己参照一些例子练习了，会了这些布局就可以完成基本的元素显示了（列表等形式以后老师进阶的时候会讲这里就不说了）。

2、android提供的基础控件

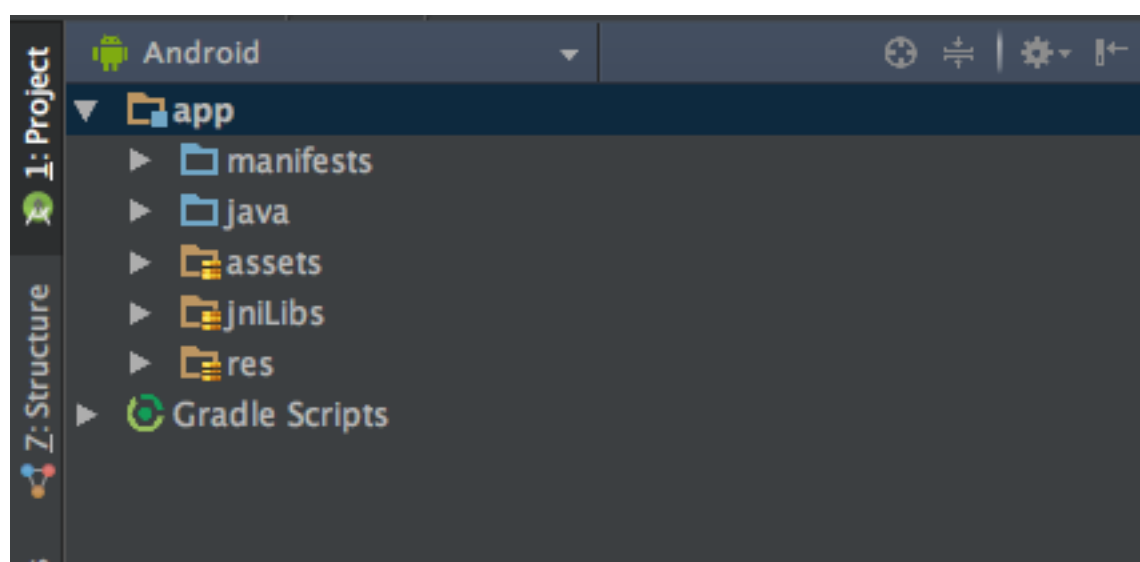


老师的课件里里面的图已经包含了一些常见的基本的控件的关系图以及展示图，展示图中除了列举的那几个之外还有TextView、ImageView、ProgressBar。这三个控件TextView是用于显示文本的，ImageView也都用过了是显示图片的（不过这里可以思考或者试一下android:src和android:background两个属性的区别，后者是每一个View以及ViewGroup都有的属性），ProgressBar是等待显示例如使用手机有些页面或者操作加载数据需要的时间比较长就会有有个转圈或者带进度的等待显示。这些控件也需要查看他

们属性的使用，分别记住他们特有的属性以及共有的属性，当然它们在不同layout下也会有一些不同的属性。这些属性可以通过多多练习的方式试试他们属性的作用，只有多练才能更好的更灵活的运用，以便以后能够灵活的布局。



三、android项目的结构（补充一下结构说明可能会有同学对结构不了解）



将新建的项目切到android试图下将会有这样一个结构，app就是代表这个项目。

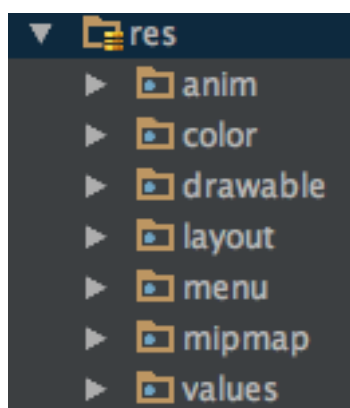
1、manifests这个目录是存放整个项目的注册清单文件，最开始说的那四大组件都是需要在这里注册（当然像broadcast还有另外的注册方式也可以用java代码进行住的）。

2、java这个目录就是包含java代码的地方了，通过java代码来实现一些业务逻辑，当然之前说的四大组件的实现部分也都被包含在这里面。

3、assets里包含的是一些原生资源文件，程序在编译的时候是不会编译这里面的文件的，比如可以放一些铃声，文本文件等。

4、jniLibs这里面包含一些第三方库文件，还有一些C，C++的库文件。就是扩展app基础库的地方。

5、res这个文件是app的一些资源，首先这里面除了图片都是用xml文件定义的。



1) anim这个目录是放一些动画文件的，在android中一些动画是可以用xml文件定义的。

2) color这个文件就是存放selector文件，selector其实也是一种xml文件，它可以设置一个控件的多种状态显示，如可以设置一个TextView字的颜色按下时显示的颜色，设置成没有按下的颜色为白色当你按下的时候可以变成黑色。在drawable里面也可以存在selector，selector的显示样式可以是图片也可以是颜色值。为了做区分最好是颜色的放在这里，使用图片的放在drawable里面。

3) drawable和mipmap这两个文件是存放图片的，区别是mipmap只能放png图片，在这里放图片的时候会对图片做一些优化以减小GPU压力（推荐把png图片放在这个目录里），drawable完全包含mipmap的作用，但是现在有了mipmap就要给它们做一下区分drawable尽量放一些xml自定义的图片（包括selector）还有一些9patch文件（9patch文件是为了更好的拉伸图片而定义的一种图片格式），如果这些文件放到mipmap会编译不过。

3) menu是放一些菜单文件的。这个可以查看一些应用的右上角都会有菜单，那些菜单都是在这里面定义的。

4) 主要是一些颜色，字符串、数字，样式，属性等文件，目前先掌握颜色和字符串就好。

6、GradleScripts

Gradle应该都有些了解了，安装的时候很费力气。它是一种编译打包的工具，它可以让你定义满足自己需要的构建逻辑，只要按照它的语法写入到build.gradle即可。初学的时候这里的配置应该不用修改。