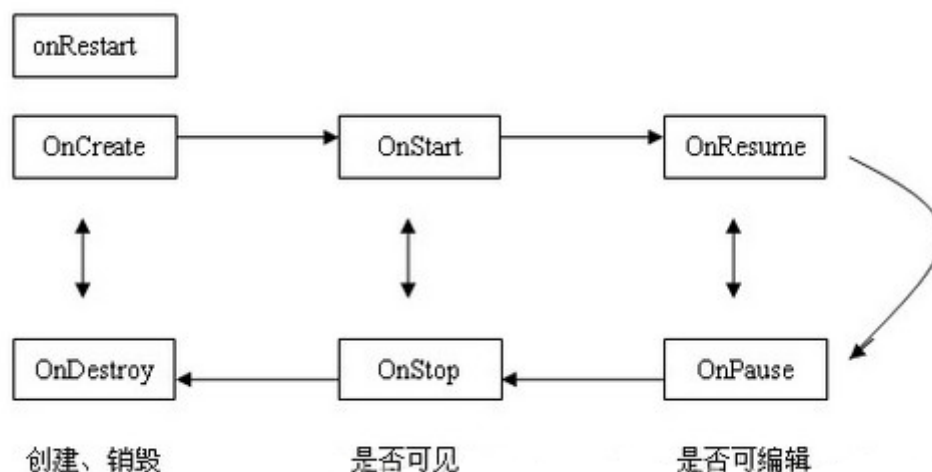


之前学习安卓的时候只是知道生命周期是什么，有哪几个，但具体的详细的的东西却不知道，后来看过《Android开发艺术探索》和大量博客之后，才觉得自己真正有点理解生命周期，本文是我对生命周期的认识的总结。

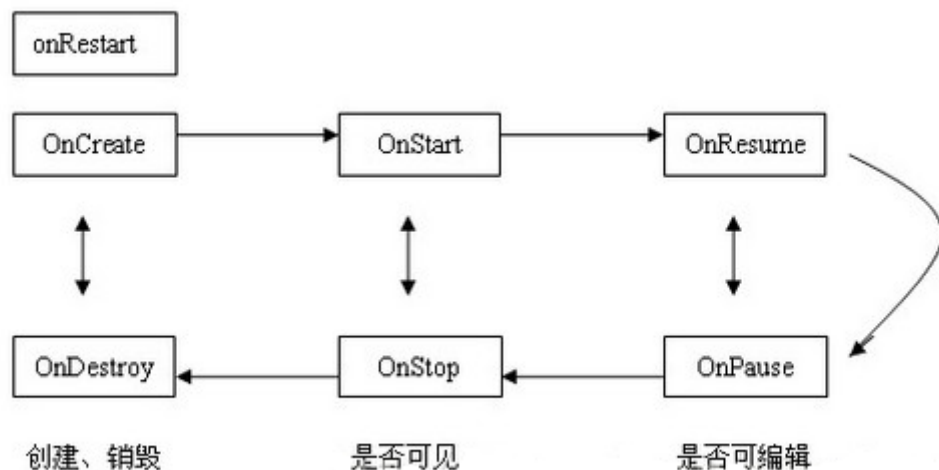
废话少说先上图。



相信学习安卓的人对这幅图都很熟悉，这是安卓Activity的生命周期活动图，详细而直观得表现了Activity各生命周期期间的关系。下面我来通过问答的方式来谈谈我对它们的认识。

1.生命周期中各个方法的含义和作用

1. `onCreate`: `create`表示创建，这是Activity生命周期的第一个方法，也是我们在android开发中接触的最多的生命周期方法。它本身的作用是进行Activity的一些初始化工作，比如使用 `setContentView` 加载布局，对一些控件和变量进行初始化等。但也有很多将很多与初始化无关的代码放在这，其实这是不规范的。此时**Activity还在后台，不可见**。所以动画不应该在这里初始化，因为看不到.....
2. `onStart`: `start`表示启动，这是Activity生命周期的第二个方法。此时Activity已经**可见了**，但是还没出现在前台，我们还看不到，无法与Activity交互。其实将Activity的初始化工作放在这也没有什么问题，放在 `onCreate` 中是由于官方推荐的以及我们开发的习惯。
3. `onResume`: `resume`表示继续、重新开始，这名字和它的职责也相同。此时Activity经过前两个阶段的初始化已经蓄势待发。Activity在这个阶段已经出现在前台并且**可见了**。这个阶段可以打开独占设备
4. `onPause`: `pause`表示暂停，当Activity要跳到另一个Activity或应用正常退出时都会执行这个方法。此时Activity在前台并**可见**，我们可以进行一些轻量级的存储数据和去初始化的工作，不能太耗时，因为在跳转Activity时只有当一个Activity执行完了 `onPause` 方法后另一个Activity才会启动，而且android中指定如果 `onPause` 在500ms即0.5秒内没有执行完毕的话就会强制关闭Activity。从生命周期图中发现可以在这快速重启，但这种情况其实很罕见，比如用户切到下一个Activity的途中按back键快速得切回来。
5. `onStop`: `stop`表示停止，此时Activity已经**不可见了**，但是Activity对象还在内存中，没有被销毁。这个阶段的主要工作也是做一些资源的回收工作。
6. `onDestroy`: `destroy`表示毁灭，这个阶段Activity被销毁，**不可见**，我们可以将还没释放的资源释放，以及进行一些回收工作。
7. `onRestart`: `restart`表示重新开始，Activity在这时**可见**，当用户按Home键切换到桌面后又切回来或者从后一个Activity切回前一个Activity就会触发这个方法。这里一般不做什么操作。



通过上面的了解我们发现其实Activity中的方法大都是两两对应的，只有onRestart方法散发着单身狗的清香。那么相邻的方法之间有什么区别呢？

2.onCreate和onStart之间有什么区别？

1. 可见与不可见的区别。前者不可见，后者可见。
2. 执行次数的区别。onCreate方法只在Activity创建时执行一次，而onStart方法在Activity的切换以及按Home键返回桌面再切回应用的过程中被多次调用。因此Bundle数据的恢复在onStart中进行比onCreate中执行更合适。
3. onCreate能做的事onStart其实都能做，但是onstart能做的事onCreate却未必适合做。如前文所说的，setContentView和资源初始化在两者都能做，然而想动画的初始化在onStart中做比较好。

3.onStart方法和onResume方法有什么区别？

1. 是否在前台。onStart方法中Activity可见但不在前台，不可交互，而在onResume中在前台。
2. 职责不同，onStart方法中主要还是进行初始化工作，而onResume方法，根据官方的建议，可以做开启动画和独占设备的操作。

4.onPause方法和onStop方法有什么区别？

1. 是否可见。onPause时Activity可见，onStop时Activity不可见，但Activity对象还在内存中。
2. 在系统内存不足的时候可能不会执行onStop方法，因此程序状态的保存、独占设备和动画的关闭、以及一些数据的保存最好在onPause中进行，但要注意不能太耗时。

5.onStop方法和onDestroy方法有什么区别？

onStop阶段Activity还没有被销毁，对象还在内存中，此时可以通过切换Activity再次回到该Activity，而onDestroy阶段Activity被销毁

6.为什么切换Activity时各方法的执行次序是

**(A)onPause→(B)onCreate→(B)onStart→(B)onResume→(A)onStop而不是
(A)onPause→(A)onStop→(B)onCreate→(B)onStart→(B)onResume**

1. 一个Activity或多或少会占有系统资源，而在官方的建议中，onPause方法将会释放掉很多系统资源，为切换Activity提供流畅性的保障，而不需要再等多两个阶段，这样做切换更快。
2. 按照生命周期图的表示，如果用户在切换Activity的过程中再次切回原Activity，是在onPause方法后直接调用onResume方法的，这样比onPause→onStop→onRestart→onStart→onResume要快得多。

7.与生命周期密切相关的onSaveInstanceState方法和onRestoreInstanceState方法在什么时候执行？

通过阅读源码会发现，当targetSdkVersion小于3时onSaveInstanceState是在onPause方法中调用的，而大于3时是在onStop方法中调用的。
而onRestoreInstanceState是在onStart之后、onResume之前调用的。