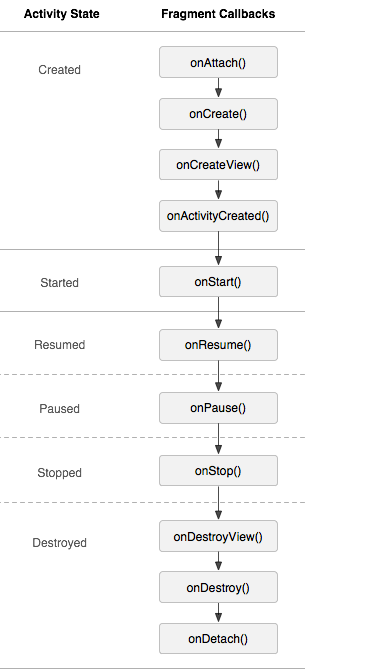
Activity与Fragment生命周期图：



Activity介绍：

**1.启动Activity：系统会先调用onCreate方法，然后调用onStart方法，最后调用onResume，Activity进入运行状态。**

**2.当前Activity被其他Activity覆盖其上或被锁屏：系统会调用onPause方法，暂停当前Activity的执行。**

**3.当前Activity由被覆盖状态回到前台或解锁屏：系统会调用onResume方法，再次进入运行状态。**

**4.当前Activity转到新的Activity界面或按Home键回到主屏，自身退居后台：系统会先调用onPause方法，然后调用onStop方法，进入停滞状态。**

**5.用户后退回到此Activity：系统会先调用onRestart方法，然后调用onStart方法，最后调用onResume方法，再次进入运行状态。**

**6.当前Activity处于被覆盖状态或者后台不可见状态，即第2步和第4步，系统内存不足，杀死当前Activity，而后用户退回当前Activity：再次调用onCreate方法、onStart方法、onResume方法，进入运行状态。**

**7.用户退出当前Activity：系统先调用onPause方法，然后调用onStop方法，最后调用onDestory方法，结束当前Activity。**

Fragment介绍：

**场景演示 : 切换到该Fragment**

**11-29 14:26:35.095: D/AppListFragment(7649): onAttach  
11-29 14:26:35.095: D/AppListFragment(7649): onCreate  
11-29 14:26:35.095: D/AppListFragment(7649): onCreateView  
11-29 14:26:35.100: D/AppListFragment(7649): onActivityCreated  
11-29 14:26:35.120: D/AppListFragment(7649): onStart  
11-29 14:26:35.120: D/AppListFragment(7649): onResume**

**屏幕灭掉：**

**11-29 14:27:35.185: D/AppListFragment(7649): onPause  
11-29 14:27:35.205: D/AppListFragment(7649): onSaveInstanceState  
11-29 14:27:35.205: D/AppListFragment(7649): onStop**

**屏幕解锁**

**11-29 14:33:13.240: D/AppListFragment(7649): onStart  
11-29 14:33:13.275: D/AppListFragment(7649): onResume**

**切换到其他Fragment:  
11-29 14:33:33.655: D/AppListFragment(7649): onPause  
11-29 14:33:33.655: D/AppListFragment(7649): onStop  
11-29 14:33:33.660: D/AppListFragment(7649): onDestroyView**

**切换回本身的Fragment:**

**11-29 14:33:55.820: D/AppListFragment(7649): onCreateView  
11-29 14:33:55.825: D/AppListFragment(7649): onActivityCreated  
11-29 14:33:55.825: D/AppListFragment(7649): onStart  
11-29 14:33:55.825: D/AppListFragment(7649): onResume**

**回到桌面**

**11-29 14:34:26.590: D/AppListFragment(7649): onPause  
11-29 14:34:26.880: D/AppListFragment(7649): onSaveInstanceState  
11-29 14:34:26.880: D/AppListFragment(7649): onStop**

**回到应用**

**11-29 14:36:51.940: D/AppListFragment(7649): onStart  
11-29 14:36:51.940: D/AppListFragment(7649): onResume**

**退出应用**

**11-29 14:37:03.020: D/AppListFragment(7649): onPause  
11-29 14:37:03.155: D/AppListFragment(7649): onStop  
11-29 14:37:03.155: D/AppListFragment(7649): onDestroyView  
11-29 14:37:03.165: D/AppListFragment(7649): onDestroy  
11-29 14:37:03.165: D/AppListFragment(7649): onDetach**

Fragment每个生命周期方法的意义、作用：  
setUserVisibleHint()：设置Fragment可见或者不可见时会调用此方法。在该方法里面可以通过调用getUserVisibleHint()获得Fragment的状态是可见还是不可见的，如果可见则进行懒加载操作。  
onAttach()：执行该方法时，Fragment与Activity已经完成绑定，该方法有一个Activity类型的参数，代表绑定的Activity，这时候你可以执行诸如mActivity = activity的操作。  
onCreate()：初始化Fragment。可通过参数savedInstanceState获取之前保存的值。  
onCreateView()：初始化Fragment的布局。加载布局和findViewById的操作通常在此函数内完成，但是不建议执行耗时的操作，比如读取数据库数据列表。  
onActivityCreated()：执行该方法时，与Fragment绑定的Activity的onCreate方法已经执行完成并返回，在该方法内可以进行与Activity交互的UI操作，所以在该方法之前Activity的onCreate方法并未执行完成，如果提前进行交互操作，会引发空指针异常。  
onStart()：执行该方法时，Fragment由不可见变为可见状态。  
onResume()：执行该方法时，Fragment处于活动状态，用户可与之交互。  
onPause()：执行该方法时，Fragment处于暂停状态，但依然可见，用户不能与之交互。  
onSaveInstanceState()：保存当前Fragment的状态。该方法会自动保存Fragment的状态，比如EditText键入的文本，即使Fragment被回收又重新创建，一样能恢复EditText之前键入的文本。  
onStop()：执行该方法时，Fragment完全不可见。  
onDestroyView()：销毁与Fragment有关的视图，但未与Activity解除绑定，依然可以通过onCreateView方法重新创建视图。通常在ViewPager+Fragment的方式下会调用此方法。  
onDestroy()：销毁Fragment。通常按Back键退出或者Fragment被回收时调用此方法。  
onDetach()：解除与Activity的绑定。在onDestroy方法之后调用。  
  
Fragment生命周期执行流程（注意红色的不是生命周期方法）：  
Fragment创建：setUserVisibleHint()->onAttach()->onCreate()->onCreateView()->onActivityCreated()->onStart()->onResume()；  
Fragment变为不可见状态（锁屏、回到桌面、被Activity完全覆盖）：onPause()->onSaveInstanceState()->onStop()；  
Fragment变为部分可见状态（打开Dialog样式的Activity）：onPause()->onSaveInstanceState()；  
Fragment由不可见变为活动状态：onStart()->OnResume()；  
Fragment由部分可见变为活动状态：onResume()；  
退出应用：onPause()->onStop()->onDestroyView()->onDestroy()->onDetach()（注意退出不会调用onSaveInstanceState方法，因为是人为退出，没有必要再保存数据）；  
Fragment被回收又重新创建：被回收执行onPause()->onSaveInstanceState()->onStop()->onDestroyView()->onDestroy()->onDetach()，重新创建执行onAttach()->onCreate()->onCreateView()->onActivityCreated()->onStart()->onResume()->setUserVisibleHint()；  
横竖屏切换：与Fragment被回收又重新创建一样。