

另外还有补充

onWindowFocusChanged、onSaveInstanceState、onRestoreInstanceState

下页🡪

1.onWindowFocusChanged方法：在Activity窗口获得或失去焦点时被调用，例如创建时首次呈现在用户面前；当前Activity被其他Activity覆盖；当前Activity转到其他Activity或按Home键回到主屏，自身退居后台；用户退出当前Activity。以上几种情况都会调用onWindowFocusChanged，并且当Activity被创建时是在onResume之后被调用，当Activity被覆盖或者退居后台或者当前Activity退出时，它是在onPause之后被调用，如图所示：

这个方法在某种场合下还是很有用的，例如程序启动时想要获取视特定视图组件的尺寸大小，在onCreate中可能无法取到，因为窗口Window对象还没创建完成，这个时候我们就需要在onWindowFocusChanged里获取；如果大家已经看过我写的Android动画之Frame Animation这篇文章就会知道，当时试图在onCreate里加载frame动画失败的原因就是因为窗口Window对象没有初始化完成，所以最后我将加载动画的代码放到了onWindowFocusChanged中，问题迎刃而解。不过大家也许会有疑惑，为什么我在代码里将它注释掉了，因为对当前Activity每一个操作都有它的执行log，我担心这会影响到整个流程的清晰度，所以将它注掉，大家只要了解它应用的场合和执行的顺序就可以了。

2.onSaveInstanceState：(1)在Activity被覆盖或退居后台之后，系统资源不足将其杀死，此方法会被调用；(2)在用户改变屏幕方向时，此方法会被调用；(3)在当前Activity跳转到其他Activity或者按Home键回到主屏，自身退居后台时，此方法会被调用。第一种情况我们无法保证什么时候发生，系统根据资源紧张程度去调度；第二种是屏幕翻转方向时，系统先销毁当前的Activity，然后再重建一个新的，调用此方法时，我们可以保存一些临时数据；第三种情况系统调用此方法是为了保存当前窗口各个View组件的状态。onSaveInstanceState的调用顺序是在onPause之前。

3.onRestoreInstanceState：(1)在Activity被覆盖或退居后台之后，系统资源不足将其杀死，然后用户又回到了此Activity，此方法会被调用；(2)在用户改变屏幕方向时，重建的过程中，此方法会被调用。我们可以重写此方法，以便可以恢复一些临时数据。onRestoreInstanceState的调用顺序是在onStart之后。