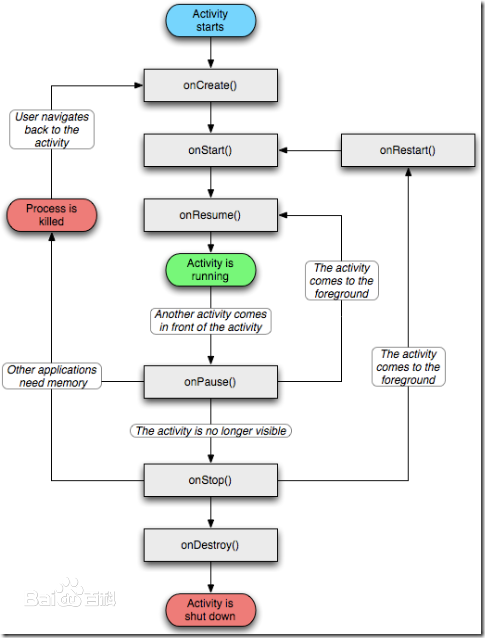
1. **Activity的生命周期**



被激活时 ： oncreate() --onstart()(activity处于可见状态时)--onresume()(得到用户焦点)

被遮挡时: onpause() --onstop()(处于不可见状态)

(只按锁屏时的效果) ------------onpause()

(按home键时的效果)---------onpause()------onstop()

恢复显示时：onreatart()---onstart()---onresume()

销毁时：onpause()----onstop()--ondestory()

onWindowFocusChanged :当用户获得 或者失去焦点时 调用 一般在onresume() onpause()

、onSaveInstanceState、onRestoreInstanceState方法 :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

**当你的程序中某一个Activity A在运行时,主动或被动地运行另一个新的Activity B,这个时候A会执行onSaveInstanceState().B完成以后又会来找A,这个时候就有两种情况:一是A被回收,二是A没有被回收,被回收的A就要重新调用onCreate()方法,不同于直接启动的是这回onCreate()里是带上了参数savedInstanceState；而没被收回的就直接执行onResume(),跳过onCreate()了.**

**2.Activity有三个关键的循环：**

1. 完整生命周期: 即一个activity从出现到消失,对应的周期方法是从onCreate到onDestroy .例如：某个Activity有一个在后台运行的线程，用于从网络下载数据，则该Activity可以在onCreate()中创建线程,在onDestory()中停止线程。

2. 可见生命周期: 从onStart()到onStop()方法,在这个周期中,用户可以看见Activity,但不一定能够与之交互,这两个方法可以随着状态的变化被执行多次.

3. 前景生命周期(焦点生命周期) :从onResume()到onPause()方法,在这个周期中,Activity处在其它Activity之前,即在Activity栈的最顶端,负责与用户进行交互,伴随着状态的变化,这两个方法可以被执行多次.

1. **请介绍下Android中常用的五种布局.**

最常用的布局方式为LinearLayout. RelativeLayout. FrameLayout. TableLayout AbsoluteLayout.

LinearLayout ：可以通过orientation属性设置线性排列的方向是垂直(vertical)还是纵向 width比重,可以进行复杂的布局.

AbsoluteLayout. ：可以让子元素指定准确的x/y坐标值,他是绝对坐标,所以在实际中不提倡使用.

RelativeLayout ：允许子元素指定他们相对于其它元素或父元素的位置,在实际中比较常用.

TableLayout 将子元素的位置分配到行或列中 用GridView排版的. Padding是文字相对于边框,而Margin是边框相对于父窗体.

FrameLayout 最简单的布局方式,放置的控件都只能罗列到左上角,控件会有重叠,不能进行复杂的布局

5**. Android 动画有哪几种？** **描述一下**

两种。 Tween动画和Frame动画。 Tween动画主要是透明度、尺寸伸缩、旋转、位移等效果。

Frame动画可以理解成gif， 一帧一帧的显示图片。比较常用的有滚动条效果。