**Android 滑动**

[**Android**](http://lib.csdn.net/base/15)系统手机屏幕的**左上角**为坐标系，同时y轴方向与笛卡尔坐标系的y轴方向想反。通过提供的api如getLeft , getTop, getBottom, getRight可以获得控件在parent中的相对位置。

调用View的scrollTo()和scrollBy()是用于滑动View中的内容，而不是把某个View的位置进行改变。如果想改变莫个View在屏幕中的位置，可以使用如下的方法。

调用public void offsetLeftAndRight(int offset)用于左右移动方法或public void [offsetTopAndBottom](http://developer.android.com/reference/android/view/View.html#offsetTopAndBottom(int))(int offset)用于上下移动。如：button.offsetLeftAndRignt(300)表示将button控件向左移动300个像素。

scrollTo(int x, int y) 是将View中**内容**滑动到相应的位置；

scrollBy(int x, int y)其实是对scrollTo的包装，移动的是相对位置。

public final int getScrollX() {

return mScrollX;

}

public final int getScrollY() {

return mScrollY;

}

public void scrollBy(int x, int y) {

scrollTo(mScrollX + x, mScrollY + y);

}

**Scroller的方法：**

mScroller.getCurrX() //获取mScroller当前水平滚动的位置

mScroller.getCurrY() //获取mScroller当前竖直滚动的位置

mScroller.getFinalX() //获取mScroller最终停止的水平位置

mScroller.getFinalY() //获取mScroller最终停止的竖直位置

mScroller.setFinalX(**int** newX) //设置mScroller最终停留的水平位置，无动画效果，直接跳到目标位置

mScroller.setFinalY(**int** newY) //设置mScroller最终停留的竖直位置，无动画效果，直接跳到目标位置

//滚动，startX, startY为开始滚动的位置，dx,dy为滚动的偏移量, duration为完成滚动的时间

mScroller.startScroll(int startX, int startY, int dx, int dy) //使用默认完成时间250ms

mScroller.startScroll(int startX, int startY, int dx, int dy, int duration)

mScroller.computeScrollOffset() //返回值为boolean，true为滚动尚未完成，false为滚动完成通常放在View.computeScroll()中，用来判断是否滚动是否结束。

来看看View滚动的实现原理，我们先调用Scroller的startScroll()方法来进行一些滚动的初始化设置，然后迫使View进行绘制，我们调用View的invalidate()或postInvalidate()就可以重新绘制View，绘制View的时候会触发computeScroll()方法，我们重写computeScroll()，在computeScroll()里面先调用Scroller的computeScrollOffset()方法来判断滚动有没有结束，如果滚动没有结束我们就调用scrollTo()方法来进行滚动，该scrollTo()方法虽然会重新绘制View,但是我们还是要手动调用下invalidate()或者postInvalidate()来触发界面重绘，重新绘制View又触发computeScroll()，所以就进入一个循环阶段，这样子就实现了在某个时间段里面滚动某段距离的一个平滑的滚动效果