1、item内如果有button等控件时，在监听listview的onitemclick事件时，焦点会被item内的button、imagebutton等控件抢走，从而导致在listview设置了onitemclick事件后不会被触发。解决方法是在初始化item的时候屏蔽掉其内部button等控件的焦点获取，具体方法可以在自定义item的根控件中调用：

setDescendantFocusability(ViewGroup.FOCUS\_BLOCK\_DESCENDANTS);

这样就能阻塞字控件抢夺焦点，listview的onitemclick就能被正确触发，同时对item内部的button等控件也没有影响，他们在被点击时照样可以触发自身的点击事件。

2、当listview需要添加headerview时，可以通过调用listview的addHeaderView(headView, null, false) 方法，该方法还有一个重载方法 addHeaderView(headView);这两个方法的区别是前一个方法可以控制header是否可以被selected，如果不想被selected则将第三个参数设置成false；

3、接着上面说的添加header，添加header时调用的addHeaderView方法必须放在listview.setadapter前面，意思很明确就是如果想给listview添加头部则必须在给其绑定adapter前添加，否则会报错。原因是当我们在调用setAdapter方法时会android会判断当前listview是否已经添加header，如果已经添加则会生成一个新的HeaderViewListAdapter，这个新的HeaderViewListAdapter包含我们设置的adapter所有内容以及listview的header和footer。所以当我们在给listview添加了header后在程序中调用listview.getadapter时返回的是tempadapter而不是我们通过setadapter传进去的adapter。如果没有设置adapter则HeaderViewListAdapter与我们自己的adapter是一样的。listview.getadapter().getcount()方法返回值会比我们预期的要大，原因是添加了header。

4、接着上面的HeaderViewListAdapter说，我们自定义adapter里面的getitem方法里面返回的position是不包括header的，是我们自定义adapter中数据position编号从0开始，也就是说与我们传进去的list的位置是一样的。

@Override

public View getView(int position, View convertView, ViewGroup parent) {

// TODO Auto-generated method stub

Log.i("adapter", "position:"+position); //这个position就是我们数据的真实位置

}

而listview的onitemclick方法中：

public void onItemSelected(AdapterView<?> parent, View view, int position, long id)

position是当前click的位置，这个位置是指在HeaderViewListAdapter中的位置，从0开始如果listview中添加了header则0代表header。

在代码中使用 listView.addHeaderView(…) 方法可以在ListView组件上方添加上其他组件，并且连结在一起像是一个新组件。如果多次使用 .addHeaderView(…) ,则最先添加的组件在最上方，按添加的先后顺序由上到下罗列。   
此时listView 的 position = 0 的位置对应的是view1，而不再是原来listView中的第一条了。如下图所示：   
