上一节中，我们使用SAX方式解析xml文档， SAX方式是基于事件驱动的。当然android的事件机制是基于回调函数的。在这一节中，我们用另外一种方式解析xml文档，这种方式也是基于事件驱动的，与SAX方式一样，它就是PULL方式。只不过PULL方式读xml回调方法返回的是数字。

读取到xml的声明返回 START\_DOCUMENT;

读取到xml的结束返回 END\_DOCUMENT ;

读取到xml的开始标签返回 START\_TAG

读取到xml的结束标签返回 END\_TAG

读取到xml的文本返回 TEXT

xml数据结果还是采用我们先前使用的river.xml文件。

采用PULL方式与SAX大同小异，重点在于我们需要知道导航到什么标签时候做什么就行了，依据上一节SAX处理方式的思路，我们也可以在这一节中一样处理，基本方法是：

基本处理方式是：当PULL解析器导航到文档开始标签时就开始实例化list集合用来存贮数据对象。导航到元素开始标签时回判断元素标签类型，如果是river标签，则需要实例化River对象了，如果是其他类型，则取得该标签内容并赋予River对象。当然它也会导航到文本标签，不过在这里，我们可以不用。

根据以上的解释，我们可以得出以下处理xml文档逻辑：

1：当导航到XmlPullParser.START\_DOCUMENT，可以不做处理，当然你可以实例化集合对象等等。

2：当导航到XmlPullParser.START\_TAG，则判断是否是river标签，如果是，则实例化river对象，并调用getAttributeValue方法获取标签中属性值。

3：当导航到其他标签，比如Introduction时候，则判断river对象是否为空，如不为空，则取出Introduction中的内容，nextText方法来获取文本节点内容

4：当然啦，它一定会导航到XmlPullParser.END\_TAG的，有开始就要有结束嘛。在这里我们就需要判读是否是river结束标签，如果是，则把river对象存进list集合中了，并设置river对象为null.

由以上的处理逻辑，我们可以得出以下代码：

public List<River> parse(String xmlPath){

List<River> rivers=new ArrayList<River>();

River river=null;

InputStream inputStream=null;

//获得XmlPullParser解析器

XmlPullParser xmlParser = Xml.newPullParser();

try {

//得到文件流，并设置编码方式

inputStream=this.context.getResources().getAssets().open(xmlPath);

xmlParser.setInput(inputStream, "utf-8");

//获得解析到的事件类别，这里有开始文档，结束文档，开始标签，结束标签，文本等等事件。

int evtType=xmlParser.getEventType();

//一直循环，直到文档结束

while(evtType!=XmlPullParser.END\_DOCUMENT){

switch(evtType){

case XmlPullParser.START\_TAG:

String tag = xmlParser.getName();

//如果是river标签开始，则说明需要实例化对象了

if (tag.equalsIgnoreCase(RIVER)) {

river = new River();

//取出river标签中的一些属性值

river.setName(xmlParser.getAttributeValue(null, NAME));

river.setLength(Integer.parseInt(xmlParser.getAttributeValue(null, LENGTH)));

}else if(river!=null){

//如果遇到introduction标签，则读取它内容

if(tag.equalsIgnoreCase(INTRODUCTION)){

river.setIntroduction(xmlParser.nextText());

}else if(tag.equalsIgnoreCase(IMAGEURL)){

river.setImageurl(xmlParser.nextText());

}

}

break;

case XmlPullParser.END\_TAG:

//如果遇到river标签结束，则把river对象添加进集合中

if (xmlParser.getName().equalsIgnoreCase(RIVER) && river != null) {

rivers.add(river);

river = null;

}

break;

default:break;

}

//如果xml没有结束，则导航到下一个river节点

evtType=xmlParser.next();

}

} catch (XmlPullParserException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}catch (IOException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

return rivers;

}