

```

package com.sxt.server.basic;

import java.io.IOException;

import javax.xml.parsers.ParserConfigurationException;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

/**
 * 熟悉SAX解析流程
 * @author 江
 *
 */

public class XmlTest02 {
    public static void main(String[] args) throws ParserConfigurationException,
SAXException, IOException {
        //SAX解析
        //1. 获取解析工厂
        SAXParserFactory factory=SAXParserFactory.newInstance();
        //2. 从解析工厂获取解析器
        SAXParser parse=factory.newSAXParser();
        //3. 编写处理器
        //4. 加载文档Document注册处理器
        PHandler handler=new PHandler();
        //5. 解析
        parse.parse(Thread.currentThread().getContextClassLoader()

.getResourceAsStream("com/sxt/server/basic/p.xml"), handler);
    }
}

```

```
}  
}
```

```
class PHandler extends DefaultHandler{  
    public void startDocument() throws SAXException{  
        System.err.println("---解析文档开始---");  
    }  
  
    @Override  
    public void characters(char[] ch, int start, int length) throws  
SAXException {  
        String contends=new String(ch,start,length).trim();  
        if(contends.length()>0){  
            System.err.println("内容为->" +contends);  
        }else {  
            System.err.println("内容为->空");  
        }  
    }  
  
    @Override  
    public void startElement(String uri, String localName, String qName,  
Attributes attributes) throws SAXException {  
        System.err.println(qName+"-->解析开始");  
    }  
  
    @Override  
    public void endElement(String uri, String localName, String qName)  
throws SAXException {  
        System.err.println(qName+"-->解析结束");  
    }  
  
    @Override  
    public void endDocument() throws SAXException {  
        System.err.println("---解析文档结束---");  
    }  
}
```

