# XML简介

## 认识XML

在项目开发中，HTML的主要功能是进行数据显示，而要想进行数据存储结构的规范化 就需要使用XML。而在项目的实际开发之中更是大量地应用了XML技术，包括Tomcat 中的配置文件、Spring等框架配置文件都应用了XML技术。XML有自己的语法，而且 所有的标签元素都可以由用户任意定义。

XML提供了一套跨平台、跨网络、跨程序的语言的数据描述方式，使用XML可以方便地实现数据交换或存储、系统配置、内容管理等常见功能。

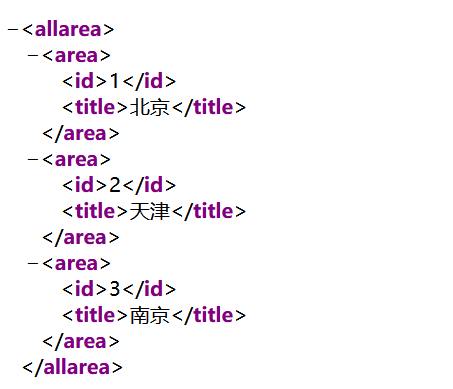
## XML与HTML

XML与HTML类似，都属于标记语。但两者有明显的不同。 最大的区别就是HTML语 言中的标签都是固定的，已经提供好了的，且HTML以显示 为主，而XML语言中的标 签都是由用户自定义的，以保存数据为主。比较如下：

|  |  |  |
| --- | --- | --- |
| **比较内容** | **HTML** | **XML** |
| 可扩展性 | 标签都已定义好，不可扩展 | 是无标签语言，自定义标签 |
| 侧重点 | 侧重于如何显示信息 | 侧重如何结构化的描述与存储信息 |
| 语法要求 | 相对健壮，可缺少结束标签，语法相对松散 | 严格要求嵌套、配对，遵循统一的顺序结构要求 |
| 大小写 | 不区分大小写 | 严格区分大小写 |

## 运行XML文件

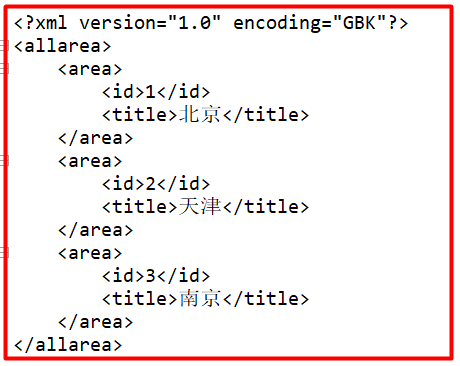
XML文件也可以像HTMl文件一样在浏览器打开，但XML在浏览器中显示时是以一种树 状的形式显示的。如下：



在开发中，当我们编写好一个XML文件后，可以通过浏览器运行它，如果能够以树状 的形式显示，则表示该XML文件没有错误；如果无法显示，则会提示错误的位置。借 此，我们可以通过在浏览器运行XML文件来判断该XML中是否存在错误。

## XML文件的定义

所有的XML文件由前导区和数据区两部分组成。一个完整的XML文件示例如下：



1. **前导区**

前导区用于规定XML文件的一些属性。如下：  
 **<?xml version=”1.0” encoding=”GBK” standalone=”yes”?>**

·version属性表示使用的XML版本

·encoding属性表示XML文件的编码格式，如果有XML文件中有中文，则必须

使用中文编码。

·standalone属性表示此文件是否是独立运行，如果需要进行显示可以使用CSS， 取值为‘yes’或‘no’，一般都忽略此属性不写。

注意：以上属性必须按照固定的顺序编写，顺序是：version、encoding、standalone。

1. **数据区**

数据区是真正的主体部分，用来存放数据。数据区必须有一个根标签，根标签下可

以存放多个子标签，且标签都是成对的，需要完结，每一个标签都区分大小写。

## XML自定义标签的属性

学习HTML可知，在HTML的各个元素中都会存在属性，如在<form>标签中可以编写 action属性。在XML中的标签虽然是自定义的，但也可以为其定义属性，属性值用”” 括起来。

## 实体参照

在XML文件中，由于‘<’和‘>’等符号都有特殊的含义。所以为了可以在XML文件 中可以显示这些符号，提供了特殊的表示方式。

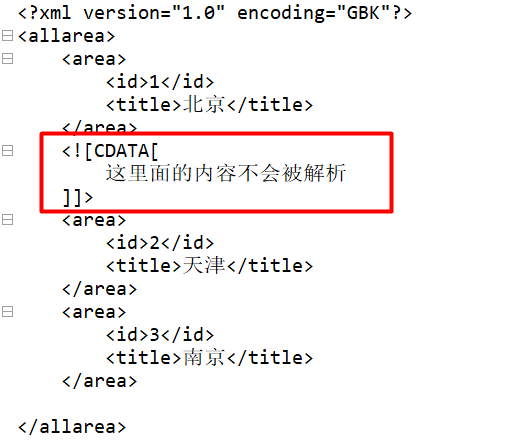
|  |  |
| --- | --- |
| **表示方法** | **对应字符** |
| &amp; | & |
| &lt; | < |
| &gt; | > |
| &quot; | “ |
| &apos; | ‘ |

## CDATA语法

XML语言中提供了CDATA标签来标识数据，当解析XML文件时不会处理CDATA标签中 的内容，相当于HTML文件中的注释。其语法格式为：

**<![CDATA[ 不会解析的内容 ]]>**

示例如下：



# XML解析

XML解析指的是利用程序取出XML文档中的标签中的内容。在XML解析中，W3C定义 了SAX和DOM两种解析方式。

## DOM解析

DOM解析的原理是指XML文档转成DOM节点树，通过对节点树进行操作，来实现对 XML 文档数据的操作，包括读取数据，修改数据，输出XML文件，带来了很大的灵活 性。但由于DOM分析器把整个XML文档转化成DOM树放在了内存中，因此，当文档 比较大或者结构比较复杂时，对内存的需求就比较高，而且对于结构复杂的树的遍历也 是一项耗时的操作。所以，DOM分析器对机器性能的要求比较高，程序的效率并不十 分理想，但由于DOM分析器所采用的树结构的思想与XML文档的结构相吻合，同时鉴 于其带来的方便，因此DOM分析器还是由很广泛的使用价值的。（具体使用方法请参 数教材书）

## SAX解析

SAX（Simple APIs for XML，操作XML的简单接口）解析采用顺序模式进行访问，是一种 快速读取XML数据的方式，当使用SAX解析器进行操作时会触发一系列的事件。当扫 描到文档开始与结束、元素开始与结束时都会调用相关的处理方法进行处理。SAX主 要事件如下：

|  |  |  |
| --- | --- | --- |
| **方法** | **类型** | **描述** |
| public void startDocument( ) throws SAXException | 普通 | 文档开始 |
| public void endDocument( )throws SAXException | 普通 | 文档结束 |
| public void startElement( String uri , String localName , String qname , Attributes attributes ) throws SAXException | 普通 | 元素开始，可以取得元素的名称和元素的全部属性 |
| public void endElement( String uri , String localName , String qName ) throws SAXException | 普通 | 元素结束 |
| public void characters( char[] ch , int start , int length ) throws SAXException | 普通 | 元素内容 |

（具体使用方法请参数教材书）

## DOM解析与SAX解析的区别

DOM解析和SAX解析都可以完成对XML文件的解析，但两者各有优点和缺点。

|  |  |  |
| --- | --- | --- |
| **区别点** | **DOM解析** | **SAX解析** |
| 操作 | 将XML文件读取到内存中形成DOM树，如果文件量过大，则不宜使用 | 顺序读入所需要的文件内容，不会一次性读取，不受文件大小 |
| 访问限制 | DOM树在内存中形成，可以随意存取和读取文件树的任何部分，没有次数限制 | 由于采取顺序读取，只能对文件顺序从头到尾解析一遍，不支持对文件的随意存取 |
| 修改 | 可以读取、修改和输出XML文件 | 只能读取XML文件的内容 |
| 复杂度 | 易于理解，易于开发 | 需要用于自定义解析器编写解析事件操作 |
| 对象模型 | 系统为使用者自动建立DOM树 | 对开发人员更加灵活，可以自定义SAX解析器 |

因此，DOM解析适合于对文件进行修改和随机存取的操作，但不适合大型文件。SAX 解析采用顺序读取的方式，可处理大型文件，但只能读取特定内容，由用户自定义解析 器。但在实际开发中，解析XML文件通常借助一些工具，如JDOM组件或DOM4J组件。

## 解析工具—JDOM

JDOM是使用Java语言编写的、用于读、写、操作XML文件的一套组件，可以直接从 <http://www.jdom.org/>上下载，具备了DOM解析和SAX解析的优点。（具体使用方法请 参考教材书）

## 解析工具—DOM4J

DOM4J也是一套操作XML文件的组件，主要用于读写XML文件，在框架上应用居多。 由于DOM4J性能优异、功能强大、具有易用性，所以现在已经被广泛应用开来。要想 的DOM4J的开发包，可以在<http://sourceforge.net/projects/dom4j/files/>下载。（具体使 用方法请参考教材书）