**xml内容综述**

**一、什么是 XML？**

XML 指可扩展标记语言（EXtensible Markup Language）。

XML 是一种很像HTML的标记语言。

XML 的设计宗旨是传输数据，而不是显示数据。

XML 标签没有被预定义。您需要自行定义标签。

XML 被设计为具有自我描述性。

XML 是 W3C 的推荐标准。

**二、HTML和XML区别**

HTML 旨在显示信息，而 XML 旨在传输信息。

XML大小写敏感.

XML结束标签不能省略.

XML没有结束标签，必须以 /> 形式结尾，如：<img src ="aaa.jpg"/>.

XML属性值必须用 “ ” 扩起来，且属性必须有值

**三、XML用途**

XML 应用于 Web 开发的许多方面，常用于简化数据的存储和共享。

**四、xml文件基本规范**

XML全称（Extensible Markup Language），即可扩展标记语言。主要用来传输和存储数据。XML文件具备严格的格式，一份格式良好的XML文件至少要包括以下四点：

**（1）整个XML文档有且仅有一个根元素；**

**（2）每个元素都有开始标签和结束标签，除非使用空元素；**

**（3）元素与元素之间合理嵌套；**

**（4）元素属性必须有值，且使用引号。**

遵循以上四个基本规范，XML文档就可以转换为树状结构，因此XML文档也称之为结构化文档。

熟悉Java开发的读者都清楚，Java软件开发同XML文件有着十分紧密的联系，在当前仍然比较流行的开源框架中,XML格式是Java应用的最主要的配置文件格式。如SpringMVC中的web.xml，就配置核心的servlet和filter等。

**五、文件声明**

大部分XML文档都以声明开始，XML声明向解析器提供了关于该文档的基本信息，包括version(必选)，encoding，standalone后两者是可选的。

·version：通常为1.0，表明该XML文档遵守XML1.0规范;

·encoding：指定该XML文档进行解码所使用的字符集，默认是UTF-8；

·standalone：yes|no，该文档是否引用其他资源。

一份完整的XML声明如下：

<?xml version='1.0' encoding='UTF-8' standalone='yes'>

### **六、实体引用和CDATA**

XML文档中的内容几乎可以接收任意字符，但针对特殊字符，还是需要特殊处理。

实体引用：处理XML文档中的特殊字符，预置了5个实体引用。

| **实体引用** | **所代表的符号** | **说明** |
| --- | --- | --- |
| &lt； | < | 小于符号 |
| &gt； | < | 大于符号 |
| &amp； | & | &符号 |
| &apos； | ‘ | 英文单引号 |
| &quot； | “ | 英文双引号号 |

·CDATA：实体引用只是针对单个的特殊字符；CDATA特殊标记下，所有的特殊字符，正常字符，都会当成简单字符处理。

<![CDATA[···原始的文本···]]>

**七、XML文件语义约束**

上面介绍了XML文档的基本格式，规则等；但一份XML是否是有效的，就需要定义语义约束。

所谓语义约束就是规定XML文档中允许出现哪些元素，各个元素之间的父子关系，以及每一个元素内部可以出现哪些子元素，支持哪些属性等。目前有两种主要的语义约束方式DTD和XML Schema。

为什么使用DTD:

通过 DTD，您的每一个 XML 文件均可携带一个有关其自身格式的描述。

通过 DTD，独立的团体可一致地使用某个标准的 DTD 来交换数据。

而您的应用程序也可使用某个标准的 DTD 来验证从外部接收到的数据。

您还可以使用 DTD 来验证您自身的数据。

示例：

<?xml version="1.0"?>

<!DOCTYPE note [

<!ELEMENT note (to,from,heading,body)>

<!ELEMENT to (#PCDATA)>

<!ELEMENT from (#PCDATA)>

<!ELEMENT heading (#PCDATA)>

<!ELEMENT body (#PCDATA)>

]>

<note>

<to>George</to>

<from>John</from>

<heading>Reminder</heading>

<body>Don't forget the meeting!</body>

</note>

以上 DTD 解释如下：

!DOCTYPE note (第二行)定义此文档是 note 类型的文档。

!ELEMENT note (第三行)定义 note 元素有四个元素：“to、from、heading,、body”

!ELEMENT to (第四行)定义 to 元素为 “#PCDATA” 类型

!ELEMENT from (第五行)定义 from 元素为 “#PCDATA” 类型

!ELEMENT heading (第六行)定义 heading 元素为 “#PCDATA” 类型

!ELEMENT body (第七行)定义 body 元素为 “#PCDATA” 类型

将DTD文件定义在外部：

<!DOCTYPE 根元素 SYSTEM "文件名.dtd">

**八、XML Schema**

DTD的替代者，定义XML文档合法构建模块，定义XML语义约束更加科学，更强大。

·定义可出现在文档中的元素

·定义可出现在文档中的属性

·定义哪个元素是子元素

·定义子元素的次序

·定义子元素的数目

·定义元素是否为空，或者是否可包含文本

·定义元素和属性的数据类型

·定义元素和属性的默认值以及固定值

<?xml version="1.0"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

targetNamespace="http://www.w3school.com.cn"

xmlns="http://www.w3school.com.cn"

elementFormDefault="qualified">

<xs:element name="note">

<xs:complexType>

<xs:sequence>

<xs:element name="to" type="xs:string"/>

<xs:element name="from" type="xs:string"/>

<xs:element name="heading" type="xs:string"/>

<xs:element name="body" type="xs:string"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>1

引用[****xxx.xsd****]文件：

<?xml version="1.0"?>

<note

xmlns="http://www.w3school.com.cn"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.w3school.com.cn note.xsd">

<to>George</to>

<from>John</from>

<heading>Reminder</heading>

<body>Don't forget the meeting!</body>

</note>1

**九、XML文档命名空间**

XML Namespace (xmlns)，在 XML 中，元素名称是由开发者定义的，当两个不同的文档使用相同的元素名时，就会发生命名冲突，所以就需要命名空间。

**XPath语言**

**Dom、SAX、JAXP解析XML文档**

**使用DOM4J处理XML文档**

掌握一种文档解析工具就可以了。

# 十、****XML 树结构****

**⑴XML 文档必须包含根元素。该元素是所有其他元素的父元素。**

**⑵XML 文档中的元素形成了一棵文档树。这棵树从根部开始，并扩展到树的最底端。**

**⑶所有元素均可拥有子元素。**

**⑷父、子以及同胞等术语用于描述元素之间的关系。父元素拥有子元素。相同层级上的子元素成为同胞（兄弟或姐妹）。**

**⑸所有元素均可拥有文本内容和属性（类似 HTML 中）。**

## **一个 XML 文档实例**

XML 使用简单的具有自我描述性的语法：

<?xml version="1.0" encoding="ISO-8859-1"?>

<note>

<to>George</to>

<from>John</from>

<heading>Reminder</heading>

<body>Don't forget the meeting!</body>

</note>

第一行是 XML 声明。它定义 XML 的版本 (1.0) 和所使用的编码 (ISO-8859-1 = Latin-1/西欧字符集)。

下一行描述文档的根元素（像在说：“本文档是一个便签”）：

<note>

接下来 4 行描述根的 4 个子元素（to, from, heading 以及 body）：

<to>George</to>

<from>John</from>

<heading>Reminder</heading>

<body>Don't forget the meeting!</body>

最后一行定义根元素的结尾：

</note>

从本例可以设想，该 XML 文档包含了 John 给 George 的一张便签。

XML 具有出色的自我描述性，你同意吗？