XML知识点总结

# XML基础

XML被设计用来传输和存储数据

## 1.什么是xml

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

XML 是一种标记语言，很类似 HTML

XML 的设计宗旨是传输数据，而非显示数据

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

## 2. XML 与 HTML 的主要差异

XML 不是 HTML 的替代。

XML 和 HTML 为不同的目的而设计：

XML 被设计为传输和存储数据，其焦点是数据的内容。

HTML 被设计用来显示数据，其焦点是数据的外观。

HTML 旨在显示信息，而 XML 旨在传输信息。

## 3. XML的用途

XML 把数据从 HTML 分离

XML 简化数据共享

XML 简化数据传输

XML 简化平台的变更

XML 使您的数据更有用

XML 用于创建新的 Internet 语言

## 4. XML树结构

**例如：**

<bookstore>

<book category="COOKING">

<title lang="en">Everyday Italian</title>

<author>Giada De Laurentiis</author>

<year>2005</year>

<price>30.00</price>

</book>

<book category="CHILDREN">

<title lang="en">Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

<book category="WEB">

<title lang="en">Learning XML</title>

<author>Erik T. Ray</author>

<year>2003</year>

<price>39.95</price>

</book>

</bookstore>

**其中**：

根元素是 <bookstore>。文档中的所有 <book> 元素都被包含在 <bookstore> 中。

<book> 元素有 4 个子元素：<title>、< author>、<year>、<price>

## 4. XML语法规则

**1.所有 XML 元素都须有关闭标签**

<p>This is a paragraph</p>

**2.XML 标签对大小写敏感**

<message>这是正确的。</message>

**3.XML 必须正确地嵌套**

<b><i>This text is bold and italic</i></b>

**4.XML 文档必须有根元素**

<root>

<child>

<subchild>.....</subchild>

</child>

</root>

**5.XML 的属性值须加引号**

<note date="08/08/2008">

<to>George</to>

<from>John</from>

</note>

**6.XML 中的注释**

<!-- This is a comment -->

**7.在 XML 中，空格会被保留**

## 5. XML元素

**1.什么是 XML 元素？**

XML 元素指的是从（且包括）开始标签直到（且包括）结束标签的部分。

元素可包含其他元素、文本或者两者的混合物。元素也可以拥有属性。

**例如：**

<bookstore>

<book category="CHILDREN">

<title>Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

</bookstore>

在上例中，<bookstore> 和 <book> 都拥有元素内容，因为它们包含了其他元素。<author> 只有文本内容，因为它仅包含文本。

在上例中，只有 <book> 元素拥有属性

**2.XML 命名规则**

XML 元素必须遵循以下命名规则：

名称可以含字母、数字以及其他的字符

名称不能以数字或者标点符号开始

名称不能以字符 “xml”（或者 XML、Xml）开始

名称不能包含空格

可使用任何名称，没有保留的字词。

**3.XML 元素是可扩展的**

## 6. XML属性

**1.XML 属性必须加引号**

属性值必须被引号包围，不过单引号和双引号均可使用。比如一个人的性别，person 标签可以这样写：

<person sex="female">

**2.避免 XML 属性**

因使用属性而引起的一些问题：

属性无法包含多重的值（元素可以）

属性无法描述树结构（元素可以）

属性不易扩展（为未来的变化）

属性难以阅读和维护

请尽量使用元素来描述数据。而仅仅使用属性来提供与数据无关的信息

**3.针对元数据的 XML 属性**

有时候会向元素分配 ID 引用。这些 ID 索引可用于标识 XML 元素，它起作用的方式与 HTML 中 ID 属性是一样的。

## 7. XML验证

拥有正确语法的 XML 被称为“形式良好”的 XML。

通过 DTD 验证的 XML 是“合法”的 XML。

**形式良好的 XML 文档**

“形式良好”或“结构良好”的XML 文档拥有正确的语法**。**

“形式良好”（Well Formed）的 XML 文档会遵守前几章介绍过的 XML 语法规则：

**XML 文档必须有根元素**

**XML 文档必须有关闭标签**

**XML 标签对大小写敏感**

**XML 元素必须被正确的嵌套**

**XML 属性必须加引号**

## 8. XML的特点

**1.良好的格式**

XML文件格式属于良好格式的文件，XML标记一定是成双成对的。

**2.具有验证机制**

XML的标记是程序员自己定义的，标记的定义和使用是否符合语法，需要验证。

**3.灵活的web应用**

在XML中数据和显示格式是分离设计的，XML原数据文件就是纯数据的文件，可以作为数据源，像HTML提供显示的内容，显示样式可以随HTML的变化而丰富多彩。也就是说HTML描述数据的外观，而XML描述数据本身，是文本化的小型数据库表达语言。HTML数据和显示格式混在一起，显示出一种样式。

**4.丰富的显示样式**

XML数据定义打印、显示排版信息主要有三种方式：用css定义打印和显示排版信息；用xslt转换到HTML进行显示和打印；用XSLT转换成xsl的fo进行显示和打印。

XML允许开发各种不同专业领域的标记语言

XML是非专有的，易于阅读和理解

XML与平台无关

## 9. XML浏览器

**几乎所有的主流浏览器均支持 XML 和 XSLT。**

Mozilla Firefox

从 1.0.2 版本开始，Firefox 就已开始支持 XML 和 XSLT（包括 CSS）。

Mozilla

Mozilla 含有用于 XML 解析的 Expat，并支持显示 XML + CSS。Mozilla 同时拥有对 Namespaces 的某些支持。

Mozilla 同样可做到对 XSLT 的执行（XSLT implementation）。

Netscape

自从版本 8 开始，Netscape 开始使用 Mozilla 的引擎，因此它对 XML / XSLT 的支持与 Mozilla 是相同的。

Opera

自从版本 9 开始，Opera 已经拥有对 XML / XSLT（以及 CSS）的支持。版本 8 仅支持 XML + CSS。

Internet Explorer

自从版本 6 开始，Internet Explorer 就开始支持 XML、Namespaces、CSS、XSLT 以及 XPath。

注释：Internet Explorer 5 同样拥有对 XML 的支持，但是 XSL 部分与 W3C 的官方标准不兼容！

## 10. DTD约束

**什么是DTD约束**

比如现在定义一个person的xml文件，只想要这个文件里面保存人的信息，比如name age等，但是如果在xml文件中写了一个标签<猫>，发现可以正常显示，因为符合语法规范。但是猫肯定不是人的信息，xml的标签是自定义的，需要技术来规定xml中只能出现的元素，这个时候需要约束。

**创建dtd约束**

步骤：

（1）创建一个dtd文件

（2）看xml中有多少个元素 ，有几个元素，在dtd文件中写几个 <!ELEMENT>

（3）判断元素是简单元素还是复杂元素

- 复杂元素：

<!ELEMENT 元素名称 (所有子元素)>

a: + : 表示一次或者多次

? ：表示零次或者一次

\* ：表示零次或者多次

b:子元素直接使用逗号进行隔开 ,

表示元素出现的顺序

c:子元素直接使用|隔开

表示元素只能出现其中的任意一个

- 简单元素：

<!ELEMENT 元素名称 (#PCDATA)>

a:(#PCDATA): 约束标签里面内容是字符串类型

b:EMPTY : 元素为空（没有内容）

c: ANY:任意

（3）需要在xml文件中引入dtd文件

**约束的引用**

：<!DOCTYPE 根元素名称 SYSTEM "dtd文件的路径">

：使用内部的dtd文件

<!DOCTYPE 根元素名称 [

<!ELEMENT person (name,age)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT age (#PCDATA)> ]>

(3):使用外部的dtd文件（网络上的dtd文件）

<!DOCTYPE 根元素 PUBLIC "DTD名称" "DTD文档的URL">

- 后面学到框架 struts2 使用配置文件 使用 外部的dtd 文件

**使用dtd约束标签里面属性**

语法： <!ATTLIST 元素名称

属性名称 属性类型 属性的约束>

属性类型

（1）： - CDATA: 字符串

<!ATTLIST birthday

ID1 CDATA #REQUIRED

>

（2） - 枚举 ： 表示只能在一定的范围内出现值，但是只能每次出现其中的一个

(aa|bb|cc)

<!ATTLIST age

ID2 (AA|BB|CC) #REQUIRED

>

（3） - ID: 值只能是字母或者下划线开头

<!ATTLIST name

ID3 ID #REQUIRED

>

**属性的约束**

（1）- #REQUIRED：属性必须存在

（2） - #IMPLIED：属性可有可无

（3）- #FIXED: 表示一个固定值 #FIXED "AAA"

- 属性的值必须是设置的这个固定值

<!ATTLIST sex

ID4 CDATA #FIXED "ABC"

>

（4） - 直接值

\* 不写属性，使用直接值

\* 写了属性，使用设置那个值

<!ATTLIST school

ID5 CDATA "WWW"