



LANZHOU UNIVERSITY OF TECHNOLOGY

结课作业

题 目 XML概述

学生姓名 袁晓璐

学 号 1817700253

专业班级 软件二班

学 院 软件学院

目录

[一、xml基本详解 3](#_Toc42863725)

[1、xml概述 3](#_Toc42863726)

[2、xml的优点及用途 4](#_Toc42863727)

[3、XML使用 5](#_Toc42863728)

[4、xml声明 6](#_Toc42863729)

[5.XML文档整体结构 7](#_Toc42863730)

[6、属性 7](#_Toc42863731)

[7、实体 7](#_Toc42863732)

[8、注释 8](#_Toc42863733)

[9、总结 8](#_Toc42863734)

[10、解析器 9](#_Toc42863735)

[11、命名空间 9](#_Toc42863736)

[11.1、xml命名空间（xml Namespaces） 9](#_Toc42863737)

[11.2、xml命名空间示例 9](#_Toc42863738)

[二、xml之DTD技术：xml文件的验证机制 9](#_Toc42863739)

[1、DTD概述 9](#_Toc42863740)

[2、DTD用途 10](#_Toc42863741)

[3.DTD的局限性 10](#_Toc42863742)

[4、DTD编写 10](#_Toc42863743)

[5、DTD的具体定义——元素的定义 10](#_Toc42863744)

[6、定义属性的语法结构 11](#_Toc42863745)

[7、实体 11](#_Toc42863746)

[三、xml之Schema技术 11](#_Toc42863747)

[1、Schema概述 11](#_Toc42863748)

[2、Schema特点 11](#_Toc42863749)

[3、Shema（模式） 12](#_Toc42863750)

[4、Schema文档 12](#_Toc42863751)

[5、Schema的数据类型 12](#_Toc42863752)

[6、Schema的元素类型 12](#_Toc42863753)

[四、xml之XSL技术 13](#_Toc42863754)

[1.XSL概述 13](#_Toc42863755)

[2.XSLT特点 13](#_Toc42863756)

[3. XSL语言 13](#_Toc42863757)

[五、xml之CSS技术 13](#_Toc42863758)

[1.CSS 概述 13](#_Toc42863759)

[六、XML XSLT 14](#_Toc42863760)

[七、XML架构 14](#_Toc42863761)

# 一、xml基本详解

## 1、xml概述

XML:（extensible Markup Language）即可扩展标记语言，xml是互联网数据传输的重要工具，它可以跨越互联网任何的平台，不受编程语言和操作系统的限制，它是一个拥有互联网最高级别通行证的数据携带者。xml是当前处理结构化文档信息中相当给力的技术，xml有助于在服务器之间穿梭结构化数据，这使得开发人员更加得心应手的控制数据的存储和传输。

XML用于标记电子文件使其具有结构性的标记语言，可以用来标记数据、定义数据类型，是一种允许用户对自己的标记语言进行定义的源语言。Xml是标准通用标记语言（SGML）的子集，非常适合Web传输，目前，XML 在 Web 中起到的作用不会亚于一直作为 Web 基石的 HTML。

XML提供统一的方法来描述和交换独立于应用程序或供应商的结构化数据。

总结：

<1>XML:

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

b.XML 是一种标记语言，描述事物本身（可扩展），很类似 HTML

c.XML 的设计宗旨是传输数据,而非显示数据

d.XML 标签没有被预定义。您需要自行定义标签。

e.XML 被设计为具有自我描述性。

<2>XML 与 HTML 的主要差异:

a.XML 不是 HTML 的替代。

b.XML 和 HTML 为不同的目的而设计：

c.XML 被设计为传输和存储数据,其焦点是数据的内容。而HTML 被设计用来显示数据,其焦点是数据的外观。

d.HTML 旨在显示信息,而XML旨在传输信息。

|  |  |  |
| --- | --- | --- |
| 比较内容 | HTML | XML |
| 可扩展性 | 不具有可扩展性 | 是元素标记语言，可以定义新的标记语言，标记由用户定义 |
| 侧重点 | 侧重于信息的表现形式，数据显示为什么格式为HTML所关注 | 侧重于结构化的描述信息，数据是什么为XML所关注 |
| 语法 | 不严格（嵌套、配对） | 严格要求嵌套、配对严格按照DTD的要求 |
| 可读、可维护 | 难于阅读，难于维护 | 结构清晰，便于阅读与进行维护 |
| 数据本身显示 | 数据与显示结合在一起 | 数据显示与分离 |
| 可重用 | 差 | 可重用性很高 |
| 大小写 | 不区分 | 严格区分 |
| 结束标记 | 可省略 | 不可省略 |
| 属性值 | 引号可用可不用 | 必须分装在引号中 |
| 标记 | 使用固有标记 | 没有固有标记 |

## 2、xml的优点及用途

优点：

（1）机器语言，xml与操作系统、编程语言的开发平台都无关；

（2）异质信息互通，实现不同系统之间的数据交互。

（3）简单

用途：

<1>配置应用程序和网站：在配置文件里边所有的配置文件都是以XMl的格式来编写的。

<2>数据交互：跨平台进行数据交互，它可以跨操作系统，也可以跨编程语言的平台。

<3>Ajax基石：Ajax是现在目前比较流行的一个网络交互的技术。Ajax里面最后一个x实际上就是xml的缩写。一个xml文档必须要有第一行的声明和它的文档元素的描述信息就可以了。

## 3、XML使用

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。

<1>XML 把数据从HTML分离

如果你需要在 HTML 文档中显示动态数据，那么每当数据改变时将花费大量的时间来编辑 HTML。

通过 XML，数据能够存储在独立的 XML 文件中。这样你就可以专注于使用 HTML 进行布局和显示，并确保修改底层数据不再需要对 HTML 进行任何的改变。

通过使用几行 JavaScript，你就可以读取一个外部 XML 文件，然后更新 HTML 中的数据内容。

<2>XML 简化数据共享

在真实的世界中，计算机系统和数据使用不兼容的格式来存储数据。XML 数据以纯文本格式进行存储，因此提供了一种独立于软件和硬件的数据存储方法。这让创建不同应用程序可以共享的数据变得更加容易。

<3>XML 简化数据传输

通过 XML，可以在不兼容的系统之间轻松地交换数据。

对开发人员来说，其中一项最费时的挑战一直是在因特网上的不兼容系统之间交换数据。由于可以通过各种不兼容的应用程序来读取数据，以 XML 交换数据降低了这种复杂性。

<4>XML简化平台的变更

升级到新的系统（硬件或软件平台），总是非常费时的。必须转换大量的数据，不兼容的数据经常会丢失。XML 数据以文本格式存储。这使得 XML 在不损失数据的情况下，更容易扩展或升级到新的操作系统、新应用程序或新的浏览器。

<5>XML使数据更有用

由于 XML 独立于硬件、软件以及应用程序，XML 使您的数据更可用，也更有用。不同的应用程序都能够访问您的数据，不仅仅在 HTML 页中，也可以从 XML 数据源中进行访问。

<6>XML 用于创建新的 Internet 语言

很多新的 Internet 语言是通过 XML 创建的

其中的例子包括：

XHTML - 最新的 HTML 版本

WSDL - 用于描述可用的 web service

WAP 和 WML - 用于手持设备的标记语言

RSS - 用于 RSS feed 的语言

RDF 和 OWL - 用于描述资源和本体

SMIL - 用于描述针针对 web 的多媒体

## 4、xml声明

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

<note>

<to></to>

<from></from>

<heading></heading>

<body></body>

</note>

<1>xml声明一般是xml文档的第一行；

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

<2>xml声明由以下几个部分组成：

a.根元素: 它是xml文档里面唯一的；它的开始是放在最前面，结束是放在最后面

<note>

b.子元素:

(1)所有的xml元素都必须有结束标签；

(2)xml标签对大小写敏感；

(3)xml必须正确地嵌套；

(4)元素的命名规则：

名称中可以包含字母、数字或者其他的字符；

名称不能以数字或者标点符号开始；

名称中不能包含空格。

to, from, heading,body;

注：里面唯一的；它的开始是放在最前面，结束是放在最后面。

## 5.XML文档整体结构

1. 有且只有一个根元素
2. 元素必须合理结束
3. 元素之间必须合理嵌套
4. 元素的属性必须有值

## 6、属性

（1）语法：

<元素名 属性名=“属性值”/>

例：<Student ID=“S100”>

<Name>Tom</Name>

</Student>

(2)注意：

属性值用双引号包裹；一个元素可以有多个属性，它的基本格式为：

<元素名 属性名=“属性值” 属性名=“属性值”>;

属性值中不能够直接包含<.”,&。

## 7、实体

在xml中，一些字符拥有特殊的意义。如果把字符“<”放在xml元素中，会发生错误，这是因为解析器会把它当作新元素的开始，这样会产生xml错误

为了避免这个错误，请用实体引用来代替“<”字符：

xml中5个预定义实体

|  |  |  |
| --- | --- | --- |
| &lt; | < | 小于 |
| &gt; | > | 大于 |
| &amp; | & | 和号 |
| &apos; | ‘ | 英文单引号 |
| &quot; | “ | 英文双引号 |

注释：严格地讲，在 XML 中仅有字符 "<"和"&" 是非法的。省略号、引号和大于号是合法的，但是把它们替换为实体引用是个好的习惯。

## 8、注释

XML文档还可以加入解释用的字符数据，这些解释用的字符串不会被XML解析器处理。这些解释用的文本为注释

语法：<!-- -->

注意：

1.不要把注释放在标签之内，否则，该文档不是一个格式良好的XML文档

2.不要把注释放在XML声明之前，XML声明应该永远处于XML文档的第一行

3.不要在注释中使用双中划线(--)

4.注释不能嵌套。

## 9、总结

（1）xml描述的是文档的内容与语义，而不是文档应当如何显示；

（2）格式正规（well formed）的xml文档遵循如下规则的xml文档称为格式正规的xml文档：

<1>必须有xml声明语句；

<2>必须有且仅有一个根元素；

<3>标签大小写敏感；

<4>属性值用双引号；

<5>标签成对；

<6>空标签关闭；

<7>元素正确嵌套。

(3)有效的xml文档。首先xml文档是个格式正规的xml文档，然后又需要满足DTD的要求，这样的xml文档称为有效的xml文档；

## 10、解析器

所有现代浏览器都内建了供读取和操作 XML 的 XML 解析器。

解析器把 XML 载入内存，然后把它转换为可通过 JavaScript 访问的 XML DOM 对象。

## 11、命名空间

XML 命名空间提供避免元素命名冲突的方法。

### 11.1、xml命名空间（xml Namespaces）

（1）xml命名空间提供避免元素命名冲突的方法。

（2）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。

### 11.2、xml命名空间示例

使用前缀示例

与仅仅使用前缀不同，为标签添加了一个xmlns属性，这样就为前缀赋予了一个与某个命名空间相关联的限定名称。

# 二、xml之DTD技术：xml文件的验证机制

## 1、DTD概述

文档类型定义——Document Type Definition

DTD用来描述xml文档的结构，一个DTD文档包含：

元素的定义规则；元素之间的关系规则；属性的定义规则。

## 2、DTD用途

<1>有了DTD，每个xml文件可以携带一个自身格式的描述,所谓格式描述就是我的xml文档里面可以写哪些东西，比如元素、属性；

<2>有了DTD，不同组织的人可以使用一个通用DTD用来交换数据，xml是随意定义的，有了DTD，我们可以限制多个公司之间按照这种DTD的规则来编写xml文档，由于DTD都是统一格式，所以不同的公司之间，不同的组织之间就可以用这种通用的xml文档格式进行交互数据了；

<3>应用程序可以使用一个标准DTD校检从外部接收的xml数据是否有效。

## 3.DTD的局限性

DTD不遵守xml语法（写xml文档实例时用一种语法，写DTD的时候用另外一种语法）；DTD数据类型有限（与数据库数据类型不一致）；DTD不可扩展；DTD不支持命名空间（命名冲突）。

## 4、DTD编写

DTD分成三大类，第一是内部DTD,第二是外部DTD,第三是内外结合的DTD。

1. 内部DTD文档(Students.xml)：

就是说这个xml文档里面既包含xml定义，又包含DTD的定义。这个[]里面是DTD的定义的内容，就是第一个叫内部DTD文档。

（2）外部DTD文档（Students.xml Students.dtd）:

所谓外部就是DTD的定义和我们的xml文档是在不同的文件里面。

外部的DTD的使用就需要在前面的xml文档的根元素的上面这个位置跟刚才内部的是一样的，要写上这类年改革文档在同一个文件夹下面，就可以了。

内部的DTD,它只能验证当前的这个xml文档。比如，有一个DTD需要验证多个xml文档的话，那就直接使用外部DTD就行。

（4）内外部DTD文档结合

## 5、DTD的具体定义——元素的定义

元素名称：是指xml文档里面的元素名称

元素类型：

（1）EMPTY；

（2）#PCDATA；

（3）纯元素类型的DTD语法；

（4）ANY；

## 6、定义属性的语法结构

xml文档携带数据的方式，不仅仅是xml的元素的内容，还可以是元素的属性。所以说在DTD里面我们有必要对属性进行限制。

## 7、实体

实体，简单理解就是C#中的常量，我们在DTD里面定义，在xml文档里面去使用。（实际上就是自定义实体，与前面的预定义实体差不多是一回事）。

# 三、xml之Schema技术

## 1、Schema概述

xml Schema是用一套预先规定的xml元素和属性创建的，这些元素和属性定义了xml文档的结构和内容模式。Xml Shema规定xml文档实例的结构和每个元素/属性的数据类型。对于Schema来讲，我们可以将shema比喻成为是表结构。在表结构里，定义一些数据的限制要求。然后我们xml文档呢就相当于是数据表将来要存储的数据u，也就是数据库里面的数据表数据了，所以整体来讲的话schema它就是用于验证xml文档的。

## 2、Schema特点

Schema的新特性：

Shema基于xml语法的；Shema可以用能处理xml文档的工具处理；Schema大大扩充了数据类型，可以自定义数据类型；Schema支持元素的继承——Object-Oriented;Shema支持属性组。

## 3、Shema（模式）

其作用与DTD一样，也是用于验证xml文档的有效性，只不过它提供了比DTD更强大的功能和更细粒度的数据类型，另外Schema还可以自定义数据类型。此外，Schema也是一个xml文件，而DTD则不是。

## 4、Schema文档

所有的Schema文档，其根元素必须为Schema。

## 5、Schema的数据类型

（1）内置的数据类型：

基本的数据类型；

扩展的数据类型；

（2）用户自定义简单类型（通过simpleType定义）。

## 6、Schema的元素类型

<1>Schema根元素

作用：包含已经定义的Schema

用法：<xs:schema>

属性：xmlns/targetNamespace/elementFormDefault:

<2>element元素

作用：声明一个元素

属性：name/type/ref/minOccurs/maxOccurs

<3>group元素

作用：把一组元素声明组合在一起

属性:name

# 四、xml之XSL技术

## 1.XSL概述

XSLT 是一种用于将 XML 文档转换为 XHTML 文档或其他 XML 文档的语言。展现事物表现形式

## 2.XSLT特点

<1>XSLT 指XSL转换（XSL Transformations）。

<2>XSLT 是XSL中最重要的部分。

<3>XSLT 可将一种XML文档转换为另外一种XML文档。

## 3. XSL语言

<1>XSL:不仅仅是样式表语言

<2>XSL包括三部分：

a.XSLT:一种用于转换 XML 文档的语言。

b.XPath:一种用于在 XML 文档中导航的语言。

c.XSL-FO:一种用于格式化 XML 文档的语言。

# 五、xml之CSS技术

## 1.CSS 概述

通过使用CSS,可为XML文档添加显示信息。

<1>CSS 指层叠样式表 (Cascading Style Sheets)

<2>样式定义如何显示 HTML 元素

<3>样式通常存储在样式表中

<4>把样式添加到 HTML 4.0 中，是为了解决内容与表现分离的问题

<5>外部样式表可以极大提高工作效率

<6>外部样式表通常存储在 CSS 文件中;

<7>多个样式定义可层叠为一;

# 六、XML XSLT

通过使用XSLT,您可以向XML文档添加显示信息。

XSLT是首选的XML样式表语言。

XSLT远比CSS更加完善。八使用XSLT的方法之一是在浏览器显示XML文件之前；

# 七、XML架构

<1>XML 架构介绍:

XML 架构是用于定义和验证 XML 数据的内容和结构的文档，就像数据库架构定义和验证组成数据库的表、列和数据类型一样。



<2>XML 架构优越性：

XML 架构使用 XML 语法，因此您不必学习新语法来定义数据结构。

XML 架构支持可再次使用的类型，并允许您使用继承创建新类型。

XML 架构使您可以对元素分组以控制元素和属性的重复。

有关架构以及它们所代表元素约束数据的方式的更多信息，请参见 XML 的元素、属性和类型。