[**XML详解**](https://www.cnblogs.com/stroll/p/7064503.html)

一、xml基础详解：

1、概述：

    xml: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](http://lib.csdn.net/base/operatingsystem)的限制，可以说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。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。

2、xml的特点及作用:

特点：

v xml与[操作系统](http://lib.csdn.net/base/operatingsystem)、编程语言的开发平台都无关；

v 实现不同系统之间的数据交互。

作用：

v 配置应用程序和网站；

v 数据交互；

v Ajax基石。

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。

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。

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。Xml（Extensible Markup Language）是可扩展标记语言

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声明

例子：

注：xml声明一般是xml文档的第一行；xml声明由以下几个部分组成：

4、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。

5、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。

（5）空元素

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

8、注释

注:注释内容中不要出现”--”;不要把注释放在标签中间；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：

v 必须有xml声明语句；

v 必须有且仅有一个根元素；

v 标签大小写敏感；

v 属性值用双引号；

v 标签成对；

v 空标签关闭；

v 元素正确嵌套。

(3)有效的（valid）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

11、命名空间

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

（1）使用前缀示例

与仅仅使用前缀不同，我们为标签添加了一个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、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、为什么要用DTD呀

v 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；

v 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；

v 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、如何编写一个DTD

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)：

（2）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。

（3）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。

那么外部的DTD如何使用呢？那就需要在前面的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。

有内部DTD了，为什么还要学习外部DTD呢？？

内部的DTD,它只能验证当前的这个xml文档。那假如说，我们有一个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

4、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

示例第一个是对的，第二个是错误的，因为它有内容了。

（2）#PCDATA

第一个是对的，第二个是错误的。

（3）纯元素类型的DTD语法

（4）ANY

5、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。

(1)编写属性验证的语法规则：

（2）属性类型——CDATA

CDATA是表示属性的值，可以是任何字符，这里包括数字和中文。

问题一：属性的CDATA与元素的CDATA节有何区别呢？？

属性的CDATA表示属性值是任何的字符，元素里的CDATA节是表示在CDATA节里面的内容不被xml文档进行解析。

问题二：#PCDATA与CDATA有何区别？？

#PCDATA是限制元素里面的内容是字符类型的，而这个CDATA是限制属性里面的内容是字符类型。

（3）属性类型——ID

表明该属性的取值必须是唯一的。

（4）属性类型——IDREF/IDREFS

v IDREF是指它的值要指向文档当中其它地方声明的ID。把IDREF当成[数据库](http://lib.csdn.net/base/mysql)的外键，把ID当成主键理解就行；

v IDREFS同IDREF，但是可以具有由空格分开的多个引用；

（5）属性类型——Enumerated

预先定义了一些值，属性的值必须在所列出的值的范围内

（6）属性特点——#REQUIRED

它表示元素的所有的实例都必须有该属性的值（NOT NULL）

(7)属性特点——#IMPLIED

这就表示该属性的值是可以被忽略的。

（8)属性特点——#FIXED

元素中该属性的值必须为指定的固定值。

（9）属性特点——Default

为属性提供一个默认的值。

三、实体

1、实体概述

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）。

2、实体的定义

四、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。

理解：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](http://lib.csdn.net/base/mysql)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、有了DTD，为什么要用Schema呢？

（1）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）。

（2）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、所有的Schema文档，其根元素必须为Schema。

5、Schema的文档结构

6、Schema的数据类型

6.1、Schema的数据类型概述

6.2、简单类型：

（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）。

6.3、复合类型（通过complexType定义）

6.4、数据类型的特征

7、Schema的元素类型

7.1、Schema根元素

作用：包含已经定义的Schema

用法：<xs:schema>

属性：xmlns/targetNamespace/elementFormDefault

例子：

7.2、element元素

作用：声明一个元素

属性：name/type/ref/minOccurs/maxOccurs

例子：

7.3、group元素

作用：把一组元素声明组合在一起

属性:name

例子：

7.4、attribute元素

作用：声明一个属性

属性：name/type/use/default/fixed

例子：

7.5、attributeGroup元素

作用：把一组属性声明组合在一起

属性：name/ref

例子：

7.6、simpleType元素

作用：定义一个简单类型，它决定了元素和属性值的约束和相关信息

属性：name

常用的两种方式：restriction——>一个约束；list——>从列表中选择。

（1）simpleType元素的子元素——>restriction

子元素为：<xs:restriction>——>定义一个约束条件

例子：

（2）simpleType元素的子元素——>list

子元素为：<xs:list>——>从一个特定数据类型的集合中选择定义一个简单类型元素。

例子：

7.7、complexType元素

作用：定义一个复合类型，它决定了一组元素和属性值的约束和相关信息。

属性：name

常用的两种方式：sequence——>一个序列；choice——>设置选择项。

（1）complexType元素——sequence

作用：给一组元素一个特定的序列

例子：

（2）complexType元素——choice

作用：把一组属性声明组合在一起，以便可以被复合类型应用

属性：name/ref

例子：

7.8、complexType与simpleType区别

simpleType类型的元素中不能包含元素或者属性；

当需要声明一个元素的子元素和或属性时，用complexType；

当需要基于内置的基本数据类型定义一个新的数据类型时，用simpleType。