

## 目录

1.XML 简介.....	1
2.XML 用途.....	1
3.XML 的结构.....	1
4.XML 的语法.....	1
5.XML 元素.....	2
6.XML 属性.....	2
7.XML 验证与显示.....	3
8.XML 命名空间.....	3
9.XML 解析 API.....	3

### 1.XML 简介

- XML (Extensible Markup Language): 是一种语法结构类似 HTML (细致差别仍然很大) 的标记语言, 用来记录和传输数据。XML 不同于一般的编程语言, 没有很多明确的关键字, 只是有一个大致的结构和规范, 来储存自定义的数据结构, 同时它也没有明示任何操作指令只是存储数据。而 XML 也不存在着一般编程语言似的编译过程, 如何解读 XML 语言完全看个人及程序需求。
- XML 的“关键字”即标签, 是自定义的, 即没有预定义。但是 HTML 中的标签是预定义的, 可以说 HTML 更像是一种网页编程语言。
- XML 的可扩展 extensible 性体现在 XML 的每个同级标签都是独立的, 因为只是存储数据的文本, 所以增加更多标签不会影响读取 XML 文件的程序运行, 只要保持原有的基本数据结构即可。

### 2.XML 用途

①补充 HTML 用于网页数据存储

使用 JavaScript 可以读取外部存储的 XML 文件更新网页, 而不需改动 HTML

### 3.XML 的结构

例: (注意: XML 中的注释不是//, 这样写只是为了便于记录)

```
<?xml version="1.0" encoding="ISO-8859-1"?>      //xml 文件的声明
<note date = "08/08/2008">                        //最底层的叫“根元素”
<to>George</to>                                    //里面的叫“子元素”
<from>John</from>
<heading>Reminder</heading>
<body>Don't forget the meeting!</body>
</note>
```

### 4.XML 的语法

- 所有的标签除了开头声明, 必须是闭合的 (HTML 中可以省略“结束标签”)
- XML 支持自闭标签, 即没有元素内容的空标签, 但是可以带有属性, 没有属性的自闭标签须在斜杠前和标签名后加一空格

例: <br name = “abc”/>

- XML 标签对大小写敏感
- XML 的标签必须正确嵌套，先开始的标签后结束（HTML 中允许其他情况）
- XML 必须有根元素，即包裹所有元素的元素
- XML 的属性值必须用引号包围（无论是数字还是字符），单双引号均可
- 实体引用：（XML 中不能任意使用<和&符号，会和基本语义冲突，可以使用以下字符直接代替符号）

&lt;	<	小于
&gt;	>	大于
&amp;	&	和号
&apos;	'	单引号
&quot;	"	引号

（注意：实体引用字符无需引号，直接代替符号即可，另外分号也是实体引用字符的一部分）

- XML 中多个空格不会被删节（HTML 会裁并空格）
- XML 中的注释同 HTML：

<!-- 注释内容，可任意回车换行 -->

## 5.XML 元素

- XML 元素必须遵循以下命名规则：

- ①名称可以含字母、数字以及其他的字符
- ②名称不能以数字或者标点符号开始
- ③名称不能以字符“xml”（或者 XML、Xml）开始
- ④名称不能包含空格
- ⑤最好避免使用“-”连字符和“.”点号

可使用任何名称，没有保留的字词。

- 使名称具有描述性。使用下划线的名称也很不错。

名称应当比较简短，比如：<book\_title>，而不是：<the\_title\_of\_the\_book>。

- ①避免“-”字符。如果您按照这样的方式进行命名：“first-name”，一些软件会认为您需要提取第一个单词。
- ②避免“.”字符。如果您按照这样的方式进行命名：“first.name”，一些软件会认为“name”是对象“first”的属性。
- ③避免“:”字符。冒号会被转换为命名空间来使用（稍后介绍）。

XML 文档经常有一个对应的数据库，其中的字段会对应 XML 文档中的元素。有一个实用的经验，即使用数据库的名称规则来命名 XML 文档中的元素。

## 6.XML 属性

- 属性由名称和值组成，如

<enemy type=“goblin” size=“small” speed=“15.0”/>

- 属性的值必须加引号（单引号或双引号均可），也可以用实体引用表示引号
- 设计 XML 文件时，有时候属性量完全可以用子元素量代替，并没有明显界限
- XML 中属性有很多缺点（HTML 中的属性易用很多），使用和设计时应倾向于用子元素代替。

- ①属性无法包含多重的值（元素可以）
- ②属性无法描述树结构（元素可以）
- ③属性不易扩展（为未来的变化）
- ④属性难以阅读和维护

- 最常应用 XML 属性的地方：对于同名同级的不同标签，用属性（比如 id）加以区分或索引。推广而言，元数据（metadata，用来查询某数据的数据）设计为属性，数据本身的部分存储为元素。

## 7.XML 验证与显示

- 使用 XML DTD 或 XML Schema 查验 XML 的语法规则

①<!DOCTYPE rootelement PUBLIC “...” [...]>

②

- 如果想要以非源代码形式显示 XML 文件，即类似 HTML 的浏览，可以使用 CSS 和 XSLT 两种语言来转化 XML 文件为浏览器可显示的文件。后者 XSLT 将 XML 转化为 HTML 文件。

## 8.XML 命名空间

①一般命名空间的使用举例

<h:table xmlns:h="http://www.w3.org/TR/html4/">      //用来声明 h 空间的语句

    <h:tr>

        <h:td>Apples</h:td>

        <h:td>Bananas</h:td>

    </h:tr>

</h:table>

②默认命名空间的使用举例

<table xmlns="http://www.w3.org/TR/html4/">

    <tr>

        <td>Apples</td>

        <td>Bananas</td>

    </tr>

</table>

- 使用具名的命名空间要用名称加冒号冠名所有标签
- 命名空间是一个属性，其值一般是一个网页 url，本来只是用以区分命名空间即可，实际应用中往往用有意义的网页（比如介绍该空间内容的网页）

## 9.XML 解析 API

- 解析 parse XML 文件的模式有构建 DOM（Document Object Model）方式和不构建方式 SAX（Simple API for XML）。DOM 方式下先将 XML 文件内容构建成一个树结构于内存，再来进行访问。SAX 方式不转换 XML 文件，是即时输入/输出的，因此 API 更加复杂且难用。

- API 举例：

    DOM 方式：TinyXML

    SAX 方式：eXpat