
XML简介

杨刚

什么是XML

- XML(**eXtensible Markup Language**): 可扩展的标记语言
- 可扩展的意思: 相对与我们接触的HTML(超文本标记语言), 我们在编辑网页文件时, 所有能够在网页文件中使用的HTML标签都是预先规定好的, 不能根据自己意愿添加规定外的标签。而如果编辑一个XML文件, 我们可以根据自己意愿定义标签来完成。比如:

```
<关于我>
```

```
  <姓名>张三</姓名>
```

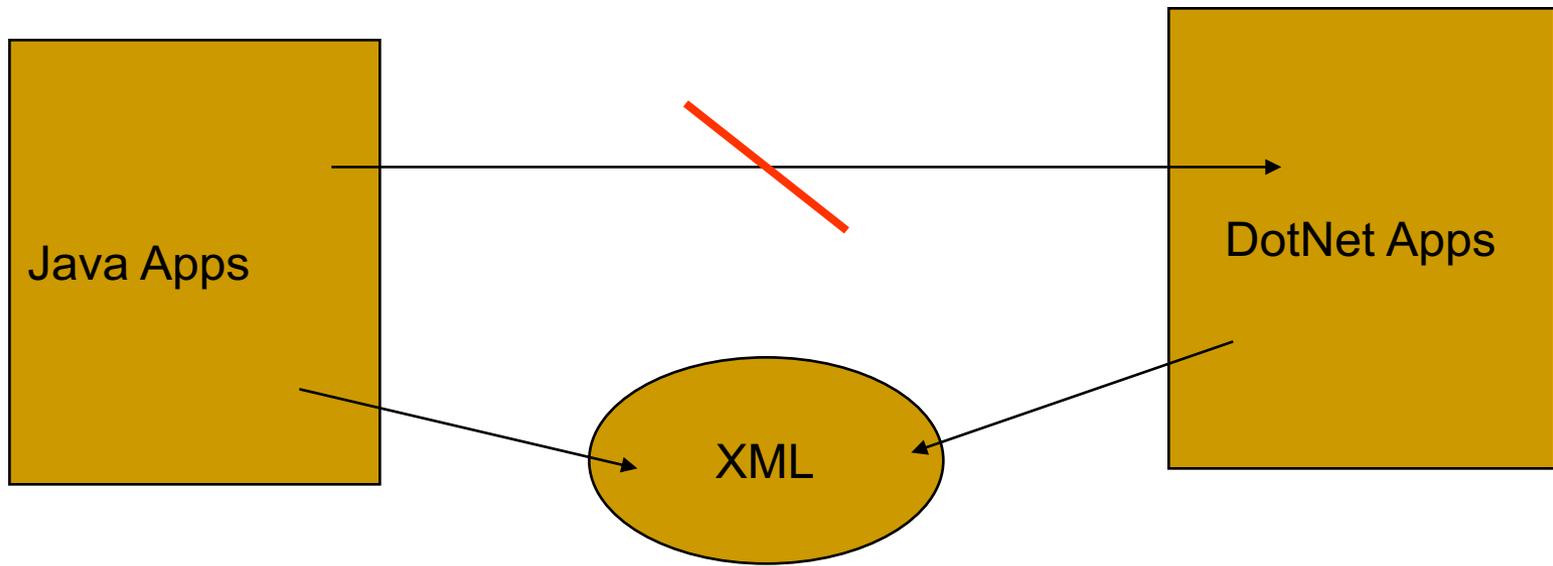
```
  <性别>男</性别>
```

```
  ...
```

```
</关于我>
```

XML特点和运用

- 可扩展性是XML最主要的特点
- 另外XML是一种元标记语言(它只表示数据本身, 没有显示的样式)
 - 和HTML比较: HTML不仅表示数据而且还有显示的样式
- 真正的数据和显示分离
- 基于这样的特点, XML主要用来交换数据
 - 用交换双方都可以明白的标签来封装数据
 - 比如WebServices, 就是在不同平台和不同语言之间交换数据
- 写配置文件



XML文档构成

一个XML文档就是扩展名为: .xml的文本文件

XML声明	{	<?xml version="1.0" encoding="utf-8"?>
文档类型声明	{	<! DOCTYPE books SYSTEM "books.dtd" [<! ENTITY bookName "XML简明教 程" >]>
文档元素	{	<books> <book> <name>Java超级编程</name> <author>比尔盖茨</author> </book> </books>

XML文档－XML声明

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

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：

version - -文档符合XML1.0规范，现在只有1.0

encoding - -文档字符编码，默认为"UTF-8"

standalone - -文档定义是否在一个文件内

standalone="yes"

standalone="no"

XML文档一根元素

- 每个XML文档必须有且只有一个根元素。
 - 根元素是一个完全包括文档中其他所有元素的元素。
 - 根元素的起始标记要放在所有其他元素的起始标记之前。
 - 根元素的结束标记要放在所有其他元素的结束标记之后。
-

XML文档—元素

元素TITLE

`<TITLE> YIKEJI.LTD.</TITLE>`

开始标记

内容

结束标记

`<TITLE/>`

空元素TITLE

`<TITLE ID="101"/>`

空元素TITLE

元素之间的关系：子元素/父元素、祖先/后代
元素内容的类型：嵌套元素/字符数据/实体引用/
CDATA节/处理指令/注释

XML文档一属性

```
<Student ID="100">  
  <Name>TOM</Name>  
</Student>
```

- 属性值用双引号 (") 或单引号 (') 分隔 (如果属性值中有', 用"分隔; 有"用'分隔)
- 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">
- 特定的属性名称在同一个元素标记中只能出现一次
- 属性值不能包括<, >, &

实体

- 作用：避免重复输入 —— (宏、变量)
- XML中五个预定义实体
- 自定义实体语法：
 <!DOCTYPE 根元素 [
 <!ENTITY 实体名 "实体内容">
]>
- 引用已定义的实体：
 &实体名;



实体	符号
<	<
>	>
&	&
"	"
'	'

XML专用标记（1） — 注释

`<!--这是一个注释-->`

- 注释内容中不要出现--;
- 不要把注释放在标记中间;
 - `<Name <!--the name-->>TOM</Name>`
- 注释不能嵌套;
- 可以在除标记以外的任何地方放注释。
- 不能位XML文档的第一行

XML专用标记（2） — 处理指令_{PI}

- 处理指令用于XML解析器传递信息到应用程序。
- 语法：<?目标 指令?>
 - PI必须以一个叫做目标的标识符开头，这个标识符遵从如同元素和属性一样的规则，目标是指令所指向的应用的名称。
 - 指令是传递给应用程序的信息。
- <?开始, ?>结束
- 样式表处理指令
 - 样式表处理指令xml-stylesheet。这个处理指令必须出现在序言部分，在根元素之前。
 - <?xml-stylesheet type="*type*" href="*uri*" ?>
 - type="text/css" 或者 type="text/xsl"

XML专用标记（3） — CDATA节

- 用于把整段文本解释为纯字符数据而不是标记的情况。包含大量<、>、&或者“字符。
CDATA(Character Data)节中的所有字符都会被当作元素字符数据的常量部分，而不是XML标记。
- 语法：

```
<![CDATA[  
    <html><head></head></html>  
]]>
```
- 可以输入任意字符（除]]>外）
- 不能嵌套

文档类型定义—DOCTYPE

- 文档类型声明，紧跟XML声明之后，包含所有实体声明

- 语法：

```
<?xml version="1.0"?>
```

```
<!DOCTYPE 根元素标记名 [
```

```
  <!--实体声明-->
```

```
]>
```

格式良好的XML文档

遵循如下规则的XML文档称为格式良好的XML文档：

- 语法规范

- 必须有XML声明语句

```
<?xml version="1.0" encoding="gb2312"?>
```

- 必须有且仅有一个根元素
- 标记大小写敏感
- 属性值用引号
- 标记成对
- 空标记关闭
- 元素正确嵌套

格式良好的XML文档(续)

■ 元素语法

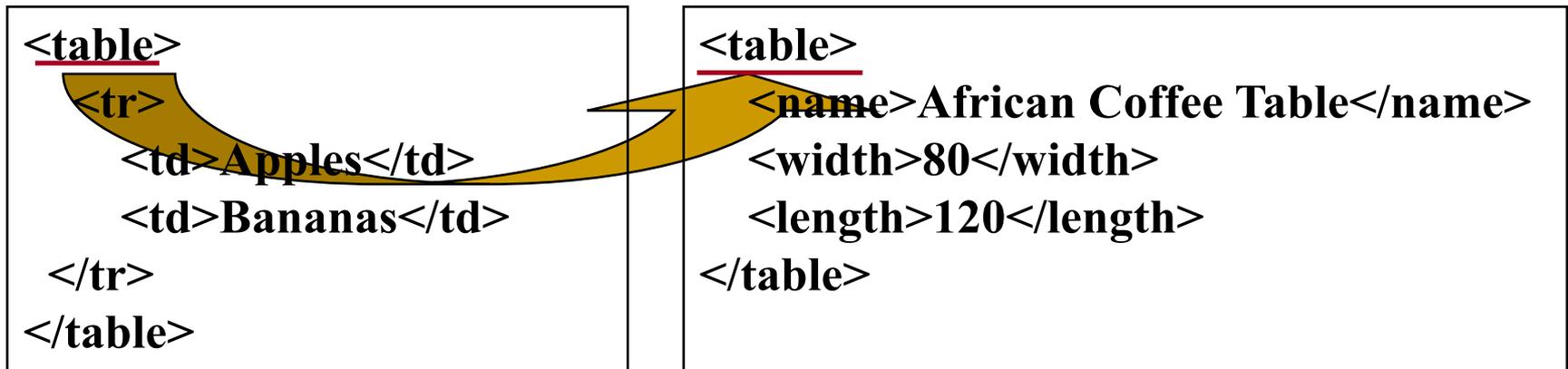
- 名称中可以包含字母、数字或者其它字符
- 名称不能以数字和"_"（下划线）开头
- 不能以XML/xml/Xml/...开头
- 名称中不能含空格
- 名称中不能含冒号(注：冒号留给命名空间使用)

命名空间 (Namespace)

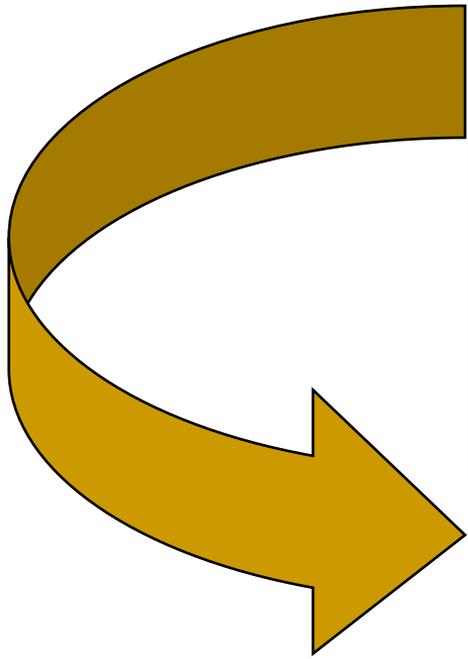
- 为什么要使用命名空间
 - 避免元素名冲突
 - 什么是命名空间
 - 使用URL作为XML的Namespaces
 - 命名空间的语法定义: `xmlns:[prefix]="URL"`
 - 元素和属性都可以应用命名空间
-

命名冲突

- XML的元素名是不固定的，当两个不同的文档使用同样的名称描述两个不同类型的元素的时候，就会发生命名冲突。



使用前綴解決命名沖突問題



```
<h:table>  
<h:tr>  
  <h:td>Apples</h:td>  
  <h:td>Bananas</h:td>  
</h:tr>  
</h:table>
```

```
<f:table>  
  <f:name>African Coffee Table</f:name>  
  <f:width>80</f:width>  
  <f:length>120</f:length>  
</f:table>
```

使用命名空间属性

```
<h:table xmlns:h="http://www.w3.org/TR/html4/">  
  <h:tr>  
    <h:td>Apples</h:td>  
    <h:td>Bananas</h:td>  
  </h:tr>  
</h:table>
```

```
<f:table xmlns:f="http://www.w3schools.com/furniture">  
  <f:name>African Coffee Table</f:name>  
  <f:width>80</f:width>  
  <f:length>120</f:length>  
</f:table>
```