



XML入门

课程介绍

◆ XML介绍与用途

◆ XML语义约束

◆ XPath路径表达式

◆ XML的语法规则

◆ Java解析XML

XML是什么

- ◆ XML的全称是EXtensible Markup Language,可扩展标记语言
- ◆ 编写XML就是编写**标签**，与HTML非常类似，扩展名.xml
- ◆ 良好的人机可读性

hr.xml

```
<employee>  
  <name>张三</name>  
  <age>31</age>  
  <height>178</height>  
</employee>
```



XML与HTML的比较

html:
<body>
 首页
</body>

xml:
<school>
 <class no="G3N1">
 <grade>三年级</grade>
 <name>1班</name>
 </class>
 <class no="G3N2">
 <grade>三年级</grade>
 <name>2班</name>
 </class>
</school>

XML的用途

web.xml - web应用配置文件

```
<web-app>
  <servlet>
    <servlet-name>InitTest</servlet-name>
    <servlet-class>moreservlets.InitServlet</servlet-class>
    <init-param>
      <param-name>param1</param-name>
      <param-value>value1</param-value>
    </init-param>
    <init-param>
      <param-name>param2</param-name>
      <param-value>2</param-value>
    </init-param>
  </servlet>
</web-app>
```

XML文档结构

- ◆ 第一行必须是XML声明。
- ◆ 有且只有一个根节点。
- ◆ XML标签的书写规则与HTML相同。

XML声明

- ◆ XML声明说明XML文档的基本信息，包括版本号与字符集，写在XML第一行。

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

version 代表版本号1.0/1.1

encoding UTF-8设置字符集，用于支持中文



我的第一份XML



XML标签书写规则

- ◆ 合法的标签名

- ◆ 合理使用属性

- ◆ 有序的子元素

- ◆ 适当的注释与缩进

- ◆ 特殊字符与CDATA标签

合法的标签名

<abc>abc</abc> ×

<考试\$>数学期末</考试\$> ×

<class><class>班级</class></class> ×

<shop-cart><item>相册</item></shop-cart> ✓

适当的注释与缩进

- ◆ 适当的注释与缩进可以让XML文档更容易阅读。

```
<employee> <name>张三  
</name> <age>31</age>  
<height>178</height> </e  
mployee>
```

```
<!-- 员工信息 -->  
<employee>  
  <name>张三</name>  
  <age>31</age>  
  <!--身高cm-->  
  <height>178</height>  
</employee>
```

合理使用属性

```
<shop-cart>
  <item sn="771938" category="电器">
    <name>XX空调</name>
    <price>2000.00</price>
    <num>1</num>
  </item>
  <item sn="890321" category="食品">
    <name>法式面包</name>
    <price>10.00</price>
    <num>5</num>
  </item>
</shop-cart>
```

处理特殊字符

- ◆ 标签体中，出现"<"、">"特殊字符，会破坏文档结构。
- ◆ 解决方案1：使用实体引用。
- ◆ 解决方案2：使用CDATA标签。

无效的XML：

```
<exam>
```

```
  <question>1+4<3是否正确？</question>
```

```
  <question>3+5>8是否正确？</question>
```

```
</exam>
```

XML支持五种实体引用

实体引用	对应符号	说明
<	<	小于
>	>	大于
&	&	和号
'	'	单引号
"	"	双引号

修改后的XML

```
<exam>
```

```
  <question>1+4&lt;3是否正确？</question>
```

```
  <question>3+5&gt;8是否正确？</question>
```

```
</exam>
```

CDATA标签

- ◆ CDATA 指的是不应由 XML 解析器进行解析的文本数据
- ◆ 从""开始，到"]]"结束</div><div data-bbox="185 515 650 935" data-label="Text"><pre><lesson>
 <content>
 本节我们来学习html中a标签的使用：
 <body>
 首页
 </body>
 </content>
</lesson></pre></div>

CDATA标签

```
<lesson>  
  <content>  
    <![CDATA[  
      本节我们来学习html中a标签的使用 :  
      <body>  
        <a href="index.html">首页</a>  
      </body>  
    ]]>  
  </content>  
</lesson>
```


有序的子元素

- ◆ 在XML多层嵌套的子元素中，标签前后顺序应保持一致。

```
<shop-cart>
  <item sn="771938" category="电器">
    <name>XX空调</name>
    <price>2000.00</price>
    <num>1</num>
  </item>
  <item sn="890321" category="食品">
    <name>法式面包</name>
    <price>10.00</price>
    <num>5</num>
  </item>
</shop-cart>
```

XML语义约束

- ◆ XML文档结构正确，但可能不是有效的。
 - 例如，员工档案XML中绝不允许出现“植物品种”标签。XML语义约束就是用于规定XML文档中允许出现哪些元素。
- ◆ XML语义约束有两种定义方式:**DTD**与**XML Schema**。

Document Type Definition

- ◆ DTD(Document Type Definition , 文档类型定义)是一种简单易用的语义约束方式。
- ◆ DTD文件的扩展名为.dtd。

hr.dtd

```
<!ELEMENT hr (employee+)>  
<!ELEMENT employee (name,age,salary,department)>  
<!ATTLIST employee no CDATA "">  
<!ELEMENT name (#PCDATA)>
```

...

DTD定义节点

- ◆ 利用DTD中的<!ELEMENT>标签，我们可以定义XML文档中允许出现的节点及数量，以hr.xml为例：

定义hr节点下只允许出现1个employee子节点。

```
<!ELEMENT hr (employee)>
```

employee节点下必须包含以下四个节点，且按顺序出现。

```
<!ELEMENT employee (name,age,salary,department)>
```

定义name标签体只能是文本，#PCDATA代表文本元素。

```
<!ELEMENT name (#PCDATA)>
```

DTD定义节点数量

- ◆ 如某个子节点需要多次重复出现，则需要在此子节点后增加相应的描述符。

hr节点下最少出现1个employee子节点。

<!ELEMENT hr (employee+)>

hr节点下可出现0..n个employee子节点。

<!ELEMENT hr (employee*)>

hr节点下最多出现1个employee子节点。

<!ELEMENT hr (employee?)>

XML引用DTD文件

- ◆ 在XML中使用<!DOCTYPE>标签来引用DTD文件。

书写格式：

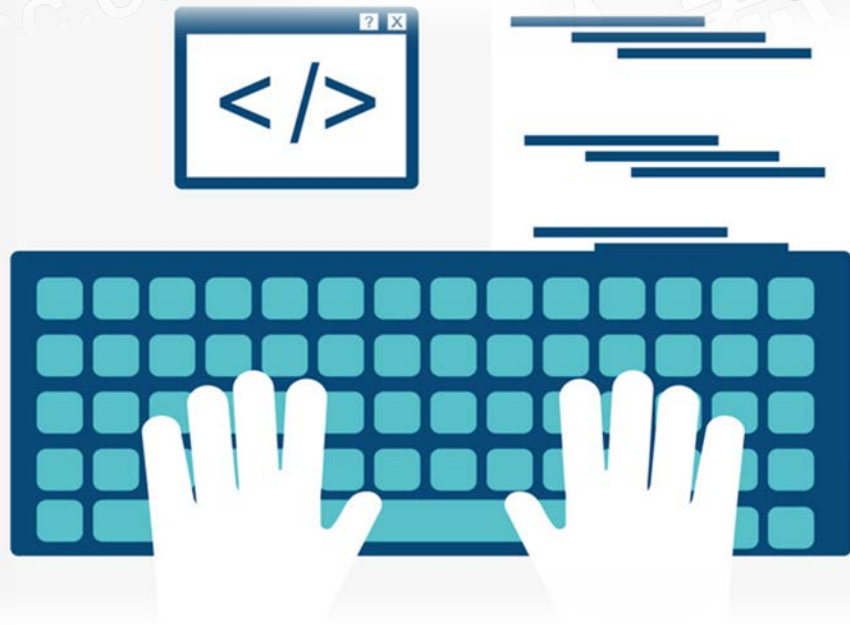
```
<!DOCTYPE 根节点 SYSTEM "dtd文件路径">
```

示例：

```
<!DOCTYPE hr SYSTEM "hr.dtd">
```



创建DTD文件

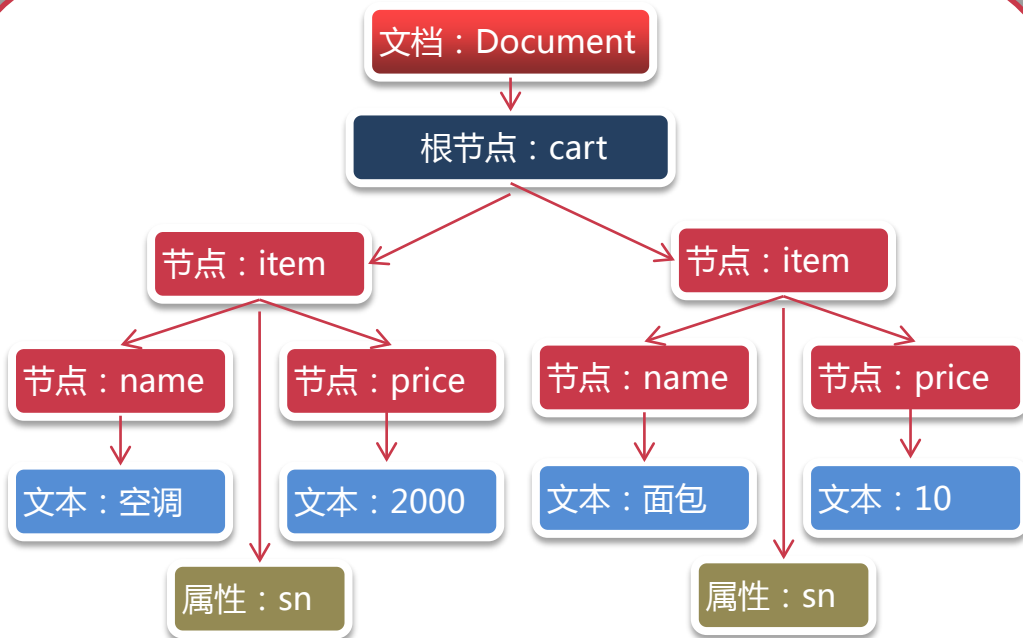


XML Schema

- ◆ XML Schema比DTD更为复杂，提供了更多功能。
- ◆ XML Schema 提供了数据类型、格式限定、数据范围等特性。
- ◆ XML Schema 是 W3C 标准。

DOM文档对象模型

```
<cart>
  <item sn="771938">
    <name>XX空调</name>
    <price>2000.00</price>
  </class>
  <item sn="890321" >
    <name>法式面包</name>
    <price>10.00</price>
  </class>
</cart>
```



Dom4j

- ◆ Dom4j是一个易用的、开源的库，用于解析XML。它应用于Java平台，具有性能优异、功能强大和极其易使用的特点。
- ◆ Dom4j将XML视为Document对象。
- ◆ XML标签被Dom4j定义为Element对象。

dom4j-2.1.1

- XML Document Object Model based on Java Collections Framework
- Java 8+
- Generics support

Released on July 1, 2018

Download 2.1.1 ▾



利用Dom4j遍历XML





利用Dom4j更新XML



XPath路径表达式

- ◆ XPath路径表达式是XML 文档中查找数据的语言。
- ◆ 掌握XPath可以极大的提高在提取数据时的开发效率。
- ◆ 学习XPath本质就是掌握各种形式表达式的使用技巧。

XPath基本表达式

◆ 最常用的基本表达式

表达式	描述
nodename	选取此节点的所有子节点。
/	从根节点选取。
//	从匹配选择的当前节点选择文档中的节点，而不考虑它们的位置。
.	选取当前节点。
..	选取当前节点的父节点。
@	选取属性。

XPath基本表达式案例

路径表达式	结果
bookstore	选取 bookstore 元素的所有子节点。
/bookstore	选取根元素 bookstore。 注释：假如路径起始于正斜杠(/)，则此路径始终代表到某元素的绝对路径！
bookstore/book	选取属于 bookstore 的子元素的所有 book 元素。
//book	选取所有 book 子元素，而不管它们在文档中的位置。
bookstore//book	选择属于 bookstore 元素的后代的所有 book 元素，而不管它们位于 bookstore 之下的什么位置。
//@lang	选取名为 lang 的所有属性。

XPath谓语表达式

路径表达式	结果
/bookstore/book[1]	选取属于 bookstore 子元素的第一个 book 元素。
/bookstore/book[last()]	选取属于 bookstore 子元素的最后一个 book 元素。
/bookstore/book[last()-1]	选取属于 bookstore 子元素的倒数第二个 book 元素。
/bookstore/book[position()<3]	选取最前面的两个属于 bookstore 元素的子元素的 book 元素。
//title[@lang]	选取所有拥有名为 lang 的属性的 title 元素。
//title[@lang='eng']	选取所有 title 元素，且这些元素拥有值为 eng 的 lang 属性。
/bookstore/book[price>35.00]	选取 bookstore 元素的所有 book 元素，且其中的 price 元素的值须大于 35.00。
/bookstore/book[price>35.00]/title	选取 bookstore 元素中的 book 元素的所有 title 元素，且其中的 price 元素的值须大于 35.00。



XPath实验室

- ◆ 利用Dom4j开发“XPath实验室”，一起见证XPath神奇之处。





Jaxen介绍

- ◆ Jaxen是一个Java编写的开源的XPath库。这是适应多种不同的对象模型，包括DOM，XOM，dom4j和JDOM。
- ◆ Dom4j底层依赖Jaxen实现XPath查询
- ◆ Jaxen 下载地址：jaxen.codehaus.org

本次课程总结

- ◆ XML文档语法规则
- ◆ DTD与XML Schema
- ◆ Dom4j与XPath表达式