## Day12\_Xml

### 0.1学习目标

1. 理解Xml用途
2. 理解xml的约束
3. 掌握jsoup解析Xml

### 0.2学习指南

* Xml概述（重要）
* Xml语法（重点）
* Xml的约束（理解）
* Xml的解析思想（理解）
* Jsoup解析Xml（重点）

### 第一堂课

本节知识点

1. 今日内容
2. xml\_概述
3. xml\_快速入门

本节目标

xml是什么,具体用在哪里

#### 主要内容概述

##### 1.1.1知识概述

内容概述

##### 1.1.2视频详情



##### 1.1.3总结与补充

无

##### 1.1.4课堂提问与练习

##### 1.1.5习题答案

##### 1.1.6视频缺陷

无

##### 1.1.7视频扩展

无

#### xml\_概述

##### 1.2.1知识概述

 xml是可扩展的标记语言,主要用途作为配置文件和网络传输数据。

##### 1.2.2视频详情



##### 1.2.3总结与补充

##### 1.2.4课堂提问与练习

* 1.什么是xml,用途是什么?

##### 1.2.5习题答案

* xml是可扩展的标记语言
* 用途 1.配置文件 2.传输数据

##### 1.2.6视频缺陷

无

##### 1.2.7视频扩展

无

#### xml\_快速入门

##### 1.3.1知识概述

通过快速入门,了解一下xml文档是怎么书写的

##### 1.3.2视频详情



##### 1.3.3总结与补充

##### 1.3.4课堂提问与练习

* 自己书写一个xml文件

##### 1.3.5习题答案

<?xml version='1.0' ?>

<users>

<user id='1'>

<name>zhangsan</name>

<age>23</age>

<gender>male</gender>

<br/>

</user>

<user id='2'>

<name>lisi</name>

<age>24</age>

<gender>female</gender>

</user>

</users>

##### 1.3.6视频缺陷

无

##### 1.3.7视频扩展

无

### 第二堂课

本节知识点：

1、xml\_组成部分

1. xml\_约束概述

本节目标 ：

xml的基本语法要记住

#### xml\_组成部分

##### 2.1.1知识概述

xml是由文档声明,指令,标签,属性和文本组成

##### 2.1.2视频详情



##### 2.1.3总结与补充

标签：标签名称自定义的规则：

\* 名称可以包含字母、数字以及其他的字符

\* 名称不能以数字或者标点符号开始

\* 名称不能以字母 xml（或者 XML、Xml 等等）开始

\* 名称不能包含空格

##### 2.1.4课堂提问与练习

##### 2.1.5习题答案

##### 2.1.6视频缺陷

无

##### 2.1.7视频扩展

无

#### xml\_约束概述

##### 2.2.1知识概述

xml约束,其实就是约束xml标签的定义的.

##### 2.2.2视频详情



##### 2.2.3总结与补充

##### 2.2.4课堂提问与练习

* 什么是xml约束

##### 2.2.5习题答案

Xml约束是用来约束Xml文件什么可以写,什么不可以写.如果没有约束,Xml里面可以随意写任何标签

##### 2.2.6视频缺陷

无

##### 2.2.7视频扩展

无

### 第三堂课

本节知识点

1. xml\_约束\_dtd
2. xml\_约束\_schema
3. xml\_解析\_解析方式

本节目标

理解xml的约束

#### xml\_约束\_dtd

##### 3.1.1知识概述

理解DTD约束的基本语法

##### 3.1.2视频详情



##### 3.1.3总结与补充

##### 3.1.4课堂提问与练习

##### 3.1.5习题答案

##### 3.1.6视频缺陷

无

##### 3.1.7视频扩展

无

#### xml\_约束\_schema

##### 3.2.1知识概述

schema约束相对于DTD来说,语法更加复杂,当然约束xml标签也更加细化.

##### 3.2.2视频详情



##### 3.2.3总结与补充

##### 3.2.4课堂提问与练习

##### 3.2.5习题答案

##### 3.2.6视频缺陷

无

##### 3.2.7视频扩展

无

#### xml\_解析\_解析方式

##### 3.3.1知识概述

xml具体的解析方式常见的是dom和sax

##### 3.3.2视频详情



##### 3.3.3总结与补充

##### 3.3.4课堂提问与练习

* Xml解析方式分别是什么,区别在哪里?

##### 3.3.5习题答案

解析xml的方式：

1. DOM：将标记语言文档一次性加载进内存，在内存中形成一颗dom树

\* 优点：操作方便，可以对文档进行CRUD的所有操作

\* 缺点：占内存

2. SAX：逐行读取，基于事件驱动的。

\* 优点：不占内存。

\* 缺点：只能读取，不能增删改

##### 3.3.6视频缺陷

无

##### 3.3.7视频扩展

无

### 第四堂课

本节知识点：

1. xml\_解析\_常见的解析器
2. xml\_解析\_Jsoup\_快速入门
3. xml\_解析\_Jsoup\_Jsoup对象

本节目标 ：

掌握jsoup常用的操作

#### xml\_解析\_常见的解析器

##### 4.1.1知识概述

xml具体解析的方式有很多,比如Dom4j,pull,jaxp,常用的jsoup的解析方式

##### 4.1.2视频详情



##### 4.1.3总结与补充

##### 4.1.4课堂提问与练习

* 常见的解析器有哪些?

##### 4.1.5习题答案

* 1. JAXP：sun公司提供的解析器，支持dom和sax两种思想
* 2. DOM4J：一款非常优秀的解析器
* 3. Jsoup：jsoup 是一款Java 的HTML解析器，可直接解析某个URL地址、HTML文本内容。它提供了一套非常省力的API，可通过DOM，CSS以及类似于jQuery的操作方法来取出和操作数据。
* 4. PULL：Android操作系统内置的解析器，sax方式的

##### 4.1.6视频缺陷

##### 4.1.7视频扩展

#### xml\_解析\_Jsoup\_快速入门

##### 4.2.1知识概述

jsoup解析xml,是第三方提供的,使用时,需要导入jar包

##### 4.2.2视频详情



##### 4.2.3总结与补充

##### 4.2.4课堂提问与练习

* 练习jsoup入门代码

##### 4.2.5习题答案

/\*\*

\* Jsoup快速入门

\*/

public class JsoupDemo1 {

public static void main(String[] args) throws IOException {

//2.获取Document对象，根据xml文档获取

//2.1获取student.xml的path

String path = JsoupDemo1.class.getClassLoader().getResource("student.xml").getPath();

//2.2解析xml文档，加载文档进内存，获取dom树--->Document

Document document = Jsoup.parse(new File(path), "utf-8");

//3.获取元素对象 Element

Elements elements = document.getElementsByTag("name");

System.out.println(elements.size());

//3.1获取第一个name的Element对象

Element element = elements.get(0);

//3.2获取数据

String name = element.text();

System.out.println(name);

}

}

##### 4.2.6视频缺陷

##### 4.2.7视频扩展

#### xml\_解析\_Jsoup\_Jsoup对象

##### 4.3.1知识概述

jsoup解析xml,提供了很多的api,常用对象Jsoup对象,里面的方法也很多.

##### 4.3.2视频详情



##### 4.3.3总结与补充

##### 4.3.4课堂提问与练习

##### 4.3.5习题答案

##### 4.3.6视频缺陷

##### 4.3.7视频扩展

无

### 第五堂课

本节知识点：

1. xml\_解析\_Jsoup\_Document对象
2. xml\_解析\_Jsoup\_Element对象

本节目标 ：

掌握Document和Element对象的常用方法

#### xml\_解析\_Jsoup\_Document对象

##### 5.1.1知识概述

通过Document对象,来操作xml标签,其实Document就对应整个xml文档

##### 5.1.2视频详情



##### 5.1.3总结与补充

##### 5.1.4课堂提问与练习

* 练习Document对象常用方法

##### 5.1.5习题答案

//1.获取student.xml的path

String path = JsoupDemo3.class.getClassLoader().getResource("student.xml").getPath();

//2.获取Document对象

Document document = Jsoup.parse(new File(path), "utf-8");

//3.获取元素对象了。

//3.1获取所有student对象

Elements elements = document.getElementsByTag("student");

System.out.println(elements);

System.out.println("-----------");

//3.2 获取属性名为id的元素对象们

Elements elements1 = document.getElementsByAttribute("id");

System.out.println(elements1);

System.out.println("-----------");

//3.2获取 number属性值为heima\_0001的元素对象

Elements elements2 = document.getElementsByAttributeValue("number", "heima\_0001");

System.out.println(elements2);

System.out.println("-----------");

//3.3获取id属性值的元素对象

Element itcast = document.getElementById("itcast");

System.out.println(itcast);

##### 5.1.6视频缺陷

##### 5.1.7视频扩展

#### xml\_解析\_Jsoup\_Element对象

##### 5.2.1知识概述

Element对象,就是标签对象,可以操作标签的属性和文本等.

##### 5.2.2视频详情



##### 5.2.3总结与补充

##### 5.2.4课堂提问与练习

* 练习Element对象常用方法

##### 5.2.5习题答案

//1.获取student.xml的path

String path = JsoupDemo4.class.getClassLoader().getResource("student.xml").getPath();

//2.获取Document对象

Document document = Jsoup.parse(new File(path), "utf-8");

/\*

Element：元素对象

1. 获取子元素对象

\* getElementById​(String id)：根据id属性值获取唯一的element对象

\* getElementsByTag​(String tagName)：根据标签名称获取元素对象集合

\* getElementsByAttribute​(String key)：根据属性名称获取元素对象集合

\* getElementsByAttributeValue​(String key, String value)：根据对应的属性名和属性值获取元素对象集合

2. 获取属性值

\* String attr(String key)：根据属性名称获取属性值

3. 获取文本内容

\* String text():获取所有字标签的纯文本内容

\* String html():获取标签体的所有内容(包括子标签的标签和文本内容)

\*/

//通过Document对象获取name标签，获取所有的name标签，可以获取到两个

Elements elements = document.getElementsByTag("name");

System.out.println(elements.size());

System.out.println("----------------");

//通过Element对象获取子标签对象

Element element\_student = document.getElementsByTag("student").get(0);

Elements ele\_name = element\_student.getElementsByTag("name");

System.out.println(ele\_name.size());

//获取student对象的属性值

String number = element\_student.attr("NUMBER");

System.out.println(number);

System.out.println("------------");

//获取文本内容

String text = ele\_name.text();

String html = ele\_name.html();

System.out.println(text);

System.out.println(html);

##### 5.2.6视频缺陷

无

##### 5.2.7视频扩展

无

### 第六堂课

本节知识点：

1. xml\_解析\_Jsoup\_根据选择器查询
2. xml\_解析\_Jsoup\_根据Xpath查询

本节目标 ：

掌握innerHTML属性的使用

#### xml\_解析\_Jsoup\_根据选择器查询

##### 6.1.1知识概述

其实jsoup的选取器 ,和前面讲过的jQuery的选择器道理是一样的,就是通过选择器快速定位标签,获取标签对象

##### 6.1.2视频详情



##### 6.1.3总结与补充

##### 6.1.4课堂提问与练习

##### 6.1.5习题答案

##### 6.1.6视频缺陷

无

##### 6.1.7视频扩展

无

#### xml\_解析\_Jsoup\_根据Xpath查询

##### 6.2.1知识概述

xpath语法就是为了支持jsoup的选择器的,使用时要注意导入对应的jar包

##### 6.2.2视频详情



##### 6.1.3总结与补充

##### 6.2.4课堂提问与练习

* 练习xml完整的解析代码

##### 6.2.5习题答案

package cn.itcast.xml.jsoup;

import cn.wanghaomiao.xpath.exception.XpathSyntaxErrorException;

import cn.wanghaomiao.xpath.model.JXDocument;

import cn.wanghaomiao.xpath.model.JXNode;

import org.jsoup.Jsoup;

import org.jsoup.nodes.Document;

import org.jsoup.select.Elements;

import java.io.File;

import java.io.IOException;

import java.util.List;

/\*\*

\*XPath查询

\*/

public class JsoupDemo6 {

public static void main(String[] args) throws IOException, XpathSyntaxErrorException {

//1.获取student.xml的path

String path = JsoupDemo6.class.getClassLoader().getResource("student.xml").getPath();

//2.获取Document对象

Document document = Jsoup.parse(new File(path), "utf-8");

//3.根据document对象，创建JXDocument对象

JXDocument jxDocument = new JXDocument(document);

//4.结合xpath语法查询

//4.1查询所有student标签

List<JXNode> jxNodes = jxDocument.selN("//student");

for (JXNode jxNode : jxNodes) {

System.out.println(jxNode);

}

System.out.println("--------------------");

//4.2查询所有student标签下的name标签

List<JXNode> jxNodes2 = jxDocument.selN("//student/name");

for (JXNode jxNode : jxNodes2) {

System.out.println(jxNode);

}

System.out.println("--------------------");

//4.3查询student标签下带有id属性的name标签

List<JXNode> jxNodes3 = jxDocument.selN("//student/name[@id]");

for (JXNode jxNode : jxNodes3) {

System.out.println(jxNode);

}

System.out.println("--------------------");

//4.4查询student标签下带有id属性的name标签 并且id属性值为itcast

List<JXNode> jxNodes4 = jxDocument.selN("//student/name[@id='itcast']");

for (JXNode jxNode : jxNodes4) {

System.out.println(jxNode);

}

}

}

##### 6.2.6视频缺陷

无

##### 6.2.7视频扩展

无