day11_XML

XML(Extensible Markup Language)：可扩展标记语言。标记语言：由标签构成的语言，可扩展：所有标签都是自定义的。xml文档是用来存储数据的，常常用来作为 "配置文件" 来存储相关的数据信息。html文档、xml文档的区别如下：

- xml中的标签都是自定义的，html标签都是预定义好的
- xml是用来存储数据的，html是用来展示数据的
- xml的语法严格，html的语法松散

xml的入门程序、基本组成部分如下：

```xml
<?xml version="1.0" ?>
<!--xml入门程序-->
<students>

    <student id="1">
        <name>zhangsan</name>
        <age>23</age>
        <gender>male</gender>
    </student>

    <student id="2">
        <name>lisi</name>
        <age>24</age>
        <sex>female</sex>
    </student>

</students>
```

1.文档声明：<?xml 属性列表 ?>

- version属性：版本号，是必需的属性（version="1.0" 为固定写法）
- encoding属性：编码方式，指定该xml文档的编码方式，不写则默认为 "UTF-8" 编码
- standalone属性：是否独立，属性值为 "yes" 则不依赖其他文件，属性值为 "no" 则依赖其他文件（了解即可，基本不使用）

#.注意：xml文档的第一行必须是"文档声明"，否则会报错（第一行是注释都不行）

2.标签：所有标签名称都是自定义的，自闭和标签、围堵标签都可以定义

- xml文档中有且仅有一个根标签（上述入门程序的根标签就是：<student></students>）
- 标签名称可以包含字母、数字以及其他符号，不能包含空格
- 标签名称不能以数字、标点符号、字母 xml、XML、Xml等开头

3.属性：属性也可以自定义，属性值必须使用引号引起来(单、双引号都可以)

4.文本：包含在标签中的文本内容

#.CDATA 区： <![CDATA[数据内容]]> 在该区域中的数据会被原样展示

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

<code>

    <!--特殊字符直接写的话，会报错-->
    if(a > b && a < c) { }

    <!--需要使用转义字符来写，但是看起来不直观-->
    if(a &gt; b &amp;&amp; a &lt; c) { }

    <!--将其放入CDATA区中，就可以原样展示了-->
    <![CDATA[ if(a > b && a < c) { } ]]>

</code>
```

一.XML约束

xml约束：规定xml文档的书写规则，一般有DTD约束、Schema约束两种方式。

1.DTD约束（简单）

(1).DTD约束的引入方式

①.内部DTD：将约束规则定义在xml文档中
<!DOCTYPE 根标签名 [约束规则]>

②.外部DTD：将 "约束规则" 定义在外部的.dtd文件中，然后将文件引入xml文档
- 引入本地.dtd文件：<!DOCTYPE 根标签名 SYSTEM "dtd文件路径">
- 引入网络.dtd文件：<!DOCTYPE 根标签名 PUBLIC "dtd文件名称" "dtd文件路径URL">

(2).DTD约束规则的编写

①.约束标签：　　<!ELEMENT 标签名称 (标签体内容)>

②.约束标签属性：<!ATTLIST 标签名称 属性名 属性类型 默认值>

2.Schema约束（复杂）

(1).Schema约束文档的引入

- 引入xsi前缀：xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
- 引入xsd文件的命名空间：xsi:schemaLocation="http://www.itcast.cn/xml .xsd文件路径"
- 为每一个xsd文件的命名空间声明一个前缀，相当于命名空间的别名：xmlns="http://www.itcast.cn/xml"

通过在根标签中定义上述3个属性，即可引入Schema约束文档。把视频再看一遍，注意理解：当引入多个.xsd约束文件时，通过为不同.xsd文件的命名空间加前缀的方式，可以自定义标签该使用哪个.xsd约束文件中的约束规则。

(2).Schema约束文档的编写：把视频再看一遍，能大概看懂.xsd文档中的Schema约束规则即可

二.XML解析

xml解析：将xml文档中存储的数据读取到内存中。xml文档的解析主要有以下两种思想：

(1).DOM思想：将xml文档一次性加载进内存，在内存中形成一颗DOM树，即：使用XML DOM来操作xml文档
- 优点：操作方便，可以对文档进行 CRUD 的操作
- 缺点：DOM树特别占内存（所以常用于服务端的xml解析）

(2).SAX思想：逐行读取xml文档中的数据到内存中，即：每次只读取一行数据到内存中，再读取下一行的时候，上一行的数据会在内存中被释放掉
- 优点：占用内存少（所以常用于移动设备端的xml解析）
- 缺点：只能读取文档中的数据，不能对文档进行增、删、改的操作

1.xml常见的解析器有以下几种：

- JAXP：sun公司提供的解析器，支持 DOM、SAX 两种方式
- DOM4J：一款常用于服务端的解析器，仅支持 DOM 方式
- PULL：Android操作系统内置的解析器，仅支持 SAX 方式
- 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。Jsoup主要是用来解析html文档的，它也可以用来解析xml文档，仅支持 DOM 方式。

2.Jsoup的基本使用

Jsoup的入门程序和使用步骤如下：

第一步：导入 "jsoup-1.11.2.jar" 包，然后再右键 "Add As Library" 将其添加成库（一般把所有的jar包放在 "lib" 目录下）
第二步：使用Jsoup工具类来获取Document文档对象（加载xml文档进内存形成对应的DOM树）
第三步：使用Document文档对象来获取Element元素对象（获取xml文档中的各个标签）
第四步：使用Element元素对象来获取标签属性、标签内容（获取xml文档中各个标签的属性、内容）

(1).Jsoup工具类

Jsoup工具类：里面含有很多静态方法，用于将指定的html、xml文档加载进内存形成一颗DOM树，并返回DOM树中的Document文档对象。

- static Document parse(File in, String charsetName)：传递html、xml文档对应的File类对象
- static Document parse(String html)：传递html、xml文档的字符串
- static Document parse(URL url, int timeoutMillis)：传递指定网络路径的html、xml文档

(2).Document文档对象

Document文档对象：使用Jsoup工具类获取到html、xml文档对应的Document对象后，就可以使用Document文档对象中的方法来获取文档中的标签元素了，即：DOM树中的Element元素对象。

- Element getElementById(String id)：根据id属性值获取唯一的Element元素对象
- Elements getElementsByTag(String tagName)：根据标签名称获取Element元素对象集合
- Elements getElementsByAttribute(String key)：根据属性名称获取Element元素对象集合
- Elements getElementsByAttributeValue(String key, String value)：根据对应的属性名和属性值获取Element元素对象集合

后面三个方法的返回值类型都是Elements类，它的定义如下：public class Elements extends ArrayList<Element>，所以Elements类是一个 "用来存储Element元素对象的ArrayList集合"，可以通过进一步操作这个ArrayList集合来得到其中的Element元素对象。

(3).Element元素对象

Element元素对象：使用Document文档对象获取到Element对象后，就可以使用Element元素对象中的方法来获取子标签对象、标签属性、标签内容。

①.获取子标签元素对象的方法

- Element getElementById(String id)：根据id属性值获取唯一的Element元素对象
- Elements getElementsByTag(String tagName)：根据标签名称获取Element元素对象集合
- Elements getElementsByAttribute(String key)：根据属性名称获取Element元素对象集合
- Elements getElementsByAttributeValue(String key, String value)：根据对应的属性名和属性值获取Element元素对象集合

②.获取属性值的方法

String attr(String key)：根据属性名称获取属性值

③.获取文本内容的方法

- String text()：获取标签的纯文本内容
- String html()：获取标签体的所有内容，包括子标签及其文本内容

(4).Node节点对象

Node节点对象：是Document文档对象、Element元素对象的父类，即：Document、Element extends Node 。所有的Document、Element对象都是一个Node节点对象。与HTML DOM中的Node节点对象类似，可以用来对DOM树进行CRUD的操作。（代码在此不再赘述）

(5).快捷查询xml标签元素的方式

①.selector选择器：Document文档对象中的select方法可以快速查询xml标签元素

Elements select(String cssQuery)：参数"cssQuery"的具体定义规则，可参考w3school

②.Xpath：即为XML路径语言，它是一种用来确定XML文档中某部分位置的语言，可以实现快速查询xml标签元素

第一步：使用Jsoup的Xpath需要额外导入 "JsoupXpath-0.3.2.jar" 包
第二步：使用Document文档对象来创建JXDocument对象
第三步：使用JXDocument对象中的 selN(String xpath) 方法来查询xml标签元素。参数"xpath"的具体定义规则，可参考w3school