

今日内容

- 1 1. XML
- 2 1. 概念
- 3 2. 语法
- 4 3. 解析

XML:

- 1 1. 概念: Extensible Markup Language 可扩展标记语言
- 2 * 可扩展: 标签都是自定义的。 <user> <student>
- 3
- 4 * 功能
- 5 * 存储数据
- 6 1. 配置文件
- 7 2. 在网络中传输
- 8 * xml与html的区别
- 9 1. xml标签都是自定义的, html标签是预定义。
- 10 2. xml的语法严格, html语法松散
- 11 3. xml是存储数据的, html是展示数据
- 12
- 13 * w3c:万维网联盟
- 14
- 15 2. 语法:
- 16 * 基本语法:
- 17 1. xml文档的后缀名 .xml
- 18 2. xml第一行必须定义为文档声明
- 19 3. xml文档中有且仅有一个根标签
- 20 4. 属性值必须使用引号(单双都可)引起来
- 21 5. 标签必须正确关闭
- 22 6. xml标签名称区分大小写
- 23 * 快速入门:
- 24 <?xml version='1.0' ?>
- 25 <users>
- 26 <user id='1'>
- 27 <name>zhangsan</name>
- 28 <age>23</age>
- 29 <gender>male</gender>
- 30

- 31 </user>
- 32
- 33 <user id='2'>
- 34 <name>lisi</name>
- 35 <age>24</age>
- 36 <gender>female</gender>
- 37 </user>
- 38 </users>
- 39
- 40 * 组成部分:
- 41 1. 文档声明
- 42 1. 格式: <?xml 属性列表 ?>
- 43 2. 属性列表:

```

44      * version: 版本号, 必须属性
45      * encoding: 编码方式。告知解析引擎当前文档使用的字符集, 默认值:
ISO-8859-1
46      * standalone: 是否独立
47      * 取值:
48      * yes: 不依赖其他文件
49      * no: 依赖其他文件
50      2. 指令(了解): 结合css的
51      * <?xml-stylesheet type="text/css" href="a.css" ?>
52      3. 标签: 标签名称自定义的
53      * 规则:
54      * 名称可以包含字母、数字以及其他的字符
55      * 名称不能以数字或者标点符号开始
56      * 名称不能以字母 xml (或者 XML、Xml 等等) 开始
57      * 名称不能包含空格
58
59      4. 属性:
60      id属性值唯一
61      5. 文本:
62      * CDATA区: 在该区域中的数据会被原样展示
63      * 格式: <![CDATA[ 数据 ]]>

```

```

1      * 约束: 规定xml文档的书写规则
2      * 作为框架的使用者(程序员):
3      1. 能够在xml中引入约束文档
4      2. 能够简单的读懂约束文档

```

```

1      * 分类:
2      1. DTD: 一种简单的约束技术
3      2. Schema: 一种复杂的约束技术

```

```

1      * DTD:
2      * 引入dtd文档到xml文档中
3      * 内部dtd: 将约束规则定义在xml文档中
4      * 外部dtd: 将约束的规则定义在外部的dtd文件中
5      * 本地: <!DOCTYPE 根标签名 SYSTEM "dtd文件的位置">
6      * 网络: <!DOCTYPE 根标签名 PUBLIC "dtd文件名字" "dtd文件的位
置URL">

```

```

1      * Schema:
2      * 引入:
3      1. 填写xml文档的根元素
4      2. 引入xsi前缀.
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5      3. 引入xsd文件命名空间.
xsi:schemaLocation="http://www.itcast.cn/xml student.xsd"
6      4. 为每一个xsd约束声明一个前缀, 作为标识
xmlns="http://www.itcast.cn/xml"
7
8      <students xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
9      xmlns="http://www.itcast.cn/xml"
10     xsi:schemaLocation="http://www.itcast.cn/xml
student.xsd">

```

```

1  3. 解析：操作xml文档，将文档中的数据读取到内存中
2      * 操作xml文档
3          1. 解析(读取)：将文档中的数据读取到内存中
4          2. 写入：将内存中的数据保存到xml文档中。持久化的存储
5
6      * 解析xml的方式：
7          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
8              * 优点：操作方便，可以对文档进行CRUD的所有操作
9              * 缺点：占内存
10         2. SAX：逐行读取，基于事件驱动的。
11             * 优点：不占内存。
12             * 缺点：只能读取，不能增删改

```

* xml常见的解析器：

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。

```

1      * 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。
2      * 快速入门：
3          * 步骤：
4              1. 导入jar包
5              2. 获取Document对象
6              3. 获取对应的标签Element对象
7              4. 获取数据
8
9      * 代码：
10         //2.1获取student.xml的path
11         String path =
JsoupDemo1.class.getClassLoader().getResource("student.xml").getPath();
12         //2.2解析xml文档，加载文档进内存，获取dom树--->Document
13         Document document = Jsoup.parse(new File(path), "utf-8");
14         //3.获取元素对象 Element
15         Elements elements = document.getElementsByTag("name");
16
17         System.out.println(elements.size());
18         //3.1获取第一个name的Element对象
19         Element element = elements.get(0);
20         //3.2获取数据
21         String name = element.text();
22         System.out.println(name);
23
24     * 对象的使用：
25         1. Jsoup：工具类，可以解析html或xml文档，返回Document
26             * parse：解析html或xml文档，返回Document
27                 * parse•(File in, String charsetName)：解析xml或html文件的。
28                 * parse•(String html)：解析xml或html字符串
29                 * parse•(URL url, int timeoutMillis)：通过网络路径获取指定的
    html或xml的文档对象

```

```

30      2. Document: 文档对象。代表内存中的dom树
31      * 获取Element对象
32      * getElementById.(String id): 根据id属性值获取唯一的element对
象
33      * getElementsByTagName.(String tagName): 根据标签名称获取元素对象
集合
34      * getElementsByAttribute.(String key): 根据属性名称获取元素对象
集合
35      * 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
36      3. Elements: 元素Element对象的集合。可以当做 ArrayList<Element>来使用
37      4. Element: 元素对象
38      1. 获取子元素对象
39      * getElementById.(String id): 根据id属性值获取唯一的element对
象
40      * getElementsByTagName.(String tagName): 根据标签名称获取元素对象
集合
41      * getElementsByAttribute.(String key): 根据属性名称获取元素对象
集合
42      * 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
43
44      2. 获取属性值
45      * String attr(String key): 根据属性名称获取属性值
46      3. 获取文本内容
47      * String text(): 获取文本内容
48      * String html(): 获取标签体的所有内容(包括子标签的字符串内容)
49      5. Node: 节点对象
50      * 是Document和Element的父类

```

```

1      * 快捷查询方式:
2      1. selector: 选择器
3      * 使用的方法: Elements select.(String cssQuery)
4      * 语法: 参考Selector类中定义的语法
5      2. XPath: XPath即为XML路径语言, 它是一种用来确定XML (标准通用标记语言的子
集) 文档中某部分位置的语言
6      * 使用Jsoup的XPath需要额外导入jar包。
7      * 查询w3cshool参考手册, 使用xpath的语法完成查询
8      * 代码:
9      //1. 获取student.xml的path
10     String path =
JsoupDemo6.class.getClassLoader().getResource("student.xml").getPath();
11     //2. 获取Document对象
12     Document document = Jsoup.parse(new File(path), "utf-8");
13
14     //3. 根据document对象, 创建JXDocument对象
15     JXDocument jxDocument = new JXDocument(document);
16
17     //4. 结合xpath语法查询
18     //4.1 查询所有student标签
19     List<JXNode> jxNodes = jxDocument.selN("//student");
20     for (JXNode jxNode : jxNodes) {
21         System.out.println(jxNode);
22     }
23

```

```
24         System.out.println("-----");
25
26         //4.2查询所有student标签下的name标签
27         List<JXNode> jxNodes2 =
jxDocument.selN("//student/name");
28         for (JXNode jxNode : jxNodes2) {
29             System.out.println(jxNode);
30         }
31
32         System.out.println("-----");
33
34         //4.3查询student标签下带有id属性的name标签
35         List<JXNode> jxNodes3 =
jxDocument.selN("//student/name[@id]");
36         for (JXNode jxNode : jxNodes3) {
37             System.out.println(jxNode);
38         }
39         System.out.println("-----");
40         //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
41
42         List<JXNode> jxNodes4 =
jxDocument.selN("//student/name[@id='itcast']");
43         for (JXNode jxNode : jxNodes4) {
44             System.out.println(jxNode);
45         }
```