

第 8 章 JSTL 标签

8.1 配置 jar 文件

1. 下载 jar 文件

(1) jstl.jar (2) standard.jar

2. 安装 jar 文件

复制到/WebContent/WEB-INF/lib/目录下

8.2 测试 jar 文件是否配置成功

1. 添加 taglib 指令标记

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

2. 使用 JSTL 输出语句

```
<c:out value="测试 JSTL 安装与配置" />
```

8.3 JSTL 标签主要分类

1. 核心标签库（前缀：c）

2. 国际化标签库（前缀：fmt）

3. 函数标签库（前缀：\${fn:函数名()})

4. SQL 数据库标签库（前缀：sql）

5. XML 标签库（前缀：xml）

8.4 JSTL 核心标签

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

1. <c:out>

(1) 作用：在页面中输出结果

(2) 基本语法

```
<c:out value="输出值" />
```

2. <c:set>

(1) 作用：在一个作用域内（request、session、application）设置某个变量的值

(2) 基本语法

```
<c:set var="变量名" value="变量值" [scope="范围"] />
```

(3) 示例:

```
<c:set var="name" value="张三" />
```

【复习】

3.<c:remove>

(1) 作用: 删除变量

(2) 基本语法

```
<c:remove var="变量名" [scope="范围"] />
```

(3) 示例: 设置各范围变量和删除变量实例

```
<c:remove var="name" />
```

```
<c:set var="number" value="4" scope="request" />
<c:set var="number" value="5" scope="session" />
<c:set var="number" value="6" scope="application" />
<h1>
  <c:out value="${requestScope.number}" />
  <c:out value="${sessionScope.number}" />
  <c:out value="${applicationScope.number}" />
</h1>
<c:remove var="number"/>
<c:out value="${requestScope.number}" />
<c:out value="${sessionScope.number}" />
<c:out value="${applicationScope.number}" />
```

4.<c:if>

(1) 作用: 条件判断

(2) 基本语法

```
<c:if test="判断条件">输出内容</c:if>
```

(3) 示例:

```
<c:set var="age" value="100" />
```

```
<c:if test="${age eq 100}"> 年龄是 100 岁 </c:if>
```

5.<c:choose>

(1) 作用：流程控制选择，条件判断（类似 switch）

(2) 基本语法

```
<c:choose>
    多个<c:when></c:when>
    <c:otherwise></c:otherwise>
</c:choose>
```

6.<c:when>

(1) 作用：<c:choose>标签的子标签，分支标签（类似 case）

(2) 基本语法

```
<c:when test="判断条件">输出内容</c:when>
```

7.<c:otherwise>

(1) 作用：<c:choose>标签的子标签，默认分支标签（类似 default）

(2) 基本语法

```
<c:when test="判断条件">输出内容</c:when>
```

(3) 流程控制标签实例

```
<c:set var="weekday" value="Monday" />
<c:choose>
    <c:when test="${weekday == 'Monday'}">今天是星期一</c:when>
    <c:when test="${weekday == 'Tuesday'}">今天是星期二</c:when>
    <c:when test="${weekday == 'Wednesday'}">今天是星期三</c:when>
    <c:when test="${weekday == 'Thursday'}">今天是星期四</c:when>
    <c:when test="${weekday == 'Friday'}">今天是星期五</c:when>
    <c:when test="${weekday == 'Saturday'}">今天是星期六</c:when>
    <c:otherwise>今天是星期天</c:otherwise>
</c:choose>
```

8.<c:forEach>

(1) 作用：循环（迭代）标签，用于循环控制并可以将集合中的成员循环浏览一遍。

(2) 基本语法

```
<c:forEach items="${集合名}" var="变量名" begin="初始值" end="结束值">
```

输出内容（\${变量名}）

```
</c:forEach>
```

（3）示例 1：简单循环控制

```
<c:forEach var="i" begin="0" end="9">${i } </c:forEach>
```

（4）示例 2：集合循环控制

```
<%
    String Characters[] = { "A", "B", "C", "D", "E", "F", "G" };
    request.setAttribute("charArrays", Characters);
%>
<c:forEach items="${charArrays}" var="i"> ${i }&nbsp;</c:forEach>
```

9.<c:forTokens>

（1）作用：循环（迭代）标签，用于循环控制并可以按照分隔符将字符串分割。

（2）基本语法

```
<c:forTokens items="${集合名}" var="变量名" delims="分隔符">
```

输出内容（\${变量名}）

```
</c:forTokens>
```

（3）示例：循环控制

```
<%
    String names = "Jhon-Brown-Dell";
    request.setAttribute("names", names);
%>
<c:forTokens items="${names}" delims="-" var="i">${i }&nbsp;</c:forTokens>
```

8.5 JSTL 函数标签

0.导入库

```
<%@ taglib uri="" prefix="c"%>
```

```
<%@ taglib uri="" prefix="fn"%>
```

1.\${fn:contains(String,substring)}

（1）函数说明：判断源字符串中是否包含指定的子串

(2) 基本语法:

`${fn:contains(源字符串, 子字符串)}`

(3) 返回值: true/false

(4) 示例:

```
<c:set var="theString" value="I am a test String" />
<c:if test="${fn:contains(theString, 'test')}">
    <p>Found test string</p>
</c:if>
```

【复习】

2.`${fn:indexOf(string,substring)}`

(1) 函数说明: 子字符串在源字符串中第一次出现的位置下标 (索引), 从零开始。

(2) 基本语法:

`${fn:indexOf(源字符串, 子字符串)}`

(3) 返回值: 数字

(4) 示例:

```
<c:set var="string1" value="ABCDEFGH" />
<p>Index (1) : ${fn:indexOf(string1, "DEF")}</p>
```

3.`${fn:length(string)}`

(1) 函数说明: 返回字符串长度

(2) 基本语法:

`${fn:length(字符串)}`

(3) 返回值: 数字

(4) 示例:

```
<c:set var="string1" value="1234567890"/>
<p>"${string1}"的长度是: ${fn:length(string1)}</p>
```

4.`${fn:split(string,delimiters)}`

(1) 函数说明: 根据“分隔符字符串”将一个字符串分割成一个数组字符串

(2) 基本语法:

`${fn:split(源字符串, 分隔符字符串)}`

(3) 返回值: 字符串数组

(4) 示例:

```
<c:set var="s1" value="John-Brown-Dell" />
<c:set var="s2" value="${fn:split(s1, '-')}" />
<c:forEach items="${ s2}" var="str">
    ${str }&nbsp;
</c:forEach>
```