JSTL

1. JSP标准标签库（JSTL）是一个JSP标签集合，它**封装**了JSP应用的通用核心功能。

JSTL支持通用的、结构化的任务，比如迭代，条件判断，XML文档操作，国际化标签，SQL标签。 除了这些，它还提供了一个框架来使用集成JSTL的自定义标签。

根据JSTL标签所提供的功能，可以将其分为5个类别。

* **核心标签**
* **格式化标签**
* **SQL 标签**
* **XML 标签**
* **JSTL 函数**

## 2. JSTL 库安装

Apache Tomcat安装JSTL 库步骤如下：

从Apache的标准标签库中下载的二进包(jakarta-taglibs-standard-current.zip)。

* 官方下载地址：<http://archive.apache.org/dist/jakarta/taglibs/standard/binaries/>
* 本站下载地址：[jakarta-taglibs-standard-1.1.2.zip](http://static.runoob.com/download/jakarta-taglibs-standard-1.1.2.tar.gz)

下载jakarta-taglibs-standard-1.1.2.zip 包并解压，将jakarta-taglibs-standard-1.1.2/lib/下的两个jar文件：standard.jar和jstl.jar文件拷贝到/WEB-INF/lib/下。

接下来我们在 web.xml 文件中添加以下配置：

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

<web-app version="2.4"

xmlns="http://java.sun.com/xml/ns/j2ee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee

http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd">

<jsp-config>

<taglib>

<taglib-uri>http://java.sun.com/jstl/fmt</taglib-uri>

<taglib-location>/WEB-INF/fmt.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>http://java.sun.com/jstl/fmt-rt</taglib-uri>

<taglib-location>/WEB-INF/fmt-rt.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>http://java.sun.com/jstl/core</taglib-uri>

<taglib-location>/WEB-INF/c.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>http://java.sun.com/jstl/core-rt</taglib-uri>

<taglib-location>/WEB-INF/c-rt.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>http://java.sun.com/jstl/sql</taglib-uri>

<taglib-location>/WEB-INF/sql.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>http://java.sun.com/jstl/sql-rt</taglib-uri>

<taglib-location>/WEB-INF/sql-rt.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>http://java.sun.com/jstl/x</taglib-uri>

<taglib-location>/WEB-INF/x.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>http://java.sun.com/jstl/x-rt</taglib-uri>

<taglib-location>/WEB-INF/x-rt.tld</taglib-location>

</taglib>

</jsp-config>

</web-app>

**使用任何库，你必须在每个JSP文件中的头部包含<taglib>标签。**

## 3. 核心标签 core <c:xxx>

核心标签是最常用的JSTL标签。引用核心标签库的语法如下：

<%@ taglib prefix="c"

uri="http://java.sun.com/jsp/jstl/core" %>

|  |  |
| --- | --- |
| **标签** | **描述** |
| [<c:out>](http://www.runoob.com/jsp/jstl-core-out-tag.html) | 用于在JSP中**显示数据**，就像<%= ... > |
| [<c:set>](http://www.runoob.com/jsp/jstl-core-set-tag.html) | 用于**保存数据** |
| [<c:remove>](http://www.runoob.com/jsp/jstl-core-remove-tag.html) | 用于**删除数据** |
| [<c:catch>](http://www.runoob.com/jsp/jstl-core-catch-tag.html) | 用来处**理产生错误的异常状况**，并且将错误信息储存起来 |
| [<c:if>](http://www.runoob.com/jsp/jstl-core-if-tag.html) | 与我们在**一般程序中用的if一样** |
| [<c:choose>](http://www.runoob.com/jsp/jstl-core-choose-tag.html) | 本身只当做<c:when>和<c:otherwise>的父标签 |
| [<c:when>](http://www.runoob.com/jsp/jstl-core-choose-tag.html) | <c:choose>的子标签，用来判断条件是否成立 |
| [<c:otherwise>](http://www.runoob.com/jsp/jstl-core-choose-tag.html) | <c:choose>的子标签，接在<c:when>标签后，当<c:when>标签判断为false时被执行 |
| [<c:import>](http://www.runoob.com/jsp/jstl-core-import-tag.html) | 检索一个绝对或相对 URL，然后将其内容暴露给页面 |
| [<c:forEach>](http://www.runoob.com/jsp/jstl-core-foreach-tag.html) | 基础迭代标签，接受多种集合类型 |
| [<c:forTokens>](http://www.runoob.com/jsp/jstl-core-foreach-tag.html) | 根据指定的分隔符来分隔内容并迭代输出 |
| [<c:param>](http://www.runoob.com/jsp/jstl-core-param-tag.html) | 用来给包含或重定向的页面传递参数 |
| [<c:redirect>](http://www.runoob.com/jsp/jstl-core-redirect-tag.html) | 重定向至一个新的URL. |
| [<c:url>](http://www.runoob.com/jsp/jstl-core-url-tag.html) | 使用可选的查询参数来创造一个URL |

1）<c:out>

<c:out>标签有如下属性：

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **描述** | **是否必要** | **默认值** |
| value | 要输出的内容 | 是 | 无 |
| default | 输出的默认值 | 否 | 主体中的内容 |
| escapeXml | 是否忽略XML特殊字符 | 否 | true |

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

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

<html>

<head>

<title>c:out 标签实例</title>

</head>

<body>

<html>

<head>

<title>&lt;c:out&gt;实例</title>

</head>

<body>

<h1>&lt;c:out&gt; 实例</h1>

<c:out value="&lt要显示的数据对象（未使用转义字符）&gt" escapeXml="true" default="默认值"></c:out><br/>

<c:out value="&lt要显示的数据对象（使用转义字符）&gt" escapeXml="false" default="默认值"></c:out><br/>

<c:out value="${null}" escapeXml="false">使用的表达式结果为null，则输出该默认值</c:out><br/>

</body>

</body>

</html>

运行结果如下：

<c:out> 实例

&lt要显示的数据对象（未使用转义字符）&gt

<要显示的数据对象（使用转义字符）>

使用的表达式结果为null，则输出该默认值

2）<c:set>

<c:set>标签有如下属性：

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **描述** | **是否必要** | **默认值** |
| value | 要存储的值 | 否 | 主体的内容 |
| target | 要修改的属性所属的对象 | 否 | 无 |
| property | 要修改的属性 | 否 | 无 |
| var | 存储信息的变量 | 否 | 无 |
| scope | var属性的作用域 | 否 | Page |

如果指定了target属性，那么property属性也需要被指定。

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

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

<html>

<head>

<title>c:set 标签实例</title>

</head>

<body>

<c:set var="salary" scope="session" value="${2000\*2}"/>

<c:out value="${salary}"/>

</body>

</html>

运行结果如下：

4000

3）<c:>

**4.格式化标签**

JSTL格式化标签用来格式化并输出文本、日期、时间、数字。引用格式化标签库的语法如下：

<%@ taglib prefix="fmt"

uri="http://java.sun.com/jsp/jstl/fmt" %>

|  |  |
| --- | --- |
| **标签** | **描述** |
| [<fmt:formatNumber>](http://www.runoob.com/jsp/jstl-format-formatnumber-tag.html) | 使用指定的格式或精度格式化数字 |
| [<fmt:parseNumber>](http://www.runoob.com/jsp/jstl-format-parsenumber-tag.html) | 解析一个代表着数字，货币或百分比的字符串 |
| [<fmt:formatDate>](http://www.runoob.com/jsp/jstl-format-formatdate-tag.html) | 使用指定的风格或模式格式化日期和时间 |
| [<fmt:parseDate>](http://www.runoob.com/jsp/jstl-format-parsedate-tag.html) | 解析一个代表着日期或时间的字符串 |
| [<fmt:bundle>](http://www.runoob.com/jsp/jstl-format-bundle-tag.html) | 绑定资源 |
| [<fmt:setLocale>](http://www.runoob.com/jsp/jstl-format-setlocale-tag.html) | 指定地区 |
| [<fmt:setBundle>](http://www.runoob.com/jsp/jstl-format-setbundle-tag.html) | 绑定资源 |
| [<fmt:timeZone>](http://www.runoob.com/jsp/jstl-format-timezone-tag.html) | 指定时区 |
| [<fmt:setTimeZone>](http://www.runoob.com/jsp/jstl-format-settimezone-tag.html) | 指定时区 |
| [<fmt:message>](http://www.runoob.com/jsp/jstl-format-message-tag.html) | 显示资源配置文件信息 |
| [<fmt:requestEncoding>](http://www.runoob.com/jsp/jstl-format-requestencoding-tag.html) | 设置request的字符编码 |

**5.SQL标签**

JSTL SQL标签库提供了与关系型数据库（Oracle，MySQL，SQL Server等等）进行交互的标签。引用SQL标签库的语法如下：

<%@ taglib prefix="sql"

uri="http://java.sun.com/jsp/jstl/sql" %>

|  |  |
| --- | --- |
| **标签** | **描述** |
| [<sql:setDataSource>](http://www.runoob.com/jsp/jstl-sql-setdatasource-tag.html) | 指定数据源 |
| [<sql:query>](http://www.runoob.com/jsp/jstl-sql-query-tag.html) | 运行SQL查询语句 |
| [<sql:update>](http://www.runoob.com/jsp/jstl-sql-update-tag.html) | 运行SQL更新语句 |
| [<sql:param>](http://www.runoob.com/jsp/jstl-sql-param-tag.html) | 将SQL语句中的参数设为指定值 |
| [<sql:dateParam>](http://www.runoob.com/jsp/jstl-sql-dateparam-tag.html) | 将SQL语句中的日期参数设为指定的java.util.Date 对象值 |
| [<sql:transaction>](http://www.runoob.com/jsp/jstl-sql-transaction-tag.html) | 在共享数据库连接中提供嵌套的数据库行为元素，将所有语句以一个事务的形式来运行 |

**6.XML 标签**

JSTL XML标签库提供了创建和操作XML文档的标签。引用XML标签库的语法如下：

<%@ taglib prefix="x"

uri="http://java.sun.com/jsp/jstl/xml" %>

在使用xml标签前，你必须将XML 和 XPath 的相关包拷贝至你的<Tomcat 安装目录>\lib下:

* **XercesImpl.jar**

下载地址： <http://www.apache.org/dist/xerces/j/>

* **xalan.jar**

下载地址： <http://xml.apache.org/xalan-j/index.html>

|  |  |
| --- | --- |
| **标签** | **描述** |
| [<x:out>](http://www.runoob.com/jsp/jstl-xml-out-tag.html) | 与<%= ... >,类似，不过只用于XPath表达式 |
| [<x:parse>](http://www.runoob.com/jsp/jstl-xml-parse-tag.html) | 解析 XML 数据 |
| [<x:set>](http://www.runoob.com/jsp/jstl-xml-set-tag.html) | 设置XPath表达式 |
| [<x:if>](http://www.runoob.com/jsp/jstl-xml-if-tag.html) | 判断XPath表达式，若为真，则执行本体中的内容，否则跳过本体 |
| [<x:forEach>](http://www.runoob.com/jsp/jstl-xml-foreach-tag.html) | 迭代XML文档中的节点 |
| [<x:choose>](http://www.runoob.com/jsp/jstl-xml-choose-tag.html) | <x:when>和<x:otherwise>的父标签 |
| [<x:when>](http://www.runoob.com/jsp/jstl-xml-choose-tag.html) | <x:choose>的子标签，用来进行条件判断 |
| [<x:otherwise>](http://www.runoob.com/jsp/jstl-xml-choose-tag.html) | <x:choose>的子标签，当<x:when>判断为false时被执行 |
| [<x:transform>](http://www.runoob.com/jsp/jstl-xml-transform-tag.html) | 将XSL转换应用在XML文档中 |
| [<x:param>](http://www.runoob.com/jsp/jstl-xml-param-tag.html) | 与<x:transform>共同使用，用于设置XSL样式表 |

**7.JSTL函数**

JSTL包含一系列标准函数，大部分是通用的字符串处理函数。引用JSTL函数库的语法如下：

<%@ taglib prefix="fn"

uri="http://java.sun.com/jsp/jstl/functions" %>

|  |  |
| --- | --- |
| **函数** | **描述** |
| [fn:contains()](http://www.runoob.com/jsp/jstl-function-contains.html) | 测试输入的字符串是否包含指定的子串 |
| [fn:containsIgnoreCase()](http://www.runoob.com/jsp/jstl-function-containsignoreCase.html) | 测试输入的字符串是否包含指定的子串，大小写不敏感 |
| [fn:endsWith()](http://www.runoob.com/jsp/jstl-function-endswith.html) | 测试输入的字符串是否以指定的后缀结尾 |
| [fn:escapeXml()](http://www.runoob.com/jsp/jstl-function-escapexml.html) | 跳过可以作为XML标记的字符 |
| [fn:indexOf()](http://www.runoob.com/jsp/jstl-function-indexof.html) | 返回指定字符串在输入字符串中出现的位置 |
| [fn:join()](http://www.runoob.com/jsp/jstl-function-join.html) | 将数组中的元素合成一个字符串然后输出 |
| [fn:length()](http://www.runoob.com/jsp/jstl-function-length.html) | 返回字符串长度 |
| [fn:replace()](http://www.runoob.com/jsp/jstl-function-replace.html) | 将输入字符串中指定的位置替换为指定的字符串然后返回 |
| [fn:split()](http://www.runoob.com/jsp/jstl-function-split.html) | 将字符串用指定的分隔符分隔然后组成一个子字符串数组并返回 |
| [fn:startsWith()](http://www.runoob.com/jsp/jstl-function-startswith.html) | 测试输入字符串是否以指定的前缀开始 |
| [fn:substring()](http://www.runoob.com/jsp/jstl-function-substring.html) | 返回字符串的子集 |
| [fn:substringAfter()](http://www.runoob.com/jsp/jstl-function-substringafter.html) | 返回字符串在指定子串之后的子集 |
| [fn:substringBefore()](http://www.runoob.com/jsp/jstl-function-substringbefore.html) | 返回字符串在指定子串之前的子集 |
| [fn:toLowerCase()](http://www.runoob.com/jsp/jstl-function-tolowercase.html) | 将字符串中的字符转为小写 |
| [fn:toUpperCase()](http://www.runoob.com/jsp/jstl-function-touppercase.html) | 将字符串中的字符转为大写 |
| [fn:trim()](http://www.runoob.com/jsp/jstl-function-trim.html) | 移除首位的空白符 |