## 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>

语法：

<c:out value="<string>" default="<string>" escapeXml="<true|false>"/>

value是要输出的内容，如果value是null，那么输出default中的内容，escapeXml决定是否使用XML的转义，不使用转义则将转义字符也输出

例子：

<%@ 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>标签用于设置变量值和对象属性。

<c:set>标签就是<jsp:setProperty>行为标签的孪生兄弟。

这个标签之所以很有用呢，是因为**它会计算EL表达式的值**，然后使用计算结果来设置 JavaBean 对象或 java.util.Map 对象的值。

语法：

<c:set

var="<string>"

value="<string>"

target="<string>"

property="<string>"

scope="<string>"/>

例子：

<%@ 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:forEach>**

用于迭代数组和集合

servlet代码：

…

String[] movieList = {“a”,”b”,”c”};

request.setAttribute(“movieList”, movieList);

…

JSP中用c:forEach：

<c:forEach var=”movie” items=”${movieList}” >

${movie} <%-- 这里取到movie --%>

</c:forEach>

迭代取到数据

**（1）适用对象**：集合（数组、Collection、Map、逗号分隔的String）

**（2）额外属性**：varStatus=”movieLoopCount”

可以得到一个循环计数器${movieLoopCount.count}

显示Count:1 Count:2 etc.

**（3）c:forEach可以嵌套**

<c:forEach var=”movie” items=”${movieList}” > <!—外循环 -->

<c:forEach var=”movie” items=”${movieList}” > <!—内循环 -->

${movie} <%-- 这里取到movie --%>

</c:forEach>

</c:forEach>

**4.<c:if>**