**http://ecourse.gdqy.edu.cn/2009/wlkc/shouye/images/05.gifJSTL标签使用说明**

**1.EL语言介绍**

EL语言是JSTL输出（输入）一个JAVA表达式的表示形式。

在JSTL中，EL语言只能在属性值中使用。EL语言只能通过建立表达式${exp1}来进行调用。在属性值中使用表达式有三种方式。

1、 value属性包含一个表达式

在这种情况下，表达式值被计算出来并根据类型转换规则赋值给value属性。比如：中的${username}就是一个EL，它相当于JSP语句<%=request.getAttribute(“username”)%>或<%=session.getAttribute(“username”)%>

2、 value属性包含一个或多个属性，这些属性被文本分割或围绕

在这种情况下，表达式从左到右进行计算，并将结果转换为字符串型（根据类型转换规则），并将结果赋值给value属性

3、 value属性仅仅包含文本

在这种情况下，字符串型属性value将根据类型转换规则转换为标签所希望的类型。

**2.EL语言的操作符**

取得某个对象或集合中的属性值

为了获得集合中的属性，EL支持以下两种操作

1． 使用.操作符来获得有名字的属性。例如表达式${user.username}表明对象user的username属性

2． 使用[]操作符来获得有名字或按数字排列的属性。

表达式${user["username"]}和表达式${user. username }含义相同

表达式${row[0]} 表明row集合的第一个条目。

在这里user是一个类的对象，它的属性username必须符合标准JavaBean的规范，即必须为username属性定义相应的getter、setter方法。

**3.Empty操作符(空值检查)**

使用empty操作符来决定对象、集合或字符串变量是否为空或null。例如

${empty param.username}

如果request的参数列表中的username值为null，则表达式的值为true。

EL也可以直接使用比较操作符与null进行比较。如${param.firstname == null}。

|  |  |
| --- | --- |
| **比较操作符** | **描述** |
| == 或 eq | 相等检查 |
| != 或 ne | 不等检查 |
| < 或 lt | 小于检查 |
| > 或 gt | 大于检查 |
| <= 或 le | 小于等于检查 |
| >= 或 ge | 大于等于检查 |

**4.Core通用标签**

c:out 标签用于在JSP中显示数据，它有如下属性

|  |  |  |  |
| --- | --- | --- | --- |
| **属 性** | **描 述** | **是否必须** | **缺省值** |
| value | 输出的信息，可以是EL表达式或常量 | 是 | 无 |
| default | value为空时显示信息 | 否 | 无 |
| escapeXml | 为true则避开特殊的xml字符集 | 否 | true |

例子: 您的用户名是:

显示用户的用户名，如为空则显示guest，指定从session中获取username的值显示；

显示username的值，默认是从request(page)中取，如果request中没有名为username的对象则从session中取，session中没有则从application(servletContext)中取，如果没有取到任何值则不显示。

<c:out value="${session.username}" default="guest" escapeXml="false"/>

c:set 标签用于保存数据，它有如下属性

|  |  |  |  |
| --- | --- | --- | --- |
| **属 性** | **描 述** | **是否必须** | **缺省值** |
| value | 要保存的信息，可以是EL表达式或常量 | 否 | 无 |
| target | 需要修改属性的变量名，一般为javabean的实例 | 否 | 无 |
| property | 需要修改的javabean属性 | 否 | 无 |
| var | 需要保存信息的变量 | 否 | 无 |
| scope | 保存信息的变量的范围 | 否 | page |

如果指定了target属性, 那么property属性也必须指定。

例子：

将test.testinfo的值保存到session的test2中，其中test是一个javabean的实例，testinfo是test对象的属性。

<c:set var="test2" scope="session" target="test" property="testinfo"/>, 或者：

<c:set var="test2" scope="session" value="${test.testinfo}"/>

将对象cust.address的city属性值保存到变量city中

<c:set var="city" value="${cust.address.city}"/>

c:remove 标签用于删除数据，它有如下属性

|  |  |  |  |
| --- | --- | --- | --- |
| **属 性** | **描 述** | **是否必须** | **缺省值** |
| var | 要删除的变量 | 是 | 无 |
| scope | 被删除变量的范围 | 否 | 所有范围，包括page、request、session、application等 |

例子:

从session中删除test2变量。

<c:remove var="city" scope="request"/>

**5.流控制标签**

c:if 标签有如下属性

|  |  |  |  |
| --- | --- | --- | --- |
| **属 性** | **描 述** | **是否必须** | **缺省值** |
| test | 需要评价的条件，相当于if (...){}语句中的条件 | 是 | 无 |
| var | 要求保存条件结果的变量名 | 否 | 无 |
| scope | 保存条件结果的变量范围 | 否 | page |

c:choose,c:when,c:otherwise 组合（如果...如果...如果...其他...）标签

c:choose 这个标签不接受任何属性

c:when 标签有以下属性

|  |  |  |  |
| --- | --- | --- | --- |
| **属 性** | **描 述** | **是否必须** | **缺省值** |
| test | 需要评价的条件 | 是 | 无 |

c:otherwise 这个标签同样不接受任何属性

例子：

只有当条件user.generous返回值是true时，才显示user.generous

只有当条件user.stingy返回值是true时，才显示user.stingy

其它所有的情况（即user.generous和user.stingy的值都不为true）全部显示user.generous and user.stingy.

<c:choose>

<c:when test="${user.generous}"><c:out value="${user.generous}"/></c:when>

<c:when test="${user.stingy}"><c:out value="${user.stingy}"/></c:when>

<c:otherwise><c:out value="${user.generous} and {user.stingy}"/></otherwise>

</c:choose>

由于JSTL没有形如if (){…} else {…}的条件语句，所以这种形式的语句只能用、和标签共同来完成了。

**6.循环控制标签**

c:forEach 标签用于通用数据，它有以下属性

|  |  |  |  |
| --- | --- | --- | --- |
| **属 性** | **描 述** | **是否必须** | **缺省值** |
| items | 进行循环的项目(是一个集合) | 否 | 无 |
| begin | 开始条件 | 否 | 0 |
| end | 结束条件 | 否 | 集合中的最后一个项目 |
| step | 步长 | 否 | 1 |
| var | 代表当前项目的变量名 | 否 | 无 |
| varStatus | 显示循环状态的变量 | 否 | 无 |

varStatus 变量的其他属性：

.index 索引号，从 0 开始计算

.count 量值，从 1 开始计算

.first 判断当前对象是不是集合中第一个对象

.last 判断当前对象是不是集合中最后一个对象