1.JSTL:JSP Standard Tag Library，JSP标准标签库

2.使用JSTL的特点：

A.**简化**了JSP和Web应用程序的开发；

B.在JSP中，以标签的形式统一地**减少**Java脚本量，力求实现无脚本化；

C.在应用程序和服务器之间提供了统一的接口，提高了Web应用程序在各服务器之间的**可移植性**；

D.支持JSP设计工具和Web应用程序开发的进一步**集成**；

3.根据使用功能的不同，JSTL标签库分为五类，分别是：

**核心标签库**、SQL标签库、**I18N标签库**（国际化输出标签库）、XML标签库、**EL函数库**。

|  |  |  |
| --- | --- | --- |
| 标签库 | uri | 说明 |
| 核心标签库 | http://java.sun.com/jsp/jstl/core | 包含Web应用的常用功能，例如循环、表达式赋值、基本输入输出等等 |
| SQL标签库 | http://java.sun.com/jsp/sql | 包含访问数据库的功能 |
| I18N标签库 | http://java.sun.com/jsp/jstl/fmt | 包含格式化显示数据的功能，例如对不同区域的日期格式化等等 |
| XML标签库 | http://java.sun.com/jsp/jstl/xml | 包含访问XML文件的功能 |
| EL函数库 | http://java.sun.com/jsp/jstl/functions | 包含读取已经定义的函数的功能 |

4.使用JSTL的步骤：

A.在项目中引入jar包；---在动态Web项目的WEB-INF/lib下，添加jstl.jar和standard.jar文件；

B.添加标签库指令；---在JSP页面顶部，添加指令，例如：

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

C.在页面中写标签。

**通用标签包括以下四种：<c:set> <c:out> <c:remove> <c:catch>**

5.<c:set>用来定义变量，可以将变量存储到JSP不同的范围中或者JavaBean属性中。

其语法格式一般有四种：

**A.<c:set var=”name” value=”value” scope=”page|request|session|application”/>**

**var用于指定变量的名称，value用于指定变量的值，scope用于指定变量所属的范围，默认为page。上述标签将属性名为name，属性值为value的变量存储到相应的范围中。**

**B.<c:set var=”name” scope=” page|request|session|application”>**

**标签体内容**

**</c:set>**

将标签体内容存储到相应的范围中。**没有value属性，有的话反而报错！**

**C.<c:set target=”target” property=”propertyName” value=”value”/>**

**将值存储到target对象的某个属性中。**

**D.<c:set target=”target” property=”propertyName”>**

**标签体内容**

**</c:set>**

将标签体内容存储到target对象的某个属性中。**没有value属性，有的话反而报错！**

6**.**<c:out>用于输出数据，类似于JSP中的<%=表达式%>，但是功能更强，而且使用更方便。

其语法格式有：

A：<c:out value=”value”/>

用于输出要显示的值，value可以是EL表达式或静态值。

B：<c:out value=”value” escapeXml=”true|false”/>

用于输出要显示的值，escapeXml属性指是否将转义字符转换后输出。如果要原样输出，不进行转换，可以将escapeXml属性设置为false。

C：<c:out value=”value” default=”default value”/>

用于输出要显示的值，如果value的值为null，则显示default的值。---value值使用EL表达式时，可能为null

D：<c:out value=”value”> ----value和标签体可以同时存在！

标签体内容

</c:out>

用于输出要显示的值，如果value的值为null，则显示标签体内容。---value值使用EL表达式时，可能为null

7.<c:remove>用于移除指定范围的变量，其语法格式如下：

**<c:remove var=”name” scope=”page|request|session|application”/>**

其中，var用来指定删除的变量的名字，scope用来指定范围，缺省值为page。

8.<c:catch>用于捕获内部代码抛出的异常，其语法格式如下：

**<c:catch var=”name” scope=”page|request|session|application”>**

**标签体内容** // ${10/0}--不会报错，无限大为其输出的值，JAVA代码则会报异常

**</c:catch>**

用于捕获JSP代码中出现的异常，并将其赋值给名称为name的变量，scope指name所属的范围。

9.**条件标签包括以下四种：<c:if> <c:choose>  <c:when> <c:otherwise> 后三种通常一起使用。**

10.<c:if>用于进行条件判断，其语法格式有两种。

**A.：<c:if test=”condition” var=”name” scope=”page|request|session|application”/>**

**其中，test是必须的，表示进行判断的表达式。**var定义变量存放判断后的结果，scope表示var定义的变量的存储范围。var和scope可选。

**B.：<c:if test=”condition” var=”name” scope=”page|request|session|application”>**

**标签体内容**

**</c:if>**

当test中的表达式结果为true时，则会执行标签体内容，否则不会执行标签体内容。

<c:if>只能判断一个表达式，即只能判断一个条件，而且无法指定如果表达式不成立时执行的语句，即不能使用一个<c:if>完成类似于if…else…语句的判断。多种条件的判断可使用其它的条件标签。

11.<c:choose>用于条件选择，一般和<c:when>和<c:otherwise>一起使用。

<c:when>代表<c:choose>的一个分支，<c:otherwise >代表最后的其它的分支。其语法格式如下：

<c:choose>

<c:when test=”condition1”>

         condition1为true时，执行的代码

</c:when>

<c:when test=”condition2”>

         condition2为true时，执行的代码

</c:when>

<c:when test=”condition3”>

         condition3为true时，执行的代码

</c:when>

……

<c:otherwise>

      其它的条件成立时，执行的代码

</c:otherwise>

</c:choose>

其中，<c:otherwise>必须作为<c:choose>的子标签，并且必须是最后一个标签。

12.**迭代标签包括：<c:forEach> <c:forTokens>**

13.<c:forEach>有两种语法格式。

第一种用来遍历集合对象中的成员**：**

<c:forEach var=”name” items=”collection” varStatus=”varStatusName” begin=”begin” end=”end” step=”step”>

标签体内容

</c:forEach>

其中：

var用于指定存放集合中当前遍历的元素的变量名称；

items用于指定被迭代的集合对象的名称；----常是EL表达式

varStatus用来存放当前遍历的元素的状态信息，varStatus常用的属性有current（当前迭代的项）、index（当前迭代从0开始的索引）、count（当前迭代从1开始的迭代计数）；

begin用来指定遍历的起始索引，必须为整数值；

end用来指定遍历的结束索引，必须为整数值，必须大于等于begin；

step用来指定迭代的步长，必须为整数值。var属性和items属性必须指定，其它属性可选。

第二种用来使循环执行指定的次数：**----没有容器变量**

<c:forEach var=”name” varStatus=”varStatusName” begin=”begin” end=”end” step=”step”>

   标签体内容

</c:forEach>

14.<c:forTokens>用来根据指定的分隔符分割字符串。其语法格式如下：

<c:forTokens var=”name” items=”StringOfTokens” delims=”delimiters” varStatus=”varStatusName” begin=”begin” end=”end” step=”step”>

标签体内容

<c:forTokens>

其中：

items属性指定被分割的字符串；

delims属性指定分隔符，可以指定一个或多个分隔符；

其它属性的用法与<c:forEach>中的相同；items属性和delims属性是必须的，其它属性可选。

15.**URL标签包括以下四种：<c:import> <c:redirect> <c:param> <c:url>**

16.<c:import>用于将其它的静态或动态资源包含到当前的JSP页面中。所包含的资源不再局限于当前的Web应用程序，其它Web应用程序或远程的文件也可以被包含。其语法格式如下：

<c:import url=”url” var=”name” scope=”page|request|session|application” charEncoding=”charEncoding”/>

其中：

url用于指定被包含资源的url，可以是绝对地址或者相对地址；

var用于指定存放被包含资源的变量名；

scope用于指定变量的存储范围；

charEncoding用于指定被包含资源的字符编码方式。

---若没有var和scope，这样可以引入页面的内容 ，加上var和scope反而不行！加上了则将网页保存到var里了！

<c:redirect>的url引用此变量值则不能转向，此处的URL的值不是url而是保存网页的内容

有var属性时可以单独用EL表达式输出

17.<c:redirect>可以使当前JSP页面自动跳转到其它资源，并且可以通过<c:param>向跳转到的资源传递参数。其格式如下：

<c:redirect url=”url” context=”context” />

其中：

url用于指定跳转资源的地址；

context用于指定url的参照路径，默认值为当前Web应用程序的上下文路径，如果声明了context值，则context和url必须以“/”开头；

18.<c:param>通常作为<c:redirect>或<c:import>的子标签使用，用来向跳转的资源或者包含的资源传递参数。其格式如下：

<c:param name=”name” value=”value”/>

19.<c:url>用于按照特定的规则重新构造url，语法格式如下：

<c:url value=”original url” var=”name” scope=”page|request|session|application”/>

例如：

<c:url value=”admin/login.jsp” var=”loginUrl” scope=”page”/>

以上语句在page范围内创建一个名字loginUrl的url变量，其值为admin/login.jsp。

20.要在JSP中使用I18N标签库，必须使用taglib指令导入，例如：

<%@ taglib uri=”http://java.sun.com/jsp/jstl/fmt” prefix=”fmt”%>

21.**I18N标签库主要包括：**

**<fmt:setLocale>**

**<fmt:bundle>**

**<fmt:setBundle>**

**<fmt:message>**

**<fmt:param>**

**<fmt:requestEncoding>**

**<fmt:formatNumber>**

**<fmt:formatDate>**

22.<fmt:formatDate>的使用：

<fmt:formatDate value="${date}"/><br/>

<fmt:formatDate value="${date}" pattern="yyyy/MM/dd HH:mm:ss:sss"/>

23.要使用EL函数库，需要使用taglib指令导入：

<%@ taglib uri=”http://java.sun.com/jsp/jstl/functions” prefix=”fn”%>

24.使用EL函数的例子：

大写转小写：I LOVE CHINA!&nbsp;&nbsp;**${fn:toLowerCase("I LOVE CHINA!")}**<br/>

取长度：I LOVE CHINA!的长度为：**${fn:length("I LOVE CHINA!")}**<br/>

video的长度为：**${fn:length(video)}**<br/>

video的值连接成一个字符串为：**${fn:join(video,"、")}**

25.更多信息，参考：

JSP 标准标签库（JSTL） | 菜鸟教程 http://www.runoob.com/jsp/jsp-jstl.html

实现页面跳转的方法：

A.使用<a>标签，超链接跳转；----客户端跳转

B.response.setHeader() 跳转；----客户端跳转

C.<jsp:forward>跳转；----属于服务器跳转

D.response.sendRedirect();----客户端跳转

E.<c:redirect>；

F.JavaScript,对象；

G.pageContext.forward（"pagecontext\_forward\_demo02 .jsp?info=AABB") ;

实现页面包含：

A.<%@ include>

B.<jsp:include>

C.<c:import>