# JSTL ：jsp Standard tag library

## JSTL简介；

Jstl：（jsp standard tag library）是由sun公司定义的规范Apache下的jakarta小组负责实现的开放源代码的标签组件。目标是：简化jsp页面的设计。

## JSTL中主要的标签库：

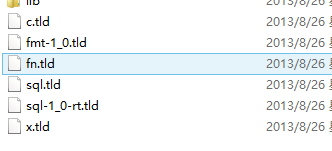


# JSTL的使用

## 安装JSTL

1. 添加jstl.jar包到项目的WEB-INF/lib目录下
2. 复制jstl-impl/META-INF文件夹下的几个主要标签配置文件到WEB-INF目录下

Tld如下：

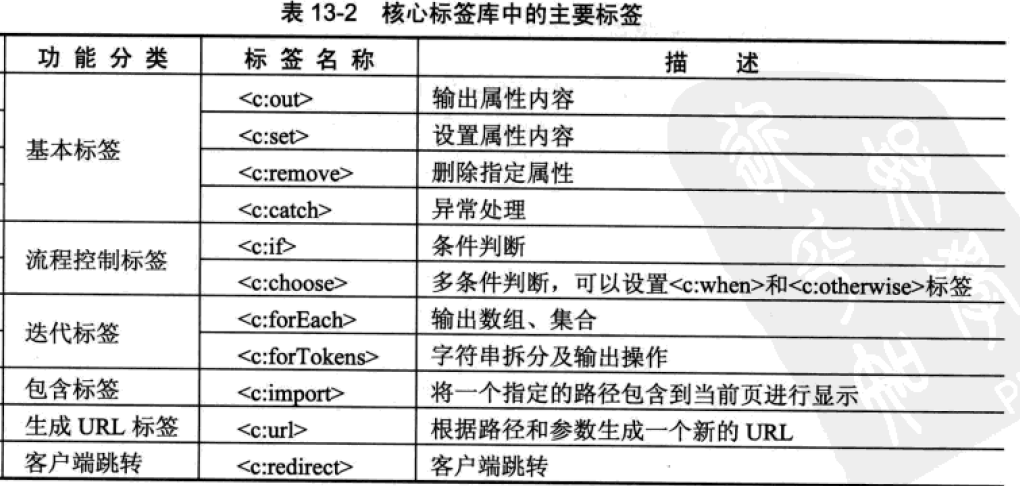


配置完成之后重新启动服务器就可以了。

完成之后我们写一个jsp进行测试：

|  |
| --- |
| <%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>  <%@ taglib uri="/WEB-INF/c.tld" prefix="c" %>  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  <html>  <head>    <title>第一个jstl标签</title>  </head>    <body>  <c:out value="这是用jstl的c：out输出的属性"></c:out>  </body>  </html> |

# 核心库的主要标签：



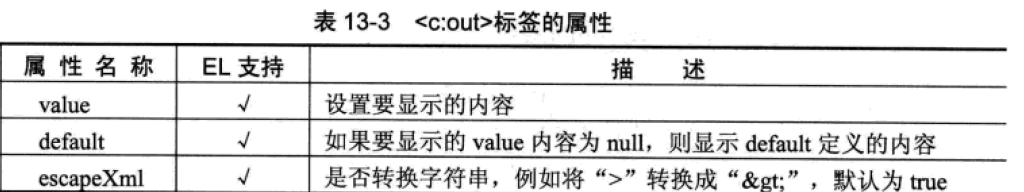
## <c:out>输出属性内容：

和表达式${}，scriptlet的输出是一样的

### 语法：

1. 没有标签体的：
   1. <c:out value=”打印的内容” [escapeXML=”[true|false]”] default=”默认值”/>
2. 有标签体的：
   1. <c:ouit value=”打印的内容” [escapeXML=”true|false”]>默认值</c:out>

### Out标签的属性：



## <c:set>标签：

设置四种属性范围的值（page，session，request，application）

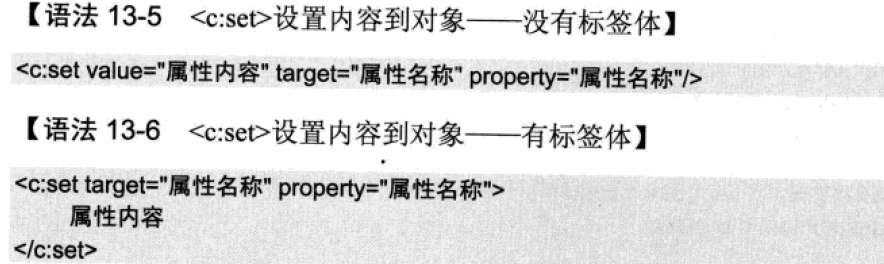
### 有标签体

* 1. <c:set var=”属性名 scope=“page|request|session|application”>属性内容</c:set>

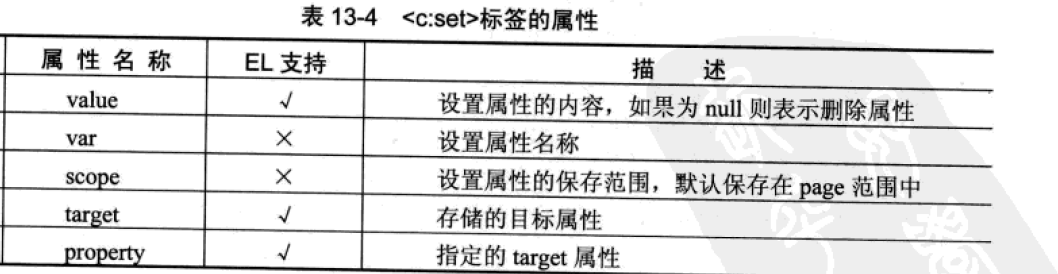
### 没有标签体

<c:set var=”属性名” scope=”page|request|session|application” value=”属性内容”/>

和out输出不同的是：



### Set标签的属性：



### 对于<c:set>设置target的值有可能遇到的问题：

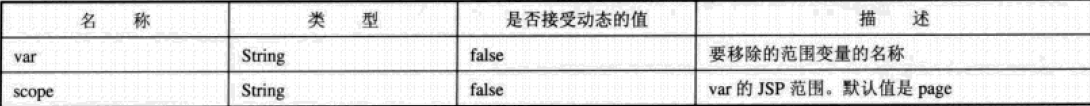
1. target为null
2. target不是java.util.Map对象或者支持setter方法的javabean对象
3. 如果value为null
   1. 由var和scope定义的范围将移除
      1. 如果指定了scope那么将按照PageContext.removeAttribute(varName,Scope);否则则按照Pagecontext.removeAttribute(varName);的方法移除

## <c:remove>标签：

### 语法：

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

### <c:remove>的属性：

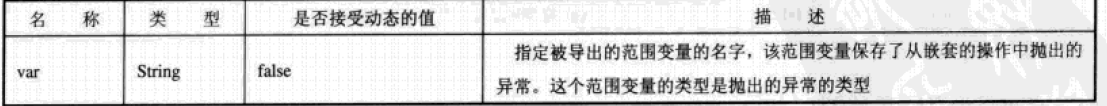


## <c:catch>标签：

### 语法：

<c:catch var=”varName”>有可能异常的部分</c:catch>

### 属性



#### 解释：

我们使用<c:catch>标签进行异常处理。如果catch标签之间的代码抛出异常异常将被捕获并且保存到var变量所标识的scope中

如果没有var则异常只是捕获并不会保存异常信息。

Var变量是保存异常信息的，该变量总是有page范围如果没有异常发生并且而var标识的范围变量存在那么var将被移除

## <c:if>标签：

### 语法：

1. 没有标签体：
   1. <c:if test=”是boolean类型的值” var=“存储判断的结果” scope=“if的作用域范围”/>
2. 有标签体：
   1. <c:if test=”判断条件” var=”存储判断结果” scope=“page|reqest|session|application”>test判断条件正确时会执行的地方</c:if>

## <c:choose><c:when><c:otherwise>标签：

### 语法：

<c:choose>

标签体：（<c:when>,<c:otherwise>）

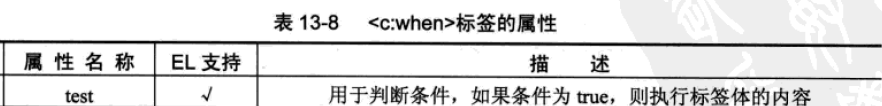
</c:choose>

注意：<c:choose>标签没有任何的属性

标签的内容只能有：

<c:when test=”条件语句”>

满足条件时执行的语句

</c:when>

<c:otherwise>当所有<c:when><c:when>里面的标签内容都不执行时就执行这里面的内容</c:otherwise>标签和switch中的default相同的功能相似也就是当<c:when><c:when>里面的都不执行时就返回<c:otherwise></c:otherwise>标签里的内容

所有的</c:otherwise>要放到c:when>之后，只有当c:when>里面的都不执行时才会有</c:otherwise>。

## <c:forEach>标签：

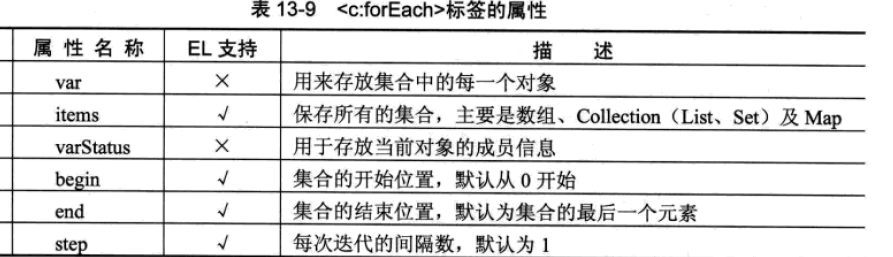
主要功能为循环控制可以将集合中的成员进行迭代输出类似于Iterator迭代

#### 语法：

<c:forEach var=”每一个对象的属性名称” items=“集合” varstatus=“保存相关成员信息” begin=“集合的开始位置” end=“集合的输出位置” step=“每次增长的步长”>

标签体

</forEach>



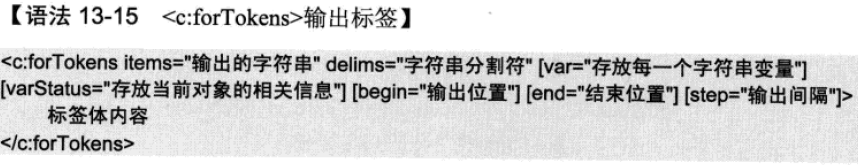
注意：做这个时候遇到异常：

deferredExpression

解决：

|  |
| --- |
| 遇到这个异常的时候是用jstl标签，是版本问题，由于MyEclipse添加Java EE5，其中自动包括了jstl1.2的版本，lib中又存在一个jstl1.1.2的jar包，把旧版本的删掉就可以了 |

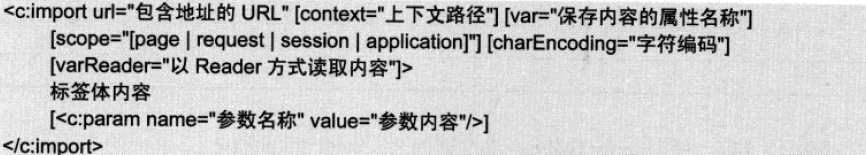
## <c:forTokens>标签：

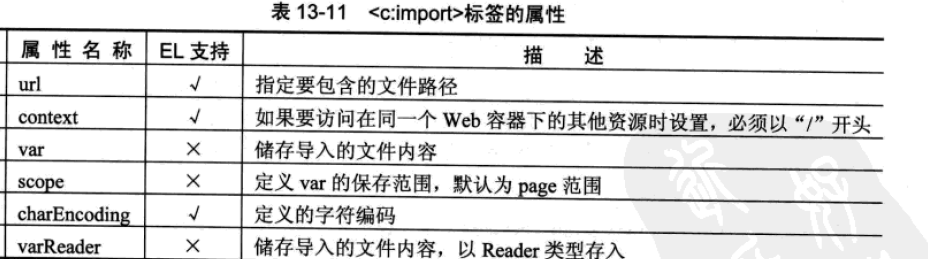




## <c:import>标签：

#### 语法：



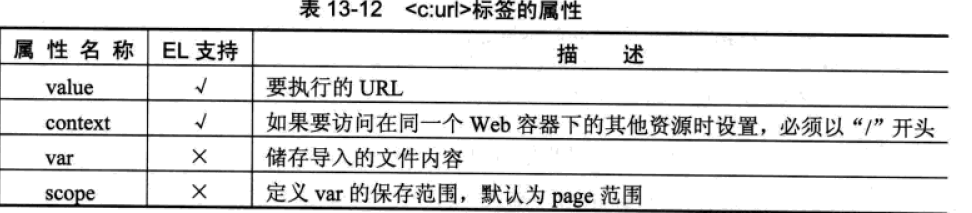


## <c:url>标签：

产生一个新的url

### 语法：





## <c:redirect>标签：

客户端的跳转

