# Java Web（4）

## Jstl标签库

### 概述

1. Apache的产品；
2. 使用时需要导入jar包
3. 四大库：

* Core:核心库，重点；
* Fmt:格式化，日期、数字
* Sql:过时
* Xml:过时

### 导入标签库

1. Jar包；
2. 在jsp页面中：<%@taglib prefix=”前缀” uri=”路径”%>

### Core --> c 标签

1. Out和set

* <c:out>:输出
* Value:可以是字符串常量，也可以是EL表达式；
* Default:当要输出的内容为null 时，会输出default指定的值；
* escapeXml：默认值为true,表示转义；
* <c:set>:设置（创建域的属性）
* Var:变量名；
* Value:变量值，可以是 EL表达式；
* Scope:域，默认为page,可选值为：page,request,session,application

1. Remove

* <remove>:删除域变量
* Var:变量名
* Scope:如果不给出scope,表示删除所有域中的该名称的变量；如果指定了域，那么只删除该域的变量。

1. url

* value:指定一个路径！他会在路径前面自动添加项目名；
* <c:url value=”/index.jsp”/>,他会输出/day13\_1/index.jsp;
* 子标签：<c:param>，用来给url后面添加参数(并进行url编码)，例如：

<c:url value=”/index.jsp”>

<c:param name=”username” value=”张三”>

</c:url>

结果：

/day13\_1/index.jsp?username=%ED%2C%3F%ED%2C%3F

* Var:指定变量名，一旦添加了这个属性，那么url标签就不会在输出到页面，而是把生成的url保存到域中。
* Scope:与var一起使用，用来保存url;

1. If：对用java中的if语句

* <c:if test=”布尔类型”>…</c:if>,当test为true时，执行标签体的内容。

1. Choose：对应java中的if/…/else

* 用法：

<**c:choose**>  
 <**c:when test="${empty** param.name**}"**>  
 没有名为name的参数！  
 </**c:when**>  
 <**c:otherwise**>  
 谁让你给出name参数的？**${**param.name**}** </**c:otherwise**>  
</**c:choose**>

1. forEach

* 用来循环遍历数组、集合
* 还可以用计数方式来循环
* 计数方式：

<c:forEach var=”i” begin=”1” end=”10” step=”2”> (include 10)

${i}

</c:forEach>

* 输出数组和集合：

*<%--输出数组--%>***<%** String[] strs = {**"one"**,**"two"**};  
 request.setAttribute(**"strs"**,strs);  
**%>**<**c:forEach items="${**requestScope.strs**}" var="s"**>  
 **${**s**}**</**c:forEach**><**br**/>

* Items:指定要循环谁，可以是一个数组或一个集合

其中 **}" 之间不能有空格。**

* Var:把数组或集合中的每个元素赋值给var指定的变量。
* 循环状态：

可以使用varStatus来创建循环状态变量！有如下属性；

* Count:循环元素的个数；
* Index: 循环元素的下标；
* First：是否为第一个元素；
* Last:是否为最后一个元素；
* Current:当前元素。
* 循环状态的例子：

*<%--循环状态变量--%>* **<%** ArrayList<String> list = **new** ArrayList<String>();  
 list.add(**"yi"**);  
 list.add(**"er"**);  
 list.add(**"san"**);  
*// 添加到域中* pageContext.setAttribute(**"list"**,list);  
 **%>** <**c:forEach items="${**list **}" var="ele" varStatus="vs"**>  
 **${**vs.index**} ${**vs.count**} ${**vs.first**} ${**vs.last**} ${**vs.current**}**<**br**/>  
 </**c:forEach**><**br**/>

## Fmt标签库常用标签

### <fmt:formatDate value=’’ pattern=’’>

* Value:指定一个Date类型的变量；
* Pattern:用来指定输出的模板。例如：yyyy/MM/dd HH:mm:ss
* *<%--日期格式化--%>*

**<%** Date date = **new** Date();  
 request.setAttribute(**"date"** ,date);  
**%>**<**fmt:formatDate value="${**date**}" pattern="yyyy/MM/dd HH:mm:ss"**/>

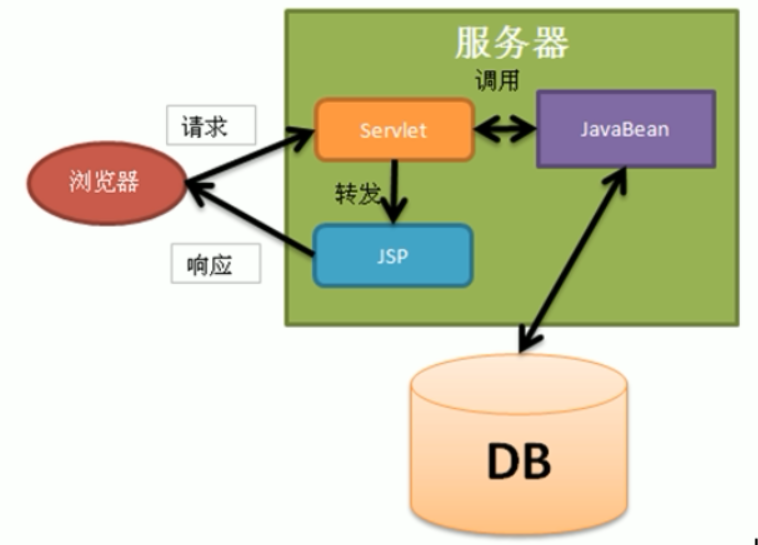
### <fmt:formatNumber value=’’ pattern=’’>

* *<%--数字格式化--%>***<%** request.setAttribute(**"num"**,3.141592653);  
  **%>**<**fmt:formatNumber value="${**num**}" pattern="0.000"**/>
* 会四舍五入，会补0
* 若**pattern="#.###"**

则不会补0，但还会四舍五入。

## MVC设计模式\*

* 不是java独有，所有的B/S结构的项目都在使用它！
* M – Model （模型）自己写代码
* V – view （视图）jsp
* C – Controller （控制器）Servlet
* 示意图：



## JavaWeb三层框架\*

* Web层 – 与Web相关的内容（servlet,jsp,servlet相关的API:request,response,session,ServletContext）
* 业务层 – 业务对象（Service）
* 数据层 – 操作数据库（DAO Data Access Object，所有对数据库的操作，不能跳出DAO之外）
* 示意图：

