# JSP

## jsp的概述

### 什么是jsp

Java server pages ，实际一个jsp就是一个servlet，用于开发动态的web资源的技术.就是在html中写Java代码，写动态web资源的一种技术

JSP = JAVA+HTML 实质就是（servlet）

Jsp动态页面技术，他的执行过程（原理）

Jsp-----翻译成servlet（.Java）------编译成（.class）-----初始化-----实例化-------服务-----销毁

所以第一次访问jsp的时候回比较慢 。

### Jsp的作用

使用servlet来处理用户的请求，并封装相应数据到页面，在页面接收数据的时候需要动态的接收数据，和动态的展示数据，这个时候使用servlet写动态的页面时可以实现的，但是相当的复杂和繁琐。

有了jsp以后我们的展示数据的时候就可以使用jsp来动态的展示数据，用servlet来处理请求和封装相应数据。  
就是java+html 动态处理数据 动态展示数据---在html中写java代码

## Jsp的基本语法

### 1 注释：以使用三种注释

1可以使用html中注释

<!-- html的注释 -->

2可以对jsp脚本内容使用java注释

//单行注释 /\* 多行注释 \*/

3可以使用jsp注释

<%-- jsp注释 --%>

### 2 jsp的小脚本（写Java代码的地方）

1) <% java代码....%>

2) <%=要输出的内容%>

3) <%! 声明的变量%> --------Java中的成员变量的位置（）

### 3 jsp的指令 <%@ 指令名 属性=””%>

指令作用域整个页面，

1)page指令

|  |
| --- |
| <%@ page language="Java" contentType="text/html; charset=UTF-8"  import="java.util.\*，java.lang.\*" pageEncoding="UTF-8"%> |

2)include指令 用于静态包含

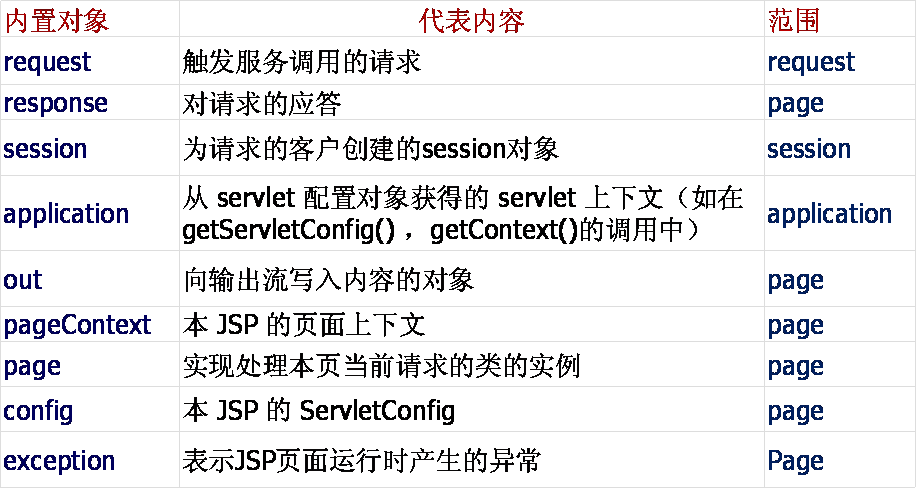
|  |
| --- |
| <%@ include file="myjsp1.jsp" %> |

3) taglib 指令 用于引入一些jsp的标签库

|  |
| --- |
| <%@taglib prefix="" uri="" %>  uri 标签文件的URI地址  prefix 标签组的命名空间前缀 |

## JSP的内置对象

什么是jsp的内置对象 ，jsp是一个servlet。Servlet就要对数据进行处理，就要有对象，这些对象不要要我们去定义，由容器生成好的一些对象



request HttpServletRequest

response HttpServletResponse

session HttpSession

application ServletContext

config ServletConfig

page this (HttpServlet)

pageContext PageContext

exception Throwable (所有异常父类)

out JspWriter

###### jsp中四个域/对象

jsp有四个域对象，分别是：

* + - * application：域范围是整个应用；
      * session：域范围是整个会话，注意一个会话中只有一个用户，如果要统计站点的访问次数，使用的域是ServletContext，因为这个时候统计的是多个用户，而非一个用户；
      * request：域范围是整个请求链；
      * pageContext：（page域）域范围是一个jsp页面，在一个页面中存数据再取数据没有任何意义，所以这里所说的域指的是在当前jsp页面和当前jsp页面中使用的标签之间共享数据。

域范围由小到大 pageContext----->request ---->session---->application

所有的域中都维护的是一个map集合和储存数据

pageContext域----------作用范围当前的页面

Request 域----------------作用范围一次请求

Session域------------------作用范围一次会话

Application域-------------作用范围整个应用程序

# EL表达式

什么是EL表达式？

是jsp的内置（Expression Language）表达式语言，它的出现时用来取代jsp中的Java代码的。

格式：${表达式} 主要的作用是获取放在域中的数据

可以取代任何的<%= %>等输出。并且支持算数运算。

作用

1获取数据

EL表达式主要用于替换JSP页面中的脚本表达式，以从各种类型的web域 中检索java对象、获取数据。(某

个web域中的对象，访问javabean的属性、访问list集合、访问map集合、访问数组)

2执行算数运算

3获取web对象

4抵用Java方法

###### EL表达式之域查找（获取数据）

EL 表达式实现取数据的原理是

pageContext.findAttribute("");

1.page-----------------------------${pageScope.xxx}

2.request--------------------------${requestScope.xxx}

3.session--------------------------${sessionScope.xxx}

4.application----------------------${applicationScope.xxx}

通过el获取时，如果没有查找到，会返回"".

EL 表达式的运算符

. [] empty

|  |
| --- |
| EL表达式可与取出四大域中的数据，以后servlet处理的数据放在相应的域对象<br/>  中，EL表达式就取出来展示到页面上,jsp 用于动态的展示数据，jsp负责处理</h3><br/>  ${"可用于输出数据"}<br/>  <%  //将数据放在四大域中域中  request.setAttribute("request", "request--value");  session.setAttribute("session", "session--value");  pageContext.setAttribute("pageContext1", "pageContext--value");  application.setAttribute("application", "application--value");    %>  <h2>取出四大域中的数据---字符串：</h2><br/>  ${request }<br/>  ${session }<br/>  ${pageContext1 }<br/>  ${application }<br/>  <%  Book b = **new** Book();  b.setId(1);  b.setName("<<我与阿娇的故事>>") ;  b.setAuther("陈冠希");  b.setPrice(12.00);  //将数据放在域中  request.setAttribute("bookrequest", b);  session.setAttribute("booksession", b);  pageContext.setAttribute("bookpage", b);  application.setAttribute("bookapplication", b);  %>  <h2>取JavaBeans中的数据</h2><br/>  ${requestScope.bookrequest.sd }<br/>  ${sessionScope.booksession.name }<br/>  ${pageScope.bookpage.price }<br/>  <%  String arr[] = {"wo","ai","ni","China"} ;  request.setAttribute("requestarr", arr);  session.setAttribute("sessionarr", arr);  pageContext.setAttribute("pagearr", arr);  application.setAttribute("applicationarr", arr);  %>  <h2>取数组中的数据</h2><br/>  ${requestarr[0]}<br/>  ${sessionarr[1]}<br/>  ${pagearr[2]}<br/>  ${applicationarr[3]}<br/>  <%  List list = **new** ArrayList();  list.add("aaa");  list.add("bbb");  list.add("ppp");  list.add("CCC");  //将数据放在域中  request.setAttribute("listrequest", list);  session.setAttribute("listsession", list);  pageContext.setAttribute("listpage", list);  application.setAttribute("listapplication", list);  %>  <h2>取List中的数据</h2><br/>  ${listrequest[0]}<br/>  ${listsession[1]}<br/>  ${listpage[2]}<br/>  ${listapplication[3]}<br/>  ${**empty** listrequest }//判断是否为空    <%  Map<String ,Object>map = **new** HashMap<String,Object>() ;  map.put("p", "ppp");  map.put("b", b);  pageContext.setAttribute("map", map,pageContext.SESSION\_SCOPE);  %>  <h2>取map中的数据</h2><br/>  ${map["p"]}----------- ${map.p}<br/>  ${map.b.name} |

###### EL表达式11个内置对象介绍

* pageScope page域
* requestScope
* sessionScope
* applicationScope
* param；
* paramValues；
* header；
* headerValues；
* initParam；
* cookie；
* pageContext； 他是jsp的内置对象 this （HttpServlet）

### 使用EL设置应用程序名

${pageContext.request.contextPath} 重要常用

为什么可以直接使用pageContext对象是因为pageContext作为el表达式11个隐式对象中的一个。而pageContext对象又可以调用一个getRequest()方法，这样就可以写request,而request对象又有一个getContextPath()方法，这样就可以写contextPath.

# JSTL标签库

JSTL简介

STL(JSP Standard Tag Library，jsp标准标签库)是Apache对EL表达式的扩展，也就是说JSTL依赖EL表达式。JSTL是标签语言，使用起来非常方便。但是它不是jsp内置的标签，所以用的时候需要我们自己导包，以及指定标签库

Jstl标签库一共包含了四大标签库：

* core：核心标签库，这是我们学习的重点；
* fmt：格式化标签库，
* sql：数据库标签库，这个标签库已经过时，本文档不再对其进行介绍；
* xml：xml标签库，这个标签库已经过时，本文档不再对其进行介绍。

另外Jstl标签库中还包含了jstl函数。

###### 怎样使用jstl标签库

* 导入jar包；
* 在使用标签的jsp页面中使用taglib指令导入标签库。

例如，在jsp页面导入jstl的核心标签库：

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

 prefix="c"：指定标签库的前缀，这个前缀可以赋任意的值，但大家都会在使用core标签库时指定前缀为c；

 uri="http://java.sun.com/jsp/jstl/core"：指定标签库的uri，它不一定是真实存在的网址，但它可以让JSP找到标签库的描述文件。

###### stl核心标签库标签介绍

JSTL core标签库中 12个标签

1、c:out 输出内容到浏览器端

2、c:set 设置值到四种数据范围，设置四个范围中java对象的一个属性值

3、c:remove 用来删除四个数据范围的数据

4、c:catch 捕获程序异常

5、c:if 取代页面中if条件语句，记住没有c:else

<c:if test="${m>5}" > </c:if>

6、c:choose c:when c:otherwise 一起使用 ，取代if elseif else 结构 、switch-case 结构

7、c:forEach 取代页面中for循环

8、c:forTokens 切割字符串

9、c:import 效果和include类似

10、c:url 完成URL重写 （禁用cookie的session追踪 ） ---- 结合c:param完成 中文URL编码

11、c:redirect 完成请求重定向

重点: c:if c:foreach 的使用

###### forEach属性介绍

* begin属性

设置循环变量从几开始

* end属性

设置循环变量到几结束；

* step属性

设置循环变量的步长

* var属性

定义一个变量，用于接收循环或把数组或集合中遍历的每一个元素赋值给var指定的变量

* varStatus属性

varStatus属性就是用来记录循环状态的，它可以创建一个循环变量vs，该循环变量有如下属性：

* count：用来记录循环元素的个数；
* index：用来记录所循环元素的下标；
* first：判断当前循环的元素是否是第一个元素；
* last：判断当前循环的元素是否是最后一个元素；
* current：代表当前循环的元素。
* items属性

指定要循环的变量，可以是一个数组也可以是一个集合，默认是支持EL表达式

###### forEach遍历容器

* 数组
* List集合
* Map集合

|  |
| --- |
| <h3>遍历数组</h3>  <br />  <%  **int**[] a = { 1, 2, 3, 4, 5 };  session.setAttribute("arr", a);  Map map = **new** HashMap();  map.put("a", "a");  map.put("b", "b");  map.put("c", "c");  request.setAttribute("map", map);  List list = **new** ArrayList();  list.add(**new** Book(1, 12.00, "JAVA1", "ME1"));  list.add(**new** Book(2, 12.00, "JAVA2", "ME2"));  list.add(**new** Book(3, 12.00, "JAVA3", "ME3"));  list.add(**new** Book(4, 12.00, "JAVA4", "ME4"));  session.setAttribute("bookList", list);  %>  <c:forEach var=*"a"* items=*"*${arr}*"* varStatus=*"vs"*>  ${a }<br />  第几个元素${vs.count}<br />  </c:forEach>  <h3>遍历map</h3>  <br />  <c:forEach var=*"map"* items=*"*${map}*"*>  ${map.key}-------${map.value }  </c:forEach>  <h3>遍历List</h3>  <br />  <table border=*"1px"*>  <c:forEach var=*"book"* items=*"*${bookList}*"* varStatus=*"vs"*>  <c:if test=*"*${vs.count%2==1}*"*>  <tr bgcolor=*"green"*>  </c:if>  <c:if test=*"*${vs.count%2==0}*"*>  <tr bgcolor=*"pink"*>  </c:if>  <td width=*"100"* height=*"30"* id=*"*${vs.count }*"*>${book.sd}</td>  <td width=*"100"* height=*"30"* id=*"*${vs.count }*"*>${book.price}</td>  <td width=*"100"* height=*"30"* id=*"*${vs.count }*"*>${book.name}</td>  <td width=*"100"* height=*"30"* id=*"*${vs.count }*"*>${book.auther}</td>  </tr>  </c:forEach>  </table> |

###### forEach案例-遍历商品信息

总结： jsp支持写Java代码，但是HTML和Java代码写在一块，不容易维护和开发

EL表达式和JSTL的出现就是为了替代jsp中的<% %>小脚本 。使jsp中的能实现动态的数据显示又使Java和HTML代码分离开来 。