第6章　JSP　标签

- 6.1　JSP指令元素
- 6.2　JSP动作
- 习题6

6.1 JSP指令元素

JSP指令元素为翻译阶段提供全局信息。例如，设置全局变量的值和输出内容的类型，声明要引用的类，指明页面中包含的文件等。所有的指令元素在整个文件的范围内都有效。指令元素从JSP发送这些信息到JSP/Servlet容器上，但它们并不向客户机产生任何输出。

目前，JSP中有三个指令元素，分别是page、include和taglib。

6.1.1 page指令

page指令用来定义JSP文件的全局属性。其语法格式如下：

<%@ page 属性1 ="属性1的值"　属性2 ="属性2的值" ……. %>

例如：

<%@ page contentType ="text/html;charset=GB2312" %>

这条page指令就指定了contentType属性的值是"text/html;charset=GB2312"，即JSP页面的MIME类型是text/html，使用的字符集是GB2312，这样可以显示标准的汉字。

page指令的属性主要包括：language，import，contentType，info，pageEncoding，buffer，autoFlush，session，errorPage，isErrorPage，isThreadSafe等。表6.1列出了page指令的属性及其作用。

表6.1 page指令的属性及其用法说明

属　性	作　　用	默认值	例　　子
language	定义要使用的脚本语言	java	language ="java"
import	为JSP页面引入Java类和包。各类或包间用逗号分隔	Java.lang.*; javax.servlet.* javax.servlet.jsp.* javax.servlet.http.*	import="java.io.* ", "java.util.Hashtable", "javax.servlet.jsp.* "
buffer	指定out使用的缓冲区大小	通常不小于8 KB	buffer="64 KB"
contentType	定义 JSP 字符编码和页面响应的 MIME 类型	contentType="text/html" charset=ISO-8859-1	contentType="text/html; charset=GB2312"
pageEncoding	JSP页面的字符编码	pageEncoding="ISO-8859-1"	pageEncoding="GB2312"
info	提供JSP页面的信息	无	info="一个测试页面"

续表

属　性	作　　用	默认值	例　　子
autoflush	指定 out 的缓冲区是否自动刷新	true	autoFlush="true"
session	用于设置是否需要使用内置的 session 对象	true	session="false"
errorPage	定义页面出现异常时调用的页面	无	errorPage="error/error.jsp"
isErrorPage	表明当前页是否是其它页的 errorPage 目标。如果被设置为 true，则可以使用 exception 对象；为 false 时不能使用 exception 对象	false	isErrorPage="true"
isThreadSafe	用来设置 JSP 文件是否可多线程访问。如果设置为 true，则一个 JSP 文件能够同时处理多个用户请求	true	isThreadSafe="true"

在一个JSP页面中，可以用一条page指令来指定多个属性的值，也可以用多条page指令来指定各个属性的值。但是，需要注意的是，除import属性外，其它属性只能使用一次page指令给该属性指定一个值。由于import属性的取值较多，因此，可以在一条page指令中为import属性指定多个值，各个值间用逗号分隔；也可以使用多个page指令给import属性指定几个值。例如，下面两种写法都是正确的：

<%@ page import="java.io.* ", "javax.servlet.* ", "java.util.Date"%>

或者

<%@ page import="java.io.* " %>

<%@ page import= "javax.servlet.* ", "java.util.Date"%>

注意：page指令对整个页面有效，可以在JSP页面的任何地方写这种代码。但是，好的习惯是把它写在JSP程序的最前面，而且因为它是JSP页面指令，请记住一定要写在<HTML>标记的前面。

6.1.2 include指令

include指令的作用是在JSP页面出现该指令的位置处静态插入一个文件，即通知JSP容器在当前页面的include指令位置上嵌入指定的资源文件的内容。include指令的语法如下：

<%@ include file="文件名"%>

所谓静态插入，就是将当前JSP页面和插入的部分合并成一个新的JSP页面，然后再由JSP引擎将这个新的JSP页面转译成Java类文件。因此，使用include指令插入文件时，应注意下面几个问题：

(1) include指令所包含文件的文件名不能是变量，文件名后也不能带任何参数。文件的扩展名可以是 .jsp、.html、.txt和 .inc等，且必须保证被插入的文件是可获得和可访问的。

(2) 如果在文件名中包含有路径信息，则路径必须是相对于当前JSP网页文件的路径，一般情况下该文件必须和当前JSP页面在同一Web服务目录中。如果路径以“/”开头，则这个路径主要是参照JSP应用的上下关系路径；如果路径是以目录名开头，则这个路径就是正在使用的JSP文件的当前路径。

(3) 使用include指令插入文件后，必须保证新合并成的JSP页面符合JSP语法规则，即形成一个新的JSP页面文件后，不存在语法冲突。例如，如果在一个JSP页面中使用include指令插入另一个JSP文件，而这两个文件中都用page指令设置了页面contentType属性，如果二者不一致时就会出现语法错误，当转译合并JSP页面到Java文件时就会失败。

(4) 如果修改了被包含的文件，则也应将当前的JSP文件修改一下(在实际操作中，就是在编辑状态下打开该JSP文件，重新保存一次)。这是因为JSP引擎是通过比较JSP页面文件和相应的字节码文件的最后修改日期来判断JSP页面是否被修改过，如果两者相同，则认为JSP网页未被修改。由于在include指令中被包含的文件是在编译成字节码文件之前插入到源JSP页面中的，所以，如果只是修改了被包含的文件，而没有修改JSP页面文件，则得到的结果将和修改前是一样的。

【示例程序include.jsp和calculate.jsp】

include指令的使用。在这个例子中我们编写了一个计算平方根的程序calculate.jsp，然后在include.jsp中使用<%@ include file="calculate.jsp"%>指令将这个文件包含进来，使include.jsp能完成计算平方根的功能。其执行效果如图6.1所示。

图6.1 在include.jsp中包含calculate.jsp的效果

(1) include.jsp文件的源代码如下。

```
<%@ page contentType="text/html; charset=gb2312"%>
<HTML>
<HEAD>
<TITLE>使用include包含文件</TITLE>
</HEAD>
<BODY>
<P ALIGN=center>
请输入一个正数，单击按钮计算这个数的平方根！
<%@ include file="calculate.jsp"%>
</P>
</BODY>
</HTML>
```

(2) calculate.jsp文件的源代码如下。

```
<%@ page contentType="text/html; charset=gb2312"%>
<FORM  ACTION="" METHOD="post">
<INPUT  TYPE="text" NAME="ok">
<INPUT  TYPE="submit" VALUE="计算">
</FORM>
<% String a=request.getParameter("ok");
   if(a==null) a="1";
   try
```

```
        {
          double number=Integer.parseInt(a);
          out.println("<BR>计算结果是：
"+Math.sqrt(number));
        }
        catch(NumberFormatException e)
        {   out.println("<BR>请输入数字");   }
      %>
```

6.1.3 taglib指令

这个指令用于引入一些特定的标签库以简化JSP页面的开发。这些标签可以是JSP标准标签库(JSP Standard Tag Library，JSTL)中的标签，也可以是使用者自己定义和开发的标签。使用JSP标签库的语法格式如下：

<%@ taglib prefix="标签前缀" uri="标签库的统一资源定位符" %>

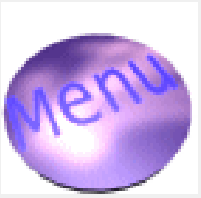
其中，prefix指出要引入的标签的前缀；uri(Uniform Resource Identifer，统一资源定位符)用于指出所引用标签资源的位置，可以使用绝对路径或相对路径。例如：

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

表示从JSP标准标签库的core库中引入前缀为c的标签。

使用标签库的主要好处是增加了代码的重用度，使页面易于维护。例如，可以把一些需要迭代显示的内容做成一个标签，在每次迭代显示时，使用这个标签就可以了，不必重复书写这些代码。

然而，由于目前的tomcat中还没有将JSTL集成进去，如果要使用标准标签库(JSTL)，需要下载和安装标签库文件，修改web.xml文件并进行相关的设置等，操作比较繁杂。虽然通过下载并安装MyEclipse插件可以解决一些问题，但仍然需要具备一些相关的知识。鉴于此，我们将在第11章介绍了XML语言后，在第12章专门介绍JSP中开发自定义标签库的技术和使用标准标签库的方法。



6.2 JSP 动 作

与在编译阶段提供全局信息的JSP指令元素相对应，JSP动作元素在执行阶段起作用，动态地为页面提供一些信息和插件等。JSP动作元素采用类似HTML/XML语法书写，并采用以下两种语法格式中的一种：

<jsp:动作名 属性1="值1" 属性2="值2"…… />

或者

<jsp:动作名 属性1="值1" 属性2="值2"…… >

……

</jsp: 动作名>

JSP规范定义了一系列的标准动作，它们均以jsp作为前缀。这些动作元素中使用比较频繁的主要有：<jsp:param>，<jsp:include>，<jsp:forword>，<jsp:plugin>，<jsp:fallback>，<jsp:useBean>，<jsp:setProperty>，<jsp:getProperty>，<jsp:attribute>等。

6.2.1 param动作

param动作以“名—值”对的形式为其它标签提供附加信息。这个动作与<jsp:include>、<jsp:forward>、<jsp:plugin>动作一起使用。它的使用方式如下：

<jsp:param name="名字" value="指定给param的值"/>

我们将在下面的几个小节中结合<jsp:include>、<jsp:forward>、<jsp:plugin>动作的使用来说明<jsp:param>动作。

6.2.2 include动作

如果需要在JSP页面内某处动态地加入一个文件，可以使用include动作。该动作告诉JSP页面，在JSP页面执行时将指明的文件加入进来。其使用格式如下：

```
<jsp:include page="文件名" flush="true"/>
或者
<jsp:include page="文件名" flush="true">
<jsp:param name="名字" value="指定给param的值"/>
……
</jsp:include>
```

include动作与include指令有下述几点不同：

(1) include动作动态地插入文件到JSP页面中，而include指令静态地插入文件到JSP页面中。即当JSP引擎把JSP页面转译成Java文件时，不把JSP页面中用include动作所包含的文件与原JSP页面合并成一个新的JSP页面，而是告诉Java解释器，这个文件在JSP运行时(Java文件的字节码文件被加载执行时)才包含进来。如果被包含的文件是普通的HTML文件(静态文件)，就将文件的内容发送到客户端，由客户端负责显示；如果被包含的文件是JSP文件(动态文件)，JSP引擎就执行这个文件，然后将执行结果发送到客户端，由客户端负责显示执行结果。

(2) 由于include动作在执行时才对包含的文件进行处理，因此，JSP页面和它所包含的文件在逻辑上和语法上都是独立的。如果对include动作中包含的文件进行了修改，那么运行时可以看到所包含文件修改后的结果；而如果对include指令中包含的文件进行了修改，则必须重新编译JSP页面文件，否则只能看到所包含文件修改前的内容。

(3) 当include动作与param动作一起使用时，可以将param动作中的参数值传递到include动作要加载的文件中去。因此，include动作如果结合param动作，可以在加载文件的过程中向该文件提供信息。

(4) include动作可以动态增加内容，但它的运行效率比include指令低。

【示例程序jsp_include.jsp、twoParam.jsp 和login.html】

include动作与include指令的应用对比。在这个例子的jsp_include.jsp程序中，我们分别使用<%@ include %>指令和<jsp:include>动作包含了twoParam.jsp程序，并在twoParam.jsp和jsp_include.jsp两个文件中都使用JSP内置对象request的getParameter方法获取两个参数yourname和yourpass的值，在jsp_include.jsp中还使用<jsp:param>动作传递这两个参数的值。下面是这几个文件的源代码。

(1) 程序jsp_include.jsp的源代码如下。

```
<%@ page contentType="text/html; charset=gb2312" language="java" %>
<HTML><BODY>
两种不同的文件包含方式执行效果比较:<BR>
<%@ include file="login.html" %>
```

1.使用include指令静态包含twoParam.jsp的执行效果:

```
<%@ include file="twoParam.jsp" %>
<BR><BR>
```

2.使用include动作动态包含twoParam.jsp的执行效果：

```
<jsp:include page="twoParam.jsp" flush="true">
<jsp:param name="yourname"
value="<%=request.getParameter("user")%>" />
<jsp:param name="yourpass"
value="<%=request.getParameter("passw")%>" />
</jsp:include>
</BODY></HTML>
```