Web语言程序设计

第五章： Java Web编程技术

第五章： Java Web编程技术

1. 理解服务器端执行
2. Java web页面结构
3. Java web编译指令
4. Java web操作指令
5. Java web代码

第五章： Java Web编程技术

1 Java Web理解服务器端执行

HTML可以直接在客户端浏览器中执行，但是Java web必须通过Web服务器执行，然后通过HTTP进行访问，<%%>内的代码是被Web服务器编译后解释执行的（Java代码），以下通过程序让大家理解服务器端和客户端的执行过程。

第五章： Java Web编程技术

程序5-1. jsp

理解服务器端执行

服务器执行

```
<%
  java.util.Date dnow=new java.util.Date();
  int dhours=dnow.getHours();
  int dminutes=dnow.getMinutes();
  int dseconds=dnow.getSeconds();
  out.print("服务器时间： "+dhours+":"+dminutes+":"+dseconds);
  System.out.println("服务器时间： "+dhours+":"+dminutes+":"+dseconds);
  %>
```

客户端执行

```
      <SCRIPT language="JavaScript">
   var dnow=new Date();
   dhours=dnow.getHours();
   dminutes=dnow.getMinutes();
   dseconds=dnow.getSeconds();
document.write("<br>浏览器时间： "+dhours+":"+dminutes+":"+dseconds);
  </SCRIPT>
```

第五章： Java Web编程技术

1 Java Web理解服务器端执行

查看以上程序的源代码，所有的<%%>中的内容是看不见的，只有执行的结果，是在web服务器执行的，JavaScript代码我们是可以看见的，因为JavaScript是在浏览器执行的。

2 Java Web页面结构

2. 1 Java Web 注释

注释分为两种，包括HTML注释和隐藏注释。

1. HTML注释：在客户端显示一个注释。

HTML注释的Java Web语法：

```
<!-- comment [ <%= expression %> ] -->
```

2 Java Web页面结构

2. 1 Java Web 注释

注释分为两种，包括HTML注释和隐藏注释。

2. 隐藏注释：写在JSP程序中，但是不发给客户。

隐藏注释的Java Web语法：

```
<%-- comment --%>
```

2 Java Web页面结构

2. 2 Java Web 页面执行元素

Java Web页面主要有四种元素：

编译指令；操作指令；Java Web代码；EL表达式。

1.编译指令主要是告诉JSP解释引擎（Tomcat），在编译的时候需要做什么动作，引入类，字符集，编译语言等。

2 Java Web页面结构

2. 2 Java Web 页面执行元素

2.操作指令主要是在Java Web页面被请求时动态执行的，可以根据某个条件动态跳转。

3.Java Web代码主要是指嵌入在Web页面当中的Java代码，主要包括三种

2 Java Web页面结构

2. 2 Java Web 页面执行元素

Java Web代码 主要包括三种：

1) 声明， <%! 变量、函数或方法 %>；

2) 脚本代码， <% Java 代码%> ；

3)表达式， <%=Java 表达式%>。

3 编译指令

Java Web语法规范定义了以下3种不同的指令。

page指令：定义与JSP页面相关的属性，并和JSP引擎进行通信。

include指令：定义JSP编译时需要插入的资源。

taglib指令：定义JSP页面可以调用的一个客户标记库。

<%@ page language="java" %>

<%@ include file="hearder.htm"%>

<%@ taglib uri=#prefix="util" %>

3.1 page指令

page指令主要用来定义整个JSP页面的属性和相关功能，并由该指令和JSP引擎进行通信。一个JSP页面可以包含多个page指令，指令之间是相互独立的，并且指令中除import属性之外的每个属性只能定义一次，否则在JSP页面的编译过程中将出现错误。

page指令可以运用于整个JSP文件，一般来说，page指令可以放在JSP页面的任何位置，但为了便于程序的阅读和格式规范，通常将page指令放在JSP页面的开始部分。

- **page指令的详细格式如下：**
  - <%@ page
  - [language="java"] [extends="package.class"]
  - [import="{package.class | package.*} , ... "]
  - [session="true|false"]
  - [buffer="none| 8kb|sizekb"] [autoFlush="true|false"]
  - [isThreadSafe="true|false"] [info="text"]
  - [errorPage="relativeURL"] [isErrorPage="true| false"]
  - [contentType="{mimeType [; charset=characterSet] |
  - text/html ; charset=ISO-8859-1}"]
  - [pageEncoding="{characterSet | ISO-8859-1}"]
  - [isELIgnored="true | false"]
  - %>

3.1 page指令

page指令针对于当前页面，由<%@%>进行标记，常用的指令有8个：language; extends; import; errorPage; isErrorPage; contentType, isTreadSafe; session;

3.1 page指令

（1）language="java"

声明脚本语言的种类，暂时只能用“java”。

（2）extends="package.class“

Extends是当前JSP要继承的父类，一般无需设置。在默认情况下，JSP页面的默认父类是HttpJspBase。例如：当前JSP页面要继承mypackage包下的myclass类，相应的声明语句为：“<%@ page extends="mypackage.myclass"%>”。

第五章： Java Web编程技术

3.1 page指令

（3） import="{package.class | package.* }, ..."

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

或者 <%@ page import ="java.io.*" %>

<%@ page import ="java.util.Hashtable" %>

需要引入的包，或是引入的类，是JDK当中的类，也可以是用户自定义的类。有些类在默认情况下已经被加入到当前JSP页面，而不需要特殊声明，包括四个类：java.lang.*;、java.servlet.*;、java.servlet.jsp.*;和java.servlet.http.*;。

3.1 page指令

（4）<%@ page isErrorPage="true" %>

isErrorPage用来设定当前的JSP页面是否作为传回错误页面的网页，默认值是“false”。如果设定为“true”，则JSP容器会在当前的页面中生成一个exception对象。

3.1 page指令

（5）<%@ page errorPage=“/error.jsp" %>

errorPage用来设定当JSP页面出现异常（Exception）时，所要转向的页面。如果没有设定，则JSP容器会用默认的当前网页来显示出错信息。例如：“<%@page errorPage="/error/error_page.jsp"%>”

程序5-2.jsp和5-3.jsp

第五章： Java Web编程技术

5-2.jsp

```
<%
  int a=0;
  int b=0;
  int c=0;
  try{
  c=a/b; }
  catch(ArithmeticException aa){
  throw new ArithmeticException("除数不能为零!!");
  }  %>
```

5-3.jsp

```
 <%out.print(exception.toString());%>
```

3.1 page指令

（6）<%@ page contentType = "text/html" %>

设置传回网页的文件格式及编码方式,一般使用“text/html;charset=GBK”。

（7）isTreadSafe

定义web服务器执行JSP程序的方式，默认“true”，代表多线程的方式执行jsp页面，“false”代表单线程的方式执行。

（8）session

当前jsp页面是否用到session对象。

3.2 include指令

include指令用来指定JSP文件被编译时需要插入的资源，这个资源可以是文本、代码、HTML文件或JSP文件。该指令的格式如下：

<%@include file="relativeURL"%>

其中，relativeURL表示要包含的文件路径。如果路径以“/”开头，则表示该路径是参照JSP应用的上下关系路径，如果路径直接以目录名或文件名开头，则表示该路径是正在使用的JSP文件的当前路径。一旦JSP文件完成编译，该资源内容就不可变，要改变该资源内容，必须重新编译JSP文件。

3.2 include指令

利用include指令，可以将一个复杂的JSP页面分为若干个部分，这样可以方便管理JSP页面。一个JSP页面一般可以分为三段：head（页头）、body（页体）和tail（页尾）。

可以将一个JSP页面分为3个不同的JSP页面：head.jsp、body.jsp和tail.jsp，其中head.jsp表示页头，body.jsp表示页体，tail.jsp表示页尾，这样对于同一网站的不同JSP页面，可以直接利用include指令调用head.jsp和tail.jsp，仅body.jsp不同。

程序5-4.jsp

3. 2 include指令

程序5-4.jsp

```
<%@include file="3-1.jsp"%>
```

3.3 taglib指令

taglib指令是页面使用者用来自定义标签。可以把一些需要重复显示的内容自定义成为一个标签，以增加代码的重用程度，并使页面易于维护。

随着JSP语言规范的升级，标签库不断得到加强，它在页面中的定义如下：

```
<%@taglib uri="taglibURI" prefix="tagPrefix"%>
```

其中，uri用来表示标签描述符，也就是提供怎么知道标签描述文件和标签库的路径，tagPrefix定义了JSP页面里要引用该标签时的前缀，需要注意的是，这些前缀不可以是jsp、jspx、java、javax、sun、servlet和sunw 。

4 操作指令

jsp编译指令时让web服务自动采取的动作，如果想要控制jsp页面的运行，可以采用jsp的操作指令：

jsp:include指令、jsp:forward指令、jsp:param指令、jsp:useBean指令、jsp:setProperty指令、jsp:getProperty指令和jsp:plugin指令

等。

4.1 jsp:include指令

<jsp:include>允许在JSP页面中包含静态和动态页面。如果包含的是静态页面，则只是将静态页面的内容加入至JSP页面中，如果包含的是动态页面，则所包含的页面将会被JSP服务器编译执行。

语法格式为：

```
<jsp:include page="test.htm"/>
```

jsp:include指令必须以“/”结束，功能和include指令相同。

4.1 jsp:include指令

<jsp:include>操作的格式如下：

<jsp:include page="relativeURL|<%=expression%>" flush="true|false"/>

page：表示所要包含的文件的相对URL，它可以是一个字符串，也可以是一个JSP表达式。

flush：默认值为false，若该值为true则表示当缓冲区满时，缓冲区将被清空。

4.1 jsp:include指令

<jsp:include>与<%@include%>指令的区别？

功能上看不出任何区别。程序5-5.jsp

查看生成servlet的源文件知道生成的区别。

```
<%@ include file="/5-1.jsp"%>
out.write("document.write(\"<br>浏览器时间：
\"+dhours+\":\"+dminutes+\":\"+dseconds);\r\n");
```

包含的文件的内容，指令的方式包含

```
<jsp:include page="/5-1.jsp"/>
org.apache.jasper.runtime.JspRuntimeLibrary.include(r
equest, response, "/5-1.jsp", out, false);
```

是调用JspRuntimeLibrary.的include方法

4.2 jsp:forward指令

<jsp:forward>操作允许将当前的请求运行转发至另外一个静态的文件、JSP页面或含有与当前页面相同内容的Servlet。操作指令用于把当前的JSP页面转发到另一个页面上。

<jsp:forward>的格式如下：

```
<jsp:forward page="relativeURL|<%=expression%>" />
```

程序5-6.jsp

```
<jsp:forward page="5-1.jsp"></jsp:forward>
```

4.2 jsp:forward指令

<jsp:forward>操作允许将当前的请求运行转发至另外一个静态的文件、JSP页面或含有与当前页面相同内容的Servlet。操作指令用于把当前的JSP页面转发到另一个页面上。

<jsp:forward>的格式如下：

```
<jsp:forward page="relativeURL|<%=expression%>" />
```

程序5-6.jsp

```
<jsp:forward page="5-1.jsp"></jsp:forward>
```

4.3 jsp:include指令

<jsp:include>用来根据浏览器的类型，插入通过**Java**插件 运行**Java Applet**所必需的**Object**或**embed**元素。

如果需要的插件不存在，它会下载插件，然后执行**Java**组件。 **Java**组件可以是一个**applet**或一个**JavaBean**

<jsp:plugin>的格式如下：

```
<jsp:plugin  type="applet"   codebase="dirname"
code="MyApplet.class"    width="60" height="80">
    <jsp:fallback> Unable to initialize Java Plugin   </jsp:fallback>
</jsp:plugin>
```

4.4 jsp:param指令

<jsp:param>操作提供了“名称——值”信息，通常和<jsp:include>、<jsp:forward>一起使用，包含的页面或重定向的页面将看到新参数增加的原始request对象。可以按照“名字/值”的形式进行参数传递。

该操作若独立于<jsp:include>、<jsp:forward>这些操作将没有任何作用。

<jsp:param>操作的格式如下：

```
<jsp:param name="paramName" value="paramValue"/>
```

其中paramName表示参数名称，paramValue表示参数值。

程序5-7.jsp 和5-8.jsp

4.4 jsp:param指令

程序5-7.jsp

```
<jsp:forward page="/3-8.jsp">
<jsp:param name="param" value="web"/>
```

程序5-8.jsp

```
<%String aa=request.getParameter("param");%>
<%out.println(aa);%>
```

5 Java Web代码

脚本元素（Scripting Elements）是JSP代码中使用最频繁的元素，它是用Java写的脚本代码。所有的脚本元素均是以“<%”标记开始，以“%>”标记结束，它可以分为如下三类：

声明、表达式、Scriptlet

5.1 声明（变量和方法）

在JSP中，声明是用来定义在程序中使用的实体，它是一段Java代码，可以声明变量也可以声明方法，它以“<%!”标记开始，以“%>”标记结束，格式如下：

<%!declaration; [declaration;]…… %>

每个声明仅在一个JSP页面内有效，如果要想在每个页面中都包含某些声明，可将这些声明包含在一个JSP页面中，然后利用前面介绍的include指令将该页面包含在每个JSP页面中。

5.1 声明（变量和方法）

变量的类型可以是Java语言支持的任意数据类型，在编译的过程中，jsp页面被编译为一个类，这些变量就是页面类的成员变量。这些变量是共享的，任何一个用户的操作都会影响其他用户。

程序5-9.jsp

5.1 声明（变量和方法）

程序5-9.jsp 点击率

```
<%! int i=0;%>
 <% i++;
 out.println(i);
 %>
```

人访问网站。

5.2 表达式

表达式（Expression）以“<%=”标记开始，以“%>”标记结尾，中间的内容为Java一个合法的表达式，格式如下：

<%=expression%>

其中expression表示Java表达式。表达式在执行时会被自动转换为字符串，然后显示在JSP页面中

程序5-10.jsp

5.3 Scriptlet（代码块）

Scriptlet是以“<%”标记开始，以“%>”标记结尾的一段Java代码，它可以包含任意合乎Java语法标准的Java代码，格式如下：

```
<%
Java代码
%>
```

5. 3 Scriptlet（代码块）

一个java wb面可以有多个Scriptlet（代码块），将按照顺序执行，声明的是局部变量，只在当前页面有效。

```
<% int i=10; %>
<%if(time<12)
{%> How are you this morning? <%}else
{%> How are you this afternoon? <%}%>
```

程序5-11.jsp

Web语言程序设计

下　课！！