### JSP基础简介

1. jsp九大内置对象:  
    1、out 向客户端输出数据,字节流.如out.print(" dgaweyr");  
    2、request 接收客户端的http请求.  
    String getParameter(String name):得到表单参数名name的值.  
    String[] getParameterValues(String name):(得到String[]复选框时常用).  
    setAttribute(String name,Object obj):设置属性名为name,属性值为obj.  
    getAttribute(String name);得到属性值.  
    3、response:封装jsp产生的回应,然后发送到客户端以响应客户的请求.重 定向跳转任意界面.(服务器跳转)  
    addCookie(Cookie cookie):  
    sendRedirect("/wel.jsp"):跳转到指定页面  
    4、session:用于保存用户信息,跟踪用户行为,当前打开的浏览器内,多个页面 共享数据. session对象指的是客户端与服务器的一次会话,从客户连到服务 器的一个WebApplication开始,直到客户端与服务器断开连接为止.它是 HttpSession类的实例.  
    setAttribute(String name,Object obj):设置属性名为name,属性值为obj.  
    getAttribute(String name):得到属性值.  
    5、application对象:实现了用户间数据的共享,可存放全局变量.它开始于服 务器的启动,直到服务器的关闭,在此期间,此对象将一直存在;这样在用户的 前后连接或不同用户之间的连接中,可以对此对象的同一属性进行操作;在任 何地方对此对象属性的操作,都将影响到其他用户对此的访问.服务器的启动 和关闭决定了application对象的生命.它是ServletContext类的实例.  
    session,application,request的区别:  
    一个项目中session尽量少用几个,因为过多的session会影响程序的执行效 率.它主要用于保存登录信息(用户信息,权限,资源)即频繁使用的信息.  
    application: 用于多个浏览器之间共享数据,多个用户共享该对象,可以做计 数器.它的用法与session完全一样.  
    数据范围:  
    application(服务器关闭时失效)>session(浏览器关闭时失效)>request(只能用于两个跳转页面之间)  
      
    6、page对象代表jsp这个实体本身,即当前页面有效.相当于java中的this.  
    数据范围:page<session<application  
     7、exception:代表运行时的异常.  
    在会发生异常的页面加入指令:

<%@ page errorPage="处理错误的页面.jsp"%>  
 在处理异常的页面写入:

<%@ page isErrorPage="true"%>  
  8、pageContext对象 pageContext对象提供了对JSP页面内所有的对象 及名字空间的访问,也就是说他可以访问到本页所在的SESSION,也可以取本 页面所在的application的某一属性值,他相当于页面中所有功能的集大成者, 它的本类名也叫pageContext.  
  9、config jsp对应的servlet的配置,可以得到web.xml中的初使化参数.  
   
二、jsp七大动作:  
1、include 动态包含(分别编译):  
 用jsp:include动作实现

**<jsp: include page="included.jsp" flush="true" />**

flush属性: 用true ，表示页面可刷新。默认为false;  
 它总是会检查所含文件中的变化，适合用于包含动态页面，并且可以带参数。

2、useBean动作(jsp页面使用javaBean的第二种方式):  
**<jsp:useBean id="对象名" class="包名.类名" scope="作用范围(request/page/application/session)"/>**  
 作用域默认为page(本页面有效).

3、getProperty动作(name为useBean动作中的id).  
 从对象中取出属性值：

**<jsp:getProperty name="javaBean对象" property="javaBean对象属性名" />**

4、setProperty动作(name为useBean动作中的id):  
 为对象设置属性值：

<jsp:setProperty name="javaBean对象" property="javaBean对象属性名"value=http://www.hake.cc/kf/201109/"值"/>  
 为对象设置属性值:

<jsp:setProperty property="javaBean对象属性名" name="javaBean对象" param="username"/>  
(param="username" 相当于 value=http://www.hake.cc/kf/201109/<%=request.getParameter("username")%>)  
   
5、param动作(传递参数)):  
 到达跳转页面可以通过 request.getParameter(“参数名”)方式取出参数值  
 <jsp:include page="转向页面的url" >  
 <jsp:param name="参数名1" value=http://www.hake.cc/kf/201109/"参数值1">

<jsp:param name="参数名2" value=http://www.hake.cc/kf/201109/"参数值2">  
           ...........  
  </jsp:include>  
**或:**  
 <jsp:forward page="转向页面的url" >

<jsp:param name="参数名1" value=http://www.hake.cc/kf/201109/"参数值1">  
 <jsp:param name="参数名2" value=http://www.hake.cc/kf/201109/"参数值2">  
           ...........  
 </jsp:forward>  
   
6、forward动作:  
 跳转页面:

<jsp:forward page="login.jsp" />

7、plugin动作:<jsp:plugin>:用于指定在客户端运行的插件  
   
三、JSP三大指令之:  
1、page指令:  
 指定页面编码.例:  
<%@pagelanguage="java"contentType="text/html;charset=gbk"pageEncoding="gbk" %>  
 导入包,例:  
<%@ page import="java.util.\*,java.text.\*" %>.

2、include 指令  
 静态包含(统一编译):

<%@ include file="included.jsp"%>  
   
3、**taglib指令：声明用户使用的自定义标签：**

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

声明用户使用自定义的标签，将标签库描述符文件导入到jsp页面：

taglib指令语法：

**<%@ taglib (uri="tigLibURL" 或 tagDir="tagDir") prefix="tagPrefix" %>**

uri属性:定位标签库描述符的位置。唯一标识和前缀相关的标签库描述符，可以使用绝对 或相对URL。

tagDir属性：指示前缀将被用于标识在WEV-INF/tags目录下的标签文件。

prefix属性：标签的前缀，区分多个自定义标签。不可以使用保留前缀和空前缀，遵循XML 命名空间的命名约定。

注意：Jsp中的静态包含与动态包含  
动态include(<jsp: include page="included.jsp"/>)  
静态include(<%@ include file="included.jsp"%>)  
1. 静态include的结果是把其他jsp引入当前jsp,两者合为一体,可以达到数据的共享即可以说是统一编译的,而  
动态include的结构是两者独立的,直到输出时才合并即为分别编译的.  
2.动态include的jsp文件独立性很强,是一个单独的jsp文件,需要使用的对象,页面设置,都由自己创建,而静态include纯粹是把代码写在外面的一种共享方法,所有的变量都是可以和include它的主文件共享,两者高度紧密结合,不能 有变量同名的冲突.而页面设置也可以借用主文件的.  
3.动态包含总是检查被包含页面的变化,静态包含不一定检查被包含页面的变化.  
4.动态包含可带参数,静态包含不能带参数.如(<jsp: include page="included.jsp">放入参数</jsp:include>);