

## jsp的6个动作

使用标签的形式来表示一段java代码

<jsp:include page="uri"></jsp:include>

<jsp:include>动态包含

<jsp:forward>请求转发

<jsp:param>设置请求参数

<jsp:useBean>创建一个对象

<jsp:setProperty>给指定的对象属性赋值

<jsp:getProperty>给指定的对象属性赋值

- 1, <jsp:useBean id="stu1" class="com.entity.student">  
    </jsp:useBean>
- 2, <jsp:setProperty property="name" name="stu1" value="jerry"/>
- 3, <jsp:getProperty property="name" name="stu1"/>

## jsp的9个内置对象

是指在jsp的<%=%>和<%%>中可以直接使用的对象

- 1, request      javax.servlet.http.HttpServletRequest
- 2, response      javax.servlet.http.HttpServletResponse
- 3, session      javax.servlet.http.HttpSession 由session= "true" 开关
- 4, application      javax.servlet.ServletContext
- 5, exception      java.lang.Throwable 由isErrorPage= "true" 开关
- 6, page      java.lang.Object 当前对象this      当前servlet实例
- 7, config      javax.servlet.ServletConfig
- 8, out      javax.servlet.jsp.JspWriter      字符输出流, 相当于printWriter
- 9, pagecontext      javax.servlet.jsp.pageContext

pagecontext 重要(只在当前页面有效)

一,

- 1, 本身也是一个域对象, 它可以操作其他三个域对象 (request, session, application) 的数据
- 2, 通过这个对象可以得到其他八个对象
- 3, 只在当前页面有效

```
//操作自己
pageContext.setAttribute("name", "tom");
String string = (String)pageContext.getAttribute("name");
pageContext.removeAttribute("name");
//操作其他对象
pageContext.setAttribute("name","name",int scope);
String str = (String)pageContext.getAttribute("name", int scope);
pageContext.removeAttribute("name",int scope);
//scope的取值:
pageContext.PAGE_SCOPE;
pageContext.REQUEST_SCOPE;
pageContext.SESSION_SCOPE;
pageContext.APPLICATION_SCOPE;
```

最重要:

*pagecontext.findAttribute(String name);自动从page request session application  
依次查找, 找到了就取值, 结束查找, 如果没有找到就返回null*

重定向不能取到request中的值, 但是可以取到session中的值

二, 它可以创建其他的8个隐式对象

在普通类中可以通过pageContext获取其他jsp隐式对象, 自定义标签时就使用

三, 提供了的简易方法

```
pageContext.forward("2.jsp");
pageContext.include("2.jsp");
```