

常用标签解释

property标签

- property标签用于输出指定值:
- `<s:set name="name" value="kk" />`
- `<s:property value="#name"/>`
- **default:** 可选属性, 如果需要输出的属性值为null, 则显示该属性指定的值
- **escape:** 可选属性, 指定是否格式化HTML代码。
- **value:** 可选属性, 指定需要输出的属性值, 如果没有指定该属性, 则默认输出ValueStack栈顶的值。
- **id:** 可选属性, 指定该元素的标识

iterator标签

- **iterator**标签用于对集合进行迭代，这里的集合包含**List**、**Set**和数组。这个标签类似于**JSTL**的**forEach**标签
- 常用属性：
- **value**: 可选属性，指定被迭代的集合，如果没有设置该属性，则使用**ValueStack**栈顶的集合。
- **var**: 可选属性，指定接收集合里元素的变量名。
- **status**: 可选属性，该属性指定迭代时的**IteratorStatus**实例。该实例包含如下几个方法：
 - `int getCount()`，返回当前迭代了几个元素。
 - `int getIndex()`，返回当前迭代元素的索引。
 - `boolean isEven()`，返回当前被迭代元素的索引是否是偶数
 - `boolean isOdd()`，返回当前被迭代元素的索引是否是奇数
 - `boolean isFirst()`，返回当前被迭代元素是否是第一个元素。
 - `boolean isLast()`，返回当前被迭代元素是否是最后一个元素。

iterator标签应用举例

```
<%  
ArrayList<Student> list = new ArrayList<Student>();  
list.add(new Student(101,"tom"));  
list.add(new Student(102,"jack"));  
list.add(new Student(103,"rose"));  
list.add(new Student(104,"john"));  
request.setAttribute("list",list);  
%>  
<s:iterator value="#request.list" var="stu" status="st">  
    遍历的第<s:property value="#st.count"/>个,  
    索引: <s:property value="#st.index"/>,  
    偶数: <s:property value="#st.even"/>,  
    奇数: <s:property value="#st.odd"/>,  
    首个: <s:property value="#st.first"/>,  
    末个: <s:property value="#st.last"/>,  
    <s:property value="#stu.sid"/>,  
    <s:property value="#stu.sname"/><br>  
</s:iterator>
```

if/elseif/else标签

- 和jstl里的if标签差不多

```
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
```

```
<%@ taglib uri="/struts-tags" prefix="s"%>
```

```
<%
```

```
request.setAttribute("score",58);
```

```
%>
```

```
<s:if test="#request.score>70">
```

高分

```
</s:if>
```

```
<s:elseif test="#request.score>50">
```

中分

```
</s:elseif>
```

```
<s:else>
```

低分

```
</s:else>
```

表单实例

- `<s:form action="Login" namespace="/test">`
- `<s:textfield name="username"/>
`
- `<s:password name="password"/>
`
- `<s:submit value="登陆"/>`
- `</s:form>`

单行文本框，密码框，和多行文本框

文本框：<s:textfield name="username"></s:textfield>

密码框：<s:password name="pwd"></s:password>

文本区域：

<s:textarea name="info" cols="50" rows="7" value="这家伙很懒"></s:textarea>

单选按钮和复选按钮

<!--

创建一组4个复选按钮，使用List集合按钮文字和按钮value值都相同
默认选中，VB,PB

-->

```
<s:checkboxlist name="skill" list="{ 'JAVA','VB','C#','PB' }" value="{ 'VB','PB' }"/>
```

<hr>

<!--功能同上，只不过数据用OGNL表达式设置了动态数据 -->

<%

```
String[] list = { "JAVA", "VB", "C#", "PB" };
```

```
String[] value = { "JAVA", "PB" };
```

```
request.setAttribute("list", list);
```

```
request.setAttribute("value", value);
```

%>

```
<s:checkboxlist name="skill2" list="#request.list" value="#request.value"/>
```


单选按钮和复选按钮

<!--

创建一组4个复选按钮，使用MAP集合按钮文字和按钮value值分开设置
默认选中JAVA和C#

-->

<s:checkboxlist name="skill3" list="#{1:'JAVA',2:'VB',3:'C#',4:'PB'}" value="{1,3}"/>

<hr>

<!--功能同上，只不过数据用OGNL表达式设置了动态数据 -->

<%

```
Map<Integer,String> map = new HashMap<Integer,String>();
map.put(100,"TOM");
map.put(101,"JACK");
map.put(102,"ROSE");
map.put(103,"JOHN");
request.setAttribute("map",map);
```

%>

<s:checkboxlist name="skill4" list="#request.map" value="{100,102}"/>

单选按钮和复选按钮

<!--

创建一组复选按钮，数据采用JavaBean对象的属性值

下例按钮文字使用学生姓名，按钮value值使用学生编号,默认选中103学生

-->

<%

```
ArrayList<Student> stuList = new ArrayList<Student>();  
stuList.add(new Student(100,"tom"));  
stuList.add(new Student(101,"jack"));  
stuList.add(new Student(102,"rose"));  
stuList.add(new Student(103,"john"));  
request.setAttribute("stuList",stuList);
```

%>

```
<s:checkboxlist name="skill5" list="#request.stuList" listKey="sid" listValue="sname"  
value="{103}"/>
```

<hr>

<!--创建一组单选按钮，使用上例JavaBean数据，默认选中101学生 -->

```
<s:radio name="skill6" list="#request.stuList" listKey="sid" listValue="sname" value="101"/>
```

列表菜单

```
<!--利用Map集合产生一个列表，键作为value值，值作为显示文字-->
<%
    Map<Integer,String> map = new HashMap<Integer,String>();
    map.put(100,"TOM");
    map.put(101,"JACK");
    map.put(102,"ROSE");
    map.put(103,"JOHN");
    request.setAttribute("map",map);
%>
<s:select name="select1" list="#request.map"></s:select>
<hr>
<!--利用一个JavaBean的list集合产生一个列表
sid作为value值，sname作为显示文字，默认选中103学生-->
<%
    ArrayList<Student> stuList = new ArrayList<Student>();
    stuList.add(new Student(100,"tom"));
    stuList.add(new Student(101,"jack"));
    stuList.add(new Student(102,"rose"));
    stuList.add(new Student(103,"john"));
    request.setAttribute("stuList",stuList);
%>
<s:select name="select2" list="#request.stuList" listKey="sid" listValue="sname" value="103"></s:select>
```