**我们自定义了一个带属性的标签，并使用它完成了一个简单的案例。**

**其实到这我们已经可以看出来，前端jsp页面只需要写一个类似html语法的标签，就可以完成将集合中的数据取出来并展示这么相对复杂的操作。**

**这就是我们使用标签的意义。接着我们看看怎么自定义一个带有标签体的标签。**

**我们可以利用标签体来简化我们上一个案例中的标签处理类。**

**<%**

**HashMap<String,Integer> maps = new HashMap<String, Integer>();**

**maps.put("李四",53);**

**maps.put("张三",23);**

**maps.put("walker",22);**

**pageContext.setAttribute("map",maps);**

**%>**

**<table>**

**<mytag:hello map="map">**

**<tr>**

**<td>${name}</td>**

**<td>${age}</td>**

**</tr>**

**</mytag:hello>**

**</table>**

**<tag>**

**<description>Outputs a colored tile</description>**

**<name>hello</name>**

**<tag-class>Test.MyTag</tag-class>**

**<body-content>scriptless</body-content>**

**<attribute>**

**<name>map</name>**

**<required>false</required>**

**<fragment>true</fragment>**

**</attribute>**

**</tag>**

**public void doTag() throws JspException, IOException {**

**HashMap<String,Integer> maps = (HashMap<String,Integer>)(getJspContext().getAttribute(map));**

**for (String str : maps.keySet()){**

**getJspContext().setAttribute("name",str);**

**getJspContext().setAttribute("age",maps.get(str));**

**getJspBody().invoke(null);**

**}**

**}**

**首先看jsp页面，有标签体的标签使用的时候是需要有开始标签和结束标签的，这一点是和html很像的。**

**至于td元素中的内容，这是一个EL表达式，这里的意思就是在当前页面寻找共享数据名为name和age的数据，找到就获取其值，否则为“”。**

**tld文件中的改动不多，就是将body-content的值改动成scriptless，这表示标签体可以是静态的html，但是不能是jsp脚本。而我们之前一直是empty，它指定该标签是没有标签体的。**

**getJspBody()表示获取整个标签体的所有内容，返回的是一个fragment对象，这个对象的一个方法invoke就是用于输出整个内容到jsp页面，如果参数为null表示直接输出。**

**总结:可以自定义的标签形式还有很多,比如以页面片段为属性的标签,动态属性标签,有父标签的标签等等,这些有兴趣的同学可以中百度中查找来!真正项目需要的时候再查查文档后在做!**

**最后还是要强调一点，属性的的只能使基本的数据类型，对于一些复杂的类型 ，例如Date，等，建议将该对象置于本页共享范围，然后标签处理类可以直接获取并做相应的处理。**