JSP标记

**0.JSP标记的三大内容都属于脚本代码（scriptlet、表达式、声明（注意不是指令）），现代的规范是消除掉JSP中的脚本代码，但我们也要学习，这样才知道那些替代者们的原理。**

**1.<% %> scriptlet**

用来放Java代码

**2.<%= %> 表达式**

表达式元素会自动打印放在标记中间的内容（等同于out.println(….)）

eg:<% out.println(Counter.getCount()); %> 相当于

<%= Counter.getCount() %> （没有分号是因为这里是取变量值打印）

表达式中间的参数永远是out.print()中的参数

不能把一个返回类型为void的方法作为表达式

**3.<%@ %> 指令（page、include、taglib）**

在页面转换时，向容器发出指示

eg:<%@ page import=”foo.\*” %> （指令同样没有分号）

**4.JSP中的变量声明**

1）尝试用<% %>声明变量

<% int count=0; %>

<%= ++count %>

如此，按我们设想，每次刷新JSP页面count应该+1，但是无论刷新多少次，始终显示1。

于是我们就到jsp转化成的servlet中去看，发现，int count声明的变量是在一个叫做\_jspService的方法中（类似于doGet和doPost的结合），而我们知道每一个请求容器都会分配一个单独的线程来处理，而且每一个请求在发出响应后就不可用了，因此，count作为局部变量是在下一个线程使用不了的。

为了实现，我们必须要声明一个全局变量才行（实例变量）。

2）因此，scriptlet中的代码只会原样放在 \_jspService中。

我们可以使用另一个声明方法，**用于声明JSP元素（全局的，包括变量和方法）**

**<%! %>**

eg:<%! int count=0; %>

这样声明后，count就是jsp对应servlet的全局变量；

而这也可以用来声明全局方法。

<%! int getMutiCount(){

count = count\*2;

return count;

}

%>

5.JSP所谓的隐式对象，大部分就在\_jspServive中声明了一个变量

6.<% %>声明在\_jspService中，局部

<%! %>声明在JspServlet中 ，全局