



北京圣思园科技有限公司  
<http://www.shengsiyuan.com>

主讲人：张龙



# JavaServer Pages (JSP)

深入学习

# JavaServer Pages (JSP)

- 3.1 Java基础知识
- 3.2 JSP语法
- 3.3 JSP内置对象
- 3.4 HTML表单
- 3.5 JSP与客户机的交互
- 3.6 实践：练习JSP编程





# JSP简介

- 在传统的网页HTML文件 (\*.htm, \*.html) 中加入Java程序片段 (Scriptlet) 和JSP标签, 就构成了JSP网页
- Java程序片段可以操纵数据库、重新定向网页以及发送E-mail等, 实现建立动态网站所需要的功能。
- 所有程序操作都在服务器端执行, 网络上传送给客户端的仅是得到的结果, 这样大大降低了对客户浏览器的要求, 即使客户浏览器端不支持Java, 也可以访问JSP网页。



# 复习JSP的概念

- **Java Server Pages**
- **Servlet**简化设计，逻辑与界面设计分开，开发更方便；
- **HTML**语法的 **Java** 扩展，加入新的标签 (`<% , %>, ...`)；
- 强大的组件 (**Java Bean**) 支持功能；



# JSP文件结构及主要标签

```
<%@ page contentType="text/html;charset=gb2312" %>
```

```
<%@ page import="java.util.*" %>
```

```
...
```

```
<HTML>
```

```
<BODY>
```

其他 HTML 语言

```
<%
```

符合JAVA语法的 JAVA 语句

```
%>
```

其他 HTML 语言

```
</BODY>
```

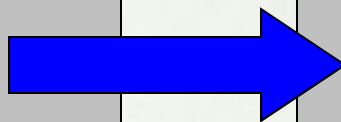
```
</HTML>
```



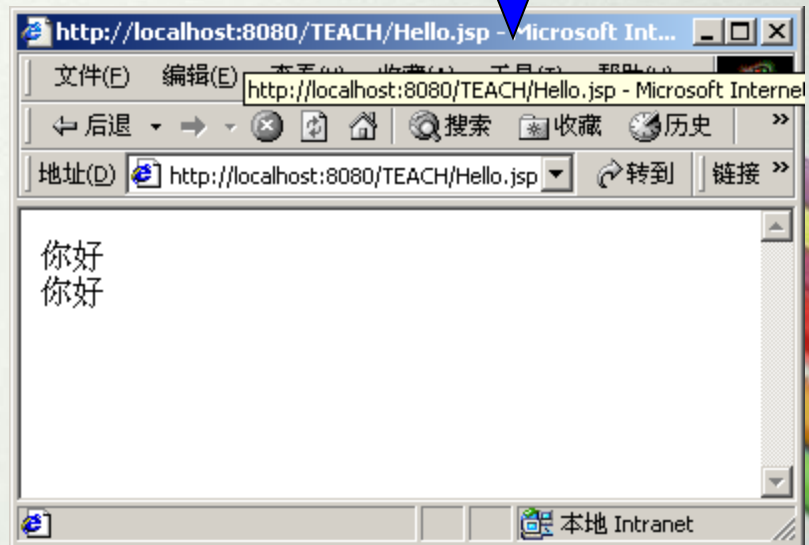
# JSP示例

```
<%@ page contentType="text/html;charset=gb2312" %>
<%@ page import="java.util.*"%>
```

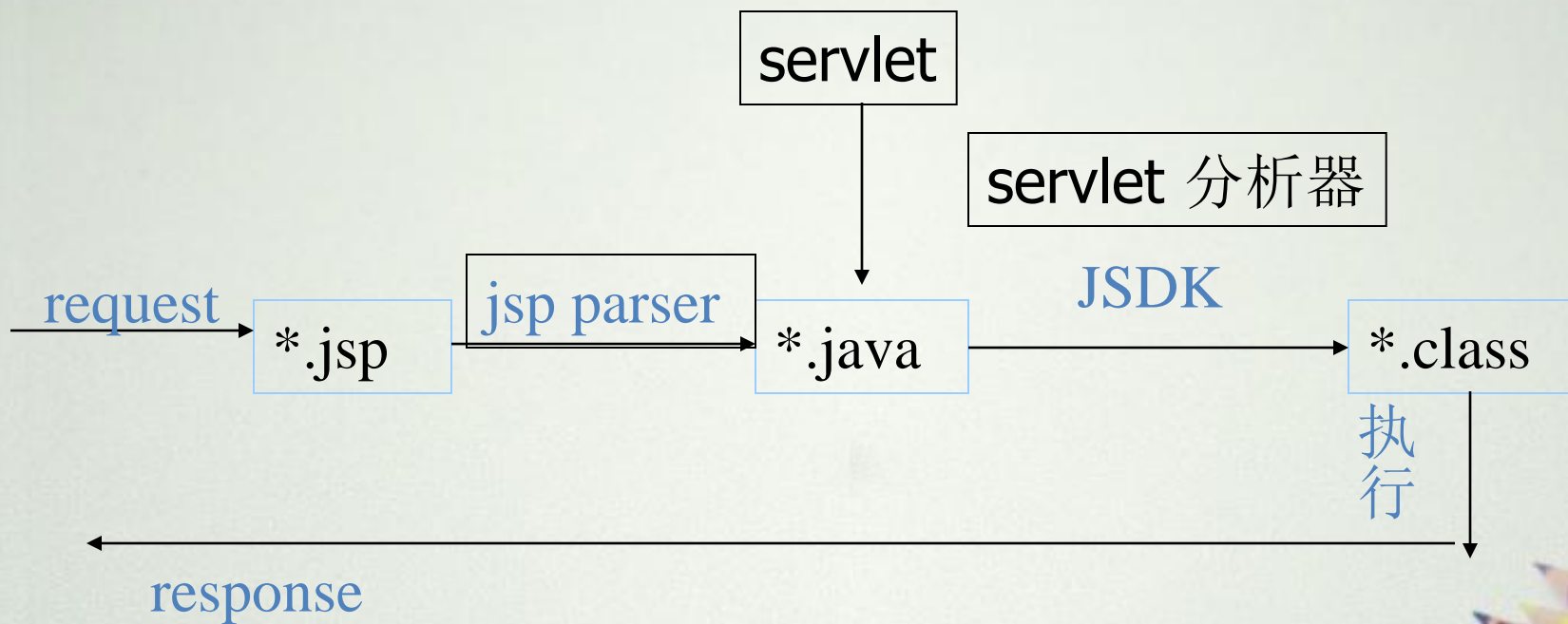
```
<HTML>
<BODY>
<%
for ( int i=0; i<2; i++ )
{
%>
    你好<br>
<%
}
%>
</BODY>
</HTML>
```



```
<HTML>
<BODY>
    你好<br>
    你好<br>
</BODY>
</HTML>
```

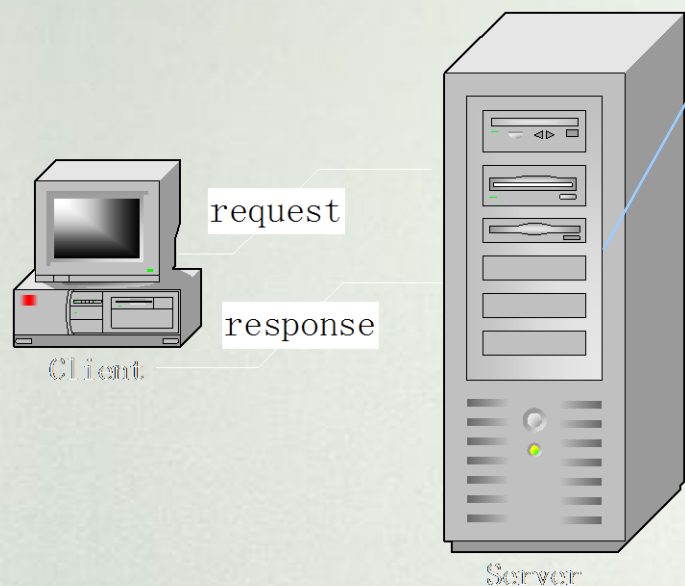


# JSP执行过程





## • 执行过程



查找对应的JSP文件

是否存在

Y

N

是否是修改或创建  
后第一次调用

Y

N

调用Jsp Parser将其  
编译成Servlet程序

调用JSDK将对应的Servlet  
程序编译成Servlet字节码

执行(若未载入则先载入)  
已有的对应的Java字节码

# JSP示例(Hello\_Date.jsp)

```
<%@ page import="java.util.*" %>
<%@ page contentType="text/html;
      charset=gb2312" %>
<HTML>
  <BODY>
    你好,今天是
  <%
    Date today=new Date();
  %>
    <%=today.getDate()%>号,
    星期<%=today.getDay()%>
  </BODY>
</HTML>
```

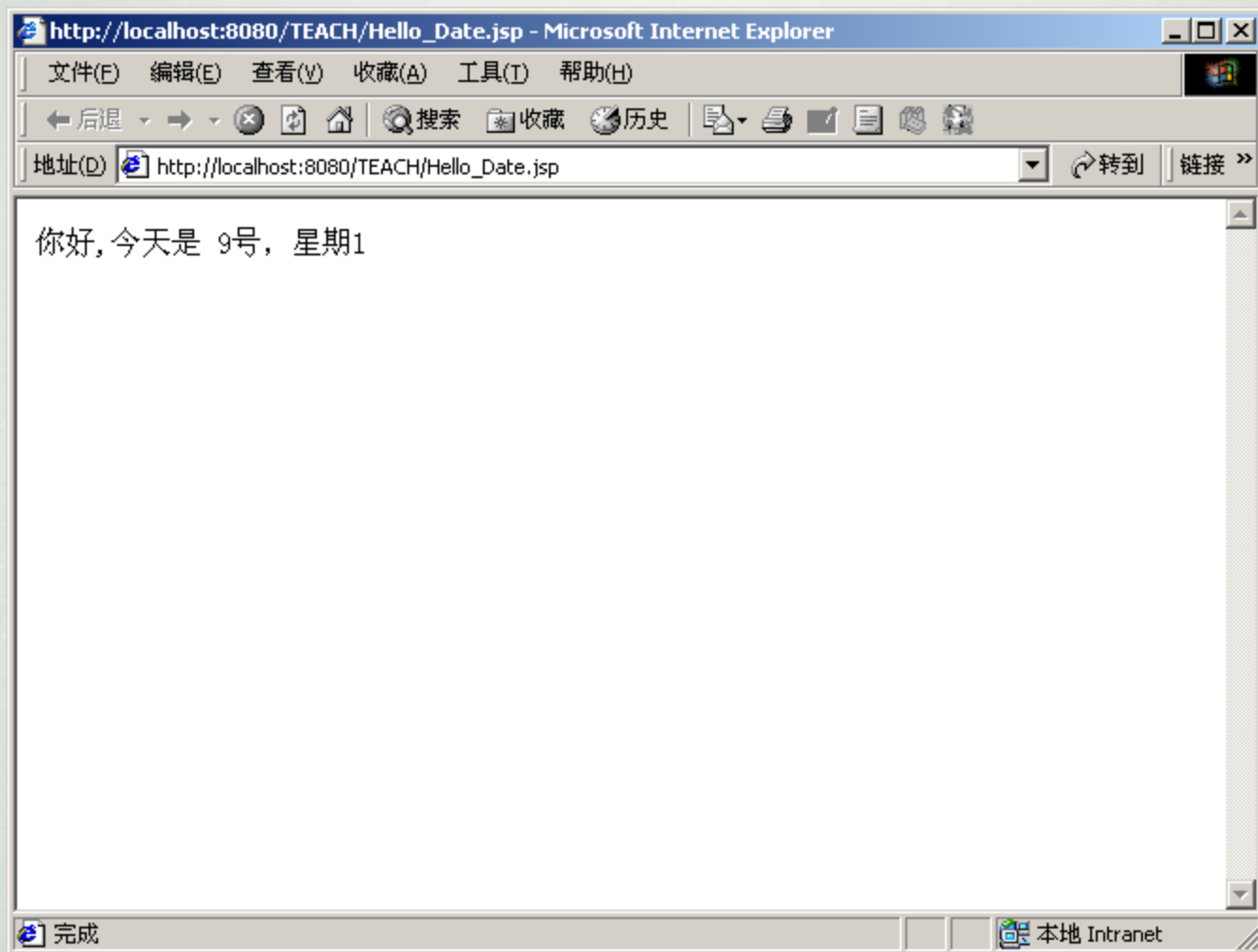
JSP文件

```
import java.util.*;

response.setContentType("text/html;
      charset=gb2312");
out = pageContext.getOut();
out.write("\r\n\r\n<HTML>\r\n
      <BODY>\r\n你好,今天是\r\n");
Date today=new Date();
out.print(today.getDate());
out.write("号, 星期");
out.print(today.getDay());
out.write(" \r\n </BODY>\r\n</HTML>\r\n ")
```

servlet文件

# JSP示例(浏览结果)



# JSP语法概述

- JSP原始代码中包含了JSP元素和Template(模板) data两类
- Template data指的是JSP引擎不处理的部分，即标记<%.....%>以外的部分，例如代码中的HTML的内容等，这些数据会直接传送到客户端的浏览器
- JSP元素则是指将由JSP引擎直接处理的部分，这一部分必须符合JAVA语法，否则会导致编译错误。





# jsp例子

- ```
<html>
<head>
<title>Hi-JSP实验</title>
</head>
<body>
<%
String msg = "This is JSP test.";
out.print("Hello World!");
%>
<h2><%=msg%></h2>
</body>
</html>
```

Template data

JSP元素



# JSP语法

- JSP语法分为三种不同的类型

- ▶ 编译器指令 (DIRECTIVE) 例如: `<%@ page import="java.io.*" %>`

- ▶ 脚本语法 (SCRIPTING)

- ▶ 动作语法 (ACTION) 例如: `<jsp:forward>`, `<jsp:getProperty>`, `<jsp:include>`



## 脚本语法

- 脚本语法包括

- “HTML注释” : `<!-- comments -->`
- “隐藏注释” : `<%-- comments --%>`
- “声明”
- “表达式”
- “脚本段”



## 编译器指令

- 编译器指令包括“包含指令”，“页指令”和“taglib指令”
- 它们包含在“<%@ %>”卷标里。
- 两个主要的指令是page与include。





## 动作语法

- 动作语法包括 `<jsp:forward>`,  
`<jsp:include>`, `<jsp:getProperty>`,  
`<jsp:setProperty>` 和 `<jsp:useBean>`。



# HTML注释

- JSP 语法格式如下:
- `<!-- comment [ <%= expression %> ] -->` 或
- `<!-- 注释 [<%= 表达式 %> ] -->`
- 这种注释发送到客户端，但不直接显示，在源代码中可以查看到。



# HTML注释(peixun2.1.jsp)

```
<html>
```

```
<head>
```

```
<title> HTML注释 </title>
```

```
</head>
```

```
<body>
```

```
    <!-- This file displays the user login screen --  
    >
```

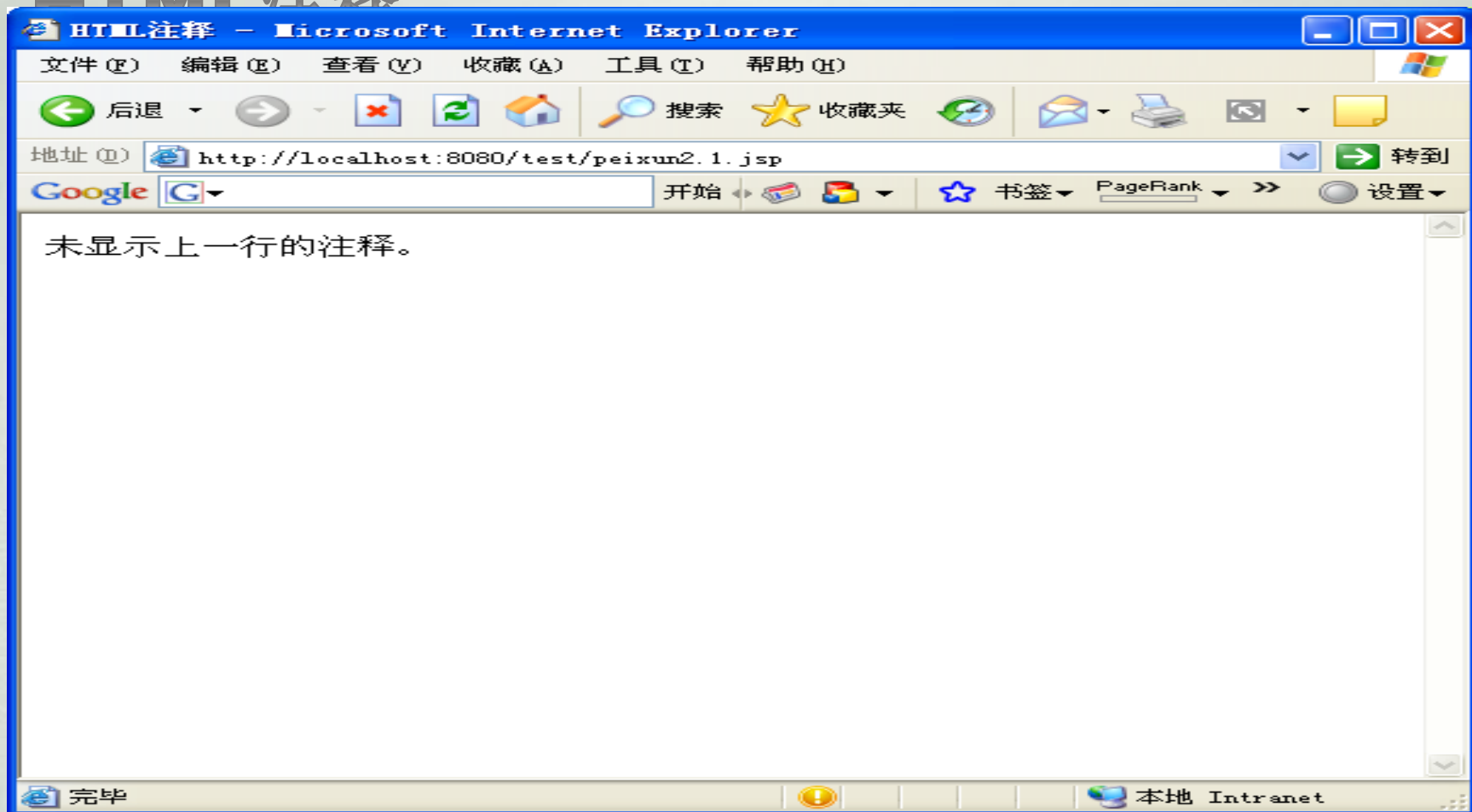
未显示上一行的注释。

```
</body>
```

```
</html>
```



# HTML 注释





# HTML注释(peixun2.2.jsp)

```
<html>
```

```
<head>
```

```
<title>要多加练习</title>
```

```
</head>
```

```
<body>
```

```
<!--This      page      was      loaded      on      <%=      (new  
    java.util.Date()).toLocaleString() %>  -->
```

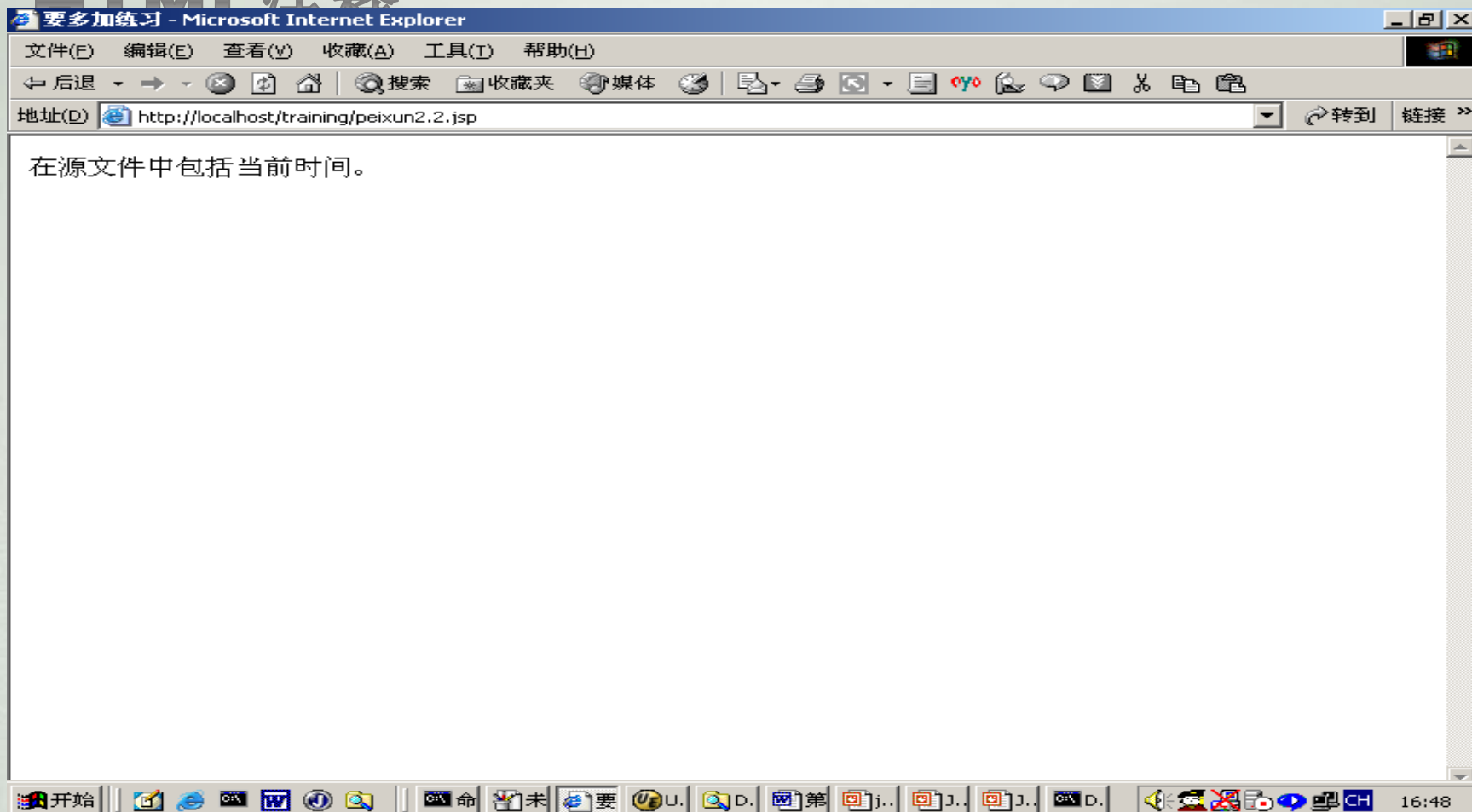
在源文件中包括当前时间。

```
</body>
```

```
</html>
```



# HTML 注释



## 隐藏注释

- JSP 语法格式如下：  
    <%-- 注释 --%>
- 不发送到客户端。



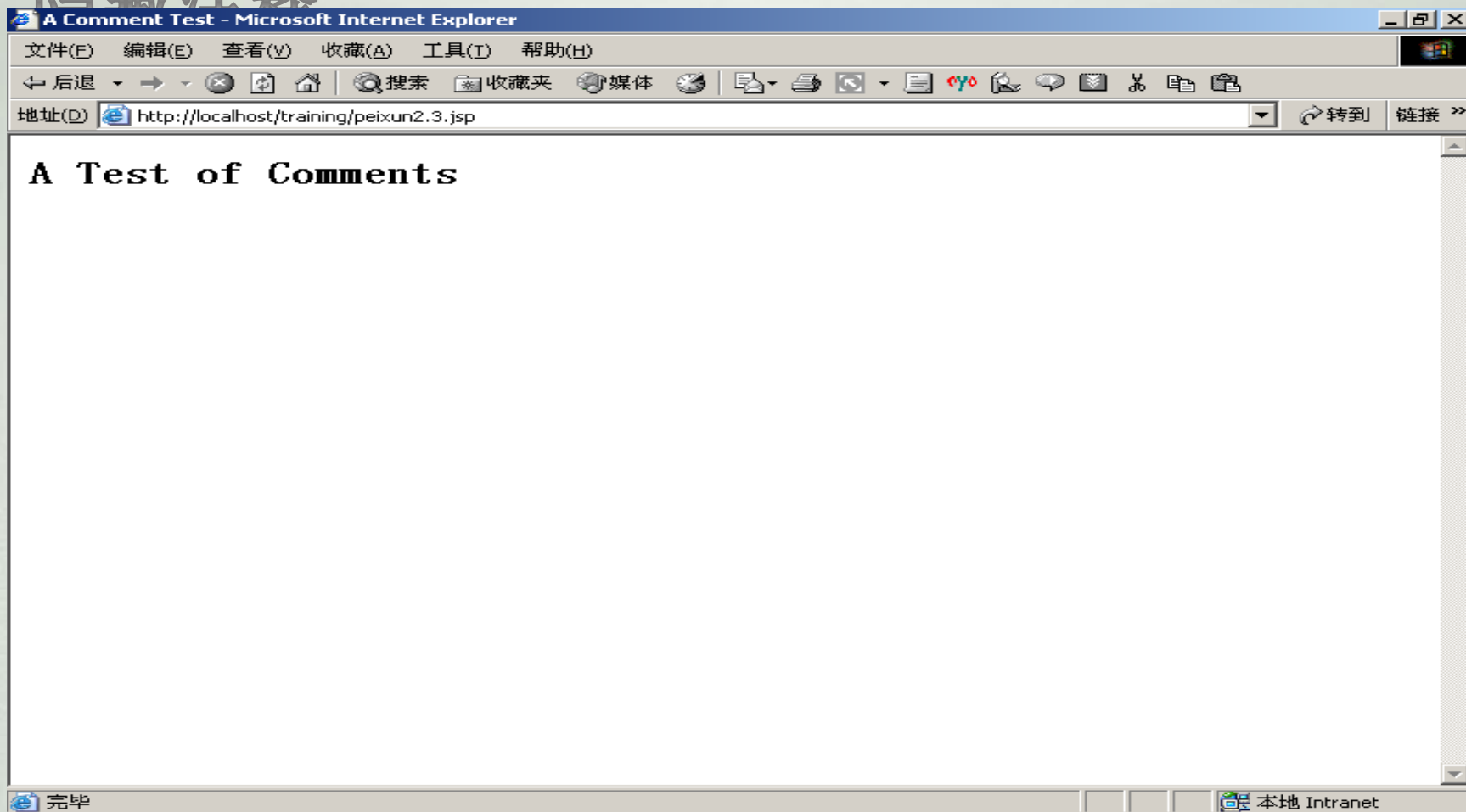
## 隐藏注释

```
<html>  
<head>  
<title>A Comment Test</title>  
</head>  
<body>  
<h2>A Test of Comments</h2>  
<%-- This comment will not be visible in the  
    page source --%>  
</body>  
</html>
```





陷藏汁察



# 在IE中查看源代码



```
<html>
<head>
<title>A Comment Test</title>
</head>
<body>
<h2>A Test of Comments</h2>

</body>
</html>
```

# 声明

- JSP 语法格式如下:
- `<%! declaration; [ declaration; ] ... %>` 或
- `<%! 声明; [声明; ] ... %>`



## 声明(peixun2.4.jsp)

```
<%@ page language="java" import="java.util.*"  
    %>  
<html>  
<head>  
<title> test2.4.jsp </title>  
</head>  
<body>  
<%! int i = 0; %>  
<%! int a, b, c; %>  
<%! Date date; %>  
</body>  
</html>
```



# 表达式


- 用于在页面上输出信息，语法格式如下：
- `<%= expression %>` 或
- `<%= 表达式 %>`





# 表达式(peixun2.4.jsp)

```
<%@ page language="java"
        import="java.util.*" %>
<html>
<head>
<title> test </title>
</head>
<body>
<center>
<%! Date date=new Date(); %>
<%! int a, b, c; %>
```



## 表达式

```
<% a=12;b=a; c=a+b;%>
```

```
<font color="blue">
```

```
<%=date.toString()%>
```

```
</font> <br>
```

```
<b>a=<%= a %></b><br>
```

```
<b>b=<%= b %></b><br>
```

```
<b>c=<%= c %></b><br>
```

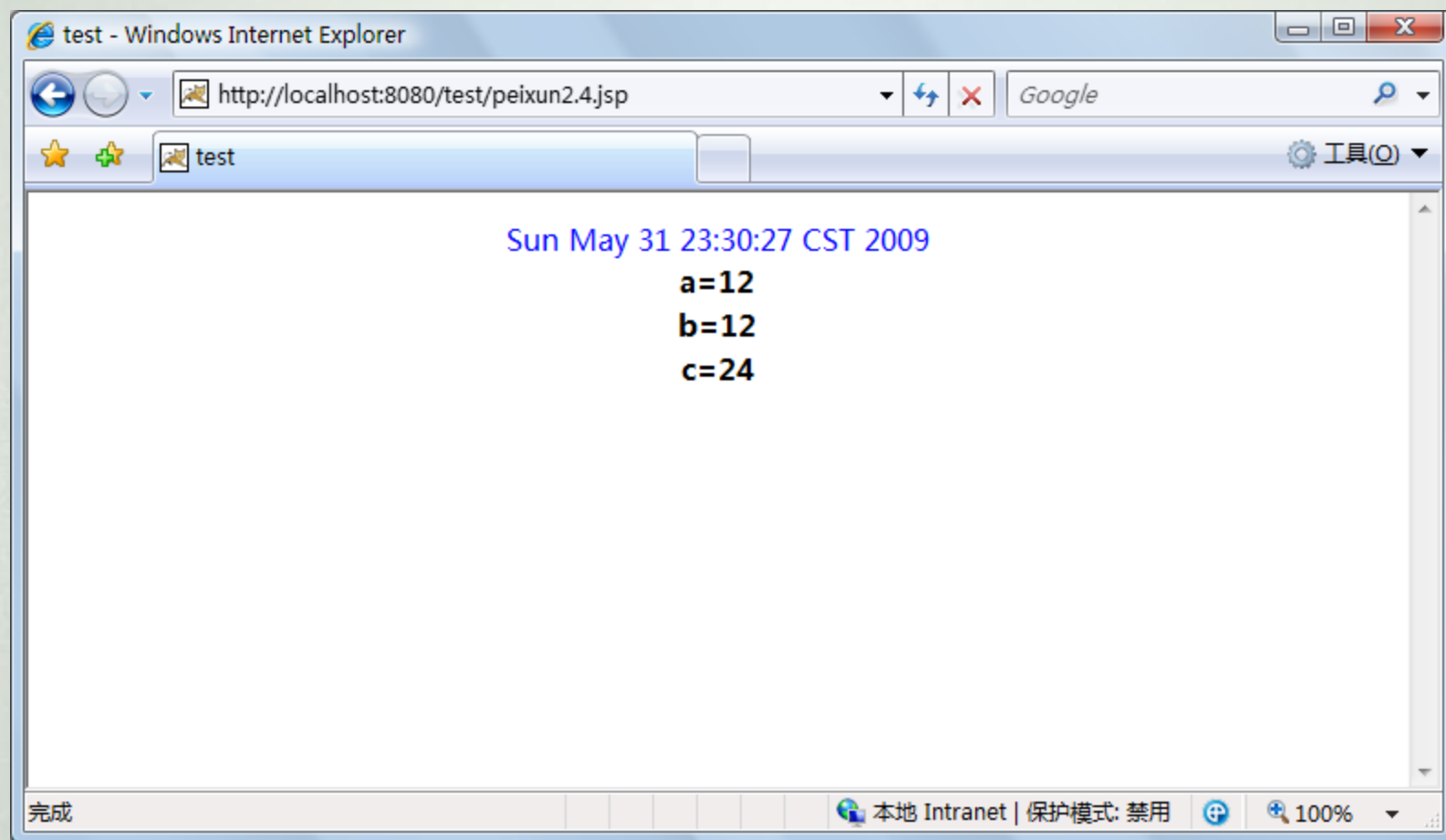
```
</center>
```

```
</body>
```

```
</html>
```



# peixun2.4.jsp执行结果



## 脚本段

- JSP 语法格式如下：
- `<% code fragment %>` 或
- `<% 代码 %>`



## 脚本段 (peixun2.5.jsp)

```
<%@ page language="java" import="java.util.*"  
    %>
```

```
<%! int condition;%>
```

```
<html>
```

```
    <head>
```

```
        <title> test </title>
```

```
    </head>
```

```
    <body>
```





# 脚本段 (peixun2.5.jsp)

```
<%
```

```
    condition=1;
```

```
    switch(condition){
```

```
        case 0:
```

```
            out.println("You must select condition 0!"+"<br>");
```

```
            break;
```

```
        case 1:
```

```
            out.println("You must select condition 1!"+"<br>");
```



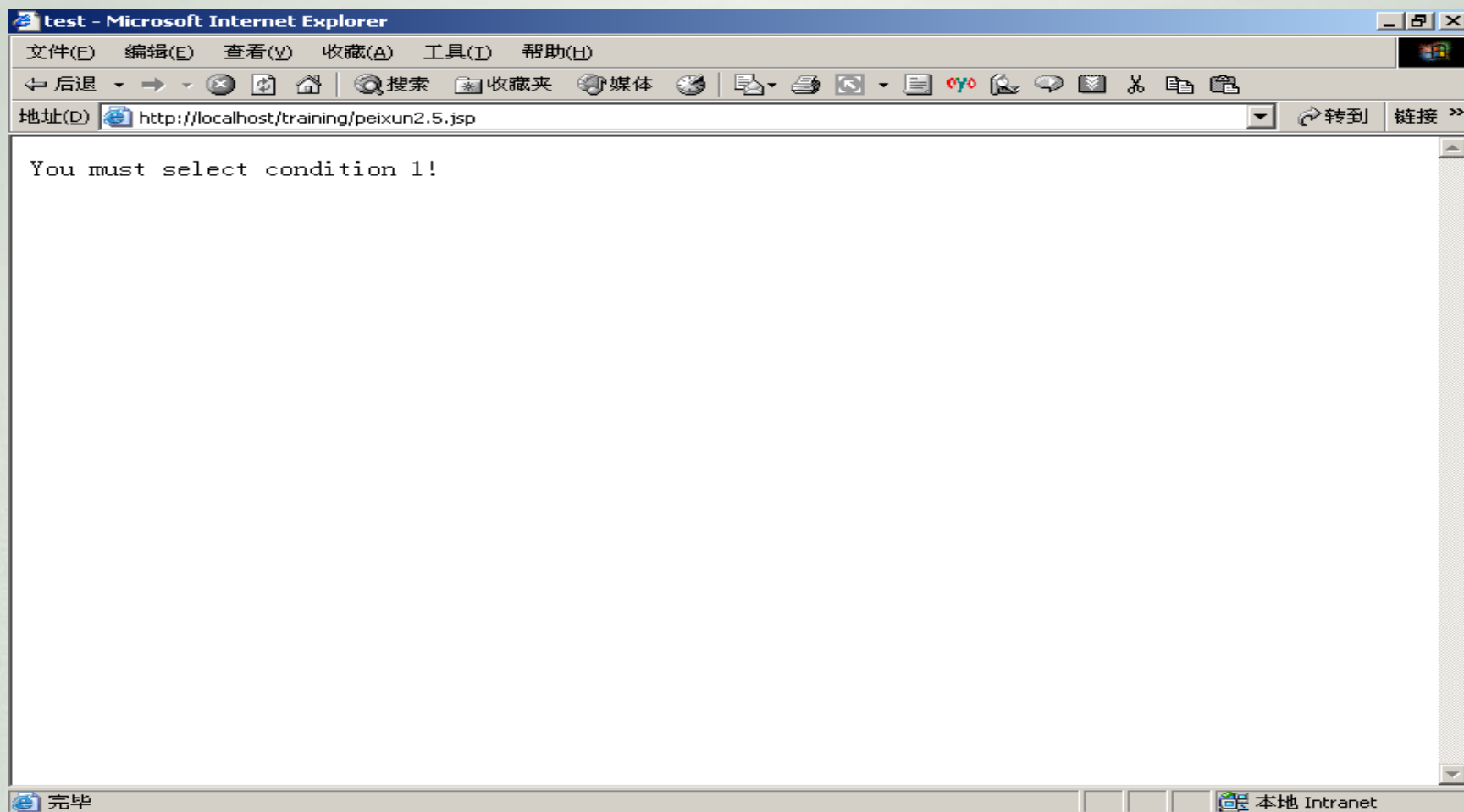
## 脚本段 (peixun2.5.jsp)

```
break;
    case 2:
        out.println("You must select condition 2!"+"<br>");
        break;
    default:
        out.println("Your      select      not      in      \"0,1,2\",select
again!!"+"<br>");
    }
%>
</body>
</html>
```

转义符：双引号 \" ，单引号\'，反斜杠\\，回车\r，换行\n，制表符\t，  
退格 \b



# 脚本段执行结果



## 包含指令

- include指令：向当前页中插入一个静态文件的内容。
- JSP 语法格式如下：
- `<%@ include file="relativeURL" %>` 或
- `<%@ include file="相对位置" %>`



## 包含指令(peixun2.6.jsp)

```
<html>
```

```
  <head>
```

```
    <title>test</title>
```

```
  </head>
```

```
  <body bgcolor="white">
```

```
    <font color="blue">
```

The current date and time are

```
<%@ include file="peixun2.6.1.jsp" %>
```

```
    </font>
```

```
  </body>
```

```
</html>
```





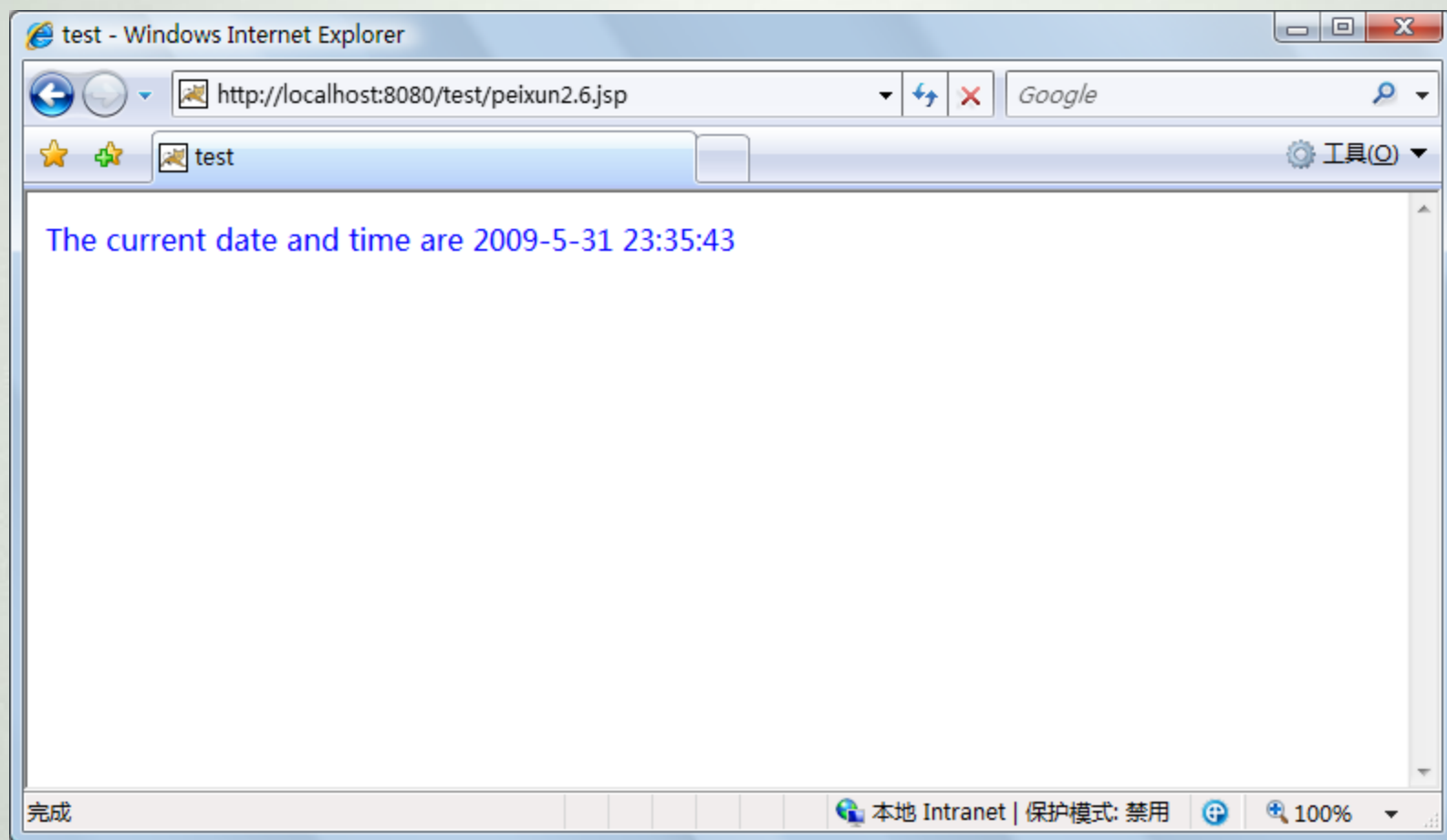
## “peixun2.6.1.jsp”

```
<%@ page import="java.util.*" %>
```

```
<%= (new java.util.Date() ).toLocaleString() %>
```



# 包含指令



# Page指令

- “Page”指令用于定义JSP文件中的全局属性。

JSP 语法格式如下：

```
<%@ page
```

```
[ language="java" ]
```

```
[ extends="package.class" ]
```

```
[ import="{package.class | package.*},..." ]
```

```
[ session="true | false" ]
```



## Page指令

```
[ isThreadSafe="true | false" ]
```

```
[ errorPage="relativeURL" ]
```

```
[ contentType="mimeType
```

```
    [;charset=characterSet]" | "text/html
```

```
    ; charset=ISO-8859-1" ]
```

```
[ isErrorPage="true | false" ]
```

```
%>
```



# Page指令

- 属性:

1. `language="java"`

声明脚本语言的种类，目前只能用“java”。

2. `import="{package.class | package.* },..."`

需要导入的Java包的列表，这些包作用于程序段，表达式，以及声明。下面的包在JSP编译时已经导入了，所以就不需要再指明了：

```
java.lang.* javax.servlet.*  
javax.servlet.jsp.*  
javax.servlet.http.*
```





## Page指令

3. `errorPage="relativeURL"`

设置处理异常事件的JSP文件。

4. `isErrorPage="true | false"`

设置此页是否为出错页，如果被设置为true，你就能使用exception对象



# page指令详说

- “<%@ page %>”指令作用于整个JSP页面，同样包括静态的包含文件。但是“<%@ page %>”指令不能作用于动态的包含文件，比如“[<jsp:include>](#)”。
- 可以在一个页面中用上多个“<%@ page %>”指令，但是其中的属性只能用一次，不过也有例外，那就是import属性。因为import属性和Java中的import语句类似(参照Java Language, import语句引入的是Java语言中的类)，所以此属性就能用多次。
- 无论把“<%@ page %>”指令放在JSP的文件的哪个地方，它的作用范围都是整个JSP页面。不过，为了JSP程序的可读性，以及好的编程习惯，最好还是把它放在JSP文件的顶部。

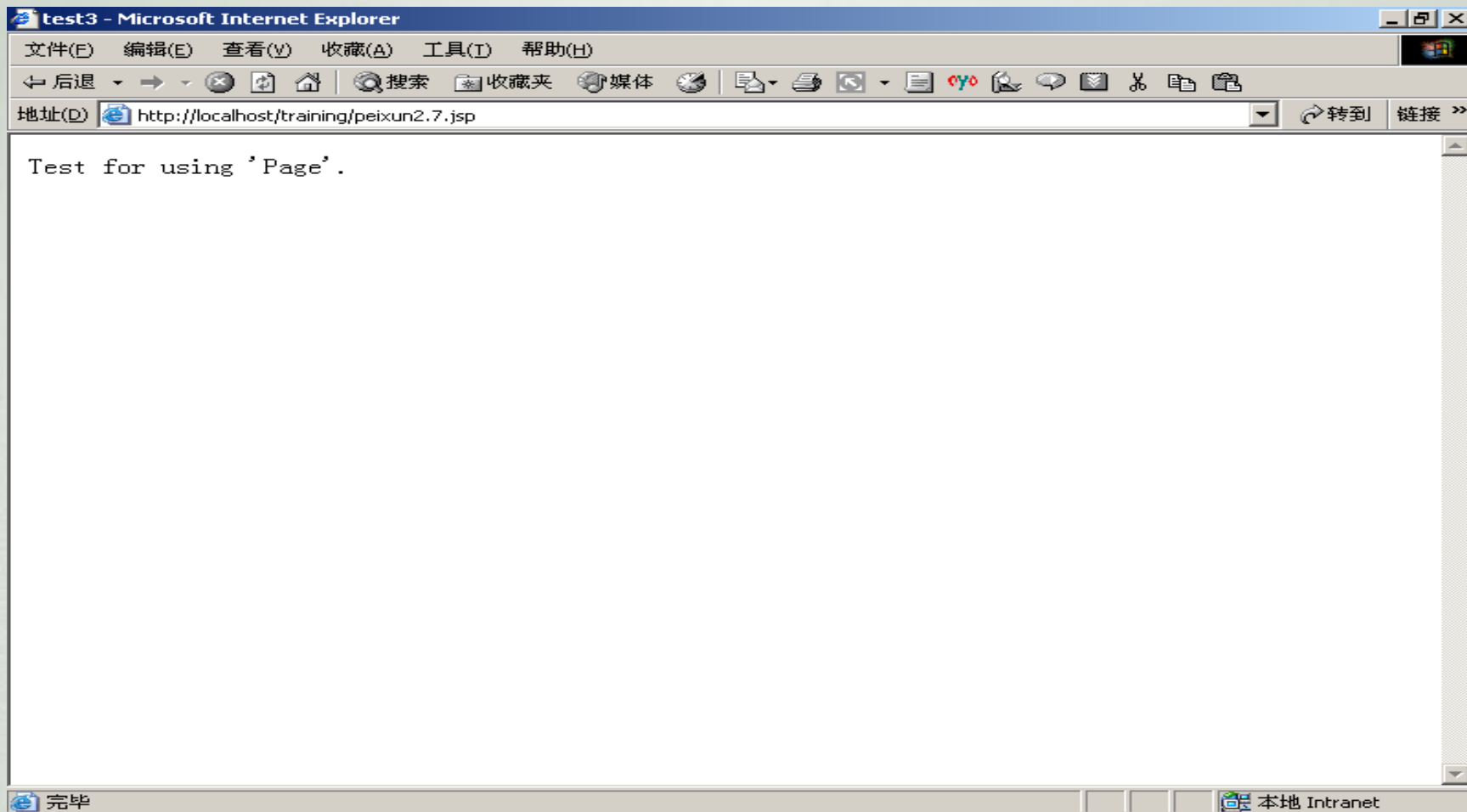


## page指令

```
<%@ page import="java.util.*, java.lang.*" %>
<%@ page errorPage="error.jsp" %>
<html>
<head>
<title>test3</title>
</head>
<body>
Test for using 'Page'.
</body>
</html>
```



# page指令例子运行结果



# taglib指令

- JSP语法

```
<%@      taglib      uri="URIToTagLibrary"  
    prefix="tagPrefix" %>
```

- 用于引入定制标签库





# taglib指令

- 属性
- 1. uri="URIToTagLibrary"
- Uniform Resource Identifier (URI) 根据标签的前缀对自定义的标签进行唯一的命名，URI可以是以下内容：
  - Uniform Resource Locator (URL)，由 RFC 2396 定义，可以从以下位置查看 (<http://www.hut.fi/u/jkorpela/rfc/2396/full.html>)。
  - Uniform Resource Name (URN)，由 RFC 2396 定义
  - 一个相对或绝对的路径
  - 可以是标签库描述文件



# taglib指令

- 2. prefix="tagPrefix"
  - 表示标签在JSP中的名称。
  - 在自定义标签之前的前缀，比如，在<public:loop>中的public，如果这里不写public，那么这就是不合法的。请不要用jsp, jsp, java, javax, servlet, sun, 和sunw做为你的前缀，这些已被Sun公司声明保留。



taglib指令

- taglib指令待讲到自定义jsp标签库时进行详细解说



## <jsp:forward>

- JSP 语法格式如下:

1. `<jsp:forward page="{relativeURL" | "<%= expression %>" } />`

2. `<jsp:forward page="{relativeURL" | "<%= expression %>" } >`

`<jsp:param name="parameterName"  
value="{parameterValue | <%= expression %>}" />`

`[<jsp:param ... />]`

`</jsp:forward>`

( “[<jsp:param ... />]”指可以有多个 “<jsp:param/>”  
标记。 )



# 属性

- 1、page="{relativeURL | <%= expression %>}"
- 这里是一个表达式或是一个字符串用于说明你将要定向的文件或URL。这个文件可以是JSP, 或程序段





# 属性

- 2 . `<jsp:param name="parameterName" value="{parameterValue | <%= expression %>}" />`
- 向一个动态文件发送一个或多个参数，这个文件必须是动态文件。如果想传递多个参数，可以在一个JSP文件中使用多个“`<jsp:param>`”；“name”指定参数名，“value”指定参数值。



# 注意

- “<jsp:forward>”标签从一个JSP文件向另一个文件传递一个包含用户请求的request对象。 “<jsp:forward>”标签以后的代码，将不能执行。



## <jsp:forward>例子 (peixun2.9.jsp)

```
<%@ page contentType="text/html;charset=gb2312" %>
<html>
  <head>
    <title>test</title>
  </head>
  <body>
    <jsp:forward page="forwardTo.jsp">
      <jsp:param name="userName" value="riso"/>
    </jsp:forward>
  </body>
</html>
```

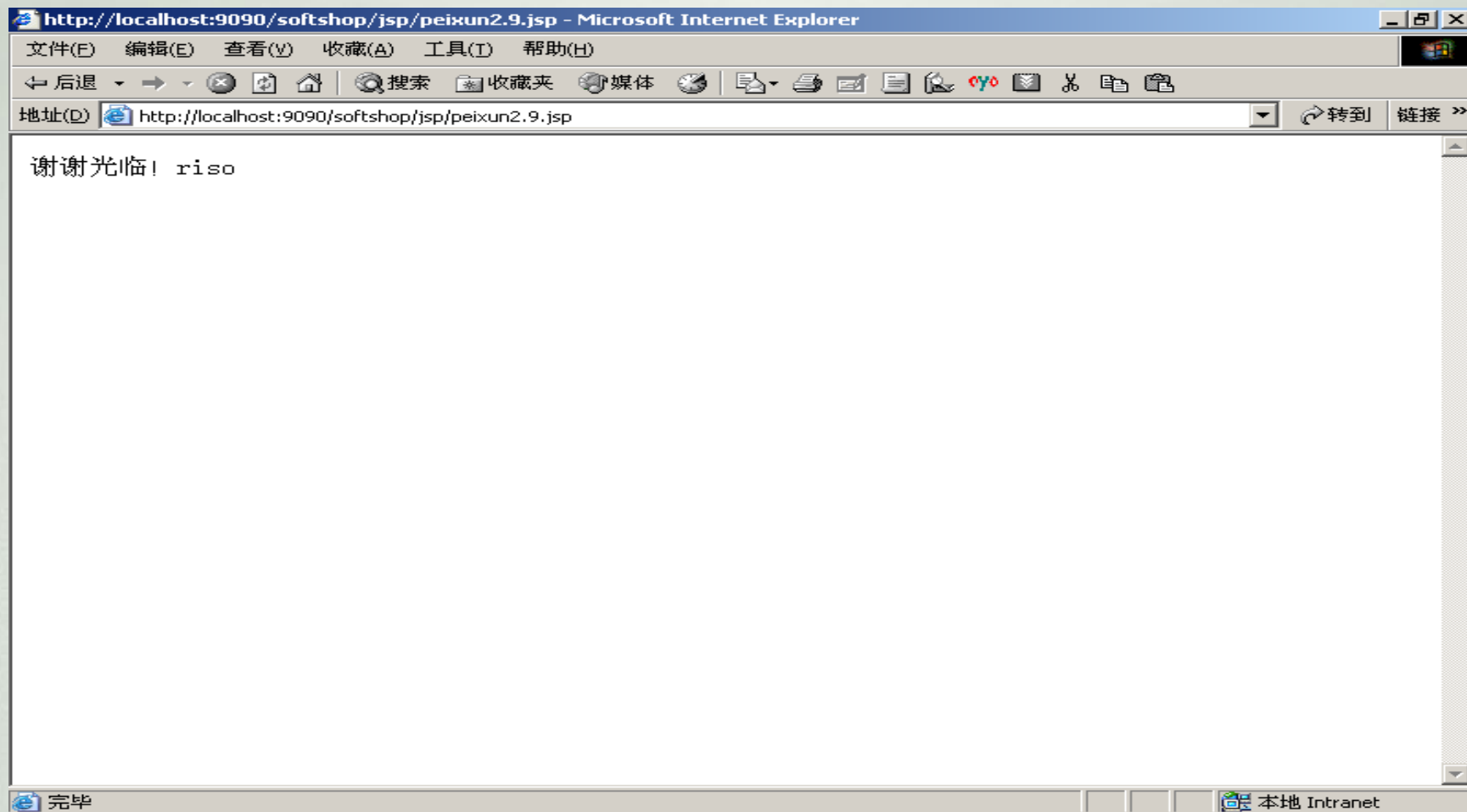


# forwardTo.jsp

```
<%@ page
    contentType="text/html;charset=gb2312" %>
<!--forwardTo.jsp-->
<%
    String
    userName=request.getParameter("userName");
    String outStr= "谢谢光临! ";
    outStr+=userName;
    out.println(outStr);
%>
```



# 运行结果





# <jsp:include>

- 包含一个静态或动态文件.

- JSP 语法格式如下:

1. `<jsp:include page="{relativeURL |  
 <%=expression%>}" flush="true" />`

2. `<jsp:include page="{relativeURL | <%=expression  
 %>}" flush="true" >`

`<jsp:param name="parameterName" value="{parameterValue |  
 <%= expression %>}" />`

`[<jsp:param .../>]`

`</jsp:include>`

( “[<jsp:param ... />]”指可以有多个 “<jsp:param/>”标记。 )



## <jsp:include>的属性

1. page="{relativeURL | <%=expression %>}"

参数为一相对路径，或者是代表相对路径的表达式。



## <jsp:include>的属性

3 .     <jsp:param     name="parameterName"  
          value="{parameterValue                 |                 <%=  
          expression %> }" />

“<jsp:param>”用来传递一个或多个参数到指定的动态文件，能在一个页面中使用多个“<jsp:param>”来传递多个参数，



# <jsp:include> 例子 (peixun2.10.jsp)

```
<html>  
  <head>  
    <title>peixun.2.10.jsp</title>  
  </head>  
  <body>  
    <jsp:include page="peixun2.10.1.jsp" >  
      <jsp:param name="User" value="HiFi King" />  
    </jsp:include>  
  </body>  
</html>
```



peixun2.10.1.jsp

<%

String username;

username=request.getParameter("User");

out.println("Username is "+username+"<br>");

%>





<isnclude>运行结果

