



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

主讲人：张龙

JSP内置对象

- JSP 有以下九种内置对象，包括：
- request，请求对象
- response，响应对象
- pageContext，页面上下文对象
- session，会话对象
- application，应用程序对象
- out，输出对象
- config，配置对象
- page，页面对象
- exception，异常对象



重点介绍

- request, 请求对象
- session, 会话对象
- application, 应用程序对象



综述

- 有几种对象看起来和**ASP**的内置对象差不多，功能也类似，这是因为这些内置对象的构建基础是标准化的**HTTP**协议。如果使用过**ASP**，又对**Java**有一定的了解的话，那么对这几种**JSP**内置对象的使用应该能迅速掌握。需要注意的问题是对象名的写法，包括这些对象方法的调用时也要书写正确，因为**Java**语言本身是大小写敏感的。



综述

- 从本质上讲，JSP的这些内置对象其实都是由特定的Java类所产生的，在服务器运行时根据情况自动生成，所以如果你有较好的Java基础，可以参考相应的类说明，表3.1给出了他们的对应关系。更进一步的讲解将会在讲解Servlet的一章中展开，如果你的Java基础不太好，可以先熟悉在这一章里所介绍的各种方法的使用。



表 3.1 JSP 内置对象映射表

| 对象名 | 类型 | 作用域 |
|-------------|---|-------------|
| request | javax.servlet.HttpServletRequest 的子类 | Request |
| response | javax.servlet.HttpServletResponse 的子 类 | Page |
| pageContext | javax.servlet.jsp.PageContext | Page |
| session | javax.servlet.http.HttpSession | Session |
| application | javax.servlet.ServletContext | Application |
| Out | javax.servlet.jsp.JspWriter | Page |
| config | javax.servlet.ServletConfig | Page |
| Page | java.lang.Object | Page |
| exception | java.lang.Throwable | Page |

续表 3.1

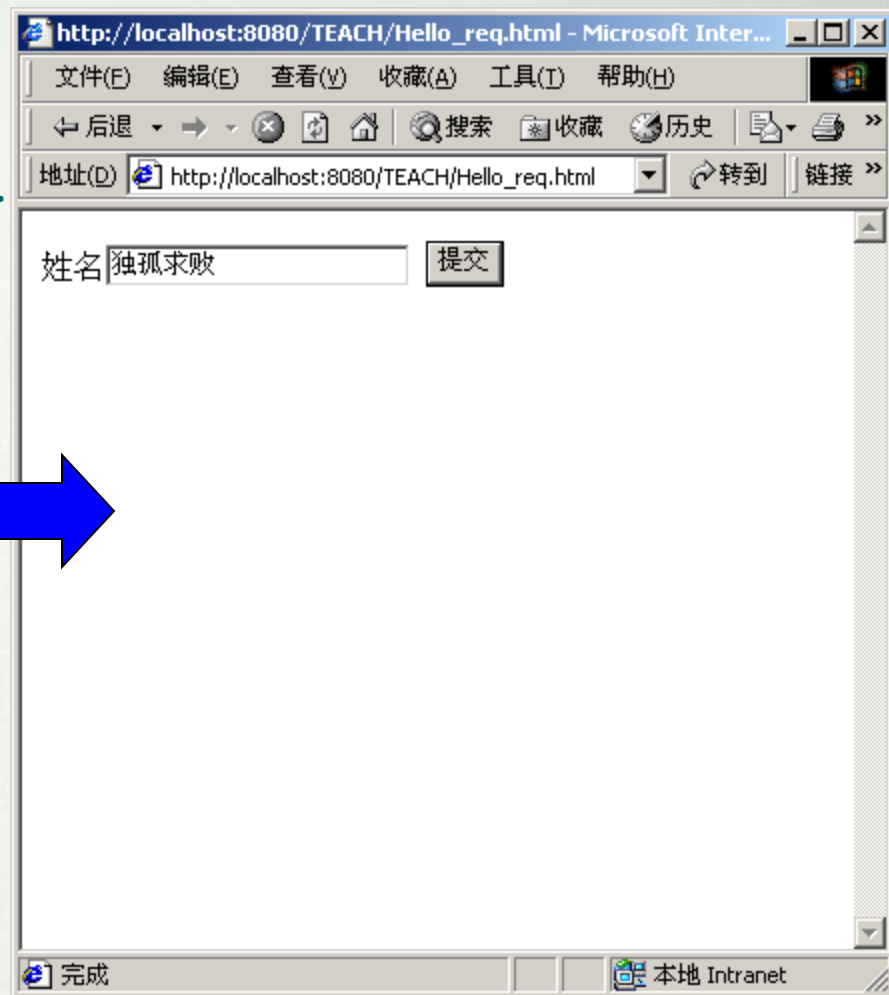
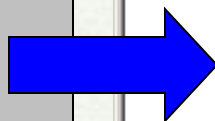
request和response

- “request” 对象代表的是来自客户端的请求，例如我们在FORM表单中填写的信息等，是最常用的对象。关于它的方法使用较多的是getParameter、getParameterNames 和getParameterValues，通过调用这几个方法来获取请求对象中所包含的参数的值
- “response” 对象代表的是对客户端的响应，也就是说可以通过“response”对象来组织发送到客户端的数据。但是由于组织方式比较底层，所以不建议普通读者使用，需要向客户端发送文字时直接使用“out”对象即可。



request对象示例(hello.jsp)

```
<html><body>  
  <form action="./Hello_req.jsp">  
    姓名<input type="text" name="UserName" >  
    <input type="submit" value="提交">  
  </form>  
</body></html>
```



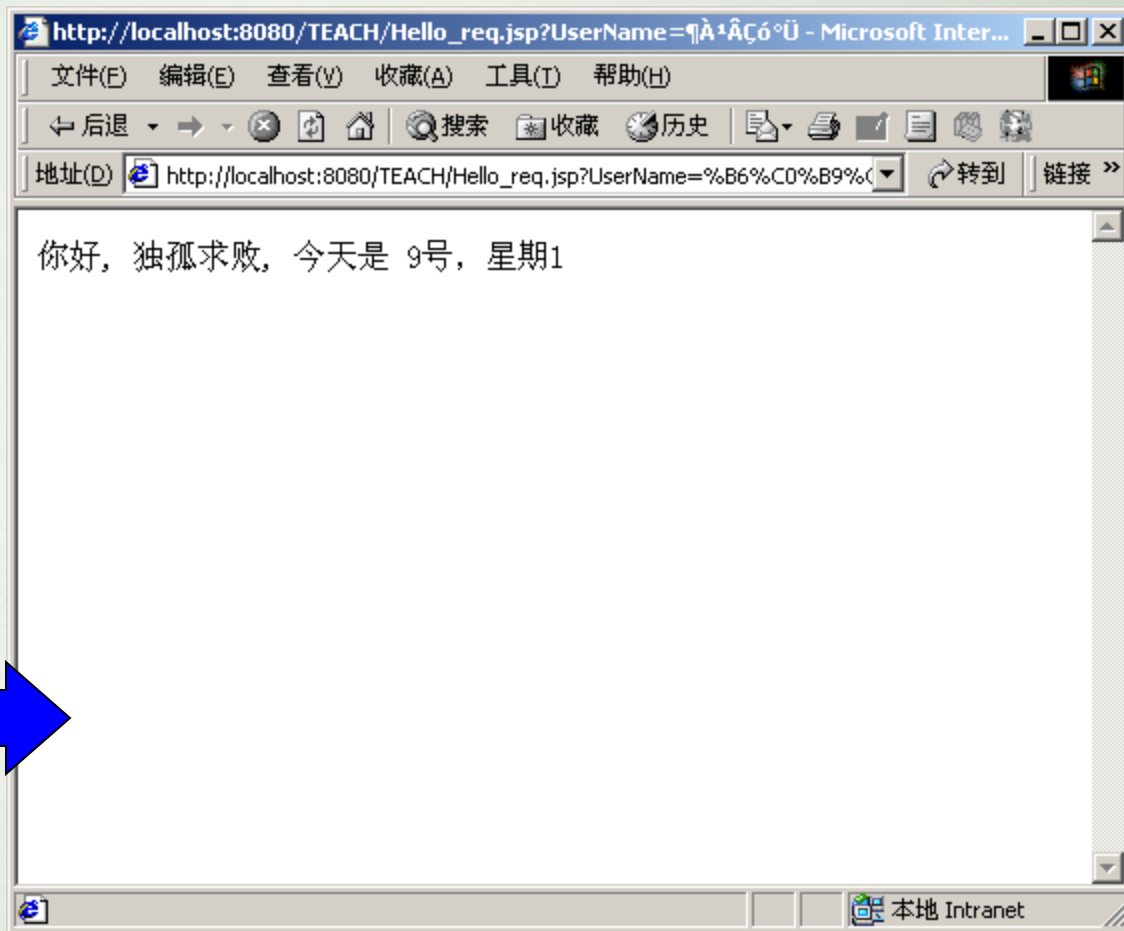
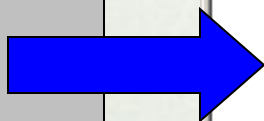
request对象示例(hello_req.jsp)

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



request对象示例（输出结果）

```
<HTML>
<BODY>
  你好，
  独孤求败，
  今天是
  9号，星期1
</BODY>
</HTML>
```



pageContext

- “pageContext” 对象直译时可以称作“页面上下文”对象，代表的是当前页面运行的一些属性，常用的方法包括 findAttribute 、 getAttribute 、 getAttributesScope 和 getAttributeNamesInScope ，一般情况下“pageContext”对象用到得也不是很多，只有在项目所面临的情况比较复杂的情况下，才会利用到页面属性来辅助处理。一般Servlet容器会使用该对象

-



session对象

- “**session**” 对象代表服务器与客户端所建立的会话，当需要在不同的**JSP**页面中保留客户信息的情况下使用，比如在线购物、客户轨迹跟踪等。



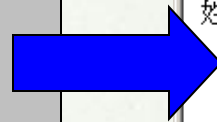
session对象

- 概要
 - HTTP是无状态（**stateless**）协议；
 - Web Server 对每一个客户端请求都没有历史记忆；
 - Session用来保存客户端状态信息；



session对象示例(Form表单—Logon.jsp)

```
<html><body>
<form action="./Logon_session.jsp">
  姓名<input type="text" name="UserName">
  <input type="submit" value="提交">
</form>
</body></html>
```

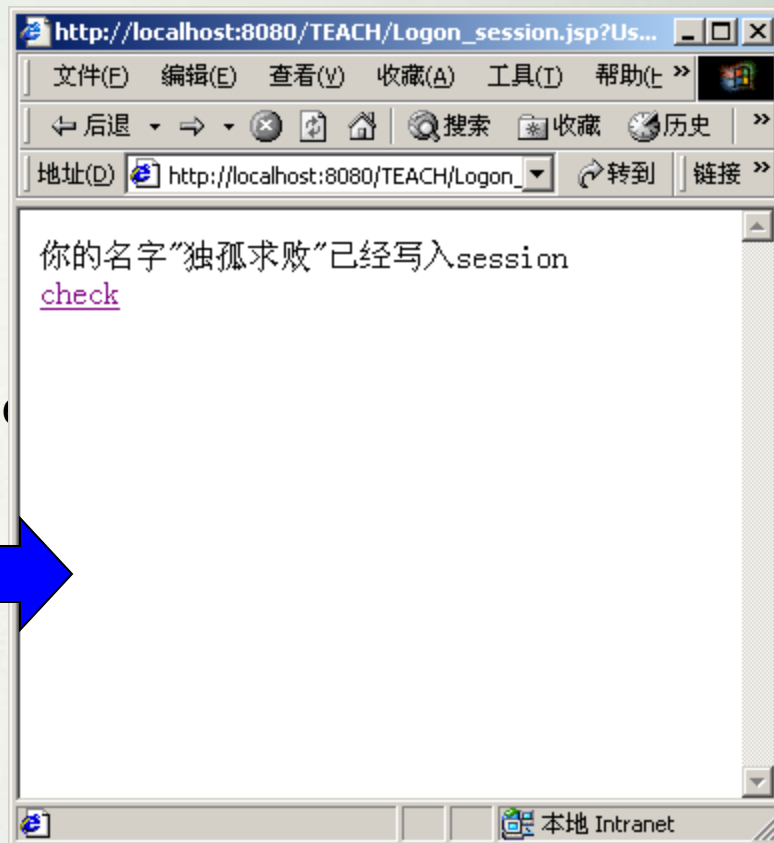


The screenshot shows a Microsoft Internet Explorer window with the title "http://localhost:8080/TEACH/Logon_session.html - Microsoft Int...". The address bar shows "http://localhost:8080/TEACH/Logon_session.html". The main content area displays a web form with the label "姓名" (Name) followed by a text input field and a "提交" (Submit) button. The browser's status bar at the bottom shows "完成" (Done) and "本地 Intranet" (Local Intranet).

session对象示例logon_session.jsp

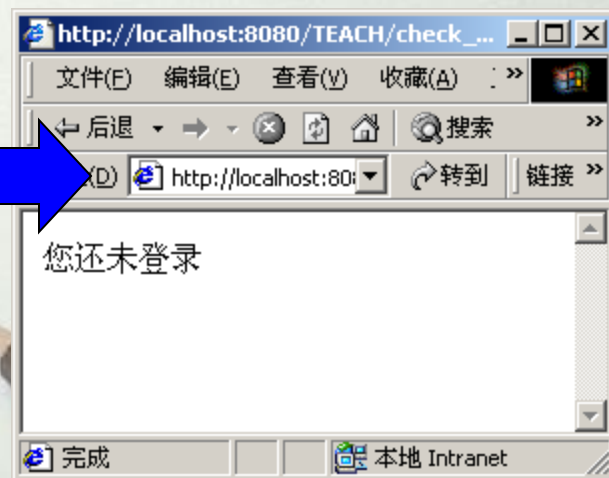
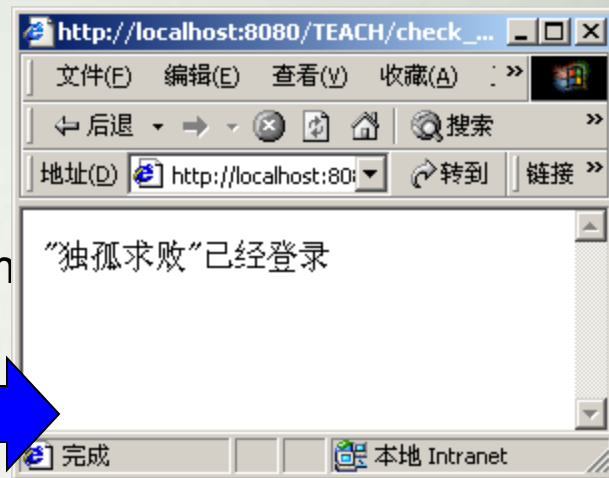
```
<%@page contentType="text/html;gb2312" %>
<%@page import="java.util.*" %>

<HTML>
  <BODY>
    <%
      String Name=request.getParameter("UserName");
      session.setAttribute("LogName", Name);
    %>
    你的名字"<%=Name%>"已经写入session
    <br>
    <a href='./check_session.jsp'>check</a>
  </BODY>
</HTML>
```



session对象示例(check_session.jsp)

```
<%@page contentType="text/html;gb2312" %>
<HTML><BODY>
  <%
    String yourName=(String)session.getAttribute("LogName")
    if (yourName==null)
    {
      %>您还未登录
    }
    %>
  }else
  {
    %>
    "<%=yourName%>"已经登录
  }
  %>
</BODY></HTML>
```



application对象

- “application” 对象负责提供应用程序在服务器中运行时的一些全局信息，常用的方法有getMimeType和getRealPath等。



out

- “out” 对象代表了向客户端发送数据的对象，与 “response” 对象不同，通过 “out” 对象发送的内容将是浏览器需要显示的内容，是文本一级的，可以通过 “out” 对象直接向客户端写一个由程序动态生成HTML文件。常用的方法除了print和println之外，还包括clear、clearBuffer 、 flush 、 getBufferSize 和 getRemaining，这是因为 “out” 对象内部包含了一个缓冲区，所以需要一些对缓冲区进行操作的方法。



config、page、exception

- “config” 对象提供一些配置信息，常用的方法有 `getInitParameter` 和 `getInitParameterNames`，以获得 `Servlet` 初始化时的参数。
- “page” 对象代表了正在运行的由 `JSP` 文件产生的类对象，不建议一般读者使用。
- “exception” 对象则代表了 `JSP` 文件运行时所产生的异常对象，此对象不能在一般 `JSP` 文件中直接使用，而只能在使用了 “`<%@ page isErrorPage="true" %>`” 的 `JSP` 文件中使用



request请求对象

- 1. `getAttribute`
- 原型: `public java.lang.Object
getAttribute(java.lang.String name)`
- 说明: 以对象的形式返回特定名称的属性值。这个方法允许Servlet引擎给出一个客户请求的特定信息。



request请求对象

- 如果所给名称的属性不存在将返回空值。
- 参数: **name** 一个标示属性名的字符串
- 返回值: 一个包含属性值的对象, 当属性不存在时空。



request对象使用示例（ peixun2.11.jsp ）

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

<html>

<head>

<title>request</title>

</head>

<body>

```
<form method="POST" action="post.jsp">
```

用户登录

<p align="center">

用户名:

密码:



浏览器类型: <input type="radio" name="radio" value="ie" checked="">IE

```
<input type="radio" name="radio" value="ns">Netscape <br>
```

资料选项:

```
<input type="checkbox" name="check1" value="ON" checked>标题
```

☐ 详细

</p><p align="center">

</form>

</body>

</html>



Post.jsp代码

```
<%@ page contentType="text/html;charset=gb2312"%>
<html>
<head><title>test</title></head>
<body>
<%
    out.println("用户名: "+request.getParameter("User")+"<br>");
    out.println("密码: "+request.getParameter("pwd")+"<br>");
    out.println("浏览器: "+request.getParameter("radio")+"<br>");

    out.println("标题: "+request.getParameter("check1")+"<br>");
    out.println("详细: "+request.getParameter("check2")+"<br>");
    out.println("隐含: "+request.getParameter("hidden")+"<br>");
    out.println("getContentLength()="+request.getContentLength()
    ()+"<br>");
```



```
out.println("getContentType()="+request.getContentType()+"<br>");
out.println("getRequestURI()="+request.getRequestURI()+"<br>");
out.println("getServerPort()="+request.getServerPort()+"<br>");
out.println("getServerName()="+request.getServerName()+"<br>");
out.println("getProtocol()="+request.getProtocol()+"<br>");
out.println("getRemoteAddr()="+request.getRemoteAddr()+"<br>");
out.println("getHeaderNames()="+request.getHeaderNames()+"<br>");
out.println("getMethod()="+request.getMethod()+"<br>");
out.println("getServletPath()="+request.getServletPath()+"<br>");
out.println("isRequestedSessionIdValid()="+request.isRequestedSessionIdValid()+
    "<br>");
```

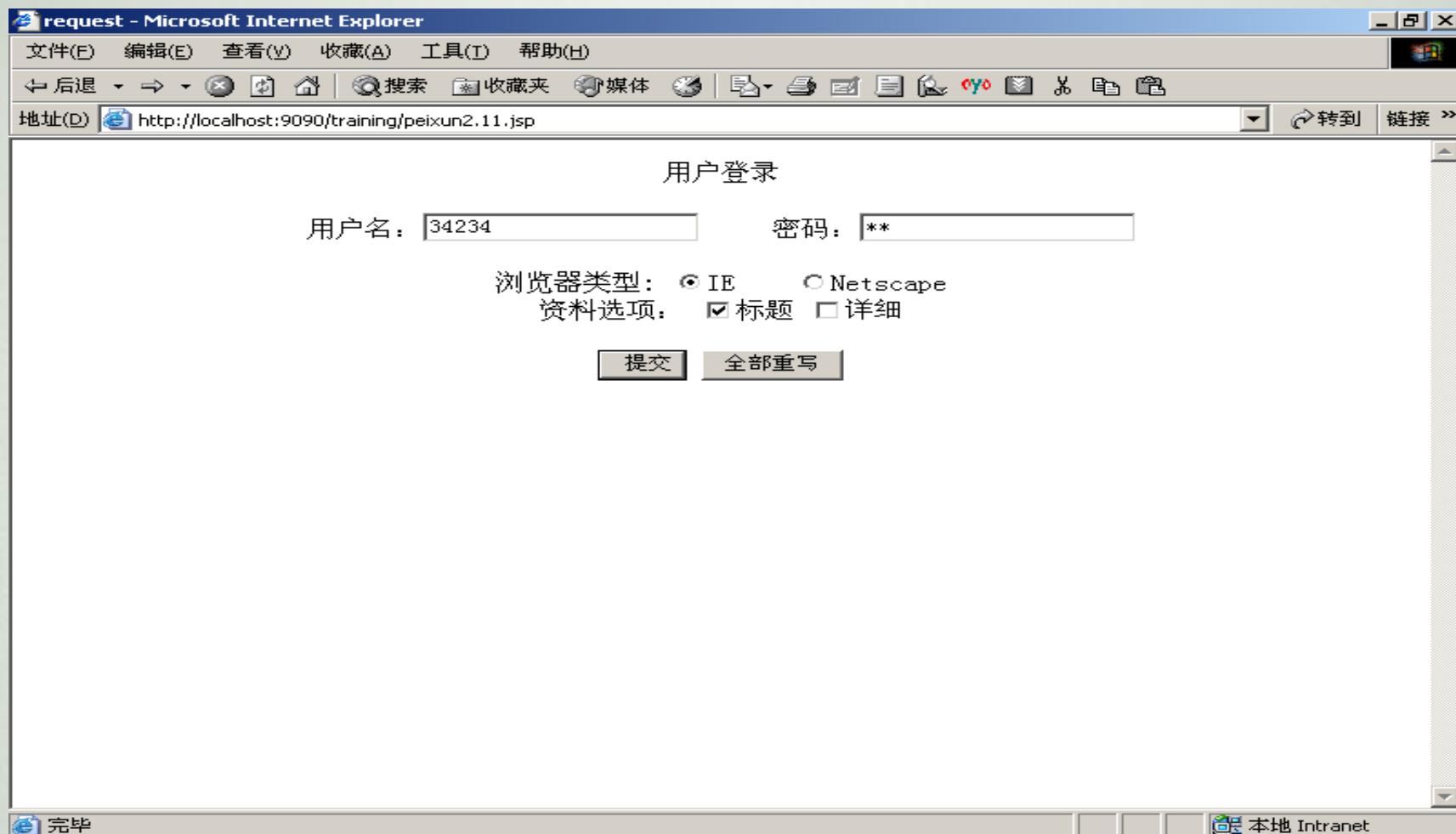
```
%>
```

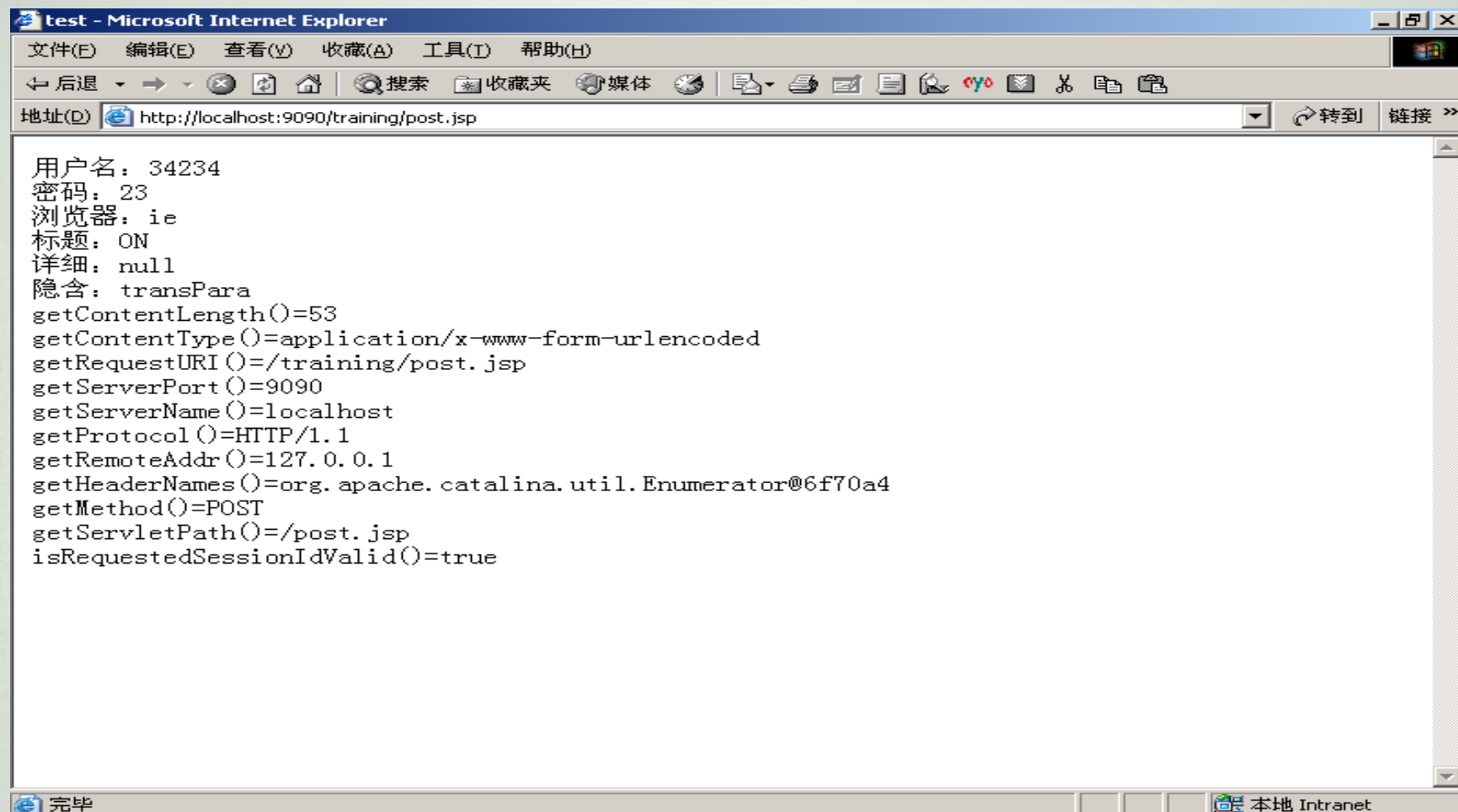
```
</body>
```

```
</html>
```



运行结果





session对象示例 (peixun2.12.jsp)

<%--session对象示例, peixun2.12.jsp文件代码--%>

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

<html>

<head><title>session</title></head>

<body>

<form method="post" action="post1.jsp">

请输入您的姓名:

<input type="text" name="username">

<input type="submit" value="submit">

</form >

</body>

</html>



post1.jsp

```
<%@ page contentType="text/html;charset=gb2312"%>
<html>
<head><title>post.jsp</title></head>
<body>
<%@ page language="java" %>
  <%! String name=""; %>
<p>
<%
    name = request.getParameter("username");
    //将username存入session
    session.setAttribute("username", name);
%>
您的姓名是: <%=name%>
<p><form method="post" action="post11.jsp">
您最喜欢的娱乐是:
<input type="text" name="amuse">
<input type="submit" value="submit"> </p>
</form>
</body>
</html>
```

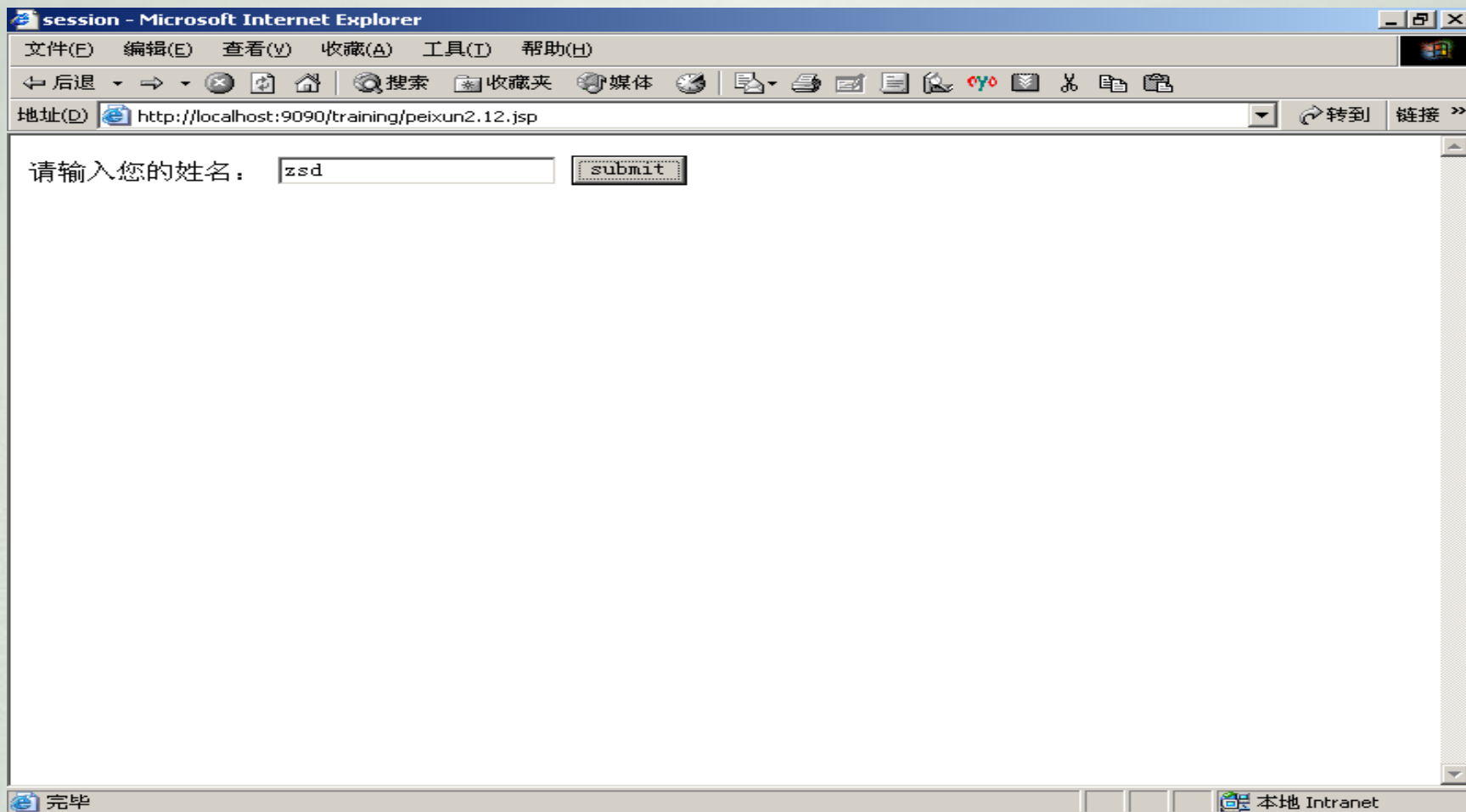


post11.jsp

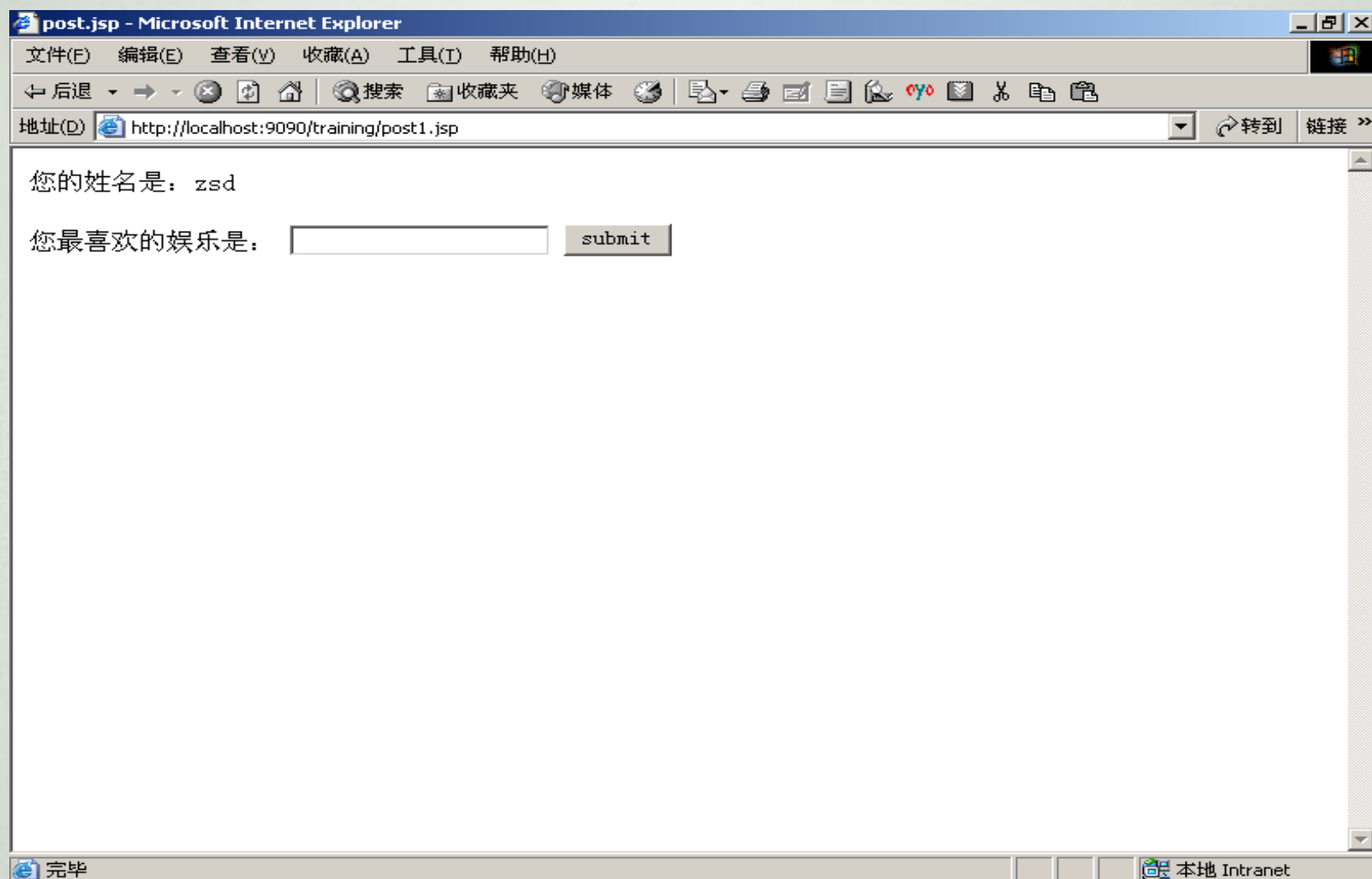
```
<%@ page contentType="text/html;charset=gb2312"%>
<html>
    <head><title>post1.jsp</title></head>
<body>
<%@ page language="java"%>
<%! String amuse=""; %>
<%
    amuse=request.getParameter("amuse");
    String user=(String)session.getAttribute("username");
%>
您的姓名是:  <%=user%>
您喜欢的娱乐是:  <%=amuse%>
</body>
</html>
```



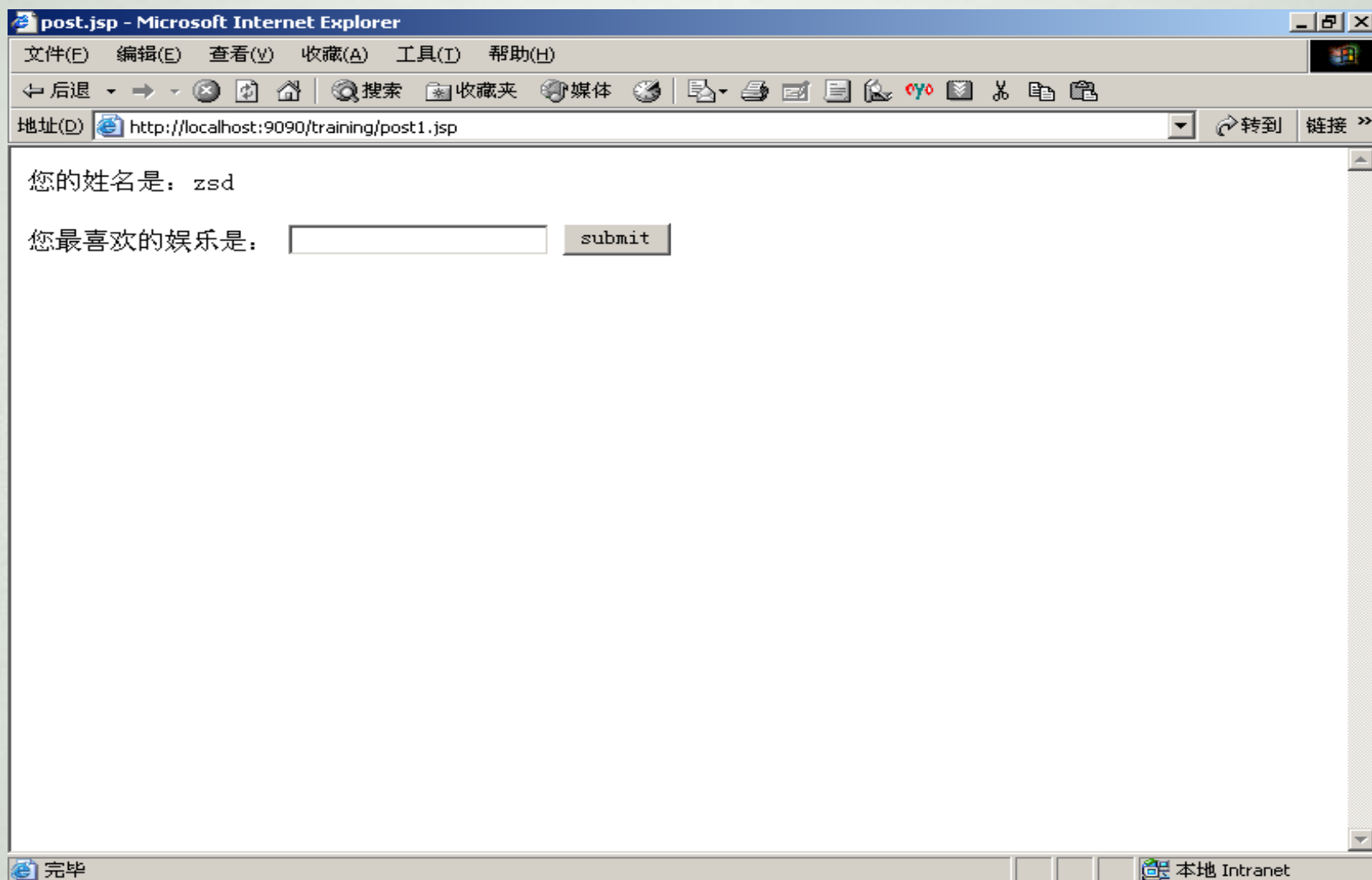
运行结果



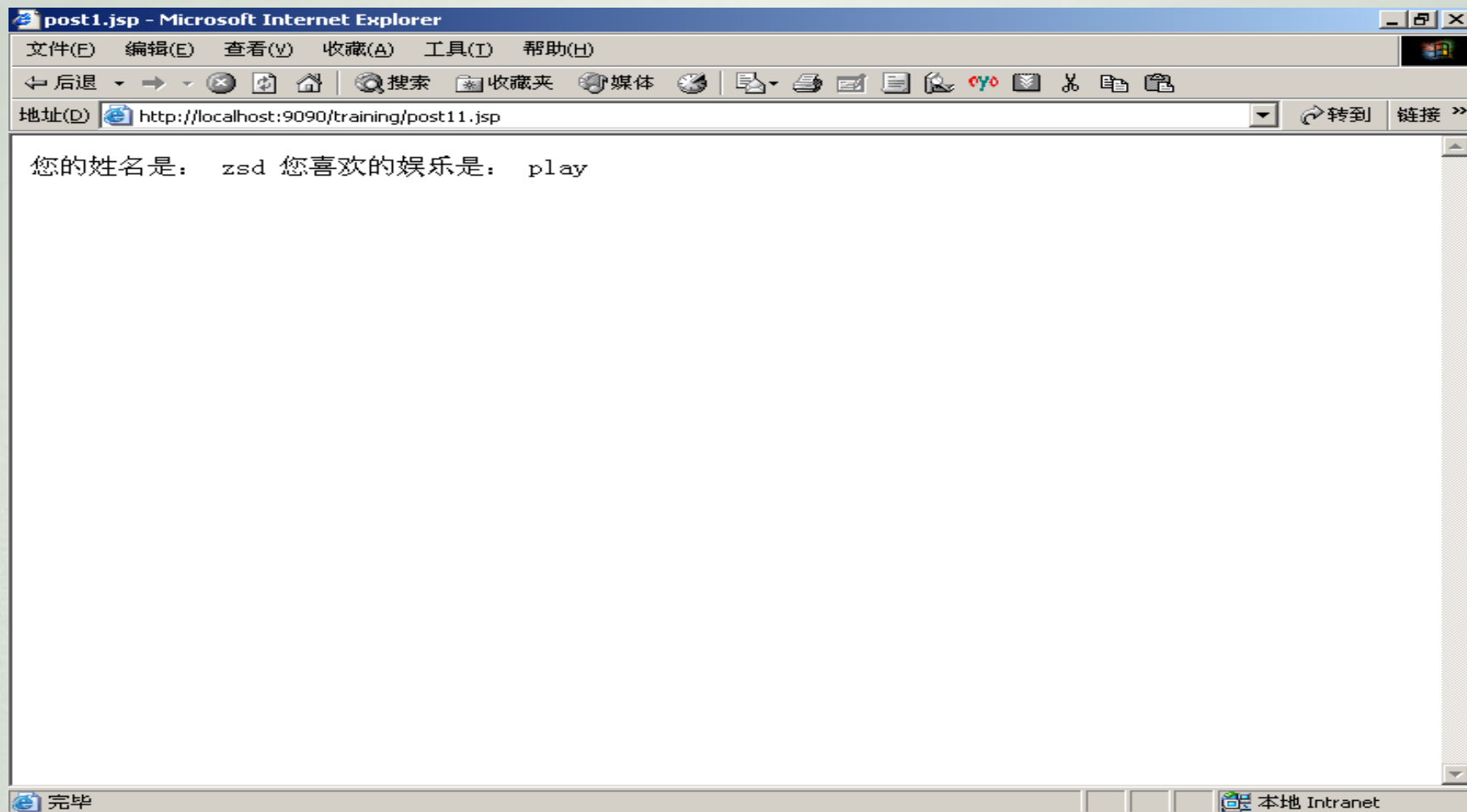
提交后



输入值




再次提交（保存名字）



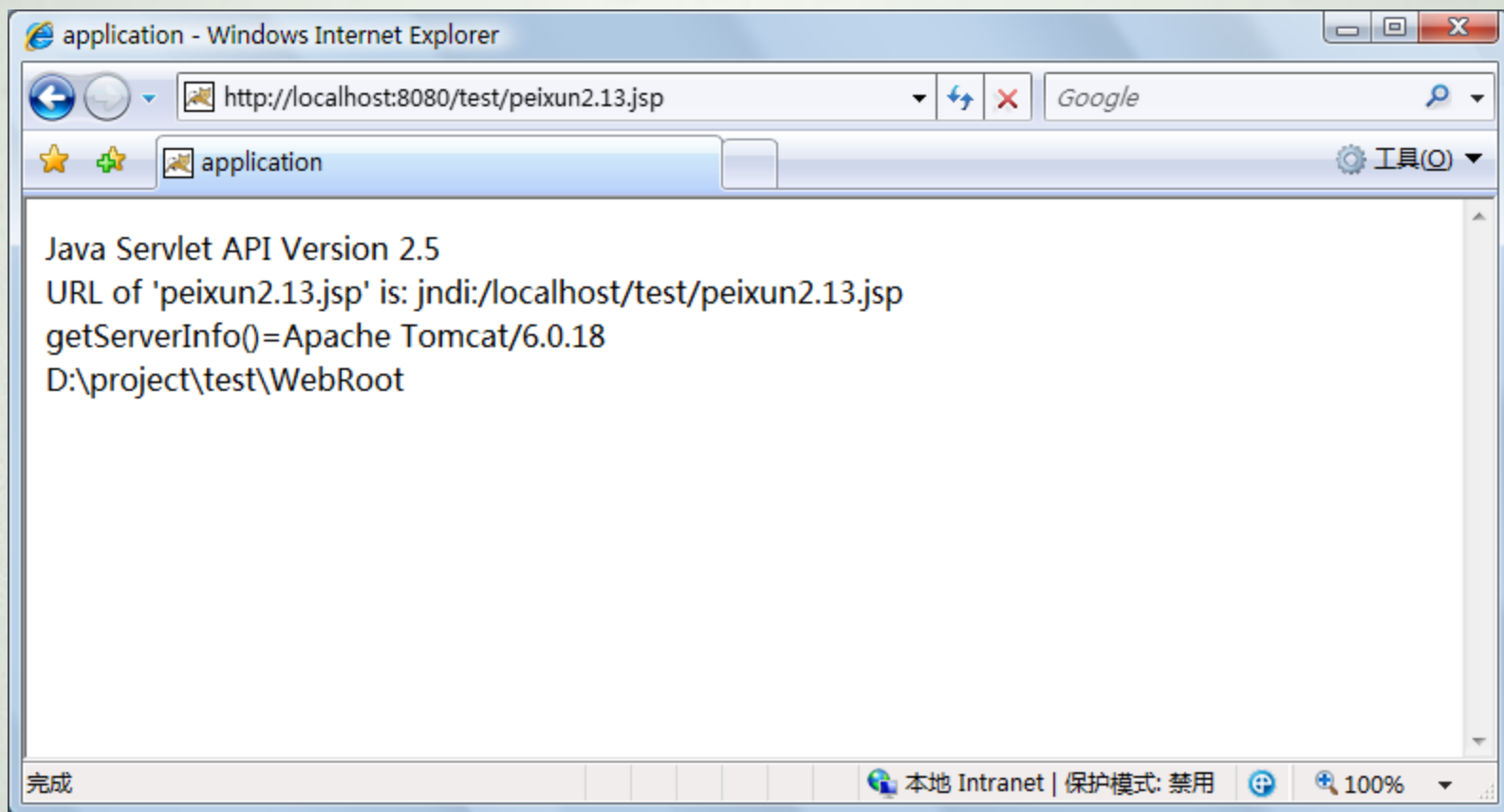
application对象运行示例

(peixun2.13.jsp)

```
<%--application对象示例， peixun2.13.jsp文件代码--%>
<%@ page contentType="text/html;charset=gb2312"%>
<html>
    <head><title>application</title><head>
<body>
<% out.println("Java Servlet API Version
    "+application.getMajorVersion()
    +"."+application.getMinorVersion()+"<br>");
    out.println("URL of 'peixun2.13.jsp' is:
    "+application.getResource("/peixun2.13.jsp")+"<br>");
    out.println("getServerInfo()="+application.getServerInfo()+"
    <br>");
    out.println(application.getRealPath(" "));
    application.log("Add a Record to log_file"); %>
</body>
</html>
```



运行结果

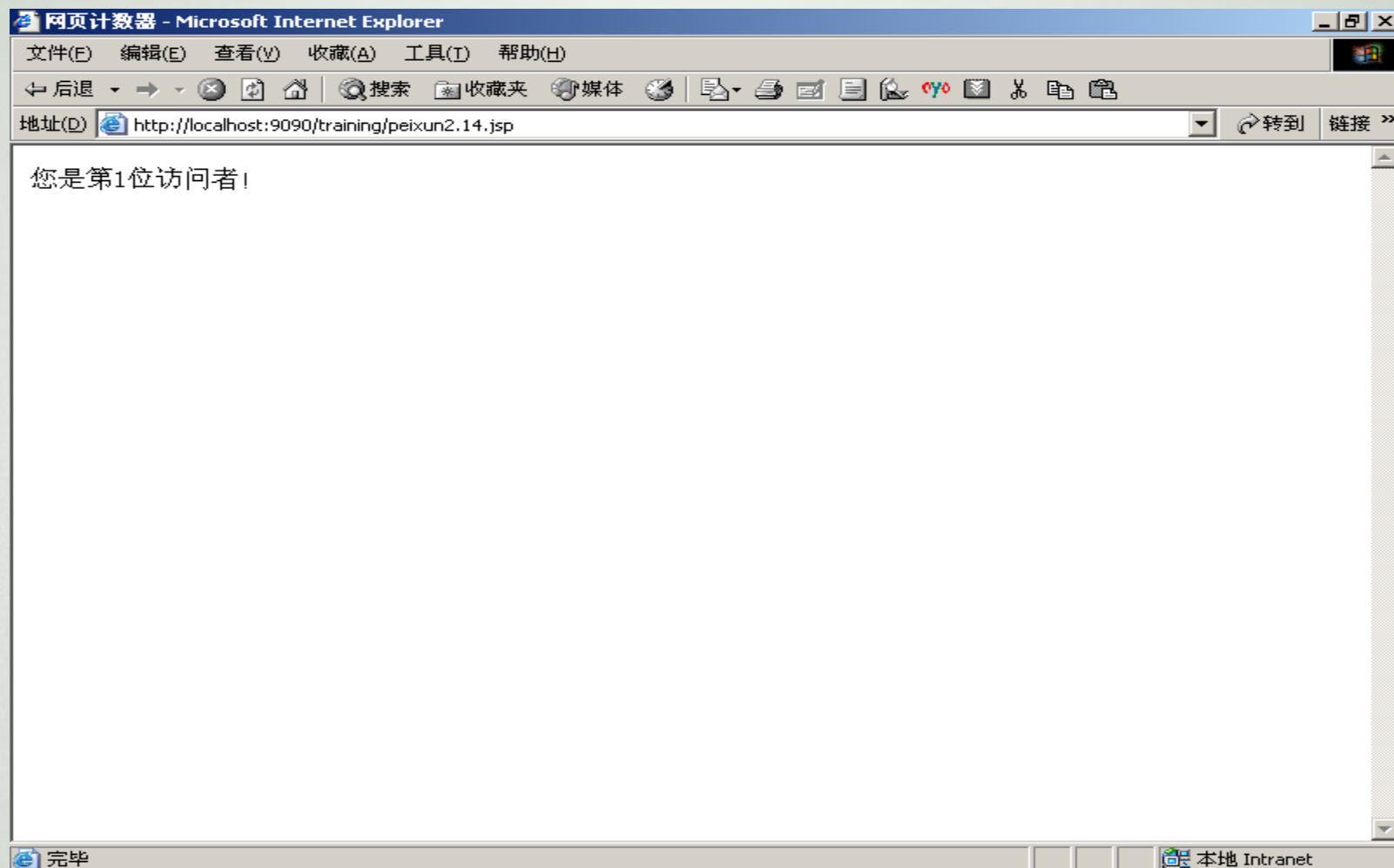


application对象实现网页计数器

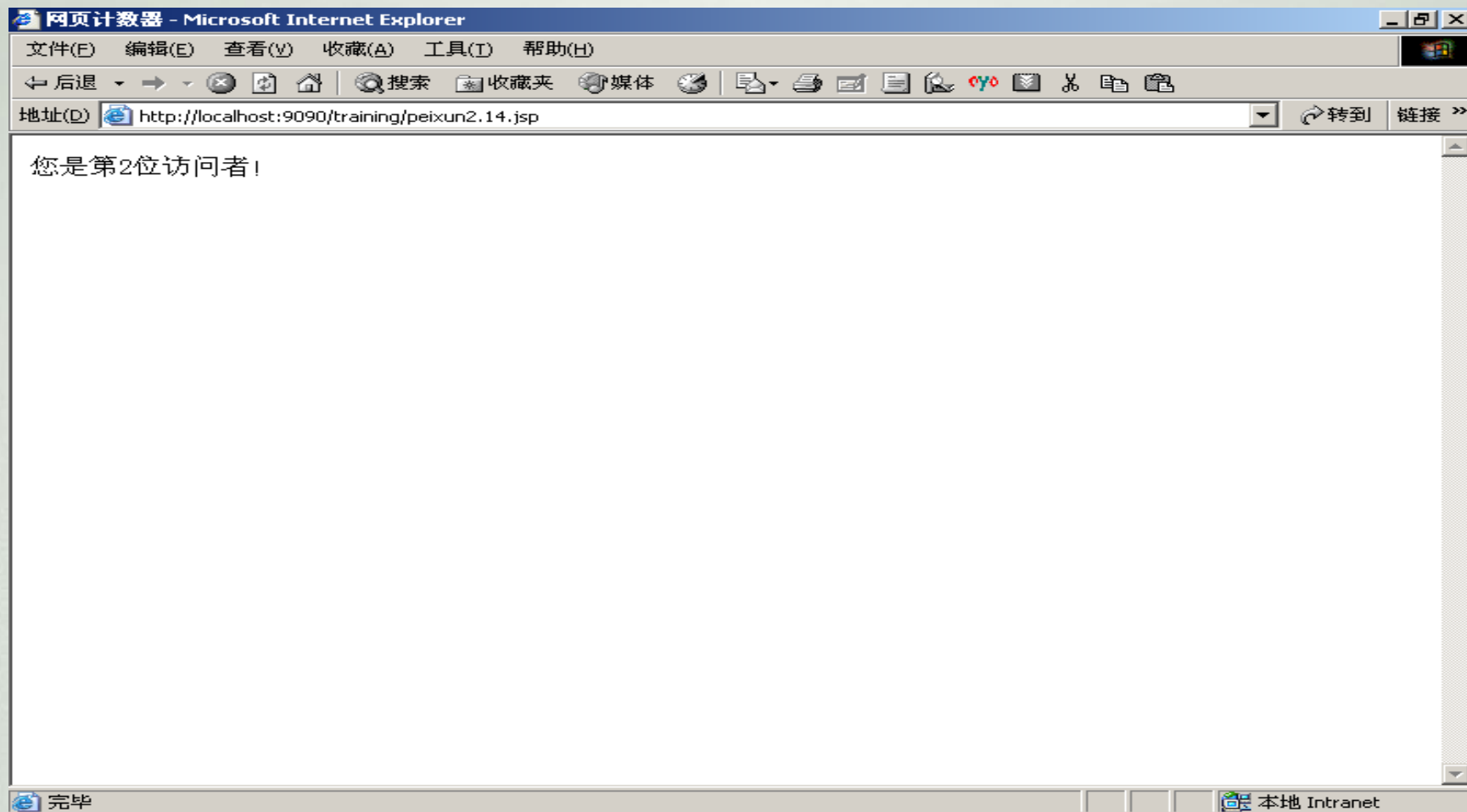
```
<!--用application对象实现网页计数器peixun2.14.jsp文件代码--%>
<%@ page contentType="text/html;charset=gb2312"%>
<html>
    <head><title>网页计数器</title></head>
<body>
<%  if (application.getAttribute("counter")==null)
    application.setAttribute("counter","1");
    else{
        String strnum=null;
        strnum=application.getAttribute("counter").toString();
        int icount=0;
        icount=Integer.valueOf(strnum).intValue();
        icount++;
        application.setAttribute("counter",Integer.toString(icount));
    }  %>
您是第<%=application.getAttribute("counter")%>位访问者!
</body>
</html>
```



运行结果(第一次访问)



刷新一次



内置对象小结

- 全面介绍了**JSP**的几种内置对象。熟练使用这些内置对象是开发**JSP**应用程序的基本要求，尤其是对于 **request**、**session** 和 **application**对象更是必须要掌握的。
- 会使用对象所具有的方法，能熟练使用每一种对象的几种常用方法。

