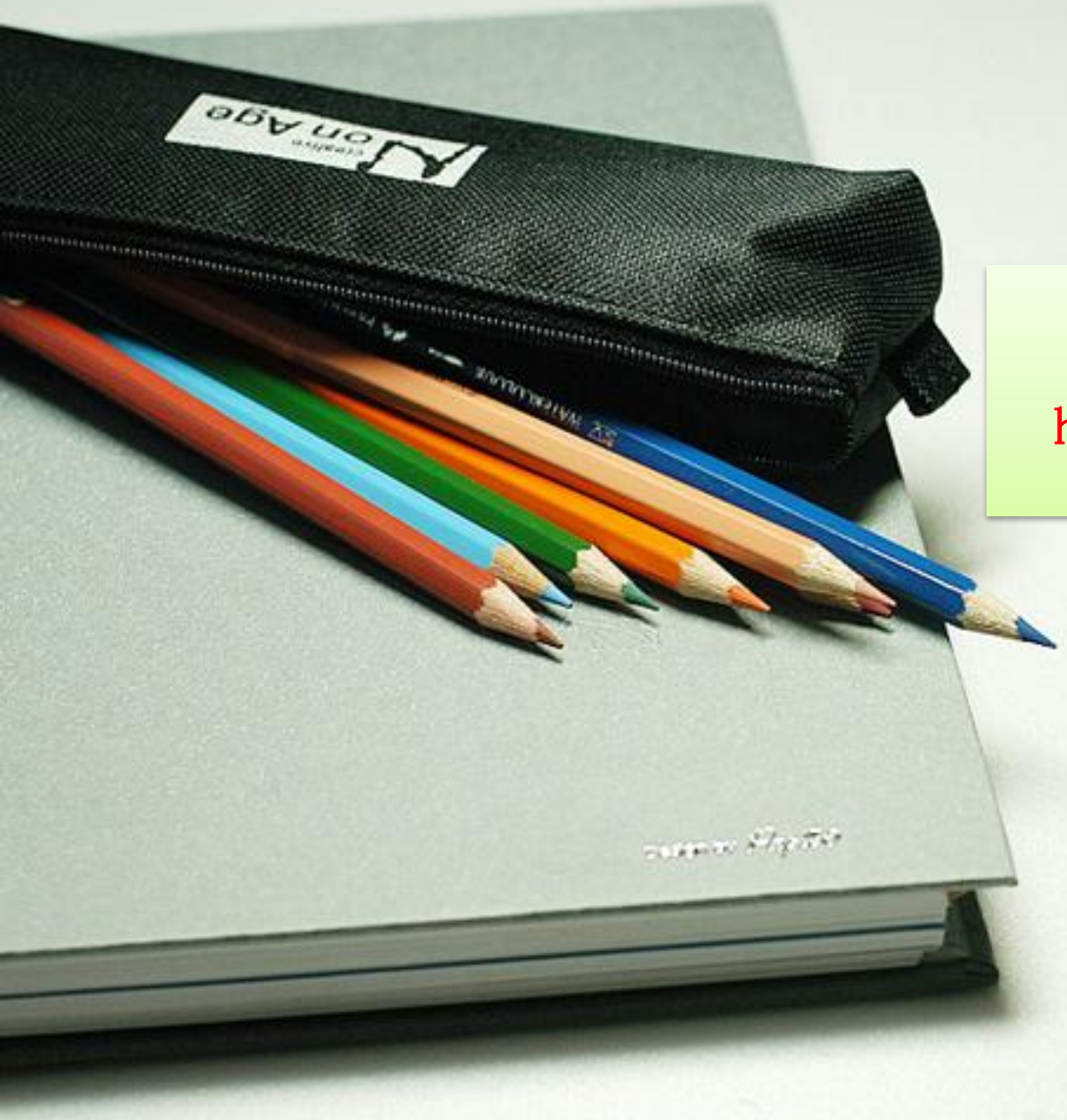


北京圣思园科技有限公司
http://www.shengsiyuan.com

主讲人：张龙

使用JavaBean

- 教学目标
  - 了解JavaBean的概念
  - 了解JavaBean的规范
  - 掌握JSP访问JavaBean的语法
  - 理解JavaBean的四种范围

JavaBean的概念

- JavaBean是一种可重复使用、且跨平台的软件组件。JavaBean可分为两种：一种是有用户界面（UI，User Interface）的JavaBean；还有一种是没有用户界面，主要负责处理事务（如数据运算，操纵数据库）的JavaBean。JSP通常访问的是后一种JavaBean。

JSP与JavaBean搭配使用的优点

- 使得HTML与Java程序分离，这样便于维护代码。如果把所有的程序代码都写到JSP网页中，会使得代码繁杂，难以维护
- 可以降低开发JSP网页人员对Java编程能力的要求
- JSP侧重于生成动态网页，事务处理由JavaBean来完成，这样可以充分利用JavaBean组件的可重用性特点，提高开发网站的效率

JavaBean的特征

- 一个标准的JavaBean有以下几个特性：
  - JavaBean是一个公共的（public）类
  - JavaBean有一个不带参数的构造方法
  - JavaBean通过setXXX方法设置属性，通过getXXX方法获取属性

CounterBean类

```
public class CounterBean{
    private int count=0;
    public CounterBean(){}
    public int getCount(){
        return count;
    }
    public void setCount(int count){
        this.count=count;
    }
}
```

JSP访问JavaBean的语法

- 1.　导入JavaBean类
- 2　　声明JavaBean对象
- 3．访问JavaBean属性

导入JavaBean类

- 通过 <%@ page import> 指令导入JavaBean类，例如：

```
<%@ page import="mypack.CounterBean" %>
```

声明JavaBean对象

- <jsp:useBean>标签用来声明JavaBean对象，例如：

```
<jsp:useBean id="myBean"
      class="mypack.CounterBean" scope="session" />

<jsp:useBean id="myBean_1"
     class="mypack.CounterBean" scope="session" />
```

属性以及用法

- id="beanInstanceName"
- 在所定义的范围中确认Bean的变量，使之能在后面的程序中使用此变量名来分辨不同的Bean ，这个变量名对大小写敏感，必须符合所使用的脚本语言的规定，这个规定在Java Language 规范已经写明。如果Bean已经在别的“<jsp:useBean>”标记中创建，则当使用这个已经创建过的Bean时，id的值必须与原来的那个id值一致；否则则意味着创建了同一个类的两个不同的对象。

访问JavaBean属性

- JSP提供了访问JavaBean属性的标签，如果要将JavaBean的某个属性输出到网页上，可以用<jsp:getProperty>标签，例如：

```
<jsp:getProperty name="myBean" property="count" />
```

- 如果要给JavaBean的某个属性赋值，可以用<jsp:setProperty>标签，例如：

```
<jsp:setProperty name="myBean" property="count"
  value="0" />
```

<jsp:getProperty>

- JSP 语法格式如下：

<jsp:getProperty name="beanInstanceName" property="propertyName" />

属性：

1．name="beanInstanceName"

bean的名字，由<jsp:useBean>指定。

2．property="propertyName"

所指定的Bean的属性名。

<jsp:getProperty>例子

```
<html>
<jsp:useBean id="calendar" scope="page"
  class="employee.Calendar" />
  <head>
  <title>test</title>
  </head>
  <body>
  Calendar of <jsp:getProperty name="calendar"
  property="username" />
  </body>
</html>
```

<jsp:setProperty>

- 设置Bean的属性值.
- JSP 语法格式如下：

```
<jsp:setProperty
    name="beanInstanceName"
   {
        property=  "*"  |  property="propertyName"  [
   param="parameterName" ]  |
          property="propertyName" value="{string | <%=
   expression %>}"
    }
/>
```

属性

- 1. name="beanInstanceName"
  - 表示已经在“<jsp:useBean>”中创建的Bean实例的名字。
- 2. property="*"
  - 储存用户在jsp输入的所有值，用于匹配Bean中的属性。

属性

- 3. property="propertyName" [param="parameterName"]
  - 用一个参数值来指定Bean中的一个属性值，一般情况下是从request对象中获得的。其中property指定Bean的属性名，param指定request中的参数名。

属性

- 4. property="propertyName" value="{string | <%= expression %>}"
  - 使用指定的值来设定Bean属性。这个值可以是字符串，也可以是表达式。如果是字符串，那么它就会被转换成Bean属性的类型。如果是一个表达式，那么它的类型就必须和将要设定的属性值的类型一致。
  - 不能在同一个“<jsp:setProperty>”中同时使用param和value参数 。

JavaBean的范围

- scope属性决定了JavaBean对象存在的范围
  - 。scope的可选值包括：
    - page（默认值）
    - request
    - session
    - application

JavaBean在page范围内

- 客户每次请求访问JSP页面时，都会创建一个JavaBean对象。JavaBean对象的有效范围是客户请求访问的当前JSP网页。JavaBean对象在以下两种情况下都会结束生命期：
  - 客户请求访问的当前JSP网页通过<forward>标记将请求转发到另一个文件
  - 客户请求访问的当前JSP页面执行完毕并向客户端发回响应。

JavaBean在request范围内

- 客户每次请求访问JSP页面时，都会创建新的JavaBean对象。JavaBean对象的有效范围为：
  - 客户请求访问的当前JSP网页
  - 和当前JSP网页共享同一个客户请求的网页，即当前JSP网页中<%@ include>指令以及<forward>标记包含的其他JSP文件
  - 当所有共享同一个客户请求的JSP页面执行完毕并向客户端发回响应时，JavaBean对象结束生命周期。

JavaBean在request范围内

- **JavaBean对象作为属性保存在HttpRequest对象中，属性名为JavaBean的id，属性值为JavaBean对象，因此也可以通过HttpRequest.getAttribute()方法取得JavaBean对象，例如：**

```
CounterBean
    obj=(CounterBean)request.getAttribute(“myBean”);
```

JavaBean在session范围内

- **JavaBean对象被创建后，它存在于整个Session的生存周期内，同一个Session中的JSP文件共享这个JavaBean对象。**

JavaBean在session范围内

- **JavaBean对象作为属性保存在HttpSession对象中，属性名为JavaBean的id，属性值为JavaBean对象。除了可以通过JavaBean的id直接引用JavaBean对象外，也可以通过HttpSession.getAttribute()方法取得JavaBean对象，例如：**

```
CounterBean
    obj=(CounterBean)session.getAttribute("myBean");
```

JavaBean在application范围内

- **JavaBean对象被创建后，它存在于整个Web应用的生命周期内，Web应用中的所有JSP文件都能共享同一个JavaBean对象。**

JavaBean在application范围内

- **JavaBean对象作为属性保存在application对象中，属性名为JavaBean的id，属性值为JavaBean对象，除了可以通过JavaBean的id直接引用JavaBean对象外，也可以通过application.getAttribute()方法取得JavaBean对象，例如：**

```
CounterBean obj=

    (CounterBean)application.getAttribute("myBean");
```

练习题

- 问题：

```
<jsp:useBean id="myBean"
class="mypack.CounterBean" scope="application" />
```

如何输出myBean的count属性？

- 选项(A)

```
<jsp:getProperty name="myBean" property="count" />
```

练习题

```
<jsp:useBean id="myBean"
class="mypack.CounterBean" scope="application" />
```

- 选项(B)

```
<% CounterBean
   counterBean=(CounterBean)application.getAttribute("myBean");
%>
<%=counterBean.getCount()%>
```

练习题

```
<jsp:useBean id="myBean"
class="mypack.CounterBean"    scope=“application"
   />
```

- 选项(C)

```
<% CounterBean
   counterBean=application.getAttribute(“myBean”);
%>
<%=counterBean.getCount()%>
```

练习题

- 问题：为什么JavaBean必须遵守特定的规范，比如对于CounterBean的count属性，必须提供getCount()和setCount()方法，而不能随心所欲的定义为getCOunt()、insertcount()等等其他名字？

练习题

- 答案：当所有JavaBean遵守相同的规范，<jsp:getProperty>和<jsp:setProperty>标记就能自动根据JavaBean的属性来推断出它的get和set访问方法。

实验

```
package com.test.bean;
public class student {
 private long classNo;
 private String name;
 private int age;
 private boolean sexy;

  public student() {    //构造函数
    classNo=970431;
    name="ZZZl";
    age=34;
    sexy=true;
  }
```

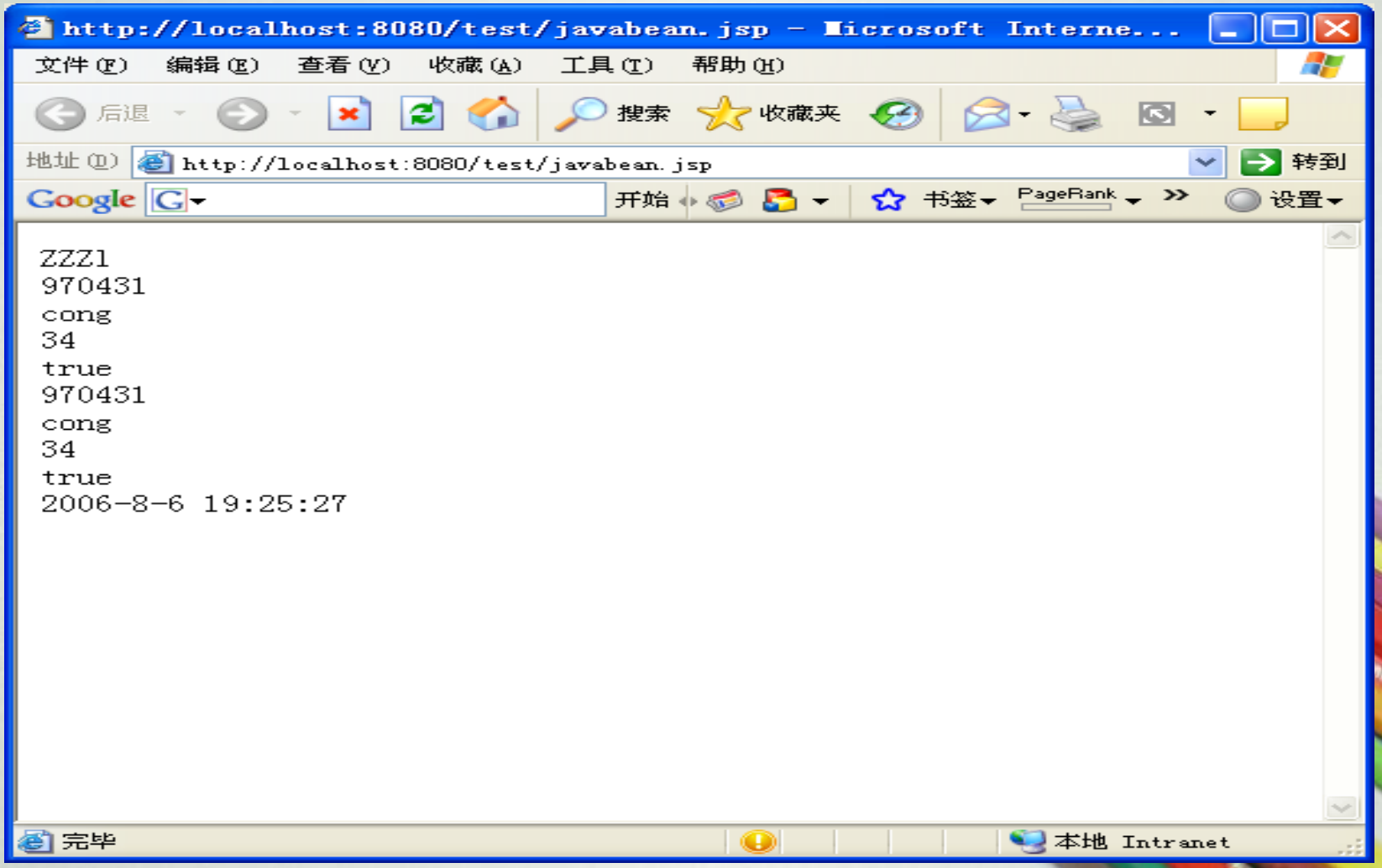
```
public long getClassNo() {
   return classNo;
 }
 public void setClassNo(long no) {
   this.classNo=no;
 }
 public String getName() {
   return name;
 }
 public void setName(String name) {
   this.name=name;
 }
```

```
 public int getAge() {
  return age;
 }

 public void setAge(int age) {
  this.age=age;
 }

 public boolean isSexy() {
  return sexy;
 }
 public void setSexy(boolean sexy) {
  this.sexy=sexy;
 }
}
```

```
<%--JavaBean.jsp文件，理解useBean动作的scope作用范围 --%>

<html>
<body>
<%-- 引用自己编写的javaBean生成的bean组件 --%>
<jsp:useBean id="student" scope="application"
   class="com.test.bean.student" />
<%=student.getname()%><br>
<% student.setname("cong"); %>

<%-- 用getProperty动作来返回bean组件的属性值 --%>
<jsp:getProperty name="student" property="classNo" /> <br>
<jsp:getProperty name="student" property="name" /> <br>
<jsp:getProperty name="student" property="age" /> <br>
<jsp:getProperty name="student" property="sexy" /> <br>
<%-- 取属性值 --%>
<%=student.getClassNo() %> <br>
<%=student.getName() %> <br>
<%=student.getAge() %> <br>
<%=student.getSexy() %> <br>
```

```
<%--引用Java类库中生成的bean组件 --%>
<jsp:useBean id="date" scope="session"
   class="java.util.Date" />
<%=date%><br>
</body>
</html>
```

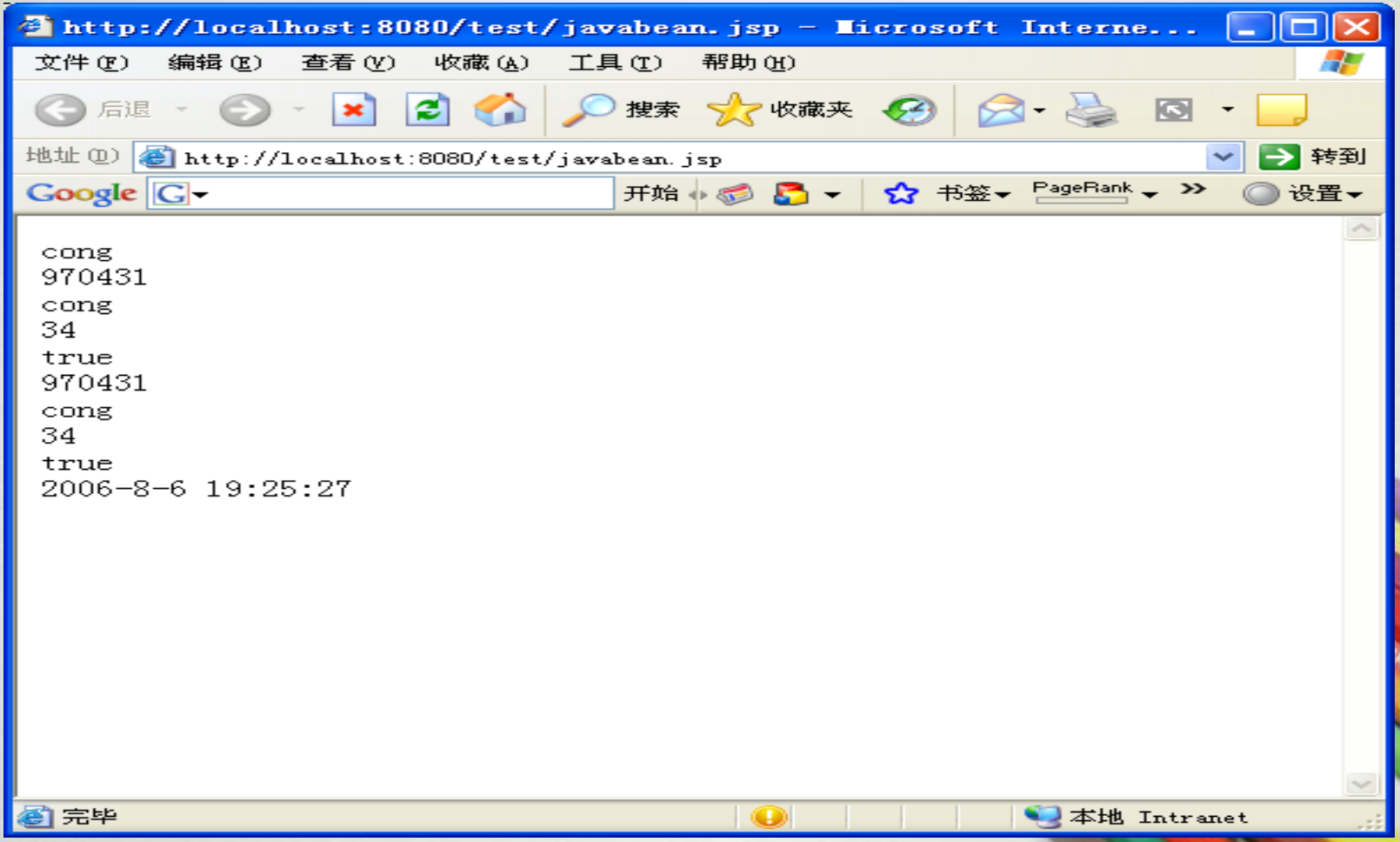
第一次运行结果


http://localhost:8080/test/javabean.jsp - Microsoft Interne...
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
后退 搜索 收藏夹
地址(D) http://localhost:8080/test/javabean.jsp
转到
Google 开始 书签 PageRank 设置
ZZZ1
970431
cong
34
true
970431
cong
34
true
2006-8-6 19:25:27
完毕
本地 Intranet

以后运行结果

第一行发生变化

为什么时间没有变化？

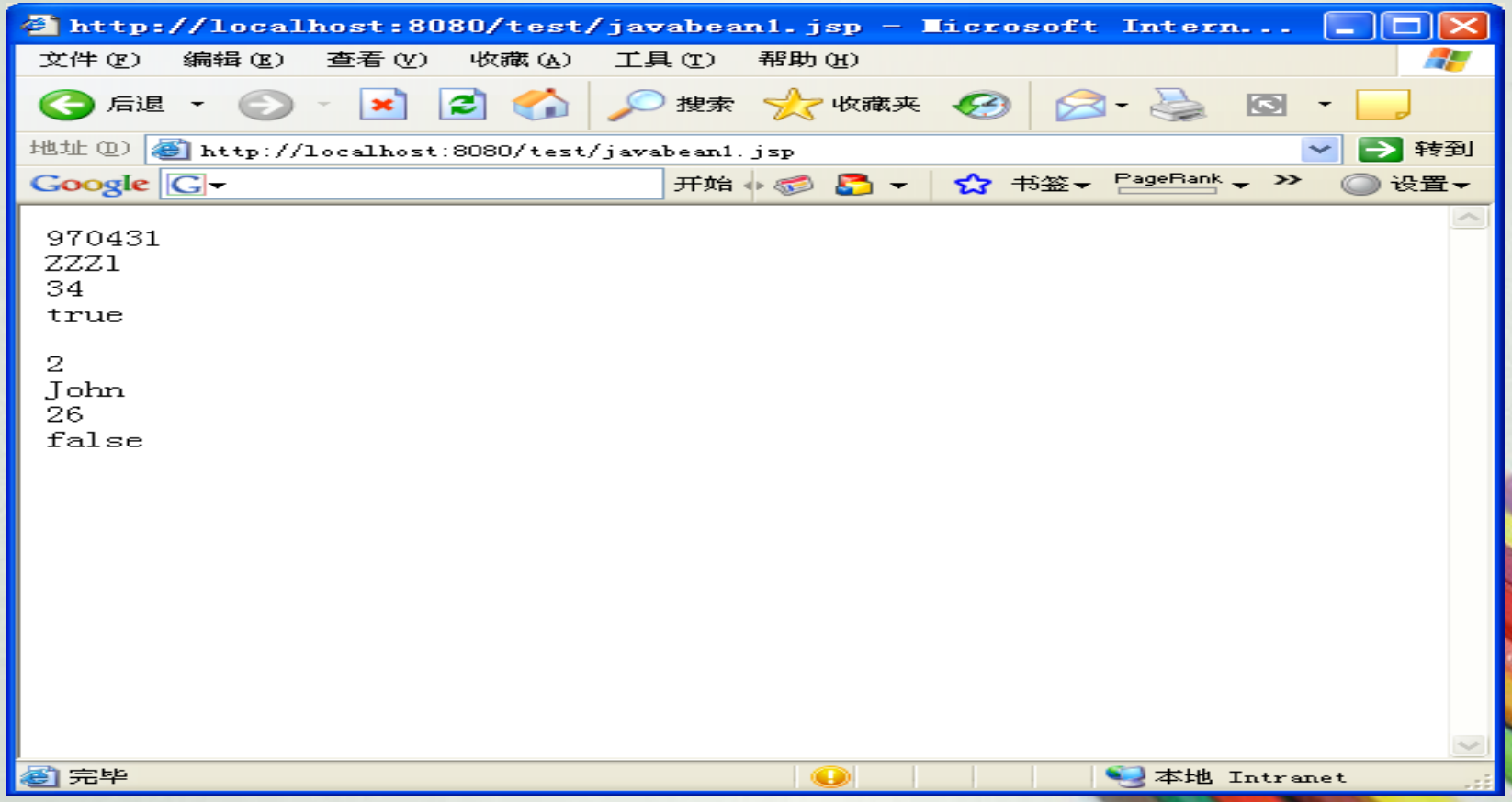


测试session作用域

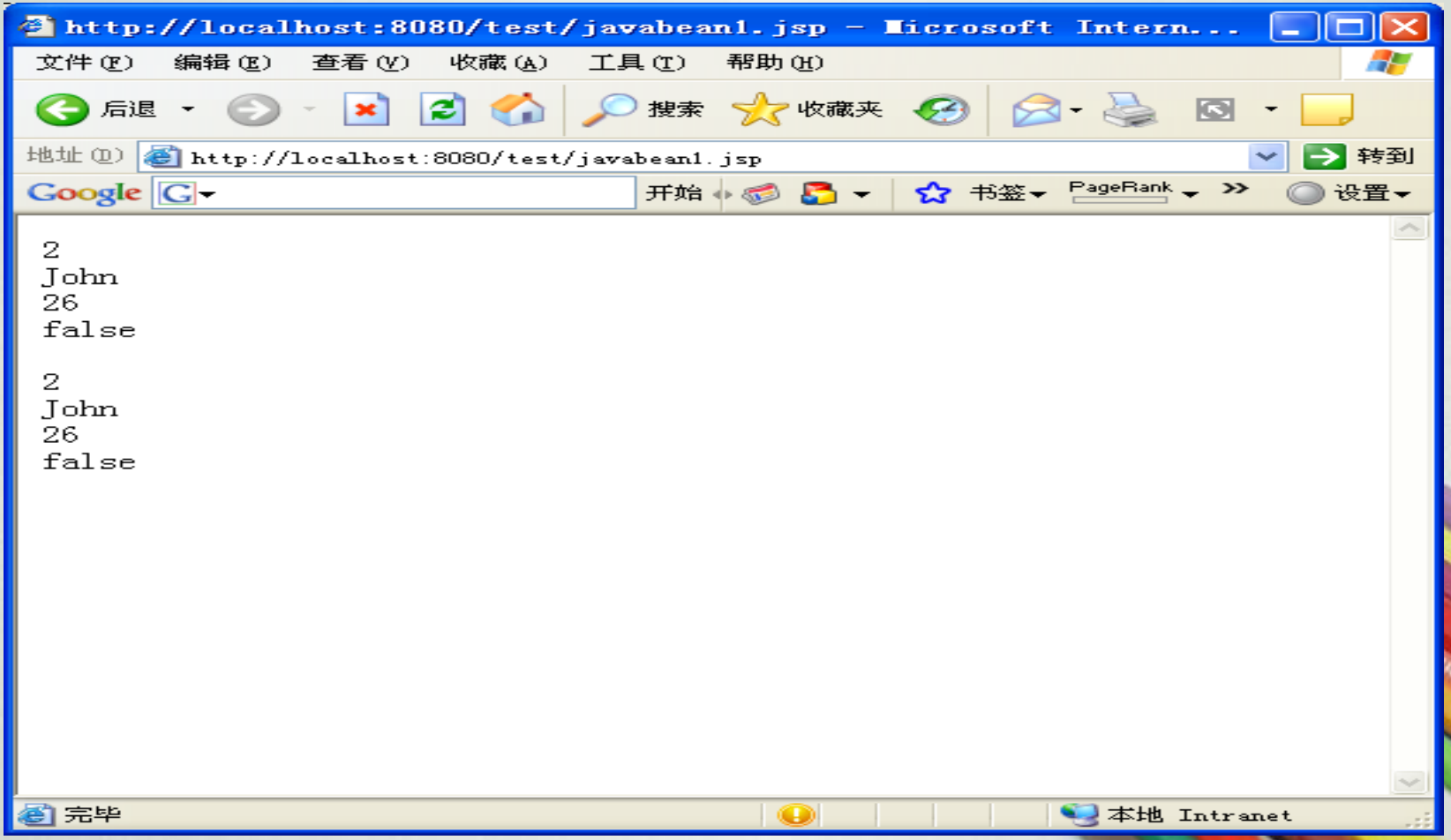
```
<%--JavaBean1.jsp文件，理解useBean动作的scope作用范围--
   ->
<%@ page contentType="text/html;charset=gb2312" %>

<html>
<body>
<%-- 引用自己编写的javaBean生成的bean组件 --%>
<jsp:useBean id="student" scope="session"
   class=“com.test.bean.student" />
```

```
<%-- 取属性值 --%>
<%=student.getClassNo() %> <br>
<%=student.getName() %> <br>
<%=student.getAge() %> <br>
<%=student.getSexy() %> <br>
<!--用JSP类属性设定bean组件的属性值-->
<%student.setClassNo(000002); %>
<%student.setName("John"); %>
<%student.setAge(26); %>
<%student.setSexy(false); %>
<br>
<%=student.getClassNo() %> <br>
<%=student.getName() %> <br>
<%=student.getAge() %> <br>
<%=student.getSexy() %> <br>
</body>
</html>
```

运行结果


http://localhost:8080/test/javabean1.jsp - Microsoft Intern...
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
后退 搜索 收藏夹
地址(D) http://localhost:8080/test/javabean1.jsp
转到
Google 开始 书签 PageRank 设置
970431
ZZZ1
34
true
2
John
26
false
完毕
本地 Intranet

刷新


http://localhost:8080/test/javabean1.jsp - Microsoft Intern...
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
后退 搜索 收藏夹
地址(D) http://localhost:8080/test/javabean1.jsp
转到
Google 开始 书签 PageRank 设置
2
John
26
false
2
John
26
false
完毕
本地 Intranet

小结与习题

- 本节集中介绍了JSP的各种语法，掌握这些语法是同学们进行JSP开发的基本要求。
- 随后的几章将各有重点的讲述与语法相关的一些问题。

习题

- 1．举例说明HTML注释与隐藏注释有何异同。
- 2．试写一个JSP文件，包含所有JSP语法元素
- 3．Scriptlet中应遵循什么样的语法规则？
- 4．<jsp:include>元素包含静态文件与包含动态文件处理上有何不同？
- 5．<jsp:forward>起到什么样的作用？使用时有哪些注意事项？
- 6．page指令能起到那些作用？有效范围有多大？