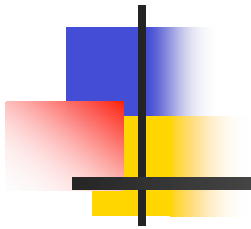
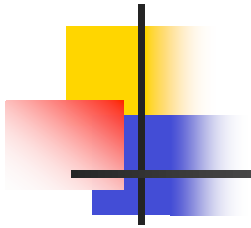


JSP Implicit Objects and EL

Name: Kaijie Sheng

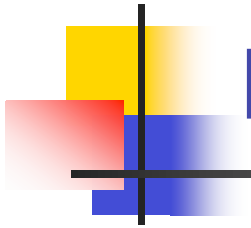
Personal Number: 830831-5053



Content

- 1. Scope of JSP objects
- 2. Implicit Objects
- 3. An JSP Object Application Example
- 4. Expression Language (EL)

1. Scope of JSP Objects

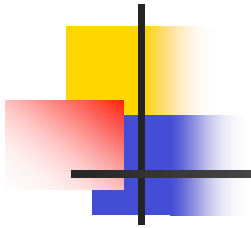


Definition on Scope of JSP Objects

The availability of a JSP object

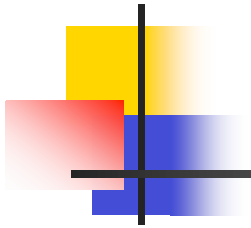
for use from a particular place of the application

is defined as the scope of that JSP object.



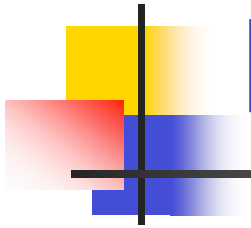
Kinds

- Object scope in JSP is segregated into four parts according to their available rang and life time。
- Page scope
- Request scope
- Session scope
- Application scope



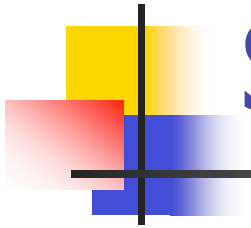
Page scope

- The JSP object can be accessed only from within the same page where it was created.
- The default scope for JSP objects created using <jsp:useBean> tag is page.
- JSP implicit objects out, exception, response, config, pageContext, and page have 'page' scope.



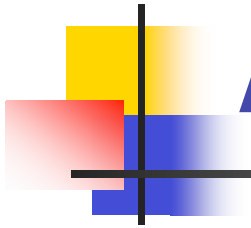
Request scope

- A JSP object created using the 'request' scope can be accessed from any pages that serves that request.
- The JSP object will be bound to the request object.
- Implicit object request has the 'request' scope.



Session scope

- the JSP object is accessible from pages that belong to the same session from where it was created.
- The JSP object that is created using the session scope is bound to the session object.
- Implicit object session has the 'session' scope.



Application scope

- A JSP object created using the ‘application’ scope can be accessed from any pages across the application.
- The JSP object is bound to the application object.
- Implicit object application has the ‘application’ scope.

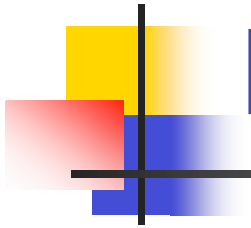
Operation on Attributes

Attribute operation method	Annotation
public void setAttribute(String name, Object ob)	Set the "name" as the reference of the object "ob"
public Object getAttribute(String name)	Get the value of attribute "name"
public void removeAttribute(String name)	Remove the attribute "name"
public Enumeration getAttributeNames ()	Get all the attributes within the available scope
public Object getAttribute(String name, int scope) public void setAttribute(String name, Object object, int scope) public void removeAttribute(String name, int scope) public Enumeration getAttributeNamesInScope(int scope)	Special method for PageContext. Used to access attribute "name" within scope " scope".

2. Implicit Objects

Overview

JSP Implicit Object	Super Class	Annotation
request	HttpServletRequest	Provides HTTP request information.
response	HttpServletResponse	Send data back to the client
out	JspWriter	Write the data to the response stream
session	HttpSession	Track information about a user from one request to another
application	ServletContext	Data shared by all JSPs and servlets in the application.
pageContext	PageContext	Contains data associated with the whole page
config	ServletConfig	Provides servlet configuration data.
page	Object	Similar with “this” in Java
exception	Throwable	Exceptions not caught by application code.



Request object

- Methods of request Object:
- getCookies()

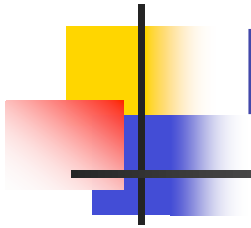
  return all cookies sent with the request information by the client

- getHeader(String name)

  return the value of the requested header

- getHeaderNames()

  return all the header names in the request



Request object

- Methods of request Object:
- getAttribute(String name)

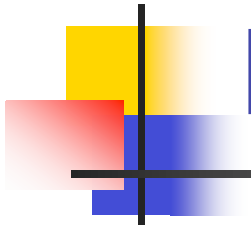
  return the value of the attribute

- getAttributeNames()

  return the object associated with the particular given attribute

- getMethod()

  return the methods GET, POST, or PUT corresponding to the requested HTTP method used



Request object

- Methods of request Object:
- getParameter(String name)

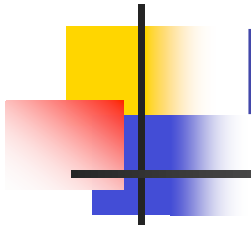
  return the value of a requested parameter

- getParameterNames()

  return the names of the parameters given in the current request

- getParameterValues(String name)

  return the value of a requested given parameter



Request object

- Methods of request Object:
- getQueryString()

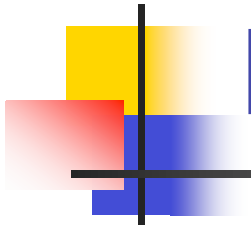
  return the query string from the request

- getRequestURI()

  return the URL of the current JSP page

- getServletPath()

  return the part of request URL that calls the servlet



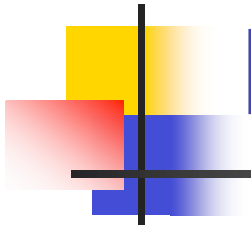
Request object

- Methods of request Object:
- setAttribute(String,Object)

  used to set object to the named attribute.

- removeAttribute(String)

  used to remove the object bound with specified name from the corresponding session



Response object

- Methods of Response Object:
- setContentType()

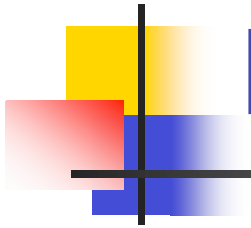
  set the MIME type and character encoding for the page

- addCookie(Cookie cookie)

  used to add the specified cookie to the response

- addHeader(String name, String value)

  write the header as a pair of name and value to the response



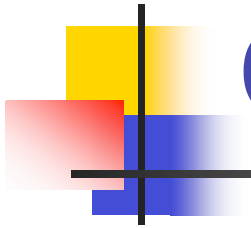
Response object

- Methods of Response Object:
- containsHeader(String name)
  check whether the response already includes the header given as parameter

- setHeader(String name, String value)
  create an HTTP Header with the name and value given as string

- sendRedirect(String)
  send a redirect response to the client temporarily by making use of redirect location URL given in parameter

- sendError(int status_code)
  send an error response to the client containing the specified status code given in parameter



Out object

- Methods of Out Object:
- Clear()

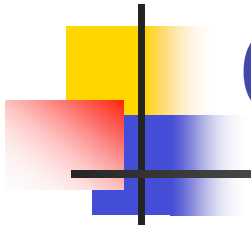
  clear the output buffer

- clearBuffer()

  clear the output buffer

- Flush()

  flush the buffer by writing the contents to the client



Out object

- Methods of Out Object:
- isAutoFlush

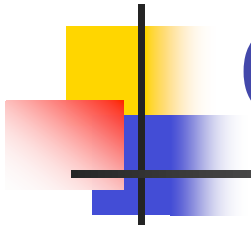
  return a true value if the output buffer is automatically flushed

- getBufferSize

  return the size of the buffer

- getRemaining

  return the number of empty bytes in the buffer



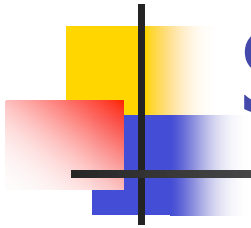
Out object

- Methods of Out Object:
- print

  write the value to the output without a newline character

- println

  write the value to the output, including the newline character



Session object

- Methods of Session Object:
- getAttribute(String name)

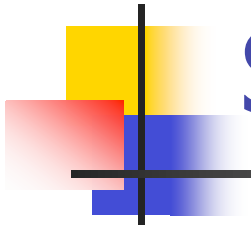
  return the object with the specified name given in parameter

- getAttributeNames()

  retrieve all attribute names associated with the current session

- isNew()

  return a true value if the session is new



Session object

- Methods of Session Object:
- getCreationTime

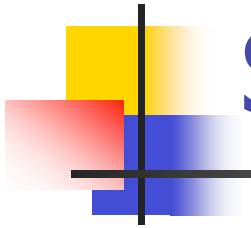
  return the session created time

- getLastAccessedTime

  return the latest time of the client request associated with the session

- getId

  return the unique identifier associated with the session.



Session object

- Methods of Session Object:
- invalidate()

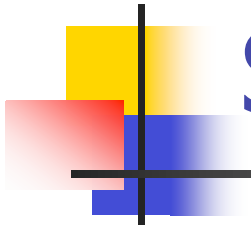
  discard the session and releases any objects stored as attributes

- getMaxInactiveInterval ()

  return the maximum amount of time the JRun keeps the session open between client accesses

- setMaxInactiveInterval()

  set the timeout explicitly for each session



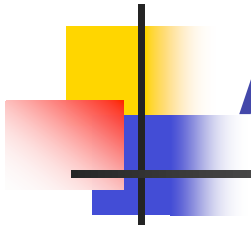
Session object

- Methods of Session Object:
- removeAttribute(String name)

  remove the attribute and value from the session

- setAttribute(String, object)

  set the object to the named attribute



Application object

- Methods of Application Object:
- getAttribute(String name)

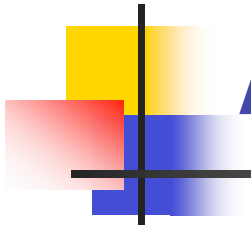
  return the attribute with the specified name

- getAttributeNames

  return the attribute names available within the application

- setAttribute(String objName, Object object)

  store the object with the given object name in the application



Application object

- Methods of Application Object:
- removeAttribute(String objName)

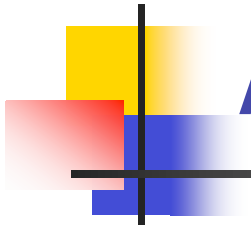
  remove the name of the object mentioned in parameter of this method from the object of the application

- getMajorVersion()

  return the major version of the Servlet API for the JSP Container

- getMinorVersion()

  return the minor version of the Servlet API for the JSP Container



Application object

- Methods of Application Object:
- getServerInfo()

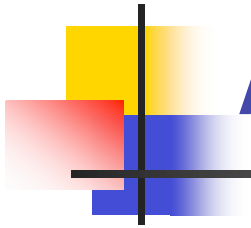
  return the name and version number of the JRun servlet engine

- getInitParameter(String name)

  return the value of an initialization parameter

- getInitParameterNames()

  return the name of each initialization parameter



Application object

- Methods of Application Object:
- getResourceAsStream(Path)

  translate the resource URL mentioned as parameter in the method into an input stream to read

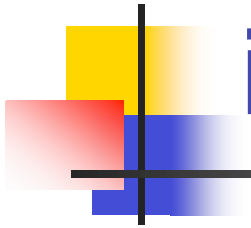
- log(Message)

  write a text string to the JSP Container's default log file

3. An Example about JSP Object

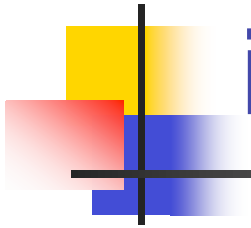
—— Simple Chat Room


Index - Windows Internet Explorer
http://localhost:8080/myapp/
百度
文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)
Index
页面(P) 工具(O)
Login
username: kaijie
password:
Login
cancel
完成
Internet
100%



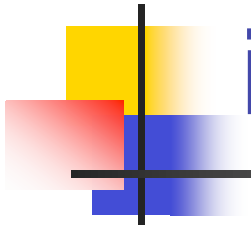
index.html

```
<html>
<head>
<title>Index</title>
</head>
<body>
<div align="center">
  <center>
  <p>    </p>
<p><font face="Times New Roman"
    size="6"><b>Login</b></font></p>
<table border="2" width="400" bordercolorlight="#FFFFFF"
    bordercolordark="#000000" cellspacing="0" cellpadding="0">
```



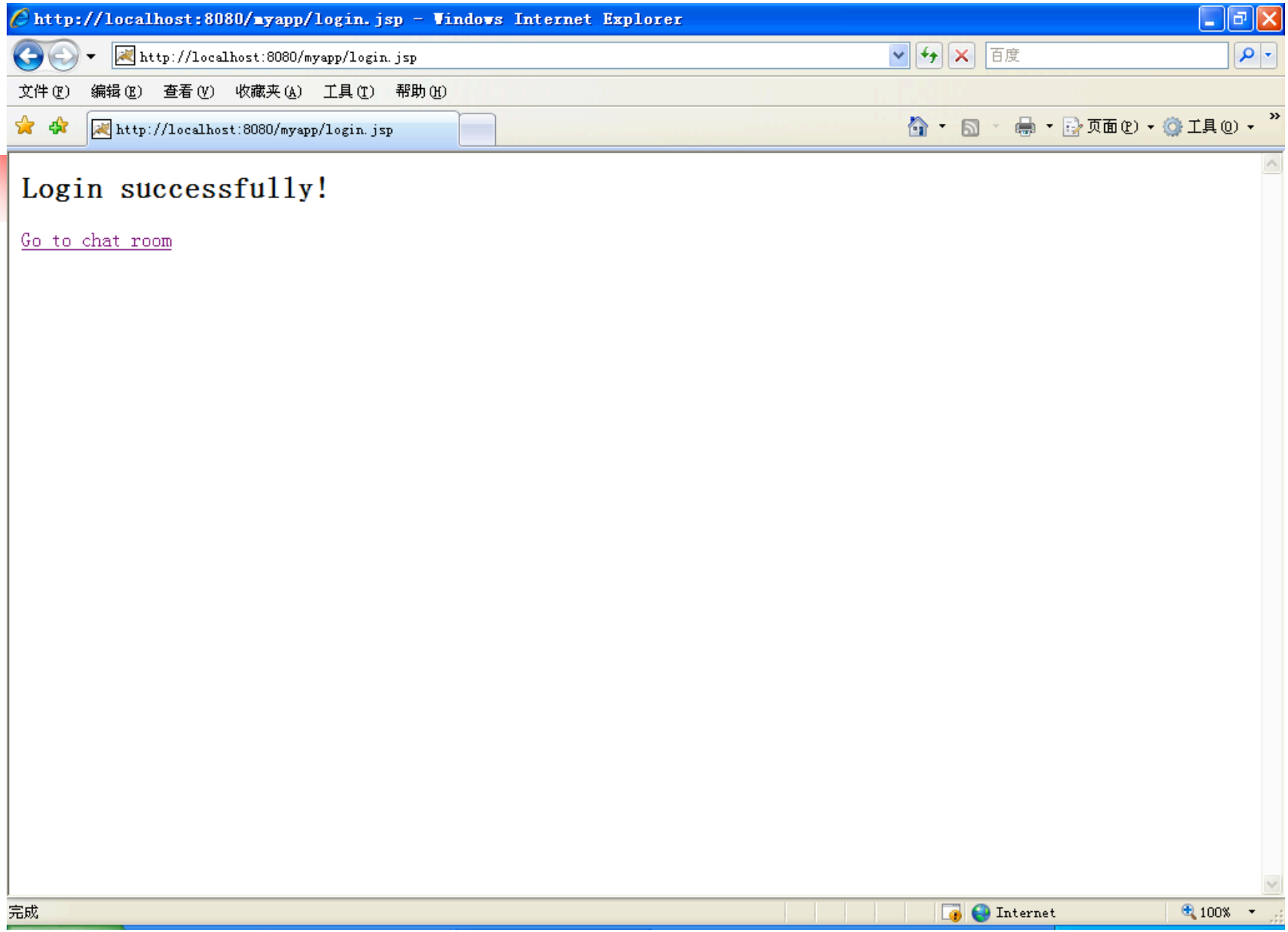
index.html

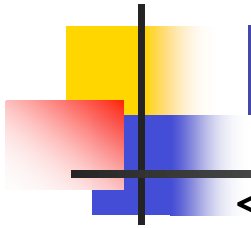
```
<tr>
   <td width="100%">
    <form method="POST" action="login.jsp"
   name="login_form">
      <p align="center">    </p>
         <p align="center">username：  <input type="text"
   name="userName" size="20"></p>
      <p align="center">password：  <input type="password"
   name="passwd" size="20"></p>
      <p align="center">
         <input class="buttonface" type="submit" value="Login"
   name="B1">  
```



index.html

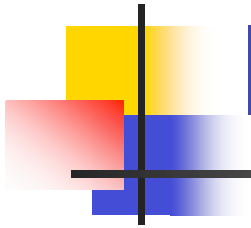
```
<input class="buttonface" type="reset" value="cancel"
    name="B2"></p>
      </form>
</td>
  </tr>
</table>
  </center>
</div>
</body>
</html>
```


http://localhost:8080/myapp/login.jsp - Windows Internet Explorer
http://localhost:8080/myapp/login.jsp
百度
文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)
http://localhost:8080/myapp/login.jsp
页面(P) 工具(O)
Login successfully!
Go to chat room
完成
Internet
100%



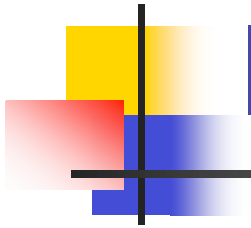
login.jsp

```
<%@ page session="true" contentType="text/html;charset=GBK"
    %>

<%
//init session as false
session.setAttribute("login","false");

//get username and password
String userName,passwd;
userName=request.getParameter("userName");
passwd=request.getParameter("passwd");

//remove redundant space
userName=userName.trim();
passwd=passwd.trim();
```



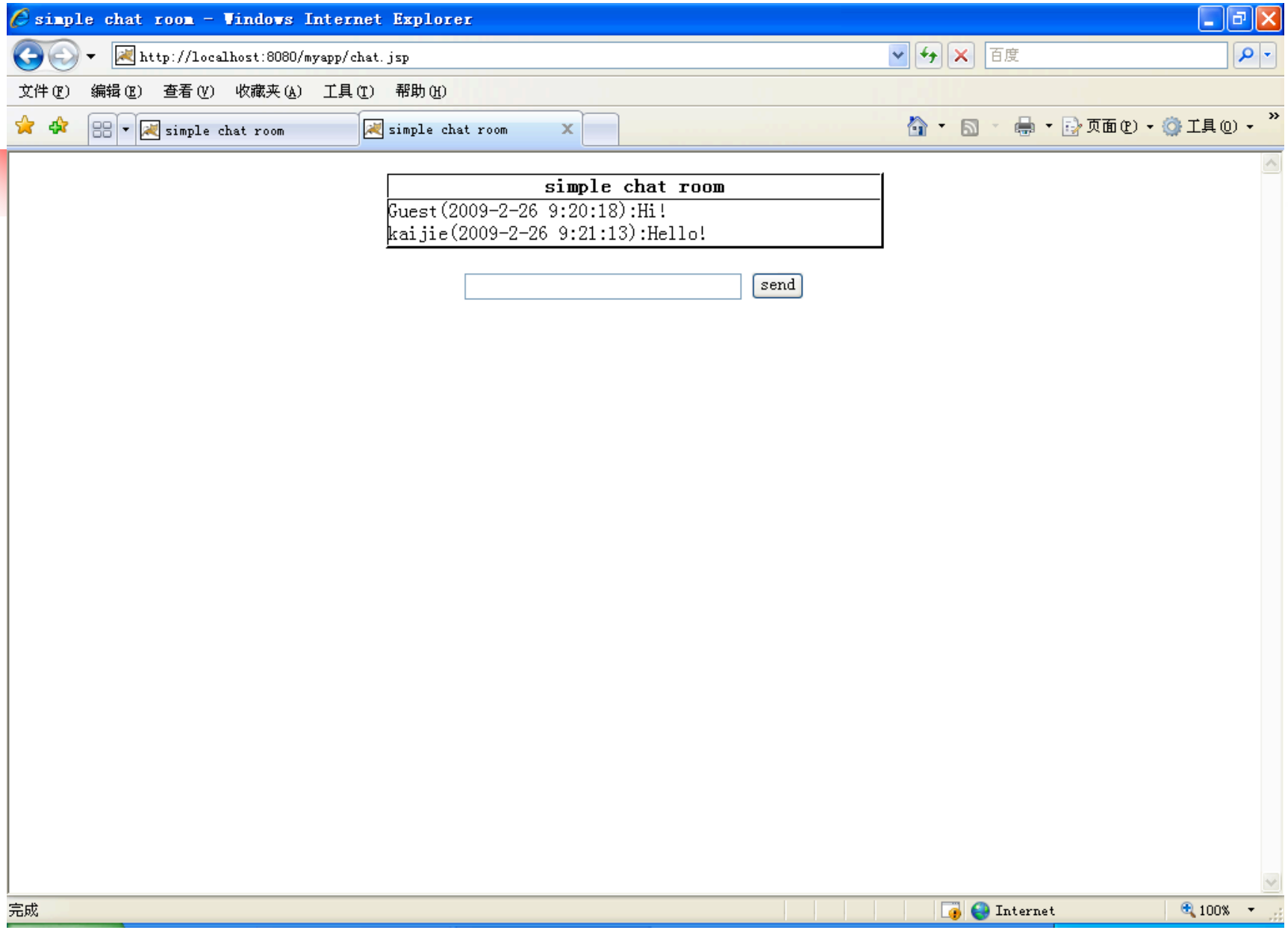
login.jsp

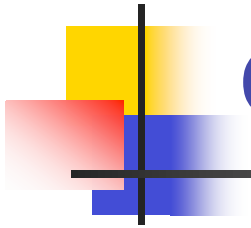
```
//if username or password is null, relogin
if ((userName==null)||(passwd==null)) {
    response.sendRedirect("./index.html");
}

else{
    // both username and password are correct
    if (userName.equals("kaijie")&&passwd.equals("123")){
        session.setAttribute("login","true");//set attribute "login" as
    true in session
        session.setAttribute("userName",userName);
%>
```



login.jsp

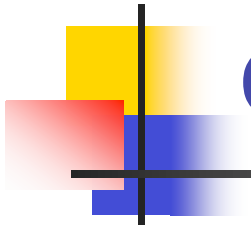
```
            <h2>Login successfully!</h2>
            <a href="chat.jsp">Go to chat room</a>

<%
    }//if
    else{
            response.sendRedirect("./index.html");
    }
}//else
%>
```


simple chat room - Windows Internet Explorer
http://localhost:8080/myapp/chat.jsp
百度
文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)
simple chat room
simple chat room
页面(P) 工具(O)
simple chat room
Guest(2009-2-26 9:20:18):Hi!
kaijie(2009-2-26 9:21:13):Hello!
send
完成
Internet
100%



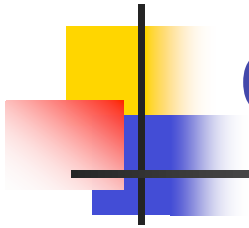
chat.jsp

```
<%@ page contentType="text/html;charset=GBK"
    import="java.util.*"%>
<html>
<head>
<title>simple chat room</title>
</head>
<body>
<center>
<%
if (request.getProtocol().compareTo("HTTP/1.0") == 0)
    response.setHeader("Pragma", "no-cache");
else if (request.getProtocol().compareTo("HTTP/1.1") == 0)
    response.setHeader("Cache-Control", "no-cache");
```



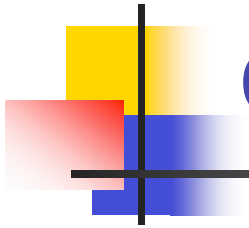
chat.jsp

```
response.setDateHeader("Expires", -1);

response.setHeader("Refresh","10");

//use session object，get current username
String userName = (String)session.getAttribute("userName");

//If the user isn't a login user，then set user as "Guest"
if (userName == null )
    userName = "Guest";

//use request object get input chat data from the input textfield
String curChat = request.getParameter("inputChatTextField");
```



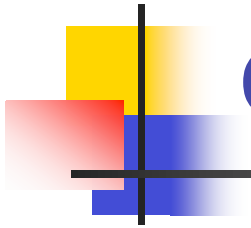
chat.jsp

```
//use application object store the chat data
String chatRecord = (String)application.getAttribute("chatRecord");

//add time,username and user input into chat data
if (curChat!= null){
    Date d = new Date();
    curChat = userName + "(" + d.toLocaleString() + "):" +
    curChat;
    if (chatRecord == null)
            chatRecord = curChat;
    else
            chatRecord = chatRecord + "<br>"  + curChat;
    }
```



chat.jsp

```
//show char data within a table
if (chatRecord!=null){
    application.setAttribute("chatRecord", chatRecord); %>
<table border="2" width="400" bordercolorlight="#FFFFFF"
    bordercolordark="#000000" cellspacing="0" cellpadding="0">
    <tr>
        <th>simple chat room</th>
    </tr>
    <tr>
        <td width="100%"><%=
    application.getAttribute("chatRecord")%></td>
    </tr>
</table>
<%}%>
```



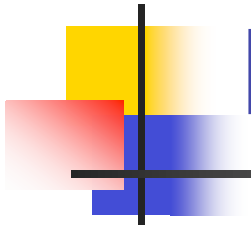
chat.jsp

```
<FORM ACTION="chat.jsp" METHOD="post">
<p><INPUT TYPE="TEXT" SIZE="30" NAME="inputChatTextField"
   VALUE="">
<INPUT TYPE="SUBMIT" name="submit" VALUE="send"></p>
</FORM>
</center>
</BODY>
</HTML>
```

- 4. Expression Language (EL)

- **Expression Language** (EL) is used to extend the "coding capability" of a JSP, thereby reducing the requirement for Java-scriptlets.

- You reach a higher level of abstraction



Lane Syntax

- ${expr}

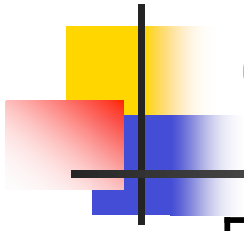
  expr stands for a valid expression

- Valid Expression:
  - Literals
  - Operators
  - Variables (object references)
  - Function call

Literals

Literals	Literal values
Boolean	true and false
Integer	Any positive or negative number e.g, 24, -45, 567
Floating	Any positive or negative floating point numbers e.g, -1.8E-45, 4.567
String	Any string delimited by single or double quotes.
Null	null

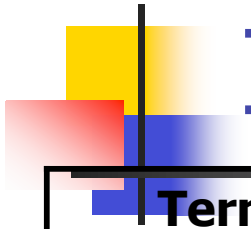
e.g. **${false} <%-- evaluates to false --%>**
${3*8)



Operator

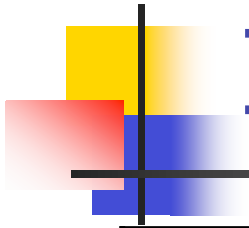
Term	Definition
Arithmetic	+, - (binary), *, /, div, %, mod, - (unary)
Logical	and, &&, or, ||, !, not
Relational	==, eq, !=, ne, <, lt, >, gt, <=, le, >=
Empty	used to determine if a value is null or empty.
Conditional	A ? B : C. Evaluate B or C, depending on the result of the evaluation of A.

e.g. ${ (6 * 5) + 5 } <%-- evaluates to 35 --%>
${empty name}



Implicit Object

Term	Definition
pageContext	**used to access the JSP implicit objects such as request, response, session, out, servletContext etc.** **${pageContext.response} evaluates to the response object for the page.**
param	**maps a request parameter name to a single String parameter value,** **$(param.name) is equivalent to request.getParameter (name).**
paramValues	**maps a request parameter name to an array of values,** **${paramvalues.name) is equivalent to request.getParamterValues(name).**
header	**maps a request header name to a single String header value. ${header.name} is equivalent to request.getHeader(name).**
headerValues	**maps a request header name to an array of values .** **${headerValues.name} is equivalent to request.getHeaderValues(name).**



Implicit Object

Term	Definition
cookie	maps cookie names to a single cookie object
initParam	maps a context initialization parameter name to a single value
pageScope	maps page-scoped variable names to their values
requestScope	maps request-scoped variable names to their values
sessionScope	maps session-scoped variable names to their values
applicationScope	maps application-scoped variable names to their values

Thank you very much !