Module 5: EL, JSTL and Custom Tags

IMC Institute

Thanisa Kruawaisayawan

Thanachart Numnonda

www.imcinstitute.com

Objectives

- Expression Language
- JSTL (JSP Standard Tag Library) 1.1
- Custom Tags

Expression Language

■ EL expressions are **ALWAYS** within curly braces, and prefixed with the dollar sign

```
${firstThing.secondThing}
```

■ **firstThing** can be
- EL Implicit Object
- Attribute

EL Implicit Object and Attribute

- **EL Implicit Object**

  - `param`
  - `paramValues`

  - `header`
  - `headerValues`

  - `cookie`

  - `initParam`

  - `pageContext`

  - `pageScope`
  - `requestScope`
  - `sessionScope`
 - `applicationScope`

- **Attribute**

  - in page scope
  - in request scope
  - in session scope
  - in application scope

EL Implicit Objects

Implicit Object	Description
`param` `paramValues`	Maps of all the **form parameters** that were passed to your JSP
`header` `headerValues`	Maps of all the **request headers**
`cookie`	A Map of all the **cookies** passed to your JSP
`initParam`	A Map of the **context init parameters**

`pageScope`
`requestScope`
`sessionScope`
`applicationScope`
	A Map of all the objects that have page, request, session and application scope

hello.html

```
<form action="helloName.jsp" method="post">
   Name: <input name="username">
    <input type="submit">
</form>
```
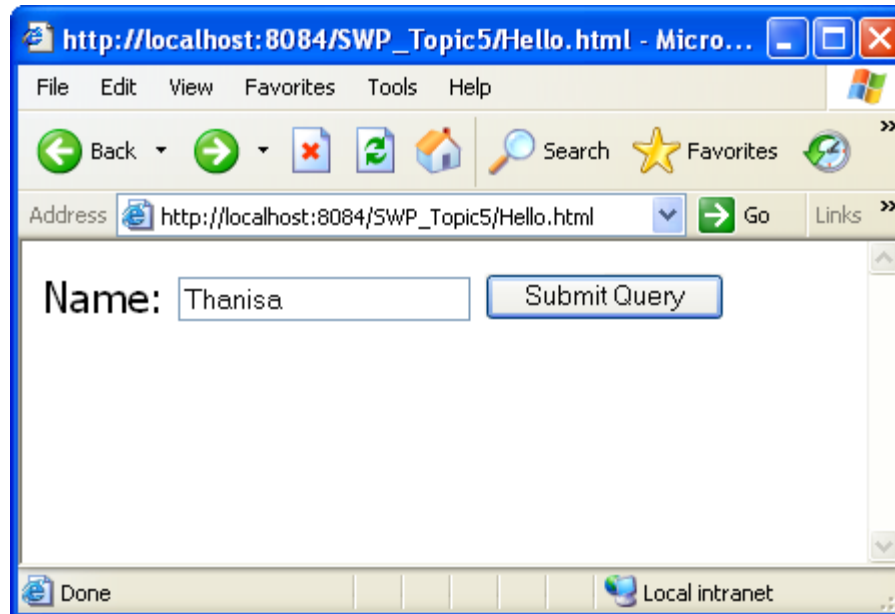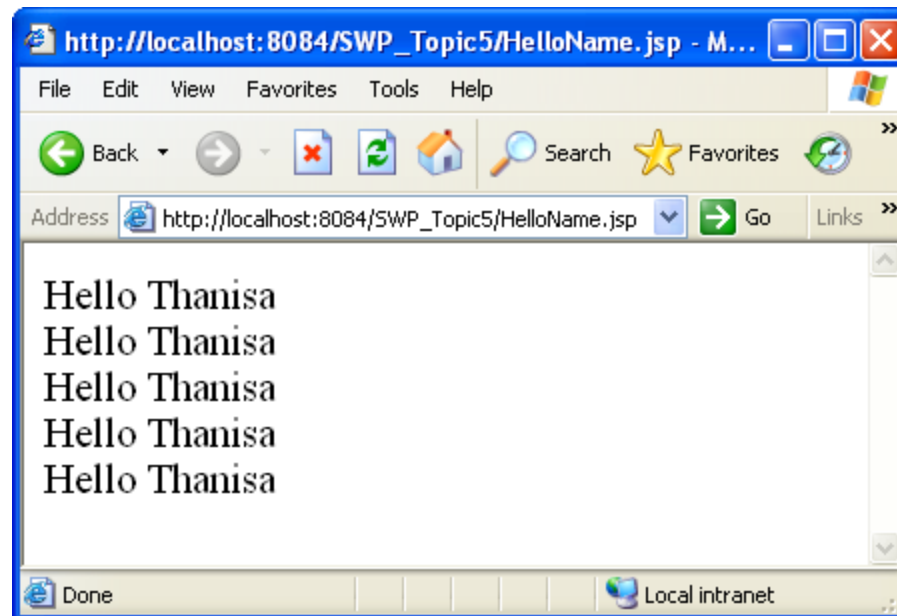
param

```jsp
<%-- helloName.jsp --%>
Hello <%= request.getParameter("username") %><br>
Hello <% out.print(request.getParameter("username"));%><br>
Hello ${param.username}<br>
Hello ${param['username']}<br>
Hello ${param["username"]}<br>
```
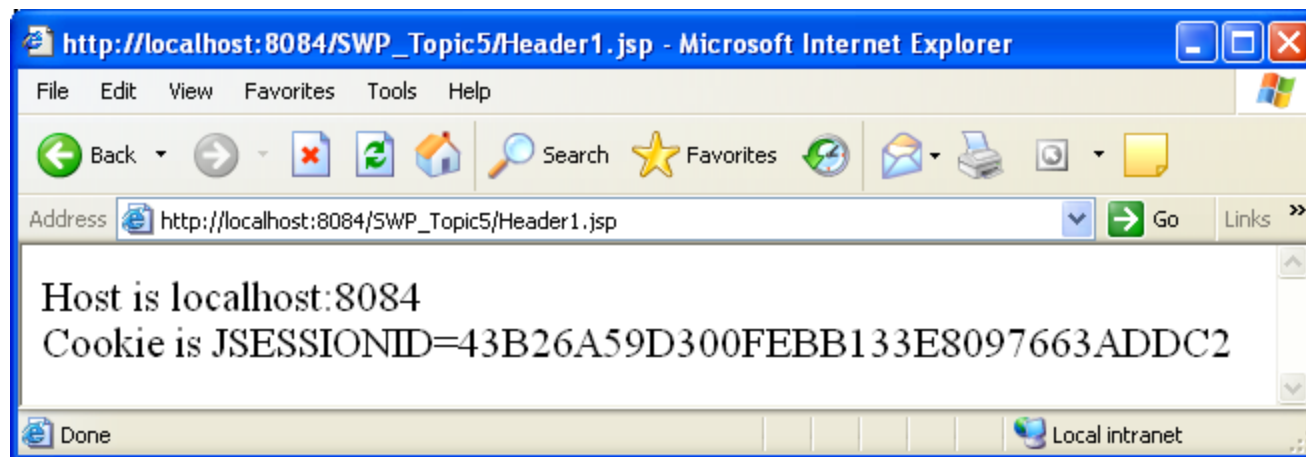
header
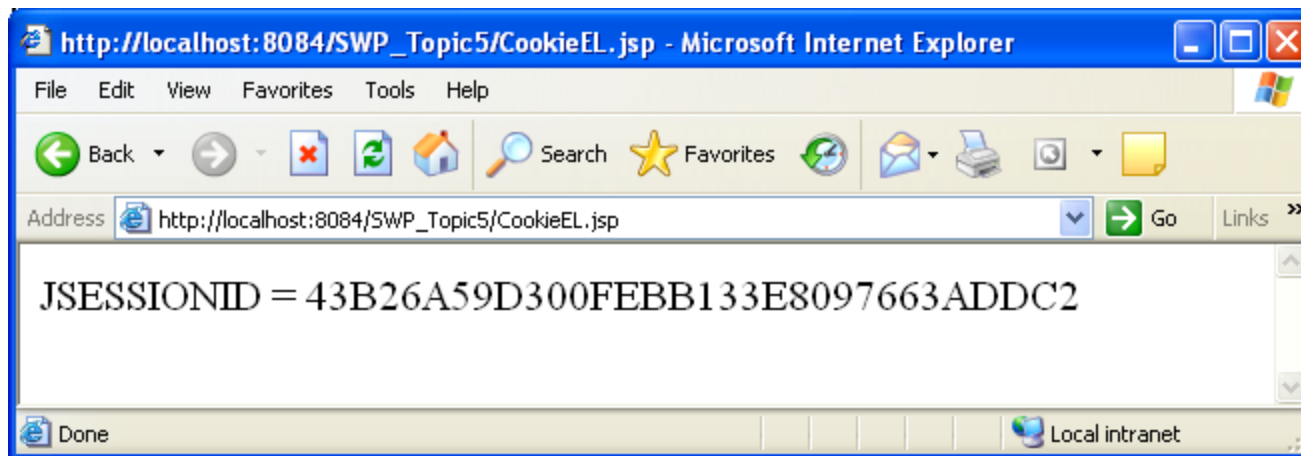
`Host is ${header.host} <br>`

`Cookie is ${header.cookie}`

cookie

`JSESSIONID = ${``cookie``.JSESSIONID.value}`

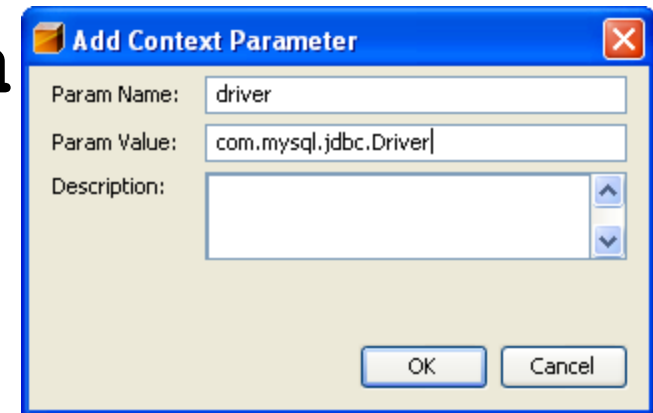# initParam
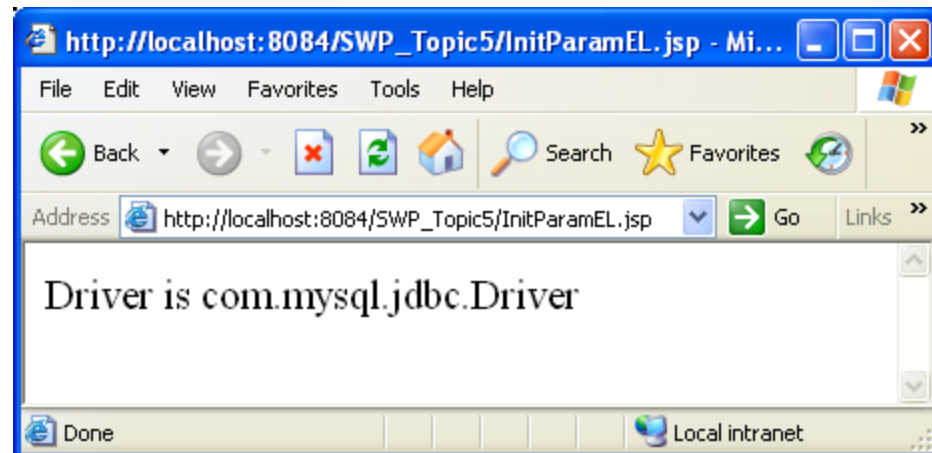

//web.xml
```
<web-app ...>
    :
    <context-param>
        <param-name> driver </param-name>
        <param-value> com.mysql.jdbc.Driver </param-value>
    </context-param>
    :
</web-app>
```

//InitParamEL.jsp
```
Driver is ${initParam.driver}
```
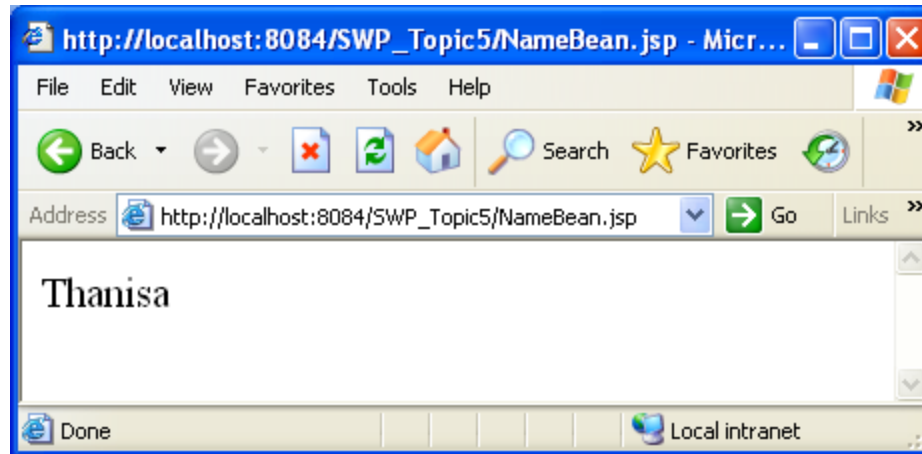

10

Person.java

```java
package myBeans;

public class Person {
    private String name;
    public void setName(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
}
```

EL and JavaBeans

```
<jsp:useBean id="personName" class="myBeans.Person" />
<jsp:setProperty name="personName" property="name" value="Thanisa" />


<%-- <jsp:getProperty name="personName" property="name" /> --%>


${personName.name}
```

Disable the EL

- ## For a single page

```
<%@ page isELIgnored="true" %>
```

- ## For an entire application

```
<jsp-property-group>
    <url-pattern>*.jsp</url-pattern>
    <el-enabled>false</el-enabled>
    <scripting-invalid>true</scripting-invalid>
</jsp-property-group>
```

13

The `taglib` Directive

- Tag libraries come in two different flavors:
    - JSTL (**J**avaServerPages **S**tandard **T**ag **L**ibrary)
    - Custom Tag Libraries

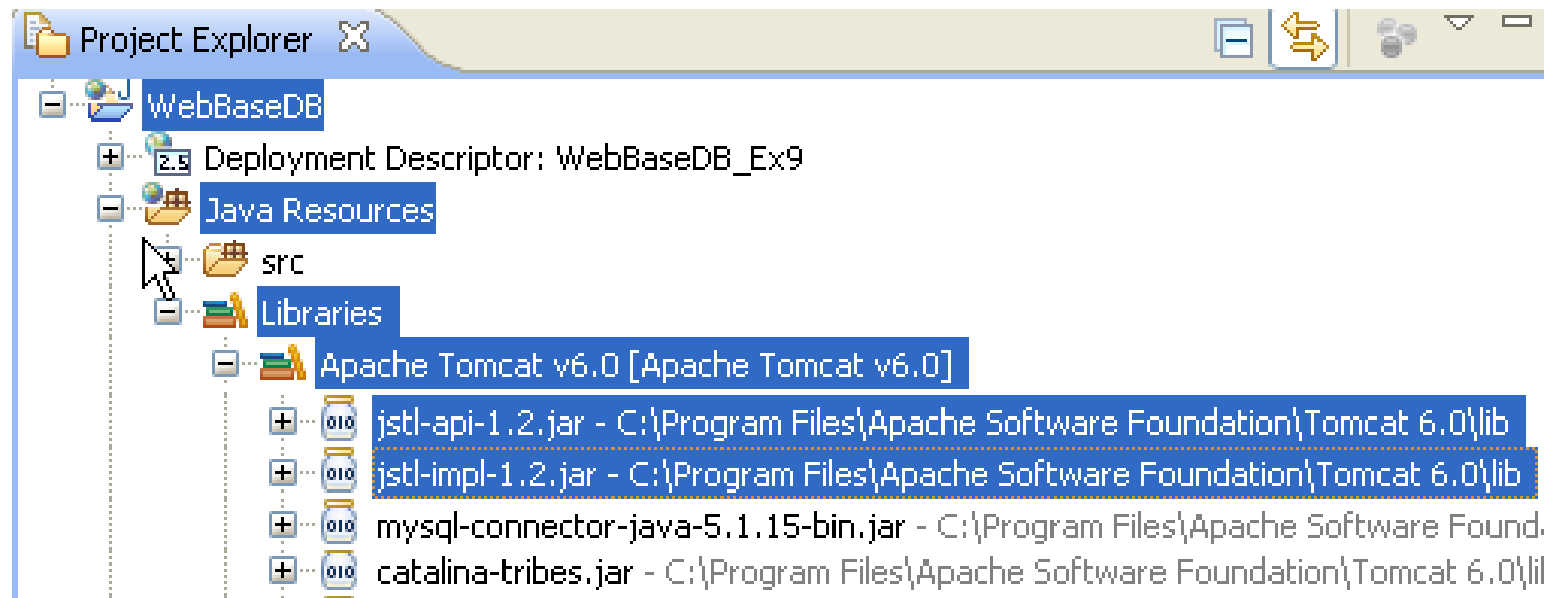- The syntax for the taglib directive is as follows:

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

JSTL 1.1

- The "Core" library
  - Looping and Iteration
    - **`<c:forEach>`**
    - **`<c:forTokens>`**
  - Conditional
    - **`<c:if>`**
    - **`<c:choose>`**
    - **`<c:when>`**
    - **`<c:otherwise>`**
  - General-purpose
    - **`<c:out>`**
    - **`<c:remove>`**
    - **`<c:catch>`**
  - URL related
    - **`<c:import>`**
    - **`<c:url>`**
    - **`<c:redirect>`**
    - **`<c:param>`**

- The "SQL" library
  - Database access
    - **`<sql:query>`**
    - **`<sql:update>`**
    - **`<sql:setDataSource>`**
    - **`<sql:param>`**
    - **`<sql:dateParam>`**

- The "Formatting" library
  - **`<fmt:message>`**
    :

- The "XML" library
  - **`<x:parse>`**
    :

Download and copy JSTL Libraries to Tomcat

16
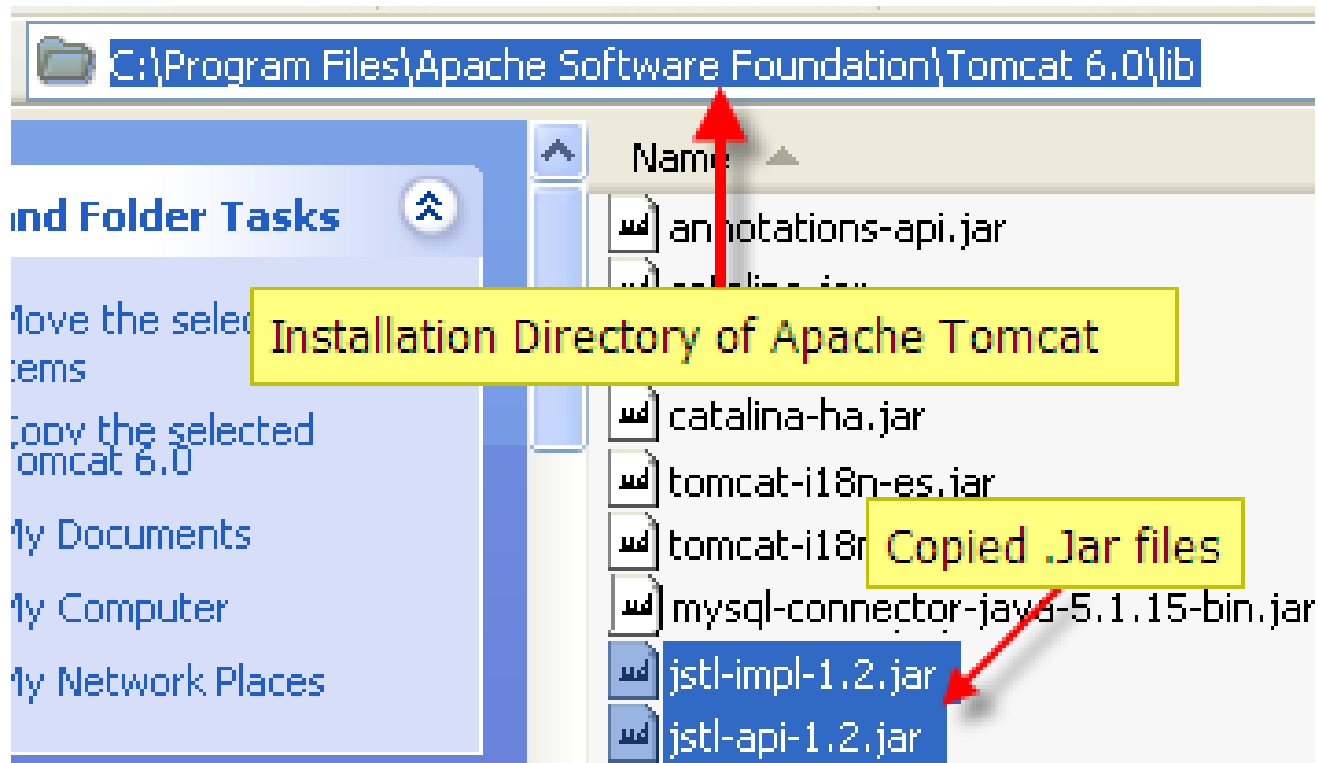
Check JSTL in Project

Core Tag Library

- Looping and Iteration
- Conditional
- General-purpose
- URL related

Looping and Iteration

- **`<c:forEach>`**
- **`<c:forTokens>`**

<c:forEach>

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<table border="1" align="center">
    <tr bgcolor="orange">
        <th>Header Names and Values</th>
    </tr>
    <c:forEach var="h" items="${header}">
        <tr>
            <td>${h.value}</td>
        </tr>
    </c:forEach>
</table>
```
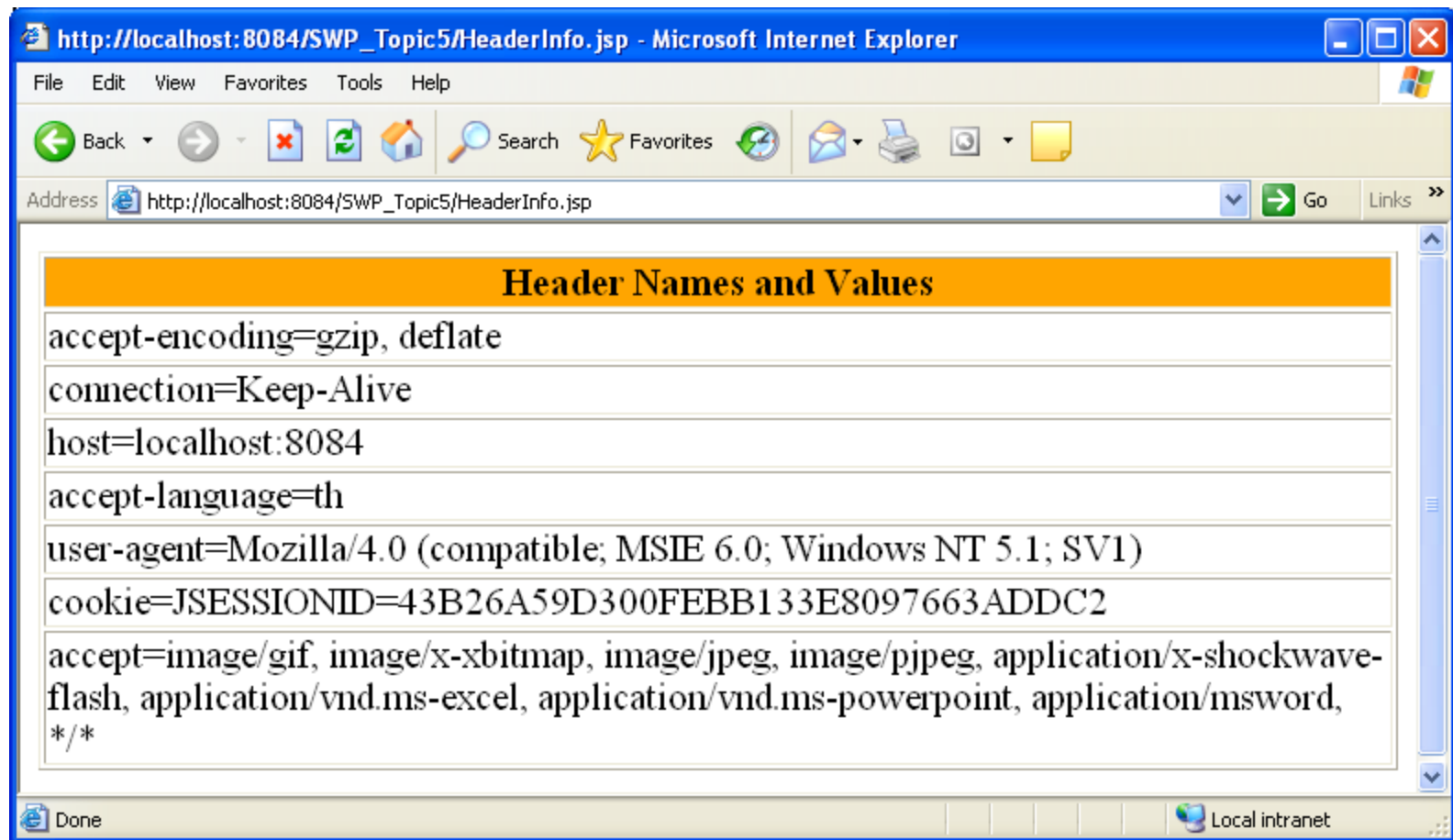
Result

Conditionals

- <c:if>
- <c:choose>, <c:when> and <c:otherwise>

Example

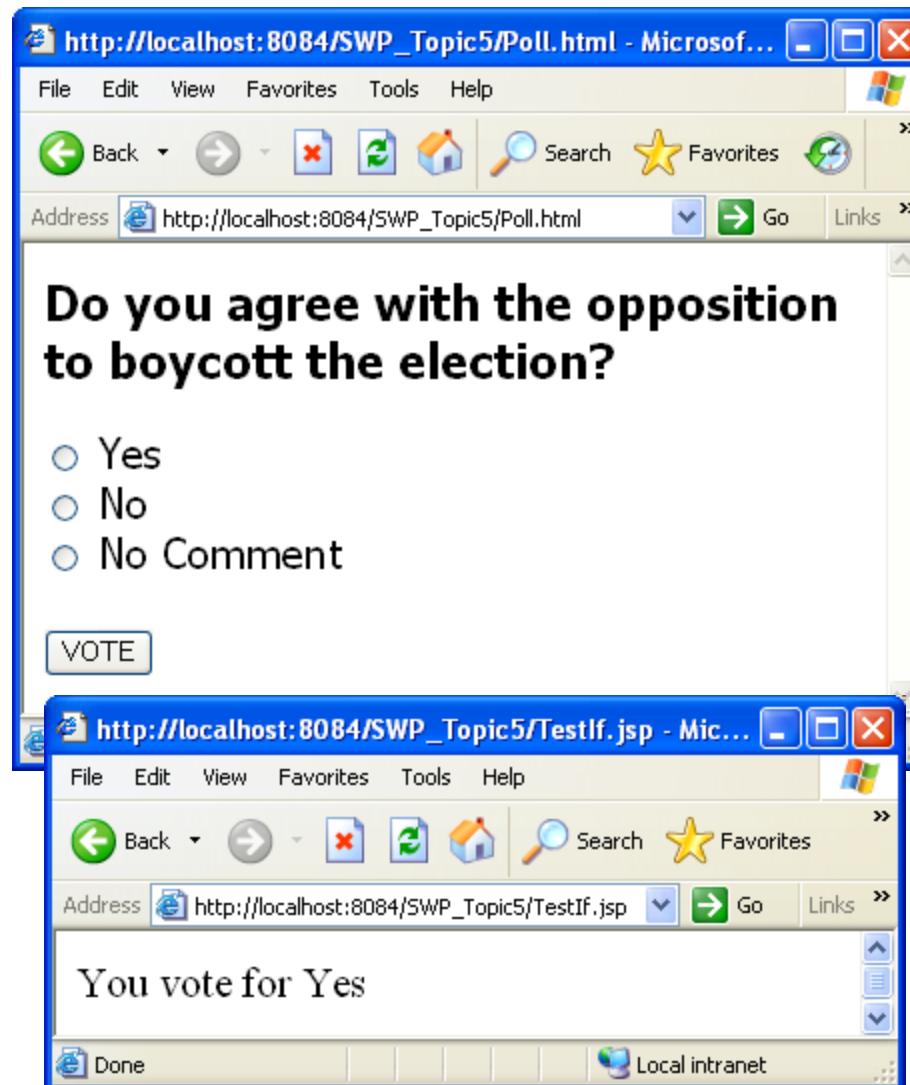# poll.html

```
:
<body>
   <h3>Do you agree with the opposition
       to boycott the election?</h3>
    <form action = "vote.jsp" method = "post">
      <input type=radio name=answer value="Yes"> Yes<br>
      <input type=radio name=answer value="No"> No<br>
      <input type=radio name=answer value="No Comment"> No Comment<br>
      <br><input type=submit value="VOTE">
    </form>
</body>
:
```

<c:if>

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core"
  prefix="c" %>

<c:if test="${param.answer == 'Yes'}" >
    You vote for Yes
</c:if>
<c:if test="${param.answer == 'No'}" >
    You vote for No
</c:if>
<c:if test="${param.answer == 'No Comment'}" >
    You vote for No Comment
</c:if>
```

<c:choose>, <c:when> and <c:otherwise>

```jsp
<%@taglib uri="http://java.sun.com/jsp/jstl/core"
  prefix="c" %>

<c:choose>
    <c:when test="${param.field == 'Network'}" >
        You choose Network
    </c:when>
    <c:when test="${param.field == 'Database'}" >
        You choose Database
    </c:when>
    <c:otherwise>
        You choose Programming
    </c:otherwise>
</c:choose>
```

Scoped Variable Manipulation

- <c:out>

- <c:set>

- <c:remove>

- <c:catch>

<c:out>

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<c:forEach begin="3" end="15" step="3" var="index" varStatus="num">
    <c:out value="${index}" />: <c:out value="${num.count}" /><br>
</c:forEach>
```
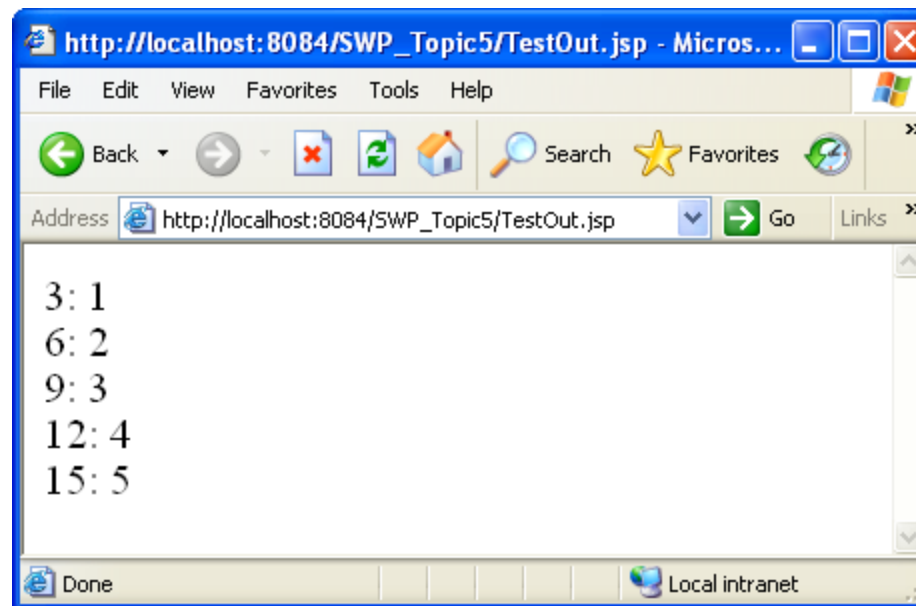
\<c:set> and \<c:remove>

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<c:set var="status" scope="request" value="On-line" />
Status is ${status} <br>

<c:remove var="status" scope="request" />
Now, status is ${status}
```
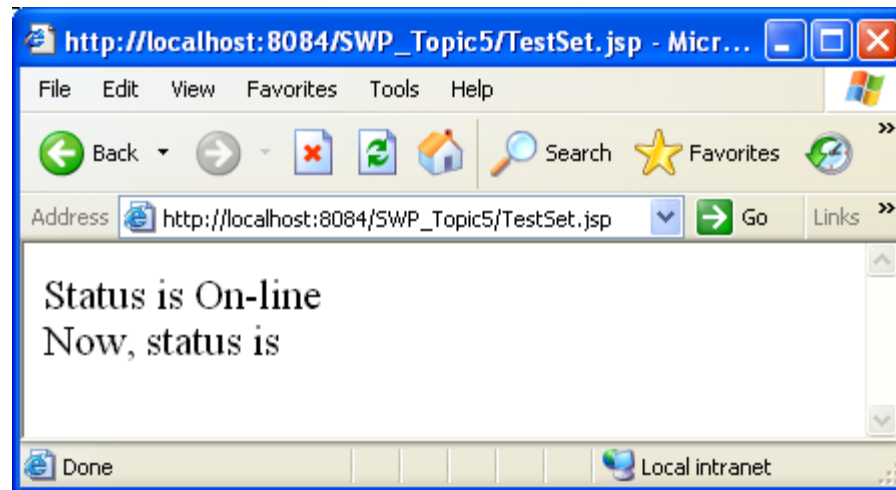
<c:catch>

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

About to do a risky thing <br>
<c:catch var="myException" >
    <% int x = 10/0; %>
</c:catch>
<c:if test="${myException != null}">
    There was an exception: ${myException.message} <br>
</c:if>
If you see this, we survived.
```
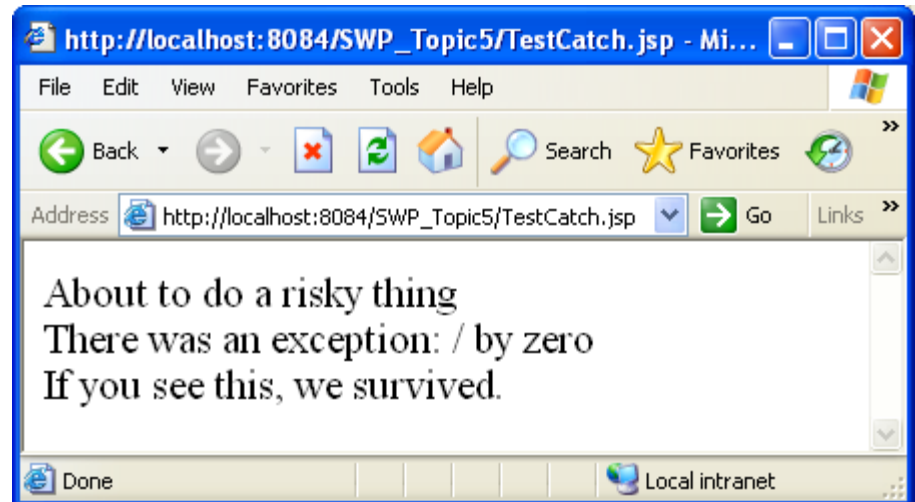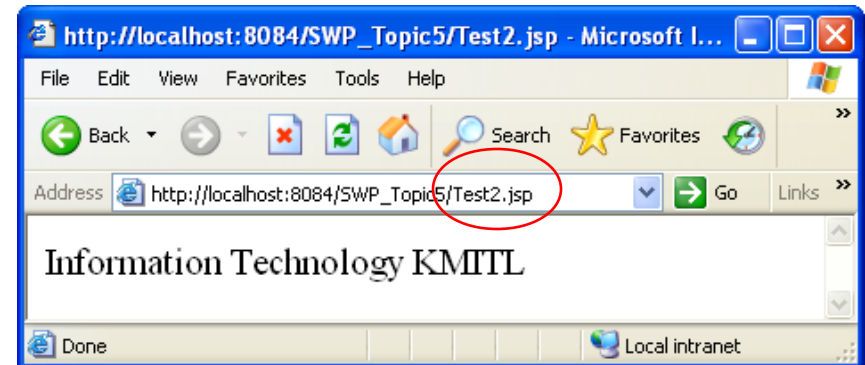
URL Manipulation

- \<c:import\>
- \<c:redirect\>
- \<c:url\>
- \<c:param\>

\<c:import\>

```
<%-- Header2.jsp --%>
Information Technology KMITL
--------------------------------------------------------------
<%-- Test2.jsp --%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<c:import url="http://localhost:8084/SWP_Topic5/Header2.jsp" />
--------------------------------------------------------------
```

- The \<c:import\> action can also be used to specify absolute, relative and FTP URL resources to provide a lot more functionality than the standard \<jsp:include\> action.
- \<c:import\> can reach OUTSIDE the web app

<c:redirect>

```
<%-- Header3.jsp --%>
Information Technology KMITL
-------------------------------------------------------------
<%-- Test3.jsp --%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<c:redirect url="http://localhost:8084/SWP_Topic5/Header3.jsp" />
-------------------------------------------------------------
```

- The <c:redirect> action simply sends an HTTP redirect to a client.
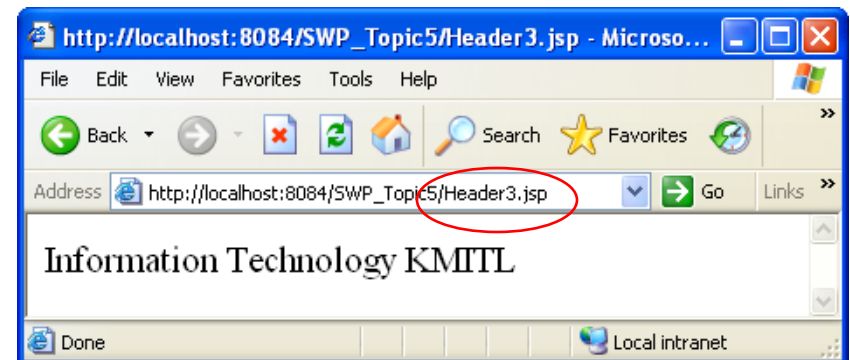
\<c:url\> and \<c:param\>

```
<%-- Header4.jsp --%>
${param.faculty}
-------------------------------------------------------------
<%-- Test4.jsp --%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<c:url value="http://localhost:8084/SWP_Topic5/Header4.jsp " >
    <c:param name="faculty" value="Information Technology" />
</c:url>
-------------------------------------------------------------
```

- The \<c:url\> action takes care of the encoding and all the URL rewriting.

```
http://localhost:8084/JSP2/Header.jsp;jsessionid=543ferew432esd23
```


34
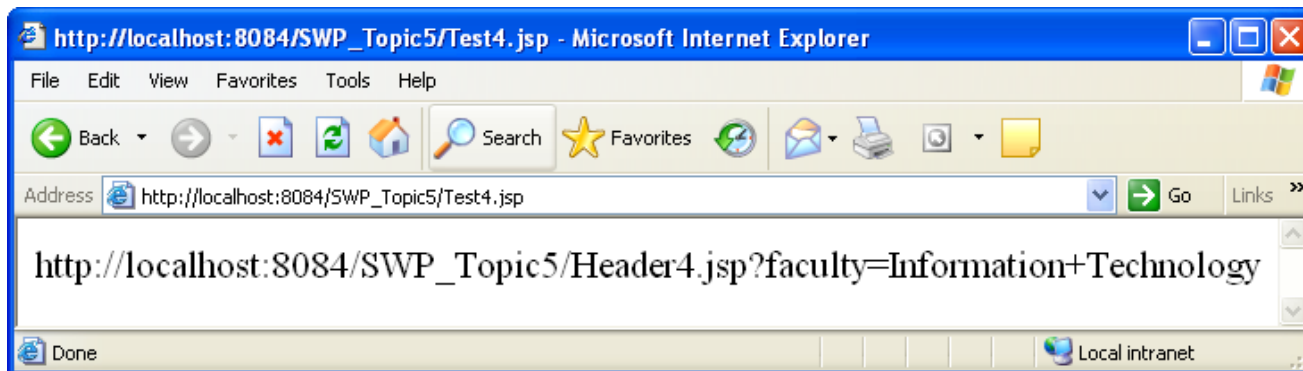
SQL Tag Library

- <sql:setDataSource>
- <sql:query>
- <sql:update>
- <sql:param> and <sql:dateParam>
- <sql:transaction>

BookStore.jsp

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>

<html>
    <head><title>ABC Book Store </title></head>
    <body>
     <center>
      <form action="BookStore.jsp">
      <h1> ABC Book Store </h1>
      <br> Please select a Book and add it to your shopping cart
      </p>

      <sql:setDataSource var="datasource"
              driver="com.mysql.jdbc.Driver"
    url="jdbc:mysql:///test"                          user="root"
    password="root" />

      <sql:query var="books" dataSource="${datasource}" >
              select * from books
      </sql:query>
```
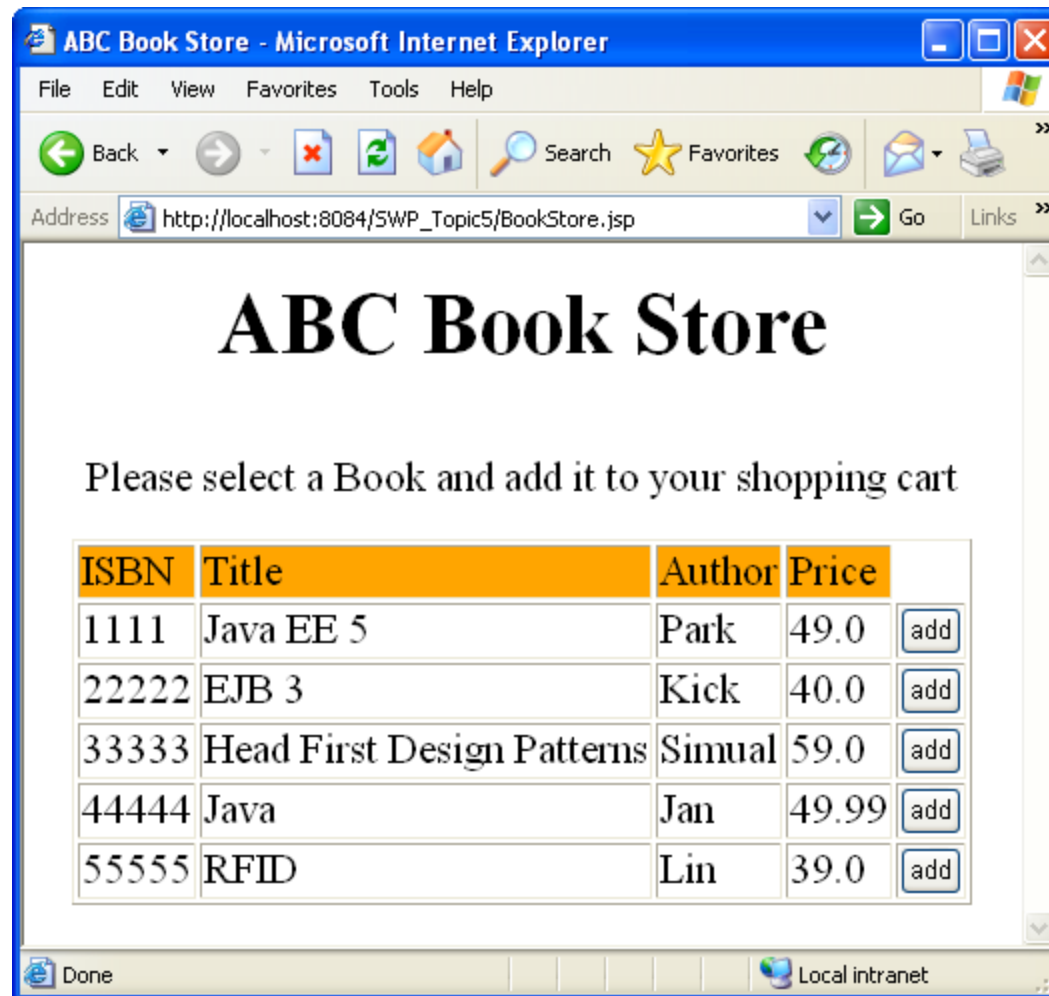
BookStore.jsp (cont.)

```
<table border="1" align="center">
  <tr bgcolor="orange">
    <td>ISBN</td><td>Title</td>
    <td>Author</td><td>Price</td></tr>
      <c:forEach var="row" items="${books.rows}">
      <tr>
          <td>${row.isbn}</td>
          <td>${row.title} /></td>
          <td>${row.author} /></td>
          <td>${row.price} /></td>
          <td><input type=submit value="add"></td>
      </tr>
    </c:forEach>
  </table>
  </form>
  </center>
  </body>
</html>
```

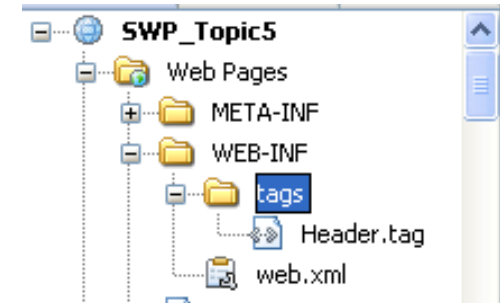# Result

Custom Tag Libraries

- The JSP 1.1 specifications introduced the ability to define new tags called custom tags
- Can be used in any number of JSP files
- A user can define how the tag, its attributes and its body are to be interpreted, and then group these tags into collections, called tag libraries.

Simple Tag Files

1. Take an included file (such as "Header.jsp") and rename it with a **.tag** extension

    ```
    <%-- Header.jsp --%>
    <img src="Duke_Cont.png"><br><br>
    Welcome to Java World!!
    ```
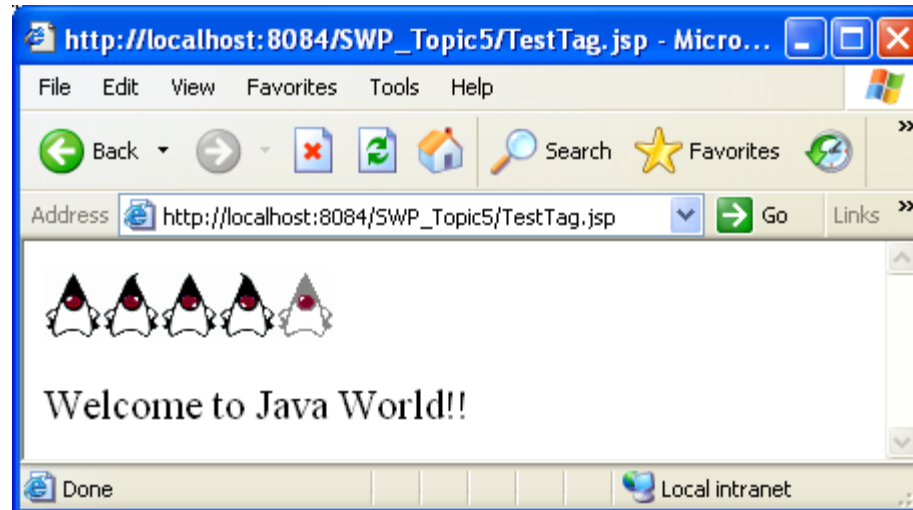
1. Put the tag file (such as "Header.tag") in a directory named "**tags**" inside the "WEB-INF" directory

1. Put a taglib directive (with a tagdir atttribute) in the JSP **(TestTag.jsp)**

    ```
    <%@taglib tagdir="/WEB-INF/tags" prefix="myTags" %>
    <myTags:Header/>
    ```

Result

Components that make up a Tag library

1. The Tag Library Descriptor (TLD) file
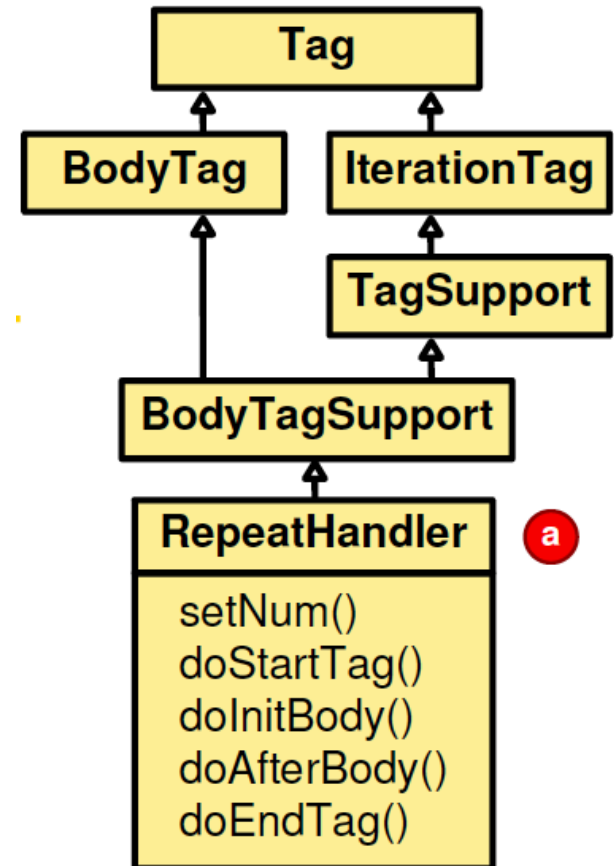2. The Tag Handler class
   - Classic
   - Simple
1. The JSP file

Repeat Tag Implemented as a Classic JSP 1.2 Tag Extension

Usage

```
<%@ taglib prefix="my"
        uri="/mytags" %>
<my:repeat num="3">
    tag body
</my:repeat>
```

Implementation

```
int doStartTag() {
    this.count = this.num;
    return Tag.EVAL_BODY_INCLUDE;
}

int doAfterBody() {
    this.count--;
    return (this.count > 0) ?
        Tag.EVAL_BODY_AGAIN :
        Tag.SKIP_BODY;
}
```
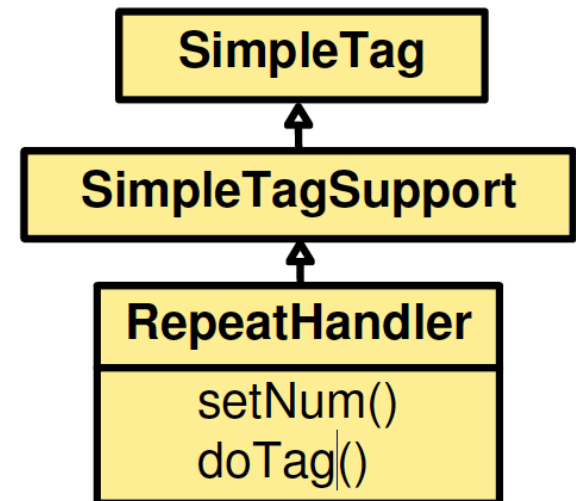
Tag

BodyTag — **IterationTag**

TagSupport

BodyTagSupport

RepeatHandler (a)
```
setNum()
doStartTag()
doInitBody()
doAfterBody()
doEndTag()
```

Repeat Tag Implemented as a Simple JSP 2.0 Tag Extension

Usage

```
<%@ taglib prefix="my"
        uri="/mytags" %>
<my:repeat num="3">
    tag body
</my:repeat>
```

Implementation

```
void doTag() {
    for( int i = 0; i < num; i++ ) {
        getJspBody().invoke( null );
    }
}
```

```
SimpleTag
    ▲
SimpleTagSupport
    ▲
RepeatHandler
    setNum()
    doTag()
```

44

Create a TLD for the tag

```xml
<?xml version="1.0" encoding="UTF-8"?>
<taglib version="2.1" xmlns="http://java.sun.com/xml/n
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instan
    xsi:schemaLocation="http://java.sun.com/xml/ns/jav
    <tlib-version>1.0</tlib-version>
    <short-name>MyTags</short-name>
    <uri>/WEB-INF/tlds/MyTags</uri>
    <tag>
        <name>test</name>
        <tag-class>handler.NameTagHandler</tag-class>
        <body-content>empty</body-content>
    </tag>
</taglib>
```

Making a Simple Tag Handler

```java
@Override
public void doTag() throws JspException, IOException {
    super.doTag();
    JspWriter out = getJspContext().getOut();
    try {
            out.print("This is a simple tag.");
    } catch (java.io.IOException ex) {
        throw new JspException("Error in NameTagHandler
    }
}
```

JSP File

- Import the tag library
  - Specify location of TLD file and define a tag prefix (namespace)

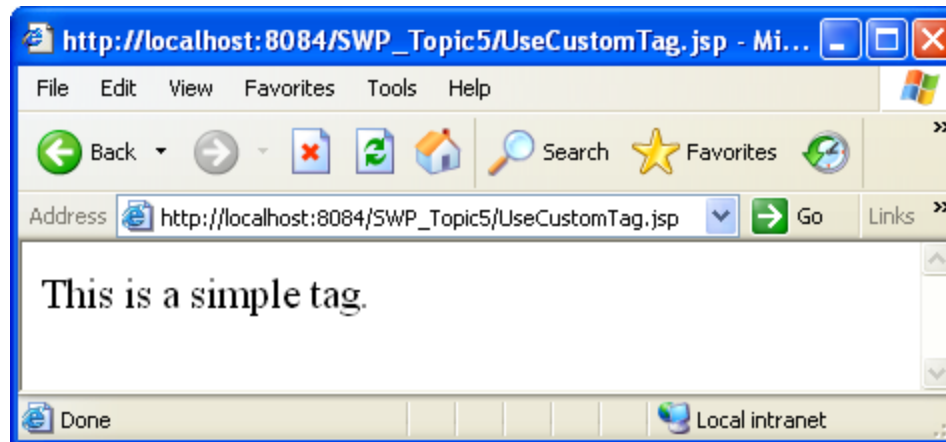   `<%@ taglib uri="myTaglib" prefix="myPrefix" %>`

- Use the tags
  - `<prefix:tagName />`
  - Prefix comes from taglib directive
  - Tag name comes from tag added into TLD file
  - Example `<myPrefix:myTag />`

Write a JSP that uses the tag

```
<%-- UseCustomTag.jsp --%>
<%@taglib uri="TestMyTag1" prefix="my" %>
<my:test1/>
```


This is a simple tag.

Thank you

thananum@gmail.com

www.facebook.com/imcinstitute

www.imcinstitute.com