CHAPTER 5
JSTL

1. INTRODUCTION

1.1. Introduction:

JSP Standard Tag Library (**JSTL**) is the standard tag library that provides tags to control the JSP page behavior, iteration and control statements, internationalization tags, and SQL tags.
JSTL is part of the Java EE API and included in most servlet containers. But to use JSTL in our JSP pages, we need to download the JSTL jars for your servlet container. Most of the times, you can find them in the example projects and you can use them. You need to include these libraries in the project **WEB-INF/lib** directory. These jars are container specific, for example in Tomcat, we need to include jstl.jar and standard.jar jar files in project build path. If they are not present in the container lib directory, you should include them into your application

1.2. JSTL Tags:

Based on the JSTL functions, the tags are categorized into five types.

1. Core Tags: Core tags provide support for iteration, conditional logic, catch exception, url, forward or redirect response etc. To use JSTL core tags, we should include it in the JSP page like below.

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

2. Formatting and Localization Tags: These tags are provided for formatting of Numbers, Dates and i18n support through locales and resource bundles. We can include these tags in JSP with below

Syntax:

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

3. SQL Tags: JSTL SQL Tags provide support for interaction with relational databases such as Oracle, MySql etc. Using SQL tags we can run database queries, we include it in JSP with below

Syntax:

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

4. XML Tags: XML tags are used to work with XML documents such as parsing XML, transforming XML data and XPath expressions evaluation.

Syntax to include XML tags in JSP page is:

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

5. JSTL Functions Tags: JSTL tags provide a number of functions that we can use to perform common operation, most of them are for String manipulation such as String Concatenation, Split String etc.

Syntax to include JSTL functions in JSP page is:

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

Note: All the JSTL standard tags URI starts with http://java.sun.com/jsp/jstl/ and we can use any prefix we want but it’s best practice to use the prefix defined above because everybody uses them, so it will not create any confusion.

1.3. JSTL Core Tags:

JSTL Core Tags are listed in the below table.

Tag	Description
<c:out>	To write something in JSP page, we can use EL also with this tag
<c:import>	Same as <jsp:include> or include directive
<c:redirect>	redirect request to another resource
<c:set>	To set the variable value in given scope.
<c:remove>	To remove the variable from given scope
<c:catch>	To catch the exception and wrap it into an object.
<c:if>	Simple conditional logic, used with EL and we can use it to process the exception from <c:catch>
<c:choose>	Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by <c:when> and <c:otherwise>
<c:when>	Subtag of <c:choose> that includes its body if its condition evalutes to ‘true’.
<c:otherwise>	Subtag of <c:choose> that includes its body if its condition evalutes to ‘false’.
<c:forEach>	for iteration over a collection
<c:forTokens>	for iteration over tokens separated by a delimiter.
<c:param>	used with <c:import> to pass parameters
<c:url>	to create a URL with optional query string parameters

Example: cout

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
```

```
   <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>

<!-- Simple use of cout usig value -->
<c:out value="Program for cout"></c:out><br>

<!-- With using escapeXml to use html tag -->
<c:out value="${'<b>Program for cout</b>'}" escapeXml="false"></c:out><br>

<!-- without using escapeXml which displays text as it is -->
<c:out value="${'<h1>program without escapeXml</h1>'}"></c:out>
</body>
</html>
```

Example: set variable

```
<%@ page language="java" contentType="text/html; charset=UTF-8"

   pageEncoding="UTF-8"%>
    <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<c:set var="jstl" value="Jsp Standard Tag Library" ></c:set>
<c:out value="${jstl}" ></c:out>
</body>
</html>
```

Example: remove variable

```
<%@ page language="java" contentType="text/html; charset=UTF-8"

   pageEncoding="UTF-8"%>
    <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<c:set var="data" scope="session" value="Jstl Program for remove attribute"/>
<c:set var="author" scope="session" value="John"/>
<c:remove var="author"/>
<a href="display.jsp">check attributes</a>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>
     <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<c:out value="${data}"/><br>
<c:out value="${author}" default="Attribute has no value"/>
</body>
</html>
```

Example: c:if

```
<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>
    <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<c:set var="age" value="26"/>
<c:if test="${age >= 18}">
 <c:out value="You are eligible for voting!"/>
</c:if>
<c:if test="${age < 18}">
 <c:out value="You are not eligible for voting!"/>
```

```
</c:if>
</body>
</html>
```

Example: forEach

```
<%@ page language="java" contentType="text/html; charset=UTF-8"

   pageEncoding="UTF-8"%>
      <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<c:forEach var="counter" begin="1" end="10">
 <c:out value="${counter}"/>
 </c:forEach>
</body>
</html>
```

1.4. JSTL Formatting tags:

Following is the list of Formatting JSTL Tags:

Tag	Description
<fmt:formatNumber>	To render numerical value with specific precision or format.
<fmt:parseNumber>	Parses the string representation of a number, currency, or percentage.
<fmt:formatDate>	Formats a date and/or time using the supplied styles and pattern
<fmt:parseDate>	Parses the string representation of a date and/or time
<fmt:bundle>	Loads a resource bundle to be used by its tag body.
<fmt:setLocale>	Stores the given locale in the locale configuration variable.
<fmt:setBundle>	Loads a resource bundle and stores it in the named scoped variable or the bundle configuration variable.
<fmt:timeZone>	Specifies the time zone for any time formatting or parsing actions nested in its body.
<fmt:setTimeZone>	Stores the given time zone in the time zone configuration variable
<fmt:message>	To display an internationalized message.
<fmt:requestEncoding>	Sets the request character encoding

Example: <fmt:formatNumber>

```
<%@ page language="java" contentType="text/html; charset=UTF-8"

   pageEncoding="UTF-8"%>
   <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<h3>Number Format:</h3>
<c:set var="balance" value="120000.2309" />
<p>Formatted Number (1): <fmt:formatNumber value="${balance}"
         type="currency"/></p>
<p>Formatted Number (2): <fmt:formatNumber type="number"
         maxIntegerDigits="3" value="${balance}" /></p>
<p>Formatted Number (3): <fmt:formatNumber type="number"
         maxFractionDigits="3" value="${balance}" /></p>
<p>Formatted Number (4): <fmt:formatNumber type="number"
         groupingUsed="false" value="${balance}" /></p>
<p>Formatted Number (5): <fmt:formatNumber type="percent"
         maxIntegerDigits="3" value="${balance}" /></p>
<p>Formatted Number (6): <fmt:formatNumber type="percent"
         minFractionDigits="10" value="${balance}" /></p>
<p>Formatted Number (7): <fmt:formatNumber type="percent"
         maxIntegerDigits="3" value="${balance}" /></p>
<p>Formatted Number (8): <fmt:formatNumber type="number"
         pattern="###.###E0" value="${balance}" /></p>
<p>Currency in USA :
<fmt:setLocale value="en_US"/>
<fmt:formatNumber value="${balance}" type="currency"/></p>

</body>
</html>
```

1.5. JSTL SQL tags:

Following is the list of SQL JSTL Tags:

Tag	Description
<sql:setDataSource>	Creates a simple DataSource suitable only for prototyping

<sql:query>	Executes the SQL query defined in its body or through the sql attribute.
<sql:update>	Executes the SQL update defined in its body or through the sql attribute.
<sql:param>	Sets a parameter in an SQL statement to the specified value.
<sql:dateParam>	Sets a parameter in an SQL statement to the specified java.util.Date value.
<sql:transaction >	Provides nested database action elements with a shared Connection, set up to execute all statements as one transaction.

1. JSTL <sql:setDataSource> tag:

JSTL setDataSource Tag provides capability to create a datasource directly from JSP and it can be stored in a variable to use later with the help of scope attribute.

JSP setDataSource Tag has following attributes.

1. driver Attribute: Specifies the driver to connect database. For each database there is a separate driver. In this example, we are connecting to MySQL database and hence driver is com.mysql.jdbc.Driver

2. url Attribute: Specifies the location of the database.

3. var Attribute: Specifies the variable which holds the dataSource once it is created.

4. user Attribute: Specifies the user you already created to access database.

5. password Attribute: Specifies the password for the user you already assigned to access database.

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
   <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
   <%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<sql:setDataSource var="localdata" driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/Sample" user="root" password="c0d3r123"/>
<sql:query dataSource="${localdata}" var="record">select * from student;</sql:query>
<table border="1" width="100%">
<tr>
<th>ID</th>
<th>Name</th>
<th>City</th>
```

```
<th>Marks</th>
</tr>
<c:forEach var="data" items="${record.rows}">
<tr>
<td><c:out value="${data.s_id}"/></td>
<td><c:out value="${data.s_name}"/></td>
<td><c:out value="${data.s_city}"/></td>
<td><c:out value="${data.s_marks}"/></td>
</tr>
</c:forEach>
</table>
</body>
</html>
```

As shown above setDataSource Tag set data source in a varaible localSource. Using this data source, you can do SQL operations directly from JSP.

1.6. JSTL XML tags:

Before you proceed with the examples, you would need to copy following two XML and XPath related libraries into your <Tomcat Installation Directory>\lib:

- **XercesImpl.jar:** Download it from http://www.apache.org/dist/xerces/j/
- **xalan.jar:** Download it from http://xml.apache.org/xalan-j/index.html

Following is the list of XML JSTL Tags:

Tag	Description
<x:out>	Like <%= ... >, but for XPath expressions.
<x:parse>	Use to parse XML data specified either via an attribute or in the tag body.
<x:set >	Sets a variable to the value of an XPath expression.
<x:if >	Evaluates a test XPath expression and if it is true, it processes its body. If the test condition is false, the body is ignored.
<x:forEach>	To loop over nodes in an XML document.
<x:choose>	Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by <when> and <otherwise>
<x:when >	Subtag of <choose> that includes its body if its expression evalutes to 'true'
<x:otherwise >	Subtag of <choose> that follows <when> tags and runs only if all of the prior conditions evaluated to 'false'
<x:transform >	Applies an XSL transformation on a XML document
<x:param >	Use along with the transform tag to set a parameter in the XSLT stylesheet

1.7. JSTL Functions:

Following is the list of JSTL Functions:

Function	Description
fn:contains()	Tests if an input string contains the specified substring.
fn:containsIgnoreCase()	Tests if an input string contains the specified substring in a case insensitive way.
fn:endsWith()	Tests if an input string ends with the specified suffix.
fn:escapeXml()	Escapes characters that could be interpreted as XML markup.
fn:indexOf()	Returns the index withing a string of the first occurrence of a specified substring.
fn:join()	Joins all elements of an array into a string.
fn:length()	Returns the number of items in a collection, or the number of characters in a string.
fn:replace()	Returns a string resulting from replacing in an input string all occurrences with a given string.
fn:split()	Splits a string into an array of substrings.
fn:startsWith()	Tests if an input string starts with the specified prefix.
fn:substring()	Returns a subset of a string.
fn:substringAfter()	Returns a subset of a string following a specific substring.
fn:substringBefore()	Returns a subset of a string before a specific substring.
fn:toLowerCase()	Converts all of the characters of a string to lower case.
fn:toUpperCase()	Converts all of the characters of a string to upper case.
fn:trim()	Removes white spaces from both ends of a string.

Example: indexof function

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
   <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
```

```
Index of "n" in the string "Indian city"
::${fn:indexOf("indiancity","n")}
</body>
</html>
```

Example: toUpperCase and to LowerCase function

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
    <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<form>
USERNAME:<input type="text" name="usr" size="25">
<p></p>
<input type="submit" value="Submit">
</form>
Your Username in Uppercase::
<c:out value="${fn:toUpperCase(param.usr)}" /> <br><br>
Your Username in LowerCase::
<c:out value="${fn:toLowerCase(param.usr)}" />
</body>
</html>
```

ASSIGNMENT

1) Write a program to send request to another resource.
2) Write a program for <C:forTokens>.
3) How many numbers are printed, when the following JSTL code fragment is executed?

```
<%@ taglib uri=”http://java.sun.com/jsp/jstl/core” prefix=”c” %>
<c:forEach var=”item” begin=”0″ end=”10″ step=”2″>
${item}
</c:forEach>
```

4) What gets printed when the following JSTL code fragment is executed?

```
<%@ taglib uri=”http://java.sun.com/jsp/jstl/core” prefix=”c” %>
<c:set var=”item” value=”2″/>
<c:if test=”${var==1}” var=”result” scope=”session”>
<c:out value=”${result}”/>
</c:if>
```

5) Write a program to see formats of dates.

6) Write a program to parse date and number.

7) Write a program to set locale.

8) Write a program to display all content from table.

9) Create a table for employee. Use 3 columns Name, City nad Salary. Write a program to perform following query:

```
select * from employee where salary>15000
```

10) Write a program to execute update query.

11) Write a program to retrive xml file data.

12) Write a program to perform following functions:

1. spliite()
2. startsWith()
3. trim()
4. length()
5. substringBefore()
6. split
7. substringAfter()