#### Eclipse support for creating an Testing Web Applications

The following components will be used: Apache Jakarta Tomcat servlet engine Derby

JavaServer Pages Standard Tag Library (JSTL) WTP

JavaServer Pages Standard Tag Library (JSTL)

Derby

***HTML***

**Scripted page**

**database**

Internet Explorer, Firefox, or internal browser

**Web browser**

**request**

Apache Jakarta Tomcat servlet engine

**Web server**

#### JSTL (JavaServerPages Standard Tag Library)

##### The JSP Standard Template Library (JSTL) is a very new component released by Sun for JSP programming.

* JSTL allows you to program your JSP pages using tags, rather than the scriptlet code that most JSP programmers are already accustomed to.
* JSTL was introduced was to allow JSP programmers to program using HTML-style tags rather than Java code.

JSP vs. JSTL

<html>

<head> <title>Count to 10 in JSP scriptlet</title> </head>

<body>

<% for(int i=1;i<=10;i++) { %>

<%= i %>

<br/>

<% } %>

</body>

</html>

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

<html>

<head> <title>Count to 10 Example (using JSTL)</title> </head>

<body>

<c:forEach var="i" begin="1" end="10" step="1">

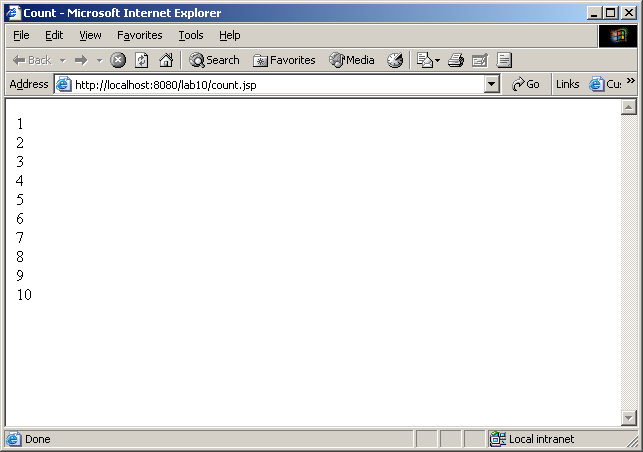
<c:out value="${i}" />

<br />

</c:forEach>

</body>

</html>



#### JSTL (JavaServerPages Standard Tag Library)

The JSP Standard Tag Library (JSTL) is a collection of custom tag libraries that implement general-purpose functionality common to Web applications, including iteration and conditionalization, data management formatting, manipulation of XML, and database access**.**

* + Core Tag Library – looping, expression evaluation, basic input/output
  + Formatting/Internationalization Tag Library – parsing data, such as dates.
  + Database Tag Library – tags that can be used to access SQL databases
  + XML Tag Library – tags can be used to access XML elements

#### JSTL (JavaServerPages Standard Tag Library)

Core Tag Library Example

**<%@ taglib [uri="h](http://java.sun.com/jsp/jstl/core)ttp**[**://java.s**](http://java.sun.com/jsp/jstl/core)**un**[**.com/jsp/jstl/core**](http://java.sun.com/jsp/jstl/core)**" prefix="c" %>**

**<html>**

**<head>**

**<title>Count to 10 Example (using JSTL)</title>**

**</head>**

**<body> <c:forEach var="i" begin="1" end="10" step="1">**

**<c:out value="${i}" />**

**<br />**

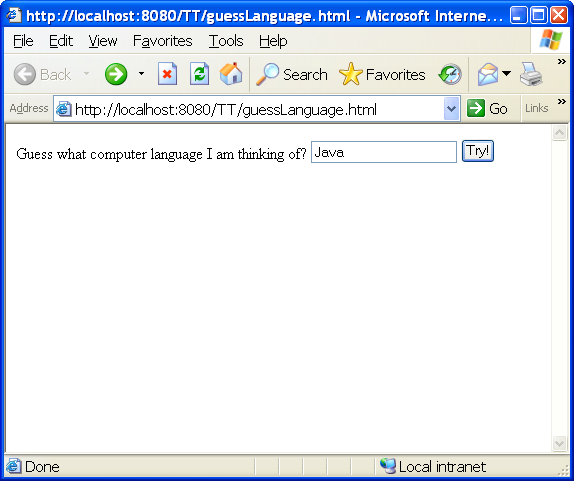
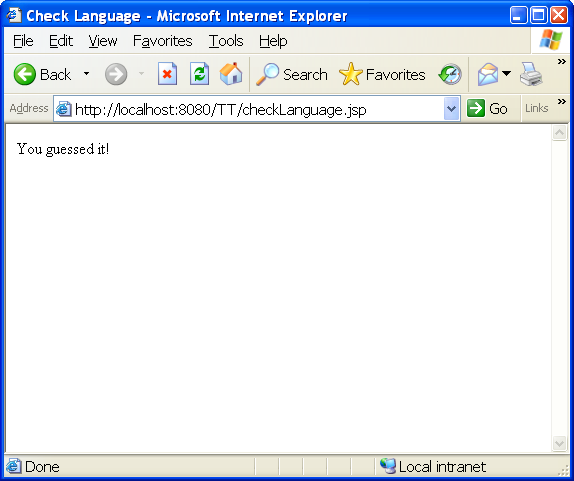
**</c:forEach>**

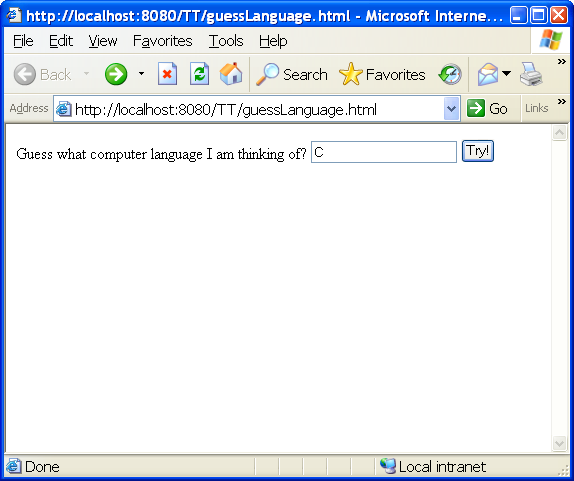
**</body>**

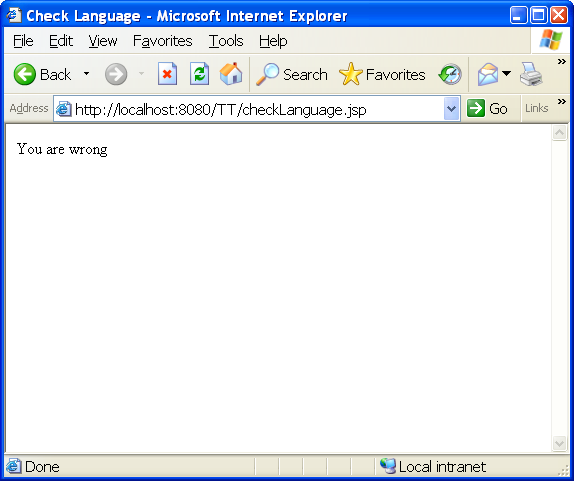
**</html>**

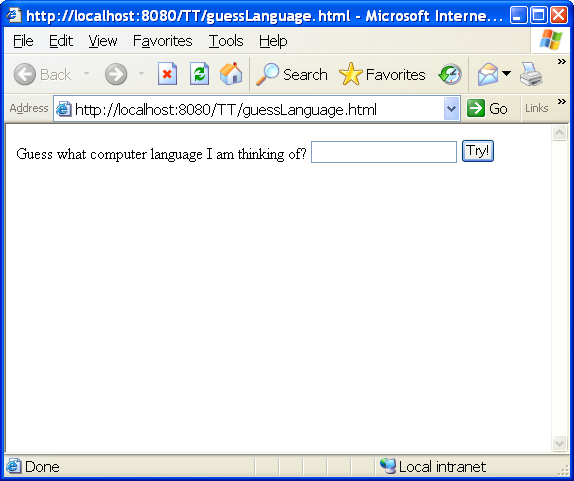
**Taglib for JSTL Components**

**Core <@tablib prefix=“c”** [**uri=“http://j**](http://java.sun.com/jsp/jstl/core)**a**[**va.sun.com/jsp/jstl/core**](http://java.sun.com/jsp/jstl/core) **%> XML <@tablib prefix=“c” uri=**[**“http://java.sun.com/jsp/jstl/xm**](http://java.sun.com/jsp/jstl/xml)**l %> Formatting <@tablib prefix=“c” uri=**[**“http://java.sun.com/jsp/jstl/fmt**](http://java.sun.com/jsp/jstl/fmt) **%> Database <@tablib prefix=“c” uri=**[**“http://j**](http://java.sun.com/jsp/jstl/sql)**a**[**va.sun.com/jsp/jstl/sql**](http://java.sun.com/jsp/jstl/sql) **%>**







<html>

<body>

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

Guess what computer language I am thinking of?

<input type="text" name="guess" />

<input type="submit" value="Try!" />

<br />

</form>

</body>

</html>

<%@ taglib [uri="htt](http://java.sun.com/jsp/jstl/core)p://[java.sun.com/jsp/jstl/cor](http://java.sun.com/jsp/jstl/core)e" prefix="c" %>

<html>

<head> <title>Check Language</title> </head>

<body>

<c:if test="${param.guess=='Java'}"> You guessed it!

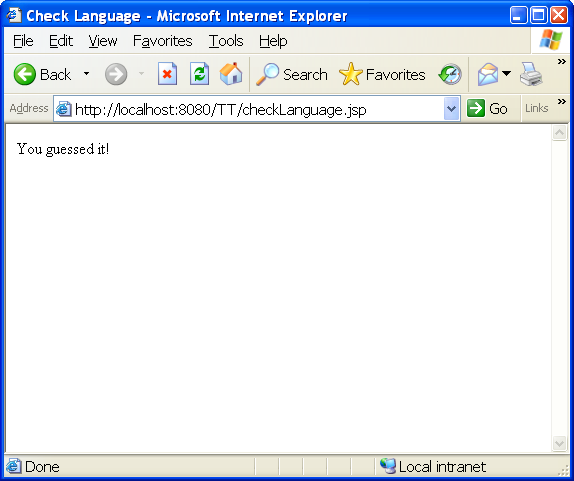
</c:if>

<c:if test="${param.guess!='Java'}"> You are wrong

</c:if>

</body>

</html>



<%@ taglib [uri="ht](http://java.sun.com/jsp/jstl/core)tp://[java.sun.com/jsp/jstl/core"](http://java.sun.com/jsp/jstl/core) prefix="c" %>

<html>

<head> <title>Check Language</title> </head>

<body>

<c:if test="${param.guess=='Java'}"> You guessed it!

</c:if>

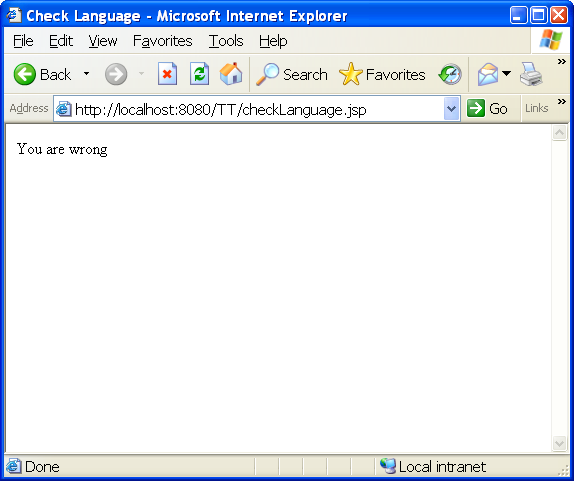
<c:if test="${param.guess!='Java'}">

You are wrong

</c:if>

</body>

</html>



<%@ taglib [uri="ht](http://java.sun.com/jsp/jstl/core)tp://[java.sun.com/jsp/jstl/core"](http://java.sun.com/jsp/jstl/core) prefix="c" %>

<html>

<head> <title>Check Language</title> </head>

<body>

<c:if test="${param.guess=='Java'}"> You guessed it!

</c:if>

<c:if test="${param.guess!='Java'}">

You are wrong

</c:if>

</body>

</html>

#### JSTL (JavaServerPages Standard Tag Library)

Database Tag Library

The Available <SQL>Actions

**There are six actions provided in this tag library:**

<sql:setDataSource> for exporting a variable that defines a data source

<sql:query> for querying to the database

<sql:update> for updating the database

<sql:transaction> for establishing a transaction context for doing queries and updates

<sql:param> for setting parameter markers ("?") used in SQL statements.

#### JSTL (JavaServerPages Standard Tag Library)

Database Tag Library

Setting the datasource:

<sql:setDataSource var=“datasource”

driver=“org.gjt.mm.mysql.driver“ url="jdbc:mysql://localhost/db” user="guest" password="guest"/>

<sql:query datasource="${datasource}" ... />

**JSTL (JavaServerPages Standard Tag Library)**

Database Tag Library

Issuing a query:

<sql:query

var="titles"

sql="SELECT \* FROM books WHERE title = 'JSTL' ORDER BY author"

</sql:query>

<sql:query var="titles“

SELECT \* FROM books WHERE title = 'JSTL' ORDER BY author

</sql:query>

#### JSTL (JavaServerPages Standard Tag Library)

Database Tag Library

**<%@taglib uri=**[**"http://java.sun.com/jsp/jstl/sql**](http://java.sun.com/jsp/jstl/sql)**" prefix="sql"%>**

**<sql:query var="cities">**

**SELECT CITY\_NAME, COUNTRY, AIRPORT FROM APP.CITIES ORDER BY \ CITY\_NAME, COUNTRY**

**</sql:query>**

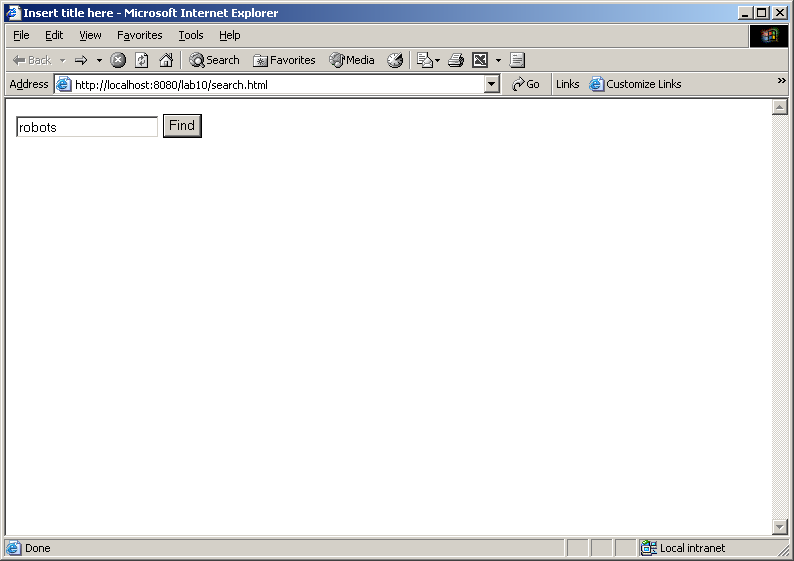
**<c:forEach var="city" items="${cities.rows}">**

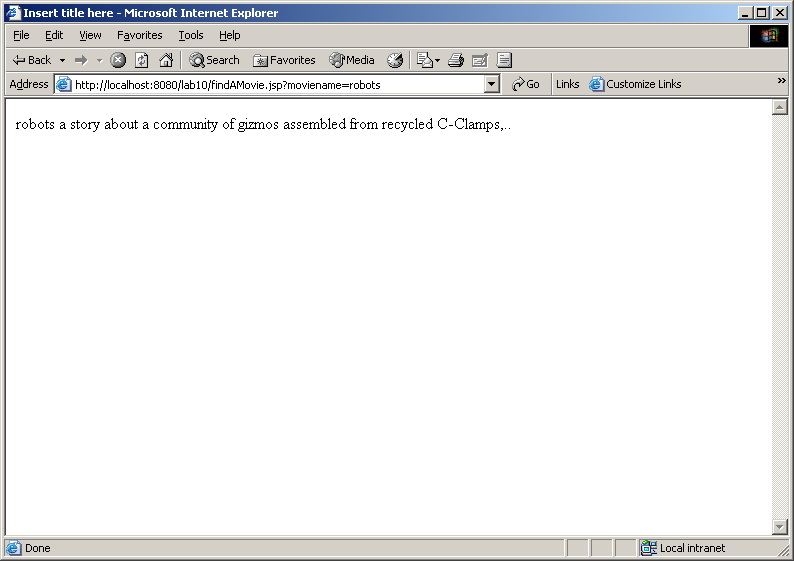
**<c:out value="${city.airport}” />**

**<c:out value=“${city.city\_name}” />**

**<br/>**

**</c:forEach>**





**Search.html**

**<html>**

**<head>**

**<title>Find a Movie</title>**

**</head>**

**<body>**

**<form method=get action="findAMovie.jsp">**

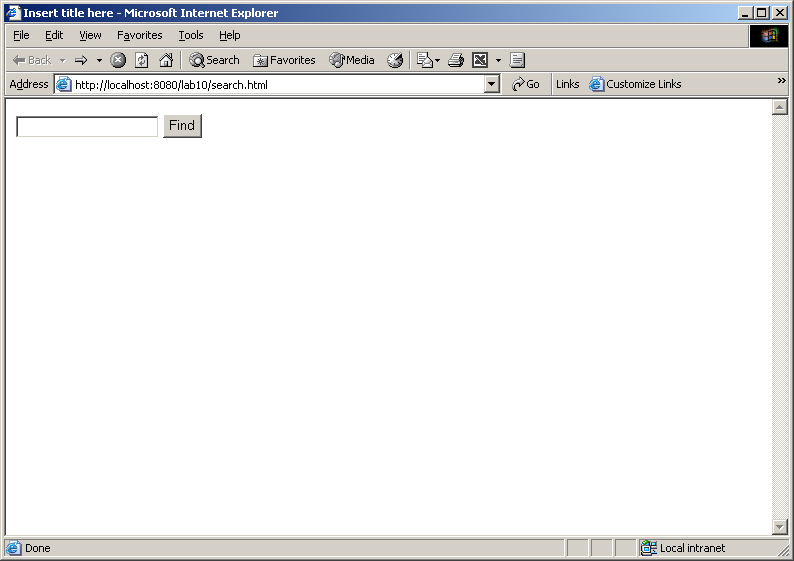
**<input type=text name=moviename>**

**<input type=submit>**

**</form>**

**</body>**

**</html>**



**<%@taglib** [**uri="htt**](http://java.sun.com/jsp/jstl/sql)**p://**[**ja**](http://java.sun.com/jsp/jstl/sql)**v**[**a.sun.com/jsp/jstl/sql"**](http://java.sun.com/jsp/jstl/sql) **prefix="sql"%>**

**<%@ taglib** [**uri="htt**](http://java.sun.com/jsp/jstl/core)**p://**[**ja**](http://java.sun.com/jsp/jstl/core)**v**[**a.sun.com/jsp/jstl/core**](http://java.sun.com/jsp/jstl/core)**" prefix="c" %>**

**<html>**

**<head>**

**<title>Movie Locator</title>**

**</head>**

**<body>**

**<sql:setDataSource driver="org.apache.derby.jdbc.ClientDriver"**

**url ="jdbc:derby://localhost:1527/famudb" user="bob" password="bob" />**

**<sql:query var="movies">**

**SELECT \* from movies where title='${param.moviename}'**

**</sql:query>**

**<c:forEach var="film" items="${movies.rows}">**

**<c:out value="${film.title}" />**

**<c:out value="${film.description}" />**

**<br/>**

**</c:forEach>**

**</body>**

**</html>**

findAMovie.jsp

JSTL

JSP Standard Tag Library

1

What is JSTL?

* **JSTL (JSP Standard Tag Libraries)** is a collection of JSP custom tags developed by Java Community Process, [www.jcp.org.](http://www.jcp.org/) The reference implementation is developed by the Jakarta project, jakarta.apache.org.
* **Full JSTL**

– Contains many common and useful JSP custom tags

##### JSTL allows you to program your JSP pages using tags, rather than the scriptlet code that most JSP programmers are already accustomed to. JSTL can do nearly everything that regular JSP scriptlet code can do.

2

**COUNT TO TEN EXAMPLE USING SCRIPTLET:**

* JSTL was introduced to allow JSP programmers to program using tags rather than Java code.
* To show why this is preferable, a quick example is in order. We will examine a very simple JSP page that counts to ten.
* We will examine this page both as regular scriptlet-based JSP, and then as JSTL. When the count to ten example is programmed using scriptlet based JSP, the JSP page appears as follows.`

3

<html>

<head>

<title>Count to 10 in JSP scriptlet</title>

</head>

<body>

<% for(int i=1;i<=10;i++)

{%>

<%=i%><br/>

<%

}

%>

</body>

</html>

## COUNT TO TEN EXAMPLE USING JSTL:

The following code shows how the count to ten example would be written using JSTL. As you can see, this code listing is much more constant, as only tags are used. HTML and JSTL tags are mixed to produce the example.

<%@ taglib uri[="http://ja](http://java.sun.com/jstl/core)v[a.sun.com/jstl/core"](http://java.sun.com/jstl/core) prefix="c" %>

<html>

<head>

<title>Count to 10 Example (using JSTL)</title>

</head>

<body>

<c:forEach var="i" begin="1" end="10" step="1">

<c:out value="${i}" />

<br />

</c:forEach>

</body>

</html>

4

# COUNT TO TEN EXAMPLE USING JSTL:

### When you examine the preceding source code, you can see that the JSP page consists entirely of tags.

* + The code makes use of HTML tags such as <head> and

<br>. The use of tags is not confined just to HTML tags. We can modify that code to print numbers from 1 to n. The code is given in the next slide.

* + This code also makes use of JSTL tags such as

<c:forEach> and <c:out>. In this article you will be introduced to some of the basics of JSTL.

5

**PRINT 1 TO N EXAMPLE USING JSTL:**

The following code shows how the count to n example would be written using JSTL. As you can see, this code listing is much more constant, as only tags are used. HTML and JSTL tags are mixed to produce the example. You can just use http://<jsp file>?number=10 to see the result.

<%@ taglib prefix="c" uri[="http://ja](http://java.sun.com/jstl/core)v[a.sun.com/jstl/core"](http://java.sun.com/jstl/core) %>

<HTML>

<HEAD>

</HEAD>

<BODY>

<c:set var="n" value="${param.number}" /> Hello ...Printing numbers till N=

<c:out value="${n}"/>

<br/>

<br/>

<c:forEach var="i" begin="1" end="${n}" step="1">

<c:out value="${i}"/>

</c:forEach>

</BODY>

6

# THE JSTL TAG LIBRARIES

### The JavaServer Pages Standard Tag Library (JSTL) is a collection of useful JSP tags which encapsulates core functionality common to many JSP applications. JSTL is often spoken of as a single-tag library.

* + JSTL has support for common, structural tasks such as iteration and conditionals, tags for manipulating XML documents, internationalization tags, and SQL tags. It also provides a framework for integrating existing custom tags with JSTL tags.
  + The JSTL tags can be classified, according to their functions, into following JSTL tag library groups that can be used when creating a JSP page:

7

# THE JSTL TAG LIBRARIES

###### The JSTL tags can be classified, according to their functions, into following JSTL tag library groups that can be used when creating a JSP page:

* **Core Tags**
* **Formatting tags**
* **SQL tags**
* **XML tags**
* **JSTL Functions**

8

# THE JSTL TAG LIBRARIES

* + **Core Tag Library**—Contains tags that are essential to nearly any Web application. Examples of core tag libraries include looping, expression evaluation, and basic input and output.
  + **Formatting/Internationalization Tag Library**—Contains tags that are used to parse data. Some of these tags will parse data, such as dates, differently based on the current locale.
  + **Database Tag Library**—Contains tags that can be used to access SQL databases. These tags are normally used only to create prototype programs. This is because most programs will not handle database access directly from JSP pages. Database access should be embedded in EJBs that are accessed by the JSP pages.
  + **XML Tag Library**—Contains tags that can be used to access XML elements. Because XML is used in many Web applications, XML processing is an important feature of JSTL.
  + **JSTL Functions --**JSTL includes a number of standard functions, most of which are common string manipulation functions.

9

# Core Tags:

The core group of tags are the most frequently used JSTL tags. Following is

the syntax to include JSTL Core library in your JSP:

**<%@ taglib prefix="c"** [**uri="h**](http://java.sun.com/jsp/jstl/core)**t**[**tp://ja**](http://java.sun.com/jsp/jstl/core)**va.**[**sun.com/jsp/jstl/cor**](http://java.sun.com/jsp/jstl/core)**e" %>**

There are following Core JSTL Tags:

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<c:out >](http://www.tutorialspoint.com/jsp/jstl_core_out_tag.htm) | Like <%= ... >, but for expressions. |
| [<c:set >](http://www.tutorialspoint.com/jsp/jstl_core_set_tag.htm) | Sets the result of an expression evaluation in a 'scope' |
| [<c:remove >](http://www.tutorialspoint.com/jsp/jstl_core_remove_tag.htm) | Removes a scoped variable (from a particular scope, if specified). |
| [<c:catch>](http://www.tutorialspoint.com/jsp/jstl_core_catch_tag.htm) | Catches any Throwable that occurs in its body and optionally exposes it. |
| [<c:if>](http://www.tutorialspoint.com/jsp/jstl_core_if_tag.htm) | Simple conditional tag which evalutes its body if the supplied condition is true. |
| [<c:choose>](http://www.tutorialspoint.com/jsp/jstl_core_choose_tag.htm) | Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by <when> and <otherwise> |
| [<c:when>](http://www.tutorialspoint.com/jsp/jstl_core_choose_tag.htm) | Subtag of <choose> that includes its body if its condition evalutes to 'true'. |
| [<c:otherwise >](http://www.tutorialspoint.com/jsp/jstl_core_choose_tag.htm) | Subtag of <choose> that follows <when> tags and runs only if all of the prior  conditions evaluated to 'false'. |
| [<c:import>](http://www.tutorialspoint.com/jsp/jstl_core_import_tag.htm) | Retrieves an absolute or relative URL and exposes its contents to either the page, a String in 'var', or a Reader in 'varReader'. |
| [<c:forEach >](http://www.tutorialspoint.com/jsp/jstl_core_foreach_tag.htm) | The basic iteration tag, accepting many different collection types and supporting subsetting and other functionality . |
| [<c:forTokens>](http://www.tutorialspoint.com/jsp/jstl_core_foreach_tag.htm) | Iterates over tokens, separated by the supplied delimeters. |
| [<c:param>](http://www.tutorialspoint.com/jsp/jstl_core_param_tag.htm) | Adds a parameter to a containing 'import' tag's URL. |
| [<c:redirect >](http://www.tutorialspoint.com/jsp/jstl_core_redirect_tag.htm) | Redirects to a new URL. |
| [<c:url>](http://www.tutorialspoint.com/jsp/jstl_core_url_tag.htm) | Creates a URL with optional query parameters |

10

# Formatting tags:

The JSTL formatting tags are used to format and display text, the date, the

time, and numbers for internationalized Web sites. Following is the syntax to include Formatting library in your JSP:

**<%@ taglib prefix="fmt"** [**uri="h**](http://java.sun.com/jsp/jstl/fmt)**t**[**tp://ja**](http://java.sun.com/jsp/jstl/fmt)**va.**[**sun.co**](http://java.sun.com/jsp/jstl/fmt)**m**[**/jsp/jstl/fmt"**](http://java.sun.com/jsp/jstl/fmt) **%>**

Following is the list of Formatting JSTL Tags:

11

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<fmt:formatNumber>](http://www.tutorialspoint.com/jsp/jstl_format_formatnumber_tag.htm) | To render numerical value with specific precision or format. |
| [<fmt:parseNumber>](http://www.tutorialspoint.com/jsp/jstl_format_parsenumber_tag.htm) | Parses the string representation of a number, currency, or  percentage. |
| [<fmt:formatDate>](http://www.tutorialspoint.com/jsp/jstl_format_formatdate_tag.htm) | Formats a date and/or time using the supplied styles and pattern |
| [<fmt:parseDate>](http://www.tutorialspoint.com/jsp/jstl_format_parsedate_tag.htm) | Parses the string representation of a date and/or time |
| [<fmt:bundle>](http://www.tutorialspoint.com/jsp/jstl_format_bundle_tag.htm) | Loads a resource bundle to be used by its tag body. |
| [<fmt:setLocale>](http://www.tutorialspoint.com/jsp/jstl_format_setlocale_tag.htm) | Stores the given locale in the locale configuration variable. |
| [<fmt:setBundle>](http://www.tutorialspoint.com/jsp/jstl_format_setbundle_tag.htm) | Loads a resource bundle and stores it in the named scoped variable or  the bundle configuration variable. |
| [<fmt:timeZone>](http://www.tutorialspoint.com/jsp/jstl_format_timezone_tag.htm) | Specifies the time zone for any time formatting or parsing actions nested in its body. |
| [<fmt:setTimeZone>](http://www.tutorialspoint.com/jsp/jstl_format_settimezone_tag.htm) | Stores the given time zone in the time zone configuration variable |
| [<fmt:message>](http://www.tutorialspoint.com/jsp/jstl_format_message_tag.htm) | To display an internationalized message. |
| [<fmt:requestEncoding>](http://www.tutorialspoint.com/jsp/jstl_format_requestencoding_tag.htm) | Sets the request character encoding |

# SQL tags:

* + The JSTL SQL tag library provides tags for interacting with relational databases

(RDBMSs) such as Oracle, mySQL, or Microsoft SQL Server.

* + Following is the syntax to include JSTL SQL library in your JSP:

**<%@ taglib prefix="sql"** [**uri="h**](http://java.sun.com/jsp/jstl/sql)**t**[**tp://ja**](http://java.sun.com/jsp/jstl/sql)**va.**[**sun.com/jsp/jstl/**](http://java.sun.com/jsp/jstl/sql)**s**[**ql"**](http://java.sun.com/jsp/jstl/sql) **%>**

Following is the list of SQL JSTL Tags:

12

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<sql:setDataSource>](http://www.tutorialspoint.com/jsp/jstl_sql_setdatasource_tag.htm) | Creates a simple DataSource suitable only for prototyping |
| [<sql:query>](http://www.tutorialspoint.com/jsp/jstl_sql_query_tag.htm) | Executes the SQL query defined in its body or  through the sql attribute. |
| [<sql:update>](http://www.tutorialspoint.com/jsp/jstl_sql_update_tag.htm) | Executes the SQL update defined in its body or  through the sql attribute. |
| [<sql:param>](http://www.tutorialspoint.com/jsp/jstl_sql_param_tag.htm) | Sets a parameter in an SQL statement to the  specified value. |
| [<sql:dateParam>](http://www.tutorialspoint.com/jsp/jstl_sql_dateparam_tag.htm) | Sets a parameter in an SQL statement to the specified java.util.Date value. |
| [<sql:transaction >](http://www.tutorialspoint.com/jsp/jstl_sql_transaction_tag.htm) | Provides nested database action elements with a shared Connection, set up to execute all statements as one transaction. |

# XML TAGS:

###### The JSTL XML tags provide a JSP-centric way of creating and manipulating XML documents. Following is the syntax to include JSTL XML library in your JSP:

**<%@ taglib prefix="x"** [**uri="h**](http://java.sun.com/jsp/jstl/xml)**t**[**tp://**](http://java.sun.com/jsp/jstl/xml)**jav**[**a.sun.com/jsp**](http://java.sun.com/jsp/jstl/xml)**/[jstl/xml"](http://java.sun.com/jsp/jstl/xml) %>**

###### The JSTL XML tag library has custom tags for interacting with XML data. This includes parsing XML, transforming XML data, and flow control based on XPath expressions.

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>

13

Following is the list of XML JSTL Tags:

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<x:out>](http://www.tutorialspoint.com/jsp/jstl_xml_out_tag.htm) | Like <%= ... >, but for XPath expressions. |
| [<x:parse>](http://www.tutorialspoint.com/jsp/jstl_xml_parse_tag.htm) | Use to parse XML data specified either via an attribute or in the tag body. |
| [<x:set >](http://www.tutorialspoint.com/jsp/jstl_xml_set_tag.htm) | Sets a variable to the value of an XPath expression. |
| [<x:if >](http://www.tutorialspoint.com/jsp/jstl_xml_if_tag.htm) | 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>](http://www.tutorialspoint.com/jsp/jstl_xml_foreach_tag.htm) | To loop over nodes in an XML document. |
| [<x:choose>](http://www.tutorialspoint.com/jsp/jstl_xml_choose_tag.htm) | Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by  <when> and <otherwise> |
| [<x:when >](http://www.tutorialspoint.com/jsp/jstl_xml_choose_tag.htm) | Subtag of <choose> that includes its body if its expression evalutes to 'true' |
| [<x:otherwise >](http://www.tutorialspoint.com/jsp/jstl_xml_choose_tag.htm) | Subtag of <choose> that follows <when> tags and runs only if all of the prior conditions evaluated to 'false' |
| [<x:transform >](http://www.tutorialspoint.com/jsp/jstl_xml_transform_tag.htm) | Applies an XSL transformation on a XML document |
| [<x:param >](http://www.tutorialspoint.com/jsp/jstl_xml_param_tag.htm) | Use along with the transform tag to set a parameter in the XSLT stylesheet |

14

# JSTL Functions:

JSTL includes a number of standard functions, most of which are common string manipulation functions. Following is the syntax to include JSTL Functions library in your JSP:

**<%@ taglib prefix="fn" uri="**[**http://java.sun.com/jsp/jstl/functions"**](http://java.sun.com/jsp/jstl/functions) **%>**

Following is the list of JSTL Functions:

|  |  |
| --- | --- |
| **Function** | **Description** |
| [fn:contains()](http://www.tutorialspoint.com/jsp/jstl_function_contains.htm) | Tests if an input string contains the specified substring. |
| [fn:containsIgnoreCase(](http://www.tutorialspoint.com/jsp/jstl_function_containsignorecase.htm)  [)](http://www.tutorialspoint.com/jsp/jstl_function_containsignorecase.htm) | Tests if an input string contains the specified substring in a case insensitive way. |
| [fn:endsWith()](http://www.tutorialspoint.com/jsp/jstl_function_endswith.htm) | Tests if an input string ends with the specified suffix. |
| [fn:escapeXml()](http://www.tutorialspoint.com/jsp/jstl_function_escapexml.htm) | Escapes characters that could be interpreted as XML markup. |
| [fn:indexOf()](http://www.tutorialspoint.com/jsp/jstl_function_indexof.htm) | Returns the index within a string of the first occurrence  of a specified substring. |
| [fn:join()](http://www.tutorialspoint.com/jsp/jstl_function_join.htm) | Joins all elements of an array into a string. |
| [fn:length()](http://www.tutorialspoint.com/jsp/jstl_function_length.htm) | Returns the number of items in a collection, or the number of characters in a string. |
| [fn:replace()](http://www.tutorialspoint.com/jsp/jstl_function_replace.htm) | Returns a string resulting from replacing in an input string all occurrences with a given string. |
| [fn:split()](http://www.tutorialspoint.com/jsp/jstl_function_split.htm) | Splits a string into an array of substrings. |

Made By:

1. Gowtham Ram
2. Reddy Srinivas