

# JSP - STANDARD TAG LIBRARY (JSTL) TUTORIAL

[http://www.tutorialspoint.com/jsp/jsp\\_standard\\_tag\\_library.htm](http://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm)

Copyright © tutorialspoint.com

The JavaServer Pages Standard Tag Library (JSTL) is a collection of useful JSP tags which encapsulates core functionality common to many JSP applications.

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:

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

## Install JSTL Library:

If you are using Apache Tomcat container then follow the following two simple steps:

- Download the binary distribution from [Apache Standard Tag lib](#) and unpack the compressed file.
- To use the Standard Tag lib from its Jakarta Tag libs distribution, simply copy the JAR files in the distribution's 'lib' directory to your application's webapps\ROOT\WEB-INF\lib directory.

To use any of the libraries, you must include a <taglib> directive at the top of each JSP that uses the library.

## 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="http://java.sun.com/jsp/jstl/core" %>
```

There are following Core JSTL Tags:

Tag	Description
<a href="#">&lt;c:out&gt;</a>	Like <%= ... >, but for expressions.
<a href="#">&lt;c:set&gt;</a>	Sets the result of an expression evaluation in a 'scope'
<a href="#">&lt;c:remove&gt;</a>	Removes a scoped variable (from a particular scope, if specified).
<a href="#">&lt;c:catch&gt;</a>	Catches any Throwable that occurs in its body and optionally exposes it.
<a href="#">&lt;c:if&gt;</a>	Simple conditional tag which evaluates its body if the supplied condition is true.
<a href="#">&lt;c:choose&gt;</a>	Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by <when> and <otherwise>
<a href="#">&lt;c:when&gt;</a>	Subtag of <choose> that includes its body if its condition evaluates to 'true'.

<a href="#"><u>&lt;c:otherwise&gt;</u></a>	Subtag of <choose> that follows <when> tags and runs only if all of the prior conditions evaluated to 'false'.
<a href="#"><u>&lt;c:import&gt;</u></a>	Retrieves an absolute or relative URL and exposes its contents to either the page, a String in 'var', or a Reader in 'varReader'.
<a href="#"><u>&lt;c:forEach&gt;</u></a>	The basic iteration tag, accepting many different collection types and supporting subsetting and other functionality.
<a href="#"><u>&lt;c:forTokens&gt;</u></a>	Iterates over tokens, separated by the supplied delimiters.
<a href="#"><u>&lt;c:param&gt;</u></a>	Adds a parameter to a containing 'import' tag's URL.
<a href="#"><u>&lt;c:redirect&gt;</u></a>	Redirects to a new URL.
<a href="#"><u>&lt;c:url&gt;</u></a>	Creates a URL with optional query parameters

## 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="http://java.sun.com/jsp/jstl/fmt" %>
```

Following is the list of Formatting JSTL Tags:

Tag	Description
<a href="#"><u>&lt;fmt:formatNumber&gt;</u></a>	To render numerical value with specific precision or format.
<a href="#"><u>&lt;fmt:parseNumber&gt;</u></a>	Parses the string representation of a number, currency, or percentage.
<a href="#"><u>&lt;fmt:formatDate&gt;</u></a>	Formats a date and/or time using the supplied styles and pattern
<a href="#"><u>&lt;fmt:parseDate&gt;</u></a>	Parses the string representation of a date and/or time
<a href="#"><u>&lt;fmt:bundle&gt;</u></a>	Loads a resource bundle to be used by its tag body.
<a href="#"><u>&lt;fmt:setLocale&gt;</u></a>	Stores the given locale in the locale configuration variable.
<a href="#"><u>&lt;fmt:setBundle&gt;</u></a>	Loads a resource bundle and stores it in the named scoped variable or the bundle configuration variable.
<a href="#"><u>&lt;fmt:timeZone&gt;</u></a>	Specifies the time zone for any time formatting or parsing actions nested in its body.
<a href="#"><u>&lt;fmt:setTimeZone&gt;</u></a>	Stores the given time zone in the time zone configuration variable
<a href="#"><u>&lt;fmt:message&gt;</u></a>	To display an internationalized message.
<a href="#"><u>&lt;fmt:requestEncoding&gt;</u></a>	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="http://java.sun.com/jsp/jstl/sql" %>
```

Following is the list of SQL JSTL Tags:

Tag	Description
<a href="#"><u>&lt;sql:setDataSource&gt;</u></a>	Creates a simple DataSource suitable only for prototyping
<a href="#"><u>&lt;sql:query&gt;</u></a>	Executes the SQL query defined in its body or through the sql attribute.
<a href="#"><u>&lt;sql:update&gt;</u></a>	Executes the SQL update defined in its body or through the sql attribute.
<a href="#"><u>&lt;sql:param&gt;</u></a>	Sets a parameter in an SQL statement to the specified value.
<a href="#"><u>&lt;sql:dateParam&gt;</u></a>	Sets a parameter in an SQL statement to the specified java.util.Date value.
<a href="#"><u>&lt;sql:transaction&gt;</u></a>	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.

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.

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

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
<a href="#"><u>&lt;x:out&gt;</u></a>	Like <%= ... >, but for XPath expressions.
<a href="#"><u>&lt;x:parse&gt;</u></a>	Use to parse XML data specified either via an attribute or in the tag body.
<a href="#"><u>&lt;x:set&gt;</u></a>	Sets a variable to the value of an XPath expression.
<a href="#"><u>&lt;x:if&gt;</u></a>	Evaluates a test XPath expression and if it is true, it processes its body. If the test condition is false, the body is ignored.
<a href="#"><u>&lt;x:forEach&gt;</u></a>	To loop over nodes in an XML document.
<a href="#"><u>&lt;x:choose&gt;</u></a>	Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by <when> and <otherwise>
<a href="#"><u>&lt;x:when&gt;</u></a>	Subtag of <choose> that includes its body if its expression evaluates to 'true'
<a href="#"><u>&lt;x:otherwise&gt;</u></a>	Subtag of <choose> that follows <when> tags and runs only if all of the prior conditions evaluated to 'false'

<a href="#"><u>&lt;x:transform&gt;</u></a>	Applies an XSL transformation on a XML document
<a href="#"><u>&lt;x:param&gt;</u></a>	Use along with the transform tag to set a parameter in the XSLT stylesheet

## 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" %>
```

Following is the list of JSTL Functions:

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