JavaServer™ Pages Standard Tag Library (JSTL) syntax reference

OIO **Orientation in Objects**

) version 1.0)

JSTL Expression Language Definition

Implicit Object	Contains
pageScope	variables which have the scope page.
requestScope	variables which have the scope request.
sessionScope	variables which have the scope session.
applicationScope	variables which have the scope of the application.
pageContext	JSP PageContext object
cookie	Cookie values
param	Request parameters as strings.
header	HTTP request headers as strings.
headerValues	HTTP request headers as a collection of strings
initParam	Context initialization parameters

JSTL Operators

Comparison operators:	== / eq, != / ne, < / lt, > / gt, <= / le, >= / ge
Logical operators:	&& / and, || / or, ! / not
Arithmetic operators:	+, -, *, /, % / mod
Property access:	**Dot operator (.)** can be used to retrieve a named property. To access numbered or named properties use the **bracket operator ([])**

Example: ${user.name} , ${row[0]} or ${order.items.price}
Use the **empty operator** to determine whether a collection or string is empty or null *Example:* <c:if test="${empty pageScope.users}">

Legend

All tags are case sensitive. A pair of single quotes is equivalent to a pair of double quotes. Spaces are not allowed between an equals sign and an attribute value.

plain text = required   *italics* = user- defined   value = default value   attribute = using a body, the content is associated with this attribute
{ } = required choice | = or   [ ] = optional   ... = list of items   ex.= exported   sc. = scoped

Core Tag Library - General Purpose Actions

Action	Description	Syntax	*name*	*dyn**	*type*	*description*
<c:out>	Evaluates an expression and outputs the result to the current JspWriter object.	<c:out value="*value*" [escapeXml="{true|false}"] [default ="*defaultValue*"] />	value	true	Object	Expression to be evaluated
			escapeXml	true	boolean	Determines whether <,>,&,','' in the result should be converted to their corresponding character entity codes.
			default	true	Object	Default value if the resulting value is null.
<c:set>	Sets the value of a scoped variable or a property of a target object.	<c:set value="*value*" var="*varName*" [scope="{page|request|session|application}"]/> or <c:set value="*value*" target="*target*" property="*propertyName*"/>	value	true	Object	Expression to be evaluated
			var	false	String	Scoped variable to hold the value specified in the action.
			scope	false	String	Scope for *var*.
			target	true	Object	JavaBeans object with setter property, or to a java.util.Map object.
			property	true	String	Name of the property to be set in the target object.
<c:remove>	Removes a scoped variable.	<c:remove var="*varName*" [scope="{page|request|session|application}"]/>	var	false	String	Name of the scoped variable to be removed
			scope	false	String	Scope for *var*.
<c:catch>	Catches a java.lang.Throwable thrown by any of its nested actions.	<c:catch [var="*varName*"]> *nested actions* </c:catch>	var	false	String	Variable to expose information about error

Core Tag Library - Conditional Actions

Action	Description	Syntax	*name*	*dyn**	*type*	*description*
<c:if>	Evaluates its body content if the expression is true.	<c:if test="*testCondition*" [var="*varName*"] [scope="{page|request|session|application}"]> *body content* </c:if>	test	true	boolean	Test condition that determines whether or not the body content should be processed
			var	false	String	Name of the exported scoped variable for the resulting test condition value.
			scope	false	String	Scope for *var*.
						Constraints: If scope is specified, var must also be specified.
<c:choose> **<c:when>** **<c:otherwise>**	Provides the context for mutually exclusive conditional execution.	<c:choose> <c:when test="*testCondition1*">body content </c:when> <c:when test="*testCondition2*">body content </c:when> ... <c:otherwise>conditional block </c:otherwise> </c:choose>	test	true	boolean	Test condition - determines whether or not the body content should be processed
						Constraints: The body of <c:choose> can only contain 1 or more <c:when> actions (must all appear before <c:otherwise>) and 0 or 1 <c:otherwise> action (must be the last action)

Core Tag Library - Iterator Actions

Action	Description	Syntax	*name*	*dyn**	*type*	*description*
<c:forEach>	Repeats its nested body content over a collection of objects, or repeats it a fixed number of times.	<c:forEach [var="*varName*"] items="*collection*" [varStatus="*varStatusName*"] [begin="*begin*"][end="*end*"][step="*step*"]> *body content* </c:forEach>	var	false	String	Name of the exported scoped variable for the current item. Its type depends on the object of the underlying collection.
			items	true	(see descr.)	Collection of items to iterate over *Types* can contain all supported types
			varStatus	false	String	Name of the exported scoped variable for the status of the iteration.
			begin / end	true	int	Iteration begins at the specified index (items) or at the specified value.
			step	true	int	Iteration will only process every step items.
						Constraints: If items is not specified, begin and end must be specified.
<c:forTokens>	Iterates over tokens, separated by the supplied delimiters.	<c:forTokens items="*stringOfTokens*" delims="*delimiters*" [var="*varName*"][varStatus="*varStatusName*"] [begin="*begin*"] [end="*end*"] [step="*step*"]> *body content* </c:forTokens>	var	false	String	Name of the exported scoped variable for the current item of the iteration
			items	true	String	String of tokens to iterate over.
			delims	true	String	The set of delimiters (the characters that separate the tokens in the string).
			varStatus	false	String	Name of the exported scoped variable for the status of the iteration.
			begin / end	true	int	Iteration begins / ends at the token located at the specified index.
			step	true	int	Iteration will only process every step tokens.

Core Tag Library - URL Related Actions

Action	Description	Syntax	*name*	*dyn**	*type*	*description*
<c:redirect>	Sends an HTTP redirect to the client.	<c:redirect url="*value*" [context="*context*"]/> *optional* <c:param> *subtags within the body*	url	true	String	The URL of the resource to redirect to.
			context	true	String	Context name when specifying a relative URL that belongs to a foreign context.
<c:param>	Adds request parameters to a URL.	<c:param name="*name*" value="*value*"/>	name	true	String	Name of the query string parameter.
			value	true	String	Value of the parameter.
<c:import>	Imports the content of a URL-based resource.	<c:import url="*url*" [context="*context*"][var="*varName*"] [scope="{page|request|session|application}"] [varReader="*varReaderName*"] [charEncoding="*charEncoding*"]/> *optional* <c:param> *subtags within the body*	url	true	String	The URL of the resource to import.
			context	true	String	Context name when specifying a relative URL that belongs to a foreign context.
			var	false	String	Name of the exported scoped variable for the resource's content.
			scope	true	String	Scope for *var*.
			charEncoding	true	String	Character encoding of the content at the input resource.
			varReader	true	String	Name of the exported scoped variable for the resource's content. (Type: Reader)
						Constraints: If varReader is specified, var and scope must not be specified.
<c:url>	Builds a URL with the proper rewriting rules applied.	<c:url value="*value*" [context="*context*"][var="*varName*"] [scope="{page|request|session|application}"] /> *optional* <c:param> *subtags within the body*	value	true	String	URL to be processed.
			context	true	String	Name of the ex. sc. variable for the resource's content.
			var	false	String	Name of the ex. sc. variable for the processed url.
			scope	false	String	Scope for *var*.

SQL Actions

Action	Description	Syntax	*name*	*dyn**	*type*	*description*
<sql:query>	Queries a database.	<sql:query sql="*sqlQuery*" var="*varName*" [scope="{page|request|session|application}"] [dataSource="*dataSource*"] [maxRows="*maxRows*"] [startRow="*startRow*"]/> *optional* <sql:param> *actions within the body*	sql	true	String	SQL query statement.
			dataSource	true	String, javax.sql.DataSource	JNDI Data source
			maxRows	true	int	The maximum number of rows to be included in the query result.
			startRow	true	int	The returned Result object includes the rows starting at the specified index.
			var	false	String	Name of the exported scoped variable for the query result.
			scope	false	String	Scope for *var*.
<sql:update>	Executes an SQL INSERT, UPDATE, or DELETE statement. (and SQL statements that return nothing)	<sql:update sql="*sqlUpdate*" [dataSource="*dataSource*"] [var="*varName*"] [scope="{page|request|session|application}"]/> *optional* <sql:param> *actions within the body*	sql	true	String	SQL update statement.
			dataSource	true	String, javax.sql.DataSource	JNDI Data source
			var	false	String	Name of the exported scoped variable for the result of the database update.(Integer)
			scope	false	String	Scope for *var*.
<sql:transaction>	Establishes a transaction context for <sql:query> and <sql:update> subtags.	<sql:transaction [dataSource="*dataSource*"] [isolation=*isolationLevel*]> *<sql:query> and <sql:update> statements* </sql:transaction>	dataSource	true	String, javax.sql.DataSource	JNDI Data source
			isolation	true	String	Max contain read_committed | read_uncommitted | repeatable_read | serializable.
						Constraints: Any nested <sql:query> and <sql:update> actions must not specify a dataSource attribute.
<sql:setDataSource>	Exports a data source either as a scoped variable or as the data source configuration variable (javax.servlet.jsp.jstl.sql.dataSource).	<sql:setDataSource {dataSource="*dataSource*" | [url="*jdbcUrl*"] [driver="*driverClassName*"] [user="*userName*"] [password="*password*"]} [var="*varName*"] [scope="{page|request|session|application}"]/>	dataSource	true	String, javax.sql.DataSource	JNDI Data source
			driver	true	String	Class name of the JDBC driver.
			url	true	String	JDBC database URL.
			user	true	String	JDBC database user
			password	true	String	JDBC user password
			var	false	String	Name of the ex. sc. variable for the data source specified. Type: String / DataSource.
			scope	false	String	Scope for *var*.
<sql:param>	Sets the values of parameter markers ("?") in a SQL statement.	<sql:param value="*value*"/>	value	true	Object	Parameter value.
<sql:dateParam>	Like <sql:param> for values of type java.util.Date. Subtag of <sql:query> and <sql:update>.	<sql:dateParam value="*value*" type="{timestamp|time|date}"/>	value	true	java.util.Date	Parameter value for DATE, TIME, or TIMESTAMP column.
			type	true	String	One of "date", "time" or "timestamp".

* The "dyn" (dynamic) column title presents the rtexprvalue element defined in a TLD. It captures the dynamic behaviour of an attribute for both library versions.

The content is based on the JavaServer™ Pages Standard Tag Library specification, version 1.0.(http://java.sun.com)

Action	Description	Syntax	name	dyn*	type	description
XML Tag Library - XML Core Actions			*name*	*dyn**	*type*	*description*
`<x:out>`	Evaluates an XPath expression and outputs the result to the current JspWriter object.	`<x:out select="XPathExpression " [escapeXml="{true|false}"]/>`	select	false	String	XPath expression.
			escapeXml	true	boolean	Determines whether <,>,&,',” in the result should be converted to their corresponding character entity codes.
`<x:parse>`	Parses an XML document.	`<x:parse xml="XMLDocument " {var="var " [scope={page|request|session|application}] |varDom="var " [scopeDom="scope"]} [systemId="systemId"] [filter="filter"]/>`	xml	true	String, Reader	Source XML document to be parsed
			systemId	true	String	The system identifier (URI) for parsing the XML document.
			filter	true	org.xml.sax. XMLFilter	Filter to be applied to the source document.
			var	true	String	Name of the exported scoped variable for the parsed XML document.
			scope	false	String	Scope for *var*.
			varDom	false	String	See *var*. The type of the sc. variable is org.w3c.dom.Document.
			scopeDom	false	String	Scope for *varDom* .
`<x:set>`	Evaluates an XPath expression and stores the result into a scoped variable.	`<x:set select="XPathExpression " var="varName " [scope="{page|request|session|application}"]/>`	select	false	String	XPath expression to be evaluated.
			var	false	String	Name of the ex. sc. variable to hold the value specified in the action.
			scope	false	String	Scope for *var* .
XML Tag Library - XML Flow Control Actions			*name*	*dyn**	*type*	*description*
`<x:if>`	Evaluates the XPath expression specified in the select attribute and renders its body content if the expression evaluates to true.	`<x:if select="XPathExpression " [var="varName "] [scope="{page|request|session|application}"]> body content </x:if>`	select	false	String	The test condition that tells whether or not the body content should be processed
			var	false	String	Name of the ex. sc. variable for the resulting value of the condition. (Boolean
			scope	false	String	Scope for *var*
			Constraints: If scope is specified, var must also be specified.			
`<x:when>`	Provides the context for mutually exclusive conditional execution.	`<x:choose> <x:when select="XPathExpr ">body content </x:when> <x:otherwise>conditional block </x:otherwise> </x:choose>`	select	false	String	Test condition that determines whether or not the body content should be processed.
			Constraints: The body of <x:choose> can only contain:			
			1 or more <x:when> actions (must all appear before <c:otherwise>)			
			0 or 1 <x:otherwise> action (must be the last action)			
`<x:forEach>`	Evaluates the given XPath expression and repeats its nested body content over the result.	`<x:forEach [var="varName "] select="XPathExpression "> body content </x:forEach>`	var	false	String	Name of the exported scoped variable for the current item of the iteration.
			select	false	String	XPath expression to be evaluated.
XML Tag Library - XML Transform Actions			*name*	*dyn**	*type*	*description*
`<x:transform>`	Applies an XSLT stylesheet transformation to an XML document.	`<x:transform xml="XMLDocument " xslt="XSLTStylesheet " [xmlSystemId="XMLSystemId"] [xsltSystemId="XSLTSystemId"] [{var="varName " [scope="{page|request|session|application}"]| result="resultObject "}]/>` optional `<x:param>` subtags within the body	xml	true	String, Reader, javax.xml. transform.Source	Source XML document to be transformed.
			xslt	true	see above (xml)	Type also: org.w3c.dom.Document, or object exported by <x:parse>, <x:set>. Transformation stylesheet as a String, Reader,or Source object.
			xmlSystemId	true	String	The system identifier (URI) for parsing the XML document.
			xsltSystemId	true	String	The system identifier (URI) for parsing the XSLT stylesheet.
			var	false	String	Name of the ex. sc. variable for the transformed XML document.(org.w3c.dom.Document)
			scope	false	String	Scope for *var*.
			result	true	javax.xml. transform.Result	Object that captures or processes the transformation result.
`<x:param>`	Set transformation parameters. Nested	`<x:param name="name " value="value "/>`	name	true	String	Name of the transformation parameter.
			value	true	Object	Value of the parameter.
I18n-capable formatting Tag Library - Internationalization (i18n) Actions			*name*	*dyn**	*type*	*description*
`<fmt:setLocale>`	Stores the specified locale in the javax.servlet.jsp.jstl.fmt.locale configuration variable.	`<fmt:setLocale value="locale " [variant="variant "] [scope="{page|request|session|application}"]/>`	value	true	String,	A String value is interpreted as the printable representation of a locale. Language and country codes must be separated by hyphen ('-') or underscore ('_').
			variant	true	String	Vendor- or browser-specific variant.
			scope	false	String	Scope of the *locale* configuration variable.
`<fmt:bundle>`	Creates an i18n localization context to be used by its body content.	`<fmt:bundle basename="basename " [prefix="prefix "]> body content </fmt:bundle>`	basename	true	String	Resource bundle base name. (fully-qualified)
			prefix	true	String	Prefix to be prepended to the value of the message key of any nested <fmt:message> action.
`<fmt:setBundle>`	Creates an i18n localization context and stores it in the scoped variable or the localizationContext.	`<fmt:setBundle basename="basename " [var="varName "] [scope="{page|request|session|application}"]/>`	basename	true	String	Resource bundle base name. (fully-qualified)
			var	false	String	Name of the exported scoped variable which stores the i18n localization context.
			scope	false	String	Scope of var or the localization context configuration variable.
`<fmt:message>`	Looks up a localized message in a resource bundle.	`<fmt:message key="messageKey " [bundle="resourceBundle "][var="varName "] [scope="{page|request|session|application}"]/>` option key and/or optional `<fmt:param>` within the body	key	true	String	Message key to be looked up
			bundle	true	Localization Context	Localization context in whose resource bundle the message key is looked up
			var	false	String	Name of the exported scoped variable
			scope	false	String	Scope for *var*
			Constraints: If scope is specified, var must also be specified.			
I18n-capable formatting Tag Library - Internationalization (i18n) Actions			*name*	*dyn**	*type*	*description*
`<fmt:param>`	Supplies a single parameter within a <fmt:message> action.	`<fmt:param value="messageParameter "/>`	value	true	Object	Argument used for parametric replacement.
`<fmt:requestEncoding>`	Sets the request's char. encoding.	`<fmt:requestEncoding [value="charsetName "]/>`	value	true	String	Name of character encoding to be applied when decoding request parameters.
I18n-capable formatting Tag Library - Formatting Actions			*name*	*dyn**	*type*	*description*
`<fmt:timeZone>`	Specifies the time zone in which time information is to be formatted or parsed in its body content.	`<fmt:timeZone value="timeZone "> body content </fmt:timeZone>`	value	true	String, java. util.TimeZone	The time zone. A String value is interpreted as a time zone ID.
`<fmt:setTimeZone>`	Stores the specified time zone in a scoped variable or the time zone configuration variable.	`<fmt:setTimeZone value="timeZone " [var="varName "] [scope="{page|request|session|application}"]/>`	value	true	String, java. util.TimeZone	The time zone. A String value is interpreted as a time zone ID.
			var	false	String	Name of the exported scoped variable which stores the time zone of type java.util.TimeZone.
			scope	false	String	Scope of *var* or the time zone configuration variable.
`<fmt:formatNumber>`	Formats a numeric value in a locale-sensitive or customized manner as a number, currency, or percentage.	`<fmt:formatNumber value="numericValue " [type="{number|currency|percent}"] [pattern="customPattern "] [currencyCode="currencyCode "] [currencySymbol="currencySymbol "] [groupingUsed="{true|false}"] [maxIntegerDigits="maxIntegerDigits "] [minIntegerDigits="minIntegerDigits "] [maxFractionDigits="maxFractionDigits "] [minFractionDigits="minFractionDigits "] [var="varName "] [scope="{page|request|session|application}"]/>`	value	true	String, Number	Numeric value to be formatted
			type	true	String	Specifies whether the value is to be formatted as number, currency, or percentage
			pattern	true	String	Custom formatting pattern.
			currencyCode	true	String	ISO 4217 currency code. Applied only when formatting currencies; ignored otherwise
			currencySymbol	true	String	Currency symbol. Applied only when formatting currencies; ignored otherwise
			groupingUsed	true	boolean	Specifies whether the formatted output will contain any grouping separator
			maxIntegerDigits	true	int	Max. number of digits in the integer portion
			minIntegerDigits	true	int	Min. number of digits in the integer portion
			maxFractionDigits	true	int	Max. number of digits in the fractional portion
			minFractionDigits	true	int	Min. number of digits in the fractional portion
			var	false	String	Name of the exported scoped variable
			scope	false	String	Scope of *var* .
`<fmt:parseNumber>`	Parses the string representation of numbers, currencies, and percentages that were formatted in a locale-sensitive or customized manner.	`<fmt:parseNumber value="numericValue " [type="{number|currency|percent}"] [pattern="customPattern "] [parseLocale="parseLocale "] [integerOnly="{true|false}"] [var="varName "] [scope="{page|request|session|application}"]/>`	value	true	String	String to be parsed
			type	true	String	Specifies whether the value attribute should be parsed as a number, currency, or percentage
			pattern	true	String	Custom formatting pattern.
			parseLocale	true	String,	Locale whose default formatting pattern is to be used
			integerOnly	true	boolean	Specifies whether just the integer portion of the given value should be parsed
			var	false	String	Name of the exported scoped variable which stores the parsed result (Number
			scope	false	String	Scope of *var* .
			Constraints: If scope is specified, var must also be specified.			
`<fmt:formatDate>`	Allows the formatting of dates and time in a locale-sensitive or customized manner.	`<fmt:formatDate value="date " [type="{time|date|both}"] [dateStyle="{default|short|medium|long|full}"] [timeStyle="{default|short|medium|long|full}"] [pattern="customPattern "] [timeZone="timeZone "] [var="varName "] [scope="{page|request|session|application}"]/>`	value	true	java.util.Date	Date and/or time to be formatted
			type	true	String	Specifies whether the time, the date, or both are to be formatted.
			dateStyle	true	String	Predefined formatting style for dates.
			timeStyle	true	String	Predefined formatting style for times.
			pattern	true	String	Custom formatting style for dates and times.
			timeZone	true	String, java. util.TimeZone	Time zone in which to represent the formatted time.
			var	false	String	Name of the exported scoped variable which stores the formatted result as a String.
			scope	false	String	Scope of *var* .
`<fmt:parseDate>`	Parses the string representation of dates and times that were formatted in a locale-sensitive or customized manner.	`<fmt:parseDate value="dateString " [type="{time|date|both}"] [dateStyle="{default|short|medium|long|full}"] [timeStyle="{default|short|medium|long|full}"] [pattern="customPattern "] [timeZone="timeZone "] [parseLocale="parseLocale "] [var="varName "] [scope="{page|request|session|application}"]/>`	value	true	String	Date string to be parsed
			type	true	String	Specifies whether the value contains a time,date,or both
			dateStyle	true	String	Predefined formatting style for dates
			timeStyle	true	String	Predefined formatting style for times
			pattern	true	String	Custom formatting pattern for parsing the date string
			timeZone	true	String, java. util.TimeZone	Time zone in which to interpret any time information
			parseLocale	true	String, java.util.Locale	Locale whose default formatting pattern is to be used
			var	false	String	Name of the ex. sc. variable in which the parsing result (of type java.util.Date) is stored
			scope	false	String	Scope of *var* .
			Constraints: If scope is specified, var must also be specified.			

The "dyn" (dynamic) column title presents the rtexprvalue element defined in a TLD. It captures the dynamic behaviour of an attribute for both library versions.

The content is based on the JavaServer™ Pages Standard Tag Library specification, version 1.0.(http://java.sun.com)