

# Tag Library Documentation

This document was thoughtfully generated from the following files:

JSTL core (c.tld)	
<u>c</u> .....	2
JSTL fmt (fmt.tld)	
<u>fmt</u> .....	25
JSTL functions (fn.tld)	
<u>fn</u> .....	44
JSTL sql (sql.tld)	
<u>sql</u> .....	62
JSTL XML (x.tld)	
<u>x</u> .....	71

## Tag Library Summary

**Tag Library Version:**1.1

**Display Name:**JSTL core

**URI:**<http://java.sun.com/jsp/jstl/core>

**File Name:**c.tld

**Validator class:**org.apache.taglibs.standard.tlv.JstlCoreTLV Provides core validation features for JSTL tags.

JSTL 1.1 core library

## Utility Tags

Tags for rendering data, creating and modifying scoped variables and for catching exceptions

### Referenced Tags:

<a href="#">c:catch</a> .....	4
<a href="#">c:out</a> .....	5
<a href="#">c:set</a> .....	6
<a href="#">c:remove</a> .....	8

## Conditional Tags

Tags for conditional execution of the body dependent on an evaluation

### Referenced Tags:

<a href="#">c:choose</a> .....	9
<a href="#">c:when</a> .....	10
<a href="#">c:otherwise</a> .....	11
<a href="#">c:if</a> .....	12

## URL related Tags

Tags for importing content either locally, cross-context or remotely

### Referenced Tags:

<a href="#">c:import</a> .....	14
<a href="#">c:param</a> .....	17
<a href="#">c:redirect</a> .....	18
<a href="#">c:url</a> .....	19

## Iteration Tags

## Tags for iterating over items

### Referenced Tags:

---

<u>c:forEach</u> .....	21
<u>c:forTokens</u> .....	23

## c:catch

**Tag Class:**org.apache.taglibs.standard.tag.common.core.CatchTag

**Body Content:**JSP

Catches any Throwable that occurs in its body and optionally exposes it.

Attribute Summary			
Name	Required	Expressions	Type
var	false	false	String

### var

---

Name of the exported scoped variable for the exception thrown from a nested action.

**Class:** java.lang.ExceptionThe type of the scoped variable is the type of the exception thrown.

Method summary for: java.lang.Exception	
Return Type	Method
String	toString
Throwable	fillInStackTrace
Throwable	cause
String	message
String	localizedMessage
StackTraceElement[]	stackTrace
int	hashCode
Class	class

`c:out`

**Tag Class:**org.apache.taglibs.standard.tag.rt.core.OutTag

**Body Content:**JSP

Output the expression Like `<%= ... >`, but for expressions.

### Example

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

Attribute Summary			
Name	Required	Expressions	Type
value	true	true	String
default	false	true	String
escapeXml	false	true	String

### value

---

The expression to be evaluated.

### default

---

The default value if the resulting value is null.

### escapeXml

---

Determines whether characters `<`, `>`, `&`, `'`, `"` in the resulting string should be converted to their corresponding character entity codes.

Allowable Values	
<b>true</b>	Output will be xml-escaped
<b>false</b>	Output will NOT be xml-escaped
<b>Default</b>	<b>false</b>

## c:set

**Tag Class:**org.apache.taglibs.standard.tag.rt.core.SetTag

**Body Content:**JSP

Sets the result of an expression evaluation in a 'scope'

### Example

```
<c:set var="var" value="${value}" target="${target}"  
property="${property}" scope="scope" >
```

### Attribute Summary

Name	Required	Expressions	Type
var	false	false	String
value	false	true	String
target	false	true	String
property	false	true	String
scope	false	false	String

### var

---

Name of the exported scoped variable to hold the value specified in the action. The type of the scoped variable is whatever type the value expression evaluates to.

### value

---

Expression to be evaluated.

### target

---

Target object whose property will be set. Must evaluate to a JavaBeans object with setter property property, or to a java.util.Map object.

### property

---

Name of the property to be set in the target object.

### scope

---

Scope for var.

### Allowable Values

<b>pageContext</b>	JSP Page Context object
<b>pageScope</b>	Page-scoped variables (valid only on a given JSP page)

<b>requestScope</b>	Request-scoped variables (valid for a given request)
<b>sessionScope</b>	Session-scoped variables (valid for the user's session)
<b>applicationScope</b>	Application-scoped variables (valid for a given application context)

`c:remove`

**Tag Class:**org.apache.taglibs.standard.tag.common.core.RemoveTag

**Body Content:**empty

Removes a scoped variable (from a particular scope, if specified).

Attribute Summary			
Name	Required	Expressions	Type
var	true	false	String
scope	false	false	String

**var**

---

Name of the scoped variable to be removed.

**scope**

---

Scope for var.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)



## c:choose

**Tag Class:**org.apache.taglibs.standard.tag.common.core.ChooseTag

**Body Content:**JSP

Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by <when> and <otherwise>

### Example

```
<c:choose>
<c:when test="{eval}">
...
</c:when>
<c:when test="{eval}">
...
</c:when>
<c:otherwise>
...
</c:otherwise>
</c:choose>
```

## c:when

**Tag Class:**org.apache.taglibs.standard.tag.rt.core.WhenTag

**Body Content:**JSP

Subtag of <choose> that includes its body if its condition evaluates to 'true'

**Required Parent:**

[org.apache.taglibs.standard.tag.common.core.ChooseTag](#)

Attribute Summary			
Name	Required	Expressions	Type
test	true	true	boolean

### **test**

---

The test condition that determines whether or not the body content should be processed.

## c:otherwise

**Tag Class:**`org.apache.taglibs.standard.tag.common.core.OtherwiseTag`

**Body Content:**JSP

Subtag of <choose> that follows <when> tags and runs only if all of the prior conditions evaluated to 'false'

**Required Parent:**

`org.apache.taglibs.standard.tag.common.core.ChooseTag`

c:if

**Tag Class:**org.apache.taglibs.standard.tag.rt.core.IfTag

**Body Content:**JSP

Simple conditional tag, which evaluates its body if the supplied condition is true and optionally exposes a Boolean scripting variable representing the evaluation of this condition

Attribute Summary			
Name	Required	Expressions	Type
test	true	true	boolean
var	false	false	String
scope	false	false	String

### test

---

The test condition that determines whether or not the body content should be processed.

### var

---

Name of the exported scoped variable for the resulting value of the test condition. The type of the scoped variable is Boolean.

**Class:** java.lang.Boolean

Field summary for: java.lang.Boolean	
Type	Field
Boolean	TRUE
Boolean	FALSE
Class	TYPE

  

Method summary for: java.lang.Boolean	
Return Type	Method
int	hashCode
String	toString
boolean	booleanValue
Class	class

### scope

---

Scope for var.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)

# c:import

**Tag Class:**org.apache.taglibs.standard.tag.rt.core.ImportTag

**Body Content:**JSP

Retrieves an absolute or relative URL and exposes its contents to either the page, a String in 'var', or a Reader in 'varReader'.

Attribute Summary			
Name	Required	Expressions	Type
url	true	true	String
var	false	false	String
scope	false	false	String
varReader	false	false	String
context	false	true	String
charEncoding	false	true	String

## url

The URL of the resource to import.

## var

Name of the exported scoped variable for the resource's content. The type of the scoped variable is String.

**Class:** java.lang.String

Field summary for: java.lang.String	
Type	Field
Comparator	CASE_INSENSITIVE_ORDER

  

Method summary for: java.lang.String	
Return Type	Method
int	hashCode
int	length
String	toString
byte[]	getBytes
String	toLowerCase
String	toUpperCase

Method summary for: <code>java.lang.String</code>	
<code>String</code>	<code>trim</code>
<code>char[]</code>	<code>toCharArray</code>
<code>String</code>	<code>intern</code>
<code>Class</code>	<code>class</code>

## scope

---

Scope for var.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)

## varReader

---

Name of the exported scoped variable for the resource's content. The type of the scoped variable is Reader.

**Class:** `java.io.Reader`

Method summary for: <code>java.io.Reader</code>	
Return Type	Method
<code>int</code>	<code>read</code>
<code>boolean</code>	<code>markSupported</code>
<code>boolean</code>	<code>ready</code>
<code>int</code>	<code>hashCode</code>
<code>Class</code>	<code>class</code>
<code>String</code>	<code>toString</code>

## context

---

Name of the context when accessing a relative URL resource that belongs to a foreign context.

## charEncoding

---

Character encoding of the content at the input resource.



## c:param

**Tag Class:**org.apache.taglibs.standard.tag.rt.core.ParamTag

**Body Content:**JSP

Adds a parameter to a containing 'import' tag's URL.

**Required Parent:** [org.apache.taglibs.standard.tag.rt.core.ImportTag](#)

**Required Parent:** [org.apache.taglibs.standard.tag.rt.core.UrlTag](#)

Attribute Summary			
Name	Required	Expressions	Type
name	true	true	String
value	false	true	String

### name

Name of the query string parameter.

### value

Value of the parameter.

## c:redirect

**Tag Class:**org.apache.taglibs.standard.tag.rt.core.RedirectTag

**Body Content:**JSP

Redirects to a new URL.

Attribute Summary			
Name	Required	Expressions	Type
url	false	true	String
context	false	true	String

### url

The URL of the resource to redirect to.

### context

Name of the context when redirecting to a relative URL resource that belongs to a foreign context.

c:url

**Tag Class:**org.apache.taglibs.standard.tag.rt.core.UrlTag

**Body Content:**JSP

Creates a URL with optional query parameters.

Attribute Summary			
Name	Required	Expressions	Type
var	false	false	String
scope	false	false	String
value	false	true	String
context	false	true	String

## var

Name of the exported scoped variable for the processed url. The type of the scoped variable is String.

**Class:** java.lang.String

Field summary for: java.lang.String	
Type	Field
Comparator	CASE_INSENSITIVE_ORDER

  

Method summary for: java.lang.String	
Return Type	Method
int	hashCode
int	length
String	toString
byte[]	getBytes
String	toLowerCase
String	toUpperCase
String	trim
char[]	toCharArray
String	intern
Class	class

## **scope**

---

Scope for var.

<b>Allowable Values</b>	
<b>pageContext</b>	JSP Page Context object
<b>pageScope</b>	Page-scoped variables (valid only on a given JSP page)
<b>requestScope</b>	Request-scoped variables (valid for a given request)
<b>sessionScope</b>	Session-scoped variables (valid for the user's session)
<b>applicationScope</b>	Application-scoped variables (valid for a given application context)

## **value**

---

URL to be processed.

## **context**

---

Name of the context when specifying a relative URL resource that belongs to a foreign context.

# c:forEach

**Tag Class:**org.apache.taglibs.standard.tag.rt.core.ForEachTag

**Body Content:**JSP

The basic iteration tag, accepting many different collection types and supporting subsetting and other functionality

Attribute Summary			
Name	Required	Expressions	Type
items	false	true	java.lang.Object
begin	false	true	int
end	false	true	int
step	false	true	int
var	false	false	String
varStatus	false	false	String

## items

---

Collection of items to iterate over.

## begin

---

If items specified:

- Iteration begins at the item located at the specified index. First item of the collection has index 0.

If items not specified:

- Iteration begins with index set at the value specified.

## end

---

If items specified:

- Iteration ends at the item located at the specified index (inclusive).

If items not specified:

- Iteration ends when index reaches the value specified.

## step

---

Iteration will only process every step items of the collection, starting with the first one.

## **var**

---

Name of the exported scoped variable for the current item of the iteration. This scoped variable has nested visibility. Its type depends on the object of the underlying collection.

## **varStatus**

---

Name of the exported scoped variable for the status of the iteration. Object exported is of type `javax.servlet.jsp.jstl.core.LoopTagStatus`. This scoped variable has nested visibility.

**Class:** `javax.servlet.jsp.jstl.core.LoopTagStatus`

<b>Method summary for: javax.servlet.jsp.jstl.core.LoopTagStatus</b>	
<b>Return Type</b>	<b>Method</b>
int	count
boolean	last
Integer	begin
Integer	end
Integer	step
int	index
boolean	first
Object	current

# c:forTokens

**Tag Class:**org.apache.taglibs.standard.tag.rt.core.ForTokensTag

**Body Content:**JSP

Iterates over tokens, separated by the supplied delimiters

Attribute Summary			
Name	Required	Expressions	Type
items	true	true	String
delims	true	true	String
begin	false	true	int
end	false	true	int
step	false	true	int
var	false	false	String
varStatus	false	false	String

## items

---

String of tokens to iterate over.

## delims

---

The set of delimiters (the characters that separate the tokens in the string).

## begin

---

Iteration begins at the token located at the specified index. First token has index 0.

## end

---

Iteration ends at the token located at the specified index (inclusive).

## step

---

Iteration will only process every step tokens of the string, starting with the first one.

## var

---

Name of the exported scoped variable for the current item of the iteration. This scoped variable has nested visibility.

## varStatus

---

Name of the exported scoped variable for the status of the iteration. Object exported is of type javax.servlet.jsp.jstl.core.LoopTagStatus. This scoped variable has nested visibility.

**Class:** javax.servlet.jsp.jstl.core.LoopTagStatus

Method summary for: javax.servlet.jsp.jstl.core.LoopTagStatus	
Return Type	Method
int	count
boolean	last
Integer	begin
Integer	end
Integer	step
int	index
boolean	first
Object	current



**Tag Library Summary****Tag Library Version:**1.1**Display Name:**JSTL fmt**URI:**<http://java.sun.com/jsp/jstl/fmt>**File Name:**fmt.tld**Validator class:**org.apache.taglibs.standard.tlv.JstlFmtTLV Provides core validation features for JSTL tags.

JSTL 1.1 i18n-capable formatting library

## Utility Tags

## Referenced Tags:

<a href="#">fmt:requestEncoding</a> .....	27
<a href="#">fmt:setLocale</a> .....	28

## Time Zone Tags

Tags that operate on the time zone

## Referenced Tags:

<a href="#">fmt:timeZone</a> .....	29
<a href="#">fmt:setTimeZone</a> .....	30

## Resource Related Tags

Tags that load a resource bundle for use with localized messages

## Referenced Tags:

<a href="#">fmt:bundle</a> .....	31
<a href="#">fmt:setBundle</a> .....	32
<a href="#">fmt:message</a> .....	33
<a href="#">fmt:param</a> .....	34

## Numeric Formatting Tags

Tags related to the formatting of numeric values

## Referenced Tags:

<a href="#">fmt:formatNumber</a> .....	35
<a href="#">fmt:parseNumber</a> .....	37

## Date Formatting Tags

Tags that act upon date objects to be formatted

### Referenced Tags:

---

<u>fmt:formatDate</u> .....	39
<u>fmt:parseDate</u> .....	41

## fmt:requestEncoding

**Tag Class:**org.apache.taglibs.standard.tag.rt.fmt.RequestEncodingTag

**Body Content:**empty

Sets the request character encoding

Attribute Summary			
Name	Required	Expressions	Type
value	false	true	String

### **value**

---

Name of character encoding to be applied when decoding request parameters.

# fmt:setLocale

**Tag Class:**org.apache.taglibs.standard.tag.rt(fmt).SetLocaleTag

**Body Content:**empty

Stores the given locale in the locale configuration variable

Attribute Summary			
Name	Required	Expressions	Type
value	true	true	String
variant	false	true	String
scope	false	false	String

## value

A String value is interpreted as the printable representation of a locale, which must contain a two-letter (lower-case) language code (as defined by ISO-639), and may contain a two-letter (upper-case) country code (as defined by ISO-3166). Language and country codes must be separated by hyphen (-) or underscore (\_).

## variant

Vendor- or browser-specific variant. See the java.util.Locale javadocs for more information on variants.

## scope

Scope of the locale configuration variable.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)

## fmt:timeZone

**Tag Class:**org.apache.taglibs.standard.tag.rt(fmt).TimeZoneTag

**Body Content:**JSP

Specifies the time zone for any time formatting or parsing actions nested in its body

### Attribute Summary

Name	Required	Expressions	Type
value	true	true	String

### value

---

The time zone. A String value is interpreted as a time zone ID. This may be one of the time zone IDs supported by the Java platform (such as "America/Los\_Angeles") or a custom time zone ID (such as "GMT-8"). See java.util.TimeZone for more information on supported time zone formats.

## fmt:setTimeZone

**Tag Class:**org.apache.taglibs.standard.tag.rt.fmt.SetTimeZoneTag

**Body Content:**empty

Stores the given time zone in the time zone configuration variable

Attribute Summary			
Name	Required	Expressions	Type
value	true	true	String
var	false	false	String
scope	false	false	String

### value

---

The time zone. A String value is interpreted as a time zone ID. This may be one of the time zone IDs supported by the Java platform (such as "America/Los\_Angeles") or a custom time zone ID (such as "GMT-8"). See java.util.TimeZone for more information on supported time zone formats.

### var

---

Name of the exported scoped variable which stores the time zone of type java.util.TimeZone.

### scope

---

Scope of var or the time zone configuration variable.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)

## fmt:bundle

**Tag Class:**org.apache.taglibs.standard.tag.rt.fmt.BundleTag

**Body Content:**JSP

Loads a resource bundle to be used by its tag body

Attribute Summary			
Name	Required	Expressions	Type
basename	true	true	String
prefix	false	true	String

### **basename**

---

Resource bundle base name. This is the bundle's fully-qualified resource name, which has the same form as a fully-qualified class name, that is, it uses "." as the package component separator and does not have any file type (such as ".class" or ".properties") suffix.

### **prefix**

---

Prefix to be prepended to the value of the message key of any nested <fmt:message> action.

# fmt:setBundle

**Tag Class:**org.apache.taglibs.standard.tag.rt.fmt.SetBundleTag

**Body Content:**empty

Loads a resource bundle and stores it in the named scoped variable or the bundle configuration variable

Attribute Summary			
Name	Required	Expressions	Type
basename	true	true	String
var	false	false	String
scope	false	false	String

## basename

Resource bundle base name. This is the bundle's fully-qualified resource name, which has the same form as a fully-qualified class name, that is, it uses "." as the package component separator and does not have any file type (such as ".class" or ".properties") suffix.

## var

Name of the exported scoped variable which stores the i18n localization context of type javax.servlet.jsp.jstl.fmt.LocalizationContext.

## scope

Scope of var or the localization context configuration variable.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)



## fmt:message

**Tag Class:**org.apache.taglibs.standard.tag.rt(fmt).MessageTag

**Body Content:**JSP

Maps key to localized message and performs parametric replacement

Attribute Summary			
Name	Required	Expressions	Type
key	false	true	String
bundle	false	true	String
var	false	false	String
scope	false	false	String

### **key**

---

Message key to be looked up.

### **bundle**

---

Localization context in whose resource bundle the message key is looked up.

### **var**

---

Name of the exported scoped variable which stores the localized message.

### **scope**

---

Scope of var.

Allowable Values	
<b>pageContext</b>	JSP Page Context object
<b>pageScope</b>	Page-scoped variables (valid only on a given JSP page)
<b>requestScope</b>	Request-scoped variables (valid for a given request)
<b>sessionScope</b>	Session-scoped variables (valid for the user's session)
<b>applicationScope</b>	Application-scoped variables (valid for a given application context)

## fmt:param

**Tag Class:**org.apache.taglibs.standard.tag.rt.fmt.ParamTag

**Body Content:**JSP

Supplies an argument for parametric replacement to a containing <message> tag

### Attribute Summary

Name	Required	Expressions	Type
value	false	true	String

### **value**

---

Argument used for parametric replacement.

# fmt:formatNumber

**Tag Class:**org.apache.taglibs.standard.tag.rt.fmt.FormatNumberTag

**Body Content:**JSP

Formats a numeric value as a number, currency, or percentage

Attribute Summary			
Name	Required	Expressions	Type
value	false	true	String
type	false	true	String
pattern	false	true	String
currencyCode	false	true	String
currencySymbol	false	true	String
groupingUsed	false	true	String
maxIntegerDigits	false	true	String
minIntegerDigits	false	true	String
maxFractionDigits	false	true	String
minFractionDigits	false	true	String
var	false	false	String
scope	false	false	String

## value

Numeric value to be formatted.

## type

Specifies whether the value is to be formatted as number, currency, or percentage.

Allowable Values	
<b>number</b>	format the value as a number
<b>currency</b>	format the value as a currency
<b>percentage</b>	format the value as a percentage

## pattern

Custom formatting pattern.

---

**currencyCode**

---

ISO 4217 currency code. Applied only when formatting currencies (i.e. if type is equal to "currency"); ignored otherwise.

---

**currencySymbol**

---

Currency symbol. Applied only when formatting currencies (i.e. if type is equal to "currency"); ignored otherwise.

---

**groupingUsed**

---

Specifies whether the formatted output will contain any grouping separators.

---

**maxIntegerDigits**

---

Maximum number of digits in the integer portion of the formatted output.

---

**minIntegerDigits**

---

Minimum number of digits in the integer portion of the formatted output.

---

**maxFractionDigits**

---

Maximum number of digits in the fractional portion of the formatted output.

---

**minFractionDigits**

---

Minimum number of digits in the fractional portion of the formatted output.

---

**var**

---

Name of the exported scoped variable which stores the formatted result as a String.

---

**scope**

---

Scope of var.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)

# fmt:parseNumber

**Tag Class:**org.apache.taglibs.standard.tag.rt.fmt.ParseNumberTag

**Body Content:**JSP

Parses the string representation of a number, currency, or percentage

Attribute Summary			
Name	Required	Expressions	Type
value	false	true	String
type	false	true	String
pattern	false	true	String
parseLocale	false	true	String
integerOnly	false	true	String
var	false	false	String
scope	false	false	String

## **value**

---

String to be parsed.

## **type**

---

Specifies whether the string in the value attribute should be parsed as a number, currency, or percentage.

## **pattern**

---

Custom formatting pattern that determines how the string in the value attribute is to be parsed.

## **parseLocale**

---

Locale whose default formatting pattern (for numbers, currencies, or percentages, respectively) is to be used during the parse operation, or to which the pattern specified via the pattern attribute (if present) is applied.

## **integerOnly**

---

Specifies whether just the integer portion of the given value should be parsed.

## **var**

---

Name of the exported scoped variable which stores the parsed result (of type java.lang.Number).

## scope

---

Scope of var.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)

# fmt:formatDate

**Tag Class:**org.apache.taglibs.standard.tag.rt(fmt).FormatDateTag

**Body Content:**empty

Formats a date and/or time using the supplied styles and pattern

Attribute Summary			
Name	Required	Expressions	Type
value	true	true	String
type	false	true	String
dateStyle	false	true	String
timeStyle	false	true	String
pattern	false	true	String
timeZone	false	true	String
var	false	false	String
scope	false	false	String

## **value**

---

Date and/or time to be formatted.

## **type**

---

Specifies whether the time, the date, or both the time and date components of the given date are to be formatted.

## **dateStyle**

---

Predefined formatting style for dates. Follows the semantics defined in class `java.text.DateFormat`. Applied only when formatting a date or both a date and time (i.e. if type is missing or is equal to "date" or "both"); ignored otherwise.

## **timeStyle**

---

Predefined formatting style for times. Follows the semantics defined in class `java.text.DateFormat`. Applied only when formatting a time or both a date and time (i.e. if type is equal to "time" or "both"); ignored otherwise.

## **pattern**

---

Custom formatting style for dates and times.

## **timeZone**

---

Time zone in which to represent the formatted time.

**var**

---

Name of the exported scoped variable which stores the formatted result as a String.

**scope**

---

Scope of var.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)



# fmt:parseDate

**Tag Class:**org.apache.taglibs.standard.tag.rt(fmt).ParseDateTag

**Body Content:**JSP

Parses the string representation of a date and/or time.

Attribute Summary			
Name	Required	Expressions	Type
value	false	true	String
type	false	true	String
dateStyle	false	true	String
timeStyle	false	true	String
pattern	false	true	String
timeZone	false	true	String
parseLocale	false	true	String
var	false	false	String
scope	false	false	String

## **value**

---

Date string to be parsed.

## **type**

---

Specifies whether the date string in the value attribute is supposed to contain a time, a date, or both.

## **dateStyle**

---

Predefined formatting style for days which determines how the date component of the date string is to be parsed. Applied only when formatting a date or both a date and time (i.e. if type is missing or is equal to "date" or "both"); ignored otherwise.

## **timeStyle**

---

Predefined formatting styles for times which determines how the time component in the date string is to be parsed. Applied only when formatting a time or both a date and time (i.e. if type is equal to "time" or "both"); ignored otherwise.

## **pattern**

---

Custom formatting pattern which determines how the date string is to be parsed.

Allowable Values	
<b>G</b>	Era designator Example 'AD'
<b>y</b>	Year Examples '1996; 96'
<b>M</b>	Month in year Examples 'July; Jul; 07'
<b>w</b>	Week in year Example '27'
<b>W</b>	Week in month Example '2'
<b>D</b>	Day in year Example '189'
<b>d</b>	Day in month Example '10'
<b>F</b>	Day of week in month Example '2'
<b>E</b>	Day in week Examples 'Tuesday; Tue'
<b>a</b>	Am/pm marker Example 'PM'
<b>H</b>	Hour in day (0-23) Example '0'
<b>k</b>	Hour in day (1-24) Example '24'
<b>K</b>	Hour in am/pm (0-11) Example '0'
<b>h</b>	Hour in am/pm (1-12) Example '12'
<b>m</b>	Minute in hour Example '30'
<b>s</b>	Second in minute Example '55'
<b>S</b>	Millisecond Example '978'
<b>Z</b>	Time zone Examples 'Pacific Standard Time; PST; GMT-08:00'
<b>z</b>	Time zone RFC 822 Example '-0800'

## **timeZone**

Time zone in which to interpret any time information in the date string.

## **parseLocale**

Locale whose predefined formatting styles for dates and times are to be used during the parse operation, or to which the pattern specified via the pattern attribute (if present) is applied.

## **var**

Name of the exported scoped variable in which the parsing result (of type `java.util.Date`) is stored.

**Class:** `java.util.Date`

Method summary for: java.util.Date	
Return Type	Method
int	hashCode
String	toString
Object	clone
long	time
int	year
int	month
int	date
int	hours
int	minutes
int	seconds
int	day
String	toLocaleString
String	toGMTString
int	timezoneOffset
Class	class

## scope

---

Scope of var.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)

**Tag Library Summary****Tag Library Version:**1.1**Display Name:**JSTL functions**URI:**<http://java.sun.com/jsp/jstl/functions>**File Name:**fn.tld

JSTL 1.1 functions library

## Testing Functions

The following functions perform tests on the String

Referenced Functions:

<a href="#">fn:contains</a> .....	46
<a href="#">fn:containsIgnoreCase</a> .....	47
<a href="#">fn:endsWith</a> .....	48
<a href="#">fn:startsWith</a> .....	49

## Integer Based Functions

The following functions return an int value which can then be tested or used

Referenced Functions:

<a href="#">fn:indexOf</a> .....	50
<a href="#">fn:length</a> .....	51

## Array Based Functions

The following functions act on, or return an array

Referenced Functions:

<a href="#">fn:split</a> .....	52
<a href="#">fn:join</a> .....	53

## Subset Functions

The following functions return a subset of the original String

Referenced Functions:

<a href="#">fn:substring</a> .....	54
<a href="#">fn:substringAfter</a> .....	55
<a href="#">fn:substringBefore</a> .....	56

## Transformation Functions

The following functions transform the current String

### Referenced Functions:

---

<u>fn:replace</u> .....	57
<u>fn:toLowerCase</u> .....	58
<u>fn:toUpperCase</u> .....	59
<u>fn:trim</u> .....	60
<u>fn:escapeXml</u> .....	61

## fn:contains

**Function Class:**org.apache.taglibs.standard.functions.Functions

**Function Signature:**boolean contains(java.lang.String, java.lang.String)

Tests if an input string contains the specified substring.

```
<c:if test="{fn:contains(name, searchString)}">
```

## fn:containsIgnoreCase

**Function Class:**org.apache.taglibs.standard.functions.Functions

**Function Signature:**boolean containsIgnoreCase(java.lang.String, java.lang.String)

Tests if an input string contains the specified substring in a case insensitive way.

```
<c:if test="{fn:containsIgnoreCase(name, searchString)}">
```

## fn:endsWith

**Function Class:**org.apache.taglibs.standard.functions.Functions

**Function Signature:**boolean endsWith(java.lang.String, java.lang.String)

Tests if an input string ends with the specified suffix.

```
<c:if test="{fn:endsWith(filename, ".txt")}">
```



## fn:startsWith

**Function Class:**org.apache.taglibs.standard.functions.Functions

**Function Signature:**boolean startsWith(java.lang.String, java.lang.String)

Tests if an input string starts with the specified prefix.

```
<c:if test="{fn:startsWith(product.id, "100-")}">
```

## fn:indexOf

**Function Class:**org.apache.taglibs.standard.functions.Functions

**Function Signature:**int indexOf(java.lang.String, java.lang.String)

Returns the index within a string of the first occurrence of a specified substring.

```
`${fn:indexOf(name, "-")}`
```

## fn:length

**Function Class:**org.apache.taglibs.standard.functions.Functions

**Function Signature:**int length(java.lang.Object)

Returns the number of items in a collection, or the number of characters in a string.

You have `#{fn:length(shoppingCart.products)}` in your shopping cart.

## fn:split

**Function Class:**org.apache.taglibs.standard.functions.Functions

**Function Signature:**java.lang.String[] split(java.lang.String, java.lang.String)

Splits a string into an array of substrings.

```
${fn:split(customerNames, ";")}
```

## fn:join

**Function Class:**org.apache.taglibs.standard.functions.Functions

**Function Signature:**java.lang.String join(java.lang.String[], java.lang.String)

Joins all elements of an array into a string.

```
#{fn:join(array, ";")}
```

## fn:substring

**Function Class:**org.apache.taglibs.standard.functions.Functions

**Function Signature:**java.lang.String substring(java.lang.String, int, int)

Returns a subset of a string.

P.O. Box: `${fn:substring(zip, 6, -1)}`

## fn:substringAfter

**Function Class:** org.apache.taglibs.standard.functions.Functions

**Function Signature:** java.lang.String substringAfter(java.lang.String,  
java.lang.String)

Returns a subset of a string following a specific substring.

P.O. Box: `${fn:substringAfter(zip, "-")}`

## fn:substringBefore

**Function Class:**org.apache.taglibs.standard.functions.Functions

**Function Signature:**java.lang.String substringBefore(java.lang.String,  
java.lang.String)

Returns a subset of a string before a specific substring.

Zip (without P.O. Box): `${fn:substringBefore(zip, "-")}`



## fn:replace

**Function Class:**org.apache.taglibs.standard.functions.Functions

**Function Signature:**java.lang.String replace(java.lang.String, java.lang.String, java.lang.String)

Returns a string resulting from replacing in an input string all occurrences of a "before" string into an "after" substring.

```
${fn:replace(text, "-", "?")}
```

## fn:toLowerCase

**Function Class:**org.apache.taglibs.standard.functions.Functions

**Function Signature:**java.lang.String toLowerCase(java.lang.String)

Converts all of the characters of a string to lower case.

Product name: `${fn.toLowerCase(product.name)}`

## fn:toUpperCase

**Function Class:**org.apache.taglibs.standard.functions.Functions

**Function Signature:**java.lang.String toUpperCase(java.lang.String)

Converts all of the characters of a string to upper case.

Product name: `${fn.UpperCase(product.name)}`

`fn:trim`

**Function Class:**`org.apache.taglibs.standard.functions.Functions`

**Function Signature:**`java.lang.String trim(java.lang.String)`

Removes white spaces from both ends of a string.

Name: `${fn.trim(name)}`

## fn:escapeXml

**Function Class:** org.apache.taglibs.standard.functions.Functions

**Function Signature:** java.lang.String escapeXml(java.lang.String)

Escapes characters that could be interpreted as XML markup.

```
#{fn:escapeXml(param:info)}
```

# sql

## Tag Library Summary

**Tag Library Version:**1.1

**Display Name:**JSTL sql

**URI:**<http://java.sun.com/jsp/jstl/sql>

**File Name:**sql.tld

**Validator class:**org.apache.taglibs.standard.tlv.JstlSqlTLV Provides core validation features for JSTL tags.

### JSTL 1.1 sql library

<a href="#">sql:transaction</a> .....	63
<a href="#">sql:query</a> .....	64
<a href="#">sql:update</a> .....	66
<a href="#">sql:param</a> .....	67
<a href="#">sql:dateParam</a> .....	68
<a href="#">sql:setDataSource</a> .....	69

# sql:transaction

**Tag Class:**org.apache.taglibs.standard.tag.rt.sql.TransactionTag

**Body Content:**JSP

Provides nested database action elements with a shared Connection, set up to execute all statements as one transaction.

Attribute Summary			
Name	Required	Expressions	Type
dataSource	false	true	String
isolation	false	true	String

## dataSource

DataSource associated with the database to access. A String value represents a relative path to a JNDI resource or the parameters for the JDBC DriverManager facility.

## isolation

Transaction isolation level. If not specified, it is the isolation level the DataSource has been configured with.

## sql:query

**Tag Class:**org.apache.taglibs.standard.tag.rt.sql.QueryTag

**Body Content:**JSP

Executes the SQL query defined in its body or through the sql attribute.

Attribute Summary			
Name	Required	Expressions	Type
var	true	false	String
scope	false	false	String
sql	false	true	String
dataSource	false	true	String
startRow	false	true	String
maxRows	false	true	String

### var

Name of the exported scoped variable for the query result. The type of the scoped variable is javax.servlet.jsp.jstl.sql.Result (see Chapter 16 "Java APIs").

**Class:** javax.servlet.jsp.jstl.sql.Result

Method summary for: javax.servlet.jsp.jstl.sql.Result	
Return Type	Method
Object[][]	rowsByIndex
String[]	columnNames
boolean	limitedByMaxRows
int	rowCount
SortedMap[]	rows

### scope

Scope of var.

Allowable Values	
pageContext	JSP Page Context object
pageScope	Page-scoped variables (valid only on a given JSP page)
requestScope	Request-scoped variables (valid for a given request)



<b>sessionScope</b>	Session-scoped variables (valid for the user's session)
<b>applicationScope</b>	Application-scoped variables (valid for a given application context)

## **sql**

---

SQL query statement.

## **dataSource**

---

Data source associated with the database to query. A String value represents a relative path to a JNDI resource or the parameters for the DriverManager class.

## **startRow**

---

The returned Result object includes the rows starting at the specified index. The first row of the original query result set is at index 0. If not specified, rows are included starting from the first row at index 0.

## **maxRows**

---

The maximum number of rows to be included in the query result. If not specified, or set to -1, no limit on the maximum number of rows is enforced.

## sql:update

**Tag Class:**org.apache.taglibs.standard.tag.rt.sql.UpdateTag

**Body Content:**JSP

Executes the SQL update defined in its body or through the sql attribute.

Attribute Summary			
Name	Required	Expressions	Type
var	false	false	String
scope	false	false	String
sql	false	true	String
dataSource	false	true	String

### var

---

Name of the exported scoped variable for the result of the database update. The type of the scoped variable is java.lang.Integer.

### scope

---

Scope of var.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)

### sql

---

SQL update statement.

### dataSource

---

Data source associated with the database to update. A String value represents a relative path to a JNDI resource or the parameters for the JDBC DriverManager class.

`sql:param`

**Tag Class:**org.apache.taglibs.standard.tag.rt.sql.ParamTag

**Body Content:**JSP

Sets a parameter in an SQL statement to the specified value.

Attribute Summary			
Name	Required	Expressions	Type
value	false	true	String

**value**

---

Parameter value.

## sql:dateParam

**Tag Class:**org.apache.taglibs.standard.tag.rt.sql.DateParamTag

**Body Content:**empty

Sets a parameter in an SQL statement to the specified java.util.Date value.

Attribute Summary			
Name	Required	Expressions	Type
value	true	true	String
type	false	true	String

### **value**

---

Parameter value for DATE, TIME, or TIMESTAMP column in a database table.

### **type**

---

One of "date", "time" or "timestamp".

## sql:setDataSource

**Tag Class:**org.apache.taglibs.standard.tag.rt.sql.SetDataSourceTag

**Body Content:**empty

Creates a simple DataSource suitable only for prototyping.

Attribute Summary			
Name	Required	Expressions	Type
var	false	false	String
scope	false	false	String
dataSource	false	true	String
driver	false	true	String
url	false	true	String
user	false	true	String
password	false	true	String

### var

---

Name of the exported scoped variable for the data source specified. Type can be String or DataSource.

### scope

---

If var is specified, scope of the exported variable. Otherwise, scope of the data source configuration variable.

Allowable Values	
<b>pageContext</b>	JSP Page Context object
<b>pageScope</b>	Page-scoped variables (valid only on a given JSP page)
<b>requestScope</b>	Request-scoped variables (valid for a given request)
<b>sessionScope</b>	Session-scoped variables (valid for the user's session)
<b>applicationScope</b>	Application-scoped variables (valid for a given application context)

### dataSource

---

Data source. If specified as a string, it can either be a relative path to a JNDI resource, or a JDBC parameters string as defined in Section 10.1.1.

### driver

---

JDBC parameter: driver class name.

### **url**

---

JDBC parameter: URL associated with the database.

### **user**

---

JDBC parameter: database user on whose behalf the connection to the database is being made.

### **password**

---

JDBC parameter: user password

## Tag Library Summary

**Tag Library Version:**1.1

**Display Name:**JSTL XML

**URI:**<http://java.sun.com/jsp/jstl/xml>

**File Name:**x.tld

**Validator class:**org.apache.taglibs.standard.tlv.JstlXmlTLV Provides validation features for JSTL XML tags.

JSTL 1.1 XML library

## Conditional Tags

Providing conditional processing

Referenced Tags:

<a href="#">x:choose</a> .....	72
<a href="#">x:when</a> .....	73
<a href="#">x:otherwise</a> .....	74
<a href="#">x:if</a> .....	75

## Iterative Tags

Tags that iterate

Referenced Tags:

<a href="#">x:forEach</a> .....	76
---------------------------------	----

## Transformation Tags

Tags that parse and perform the transformation

Referenced Tags:

<a href="#">x:transform</a> .....	78
<a href="#">x:param</a> .....	82
<a href="#">x:parse</a> .....	83

## Utility Tags

Referenced Tags:

<a href="#">x:out</a> .....	87
<a href="#">x:set</a> .....	88

## x:choose

**Tag Class:**org.apache.taglibs.standard.tag.common.core.ChooseTag

**Body Content:**JSP

Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by <when> and <otherwise>



## x:when

**Tag Class:**org.apache.taglibs.standard.tag.common.xml.WhenTag

**Body Content:**JSP

Subtag of <choose> that includes its body if its expression evalutes to 'true'

**Required Parent:**

[org.apache.taglibs.standard.tag.common.core.ChooseTag](#)

Attribute Summary			
Name	Required	Expressions	Type
select	true	false	String

### **select**

---

The test condition that tells whether or not the body content should be processed

## x:otherwise

**Tag Class:**`org.apache.taglibs.standard.tag.common.core.OtherwiseTag`

**Body Content:**JSP

Subtag of <choose> that follows <when> tags and runs only if all of the prior conditions evaluated to 'false'

**Required Parent:**

`org.apache.taglibs.standard.tag.common.core.ChooseTag`

`x:if`

**Tag Class:**org.apache.taglibs.standard.tag.common.xml.IfTag

**Body Content:**JSP

XML conditional tag, which evaluates its body if the supplied XPath expression evaluates to 'true' as a boolean

Attribute Summary			
Name	Required	Expressions	Type
select	true	false	String
var	false	false	String
scope	false	false	String

### **select**

---

The test condition that tells whether or not the body content should be processed.

### **var**

---

Name of the exported scoped variable for the resulting value of the test condition. The type of the scoped variable is Boolean.

### **scope**

---

Scope for var.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)

# x:forEach

**Tag Class:**org.apache.taglibs.standard.tag.common.xml.ForEachTag

**Body Content:**JSP

XML iteration tag.

Attribute Summary			
Name	Required	Expressions	Type
var	false	false	String
select	true	false	String
begin	false	true	int
end	false	true	int
step	false	true	int
varStatus	false	false	String

## **var**

---

Name of the exported scoped variable for the current item of the iteration. This scoped variable has nested visibility. Its type depends on the result of the XPath expression in the select attribute.

## **select**

---

XPath expression to be evaluated.

## **begin**

---

Iteration begins at the item located at the specified index. First item of the collection has index 0.

## **end**

---

Iteration ends at the item located at the specified index (inclusive).

## **step**

---

Iteration will only process every step items of the collection, starting with the first one.

## **varStatus**

---

Name of the exported scoped variable for the status of the iteration. Object exported is of type javax.servlet.jsp.jstl.core.LoopTagStatus. This scoped variable has nested visibility.

**Class:** javax.servlet.jsp.jstl.core.LoopTagStatus

Method summary for: javax.servlet.jsp.jstl.core.LoopTagStatus	
Return Type	Method
int	count
boolean	last
Integer	begin
Integer	end
Integer	step
int	index
boolean	first
Object	current

## x:transform

**Tag Class:**org.apache.taglibs.standard.tag.rt.xml.TransformTag

**Body Content:**JSP

Conducts a transformation given a source XML document and an XSLT stylesheet

Attribute Summary			
Name	Required	Expressions	Type
var	false	false	String
scope	false	false	String
result	false	true	String
xml	false	true	String
doc	false	true	String
xmlSystemId	false	true	String
docSystemId	false	true	String
xslt	false	true	String
xsltSystemId	false	true	String

### var

Name of the exported scoped variable for the transformed XML document. The type of the scoped variable is org.w3c.dom.Document.

**Class:** org.w3c.dom.Document

Field summary for: org.w3c.dom.Document	
Type	Field
short	ELEMENT_NODE
short	ATTRIBUTE_NODE
short	TEXT_NODE
short	CDATA_SECTION_NODE
short	ENTITY_REFERENCE_NODE
short	ENTITY_NODE
short	PROCESSING_INSTRUCTION_NODE
short	COMMENT_NODE
short	DOCUMENT_NODE

Field summary for: org.w3c.dom.Document	
short	DOCUMENT_TYPE_NODE
short	DOCUMENT_FRAGMENT_NODE
short	NOTATION_NODE
short	DOCUMENT_POSITION_DISCONNECTED
short	DOCUMENT_POSITION_PRECEDING
short	DOCUMENT_POSITION_FOLLOWING
short	DOCUMENT_POSITION_CONTAINS
short	DOCUMENT_POSITION_CONTAINED_BY
short	DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC
Method summary for: org.w3c.dom.Document	
Return Type	Method
DOMImplementation	implementation
DocumentType	doctype
Element	documentElement
DocumentFragment	createDocumentFragment
String	inputEncoding
String	xmlEncoding
boolean	xmlStandalone
String	xmlVersion
boolean	strictErrorChecking
String	documentURI
DOMConfiguration	domConfig
String	localName
String	namespaceURI
String	prefix
String	nodeName
String	nodeValue
short	nodeType
Node	parentNode
NodeList	childNodes
Node	firstChild
Node	lastChild

Method summary for: org.w3c.dom.Document	
Node	previousSibling
Node	nextSibling
Document	ownerDocument
boolean	childNodes
boolean	attributes
String	baseURI
String	textContent
NamedNodeMap	attributes

## scope

---

Scope for var.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)

## result

---

Result Object that captures or processes the transformation result.

## xml

---

Deprecated. Use attribute 'doc' instead.

## doc

---

Source XML document to be transformed. (If exported by <x:set>, it must correspond to a well-formed XML document, not a partial document.)

## xmlSystemId

---

Deprecated. Use attribute 'docSystemId' instead.

## docSystemId

---

The system identifier (URI) for parsing the XML document.

## xslt

---



javax.xml.transform.Source Transformation stylesheet as a String, Reader, or Source object.

### **xsltSystemId**

---

The system identifier (URI) for parsing the XSLT stylesheet.

## x:param

**Tag Class:**org.apache.taglibs.standard.tag.rt.xml.ParamTag

**Body Content:**JSP

Adds a parameter to a containing 'transform' tag's Transformer

**Required Parent:** [org.apache.taglibs.standard.tag.rt.xml.TransformTag](#)

Attribute Summary			
Name	Required	Expressions	Type
name	true	true	String
value	false	true	String

### **name**

---

Name of the transformation parameter.

### **value**

---

Value of the parameter.

## x:parse

**Tag Class:**org.apache.taglibs.standard.tag.rt.xml.ParseTag

**Body Content:**JSP

Parses XML content from 'source' attribute or 'body'

Attribute Summary			
Name	Required	Expressions	Type
var	false	false	String
varDom	false	false	String
scope	false	false	String
scopeDom	false	false	String
xml	false	true	String
doc	false	true	String
systemId	false	true	String
filter	false	true	String

### var

Name of the exported scoped variable for the parsed XML document. The type of the scoped variable is implementation dependent.

### varDom

Name of the exported scoped variable for the parsed XML document. The type of the scoped variable is org.w3c.dom.Document.

**Class:** org.w3c.dom.Document

Field summary for: org.w3c.dom.Document	
Type	Field
short	ELEMENT_NODE
short	ATTRIBUTE_NODE
short	TEXT_NODE
short	CDATA_SECTION_NODE
short	ENTITY_REFERENCE_NODE
short	ENTITY_NODE
short	PROCESSING_INSTRUCTION_NODE

Field summary for: org.w3c.dom.Document	
short	COMMENT_NODE
short	DOCUMENT_NODE
short	DOCUMENT_TYPE_NODE
short	DOCUMENT_FRAGMENT_NODE
short	NOTATION_NODE
short	DOCUMENT_POSITION_DISCONNECTED
short	DOCUMENT_POSITION_PRECEDING
short	DOCUMENT_POSITION_FOLLOWING
short	DOCUMENT_POSITION_CONTAINS
short	DOCUMENT_POSITION_CONTAINED_BY
short	DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC
Method summary for: org.w3c.dom.Document	
Return Type	Method
DOMImplementation	implementation
DocumentType	doctype
Element	documentElement
DocumentFragment	createDocumentFragment
String	inputEncoding
String	xmlEncoding
boolean	xmlStandalone
String	xmlVersion
boolean	strictErrorChecking
String	documentURI
DOMConfiguration	domConfig
String	localName
String	namespaceURI
String	prefix
String	nodeName
String	nodeValue
short	nodeType
Node	parentNode
NodeList	childNodes

Method summary for: org.w3c.dom.Document	
Node	firstChild
Node	lastChild
Node	previousSibling
Node	nextSibling
Document	ownerDocument
boolean	childNodes
boolean	attributes
String	baseURI
String	textContent
NamedNodeMap	attributes

## scope

Scope for var.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)

## scopeDom

Scope for varDom.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)

## xml

Deprecated. Use attribute 'doc' instead.

**doc**

---

Source XML document to be parsed.

**systemId**

---

The system identifier (URI) for parsing the XML document.

**filter**

---

Filter to be applied to the source document.

x:out

**Tag Class:**org.apache.taglibs.standard.tag.rt.xml.ExprTag

**Body Content:**empty

Like <%= ... >, but for XPath expressions.

Attribute Summary			
Name	Required	Expressions	Type
select	true	false	String
escapeXml	false	true	String

### **select**

---

XPath expression to be evaluated.

### **escapeXml**

---

Determines whether characters <, >, &, ' , " in the resulting string should be converted to their corresponding character entity codes. Default value is true.

Allowable Values	
true	
false	
Default	true

## x:set

**Tag Class:**org.apache.taglibs.standard.tag.common.xml.SetTag

**Body Content:**empty

Saves the result of an XPath expression evaluation in a 'scope'

Attribute Summary			
Name	Required	Expressions	Type
var	true	false	String
select	false	false	String
scope	false	false	String

### var

---

Name of the exported scoped variable to hold the value specified in the action. The type of the scoped variable is whatever type the select expression evaluates to.

### select

---

XPath expression to be evaluated.

### scope

---

Scope for var.

Allowable Values	
<code>pageContext</code>	JSP Page Context object
<code>pageScope</code>	Page-scoped variables (valid only on a given JSP page)
<code>requestScope</code>	Request-scoped variables (valid for a given request)
<code>sessionScope</code>	Session-scoped variables (valid for the user's session)
<code>applicationScope</code>	Application-scoped variables (valid for a given application context)