

JSTL

etiquetas básicas

etiquetas básicas son las etiquetas JSTL más comúnmente utilizados. sintaxis de referencia de biblioteca de códigos de la base es de la siguiente manera:

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

etiqueta	descripción
<C: out>	Para la visualización de los datos en un JSP, como <% = ...>
<C: set>	Para el almacenamiento de datos
<C: extraer>	Para la eliminación de datos
<C: catch>	Para hacer frente a condición de excepción de error, y un almacén de mensajes de error
<C: if>	Y si estamos en el mismo procedimiento general que se usa
<C: choose>	Sí sólo como <c: when> y <c: otherwise> marca para padres
<C: when>	<C: choose> pestaña secundaria se utiliza para determinar si las condiciones establecidas
<C: de otro modo>	<C: choose> pestaña secundaria, a continuación, en <c: when> etiqueta después, cuando <c: when> etiqueta es juzgado como falso es ejecutado
<C: import>	Recuperando una URL absoluta o relativa y, a continuación, se expone a la página de contenidos
<C: forEach>	base etiqueta iterativo, aceptar una variedad de tipos de colección
<C: forTokens>	De acuerdo con el delimitador especificado para separar contenido y los resultados iterativo
<C: param>	Se utiliza para contener o redirigir la página para pasar parámetros
<C: redirigir>	Redireccionado a una nueva URL.
<C: url>	Utilice los parámetros de consulta opcionales para crear una URL

etiquetas de formato

JSTL etiquetas de formato para formatear el texto de salida, fecha, hora, número. sintaxis de referencia de la biblioteca etiqueta de formato es el siguiente:

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

etiqueta	descripción
<Fmt: formatNumber>	Utilizando el formato especificado o en formato digital de precisión
<Fmt: parseNumber>	Resolver un representante de la serie, la moneda, o el porcentaje de una cadena
<Fmt: formatDate>	Usando el estilo especificado o el patrón de formato de fecha y hora
<Fmt: parseDate>	Analizar una cadena que representa la fecha o la hora
<Fmt: bundle>	Recursos de enlace
<Fmt: setLocale>	región
<Fmt: setBundle>	Recursos de enlace
<Fmt: timeZone>	Especificar la zona horaria
<Fmt: setTimezone>	Especificar la zona horaria
<Fmt: message>	Muestra la información de perfil de recursos
<Fmt: requestEncoding>	Establecer la codificación de caracteres de la solicitud

etiqueta de SQL

biblioteca de etiquetas JSTL SQL proporciona una base de datos relacional (Oracle, MySQL, SQL Server, etc.) interactúan etiqueta. biblioteca SQL sintaxis de referencia de etiqueta de la siguiente manera:

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

etiqueta	descripción
<Sql: setDataSource>	Especificar la fuente de datos
<Sql: query>	ejecución de la consulta SQL
<Sql: actualización>	Ejecutar instrucción de actualización SQL
<Sql: param>	El parámetro instrucción SQL para el valor especificado
<Sql: dateParam>	La instrucción SQL para el parámetro de valor de objeto java.util.Date fecha especificada
<Sql: transacción>	A condición de anidado en una base de datos compartida conexión de base de elementos de comportamiento, todos los estados en la forma de una transacción para correr

Las etiquetas XML

biblioteca de etiquetas XML JSTL proporciona etiquetas para crear y manipular documentos XML. sintaxis de referencia de la biblioteca de etiquetas XML es el siguiente:

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

Antes de utilizar la etiqueta XML, usted tiene que XML y XPath paquetes relacionados copiados en el <directorio de instalación de Tomcat> \ lib bajo:

- **xercesImpl.jar**
Descargar: <http://www.apache.org/dist/xerces/j/>
- **xalan.jar**
Descargar: <http://xml.apache.org/xalan-j/index.html>

etiqueta	descripción
<X: out>	Y <% = ...>, de forma similar, pero sólo para las expresiones XPath
<X: analizar>	Analizar los datos XML
<X: conjunto>	Ajuste de expresión XPath
<X: si>	El análisis de la expresión XPath, si bien es cierto, entonces el cuerpo del contenido o cuerpo Saltar
<X: forEach>	los nodos del documento XML iterativo
<X: choose>	<X: cuando> y <x: de otro modo> marca para padres
<X: cuando>	<X: elegir> etiqueta hijo sea evaluado
<X: de otro modo>	<X: choose> etiqueta hijo cuando <x: cuando> juzgado como falso cuando se ejecuta
<X: transformar>	Las transformaciones XSL en un documento XML
<X: param>	Y <x: Transformar> se utilizan conjuntamente para establecer la hoja de estilo XSL

función JSTL

JSTL incluye una serie de funciones estándar, la mayoría de las cuales son funciones genéricas de manejo de cadenas. sintaxis de referencia de la biblioteca JSTL es el siguiente:

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

función	descripción
fn: contains ()	Probar si la cadena de entrada contiene la subcadena especificada
fn: containsIgnoreCase ()	Probar si la cadena de entrada contiene la subcadena especificada, entre mayúsculas y minúsculas
fn: endsWith ()	Comprobar si la cadena de entrada termina con el sufijo especificado
fn: escapeXml ()	Saltar como caracteres de marcado XML
fn: indexOf ()	Devuelve la posición de la cadena especificada que aparece en la cadena de entrada
fn: join ()	La síntesis de los elementos de la matriz y da salida a una cadena
fn: longitud ()	Devuelve la longitud de la cadena
fn: replace ()	La ubicación especificada en la cadena de entrada para reemplazar la cadena especificada y devuelve
fn: split ()	Las cadenas están separadas por el separador especificado y luego la formación de una matriz de sub-cadena y devuelve
fn: startsWith ()	Comprobar si la cadena de entrada comienza con el prefijo especificado
fn: substring ()	Devuelve un subconjunto de una cadena
fn: substringAfter ()	Devuelve una cadena en el subconjunto de la subcadena especificada después
fn: substringBefore ()	Devuelve un subconjunto de cadena antes de la subcadena especificada
fn: toLowerCase ()	La cadena de caracteres a minúsculas
fn: toUpperCase ()	Los caracteres de una cadena a mayúsculas
fn: trim ()	Retire el primer espacio en blanco