

JSP Standard Tag Library (JSTL) является стандартной библиотекой тегов, который обеспечивает теги для управления поведением страницы, повторение команд управления, интернационализация тегов и тегов SQL.

JSTL является частью **Java EE API**, и большинства контейнеров servlet. Но чтобы использовать **JSTL** в ваших страницах JSP, необходимо скачать библиотеки **JSTL** для вашего контейнера Servlet (container servlet).

Обзор функций JSTL

На основании функций JSTL, они делятся на пять категорий.

Functions	Description / Declaration
Core Tags	Core tags provide support for iteration, conditional logic, catch exception, url, forward or redirect response etc. To use JSTL core tags, you should include it in the JSP page like below.
	<ul style="list-style-type: none">• <code><%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %></code>
Formatting and Localization Tags	These tags are provided for formatting of Numbers, Dates and i18n support through locales and resource bundles. You can include these tags in JSP with below syntax:
	<ul style="list-style-type: none">• <code><%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %></code>
SQL Tags:	JSTL SQL Tags provide support for interaction with relational databases such as Oracle, MySql etc. Using SQL tags you can run database queries, you include it in JSP with below syntax:
	<ul style="list-style-type: none">• <code><%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %></code>
XML Tags	XML tags are used to work with XML documents such as parsing XML, transforming XML data and XPath expressions evaluation. Syntax to include XML tags in JSP page is:
	<ul style="list-style-type: none">• <code><%@ taglib uri="http://java.sun.com/jsp/jstl/xml" prefix="x" %></code>
JSTL Functions Tags	JSTL tags provide a number of functions that you can use to perform common operation, most of them are for String manipulation such as String Concatenation, Split String etc. Syntax to include JSTL functions in JSP page is:
	<ul style="list-style-type: none">• <code><%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %></code>

Вы можете скачать библиотеки, предоставленные **Apache** из **Maven Repository**:

<http://mvnrepository.com/artifact/org.apache.taglibs/taglibs-standard-spec>

- Core: содержит основные теги для наиболее распространенных задач.

- Использует префикс "c" и uri "http://java.sun.com/jsp/jstl/core"
- **Formatting:** предоставляет теги для форматирования чисел, дат, времени.
Использует префикс "fmt" и uri "http://java.sun.com/jsp/jstl/fmt"
- **SQL:** предоставляет теги для sql-запросами и источниками данных.
Использует префикс "sql" и uri "http://java.sun.com/jsp/jstl/sql"
- **XML:** предоставляет теги для работы с xml.
Использует префикс "x" и uri "http://java.sun.com/jsp/jstl/xml"
- **Functions:** предоставляет функции для работы со строками.
Использует префикс "fn" и uri "http://java.sun.com/jsp/jstl/functions"

Для подключения функционала этих библиотек на страницу jsp применяется директива taglib. Например, подключения основной библиотеки:

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

Стандартные теги JSTL (JSTL Core Tags)

Tag	Description
<c:out>	To write something in JSP page, you can use EL also with this tag
<c:import>	Same as <jsp:include> or include directive
<c:redirect>	redirect request to another resource
<c:set>	To set the variable value in given scope.
<c:remove>	To remove the variable from given scope
<c:catch>	To catch the exception and wrap it into an object.
<c:if>	Simple conditional logic, used with EL and you can use it to process the exception from <c:catch>
<c:choose>	Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by <c:when> and <c:otherwise>
<c:when>	Subtag of <c:choose> that includes its body if its condition evalutes to 'true'.
<c:otherwise>	Subtag of <c:choose> that includes its body if its condition evalutes to 'false'.
<c:forEach>	for iteration over a collection
<c:forTokens>	for iteration over tokens separated by a delimiter.
<c:param>	used with <c:import> to pass parameters
<c:url>	to create a URL with optional query string parameters

Операторы EL

Полезное свойство EL это возможность выполнения сравнения, или между числами или объектами. Это свойство в основном используется для значений кастомизированных тегов атрибутов, но можно использовать и для написания результата сравнения (правильно или неправильно) для страницы JSP. EL предоставляет следующие операторы сравнения:

Равно (equals)	==	eq
Неравно (Not equals)	!=	ne
Меньше (Less than)	<	lt
Больше (Greater than)	>	gt
Меньше или равно (Less than or equals)	<=	le
Больше или равно (Greater than or equals)	>=	ge

2-ой способ сравнения стоит использовать, чтобы избежать путаницы между знаками < или > с тегом в JSP. Но, поверение двух способов одинаково. Смотрите пример ниже:

Понятие	Условие EL	Результат
Число меньше	<code>\${ 1 < 2 }</code>	true
Число больше	<code>\${ 1 > 2 }</code>	false
Число меньше (Less than)	<code>\${ 1 lt 2 }</code>	true
Число больше (Greater than)	<code>\${ 1 gt 2 }</code>	false
Число больше или равно	<code>\${ 1 >= 1 }</code>	true
Число меньше или равно	<code>\${ 1 <= 1 }</code>	true
Число меньше или равно (Less equals)	<code>\${ 1 le 1 }</code>	true
Число больше или равно (Greater than or equal)	<code>\${ 1 ge 1 }</code>	true
Число равно	<code>\${ 1 == 1 }</code>	true
Число равно (equal)	<code>\${ 1 eq 1 }</code>	true
Число неравно	<code>\${ 1 != 2 }</code>	true
Число неравно (not equal)	<code>\${ 1 ne 2 }</code>	true
Буква меньше	<code>\${ 'abe' < 'ade' }</code>	true
Буква больше	<code>\${ 'abe' > 'ade' }</code>	false

Буква равна (Equals)	<code>\${'abe' eq 'abe'}</code>	true
Буква не равна (Not equals)	<code>\${'abe' ne 'ade'}</code>	true

Вы так же можете использовать операторы `&&` и `||`.

Оператор	Описание
<code>&&</code>	И
<code> </code>	Или

Пример:

<code>\${(guess >= 10) && (guess <= 20)}</code>	<code>\${ guess >= 10 && guess <= 20 }</code>
<code>\${(guess < 10) (guess > 20)}</code>	<code>\${ guess < 10 guess > 20 }</code>

Основные возможности JSTL

Цикл

Вывод в цикле элементов массива или коллекции:

```
<c:forEach var="user" items="${users}">
  <p>${user}</p>
</c:forEach>
```

В данном случае параметр "items" указывает на коллекцию, элементы которой выводятся. А параметр "var" задает переменную, через которую доступен текущий перебираемый элемент.

```
<ul>
  <c:forEach var="cook" items="${cookie}">
    <li>
      <p><c:out value="${cook.value.name}" /></p>
      <p><c:out value="${cook.value.value}" /></p>
    </li>
  </c:forEach>
</ul>
```

Условные выражения

Выражение if:

```
<c:if test="${isVisible == true}">
  <p>Visible</p>
</c:if>
```

В данном случае если атрибут isVisible равен true, то выводится код, который расположен между тегами `<c:if>` и `</c:if>`

Тег choose

Тег `c:choose` подобно конструкции `switch...case` в Java проверяет объект на соответствие одному из значений.

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <title>JSTL in JSP</title>
```

```

</head>
<body>
<c:choose>
  <c:when test="${val == 1}">
    <p>Equals to 1</p>
  </c:when>
  <c:when test="${val == 2}">
    <p>Equals to 2</p>
  </c:when>
  <c:otherwise>
    <p>Undefined</p>
  </c:otherwise>
</c:choose>
</body>
</html>

```

В данном случае тег `c:choose` проверяет значение атрибута "val". Для проверки применяются вложенные теги `c:when`, которые аналогичны блокам `case` в конструкции `switch..case`. С помощью их параметра `test` значение атрибута сравнивается с определенным значением. И если выражения сравнения истинно, то выводится код, который размещен внутри данного элемента `c:when`. Таким образом мы можем определить несколько блоков `c:when`. Дополнительный тег `<c:otherwise>` выполняется, если условия проверки значения во всех тегах `c:when` ложно.

Тег url

Тег `<c:url>` позволяет создать адрес относительно корня приложения. Этот тег может применяться, например, при создании ссылок.

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <title>JSTL in JSP</title>
</head>
<body>
<a href='<c:url value="/edit" />'><c:url value="/edit" /></a>
</body>
</html>

```

Редирект

С помощью тега `redirect` можно установить редирект на другой адрес. Например, в случае если атрибут `val` не определен, то делаем редирект на страницу "notfound.jsp":

```

<c:if test="${val == null}">
  <c:redirect url="/notfound.jsp" />
</c:if>

```
