Server-side Web Programming

Lecture 21:

The JSP Standard TagLib (JSTL)

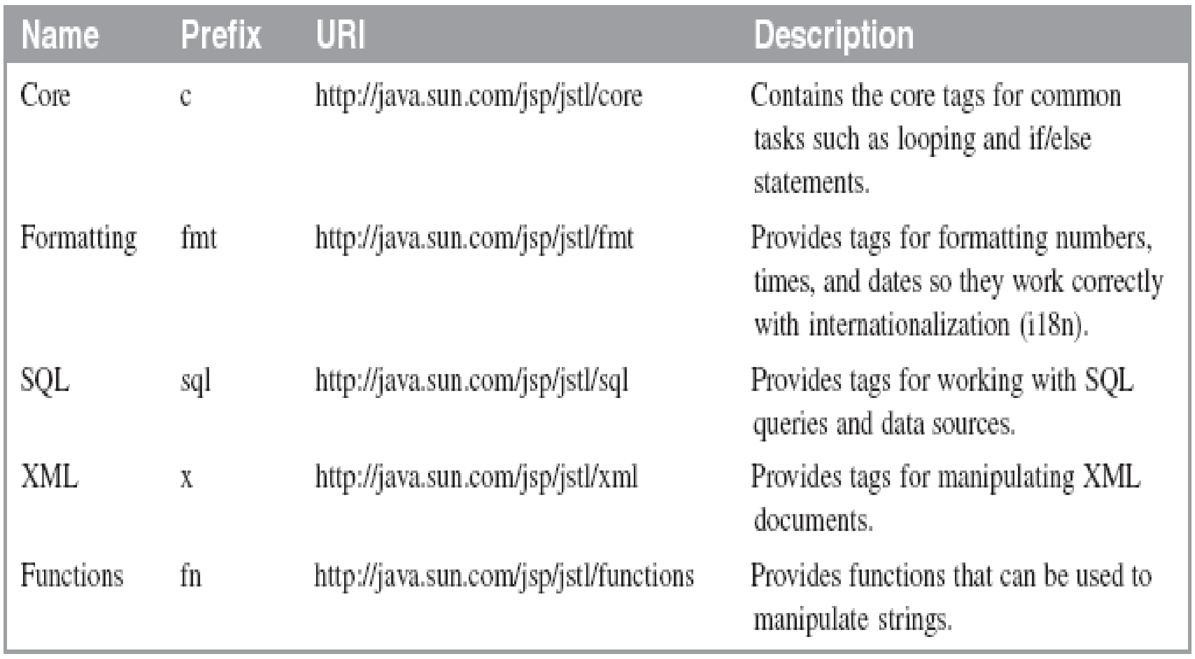
Objectives

- JSTL provides tags for common JSP tags that needed to be performed in JSP to reduce the amount of scripting in your applications, make it easier to code and read than equivalent JSP script, especially for web designers and other nonprogrammers who are used to HTML syntax

Overview

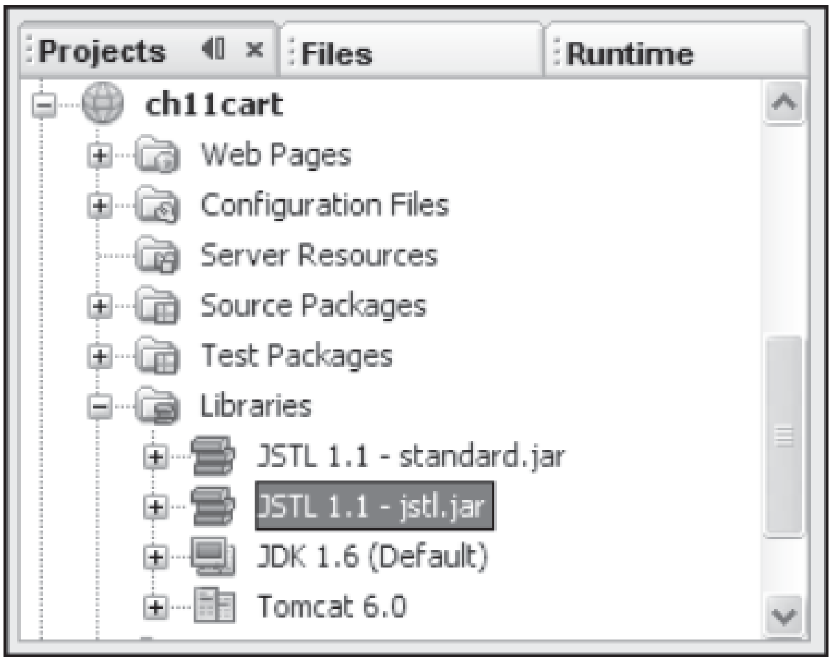
- **JSTL is not part of the JSP 1.2 Specification**
  - It is a separate specification that requires a separate download
  - Available only in servers that support servlets 2.3 and JSP 1.2 or later. Cannot be retrofitted into JSP 1.1.
- **The JSTL expression language will be part of JSP 2.0**

JSTL Components



Name	Prefix	URI	Description
Core	c	http://java.sun.com/jsp/jstl/core	Contains the core tags for common tasks such as looping and if/else statements.
Formatting	fmt	http://java.sun.com/jsp/jstl/fmt	Provides tags for formatting numbers, times, and dates so they work correctly with internationalization (i18n).
SQL	sql	http://java.sun.com/jsp/jstl/sql	Provides tags for working with SQL queries and data sources.
XML	x	http://java.sun.com/jsp/jstl/xml	Provides tags for manipulating XML documents.
Functions	fn	http://java.sun.com/jsp/jstl/functions	Provides functions that can be used to manipulate strings.

Make the JSTL JAR file available to your applications



How to code a JSTL tag

The taglib directive that specifies the JSTL core library

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

An example that uses JSTL to encode a URL

JSP code with JSTL

```
<a href="<c:url value='/index.jsp' />">Continue Shopping</a>
```

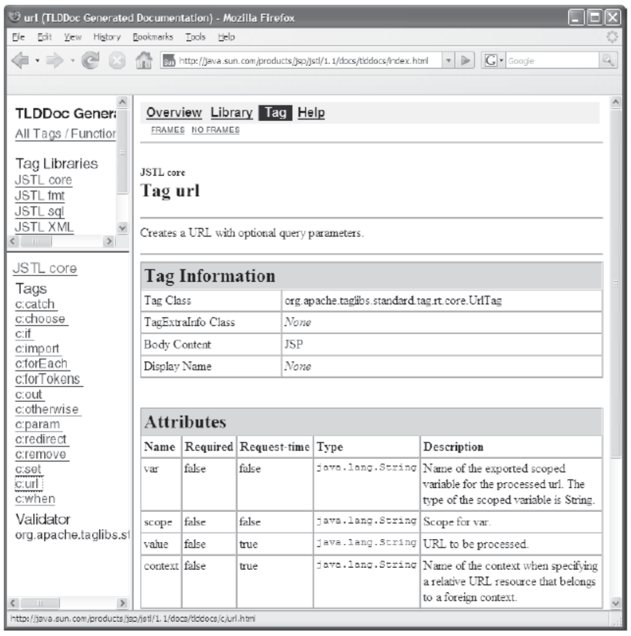
Equivalent script

```
<a href="<%=response.encodeURL("index.jsp")%>">Continue Shopping</a>
```

How to view the documentation for a library

The URL for the JSTL 1.1 documentation

`http://java.sun.com/products/jsp/jstl/1.1/docs/tlddocs/index.html`



url (TLDDoc Generated Documentation) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://java.sun.com/products/jsp/jstl/1.1/docs/tlddocs/index.html
Google

TLDDoc Genera
All Tags / Functior

Tag Libraries
JSTL core
JSTL fmt
JSTL sql
JSTL XML

JSTL core
Tags
c:catch
c:choose
c:if
c:import
c:forEach
c:forTokens
c:out
c:otherwise
c:param
c:redirect
c:remove
c:set
c:url
c:when
Validator
org.apache.taglibs.st

Overview Library Tag Help
FRAMES NO FRAMES

JSTL core
Tag url

Creates a URL with optional query parameters.

Tag Information

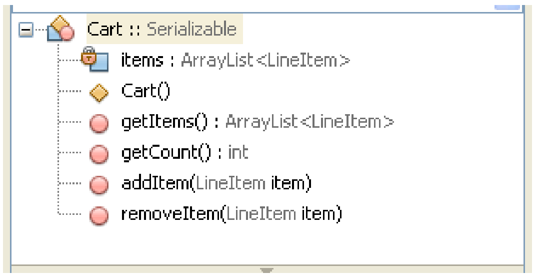
Tag Class	org.apache.taglibs.standard.tag.rt.core.UrlTag
TagExtraInfo Class	*None*
Body Content	JSP
Display Name	*None*

Attributes

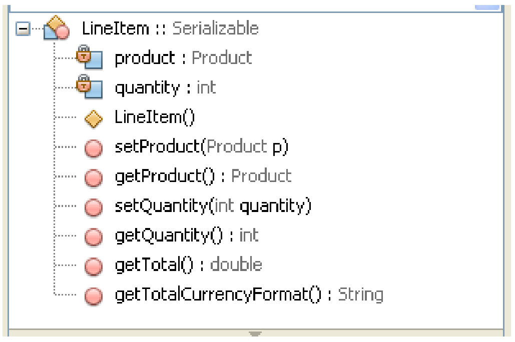
Name	Required	Request-time	Type	Description
var	false	false	`java.lang.String`	Name of the exported scoped variable for the processed url. The type of the scoped variable is String.
scope	false	false	`java.lang.String`	Scope for var.
value	false	true	`java.lang.String`	URL to be processed.
context	false	true	`java.lang.String`	Name of the context when specifying a relative URL resource that belongs to a foreign context.

http://java.sun.com/products/jsp/jstl/1.1/docs/tlddocs/c/url.html

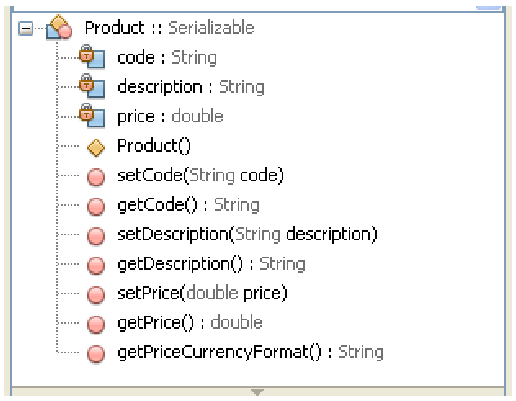
The Cart Application



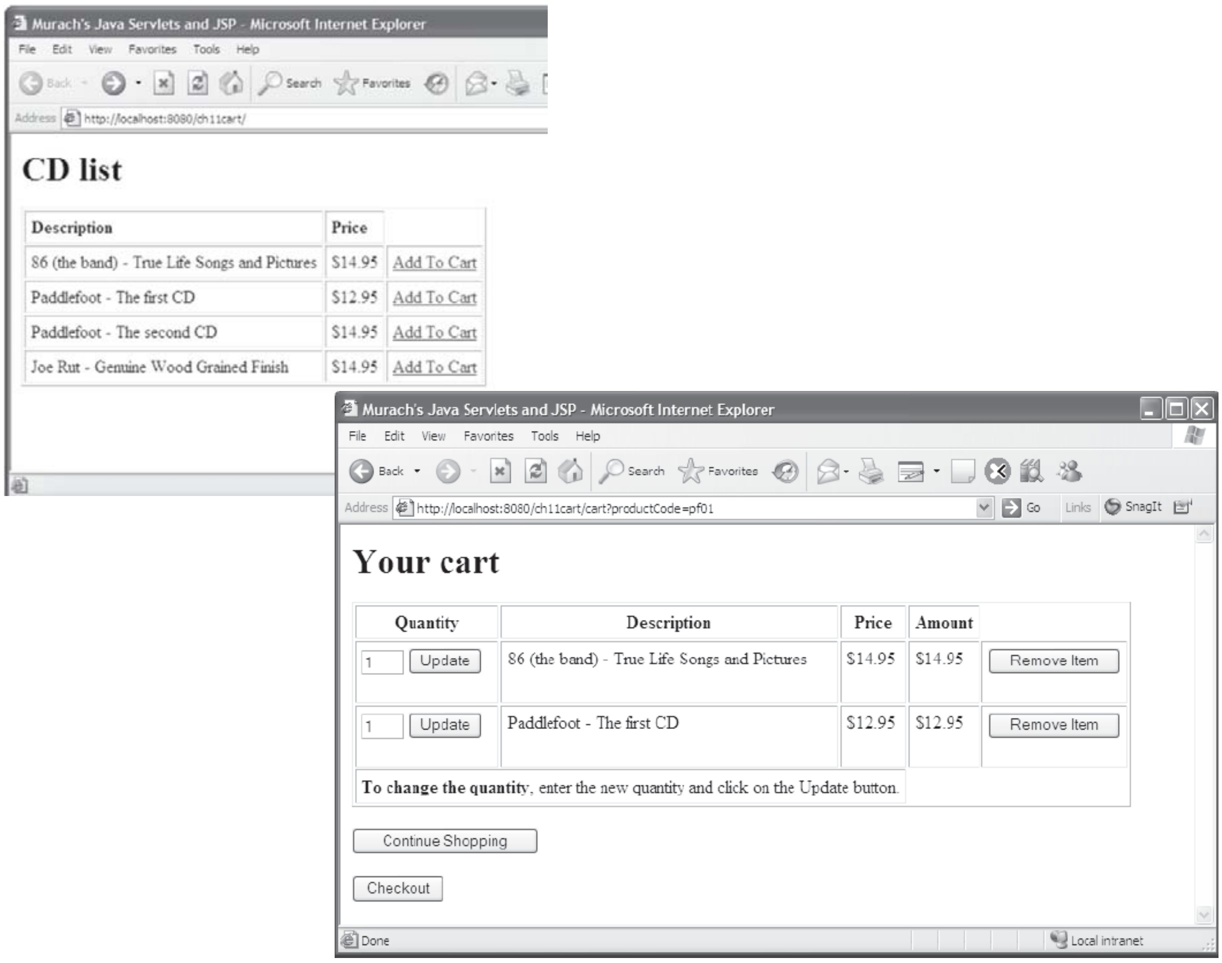
```
Cart :: Serializable
    items : ArrayList<LineItem>
    Cart()
    getItems() : ArrayList<LineItem>
    getCount() : int
    addItem(LineItem item)
    removeItem(LineItem item)
```



```
LineItem :: Serializable
    product : Product
    quantity : int
    LineItem()
    setProduct(Product p)
    getProduct() : Product
    setQuantity(int quantity)
    getQuantity() : int
    getTotal() : double
    getTotalCurrencyFormat() : String
```



```
Product :: Serializable
    code : String
    description : String
    price : double
    Product()
    setCode(String code)
    getCode() : String
    setDescription(String description)
    getDescription() : String
    setPrice(double price)
    getPrice() : double
    getPriceCurrencyFormat() : String
```


Murach's Java Servlets and JSP - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/ch11cart/

CD list

Description	Price	
86 (the band) - True Life Songs and Pictures	$14.95	Add To Cart
Paddlefoot - The first CD	$12.95	Add To Cart
Paddlefoot - The second CD	$14.95	Add To Cart
Joe Rut - Genuine Wood Grained Finish	$14.95	Add To Cart

Murach's Java Servlets and JSP - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/ch11cart/cart?productCode=pf01
Go Links SnagIt

Your cart

Quantity	Description	Price	Amount	
1 Update	86 (the band) - True Life Songs and Pictures	$14.95	$14.95	Remove Item
1 Update	Paddlefoot - The first CD	$12.95	$12.95	Remove Item

To change the quantity, enter the new quantity and click on the Update button.

url tag

JSP code with JSTL

```
<a href="<c:url value='/cart?productCode=8601' />">
   Add To Cart
</a>
```

Equivalent scripting

```
<a href="<%=response.encodeURL("cart?productCode=8601")%>">
    Add To Cart
</a>
```

- You can use the url tag to encode URLs within your web application. This tag will automatically rewrite the URL to include a unique session ID whenever the client doesn't support cookies.
- You can use the JSTL param tag if you want to automatically encode unsafe characters such as spaces with special characters such as plus signs.

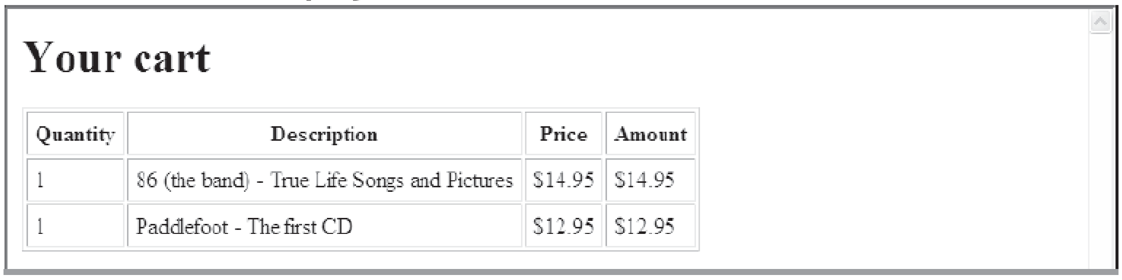
forEach tag

An example that uses JSTL to loop through a collection

JSP code with JSTL

```
<c:forEach var="item" items="${cart.items}">
<tr valign="top">
  <td>${item.quantity}</td>
  <td>${item.product.description}</td>
  <td>${item.product.priceCurrencyFormat}</td>
  <td>${item.totalCurrencyFormat}</td>
</tr>
</c:forEach>
```

The result that's displayed in the browser for a cart that has two items



Your cart

Quantity	Description	Price	Amount
1	86 (the band) - True Life Songs and Pictures	$14.95	$14.95
1	Paddlefoot - The first CD	$12.95	$12.95

Equivalent scripting

```
<%@ page import="business.*, java.util.ArrayList" %>
<%
  Cart cart = (Cart) session.getAttribute("cart");
  ArrayList<LineItem> items = cart.getItems();
  for (LineItem item : items)
  {
%>
  <tr valign="top">
    <td><%=item.getQuantity()%></td>
    <td><%=item.getProduct().getDescription()%></td>
    <td><%=item.getProduct().getPriceCurrencyFormat()%></td>
    <td><%=item.getTotalCurrencyFormat()%></td>
  </tr>
<% } %>
```

- You can use the forEach tag to loop through most types of collections, including arrays.
- You can use the var attribute to specify the variable name that will be used to access each item within the collection.
- You can use the items attribute to specify the collection that stores the data.
- If necessary, you can nest one forEach tag within another.

forTokens tag

An example that uses JSTL to loop through a comma-delimited string

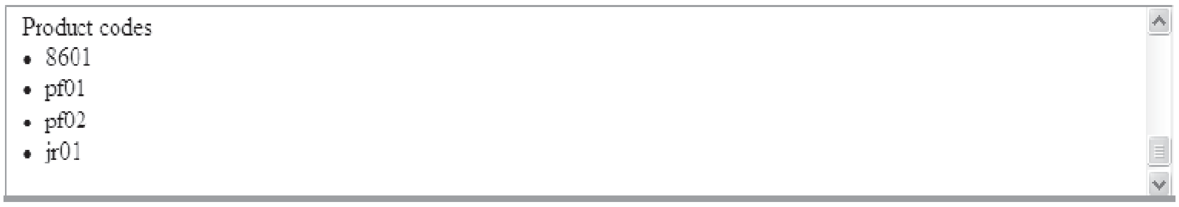
Servlet code

```
session.setAttribute("productCodes", "8601,pf01,pf02,jr01");
```

JSP code

```
<p>Product codes<br>
<c:forTokens var="productCode" items="${productCodes}" delims="," >
    <li>${productCode}</li>
</c:forTokens>
</p>
```

The result that's displayed in the browser



```
Product codes
• 8601
• pf01
• pf02
• jr01
```

An example that uses JSTL to parse a string

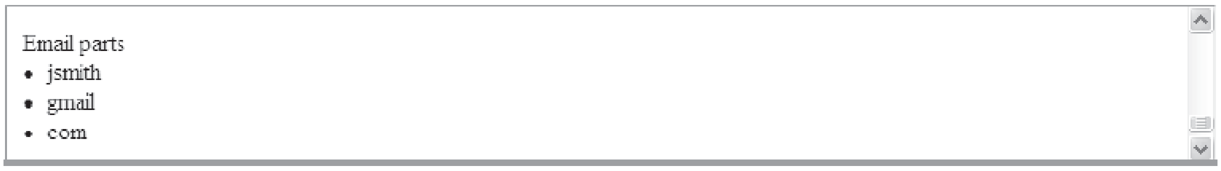
Servlet code

```
session.setAttribute("emailAddress", "jsmith@gmail.com");
```

JSP code

```
<p>Email parts<br>
<c:forTokens var="part" items="${emailAddress}" delims="@.">
    <li>${part}</li>
</c:forTokens>
</p>
```

The result that's displayed in the browser



```
Email parts
• jsmith
• gmail
• com
```

- You can use the forTokens tag to loop through delimited values that are stored in a string.
- You can use the var attribute to specify the variable name that will be used to access each delimited string.
- You can use the items attribute to specify the string that stores the data.
- You can use the delims attribute to specify the character or characters that are used as the delimiters for the string.
- If necessary, you can nest one forTokens tag within another.

Four more attributes for looping

Attributes that you can use for advanced loops

Attribute	Description
begin	Specifies the first index for the loop.
end	Specifies the last index for the loop.
step	Specifies the amount to increment the index each time through the loop.
varStatus	Specifies the name of a variable that can be used to get information about the status of the loop. This variable provides the first, last, index, and count properties.

An example that uses all four attributes

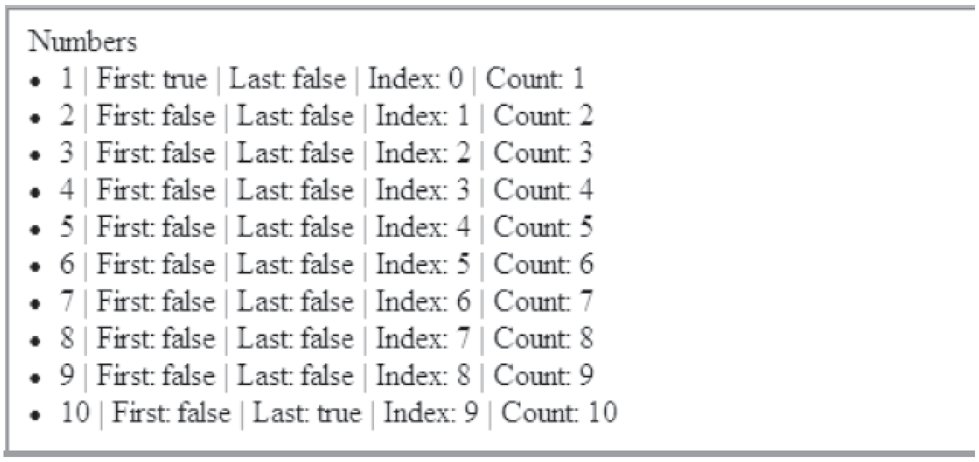
Servlet code

```
int[] numbers = new int[30];
for (int i = 0; i < 30; i++)
{
    numbers[i] = i+1;
}
session.setAttribute("numbers", numbers);
```

JSP code

```
<p>Numbers<br>
<c:forEach items="${numbers}" var="number"
           begin="0" end="9" step="1"
           varStatus="status">
    <li>${number} | First: ${status.first} | Last: ${status.last} |
        Index: ${status.index} | Count: ${status.count} </li>
</c:forEach>
</p>
```

The result that's displayed in the browser



```
Numbers
• 1 | First: true | Last: false | Index: 0 | Count: 1
• 2 | First: false | Last: false | Index: 1 | Count: 2
• 3 | First: false | Last: false | Index: 2 | Count: 3
• 4 | First: false | Last: false | Index: 3 | Count: 4
• 5 | First: false | Last: false | Index: 4 | Count: 5
• 6 | First: false | Last: false | Index: 5 | Count: 6
• 7 | First: false | Last: false | Index: 6 | Count: 7
• 8 | First: false | Last: false | Index: 7 | Count: 8
• 9 | First: false | Last: false | Index: 8 | Count: 9
• 10 | First: false | Last: true | Index: 9 | Count: 10
```

- The begin, end, step, and varStatus attributes work for both the forEach and forTokens tags.

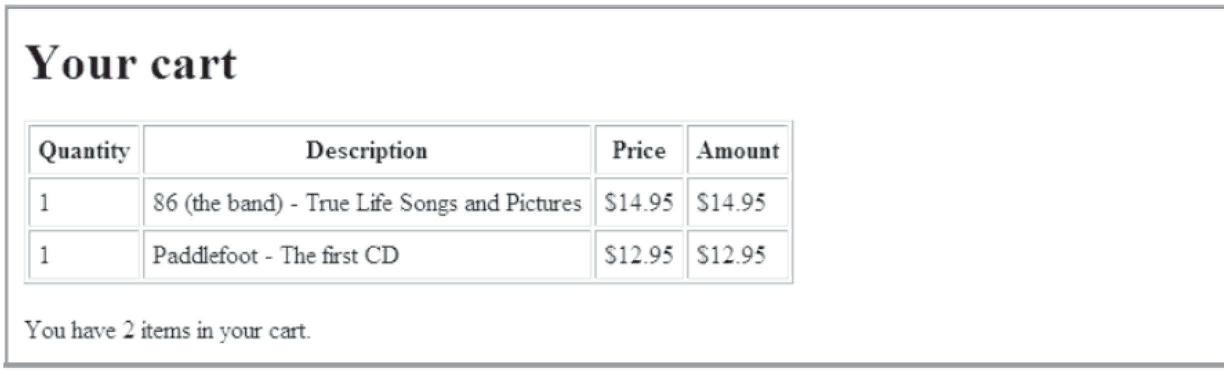
if tag

An example that uses JSTL to code an if statement

JSP code with JSTL

```
<c:if test="${cart.count == 1}">
    <p>You have 1 item in your cart.</p>
</c:if>
<c:if test="${cart.count > 1}">
    <p>You have ${cart.count} items in your cart.</p>
</c:if>
```

The result that's displayed in the browser for a cart that has two items



Your cart

Quantity	Description	Price	Amount
1	86 (the band) - True Life Songs and Pictures	$14.95	$14.95
1	Paddlefoot - The first CD	$12.95	$12.95

You have 2 items in your cart.

Equivalent scripting

```
<%@ page import="business.Cart, java.util.ArrayList" %>
<%
  Cart cart = (Cart) session.getAttribute("cart");
  if (cart.getCount() == 1)
    out.println("<p>You have 1 item in your cart.</p>");
  if (cart.getCount() > 1)
    out.println("<p>You have " + cart.getCount() +
                "  items in your cart.</p>");
%>
```

- You can use the if tag to perform conditional processing that's similar to an if statement in Java.
- You can use the test attribute to specify the Boolean condition for the if statement.
- If necessary, you can nest one if tag within another.

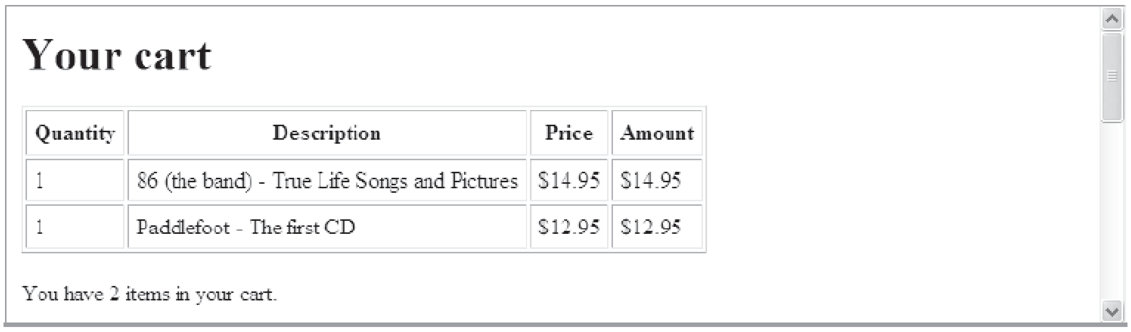
choose tag

An example that uses JSTL to code an if/else statement

JSP code with JSTL

```
<c:choose>
    <c:when test="${cart.count == 0}">
        <p>Your cart is empty.</p>
    </c:when>
    <c:when test="${cart.count == 1}">
        <p>You have 1 item in your cart.</p>
    </c:when>
    <c:otherwise>
        <p>You have ${cart.count} items in your cart.</p>
    </c:otherwise>
</c:choose>
```

The result that's displayed in the browser for a cart that has two items



Your cart

Quantity	Description	Price	Amount
1	86 (the band) - True Life Songs and Pictures	$14.95	$14.95
1	Paddlefoot - The first CD	$12.95	$12.95

You have 2 items in your cart.

Equivalent scripting

```
<%@ page import="business.Cart, java.util.ArrayList" %>
<%
  Cart cart = (Cart) session.getAttribute("cart");
  if (cart.getCount() == 0)
    out.println("<p>Your cart is empty.</p>");
  else if (cart.getCount() == 1)
    out.println("<p>You have 1 item in your cart.</p>");
  else
    out.println("<p>You have " + cart.getCount() +
                " items in your cart.</p>");
%>
```

- You can use the choose tag to perform conditional processing similar to an if/else statement in Java. To do that, you can code multiple when tags and a single otherwise tag within the choose tag.
- You can use the test attribute to specify the Boolean condition for a when tag.
- If necessary, you can nest one choose tag within another.

import tag

An example that imports a header file

JSP code with JSTL

```
<c:import url="/includes/header.html" />
```

Equivalent standard JSP tag

```
<jsp:include page="/includes/header.html" />
```

An example that imports a footer file

JSP code with JSTL

```
<c:import url="/includes/footer.jsp" />
```

Equivalent standard JSP tag

```
<jsp:include page="/includes/footer.jsp" />
```

An example that imports a file from another application

```
<c:import url="http://localhost:8080/musicStore/includes/footer.jsp" />
```

An example that imports a file from another web server

```
<c:import url="www.murach.com/includes/footer.jsp" />
```

- The import tag includes the file at *runtime*, not at compile-time, much like the standard JSP include tag described in chapter 7.
- One advantage of the import tag over the standard JSP include tag is that it lets you include files from other applications and web servers.

Other tags in JSTL core library

Tag name	Description
out	Uses EL to display a value, automatically handling most special characters such as the left angle bracket (<) and right angle bracket (>).
set	Sets the value of an attribute in a scope.
remove	Removes an attribute from a scope.
catch	Catches any exception that occurs in its body and optionally creates an EL variable that refers to the Throwable object for the exception.
redirect	Redirects the client browser to a new URL.
param	Adds a parameter to the parent tag.

If you use the MVC pattern, you probably won’t need to use these tags

An out tag that displays a message

Using the Value attribute

```
<c:out value="${message}" default="No message" />
```

Using the tag's body

```
<c:out value="${message}">
    No message
</c:out>
```

A set tag that sets a value in an attribute

```
<c:set var="message" scope="session" value="Test message" />
```

A set tag that sets a value in a JavaBean

JSP code with JSTL

```
<c:set target="${user}" property="firstName" value="John" />
```

Equivalent standard JSP tag

```
<jsp:setProperty name="user" property="firstName" value="John"/>
```

A remove tag that removes an attribute

```
<c:remove var="message" scope="session" />
```

A catch tag that catches an exception

```
<c:catch var="e">
    <%  // this scriptlet statement will throw an exception
        int i = 1/0;
    %>
    <p>Result: <%= i %></p>
</c:catch>
<c:if test="${e != null}">
    <p>An exception occurred. Message: ${e.message}</p>
</c:if>
```

A redirect tag that redirects to another page

```
<c:if test="${e != null}">
    <c:redirect url="/error_java.jsp" />
</c:if>
```

The Cart Application

Page 358