NIIT

Objectives

- In this session, you will learn to:
  - Work with Java Standard Tag Library (JSTL)
  - Work with JSTL and Expression Language (EL)

NIIT

Identifying the Tag Library

- JSP was introduced with a vision of segregating the UI-related code and the business logic-related code in Web applications.
- However, the use of scriptlets in a JSP page does not fulfill the preceding requirement.
- In addition, the use of scriptlets leads to readability and maintenance issues.
- Therefore, JSTL was introduced to solve these problems.
- JSTL is a tag library that consists of predefined tags to perform common tasks in JSP pages, such as:
  - iterating over a list of items.
  - formatting the page output according to a specific condition.

NIIT

Identifying the Tag Library (Contd.)

- JSTL contains several types of tags.
- The tags are classified into the following categories according to their functions:
  - Core tags
  - Formatting tags
  - Database tags
- The core tags are used to perform the following tasks:
  - Implementing flow control
  - Iterating through a list
  - Implementing URL rewriting
  - Redirecting users to a new URL
  - Creating a URL with query parameters

NIIT

Identifying the Tag Library (Contd.)

- Some of the core tags include:
  - `<out>`
  - `<set>`
  - `<if>`
  - `<choose>`
  - `<when>`
  - `<otherwise>`
  - `<import>`
  - `<forEach>`
  - `<url>`
  - `<redirect>`

NIIT

Identifying the Tag Library (Contd.)

- The following code snippet shows the use of a core tag in a JSP page:

```
<%@taglib prefix="c"
   uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
<title>Core Tags</title>
</head>
<body>
     <c:out value="Message shown using the <out>
   tag"/>
</body>
</html>
```

Specifies the location of the JSTL core tag library.

NIIT

Core Tags

- <c:if test =“testcondition” var=“varName” scope=“page | request | session | application”>

  Body content

  </c:if>

- <c:choose>

  <c:when test=“testCondition”>

  Body Content

  </c:when>

  <c:otherwise>

  Body Content

NIIT

Core Tags

- <c:forEach var="varName" item="collection" begin="begin" end="end" step="step">
  Body Content
- </c:forEach>

- <c:forTokens items="stringofToken" delims="delimiters" var="varName" >
  Body Content
- </c:forTokens>

NIIT

Identifying the Tag Library (Contd.)

- Formatting tags are used to format and display text, numbers, date, or time in a specific format.
- Some of the formatting tags include:
  - `<formatNumber>`
  - `<parseNumber>`
  - `<formatDate>`
  - `<timeZone>`
  - `<setLocale>`

NIIT

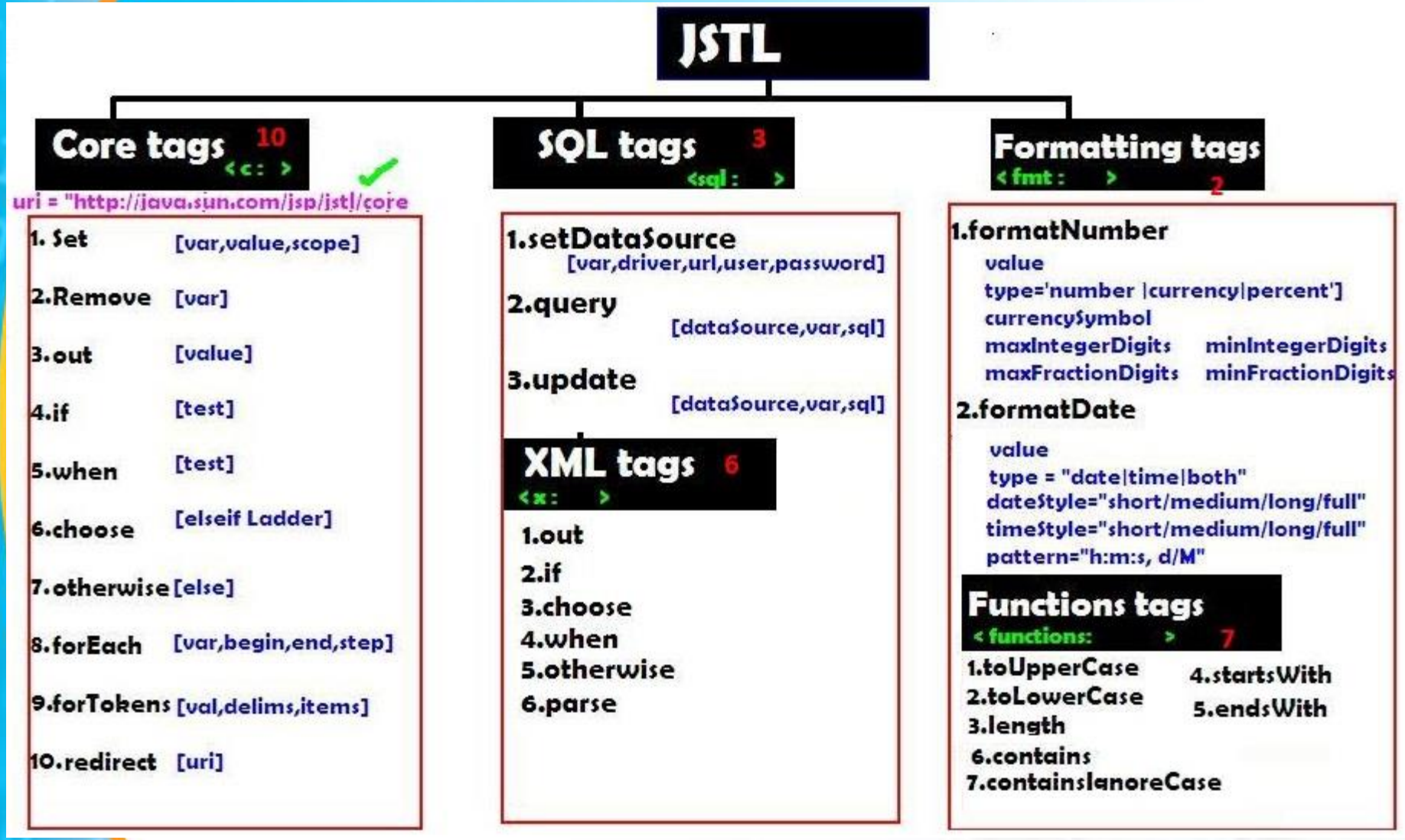
Identifying the Tag Library (Contd.)

- The following code snippet shows the use of a formatting tag in a JSP page:

```
<%@taglib prefix="fmt"
   uri="http://java.sun.com/jsp/jstl/fmt" %>
<html>
<head>
<title>Formatting Tags</title>
</head>
<body>
<fmt:formatNumber value="123000.4509"
   type="currency"/>
</body>
</html>
```

Formats the number, `123000.4509`, as `$123,000.45`.

NIIT



NIIT

Database Tags

- Database tags are used to interact with relational databases, such as Microsoft SQL Server and Oracle.
- Some of the database tags include:
  - `<setDataSource>`
  - `<query>`
  - `<param>`
  - `<transaction>`

NIIT

Database Tags (Contd.)

- The following code snippet shows the use of a database tag in a JSP page:

```
<%@taglib prefix="sql"
   uri="http://java.sun.com/jsp/jstl/sql" %>
<html>
<head>
<title>Core Tags</title>
</head>
<body>
<sql:setDataSource var="ds"
   driver="sun.jdbc.odbc.JdbcOdbcDriver”
   url="jdbc:derby://localhost:1527/ProductDetail
   s” user="niit"  password="pass123"/>
```

Specifies the data source.

NIIT

Database Tags (Contd.)

```
<sql:query dataSource="${ds}" var="result">
SELECT * from Products;
</sql:query>
</body>
</html>
```

Specifies the query that needs to be executed against the database.

NIIT

Implementing EL

- EL:
  - is used extensively in JSTL tags.
  - can be used to create dynamic JSP pages.
- EL is a scripting language that allows programmers to use simple expressions to access and manipulate:
  - application data stored in JavaBeans components.
  - implicit objects.
  - Java classes.
  - collections elements.
  - scoped variables.
- EL expressions can be used in the following ways:
  - As attribute values
  - As text in a JSP page

NIIT

Implementing EL (Contd.)

- EL provides implicit objects that can be used to easily access all implicit objects and scoped variables of JSP.
- Some of the EL implicit objects are:
  - `pageContext`
  - `pageScope`
  - `requestScope`
  - `sessionScope`
  - `applicationScope`
  - `param`
  - `paramValues`
  - `header`
  - `headerValues`
  - `initParam`
  - `cookie`

NIIT

Implementing EL (Contd.)

- The following code snippet shows how to retrieve the value of an init parameter using an EL implicit object:

```
My Email is: ${initParam.myEmail}
```

- The following code snippet retrieves the information from a cookie named `yourname` using the `cookie` implicit object of EL:

```
${cookie.yourname.value}
```

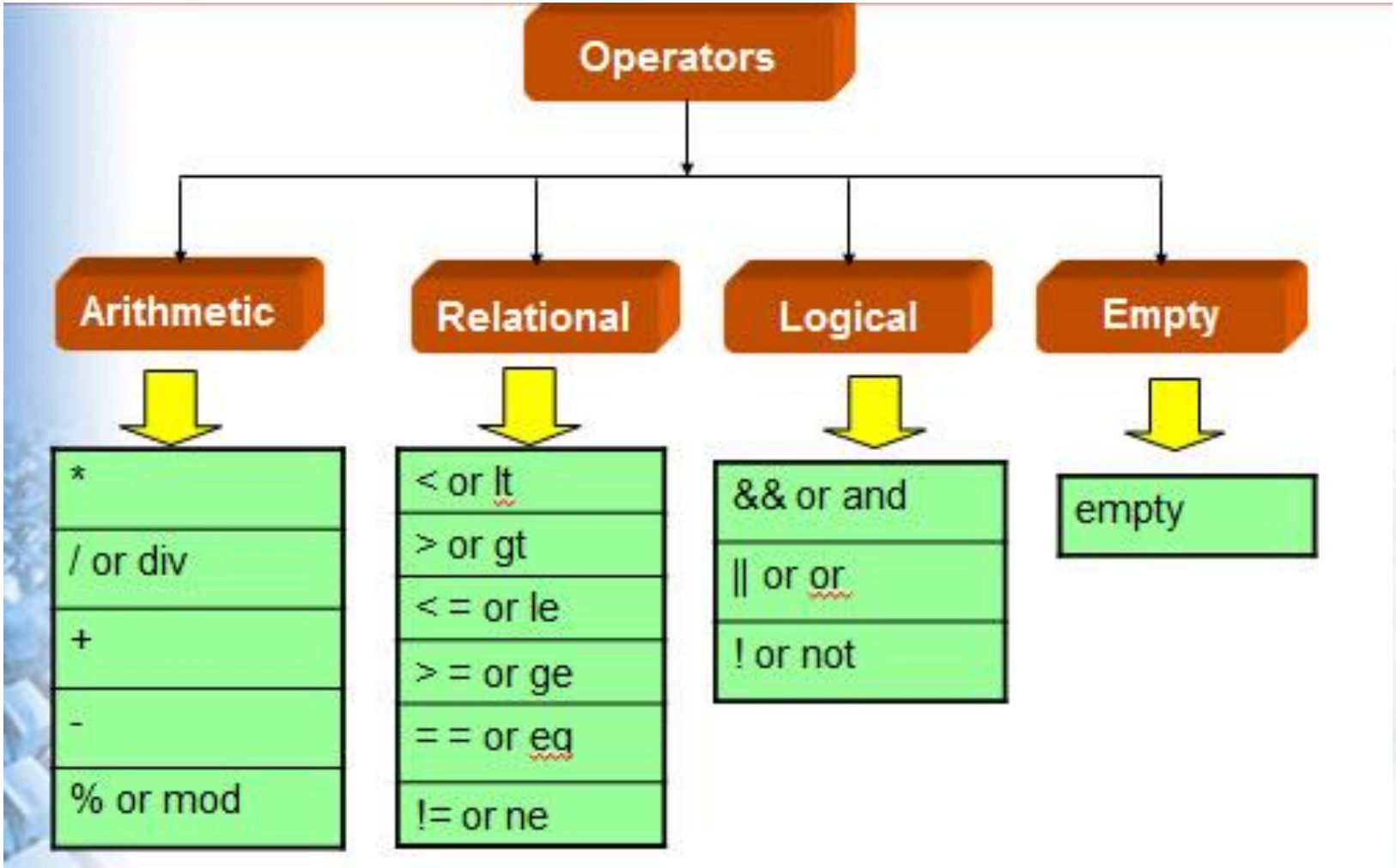
NIIT

Implementing EL

- EL can also be used to access JavaBeans.
- A JavaBean:
  - is a simple Java class that exposes internal fields as properties using the corresponding getter and setter methods.
  - is a reusable and a self-contained software component that takes advantage of all the security and platform-independent features of Java.
  - can be included in a JSP page to start processing user requests.
- The following code snippet sets and retrieves the value of the `firstName` property from a bean named `customer`:

```
<jsp:setProperty name="customer"
property="firstName" value="Sam" />
<h2>${customer.firstName}</h2>
```

NIIT



NIIT

Summary

- In this session, you learned that:
  - JSTL is a tag library that consists of predefined tags to perform common tasks in JSP pages.
  - EL is a scripting language that allows programmers to use simple expressions to access and manipulate the properties of a JSP page.
  - JSTL tags are classified into the following categories according to their functions:
    - Core tags
    - Formatting tags
    - Database tags
  - EL is a scripting language that allows programmers to use simple expressions to access and manipulate the properties of a JSP page.
  - EL expressions can be used in the following ways:
    - As attribute values
    - As text in a JSP page