

# 第十一章

## JSP 标准标签库



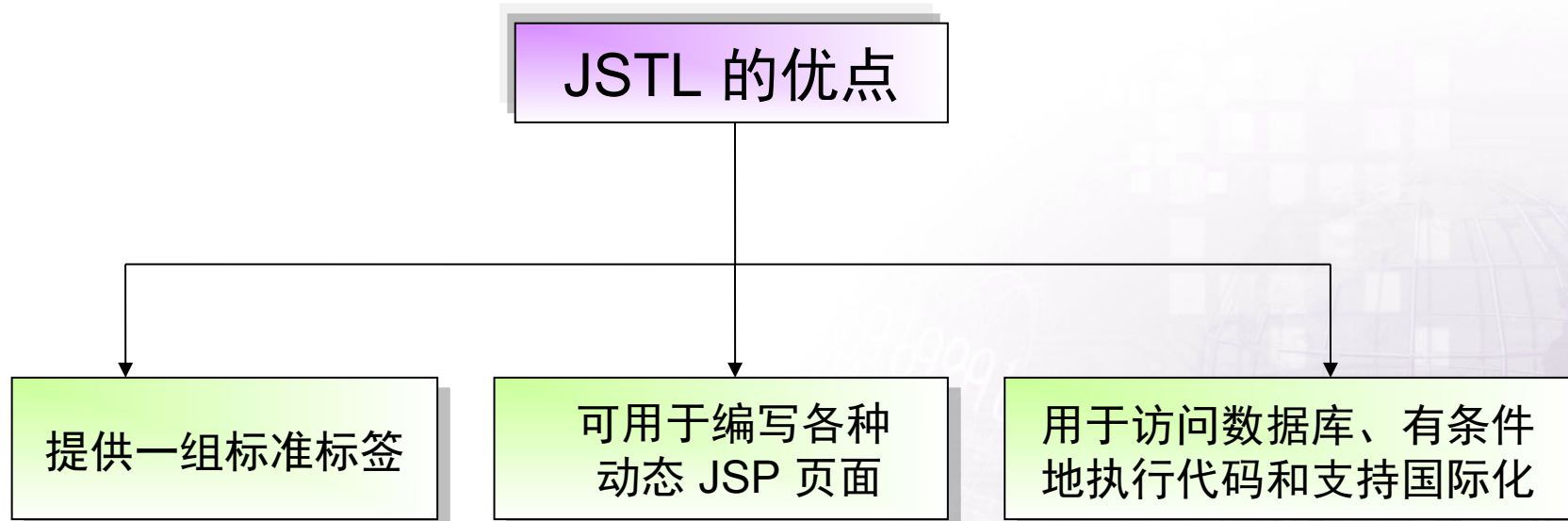
ACCP V4.0

- ❑ 自定义标签是用户定义的 JSP 语言元素
- ❑ 标签处理程序是一个对象，Web 容器调用此对象，以在执行 JSP 页面的过程中解释自定义标签
- ❑ 简单标签的处理程序必须实现 Tag 接口的 doStartTag() 和 doEndTag() 方法
- ❑ 标签库描述符 (TLD, Tag Library Descriptor) 包含库以及该库中所包含的每个标签的信息

# 目标

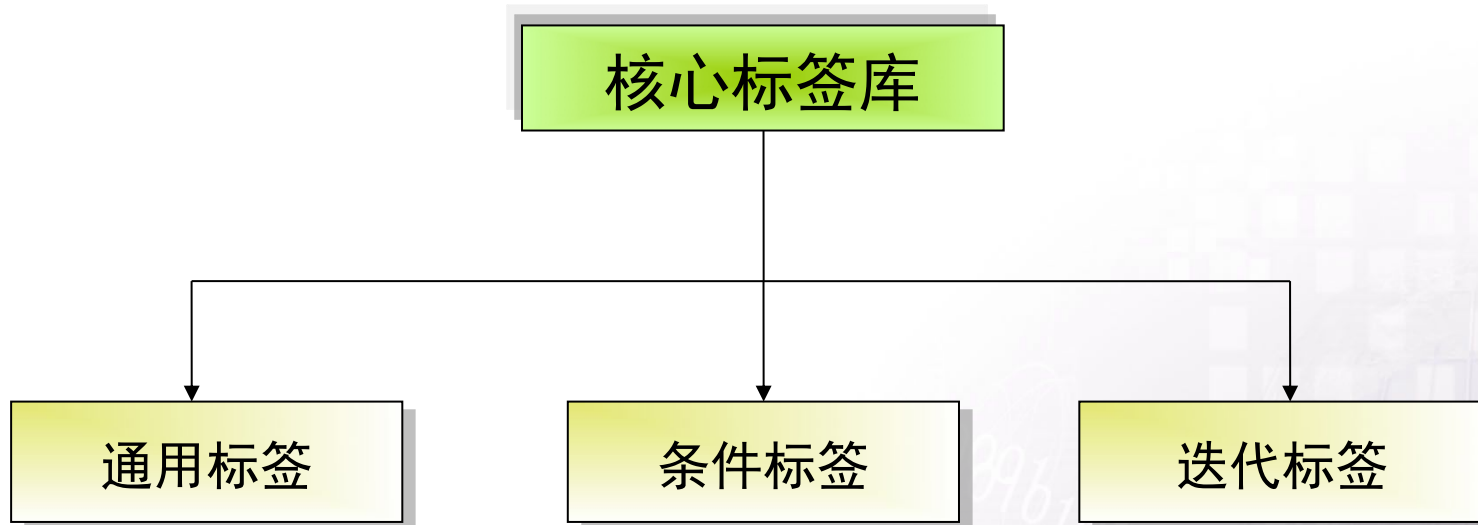
- ❑ 理解核心标签库
- ❑ 理解国际化和格式化标签
- ❑ 理解 SQL 标签库

- Java Server Pages 标准标签库 (JSTL) 包含的各种标签可用于 JSP 页面中



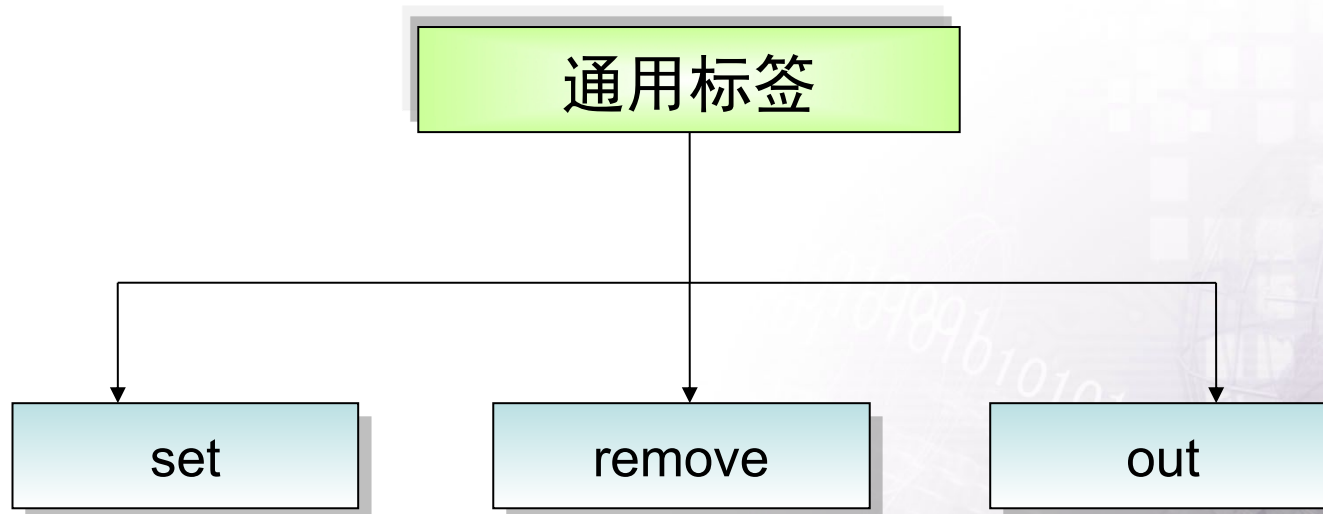
# 核心标签库

□ 核心标签库包含以下标签：



## 通用标签 3-1

- 通用标签库用于操作 JSP 页面内的作用域变量
- 通用标签库中的标签：



## 通用标签 3-2

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

插入使用通用  
标签的指令

```
<body>
```

```
<c:set var="example" value="${100+1}" scope="session" />
```

set 标签用于给变  
量设置值和作用域

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

out 标签显示值的结果

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

```
</body>
```

remove 标签用于删  
除具有作用域的变量

```
<body>
```

```
<H1>欢迎来到 Shop Stop</H1>
```

该 JSP 页面在一个 session 作用域内的变量中存储 sessionvariable, 此 Web 应用程序中的其他 JSP 页面可以访问此变量.

```
<p />
```

```
<c:set var="sessionvariable" value="{80+8}"  
scope="session" />
```

在删除 sessionvariable 之前先显示它的值:

```
<c:out value="{sessionvariable}"/>  
<c:remove var="sessionvariable" scope="session" />
```

```
<br/>
```

显示并删除后的 sessionvariable 的值. <br />

```
<c:out value="{sessionvariable}"> sessionvariable为NULL </c:out>
```

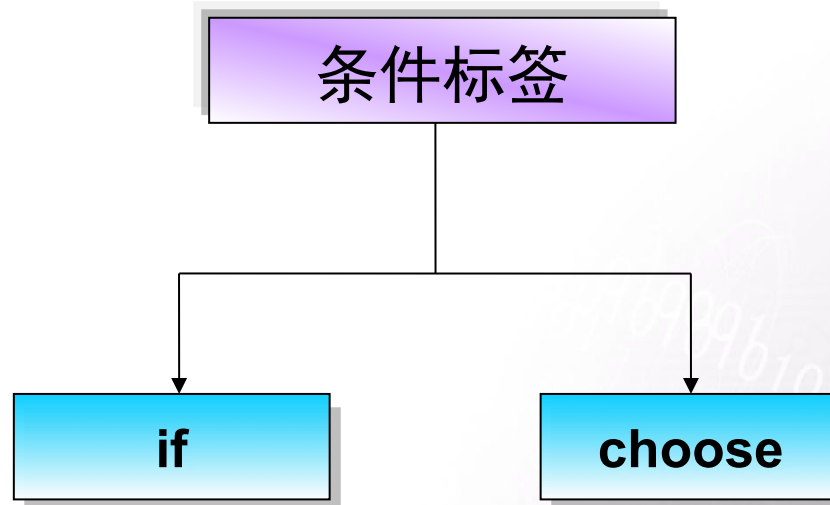
```
</body>
```

演示：示例 2



## 条件标签 2-1

- ❑ JSTL 提供条件标签以支持 JSP 页面中的各种条件
- ❑ 条件标签包括：

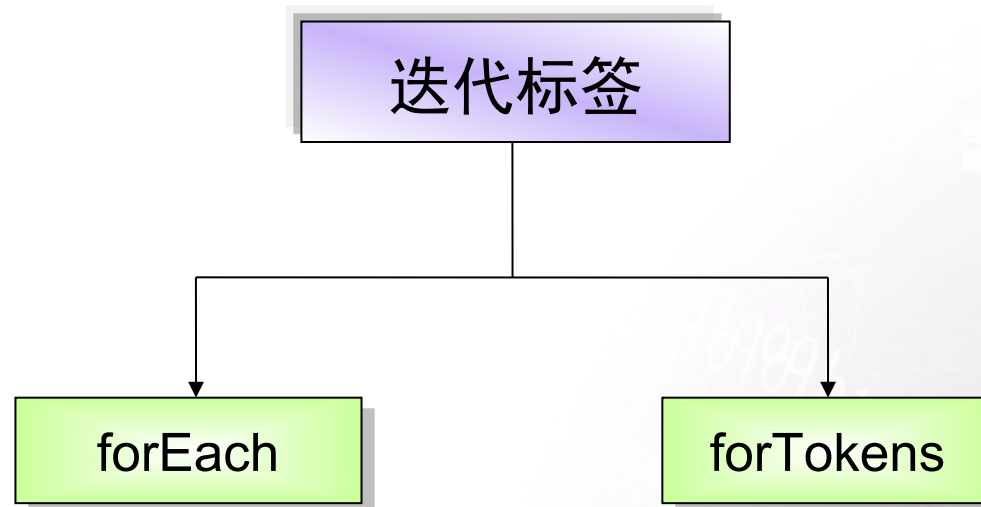


## 条件标签 2-2

```
<body>  
  <c:if test="${pageScope.signalStrength < 5}">  
    <c:set var="signalFailure" value="true"  
      scope="page" />  
  </c:if>  
  <h1> 使用 If 和 Choose</h1>  
  <c:choose>  
    <c:when test="${pageScope.signalFailure == true}">  
      信号断开  
    </c:when>  
    <c:otherwise>  
      信号打开  
    </c:otherwise>  
  </c:choose>  
</body>
```

## 迭代标签库 3-1

- ❑ 迭代标签用于多次计算标签体
- ❑ 迭代标签库中的标签有：



## 迭代标签库 3-2

```
<%@ taglib uri="http://java.sun.com/jstl/core_rt" prefix="c" %>
<%! String [] numbers = { "1", "2", "3", "4" }; %>
<c:set var="member" value="Joe:Petter;Ryan|John" scope="page" />
<BODY>
<c:forEach var="numbering" items="<%= numbers %>">
<c:out value="\${numbering}" />
</c:forEach>
```

forEach 标签将遍历对象集合并显示 1 到 4 之间的数字

```
<c:forTokens items="\${pageScope.member}" delims="·|" var="membername">
<c:out value="\${membername}" /> <br />
</c:forTokens>
```

forTokens 将遍历用所提供的分隔符分隔的值并显示所有的名称

```
</BODY>
```

## □ 在 JSP 页面中使用各种迭代标签

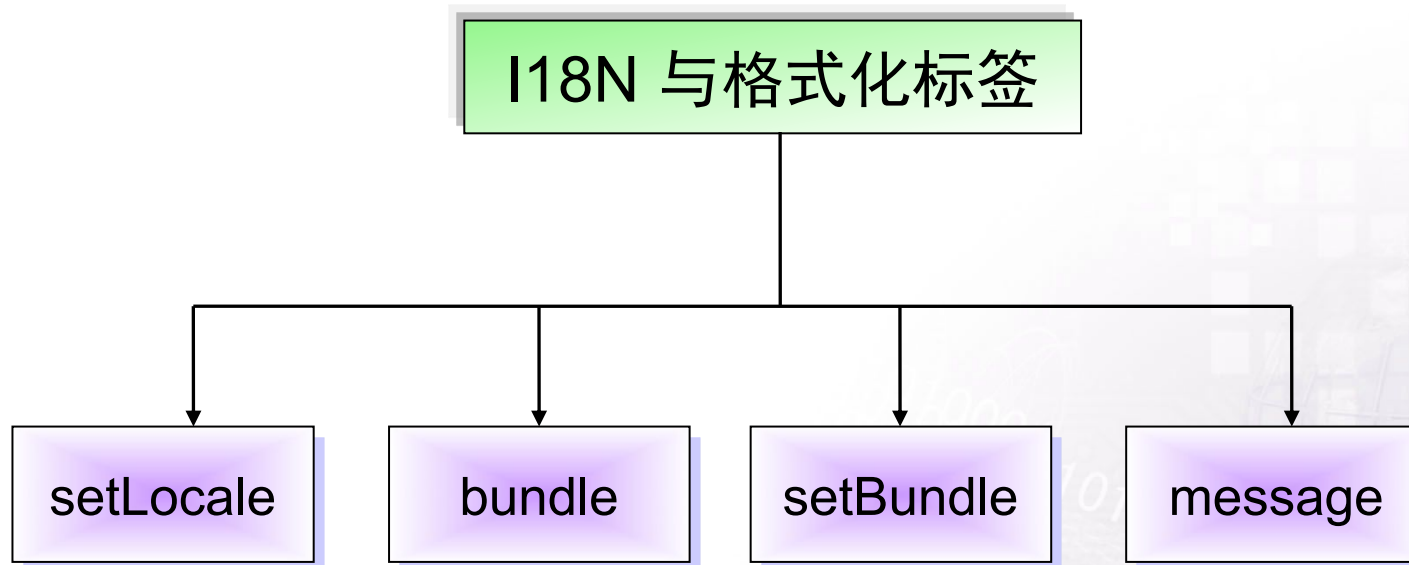
```
<body >
  <H1>显示产品与公司名称</H1>
  <c:forEach var="company" items="<%= names %>">
    ${company} <br/>
  </c:forEach>
  <br/>
  <c:forTokens items="${pageScope.language}"
    delims=":;|" var="currentName" varStatus="status">
    产品编号为 #P000<c:out value="${status.count}" />
    是<c:out value="${currentName}" /> <br />
  </c:forTokens>
</body >
```

# I18N与格式化标签库 3-1

- I18N 与格式化标签库可用于创建国际化的 Web 应用程序
- 它也可用于对经过格式化的数字和日期-时间的输出结果进行标准化
- I18N 应用程序的特点是：
  - 可以在全世界运行
  - 支持多语言
  - 可以快速本地化

## I18N与格式化标签库 3-2

□ 国际化 (I18N) 与格式化标签库中的标签有：





# I18N与格式化标签库 3-3

```
<body>
  <H2>数字格式示例</H2>
  将 <B>123.4</B> 格式化为:
  <fmt:formatNumber value="123.4" type="number"
  minFractionDigits="3" /><BR>
  <HR>
  <H2>货币格式示例</H2>
  <c:set var="salary" value="125000" />
  工资: <c:out value="\${salary}"/><BR>
  <fmt:setLocale value="en_GB"/>
  用本机的 <B>en_GB</B> 将工资格式化为:
  <fmt:formatNumber type="currency" value="\${salary}"
  /><BR>
  <fmt:setLocale value="zh_CN"/>
  用本机的 <B>zh_CN</B> 将工资格式化为:
  <fmt:formatNumber type="currency" value="\${salary}"
  /><BR>
  <HR>
</body>
```



# SetBundle 示例

```
<c:if test="${param.language == 'en' || param.language == null}">
  <fmt:setLocale value="en" scope="session" />
</c:if>
<c:if test="${param.language == 'sv'}">
  <fmt:setLocale value="sv" scope="session" />
</c:if>
<c:if test="${param.language == 'de'}">
  <fmt:setLocale value="de" scope="session" />
</c:if>
<fmt:setBundle basename="labels" scope="session" />
```

...

title=Industry Trends

title=Industri Trender  
select\_language=Välj språk  
new\_language=Nytt Språk  
english=Engelska  
swedish=Svenska  
german=Tyska  
submit=Rösta

```
<fmt:message key="german" /><br>
```

```
<p>
```

```
<input type="submit" value="<fmt:message key="new_language" />">
```

```
</form>
```

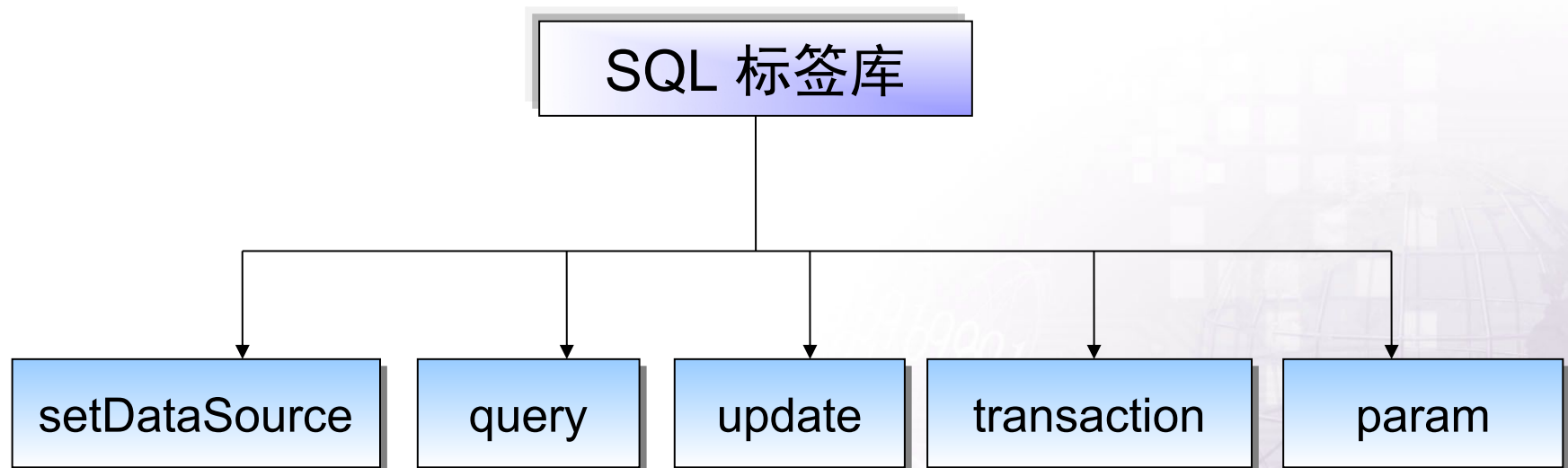
```
</body>
```

# SQL 标签库 3-1

- ❑ SQL 标签库用于从 JSP 页面访问和更新数据库。
- ❑ SQL 标签库具有以下功能：
  - ❑ 传递各种数据库查询
  - ❑ 访问查询结果
  - ❑ 数据库修改
  - ❑ 执行各种数据库事务

# SQL 标签库 3-2

□ SQL 标签库中的标签有：



```
<body>
<h1> 使用 SQL 标签库</h1>
<c:set var = "valprice" value = "1000"/>
<sql:setDataSource
driver = "com.microsoft.jdbc.sqlserver.SQLServerDriver" url
=
"jdbc:microsoft:sqlserver://localhost:1433;DataBaseName=sho
pstop;" user="sa" password="" var="conn"/>
<sql:transaction dataSource="${conn}">
<sql:update var="newTable">
CREATE TABLE ProductDetails
(
ProductId int IDENTITY (1000, 1) NOT NULL primary key,
ProductName varchar (20) NOT NULL ,
ProductType varchar (15) NOT NULL ,
Price varchar (5) NOT NULL ,
Brand varchar (25) NOT NULL ,
Description varchar (50) NOT NULL
)
</sql:update>
</sql:transaction>
<sql:update var="newrow" dataSource="${conn}">
INSERT INTO ProductDetailss(ProductName, ProductType,
Price, Brand, Description
VALUES('牛仔裤 ', '衣服', '1000', 'Lee', '品质良好的牛仔裤')
</sql:update>
<sql:query var="products" dataSource="${conn}">
select * from ProductDetailss
</sql:query>
```

- ❑ JSTL 标签用于访问数据库、操作、设置和删除作用域变量和国际化标签
- ❑ 核心标签库是最常用的标签库
- ❑ 通用标签用于设置、删除和显示表达式的输出结果
- ❑ 条件标签用于有条件地执行标签
- ❑ if 和 choose 标签用于有条件地执行标签
- ❑ 迭代标签用于多次执行标签体
- ❑ forEach 和 forEachTokens 是迭代标签
- ❑ I18N 与格式化标签库可用于创建国际化和标准化的 Web 应用程序
- ❑ SQL 标签用于访问和更新数据库
- ❑ SQL 标签适用于基于 Web 的小型应用程序