**标准标签库**

# 授课内容

授课科目：JavaEE

授课内容：标准标签库

授课类型：讲授+实践

授课时间：3+1学时

主讲教师：武永亮

# 教学目标要求

## 思想目标：

## 能力目标：

1. 希望学生了解JSTL的应用场景；
2. 培养学生掌握常见JSTL库的使用。

## 知识目标：

1. JSTL标签库安装
2. 一般用途的JSTL标签
3. 属性用途的JSTL标签
4. 控制用途的JSTL标签

# 课件分析

概 述：JSTL全称JSP标准标签库(JSP Standard Tag Library),其封装了Web应用程序中常见的通用功能标签。JSTL由以下几个子标签库组成：核心标签、XML标签、格式化标签(I18N)、SQL标签、函数标签库。通过使用JSTL标签库可以简化Java数据在前端的展示方式。

教学重点：JSTL标签库基本用法。

教学难点：JSTL标签库基本用法。

# 教学方法

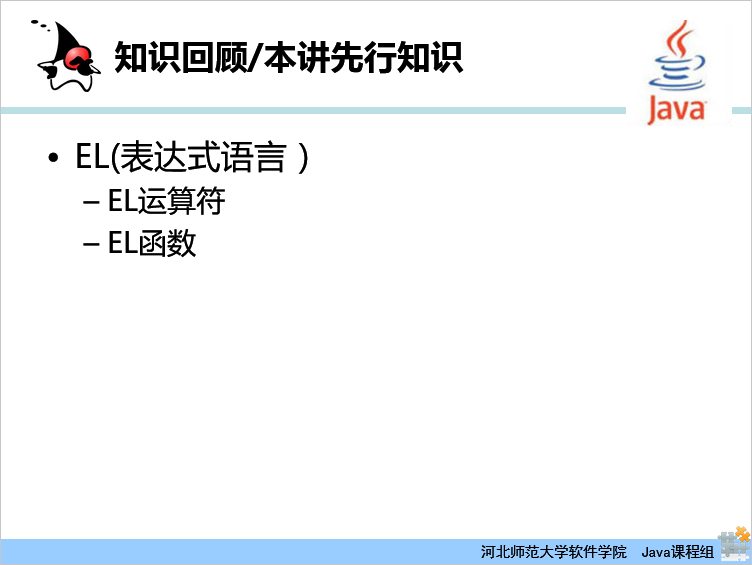
直观演示法、案例法等。

# 教学过程

## 课前补充

## 上节回顾

（4分钟）



上节课讲到在JSP页面中为了简化数据的显示形式，可以引入EL表达式进行页面数据展示。EL表达式可以替代JSP的显示标签显示数据，也可以进行一定的算数运算、逻辑运算。对于一些数据展示的复杂要求，也可以通过自定义EL函数来实现。

定义EL函数的步骤是：

1、编写方法所在源文件 .java文件，该文件包含EL函数的实现。

2、编写标签库描述符文件 .tld文件，定义EL函数名。

3、修改部署描述符文件 web.xml文件， 定义标签库描述符地址。

4、编写JSP文件 .jsp文件，通过EL函数的库引入使用EL函数。

## 作业点评

（0分钟）

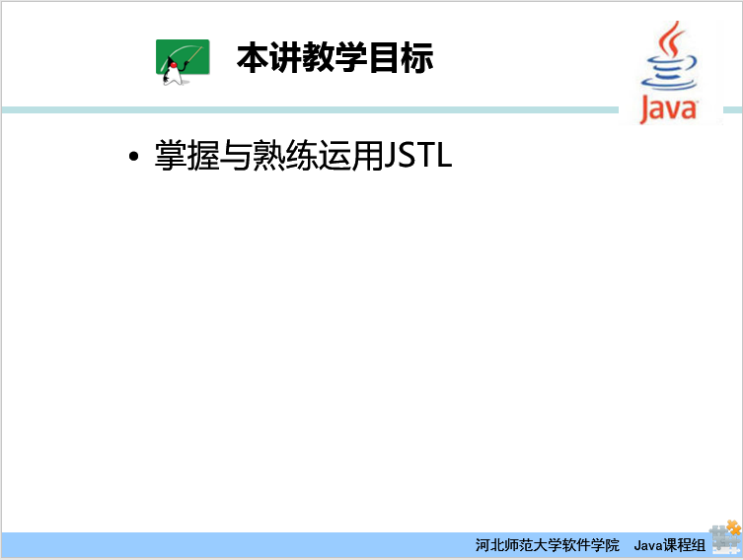
## 导入新课

（1分钟）

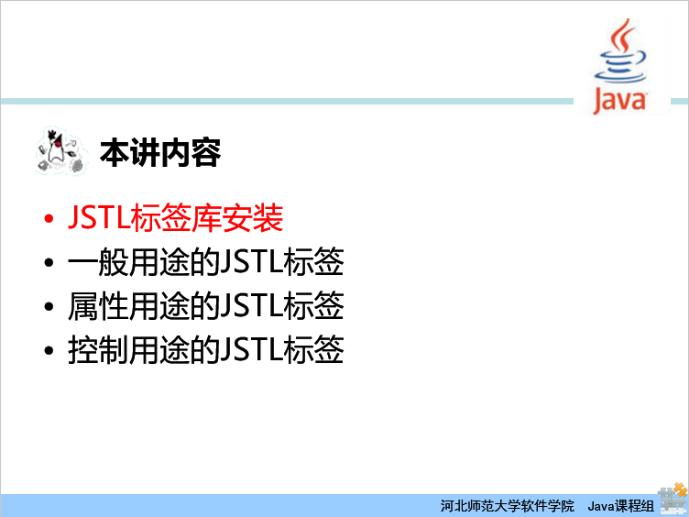
在页面中可以通过EL函数来实现所有的页面显示逻辑，但是很多都依赖EL函数实现，EL函数的实现较复杂，所以在JSP中已经将常用的EL函数进行封装，形成JSTL，帮助我们更快速的实现前台页面的显示逻辑。本章主要内容是介绍JSTL的用法。

## 讲授新课

（120分钟）

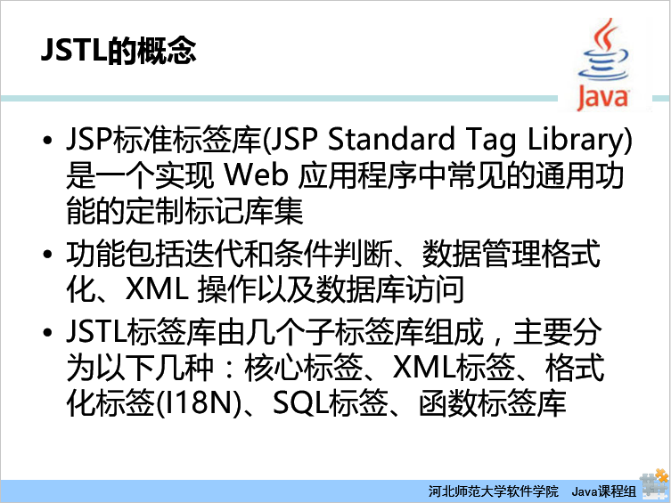


本章的教学目标：要求学生掌握JSTL的用法，熟练掌握如何引入JSTL相关库。掌握核心标签、XML标签、格式化标签(I18N)、SQL标签、函数标签库的基本用法。



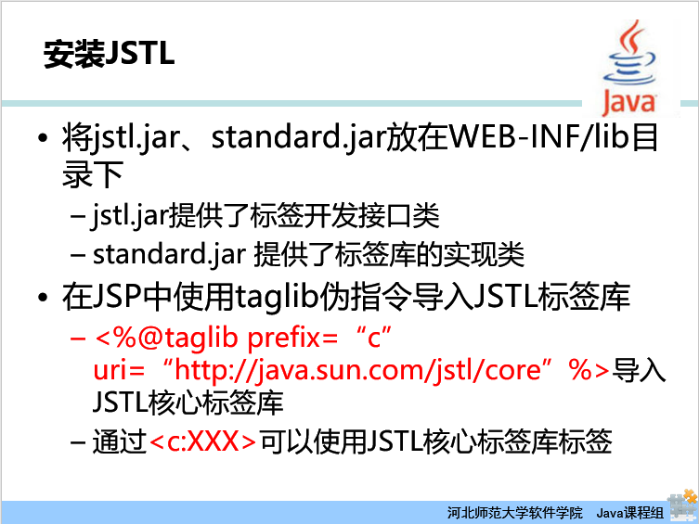
本章内容分为四个部分，第一部分介绍JSTL标签库的安装过程及核心标签库的用法，第二部分介绍JSTL中一般标签库的使用方法，第三部分介绍属性相关的JSTL标签的用法，第四部分介绍与逻辑控制相关的JSTL标签的用法。

首先介绍第一部分内容—JSTL标签库安装。



JSP标准标签库(JSP Standard Tag Library)是实现了Web 应用程序中通用功能的定制标签库集。它使用EL函数的形式封装了JSP中通用的功能。主要功能包括迭代和条件判断、数据管理格式化、XML操作、数据库访问等。JSTL标签库由几个子标签库组成，主要分为以下几种：核心标签、XML标签、格式化标签(I18N)、SQL标签、函数标签库。

下面介绍一下JSTL的安装过程。



首先将jstl.jar、standard.jar放在WEB-INF/lib目录下，并加入buildpath。其中jstl.jar主要提供标签开发接口类；standard.jar 提供了标签库的实现类。

在需要使用JSTL的JSP页面使用taglib伪指令导入JSTL标签库。taglib指令是定义一个标签库以及其自定义标签的前缀。通常引入方法为：

|  |
| --- |
| <%@ taglib uri="" prefix="c"%> |

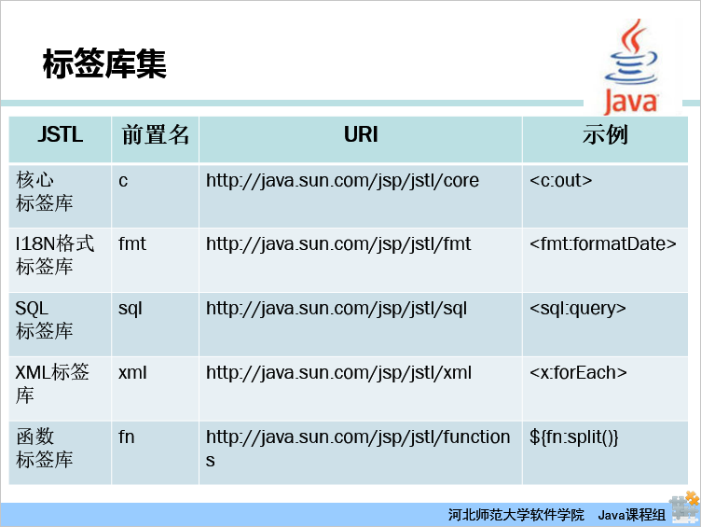
其中prefix 是一个标签库别名，也叫做前缀名，在JSP中引入taglib后，可以使用<前缀名:标签名>来一使用对应标签库中的标签。

例如，导入核心标签库的代码如下：

|  |
| --- |
| <%@taglib prefix=“c” uri=“http://java.sun.com/jstl/core”%> |

导入了核心标签库后，就可以通过<c:XXX>访问核心标签库的标签，这些标签本质就是定义在standard.jar中的EL函数。

JSTL中的标签库集的前置名和URI地址如下：



核心标签库是最常用的JSTL标签。引用核心标签库的语法如下：

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

JSTL格式化标签用来格式化并输出文本、日期、时间、数字。引用格式化标签库的语法如下：

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

JSTL SQL标签库提供了与关系型数据库（Oracle，MySQL，SQL Server等等）进行交互的标签。引用SQL标签库的语法如下：

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

JSTL XML标签库提供了创建和操作XML文档的标签。引用XML标签库的语法如下：

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

JSTL包含一系列标准函数，大部分是通用的字符串处理函数。引用JSTL函数库的语法如下：

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

其中JSTL的核心标签库中主要包含以下标签。



<c:catch> 用来处理产生错误的异常状况，并且将错误信息储存起来；

<c:out> 用于在JSP中显示数据，功能类似<%= ... %>；

<c:set> 用于保存数据，将数据保存到JSP的对象中；

<c:remove> 用于删除数据，清理JSP对象中的无用数据；

<c:if> 逻辑同一般程序中的if，可结合EL表达式获取变量值进行相应显示；

<c:choose> 当做<c:when>和<c:otherwise>的父标签；

<c:when> <c:choose>的子标签，用来判断条件是否成立；

<c:otherwise> <c:choose>的子标签，在<c:when>标签后，当<c:when>标签判断为false时被执行；

<c:forEach> 基础迭代标签，接受多种集合类型；

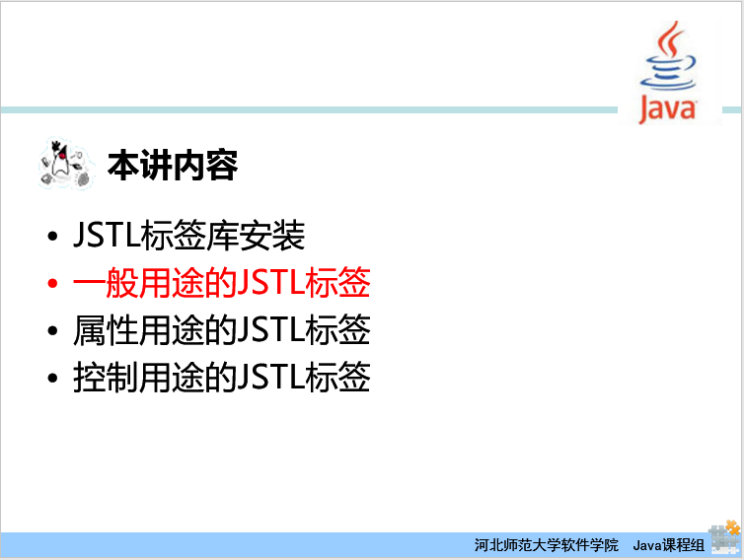
<c:forTokens> 根据指定的分隔符来分隔内容并迭代输出；

<c:url> 使用可选的查询参数来重写URL；

<c:import> 检索一个绝对或相对 URL，然后将其内容暴露在页面中；

<c:redirect> 重定向至一个新的URL。

接下来首先介绍一般用途的JSTL标签



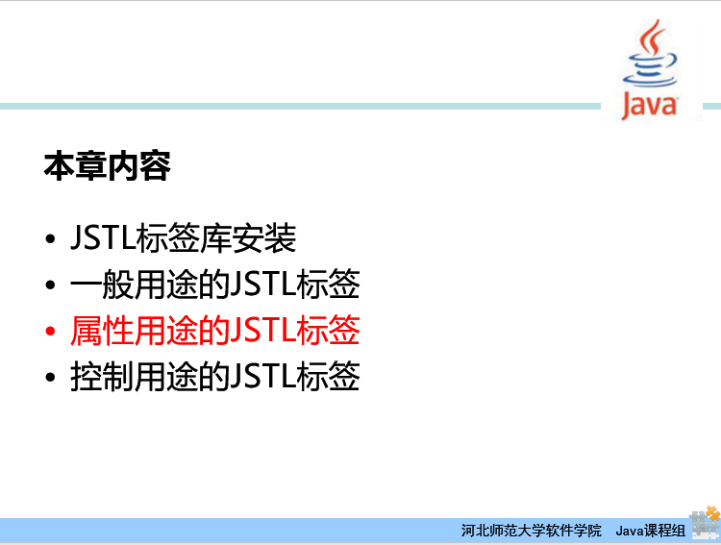
这里介绍两个标签：

（1）<c:catch>捕获JSP页面抛出的异常

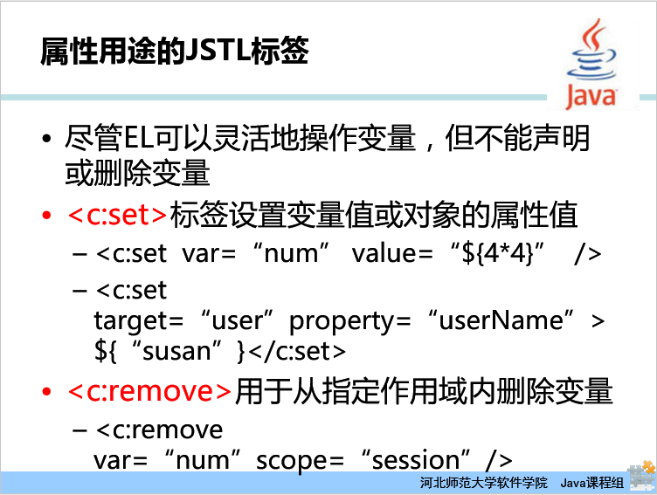
例如：<c:catch var=“e”> action that might throw an exception</c:catch>，可以捕获<c:catch>标签内部代码所抛出的异常。加入处理代码

（2）<c:out>打印输出内容到页面，作用同JSP页面的<%= %>。用于显示Java数据。如果属性value中包含特殊字符，则使用转义字符，如果属性name变量没有初始化，使用default属性可指定默认值。示例代码参考Demo04-03中的index.jsp文件。代码如下：

|  |
| --- |
| <%@taglib prefix=“c” uri=“http://java.sun.com/jstl/core”%>  <c:catch var="myException">  <%  int i = 0;  int j = 10 / i;  %>  </c:catch>    异常：  <c:out value="${myException}" />    异常Exception.getMessage()=  <c:out value="${myException.message}" default="users" />    异常Exception.getCause=  <c:out value="${myException.cause}" default="users" /> |



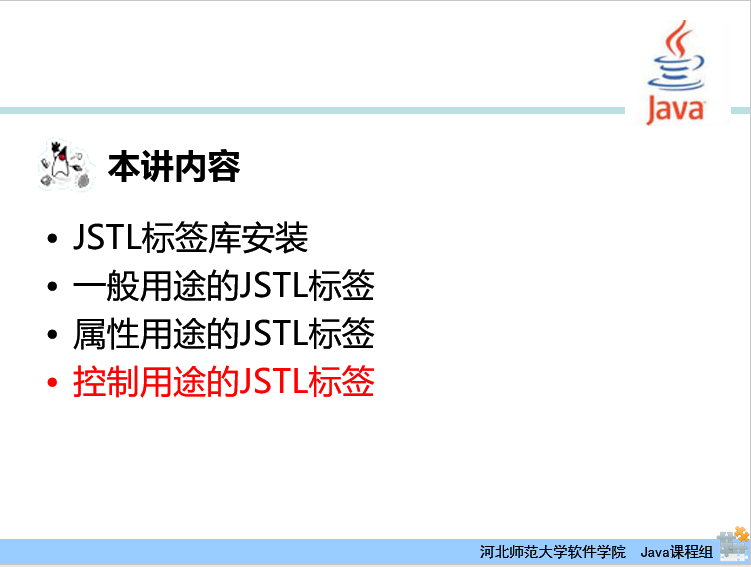
接下来介绍属性相关的JSTL标签。



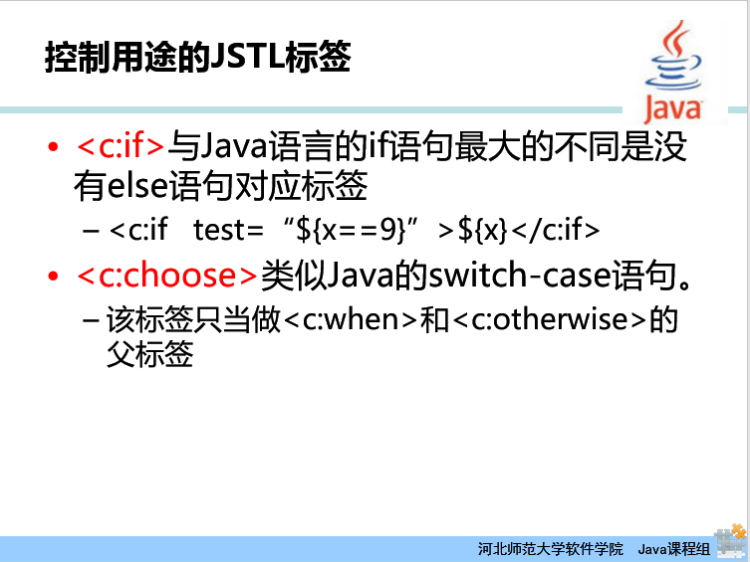
对于上节课讲到的EL表达式，可以非常方便的获取变量的值。但是它不能声明变量也不能删除变量。在JSP中可以通过<c:set>和<c:remove>标签进行变量值的设置和删除。

<c:set>标签可以设置变量值或对象的属性值，示例代码参考Demo04-03中的index.jsp文件。代码如下：

|  |
| --- |
| <%@taglib prefix=“c” uri=“http://java.sun.com/jstl/core”%>  <!-- c:set,c:remove demo -->  <c:set var="num1" value="${4\*4}" />  <c:set var="num2"> ${2\*8}</c:set>  <c:out value="num1=" /><c:out value="${num1}" />  <c:out value="num2=" /><c:out value="${num2}" />  <c:remove var="num1" scope="page" />  <c:out value="num1=" /><c:out value="${num1}" />  <c:remove var="num2" scope="session" />  <c:out value="num2=" /><c:out value="${num2}" />  <jsp:useBean id="user" class="com.Users" scope="session"></jsp:useBean>  <c:set target="${user}" property="userName" value="mengsy" />  <c:set target="${user}" property="password" value="123456" />  <c:set target="${user}" property="email" value="mengsy831120@163.com" />  <c:set target="${user}" property="limits" value="1" /> |

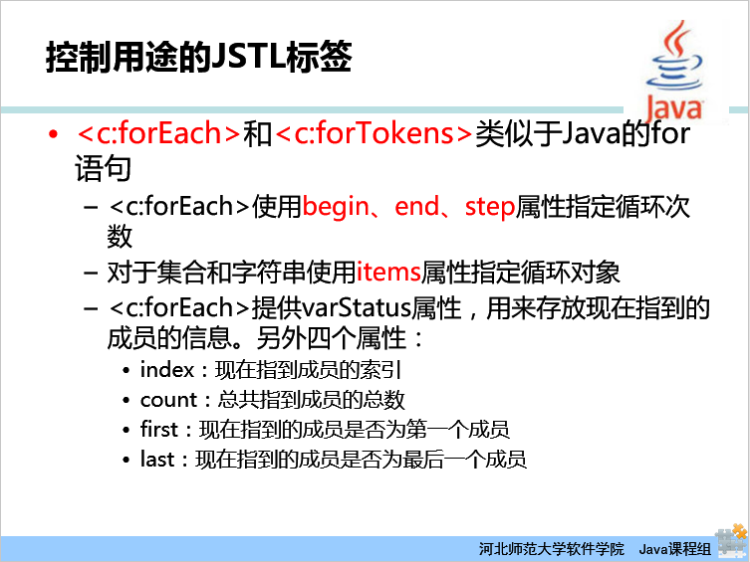


接下来讲解控制用途相关的JSTL标签。



在学习Java语言的过程中，可以通过控制语句根据不同情况进行不同的响应显示。而在EL中无法直接完成相关操作。在JSTL中引入<c:if>和<c:choose>标签进行控制处理。<c:if>与Java语言的if语句最大的不同是没有else语句对应标签。<c:choose>类似Java的switch-case语句。该标签只当做<c:when>和<c:otherwise>的父标签。示例代码参考Demo04-03中的index.jsp文件。代码如下：

|  |
| --- |
| <%@taglib prefix=“c” uri=“http://java.sun.com/jstl/core”%>  JSTL c:if 的使用  <c:if test="${user.userName!=null}">  hello ${user.userName},your password is ${user.password},your email is ${user.email}  </c:if>  <c:choose>  <c:when test="${user.limits==1}">  <c:out value="Hello ${user.userName} your are 管理员"></c:out>  </c:when>  <c:when test="${user.limits==2}">  <c:out value="Hello ${user.userName} your are 版主"></c:out>  </c:when>  <c:when test="${user.limits==3}">  <c:out value="Hello ${user.userName} your are 普通会员"></c:out>  </c:when>  </c:choose> |



在JSTL中还引入<c:forEach>和<c:forTokens>标签来解决在页面中循环展示的问题。<c:forEach>使用begin、end、step属性指定循环次数，对于集合和字符串使用items属性指定循环对象；<c:forEach>提供varStatus属性，用来存放现在指到的成员的信息。另外四个属性：

index：现在指到成员的索引；

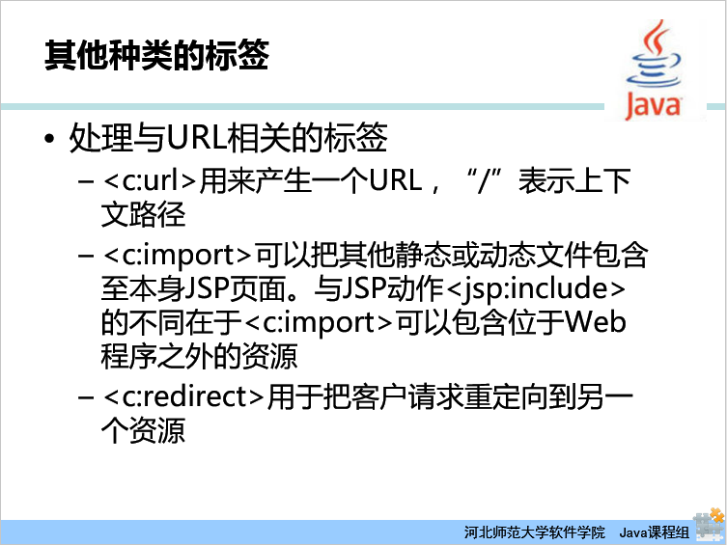
count：总共指到成员的总数；

first：现在指到的成员是否为第一个成员；

last：现在指到的成员是否为最后一个成员；

示例代码参考Demo04-03中的index.jsp文件。代码如下：

|  |
| --- |
| <%@taglib prefix=“c” uri=“http://java.sun.com/jstl/core”%>  JSTL c:foreach 的使用  <c:forEach begin="1" end="5" step="1" var="i">  <c:set target="${user}" property="userName" value="user${i}" />  <c:set target="${user}" property="password" value="password${i}" />  <c:set target="${user}" property="email" value="emial${i}" />  <c:set target="${user}" property="limits" value="${i}" />  hello ${user.userName},your password is ${user.password},your email is ${user.email},your limits is ${user.limits }<br />  </c:forEach> |



在核心标签库中还包含了与URL相关的标签，其功能如下：

<c:url>用来产生一个URL，“/”表示上下文路径；

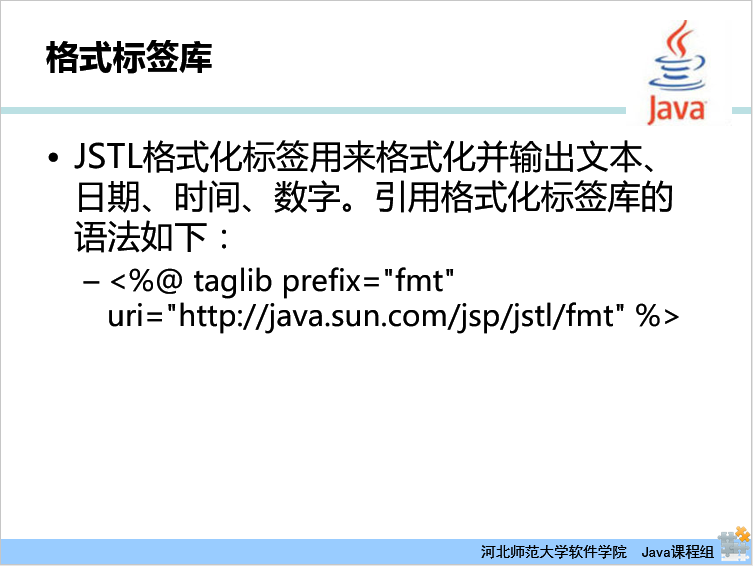
<c:import>可以把其他静态或动态文件包含至本身JSP页面。与JSP动作<jsp:include>的不同在于<c:import>可以包含位于Web程序之外的资源；

<c:redirect>用于把客户请求重定向到另一个资源。

示例代码参考Demo04-03中的index.jsp文件。代码如下：

|  |
| --- |
| <%@taglib prefix=“c” uri=“http://java.sun.com/jstl/core”%>  <%@page import="com.\*"%>  <%@page import="java.util.\*"%>  <%  DbConnection db = new DbConnection();  String userName = request.getParameter("userName");  String password = request.getParameter("password");  User u = db.login(userName, password);  session.setAttribute("user", u);  %>  <c:if test="${user!=null}">  <%  ArrayList<User> list = db.select();  session.setAttribute("list", list);  %>  <c:redirect url="success.jsp"></c:redirect>  </c:if>  <c:if test="${user==null}">  <c:redirect url="login.jsp"></c:redirect>  </c:if> |

核心标签库讲解完毕。接下来列举其他标签库的标签使用方法，希望各位同学能够了解基本用法。



格式化标签库的主要作用是用来格式化并输出文本、日期、时间、数字。引用格式化标签库的语法如下：

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

引入后就可以在当前JSP页面中使用格式化标签库的标签了，常见格式化标签库的标签及作用如下，希望各位同学能够了解：



<fmt:formatNumber> 使用指定的格式或精度格式化数字，例如：

|  |
| --- |
| <c:set var="balance" value="120000.2309" />  格式化数字 (1): <fmt:formatNumber value="${balance}" type="currency"/>  格式化数字 (1): <fmt:formatNumber value="${balance}" type="currency"/>  格式化数字 (2): <fmt:formatNumber type="number" maxIntegerDigits="3" value="${balance}" />  格式化数字 (3): <fmt:formatNumber type="number" maxFractionDigits="3" value="${balance}" />  格式化数字 (4): <fmt:formatNumber type="number" groupingUsed="false" value="${balance}" />  格式化数字 (5): <fmt:formatNumber type="percent" maxIntegerDigits="3" value="${balance}" />  格式化数字 (6): <fmt:formatNumber type="percent" minFractionDigits="10" value="${balance}" />  格式化数字 (7): <fmt:formatNumber type="percent" maxIntegerDigits="3" value="${balance}" />  格式化数字 (8): <fmt:formatNumber type="number" pattern="###.###E0" value="${balance}" /> |

<fmt:parseNumber> 解析一个代表着数字，货币或百分比的字符串，例如：

|  |
| --- |
| <c:set var="balance" value="1250003.350" />  <fmt:parseNumber var="i" type="number" value="${balance}" />  数字解析 (1) : <c:out value="${i}" />  <fmt:parseNumber var="i" integerOnly="true" type="number" value="${balance}" />  数字解析 (2) : <c:out value="${i}" /> |

<fmt:formatDate> 使用指定的风格或模式格式化日期和时间，例如：

|  |
| --- |
| <h3>日期格式化:</h3>  <c:set var="now" value="<%=new java.util.Date()%>" />  日期格式化 (1): <fmt:formatDate type="time" value="${now}" />  日期格式化 (2): <fmt:formatDate type="date" value="${now}" />  日期格式化 (3): <fmt:formatDate type="both" value="${now}" />  日期格式化 (4): <fmt:formatDate type="both" dateStyle="short" timeStyle="short" value="${now}" />  日期格式化 (5): <fmt:formatDate type="both" dateStyle="medium" timeStyle="medium" value="${now}" />  日期格式化 (6): <fmt:formatDate type="both" dateStyle="long" timeStyle="long" value="${now}" />  日期格式化 (7): <fmt:formatDate pattern="yyyy-MM-dd" value="${now}" /> |

<fmt:parseDate> 解析一个代表着日期或时间的字符串，例如：

|  |
| --- |
| <h3>日期解析:</h3>  <c:set var="now" value="20-10-2015" />  <fmt:parseDate value="${now}" var="parsedEmpDate" pattern="dd-MM-yyyy" />  <p>解析后的日期为: <c:out value="${parsedEmpDate}" /></p> |

以下几种标签不在课上讲解。

<fmt:bundle> 绑定资源；

<fmt:setLocale> 指定地区；

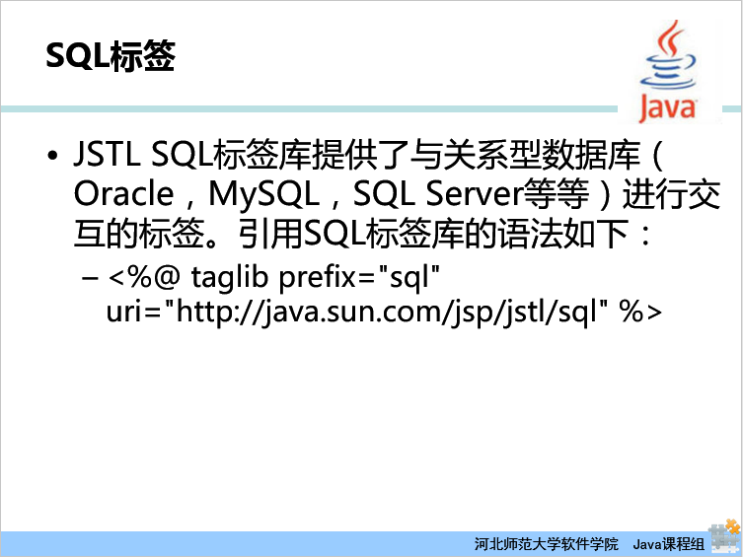
<fmt:setBundle> 绑定资源；

<fmt:timeZone> 指定时区；

<fmt:setTimeZone> 指定时区；

<fmt:message> 显示资源配置文件信息；

<fmt:requestEncoding> 设置request的字符编码；



JSTL SQL标签库提供了与关系型数据库（Oracle，MySQL，SQL Server等等）进行交互的标签。引用SQL标签库的语法如下：

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



引入后就可以在当前JSP页面中使用SQL的标签了，常见SQL标签库的标签及作用如下，希望各位同学能够了解：

<sql:setDataSource> 指定数据源；

<sql:query> 运行SQL查询语句；

<sql:update> 运行SQL更新语句；

<sql:param> 将SQL语句中的参数设为指定值；

<sql:dateParam> 将SQL语句中的日期参数设为指定的java.util.Date 对象值；

<sql:transaction> 在共享数据库连接中提供嵌套的数据库行为元素，将所有语句以一个事务的形式来运行。



JSTL XML标签库提供了创建和操作XML文档的标签。引用XML标签库的语法如下：

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

在使用xml标签前，还需要导入XercesImpl.jar和xalan.jar库，下载地址如下：<http://www.apache.org/dist/xerces/j/>，http://xml.apache.org/xalan-j/index.html。

接下来了解一下XML相关的标签。



引入后就可以在当前JSP页面中使用XML的标签了，常见XML标签库的标签及作用如下，各位同学了解一下即可：

<x:out> 与<%= ... >,类似，不过只用于XPath表达式；

<x:parse> 解析 XML 数据；

<x:set> 设置XPath表达式；

<x:if> 判断XPath表达式，若为真，则执行本体中的内容，否则跳过本体；

<x:forEach> 迭代XML文档中的节点；

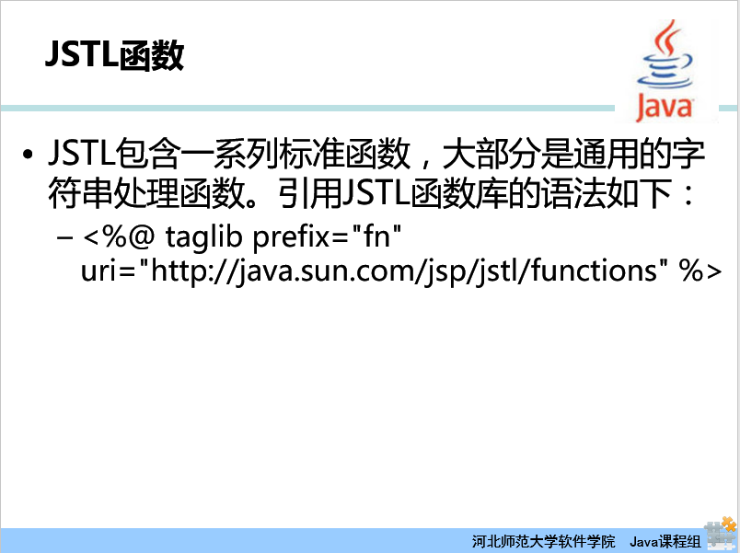
<x:choose> <x:when>和<x:otherwise>的父标签；

<x:when> <x:choose>的子标签，用来进行条件判断；

<x:otherwise> <x:choose>的子标签，当<x:when>判断为false时被执行；

<x:transform> 将XSL转换应用在XML文档中；

<x:param> 与<x:transform>共同使用，用于设置XSL样式表；



JSTL包含一系列标准函数，大部分是通用的字符串处理函数。引用JSTL函数库的语法如下：

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

接下来了解一下JSTL函数相关的标签。



引入后就可以在当前JSP页面中使用JSTL函数标签了，常见JSTL函数标签及作用如下，各位同学了解一下即可：

fn:contains() 测试输入的字符串是否包含指定的子串；

fn:containsIgnoreCase() 测试输入的字符串是否包含指定的子串，大小写不敏感；

fn:endsWith() 测试输入的字符串是否以指定的后缀结尾；

fn:escapeXml() 跳过可以作为XML标记的字符；

fn:indexOf() 返回指定字符串在输入字符串中出现的位置；

fn:join() 将数组中的元素合成一个字符串然后输出；

fn:length() 返回字符串长度；

fn:replace() 将输入字符串中指定的位置替换为指定的字符串然后返回；

fn:split() 将字符串用指定的分隔符分隔然后组成一个子字符串数组并返回；

fn:startsWith() 测试输入字符串是否以指定的前缀开始；

fn:substring() 返回字符串的子集；

fn:substringAfter() 返回字符串在指定子串之后的子集；

fn:substringBefore() 返回字符串在指定子串之前的子集；

fn:toLowerCase() 将字符串中的字符转为小写；

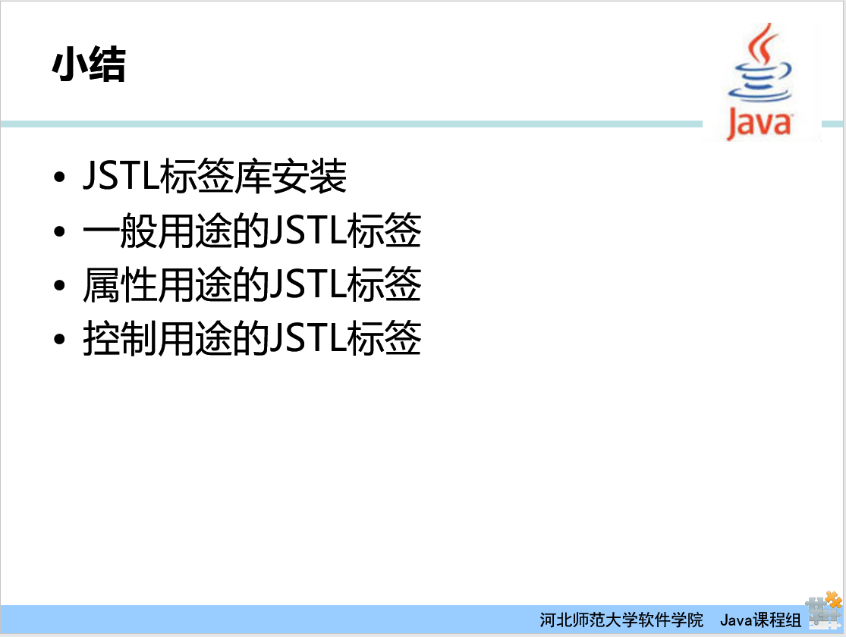
fn:toUpperCase() 将字符串中的字符转为大写；

fn:trim() 移除首位的空白符。

JSTL部分的内容讲解完毕，接下来进行总结。

## 课时小结

（2分钟）



首先介绍了JSTL标签库的安装配置，然后重点讲解核心标签的标签用法。最后简单介绍XML标签、格式化标签(I18N)、SQL标签、函数标签库的引入及基本用法。

# 布置作业

（）