

# EL与JSTL

本章内容：共2小节，14个知识点



- 第1节：表达式语言EL
- 第2节：标准标签库JSTL

# 本章目标

- 理解EL和JSTL的作用;
- 熟练使用常用的EL和JSTL;



# 第1节 【EL】



- 知识点1: EL表达式的功能
- 知识点2: EL表达式的一般格式
- 知识点3: EL表达式的内置对象
- 知识点4: EL表示式获取数据的作用域检索顺序
- 知识点5: 强制使用某个作用域的属性
- 知识点6: EL运算符
- 知识点7: EL表达式的集合运算符等特殊运算符

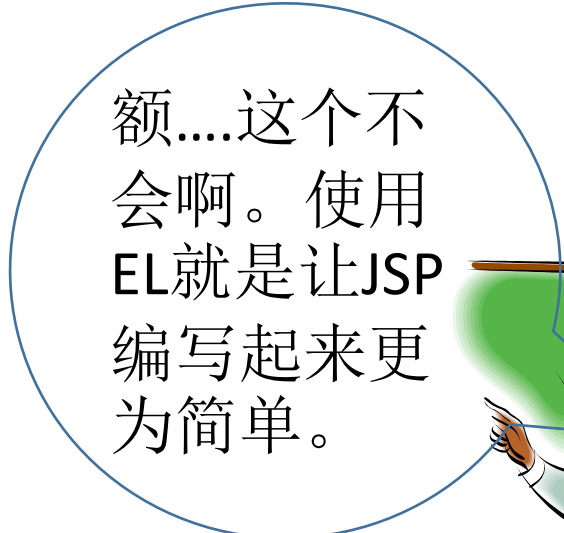
## 知识点1: EL表达式的功能-1

- EL是Expression Language的简称，即表达式语言；
- EL在JSP中使用，服务器会对其进行解析翻译，生成相应的Java代码；
- EL的作用是用来在JSP页面输出动态内容，可以替代JSP中的表达式元素

<%= %>



使用EL后，  
JSP能够运行更快么？



额....这个不会啊。使用EL就是让JSP编写起来更为简单。



## 知识点2： EL表达式的一般格式



- EL的一般格式如下：以\${开始，以}结束；

`${EL表达式}`

- 例如：

`${param.username}`

- 等同于

`<%=request.getParameter("username")%>`

上述举例中的param是EL的内置对象，用来返回请求参数的值，参数值为null时不显示。接下来学习内置对象。

## 知识点3： EL表达式的内置对象-1



- 为了能够方便地输出数据，EL提供了11个内置对象，其中：
  - 2个内置对象为了方便输出请求参数： param/paramValues;
  - 4个内置对象为了方便输出各个范围的属性： pageScope/ requestScope /sessionScope /applicationScope 【使用较多，知识点4单独学习】
  - 2个与请求头有关的内置对象： header/headerValues
  - 2个其他内置对象： cookie/initParam
  - 1个特殊的内置对象pageContext

## 知识点3： EL表达式的内置对象-2



- 内置对象param：用来输出请求参数的值，格式为\${param.请求参数名字}

使用表达式：<%=request.getParameter("name")%><br>

使用EL：\${param.name}<br>

- 内置对象paramValues：用来获取一对多的参数值，返回一个数组。比如某请求参数是通过checkbox传递的，名字为hobbies，要输出所有hobbies值中的第一个值，可以使用如下两种方式：

使用表达式：<%=request.getParameterValues("hobbies")[0]%>

使用EL：\${paramValues.hobbies[0]}



## 知识点3： EL表达式的内置对象-3

- 内置对象header：用来输出某一个请求头的值，格式为\${header.请求头名字}，例如输出请求头accept的值

```
${header.accept}
```

- 内置对象headerValues：如果某个请求头的值有多个，则使用headerValues返回一个数组。如下代码所示：

```
${headerValues.cookie[0]}
```

- 上述代码将返回请求头cookie中的第一个值，例如：

JSESSIONID=A6A22CA4AEE8F9E1111422C889740B24:

## 知识点3： EL表达式的内置对象-4



- 内置对象cookie：用来获取cookie的值，如下代码所示，将输出名字为JSESSIONID的cookie的值： JSESSIONID=A6A22CA4AEE8F9E1111422C889740B24：

```
${cookie.JSESSIONID.value}<br>
```

- 内置对象initParam：用来输出上下文参数；

在web.xml中配置上下文参数：

```
<context-param>
    <param-name>path</param-name>
    <param-value>/WEB-INF/props</param-value>
</context-param>
```

在JSP中使用EL输出path的值：

```
${initParam.path}<br>
```

## 知识点3： EL表达式的内置对象-5



- 内置对象pageContext： EL中的pageContext对象可以调用PageContext类中所有符合规范的getXxx方法，如PageContext类中有如下方法： public abstract ServletRequest getRequest(), 可以通过如下EL调用该方法：

```
${pageContext.request}
```

- 该方法将输出请求对象： org.apache.catalina.core.ApplicationHttpRequest@1b98cbb
- 既然该EL返回的是真正的请求对象，那么就可以继续调用HttpServletRequest中的其他getXxx方法，如：

```
${pageContext.request.remoteAddr}
```

- 上述表达式将调用请求中的getRemoteAddr方法，输出其返回值，例如： 127.0.0.1

## 知识点4：EL表达式获取数据的作用域检索顺序-1



- 与获取属性数据有关的内置对象有四个，分别是：
  - pageScope：页面范围
  - requestScope：请求范围
  - sessionScope：会话范围
  - applicationScope：上下文范围
- 例如获得名字为user的请求属性的pwd属性值，EL为：

```
${requestScope.user.pwd}
```

等同于`<%=((User)request.getAttribute("user")).getPwd()%>`

- 检索顺序：当不指定范围时，例如，`${user.pwd}`，将自动从pageScope开始查找，直到applicationScope，如果没查到，则什么也不显示

## 知识点4：EL表达式获取数据的作用域检索顺序-2

- 1、pageScope作用域：与页面作用域(page)中的属性相关联的Map类，主要用于获取页面范围内的属性值
- 2、requestScope作用域：与请求作用域(request)中的属性相关联的Map类，主要用于获取请求范围内的属性值
- 3、sessionScope作用域：与会话作用域(session)中的属性相关联的Map类，主要用于获取会话范围内的属性值
- 4、applicationScope作用域：与应用程序作用域(application)中的属性相关联的Map类，主要用于获取应用程序范围内的属性值
- 默认的访问顺序：pageScope→requestScope→sessionScope→applicationScope

## 知识点4：EL表达式获取数据的作用域检索顺序-3



作用域	Java代码取值	EL取值
请求作用域	request. <b>getAttribute</b> ("goods");	<code>\${ request<b>Scope</b>.goods }</code>
会话作用域	session. <b>getAttribute</b> ("username");	<code>\${ session<b>Scope</b>.username }</code>
程序作用域	application. <b>getAttribute</b> ("user");	<code>\${ application<b>Scope</b>.user }</code>
页面作用域	pageContext. <b>getAttribute</b> ("pageNum");	<code>\${ page<b>Scope</b>.pageNum }</code>

## 知识点5： 强制使用某个作用域的属性

- 使用pageScope/requestScope/sessionScope/applicationScope中的某一个内置对象获取属性时，则是强制使用该作用域的属性；
- 例如：

页面范围user对象的pwd属性： `${pageScope.user.pwd}`

请求范围user对象的pwd属性： `${requestScope.user.pwd}`

会话范围user对象的pwd属性： `${sessionScope.user.pwd}`

上下文范围user对象的pwd属性： `${applicationScope.user.pwd}`

# JSP表达式语言支持以下11种隐式对象



JSP表达式语言支持以下隐式对象 -

编号	隐式对象	描述
1	pageScope	来自页面范围的范围变量
2	requestScope	来自请求范围的范围变量
3	sessionScope	会话范围的范围变量
4	applicationScope	应用范围的范围变量
5	param	请求参数作为字符串
6	paramValues	请求参数作为字符串集合
7	header	HTTP请求标头作为字符串
8	headerValues	HTTP请求标头作为字符串集合
9	initParam	上下文初始化参数
10	cookie	Cookie值
11	pageContext	当前页面的JSP PageContext对象



## 知识点6： EL运算符-1



- EL中提供了多种运算符，可以对变量或常量进行运算，输出运算结果；
- EL中的运算符包括：
  - 算术运算符
  - 比较运算符
  - 逻辑运算符
  - 其他运算符

## 知识点6: EL运算符-2



- EL算术运算符: +, 实现加法运算; - 实现减法运算; \* 实现乘法运算; /或div实现除法运算; %或mod实现求模运算。如:

```
${19+2} <br>  
${19-2} <br> ${19*2} <br>  
${19/2} <br> ${19%2} <br>
```

- EL比较运算符: 有6种比较运算符, 可以对值进行比较, 返回值为true或false。
  - == 或eq表示等于
  - != 或ne表示不等于
  - < 或lt表示小于
  - > 或gt表示大于
  - <= 或le表示小于等于
  - >= 或ge表示大于等于

## 知识点6： EL运算符-3

- EL逻辑运算符：EL中提供了3个逻辑运算符，可以对boolean类型的值进行运算，返回值为true或false。
  - &&或and表示交集，两个值都是true才返回true。
  - ||或or表示并集，两个值只要有一个是true，即返回true。
  - !或not表示非

**注意：**EL中的多数运算符都用于流程控制中，EL自己不能实现流程控制，需要结合JSTL使用，后续学习。

## 知识点7： EL表达式的集合运算符等特殊运算符



- 使用[]指定索引的方式可以获取数组或List中的元素；

返回请求范围内商品列表集合的第一个商品的名称： `${requestScope.productsList[0].productName}`

- EL其他运算符：除了算术、比较、逻辑运算符外，还有三种其他运算符。
  - empty运算符：判断值是否为null，如果是null，返回true，否则返回false。
  - 关系运算符： `${A?B:C}`如果A为true，则执行B，如果A为false，则执行C。
  - ()运算符：通过（）可改变优先级

```
<form action = "param1.jsp" method = "post">  
    用户名: <input type = "text" name = "username"/><br/>  
    密码: <input type = "password" name = "password"/><br/>  
    性别: <input type = "radio" name = "sex" value = "男" checked/>男  
          <input type = "radio" name = "sex" value = "女">女<br/>  
    爱好: <input type = "checkbox" name = "interst" value = "看书">看书  
          <input type = "checkbox" name = "interst" value = "打篮球">打篮球  
          <input type = "checkbox" name = "interst" value = "旅行">旅行  
          <input type = "checkbox" name = "interst" value = "编程">编程<br/>  
          <input type = "submit" value = "提交"/>  
          <input type = "reset" value = "重置"/>  
</form>
```

```
<center>
    <hr size="5" color="red" width="50%" />
    <%
        request.setCharacterEncoding("utf-8");
    %>
    <hr size="5" color="red" width="50%">
    用户名: ${param.username} <br>
    密码: ${param.password} <br>
    姓名: ${param.name} <br />
    性别: ${param.sex} <br />
    爱好: ${paramValues.interst[0]}
    ${paramValues.interst[1]}${paramValues.interst[2]} <br />
    <hr size="5" color="red" width="50%">
</center>
```

## 本节总结提问【EL】



- EL的作用是什么?
- EL有哪些内置对象?
- EL中如何获取不同作用域的数据?
- EL中有哪些运算符?

## 本节总结【EL】



- EL的作用是替代表达式`<%= %>`简化JSP;
- EL中有11个内置对象, 最常用的有param和不同作用域有关的对象;
- EL中使用pageScope、requestScope、sessionScope、applicationScope表示不同的作用域, 可以用来检索共享的数据;
- EL中提供了算术、比较、逻辑以及和集合相关的其他运算符;



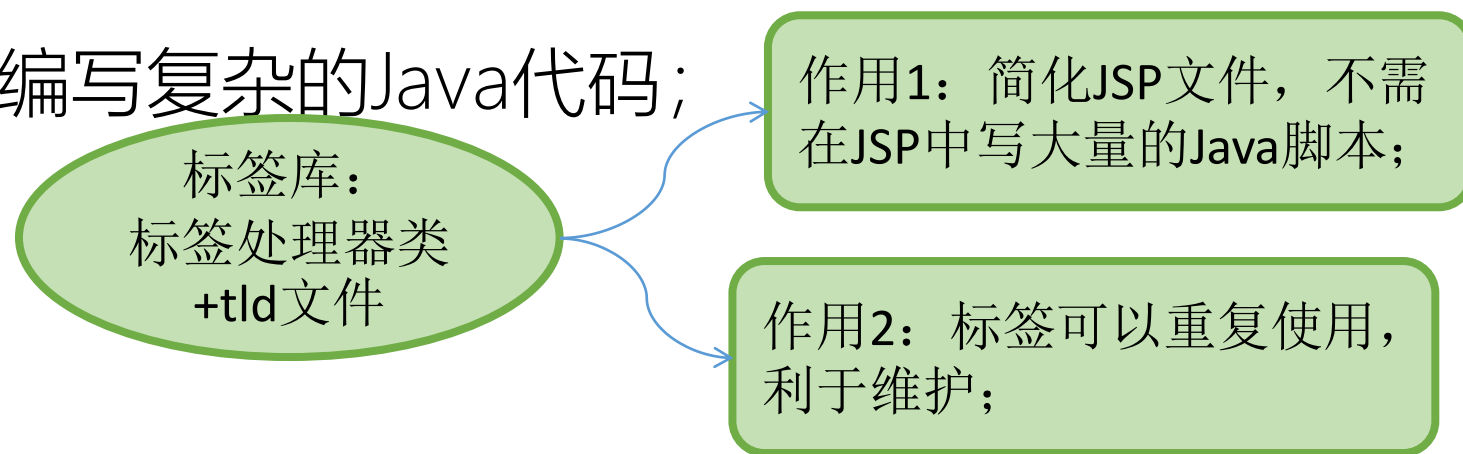
## 第2节 【JSTL】



- 知识点1： 标签库的作用
- 知识点2： JSTL简介
- 知识点3： JSP taglib指令标签的使用
- 知识点4： JSTL属性操作标签的使用
- 知识点5： JSTL条件分支标签的使用
- 知识点6： JSTL迭代标签的使用
- 知识点7： JSTL标签分类

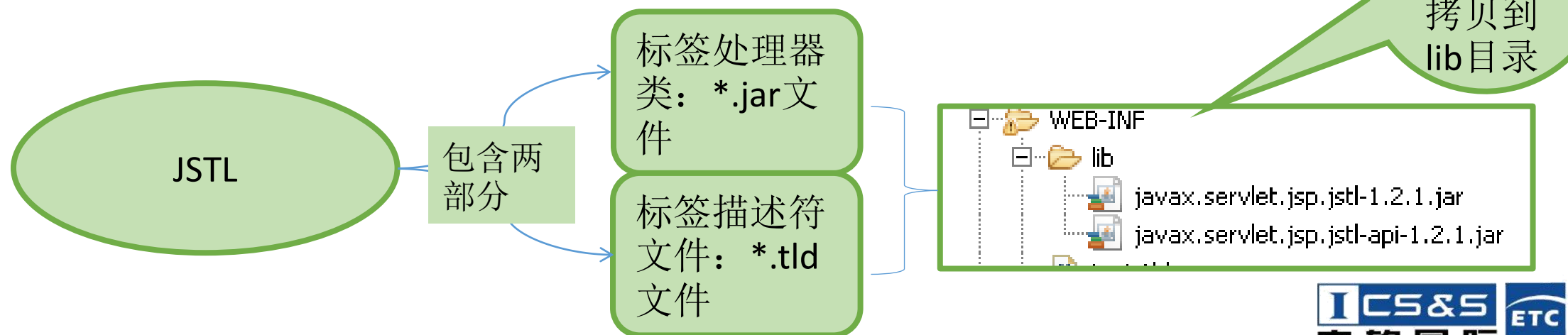
## 知识点1：标签库的作用

- JSP中的静态内容都使用HTML标签实现，如果动态内容也能够使用标签实现，将大大简化JSP；
- JSP中的标签库就是用Java类【遵守一定规范，成为标签处理器类】实现动态功能，将这些动态功能在tld文件（taglib description 标签库描述文件）中描述为标签，在JSP中通过使用标签就可以实现动态功能，不用在JSP中编写复杂的Java代码；



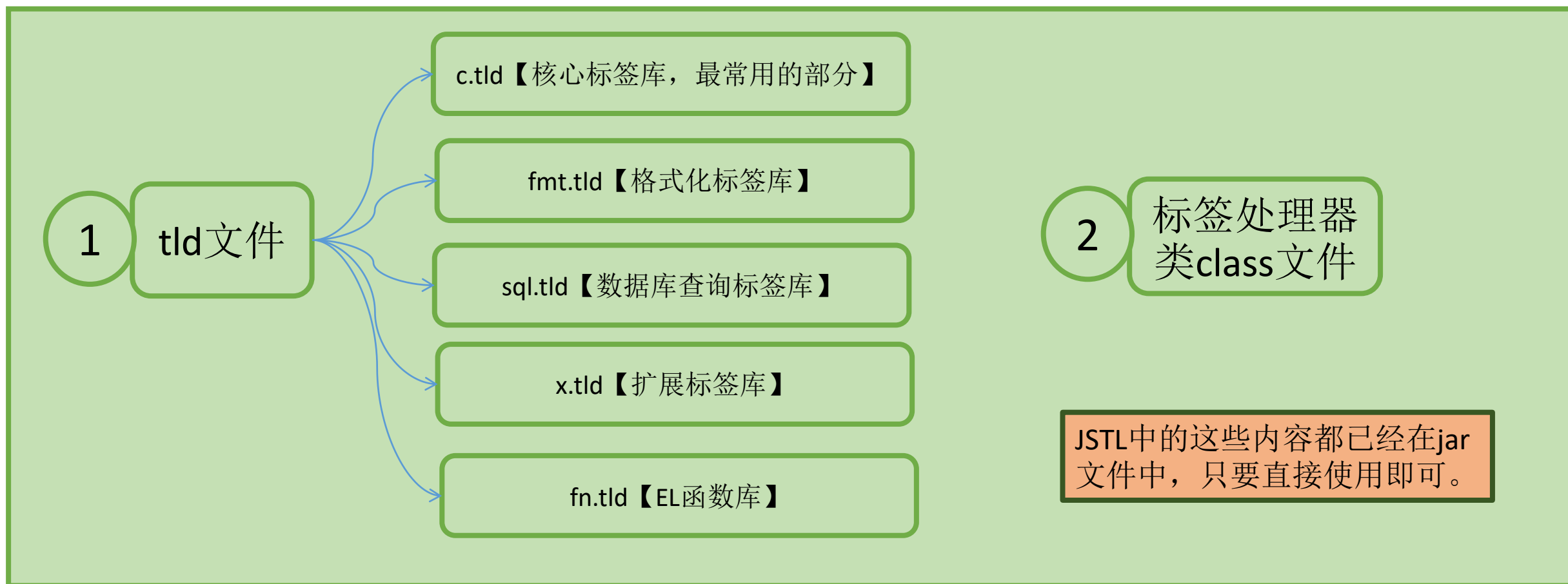
## 知识点2: JSTL简介-1

- JSTL是一套定义好的标签库，可以直接使用；
- JSTL的全称是Jsp Standard Tag Library，即JSP标准标签库；
- JSTL包含很多标签，根据其作用可以分为：属性相关的标签、条件分支相关的标签、迭代标签、其他标签；
- 标签库包括标签处理器类及描述文件tld文件，JSTL也一样：



## 知识点2: JSTL简介-2

- 如上页所示, 要使用JSTL, 首先要把JSTL相关的资源文件保存到lib目录下, 使用解压缩软件解压jar文件, 主要包含两类内容:



## 知识点2: JSTL简介-3

- tld文件中, 对标签进行了描述, 包括标签的名字, 标签处理器类, 标签的属性等, 同时一个tld文件有一个唯一标记的uri; 如下面片段是c.tld文件中的内容:

```
<?xml version="1.0" encoding="UTF-8" ?>
<taglib xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/webjsptaglibrary_2_1.xsd"
  version="2.1">
  <uri>http://java.sun.com/jsp/jstl/core</uri>
  <tag>
    <name>catch</name>
    <tag-class>org.apache.taglibs.standard.tag.common.core.CatchTag</tag-class>
    <body-content>ISP</body-content>
    <attribute>
      <name>var</name>
      <required>>false</required>
      <rtexprvalue>>false</rtexprvalue>
    </attribute>
  </tag>
```

uri是tld文件的唯一标记;

name是标签的名字

标签处理器类

标签体内容

标签的属性, 包括属性名字, 属性是否可以省略, 属性是否可以使用动态表达式。

一个tld文件中, 可以定义多个标签。

### 知识点3：JSP taglib指令标签的使用

- 至此，我们已经了解了JSTL的概念，并为使用JSTL做好了准备：下载了jar文件引入到了工程中；
- 使用标签库之前，需要在JSP中使用taglib指令引入标签库

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

**uri**是tld文件的唯一标记；需要查看tld文件获得。

**prefix**指定使用该标签库时的前缀，只要不使用保留字【比如**jsp**】，其他字符均可。不过一般如果使用**c.tld**标签库，前缀就用**c**，可增强可读性。

## 知识点4：JSTL属性操作标签的使用



- JSTL中与属性操作有关的标签有如下两个：

标签名字	所在标签库	作用	属性
set	c.tld	在指定范围内，将对象保存为属性；	var:属性名字 scope:属性范围 value:属性值，可以使用EL
remove	c.tld	在指定范围内，删除属性；	var:属性名字 scope:属性范围

- 例如，在会话范围内存一个属性，然后进行显示；

```
<c:set var="loggedIn" scope="session" value="true"/>
loggedIn:${sessionScope.loggedIn}
```

.....

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

.....

```
<font size = "3">
```

```
  <c:out value = "c:set标签示例"></c:out><br>
```

第一种语法示例： <br>

```
  <c:set var = "number" value = "2"></c:set>
```

```
  <c:out value = "number的值为： ${number}" /><br>
```

第二种语法示例： <br>

```
  <c:set var = "number">2</c:set>
```

```
  <c:out value = "number的值为： ${number}" />
```

```
</font>
```



## 知识点5：JSTL条件分支标签的使用-1



- JSTL中与条件分支有关的标签有如下四个：

标签名字	所在标签库	作用	属性
if	c.tld	实现if逻辑，当条件为true，执行标签体内容；	test:返回值为布尔值的表达式，作为分支的条件；
choose	c.tld	实现if/else if/else逻辑，需要与when,otherwise结合使用	
when	c.tld	实现else if逻辑，与choose、otherwise配合使用；	test:返回值为布尔值的表达式，作为分支的条件；
otherwise	c.tld	实现else逻辑，与choose、when配合使用；	

## 知识点5：JSTL条件分支标签的使用-2

课堂案例：  
[cif.jsp](#)

- JSTL中与条件分支有关的标签使用：

```
.....  
<%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix = "c"%>  
.....  
  
<center>  
    <c:set var = "name" value = "Brain"/>  
    <c:if test= "${not empty name}">  
        <c:out value = "${name}"/>  
    </c:if>  
</center>
```

.....

```
<%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix = "c"%>
<c:set var = "IDE" value = "Eclipse"/>
    <c:choose>
        <c:when test = "${IDE == 'IDEA'}">
            <c:out value = "你使用的工具是IDEA"/>
        </c:when>
        <c:when test = "${IDE == 'Eclipse'}">
            <c:out value = "你使用的工具是Eclipse"/>
        </c:when>
        <c:when test = "${IDE == 'MyEclipse'}">
            <c:out value = "你使用的工具是MyEclipse"/>
        </c:when>
        <c:otherwise>
            <c:out value = "我是高手我用记事本写代码！"/>
        </c:otherwise>
    </c:choose>
```

## 知识点6：JSTL迭代标签的使用-1



- JSTL中与迭代有关的标签如下所示：

标签名字	所在标签库	作用	属性
forEach	c.tld	迭代集合或数组；	items:需要迭代的数组或集合对象，可以使用EL表达式； var: 迭代过程中的临时变量；
forEachToken	c.tld	迭代字符串；	items: 指定需要迭代的字符串 delims: 表示分隔符 var: 表示使用分隔符分割items产生的字符串

## 知识点6: JSTL迭代标签的使用-2

课堂案例:  
[cforEach.jsp](#)

- JSTL中迭代标签使用如下所示:

### ◆ <c:forEach/>迭代标签

#### ■ 实现对集合的遍历

```
<%@ page import = "java.util.*"%>
```

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

```
<font size = "4">
```

```
<%
```

```
    ArrayList listt = new ArrayList();
```

```
    for(int i = 0;i<5;i++){
```

```
        listt.add(i,(i+1)*15);
```

```
    }
```

```
    request.setAttribute("list1",listt);
```

```
%>
```

```
<c:out value= "c:forEach 标签示例: "/><br>
```

```
<c:forEach items = "${list1}" var = "current1" varStatus = "status1">
```

```
    <c:out value = "序号: "/>
```

```
    <c:out value = "${status1.count}"/>
```

```
    <c:out value="值: "></c:out>
```

```
    <c:out value = "${current1}"/><br>
```

```
</c:forEach>
```

```
</font>
```

集合中元素的名称

当前循环的状态信息,  
例如循环的索引号

```
<c:forEach var="varName" items="items" varStatus="varStatus">
```

```
    .....
```

```
</c:forEach>
```

```
<%@ page import = "java.util.*"%>
<%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix = "c" %>
<hr size="5" color="red" width="50%" />
    <font size="4">
<%
    String question = "What*is*your*name*?";
    request.setAttribute("question1", question);
%>
<c:out value="c:outTokens">标签示例:" /> <br>
<c:forTokens items="${question1}" var="current1" delims="*" varStatus="status">
    第<c:out value="${status.count}" />次取出的单词为: <c:out value="${current1}" />
        <br><br>
    </c:forTokens> </font>
<hr size="5" color="red" width="50%" />
```

# 知识点7: JSTL标签分类



标签库名称	资源标示符 (uri)	前缀 (prefix)
核心标签库	http://java.sun.com/jsp/jstl/core	c
国际化/格式化标签库	http://java.sun.com/jsp/jstl/fmt	fmt
XML标签库	http://java.sun.com/jsp/jstl/xml	x
数据库标签库	http://java.sun.com/jsp/jstl/sql	sql
函数标签库	http://java.sun.com/jsp/jstl/functions	fn

## ◆ 核心标签库

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

## ◆ 国际化/格式化标签库

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

**提醒:** JSTL中有很多其他标签, 不一一学习, 使用的过程都一样, 主要熟悉标签的作用, 理解每个属性的作用即可。

## 本节总结提问【JSTL】



- JSTL有什么作用?
- JSTL使用步骤有哪些?
- 列举几个JSTL的常用标签?



## 本节总结【JSTL】



- JSTL是一套JSP的标准标签库，可以直接用于JSP中；
- 使用JSTL首先需要下载相关的jar文件并保存到工程的lib目录下；在JSP中使用taglib指令引入需要使用的标签库；
- forEach、set、if等是JSTL中常用的标签；

## 本章总结



- EL可以替代`<%= %>`，简化JSP文件；
- EL中有11个内置对象，可以用来获取不同范围的属性、请求参数等；
- EL不能实现流程控制，所以一般与JSTL一起使用；
- JSTL是一套标准标签库，可以直接使用，往往和EL一起使用；

# 本章作业



- 作业1:
- 题目：改造之前章节的课堂示例及作业，JSP中不出现`<%%>`、`<%= %>`，均使用EL、JSTL实现
- 难度：易

