**EL**

**概述**

EL(Expression Language)，中文名表达式语言，可以用来替换JSP页面中的java代码。

**JSP页面如何使用EL**

jsp页面中如果想要使用EL，必须将page指令标记中的isELIgnored属性置为false，如果置为true，则当前JSP页面无法使用EL。注: isELIgnored属性默认为false。

**EL表达式的基本语法**

格式: ${表达式} or ${常量}

如果{ }中是一个表达式，那么表达式的值可以是整型，浮点型，字符串，布尔型，null。

如果{ }中是一个常量，那么常量也可以是整型，浮点型，字符串，布尔型，null。例如: ${100}、${12.3}、${“Hello”}、${true}、${null}

注: 如果{ }中的表达式的值null或常量为null，则网页中会显示一个空字符，并不会显示null。

注: ${表达式} 就好比JSP中的java表达式<%=表达式 %>

**EL中常见的操作符**

点操作符 . -- 使用方便

中括号操作符 -- 功能强大(可以包含特殊字符，还可以获取变量值)

**EL中的运算符**

Empty运算符用于判断一个只是否为空,为空返回true，不为空返回false

条件运算符(? : )



**EL表达式中的隐式对象**

隐式对象分3类

作用域访问对象，简称域对象 (pageScope，requestScope，sessionScope，applicationScope)

EL中的pageScope对象就对应JSP中的内置对象pageContext

EL中的requestScope对象就对应JSP中的内置对象request

EL中的sessionScope对象就对应JSP中的内置对象session

EL中的applicationScope对象就对应JSP中的内置对象application

即EL中的隐式对象pageScope，requestScope，sessionScope，applicationScope分别用于获取JSP中的内置对象pageContext，request，session，application中的数据。

注: 如果获取某个属性时，不明确指定是从哪个作用域(域对象)中获取，则默认会根据作用域的大小，从小到大依次取值。

参数访问对象 (param，paramValues)

参数访问对象用于获取用户在表单中输入的数据以及在超链接中传的值（a.jsp?a=b&c=d）

JSP隐式对象 (pageContext)

注: pageContext对象是EL中的隐式对象，同样也是JSP中的内置对象。在EL中如果想要使用JSP中的内置对象，就要使用到pageContext对象，pageContext对象调用相应的方法就可以获取到对应JSP内置对象。pageContext对象就像是EL到JSP的桥梁。

**JSTL**

**概述**

JSTL(Java Server Page Standard Tag Library)，中文名JSP标准标签库。也可以用于替换JSP页面中的java代码，且功能比EL更为强大。jstl可以与EL配合使用，并且在jstl中嵌套使用EL。JSTL由5个不同功能的标签库组成，包括Core、I18N、XML、SQL和Functions。这里主要学习核心标签库(Core)。

**在使用JSTL核心标签库前，需要先对其进行配置**

第一: 先导入两个jar包，分别是jstl.jar和standard.jar

第二: 如果当前JSP页面想要使用jstl，必须使用taglib指令标记引入jstl核心标签库。例如:

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

注: prefix="c"% 用于指定标签的前缀是什么，我这里指定标签的前缀为c，可以指定任意英文字符作为标签的前缀。

拓展: 如果当前JSP页面要使用Functions标签库，同理:

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

18N、XML、SQL也同理。

注: functions(函数标签库)中的函数是放在EL表达式中使用的。

**JSTL中的核心标签库(Core)**

核心标签库包含: 通用标签，流程控制标签(条件标签)，迭代标签，URL标签

通用标签有:

<c: set > 用于赋值

用法一: 在某个作用域中，给某个变量赋值

用法二: 在某个作用域中，给普通对象(JavaBean对象)的属性重新赋值

用法三: 在某个作用域中，给map对象(map集合对象)的属性重新赋值

注意: 在某个作用域中，使用<c: set>标签给普通对象和map集合对象的某个属性重新赋值时，前提是这个对象已经被创建出来了，否则无法赋值。在某个作用域中，使用<c: set>标签给某个变量赋值，如果该变量在这个作用域(域对象)中存在，那么就是重新赋值，如果该变量在这个作用域(域对象)不存在，那么就是在这个作用域(域对象)中声明一个新的变量，并为其赋值，等价于 request.setAttribute(“xxx”, “yyy”);

<c: out> 用于输出数据 等价于 <%=? %> 和 ${表达式}

格式: <c: out value=”输出的数据” default=”xxx”>

注意: value值可以是一个字符串，也可以是一个EL表达式。default可有可无，如果value值为null时，就是出default值:xxx。

流程控制标签有:

<c:if test=”表达式”>实际就是单分支if语句，因为单分支if语句没与之匹else语句。所以<c:if>标签也不存在对应<c:else>标签。jstl中就不存在<c:else>标签

<c:choose>、<c:when>和<c:otherwise>标签联合使用实现多分支if语句，即if...else...if...else... 。注意<c:when>和<c:otherwise>标签嵌套在<c:choose>标签中。并且<c:if>标签和<c:choose>标签可以嵌套使用。

迭代标签有:

<c:forEach> 标签用来实现普通for循环和增强for循环

增强for循环

<c:forEach var=”变量名” items=”数组或Collection集合对象”>

每一次循环都从数组或Collection集合中取出一个元素赋予前面的变量var

普通for循环

<c:forEach var=”变量名” begin=”?” end=”?” step=”?” varStatus=”status”>

var是循环变量名, begin是循环的开始位置，即为循环变量赋初值，默认值为0，end是循环的结束位置，默认值为最后一个元素，step是迭代的步长，默认值为1，varStatus是循环变量的状态，varStatus包含count属性(循环的次数)，index属性(当前循环变量的值，即迭代的索引，是一个索引)。