官网文档：

<http://docs.oracle.com/javaee/7/tutorial/doc/jsf-el001.htm>

<http://docs.oracle.com/javaee/> 第9章；

el和jstl：jsp、jsf等中使用；

<c:

<fn:

el，页面直接使用；

写法，${...}或#{...}

条件判断中的，使用${...}，

表单中赋值使用的，使用#{...}；

~

简而言，表单中使用#{...}，js中使用${...}；

示例，

<h:inputText id="name" value="#{customer.name}" />

<c:if test="${sessionScope.cart.numberOfItems > 0}"> ... </c:if>

jstl，指令引入；

引入，

xmlns:c=*"http://java.sun.com/jsp/jstl/core"*

*或，*

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

<c:choose>

<c:when test=*"${contractApplyAction.currentExamination.customerName eq contractApplyAction.lastHistory.customerName}"*>

<h:outputText value=*"#{contractApplyAction.lastHistory.customerName}"* />

</c:when>

<c:otherwise>

<h:outputText value=*"#{contractApplyAction.currentExamination.customerName}"* class=*"hcolor"* />

<h:outputText value=*"#{contractApplyAction.lastHistory.customerName}"* />

</c:otherwise>

</c:choose>

~

控制判断，

<c:when test=*"${contractApplyAction.lastHistory.customerName ==null}"*>

迭代标签

语法：<c:forEach items=“collection” var=“item” varStatus=“status” begin=“int“

end=”int” step=“int” >

//循环体${item.name}

</c:forEach>

说明:1)items:是集合，用EL表达式；

2)var:变量名，存放item

3)varStatus: 显示循环状态的变量

①index:从0开始;

②count:元素位置,从1开始;

③first:如果是第一个元素则显示true;

④last:如果是最后一个元素则显示true;

4)begin:循环的初始值(整型)；

5)end: 循环结束 ;

6)step:步长,循环间隔的数值；

//加一个%号，

<c:set var=*"str"* value=*"#{contractApplyAction.lastHistory.contractRate}%"* />

<c:if test=*"${str eq contractApplyAction.lastHistory.contractRateLabel}"*>

#{contractApplyAction.lastHistory.contractRateLabel}

</c:if>

~

c:set可以拼接字符串，

<c:set var=*"str"* value=*"${param1}nihao${param2}"* />

fn：

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

//长度，

<c:if test=*"${fn:length(str)>0}"*>

fmt，格式转换：

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

格式化日期

<fmt:formatDate value=“” pattern=“yyyy-MM-dd HH:mm:ss”/>

pattern=*"#,##0.##"---金额的,号分隔显示；*

//小数，如:18.76---默认，4舍6入5奇偶； | 5的，整数为偶数舍去，奇数进1；

<fmt:formatNumber value=*"*${transfer.borrowApr}*"* pattern=*"#,##0.##"*/>%

//取整数，

<fmt:formatNumber value=*"*${(80)/30 }*"* pattern=*"#"*/>

//四舍五入，

<fmt:formatNumber value=*"*${(80)/30+0.0001 }*"* pattern=*"#"*/>

//取整数，去尾；---减去0.5

<fmt:formatNumber value=*"*${(80)/30-0.5 }*"* pattern=*"#"*/>

//取整数，进1；---加上0.5

<fmt:formatNumber value=*"*${(80)/30+0.5 }*"* pattern=*"#"*/>

EL 运算符

类别 运算符

算术运算符 +、-、\*、/（或 div）和 %（或 mod）

关系运算符 ==（或 eq）、!=（或 ne）、<（或 lt）、>（或 gt）、<=（或 le）和 >=（或 ge）

逻辑运算符 &&（或 and）、||（或 or）和 !（或 not）

验证运算符 empty

El字符转换，

|  |  |
| --- | --- |
| **Character** | **Character Entity Code** |
| < | &lt; |
| > | &gt; |
| & | &amp; |
| ’ | &#039; |
| " | &#034; |

El，计算：

//除以30，

${(transfer.intervalDay)/30 }

//除以30，然后+1；

${(transfer.intervalDay)/30+1 }

///////////////////////////////////////////////////////////////////////////////////////////