01_EL表达式介绍

- 什么是EL表达式?
  - EL是expression Language的缩写，它是jsp内置的一种表达式语言，从jsp2.0开如，就不让在使用jsp脚本，而是使用el表达式来替换jsp中jsp脚本。
- EL表达式作用
  - 获取域数据
  - 获取常用web对象
- EL表达式格式
  - ${表达式}
  - 使用el表达式的主要是代替jsp页面上的<%=%>,也就是说，el表达式主要作用是向浏览器输出数据
- 注意事项
  - EL表达式是jsp2.0中的规范，要使用el表达式必须使用支持jsp2.0技术的web服务器(tomcat)，jsp2.0对应的servlet版本是servlet2.4
  - 如果不能使用el表达式请检查isELIgnored属性
- 入门案例
  - 使用el表达式向页面输出数字3

02_EL获取域数据

- El表达式怎样获取域中数据

域	el表达式
page域	${pageScope.name}
request域	${requestScope.name}
session域	${sessionScope.name}
application域	${applicationScope.name}

  - 使用el表达式获取时，如果没有查找到结果，返回的不是null,而是一个"".
  - 使用${msg}没有指明域，那么就从pageScop->requestScop->sessionScop->applicationScop中依次查找

- 获取简单域数据
  - 直接使用el表达式
- 获取复杂数据
  - 获取数组

```
<%
    //1,使用el表达式获取数组
    String[] strs = { "aa", "bb", "cc" };
    request.setAttribute("strs", strs);
%>
//获取request域中st数组第二个值:
jsp脚本:
    <%=
        ((String[]) request.getAttribute("strs"))[1]
    %>
el表达式:
    ${requestScope.strs[1]}
```

- 获取List集合

```
<%
    List<String> list = new ArrayList<String>();
    list.add("111");
    list.add("222");
    list.add("333");
    request.setAttribute("list", list);
%>
//获取request域中list集合中第二个值:
jsp脚本:
    <%=
        ((List<String>) request.getAttribute("list")).get(1)
    %>
 el表达式:
    ${requestScope.list[1]}
```

- 获取map集合
  - el表达式map[1]中的1默认类型为Long,所以如果要使用map集合存储,键的类型应该为Long!!!

```
<%

    Map<String, String> map = new HashMap<String, String>();
    map.put("username", "张三");
    map.put("age", "18");
    request.setAttribute("map", map);
%>
jsp脚本:
    <%=
        ((Map<String, String>) request.getAttribute("map"))
                .get("username")
    %>
el表达式:
    ${requestScope.map.username }
```

- 获取java对象

```
<%
    User user = new User();
    user.setUsername("tom");
    user.setPassword("123");
    request.setAttribute("user", user);
%>
jsp脚本:
    <%=((User)
        request.getAttribute("user")).getUsername()
    %>
el表达式:
    ${requestScope.user.username}
```

03_EL执行运算

- 算术运算符

```
+ - * /(div)  %(mod)
```

  - 注意在el表达式中不能使用+号进行字符串的拼接，只能执行加法操作。

- 关系运算符

```
>(gt) >=(ge) <(lt) <=(le) == (eq) !=(ne)
```

  - 执行关系运算它得到的结果就是boolean

- 逻辑运算符

```
&&(and)  ||(or)  !(not)
```

  - 执行逻辑运算得到的结果也是boolean

- 三目运算符

  - 它与java中的三元运算符一样。

04_EL中的Web对象

- 一共有11个对象

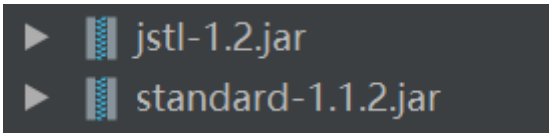
  - pageScope：获取pageContext域属性
  - requestScope：获取request域属性
  - sessionScope：获取session域属性

    - applicationScope：获取application域属性
    - param：对应参数，它是一个Map，其中key是参数，value是参数值，适用于单值的参数，相当于request.getParameter("xxx")
    - paramValues：对应参数，她是一个Map，其中key是参数，value是多个参数值，适用于多值的参数，相当于request.getParameterValues("xxx")
    - header：对应请求头，它是一个Map，其中key表示头名称，value是单个头值，适用于单值的请求头，相当于request.getHeader("xxx")
    - headerValues：对应请求头，它是一个Map，其中key表示头名称，value是多个头值，适用于多值的请求头，相当于request.getHeaders("xxx")
    - initParam：获取web.xml中<context-param>内的参数
    - cookie：用于获取cookie，Map<String,Cookie>，其中key是cookie的name，value是cookie对象
    - pageContext：可以获取JSP九大内置对象

- 常用Web对象
    - pageScope、requestScope、sessionScope、 applicationScope、cookie、pageContext

- 常用操作
    - ${pageContext.request.contextPath}
    - ${cookie.get(key).value}

05_jstl标签库介绍

- A.什么是jstl?它的作用是什么?
    - JSTL(jsp standard tag Library) jsp标签库，它是apache对el表达式的扩展，jstl与el结合使用可以完成更强大的功能。
- B.怎样在jsp页面上使用jstl标签
    - 导入jstl的jar包
    - 

    - 在jsp页面上使用taglib指令来导入jstl
        - <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

06_jstl核心标签

- set:向域中设置数据 * <c:set var="username" value="old qiu" scope="request"></c:set> * 向request域中存储一个数据，名称为username，值为old qiu
- remove：将数据从域中移除
- catch：捕获异常

```
<%--捕获异常--%>
<c:catch var="e">
    <%--产生异常--%>
    <%
        int num = 1 / 0;
    %>
</c:catch>
${e}
```

- if：条件判断
- forEach：遍历
- forToken：分割字符串

07_jstl标签库案例之显示商品信息

- 需求:完成如下效果
  -

id	商品	价格	数量	小计
1	电视	1000.0	2	2000.0
2	冰箱	2000.0	3	6000.0
				总计:8000.0

- 商品类

```
public class Product {

    private Integer pId;//id
    private String pName;//商品名称
    private Integer count;//商品数量
    private double price;//商品单价

}
```

- 商品页面

```
<table border="1px" cellspacing="0px" cellpadding="10px" width="400px">
    <tr>
        <th>id</th>
        <th>商品</th>
        <th>价格</th>
        <th>数量</th>
        <th>小计</th>
```

```
    </tr>
    <c:set var="totalPrice" value="0" scope="request"></c:set>
    <c:forEach items="${productList}" var="product">
        <tr>
            <td>${product.pId}</td>
            <td>${product.pName}</td>
            <td>${product.price}</td>
            <td>${product.count}</td>
            <td>${product.price * product.size}</td>
        </tr>
        <c:set var="totalPrice" value="${totalPrice + product.price *
product.count}"></c:set>
    </c:forEach>

    <tr >
        <td colspan="5" align="right">
            总计:${totalPrice}
        </td>
    </tr>

</table>
```