**El 表达式和 Jstl 标签库学习**

**El 表达式学习**

1、 什么是 EL 表达式

全称：Expression Language，一种写法非常简洁的表达式。语法简单易懂，便于使用。表达式语言的灵感来自于 ECMAScript 和XPath 表达式语言

2、 EL 表达式的作用

作用：让 jsp 书写起来更加的方便。简化在 jsp 中获取作用域或者请求数据的写法。也会搭配 Jstl 来进行使用。

3、 使用 EL 表达式

语法结构：${expression},提供.和[ ]两种运算符来存取数据。

使用El表达式获取作用域数据:

作用：获取作用域对象中的数据。

注意：获取的是pageContext、request、session、application四个对象中的数据，其他数据一概不理会。

找到了则获取返回，找不到则什么都不做，也不报错。

●EL表达式

> 是为了简化咱们的jsp代码，具体一点就是为了简化在jsp里面写的那些java代码。

\* 写法格式: ${表达式 }

|  |  |
| --- | --- |
| |  | | --- | | <%@ **page import**="**com.web.pojo.T\_User**" %> <%@ **page import**="**java.util.ArrayList**" %> <%@ **page import**="**java.util.HashMap**" %> <%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %> **<%** String path = request.getContextPath();  String basePath = request.getScheme() + **"://"** + request.getServerName() + **":"** + request.getServerPort() + path + **"/"**; **%>** <**html**> <**head**>  <**base href="<%=**basePath**%>"**>  <**title**>Title</**title**> </**head**> <**body**>  <**h3**>el表达式</**h3**>  *<%--  el表达式:  用来取作用域中的数据  session/request/servletcontext/pageContext  --%>* **<%** T\_User tuser = **new** T\_User();  tuser.setUname(**"小美"**);  tuser.setPwd(**"111"**);  session.setAttribute(**"tuser"**,tuser);  session.setAttribute(**"city"**, **"北京"**);  session.setAttribute(**"uname"**, tuser.getUname());  ArrayList<String> list = **new** ArrayList<>();  list.add(**"上海"**);  list.add(**"南京"**);  list.add(**"广州"**);  session.setAttribute(**"list"**, list);  HashMap<String,Integer> map = **new** HashMap<>();  map.put(**"a"**, 10);  map.put(**"b"**, 20);  map.put(**"c"**, 30);  session.setAttribute(**"m"**, map);  ArrayList<T\_User> users = **new** ArrayList<>();  users.add(**new** T\_User(**"张三"**, **"123"**) );  users.add(**new** T\_User(**"李四"**, **"321"**) );  users.add(**new** T\_User(**"王五"**, **"456"**) );  session.setAttribute(**"users"**, users);  **%>** <**h4**>通过el表达式取值</**h4**>  *<%--根据键取值,如果键是对象,那么可以:键.属性取值--%>* **${**tuser.uname**}**--**${**tuser.pwd**}**<**br**/>  **${**city**}**--**${**uname**}**<**br**/>  **${**list.get(0)**}**--**${**list.get(1)**}**--**${**list.get(2)**}**<**br**/>  **${**m.a**}** -- **${**m.b**}**-- **${**m.c**}**<**br**/>  **${**users.get(0).uname**}**--**${**users.get(0).pwd**}**<**br**/>  **${**users.get(1).uname**}**--**${**users.get(1).pwd**}**<**br**/>  **${**users.get(2).uname**}**--**${**users.get(2).pwd**}**<**br**/>  *<%--无论是session/request/servletcontext/pageContext都可以使用以上的方式取各作用域中的数据--%>* <**hr**/> **<%** */\*四种常用的作用域放入同一个键,不同的值\*/  //session.setAttribute("a", "session");  //request.setAttribute("a", "request");* application.setAttribute(**"a"**, **"application"**);  *//pageContext.setAttribute("a", "pageContext");* **%>** <**h4**>测试作用域的取值优先顺序</**h4**>  **${**a**}** <**br**/>  <**b**>  优先级: pageContext > request > session > application  </**b**> </**body**> </**html**> | |

●如何使用

**JSTL 标签库**

1、 什么是 JSTL 标签库

JSTL 是 apache 对 EL 表达式的扩展（也就是说 JSTL 依赖 EL），

JSTL 是标签语言！JSTL 标签使用以来非常方便，它与 JSP 动作

标签一样，只不过它不是 JSP 内置的标签，需要我们自己导包，以

及指定标签库而已！

如果你使用 Idea开发 JavaWeb，那么在把项目发布

* Tomcat 时，你会发现，Idea会在 lib 目录下存放 jstl 的

Jar 包！如果你没有使用 Idea开发那么需要自己来导入这个 JSTL 的 Jar 包：jstl-1.2.jar。

2、 JSTL 标签库的作用

用来提升在 JSP 页面的逻辑代码的编码效率，使用标签来

替换逻辑代码的直接书写，高效，美观，整洁，易读。

3、 使用 JSTL 标签库

1. 使用：
   1. 导包

1. 使用 taglib 标签引入资源
2. 核心标签库
3. 格式化标签库

学习网址：<http://www.runoob.com/jsp/jsp-jstl.html>

●JSTL

> 全称 ： JSP Standard Tag Library jsp标准标签库

> 简化jsp的代码编写。 替换 <%%> 写法。 一般与EL表达式配合

●怎么使用

1. 导入jar文件到工程的WEB-INF/lib jstl.jar standard.jar

1. 在jsp页面上，使用taglib 指令，来引入标签库

3. 注意： 如果想支持 EL表达式，那么引入的标签库必须选择1.1的版本，1.0的版本不支持EL表达式。

http://java.sun.com/jsp/jstl/core

●常用标签

\* c:set

\* c:if

\* c:forEach

|  |
| --- |
| <%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %> <%@**taglib prefix**="**c**" **uri**="**http://java.sun.com/jsp/jstl/core**" %> **<%** String path = request.getContextPath();  String basePath = request.getScheme() + **"://"** + request.getServerName() + **":"** + request.getServerPort() + path + **"/"**; **%>** <**html**> <**head**>  <**base href="<%=**basePath**%>"**>  <**title**>Title</**title**> </**head**> <**body**>  *<%-- 设置值:给age赋值,等同于往作用域中放数据--%>  <%-- <c:set var="age" value="26"></c:set>--%>* **<%** session.setAttribute(**"age"**, 12);  **%>** <**c:if test="${**age >=18**}"**>  成年了,可以干大事了  </**c:if**>  <**c:if test="${**age <18**}"**>  未成年,听妈妈的话  </**c:if**>    </**body**> </**html**> |