EL表达式小记

**外文名**

Expression Language

**使    用**

JSP

**作    用**

在 JSP 中简化**表达式**的方法 <% %> 和 <%= %>

**1、语法结构**

**${expression}**

**2、[ ]与.运算符**

EL 提供“.“和“[ ]“两种运算符来存取数据。

**当要存取的属性名称中包含一些特殊字符，如 . 或 - 等并非字母或数字的符号，就一定要使用“[ ]“**。例如：

${ user. My-Name}应当改为**${user["My-Name"]}**

如果要动态取值时，就可以用“[ ]“来做，而**“.“无法做到动态取值**。例如：

${sessionScope.user[data]}中data 是一个变量

**动态取值指，[]可以取第几个，而.只能取第一个，而且data可以是name，可以是sex**

**3、变量**

EL取变量数据的方法很简单，例如：${username}。它的意思是取出**某一范围中**名称为username的**变量**。

因为我们并没有指定哪一个范围的username，所以**它会依序从Page、Request、Session、Application范围查找。 从小到大**

假如途中找到username，就直接回传，不再继续找下去，但是假如全部的范围都没有找到时，就回传null。

JSP 表达式语言定义可在表达式中使用的以下文字：

|  |
| --- |
|  |
| true 和 false |
| 与 Java 类似。可以包含任何正数或负数，例如 24、-45、567 |
| 与 Java 类似。可以包含任何正的或负的浮点数，例如 -1.8E-45、4.567 |
| 任何由单引号或双引号限定的字符串。对于单引号、双引号和反斜杠，使用反斜杠字符作为转义序列。必须注意，如果在字符串两端使用双引号，则单引号不需要转义。 |
| null |

**2、操作符**

JSP 表达式语言提供以下操作符，其中大部分是 Java 中常用的操作符：

|  |  |
| --- | --- |
| **术语** | **定义** |
| 算术型 | +、-（二元）、\*、/、div、%、mod、-（一元） |
| 逻辑型 | and、&&、or、||、!、not |
| 关系型 | ==、eq、!=、ne、<、lt、>、gt、<=、le、>=、ge。可以与其他值进行比较，或与布尔型、字符串型、整型或浮点型文字进行比较。 |
| 空 | empty 空操作符是前缀操作，可用于确定值是否为空。 |
| 条件型 | A ?B :C。根据 A 赋值的结果来赋值 B 或 C。 |

**3、隐式对象**

像JSP一样，EL也自己定义了一组11个**隐式对象**，其中许多对象在 JSP scriplet 和表达式中可用：

|  |  |
| --- | --- |
| **pageContext** | JSP 页的上下文。它可以**用于访问 JSP 隐式对象**，如**请求**、**响应**、**会话**、**输出**、**servletContext** 等。例如，  **${pageContext.response} 为页面的响应对象赋值。** |

此外，还提供下列几个**隐式对象**，允许**对**以下对象进行简易访问：

相当于用一个变量表示了一个方法的调用的返回值

|  |  |
| --- | --- |
| **术语** | **定义** |
| **param** | 将**请求参数**名称映射到**单个字符串参数值**（通过调用 ServletRequest.getParameter (String name) 获得）。getParameter (String) 方法返回带有特定名称的参数。  表达式  ${**param** . name}相当于 **request.getParameter** (name)。 |
| **paramValues** | 将**请求参数**名称映射到一个**数值数组**（通过调用 ServletRequest.getParameter (String name) 获得）。它与 param 隐式对象非常类似，但它检索一个**字符串数组**而不是单个值。表达式  ${**paramvalues**. name} 相当于 **request.getParamterValues**(name)。 |
| **header** | 将**请求头**名称映射到单个字符串头值（通过调用 ServletRequest.getHeader(String name) 获得）。  表达式  ${header. name} 相当于 **request.getHeader**(name)。 |
| **headerValues** | 将**请求头**名称映射到一个数值数组（通过调用 ServletRequest.getHeaders(String) 获得）。它与头隐式对象非常类似。  表达式  ${headerValues. name} 相当于 **request.getHeaderValues**(name)。 |
| **cookie** | 将 cookie 名称映射到单个 cookie 对象。向服务器发出的客户端请求可以获得一个或多个 cookie。  表达式 ${cookie. name .value} 返回带有特定名称的第一个 cookie 值。如果请求包含多个同名的 cookie，则应该使用 ${headerValues. name} 表达式。 |
| **initParam** | 将上下文**初始化参数名称**映射到单个值（通过**调用 ServletContext.getInitparameter(String name**) 获得）。  ${initParam.name} |

除了上述两种类型的隐式对象之外，还有些对象允许**访问多种范围的变量**，如 **Web 上下文**、**会话**、**请求**、**页面**：（其实就是访问属性）

|  |  |
| --- | --- |
| **术语 从小到大** | **定义** |
| pageScope | 将**页面范围**的变量名称映射到其值。例如，EL 表达式可以使用 ${pageScope.objectName} 访问一个 **JSP 中页面范围的对象**，还可以使用  ${pageScope .objectName. attributeName} **访问对象的属性**。 |
| requestScope | 将**请求范围**的变量名称映射到其值。该对象**允许访问请求对象的属性**。例如，EL 表达式可以使用  ${requestScope. objectName} 访问一个 **JSP 请求范围的对象**，  还可以使用  ${requestScope. objectName. attributeName} **访问对象的属性**。 |
| sessionScope | 将**会话范围**的变量名称映射到其值。该对象**允许访问会话对象的属性**。例如：  ${sessionScope. name} |
| applicationScope | 将**应用程序范围**的变量名称映射到其值。该隐式对象允许访问应用程序范围的对象。 |

特别强调

[编辑](javascript:;)

1、注意当表达式根据名称引用这些对象之一时，**返回的是相应的对象**而不是相应*的属性*。例如：即使现有的 pageContext 属性包含某些其他值，${pageContext} 也返回 PageContext 对象。

2、 注意 <%@ page isELIgnored="true" %> 表示是否禁用EL语言,TRUE表示禁止.FALSE表示不禁止.JSP2.0中默认的启用EL语言。

举例说明

[编辑](javascript:;)

1、例如，

< %=request. getParameter(“username”)% > 等价于 ${ param. username }

2、例如，但是下面的那句EL语言可以完成如果得到一个username为空，则不显示null,而是不显示值。

[](http://baike.baidu.com/pic/EL%E8%A1%A8%E8%BE%BE%E5%BC%8F/1190845/0/b7fd5266d0160924ca51eaa3d40735fae6cd3415?fr=lemma&ct=single)

<%=user.getAddr( ) %> 等价于 ${user.addr}。

3、例如：

**<%=request.getAttribute(“userlist”) %> 等价于$ { requestScope.userlist }**

4、例如，原理如上例3。

${requestScope.userlist} 1

${sessionScope.userlist} 2

${applicationScope.userlist} 3

${pageScope.userlist} 4

**${userlist} 含义：执行顺序为4 1 2 3。**

“.”后面的只是一个字符串，并不是真正的内置对象，不能调用对象。

5、例如，

<%=user.getAddr( ) %> 等价于 ${user.addr}

第一句前面的user,为一个变量。

**第二句后面user**，必须为在**某一个范围里的属性**。

[](http://baike.baidu.com/pic/EL%E8%A1%A8%E8%BE%BE%E5%BC%8F/1190845/0/cdbf6c81800a19d89313ccbd33fa828ba61e4671?fr=lemma&ct=single)