# EL函数

**目的：**

    使用EL函数可以使JSP更加容易阅读。简化JSP文件中的数据访问。

**使用步骤：**

**EL函数实际上是与JAVA类中的静态方法相对应的**。EL不是一种新的语言只是JSP2.0引入的一个新特性。

    1.创建对应的Java类代码。

      对该java类有两点要求：a.这个类必需为public类型；b.被EL使用的函数必须声明为public static.

    2.创建TLD（标签库描述符）文件中，把Java类的方法映射为函数

    3.在Web.xml中声名标签所在位置（可选）

      如果在jsp文件中使用<%@ taglib prefix="xxx" uri="xxx" %>，即将uri的值直接指向TLD文件所在文置就可以跳过该步骤。（详细看后面示例）

**示例：**

    该示例按上述三个步骤逐步演示，并在第三步的时候分别演示直接使用与在Web.xml中定义的区别，请看代码。

*1.创建对应的Java类代码*

 该类在JSP中直接引用：

package com.dyp.elsample;

public class FirstELTest {  
   
 public static String getHello()  
 {  
  return "Hello World!--FirstELTest";  
 }  
   
}

 该类通过Web.xml声名后再在JSP中引用：

package com.dyp.elsample;

public class SecondELTest {  
 public static String getHello()  
 {  
  return "Hello World!--SecondELTest";  
 }  
}  
*2.创建TLD（标签库描述符）*

FirstELTest 对应的TLD

<taglib xmlns="<http://java.sun.com/xml/ns/j2ee>"  
  xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"  
  xsi:schemaLocation="<http://java.sun.com/xml/ns/javaee>   
         <http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd>"  
  version="2.5">  
      
  <description>JSTL 1.1 functions library</description>  
  <display-name>JSTL functions first</display-name>  
  <tlib-version>1.1</tlib-version>  
  <short-name>first</short-name>  
  <uri>http://java.sun.com/jsp/jstl/functionfirst</uri>

  <function>  
    <description>get hello word info</description>  
    <name>getHello</name>  
    <function-class>com.dyp.elsample.FirstELTest</function-class>  
    <function-signature>java.lang.String getHello()</function-signature>  
    <example>${first:getHello()}</example>    
  </function>  
    
</taglib>

SecondELTest 对应的TLD  
<taglib xmlns="<http://java.sun.com/xml/ns/j2ee>"  
  xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"  
  xsi:schemaLocation="<http://java.sun.com/xml/ns/javaee>   
         <http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd>"  
  version="2.5">  
      
  <description>JSTL 1.1 functions library</description>  
  <display-name>JSTL functions second</display-name>  
  <tlib-version>1.1</tlib-version>  
  <short-name>second</short-name>  
  <uri>http://java.sun.com/jsp/jstl/functionsecond</uri>

  <function>  
    <description>get hello word info</description>  
    <name>getHello</name>  
    <function-class>com.dyp.elsample.SecondELTest</function-class>  
    <function-signature>java.lang.String getHello()</function-signature>  
    <example>${second:getHello()}</example>    
  </function>  
    
</taglib>

*3.在Web.xml中声名标签所在位置*

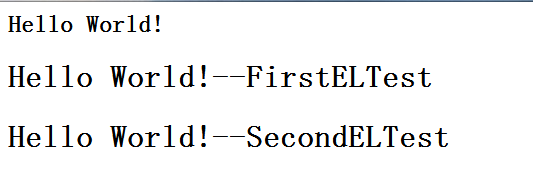
这里只声名了SecondELTest。

<web-app xmlns="<http://java.sun.com/xml/ns/javaee>"   
    xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"  
 xmlns:web="<http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd>"   
 xsi:schemaLocation="<http://java.sun.com/xml/ns/javaee>   
         <http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd>"   
 version="2.5">  
   
  <display-name>EL SAMPLE</display-name>  
    
  <jsp-config>  
    <taglib>  
        <taglib-uri>http://java.sun.com/jsp/jstl/functionsecond</taglib-uri>  
        <taglib-location>/WEB-INF/tlds/second.tld</taglib-location>  
    </taglib>  
  </jsp-config>  
    
</web-app>

*4.index.jsp代码*  
<%@ taglib prefix="first" uri="/WEB-INF/tlds/first.tld" %>   
<%@ taglib prefix="second" uri="<http://java.sun.com/jsp/jstl/functionsecond>" %>

<html>  
<body>  
<h2>Hello World!</h2>  
<h1>${first:getHello()}</h1>  
<h1>${second:getHello()}</h1>  
</body>  
</html>

注：注意标红的区别！！  
5.结果

[](http://album.sina.com.cn/pic/0029VBP6gy6Gs7xJ1mH4b)

# Tag文件

**tag文件只是以tag为后缀名的文本文件。除了jsp页面指令外，其他JSP元素都可以出现在tag文件中**

**页面引用格式**

<%@ taglib prefix="ui" tagdir="/WEB-INF/tags" %>

**tagdir：用于指定tag文件目录，当页面使用<ui:xxxx>进，会查找该目录下对应的xxxx.tag文件。**

**prefix：指定使用时标签前缀**

**使用格式**

<ui:xxxx></ui:xxxx>

例子：<ui:tagDemo><ui:tagDemo>

tag文件添加属性：当tag文件需要引用页面传入参数时，就需要在tag文件中填加属性

**定义属性格式**

<%@ attribute name="attributename" required="true" type="com.myapp.util.ListPage" %>

name(必须)：属性名

required(必须)：指定是否必须传

type(可选)：指定属性类型。

**tag文件获得传入参数值**

String attributename=(String) pageContext.getAttribute("attributename");

或者在jsp元素中使用${pageScope.attributename}

也可使用<jsp:doBody/> 获取引用页面标签内的body内容。

示例

将MyTag.tag放入WEB-INF/tags目录下

在index.jsp中用<%@ taglib prefix="tagfile" tagdir="/WEB-INF/tags" %>引入。其内空如下

<%@ tag language=*"java"* pageEncoding=*"UTF-8"*%>

<%@ attribute name=*"id"* required=*"true"* %>

<%@ attribute name=*"webletID"* required=*"true"* %>

<B>JSP TAG ATTRIBUTE GET</B>

<div id=*"*<%=id %>*"* style="border:*solid 1px #ff0000*">

Here, got the attribute id value :<%=id %>

<br>Got the attribute webletID value :<%=webletID %>

</div>

在index.jsp中使用方式为<tagfile:MyTag id="007" webletID="W1001"/> 因为atrribute的required全部要求为true所以这里的两个参数不能省

运行结果为

