



Java程序设计

第04-02讲 表达式语言 (EL)

武永亮

wuyongliang@edu2act.org



知识回顾/本讲先行知识



- 静态包含
- 动态包含
 - include动作指令
 - 静态包含与动态包含的比较
 - forward动作指令
- 使用JavaBean组件





本讲教学目标



- 掌握并灵活运用EL(表达式语言)





本讲内容

- 表达式语言简介
- 表达式语言运算符
- 表达式语言函数



EL简介



- EL(Expression Language): 表达式语言
- 在JSP2.0之后, EL正式纳入标准, 只要是支持Servlet2.4/JSP2.0的容器, 可以直接在JSP页面中使用EL
- EL语法简单, 使用方便, 在很多地方可代替JSP标签
- 输出page、request、session、application范围的数据



示例



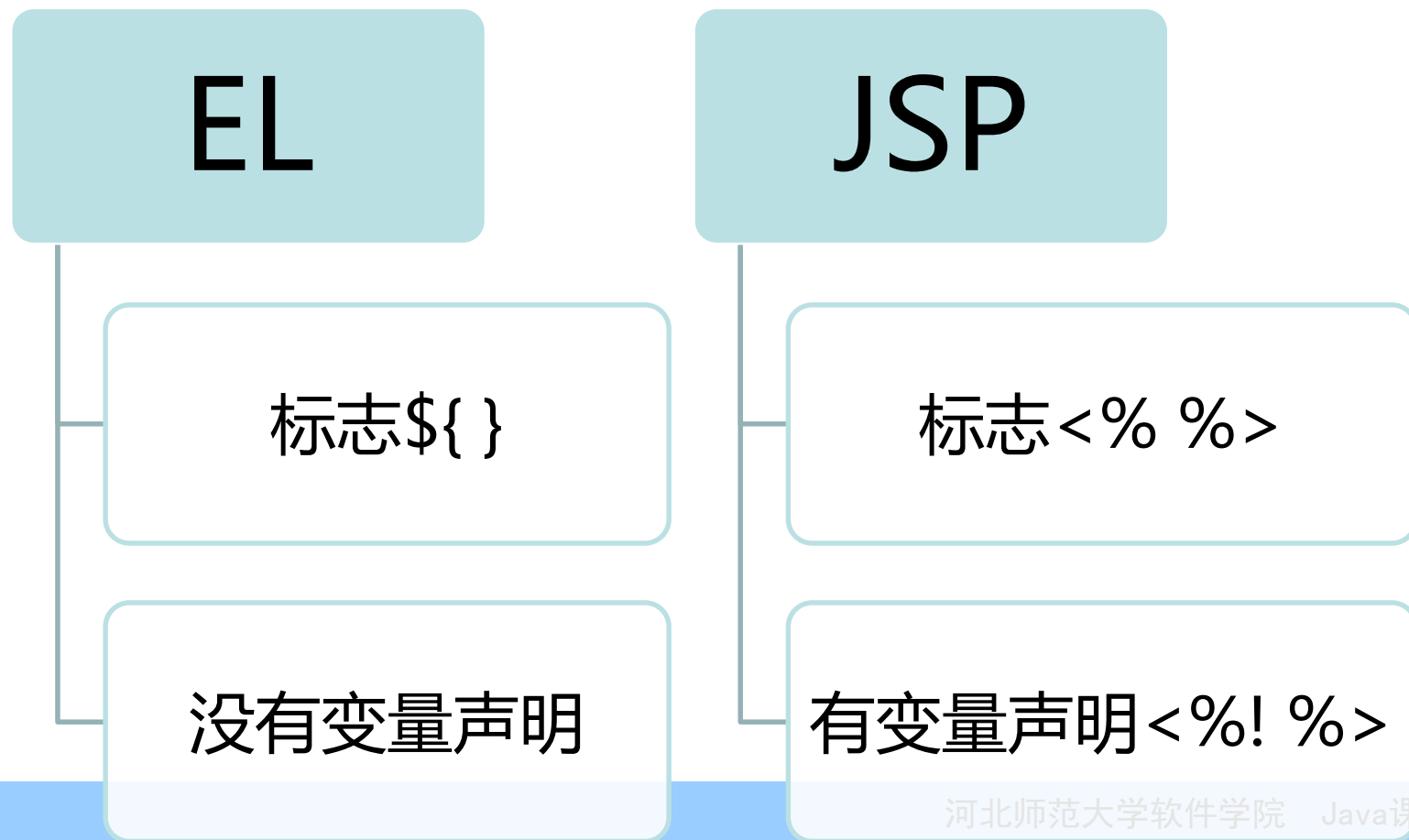
- JSP表达式的代码：
 - The outside temperature is `<%= temp%>` degrees
- 表达式语言(EL)表达式的代码：
 - The outside temperature is `${temp}` degrees



表达式语言(EL)介绍



- EL是一个完全独立的拥有操作符、语法和保留字的编程语言



EL隐含对象



- EL隐含对象有11个，与JSP的9个隐含对象区分开
- 与范围有关的
 - applicationScope:
 - sessionScope:
 - requestScope:
 - pageScope:
- 与输入有关的
 - param:
 - paramValues:
- 其他隐含对象
 - cookie:
 - header:
 - headerValues:
 - initParam:
 - pageContext

通过pageContext对象可以访问到 application、session、request 等对象，如：

```
${pageContext.out.bufferSize}
```

```
${pageContext.request.method}
```

但是，表达式限制开发人员调用Java方法，所以下面的表示是非法的：

```
${pageContext.request.getMethod()}
```

但用JSP表达式是合法的：

```
<%= request.getMethod()%>
```



EL中的隐含对象



内置对象	描述
pageContext	用于访问JSP内置对象
pageScope	代表page作用域对象
requestScope	代表request作用域对象
sessionScope	代表session作用域对象
applicationScope	代表application作用域对象
param	代表请求参数对象
paramValues	代表所有请求参数对象
header	代表请求头对象
headerValues	代表所有请求头对象
cookie	代表Cookie对象
initParam	相当于ServletContext.getInitParameter(String name)



EL变量



- EL存取变量数据的方法很简单，如`${user}`
 - 意思是取出某一范围中名称为user的变量
- 默认的范围查找顺序(自动搜索顺序)
 - Page → Request → Session → Application
- 也可以指定范围
 - `${pageScope.user}`
 - `${requestScope.user}`
 - `${sessionScope.user}`
 - `${applicationScope.user}`



EL保留字



- 变量命名不得使用EL保留字，以免程序编译时发生错误

and	eq	gt	true
or	ne	le	false
no	lt	ge	null
instanceof	empty	div	mod





本讲内容

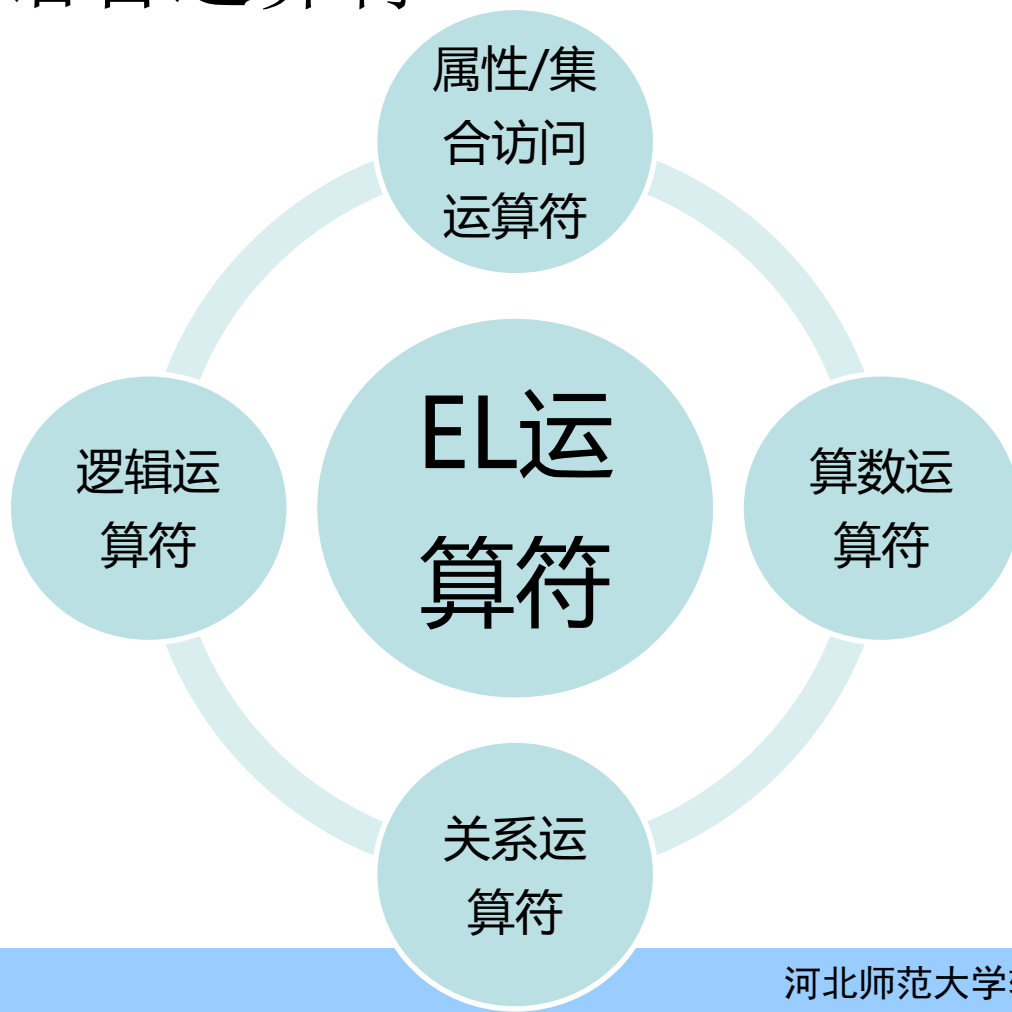
- 表达式语言简介
- 表达式语言运算符
- 表达式语言函数



运算符



- 表达式语言运算符





属性/集合访问运算符

- 属性访问运算符.
 - 例如a.b代表a对象的名为b的属性
 - `${header.host}`
- 集合访问运算符[]
 - 例如a[b]代表集合a的关键字为b所指向的元素
 - `${header["host"]}`



.与[]运算符



- 有时两者所代表的意思是一样的，如
 - `${sessionScope.user.sex}` 等价于 `${sessionScope.user["sex"]}`
- 两者也可混合使用，如
 - `${sessionScope.shoppingCart[0].price}`
- 以下两种情况，两者会有差异
 - 当要存取的属性名称中包含一些特殊字符，象“.”或“-”等并非字母或数字的符号，一定要用[]，如
 - `${user.My-Name}`应写为`${user["My-Name"]}`
 - 如想动态的取值时，也要使用[]，如
 - `${user[data]}`，data相当于一个变量



算术运算符



运算符名称	符号
加运算	+
减运算	-
乘运算	*
除运算	div和/
取余运算	mod和%

Demo 04-02示例



关系与逻辑运算符



关系运算	运算符	注意
相等	<code>==</code> 和 <code>eq</code>	使用EL关系运算符， 不可以写成： <code>\${a.pass}==\${a.pass2}</code> 或 <code>\${ \${a.pass}==\${a.pass2} }</code> 应该写成 <code>\${a.pass == a.pass2}</code> 示例：Demo 04-02示例
不相等	<code>!=</code> 和 <code>ne</code>	
小于	<code><</code> 和 <code>lt</code>	
大于	<code>></code> 和 <code>gt</code>	
小于或等于	<code><=</code> 和 <code>le</code>	
大于或等于	<code>>=</code> 和 <code>ge</code>	

逻辑运算	运算符
逻辑与	<code>&&</code> 和 <code>and</code>
逻辑或	<code> </code> 和 <code>or</code>
逻辑非	<code>!</code> 和 <code>not</code>



其他运算符



- empty: 判断值是否为null或空的
 - `${ empty param.name}`
- 条件运算符: 三元运算符, `A? B:C`
 - 也就是当A为true时, 执行B, 否则执行C
 - `${A? B:C}`
- 括号运算符: 用来改变执行优先权
 - `${A *(B+C)}`





本讲内容

- 表达式语言简介
- 表达式语言运算符
- 表达式语言函数



自定义EL函数



- 自定义EL语言函数需要包括创建、修改以下四个文件

1、编写方法所在源文件 .java文件

- 该文件包含EL函数实现

2、编写标签库描述符 .tld文件

- 定义EL函数名

3、修改部署描述符 web.xml文件

- 定义标签库描述符地址

4、编写JSP文件 .JSP文件

- 使用EL函数



示例



- 编写一个EL函数，接收一个字符串参数，函数的返回结果是字符串大写
- 编写一个EL函数，输入一个字符串，返回该字符串的长度，如：输入abc，输出3



1、编写方法所在源文件



- 方法声明为 **public**、**static**，不需要创建类的实例就可以访问方法
- 方法的返回值和参数必须是EL内有效的



1、编写方法所在源文件



```
public class StrMethods {  
    public static String upper(String x) {  
        return x.toUpperCase();  
    }  
  
    public static int length(String x) {  
        return x.length();  
    }  
}
```



2、编写标签库描述符



- `<function>` `</function>` 必须成对出现
- `<name>` `</name>` 用于指定EL语言函数名称
- `<function-class>` `</function-class>` 用于指定Java方法所在的类名
- `<function-signature>` `</function-signature>` 用于指定静态方法及其参数和返回类型
- tld文件通常放在/WEB-INF子目录下



2、编写标签库描述符



```
<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/j2eeweb-
jsptaglibrary_2_0.xsd"
version="2.0">
  <tlib-version>1.0</tlib-version>
  <function>
    <name>upper</name>
    <function-class>com.StrMethods</function-class>
    <function-signature>java.lang.String upper(java.lang.String)
    </function-signature>
  </function>
  <function>
    <name>length</name>
    <function-class>com.StrMethods</function-class>
    <function-signature>java.lang.int length(java.lang.String)
    </function-signature>
  </function>
</taglib>
```



3、修改部署描述符



- web.xml文件中<web-app> </web-app>节点内加入jsp-config 标签, 然后在其中加入<taglib> </taglib> 标签(新版本的Servlet规范中<taglib> 标签被包含在<jsp-config> 标签内)
- <taglib-uri> </taglib-uri> 用于描述JSP文件中使用的标签库描述符的URI地址
- <taglib-location> </taglib-location> 用于描述标签库描述符的实际物理位置



3、修改部署描述符



```
<jsp-config>
  <taglib>
    <!-- 配置标签的引用地址 JSP页面中引用时使用-->
    <taglib-uri>/Functions</taglib-uri>
    <!-- 配置标签的TLD文件地址 -->
    <taglib-location>
      /WEB-INF/Functions.tld
    </taglib-location>
  </taglib>
</jsp-config>
```



4、编写JSP文件调用EL函数



- 使用taglib指令访问tld文件，并添加一个代表标签库的前缀名
- 使用tld前缀和函数名创建EL的表达式,使用正确数据类型参数



4、编写JSP文件调用EL函数



```
<%@ taglib uri="/Functions" prefix="el" %>
```

```
...
```

```
<body>
```

```
    ${el:upper(param.userName)}
```

```
</body>
```



小结



- 表达式语言简介
- 表达式语言运算符
- 表达式语言函数



作业



- 编写一个EL函数，接收两个int型参数，函数的返回结果是两参数的和
- 编写一个EL函数，输入一个字符串，返回该字符串的逆序字符排列，如：输入abc，输出cba



本讲结束

- 谢谢大家

