目录

[一、 调用说明 3](#_Toc295900186)

[1. 初始化 3](#_Toc295900187)

[2. 设置属性 3](#_Toc295900188)

[3. 添加自定义标签 3](#_Toc295900189)

[二、 运算符号 3](#_Toc295900190)

[1. is 4](#_Toc295900191)

[2. isnot 4](#_Toc295900192)

[3. and 4](#_Toc295900193)

[4. or 4](#_Toc295900194)

[5. lt 5](#_Toc295900195)

[6. lte 5](#_Toc295900196)

[7. gt 5](#_Toc295900197)

[8. gte 5](#_Toc295900198)

[三、 内置函数 5](#_Toc295900199)

[1. equals 5](#_Toc295900200)

[2. notequals 6](#_Toc295900201)

[3. iseven 6](#_Toc295900202)

[4. isodd 6](#_Toc295900203)

[5. isempty 6](#_Toc295900204)

[6. isnotempty 7](#_Toc295900205)

[7. isnumber 7](#_Toc295900206)

[8. toupper 7](#_Toc295900207)

[9. tolower 7](#_Toc295900208)

[10. isdefined 7](#_Toc295900209)

[11. ifdefined 8](#_Toc295900210)

[12. length 8](#_Toc295900211)

[13. tolist 8](#_Toc295900212)

[14. isnull 9](#_Toc295900213)

[15. not 9](#_Toc295900214)

[16. iif 9](#_Toc295900215)

[17. format 9](#_Toc295900216)

[18. trim 10](#_Toc295900217)

[19. filter 10](#_Toc295900218)

[20. gt 10](#_Toc295900219)

[21. lt 10](#_Toc295900220)

[22. compare 11](#_Toc295900221)

[23. or 11](#_Toc295900222)

[24. and 11](#_Toc295900223)

[25. comparenocase 11](#_Toc295900224)

[26. stripnewlines 12](#_Toc295900225)

[27. typeof 12](#_Toc295900226)

[28. cint 12](#_Toc295900227)

[29. cdouble 12](#_Toc295900228)

[30. cdate 13](#_Toc295900229)

[31. cdatestring 13](#_Toc295900230)

[32. createtypereference 13](#_Toc295900231)

[四、 结构控制标签 13](#_Toc295900232)

[1. if 13](#_Toc295900233)

[2. foreach 15](#_Toc295900234)

[3. for 15](#_Toc295900235)

[4. set 16](#_Toc295900236)

[五、 自定义模板 17](#_Toc295900237)

[1. 内置自定义模板 17](#_Toc295900238)

[六、 自定义标签 17](#_Toc295900239)

[1. cut 17](#_Toc295900240)

[2. include 17](#_Toc295900241)

[3. removetag 18](#_Toc295900242)

# 调用说明

## 初始化

String \_templateContent="${title}";

AderTemplates.TemplateManager manager = AderTemplates.TemplateManager.FromString(\_templateContent);

//设置公用的对象

manager.SetValue("title","测试标题" );

//输出结果

Response.Write(manager.Process());

## 设置属性

1. **方法声明**

public void SetValue(string name, object value)

1. **参数说明**

**name:属性名称,在模板代码中将使用到该属性名称做输出名称**

**value:该属性对应的值，可以是任何类型**

1. **调用示例**

manager.SetValue("title","测试标题" );

或

manager.SetValue("user", new ArrayList { "张三", "李四", "王五" });

## 添加自定义标签

# 运算符号

## is

|  |  |  |  |
| --- | --- | --- | --- |
| **obj1 is obj2** | | | |
| **说明** | 测试obj1 和 obj2 是否相等. 返回boolean值. | | |
| **参数说明** |  | | |
|  | **obj1** | **任意类型** | **比较条件之一** |
|  | **obj2** | **任意类型** | **比较条件之一** |
| **调用示例** | ${1 is 1} 返回则为真，因为1等于1 | | |

## isnot

|  |  |  |  |
| --- | --- | --- | --- |
| **obj1 isnot obj2** | | | |
| **说明** | 测试obj1 和 obj2 是否不相等. 返回boolean值. | | |
| **参数说明** |  | | |
|  | **obj1** | **任意类型** | **比较条件之一** |
|  | **obj2** | **任意类型** | **比较条件之一** |
| **调用示例** | ${1 isnot 2}返回则为真，因为1不等于2 | | |

## and

|  |  |  |  |
| --- | --- | --- | --- |
| **expression1 and expression2** | | | |
| **说明** | 条件连接符号,并且 | | |
| **参数说明** |  | | |
|  | **无** |  |  |
| **调用示例** | ${equals(1,1) and equals(1,2)} 返回则为假，因为第一个条件中1等于1，而第二个条件中1不等于2，两个条件使用 and 进行连接；其中任何一个条件不满足则返回假 | | |

## or

|  |  |  |  |
| --- | --- | --- | --- |
| **expression1 or expression2** | | | |
| **说明** | 条件连接符号，或者 | | |
| **参数说明** |  | | |
|  | **无** |  |  |
| **调用示例** | ${equals(1,1) or equals(1,2)} 返回则为真；其中任何一个条件满足则返回真 | | |

## lt

|  |  |  |  |
| --- | --- | --- | --- |
| **num1 lt num2** | | | |
| **说明** | 相当于小于号,比较lt之前的值是否小于lt之后的值. | | |
| **参数说明** |  | | |
|  | **无** |  |  |
| **调用示例** | ${1 lt 2}，判断1是否小于2 | | |

## lte

|  |  |  |  |
| --- | --- | --- | --- |
| **num1 lte num2** | | | |
| **说明** | 相当于小于等于号,比较lte之前的值是否小于并等于lte之后的值. | | |
| **参数说明** |  | | |
|  | **无** |  |  |
| **调用示例** | ${1 lte 2}，判断1是否小于并等于2 | | |

## gt

|  |  |  |  |
| --- | --- | --- | --- |
| **num1 gt num2** | | | |
| **说明** | 相当于大于号,比较gt之前的值是否大于gt之后的值. | | |
| **参数说明** |  | | |
|  | **无** |  |  |
| **调用示例** | ${1 gt 2}，判断1是否大于2 | | |

## gte

|  |  |  |  |
| --- | --- | --- | --- |
| **num1 gte num2** | | | |
| **说明** | 相当于大于等于号,比较gte之前的值是否大于并等于gte之后的值. | | |
| **参数说明** |  | | |
|  | **无** |  |  |
| **调用示例** | ${1 gte 2}，判断1是否大于并等于2 | | |

# 内置函数

## equals

|  |  |  |  |
| --- | --- | --- | --- |
| **equals(obj1,obj2)** | | | |
| **说明** | 测试obj1 和 obj2 是否相等. 返回boolean值. | | |
| **参数说明** |  | | |
|  | **obj1** | **任意类型** | **比较条件之一** |
|  | **obj2** | **任意类型** | **比较条件之一** |
| **调用示例** | ${equals(1,2 )} 返回则为假，因为1不等于2 | | |

## notequals

|  |  |  |  |
| --- | --- | --- | --- |
| **notequals(obj1,obj2)** | | | |
| **说明** | 测试obj1 和 obj2 是否不相等. 返回boolean值. | | |
| **参数说明** |  | | |
|  | **obj1** | **任意类型** | **比较条件之一** |
|  | **obj2** | **任意类型** | **比较条件之一** |
| **调用示例** | ${notequals(1,2 )} 返回则为真，因为1不等于2 | | |

## iseven

|  |  |  |  |
| --- | --- | --- | --- |
| **iseven(num)** | | | |
| **说明** | 测试值是否为偶数. | | |
| **参数说明** |  | | |
|  | **num** | **数值型** |  |
| **调用示例** | ${iseven(2)} 返回则为真，因为2为偶数 | | |

## isodd

|  |  |  |  |
| --- | --- | --- | --- |
| **isodd(num)** | | | |
| **说明** | 测试值是否为奇数. | | |
| **参数说明** |  | | |
|  | **num** | **数值型** |  |
| **调用示例** | ${isodd(2)} 返回则为假，因为2为偶数 | | |

## isempty

|  |  |  |  |
| --- | --- | --- | --- |
| **isempty(str)** | | | |
| **说明** | 测试 string 长度是否等于0. 相当于：equals(string.Length, 0) . | | |
| **参数说明** |  | | |
|  | **str** | **字符型** |  |
| **调用示例** | ${isempty(“abc”)} 返回则为假，因为”abc”的长度为3 | | |

## isnotempty

|  |  |  |  |
| --- | --- | --- | --- |
| **isnotempty(str)** | | | |
| **说明** | 测试 string 长度是否大于0. | | |
| **参数说明** |  | | |
|  | **str** | **字符型** |  |
| **调用示例** | ${isnotempty(“abc”)} 返回则为真，因为”abc”的长度为3 | | |

## isnumber

|  |  |  |  |
| --- | --- | --- | --- |
| **isnumber(num)** | | | |
| **说明** | 测试 num 是否为数值型. | | |
| **参数说明** |  | | |
|  | **num** | **数值型** |  |
| **调用示例** | ${isnumber(2 )} 返回则为真，因为2为数值型 | | |

## toupper

|  |  |  |  |
| --- | --- | --- | --- |
| **toupper(str)** | | | |
| **说明** | 将str中的英文字符转换成大写. | | |
| **参数说明** |  | | |
|  | **str** | **字符型** |  |
| **调用示例** | ${toupper(“abc” )} 结果为ABC | | |

## tolower

|  |  |  |  |
| --- | --- | --- | --- |
| **tolower(str)** | | | |
| **说明** | 将str中的英文字符转换成小写 | | |
| **参数说明** |  | | |
|  | **str** | **字符类型** |  |
| **调用示例** | ${tolower(“ABC” )} 结果为abc | | |

## isdefined

|  |  |  |  |
| --- | --- | --- | --- |
| **isdefined(varname)** | | | |
| **说明** | 测试 varname 是否已经定义类型. | | |
| **参数说明** |  | | |
|  | **varname** | **字符型** |  |
| **调用示例** | ${isdefined(“username”)} | | |

## ifdefined

|  |  |  |  |
| --- | --- | --- | --- |
| **ifdefined(varname,value)** | | | |
| **说明** | 如果varname已经定义就得到value，否则为null. 例如：${ifdefined("name", name)}. | | |
| **参数说明** |  | | |
|  | **varname** | **字符型** |  |
|  | **value** | **变量** |  |
| **调用示例** | ${ifdefined("name", name)} | | |

## length

|  |  |  |  |
| --- | --- | --- | --- |
| **length(str)** | | | |
| **说明** | 获取str的长度. | | |
| **参数说明** |  | | |
|  | **str** | **字符型** |  |
| **调用示例** | ${length(“abc” )} 返回则为3 | | |

## tolist

|  |  |  |  |
| --- | --- | --- | --- |
| **tolist(collection, property, delim)** | | | |
| **说明** | 组合集合对象中的列值，使用delim作为分隔符 | | |
| **参数说明** |  | | |
|  | **collection** | **集合对象** |  |
|  | **property** | **字符型** |  |
|  | **delim** | **字符型** |  |
| **调用示例** | 将集合对象中指定的列值使用指定的分隔符号进行组合;  示例一：  mylist 集合对象中包含 one,two,three  ${tolist(mylist, " & ")} 将输出 one & two & three  示例二：  mylist 集合对象中包含 one,two,three；one,two,three为对象类型，包含name,id属性  ${tolist(mylist,”name”, " & ")} 将输出 one&two&three | | |

## isnull

|  |  |  |  |
| --- | --- | --- | --- |
| **isnull(obj)** | | | |
| **说明** | 测试obj 是否为 null . 返回boolean值. | | |
| **参数说明** |  | | |
|  | **obj** | **object** |  |
| **调用示例** | ${notequals(1,2 )} 返回则为真，因为1不等于2 | | |

## not

|  |  |  |  |
| --- | --- | --- | --- |
| **not(boolvalue)** | | | |
| **说明** | 非运算符, boolvalue为true时返回false，boolvalue为false时返回true . | | |
| **参数说明** |  | | |
|  | **boolvalue** | **object** | **比较条件之一** |
| **调用示例** | ${not(equals(1,1))} 返回假 | | |

## iif

|  |  |  |  |
| --- | --- | --- | --- |
| **iif(expression,truevalue,falsevalue)** | | | |
| **说明** | 测试obj1 和 obj2 是否不相等. 返回boolean值. | | |
| **参数说明** |  | | |
|  | **expression** | **表达式** | 布尔型表达式 |
|  | **truevalue** | **任意类型** | **当表达式返回真时返回本值** |
|  | **falsevalue** | **任意类型** | **当表达式返回假时返回本值** |
| **调用示例** | ${iif(isodd(2), "bgcolor=yellow", "bgcolor=red")} 将输出 bgcolor=red | | |

## format

|  |  |  |  |
| --- | --- | --- | --- |
| **format(obj,formating)** | | | |
| **说明** | 对obj 按照formating的要求进行格式化 | | |
| **参数说明** |  | | |
|  | **obj** | **任意类型** | **要格式化的对象** |
|  | **formating** | **字符型** | **格式化格式**  具体值请参考: <http://msdn.microsoft.com/zh-cn/library/dwhawy9k.aspx> |
| **调用示例** | ${format(1208.451256, "C")}  将输出 $1,208.45 | | |

## trim

|  |  |  |  |
| --- | --- | --- | --- |
| **trim(str)** | | | |
| **说明** | 去除str 中的前后空格. | | |
| **参数说明** |  | | |
|  | **str** | **字符类型** |  |
| **调用示例** | ${trim(“ abc ”)} 将返回:abc | | |

## filter

|  |  |  |  |
| --- | --- | --- | --- |
| **filter** **(collection, boolproperty)** | | | |
| **说明** | 返回一个新的集合对象，其中包含旧的集合中boolproperty属性为真的记录 | | |
| **参数说明** |  | | |
|  | **collection** | **集合对象** |  |
|  | **boolproperty** | **字符型** | **结合对象中的属性** |
| **调用示例** | mylist 集合对象中包含 one,two,three  ${tolist(mylist, " & ")} 将输出 one & two & three; one,two,three为对象类型，包含name,id,checked属性  <#set name="newlist" value="${filter(mylist,”checked”)}" /> | | |

## gt

|  |  |  |  |
| --- | --- | --- | --- |
| **gt(obj1,obj2)** | | | |
| **说明** | 当obj1大于obj2时为true. | | |
| **参数说明** |  | | |
|  | **obj1** | **数值型** | **比较条件之一** |
|  | **obj2** | **数值型** | **比较条件之一** |
| **调用示例** | ${gt(1,2)}返回假 | | |

## lt

|  |  |  |  |
| --- | --- | --- | --- |
| **lt(obj1,obj2)** | | | |
| **说明** | 当obj1小于obj2时为true. | | |
| **参数说明** |  | | |
|  | **obj1** | **数值型** | **比较条件之一** |
|  | **obj2** | **数值型** | **比较条件之一** |
| **调用示例** | ${lt(1,2)}返回真 | | |

## compare

|  |  |  |  |
| --- | --- | --- | --- |
| **compare(obj1,obj2)** | | | |
| **说明** | 如果obj1 < obj2返回 -1； obj1 == obj2返回0； obj1 > obj2返回1 | | |
| **参数说明** |  | | |
|  | **obj1** | **数值型** | **比较条件之一** |
|  | **obj2** | **数值型** | **比较条件之一** |
| **调用示例** | ${compare(1,2 )} 返回-1 | | |

## or

|  |  |  |  |
| --- | --- | --- | --- |
| **or(obj1,obj2)** | | | |
| **说明** | 当obj1或obj2为true时为true . | | |
| **参数说明** |  | | |
|  | **obj1** | **布尔表达式** | **比较条件之一** |
|  | **obj2** | **布尔表达式** | **比较条件之一** |
| **调用示例** | ${or(equals(state, "IL"), equals(state, "NY"))},当state的值为IL或NY的时候则返回真 | | |

## and

|  |  |  |  |
| --- | --- | --- | --- |
| **and(obj1,obj2)** | | | |
| **说明** | 当obj1与obj2都为true时返回true，否则返回false | | |
| **参数说明** |  | | |
|  | **obj1** | **布尔表达式** | **比较条件之一** |
|  | **obj2** | **布尔表达式** | **比较条件之一** |
| **调用示例** |  | | |

## comparenocase

|  |  |  |  |
| --- | --- | --- | --- |
| **comparenocase(str1,str2)** | | | |
| **说明** |  | | |
| **参数说明** |  | | |
|  | **str1** | **字符型** | **比较条件之一** |
|  | **str2** | **字符型** | **比较条件之一** |
| **调用示例** |  | | |

## stripnewlines

|  |  |  |  |
| --- | --- | --- | --- |
| **stripnewlines(str)** | | | |
| **说明** |  | | |
| **参数说明** |  | | |
|  | **str** | **object** |  |
| **调用示例** |  | | |

## typeof

|  |  |  |  |
| --- | --- | --- | --- |
| **typeof(obj)** | | | |
| **说明** | 返回obj的值类型 | | |
| **参数说明** |  | | |
|  | **obj** | **任意类型** |  |
| **调用示例** | typeof("hello") 返回 "string". typeof(3) 返回 int | | |

## cint

|  |  |  |  |
| --- | --- | --- | --- |
| **cint(obj)** | | | |
| **说明** | 将obj转换成整型. | | |
| **参数说明** |  | | |
|  | **obj** | **任意类型** |  |
| **调用示例** | ${cint(“202”)} 返回202 | | |

## cdouble

|  |  |  |  |
| --- | --- | --- | --- |
| **cdouble(obj)** | | | |
| **说明** | 将 obj 转换成双精度类型. | | |
| **参数说明** |  | | |
|  | **obj** | **任意类型** |  |
| **调用示例** | ${cdouble(“1000.250”)}返回1000.25 | | |

## cdate

|  |  |  |  |
| --- | --- | --- | --- |
| **cdate(obj)** | | | |
| **说明** | 将obj 转换成日期型. | | |
| **参数说明** |  | | |
|  | **obj** | **任意类型** |  |
| **调用示例** |  | | |

## cdatestring

|  |  |  |  |
| --- | --- | --- | --- |
| **cdatestring(date,formating)** | | | |
| **说明** | 使用指定的string1格式将当前值格式化. 例如： ${cdatestring("2011-5-1 01:00:00","yy-MM-dd")} 将返回 11-05-01 | | |
| **参数说明** |  | | |
|  | **date** | **日期型** |  |
|  | **formating** | **字符型** |  |
| **调用示例** | ${cdatestring(art.CreatedTime,"yy-MM-dd")} | | |

## createtypereference

|  |  |  |  |
| --- | --- | --- | --- |
| **createtypereference(typestring)** | | | |
| **说明** | 引入新的类库到页面中 | | |
| **参数说明** |  | | |
|  | **typestring** | **字符型** | **必须是符合.net要求的命名空间,区分大小写；**  **如:System.Math** |
| **调用示例** | ${createtypereference("System.Math").Round(3.39789)}  ${createtypereference("System.Math").PI}  或  <#set name="MyMath" value="${createtypereference("System.Math")}" />  ${MyMath.Round(3.3)}  ${MyMath.PI} | | |
|  |  | | |
|  |  | | |

## substr

|  |  |  |  |
| --- | --- | --- | --- |
| **substr(str,num)** | | | |
| **说明** | 返回从0开始长度为num的子字符串 | | |
| **参数说明** |  | | |
|  | **str** | **字符型** |  |
|  | **num** | **整型** |  |
| **调用示例** | ${substr(art.Title,2)}  ${substr(“abscs”,5)} | | |
|  |  | | |
|  |  | | |

## stringtounicode

|  |  |  |  |
| --- | --- | --- | --- |
| **stringtounicode(str)** | | | |
| **说明** | 返回string的万国码 | | |
| **参数说明** |  | | |
|  | **str** | **字符型** |  |
| **调用示例** | ${stringtounicode(art.Title)}  ${ stringtounicode (“abscs”)} | | |
|  |  | | |
|  |  | | |

# 构控制标签

## if

|  |  |  |  |
| --- | --- | --- | --- |
| **if test=”${expression}”** | | | |
| **说明** | 根据一个表达式的值，有条件地执行一组语句。 | | |
| **语法** | <#if test="${booleanexpression1}">  //如果booleanexpression1为真则输出本段内容  <#elseif test="${booleanexpression2}">  //如果booleanexpression2为真则输出本段内容  <#else>  //如果以上条件都不成立则输出本段内容  <#/if>  注：  <#elseif test="${ booleanexpression2}">与<#else>为可选；  <#/if>为必须. | | |
|  | **booleanexpression1** | 表达式 | 布尔型表达式 |
|  | **booleanexpression2** | 表达式 | 布尔型表达式 |
| **调用示例** | 例一：  <#if test="${cust.country is "US"}">  该客户来至美国.  <#else>  该客户来至: ${cust.country}.  <#/if>  例二：  <#if test="${cust.country is "US"}">  该客户来至美国.  <#elseif test=”${cust.country is “CN”}”>  该客户来至中国.  <#else>  该客户来至 ${cust.country}.  <#/if>  例三：  <#if test="${cust.country is "US"}">  该客户来至美国.  <#/if> | | |

## foreach

|  |  |  |  |
| --- | --- | --- | --- |
| **foreach collection="${collection}" var="var" index="index"** | | | |
| **说明** | 对应于一个对象的每个，或一个数组的每个元素，执行一个或多个语句。 | | |
| **语法** | <#foreach collection="${collection}" var="item" index="index">  //循环输出段内容  <#/foreach> | | |
| **参数说明** | **collection** | 集合对象 | 对应页面中的集合对象，具体见附件 |
|  | **item** | 字符型 | 名称为自由定义，注意在嵌套代码中不重复即可 |
|  | **index** | 字符型 | 代表本次执行的索引，从0开始 |
| **调用示例** | 例一：输出所有客户集合对象中的客户信息  <#foreach collection="${customers}" var=”c” index=”rowindex”>  行号:${rowindex+1} 客户名:${c.name} 来至:${c.country}  <#/foreach>  例二：输出所有客户集合对象中的客户信息，隔行换色  <#foreach collection="${customers}" var=”c” index=”rowindex”>  <!--隔行换背景色-->  <div style=”background-color:${iif(isodd(rowindex),”red”,”green”)}”>  行号:${rowindex+1} 客户名:${c.name} 来至:${c.country}  </div>  <#/foreach> | | |

## for

|  |  |  |  |
| --- | --- | --- | --- |
| **for from="num1" to="num2" index="index"** | | | |
| **说明** | 执行指定次数的循环语句块。 | | |
| **语法** | <#for from="num1" var="num2" index="index">  //循环输出段内容  <#/for> | | |
| **参数说明** | **num1** | 数值型 | 循环开始 |
|  | **num2** | 数值型 | 循环结束，num2-num1=循环次数 |
|  | **index** | 字符型 | 代表本次执行的索引，起始值为from的值 |
| **调用示例** | 例一：输出前10个客户信息  <#for from="1" to=”10” index=”rowindex”>  行号:${rowindex} 客户名:${customers[rowindex].name } 来至:${customers[rowindex].country}  <#/for>  例二：输出前10个客户信息,隔行换色  <#for from="1" to=”10” index=”rowindex”>  <div style=”background-color:${iif(isodd(rowindex),”red”,”green”)}”>  行号:${rowindex+1} 客户名:${customers[rowindex].name} 来至:${customers[rowindex].country}  </div>  <#/foreach> | | |

## set

|  |  |  |  |
| --- | --- | --- | --- |
| **set name="name" value="expression"** | | | |
| **说明** | 执行指定次数的循环语句块。 | | |
| **语法** | <#set name="name" value=” expression" /> | | |
| **参数说明** | **name** | 字符型 | 要创建的存储变量名称，之后的代码可以通过本名称进行引用 |
|  | **expression** | 表达式或常量 | 要赋予该存储变量的值，计算表达式或常量 |
| ***调用示例*** | *例一：存储第一个客户信息，方便之后调用*  *<#for from=”1” to=”1” index=”i>*  *<#set name=”customername” value=”c.name” />*  *<#/for>*  *<div>第一个客户名为:${customername}</div>* | | |

# 自定义模板

## 内置自定义模板

|  |  |  |  |
| --- | --- | --- | --- |
| **template name="string"** | | | |
| **说明** | 声明自定义模板片段；允许内部嵌套子模板。${innerText}为固定变量名称，代表模板的围绕内容. | | |
| **语法** | <#template name="string">长度超过显示宽度的文本内容<#/template> | | |
| **参数说明** | **string** | 字符型 | 模板名称 |
| **调用示例** | 例一：加粗字体模板  <#template name="bold">  <b>${innerText}</b>  <#/template>  //模板使用代码  <#bold>${customer.name}, ${customer.country}<#/bold>  或  <#apply template="bold">需要加粗显示的内容<#/apply>  本例将输出加粗的客户名称与国家。  例二：嵌套模板使用演示  <#template name="doit">  //子模板声明  <#template name="colorme">  <font color=${color}>${innerText}</font>  <#/template>  //模板使用代码  <#colorme color="blue">colorize me<#/colorme>  <#/template> | | |

# 自定义标签

## cut

|  |  |  |  |
| --- | --- | --- | --- |
| **cut len="num"** | | | |
| **说明** | 对包含的内容进行裁剪，并输出指定长度。 | | |
| **语法** | <#cut len="num">长度超过显示宽度的文本内容<#/cut> | | |
| **参数说明** | **num** | 数值型 | 要裁剪的字符长度 |
| **调用示例** | 例一：如果客户名称长度超过10，超出的部分使用省略号进行表示  <#foreach collection=”${customers}” var=”c” index=”i>  <#cut len=”10” value=”c.name” />  <#/foreach> | | |

## include

|  |  |  |  |
| --- | --- | --- | --- |
| **include virtual="filepath"** | | | |
| **说明** | 在当前位置嵌入指定的本地文件或远程网页。 | | |
| **语法** | <#include virtual="filepath"” /> | | |
| **参数说明** | **filepath** | 字符型 | 要嵌入的文件路径；当source为web时其中的~代表本站域名 |
|  |  |  |  |
|  |  |  |  |
| **调用示例** | <#include virtual="D:\CMS\_V1\code\branches\v1.0\CMS.TemplateTest\template\include.txt" /> | | |

## removetag

|  |  |  |  |
| --- | --- | --- | --- |
| **removetag** | | | |
| **说明** | 移除标签的内容中的HTML元素。 | | |
| **语法** | <#removetag><b>测试</b><#/removetag> | | |
| **参数说明** | **无** |  |  |
| **调用示例** | 例一：去除标签中的所有html元素  <#removetag><b>测试</b><#/removetag>  将输出：测试 | | |