|  |  |
| --- | --- |
| |  | | --- | | **、基本语法** | |
| **1、"#"用来标识Velocity的脚本语句**，包括#set、#if 、#else、#end、#foreach、#end、#iinclude、#parse、#macro等； 如: #if($info.imgs) <img src="$info.imgs" border=0> #else <img src="noPhoto.jpg"> #end  **2、"$"用来标识一个对象(或理解为变量)；**如 如：$i、$msg、$TagUtil.options(...)等。  **3、"{}"用来明确标识Velocity变量；** 比如在页面中，页面中有一个$someonename，此时，Velocity将把someonename作为变量名，若我们程序是想在someone这 个变量的后面紧接着显示name字符，则上面的标签应该改成${someone}name。  **4、"!"用来强制把不存在的变量显示为空白。** 如当页面中包含$msg，如果msg对象有值，将显示msg的值，如果不存在msg对象同，则在页面中将显示$msg字符。这是我们不希望的，为了把不存 在的变量或变量值为null的对象显示为空白，则只需要在变量名前加一个“!”号即可。 如：$!msg |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | |  | | --- | | **二、在EasyJWeb中的最佳实践** | | | 理论上你可以在EasyjWeb模板使用所有Velocity的脚本及功能，但我们不推荐你在界面模板中使用过多过复杂的脚本表达方式，在万不得已的情况下，不要在界面模板中加入任何复杂的逻辑，更不要在界面模板中加入变量声明、逻辑运算符等等。  　　在EasyJWeb中，我们提供了五条基本的模板脚本语句，基本上就能满足所有应用模板的要求。这四条模板语句很简单，可以直接由界面设计人员来添加。在当前很多EasyJWeb的应用实践中，我们看到，所有界面模板中归纳起来只有下面四种简单模板脚本语句即可实现： **1、$!obj 　直接返回对象结果。** 如：在html标签中显示java对象msg的值。<p>$!msg</p> 在html标签中显示经过HtmlUtil对象处理过后的msg对象的值　　<p>$!HtmlUtil.doSomething($!msg)</p>  **2、#if($!obj) #else #end 判断语句** 如：在EasyJWeb各种开源应用中，我们经常看到的用于弹出提示信息msg的例子。 #if($msg) <script> alert('$!msg'); </script> #end 上面的脚本表示当对象msg对象存在时，输出<script>等后面的内容。  **3、#foreach( $info in $list) $info.someList #end**　　**循环读取集合list中的对象，并作相应的处理。** 如：EasyJF开源论坛系统中论(0.3)坛首页显示热门主题的html界面模板脚本： #foreach( $info in $hotList1)  <a href="/bbsdoc.ejf?easyJWebCommand=show&&cid=$!info.cid" target="\_blank">$!info.title</a><br> #end  上面的脚本表示循环遍历hotList1集合中的对象，并输出对象的相关内容。  **4、#macro(macroName)#end 脚本函数(宏)调用，不推荐在界面模板中大量使用。** 如：在使用EasyJWeb Tools快速生成的添删改查示例中，可以点击列表的标题栏进行升降排序显示，这是我们在EasyJWeb应用中经常看到的一个排序状态显示的模板内容。 函数(宏)定义，一般放在最前面 #macro(orderPic $type) #if ($orderField.equals($type))  <img src="http://images.cnblogs.com/ico/${orderType}.gif">  #end #end 具体的调用如：<font color="#FFFFFF">头衔#orderPic("title")</font>  经过测试，宏不支持方法重载    **5、包含文件#inclue("模板文件名")或#parse("模板文件名")** 主要用于处理具有相同内容的页面，比如每个网站的顶部或尾部内容。 使用方法，可以参考EasyJF开源Blog及EasyJF开源论坛中的应用！ 如：#parse("/blog/top.html")或#include("/blog/top.html") parse与include的区别在于，若包含的文件中有Velocity脚本标签，将会进一步解析，而include将原样显示。 | |  | |
| |  |  | | --- | --- | | |  | | --- | | **三、关于#set的使用** | | | 在万不得已的时候，不要在页面视图自己声明Velocity脚本变量，也就是尽量少使用#set。有时候我们需要在页面中显示序号，而程序对象中又没有包 含这个序号属性同，可以自己定义。如在一个循环体系中，如下所示： #set ($i=0) #foreach($info in $list) 序号:$i #set($i=$i+1) #end | |  | |
| |  |  | | --- | --- | | |  | | --- | | **四、Velocity脚本语法摘要** | | | **1、声明:#set ($var=XXX)** 左边可以是以下的内容 Variable reference  String literal  Property reference  Method reference  Number literal #set ($i=1)  ArrayList #set ($arr=["yt1","t2"]) 算术运算符  **2、注释:** 单行## XXX 多行#\* xxx xxxx xxxxxxxxxxxx\*#  References 引用的类型 **3、变量 Variables**  以 "$" 开头，第一个字符必须为字母。character followed by a VTL Identifier. (a .. z or A .. Z). 变量可以包含的字符有以下内容： alphabetic (a .. z, A .. Z)  numeric (0 .. 9)  hyphen ("-")  underscore ("\_")   **4、Properties**  $Identifier.Identifier $user.name hashtable user中的的name值.类似：user.get("name")  **5、Methods**  object user.getName() = $user.getName()  **6、Formal Reference Notation**  用{}把变量名跟字符串分开   如 #set ($user="csy"} ${user}name  返回csyname  $username $!username $与$!的区别 当找不到username的时候，$username返回字符串"$username"，而$!username返回空字符串""   **7、双引号 与 引号**  #set ($var="helo") test"$var" 返回testhello test'$var' 返回test'$var' 可以通过设置 stringliterals.interpolate=false改变默认处理方式  **8、条件语句** #if( $foo )  <strong>Velocity!</strong> #end #if($foo) #elseif() #else #end 当$foo为null或为Boolean对象的false值执行.  **9、逻辑运算符:== && || !**  **10、循环语句#foreach($var in $arrays )** // 集合包含下面三种Vector, a Hashtable or an Array #end #foreach( $product in $allProducts ) <li>$product</li> #end  #foreach( $key in $allProducts.keySet() ) <li>Key: $key -> Value: $allProducts.get($key)</li> #end  #foreach( $customer in $customerList ) <tr><td>$velocityCount</td><td>$customer.Name</td></tr> #end  **11、velocityCount变量在配置文件中定义** # Default name of the loop counter # variable reference. directive.foreach.counter.name = velocityCount # Default starting value of the loop # counter variable reference. directive.foreach.counter.initial.value = 1  **12、包含文件**  #include( "one.gif","two.txt","three.htm" )  **13、Parse导入脚本** #parse("me.vm" )  **14、#stop 停止执行并返回**   **15、定义宏Velocimacros ,相当于函数 支持包含功能** #macro( d ) <tr><td></td></tr> #end 调用  #d()  **16、带参数的宏** #macro( tablerows $color $somelist ) #foreach( $something in $somelist ) <tr><td bgcolor=$color>$something</td></tr> #end #end  **17、Range Operator**  #foreach( $foo in [1..5] ) | |