'//ASP 模板引擎 (Taihom.Template.class v3.0)  
'//Code By Taihom  
'//Taihom 原创  
'//版权说明  
'//本类允许转载、复制和修改，但转载、复制和修改的同时请保留原始的出处和作者声明，这也是对作者Taihom劳动成果的一种尊重！  
'//如果您对本类进行修改，希望您也可以给我发一份您修改过后的类(taihom@163.com)  
本类支持单标签，支持简单的逻辑，支持嵌套模板，支持循环列表,支持嵌套循环，支持调用asp函数，支持循环列表的逻辑，支持自定义标签，支持自定义标签属性和自由获取标签……  
代码在这里下载：  
演示地址：<http://www.taihom.com/templateclass/demo.asp>  
模板地址：<http://www.taihom.com/templateclass/templates>  
代码下载：<http://www.taihom.com/templateclass/taihom.template.class.rar>  
  
先看看说明！

|  |  |  |
| --- | --- | --- |
|  | **模板类的标签规则** | **说明** |
| **标签边界：** | **{}、()** | {@}是单标签 {}是类标签，主要是应用在{include("...")}和{if ...}...{else}...{/if} <loop:name ...><.../loop:name>循环标签 |
| **包含文件：** | **{include("文件路径")}** | 该模板类支持包含同模板目录下的文件，例如 **{include("head.html")}**  **支持多级include(例如1.html可以include 2.html,2.html可以include 3.html)，但是不能include死循环，否则程序出错** |
| **ASP调用标签** | {asp 函数(参数)} | v3.0新特性：可以在模板中直接调用asp函数(只要是asp中合法的函数均可调用)，但是要注意的是，使用这个标签调用函数，带参数的情况下要以asp标准带参数，例如传递字符串要用双引号引起来 例如 ：{asp now()} 将会得到当前时间 如果您在asp里面写有一个自定义函数 myfunction(var1,var2) 在模板中你可以这么调用：{asp myfunction(参数1,参数2)} 参数支持本模板的单标签 |
| **单标签：** | {@标签名} {@标签名 属性="值"} (@字段名)、(@字段下标) | (@...)单标签只能应用在loop块 另外在loop块里的取值还可以使用**(块名.字段名)**来获得 |
| **逻辑标签：** | **{if @标签变量=="值"}** 代码块 **{else}** 代码块 **{/if} {if @标签变量=="值"}** 代码块 **{/if}** | 可以处理一些简单的逻辑关系 ， 目前支持的运算符有**==**、**<=**、**>=**、**>**、**<**、**!=**、**<>** 例如：**{if @islogin=="1"}** 您已经登录 **{else}** 您没有登录！ **{/if} 逻辑标签可以应用到循环块中，但是逻辑标签不能跟自己嵌套** **v2.0开始支持逻辑标签的条件支持，运算符号前后均可支持标签和带属性的标签， 这样使得{if @var1==@var2} 或者 {if @var xx='??' == @var1 yy='??'}**得到了充分的支持 |
| **循环标签：** | **<loop:name 属性1="值1" 属性2="值2" ... > 代码块 （@数字）（@字段名 属性=""）** **</loop:name>** | 这里要注意的是 <loop> 标签部分一定要换行之后再写代码块， 这里主要设置一个断点，以便模板可以解析到<loop>的参数。 被循环块包含着的，我称之为循环体，也就是需要循环输出的部分， 一般来说，循环体是用来显示数据用的，这些数据往往都是从数据库直接筛选的结果。 所以，**循环体也支持单标签**，只不过循环体的单标签部分的**边界符号是（）** 单标签的输出方式有两种，一种是"@字段下标，一种是"@字段名"，具体怎么样用，等下例子中会有说明 **循环部分也支持逻辑**:用法跟上面的逻辑支持也是一样的。循环部分的逻辑支持获取外部单标签的方法是 @变量 不用()边界符号 例如{if (@fieldname)>@var1}   循环块有一个保留属性 name="块名字" 除了这个，想定义什么自定义属性都可以 这个值缺省也是可以的，模板类会自动分配一个name给到<loop> 主要是给<loop>一个唯一的命名，在程序中就可以传递数据给这个<loop> 如果输出的结果为空，那么如何自定义自己的输出结果呢？很简单，只要在loop标签里面加一个empty标签：例如： <loop3:loop sql="select \* from [table] where 1=2"> <empty> <li>很抱歉，没有您要查找的数据！</li> </empty> <li>显示你的数据 (@1) (@2) (@3) (loop3.name) (@name)</li> </loop3:loop> |
| **嵌套循环：** | <loop:name1 > ...(name1.fields)       <loop:name2 **data='myfunction((name1.id))'**>       ...(name2.name)       </loop:name2> </loop:name1> | v3.0新特性，3.0版本开始，支持**循环的嵌套**。 模板中的代码如左边所示。子循中需要有一个特定的 data属性作为支持。而data属性的作用是调用asp函数,这个asp函数是可以自己定义的，作用是根据传递的参数来返回子循环的数据对象。 也就是说，你可以根据你的需要在自定义的函数中构造和筛选子循环的数据。这一点是很灵活的 如果没有指定 data 属性，或者 指定的 data属性的函数不存在 或者发生错误，那么内循环是不会工作的。这一点大家要注意。 **data 返回的数据对象必须是rs数据集的对象**。例如 返回值是 set myfunction=rs 使用循环之前提取循环的值用的是标签(@tag)这样的方式来获得。而使用**嵌套循环**后，这个当然也还可以用，但是为了区别显示，我们提供了另外一种数据提取的支持 (循环名.字段名 ) 例如左边的 **(name1.id)**，(name2.name)分别表示name1循环的id值和name2循环的name值，这样不会取值混乱。 另外一个注意的是，嵌套循环不宜用在太大量的数据显示上。 |
| **模板引擎的默认属性支持，你也可以修改程序加上自己定义的属性 这些属性可以应用在单标签上** | | |
| 截取字符 | len="数字" | 例如：{@title len="5"} 这样将会截取@title前面5个字符作为输出 |
| 返回字符串长度 | return="len" | 例如：{@content return="len"} 将返回这个变量的字符串长度 |
| 清除HTML格式 | return="clearhtml" | 例如：{@content return="clearhtml"} 这样将会清除@content输出的HTML格式 |
| 清除所有空格换行 | return="clearspace" | 例如：{@content return="clearspace"} 这样将会清除@content输出的所有的空格和换行 **一个标签可以有多个return** |
| 清除所有的格式 | return="clearformat" | 这样只会保留文字，连HTML格式，换行什么都没有了 |
| 返回字符串的server.urlencode | return="urlencode" | {@content return="urlencode"} 将返回与ASP中server.urlencode("字符串")一致的结果 |
| 日期显示方式转换 | dateformat="日期显示方式" | 年(yyyy|yy)月(mm|m)日(dd|d) 时(HH|H)分(MM|M)秒(SS|S)星期(w|W) 默认支持的方式有  dateformat="0" 显示日期和/或时间。如果有日期部分，则将该部分显示为短日期格式。如果有时间部分，则将该部分显示为长时间格式。如果都存在，则显示所有部分 dateformat="1" 使用计算机区域设置中指定的长日期格式显示日期 dateformat="2" 使用计算机区域设置中指定的短日期格式显示日期 dateformat="3" 使用计算机区域设置中指定的时间格式显示时间 dateformat="4" 使用 24 小时格式 (hh:mm) 显示时间  其他的日期格式化方式，可以参照下面的例子： 例如：{@datetime dateformat="yyyy-mm-dd"} 将会输出类似 2009-03-01这样的时间格式 例如：{@datetime dateformat="yy年mm月dd日- 星期w"} 将会输出类似 09年03月01日 - 星期？ 这样的时间格式 |
| **模板类的使用例子,假设已经实例化到tpl** | | |
| 类的功能 | 方法 | 说明 |
| 设置 |  | '全局配置信息(影响到全局的设置) 'tpl.setCharset = "utf-8" '可选设置,默认是utf-8 'tpl.setCacheType = 1 '可选设置,缓存开关,默认是关的 'tpl.setCacheTimeOut = 30 '可选设置,缓存时间,单位是分钟,默认是10分钟 tpl.setRoot = cStr(Server.MapPath("./"))'必选设置,后面不带斜杠 'tpl.setTemplatedir = "/templates/"'可选设置,设置模板目录 'tpl.setCachedir = "/cache/templates/"'可选设置,设置缓存目录 **'tpl.setAbsPath = 1** '可选设置,模板绝对路径,默认是开的,作用是输出的时候将模板相对路径替换成绝对路径,已经是绝对路径或者描点等不受影响，此开关只是转换那些不是绝对路径的 页面设置(影响到单个页面的设置) 'tpl.setPageTimeout=0'(v2.0新特性)可选设置，0=不生产前台页面缓存,>0表示缓存的时间，单位是分钟 ，这个设置需要在页面开头，也就是在赋值之前设置，如果不设置，默认不缓存，建议数据有实时更新的不进行设置  tpl.setPath = "index.html"'可选,模板文件路径相对于模板目录 **tpl.setHtml**="模板内容"'可选，加载模板代码，如果不指定tpl.setPath,程序自动使用这里,反正 tpl.setPath/tpl.setHtml二选其一,使用这个之后模板缓存自动关闭(2010年1月支持) |
| 加载并解析模板 | tpl.load() | '加载模板，这个方法负责读取并且解析模板， 当然，如果你不需要提取模板的参数，你可以放到.display()之前  例如： tpl.load()  tpl.display() |
| **提取所有块标签** | **Set getblock = tpl.getblock** | **'获得块列表,返回dic对象，key = 循环块名称,items = 块的属性(数组，0=属性名称,1=值)** **例子： Dim blockname,block Set block = tpl.getblock For blockname in block Response.Write(blockname)'//这样 就可以输出所有块的名称了 Next** |
| **指定块标签的属性** | **Set getattr = tpl.getattr** | **如果你想提取块名字是loop1的sql的内容 那么只要 getattr("loop1.sql") 如果没有，将返回empty值，也就是空值** (2010年1月支持) **如果要提取单个标签的属性值** 需要这样**getattr("@标签名字.属性")** |
| 标签赋值 | tpl.add("@标签")="值" | 设置标签的值，也就是用这个值来替换掉{@标签} |
| 块标签赋值 | tpl.add("块标签")=记录集 | 设置循环块的值，这里我们只要给一个记录集，其他的输出，由模板去执行，也不用判断记录集是否为空。 |
| 显示输出 | tpl.display() | '显示模板,将结果输出 |
| 清除缓存 | tpl.ClearCache | 如果设置了模板缓存，用这个将可以清除所有的模板缓存，对于不常修改的模板，我建议开启缓存。当然如果模板有改动，必须要清空了缓存，缓存才会重建 |
| 程序中获取输出结果 | html = tpl.gethtml | 将输出的结果赋值给html变量，这里的结果是已经是最后的输出结果 |
| 生成静态页面 | tpl.OutPutPage(cStr(Server.MapPath("./"))) = "asdf.html" | 如果需要生成静态页面，使用 .OutPutPage("路径")=文件名 |
| **完整的类使用方法（来自demo.asp）** | | |
| <script language="vbscript" runat="server"> Response.Buffer = true Response.Expires = 0  Response.CacheControl = "no-cache"  Response.AddHeader "Pragma", "No-Cache" Response.CodePage="65001" Response.Charset="utf-8" </script> <!--#include file="class/Template.class.vbs.asp" --> <!--#include file="class/DBOperate.class.vbs.asp" --> <% '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ACCESS数据库\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Dim dbo Set dbo = New DBOperateClass dbo.setConnString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & server.mapPath("./test.mdb")  '//-------这里是使用该类的例子 Dim tpl Set tpl = New TemplateManageClass '配置信息 'tpl.setCharset = "utf-8" '可选设置,默认是utf-8 'tpl.setCacheType = 0 '可选设置,0=不需要缓存,1=文件缓存,2=内存缓存(只对文件模板(setPath的方式加载模板)有效) tpl.setRoot = cStr(Server.MapPath("./"))'必选设置,后面不带斜杠 'tpl.setTemplatedir = "/templates/"'可选设置,设置模板目录 'tpl.setCachedir = "/cache/templates/"'可选设置,设置文件缓存目录  tpl.setPath = "index.html"'可选,模板文件路径相对于模板目录 tpl.setHtml = "html模板内容"'可选,模板路径与setHtml二选一 tpl.load() '加载模板   '输出的数据变量 tpl.add("@title")="标题在这里" tpl.add("@username")="taihom" tpl.add("@name")="梅川内酷" tpl.add("@password")="密码" tpl.add("@runtime")=tpl.RunTimer tpl.add("@author")=tpl.Author '获得循环块的参数 Dim block,getattr Set block = tpl.getblock'获得块列表,返回dic对象，key = 循环块,items = 块的属性(数组，0=属性名称,1=值) Set getattr = tpl.getattr'获得属性列表,返回dic对象  Dim sql1 : sql1 = getattr("loop1.sql")'获取loop1块的sql属性 Dim sql2 : sql2 = getattr("loop2.sql")'获取loop2块的sql属性 Dim sql3 : sql3 = getattr("loop3.sql")'获取loop3块的sql属性  '循环块内容输出 tpl.add("loop1") = dbo.execute(sql1)'直接给loop1块数据 tpl.add("loop2") = dbo.execute(sql2)'直接给loop2块数据 tpl.add("loop3") = dbo.execute(sql3)'直接给loop3块数据  tpl.display()'显示模板  '--------------清楚所有模板缓存 'tpl.ClearCache  '--------------获取输出的代码 'html = tpl.gethtml  '--------------将结果生成静态页面 '如果需要生成静态页面，使用 .OutPutPage("路径")=文件名 'tpl.OutPutPage(cStr(Server.MapPath("./"))) = "asdf.html"  dbo.close Set tpl = Nothing Set dbo = Nothing %> | | |
|  | | |
| 简便方法： 如果某个字段中有 title,name,hit,content 等这些字段 模板中对应有 {@title} {@name} {@hit} {@content} 假设 rs已经筛选好 一般的做法是 .add("@title")=rs("title") .add("@name")=rs("name") .add("@hit")=rs("hit") .add("@content")=rs("content") 有一个更加简单的方法 .add('@')=rs 数据集 这样，模板将会自动根据数据集的字段绑定数据  如果需要自己再进行处理可以先 .add('@')=rs 数据集 之后再进行自己处理，例如 .add('@hit')= trim(rs("hit")) 这样就可以覆盖之前同名的标签数据 | | |