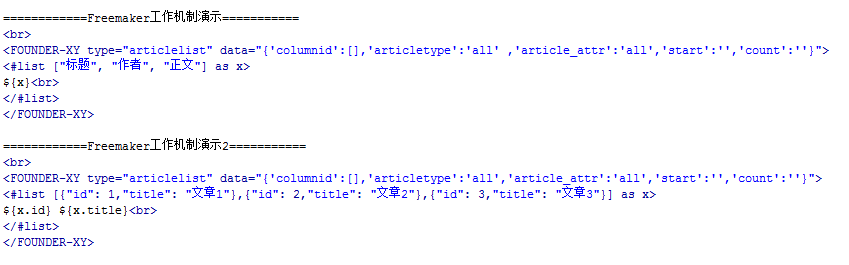
**翔宇V5模板制作经验总结**

**一、FreeMaker的工作机制**

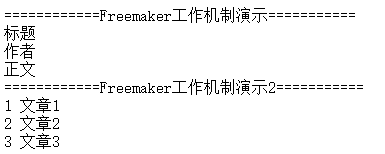
FreeMaker的工作机制就一句话：

模板 + 数据模型 = 输出

理解这句话很重要，所以我用一个模板代码实例来进行说明。



模板代码



页面效果

案例说明：

1. 首先，什么是数据模型？数据模型就是特定格式的数据。在这里为了更容易理解，我在list里手工写入了数据。第一个数据模型模拟了内容页的数据，第二个数据模型模拟了标题列表的数据；实际数据比这个字段多一些，结构复杂一些。
2. 在这里例子里，我并没有用<FOUNDER-XY>取得的数据。实际上，翔宇的Jpublish程序负责根据<FOUNDER-XY>标签里的定义从系统里取得这种**序列**或者**哈希表**数据。
3. <#list>标签里的内容就是模板，模板定义了数据的显示方式。这部分功能由Freemaker实现。

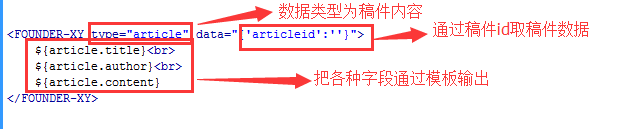
**二、常用模板组件**

翔宇5.0沿用了4.x的dw插件方式来上传下载模板，也提供了各种翔宇组件。但是当我们熟悉了以后就会发现，在html代码里复制粘贴后再修改的效率是非常高的。其实最常用的就那么几种组件。





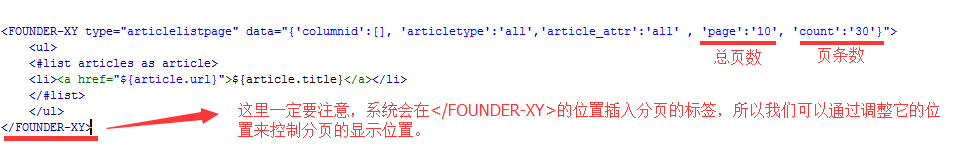
**1、文章内容组件：**



**2、标题列表组件**



**3、分页标题列表组件**



分页标签系统已经写好了css样式名，我们可以自定义样式，下面附上一段写好的样式，做项目时在此基础上微调即可。

.pages{text-align:center;font-family:arial;}

.pages a{text-decoration:none;}

.pages li{display:inline-block; border:1px #ddd solid;padding:3px 6px;margin:2px;}

.pages li:hover{background-color:#ccc;color:#1E90FF}

.pages .page-active{background-color:#ccc;}



短分页效果

如果分页在10页以内，这个效果就不错了，但是如果超过10页，系统默认还是显示所有的页码。为了显示效果更美观，可以采用js对一些页面进行隐藏。



长分页效果

附上js代码，这段代码需要放在分页标签之后，所以放到模板尾部比较合适。

<script src=/js /jquery-1.3.2.min.js></script>

<script>

str="<li class='page'>...</li>";

pages=$("ul.pages").attr("pages");

cur=$(".page-active").text();

cur=Number(cur); //转为数字

if (pages>10){

$("li.page").hide();

$("li.page-previous").hide();

$("li.page-next").hide();

$("li.page1").show();

if (cur>5 && cur<(pages-4)){

$("li.page"+(cur-3)).show();

$("li.page"+(cur-2)).show();

$("li.page"+(cur-1)).show();

$("li.page-active").show();

$("li.page"+(cur+1)).show();

$("li.page"+(cur+2)).show();

$("li.page"+(cur+3)).show();

$("li.page"+pages).show();

$("li.page1").after(str);

$("li.page"+pages).before(str);

}else if(cur<=5){

$("li.page1").show();

$("li.page2").show();

$("li.page3").show();

$("li.page4").show();

$("li.page5").show();

$("li.page6").show();

$("li.page7").show();

$("li.page8").show();

$("li.page"+pages).show();

$("li.page"+pages).before(str);

}else{

$("li.page"+(pages-7)).show();

$("li.page"+(pages-6)).show();

$("li.page"+(pages-5)).show();

$("li.page"+(pages-4)).show();

$("li.page"+(pages-3)).show();

$("li.page"+(pages-2)).show();

$("li.page"+(pages-1)).show();

$("li.page"+pages).show();

$("li.page1").after(str);

}

}

</script>

**三、****FreeMaker常用语法汇总**

FreeMaker支持算术运算、逻辑运算，条件语句等，并且有很多内建函数，使用非常灵活，能实现很多功能。具体语法请参考FreeMaker手册。

下面列出个人认为最常用的部分内容。

**算术运算：**加 + 减 - 乘 \* / 除 取余 %

**比较运算符：**赋值 = 等于 == 不等于 != 大于 gt 大于等于 gte 小于 lt

小于等于lte

**逻辑运算符：**逻辑与 && 逻辑或 || 逻辑非 !

**内建函数：**字符串长度length 截取字符串substring 替换 replace

数字转字符串 string 数字转字符 c 判断变量是否存在 ??

**数据数量：**sequence?size

**数据反序：**sequence?reverse

**数据按指定字段排序：**sequence?sort\_by(“key”)

**在 list 循环中，有两个特殊的循环变量可用：**

* *item\_index：这是一个包含当前项在循环中的步进索引的数值。*
* *item\_has\_next：来辨别当前项是否是序列的最后一项的布尔值。*

**条件语句：**

<#if condition>...

<#elseif condition>...

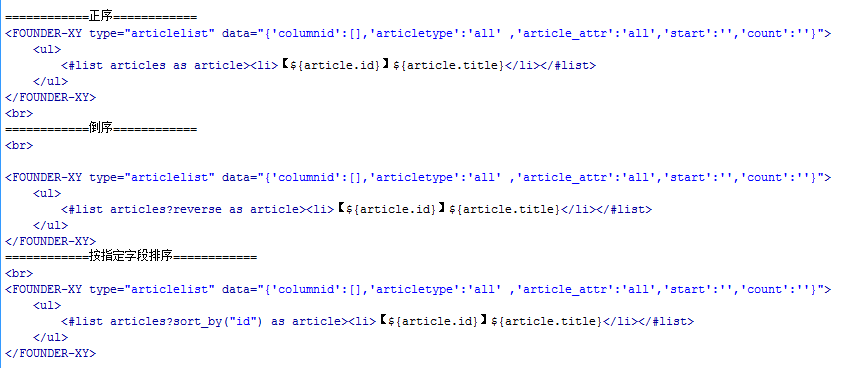
<#elseif condition>...

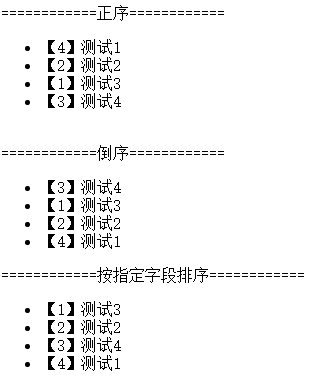
<#else> ...

</#if>

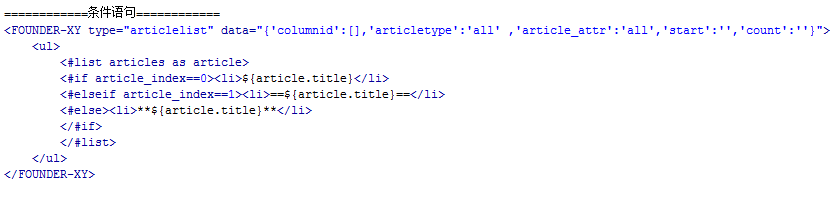
**四、翔宇系统FreeMaker常用语法示例**

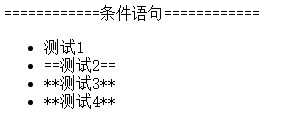
**1、稿件排序**

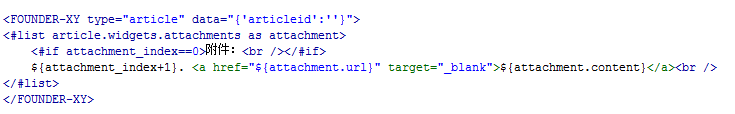


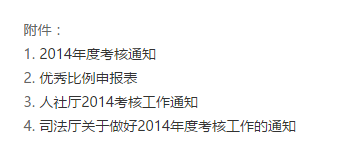
**2、条件语句**



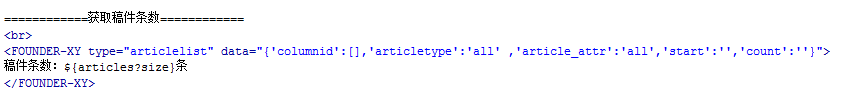


**3、实现附件列表自动加索引值**





**4、获取稿件条数**



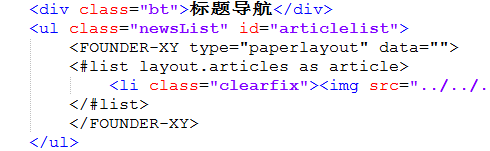


**五、用FreeMaker语法解决问题案例**

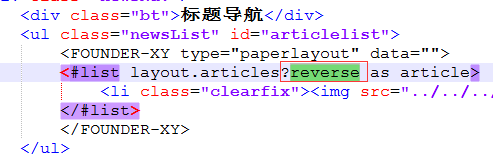
**1、数字报标题列表排序问题，因为默认发布出来的稿件列表顺序不对，需要对把所有版面的稿件反序显示。**

分析：数字报每一期的每个版都是一个单独的栏目，所以只需要让栏目下的稿件反序显示就行。用sequence?reverse即可解决。

模板里是这么写的：



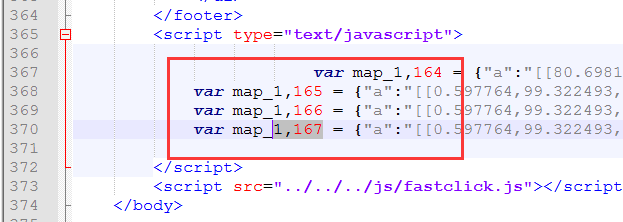
改成下图这样即可。



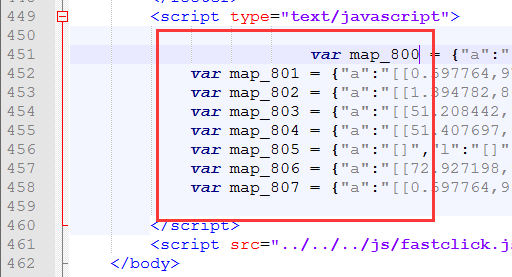
**2、数字报本来运行正常，突然有一天手机版文章内容点不开。**

分析：检查后台程序、数据都没发现什么问题。通过F12发现有js报错，再仔细查看，发现了问题所在。

先看下图，问题版面里有这样的内容，这是一段js代码，var后面跟的是变量名，但是变量名里是不可能有逗号的。



再看正常的版面，应该是下图这样的。



两相对比，就可以猜测出问题原因了，就是FreeMaker输出数值时用科学计数法给每隔三位数加了逗号，造成js程序错误；以前没问题是因为id没有增长到四位数。

这时FreeMaker的内建函数就派上用场了，内建函数c就是是用来将数字转成字符输出，正好满足需求。我们把模板里${layout.id}这种输出加了逗号的字段都改成${layout.id?c}，这样重新发布就可解决问题。

Ps.浏览器的F12功能是调试B/S程序的必备工具。谷歌浏览器的F12还有模拟移动设备的功能，可以很方便调试手机版页面。

**六、总结**

FreeMaker具有编程语言的许多特性，它让我们可以自由的改变数据的输出方式。与翔宇4.x的模板制作方式相比，FreeMaker更灵活，能实现很多过去实现不了的功能；可以直接编辑html代码，而不用再打开翔宇标签编辑器；因为它增加了一套语法规则，所以也提高了一些学习成本。