**jquery template使用**

动态请求数据来更新页面是现在非常常用的方法，比如博客评论的分页动态加载，微博的滚动加载和定时请求加载等。

这些情况下，动态请求返回的数据一般不是已拼好的HTML就是JSON或XML，总之不在浏览器端拼数据就在服务器端拼数据。不过，从传输量方面来看，返回HTML不划算，而在web传输方面，现在更多的是使用JSON而不是XML。

浏览器端根据JSON生成HTML有个很苦恼的地方就是，结构不复杂的时候还好，结构一复杂，就想死了，需要很小心很小心地写出几乎无法维护的JavaScript代码。

因此一些用模版生成HTML的的框架相继出现jquery.tmpl 就是其中的一种，下面我们来详细介绍下**jquery.tmpl的用法**：

jquery.tmpl的几种常用标签分别有：

**${}**, **{{each}}**, **{{if}}**, **{{else}}**, **{{html}}**

不常用标签

**{{=}}**,**{{tmpl}}** and **{{wrap}}**.

**${}**等同与**{{=}}**是输出变量 **${}**里面还可以放表达式 （=和变量之间一定要有空格，否则无效）

示例：

[复制代码](javascript:void(0);)

<div id="div\_demo">

</div>

<script id="demo" type="text/x-jquery-tmpl">

<div style="margin-bottom:10px;">

　　<span>${ID}</span>

　　<span style="margin-left:10px;">{{= Name}}</span>

　　<span style="margin-left:10px;">${Number(Num)+1}</span>

　　<span style="margin-left:10px;">${Status}</span>

</div>

</script>

<script type="text/javascript">

　　var users = [{ ID: 'think8848', Name: 'Joseph Chan', Num: '1', Status: 1 }, { ID: 'aCloud', Name: 'Mary Cheung', Num: '2'}];

　　$("#demo").tmpl(users).appendTo('#div\_demo');

</script>

**{{each}}** 提供循环逻辑，$value访问迭代变量 也可以自定义迭代变量(i,value)

示例：

<div id="div\_each">

</div>

<script id="each" type="text/x-jquery-tmpl">

<h3>users</h3>

{{each(i,user) users}}

<div>${i+1}:{{= user.name}}</div>

{{if i==0}}

<h4>group</h4>

{{each(j,group) groups}}

<div>${group.name}</div>

{{/each}}

{{/if}}

{{/each}}

<h3>depart</h3>

{{each departs}}

<div>{{= $value.name}}</div>

{{/each}}

</script>

<script type="text/javascript">

　　var eachData = { users: [{ name: 'jerry' }, { name: 'john'}], groups: [{ name: 'mingdao' }, { name: 'meihua' }, { name: 'test'}], departs: [{ name: 'IT'}] };

　　$("#each").tmpl(eachData).appendTo('#div\_each');

</script>

**{{if }} {{else}}**提供了分支逻辑 {{else}} 相当于else if

示例：

<div id="div\_ifelse"></div>

<script id="ifelse" type="text/x-jquery-tmpl">

<div style="margin-bottom:10px;"><span>${ID}</span><span style="margin-left:10px;">{{= Name}}</span>

{{if Status}}

<span>Status${Status}</span>

{{else App}}

<span>App${App}</span>

{{else}}

<span>None</span>

{{/if}}

</div>

</script>

<script type="text/javascript">

　　var users = [{ ID: 'think8848', Name: 'Joseph Chan', Status: 1, App: 0 }, { ID: 'aCloud', Name: 'Mary Cheung', App: 1 }, { ID: 'bMingdao', Name: 'Jerry Jin'}];

$("#ifelse").tmpl(users).appendTo('#div\_ifelse');

</script>

**{{html}}** 输出变量html,但是没有html编码，适合输出html代码

 实例

[复制代码](javascript:void(0);)

<div id="div\_html"></div>

<script id="html" type="text/x-jquery-tmpl">

<div style="margin-bottom:10px;">  
　　　　<span>${ID}</span>  
　　　　<span style="margin-left:10px;">{{= Name}}</span>

　　${html}

　　{{html html}}

</div>

</script>

<script type="text/javascript">

　　var user = { ID: 'think8848', Name: 'Joseph Chan', html: '<button>html</button>' };

　$("#html").tmpl(user).appendTo('#div\_html');

</script>

**{{tmpl}}** 嵌套模版

实例

<div id="tmpl"></div>

<script id="tmpl1" type="text/x-jquery-tmpl">

<div style="margin-bottom:10px;">

　　<span>${ID}</span>

　　<span style="margin-left:10px;">{{tmpl($data) '#tmpl2'}}</span>

</div>

</script>

<script id="tmpl2" type="type/x-jquery-tmpl">

{{each Name}}${$value} {{/each}}

</script>

<script type="text/javascript">

　　var users = [{ ID: 'think8848', Name: ['Joseph', 'Chan'] }, { ID: 'aCloud', Name: ['Mary', 'Cheung']}];

$("#tmpl1").tmpl(users).appendTo('#tmpl');

</script>

**{{wrap}}**,包装器

实例

<div id="wrapDemo">

</div>

<script id="myTmpl" type="text/x-jquery-tmpl">

The following wraps and reorders some HTML content:

{{wrap "#tableWrapper"}}

<h3>One</h3>

<div>

First <b>content</b>

</div>

<h3>Two</h3>

<div>

And <em>more</em> <b>content</b>...

</div>

{{/wrap}}

</script>

<script id="tableWrapper" type="text/x-jquery-tmpl">

<table cellspacing="0" cellpadding="3" border="1"><tbody>

<tr>

{{each $item.html("h3", true)}}

<td>

${$value}

</td>

{{/each}}

</tr>

<tr>

{{each $item.html("div")}}

<td>

{{html $value}}

</td>

{{/each}}

</tr>

</tbody></table>

</script>

<script type="text/javascript">

$(function () {

$('#myTmpl').tmpl().appendTo('#wrapDemo');

});

</script>

**$data $item** $item代表当前的模板；$data代表当前的数据。

 实例：

<div id="div\_item\_data"></div>

<script id="item\_data" type="text/x-jquery-tmpl">

<div style="margin-bottom:10px;">  
　　　　<span>${$data.ID}</span>  
　　　　<span style="margin-left:10px;">${$item.getName(" ")}</span>  
　　　</div>

</script>

<script type="text/javascript">

　　var users = [{ ID: 'think8848', Name: ['Joseph', 'Chan'] }, { ID: 'aCloud', Name: ['Mary', 'Cheung']}];

$("#item\_data").tmpl(users,

{

getName: function (spr) {

return this.data.Name.join(spr);

}

}).appendTo('#div\_item\_data');

</script>

**$.tmplItem()**方法，使用这个方法，可以获取从render出来的元素上重新获取$item

实例

<script type="text/javascript">

　　$('#demo').delegate('div', 'click', function () {

var item = $.tmplItem(this);

alert(item.data.Name);

});

</script>