
[function]

[guide 引擎]

[让界面更活跃]

林光樟

[2017-8-30]

目录

| | |
|--------------------------------|---|
| Guide 文档..... | 1 |
| 1 什么是 guide? | 1 |
| 1.1 模板标识符: | 1 |
| 1.2 兼容性考虑: | 1 |
| 2 Guide 的内置逻辑类。 | 1 |
| 2.1 什么是逻辑类? | 1 |
| 2.2 前置逻辑代码 js-before | 2 |
| 2.3 后置逻辑代码 js-after | 2 |
| 2.4 内前置逻辑代码 js-appendto | 2 |
| 2.5 内后置逻辑代码 js-append..... | 2 |
| 3 Guide 的使用方法。 | 2 |
| 3.1 回调第一个参数, 刷新。 | 3 |
| 3.2 回调第二个参数, 数据。 | 4 |
| 3.3 回调第三个参数, 接口引用。 | 4 |
| 4 用法和例子。 | 5 |
| 4.1 动态表格..... | 5 |
| 4.2 嵌套子模板: | 6 |
| 5 Guide 在 requirejs 的使用。 | 7 |
| 6 美化..... | 7 |
| 7 寄语..... | 8 |

Guide 文档

1 什么是 guide?

Guide 是一个 html 的模块加载引擎。Guide 集模板渲染，模板模块化编程，遵循的是 AMD（异步模块定义）规范。Guide 模板具有较高的可编程性，为解决复杂重复的界面 html 代码带来了无限可能。

1.1 模板标识符：

逻辑代码：<#逻辑代码#>

输出文本：<#=对应数据对象#>

输出 html：<#:对应数据对象#>

1.2 兼容性考虑：

只考虑最新的浏览器引擎，ie9 及以上，火狐、谷歌最新浏览器。

2 Guide 的内置逻辑类。

2.1 什么是逻辑类？

逻辑类是在界面添加逻辑代码的属性，通过逻辑类达到模板的编程。通过逻辑类，可以解决浏览器对 html 的格式化带来的导致逻辑代码错位的影响，还可以让 html 代码看起来更加整洁易懂。

2.2 前置逻辑代码 js-before

模板代码：`<div js-before=" alert('在之前');" >内部代码</div>`

等价于：`<# alert('在之前');#> <div>内部代码</div>`

2.3 后置逻辑代码 js-after

模板代码：`<div js-after=" alert('在之后');" >内部代码</div>`

等价于：`<div>内部代码</div> <# alert('在之前');#>`

2.4 内前置逻辑代码 js-appendto

模板代码：`<div js-appendto=" alert('在内前');" >内部代码</div>`

等价于：`<div><# alert('在内前');#>内部代码</div>`

2.5 内后置逻辑代码 js-append

模板代码：`<div js-append=" alert('在内后');" >内部代码</div>`

等价于：`<div>内部代码<# alert('在内后');#></div>`

3 Guide 的使用方法。

使用格式：

```
Guide 的 OS 类似 jquery 的$;  
OS("table",{  
    //模板初始化数据  
},function($apply,$data,$quote){
```

```
//模板内逻辑代码

return {};//模板对外开放接口

});
```

这里面有三个内置回调参数，分别是：**刷新、数据、接口引用**；

3.1 回调第一个参数，刷新。

假设初始数据为：

```
{
  title:'标题'
}
```

代码如下：

```
$apply({

  title:"新标题"

},false);//是否部分更新
```

代码效果：

模板刷新，原先的 title 将替换成新的 title 值。

第一个参数，新的数据，格式必需与原始数据匹配，只能在原始数据基础上做操作。

第二个参数，不设置或者设置为 false 为替换刷新，如果模板改动涉及到逻辑，应该使用这种方式，这种方式特点是覆盖刷新，覆盖后之前绑定到指定 dom 的事件将不复存在。如果为 true，将不影响之前的事件绑定，在

结构不改变，单纯改变内容的情况下，使用这种方式是比较方便的。

3.2 回调第二个参数，数据。

假设初始数据为：

```
{
  title: '标题'
}
```

代码如下：

```
$data.title = "新标题";
$apply($data);
```

代码效果：

效果同上。

3.3 回调第三个参数，接口引用。

假设一个名为“test”的模板，对外开放接口有：

```
{
  test:function(){
    //代码省略
  }
}
```

代码如下：

```
$quote("test",function($out){
```

```
$out.test();  
});
```

代码效果：

在当前模板，引用 test 模板的对外接口的 test 方法。

4 用法和例子。

4.1 动态表格

Html 逻辑代码：

```
<table id = "table">  
  
  <tbody>  
  
    <tr js-before="for(var i=0;i<row;i++){"  
js-after="}">  
  
      <td js-before="for(var j=0;j<col;j++){"  
js-after="}"><#=i+j*row#></td>  
  
    </tr>  
  
  </tbody>  
  
</table>
```

js 代码：

```
OS("table",{  
  
  row:3,  
  
  col:5
```

```
},function($apply,$data,$quote){  
    return {};  
});
```

代码结果：

0 3 6 9 12

1 4 7 10 13

2 5 8 11 14

4.2 嵌套子模板：

Html 代码：

```
<#  
    var itemTpl function (){  
#>
```

<div>子模板逻辑结构代码</div>

```
<#  
    };//每个逻辑变量后要带;  
#>
```

使用模板：

```
<#  
    itemTpl();//使用模板
```


#>

5 Guide 在 requirejs 的使用。

例子同 4.1 表格例子：

代码 define:table.js：

```
define(['guide'],function(OS){

    return OS("table",{

        row:3,

        col:5

    },function($apply,$data,$quote){

        return {};

    });

});
```

代码 require:api.js：

```
require(['table'],function (table) {

    //table参数为该模块的对外接口方法

});
```

6 美化

为了使体验更好，建议引入 guide.css，在模板的 class 中引入 _guide 类，可以掩盖模板渲染之前出现闪烁的效果。

7 寄语

Guide 的意思是指导、引路的意思，目的在于更好的处理复杂的界面逻辑，使得界面 html 代码更具模块的特性。