

# 模板引擎art-template

---

模板引擎用于根据定义好的模板来批量生成HTML的一个工具

## 存在的问题

我们之前希望通过这个一个数据来动态的生成我们的网页结构

```
1  var stus = [  
2    {  
3      stuName: "张三",  
4      sex: "女",  
5      age: 18,  
6      hobby: ["看书", "睡觉"]  
7    },  
8    {  
9      stuName: "李四",  
10     sex: "女",  
11     age: 22,  
12     hobby: ["看书", "睡觉"]  
13   }  
14   // ...这里还有很多  
15 ]
```

以前要实现这种动态数据的渲染，我们需要去遍历数据如何通过 `createElement` 来创建元素，如何去追加元素，这么做就很麻烦

```
1 <body>
2   <ul class="list"></ul>
3 </body>
4 <script src="./js/stus.js"></script>
5 <script>
6   var list = document.querySelector(".list")
7   stus.forEach(function (item, index) {
8     var li = document.createElement("li")
9     li.innerHTML = "姓名: " + item.stuName + "-----性别: " +
    item.sex;
10    list.appendChild(li)
11  })
12 </script>
```

## 模板的使用

前面说到了模板引擎就是根据一个定义好的摹本去批量的生成我们的HTML页面，这个固定的模板我们就叫他模板引擎

现在的模板引擎有很多

- `art-template`
- `ejs`
- `jade`
- `pug`
- `hbs`

本堂课我们以 `art-template` 为标准来学习，我们需要从网站上面下周这个程序库，然后引入到网页中

```
1 <script src="./js/template-web.js"></script>
```

## 标准语法

`art-template` 有两种语法，一种是标准语法，还有一种是简洁语法，现在我们先从标准语法开始

## 定义模板

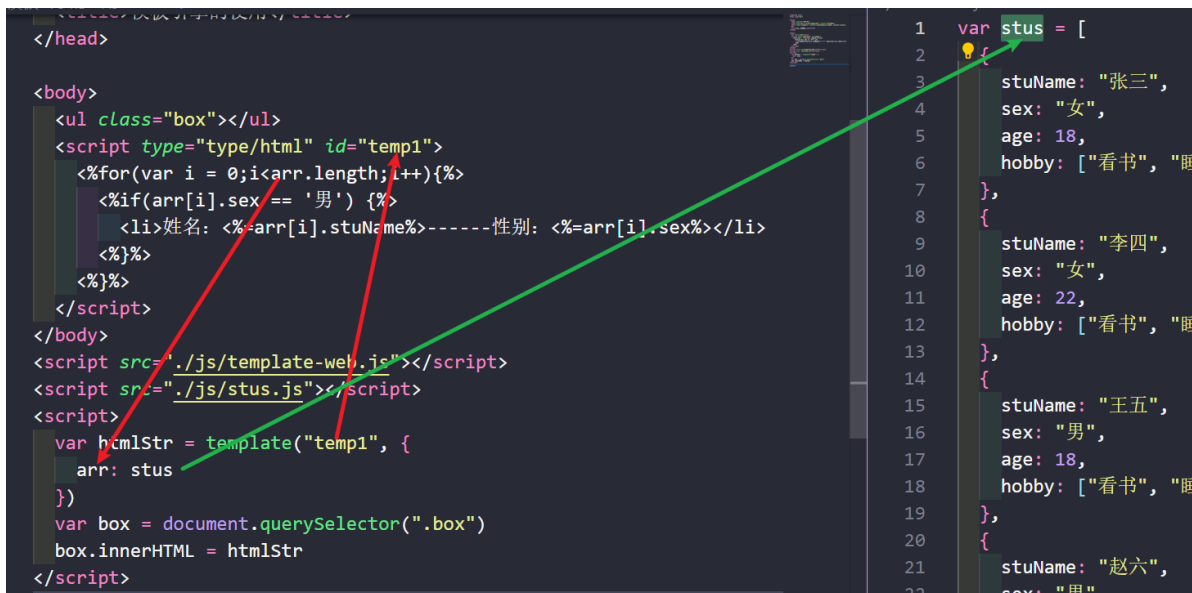
```
1 <script type="type/html" id="temp1">
2     <%for(var i = 0;i<arr.length;i++){%>
3         <%if(arr[i].sex == '男') {%>
4             <li>姓名: <%=arr[i].stuName%>-----性别:
              <%=arr[i].sex%></li>
5         <%}%>
6     <%}%>
7 </script>
```

在上面的模板定义里面，我们使用了 `<%%>` 这个符号，这个符号的作用就是在模板内部可以写 `ECMAScript` 代码，同时我们在这个符号里面写了一个 `for` 循环，同时我们也使用了一个 `if` 的条件判断语句

## 使用模板

当我们在页面中引入了 `template-web.js` 这个文件后，他就会给我们一个方法，叫 `template`，这个方法是让我们根据数据来批量的生成这个HTML标签

```
1 <script src="./js/template-web.js"></script>
2 <script src="./js/stus.js"></script>
3 <script>
4     var htmlStr = template("temp1", {
5         arr: stus
6     })
7     var box = document.querySelector(".box")
8     box.innerHTML = htmlStr
9 </script>
```



## 简洁语法

这个语法使用起来非常简单，只需要记住几个命令即可

```

1  <script type="type/html" id="temp1">
2    {{each arr item index}}
3      {{if item.sex == "男"}}
4        <li>序号: {{index}}-----姓名: {{item.stuName}}-----性
      别: {{item.sex}}</li>
5      {{/if}}
6    {{/each}}
7  </script>

```

在上面的代码里面，我们可以看到，我们使用了 `{{each}}` 来进行遍历，使用了 `{{if}}` 来进行判断，我们可以得出一个结论 `{{}}` 内部就是我们的JS代码

## 案例

现在有一组，数据需要渲染，性别为男的文字颜色就渲染为蓝色，性别为女的文字颜色就渲染为红色

```

1  var stus = [
2    {
3      stuName: "张三",
4      sex: "女",
5      age: 18,
6      hobby: ["看书", "睡觉"]
7    },

```

```

8      {
9          stuName: "李四",
10         sex: "女",
11         age: 22,
12         hobby: ["看书", "睡觉"]
13     }
14     // ...这里还有很多
15 ]

```

## 第一种模板写法

```

1  <script type="type/html" id="temp1">
2      {{each arr item index}}
3          {{if item.sex == "男"}}
4              <li class="blue"> 姓名: {{item.stuName}}-----性别:
                {{item.sex}}</li>
5          {{else if item.sex == "女"}}
6              <li class="red"> 姓名: {{item.stuName}}-----性别:
                {{item.sex}}</li>
7          {{/if}}
8      {{/each}}
9  </script>

```

## 第二种模板写法

```

1  <script type="type/html" id="temp1">
2      {{each arr item index}}
3          <li style="color: {{item.sex=="男"?'blue':'red'}}">
                姓名: {{item.stuName}}-----性别: {{item.sex}}</li>
4      {{/each}}
5  </script>

```

## 第三种模板写法

```

1  <script type="type/html" id="temp1">
2      {{each arr item index}}
3          <li class="{{item.sex=="男"?'blue':'red'}}"> 姓名:
                {{item.stuName}}-----性别: {{item.sex}}</li>
4      {{/each}}
5  </script>

```

同时我们也可以用上面的数据生成一个 `table`，如下所示

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width,
initial-scale=1.0">
8      <title>模板引擎</title>
9      <style>
10         * {
11             margin: 0;
12             padding: 0;
13             list-style-type: none;
14         }
15
16         .container {
17             width: 80%;
18             margin: auto;
19         }
20
21         .title {
22             text-align: center;
23             margin: 20px 0;
24         }
25
26         .btn {
27             padding: 6px 10px;
28             border: none;
29             color: #fff;
30             border-radius: 3px;
31         }
32
33         .btn-del {
34             background-color: red;
35         }
36
37         .btn-edit {
38             background-color: orange;
```

```
39     }
40
41     .btn-add {
42         background-color: lightseagreen;
43     }
44
45     .table1 {
46         width: 100%;
47         border: 1px solid black;
48         border-spacing: 0;
49     }
50
51     .table1 td,
52     .table1 th {
53         text-align: center;
54         border: 1px dotted black;
55         height: 40px;
56     }
57 </style>
58 </head>
59
60 <body>
61     <div class="container">
62         <h2 class="title">学生信息列表</h2>
63         <div class="btn-box">
64             <button type="button" class="btn btn-add"
65             onclick="showMask()">新增</button>
66             <button type="button" class="btn btn-del">批量删除
67             </button>
68         </div>
69         <table class="table1">
70         </table>
71     </div>
72
73     <div class="mask">
74         <form name="stuForm" class="stuForm">
75             <div class="form-item">
76                 <span class="label">姓名</span>
77                 <input type="text" id="stuName" class="aaa">
78             </div>
79             <div class="form-item">
```

```

78         <span class="label">性别</span>
79         <select id="sex" class="aaa">
80             <option value="男">男</option>
81             <option value="女">女</option>
82         </select>
83     </div>
84     <div class="form-item">
85         <span class="label">年龄</span>
86         <input type="number" id="age" class="aaa">
87     </div>
88     <div class="form-item">
89         <span class="label">爱好</span>
90         <input type="text" id="hobby" class="aaa">
91     </div>
92     <div class="form-item">
93         <button type="button" class="btn btn-add"
94         onclick="saveData()">保存</button>
95         <button type="button" class="btn btn-del"
96         onclick="closeMask()">取消</button>
97     </div>
98     </form>
99 </div>
100 <script type="type/html" id="stuTemp">
101     <tr>
102         <th><label><input type="checkbox">全选</label>
103     </th>
104     <th>姓名</th>
105     <th>性别</th>
106     <th>年龄</th>
107     <th>爱好</th>
108     <th>操作</th>
109 </tr>
110 {{each stuArr item index}}
111     <tr>
112         <td><input type="checkbox"></td>
113         <td>{{item.stuName}}</td>
114         <td>{{item.sex}}</td>
115         <td>{{item.age}}</td>
116         <td>{{item.hobby.toString()}}</td>
117         <td>

```



```
115         <button type="button" class="btn btn-edit">编辑
      辑</button>
116         <button type="button" class="btn btn-del">删除
      </button>
117     </td>
118 </tr>
119 {{/each}}
120 </script>
121 </body>
122 <script src="../js/template-web.js"></script>
123 <script src="../js/stus.js"></script>
124 <script>
125     function renderHtml() {
126         var htmlStr = template("stuTemp", {
127             stuArr: stus
128         })
129         document.querySelector(".table1").innerHTML = htmlStr
130     }
131     renderHtml()
132 </script>
133
134 </html>
```