

44jsDOM增删改查和练习

DOM方法：

- 获取元素节点：通过document对象调用 **练习2 DOM查询**
 1. **getElementById ()**：通过 **id** 属性获取 **一个** 元素节点对象
 2. **getElementsByTagName ()**：通过 **标签名** 获取 **一组** 元素节点对象，会返回一个类数组对象，所有查询到的元素都会封装到对象中，即使查询到的就只有一个
 3. **getElementsByName ()**：通过 **name** 属性获取 **一组** 元素节点对象
- 获取元素节点的子节点：通过具体的元素节点调用 **练习2 DOM查询**
 1. **getElementsByTagName ()**：方法，返回当前节点的指定标签名后代节点
 2. **childNodes**：属性，表示当前节点的所有子节点，包括文本节点在内的所有节点，根据DOM标准标签间空白也会当成文本节点，**注意**：在IE8及以下的浏览器不会把空白文档当成子节点
 3. **children**：属性，可以获取当前元素的所有子元素
 4. **firstChild**：属性，表示当前节点的第一个子节点（包括空白文本）
 5. **firstElementChild**：属性，可以获取当前元素的第一个子元素，不支持IE8及以下
 6. **lastChild**：属性，表示当前节点的最后一个子节点
- 获取父节点和兄弟节点：通过具体的节点调用 **练习2 DOM查询**
 1. **parentNode**：属性，表示当前节点的父节点
 2. **previousSibling**：属性，表示当前节点的前一个兄弟节点（也会获取空白文本）
 3. **previousElementSibling**：属性，获取前一个兄弟元素，IE8及以下不支持
 4. **nextSibling**：属性，表示当前节点的后一个兄弟节点
- 获取body标签：在document中有一个属性 **body**，它保存的是body的引用

```
1 var body = document.body;
2 console.log(body); //<body></body>
```

- 获取html根标签：在document中有一个属性 **documentElement**，它保存的是html根标签

```
1 var html = document.documentElement;
2 console.log(html); //<html></html>
```

- 获取所有元素：在document中有一个属性 **all**，它保存的是所有元素

```
1 var all = document.all;
2 //all = document.getElementsByTagName("*");这种方式也可以获得全部元素
3 for(var i = 0; i < all.length; i++){
4     console.log(all[i]);
5 }
```

- **getElementsByClassName ()**：根据元素的 **class属性值** 查询一组元素节点对象，但是IE8及以下不支持
- **document.querySelector ()**：需要一个选择器字符串作为参数，可以根据一个css选择器来查询一个元素节点

点对象，虽然 IE8 中没有 `getElementsByClassName()` 但是可以使用 `querySelector()` 代替，使用该方
法总会

返回唯一的元素，如果满足条件的元素有多个，那么他只会返回第一个

- **`document.querySelectorAll()`**：该方法和 `document.querySelector()` 类似，不同的是他会将符合
条件的元素

封装到一个数组中返回，只有一个元素也会返回

- **`document.createElement()`**：可以创建一个 元素节点对象，他需要一个标签名作为参数，将会根
据该标签名
常见元素节点对象，并将创建好的对象作为返回值返回 eg: `var li = document.createElement("li");`
- **`document.createTextNode()`**：可以用来 创建一个文本节点对象，需要一个文本内容作为参数，将
会根据该
内容创建文本节点，并将新的节点返回
- **`appendChild()`**：向一个父节点中添加一个新的子节点， 用法：父节点.`appendChild`(子节点)；
- **`insertBefore()`**：可以在指定的子节点迁出如新的节点 语法：父节点.`insertBefore`(新节点, 旧节
点)；
- **`replaceChild()`**：可以使用指定的子节点替换已有的子节点 语法：父节点.`replaceChild`(新节点,
旧节点)；
- **`removeChild()`**：可以删除一个子节点 语法：父节点.`removeChild`(子节点)；

innerHTML 通过这个属性可以获取到元素内部的html代码，但对于自结束标签此属性没有意义

innerText 该属性可以获取到元素内部的文本内容，和 innerHTML 不同的是它会自动将 html 去除

如果需要读取元素节点属性：直接使用 元素 . 属性名 例子：元素.id，元素.name，元素.value

注意：class属性不能采用这种方式，读取class属性是需要使用 元素 . className

练习：

- ✓ 1. 图片的切换



```

2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title></title>
6      <style type="text/css">
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         #outer{
12             width: 300px;
13             height: 270px;
14             margin: 50px auto;
15             background-color: pink;
16             text-align: center;
17         }
18         img{
19             width: 280px;
20             height: 200px;
21         }
22         button{
23             margin-top: 10px;
24         }
25     </style>
26     <script type="text/javascript">
27         window.onload = function() {
28             var prev = document.getElementById("prev");
29             var next = document.getElementById("next");
30             var img = document.getElementsByTagName("img")[0];
31             var imgArr =
["image/1.jpg", "image/2.jpg", "image/3.jpg", "image/4.jpg", "image/5.jpg"];
32             var index = 0;
33             var info = document.getElementById("info");
34             info.innerHTML = "一共有"+imgArr.length+"张图片，当前是"+(index+1)+"张";
35             prev.onclick = function() {
36                 //img.src = "image/1.jpg"
37                 index--;
38                 if(index < 0){
39                     index = imgArr.length-1;
40                 }
41                 img.src = imgArr[index];
42                 info.innerHTML = "一共有"+imgArr.length+"张图片，当前是"+(index+1)+"张";
43             }
44             next.onclick = function() {
45                 //img.src = "image/2.jpg";
46                 index++;
47                 if(index > imgArr.length-1){
48                     index = 0;
49                 }
50                 img.src = imgArr[index];
51                 info.innerHTML = "一共有"+imgArr.length+"张图片，当前是"+(index+1)+"张";
52             }
53         }
54     }
55 </script>
56 </head>
57 <body>
58     <div id="outer">

```

```
59     <p id="info"></p>
60     
61     <button id="prev">上一张</button>
62     <button id="next">下一张</button>
63 </div>
64 </body>
65 </html>
```



2. DOM查询和练习

◦ 获取元素节点：通过document对象调用 **练习2 DOM查询**

1. **getElementById ()**：通过 **id** 属性获取 **一个** 元素节点对象
2. **getElementsByTagName ()**：通过 **标签名** 获取 **一组** 元素节点对象，会返回一个类数组对象，所有查询到的元素都会封装到对象中，即使查询到的就只有一个
3. **getElementsByName ()**：通过 **name** 属性获取 **一组** 元素节点对象

◦ 获取元素节点的子节点：通过具体的元素节点调用 **练习2 DOM查询**

1. **getElementsByTagName ()**：方法，返回当前节点的指定标签名后代节点
2. **childNodes**：属性，表示当前节点的所有子节点，包括文本节点在内的所有节点，根据DOM标准标准间空白也会当成文本节点，**注意**：在IE8及以下的浏览器不会把空白文档当成子节点
3. **children**：属性，可以获取当前元素的所有子元素
4. **firstChild**：属性，表示当前节点的第一个子节点（包括空白文本）
5. **firstElementChild**：属性，可以获取当前元素的第一个子元素，不支持IE8及以下
6. **lastChild**：属性，表示当前节点的最后一个子节点

◦ 获取父节点和兄弟节点：通过具体的节点调用 **练习2 DOM查询**

1. **parentNode**：属性，表示当前节点的父节点
2. **previousSibling**：属性，表示当前节点的前一个兄弟节点（也会获取空白文本）
3. **previousElementSibling**：属性，获取前一个兄弟元素，IE8及以下不支持
4. **nextSibling**：属性，表示当前节点的后一个兄弟节点

◦ **innerHTML** 通过这个属性可以获取到元素内部的html代码，但对于自结束标签此属性没有意义

innerText 该属性可以获取到元素内部的文本内容，和 **innerHTML** 不同的是它会自动将 **html** 去除

你喜欢哪个城市？

北京 上海 东京 首尔

你喜欢哪款单机游戏？

红警 实况 极品飞车 魔兽

你手机的操作系统是？

IOS Android Window Phone

gender: ☐ Male ☐ Female

name:

查找#bj节点

查找所有li节点

查找name=gender的所有节点

查找#city下所有li节点

返回#city下的所有子节点

返回#phone的第一个子节点

返回#bj的父节点

返回#android的前一个兄弟节点

读取#username的value属性值

设置#username的value属性值

返回#bj的文本值

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title></title>
6      <style type="text/css">
7        body {
8          width: 800px;
9          margin-left: auto;
10         margin-right: auto;
11       }
12       button {
13         width: 300px;
14         margin-bottom: 10px;
15       }
16       #btnList {
17         float:left;
18       }
19       #total{
20         width: 450px;
21         float:left;
22       }
23       ul{
24         list-style-type: none;
25         margin: 0px;
26         padding: 0px;
27       }
28       .inner li{
29         border-style: solid;
30         border-width: 1px;
31         padding: 5px;
32         margin: 5px;
33         background-color: #99ff99;
34         float:left;
35       }
36       .inner{
37         width:400px;
38         border-style: solid;
39         border-width: 1px;
```

```

40     margin-bottom: 10px;
41     padding: 10px;
42     float: left;
43 }
44 </style>
45 <script type="text/javascript">
46     /* 定义一个函数，专门用来为指定元素绑定单击响应函数
47     参数：
48         idstr 要绑定单击响应函数的对象的id属性
49         fun 事件的回调函数, 当单击元素时, 该函数触发 */
50     function myClick(idStr, fun) {
51         var btn = document.getElementById(idStr);
52         btn.onclick = fun;
53     }
54
55     window.onload = function() {
56         var btn01 = document.getElementById("btn01");
57         btn01.onclick = function() {
58             //查找#bj节点
59             var bj = document.getElementById("bj");
60             alert(bj.innerHTML);
61         };
62
63         var btn02 = document.getElementById("btn02");
64         btn02.onclick = function() {
65             //查找所有li节点
66             var lis = document.getElementsByTagName("li");
67             //alert(lis);
68             //alert(lis.length);
69             for(var i = 0; i < lis.length; i++) {
70                 alert(lis[i].innerHTML);
71             }
72         }
73
74         var btn03 = document.getElementById("btn03");
75         btn03.onclick = function() {
76             //查找name=gender的所有节点
77             var gen = document.getElementsByName("gender");
78             //alert(gen);
79             //alert(gen.length);
80             for(var j = 0; j < gen.length; j++) {
81                 alert(gen[j].value);
82             }
83         }
84
85         var btn04 = document.getElementById("btn04");
86         btn04.onclick = function() {
87             //查找#city下所有li节点
88             var city = document.getElementById("city");
89             var lis = city.getElementsByTagName("li");
90             //alert(lis);
91             for(var i = 0; i < lis.length; i++) {
92                 alert(lis[i].innerHTML);
93             }
94         }
95
96         var btn05 = document.getElementById("btn05");
97         btn05.onclick = function() {

```

```

98         //返回#city下的所有子节点
99         var city = document.getElementById("city");
100         var zi = city.childNodes;
101         //alert(zi.length);
102         var cns = city.children;
103         alert(cns.length);
104     }
105
106     var btn06 = document.getElementById("btn06");
107     btn06.onclick = function() {
108         //返回#phone的第一个子节点
109         var phone = document.getElementById("phone");
110         //alert(phone.children[0].innerHTML) ;IOS
111         //var first = phone.firstChild;
112         var first = phone.firstElementChild;
113         alert(first.innerHTML);//IOS
114     }
115
116     myClick("btn07", function() {
117         //返回#bj的父节点
118         var bj = document.getElementById("bj");
119         var pn = bj.parentNode;
120         //alert(pn.innerHTML);
121         alert(pn.innerText);
122     });
123
124     myClick("btn08", function() {
125         //返回#android的前一个兄弟节点
126         var and = document.getElementById("Android");
127         var ps = and.previousSibling;
128         //var ps = and.previousElementSibling;
129         alert(ps);
130     })
131
132     myClick("btn09", function() {
133         //读取#username的value属性值
134         var un = document.getElementById("username");
135         //文本框的value属性值就是文本框中填写的内容
136         alert(un.value);
137     })
138
139     myClick("btn10", function() {
140         //设置#username的value属性值
141         var un = document.getElementById("username");
142         un.value = "hello";
143     })
144
145     myClick("btn11", function() {
146         //返回#bj的文本值
147         var bj = document.getElementById("bj");
148         //alert(bj.innerHTML);北京
149         //alert(bj.innerText);北京
150         /* 获取北京文本节点
151         var fc = bj.firstChild;
152         alert(fc.nodeValue);//因为文本节点的nodeValue就是文本内容 */
153         alert(bj.firstChild.nodeValue);
154     })
155 }

```

```
156     </script>
157 </head>
158 <body>
159     <div id="total">
160         <div class="inner">
161             <p>
162                 你喜欢哪个城市?
163             </p>
164             <ul id="city">
165                 <li id="bj">北京</li>
166                 <li>上海</li>
167                 <li>东京</li>
168                 <li>首尔</li>
169             </ul>
170             <br>
171             <br>
172
173             <p>
174                 你喜欢哪款单机游戏?
175             </p>
176             <ul id="game">
177                 <li id="rl">红警</li>
178                 <li>实况</li>
179                 <li>极品飞车</li>
180                 <li>魔兽</li>
181             </ul>
182             <br />
183             <br />
184
185             <p>
186                 你手机的操作系统是?
187             </p>
188             <ul id="phone">
189                 <li>IOS</li>
190                 <li id="Android">Android</li>
191                 <li>Window Phone</li>
192             </ul>
193         </div>
194         <div class="inner">
195             <br>
196             gender:
197             <input type="radio" name="gender" value="male" />
198             Male
199             <input type="radio" name="gender" value="female" />
200             Female
201             <br>
202             <br>
203             name:
204             <input type="text" name="name" id="username" value="abcde" />
205         </div>
206     </div>
207     <div id="btnList">
208         <div><button id="btn01">查找#bj节点</button></div>
209         <div><button id="btn02">查找所有li节点</button></div>
210         <div><button id="btn03">查找name=gender的所有节点</button></div>
211         <div><button id="btn04">查找#city下所有li节点</button></div>
212         <div><button id="btn05">返回#city下的所有子节点</button></div>
213         <div><button id="btn06">返回#phone的第一个子节点</button></div>
```



```

214     <div><button id="btn07">返回#bj的父节点</button></div>
215     <div><button id="btn08">返回#android的前一个兄弟节点</button></div>
216     <div><button id="btn09">读取#username的value属性值</button></div>
217     <div><button id="btn10">设置#username的value属性值</button></div>
218     <div><button id="btn11">返回#bj的文本值</button></div>
219 </div>
220 </body>
221 </html>

```

3. 全选练习

你爱好的运动是？ ☐ 全选/全不选

☒ 足球 ☒ 篮球 ☐ 羽毛球 ☒ 乒乓球

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title></title>
6      <script type="text/javascript">
7        window.onload = function() {
8          var checkedAllBtn = document.getElementById("checkedAllBtn");
9          var checkedAllBox = document.getElementById("checkedAllBox");
10         checkedAllBtn.onclick = function() {
11           //1. #checkedAllBtn全选按钮
12           var items = document.getElementsByName("items");
13           for(var i = 0; i < items.length; i++) {
14             //通过多选框的checked属性可以获取或设置多选框的选中状态
15             items[i].checked = true;
16           }
17           checkedAllBox.checked = true;
18         }
19
20         var checkedNoBtn = document.getElementById("checkedNoBtn");
21         var checkedAllBox = document.getElementById("checkedAllBox");
22         checkedNoBtn.onclick = function() {
23           //2. #checkedNoBtn
24           var items = document.getElementsByName("items");
25           for(var i = 0; i < items.length; i++) {
26             items[i].checked = false;
27           }
28           checkedAllBox.checked = false;
29         }
30
31         var checkedRevBtn = document.getElementById("checkedRevBtn");
32         var items = document.getElementsByName("items");
33         checkedRevBtn.onclick = function() {
34           //3. #checkedRevBtn
35           checkedAllBox.checked = true;
36           for(var i = 0; i < items.length; i++) {
37             /* if(items[i].checked) {
38               items[i].checked = false;
39             }else{
40               items[i].checked = true;

```

```

41         } */
42         items[i].checked = !items[i].checked;
43         if(!items[i].checked){
44             //一旦进入判断则证明不是全选状态，将checkedAllBox设置为没选中状态
45             checkedAllBox.checked = false;
46         }
47     }
48 }
49
50 var sendBtn = document.getElementById("sendBtn");
51 sendBtn.onclick = function() {
52     //4. #sendBtn
53     var items = document.getElementsByName("items");
54     for(var i = 0; i < items.length; i++) {
55         if(items[i].checked) {
56             alert(items[i].value);
57         }
58     }
59 }
60
61 var checkedAllBox = document.getElementById("checkedAllBox");
62 checkedAllBox.onclick = function() {
63     //5. #checkedAllBox
64     //alert(this === checkedAllBox); //true, 在事件的响应函数中，响应函数是给谁绑定的
    this就是谁
65     var items = document.getElementsByName("items");
66     for(var i = 0; i < items.length; i++) {
67         items[i].checked = this.checked;
68     }
69 }
70 }
71 //6. items
72 /* 如果四个全选框都选中，则checkedAllBox也应该选中
73 如果四个全选框没都选中，则checkedAllBox也不应该选中 */
74 //为四个多选框分别绑定单击响应函数
75 //6. items
76 var items = document.getElementsByName("items");
77 for(var i = 0; i < items.length; i++) {
78     items[i].onclick = function() {
79         checkedAllBox.checked = true;
80         for (var j = 0; j < items.length; j++) {
81             //判断四个多选框是否全选
82             //只要有一个没选中则就不是全选
83             if(!items[j].checked) {
84                 //一旦进入判断则证明不是全选状态，将checkedAllBox设置为没选中状态
85                 checkedAllBox.checked = false;
86                 //一旦进入判断，则已经得出结果，不用再继续执行循环
87                 break;
88             }
89         }
90     }
91 }
92 }
93 </script>
94 </head>
95 <body>
96 <form action="" method="post">
97     你爱好的运动是? <input type="checkbox" id="checkedAllBox" />全选/全不选

```

```

98     <br />
99     <input type="checkbox" name="items" id="" value="足球" />足球
100    <input type="checkbox" name="items" id="" value="篮球" />篮球
101    <input type="checkbox" name="items" id="" value="羽毛球" />羽毛球
102    <input type="checkbox" name="items" id="" value="乒乓球" />乒乓球
103    <br/>
104    <input type="button" name="" id="checkedAllBtn" value="全 选" />
105    <input type="button" name="" id="checkedNoBtn" value="全不选" />
106    <input type="button" name="" id="checkedRevBtn" value="反 选" />
107    <input type="button" name="" id="sendBtn" value="提 交" />
108    </form>
109    </body>
110    </html>

```



4. DOM增删改练习

- **document.createElement()**：可以创建一个 元素节点对象， 他需一个标签名作为参数， 将会根据该标签

名常见元素节点对象，并将创建好的对象作为返回值返回 eg: `var li = document.createElement("li");`

- **document.createTextNode()**：可以用来 创建一个文本节点对象， 需要一个文本内容作为参数， 将会根据

该内容创建文本节点，并将新的节点返回

- **appendChild()**：向一个父节点中添加一个新的子节点， 用法: 父节点.appendChild(子节点);
- **appendChild()**：向一个父节点中添加一个新的子节点， 用法: 父节点.appendChild(子节点);
- **insertBefore()**：可在指定的子节点迁出如新的节点 语法: 父节点.insertBefore(新节点, 旧节点);
- **replaceChild()**：可使用指定的子节点替换已有的子节点 语法: 父节点.replaceChild(新节点, 旧节点);
- **removeChild()**：可以删除一个子节点 语法: 父节点.removeChild(子节点);

你喜欢那个城市？

北京 上海 东京 首尔

建一个“广州”节点，添加到#city下(法1)

建一个“广州”节点，添加到#city下(法2)

将“广州”节点插入到#bj前面

使用“广州”节点替换#bj节点

删除#bj节点

读取#city内的HTML代码

设置#bj内的HTML代码

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title></title>
6      <style type="text/css">
7        body{

```

```

8      width: 600px;
9      margin: 0 auto;
10     }
11     #total{
12         width: 300px;
13         border: black 1px solid;
14         float: left;
15         margin-top: 10px;
16         padding-left: 10px;
17     }
18     ul{
19         list-style: none;
20         padding: 0;
21         margin: 0;
22     }
23     li{
24         float: left;
25         border: black 1px solid;
26         margin: 5px;
27         padding: 5px;
28         background-color: chartreuse;
29     }
30     #btnList{
31         margin-left: 10px;
32         float: left;
33     }
34     button{
35         width: 250px;
36         margin-top: 10px;
37     }
38 </style>
39 <script type="text/javascript">
40     //DOM增删改练习
41     window.onload = function() {
42         function myClick (idStr, fun){
43             var btn = document.getElementById(idStr);
44             btn.onclick = fun;
45         }
46
47         //建一个“广州”节点，添加到#city下(法1)
48         myClick("btn01",function() {
49             //创建li元素节点
50             var li = document.createElement("li");
51             //创建广州文本节点
52             var gzText = document.createTextNode("广州");
53             li.appendChild(gzText);
54             //获取id为city的节点
55             var city = document.getElementById("city");
56             city.appendChild(li);
57         })
58
59
60         //建一个“广州”节点，添加到#city下(法2)
61         myClick("btn07",function() {
62             var city = document.getElementById("city");
63             /* 使用innerHTML也可以实现完成DOM的增删改操作，但他是将所有的li都重新建立，动作太大
64             我们一般两种方式结合使用 */
65             //city.innerHTML = "<li>广州</li>";

```

```

66         //结合两种方法(推荐使用)
67         var li = document.createElement("li");
68         li.innerHTML = "广州";
69         city.appendChild(li);
70     })
71
72     //将“广州”节点插入到#bj前面
73     myClick("btn02",function() {
74         var bj = document.getElementById("bj");
75         var li = document.createElement("li");
76         var gzText = document.createTextNode("广州");
77         li.appendChild(gzText);
78         //获取city
79         var city = document.getElementById("city");
80         city.insertBefore(li,bj);
81     })
82
83     //使用“广州”节点替换#bj节点
84     myClick("btn03",function() {
85         var bj = document.getElementById("bj");
86         var li = document.createElement("li");
87         var gzText = document.createTextNode("广州");
88         li.appendChild(gzText);
89         var city = document.getElementById("city");
90         city.replaceChild(li,bj);
91     })
92
93     //删除#bj节点
94     myClick("btn04",function() {
95         var bj = document.getElementById("bj");
96         var city = document.getElementById("city");
97         //city.removeChild(bj);
98         bj.parentNode.removeChild(bj); //常用
99     })
100
101     //读取#city内的HTML代码
102     myClick("btn05",function() {
103         var city = document.getElementById("city");
104         alert(city.innerHTML);
105     })
106
107     //设置#bj内的HTML代码
108
109     myClick("btn06",function() {
110         var bj = document.getElementById("bj");
111         bj.innerHTML = "郑州";
112     })
113 }
114 </script>
115 </head>
116 <body>
117     <div id="total">
118         <div class="inner">
119             <p>
120                 你喜欢那个城市?
121             </p>
122             <ul id="city">
123                 <li id="bj">北京</li>

```

```

124         <li>上海</li>
125         <li>东京</li>
126         <li>首尔</li>
127     </ul>
128 </div>
129 </div>
130 <div id="btnList">
131     <div><button id="btn01">建一个“广州”节点，添加到#city下(法1)</button></div>
132     <div><button id="btn07">建一个“广州”节点，添加到#city下(法2)</button></div>
133     <div><button id="btn02">将“广州”节点插入到#bj前面</button></div>
134     <div><button id="btn03">使用“广州”节点替换#bj节点</button></div>
135     <div><button id="btn04">删除#bj节点</button></div>
136     <div><button id="btn05">读取#city内的HTML代码</button></div>
137     <div><button id="btn06">设置#bj内的HTML代码</button></div>
138 </div>
139 </body>
140 </html>

```

✓ 5. 向表格中添加和删除记录练习

Name	Email	Salary	
Tom	tom@tom.com	5000	Delete
Jerry	jerry@sohu.com	8000	Delete
Bob	bob@tom.com	10000	Delete

添加新员工

name:

email:

salary:

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title></title>
6      <style type="text/css">
7        #employeeTable{
8          width: 300px;
9          margin: 0 auto;
10         border-spacing: 1px;
11         background-color: black;
12       }
13     }
14     #formDiv{
15       margin: 50px auto;
16       width: 250px;

```

```

17         border: #000000 1px solid;
18         padding: 20px;
19     }
20     td, th {
21         background-color: white;
22     }
23 </style>
24 <script type="text/javascript">
25
26     /* 删除tr的响应函数 我们在新添加了员工之后点击删除会没有效果，因为点击的响应函数已经执行完了
27
28         因此我们还需要在添加新员工功能中添加删除响应函数，因此我们把删除响应函数代码提取出来，
29
30         在下方直接调用*/
31     function delA() {
32         //点击超链接后，删除所在行员工的信息
33         var tr = this.parentNode.parentNode;
34
35         //获取要删除员工的名字(两种)
36         //var name = tr.getElementsByTagName("td")[0].innerHTML;
37         var name = tr.children[0].innerHTML;
38
39         //alert("确认删除吗？")这个alert会弹出只能确认的窗口
40         /* confirm()用于弹出一个带有确认和取消按钮的提示框，
41            需要一个字符串作为参数，会作为提示文字显示，如果用户点击确认会返回true，点击取消会返回false*/
42         var flag = confirm("确认删除"+name+"吗");
43         if(flag) {
44             tr.parentNode.removeChild(tr);
45         }
46
47         /* 点击超链接以后，超链接会跳转页面，这个是超链接的默认行为
48            但此时我们不想让他跳转，可以在响应函数最后return false;取消，或者href="javascript:;"
49
50            */
51         return false;
52     }
53
54     window.onload = function() {
55         /* 删除功能实现 */
56         var allA = document.getElementsByTagName("a");
57         /* for循环会在页面加载完成之后立即执行，而allA的onclick响应函数会在点击超链接时才执行
58
59            当响应函数执行时，for循环早已执行完毕，
60            当使用另一种方法在for循环写allA的onclick响应函数时：
61            for(var i = 0;i < allA.length;i++){
62            allA[i].onclick = function() {
63
64                本来this和allA[i]是一样的，但for循环执行完了i的值已经变了，allA[i]的值是循环体执行完了的i的值
65
66                var tr = this.parentNode.parentNode;
67                .....
68            };
69            } */
70         for(var i = 0;i < allA.length;i++) {
71             allA[i].onclick = delA;
72         }
73
74         /* 添加员工功能（两种方法）*/

```

```

69     var addEmpButton = document.getElementById("addEmpButton");
70     addEmpButton.onclick = function() {
71         var name = document.getElementById("empName").value;
72         var salary = document.getElementById("salary").value;
73         var email = document.getElementById("email").value;
74
75         /* //(方法一) 需要我们将获取到的信息添加到tr中
76         var tr = document.createElement("tr");
77         var nameTd = document.createElement("td");
78         var emailTd = document.createElement("td");
79         var salaryTd = document.createElement("td");
80         var aTd = document.createElement("td");
81         var a = document.createElement("a");
82         var nameText = document.createTextNode(name);
83         var emailText = document.createTextNode(email);
84         var salaryText = document.createTextNode(salary);
85         var delText = document.createTextNode("Delete");
86         nameTd.appendChild(nameText);
87         emailTd.appendChild(emailText);
88         salaryTd.appendChild(salaryText);
89         a.appendChild(delText);
90         aTd.appendChild(a);
91         tr.appendChild(nameTd);
92         tr.appendChild(emailTd);
93         tr.appendChild(salaryTd );
94         tr.appendChild(aTd);
95         a.href = "javascript:;";
96         a.onclick = delA; */
97
98         //(推荐方法二) 需要我们将获取到的信息添加到tr中
99         var tr = document.createElement("tr");
100         tr.innerHTML = "<td>"+name+"</td>"+
101             "<td>"+email+"</td>"+
102             "<td>"+salary+"</td>"+
103             "<td><a href=' javascript:;'>Delete</a></td>";
104         var a = tr.getElementsByTagName("a")[0];
105         a.onclick = delA;
106
107         var employeeTable = document.getElementById("employeeTable");
108         //浏览器会自动给table添加tbody, 因此我们要获取employeeTable的tbody
109         var tbody = employeeTable.getElementsByTagName("tbody")[0];
110         //将tr添加到tbody中, 也可以使用innerHTML添加但是他会将内容全部更新, 也会导致超链
接无效
111         /* (不建议使用)tbody.innerHTML += "<tr>"+
112             "<td>"+name+"</td>"+
113             "<td>"+email+"</td>"+
114             "<td>"+salary+"</td>"+
115             "<td><a href=' javascript:;'>Delete</a></td>"
116             +"</tr>"; */
117         tbody.appendChild(tr); //(建议使用)
118     }
119 }
120 </script>
121 </head>
122 <body>
123     <table id="employeeTable">
124         <tr>
125             <th>Name</th>

```



```

126         <th>Email</th>
127         <th>Salary</th>
128         <th>&nbsp;</th>
129     </tr>
130     <tr>
131         <td>Tom</td>
132         <td>tom@tom.com</td>
133         <td>5000</td>
134         <td><a href="javascript:;">Delete</a></td>
135     </tr>
136     <tr>
137         <td>Jerry</td>
138         <td>jerry@sohu.com</td>
139         <td>8000</td>
140         <td><a href="deleteEmp?id=002">Delete</a></td>
141     </tr>
142     <tr>
143         <td>Bob</td>
144         <td>bob@tom.com</td>
145         <td>10000</td>
146         <td><a href="deleteEmp?id=003">Delete</a></td>
147     </tr>
148 </table>
149 <div id="formDiv">
150     <h4>添加新员工</h4>
151     <table >
152         <tr>
153             <td class="word">name:</td>
154             <td class="inp">
155                 <input type="text" name="empName" id="empName">
156             </td>
157         </tr>
158         <tr>
159             <td class="word">email:</td>
160             <td class="inp">
161                 <input type="text" name="email" id="email">
162             </td>
163         </tr>
164         <tr>
165             <td class="word">salary:</td>
166             <td class="inp">
167                 <input type="text" name="salary" id="salary">
168             </td>
169         </tr>
170         <tr>
171             <td colspan="2" align="center">
172                 <button id="addEmpButton" value="abc">Submit</button>
173             </td>
174         </tr>
175     </table>
176 </div>
177 </body>
178 </html>

```