

jQuery

jQuery

- jQuery介绍

- jQuery的Hello程序示例

 - 常见问题

- jQuery的核心函数

 - 四个作用

- 如何区分jQuery对象和DOM对象

 - DOM对象

 - jQuery对象

 - jQuery对象的本质

 - jQuery对象和DOM对象使用区别

- DOM对象和jQuery对象互转

 - 1.DOM对象转化为jQuery对象

 - 2.jQuery对象转化为DOM对象

- jQuery选择器

 - 基础选择器

 - 层级选择器

 - 过滤选择器

 - 基本过滤器

 - 内容过滤选择器

 - 属性过滤选择器

 - 表单过滤选择器

 - 表单对象属性过滤选择器

 - 补充

- jQuery元素的筛选

- jQuery的属性操作

 - html()、text()、val方法

 - val方法同时设置多个表单项的值

 - att()和prop()方法

- DOM的增、删、改

 - 内部插入

 - 外部插入

 - 替换

 - 删除

- jQuery练习

 - 全选、全不选、反选

两列表的内容的相互移动

动态添加、删除表格记录

品牌展示

图片跟随

CSS样式操作

jQuery动画操作

基本动画

淡入淡出动画

jQuery事件操作

原生JavaScript和jQuery页面加载完成之后的区别

他们分别是在什么时候触发？

他们触发的顺序？

他们执行的次数？

jQuery中其他的事件处理办法

事件的冒泡

什么是事件的冒泡？

如何阻止事件冒泡？

JavaScript事件对象

如何获取JavaScript事件对象？

jQuery介绍

jQuery是JavaScript和查询(Query)，是辅助JavaScript开发的js类库。jQuery的核心思想是write less, do more,它实现了很多浏览器的兼容问题。查询可点击[jQuery API 中文在线手册](#)和[菜鸟教程](#)

jQuery的Hello程序示例

```

1 <script type="text/javascript"
  src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
2 <script type="text/javascript">
3     $(function(){//表示页面完成后加载之后，相当于
      window.onload=function(){
4         var $btnObj = $("#btnId");//表示按id查询标签对象
5         $btnObj.click(function(){//绑定单击事件
6             alert("jQuery的单击事件");
7         })
8     })
9 </script>

```

常见问题

- 使用jQuery需用jQuery库
- \$是jQuery的函数

jQuery的核心函数

\$为jQuery的核心函数，可完成jQuery的很多功能。

\$()就是调用此函数

四个作用

- 1.传入参数为函数时，表示页面加载完成之后，自动调用。
- 2.传入参数为HTML字符串时，根据这个字符串创建元素节点对象。

```

1 $(function(){
2     alert("页面加载完成之后，自动调用");
3     $(
4         "<div>" +
5         "<span>div-span1</span>"
6         "<span>div-span2</span>"
7         "</div>"
8     ).appendTo("body");
9 });

```

3.传入参数为选择器字符串时，根据输入的选择器类型查询标签对象。

- \$("#id属性值");
- \$("标签名");
- \$(".class属性值");

4.传入参数为DOM对象时，会把DOM对象转化为jQuery对象。

如何区分jQuery对象和DOM对象

DOM对象

- 通过getElementById() 查询出来的对象是DOM对象。
- 通过getElementByName() 查询出来的对象是DOM对象。
- 通过getElementsByTagName() 查询出来的对象是DOM对象。
- 通过createElement() 查询出来的对象是DOM对象。
- DOM对象alert()出来的效果是[object HTMLxxxElement]

jQuery对象

- 通过jQuery提供的API创建的对象是jQuery对象。
- 通过jQuery包装的DOM对象是jQuery对象。
- 通过jQuery提供的API查询到的对象是jQuery对象。
- jQuery对象alert()出来的效果是[object Object]

jQuery对象的本质

jQuery对象是DOM对象的数组 + jQuery提供的一系列功能函数。

jQuery对象和DOM对象使用区别

jQuery对象与DOM对象的属性和方法无法通用。

DOM对象和jQuery对象互转

1.DOM对象转化为jQuery对象

1. 先有DOM对象
2. \$(DOM对象)就可以转换为jQuery对象

```
1 var $obj = $(DOM对象);
```

2.jQuery对象转化为DOM对象

1. 先有jQuery对象
2. jQuery对象[下标]取出相应的DOM对象

```
1 var DOM = $obj[下标]
```

jQuery选择器

获取jQuery相关资料请[点击!](#)

基础选择器

```
1 $(function(){//
2     $("#btn1").click(function(){
3         //css()方法可以设置和获取样式
4         $("#one").css("样式");
5     });
6     $("#btn2").click(function(){
7         $(".mini").css("样式");
8     });
9     $("#btn3").click(function(){
10        $("div").css("样式");
11    })
12    $("#btn4").click(function(){
13        $("*").css("样式");
14    })
15    $("#btn5").click(function(){
```

```

16     $("span,#two").css("样式");
17     })
18 })

```

层级选择器

- ancestor descendant 后代选择器
 - 在给定的祖先元素下匹配所有的后代元素
- parent > child 子元素选择器
 - 在给定的父元素下匹配所有的子元素
- prev + next 相邻兄弟选择器
 - 匹配所有紧接在prev元素后的next元素(单个)
- prev ~ siblings 之后的兄弟元素选择器
 - 匹配prev元素之后的所有siblings元素

```

1  $("from input");
2  $("form > input")
3  $("label + input");
4  $("form ~ input");

```

过滤选择器

基本过滤器

```

1  :first //匹配第一个
2  $("li:first");
3  :last //匹配第二个
4  $("li:last");
5  :not(selector)
6  $("input:not(:checked)");
7  :even //匹配所有索引值为偶数的元素，从0开始计数
8  $("tr:even");
9  :odd //匹配所有索引值为奇数的元素，从0开始计数
10 $("tr:odd");
11 :eq(index) //匹配一个给定索引值的元素，从0开始计数
12 $("tr:eq(1)");

```

```

13 :gt(index) //匹配所有大于给定索引值的元素
14 $("tr:gt(0)");
15 :lt(index) //匹配所有小于给定索引值的元素
16 $("tr:lt(2)");
17 :header //匹配标题元素
18 $(":header").css("样式");
19 :animated //匹配所有正在执行动画效果的元素
20

```

内容过滤选择器

```

1 :contains(text) //匹配包含给定文本的元素
2 $("div:contains('John')");
3 :empty //匹配所有不包含子元素或者文本的空元素
4 $("td:empty");
5 :has(selector) //匹配含有子元素或者文本的元素的元素
6 $("div:has(p).addClass('test')");
7 :parent //匹配含有选择器所匹配的元素（非空）
8 $("td:parent");

```

属性过滤选择器

```

1 [attribute] //匹配包含给定属性的元素
2 $("div[id]");
3 [attribute=value] //匹配包含给定属性是某个特定值的元素
4 $("input[name='newsletter']").attr("checked",true);
5 [attribute!=value] //匹配所有属性不等于特定值的元素，或者不含有指定的属性
6 $("input[name!='newsletter']").attr("checked",true);
7 [attribute^=value] //匹配给定的属性是以某些值开始的元素
8 $("input[name^='news']");
9 [attribute$=value] //匹配给定的属性是以某些值结尾的元素
10 $("input[name$='letter']");
11 [attribute*=value] //匹配给定的属性是以包含某些值的元素
12 $("input[name*='man ']");
13 [attrSet1][attrSet2][attrSet3] //复合属性选择器，需要同时满足多个条件时使用
14 $("input[id][name$='man']");

```

表单过滤选择器

```
1  :input //匹配所有 input,textarea,select和button元素
2  $("input");
3  :text //匹配所有的单行文本框
4  $("text");
5  :password //匹配所有的密码框
6  $("password");
7  :radio //匹配所有的单选按钮
8  $("radio");
9  :checkbox //匹配所有的复选框
10 $("checkbox");
11 :submit //匹配所有的提交按钮
12 $("submit");
13 :image //匹配所有的图像
14 $("image");
15 :reset //匹配所有的重置按钮
16 $("reset");
17 :button //匹配所有的所有按钮
18 $("button");
19 :file //匹配所有的文件域
20 $("file");
21 :hidden //匹配所有不可见元素，或者type为hidden的元素
22 $("hidden");
```

表单对象属性过滤选择器

```
1  :enabled //匹配所有可用元素
2  $("input:enabled");
3  :disabled //匹配所有不可用元素
4  $("input:disabled");
5  :checked //匹配所有选中的被选中元素（复选框、单选框，不包括select中的
6  option）
7  $("input:checked");
8  :selected //匹配所有选中的option元素
9  $("select option:selected");
```


补充

```
1 //val()可以操作表单的value属性值
2 $(".text:enabled").val();
3
4 //each方法是jQuery对象提供用来遍历元素的方法
5 //在遍历的function函数，有一个this对象，就是当前遍历到的DOM对象
6 $checkboxes.each(function(){
7     alert(this.value);
8 });
9
10 $("#btn5").click(function(){
11     var $options = $("select option:selected");
12     $options.each(function(){
13         alert(this.innerHTML);
14     });
15 });
```

jQuery元素的筛选

[jQuery API 中文在线手册](#)

```
1 eq(index)
2 first()
3 last()
4 filter(expr)
5 is(exp)
6 has(expr)
7 not(exp)
8 children(exp)
9 find(exp)
10 next()
11 nextAll()
12 nextUntil()
13 parent()
14 prev(exp)
15 prevAll()
16 prevUntil(exp)
```

```
17 siblings(exp)
18 add()
```

jQuery的属性操作

html()、text()、val方法

html()

- 可设置和获取起始标签和结束标签中的内容。跟DOM属性innerHTML一样。

text()

- 可设置和获取起始标签和结束标签中的文本。跟DOM属性innerText一样。

val()

- 可设置和获取表单项的value属性值。

val方法同时设置多个表单项的值

```
1  $(function(){
2      //不传参数，是获取，传递参数是设置
3      alert($("#div").html());
4      $("#div").html("可往div中设置内容");
5
6      //不传参数，是获取，传递参数是设置
7      alert($("#div").text());
8      $("#div").text("可设置div中的文本");
9
10     //不传参数，是获取，传递参数是设置
11     $("#button").click(function(){
12         alert("#username").val();
13         $("#username").val("");
14     })
15
16     //批量操作单选
17     $(function(){
18         //批量操作单选
```

```

19     $(":radio").val(["radio01"]);
20     //批量操作筛选框的选中状态
21     $(":checkbox").val(["checkbox01", "checkbox02"]);
22     //批量操作多选的下拉框选中状态
23     $("#multiple").val(["mul1", "mul2"]);
24     //批量操作单选的下拉框选中状态
25     $("#single").val(["sin1"]);
26     //批量操作
27     $(":radio, :checkbox, ...").val(["radio01", ...]);
28 }
29 }

```

att()和prop()方法

attr()

- 可以设置和获取属性的值
- 不推荐操作checked、readOnly、selected、disabled等
- 还可以操作自定义属性

prop()

- 可以设置和获取属性的值，只推荐操作checked、readOnly、selected、disabled等

```

1  $(function(){
2      $(":checkbox:first").attr("name");//获取
3      $(":checked:first").attr("name", "zjx");//设置
4  })

```

DOM的增、删、改

内部插入

append()

```

1  a.appendTo(b) //把a插入到b子元素末尾，成为最后一个子元素
2  $("<h1>标题</h1>").appendTo("div");

```

prependTo()

```
1 a.prependTo(b) //把a插入到b所有子元素前面，成为第一个子元素
```

外部插入

insertAfter()

```
1 a.insertAfter(b) //得到ba，平级插入
```

insertBefore()

```
1 a.insertBefore(b) //ab
```

替换

replaceWith()

```
1 a.replaceWith(b) //用b替换a
```

replaceAll()

```
1 a.replaceAll(b) //用a替换所有的b
```

删除

remove()

```
1 a.remove() //删除a标签
```

empty()

```
1 a.empty() //清空a标签的内容
```

jQuery练习

全选、全不选、反选

1

两列表的内容的相互移动

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset = "UTF-8">
5     <title></title>
6     <script type = "text/javascript" src =
7         "https://code.jquery.com/jquery-3.5.1.min.js"></script>
8     <script type = "text/javascript">
9         //页面加载完成
10        $(function(){
11            //第一个按钮 / 选中添加到右边
12            $("button:eq(0)").click(function(){
13                $("select:eq(0)
14option:selected").appendTo($("select:eq(1)"));
15            });
16            $("button:eq(1)").click(function(){
17                $("select:eq(0)
18option").appendTo($("select:eq(1)"));
19            });
20            $("button:eq(2)").click(function(){
21                $("select:eq(1)
22option:selected").appendTo($("select:eq(0)"));
23            });
24            $("button:eq(3)").click(function(){
25                $("select:eq(1)
26option").appendTo($("select:eq(0)"));
27            });
28        });
```

```

27     </script>
28
29 </head>
30 <body>
31 <div id = "left">
32     <select multiple = "multiple" name = "se101">
33         <option value = "opt01">选项1</option>
34         <option value = "opt02">选项2</option>
35         <option value = "opt03">选项3</option>
36         <option value = "opt04">选项4</option>
37         <option value = "opt05">选项5</option>
38         <option value = "opt06">选项6</option>
39         <option value = "opt07">选项7</option>
40         <option value = "opt08">选项8</option>
41     </select>
42
43     <button>选中添加到右边</button>
44     <button>全部添加到右边</button>
45 </div>
46 <div id = "right">
47     <select multiple = "multiple" name = "se102"></select>
48     <button>选中删除到左边</button>
49     <button>全部删除到左边</button>
50 </div>
51 </body>
52 </html>

```

动态添加、删除表格记录

```

1 <html>
2     <head>
3         <meta charset = "UTF-8">
4         <title></title>
5         <script type = "text/javascript"
src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
6         <script type = "text/javascript">
7             $(function(){
8                 //复用
9                 var deleteFun = function(){

```

```

10 //在事件响应的function函数中，有一个this对
    象。这个this对象是当前正在响应事件的DOM对象
11     var $trObj = $(this).parent().parent();
12     var name =
    $teObj.find("td:first").text();
13
14 //confirm是JavaScript语言提供的一个确认提示
    框函数，你给他传什么，它就提示什么
15 //当用户点击了确定，就返回true。当用户点击了取
    消就返回false
16     if(confirm("你确定要删除[" + name + "]
    吗? ")){
17         $trObj.remove();
18     }
19     //return false; 可以阻止元素的默认行为
20     return false;
21 }
22
23 //给submit按钮绑定单击事件
24 $("#addEmpButton").click(function(){
25     //获取输入框的姓名、邮箱、工资的内容
26     var name = $("#empName").val();
27     var email = $("#email").val();
28     var salary = $("#salary").val();
29     //创建一个行标签对象，添加到显示器数据的表格中
30     var $trObj = ("<tr> +
31     "<td>" + name + "</td>" +
32     "<td>" + email + "</td>" +
33     "<td>" + salary + "</td>" +
34     "<td><a href = \"deleteEmp?id =
    001\">Delete</a></td>" +
35     "</tr>");
36     //添加到显示数据的表格中
37     $trObj.appendTo("#employeeTable");
38     //给添加的行的a标签绑上单击事件
39     $trObj.find("a").click(deleteFun);
40     //给删除的a标签绑定单击事件
41     $("a").click(deleteFun);
42 });

```

```

43         });
44     </script>
45 </head>
46 <body>
47     <table id = "">
48         <tr>
49             <th>name</th>
50             <th>Email</th>
51             <th>Salary</th>
52             <th> </th>
53         </tr>
54         <tr>
55             <td>Tom</td>
56             <td>tom@tom.com</td>
57             <td>5000</td>
58             <td><a href = "deleteEmp?id = 001">Delete</a>
59 </td>
60         </tr>
61         <tr>
62             <td>Jerry</td>
63             <td>jerry@sohu.com</td>
64             <td>8000</td>
65             <td><a href = "deleteEmp?id = 002">Delete</a>
66 </td>
67         </tr>
68         <tr>
69             <td>Bob</td>
70             <td>bob@tom.com</td>
71             <td>10000</td>
72             <td><a href = "deleteEmp?id = 003">Delete</a>
73 </td>
74         </tr>
75     </table>
76     <div id = "">
77         <tr>
78             <td class = "">name</td>
79             <td class = "">
80                 <input type = "text" name = "empName" id =
81 "empName" />

```



```

78         </td>
79     </tr>
80     <tr>
81         <td class = "">email</td>
82         <td class = "">
83             <input type = "text" name = "email" id =
"email" />
84         </td>
85     </tr>
86     <tr>
87         <td class = "">salary</td>
88         <td class = "">
89             <input type = "text" name = "salary" id =
"salary" />
90         </td>
91     </tr>
92     <tr>
93         <td colspan = "2" align = "center">
94             <button id = "addEmpButton" value = "abc">
95                 Submit
96             </button>
97         </td>
98     </tr>
99 </div>
100 </body>
101 </html>

```

品牌展示

```
1
```

图片跟随

```

1 <script type="text/javascript"
  src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
2 <script type="text/javascript">
3     $(function(){
4         $("#small").bind("mouseover mouseout", function(event){

```

```
5         if(event.type == "mouseover"){
6             $("#showBig").show();
7         }else if(event.type == "mouseout"){
8             $("#showBig").hide();
9         }else if(event.type == "mousemove"){
10            $("#showBig").offset({
11                //避免两个图片重叠而导致event.type的判定出现问题
12                left:event.pageX + 10;
13                top:event.pageY + 10;
14            });
15        }
16    });
17 });
18 </script>
```

CSS样式操作

```
1  addClass(样式1 样式2 ...); //添加样式
2
3  removeClass(); //删除样式
4
5  toggleClass(); //有则删除，无则添加
6
7  offset(); //获取和设置元素的坐标还可添加
8  $divEle.offset({
9      top:100,
10     left:50
11 });
```

jQuery动画操作

基本动画

```
1 show() //显示
2 $("#btn1").click(function(){
3     $("#div1").show(2000,function(){
4         alert("show动画完成");
5     });
6 });
7
8 hide() //隐藏
9
10 toggle() //显示隐藏切换
```

以上动画方法都可以添加参数

1. 第一个参数是动画执行的时长，以毫秒为单位
2. 第二个参数是动画的回调函数（动画完成后自动调用的函数）

淡入淡出动画

```
1 fadeIn() //淡入（慢慢可见）
2
3 fadeOut() //淡出（慢慢消失）
4
5 fadeTo() //在指定时长内慢慢的将透明度修改到指定的值(0 ~ 1)
6
7 fadeToggle() //淡化切换
```

jQuery事件操作

原生JavaScript和jQuery页面加载完成之后的区别

```
1 $(function(){
2
3 });
4
5 window.onload = function(){
6
7 };
```

他们分别是在什么时候触发？

1. jQuery的页面加载完成之后是浏览器的内核解析完页面的标签创建好DOM对象之后就会马上执行。
2. 原生JavaScript的页面加载完成之后，除了要等浏览器内核解析完标签创建好DOM对象，还要等标签显示时需要的内容加载完成。

他们触发的顺序？

1. jQuery页面加载完成后先执行
2. 原生JavaScript页面加载完成后执行

他们执行的次数？

1. 原生JavaScript的页面加载完成后，只会执行最后一次的赋值函数
2. jQuery的页面加载完成之后，是把所有的注册的function函数，按顺序依次执行。

jQuery中其他的事件处理办法

```
1  click() //可以绑定单击事件以及触发单击事件
2  $(function(){
3      $("h5").click(function(){//传function是绑定事件
4          alert("");
5      });
6      $("button").click(function(){//不传function是触发事件
7          $("h5").click();
8      });
9  });
10
11 mouseover() //鼠标移入事件
12
13 mouseout() //鼠标移出事件
14
15 bind() //可以给元素一次性绑定一个或多个事件
16
17 one() //与bind()类似，但one()方法绑定的事件只会响应一次
18
19 live() //可以用来绑定选择器匹配的所有元素事件， 尽管这个元素是后面动态创建出来的也有效
```

20

21 `unbind()` //解除事件的绑定

事件的冒泡

什么是事件的冒泡？

是指，父子元素同时监听同一个事件。当触发子元素的事件的时候，同一个事件也被传递到了父元素的事件里去响应。

如何阻止事件冒泡？

在事件函数体内，`return false;`可以阻止事件的冒泡传递。

JavaScript事件对象

事件对象，是封装有触发的事件信息的一个JavaScript对象。我们重点关系是怎么拿到这个JavaScript的事件对象，以及使用。

如何获取JavaScript事件对象？

在给元素绑定事件的时候，在事件的`function(event)`参数列表中添加一个参数，这个参数名习惯取名为`event`。这个`event`就是JavaScript传递参事件处理函数的对象。

1. 原生JavaScript获取事件对象

```
1 window.onload = function(){
2     document.getElementById("areaDiv").onclick = function(event){
3         console.log(event);
4     }
5 }
```

2. jQuery获取事件对象

```
1 $(function(){
2     $("#areaDiv").click(function(event){
3         console.log(event);
4     });
5 });
```

3. 使用**bin()**同时对多个事件绑定同一个函数。怎么获取当前操作时什么事件

```
1 $(function(){
2     $("#areaDiv").bind("mouseover mouseout", function(event){
3         if(event.type == "mouseover"){
4             console.log("鼠标移入");
5         }else if(event.type == "mouseout"){
6             console.log("鼠标移出");
7         }
8     });
9 });
```

[返回文首](#)