

# JQuery

## 1、\$是函数名

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title></title>
6          <script type="text/javascript">
7              //单机按钮获取 文本框的 value值
8              function fun1(){
9                  //通过js中的 id 获取 dom 对象
10                 // var obj = document.getElementById("txt1");
11                 var obj = $("txt1");
12                 alert(obj.value);
13             }
14             function fun2(){
15                 //通过js中的 id 获取 dom 对象
16                 // var obj = document.getElementById("txt2");
17                 var obj = $("txt2");
18                 alert(obj.value);
19             }
20
21             //自定义方法，获取js中的dom对象,$ 是一个函数的名称
22             function $(domId){
23                 var obj = document.getElementById(domId);
24                 return obj;
25             }
26         </script>
27     </head>
28     <body>
29         <input type="text" id="txt1" value="我是txt1" /><br>
30         <input type="text" id="txt2" value="我是txt2" /><br>
31         <input type="button" value="单击按钮 1" onclick="fun1()" /><br>
32         <input type="button" value="单击按钮 2" onclick="fun2()" />
33     </body>
34 </html>
```

## 2、JQuery简介

- jQuery是是一款跨主流浏览器的 JavaScript 库，封装了 JavaScript 相关方法调用，简化 JavaScript 对 HTML DOM 操作
- 库：相当于java的工具类，库是存放东西的，jQuery是存放js代码的地方， 放的用js代码写的function
- 官网首页介绍：jQuery 是一个快速，小巧，功能丰富的 JavaScript 库。它通过易于使用的 API 在大 量浏览器中运行，使得 HTML 文档遍历和操作，事件处理，动画和 Ajax 变得更加简单。通 过多功能性和可扩展性的结合，jQuery 改变了数百万人编写 JavaScript 的方式。

## 3、JQuery优点

- 写少代码，做多事情【write less do more】
- 免费，开源且轻量级的 js 库，容量很小
- 兼容市面上主流浏览器，例如 IE，Firefox，Chrome
- 能够处理 HTML、JSP、XML、CSS、DOM、事件、实现动画效果，也能提供异步 AJAX 功能
- 文档手册很全，很详细；<https://jquery.cuishifeng.cn/>
- 成熟的插件可供选择，多种 js 组件，例如日历组件（点击按钮显示下来日期）
- 出错后，有一定的提示信息
- 不用再在 html 里面通过标签插入一大堆js来调用命令了

JavaScript	JQuery	描述
document.getElementById()	\$("#id名")	通过 ID 属性
getElementsByClassName()	\$(".class名")	通过 class 属性
document.getElementsByTagName()	\$("标签名")	通过标签名

## 4、JQuery下载

官网下载地址：<https://jquery.com/download/>

jQuery

For help when upgrading jQuery, please see the [upgrade guide](#) most relevant to your version. We also recommend using the [jQuery Migrate plugin](#).

[Download the compressed, production jQuery 3.6.0](#)    压缩版本，线上使用

[Download the uncompressed, development jQuery 3.6.0](#)    未压缩版本，开发阶段使用

[Download the map file for jQuery 3.6.0](#)

5、DOM对象、JS对象和jQuery对象

- DOM对象：  
文档对象模型（Document Object Model，简称 DOM），是 W3C 组织推荐的处理可扩展标志语言的标准编程接口。通过 DOM 对 HTML 页面的解析，可以将页面元素解析为元素节点、属性节点和文本节点，这些解析出的节点对象，即 DOM 对象。DOM 对象可以使用 JavaScript 中的方法。

```
1  //使用JavaScript的语法创建的对象叫做dom对象，也就是js对象
2  var obj = document.getElementById("txt1");
```

- JavaScript对象：  
用 JavaScript 语法创建的对象叫做 JavaScript 对象, JavaScript 对象只能调用 JavaScript 对象的 API。
- JQuery对象：  
用 JQuery 语法创建的对象叫做 JQuery 对象, JQuery 对象只能调用 jquery 对象的 API。jquery 对象是一个数组。在数组中存放本次定位的 DOM 对象。

```
1  //使用JQuery语法表示的对象叫做JQuery对象，注意：jquery表示的对象都是数组
2  var jobj = $("#txt1");
3  //jobj就是使用jquery语法表示的对象，也就是JQuery对象，它是一个数组，现在数组中就是一个值
```

- JQuery 对象与 JavaScript 对象是可以互相转化的，一般地，由于 JQuery 用起来更加方便，我们都是将 JavaScript 对象转化成 JQuery 对象

```
1  //dom --> jquery
2  $(dom对象)
3  //jquery --> dom
4  从数组中获取第一个对象，第一个对象就是dom值，使用[0]或者get(0)
5
6  为什么要进行dom和jquery的转换：目的是要使用对象的方法，或者方法
7  当你的dom对象时，可以使用dom对象的属性或者方法，如果你要想使用jquery提供的函数，必须是jquery对象才可以。
```

6、JQuery学习

6.1 JQuery第一个例子s

```
1  //使用 jquery，首先要将 jquery 库引入。使用如下语句：
2  <script type="text/javascript" src="js/jquery-3.6.0.js"></script>
```

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>JQuery第一个例子</title>
6      <!--指定jquery的库文件位置，使用相对路径，当前项目的js目录下的指定文件-->
7      <script type="text/javascript" src="js/jquery-3.6.0.js"></script>
8      <script type="text/javascript">
9        /*
10           1.$(document), $是jQuery中的函数名称，document是函数的参数
11           作用是document对象变成jQuery函数库可以使用的对象。
12           2.ready: 是jQuery中的函数，是准备的意思，当页面的dom对象加载成功后
13           会执行ready函数的内容。ready 相当于js中的onLoad事件
14           3.function()自定义的表示onLoad后要执行的功能。
15         */
16         // $(document).ready(function(){
17         //   alert("Hello JQuery");
18         // })
19
20         //简化
21         $(function(){
22           alert("Hello JQuery");
23         })
24         /*
25           $(document).ready()与$()、jQuery()、window.jQuery()是等价的，
26           所以$(document).ready()可以写成 $(function() { alert("Hello jquery") } )
27         */
28       </script>
```

```
29     </head>
30     <body>
31     </body>
32 </html>
```

6.2 DOM对象转为JQuery对象

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>dom对象转为JQuery</title>
6          <script type="text/javascript" src="./js/jquery-3.6.0.js"></script>
7          <script>
8              function btnClick(){
9                  //获取dom对象
10                 var obj = document.getElementById("btn");
11                 //使用dom的value属性,获取值
12                 alert("使用dom对象的属性="+obj.value)
13
14                 //把dom对象转jquery,使用jquery库中的函数
15                 var jobj = $(obj);
16                 //调用jquery中的函数,获取value的值
17                 alert( jobj.val() )
18             }
19         </script>
20     </head>
21     <body>
22         <input type="button" id="btn" value="我是按钮" onclick="btnClick()" />
23     </body>
24 </html>
```

6.3 JQuery对象转为DOM对象

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title></title>
6          <script type="text/javascript" src="js/jquery-3.6.0.js"></script>
7          <script>
8              function btnClick(){
9                  //使用JQuery语法获取页面中的dom对象
10                 var obj = $("#txt")[0];
11                 var num = obj.value;
12                 obj.value = num * num;
13             }
14         </script>
15     </head>
16     <body>
17         <input type="button" value="计算平方" onclick="btnClick()" /><br />
18         <input type="text" id="txt" value="整数" />
19     </body>
20 </html>
```

6.4 基本选择器

选择器: 就是定位条件; 通知 jquery 函数定位满足条件的 DOM 对象, 从而可以通过jquery的函数操作dom

- 1. id 选择器: 语法: \$("#dom对象的id名")  
通过dom对象的id定位dom对象的。通过id找对象, id在当前页面中是唯一值。
- 2. class选择器: 语法: \$(".class样式名")  
class表示css中的样式, 使用样式的名称定位dom对象的。
- 3. 标签选择器: 语法: \$("标签名称")  
使用标签名称定为dom对象的
- 4. 所有选择器: 语法: \$("\*") 选取页面中所有 DOM 对象。
- 5. 组合选择器: 组合选择器是多个被选对象间使用逗号分隔后形成的选择器, 可以组合 id, class, 标签名等。  
语法: \$("id选择器,class选择器,标签选择器")

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title></title>
6          <style type="text/css">
7              div{
8                  background: gray;
9                  width: 200px;
```

```
10         height: 100px;
11     }
12
13     .two{
14         background: gold;
15         font-size: 20pt;
16     }
17 </style>
18 <script type="text/javascript" src="js/jquery-3.6.0.js"></script>
19 <script>
20     function fun1(){
21         //id选择器
22         var obj = $("#one");
23         //使用jquery中改变样式的函数
24         obj.css("background", "red");
25     }
26
27     function fun2(){
28         //使用样式(class)选择器
29         var obj = $(".two");
30         obj.css("background","yellow");
31     }
32
33     function fun3(){
34         //标签选择器
35         var obj = $("div"); //数组有3个对象
36         //jquery的操作都是操作数组中的全部成员。
37         //所以是给所有的div都设置的背景色
38         obj.css("background","blue");
39     }
40
41     function fun4(){
42         var obj = $("*");
43         obj.css("background","green");
44     }
45
46     function fun5(){
47         var obj = $("#one,span");
48         //obj.css("background","red");
49
50         //obj是一个数组，有两个成员，1 是span dom对象
51         //$( obj[1] ) : jquery对象
52         // $( dom 对象) : 是把dom对象转为jquery对象，之后就可以调用jquery的css函数了
53         $(obj[1]).css("background", "green");//就是span
54     }
55
56 </script>
57 </head>
58 <body>
59     <div id="one">我是one的div</div><br/>
60     <div class="two">我是样式是two的div</div><br/>
61     <div>我是没有id, class的div</div><br/>
62     <span class="two">我是span标签</span><br/>
63     <input type="button" value="获取id是one的dom对象" onclick="fun1()" /><br/>
64     <input type="button" value="使用class样式获取dom对象" onclick="fun2()" /><br/>
65     <input type="button" value="使用标签选择器" onclick="fun3()" /> <br/>
66     <input type="button" value="所有选择器" onclick="fun4()" /><br/>
67     <input type="button" value="组合选择器" onclick="fun5()" />
68 </body>
69 </html>
```

6.5 表单选择器

表单相关元素选择器是指文本框、单选框、复选框、下拉列表等元素的选择方式。该方法无论是否存在表单，均可做出相应选择。表单选择器是为了能更加容易地操作表单， 表单选择器是根据元素类型来定义的。

- 
- 
- ☐
- ☐
- 
- 未选择任何文件
- 
- 

语法格式：\$(" :type 属性值")

- \$(" :text") 选取所有的单行文本框
- \$(" :password") 选取所有的密码框
- \$(" :radio") 选取所有的单选框
- \$(" :checkbox") 选取所有的多选框

- \$(":file") 选取所有的上传按钮

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>表单选择器</title>
6          <style type="text/css">
7
8      </style>
9      <script type="text/javascript" src="js/jquery-3.6.0.js"></script>
10     <script type="text/javascript">
11         function fun1(){
12             //使用表单选择器 $(":type的值")
13             var obj = $(":text");
14             //获取value属性的值 val()是jquery中的函数，读取value属性值
15             alert( obj.val());
16         }
17
18         function fun2() {
19             //定位radio
20             var obj = $(":radio");//数组,目前是二个对象 man ,woman
21             //循环数组,数组中的成员是dom对象，可以dom的属性或者函数
22             for(var i=0;i<obj.length;i++){
23                 //从数组值获取成员,使用下标的方式
24                 var dom = obj[i];
25                 //使用dom对象的属性,获取value值
26                 alert(dom.value)
27             }
28         }
29
30         function fun3(){
31             //定位checkbox
32             var obj = $(":checkbox"); //数组,有三个对象
33             for(var i=0;i<obj.length;i++){
34                 var dom = obj[i];
35                 //alert(dom.value);
36                 //使用jqeueyr的val函数，获取value的值
37                 //1. 需要jquery对象
38                 var jObj = $(dom); // jObj 是jquery对象
39                 //2. 调用jquery函数
40                 alert("jquery的函数调用=" + jObj.val());
41             }
42         }
43     </script>
44 </head>
45 <body>
46     <input type="text" value="我是type=text" /><br/>
47     <br/>
48     <input type="radio" value="man" /> 男 <br/>
49     <input type="radio" value="woman" /> 女 <br/>
50     <br/>
51     <input type="checkbox" value="bike" /> 骑行 <br/>
52     <input type="checkbox" value="football" /> 足球 <br/>
53     <input type="checkbox" value="music" /> 音乐 <br/>
54     <br/>
55     <input type="button" value="读取text的值" onclick="fun1()"/>
56     <br/>
57     <input type="button" value="读取radio的值" onclick="fun2()"/>
58     <br/>
59     <input type="button" value="读取checkbox的值" onclick="fun3()"/>
60 </body>
61 </html>
```

6.6 基本过滤器

过滤器：在定位了dom对象后，根据一些条件筛选dom对象。

过滤器又是一个字符串，用来筛选dom对象的。

过滤器不能单独使用，必须和选择器一起使用。

```
1  <div>1</div> dom1
2  <div>2</div> dom2
3  <div>3</div> dom3
4  $("div") == [dom1,dom2,dom3]
```

基本过滤器：

- \$("选择器:first")：第一个dom对象
- \$("选择器:last")：数组中的最后一个dom对象
- \$("选择器:eq(数组的下标)")：获取指定下标的dom对象
- \$("选择器:lt(下标)")：获取小于下标的所有dom对象

- \$("选择器:gt(下标)"): 获取大于下标的所有dom对象

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <style type="text/css">
6        div{
7          background: gray;
8        }
9      </style>
10     <script type="text/javascript" src="js/jquery-3.6.0.js"></script>
11     <script type="text/javascript">
12       // $(document).ready( 函数 ): 当页面中的dom对象加载成功后,会执行ready(),
13       // 相当于是onLoad().
14       $(function() {
15         //当页面dom对象加载后,给对象绑定事件,因为此时button对象已经在内存中创建好了.才能使用.
16         $("#btn1").click(function(){
17           //过滤器
18           var obj = $("div:first");
19           obj.css("background","red");
20         })
21
22         //绑定事件
23         $("#btn2").click(function(){
24           var obj = $("div:last");
25           obj.css("background","green");
26         })
27
28         //绑定btn3的事件
29         $("#btn3").click(function(){
30           var obj = $("div:eq(3)");
31           obj.css("background","blue");
32         })
33
34         $("#btn4").click(function(){
35           var obj = $("div:lt(3)");
36           obj.css("background","orange");
37         })
38
39         $("#btn5").click(function(){
40           var obj = $("div:gt(3)");
41           obj.css("background","yellow");
42         })
43
44         $("#txt").keydown(function(){
45           alert("keydown")
46         })
47       })
48     </script>
49   </head>
50   <body>
51     <input type="text" id="txt" />
52     <div id="one">我是div-0</div>
53     <div id="two">我是div-1</div>
54     <div>我是div-2
55       <div>我是div-3</div>
56       <div>我是div-4</div>
57     </div>
58     <div>我是div-5</div>
59     <br />
60     <span>我是span</span>
61
62     <br/>
63     <input type="button" value="获取第一个div" id="btn1"/>
64     <br/>
65     <input type="button" value="获取最后一个div" id="btn2"/>
66     <br/>
67     <input type="button" value="获取下标等于3的div" id="btn3"/>
68     <br/>
69     <input type="button" value="获取下标小于3的div" id="btn4"/>
70     <br/>
71     <input type="button" value="获取下标大于3的div" id="btn5"/>
72   </body>
73 </html>
```

6.7 表单属性过滤器

根据表单中dom对象的状态情况，定位dom对象的。

- \$(".:text:enabled") 选择可用的文本
- \$(".:text:disabled") 选择不可用的文本
- \$(".:checkbox:checked") 复选框选中的元素



- 选择器>option:selected 选择指定下拉列表的被选中元素

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <script type="text/javascript" src="js/jquery-3.6.0.js"></script>
6      <script type="text/javascript">
7        // $(document).ready( 函数 ): 当页面中的dom对象加载成功后,会执行ready(),
8        // 相当于是onLoad().
9        $(function() {
10          //当页面dom对象加载后,给对象绑定事件,因为此时button对象已经在内存中创建好了.才能使用.
11          $("#btn1").click(function(){
12            //获取所有可以使用的text
13            var obj  = $(".text:enabled");
14            //设置 jquery数组值所有dom对象的value值
15            obj.val("hello");
16          })
17
18          $("#btn2").click(function(){
19            //获取选中的checkbox
20            var obj  = $(".checkbox:checked");
21            for(var i=0;i<obj.length;i++){
22              //alert( obj[i].value);
23              alert(    $(obj[i]).val() )
24            }
25          })
26
27          $("#btn3").click(function(){
28            //获取select选中的值
29            //var obj= $(".select>option:selected");
30            var obj = $(".#yuyan>option:selected");
31            alert(obj.val());
32          })
33        })
34      </script>
35    </head>
36    <body>
37      <input type="text" id="txt1" value="text1" /><br/>
38      <input type="text" id="txt2" value="text2" disabled="true"/><br/>
39      <input type="text" id="txt3" value="text3" /><br/>
40      <input type="text" id="txt4" value="text4" disabled/><br/>
41      <br/>
42      <input type="checkbox" value="游泳" />游泳 <br/>
43      <input type="checkbox" value="健身" checked />健身 <br/>
44      <input type="checkbox" value="电子游戏" checked />电子游戏 <br/>
45      <br/>
46      <select id="yuyan">
47        <option value="java">java语言</option>
48        <option value="go" selected>go语言</option>
49        <option value="python">python语言</option>
50      </select>
51
52      <br/><br/>
53      <input type="button" value="设置可以的text的value是hello" id="btn1"/>
54      <br/>
55      <button id="btn2">显示选中的复选框的值</button>
56      <br/>
57      <button id="btn3">显示选中下拉列表框的值</button>
58    </body>
59  </html>
```

6.8 函数

1. val

操作数组中 DOM 对象的 value 属性;

\$(选择器).val(): 无参数调用形式, 读取数组中第一个 DOM 对象的 value 属性值

\$(选择器).val(值): 有参形式调用; 对数组中所有 DOM 对象的 value 属性值进行统一赋值

2. text

操作数组中所有 DOM 对象的【文字显示内容属性】

\$(选择器).text():无参数调用, 读取数组中所有 DOM 对象的文字显示内容, 将得到内容拼接为一个字符串返回

\$(选择器).text(值):有参数方式, 对数组中所有 DOM 对象的文字显示内容进行统一赋值

3. attr

对 val, text 之外的其他属性操作

\$(选择器).attr("属性名"): 获取 DOM 数组第一个对象的属性值

\$(选择器).attr("属性名","值"): 对数组中所有 DOM 对象的属性设为新值

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <style type="text/css">
6       div{
7         background: yellow;
8       }
9     </style>
10    <script type="text/javascript" src="js/jquery-3.6.0.js"></script>
11    <script type="text/javascript">
12      //在dom对象创建好后,绑定事件
13      $(function(){
14        $("#btn1").click(function(){
15          //val() 获取dom数组中第一个对象的value属性值
16          var text = $(":text").val();
17          alert(text) // 刘备
18        })
19
20        $("#btn2").click(function(){
21          //设置所有的text的value为新值
22          $(":text").val("三国演义");
23        })
24
25        $("#btn3").click(function(){
26          //获取div ,text()无参数,获取dom对象的文本值,连接成一个字符串
27          alert($("#div").text());
28          // 1.我第一个div2.我第二个div3.我第三个div返回顶部
29        })
30
31        $("#btn4").click(function(){
32          //设置div的文本值
33          $("#div").text("新的div文本内容");
34        })
35
36        $("#btn5").click(function(){
37          //读取指定属性的值
38          alert($("#img").attr("src")); // img/ex1.jpg
39        })
40
41        $("#btn6").click(function(){
42          //设置指定属性的,指定值
43          $("#img").attr("src","img/ex2.jpg");
44          //val(), text();
45        })
46      })
47    </script>
48  </head>
49  <body>
50    <input type="text" value="刘备" /><br/>
51    <input type="text" value="关羽" /><br/>
52    <input type="text" value="张飞" /><br/>
53    <br/>
54    <div>1.我第一个div</div>
55    <div>2.我第二个div</div>
56    <div>3.我第三个div</div>
57    <br/>
58    
59    <br/>
60
61    <input type="button" value="获取第一文本框的值" id="btn1"/>
62    <br/>
63    <br/>
64    <input type="button" value="设置所有文本框的value值" id="btn2"/>
65    <br/>
66    <br/>
67    <input type="button" value="获取所有div的文本值" id="btn3"/>
68    <br/>
69    <br/>
70    <input type="button" value="设置div的文本值" id="btn4"/>
71    <br/>
72    <br/>
73    <input type="button" value="读取src属性的值" id="btn5"/>
74    <br/>
75    <br/>
76    <input type="button" value="设置指定的属性值" id="btn6"/>
77  </body>
78 </html>
```

#### 4. hide

\$(选择器).hide(): 将数组中所有 DOM 对象隐藏起来

#### 5. show



\$(选择器).show():将数组中所有 DOM 对象在浏览器中显示起来

6. remove

\$(选择器).remove() : 将数组中所有 DOM 对象及其子对象一并删除

7. empty

\$(选择器).empty(): 将数组中所有 DOM 对象的子对象删除

8. append

为数组中所有 DOM 对象添加子对象

```
$(选择器).append("
我动态添加的 div
")
```

9. html

设置或返回被选元素的内容 (innerHTML) 。

\$(选择器).html(): 无参数调用方法, 获取 DOM 数组第一个匹元素的内容。

\$(选择器).html(值): 有参数调用, 用于设置 DOM 数组中所有元素的内容。

10. each

each 是对数组、json 和 dom 数组等的遍历,对每个元素调用一次函数。

语法 1: \$.each( 要遍历的对象, function(index,element) { 处理程序 } )

语法 2: jQuery 对象.each( function( index, element ) { 处理程序 } )

index: 数组的下标

element: 数组的对象

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <style type="text/css">
6              div{
7                  background: yellow;
8              }
9          </style>
10         <script type="text/javascript" src="js/jquery-3.6.0.js"></script>
11         <script type="text/javascript">
12             //在dom对象创建好后,绑定事件
13             $(function(){
14                 $("#btn1").click(function(){
15                     //使用remove:删除父和子所有的dom对象
16                     $("select").remove();
17                 })
18
19                 $("#btn2").click(function(){
20                     //使用empty 删除子dom对象
21                     $("select").empty();
22                 })
23
24                 $("#btn3").click(function(){
25                     //使用append,增加dom对象
26                     // $("#fatcher").append("<input type='button' value='我是增加的按钮' />")
27                     //增加一个table
28                     $("#fatcher").append("<table border=1><tr><td>第一列</td><td>第二列</td></tr>
29
30
31                 $("#btn4").click(function(){
32                     //使用html()函数,获取数组中第一个dom对象的文本值(innerHTML)
33                     alert($("#span").text()); // 我是mysql 数据库我是jdbc
34                     alert( $("#span").html() ); //我是mysql <b>数据库</b>
35                 })
36
37                 $("#btn5").click(function(){
38                     //使用 html(有参数):设置dom对象的文本值
39                     $("#span").html("我是新的<b>数据</b>");
40                 })
41
42                 $("#btn6").click(function(){
43                     //循环普通数组,非dom数组
44                     var arr = [ 11, 12, 13];
45                     $.each(arr, function(i,n){
46                         alert("循环变量: "+i + "====数组成员:"+ n);
47                         //循环变量: 0====数组成员:11
48                         //循环变量: 1====数组成员:12
49                         //循环变量: 2====数组成员:13
50                     })
51                 })
52             })
```

```
53     $("#btn7").click(function(){
54         //循环json
55         var json={"name":"张三","age":20};
56         //var obj = eval("{'name':'张三','age':20}");
57         $.each(json,function(i,n){
58             alert("i是key="+i+",n是值="+n);
59             //i是key=name,n是值=张三
60             //i是key=age,n是值=20
61         })
62     })
63
64     $("#btn8").click(function(){
65         //循环dom数组
66         var domArray = $(".:text");//dom数组
67         $.each(domArray, function(i,n){
68             // n 是数组中的dom对象
69             alert("i="+i+" , n="+n.value);
70             //i=0 , n=刘备
71             //i=1 , n=关羽
72             //i=2 , n=张飞
73         })
74     })
75
76     $("#btn9").click(function(){
77         //循环jquery对象，jquery对象就是dom数组
78         $(".:text").each(function(i,n){
79             alert("i="+i+", n="+ n.value);
80             //i=0 , n=刘备
81             //i=1 , n=关羽
82             //i=2 , n=张飞
83         })
84     })
85     })
86     </script>
87 </head>
88 <body>
89     <input type="text" value="刘备" />
90     <input type="text" value="关羽" />
91     <input type="text" value="张飞" />
92
93     <br/>
94     <select>
95         <option value="老虎">老虎</option>
96         <option value="狮子">狮子</option>
97         <option value="豹子">豹子</option>
98     </select>
99     <br/>
100    <br/>
101    <select>
102        <option value="亚洲">亚洲</option>
103        <option value="欧洲">欧洲</option>
104        <option value="美洲">美洲</option>
105    </select>
106    <br/>
107    <br/>
108    <div id="fatcher">我是第一个div</div>
109    <br/>
110    <br/>
111    <span>我是mysql <b>数据库</b></span>
112    <br/>
113    <span>我是jdbc</span>
114    <br/>
115    <br/>
116
117    <input type="button" value="使用remove删除父和子对象" id="btn1"/>
118    <br/>
119    <br/>
120    <input type="button" value="使用empty删子对象" id="btn2"/>
121    <br/>
122    <br/>
123    <input type="button" value="使用append,增加dom对象" id="btn3"/>
124    <br/>
125    <br/>
126    <input type="button" value="获取第一个dom的文本值" id="btn4"/>
127    <br/>
128    <br/>
129    <input type="button" value="设置span的所以dom的文本值" id="btn5"/>
130    <br/>
131    <br/>
132    <input type="button" value="循环普通数组" id="btn6"/>
133    <br/>
134    <br/>
135    <input type="button" value="循环json" id="btn7"/>
```

```
136         <br/>
137         <br/>
138         <input type="button" value="循环dom数组" id="btn8"/>
139         <br/>
140         <br/>
141         <input type="button" value="循环jquery对象" id="btn9"/>
142     </body>
143 </html>
```

6.9 JQuery绑定事件方式

1. 定义元素监听事件

\$(选择器).事件名称( 事件的处理函数)  
\$(选择器): 定位dom对象， dom对象可以有多个， 这些dom对象都绑定事件了  
事件名称: 就是js中事件去掉on的部分， 例如 js中的单击事件 onclick(),  
jquery中的事件名称， 就是click， 都是小写的。  
事件的处理函数: 就是一个function ， 当事件发生时， 执行这个函数的内容。

```
1  例如给id是"btn"的按钮绑定单击事件
2  $("#btn").click(function(){
3      alert("btn按钮单击了")
4  })
```

2. on() 绑定事件

on() 方法在被选元素上添加事件处理程序。该方法给 API 带来很多便利， 推荐使用该方法。

语法: \$(选择器).on(event, data, function)  
event: 事件一个或者多个， 多个之间空格分开  
data: 可选。规定传递到函数的额外数据， json 格式  
function: 可选。规定当事件发生时运行的函数。

```
1  <input type="button" id="btn">
2  $("#btn").on("click", function() {
3      //处理按钮单击
4  })
```

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <style type="text/css">
6              div{
7                  background: yellow;
8              }
9          </style>
10         <script type="text/javascript" src="js/jquery-3.6.0.js"></script>
11         <script type="text/javascript">
12             //在dom对象创建好后,绑定事件
13             $(function(){
14                 $("#btn1").click(function(){
15                     //使用append增加dom对象
16                     $("#mydiv").append("<input id='newBtn' type='button' value='我是新加入的按钮' />");
17                     //使用on给按钮绑定事件
18                     $("#newBtn").on("click",function(){
19                         alert("新建的按钮被单击了");
20                     })
21                 })
22             })
23         </script>
24     </head>
25     <body>
26         <div id="mydiv">
27             我是一个div ， 需要增加一个button
28         </div>
29         <input type="button" value="创建一个button,绑定一个click" id="btn1"/>
30     <br/>
31 </body>
32 </html>
```

6.10 AJAX语法

- 没有jquery之前， 使用XMLHttpRequest做ajax， 有4个步骤。 jquery简化了ajax请求的处理。
- 使用三个函数可以实现ajax请求的处理。
  - \$.ajax(): jquery中实现ajax的核心函数。
  - \$.post(): 使用post方式做ajax请求。
  - \$.get(): 使用get方式发送ajax请求。
  - \$.post() 和 \$.get() 他们在内部都是调用的 \$.ajax()

1. \$.ajax()函数的使用

\$.ajax() 是jQuery 中 AJAX 请求的核心方法，所有的其他方法都是在内部使用此方法。

语法：\$.ajax( { name:value, name:value, ... } )

说明：参数是 json 的数据，包含请求方式，数据，回调方法等

参数：

- async：布尔值，表示请求是否异步处理。默认是 true（异步）
- contentType：发送数据到服务器时所使用的内容类型。默认是："application/x-www-form-urlencoded"  
例如你想表示请求的参数是json格式的，可以写："application/json"
- data：规定要发送到服务器的数据，可以是：string，数组，多数是 json
- dataType：**期望**从服务器端响应的数据类型。jQuery 从 xml, json, text,, html 这些中测试最可能的类型
  - "xml": 一个 XML 文档
  - "html": HTML 作为纯文本
  - "text": 纯文本字符串
  - "json": 以 JSON 运行响应，并以对象返回
- error(xhr,status,error)：如果请求失败要运行的函数，其中 xhr, status, error 是自定义的形参名

```
1 | error:function() {    发生错误时执行    }
```

- success(result,status,xhr)：当请求成功时运行的函数，其中 result, status, xhr 是自定义的形参名

```
1 | success:一个function，请求成功了，从服务器端返回了数据，会执行success指定函数
2 |    之前使用XMLHttpRequest对象， 当readyState==4 && status==200的时候。
```

- type：规定请求的类型（GET 或 POST 等），默认是 GET；get, post 不用区分大小写
- url：规定发送请求的 URL

**注意：error() , success()中的 xhr 是 XMLHttpRequest 对象**

```
1 | $.ajax({
2 |     async:true ,
3 |     contentType:"application/json" ,
4 |     data: {name:"lisi",age:20 },
5 |     dataType:"json",
6 |     error:function(){
7 |         请求出现错误时，执行的函数
8 |     },
9 |     success:function( data ) {
10 |         //data就是responseText， 是jquery处理后的数据。
11 |     },
12 |     url:"bmiAjax",
13 |     type:"get"
14 | })
```

```
1 | <!--
2 |   Created by IntelliJ IDEA.
3 |   User: Amadeus
4 |   Date: 2022/1/10
5 |   Time: 14:30
6 |   To change this template use File | Settings | File Templates.
7 | -->
8 | <%@ page contentType="text/html; charset=UTF-8" language="java" %>
9 | <html>
10 | <head>
11 |     <title>使用JSON格式的数据</title>
12 |     <script type="text/javascript" src="/js/jquery-3.6.0.js"></script>
13 |     <script type="text/javascript">
14 |         $(function () {
15 |             $("#btn").click(function () {
16 |                 //获取dom的value值
17 |                 var proid = $("#proid").val();
18 |                 //发起ajax请求
19 |                 $.ajax({
20 |                     url:"queryJson",
21 |                     data:{"proid": proid},
22 |                     dataType:"json",
23 |                     success:function (resp){
24 |                         $("#praname").val(resp.name);
25 |                         $("#prosh").val(resp.shenghui);
26 |                         $("#projc").val(resp.jiancheng);
27 |                     }
28 |                 })
29 |             })
30 |         })
31 |     </script>
32 | </head>
33 | <body>
```

```
34     <p>ajax请求使用json格式的数据</p>
35     <table border="2">
36         <tr>
37             <td>省份编号</td>
38             <td>
39                 <input type="text" id="proid">
40                 <input type="button" value="搜索" id="btn">
41             </td>
42         </tr>
43         <tr>
44             <td>省份名称</td>
45             <td><input type="text" id="proname"></td>
46         </tr>
47         <tr>
48             <td>省份简称</td>
49             <td><input type="text" id="projc"></td>
50         </tr>
51         <tr>
52             <td>省会名称</td>
53             <td><input type="text" id="prosh"></td>
54         </tr>
55     </table>
56 </body>
57 </html>
```

2. \$.get()

\$.get() 方法使用 HTTP GET 请求从服务器加载数据。

语法：\$.get(url,data,function(data,status,xhr),dataType)

- url：必需。规定您需要请求的 URL。
- data：可选。规定连同请求发送到服务器的数据。
- function(data,status,xhr) 可选。当请求成功时运行的函数。data,status,xhr 是自定义形参名。
  - data：包含来自请求的结果数据
  - status：包含请求的状态 ("success"、"notmodified"、"error"、"timeout"、"parsererror")
  - xhr：包含 XMLHttpRequest 对象
- dataType：可选。规定预期的服务器响应的数据类型。默认地，jQuery 会智能判断。

可能的类型：

- "xml"：一个 XML 文档
- "html"：HTML 作为纯文本
- "text"：纯文本字符串
- "json" - 以 JSON 运行响应，并以对象返回

3. \$.post()

\$.post() 方法使用 HTTP POST 请求从服务器加载数据。

语法：\$.post(URL,data,function(data,status,xhr),dataType)

参数同\$.get()

7、级联查询功能

7.1 准备

数据库：

- province:

province @springdb (XK_MySQL) - 表			
文件 编辑 查看 窗口 帮助			
导入向导 导出向导 筛选向导 网格查看 表单查看 备注 十六进制 图像 Az 升序排序 Za 降序排序 移除排序			
id	name	jiancheng	shenghui
1	河北	冀	石家庄
2	山西	晋	太原市
3	内蒙古	蒙	呼和浩特市
4	辽宁	辽	沈阳
5	江苏	苏	南京
6	浙江	浙	杭州
7	安徽	皖	合肥
8	福建	闽	福州
9	江西	赣	南昌

- city:

city @springdb (XK\_MySQL) - 表

文件 编辑 查看 窗口 帮助

导入向导 导出向导 筛选向导 网格查看 表单查看 备注 十六进制 图像 A-Z 升序排序 Z-A 降序排序 移除排序

id	name	provinceid
1	石家庄市	1
2	秦皇岛	1
3	保定市	1
4	张家口	1
5	南昌市	9
6	九江市	9
7	宜春市	9
8	福州市	8
9	厦门市	8
10	泉州市	8
11	龙岩市	8
12	太原	2
13	大同	2
14	呼和浩特市	3
15	包头	3
16	呼伦贝尔	3

SELECT \* FROM `city` LIMIT 0, 100 第 12 条记录 (共 16 条) 于 1 页

JAVA实体类：

```

1  public class Province {
2      private Integer id;
3      private String name;
4      private String jiancheng;
5      private String shenghui;
6
7      public Integer getId() {
8          return id;
9      }
10     public void setId(Integer id) {
11         this.id = id;
12     }
13     public String getName() {
14         return name;
15     }
16     public void setName(String name) {
17         this.name = name;
18     }
19     public String getJiancheng() {
20         return jiancheng;
21     }
22     public void setJiancheng(String jiancheng) {
23         this.jiancheng = jiancheng;
24     }
25     public String getShenghui() {
26         return shenghui;
27     }
28     public void setShenghui(String shenghui) {
29         this.shenghui = shenghui;
30     }
31     @Override
32     public String toString() {
33         return "Province{" +
34             "id=" + id +
35             ", name='" + name + '\'' +
36             ", jiancheng='" + jiancheng + '\'' +
37             ", shenghui='" + shenghui + '\'' +
38             '}';
39     }
40 }

```

```

1  public class City {
2      private Integer id;
3      private String name;
4      private Integer provinceId;
5
6      public Integer getId() {
7          return id;
8      }
9      public void setId(Integer id) {
10         this.id = id;
11     }
12     public String getName() {
13         return name;
14     }
15     public void setName(String name) {

```



```

16         this.name = name;
17     }
18     public Integer getProvinceId() {
19         return provinceId;
20     }
21     public void setProvinceId(Integer provinceId) {
22         this.provinceId = provinceId;
23     }
24     @Override
25     public String toString() {
26         return "City{" +
27             "id=" + id +
28             ", name='" + name + '\'' +
29             ", provinceId=" + provinceId +
30             '}';
31     }
32 }

```

## 7.2 查询所有的省份信息

```

1  public class QueryProvinceServlet extends HttpServlet {
2      @Override
3      protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
4          String json = "{}";
5          //调用Dao, 获取所有的省份信息, 是一个List集合
6          List<Province> provinceList = QueryDao.queryProvinceList();
7          //把List转为一个JSON格式的数据, 输出给ajax请求
8          if(provinceList != null && provinceList.size() != 0){
9              //调用jackson工具库, 实现list --> json
10             ObjectMapper om = new ObjectMapper();
11             json = om.writeValueAsString(provinceList);
12         }
13         //输出json数据, 响应ajax请求的, 返回数据
14         response.setContentType("application/json;charset=utf-8");
15         PrintWriter pw = response.getWriter();
16         pw.println(json);
17         pw.flush();
18         pw.close();
19     }
20 }

```

```

1  public class QueryDao {
2      //查询所有的省份信息
3      public static List<Province> queryProvinceList(){
4          Connection conn = null;
5          PreparedStatement ps = null;
6          ResultSet rs = null;
7          Province province = null;
8          List<Province> provinceList = new ArrayList<>();
9          try {
10             String sql = "select id, name, jiancheng, shenghui from province order by id";
11             conn = JdbcUtil.getConnection();
12             ps = conn.prepareStatement(sql);
13             rs = ps.executeQuery();
14             while(rs.next()){
15                 province = new Province();
16                 province.setId(rs.getInt("id"));
17                 province.setName(rs.getString("name"));
18                 province.setJiancheng(rs.getString("jiancheng"));
19                 province.setShenghui(rs.getString("shenghui"));
20                 provinceList.add(province);
21             }
22         } catch (SQLException e) {
23             e.printStackTrace();
24         } finally {
25             JdbcUtil.close(conn, ps, rs);
26         }
27         return provinceList;
28     }
29 }

```

## 7.3 查询一个省份下面所有的城市

```

1  public class QueryCityServlet extends HttpServlet {
2      @Override
3      protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
4          //获取请求传过来的省份ID
5          String proid = request.getParameter("proid");
6          String json = "{}";
7          if(proid != null && !"".equals(proid)){

```

```

8         List<City> cityList = QueryDao.queryCityList(Integer.valueOf(proid));
9         //list --> json
10        ObjectMapper om = new ObjectMapper();
11        json = om.writeValueAsString(cityList);
12    }
13    //输出json数据，响应ajax请求的，返回数据
14    response.setContentType("application/json;charset=utf-8");
15    PrintWriter pw = response.getWriter();
16    pw.println(json);
17    pw.flush();
18    pw.close();
19 }
20 }

```

```

1 public class QueryDao {
2     //查询一个省份下面所有的城市
3     public static List<City> queryCityList(Integer provinceId){
4         Connection conn = null;
5         PreparedStatement ps = null;
6         ResultSet rs = null;
7         City city = null;
8         List<City> cityList = new ArrayList<>();
9         try {
10            String sql = "select id, name from city where provinceId = ?";
11            conn = JdbcUtil.getConnection();
12            ps = conn.prepareStatement(sql);
13            ps.setInt(1, provinceId);
14            rs = ps.executeQuery();
15            while(rs.next()){
16                city = new City();
17                city.setId(rs.getInt("id"));
18                city.setName(rs.getString("name"));
19                cityList.add(city);
20            }
21        } catch (SQLException e) {
22            e.printStackTrace();
23        } finally {
24            JdbcUtil.close(conn, ps, rs);
25        }
26        return cityList;
27    }
28 }

```

## 7.4 index.jsp页面

```

1 <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
2 <!DOCTYPE html>
3 <html>
4 <head>
5     <title>省市级联查询</title>
6     <script type="text/javascript" src="js/jquery-3.6.0.js"></script>
7     <script type="text/javascript">
8         function loadDataAjax() {
9             //做ajax请求，使用jquery的$.ajax()
10            $.ajax({
11                url:"queryProvince",
12                dataType:"json",
13                success:function( resp ){
14                    //删除旧的数据，把已经存在的数据清空
15                    $("#province").empty();
16                    //[{ "id":1,"name":"河北","jiancheng":"冀","shenghui":"石家庄"},{}]
17                    $.each( resp, function (i,n) {
18                        //获取select这个dom对象
19                        $("#province").append("<option value='"+n.id+ "'>" + n.name + "</option>");
20                    })
21                }
22            })
23        }
24
25        $(function(){
26            // $(function()) 在页面的dom的对象加载成功后执行的函数， 在此发起ajax。
27            loadDataAjax();
28            //绑定事件
29            $("#btnLoad").click(function(){
30                loadDataAjax();
31            })
32            //给省份的select绑定一个change事件，当select内容发生变化时，触发事件
33            $("#province").change(function () {
34                //获取选中的列表框的值
35                var privinceId = $("#province>option:selected").val();
36                //做一个ajax请求，获取省份的所有城市信息
37                $.post("queryCity", { proid: privinceId }, function (resp){

```

```
38         //删除旧的数据，把已经存在的数据清空
39         $("#city").empty();
40         $.each(resp, function (index, value){
41             $("#city").append("<option value="+ value.id +">" + value.name + "</option>")
42         })
43     }, "json");
44 })
45 })
46 </script>
47 </head>
48 <body>
49 <p>省市级联查询</p>
50 <div>
51     <table>
52         <tr>
53             <td>省份: </td>
54             <td>
55                 <select id="province">
56                     <option value="0">请选择.....</option>
57                 </select>
58             </td>
59             <td>
60                 <input type="button" value="load数据" id="btnLoad">
61             </td>
62         </tr>
63         <tr>
64             <td>城市: </td>
65             <td>
66                 <select id="city">
67                     <option value="0">请选择.....</option>
68                 </select>
69             </td>
70         </tr>
71     </table>
72 </div>
73 </body>
74 </html>
```