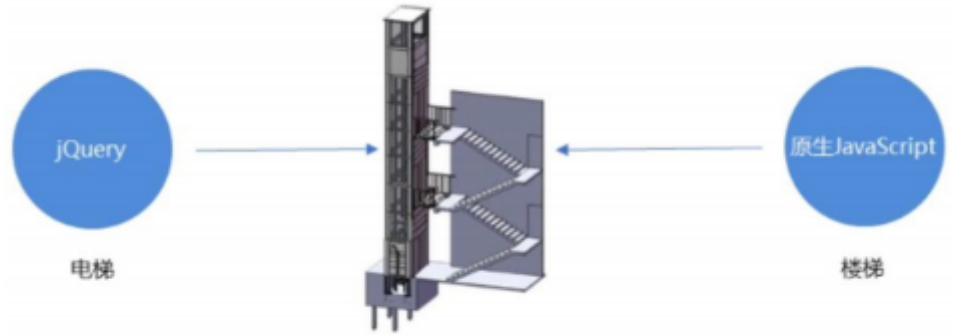
1、JQuery快速入门

1.1、JQuery介绍

- jQuery 是一个 JavaScript 库。
- 所谓的库，就是一个 JS 文件，里面封装了很多预定义的函数，比如获取元素，执行隐藏、移动等，目的就 是在使用时直接调用，不需要再重复定义，这样就可以极大地简化了 JavaScript 编程。
- jQuery 官网：https://www.jquery.com



1.2、JQuery快速入门

- **开发思路**

1. 编写 HTML 文档。
2. 引入 jQuery 文件。
3. 使用 jQuery 获取元素。
4. 使用浏览器测试。

- **代码实现**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>快速入门</title>
</head>
<body>
    <div id="div">我是div</div>
</body>
<!--引入 jQuery 文件-->
<script src="js/jquery-3.3.1.min.js"></script>
<script>
    // JS方式，通过id属性值来获取div元素
    let jsDiv = document.getElementById("div");
    //alert(jsDiv);
    //alert(jsDiv.innerHTML);

    // jQuery方式，通过id属性值来获取div元素
    let jqDiv = $("#div");
    alert(jqDiv);
```

```
    alert(jqDiv.html());
</script>
</html>
```

1.3、小结

- jQuery 是一个 JavaScript 库。
- 说白了就是定义好的一个 JS 文件，内部封装了很多功能，可以大大简化我们的 JS 操作步骤。
- jQuery 官网：https://www.jquery.com。
- 要想使用，必须要引入该文件。
- jQuery 的核心语法 $();

2、JQuery基本语法

2.1、JS对象和JQuery对象转换

- jQuery 本质上虽然也是 JS，但如果想使用 jQuery 的属性和方法那么必须保证对象是 jQuery 对象，而不是 JS 方式获得的 DOM 对象，二者的 API 方法不能混合使用，若想使用对方的 API，需要进行对象的转换。
- **JS 的 DOM 对象转换成 jQuery 对象**

```
//$(JS 的 DOM 对象);

// JS方式，通过id属性值获取div元素
let jsDiv = document.getElementById("div");
alert(jsDiv.innerHTML);
//alert(jsDiv.html());  JS对象无法使用jQuery里面的功能

// 将 JS 对象转换为jQuery对象
let jq = $(jsDiv);
alert(jq.html());
```

- **jQuery 对象转换成 JS 对象**

```
/*jQuery 对象[索引];
jQuery 对象.get(索引);*/

// jQuery方式，通过id属性值获取div元素
let jqDiv = $("#div");
alert(jqDiv.html());
// alert(jqDiv.innerHTML);   jQuery对象无法使用JS里面的功能

// 将 jQuery对象转换为JS对象
let js = jqDiv[0];
alert(js.innerHTML);
```

2.2、事件的基本使用

- **常用的事件**

事件名	说明
onload	某个页面或图像被完成加载
onsubmit	当表单提交时触发该事件
onclick	鼠标单击事件
ondblclick	鼠标双击事件
onblur	元素失去焦点
onfocus	元素获得焦点
onchange	用户改变域的内容

- **在 jQuery 中将事件封装成了对应的方法。去掉了 JS 中的 .on 语法。**
- **代码实现**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>事件的使用</title>
</head>
<body>
    <input type="button" id="btn" value="点我">
    <br>
    <input type="text" id="input">
</body>
<script src="js/jquery-3.3.1.min.js"></script>
<script>
    //单击事件
    $("#btn").click(function(){
        alert("点我干嘛?");
    });

    //获取焦点事件
    // $("#input").focus(function(){
    //     alert("你要输入数据啦...");
    // });

    //失去焦点事件
    $("#input").blur(function(){
        alert("你输入完成啦...");
    });
</script>
</html>
```

2.3、时间的绑定和解绑

- **绑定事件**

  //jQuery 对象.on(事件名称,执行的功能);

```
//给btn1按钮绑定单击事件
$("#btn1").on("click",function(){
alert("点我干嘛?");
});
```

- **解绑事件**

  如果不指定事件名称，则会把该对象绑定的所有事件都解绑

  //jQuery 对象.off(事件名称);

```
//通过btn2解绑btn1的单击事件
$("#btn2").on("click",function(){
$("#btn1").off("click");
});
```

2.4、时间的切换

事件的切换：需要给同一个对象绑定多个事件，而且多个事件还有先后顺序关系。

- **方式一：单独定义**

  $(元素).事件方法名1(要执行的功能);

  $(元素).事件方法名2(要执行的功能);

```
//方式一 单独定义
$("#div").mouseover(function(){
    //背景色：红色
    //$("#div").css("background","red");
    $(this).css("background","red");
    });
$("#div").mouseout(function(){
    //背景色：蓝色
    //$("#div").css("background","blue");
    $(this).css("background","blue");
});
```

- **方式二：链式定义**

  $(元素).事件方法名1(要执行的功能)

  .事件方法名2(要执行的功能);

```
//方式二 链式定义
$("#div").mouseover(function(){
   $(this).css("background","red");
}).mouseout(function(){
   $(this).css("background","blue");
});
```

2.5、遍历操作

- **方式一：传统方式**

```
for(let i = 0; i < 容器对象长度; i++){
        执行功能;
}
```

```
//方式一：传统方式
$("#btn").click(function(){
    let lis = $("li");
    for(let i = 0 ; i < lis.length; i++) {
        alert(i + ":" + lis[i].innerHTML);
    }
});
```

- **方式二：对象.each()方法**

```
容器对象.each(function(index,ele){
    执行功能;
});
```

```
//方式二：对象.each()方法
$("#btn").click(function(){
    let lis = $("li");
    lis.each(function(index,ele){
        alert(index + ":" + ele.innerHTML);
    });
});
```

- **方式三：$.each()方法**

```
$.each(容器对象,function(index,ele){
    执行功能;
});
```

```
//方式三：$.each()方法
$("#btn").click(function(){
    let lis = $("li");
    $.each(lis,function(index,ele){
        alert(index + ":" + ele.innerHTML);
    });
});
```

- **方式四：for of语句**

```
for(ele of 容器对象){
    执行功能;
}
```

```
//方式四：for of 语句遍历
$("#btn").click(function(){
    let lis = $("li");
    for(ele of lis){
        alert(ele.innerHTML);
    }
});
```

2.6、小结

- **JS 对象和 jQuery 对象相互转换**
  - $(JS 的 DOM 对象)：将 JS 对象转为 jQuery 对象。
  - jQuery 对象[索引] jQuery
  - 对象.get(索引)：将 jQuery 对象转为 JS 对象。
- **事件**
  - 在 jQuery 中将事件封装成了对应的方法。去掉了 JS 中的 .on 语法。
  - on(事件名称,执行的功能)：绑定事件。
  - off(事件名称)：解绑事件。
- **遍历**
  - 传统方式。
  - 对象.each() 方法。
  - $.each() 方法。
  - for of 语句。

3、JQuery选择器

3.1、基本选择器

- 选择器：类似于 CSS 的选择器，可以帮助我们获取元素。
- 例如：id 选择器、类选择器、元素选择器、属性选择器等等。
- jQuery 中选择器的语法：$();

选择器	语法	作用
元素选择器	$("元素的名称");	根据元素名称获取元素对象们
id 选择器	$("#id的属性值");	根据id属性值获取元素对象
类选择器	$(".class的属性值");	根据class属性值获取元素对象们

代码实现

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>基本选择器</title>
</head>
<body>
```

```
    <div id="div1">div1</div>
    <div class="cls">div2</div>
    <div class="cls">div3</div>
</body>
<script src="js/jquery-3.3.1.min.js"></script>
<script>
    //1.元素选择器    $("元素的名称")
    let divs = $("div");
    //alert(divs.length);

    //2.id选择器     $("#id的属性值")
    let div1 = $("#div1");
    //alert(div1);

    //3.类选择器      $(".class的属性值")
    let cls = $(".cls");
    alert(cls.length);

</script>
</html>
```

3.2、层级选择器

选择器	语法	作用
后代选择器	$("A B");	A下的所有B(包括B的子级)
子选择器	$("A > B");	A下的所有B(不包括B的子级)
兄弟选择器	$("A + B");	A相邻的下一个B
兄弟选择器	$("A ~ B");	A相邻的所有B

- **代码实现**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>层级选择器</title>
</head>
<body>
    <div>
        <span>s1
            <span>s1-1</span>
            <span>s1-2</span>
        </span>
        <span>s2</span>
    </div>

    <div></div>
    <p>p1</p>
    <p>p2</p>
</body>
<script src="js/jquery-3.3.1.min.js"></script>
```

```
<script>
    // 1. 后代选择器 $("A B");       A下的所有B(包括B的子级)
    let spans1 = $("div span");
    //alert(spans1.length);

    // 2. 子选择器   $("A > B");     A下的所有B(不包括B的子级)
    let spans2 = $("div > span");
    //alert(spans2.length);

    // 3. 兄弟选择器 $("A + B");     A相邻的下一个B
    let ps1 = $("div + p");
    //alert(ps1.length);

    // 4. 兄弟选择器 $("A ~ B");     A相邻的所有B
    let ps2 = $("div ~ p");
    alert(ps2.length);

</script>
</html>
```

3.3、属性选择器

选择器	语法	作用
属性名选择器	$("A[属性名]");	根据指定属性名获取元素对象们
属性值选择器	$("A[属性名=属性值]");	根据指定属性名和属性值获取元素对象们

- **代码实现**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>属性选择器</title>
</head>
<body>
    <input type="text">
    <input type="password">
    <input type="password">
</body>
<script src="js/jquery-3.3.1.min.js"></script>
<script>
    //1.属性名选择器   $("元素[属性名]")
    let in1 = $("input[type]");
    //alert(in1.length);


    //2.属性值选择器   $("元素[属性名=属性值]")
    let in2 = $("input[type='password']");
    alert(in2.length);

</script>
</html>
```

3.4、过滤器选择器

选择器	语法	作用
首元素选择器	$("A:first");	获得选择的元素中的第一个元素
尾元素选择器	$("A:last");	获得选择的元素中的最后一个元素
非元素选择器	$("A:not(B)");	不包括指定内容的元素
偶数选择器	$("A:even");	偶数，从 0 开始计数
奇数选择器	$("A:odd");	奇数，从 0 开始计数
等于索引选择器	$("A:eq(index)");	指定索引元素
大于索引选择器	$("A:gt(index)");	大于指定索引元素
小于索引选择器	$("A:lt(index)");	小于指定索引元素

- **代码实现**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>过滤器选择器</title>
</head>
<body>
    <div>div1</div>
    <div id="div2">div2</div>
    <div>div3</div>
    <div>div4</div>
</body>
<script src="js/jquery-3.3.1.min.js"></script>
<script>
    // 1.首元素选择器 $("A:first");
    let div1 = $("div:first");
    //alert(div1.html());

    // 2.尾元素选择器 $("A:last");
    let div4 = $("div:last");
    //alert(div4.html());

    // 3.非元素选择器 $("A:not(B)");
    let divs1 = $("div:not(#div2)");
    //alert(divs1.length);

    // 4.偶数选择器       $("A:even");
    let divs2 = $("div:even");
    //alert(divs2.length);
    //alert(divs2[0].innerHTML);
    //alert(divs2[1].innerHTML);

    // 5.奇数选择器       $("A:odd");
    let divs3 = $("div:odd");
```

```
    //alert(divs3.length);
    //alert(divs3[0].innerHTML);
    //alert(divs3[1].innerHTML);

    // 6.等于索引选择器      $("A:eq(index)");
    let div3 = $("div:eq(2)");
    //alert(div3.html());

    // 7.大于索引选择器      $("A:gt(index)");
    let divs4 = $("div:gt(1)");
    //alert(divs4.length);
    //alert(divs4[0].innerHTML);
    //alert(divs4[1].innerHTML);

    // 8.小于索引选择器      $("A:lt(index)");
    let divs5 = $("div:lt(2)");
    alert(divs5.length);
    alert(divs5[0].innerHTML);
    alert(divs5[1].innerHTML);

</script>
</html>
```

3.5、表单属性选择器

选择器	语法	作用
可用元素选择器	$("A:enabled");	获得可用元素
不可用元素选择器	$("A:disabled");	获得不可用元素
单选/复选框被选中的元素	$("A:checked");	获得单选/复选框选中的元素
下拉框被选中的元素	$("A:selected");	获得下拉框选中的元素

- **代码实现**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>表单属性选择器</title>
</head>
<body>
    <input type="text" disabled>
    <input type="text" >
    <input type="radio" name="gender" value="men" checked>男
    <input type="radio" name="gender" value="women">女
    <input type="checkbox" name="hobby" value="study" checked>学习
    <input type="checkbox" name="hobby" value="sleep" checked>睡觉
    <select>
        <option>---请选择---</option>
        <option selected>本科</option>
        <option>专科</option>
```

```
    </select>
</body>
<script src="js/jquery-3.3.1.min.js"></script>
<script>
    // 1.可用元素选择器  $("A:enabled");
    let ins1 = $("input:enabled");
    //alert(ins1.length);

    // 2.不可用元素选择器  $("A:disabled");
    let ins2 = $("input:disabled");
    //alert(ins2.length);

    // 3.单选/复选框被选中的元素  $("A:checked");
    let ins3 = $("input:checked");
    //alert(ins3.length);
    //alert(ins3[0].value);
    //alert(ins3[1].value);
    //alert(ins3[2].value);

    // 4.下拉框被选中的元素   $("A:selected");
    let select = $("select option:selected");
    alert(select.html());

</script>
</html>
```

3.6、小结

- 选择器：类似于 CSS 的选择器，可以帮助我们获取元素。
- jQuery 中选择器的语法：$();
- 基本选择器
  - $("元素的名称");
  - $("#id的属性值");
  - $(".class的属性值");
- 层级选择器
  - $("A B");
  - $("A > B");
- 属性选择器
  - $("A[属性名]");
  - $("A[属性名=属性值]");
- 过滤器选择器
  - $("A:even");
  - $("A:odd");
- 表单属性选择器
  - $("A:disabled");
  - $("A:checked");
  - $("A:selected");

4、JQuery DOM

4.1、操作文本

- **常用方法**

方法	作用
html()	获取标签的文本
html(value)	设置标签的文本内容，解析标签

- **代码实现**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>操作文本</title>
</head>
<body>
    <div id="div">我是div</div>
    <input type="button" id="btn1" value="获取div的文本">
    <input type="button" id="btn2" value="设置div的文本">
</body>
<script src="js/jquery-3.3.1.min.js"></script>
<script>
     //1. html()    获取标签的文本内容
     $("#btn1").click(function(){
         //获取div标签的文本内容
         let value = $("#div").html();
         alert(value);
     });

     //2. html(value)    设置标签的文本内容，解析标签
     $("#btn2").click(function(){
         //设置div标签的文本内容
         //$("#div").html("我真的是div");
         $("#div").html("<b>我真的是div</b>");
     });
</script>
</html>
```

4.2、操作对象

- **常用方法**

方法	作用
$("元素")	创建指定元素
append(element)	添加成最后一个子元素，由添加者对象调用
appendTo(element)	添加成最后一个子元素，由被添加者对象调用
prepend(element)	添加成第一个子元素，由添加者对象调用
prependTo(element)	添加成第一个子元素，由被添加者对象调用
before(element)	添加到当前元素的前面，两者之间是兄弟关系，由添加者对象调用
after(element)	添加到当前元素的后面，两者之间是兄弟关系，由添加者对象调用
remove()	删除指定元素(自己移除自己)
empty()	清空指定元素的所有子元素

- **代码实现**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>操作对象</title>
</head>
<body>
    <div id="div"></div>
    <input type="button" id="btn1" value="添加一个span到div"> <br><br><br>

    <input type="button" id="btn2" value="将加油添加到城市列表最下方">
   
    <input type="button" id="btn3" value="将加油添加到城市列表最上方">
   
    <input type="button" id="btn4" value="将雄起添加到上海下方">
   
    <input type="button" id="btn5" value="将雄起添加到上海上方">
   
    <ul id="city">
        <li id="bj">北京</li>
        <li id="sh">上海</li>
        <li id="gz">广州</li>
        <li id="sz">深圳</li>
    </ul>
    <ul id="desc">
        <li id="jy">加油</li>
        <li id="xq">雄起</li>
    </ul>  <br><br><br>
    <input type="button" id="btn6" value="将雄起删除">    
    <input type="button" id="btn7" value="将描述列表全部删除">
   
</body>
<script src="js/jquery-3.3.1.min.js"></script>
<script>
    /*
        1. $("元素")    创建指定元素
        2. append(element)    添加成最后一个子元素，由添加者对象调用
        3. appendTo(element) 添加成最后一个子元素，由被添加者对象调用
        4. prepend(element)   添加成第一个子元素，由添加者对象调用
```

```
        5. prependTo(element) 添加成第一个子元素，由被添加者对象调用
        6. before(element)    添加到当前元素的前面，两者之间是兄弟关系，由添加者对象调用
        7. after(element)     添加到当前元素的后面，两者之间是兄弟关系，由添加者对象调用
        8. remove()           删除指定元素(自己移除自己)
        9. empty()            清空指定元素的所有子元素
    */

    // 按钮一：添加一个span到div
    $("#btn1").click(function(){
        let span = $("<span>span</span>");
        $("#div").append(span);
    });


    //按钮二：将加油添加到城市列表最下方
    $("#btn2").click(function(){
        //$("#city").append($("#jy"));
        $("#jy").appendTo($("#city"));
    });

    //按钮三：将加油添加到城市列表最上方
    $("#btn3").click(function(){
        //$("#city").prepend($("#jy"));
        $("#jy").prependTo($("#city"));
    });


    //按钮四：将雄起添加到上海下方
    $("#btn4").click(function(){
        $("#sh").after($("#xq"));
    });


    //按钮五：将雄起添加到上海上方
    $("#btn5").click(function(){
        $("#sh").before($("#xq"));
    });

    //按钮六：将雄起删除
    $("#btn6").click(function(){
        $("#xq").remove();
    });


    //按钮七：将描述列表全部删除
    $("#btn7").click(function(){
        $("#desc").empty();
    });

</script>
</html>
```

4.3、操作样式

- **常用方法**

方法	作用
css(name)	根据样式名称获取css样式
css(name,value)	设置CSS样式
addClass(value)	给指定的对象添加样式类名
removeClass(value)	给指定的对象删除样式类名
toggleClass(value)	如果没有样式类名，则添加。如果有，则删除

- **代码实现**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>操作样式</title>
    <style>
        .cls1{
            background: pink;
            height: 30px;
        }
    </style>
</head>
<body>
    <div style="border: 1px solid red;" id="div">我是div</div>
    <input type="button" id="btn1" value="获取div的样式">   
    <input type="button" id="btn2" value="设置div的背景色为蓝色">  
    <br><br><br>
    <input type="button" id="btn3" value="给div设置cls1样式">   
    <input type="button" id="btn4" value="给div删除cls1样式">   
    <input type="button" id="btn5" value="给div设置或删除cls1样式">
  
</body>
<script src="js/jquery-3.3.1.min.js"></script>
<script>
    // 1.css(name)   获取css样式
    $("#btn1").click(function(){
        alert($("#div").css("border"));
    });

    // 2.css(name,value)   设置CSS样式
    $("#btn2").click(function(){
        $("#div").css("background","blue");
    });

    // 3.addClass(value)   给指定的对象添加样式类名
    $("#btn3").click(function(){
        $("#div").addClass("cls1");
    });

    // 4.removeClass(value)  给指定的对象删除样式类名
```

```
    $("#btn4").click(function(){
        $("#div").removeClass("cls1");
    });

    // 5.toggleClass(value)  如果没有样式类名，则添加。如果有，则删除
    $("#btn5").click(function(){
        $("#div").toggleClass("cls1");
    });

</script>
</html>
```

4.4、操作属性

- **常用方法**

方法	作用
attr(name,[value])	获得/设置属性的值
prop(name,[value])	获得/设置属性的值(checked，selected)

- **代码实现**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>操作属性</title>
</head>
<body>
    <input type="text" id="username">
    <br>
    <input type="button" id="btn1" value="获取输入框的id属性">    
    <input type="button" id="btn2" value="给输入框设置value属性">
    <br><br>

    <input type="radio" id="gender1" name="gender">男
    <input type="radio" id="gender2" name="gender">女
    <br>
    <input type="button" id="btn3" value="选中女">
    <br><br>

    <select>
        <option>---请选择---</option>
        <option id="bk">本科</option>
        <option id="zk">专科</option>
    </select>
    <br>
    <input type="button" id="btn4" value="选中本科">
</body>
<script src="js/jquery-3.3.1.min.js"></script>
<script>
    // 1.attr(name,[value])   获得/设置属性的值
    //按钮一：获取输入框的id属性
```

```
    $("#btn1").click(function(){
        alert($("#username").attr("id"));
    });

    //按钮二：给输入框设置value属性
    $("#btn2").click(function(){
        $("#username").attr("value","hello...");
    });


    // 2.prop(name,[value])    获得/设置属性的值(checked，selected)
    //按钮三：选中女
    $("#btn3").click(function(){
        $("#gender2").prop("checked",true);
    });

    //按钮四：选中本科
    $("#btn4").click(function(){
        $("#bk").prop("selected",true);
    });
</script>
</html>
```

4.5、小结

- 操作文本
  - html() html(...)：获取或设置标签的文本，解析标签。
- 操作对象
  - $(“元素”)：创建指定元素。
  - append(element)：添加成最后一个子元素，由添加者对象调用。
  - prepend(element)：添加成第一个子元素，由添加者对象调用。
  - before(element)：添加到当前元素的前面，两者之间是兄弟关系，由添加者对象调用。
  - after(element)：添加到当前元素的后面，两者之间是兄弟关系，由添加者对象调用。
  - remove()：删除指定元素(自己移除自己)。
- 操作样式
  - addClass(value)：给指定的对象添加样式类名。
  - removeClass(value)：给指定的对象删除样式类名。
- 操作属性
  - attr(name,[value])：获得/设置属性的值。
  - prop(name,[value])：获得/设置属性的值(checked，selected)。

5、综合案例 复选框

5.1、案例效果

全选 全不选 反选	分类ID	分类名称	分类描述	操作
☐	1	手机数码	手机数码类商品	修改|删除
☐	2	电脑办公	电脑办公类商品	修改|删除
☐	3	鞋靴箱包	鞋靴箱包类商品	修改|删除
☐	4	家居饰品	家居饰品类商品	修改|删除

5.2、分析和实现

功能分析

- 全选
  - 1. 为全选按钮绑定单击事件。
    2. 获取所有的商品项复选框元素，为其添加 checked 属性，属性值为 true。
- 全不选
  - 1. 为全不选按钮绑定单击事件。
    2. 获取所有的商品项复选框元素，为其添加 checked 属性，属性值为 false。
- 反选
  - 1. 为反选按钮绑定单击事件
    2. 获取所有的商品项复选框元素，为其添加 checked 属性，属性值是目前相反的状态。

代码实现

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>复选框</title>
</head>
<body>
    <table id="tab1" border="1" width="800" align="center">
        <tr>
            <th style="text-align: left">
                <input style="background:lightgreen" id="selectAll"
type="button" value="全选">
                <input style="background:lightgreen" id="selectNone"
type="button" value="全不选">
                <input style="background:lightgreen" id="reverse" type="button"
value="反选">
            </th>

            <th>分类ID</th>
            <th>分类名称</th>
            <th>分类描述</th>
            <th>操作</th>
        </tr>
        <tr>
            <td><input type="checkbox" class="item"></td>
            <td>1</td>
            <td>手机数码</td>
            <td>手机数码类商品</td>
            <td><a href="">修改</a>|<a href="">删除</a></td>
        </tr>
        <tr>
            <td><input type="checkbox" class="item"></td>
            <td>2</td>
            <td>电脑办公</td>
            <td>电脑办公类商品</td>
            <td><a href="">修改</a>|<a href="">删除</a></td>
        </tr>
        <tr>
```

```
            <td><input type="checkbox" class="item"></td>
            <td>3</td>
            <td>鞋靴箱包</td>
            <td>鞋靴箱包类商品</td>
            <td><a href="">修改</a>|<a href="">删除</a></td>
        </tr>
        <tr>
            <td><input type="checkbox" class="item"></td>
            <td>4</td>
            <td>家居饰品</td>
            <td>家居饰品类商品</td>
            <td><a href="">修改</a>|<a href="">删除</a></td>
        </tr>
    </table>
</body>
<script src="js/jquery-3.3.1.min.js"></script>
<script>
    //全选
    //1.为全选按钮添加单击事件
    $("#selectAll").click(function(){
        //2.获取所有的商品复选框元素，为其添加checked属性，属性值true
        $(".item").prop("checked",true);
    });


    //全不选
    //1.为全不选按钮添加单击事件
    $("#selectNone").click(function(){
        //2.获取所有的商品复选框元素，为其添加checked属性，属性值false
        $(".item").prop("checked",false);
    });


    //反选
    //1.为反选按钮添加单击事件
    $("#reverse").click(function(){
        //2.获取所有的商品复选框元素，为其添加checked属性，属性值是目前相反的状态
        let items = $(".item");
        items.each(function(){
            $(this).prop("checked",!$(this).prop("checked"));
        });
    });
</script>
</html>
```

6、综合案例 随机图片

6.1、案例效果

开始 停止

6.2、动态切换小图的分析和实现

- **功能分析**
  1. 准备一个数组
  2. 定义计数器
  3. 定义定时器对象
  4. 定义图片路径变量
  5. 为开始按钮绑定单击事件
  6. 设置按钮状态
  7. 设置定时器，循环显示图片
  8. 循环获取图片路径
  9. 将当前图片显示到小图片上
  10. 计数器自增
- **代码实现**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>随机图片</title>
</head>
<body>
<!-- 小图 -->
<div style="background-color:red;border: dotted; height: 50px; width: 50px">
    <img src="img/01.jpg" id="small" style="width: 50px; height: 50px;">
</div>
<!-- 大图 -->
<div style="border: double ;width: 400px; height: 400px; position: absolute;
left: 500px; top:10px">
    <img src="" id="big" style="width: 400px; height: 400px; display:none;">
</div>

<!-- 开始和结束按钮 -->
<input id="startBtn" type="button" style="width: 150px;height: 150px; font-
size: 20px" value="开始">
<input id="stopBtn" type="button" style="width: 150px;height: 150px; font-
size: 20px" value="停止">
</body>
```

```
<script src="js/jquery-3.3.1.min.js"></script>
<script>
    //1.准备一个数组
    let imgs = [
        "img/01.jpg",
        "img/02.jpg",
        "img/03.jpg",
        "img/04.jpg",
        "img/05.jpg",
        "img/06.jpg",
        "img/07.jpg",
        "img/08.jpg",
        "img/09.jpg",
        "img/10.jpg"];

    //2.定义计数器变量
    let count = 0;

    //3.声明定时器对象
    let time = null;

    //4.声明图片路径变量
    let imgSrc = "";

    //5.为开始按钮绑定单击事件
    $("#startBtn").click(function(){
        //6.设置按钮状态
        //禁用开始按钮
        $("#startBtn").prop("disabled",true);
        //启用停止按钮
        $("#stopBtn").prop("disabled",false);

        //7.设置定时器，循环显示图片
        time = setInterval(function(){
            //8.循环获取图片路径
            let index = count % imgs.length; // 0%10=0  1%10=1  2%10=2 ..
9%10=9  10%10=0

            //9.将当前图片显示到小图片上
            imgSrc = imgs[index];
            $("#small").prop("src",imgSrc);

            //10.计数器自增
            count++;
        },10);
    });
</script>
</html>
```

6.3、显示大图的分析和实现

- **功能分析**
    1. 为停止按钮绑定单击事件
    2. 取消定时器
    3. 设置按钮状态
    4. 将图片显示到大图片上

- **代码实现**

```
//11.为停止按钮绑定单击事件
$("#stopBtn").click(function(){
    //12.取消定时器
    clearInterval(time);

    //13.设置按钮状态
    //启用开始按钮
    $("#startBtn").prop("disabled",false);
    //禁用停止按钮
    $("#stopBtn").prop("disabled",true);

    //14.将图片显示到大图片上
    $("#big").prop("src",imgSrc);
    $("#big").prop("style","width: 400px; height: 400px;");
});
```